Guía de Operaciones de OmniCRM

OmniCRM es la solución integral BSS/OSS de Omnitouch para proveedores de servicios móviles y de línea fija. Una plataforma completa que maneja todo, desde la incorporación de clientes hasta la facturación, aprovisionamiento y soporte, todo en un sistema integrado.

Comenzando

Para el Personal de Atención al Cliente

Tus primeros pasos:

1. **Aprender la interfaz** - Familiarízate con la navegación y búsqueda
2. **Crear un cliente** - Creación de clientes paso a paso
3. **Agregar un servicio** - Aprovisiona tu primer servicio
4. **Procesar un pago** - Maneja los pagos de los clientes
5. **Recargar servicios** - Agrega crédito a las cuentas de los clientes

Tareas diarias:

- **Gestión de servicios** - Administra los servicios de los clientes
- **Ver uso** - Verifica saldos y uso
- **Ver registros de llamadas** - Inspecciona registros de llamadas, SMS y datos (CDRs)
- **Modificar servicios** - Cambia configuraciones de servicios
- **Generar facturas** - Crea y envía facturas

Para Administradores del Sistema

Configuración y configuración:

1. **Entender la arquitectura** - Visión general del sistema
2. **Configurar el sistema** - Configuraciones del sistema
3. **Crear productos** - Construye tu catálogo
4. **Escribir playbooks** - Automatiza el aprovisionamiento
5. **Configurar usuarios** - Crea cuentas y asigna roles

Temas avanzados:

- **Gestión de inventario** - Administra activos
- **Integración de API** - Conecta sistemas externos
- **Configuración de seguridad** - Configura 2FA y permisos

Para Clientes

Uso del Portal de Autocuidado:

- **Acceder a tu cuenta** - Inicia sesión y navega
- **Ver servicios** - Consulta tus servicios activos
- **Ver uso** - Monitorea datos y saldos
- **Pagar facturas** - Realiza pagos en línea
- **Recargar servicios** - Agrega crédito al instante

Referencia Rápida de Tareas

Quiero...	Documentación
Crear un nuevo cliente	Crear Cliente
Agregar un servicio a un cliente	Agregar Servicio
Ver uso y saldos de servicio	Uso de Servicio
Ver registros de llamadas/SMS/datos de un servicio	Registros de Detalle de Llamadas
Procesar un pago	Procesar Pago
Generar una factura	Gestión de Facturas
Recargar un servicio	Recarga y Recarga
Gestionar inventario	Sistema de Inventario
Crear un producto	Ciclo de Vida del Producto
Escribir un playbook de aprovisionamiento	Playbooks de Ansible
Configurar cuentas de usuario	RBAC
Habilitar 2FA	Autenticación de Dos Factores
Buscar cualquier cosa	Búsqueda Global
Entender el sistema	Arquitectura del Sistema
Usar la API	Documentación de API

Biblioteca de Documentación

Lectura Esencial

- **Arquitectura del Sistema** - Visión técnica completa con diagramas
- **Guía del Ciclo de Vida del Producto** - Gestión de productos de principio a fin
- **Guía de Playbooks de Ansible** - Domina la automatización del aprovisionamiento

Gestión de Clientes

- Clientes, Contactos y Sitios - Modelo de datos y relaciones
- Creación de Clientes - Guía paso a paso
- Etiquetas de Clientes - Organiza con etiquetas
- Atributos de Clientes - Metadatos personalizados
- Registro de Actividades - Rastrea todos los cambios

Soporte

- Sistema de Tickets de Soporte - Gestiona tickets de soporte al cliente
- Base de Conocimientos FAQ - Respuestas de autoservicio para problemas comunes

Gestión del Trabajo

- Tableros - Gestión de trabajo estilo Kanban

Operaciones de Servicio

- Gestión de Servicios - Visión general
- Adición de Servicios - Flujo de trabajo de aprovisionamiento
- Asignación de Planes - Asignación de planes
- Modificación de Servicios - Realizando cambios
- Uso y Saldos de Servicio - Monitorea el uso

- Registros de Detalle de Llamadas (CDRs) - Ver registros de llamadas, SMS y datos
- Recarga y Recarga - Agregar créditos

Facturación y Finanzas

- Visión General de Facturación - Guía completa de facturación
- Guía del Sistema de Pagos - Arquitectura del sistema de pagos
- Métodos de Pago - Gestión de métodos de pago
- Transacciones - Cargos y créditos
- Procesamiento de Pagos - Flujos de trabajo de pagos
- Gestión de Facturas - Generar y gestionar facturas

Administración

- Configuración del Sistema - Configura OmniCRM
- Gestión de Inventario - Seguimiento de activos
- Aprovisionamiento de Tarjetas SIM - Gestión del ciclo de vida de SIM
- Gestión de Inventario de eSIM - Seguimiento y asignación de eSIM
- Claves de API - Gestión de acceso a la API
- Monitoreo y Métricas - Salud y observabilidad del sistema

Seguridad y Acceso

- Flujos de Autenticación - Cómo funciona la autenticación
- Autenticación de Dos Factores - Configura 2FA
- Control de Acceso Basado en Roles - Usuarios, roles y permisos

Integraciones

- Campañas de Email y SMS (EDM) - Campañas de comunicación masiva
- Integración de Mailjet - Servicio de correo electrónico
- API de Notificaciones - Integración de notificaciones salientes
- Documentación de API - Referencia de API REST

Orientado al Cliente

- Portal de Autocuidado - Guía del portal para clientes
- Atención al Cliente - Funciones de autoservicio
- Sistema de Difusión Celular - Alertas de emergencia

Referencia

- Registro de Cambios - Historial de versiones
- Glosario de Términos - Referencia de terminología

¿Qué Hace Especial a OmniCRM?

OmniCRM reúne todas las herramientas que necesitas para gestionar un negocio de telecomunicaciones moderno:

Aprovisionamiento de Servicios Automatizado

OmniCRM elimina la configuración manual. Utiliza **automatización de Ansible** para aprovisionar servicios en segundos. El sistema maneja el trabajo automáticamente cuando activas una tarjeta SIM, configuras el equipo del cliente o configuras un paquete complejo.

Lo que hace:

- Aprovisiona servicios con un solo clic
- Configura equipos de red automáticamente
- Crea cuentas de facturación en tiempo real
- Envía correos electrónicos de bienvenida y notificaciones SMS
- Revierte automáticamente si algo falla

Aprende más sobre Aprovisionamiento | **Ver Guía de Playbooks de Ansible**

Facturación y Pagos Inteligentes

La integración incorporada con **CGRateS** proporciona tarifas y cargos en tiempo real, mientras que Stripe maneja el procesamiento de pagos sin problemas.

Lo que hace:

- Seguimiento y tarificación de uso en tiempo real
- Generación automática de facturas
- Procesamiento de tarjetas de crédito a través de Stripe
- Modelos de facturación prepago y postpago
- Precios flexibles y códigos promocionales
- Cálculo e informes de impuestos

Explora Funciones de Facturación | **Procesamiento de Pagos** | **Gestión de Facturas**

Gestión Completa de Clientes

Gestiona clientes, contactos, múltiples sitios y servicios con un modelo de relación integral.

Lo que hace:

- Rastrea clientes, contactos y ubicaciones de servicio
- Administra múltiples servicios por cliente
- Almacena atributos y metadatos personalizados
- Etiqueta clientes para organización
- Historial de actividades completo y registros de auditoría
- Integración de Google Maps para geocodificación de sitios

Guía de Gestión de Clientes | **Crear un Cliente** | **Etiquetas de Clientes**

Gestión de Inventario

Rastrea y gestiona todos tus activos físicos y virtuales, desde tarjetas SIM hasta enrutadores y bloques de direcciones IP.

Lo que hace:

- Rastrea tarjetas SIM, equipos y accesorios
- Gestiona números de teléfono y bloques de direcciones IP
- Asignación automatizada durante el aprovisionamiento
- Capacidades de importación y exportación por lotes
- Almacenamiento de configuraciones de equipos
- Plantillas de inventario personalizadas

Documentación del Sistema de Inventario

Portal de Autocuidado para Clientes

Empodera a tus clientes con un **portal de autoservicio** donde pueden gestionar sus propios servicios.

Lo que los clientes pueden hacer:

- Ver servicios y uso
- Verificar saldos y asignaciones de datos
- Pagar facturas en línea
- Descargar facturas como PDFs
- Actualizar información de contacto
- **Recargar sus servicios** al instante

Guía del Portal de Autocuidado | **Funciones de Atención al Cliente**

Seguridad Empresarial

Construido con la seguridad en su núcleo, con autenticación y autorización completas.

Características de seguridad:

- **Autenticación basada en JWT**
- **Autenticación de dos factores (2FA)** con TOTP y códigos de respaldo
- **Control de acceso basado en roles (RBAC)** con permisos granulares
- Verificación de correo electrónico para cambios de cuenta
- Registro de auditoría completo a través del registro de actividades
- Gestión de sesiones y controles de tiempo de espera

Guía de Autenticación | **Documentación de RBAC** | **Configuración de 2FA**

Integraciones Potentes

OmniCRM se integra sin problemas con servicios líderes en la industria:

- **CGRateS** - Motor de facturación y tarificación en tiempo real para cargos de grado telecomunicaciones
- **Stripe** - Procesamiento de pagos seguro y almacenamiento de tarjetas
- **Mailjet** - Entrega de correo electrónico profesional con plantillas
- **Google Maps** - Validación de direcciones y geocodificación para ubicaciones de sitios precisas
- **API RESTful** - Integra con tus sistemas existentes

Documentación de API | **Integración de Mailjet**

Catálogo de Productos Flexible

Crea cualquier tipo de oferta de producto, desde servicios independientes simples hasta paquetes complejos con múltiples componentes.

Tipos de productos soportados:

- **Independientes** - Servicios únicos (planes móviles, paquetes de internet)
- **Paquetes** - Ofertas combinadas (internet + TV + teléfono)
- **Complementos** - Servicios suplementarios (recargas de datos, llamadas internacionales)
- **Promociones** - Ofertas especiales y descuentos

Cada producto puede tener su propia automatización de aprovisionamiento, reglas de precios y lógica comercial.

Guía del Ciclo de Vida del Producto | **Conceptos de Productos y Servicios**

Sistema de Difusión de Emergencia

Para operadores móviles, OmniCRM incluye un **Sistema de Difusión Celular** para alertas de seguridad pública.

Lo que hace:

- Envía alertas de emergencia a áreas geográficas
- Soporte de mensajes multilingües
- Segmentación por operador de red móvil
- Cumplimiento con estándares de alertas gubernamentales

Documentación de Difusión Celular

Búsqueda y Navegación Poderosas

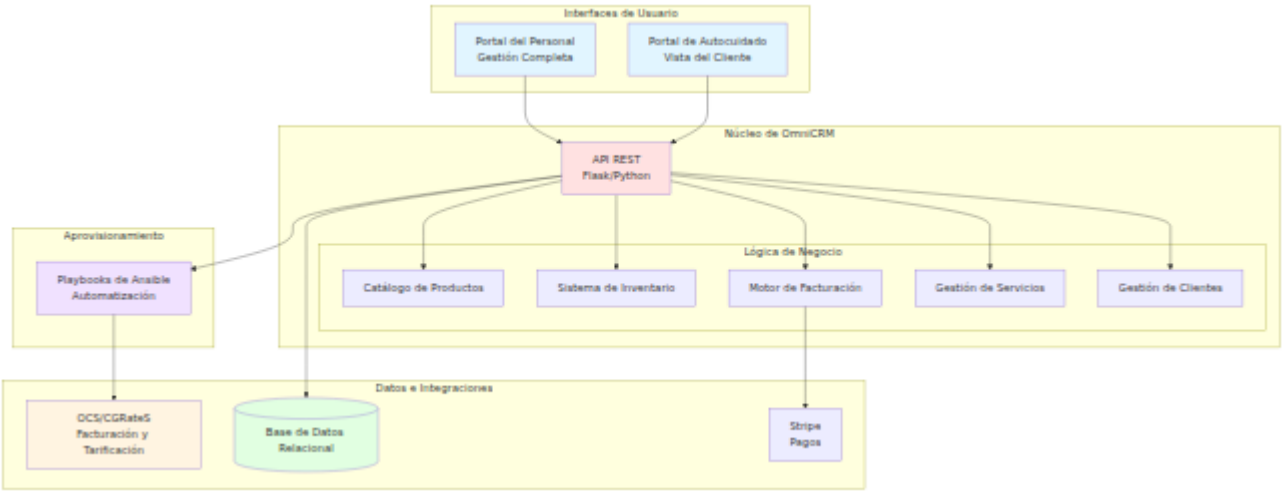
Encuentra cualquier cosa al instante con **búsqueda global** a través de clientes, servicios, facturas y más.

Características de búsqueda:

- Busca en todas las entidades
- Filtra y ordena resultados
- Accesos directos de navegación rápida
- Sugerencias inteligentes

Guía de Navegación | **Búsqueda Global**

Arquitectura del Sistema



Ver Documentación Completa de Arquitectura

Conceptos Clave

Entendiendo Productos vs Servicios

Este es uno de los conceptos más importantes en OmniCRM:

- **Producto** = Una plantilla u oferta en tu catálogo (por ejemplo, "Plan Móvil Ilimitado")
- **Servicio** = Una instancia activa de un producto para un cliente específico (por ejemplo, "Plan Móvil Ilimitado de John Smith")

Cuando aprovisionas un producto, el sistema utiliza **automatización de Ansible** para crear el/los servicio(s) real(es). Un producto puede crear múltiples servicios, ningún servicio (solo configuración) o modificar servicios existentes.

Aprende más sobre Productos y Servicios

La Magia del Aprovisionamiento

La salsa secreta de OmniCRM es su **sistema de aprovisionamiento automatizado**:



El playbook maneja todo, desde la creación de cuentas en CGRateS hasta la configuración de enrutadores y el envío de correos electrónicos de bienvenida. Y si algo falla, revierte automáticamente.

Profundiza: Playbooks de Ansible | **Sistema de Aprovisionamiento**

Facturación Hecha Simple



Cada servicio genera **transacciones** (cargos o créditos). Estas se agrupan en **facturas** y se envían a los clientes. Con la integración de Stripe, los pagos pueden procesarse automáticamente.

Visión General de Facturación | **Procesamiento de Pagos**

Jerarquía de Clientes

```
Cliente (Compañía ABC)
├── Contactos (John Smith - Facturación, Jane Doe - Técnico)
├── Sitios (Oficina Principal, Oficina Sucursal)
└── Servicios
    ├── Internet - Oficina Principal
    ├── Plan Móvil - John Smith
    └── VoIP - Oficina Sucursal
```

Todo está organizado jerárquicamente, lo que facilita la gestión de clientes con múltiples ubicaciones y servicios.

Clientes, Contactos y Sitios

¿Por Qué Elegir OmniCRM?

Plataforma Todo en Uno

Todo lo que necesitas en un solo sistema integrado: no hay necesidad de manejar múltiples herramientas o proveedores.

Automatización Primero

El aprovisionamiento impulsado por Ansible significa que los servicios se despliegan en segundos, no en horas. Reduce errores, ahorra tiempo y aumenta la satisfacción del cliente.

Flexible y Personalizable

Desde definiciones de productos hasta playbooks de aprovisionamiento y atributos personalizados: adapta OmniCRM para que coincida con tus procesos comerciales exactos.

Diseñado para Telecomunicaciones

Diseñado específicamente para proveedores de servicios con características como integración con CGRateS, gestión de inventario y tarificación en tiempo real.

Seguridad Empresarial

Autenticación JWT, 2FA, RBAC y registro de auditoría completo aseguran que tus datos estén seguros y cumplan con las normativas.

Impulsado por API

Una API REST completa significa que puedes integrar OmniCRM con cualquier sistema existente o construir herramientas personalizadas.

Obtén Soporte

Recursos de Documentación

- Comienza con **Arquitectura del Sistema** para una visión técnica
- Consulta las **guías de inicio** para la incorporación específica por rol
- Revisa la **Referencia Rápida de Tareas** para operaciones comunes
- Consulta el glosario para terminología

¿Necesitas Más Ayuda?

Explora la biblioteca de documentación completa arriba o utiliza la **búsqueda global** para encontrar temas específicos.

OmniCRM - BSS/OSS Completo para Proveedores de Servicios Modernos

Última Actualización: 2025-12-23

Autenticación de Dos Factores (2FA)

Seguridad Mejorada con Contraseñas de Un Solo Uso Basadas en Tiempo

OmniCRM soporta **autenticación de dos factores (2FA)** utilizando contraseñas de un solo uso basadas en tiempo (TOTP). Esto añade una capa extra de seguridad al requerir que los usuarios proporcionen tanto su contraseña como un código sensible al tiempo de una aplicación de autenticación.

Consulta también: RBAC para permisos de gestión de 2FA, Flujos de Autenticación para detalles del proceso de inicio de sesión.

Propósito

2FA proporciona:

1. **Seguridad Mejorada**: Protege las cuentas incluso si las contraseñas son comprometidas.
2. **Cumplimiento**: Cumple con los requisitos de seguridad para industrias reguladas.
3. **Elección del Usuario**: Opcional para los usuarios, puede ser impuesto por rol o globalmente.
4. **Estándar de la Industria**: Utiliza el protocolo TOTP compatible con Google Authenticator, Authy, Microsoft Authenticator y otras aplicaciones estándar.

Cómo Funciona 2FA

Cuando 2FA está habilitado para un usuario:

1. **Configuración**: El usuario escanea un código QR con su aplicación de autenticación durante la inscripción.
2. **Inicio de Sesión**: Después de que el usuario ingresa su nombre de usuario y contraseña, el usuario proporciona el código de 6 dígitos de su aplicación.
3. **Verificación**: El sistema valida que el código basado en tiempo coincide con el valor esperado.
4. **Acceso**: El usuario obtiene acceso solo después de que ambos factores son verificados.

Habilitando 2FA

Para Usuarios Individuales

Los usuarios pueden habilitar 2FA para su propia cuenta:

1. Navegar a **Configuración del Usuario** o **Perfil**
2. Seleccionar **Habilitar Autenticación de Dos Factores**
3. Escanear el código QR con una aplicación de autenticación
4. Ingresar el código de verificación para confirmar la configuración
5. Guardar los códigos de respaldo en un lugar seguro

Respaldo y Recuperación

Códigos de Respaldo

Al habilitar 2FA, los usuarios reciben códigos de respaldo que pueden ser utilizados si su dispositivo no está disponible:

- Cada código es de un solo uso
- Almacenar códigos de forma segura fuera de línea
- Generar nuevos códigos si todos son utilizados

Reinicio de Administrador

Si un usuario pierde el acceso a su autenticador y todos los códigos de respaldo están agotados, un administrador con acceso a la base de datos puede restablecer manualmente el 2FA del usuario borrando el campo *totp_secret* en la base de datos. El usuario puede luego volver a habilitar 2FA.

Integración de API

Habilitar 2FA para un usuario

POST /2fa/enable/user/{user_id}

```
{
  "password": "current_password"
}
```

La respuesta incluye la URI de aprovisionamiento (para el código QR) y los códigos de respaldo.

Verificar la configuración de 2FA

POST /2fa/verify-setup/user/{user_id}

```
{
  "code": "123456"
}
```

Verificar 2FA durante el inicio de sesión

POST /2fa/verify/user/{user_id}

```
{
  "code": "123456"
}
```

Devuelve el token de acceso, el token de actualización y los datos del usuario tras una verificación exitosa.

Regenerar códigos de respaldo

POST /2fa/backup-codes/regenerate/user/{user_id}

Requiere autenticación. Devuelve un nuevo conjunto de códigos de respaldo.

Mejores Prácticas

- **Códigos de respaldo primero.** Siempre guarda los códigos de respaldo antes de completar la configuración de 2FA.
- **Educar a los usuarios.** Proporcionar instrucciones claras para la configuración y recuperación.
- **Proceso de reinicio seguro.** Verificar la identidad del usuario antes de restablecer manualmente 2FA en la base de datos.

Preguntas Frecuentes

¿Qué aplicaciones de autenticación son compatibles? Cualquier aplicación compatible con TOTP (Google Authenticator, Authy, Microsoft Authenticator, 1Password, etc.).

¿Qué hago si pierdo mi teléfono? Usa un código de respaldo o contacta a un administrador para restablecer 2FA.

¿Puedo usar SMS en lugar de una aplicación? Actualmente, solo se admiten aplicaciones de autenticación TOTP.

¿Es 2FA obligatorio? Depende de la política de tu organización. 2FA generalmente es requerido para roles de personal administrativo y de soporte, pero es opcional para usuarios clientes. El sistema no impone 2FA para cuentas de clientes (los usuarios en rol de cliente no ven los avisos de inscripción a 2FA).

¿Cuánto tiempo son válidos los códigos TOTP? Los códigos se actualizan cada 30 segundos y tienen una pequeña ventana de tiempo para validación (normalmente acepta códigos de la ventana de 30 segundos actual más las ventanas anterior/siguiente para tolerancia de desincronización del reloj).

Gestión de Claves API

La interfaz de Gestión de Claves API proporciona una **interfaz web** para crear, monitorear y gestionar claves API utilizadas para el acceso programático a la API de OmniCRM.

> ⓘ **NOTA**
>
> Para conceptos generales de autenticación API y ejemplos de uso, consulta `concepts_api`.

Descripción General

Las claves API permiten una **autenticación segura y de larga duración** para:

- Integraciones de servidor a servidor
- Scripts de automatización
- Aplicaciones de terceros
- Tareas programadas y trabajos cron
- Sistemas de monitoreo externos

A diferencia de los tokens JWT (que expiran después de minutos/horas), las claves API permanecen válidas hasta que se revoquen manualmente o hasta su fecha de expiración.

Accediendo a la Gestión de Claves API

Navega a:

O directamente:

Permiso Requerido: `MANAGE_API_KEYS` (rol de administrador)

Vista de Lista de Claves API

La página principal muestra todas las claves API en un formato de tabla:

Columnas:

- **Nombre** - Etiqueta descriptiva para la clave API (por ejemplo, "Sistema de Aprovisionamiento", "Herramienta de Monitoreo")
- **Creado Por** - Nombre de usuario de la persona que creó la clave
- **Clave API** - La cadena de clave real (parcialmente enmascarada por seguridad)
- **Estado** - Activa, Expirada o Revocada
- **Fecha de Creación** - Cuando se generó la clave
- **Fecha de Expiración** - Cuando la clave expirará automáticamente
- **Acciones** - Botones Editar, Eliminar, Regenerar

Ejemplo de Visualización:

Widgets del Tablero

En la parte superior de la página, se muestran estadísticas resumidas:

- **Total de Claves API** - Conteo de todas las claves API (activas e inactivas)
- **Claves Activas** - Claves actualmente válidas
- **Expirando Pronto** - Claves que expiran en los próximos 30 días
- **Claves Expiradas** - Claves que han pasado su fecha de expiración

Creando una Clave API

Paso 1: Haz clic en "Agregar Clave API"

Haz clic en el botón **+ Agregar** en la parte superior derecha de la lista de Claves API.

Paso 2: Completa los Detalles

Aparecerá un formulario modal solicitando:

> Nombre: _____
> (por ejemplo, "Sistema de Aprovisionamiento")
>
> Descripción: _____
> (Opcional - propósito de esta clave)
>
> Fecha de Expiración: [Selector de Fecha]
> (Opcional - dejar en blanco si no hay expiración)
>
> Permisos: ☐ Ver Clientes ☐ Crear Clientes ☐ Ver Servicios ☐ Crear Servicios ☐ Aprovisionamiento ☐ Ver Inventario ☑ Administrador (todos los permisos)
>
> [Cancelar] [Generar Clave]

Directrices de Campo:

Nombre (requerido)

- Identificador corto y descriptivo
- Ejemplos: "Sistema de Aprovisionamiento", "Integración de Facturación", "Monitoreo"
- Usado en registros de auditoría y mostrado en la lista

Descripción (opcional)

- Explicación más detallada
- Ejemplos: "Usado por el servidor de aprovisionamiento de Ansible", "Sincronización de facturación de terceros"
- Ayuda a futuros administradores a entender el propósito de la clave

Fecha de Expiración (opcional)

- Si está en blanco: La clave nunca expira (no recomendado)
- Si se establece: La clave se vuelve automáticamente inválida después de esta fecha
- Recomendado: Establecer una expiración por seguridad (90 días a 1 año)

Permisos

- Selecciona permisos específicos o marca "Administrador" para acceso completo
- Sigue el mismo modelo de permisos basado en roles que las cuentas de usuario
- **Mejor Práctica:** Asignar los permisos mínimos necesarios

Paso 3: Generar y Copiar Clave

Después de hacer clic en **"Generar Clave"**, el sistema muestra la nueva clave API creada:

> ⚠ ¡Copia esta clave ahora - no se mostrará de nuevo!
>
> sk_live_a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6q7r8s9t0
>
> [Copiar al Portapapeles]
>
> [Cerrar]

> **⚠ AVISO**
>
> **¡Guarda la clave API inmediatamente!**
>
> Una vez que cierres este diálogo, la clave completa no podrá ser recuperada de nuevo. Solo verás una versión enmascarada (`sk_live_...XYZ`) en la vista de lista.
>
> Si pierdes la clave, debes **regenerarla**, lo que invalida la clave antigua y puede romper integraciones existentes.

Paso 4: Configura Tu Aplicación

Usa la clave API en las solicitudes de tu aplicación:

```
curl -X GET "https://yourcrm.com/crm/customers" \
  -H "X-API-KEY: sk_live_a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6q7r8s9t0"
```

O en variables de entorno:

```
export
CRM_API_KEY="sk_live_a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6q7r8s9t0"
```

Gestión de Claves Existentes

Visualizando Detalles de la Clave

Haz clic en cualquier nombre de clave API para ver detalles completos:

- Nombre completo de la clave y descripción
- Marca de tiempo de creación
- Nombre de usuario del creador
- Permisos asociados
- Estadísticas de uso (si se implementan)
- Registros de acceso recientes

Regenerando una Clave API

Si una clave API está comprometida o se pierde, regénérala:

1. Haz clic en el ⋮ **(tres puntos)** en la columna de Acciones
2. Selecciona **"Regenerar Clave"**
3. Confirma la acción

> ⚠ **AVISO**
>
> **Regenerar invalida la clave antigua inmediatamente.**
>
> Cualquier aplicación que use la clave antigua dejará de funcionar.
> Actualiza todas las integraciones con la nueva clave antes de regenerar.

Qué Ocurre:

- La clave antigua es revocada
- Se genera una nueva clave con los mismos permisos
- Se muestra la nueva clave (cópiala inmediatamente)
- El nombre, la descripción y los permisos permanecen sin cambios

Revocando (Eliminando) una Clave API

Para eliminar permanentemente una clave API:

1. Haz clic en el **⋮ (tres puntos)** en la columna de Acciones
2. Selecciona **"Eliminar"**
3. Confirma la eliminación

Qué Ocurre:

- La clave es revocada inmediatamente
- Todas las solicitudes que usen esta clave devuelven `401 Unauthorized`
- La clave es eliminada de la base de datos
- **No se puede deshacer** - la clave no puede ser recuperada

Cuándo Revocar:

- La integración ya no es necesaria
- La clave ha sido comprometida
- El sistema que usa la clave ha sido desmantelado
- Reemplazando con una nueva clave con permisos diferentes

Editando Detalles de la Clave API

Para modificar los detalles de una clave API:

1. Haz clic en el **⋮ (tres puntos)** en la columna de Acciones
2. Selecciona **"Editar"**
3. Actualiza el nombre, la descripción, la expiración o los permisos
4. Haz clic en **"Guardar Cambios"**

Campos Editables:

- Nombre
- Descripción
- Fecha de expiración
- Permisos

No Editables:

- El valor de la clave en sí (usa Regenerar para cambiar)
- Fecha de creación
- Creado por usuario

Estado de la Clave API

Las claves API pueden tener varios estados:

Activa

- La clave es válida y puede ser utilizada
- Dentro de la fecha de expiración (o sin expiración establecida)
- No revocada manualmente
- Mostrada con una insignia verde

Expirando Pronto

- Activa pero expirará en los próximos 30 días
- Mostrada con una insignia naranja/de advertencia
- Considera rotar antes de la expiración

Expirada

- Pasada la fecha de expiración
- Ya no acepta autenticación
- Mostrada con una insignia roja
- Puede ser eliminada o la expiración extendida

Revocada

- Eliminada/deshabilitada manualmente
- Permanentemente inválida
- Ya no se muestra en la lista activa

Filtrado y Búsqueda

La lista de Claves API soporta:

Buscar:

Buscar por nombre, descripción o clave parcial:

Filtrar por Estado:

Desplegable de filtro para mostrar:

- Todas las Claves
- Solo Activas
- Expirando Pronto (próximos 30 días)
- Expiradas

Ordenar:

Haz clic en los encabezados de columna para ordenar por:

- Nombre
- Fecha de Creación
- Fecha de Expiración
- Creado Por

Mejores Prácticas de Seguridad

Generación de Claves API

- **Longitud:** Las claves deben tener al menos 32 caracteres (el sistema aplica esto)

- **Aleatoriedad:** Generadas utilizando generadores de números aleatorios criptográficamente seguros
- **Formato:** Típicamente prefijadas (por ejemplo, `sk_live_`) para identificación

Almacenamiento de Claves API

En el CRM:

- Las claves son encriptadas antes de su almacenamiento (como contraseñas)
- La clave completa solo se muestra una vez durante la creación
- La base de datos almacena el hash para verificación
- Incluso los administradores no pueden recuperar la clave completa más tarde

En Tu Aplicación:

- Almacena en variables de entorno, no en el código
- Usa sistemas de gestión de secretos (AWS Secrets Manager, HashiCorp Vault)
- Nunca comites claves en el control de versiones
- Rota las claves periódicamente (90-365 días)

Gestión de Permisos

- **Principio de Mínimo Privilegio** - Solo otorga los permisos necesarios
- Evita crear claves de administrador a menos que sea absolutamente necesario
- Usa claves separadas para diferentes sistemas/purposes
- Revisa los permisos regularmente

Monitoreo y Auditoría

- Monitorea el uso de claves API a través de registros de actividad
- Configura alertas para patrones de acceso inusuales

- Revisa las marcas de tiempo de "último uso" regularmente
- Elimina claves no utilizadas

Rotación de Claves

Establece una política de rotación de claves:

1. **Crea una nueva clave** con los mismos permisos
2. **Actualiza las aplicaciones** para usar la nueva clave
3. **Monitorea** para asegurar que la clave antigua ya no se use
4. **Revoca la clave antigua** después del período de gracia

Solución de Problemas

"401 Unauthorized" al usar la clave API

- **Causa:** Clave inválida, expirada o incorrecta
- **Solución:**
  - Verifica que la clave esté copiada correctamente (sin espacios extra)
  - Verifica el estado de la clave (Activa vs. Expirada)
  - Confirma que la clave tenga los permisos requeridos
  - Asegúrate de usar el encabezado `X-API-KEY` (no `Authorization`)

"Clave API no encontrada" después de la creación

- **Causa:** La clave puede haber sido creada pero no almacenada correctamente
- **Solución:**
  - Verifica la lista de claves API para la nueva entrada
  - Si falta, crea una nueva clave
  - Informa del problema al administrador

Clave API expirando pronto

- **Causa:** Fecha de expiración próxima (dentro de 30 días)
- **Solución:**

    - Crea una nueva clave con expiración extendida
    - Actualiza las aplicaciones para usar la nueva clave
    - Revoca la clave antigua después de la migración

No se puede eliminar la clave API

- **Causa:** Puede estar protegida o en uso
- **Solución:**
    - Asegúrate de tener permisos de administrador
    - Verifica si la clave está bloqueada/protegida
    - Contacta al administrador si el problema persiste

Puntos Finales de API (para Gestión Programática)

Las claves API también se pueden gestionar a través de la API (requiere permisos de administrador):

Listar Claves API

```
GET /crm/api-keys
Authorization: Bearer <admin-token>
```

Crear Clave API

```
POST /crm/api-keys
Authorization: Bearer <admin-token>
Content-Type: application/json

{
  "name": "Nueva Integración",
  "description": "Sincronización de facturación de terceros",
  "expiry_date": "2026-01-10",
  "permissions": ["view_customer", "view_service"]
}
```

Respuesta:

```
{
  "api_key_id": 123,
  "name": "Nueva Integración",
  "api_key": "sk_live_a1b2c3d4e5f6g7h8i9j0",
  "status": "active",
  "created": "2025-01-10T10:00:00Z",
  "expiry_date": "2026-01-10T23:59:59Z"
}
```

Revocar Clave API

```
DELETE /crm/api-keys/{api_key_id}
Authorization: Bearer <admin-token>
```

Actualizar Clave API

```
PATCH /crm/api-keys/{api_key_id}
Authorization: Bearer <admin-token>
Content-Type: application/json

{
  "name": "Nombre Actualizado",
  "expiry_date": "2026-12-31"
}
```

Casos de Uso Comunes

Caso de Uso 1: Integración del Sistema de Aprovisionamiento

Crea una clave API para tu servidor de aprovisionamiento de Ansible:

1. Navega a Claves API → Agregar
2. Nombre: "Servidor de Aprovisionamiento de Ansible"
3. Descripción: "Usado por la automatización de aprovisionamiento"

4. Permisos: Aprovisionamiento, Ver/Crear Servicios, Ver/Actualizar Inventario
5. Expiración: 1 año
6. Copia la clave y añádela a `crm_config.yaml` de Ansible

Caso de Uso 2: Integración de Facturación de Terceros

Crea una clave de solo lectura para la exportación de facturación:

1. Nombre: "Sincronización de Facturación - QuickBooks"
2. Permisos: Ver Clientes, Ver Transacciones, Ver Facturas
3. Expiración: 90 días (rotar trimestralmente)
4. Usar en el script de exportación programada

Caso de Uso 3: Monitoreo y Alertas

Crea una clave para la recolección de métricas de Prometheus/Grafana:

1. Nombre: "Monitoreo - Grafana"
2. Permisos: Ver Servicios, Ver Aprovisionamiento
3. Expiración: Nunca (el monitoreo necesita acceso continuo)
4. Configura en la fuente de datos de Grafana

Caso de Uso 4: API del Portal del Cliente

Crea una clave para el portal de autoservicio del cliente:

1. Nombre: "Backend del Portal del Cliente"
2. Permisos: Ver Propio Cliente, Ver Propios Servicios, Crear Pagos
3. Expiración: 180 días
4. Usar en las llamadas a la API del backend del portal

Documentación Relacionada

- `concepts_api` - Conceptos y ejemplos de autenticación API
- `rbac` - Control de acceso basado en roles y permisos
- `2fa` - Autenticación de dos factores para mayor seguridad

Atributos del Cliente

Los Atributos del Cliente son pares clave-valor flexibles que se pueden adjuntar a cualquier registro de cliente para almacenar metadatos personalizados, configuraciones, o información específica del negocio que no encaja en los campos estándar del cliente.

Para la categorización visual de clientes y enlaces clicables, consulte Atributos del Cliente. Para información básica del cliente, vea Clientes, Contactos, Sitios y Servicios.

A diferencia de los campos de base de datos fijos, los atributos permiten extender dinámicamente los registros de clientes sin modificar el esquema de la base de datos. Esto los hace ideales para almacenar datos específicos de implementación, parámetros de integración, o banderas de lógica comercial personalizada.

Propósito y Casos de Uso

Los casos de uso comunes para los Atributos del Cliente incluyen:

1. Datos de Integración

Almacene identificadores de sistemas externos o claves API específicas para este cliente:

- `external_crm_id` = "SF-12345" (ID de cliente de Salesforce)
- `legacy_system_id` = "OLD-CRM-789" (Referencia de migración)
- `hubspot_contact_id` = "12345678" (Integración con HubSpot)

2. Lógica Comercial Personalizada

Almacene banderas o configuraciones que controlen el comportamiento específico del cliente:

- `billing_method` = "quarterly" (Anular la facturación mensual predeterminada)

- `auto_provision` = "true" (Habilitar la provisión automática de servicios)
- `support_tier` = "premium" (Nivel de soporte personalizado)
- `credit_limit` = "10000" (Límite de crédito específico del cliente)

3. Datos de Cumplimiento y Regulatorios

Rastrear metadatos relacionados con el cumplimiento:

- `gdpr_consent_date` = "2025-01-01" (Consentimiento para el procesamiento de datos)
- `tax_exempt` = "true" (Estado de exención fiscal)
- `regulatory_entity` = "FCC-123456" (Identificador regulatorio)

4. Metadatos Operacionales

Almacene información operacional:

- `preferred_contact_method` = "email" (Preferencia de comunicación)
- `account_manager` = "<john.smith@company.com>" (Gerente de cuenta asignado)
- `onboarding_date` = "2025-01-15" (Seguimiento del ciclo de vida del cliente)
- `churn_risk_score` = "0.23" (Análisis predictivo)

5. Parámetros de Provisión

Almacene configuraciones específicas de provisión:

- `radius_username_format` = "email" (Formato RADIUS personalizado)
- `vlan_id` = "100" (Configuración de red)
- `ipv6_enabled` = "true" (Banderas de características)

Atributos vs. Campos Estándar

Use Atributos Cuando:

- Los datos son específicos de la implementación o varían según la instalación
- Los requisitos cambian con frecuencia
- Almacena metadatos específicos de integración
- Prototipando nuevas características antes de agregar campos de base de datos
- Los datos no necesitan consultas complejas o uniones

Use Campos Estándar Cuando:

- Los datos son fundamentales para el modelo de cliente (nombre, correo electrónico, dirección)
- Se requiere búsqueda, filtrado o informes frecuentes
- Los datos tienen restricciones de integridad referencial
- El rendimiento es crítico para consultas a gran escala

Gestión de Atributos a través de la UI

Visualización de Atributos del Cliente

Para ver los atributos de un cliente:

1. Navegue a la página de resumen del cliente
2. Haga clic en la pestaña **Atributos**
3. Verá una tabla de todos los atributos para este cliente, mostrando:
    - Nombre del Atributo (clave)
    - Valor del Atributo
    - Fecha de Creación
    - Fecha de Última Modificación

Creación de un Nuevo Atributo

Para crear un nuevo atributo para un cliente:

1. Navegue a la página de resumen del cliente
2. Haga clic en la pestaña **Atributos**
3. Haga clic en el botón **Agregar Atributo**
4. Complete los campos requeridos:
   - **Nombre del Atributo** (requerido): La clave/nombre para este atributo (por ejemplo, `external_crm_id`)
   - **Valor del Atributo** (requerido): El valor a almacenar (por ejemplo, `SF-12345`)
5. Haga clic en **Crear Atributo**

Convenciones de Nomenclatura:

- Use minúsculas con guiones bajos: `external_system_id` ✓
- Evite espacios: `external system id` ✗
- Mantenga los nombres descriptivos pero concisos
- Use nomenclatura consistente entre clientes para los mismos tipos de atributos

Edición de un Atributo

Para editar un atributo existente:

1. Navegue a la página de resumen del cliente
2. Haga clic en la pestaña **Atributos**
3. Encuentre el atributo que desea editar en la tabla
4. Haga clic en el botón **Editar** (lápiz)
5. Modifique el nombre o valor del atributo
6. Haga clic en **Actualizar Atributo**

> ⓘ **NOTA**
>
> Cambiar el nombre de un atributo crea un nuevo par clave-valor. Asegúrese de que esto no rompa las integraciones que dependen del nombre original del atributo.

Eliminación de un Atributo

Para eliminar un atributo:

1. Navegue a la página de resumen del cliente
2. Haga clic en la pestaña **Atributos**
3. Encuentre el atributo que desea eliminar en la tabla
4. Haga clic en el botón **Eliminar** (papelera)
5. Confirme la eliminación en el popup

> **⚠ AVISO**
> Eliminar atributos utilizados por integraciones, flujos de trabajo de provisión, o lógica de facturación puede causar fallos. Verifique las dependencias antes de la eliminación.

Referencia de Campo de Atributo

Integración API

Los atributos se pueden gestionar programáticamente a través de la API:

Crear o Actualizar un Atributo

Endpoint: `PUT /crm/attribute/`

Permiso Requerido: `create_customer_attribute`

Cuerpo de la Solicitud:

```
{
  "customer_id": 123,
  "attribute_name": "external_crm_id",
  "attribute_value": "SF-12345"
}
```

Respuesta:

```
{
  "attribute_id": 456,
  "customer_id": 123,
  "attribute_name": "external_crm_id",
  "attribute_value": "SF-12345",
  "created": "2025-01-04 10:30:00",
  "last_modified": "2025-01-04 10:30:00"
}
```

Actualizar un Atributo Existente

Endpoint: `PATCH /crm/attribute/attribute_id/{attribute_id}`

Permiso Requerido: `update_customer_attribute`

Cuerpo de la Solicitud:

```
{
  "attribute_value": "SF-54321"
}
```

Obtener Atributo por ID

Endpoint: `GET /crm/attribute/attribute_id/{attribute_id}`

Permiso Requerido: `view_customer_attribute`

Respuesta:

```
{
  "attribute_id": 456,
  "customer_id": 123,
  "attribute_name": "external_crm_id",
  "attribute_value": "SF-12345",
  "created": "2025-01-04 10:30:00",
  "last_modified": "2025-01-04 10:30:00"
}
```


Obtener Todos los Atributos por ID de Cliente

Endpoint: `GET /crm/attribute/customer_id/{customer_id}`

Permiso Requerido: `view_customer_attribute`

Respuesta:


```
[
  {
    "attribute_id": 456,
    "customer_id": 123,
    "attribute_name": "external_crm_id",
    "attribute_value": "SF-12345",
    "created": "2025-01-04 10:30:00",
    "last_modified": "2025-01-04 10:30:00"
  },
  {
    "attribute_id": 457,
    "customer_id": 123,
    "attribute_name": "billing_method",
    "attribute_value": "quarterly",
    "created": "2025-01-04 10:35:00",
    "last_modified": "2025-01-04 10:35:00"
  }
]
```


Eliminar un Atributo

Endpoint: `DELETE /crm/attribute/attribute_id/{attribute_id}`

Permiso Requerido: `delete_customer_attribute`

Respuesta:

```
{
  "result": "success"
}
```

Operaciones de Atributo en Masa

Gestión de Múltiples Atributos

Para establecer múltiples atributos para un cliente a la vez (por ejemplo, durante la incorporación o la sincronización de integración):

```
import requests

customer_id = 123
attributes = [
    {"attribute_name": "external_crm_id", "attribute_value": "SF-
12345"},
    {"attribute_name": "billing_method", "attribute_value":
"quarterly"},
    {"attribute_name": "support_tier", "attribute_value":
"premium"}
]

for attr in attributes:
    attr["customer_id"] = customer_id
    requests.put(
        "https://api.example.com/crm/attribute/",
        json=attr,
        headers={"Authorization": "Bearer YOUR_TOKEN"}
    )
```

Consultar Clientes por Atributo

Si bien los atributos no tienen endpoints de búsqueda integrados, puede filtrar clientes por atributo utilizando la API de búsqueda de clientes con filtrado personalizado:

```
# Obtener todos los clientes, luego filtrar por atributo en el código
aplicación
customers = requests.get("https://api.example.com/crm/customer/").jso

for customer in customers:
    attributes = requests.get(

f"https://api.example.com/crm/attribute/customer_id/{customer['custom
    ).json()

    # Encontrar clientes con un atributo específico
    for attr in attributes:
        if attr['attribute_name'] == 'support_tier' and
attr['attribute_value'] == 'premium':
            print(f"Cliente premium: {customer['customer_name']}")
```

> **NOTA**
>
> Para consultas frecuentes basadas en atributos, considere agregar campos de base de datos indexados o implementar un endpoint de búsqueda dedicado.

Mejores Prácticas

1. Convenciones de Nomenclatura

- Use snake_case: `external_system_id` ✓
- Sea descriptivo: `billing_method` ✓ vs `method` ✗
- Evite palabras clave reservadas o caracteres especiales
- Documente los significados de los atributos en su guía de implementación

2. Tipos de Datos

- Los atributos almacenan valores como cadenas (máx. 150 caracteres)
- Para booleanos, use "true"/"false" (minúsculas)
- Para fechas, use el formato ISO 8601: "2025-01-04 10:30:00"
- Para datos JSON grandes, considere utilizar campos de base de datos dedicados en su lugar

3. Validación

- Valide los valores de los atributos en el código de la aplicación antes de guardar
- Use formatos de valor consistentes entre clientes
- Documente los valores esperados para cada nombre de atributo

4. Documentación

- Mantenga un registro de nombres de atributos y propósitos
- Documente qué sistemas/integraciones dependen de atributos específicos
- Incluya ejemplos de valores válidos

5. Migración y Limpieza

- Audite regularmente los atributos no utilizados
- Elimine atributos obsoletos después de migraciones del sistema
- Versione los nombres de los atributos al cambiar esquemas (por ejemplo, `api_key_v2`)

Ejemplos de Flujos de Trabajo

Integración de Incorporación

Al migrar clientes desde un sistema heredado:

```
# Almacenar referencia del sistema heredado para depuración
PUT /crm/attribute/
{
  "customer_id": 123,
  "attribute_name": "legacy_crm_id",
  "attribute_value": "OLD-12345"
}

# Rastrear fecha de migración
PUT /crm/attribute/
{
  "customer_id": 123,
  "attribute_name": "migrated_date",
  "attribute_value": "2025-01-04"
}
```

Reglas de Facturación Personalizadas

Anule el ciclo de facturación predeterminado para un cliente específico:

```
# Establecer facturación trimestral
PUT /crm/attribute/
{
  "customer_id": 123,
  "attribute_name": "billing_cycle",
  "attribute_value": "quarterly"
}

# Luego en el código de facturación, verifique el atributo antes
de procesar
attributes = GET /crm/attribute/customer_id/123
billing_cycle = next(
    (a['attribute_value'] for a in attributes if
a['attribute_name'] == 'billing_cycle'),
    'monthly'  # predeterminado
)
```

Banderas de Características

Habilitar características beta para clientes específicos:

```
# Habilitar provisión IPv6
PUT /crm/attribute/
{
  "customer_id": 123,
  "attribute_name": "feature_ipv6_enabled",
  "attribute_value": "true"
}
```

Permisos

Las operaciones de atributos requieren los siguientes permisos:

- `view_customer_attribute` - Ver atributos
- `create_customer_attribute` - Crear nuevos atributos
- `update_customer_attribute` - Modificar atributos existentes
- `delete_customer_attribute` - Eliminar atributos

Consulte `rbac` para la configuración de control de acceso basado en roles.

Solución de Problemas

Atributo No Aparece en la UI

- Verifique que el atributo fue creado (verifique la respuesta de la API)
- Actualice la página para recargar los datos del cliente
- Verifique que el usuario tenga permiso `view_customer_attribute`

No se Puede Actualizar el Atributo

- Asegúrese de tener permiso `update_customer_attribute`
- Verifique que el attribute_id sea correcto
- Verifique que el atributo pertenezca al cliente especificado

Integración Fallando Después de la Eliminación de Atributo

- Restaure el atributo con el valor anterior
- Actualice el código de integración para manejar la falta de atributos de manera adecuada
- Audite las dependencias de atributos antes de la eliminación

Valor del Atributo Truncado

- Los valores de los atributos tienen un límite de 150 caracteres
- Para datos más largos, divídalos en múltiples atributos o use el campo de notas del cliente
- Considere almacenar datos grandes en campos de base de datos dedicados

Guía de Configuración y Personalización

Esta guía integral cubre todas las opciones de configuración y personalización para OmniCRM, incluyendo configuraciones de backend, branding de frontend, personalización visual y mejores prácticas de implementación.

Descripción General

OmniCRM utiliza dos sistemas de configuración principales:

Configuración de Backend:

- **Archivo:** `OmniCRM-API/crm_config.yaml`
- **Formato:** YAML
- **Requiere:** Reinicio de API después de cambios
- **Usado para:** Base de datos, integraciones, seguridad, aprovisionamiento

Configuración de Frontend (tiempo de ejecución, producción):

- **Fuente:** `crm_config.yaml` → bloque `ui:`, servido por `GET /crm/public/ui-config`
- **Formato:** JSON en tiempo de ejecución (YAML en configuración de implementación)
- **Requiere:** Recarga de API / actualización de montaje de configuración solamente: **sin reconstrucción de imagen de UI**
- **Usado para:** Branding, tema, localización, paneles de Grafana, accesos directos de aplicaciones web, banderas de características

Configuración de Frontend (fallback de desarrollo local):

- **Archivo:** `OmniCRM-UI/.env` (ver `.env.example`)
- **Formato:** variables de entorno `VITE_*`
- **Requiere:** Reinicio del servidor de desarrollo de UI

- **Usado cuando:** `/crm/public/ui-config` no está disponible; `runtimeConfig.js` recae en estos valores

La imagen de Docker genérica `omnicrm-ui` lee el branding en tiempo de ejecución. Las imágenes de UI por cliente y los argumentos de construcción `VITE_*` están en desuso para implementaciones en producción.

Configuración de Backend (crm_config.yaml)

El archivo `crm_config.yaml` contiene todos los ajustes del sistema de backend.

Ubicación: `/OmniCRM/OmniCRM-API/crm_config.yaml`

Configuración de Base de Datos

```yaml
database:
  username: omnitouch
  password: omnitouch2024
  server: localhost
```

Campos:

- `username` - Nombre de usuario de la base de datos MySQL
- `password` - Contraseña de la base de datos MySQL
- `server` - Nombre de host o IP del servidor de base de datos (por defecto: localhost)

Conexión a la Base de Datos:

- El nombre de la base de datos está codificado como `omnicrm`
- Puerto por defecto: 3306 (por defecto de MySQL)
- Cadena de conexión: `mysql+pymysql://username:password@server/omnicrm`

Nota de Seguridad: Nunca comprometas este archivo con credenciales reales en el control de versiones. Usa configuración específica del entorno o gestión de secretos.

Nota: Esto debe coincidir con tus credenciales de base de datos en `.env`.

Tipos de Servicio

```
service_types:
  - omnicharge
  - mobile
  - fixed
  - fixed-voice
  - hotspot
  - dongle
```

Propósito: Define los tipos de servicio válidos para tu implementación.

Estos se utilizan en todo el sistema para:

- Categorización de productos
- Filtrado de addons (los addons coinciden con los tipos de servicio)
- Flujos de trabajo de aprovisionamiento
- Informes y análisis

Agrega tipos de servicio personalizados aquí para tus casos de uso específicos.

Integración HSS (Servidor de Suscriptor en Casa)

Para operadores de redes móviles con integración HSS:

```
hss:
  hss_peers:
    - 'http://10.179.2.140:8080'
  apn_list: "1,2,3,4,5,6"
```

Configuración:

- `hss_peers` - Lista de puntos finales HSS para el aprovisionamiento de suscriptores
- `apn_list` - Lista separada por comas de IDs de APN disponibles para aprovisionamiento

Usado para: Aprovisionamiento de redes móviles y autenticación de suscriptores.

Configuración de Plantilla de Correo Electrónico de Mailjet

OmniCRM utiliza Mailjet para correos electrónicos transaccionales. Cada tipo de correo electrónico tiene su propia configuración de plantilla:

```yaml
mailjet:
  api_key: your_mailjet_api_key
  api_secret: your_mailjet_api_secret

  # Correo Electrónico de Bienvenida al Cliente
  api_crmCommunicationCustomerWelcome:
    from_email: "support@yourcompany.com"
    from_name: "Soporte de Tu Empresa"
    template_id: 5977509
    subject: "Bienvenido a YourCompany"

  # Correo Electrónico de Factura al Cliente
  api_crmCommunicationCustomerInvoice:
    from_email: "billing@yourcompany.com"
    from_name: "Facturación de Tu Empresa"
    template_id: 6759851
    subject: "Tu Factura - "

  # Recordatorio de Factura
  api_crmCommunicationCustomerInvoiceReminder:
    from_email: "billing@yourcompany.com"
    from_name: "Facturación de Tu Empresa"
    template_id: 5977570
    subject: "Recordatorio de Pago de Factura"

  # Correo Electrónico de Bienvenida al Usuario (Personal/Admin)
  api_crmCommunicationUserWelcome:
    from_email: "admin@yourcompany.com"
    from_name: "Admin de Tu Empresa"
    template_id: 6118112
    subject: "Bienvenido al Equipo"

  # Solicitud de Restablecimiento de Contraseña
  api_crmCommunicationUserPasswordReset:
    from_email: "security@yourcompany.com"
    from_name: "Seguridad de Tu Empresa"
    template_id: 6735666
    subject: "Solicitud de Restablecimiento de Contraseña"

  # Confirmación de Éxito en el Restablecimiento de Contraseña
  api_crmCommunicationUserPasswordResetSuccess:
    from_email: "security@yourcompany.com"
    from_name: "Seguridad de Tu Empresa"
```

```
    template_id: 6118378
    subject: "Restablecimiento de Contraseña Exitoso"

  # Notificación de Cambio de Contraseña
  api_crmCommunicationUserPasswordChange:
    from_email: "security@yourcompany.com"
    from_name: "Seguridad de Tu Empresa"
    template_id: 6118423
    subject: "Contraseña Cambiada"

  # Verificación de Correo Electrónico
  api_crmCommunicationEmailVerification:
    from_email: "verify@yourcompany.com"
    from_name: "Verificación de Tu Empresa"
    template_id: 6267350
    subject: "Verifica Tu Dirección de Correo Electrónico"

  # Notificación de Saldo Expirado
  api_crmCommunicationsBalanceExpired:
    from_email: "alerts@yourcompany.com"
    from_name: "Alertas de Tu Empresa"
    template_id: 7238252
    subject: "Saldo de Servicio Expirado"

  # Advertencia de Bajo Saldo
  api_crmCommunicationsBalanceLow:
    from_email: "alerts@yourcompany.com"
    from_name: "Alertas de Tu Empresa"
    template_id: 7238263
    subject: "Advertencia de Bajo Saldo"

  # Correo Electrónico de Entrega de eSIM
  # Establecer template_id a una plantilla válida de Mailjet para
usar una plantilla personalizada,
  # o dejar como 0 para usar el correo electrónico HTML de marca
incorporada
  api_crmCommunicationEsimDelivery:
    from_email: "support@yourcompany.com"
    from_name: "Soporte de Tu Empresa"
    template_id: 0
    subject: "Tu eSIM está Lista"
```

Correo Electrónico de Entrega de eSIM

Enviado cuando una eSIM es aprovisionada y está lista para la activación.

```
api_crmCommunicationEsimDelivery:
  from_email: "support@yourcompany.com"
  from_name: "Soporte de Tu Empresa"
  template_id: 0
  subject: "Tu eSIM está Lista"
```

Cuándo se Envía:

- eSIM aprovisionada durante el auto-registro
- eSIM asignada a un servicio al cliente

Variables de Plantilla Disponibles (cuando se usa una plantilla de Mailjet):

- `{{ var:customer_name }}` - Nombre completo del cliente
- `{{ var:iccid }}` - ICCID de la eSIM
- `{{ var:activation_code }}` - Cadena de código de activación LPA
- `{{ var:has_qr_code }}` - Booleano, verdadero si la imagen del código QR está adjunta
- `{{ var:has_activation_code }}` - Booleano, verdadero si el código de activación está disponible

Fallback Incorporado:

Cuando `template_id` se establece en `0`, OmniCRM envía un correo electrónico HTML de marca incorporada con el código QR de la eSIM en línea. El branding para este correo electrónico está controlado por la sección `branding` (ver más abajo).

Branding (Plantillas de Correo Electrónico Incorporadas)

Cuando OmniCRM envía correos electrónicos utilizando sus plantillas HTML incorporadas (en lugar de plantillas de Mailjet), la sección `branding` controla la apariencia visual.

```yaml
branding:
  brand_name: 'YourCompany'
  primary_dark: '#594712'
  dark_color: '#32280B'
  accent_bg: '#f5f0e0'
```

Campos:

Clave	Descripción	Predeterminado	Ejemplo
`brand_name`	Nombre de la empresa mostrado en el asunto del correo, saludo y despedida	Derivado de `from_name`	`'Ejemplo Móvil'`
`primary_dark`	Color principal de la marca utilizado para encabezados y bordes de acento	`#594712`	`'#1a73e8'`
`dark_color`	Color oscuro utilizado en el degradado del encabezado del correo	`#32280B`	`'#0d47a1'`
`accent_bg`	Color de fondo claro para cajas de información	`#f5f0e0`	`'#e3f2fd'`

Actualmente Usado Por:

- Correo electrónico de entrega de eSIM (cuando `api_crmCommunicationEsimDelivery.template_id` es `0`)

Creando Plantillas de Mailjet:

1. Inicia sesión en el panel de Mailjet (<https://app.mailjet.com>)

2. Navega a **Transaccional → Plantillas**
3. Crea una nueva plantilla o clona una existente
4. Toma nota del **ID de Plantilla** (valor numérico)
5. Agrega variables de plantilla que coincidan con la estructura de datos de OmniCRM
6. Actualiza `crm_config.yaml` con el ID de plantilla

Variables de Plantilla Disponibles:

Cada tipo de correo electrónico recibe variables específicas. Ejemplos comunes:

- `{{customer_name}}` - Nombre del cliente o usuario
- `{{service_name}}` - Nombre del servicio o producto
- `{{invoice_id}}` - Número de factura
- `{{invoice_amount}}` - Monto total de la factura
- `{{due_date}}` - Fecha de vencimiento del pago
- `{{reset_link}}` - URL de restablecimiento de contraseña
- `{{verification_link}}` - URL de verificación de correo electrónico
- `{{balance}}` - Saldo actual de la cuenta
- `{{expiry_date}}` - Fecha de expiración del saldo o servicio

Configuración de Aprovisionamiento

```
provisioning:
  failure_list: ['admin@yourcompany.com', 'ops@yourcompany.com']
```

Propósito:

- `failure_list` - Direcciones de correo electrónico notificadas cuando el aprovisionamiento de Ansible falla
- Las notificaciones incluyen el nombre del playbook, detalles del error y la información del cliente
- Permite al equipo de operaciones responder rápidamente a problemas de aprovisionamiento

Configuración de Factura

```yaml
invoice:
  template_filename: 'yourcompany_invoice_template.html'
```

Propósito:

Especifica qué plantilla HTML de Jinja2 usar para la generación de facturas PDF.

Ubicación de la Plantilla: `/OmniCRM-API/invoice_templates/`

Consulta la sección **Generación de PDF de Factura** a continuación para obtener detalles sobre cómo crear plantillas personalizadas.

URL Base de CRM

```yaml
crm:
  base_url: 'http://localhost:5000'
```

Propósito:

- Usado por los playbooks de Ansible para hacer llamadas a la API
- Usado en plantillas de correo electrónico para generar enlaces al CRM
- Debe ser la URL accesible públicamente de tu API

Ejemplos:

- Desarrollo: `http://localhost:5000`
- Producción: `https://api.yourcompany.com`

Importante:

- No incluyas barra diagonal final
- Usa URL accesible públicamente si los playbooks se ejecutan en diferentes servidores
- Actualiza al implementar en producción

Configuración de OCS y CGRates

```
ocs:
  ocsApi: 'http://10.179.2.142:8080/api'
  ocsTenant: 'mnc001.mcc001.3gppnetwork.org'
  cgrates: 'localhost:2080'
```

Configuración:

- `ocsApi` - Punto final de la API OCS para la gestión de suscriptores
- `ocsTenant` - Identificador de inquilino para implementaciones OCS multi-inquilino
- `cgrates` - Punto final de la API JSON-RPC de CGRates (host:puerto)

Usado para: Carga en tiempo real, gestión de saldo, seguimiento de uso.

Configuración de SMSC (Puerta de Enlace SMS)

```
smsc:
  source_msisdn: 'YourCompany'
  smsc_url: 'http://10.179.2.216/SMSc/'
  api_key: 'your_smsc_api_key'
```

Propósito:

- Enviar notificaciones SMS a los clientes (bajo saldo, alertas de servicio, códigos 2FA)
- `source_msisdn` - ID de remitente mostrado a los destinatarios (alfanumérico o número de teléfono)
- `smsc_url` - Punto final de la API de la puerta de enlace SMSC
- `api_key` - Autenticación para la API SMSC

Configuración de Difusión Celular

```
cbc_url: 'http://10.179.1.113:8080'
```

Propósito: Punto final de la API del Centro de Difusión Celular (CBC) para alertas de emergencia.

Consulta `features_cell_broadcast` para detalles de uso.

Clave Secreta JWT

```
jwt_secret: 'CHANGE_ME_ON_DEPLOYMENT'  # Generado automáticamente por Ansible
```

Seguridad:

- Usado para firmar y verificar tokens de autenticación (JWT)
- **Crítico para la seguridad de autenticación** - trátalo como una contraseña
- Cada implementación/entorno DEBE tener un secreto único
- Nunca compartas ni comprometas en el control de versiones

Gestión de Secretos Automatizada (Implementación de Ansible):

Al implementar con Ansible, el secreto JWT se gestiona automáticamente:

1. **Primera implementación:** Ansible genera una cadena hexadecimal aleatoria de 64 caracteres (256 bits de entropía)
2. **Implementaciones posteriores:** Ansible preserva el secreto existente del servidor
3. **Cada servidor obtiene su propio secreto único** - las sesiones no funcionan entre entornos

Esto asegura:

- Sin secretos codificados en repositorios git
- Cada entorno está criptográficamente aislado
- Los secretos persisten a través de implementaciones (las sesiones no se invalidan en la nueva implementación)
- Reconstruir un servidor genera un nuevo secreto (las sesiones se invalidan correctamente)

Generación Manual de Secretos (Implementaciones No Ansible):

Si implementas sin Ansible, DEBES generar manualmente un secreto único:

```
# Generar una clave aleatoria segura criptográficamente
python3 -c "import secrets; print(secrets.token_hex(32))"
# o
openssl rand -hex 32
```

Luego actualiza `crm_config.yaml`:

```
jwt_secret: 'your_generated_secret_here'
```

Importante:

- **Nunca uses el valor de marcador de posición predeterminado** `CHANGE_ME_ON_DEPLOYMENT` en producción
- **Nunca compartas públicamente** - cualquiera con el secreto puede falsificar tokens de autenticación
- Cambiar esto invalida todas las sesiones de usuario existentes (los usuarios deben volver a iniciar sesión)
- Cada entorno (dev, staging, prod) DEBE tener secretos diferentes

Configuración de Pagos de Stripe

```
stripe:
  secret_key: 'sk_live_xxxxxxxxxx'
  publishable_key: 'pk_live_xxxxxxxxxx'
  currency: 'aud'
  statement_descriptor_suffix: 'YOURCOMPANY'
```

Configuración:

- `secret_key` - Clave API secreta de Stripe (lado del servidor, mantener confidencial)

- `publishable_key` - Clave publicable de Stripe (lado del cliente, seguro de exponer)
- `currency` - Código de moneda ISO 4217 (aud, usd, gbp, eur, etc.)
- `statement_descriptor_suffix` - Aparece en los estados de cuenta de tarjetas de crédito de los clientes

Tipos de Clave:

- Claves de prueba: `sk_test_...` y `pk_test_...` (para desarrollo)
- Claves en vivo: `sk_live_...` y `pk_live_...` (para producción)

Uso del Descriptor de Estado:

- Mostrado en los estados de cuenta bancarios de los clientes como "YOURCOMPANY"
- Máximo 22 caracteres
- Ayuda a los clientes a identificar los cargos
- También se utiliza en los nombres de archivos PDF de facturas (por ejemplo, `YOURCOMPANY_12345.pdf`)

Consulta *Integraciones de Proveedores de Pago* para detalles de configuración.

Claves API y Lista Blanca de IP

Define claves API con acceso basado en roles y restricciones de IP:

```
api_keys:
  "YOUR_API_KEY_1":
    roles: ["admin"]
    ips: ["127.0.0.1", "::1"]
  "YOUR_API_KEY_2":
    roles: ["customer_service_agent_1"]
    ips: ["127.0.0.1", "::1", "10.0.1.0/24"]

# Lista Blanca de IP (independiente, sin clave API)
ip_whitelist:
  "10.179.2.142":
    roles: ["admin"]
```

Propósito:

- Permitir que sistemas externos se autentiquen a través de la clave API
- Restringir el acceso por dirección IP
- Otorgar roles específicos a los consumidores de API
- Útil para integraciones (sistemas de facturación, monitoreo, automatización)

Generación de Claves API:

```
openssl rand -base64 48
```

Roles:

- `admin` - Acceso completo a todos los puntos finales
- Roles personalizados definidos en el sistema RBAC

Mejores Prácticas de Seguridad:

- Usa claves API largas y aleatorias (mínimo 32 caracteres)
- Restringe las IP a fuentes conocidas solamente
- Otorga los roles mínimos necesarios
- Rota las claves API regularmente
- Monitorea el uso de claves API en los registros

Advertencia de Seguridad:

- Solo usa la lista blanca de IP para redes internas de confianza
- No debes usar IPs de localhost (127.0.0.1, ::1)
- Usa claves API en su lugar para acceso externo
- Considera reglas de firewall como protección adicional

Consulta `administration_api_keys` y `concepts_api` para detalles.

Configuración de UI en Tiempo de Ejecución (`ui:` en crm_config.yaml)

Las implementaciones en producción utilizan una imagen **genérica** `omnicrm-ui`. El branding, el tema y los toggles de características se cargan en tiempo de ejecución desde la API:

- **Endpoint:** `GET /crm/public/ui-config` (compatible con ETag; la UI se almacena en caché en `localStorage`)
- **Activos:** logotipos, favicons e íconos de aplicaciones web bajo `ui.assets_dir`, servidos desde `/crm/public/assets/...`
- **Bloque `branding:` legado:** aún utilizado para plantillas de correo electrónico; `ui.branding` y `ui.theme` son la fuente de verdad de la UI

Ejemplo de bloque `ui:` (ver también `OmniCRM-API/crm_config.yaml`):

```yaml
ui:
  version: '1'
  assets_dir: '/etc/omnicrm/deployment_assets'
  branding:
    client_name: 'Ejemplo Móvil'
    portal_name: 'Omnitouch'
    self_care_name: 'Ejemplo Móvil'
    logos:
      sm: logos/logoSm.png
      dark: logos/logoDark.png
      light: logos/logoLight.png
      favicon: logos/favicon.ico
  theme:
    primary: '#0D0629'
    secondary: '#CBB42B'
    font_family: Quicksand
  locale:
    default_language: en
    locale: en-GB
    default_country: Australia
  integrations:
    google_maps_api_key: ''
    google_tag_id: ''
    grafana:
      dashboard_ids: 'id1,id2'
      dashboard_labels: 'Ops,Finance'
  features:
    cbc_mode: no
    simplified_addresses: no
  links:
    faqs_url: 'https://example.com/faqs'
    support_url: 'https://example.com/support'
  web_apps:
    - name: GitHub
      url: https://github.com
      icon_path: webapp_icons/github.png
```

Los logotipos predeterminados de Omnitouch se **envían con el paquete** `omnicrm-api` bajo `/usr/share/omnicrm/deployment_assets/logos/`. Las implementaciones sin una superposición de `deployment_assets` de cliente aún sirven el branding desde esos valores predeterminados. Las sobreescrituras por cliente: sincroniza a `/etc/omnicrm/deployment_assets` (Ansible copia

`group_vars/deployment_assets/` y/o `logo_*`) y referencia rutas en `ui.branding.logos`. Establece `ui.assets_dir: '/etc/omnicrm/deployment_assets'`, nunca `/etc/omnicrm` en sí. Si esa ruta falta (Docker), la API recae en `UI_ASSETS_DIR` y luego `/app/deployment_assets`. `GET /crm/public/assets/` solo sirve archivos de imagen bajo `logos/`, `login/`, y `webapp_icons/`.

Migrando de `VITE_*`: mapea cada antiguo argumento de construcción al campo correspondiente de `ui:` (ver comentarios en `.env.example`). `npm run dev` local aún lee `VITE_*` a través de `runtimeConfig.js` cuando el endpoint de API no está disponible.

Configuración de Frontend (.env): fallback de desarrollo local

Ubicación: `/OmniCRM/OmniCRM-UI/.env`

Claves API y Servicios Externos

```
# Google Maps API (para autocompletar direcciones y
geocodificación)
VITE_GOOGLE_API_KEY=your_google_api_key

# Desactivar el lanzamiento automático del navegador al iniciar
npm
BROWSER=none
```

Configuración de Proveedores de Pago

```
# Proveedores de pago habilitados (separados por comas): stripe,
paypal, o ambos
VITE_PAYMENT_VENDORS="stripe,paypal"

# Clave publicable de Stripe (requerida si Stripe está habilitado)
VITE_STRIPE_PUBLISHABLE_KEY=pk_test_xxxxx

# ID de cliente de PayPal (requerido si PayPal está habilitado)
VITE_PAYPAL_CLIENT_ID=YOUR_PAYPAL_CLIENT_ID
```

Debe Coincidir con el Backend:

- `VITE_STRIPE_PUBLISHABLE_KEY` debe coincidir con `stripe.publishable_key` en `crm_config.yaml`
- `VITE_PAYPAL_CLIENT_ID` debe coincidir con `paypal.client_id` en `crm_config.yaml`

Importante: Estas son variables **de tiempo de construcción**. Cambiar cualquier valor `VITE_*` requiere reconstruir y reiniciar la UI. Consulta Guía del Sistema de Pagos - Configuración de Proveedores para el procedimiento completo de configuración incluyendo la configuración del backend.

Branding e Información de la Empresa

```
# Branding de la Empresa
VITE_COMPANY_NAME="ShellFone"
VITE_PORTAL_NAME="ShellManager"
VITE_SELF_CARE_NAME="ShellCare"
VITE_COMPANY_TAGLINE="Teléfonos con Conchas"
```

Estos valores aparecen en toda la UI:

- `COMPANY_NAME` - Mostrado en títulos de página, correos electrónicos y comunicaciones con clientes
- `PORTAL_NAME` - Nombre del portal de administración/personal (por ejemplo, "ShellManager")

- `SELF_CARE_NAME` - Nombre del portal de autoservicio del cliente (por ejemplo, "ShellCare")
- `COMPANY_TAGLINE` - Aparece en pantallas de inicio de sesión y materiales de marketing

Localización y Configuraciones Regionales

```
# Idioma y Localización
# Idiomas soportados: ar, ch, en, fr, gr, it, ru, sp
VITE_DEFAULT_LANGUAGE=en
VITE_LOCALE="en-GB"

# Ubicación Predeterminada (para autocompletar direcciones)
VITE_DEFAULT_LOCATION="Sídney, Australia"
VITE_DEFAULT_COUNTRY="Australia"

# Configuraciones de Moneda
VITE_CURRENCY_CODE="GBP"
VITE_CURRENCY_SYMBOL="£"
```

Idiomas Soportados:

- `ar` - Árabe
- `ch` - Chino
- `en` - Inglés (predeterminado)
- `fr` - Francés
- `gr` - Griego
- `it` - Italiano
- `ru` - Ruso
- `sp` - Español

Monedas Comunes:

- USD - $ (Dólar estadounidense)
- GBP - £ (Libra esterlina)
- EUR - € (Euro)
- AUD - $ (Dólar australiano)

- CAD - $ (Dólar canadiense)

Nota: Debe coincidir con `stripe.currency` en `crm_config.yaml`.

Configuración del Tema de Color

```
# Color Primario (color principal de la marca)
VITE_PRIMARY_COLOR=#405189

# Opciones de Color Adicionales (ejemplos comentados)
# VITE_SECONDARY_COLOR=#2bFFcf
# VITE_TERTIARY_COLOR=#1a9fbf
# VITE_SUCCESS_COLOR=#28a745
# VITE_INFO_COLOR=#17a2b8
# VITE_WARNING_COLOR=#ffc107
# VITE_DANGER_COLOR=#dc3545
```

Colores Disponibles:

- `VITE_PRIMARY_COLOR` - Color principal de la marca (botones, enlaces, resaltados)
- `VITE_SECONDARY_COLOR` - Color de acento
- `VITE_TERTIARY_COLOR` - Acento adicional
- `VITE_SUCCESS_COLOR` - Mensajes de éxito (#28a745)
- `VITE_INFO_COLOR` - Mensajes de información (#17a2b8)
- `VITE_WARNING_COLOR` - Advertencias (#ffc107)
- `VITE_DANGER_COLOR` - Errores (#dc3545)
- `VITE_LIGHT_COLOR` - Fondos claros (#f8f9fa)
- `VITE_DARK_COLOR` - Texto oscuro (#343a40)
- `VITE_PRIMARY_DARK_COLOR` - Variante oscura del color primario

Formato: Códigos de color hexadecimal (#RRGGBB)

El color primario se aplica a:

- Encabezados de navegación
- Botones de acción

- Enlaces y resaltados
- Estados activos
- Elementos de marca

Configuración de Fuente

```
# Configuración de Familia de Fuentes
# Sans-Serif: Inter, Roboto, Open Sans, Lato, Quicksand, Poppins,
Nunito, Montserrat, Work Sans, Source Sans Pro, Raleway, Ubuntu,
Josefin Sans, HKGrotesk
# Serif: Merriweather, Lora, Playfair Display
# Sistema: Sistema (fuentes nativas del dispositivo)
# Predeterminado: Quicksand
VITE_FONT_FAMILY=Quicksand
```

Importante: Todas las fuentes están **auto-alojadas localmente** dentro de la aplicación OmniCRM-UI. Esto significa:

- **Sin carga de fuentes externas** - Las fuentes están empaquetadas con la aplicación
- **Compatible con jardín amurallado** - No se requiere acceso a internet para que funcionen las fuentes
- **Operación sin conexión** - Funcionalidad completa en entornos de red restringida o aislada
- **Privacidad** - Sin solicitudes externas a Google Fonts, Adobe Fonts u otros CDNs
- **Rendimiento** - Carga más rápida sin dependencias externas
- **Seguridad** - Sin seguimiento de terceros ni filtraciones de datos a través de solicitudes de fuentes

Opciones Disponibles:

Fuentes Sans-Serif:

- Inter, Roboto, Open Sans, Lato, Quicksand (predeterminado), Poppins, Nunito, Montserrat, Work Sans, Source Sans Pro, Raleway, Ubuntu, Josefin Sans, HKGrotesk

Fuentes Serif:

- Merriweather, Lora, Playfair Display

Fuentes del Sistema:

- Sistema - Utiliza fuentes nativas del dispositivo para mejor rendimiento y menor tamaño de paquete

Agregar Fuentes Personalizadas:

¡Sí, puedes agregar fuentes adicionales! Todas las fuentes se almacenan localmente en la aplicación.

Para agregar una nueva fuente personalizada:

1. **Agrega archivos de fuente** a `OmniCRM-UI/src/assets/fonts/your-font-name/`
    - Usa formato WOFF2 para mejor compresión y soporte de navegador
    - Incluye múltiples pesos (300, 400, 500, 600, 700) para un renderizado adecuado
    - Nombra los archivos: `your-font-name-300.woff2`, `your-font-name-400.woff2`, etc.
2. **Define reglas @font-face** en `OmniCRM-UI/src/assets/scss/fonts/_google-fonts.scss`

```
//
// Tu Fuente Personalizada - Descripción
//

@font-face {
  font-family: 'Tu Nombre de Fuente';
  font-style: normal;
  font-weight: 400;
  font-display: swap;
  src: url("../../fonts/your-font-name/your-font-name-
400.woff2") format('woff2');
}

@font-face {
  font-family: 'Tu Nombre de Fuente';
  font-style: normal;
  font-weight: 700;
  font-display: swap;
  src: url("../../fonts/your-font-name/your-font-name-
700.woff2") format('woff2');
}
```

3. **Establecer en el archivo .env:**

```
VITE_FONT_FAMILY=Tu Nombre de Fuente
```

Guías de Peso de Fuente:

- 300 - Ligera (opcional, para encabezados sutiles)
- 400 - Regular (requerido, texto predeterminado)
- 500 - Medio (opcional, énfasis)
- 600 - Semi-Negrita (opcional, subtítulos)
- 700 - Negrita (requerido, encabezados y texto fuerte)

Enlaces Rápidos de Aplicaciones Web

Configura hasta 6 aplicaciones web de acceso rápido que aparecen en el panel de administración:

```
# Aplicación Web 1: GitHub
VITE_WEB_APP_1_NAME="GitHub"
VITE_WEB_APP_1_URL="https://github.com"
VITE_WEB_APP_1_ICON_PATH="resources/webapp_icons/github.png"

# Aplicación Web 2: Xero
VITE_WEB_APP_2_NAME="Xero"
VITE_WEB_APP_2_URL="https://go.xero.com/"
VITE_WEB_APP_2_ICON_PATH="resources/webapp_icons/xero.png"

# Aplicación Web 3-6: Integraciones adicionales
# (Configura de manera similar con NAME, URL e ICON_PATH)
```

Patrón:

- `VITE_WEB_APP_N_NAME` - Nombre de visualización
- `VITE_WEB_APP_N_URL` - URL de destino
- `VITE_WEB_APP_N_ICON_PATH` - Ruta del archivo de ícono (relativa a public/)

Ejemplos de Íconos: GitHub, Xero, Monday.com, Gmail, MailJet, Slack

Integración del Panel de Grafana

OmniCRM se integra con Grafana para proporcionar paneles de análisis. Los usuarios con el permiso `grafana_access` pueden acceder a Grafana en `/grafana` para crear paneles personalizados, y esos paneles pueden ser incrustados en la interfaz de OmniCRM.

```
# Integración del Panel de Grafana
# IDs de Paneles (separados por comas) - obtén estos de las URLs
/grafana/d/ID/name
VITE_GRAFANA_DASHBOARD_IDS=abc123,def456,ghi789

# Etiquetas de Paneles (separadas por comas, deben coincidir con
el orden de IDs)
VITE_GRAFANA_DASHBOARD_LABELS=Métricas del Sistema,Analítica de
Usuarios,Rendimiento de la Red
```

Campos:

- `VITE_GRAFANA_DASHBOARD_IDS` - Lista separada por comas de IDs de paneles de Grafana desde la ruta de URL
- `VITE_GRAFANA_DASHBOARD_LABELS` - Lista separada por comas de nombres de visualización de paneles (debe coincidir con el orden de IDs)

Cómo Encontrar IDs de Paneles:

Al ver un panel en Grafana, la URL se verá como `/grafana/d/abc123/dashboard-name`. El ID del panel es la parte después de `/d/` y antes del nombre (en este ejemplo, `abc123`).

Uso:

Una vez configurados, los paneles aparecen automáticamente en la barra lateral de OmniCRM bajo el menú "Paneles". Los usuarios pueden hacer clic para ver paneles incrustados mientras mantienen visible la navegación de OmniCRM.

Ejemplo:

Si tu URL de panel es `http://your-server/grafana/d/system-overview-123/system-metrics`, entonces:

- ID del panel: `system-overview-123`
- Etiqueta del panel: Elige cualquier nombre amigable como "Visión General del Sistema"

Importante:

- El número de IDs debe coincidir con el número de etiquetas
- Los paneles aparecen en la barra lateral en el orden en que se definen
- Solo los usuarios con el permiso `grafana_access` pueden ver los paneles
- Los cambios requieren reconstruir la UI frontend

URLs de Soporte y Documentación

```
# URLs de FAQs y Soporte
VITE_FAQS_URL=https://support.yourcompany.com/faqs
VITE_SUPPORT_URL=https://support.yourcompany.com
```

Propósito: Enlaces a recursos de soporte externos mostrados en la UI.

Inicios de Sesión Sociales

```
# Permitir Inicios de Sesión Sociales (sí|no)
VITE_ALLOW_SOCIAL_LOGINS=yes
```

Opciones:

- `yes` - Habilitar botones de inicio de sesión social (Google, Facebook, etc.)
- `no` - Deshabilitar inicios de sesión sociales

Nota: Los proveedores de inicio de sesión social deben configurarse por separado.

Análisis y Seguimiento

```
# Google Analytics 4 o Google Tag Manager (opcional)
# Para GA4, usa el formato: G-XXXXXXXXXX
# Para GTM, usa el formato: GTM-XXXXXXX
VITE_GOOGLE_TAG_ID=G-XXXXXXXXXX
```

Propósito: Habilitar el seguimiento de Google Analytics 4 (GA4) o Google Tag Manager (GTM) para monitorear el comportamiento del usuario y el uso de la aplicación.

Formatos Soportados:

- **ID de Medición GA4:** `G-XXXXXXXXXX` - Para el seguimiento de Google Analytics 4

- **ID de Contenedor GTM:** `GTM-XXXXXXX` - Para Google Tag Manager
- **Deshabilitado:** Deja vacío o omite para deshabilitar el seguimiento

Cómo Funciona:

El sistema detecta automáticamente el servicio de seguimiento según el formato del ID:

- Los IDs que comienzan con `G-` inicializan Google Analytics 4
- Los IDs que comienzan con `GTM-` inicializan Google Tag Manager
- Valor vacío o faltante desactiva todo el seguimiento

Características:

- **Seguimiento Automático de Páginas** - Cada cambio de ruta se rastrea como una vista de página
- **Seguimiento de Eventos Personalizados** - Rastrear interacciones de usuario, clics en botones, envíos de formularios
- **Sin Cambios de Código Requeridos** - Solo establece la variable de entorno y reconstruye la UI

Instrucciones de Configuración:

1. **Obtén Tu ID de Seguimiento:**

   Para Google Analytics 4:

   - Inicia sesión en Google Analytics
   - Crea o selecciona una propiedad GA4
   - Ve a **Admin** → **Flujos de Datos**
   - Selecciona tu flujo web
   - Copia el **ID de Medición** (formato: `G-XXXXXXXXXX`)

   Para Google Tag Manager:

   - Inicia sesión en Google Tag Manager
   - Crea o selecciona un contenedor
   - Copia el **ID de Contenedor** (formato: `GTM-XXXXXXX`)

2. **Agrega al Archivo .env:**

```
VITE_GOOGLE_TAG_ID=G-XXXXXXXXXX
```

3. **Reconstruye la UI:**

```
cd OmniCRM-UI
npm run build
```

4. **Verifica el Seguimiento:**

   - Abre tu aplicación en un navegador
   - Verifica la consola del navegador para mensajes de inicialización
   - Usa los informes en tiempo real de Google Analytics para ver usuarios activos

Uso Avanzado - Seguimiento de Eventos Personalizados:

La integración proporciona funciones de utilidad para rastrear eventos personalizados en toda tu aplicación.

Ejemplo - Rastrear Clics en Botones:

```
import { trackEvent } from './utils/googleAnalytics';

// En tu componente
const handleCheckout = () => {
  trackEvent('checkout_initiated', {
    cart_value: 99.99,
    item_count: 3,
    page: '/cart'
  });
  // Continuar con la lógica de pago
};
```

Ejemplo - Rastrear Envíos de Formularios:

```
import { trackEvent } from './utils/googleAnalytics';

const handleFormSubmit = (formType) => {
  trackEvent('form_submission', {
    form_type: formType,
    timestamp: new Date().toISOString()
  });
};
```

Ejemplo - Rastrear Vistas de Página Personalizadas:

```
import { trackPageView } from './utils/googleAnalytics';

// Rastrear manualmente una vista de página (útil para diálogos modales, pestañas, etc.)
trackPageView('/virtual/page-path', 'Título de Página');
```

Funciones de Seguimiento Disponibles:

Función	Propósito	Parámetros
`initializeGoogleTracking()`	Inicializar seguimiento al inicio de la app	Ninguno (llamado automáticamente)
`trackEvent(eventName, eventParams)`	Rastrear eventos personalizados	Nombre del evento (cadena), parámetros (objeto)
`trackPageView(path, title)`	Rastrear vistas de página manualmente	Ruta (cadena), título (cadena)

Privacidad y Cumplimiento:

El seguimiento de Google Analytics recopila datos de comportamiento del usuario. Asegúrate de cumplir con las regulaciones de protección de datos:

- **GDPR (Europa)** - Obtén el consentimiento del usuario antes de rastrear, proporciona un mecanismo de exclusión
- **CCPA (California)** - Permite a los usuarios optar por no participar en la recopilación de datos
- **Política de Privacidad** - Divulga el uso de Google Analytics en tu política de privacidad
- **Consentimiento de Cookies** - Implementa un banner de consentimiento de cookies si es requerido por las leyes locales
- **Retención de Datos** - Configura períodos de retención de datos apropiados en la configuración de Google Analytics

Solución de Problemas:

Problema	Causa Posible	Solución
No hay datos en el panel de GA	ID de seguimiento incorrecto	Verifica que el formato del ID coincida con G-XXXXXXXXXX o GTM-XXXXXXX
Errores en la consola al cargar la página	ID de seguimiento inválido	Verifica errores tipográficos en el archivo .env
Las vistas de página no se rastrean	UI no se reconstruyó después del cambio en .env	Ejecuta `npm run build` y reinicia la UI
Eventos no aparecen	Bloqueador de anuncios habilitado	Prueba con el bloqueador de anuncios deshabilitado
Datos retrasados	Comportamiento normal de GA	Usa informes en tiempo real para ver la verificación inmediata

Consideraciones de Rendimiento:

- Los scripts de seguimiento se cargan de manera asíncrona y no bloquean el renderizado de la página
- Impacto de rendimiento mínimo (< 50KB de carga adicional)
- Los scripts se almacenan en caché por los navegadores después de la primera carga
- No se realiza seguimiento si `VITE_GOOGLE_TAG_ID` está vacío

Detalles de Implementación:

- Código de seguimiento: `OmniCRM-UI/src/utils/googleAnalytics.js`
- Hook de seguimiento de página: `OmniCRM-UI/src/utils/usePageTracking.js`
- Inicialización: Automática al inicio de la app en `App.js` o `index.js`

Configuración de Recarga y Recarga

```
VITE_TOPUP_PRICE_PER_DAY=10
```

Propósito: Establece el precio por día para servicios de recarga en el portal de autoservicio.

Campos:

- `VITE_TOPUP_PRICE_PER_DAY` - Precio diario para servicios de recarga (valor numérico)

Ejemplo: Si se establece en `10` y la moneda es USD, los clientes pagan $10 por día de servicio.

Nota: Este valor debe coincidir con la configuración de precios del backend. Consulta `features_topup_recharge` para detalles completos de configuración.

Branding y Personalización Visual

OmniCRM te permite reemplazar las imágenes de branding predeterminadas con el logotipo y las pantallas de inicio de tu empresa sin modificar el código.

Archivos de Logotipo y Sistema de Fallback

Los logotipos se almacenan en `/OmniCRM-UI/src/assets/images/omnitouch/` y utilizan un sistema de fallback:

Logotipos Predeterminados (siempre presentes):

- `DefaultLogoDark.png` - Logotipo de tema oscuro (usado en fondos claros)
- `DefaultLogoLight.png` - Logotipo de tema claro (usado en fondos oscuros)

Logotipos Personalizados (opcional, tienen prioridad cuando están presentes):

- `logoSm.png` - Logotipo pequeño para la barra lateral colapsada (recomendado: 100x100px)
- `logoDark.png` - Logotipo oscuro de tamaño completo para encabezados (recomendado: 200x50px)
- `logoLight.png` - Logotipo claro de tamaño completo para pantallas de autenticación (recomendado: 200x50px)

Cómo Funciona el Fallback de Logotipo:

El sistema intenta cargar primero los logotipos personalizados. Si un archivo de logotipo personalizado no existe, recae en el predeterminado:

```
// Desde Header.js
const tryImport = (filename) => {
    try {
        return require(`../assets/images/omnitouch/${filename}`);
    } catch (err) {
        return null; // Recae en el predeterminado
    }
};

const userLogoSm = tryImport("logoSm.png");
const userLogoDark = tryImport("logoDark.png");
const userLogoLight = tryImport("logoLight.png");
```

Dónde Aparecen los Logotipos:

- **logoSm.png** - Barra lateral colapsada, navegación móvil, displays de encabezado pequeños
- **logoDark.png** - Barra de encabezado principal (modo claro), encabezado del panel de administración
- **logoLight.png** - Pantallas de inicio de sesión/registro, fondos oscuros, carrusel de autenticación

Reemplazando Logotipos:

1. **Crea Tus Archivos de Logotipo:**
    - Usa formato PNG para soporte de transparencia
    - Coincide con las dimensiones recomendadas arriba
    - Asegúrate de que los logotipos sean claros tanto en resoluciones regulares como en retina
2. **Agrega a OmniCRM:**

```
# Copia tus archivos de logotipo al directorio de imágenes de
omnitouch
cp /path/to/your/logoSm.png OmniCRM-
UI/src/assets/images/omnitouch/
cp /path/to/your/logoDark.png OmniCRM-
UI/src/assets/images/omnitouch/
cp /path/to/your/logoLight.png OmniCRM-
UI/src/assets/images/omnitouch/
```

3. **Reconstruye la UI:**

```
cd OmniCRM-UI
npm run build
```

4. **Verifica los Cambios:**
    - Verifica el encabezado en modo claro (debería mostrar `logoDark.png`)
    - Verifica el encabezado en modo oscuro (debería mostrar `logoLight.png`)
    - Verifica la barra lateral colapsada (debería mostrar `logoSm.png`)
    - Verifica la pantalla de inicio de sesión (debería mostrar `logoLight.png`)

Mejores Prácticas de Diseño de Logotipos:

- **Contraste** - Asegúrate de que los logotipos sean visibles en fondos claros y oscuros
- **Simplicidad** - Los logotipos deben ser reconocibles en tamaños pequeños
- **Formato** - Usa PNG con fondos transparentes
- **Retina** - Proporciona resolución 2x para pantallas de alta DPI
- **Consistencia** - Usa los mismos colores de marca en todas las variantes de logotipo

Pantallas de Inicio y Fondos de Autenticación

Las pantallas de autenticación (inicio de sesión, registro, restablecimiento de contraseña) utilizan un fondo de carrusel con imágenes personalizables.

Ubicación: `/OmniCRM-UI/src/pages/AuthenticationInner/authCarousel.js`

Configuración Predeterminada:

```
import logoLight from "../../assets/images/logo-light.png";

// Logotipo mostrado en pantallas de autenticación
<img src={logoLight} alt="" height="18" />
```

Personalizando Pantallas de Autenticación:

1. Reemplaza `logo-light.png` en `/OmniCRM-UI/src/assets/images/`
2. Agrega CSS de fondo personalizado a la clase `.auth-one-bg`
3. Modifica las citas del carrusel en `authCarousel.js` para que coincidan con la voz de tu marca

Ejemplo de Personalización:

```
/* Agrega a tu CSS personalizado */
.auth-one-bg {
    background-image: url('/assets/images/custom-auth-bg.jpg');
    background-size: cover;
    background-position: center;
}
```

Generación de PDF de Factura

OmniCRM genera facturas PDF profesionales utilizando plantillas HTML de Jinja2 y renderizado PDF de WeasyPrint.

Cómo Funciona la Generación de PDF

1. **Selección de Plantilla:**
    - El nombre del archivo de plantilla se especifica en `crm_config.yaml` bajo `invoice.template_filename`

- La plantilla se carga desde el directorio `/OmniCRM-API/invoice_templates/`

2. **Preparación de Datos:**
   - Los datos de la factura (ID, fechas, montos, estado) se obtienen de la base de datos
   - La información del cliente (nombre, dirección) se recupera
   - Se compila la lista de transacciones (todos los cargos/créditos en la factura)
3. **Renderizado de Plantilla:**
   - Jinja2 renderiza la plantilla HTML con los datos de la factura
   - Variables como `{{ invoice_number }}`, `{{ total_amount }}`, etc. son reemplazadas
   - El HTML renderizado se guarda en `invoice_templates/rendered/` para depuración
4. **Generación de PDF:**
   - WeasyPrint convierte el HTML renderizado a PDF
   - El PDF soporta estilos CSS, imágenes, saltos de página, encabezados/pies de página
   - Los datos binarios del PDF se generan en memoria
5. **Caché:**
   - El PDF se codifica en Base64 y se almacena en la tabla `Invoice_PDF_Cache`
   - Se calcula un hash SHA256 para verificación de integridad
   - Las solicitudes subsiguientes devuelven el PDF en caché (entrega instantánea)
6. **Invalidación de Caché:**
   - La caché se invalida cuando la factura se modifica, anula o reembolsa
   - Los cambios en la plantilla no invalidan automáticamente las cachés existentes

Estructura de Plantilla de Factura

Las plantillas de factura son archivos HTML de Jinja2 con variables y lógica incrustadas.

Ubicación de la Plantilla: `/OmniCRM-API/invoice_templates/yourcompany_invoice_template.html`

Variables Disponibles:

```
{
    'invoice_number': 12345,
    'date': '2025-01-04',
    'client': {
        'name': 'John Smith',
        'address': {
            'address_line_1': '123 Main St',
            'city': 'Sídney',
            'state': 'NSW',
            'zip_code': '2000',
            'country': 'Australia'
        }
    },
    'transaction_list': [
        [
            {
                'transaction_id': 1,
                'title': 'Servicio Móvil - Cuota Mensual',
                'retail_cost': 30.00,
                'wholesale_cost': 10.00,
                'created': '2025-01-01'
            },
            {
                'transaction_id': 2,
                'title': 'Addon de Datos - 5GB',
                'retail_cost': 15.00,
                'wholesale_cost': 5.00,
                'created': '2025-01-15'
            }
        ]
    ],
    'total_amount': 45.00,
    'due_date': '2025-01-31',
    'start_date': '2025-01-01',
    'end_date': '2025-01-31',
    'paid': False,
    'void': False
}
```

Ejemplo de Fragmento de Plantilla:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Factura {{ invoice_number }}</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 40px;
        }
        .header {
            text-align: center;
            margin-bottom: 30px;
        }
        .invoice-details {
            margin-bottom: 20px;
        }
        table {
            width: 100%;
            border-collapse: collapse;
        }
        th, td {
            border: 1px solid #ddd;
            padding: 12px;
            text-align: left;
        }
        th {
            background-color: #405189;
            color: white;
        }
        .total {
            text-align: right;
            font-size: 18px;
            font-weight: bold;
            margin-top: 20px;
        }
    </style>
</head>
<body>
    <div class="header">
        <h1>FACTURA</h1>
    </div>
```

```
    <div class="invoice-details">
        <p><strong>Número de Factura:</strong> {{ invoice_number
}}</p>
        <p><strong>Fecha:</strong> {{ date }}</p>
        <p><strong>Fecha de Vencimiento:</strong> {{ due_date }}
</p>
        <p><strong>Período de Facturación:</strong> {{ start_date
}} a {{ end_date }}</p>
    </div>

    <div class="customer-details">
        <h3>Facturar a:</h3>
        <p>{{ client.name }}</p>
        <p>{{ client.address.address_line_1 }}</p>
        <p>{{ client.address.city }}, {{ client.address.state }}
{{ client.address.zip_code }}</p>
        <p>{{ client.address.country }}</p>
    </div>

    <table>
        <thead>
            <tr>
                <th>Descripción</th>
                <th>Fecha</th>
                <th>Monto</th>
            </tr>
        </thead>
        <tbody>
            {% for transaction in transaction_list[0] %}
            <tr>
                <td>{{ transaction.title }}</td>
                <td>{{ transaction.created }}</td>
                <td>${{ "%.2f"|format(transaction.retail_cost) }}
</td>
            </tr>
            {% endfor %}
        </tbody>
    </table>

    <div class="total">
        <p>Total a Pagar: ${{ "%.2f"|format(total_amount) }}</p>
    </div>

    {% if paid %}
```

```
    <div style="text-align: center; color: green; font-weight:
bold;">
        PAGADO
    </div>
    {% endif %}

    {% if void %}
    <div style="text-align: center; color: red; font-weight:
bold;">
        ANULADO
    </div>
    {% endif %}
</body>
</html>
```

Mejores Prácticas de Plantilla:

- **Usa rutas absolutas para imágenes** - `file:///absolute/path/to/image.png`
- **CSS en línea** - WeasyPrint no carga hojas de estilo externas de manera confiable
- **Prueba con datos de muestra** - Usa `invoice_templates/rendered/` para inspeccionar HTML
- **Saltos de página** - Usa `<div style="page-break-after: always;"></div>` para facturas de varias páginas
- **Encabezados y pies de página** - Usa reglas CSS `@page` para elementos repetidos
- **Formateo de moneda** - Usa filtros de Jinja2: `{{ "%.2f"|format(amount) }}`

Creando una Plantilla de Factura Personalizada

1. **Copia la Plantilla de Ejemplo:**

```
cd /OmniCRM/OmniCRM-API/invoice_templates
cp sample_invoice_template.html
yourcompany_invoice_template.html
```

2. **Edita la Plantilla:**
    - Reemplaza el nombre de la empresa, logotipo, información de contacto
    - Ajusta el estilo (colores, fuentes, diseño) para que coincida con la marca
    - Agrega o quita secciones según sea necesario (desglose de impuestos, instrucciones de pago, etc.)
3. **Actualiza la Configuración:**

    Edita `crm_config.yaml`:

    ```yaml
    invoice:
      template_filename: 'yourcompany_invoice_template.html'
    ```

4. **Prueba la Generación de Facturas:**
    - Crea una factura de prueba en el CRM
    - Descarga el PDF y verifica el formato
    - Revisa `invoice_templates/rendered/{invoice_id}.html` para errores de depuración
5. **Invalidar Cachés Antiguos (si es necesario):**

    Si has cambiado la plantilla y deseas regenerar facturas existentes:

    ```sql
    -- Limpiar todos los PDFs en caché (forzar regeneración)
    DELETE FROM Invoice_PDF_Cache;
    ```

Sistema de Caché de PDF

Para mejorar el rendimiento, OmniCRM almacena en caché los PDFs generados:

Comportamiento de Caché:

- **Primera Solicitud** - Se genera el PDF, se almacena en caché y se devuelve

- **Solicitudes Subsiguientes** - Se devuelve inmediatamente el PDF en caché (sin regeneración)
- **Invalidación de Caché** - Ocurre cuando la factura se modifica, anula o reembolsa
- **Limpieza de Caché** - Las cachés antiguas se purgan automáticamente después de 30 días de inactividad

Almacenamiento de Caché:

- PDF codificado en Base64 almacenado en la tabla `Invoice_PDF_Cache`
- Hash SHA256 para verificación de integridad
- Incluye nombre de archivo, marca de tiempo de creación, marca de tiempo de último acceso

Gestión Manual de Caché:

```
# En la API de OmniCRM o en el shell de Python
from services.invoice_service import cleanup_old_pdf_cache,
invalidate_invoice_cache
from utils.db_helpers import get_db_session

session = get_db_session()

# Limpiar cachés más antiguas de 30 días
result = cleanup_old_pdf_cache(session, days_old=30)
print(result)  # {'status': 'success', 'deleted_count': 15}

# Invalidar caché de factura específica
invalidate_invoice_cache(session, invoice_id='12345')
```

Puntos Finales de API:

Generar/descargar PDF de factura:

GET /invoice/pdf/{invoice_id}

Respuesta: Descarga de archivo PDF con nombre de archivo del descriptor de estado de Stripe

Encabezados de Caché:

- Primera solicitud: Respuesta más lenta (tiempo de generación)
- Solicitudes en caché: Respuesta instantánea
- El acierto/fallo de caché es transparente para el usuario

Aplicando Cambios de Configuración

Backend (crm_config.yaml)

1. Edita `OmniCRM-API/crm_config.yaml`
2. Guarda los cambios
3. Reinicia el servicio de API:

```
cd OmniCRM-API
sudo systemctl restart omnicrm-api
# o
./restart_api.sh
```

Los cambios entran en efecto inmediatamente después del reinicio.

Frontend (.env)

1. Edita `OmniCRM-UI/.env`
2. Guarda los cambios
3. Reconstruye la UI:

```
cd OmniCRM-UI
npm run build
```

4. Reinicia el servicio de UI o el servidor web

Modo de Desarrollo:

Durante el desarrollo con `npm start`, reinicia el servidor de desarrollo para aplicar cambios.

Mejores Prácticas de Configuración

Seguridad

- **Nunca comprometas secretos** - Usa `.gitignore` para archivos de configuración con credenciales
- **Usa contraseñas fuertes** - Mínimo 16 caracteres con mezcla de mayúsculas, números y símbolos
- **Rota credenciales regularmente** - Especialmente para implementaciones en producción
- **Restringe el acceso a la base de datos** - Usa listas blancas de IP y reglas de firewall
- **Usa configuraciones específicas del entorno** - Configuraciones separadas para dev/staging/producción
- **Limita los permisos de las claves API** - Asigna roles mínimos necesarios
- **Usa la lista blanca de IP con moderación** - Prefiere claves API para mejor seguridad

Mantenimiento

- **Documenta los cambios** - Mantén un registro de cambios de configuración
- **Haz copias de seguridad de las configuraciones** - Almacena copias antes de cambios importantes
- **Prueba en staging** - Verifica cambios de configuración antes de la implementación en producción
- **Control de versiones** - Rastrea plantillas de configuración (sin secretos) en git

Rendimiento

- **Usa base de datos local** - Evita base de datos remota para mejor rendimiento
- **Configura caché** - Habilita caché OCS si está disponible
- **Optimiza Grafana** - Limita el número de paneles incrustados

Branding

- **Coincide colores** - Asegúrate de que los colores de la UI complementen tu logotipo
- **Prueba contraste** - Verifica la legibilidad del texto en fondos coloreados
- **Pruebas en móvil** - Verifica el branding en dispositivos móviles
- **Ubicación del logotipo** - Usa tamaños de logotipo apropiados para diferentes contextos

---

Solución de Problemas

Problemas Comunes

Cambios no aplicados

- **Causa:** Servicio no reiniciado o UI no reconstruida
- **Solución:** Reinicia los servicios de API/UI después de cambios de configuración

Errores de sintaxis YAML

- **Causa:** Formato YAML inválido (indentación, comillas, etc.)
- **Solución:** Valida YAML en línea o usa `yamllint crm_config.yaml`

Conexión a la base de datos fallida

- **Causa:** Credenciales incorrectas o servidor inalcanzable

- **Solución:** Verifica que la base de datos esté en ejecución, las credenciales son correctas

Pagos de Stripe no funcionan

- **Causa:** Claves desajustadas entre backend y frontend
- **Solución:** Asegúrate de que `publishable_key` coincida en ambos archivos

Correos electrónicos no enviados

- **Causa:** Credenciales de Mailjet inválidas o IDs de plantilla
- **Solución:** Verifica la clave/secreto de Mailjet, comprueba que existan los IDs de plantilla

Fallo en la Generación de PDF:

- Verifica que WeasyPrint esté instalado: `pip install weasyprint`
- Verifica que el nombre de la plantilla coincida con `crm_config.yaml`
- Revisa `invoice_templates/rendered/` para errores de renderizado HTML
- Revisa los registros de API para errores de plantilla Jinja2

Imágenes no aparecen en el PDF:

- Usa rutas absolutas: `file:///full/path/to/image.png`
- Asegúrate de que los archivos de imagen existan y sean legibles
- Verifica el formato de imagen (PNG y JPEG funcionan mejor)
- Verifica que las rutas de imagen no contengan caracteres especiales

Problemas de Estilo:

- Inline todo el CSS (no se admiten hojas de estilo externas)
- Evita características CSS complejas (flexbox, grid pueden no renderizar correctamente)
- Prueba con diseños simples primero, agrega complejidad gradualmente
- Usa tablas para el diseño en lugar de divs siempre que sea posible

Caché no se invalida:

- Verifica que `invalidate_invoice_cache()` se llame cuando la factura se modifique
- Verifica que las actualizaciones de transacciones desencadenen la invalidación de caché
- Elimina manualmente de la tabla `Invoice_PDF_Cache` si es necesario

Lista de Verificación de Configuración

Usa esta lista de verificación al implementar OmniCRM:

Configuración de Backend

- [ ] Copia `.env.example` a `.env`
- [ ] Establece contraseñas de base de datos fuertes
- [ ] Configura credenciales de CGRates
- [ ] Actualiza `crm_config.yaml` con tus configuraciones:
  - [ ] Conexión a la base de datos
  - [ ] Tipos de servicio
  - [ ] Claves de API de Mailjet y IDs de plantilla
  - [ ] Correos electrónicos de notificación de fallos de aprovisionamiento
  - [ ] Nombre de archivo de plantilla de factura
  - [ ] URL base de CRM (accesible públicamente)
  - [ ] Puntos finales de OCS/CGRates
  - [ ] Configuración de SMSC
  - [ ] Genera nueva clave secreta JWT
  - [ ] Claves de Stripe (en vivo, no de prueba)
  - [ ] Claves API y lista blanca de IP

Configuración de Frontend

- [ ] Copia `OmniCRM-UI/.env.example` a `OmniCRM-UI/.env`

- [ ] Establece la clave de API de Google Maps
- [ ] Establece la clave publicable de Stripe
- [ ] Actualiza el branding de la empresa:
  - [ ] Nombre de la empresa
  - [ ] Nombre del portal
  - [ ] Nombre de autoservicio
  - [ ] Lema de la empresa
- [ ] Configura la localización:
  - [ ] Idioma predeterminado
  - [ ] Localización
  - [ ] Ubicación y país predeterminados
  - [ ] Código de moneda y símbolo
- [ ] Establece el color principal de la marca
- [ ] Configura integraciones de aplicaciones web (opcional)
- [ ] Agrega URLs de soporte y FAQs (opcional)
- [ ] Establece el ID de seguimiento de Google Analytics/Tag Manager (opcional)

Activos de Branding

- [ ] Crea archivos de logotipo (logoSm.png, logoDark.png, logoLight.png)
- [ ] Sube los logotipos a `OmniCRM-UI/src/assets/images/omnitouch/`
- [ ] Crea plantilla HTML de factura personalizada
- [ ] Sube la plantilla de factura a `OmniCRM-API/invoice_templates/`
- [ ] Actualiza `crm_config.yaml` con el nombre de archivo de la plantilla de factura
- [ ] Prueba la generación de PDF de factura
- [ ] Reconstruye la UI: `npm run build`

Seguridad

- [ ] Cambia todas las contraseñas predeterminadas
- [ ] Genera un secreto JWT único
- [ ] Usa claves de Stripe en producción (no claves de prueba)
- [ ] Rota las claves de API de Mailjet

- [ ] Habilita reglas de firewall
- [ ] Configura la lista blanca de IP para acceso a la API
- [ ] Configura certificados SSL/TLS
- [ ] Habilita HTTPS para todos los puntos finales
- [ ] Revisa configuraciones de CORS
- [ ] Implementa limitación de tasa
- [ ] Configura procedimientos de respaldo y recuperación

Pruebas

- [ ] Prueba el flujo de registro de clientes
- [ ] Prueba el aprovisionamiento de servicios de extremo a extremo
- [ ] Verifica que se envíen correctamente las notificaciones por correo electrónico
- [ ] Prueba la generación de facturas y la descarga de PDF
- [ ] Verifica el procesamiento de pagos (Stripe)
- [ ] Verifica la autenticación de usuarios y 2FA
- [ ] Prueba la suplantación y el registro de auditoría
- [ ] Verifica que los datos de uso se sincronicen desde OCS
- [ ] Prueba la creación y renovación de ActionPlan
- [ ] Confirma que la asignación de inventario funcione correctamente

Implementación

- [ ] Inicia la base de datos (MySQL)
- [ ] Inicia CGRates
- [ ] Inicia la API de OmniCRM
- [ ] Inicia la UI de OmniCRM
- [ ] Configura el proxy inverso (nginx, traefik)
- [ ] Configura monitoreo (Grafana, Prometheus)
- [ ] Configura agregación de registros
- [ ] Configura respaldos automáticos
- [ ] Capacita al personal en el uso del sistema

Documentación Relacionada

- RBAC y Gestión de Usuarios
- Productos y Servicios
- Aprovisionamiento de Ansible
- Gestión de Inventario
- Facturas de Clientes
- Autenticación de Dos Factores
- Atención al Cliente y Suplantación
- *Integraciones de Proveedores de Pago*
- `administration_api_keys` - Gestión de claves API
- `integrations_mailjet` - Integración de correo electrónico
- `concepts_api` - Autenticación de API

Resumen del Inventario en OmniCRM

El sistema de **Inventario** en OmniCRM está diseñado para gestionar y rastrear tanto artículos físicos como virtuales utilizados por operadores de red y clientes.

Esto significa que podemos rastrear todo tipo de artículos, como módems, números de teléfono, bloques de direcciones IP, o incluso hardware físico como ONTs GPON o CPEs de Wireless Fijo.

Véase también: Atributos del Cliente para almacenar metadatos personalizados, y Etiquetas del Cliente para la categorización visual.

Para apoyar a los clientes con una huella de red fija, el sistema de inventario también puede rastrear los hogares pasados para un servicio determinado, permitiendo a los operadores realizar la calificación del servicio de forma remota, y para aquellos que operan una red inalámbrica fija, podemos rastrear los CPEs desplegados en el campo con sus ubicaciones.

> ⓘ **NOTA**
>
> Los artículos de inventario están vinculados a productos durante la provisión a través del campo `inventory_items_list`. Para una guía completa sobre cómo el inventario se integra con la provisión de productos, incluyendo la interfaz de usuario del selector de inventario y la integración del libro de jugadas de Ansible, consulte Guía Completa del Ciclo de Vida del Producto - Requisitos de Inventario.

Propósito

El Inventario de OmniCRM cumple varios propósitos clave:

1. **Provisión de Servicios**: Cuando un cliente se inscribe en un servicio, artículos como módems, tarjetas SIM o números de teléfono pueden

necesitar ser asignados. El sistema de inventario rastrea estos artículos y los asocia con los clientes.

2. **Gestión de Stock**: Para el stock físico, como hardware u otro equipo, el inventario ayuda a los operadores a mantener visibilidad sobre lo que está disponible, dónde se almacena, y qué ha sido asignado o vendido a los clientes.
3. **Asignación a Clientes**: El sistema permite que los artículos sean asignados a los clientes, ya sea para su uso en un servicio (por ejemplo, asignar un módem a la cuenta de internet de un cliente) o para venta directa.
4. **Calificación de Servicio / Huella de Red**: Almacenar información sobre la huella de red, como cada hogar pasado para un servicio GPON, o cada CPE de Wireless Fijo desplegado, permite al personal realizar la calificación del servicio de forma remota y ver si hay una huella de red en un área específica.

Ejemplo del Ciclo de Vida del Inventario

Para ilustrar cómo funciona el sistema de Inventario, considere los siguientes ejemplos

Ejemplo de Tarjeta SIM

Se ordena un lote de 1000 tarjetas SIM de Omnitouch. Primero se crea una Plantilla de Inventario para tarjetas SIM (si no existe ya) y las SIMs ordenadas se cargan en el Inventario en el estado *En Tránsito*.

Una vez que se reciben las SIMs, se marcan como *En Stock*, y pueden ser asignadas a diferentes tiendas minoristas, con la ubicación del artículo de inventario actualizada para reflejar en qué tienda minorista se encuentra cada tarjeta SIM - Esto es útil para rastrear los niveles de stock en cada tienda y asegurar que cada tienda tenga suficiente stock para satisfacer la demanda del cliente.

Cuando un cliente se inscribe en un servicio móvil en la tienda, se asigna una tarjeta SIM al cliente y el estado se cambia a *Asignada*. El artículo de inventario de la tarjeta SIM se asigna al cliente, y el estado se actualiza a *En Uso*.

Si el cliente cancela el servicio o el servicio queda inactivo, la tarjeta SIM se marca como *Desactivada*.

Ejemplo de Hogares Pasados GPON

Para una red GPON, el sistema de inventario puede rastrear cada hogar pasado para un servicio determinado.

Cuando se construye una nueva área, cada dirección pasada puede ser añadida al inventario.

Esto permite a los operadores ver qué hogares están pasados para un servicio determinado, y cuáles hogares aún no están pasados.

Cuando un cliente se inscribe en un servicio, OmniCRM puede ejecutar automáticamente una calificación de servicio contra la dirección del cliente, para ver si la dirección está en el inventario de hogares pasados y qué servicios se pueden ofrecer.

Plantillas de Inventario

La función de **Plantilla de Inventario** permite la creación de cualquier número de tipos de artículos con campos predefinidos. Estas plantillas actúan como planos que definen las características esenciales de diferentes artículos, tales como:

- **Módems** con una dirección MAC (`itemtext1`) y un número de serie (`itemtext2`).
- **Hogares Pasados** para un servicio determinado, con una ubicación y estado (por ejemplo, pasado o no pasado).
- **Números de Teléfono** con un número principal (`itemtext1`) y una ubicación geográfica (`itemtext2`).

- **Recursos Virtuales** como bloques de direcciones IP, con identificadores mapeados a través de la plantilla.

Cada plantilla de inventario define hasta 20 campos de texto personalizables (`itemtext1` a `itemtext20`) con etiquetas correspondientes (`itemtext1_label` a `itemtext20_label`) que describen lo que representa cada campo. Por ejemplo, una plantilla de Módem podría establecer `itemtext1_label` a "Dirección MAC" y `itemtext2_label` a "Número de Serie".

Los operadores pueden personalizar los campos para cada tipo de artículo usando **Plantillas de Inventario**. Estas plantillas permiten que los artículos sean categorizados y gestionados de manera estructurada, asegurando consistencia en cómo se rastrean los artículos.

Vinculación a Productos:

Los nombres de las plantillas de inventario se hacen referencia en las definiciones de productos a través del campo `inventory_items_list`. Al provisionar un producto, el sistema muestra un selector de inventario que solo muestra artículos que coinciden con los tipos de plantilla requeridos.

Ejemplo: Un producto con `inventory_items_list: "['SIM Card', 'Mobile Number']"` requiere que existan dos plantillas de inventario nombradas exactamente "SIM Card" y "Mobile Number". Los nombres de las plantillas son sensibles a mayúsculas.

Para obtener detalles completos sobre cómo las plantillas de inventario se conectan a la provisión de productos, consulte Producto Ciclo de Vida - Requisitos de Inventario.

Creación de Plantillas de Inventario a través de la UI

Para crear una nueva plantilla de inventario:

1. Navegue a **Inventario** → **Plantillas** desde el menú principal
2. Haga clic en el botón **Agregar Plantilla**
3. Complete los campos requeridos:

   Información Básica:

   - **Ícono** (opcional): Nombre de clase del ícono para identificación visual (por ejemplo, `fa-solid fa-sim-card`)
   - **Artículo** (requerido): El nombre de la plantilla (debe coincidir exactamente con lo que se utiliza en `inventory_items_list` para productos)

   Información de Costo (Requerido):

   - **Costo Mayorista** (requerido): Su costo para comprar o provisionar este tipo de artículo
   - **Costo Minorista** (requerido): Precio minorista estándar si se vende por separado a los clientes

   > **NOTA**
   >
   > Los costos mayoristas y minoristas establecidos aquí sirven como valores predeterminados al crear nuevos artículos de inventario a

> partir de esta plantilla. Los artículos de inventario individuales pueden tener diferentes costos si es necesario.

Etiquetas de Campo:

- **Etiqueta de Texto de Artículo 1** (requerido): Etiqueta para el primer campo personalizable (predeterminado a "Número de Modelo")
  - Ejemplos comunes: "ICCID" para tarjetas SIM, "Dirección MAC" para módems, "Número de Teléfono" para números
- **Etiqueta de Texto de Artículo 2** (requerido): Etiqueta para el segundo campo personalizable (predeterminado a "Número de Serie")
  - Ejemplos comunes: "IMSI" para tarjetas SIM, "Número de Serie" para hardware, "Región Geográfica" para números
- **Etiquetas de Texto de Artículo 3-20** (opcional): Etiquetas de campo adicionales según sea necesario
  - Haga clic en **Agregar Campo** para añadir más campos personalizados
  - Solo agregue campos que realmente utilizará para este tipo de artículo

Configuraciones de Visibilidad:

- **Permitir Dropdown Personal**: Habilitar al personal para seleccionar este tipo de inventario en menús desplegables
- **Permitir Dropdown Cliente**: Habilitar a los clientes para ver/seleccionar este tipo de inventario (portal del cliente)

4. Haga clic en **Guardar** para crear la plantilla

Edición de Plantillas de Inventario

Para editar una plantilla existente:

1. Navegue a **Inventario → Plantillas**
2. Encuentre la plantilla en la lista
3. Haga clic en el botón **Editar**
4. Modifique los campos según sea necesario
5. Haga clic en **Guardar**

> **⚠ AVISO**
>
> Cambiar las etiquetas de campo (por ejemplo, `itemtext1_label`) solo afecta a los nuevos artículos creados después del cambio. Los artículos de inventario existentes mantienen sus datos pero se mostrarán con los nuevos nombres de etiqueta.

> **⚠ PRECAUCIÓN**
>
> Los nombres de las plantillas referenciados en los campos `inventory_items_list` de productos son sensibles a mayúsculas. Renombrar una plantilla romperá el enlace a los productos que utilizan el antiguo nombre.

Ejemplos Comunes de Plantillas

Plantilla de Tarjeta SIM:

- Artículo: "SIM Card"
- Costo Mayorista: 2.50
- Costo Minorista: 10.00
- Etiqueta de Texto de Artículo 1: "ICCID"
- Etiqueta de Texto de Artículo 2: "IMSI"
- Etiqueta de Texto de Artículo 3: "Tipo de SIM" (Física/eSIM)

Plantilla de Número Móvil:

- Artículo: "Mobile Number"
- Costo Mayorista: 1.00
- Costo Minorista: 0.00
- Etiqueta de Texto de Artículo 1: "Número de Teléfono"
- Etiqueta de Texto de Artículo 2: "Región Geográfica"
- Etiqueta de Texto de Artículo 3: "Tipo de Número" (Móvil/Fijo)

Plantilla de CPE de Wireless Fijo:

- Artículo: "Fixed Wireless CPE"

- Costo Mayorista: 250.00
- Costo Minorista: 450.00
- Etiqueta de Texto de Artículo 1: "Dirección MAC"
- Etiqueta de Texto de Artículo 2: "Número de Serie"
- Etiqueta de Texto de Artículo 3: "Versión de Firmware"
- Etiqueta de Texto de Artículo 4: "Fabricante"
- Etiqueta de Texto de Artículo 5: "Modelo"

Plantilla de GPON ONT:

- Artículo: "GPON ONT"
- Costo Mayorista: 45.00
- Costo Minorista: 0.00 (incluido con el servicio)
- Etiqueta de Texto de Artículo 1: "Número de Serie"
- Etiqueta de Texto de Artículo 2: "Dirección MAC"
- Etiqueta de Texto de Artículo 3: "Ubicación PON"
- Etiqueta de Texto de Artículo 4: "Modelo"

Creación y Gestión de Artículos de Inventario

Una vez que se define una **Plantilla de Inventario**, se pueden crear artículos de **Inventario** individuales. Cada artículo de inventario representa una

instancia específica de un tipo de artículo (por ejemplo, un módem o número de teléfono específico) que puede ser:

- **Asignado a Clientes**: Los artículos están vinculados a clientes para la provisión de servicios (por ejemplo, asignar hardware para una conexión a internet).
- **Rastreado para Stock**: Los operadores pueden monitorear el inventario disponible, como artículos no vendidos o no asignados.
- **Vendidos o Desactivados**: Una vez vendidos, los artículos se marcan con marcas de tiempo relevantes (por ejemplo, `sold_date`) y ya no pueden considerarse stock disponible.

A través de este sistema, OmniCRM facilita la gestión eficiente del stock, ayuda a asignar recursos a los clientes y proporciona visibilidad detallada sobre el estado y la historia de cada artículo.

Servicios pueden estar vinculados a un artículo de **Inventario** dado, permitiendo un seguimiento fácil de qué artículos están asociados con qué clientes o servicios.

Una vez que un artículo de inventario ha sido asignado a un cliente, los juegos de Ansible pueden actualizar el estado y la historia del artículo para reflejar la asignación. Esto asegura que los operadores tengan un registro preciso de qué artículos están en uso y cuáles están disponibles para asignación, así como saber qué cliente está utilizando qué artículo.

Podemos ver los artículos asignados a un cliente desde la página de perfil del cliente en la pestaña de **Inventario**.

Para los artículos de Inventario vinculados a un **Servicio**, podemos ver eso editando el servicio, para ver los artículos de Inventario vinculados.

Campos del Artículo de Inventario

Cada artículo de inventario contiene información completa organizada en varias categorías:

Información Básica del Artículo

- **inventory_id** - Identificador único para el artículo de inventario (auto-generado)
- **item** - Tipo de artículo (coincide con el nombre de la Plantilla de Inventario, por ejemplo, "SIM Card", "Modem", "Phone Number")
- **inventory_template_id** - Enlace a la Plantilla de Inventario que define este tipo de artículo
- **customer_id** - Si está asignado a un cliente, el ID del cliente (nulo)
- **service_id** - Si está vinculado a un servicio específico, el ID del servicio (nulo)
- **item_location** - Ubicación física o lógica del artículo:

    - Para artículos físicos: edificio, almacén, ubicación en estante, nombre de la tienda, etc.
    - Para artículos virtuales: ubicación geográfica, ubicación de bloque IP, región de rango de números, etc.
- **item_state** - Estado actual del artículo de inventario (valores enumerados):
    - **Nuevo** - Artículo completamente nuevo, sin usar
    - **Usado** - Anteriormente usado pero funcional
    - **Uso Interno** - Asignado para pruebas internas o uso del personal
    - **Asignado** - Actualmente asignado a un cliente o servicio
    - **Dañado** - No funcional, requiere reparación o eliminación
    - **Fuera de Servicio** - Temporalmente no disponible
    - **Perdido** - El artículo no puede ser localizado
    - **Robado** - El artículo fue robado

Campos Personalizables del Artículo (de la Plantilla)

El sistema de inventario admite hasta 20 campos de texto personalizables cuyo significado está definido por la Plantilla de Inventario:

- **itemtext1** - Primer campo personalizable (requerido, etiqueta definida por `itemtext1_label` de la plantilla)
    - Ejemplo: Para módems, podría ser "Dirección MAC"
    - Ejemplo: Para tarjetas SIM, podría ser "ICCID"
    - Ejemplo: Para números de teléfono, podría ser "Número de Teléfono"
- **itemtext2** a **itemtext20** - Campos personalizables adicionales (opcional, etiquetas definidas por la plantilla)
    - Ejemplo: itemtext2 para módems podría ser "Número de Serie"
    - Ejemplo: itemtext2 para tarjetas SIM podría ser "IMSI"
    - Ejemplo: itemtext3 para módems podría ser "Versión de Firmware"

Cada Plantilla de Inventario define cuáles de estos campos se utilizan y qué representan a través de los campos de etiqueta correspondientes (`itemtext1_label`, `itemtext2_label`, etc.).

Información de Costo

- **wholesale_cost** - Su costo para comprar/provisionar este artículo (float)
- **retail_cost** - Precio cobrado al cliente si se vende por separado (float)
- **sold_date** - Marca de tiempo cuando el artículo fue vendido o asignado al cliente

Dirección Física (para Equipos de Red y Sitios)

Utilizada para rastrear ubicaciones de despliegue físico, particularmente para equipos de red fijos (CPEs, ONTs, módems) o hogares pasados:

- **address_line_1** - Dirección de la calle, número de edificio, número de unidad
- **address_line_2** - Información adicional de la dirección (suite, apartamento, piso)
- **city** - Ciudad o pueblo
- **state** - Estado, provincia o región
- **zip_code** - Código postal
- **country** - Nombre del país

Ubicación Geográfica (Autocompletado desde la UI Web)

Al crear artículos de inventario a través de la UI Web con autocompletado de dirección, estos campos se completan automáticamente:

- **google_maps_place_id** - ID de Lugar de Google Maps para la dirección
- **plus_code** - Código Plus de Google Maps (Código de Ubicación Abierto) para ubicación precisa
- **latitude** - Latitud geográfica (grados decimales)
- **longitude** - Longitud geográfica (grados decimales)

Estos campos permiten:

- Mapeo de ubicaciones de inventario en una vista de mapa
- Cálculos de proximidad para calificación de servicio
- Análisis de cobertura para planificación de red
- Optimización de rutas para el despacho de técnicos de campo

Gestión de Dispositivos y URLs de Acceso

El campo `management_url` proporciona acceso rápido a interfaces de dispositivos y URLs de provisión:

- **management_url** - URL de acceso para el artículo de inventario
  - **Equipos de Red**: URL de interfaz web (por ejemplo, `https://192.168.1.1` para enrutadores, conmutadores, ONTs, CPEs)
  - **Perfiles eSIM**: Dirección LPA (Asistente de Perfil Local) para activación de eSIM (por ejemplo, `LPA:1$smdp.example.com$ACTIVATION-CODE-HERE`)
  - **Otros Casos de Uso**: Cualquier URL que necesite ser fácilmente accesible a través de un dispositivo móvil

  #### Generación de Código QR

  Al ver artículos de inventario con un `management_url`, el sistema genera automáticamente un **código QR escaneable**:

  - **Vista de detalle del artículo de inventario**: código QR de 128x128 mostrado junto a la URL
  - **Tabla de inventario de servicio**: código QR de 64x64 mostrado para artículos asignados
  - **Formato**: Tanto el código QR como el hipervínculo clicable se muestran juntos

  #### Casos de Uso Comunes

  - **Técnicos de Red**: Escanear el código QR para acceder a la interfaz de gestión del dispositivo sin escribir direcciones IP
  - **Activación de eSIM**: Los clientes escanean el código QR desde el CRM para instalar el perfil eSIM en su dispositivo
  - **Autoservicio del Cliente**: Proporcionar acceso fácil a la configuración del dispositivo o portales para clientes

- **management_username** - Nombre de usuario de administrador para acceso al dispositivo

- **management_password** - Contraseña de administrador para acceso al dispositivo (encriptada en reposo)

Gestión de Configuración

Para dispositivos con archivos de configuración:

- **config_content** - Contenido completo del archivo de configuración (almacenado como texto)
  - Útil para respaldo, versionado y recuperación ante desastres
  - Puede almacenar configuraciones de enrutadores, configuraciones de conmutadores, configuraciones de CPE, etc.
- **config_file_path** - Ruta al archivo de configuración externo si se almacena por separado
  - Alternativa a almacenar la configuración completa en la base de datos
  - Ruta a un recurso compartido de red, repositorio de control de versiones o sistema de gestión de configuraciones

Notas y Metadatos

- **inventory_notes** - Notas en formato libre sobre el artículo de inventario
  - Notas de instalación
  - Historial de mantenimiento
  - Particularidades o requisitos de manejo especiales
  - Información del proveedor
  - Detalles de la garantía
- **created** - Marca de tiempo cuando se creó el artículo de inventario en el sistema (auto-establecido)
- **last_modified** - Marca de tiempo de la última actualización del artículo de inventario (auto-actualizado)

Ejemplos de Uso de Campos

Ejemplo 1: Tarjeta SIM Móvil

```
{
  "inventory_id": 1001,
  "item": "SIM Card",
  "inventory_template_id": 5,
  "itemtext1": "89612345678901234 56",
  "itemtext2": "310120123456789",
  "itemtext3": "Physical",
  "item_location": "Warehouse A, Shelf 3",
  "item_state": "Assigned",
  "customer_id": 456,
  "service_id": 789,
  "wholesale_cost": 2.50,
  "retail_cost": 10.00,
  "sold_date": "2025-01-15T10:30:00Z",
  "inventory_notes": "Activated on 2025-01-15"
}
```


Ejemplo 2: Perfil eSIM Móvil


```
{
  "inventory_id": 1002,
  "item": "eSIM",
  "inventory_template_id": 6,
  "itemtext1": "89612345678901234 57",
  "itemtext2": "310120123456790",
  "itemtext3": "eSIM",
  "item_location": "Virtual Inventory",
  "item_state": "Assigned",
  "customer_id": 457,
  "service_id": 790,
  "management_url": "LPA:1$smdp.example.com$ACTIVATION-CODE-
ABC123XYZ",
  "wholesale_cost": 0.00,
  "retail_cost": 0.00,
  "sold_date": "2025-01-16T14:20:00Z",
  "inventory_notes": "eSIM profile ready for activation"
}
```


Al ver este artículo de inventario eSIM, la interfaz muestra un código QR que contiene la dirección LPA. Los clientes escanean este código QR con su

dispositivo móvil para instalar el perfil eSIM.

Ejemplo 3: Equipos de Cliente (CPE) - Wireless Fijo


```
{
  "inventory_id": 2001,
  "item": "Fixed Wireless CPE",
  "inventory_template_id": 10,
  "itemtext1": "AA:BB:CC:DD:EE:FF",
  "itemtext2": "FW2024-12345",
  "itemtext3": "v2.4.1",
  "itemtext4": "Ubiquiti",
  "itemtext5": "LiteBeam AC Gen2",
  "item_location": "Customer Site",
  "item_state": "Assigned",
  "customer_id": 789,
  "service_id": 1234,
  "address_line_1": "123 Main Street",
  "address_line_2": "Apt 4B",
  "city": "Sydney",
  "state": "NSW",
  "zip_code": "2000",
  "country": "Australia",
  "latitude": "-33.8688",
  "longitude": "151.2093",
  "management_url": "https://192.168.100.1",
  "management_username": "admin",
  "management_password": "encrypted_password_here",
  "config_file_path": "/configs/cpe/fw2024-12345.conf",
  "inventory_notes": "Installed 2025-01-10. Customer reports
excellent signal. Pointing: Azimuth 45°, Elevation 15°"
}
```


Ejemplo 4: GPON ONT con Dirección Completa

```
{
  "inventory_id": 3001,
  "item": "GPON ONT",
  "inventory_template_id": 15,
  "itemtext1": "ALCL12345678",
  "itemtext2": "AA:BB:CC:DD:EE:FF",
  "itemtext3": "OLT-1, PON 3, ONT 42",
  "itemtext4": "Nokia G-010G-A",
  "item_location": "Customer Premises",
  "item_state": "Assigned",
  "customer_id": 321,
  "service_id": 654,
  "address_line_1": "456 Fiber Lane",
  "city": "Melbourne",
  "state": "VIC",
  "zip_code": "3000",
  "country": "Australia",
  "google_maps_place_id": "ChIJ1234567890",
  "plus_code": "4RRH+2C Melbourne VIC",
  "latitude": "-37.8136",
  "longitude": "144.9631",
  "management_url": "https://192.168.1.1",
  "management_username": "admin",
  "config_content": "# ONT Configuration\nwlan-ssid:
HomeNetwork\nwlan-password: encrypted...",
  "wholesale_cost": 45.00,
  "retail_cost": 0.00,
  "inventory_notes": "Provisioned 2025-01-20. Optical power:
-22dBm"
}
```


NOTA

Al ver artículos de inventario con un `management_url` (como los Ejemplos 2, 3 y 4 anteriores), la interfaz muestra automáticamente:

- Un código QR escaneable que contiene la URL o dirección LPA
- Un hipervínculo clicable (para URLs web)

Casos de Uso:

- **Activación de eSIM** (Ejemplo 2): Los clientes escanean el código QR para instalar el perfil eSIM en su dispositivo
- **Acceso a Equipos de Red** (Ejemplos 3 y 4): Los técnicos escanean para acceder a las interfaces de gestión de dispositivos sin escribir manualmente las direcciones IP

Ejemplo 5: Número de Teléfono (Inventario Virtual)


```
{
  "inventory_id": 4001,
  "item": "Phone Number",
  "inventory_template_id": 20,
  "itemtext1": "+61412345678",
  "itemtext2": "Melbourne",
  "itemtext3": "Mobile",
  "item_location": "Australia - VIC",
  "item_state": "Assigned",
  "customer_id": 555,
  "service_id": 888,
  "wholesale_cost": 1.00,
  "retail_cost": 0.00,
  "inventory_notes": "Ported from Telstra on 2025-01-05"
}
```


Estados del Artículo de Inventario Explicados

El campo `item_state` rastrea el ciclo de vida de los artículos de inventario:

- **Nuevo** → **Asignado** - Flujo normal al provisionar a un cliente
- **Asignado** → **Usado** - Después de la desactivación del servicio, el artículo puede ser reutilizado
- **Nuevo** → **Uso Interno** - Asignado para pruebas, demostraciones o uso del personal
- **Asignado** → **Dañado** - El artículo falló, requiere RMA o eliminación
- **Cualquier Estado** → **Perdido** - El artículo no puede ser localizado (dispara auditoría)
- **Cualquier Estado** → **Robado** - El artículo fue robado (dispara informe de seguridad)

- **Dañado/Usado → Nuevo** - Después de reacondicionamiento o reparación

Filtrar y buscar inventario por estado permite a los operadores:

- Rastrear stock disponible (Artículos Nuevos)
- Identificar artículos asignados a clientes (Asignados)
- Encontrar artículos disponibles para reutilización (Usados)
- Monitorear problemas de equipos (Dañados, Fuera de Servicio)
- Auditar artículos faltantes (Perdidos, Robados)

Etiquetas de Clientes

Las etiquetas son enlaces codificados por colores que se pueden agregar a un cliente para ayudar a categorizarlos; por ejemplo, un cliente podría tener una etiqueta para "Ticket de Soporte Abierto" o "Factura Muy Atrasada" o "Desagradable".

Para almacenar metadatos estructurados y datos personalizados de clave-valor, consulte Atributos del Cliente.

Las etiquetas se muestran como píldoras en la página de perfil del cliente, y el color de la píldora es personalizable junto con el enlace.

Un caso de uso común es etiquetar a los clientes que tienen un ticket de soporte abierto, para que el equipo de soporte pueda saltar fácilmente al ticket abierto desde la página de perfil del cliente.

Las etiquetas pueden ser creadas en el sistema por un administrador a través de la interfaz de usuario o por sistemas de terceros a través de la API y pueden tener fechas de inicio y finalización, por lo que pueden ser eliminadas automáticamente después de un cierto período.

Gestión de Etiquetas a través de la Interfaz de Usuario

Visualización de Etiquetas de Clientes

Para ver las etiquetas de un cliente:

1. Navegue a la página de resumen del cliente
2. Haga clic en la pestaña **Etiquetas**
3. Verá una lista de todas las etiquetas activas para el cliente, mostrando:
    - Vista previa de la etiqueta con el color configurado
    - Texto de la etiqueta
    - Fecha activa (cuando la etiqueta se vuelve visible)
    - Fecha de desactivación (cuando la etiqueta se ocultará)
    - Enlace (si está configurado)

Creación de una Nueva Etiqueta

Para crear una nueva etiqueta para un cliente:

1. Navegue a la página de resumen del cliente
2. Haga clic en la pestaña **Etiquetas**
3. Haga clic en el botón **Agregar Etiqueta**
4. Complete los campos requeridos:
    - **Texto de la Etiqueta** (requerido): El texto que se mostrará en la etiqueta
    - **Color de la Etiqueta** (requerido): Elija un color usando el selector de color o ingrese un código hexadecimal
    - **Enlace de la Etiqueta** (opcional): URL que se abrirá cuando se haga clic en la etiqueta
    - **Fecha Activa** (requerido): Fecha en la que la etiqueta debe comenzar a mostrarse
    - **Fecha de Desactivación** (requerido): Fecha en la que la etiqueta debe dejar de mostrarse (por defecto es 2099-01-01)

5. Vista previa de su etiqueta en la sección de vista previa
6. Haga clic en **Crear Etiqueta**

Edición de una Etiqueta

Para editar una etiqueta existente:

1. Navegue a la página de resumen del cliente
2. Haga clic en la pestaña **Etiquetas**
3. Encuentre la etiqueta que desea editar en la lista
4. Haga clic en el botón **Editar** (lápiz)
5. Modifique los campos según sea necesario
6. Haga clic en **Actualizar Etiqueta**

Eliminación de una Etiqueta

Para eliminar una etiqueta:

1. Navegue a la página de resumen del cliente
2. Haga clic en la pestaña **Etiquetas**
3. Encuentre la etiqueta que desea eliminar en la lista
4. Haga clic en el botón **Eliminar** (papelera)
5. Confirme la eliminación en el popup

Referencia de Campos de Etiquetas

Integración de API

Las etiquetas también se pueden gestionar programáticamente a través de la API:

Crear una Etiqueta:

```
PUT /crm/tag/
{
  "tag_text": "Cliente VIP",
  "tag_hex_color": "FFD700",
  "tag_link": "https://example.com/vip",
  "tag_active_date": "2025-01-01 00:00:00",
  "tag_deactivate_date": "2099-12-31 23:59:59",
  "customer_id": 12
}
```

Actualizar una Etiqueta:

```
PATCH /crm/tag/tag_id/{tag_id}
{
  "tag_text": "Texto de Etiqueta Actualizado",
  "tag_hex_color": "FF0000"
}
```

Obtener Etiquetas por Cliente:

```
GET /crm/tag/customer_id/{customer_id}
```

Eliminar una Etiqueta:

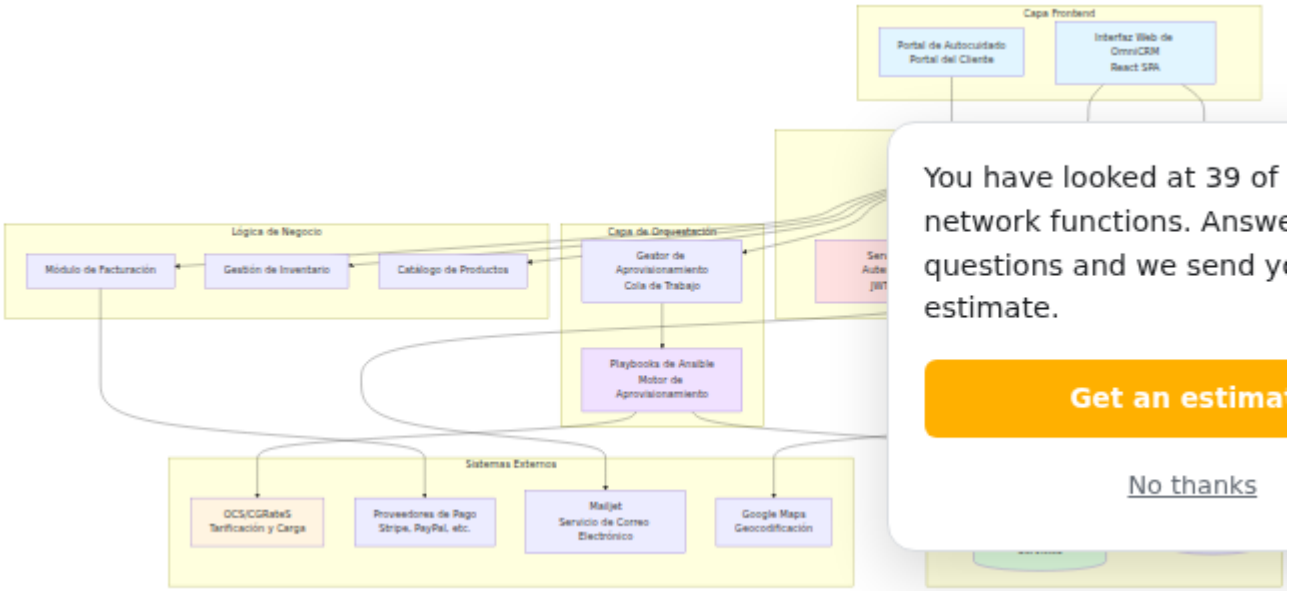
```
DELETE /crm/tag/tag_id/{tag_id}
```

Arquitectura del Sistema OmniCRM

Este documento proporciona una visión general de la arquitectura del sistema OmniCRM, incluyendo relaciones de componentes y flujo de datos.

Visión General del Sistema a Alto Nivel

OmniCRM es una plataforma integral BSS/OSS que integra varios componentes clave para proporcionar una gestión completa de servicios para proveedores de telecomunicaciones.



Componentes Clave

1. Aplicaciones Frontend

Interfaz Web de OmniCRM

- Aplicación de una sola página en React

- Interfaz para el personal para la gestión de clientes, aprovisionamiento de servicios, facturación
- Actualizaciones de estado de aprovisionamiento en tiempo real
- Control de acceso basado en roles

Portal de Autocuidado

- Portal orientado al cliente
- Gestión de servicios y seguimiento de uso
- Visualización y pago de facturas
- Código compartido con la interfaz del personal, diferentes vistas

2. Capa API

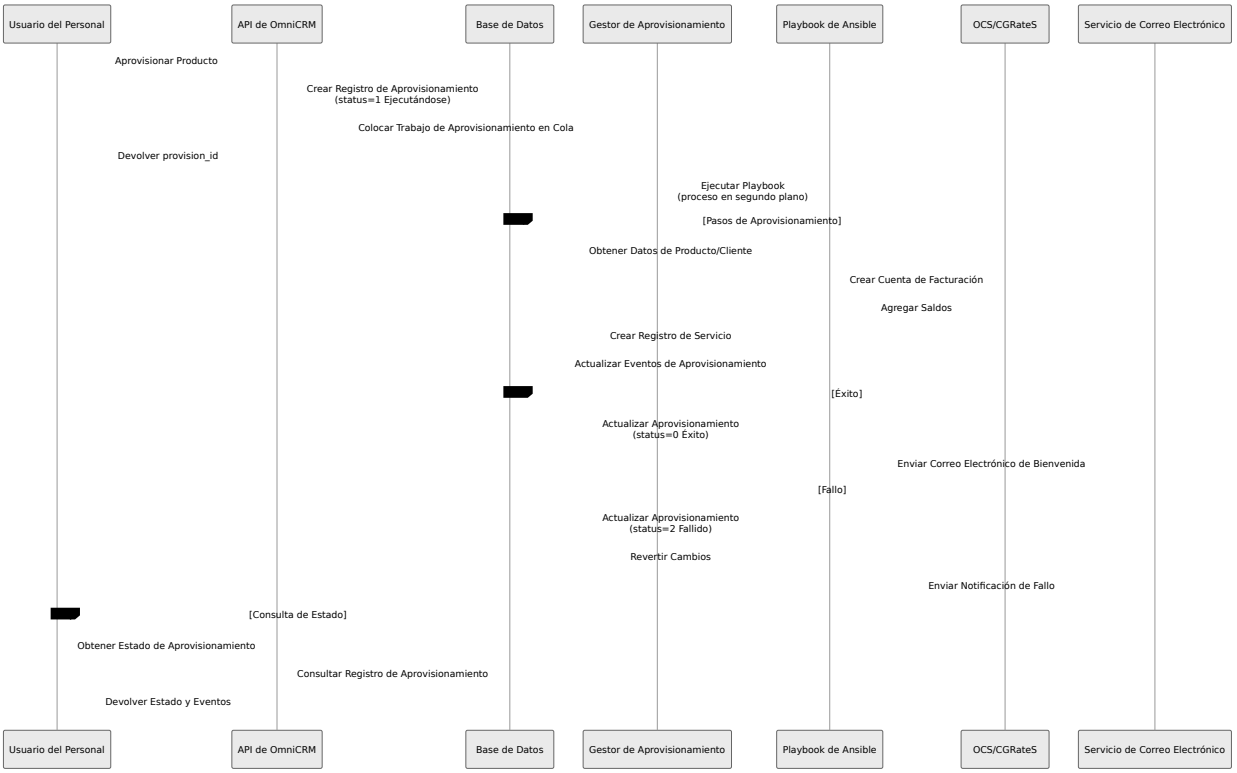
API de OmniCRM (Flask/Python)

- API RESTful para todas las operaciones
- Documentación OpenAPI/Swagger
- Autenticación basada en JWT
- Limitación de tasa y caché
- Soporte para WebSocket para actualizaciones en tiempo real

Rutas Clave de la API:

- `/crm/customer/*` - Gestión de clientes
- `/crm/service/*` - Operaciones de servicio
- `/crm/product/*` - Catálogo de productos
- `/crm/provision/*` - Operaciones de aprovisionamiento
- `/crm/transaction/*` - Transacciones de facturación
- `/crm/invoice/*` - Gestión de facturas

3. Sistema de Aprovisionamiento

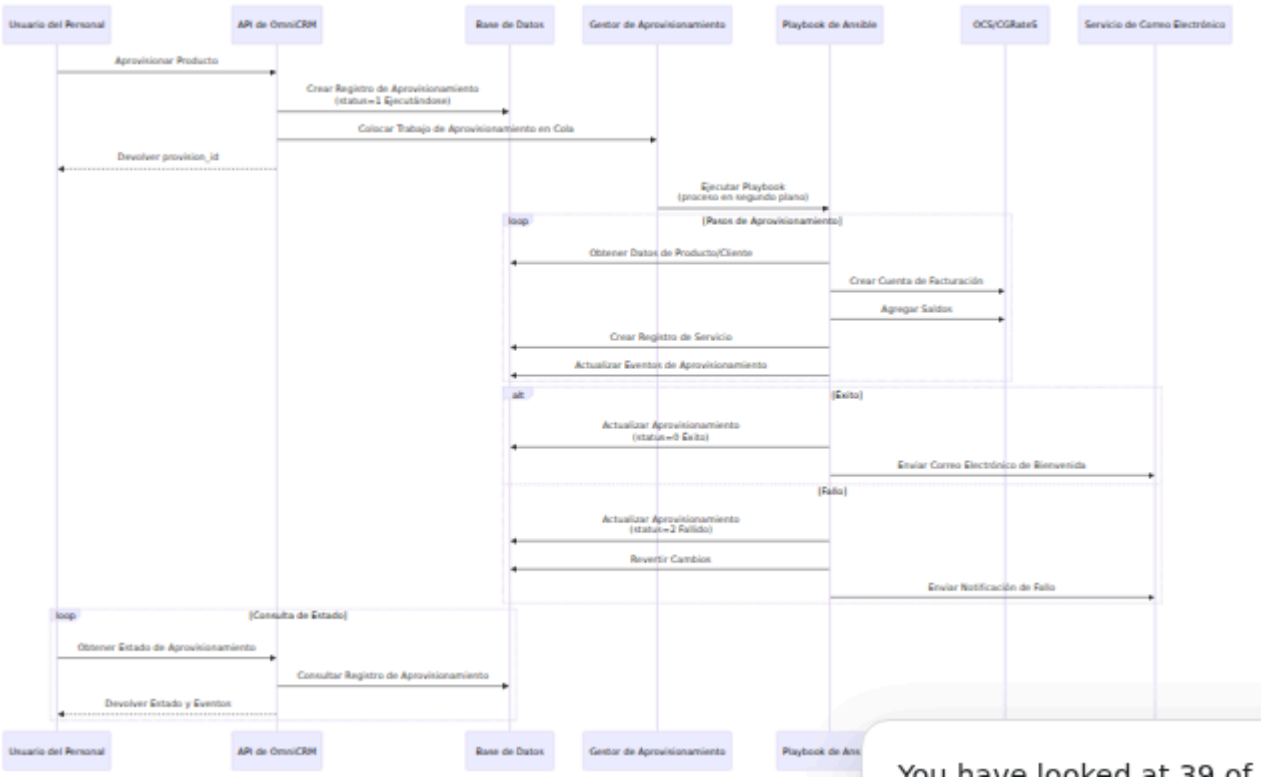


4. Facturación y Tarificación

Integración OCS/CGRateS

- Tarificación y carga en tiempo real
- Gestión de cuentas
- Seguimiento de saldos (monetario, datos, voz, SMS)
- Planes de acción para cargos recurrentes
- Notificaciones basadas en umbrales

Flujo de Trabajo de Facturación:



5. Modelo de Datos

OmniCRM utiliza una base de datos relacional con los siguientes modelos centrales. Para diagramas de relación de entidades visuales, consulte Clientes, Contactos, Sitios y Servicios.

Cliente y Entidades Relacionadas

Cliente - Entidad central que representa una empresa o individuo

Campo	Tipo	Descripción
customer_id	Entero (PK)	Identificador único
customer_name	Cadena	Nombre de la empresa o individuo
customer_account_type	Enum	'Individual' o 'Negocio'
customer_status	Enum	'Abierto', 'Cerrado', 'Suspendido', 'Archivado'
customer_payment_type	Cadena	'prepago' o 'postpago'
customer_enabled	Booleano	¿Está activa la cuenta?
tax_identifier	Cadena	Número de IVA/GST
contract_start_date	FechaHora	Inicio del contrato
contract_end_date	FechaHora	Fin del contrato

Contacto - Personas asociadas a un cliente

Campo	Tipo	Descripción
contact_id	Entero (PK)	Identificador único
customer_id	Entero (FK)	Cliente padre
contact_firstname	Cadena	Nombre
contact_lastname	Cadena	Apellido
contact_email	Cadena	Dirección de correo electrónico
contact_phone	Cadena	Número de teléfono
contact_types	Cadena	'Primario', 'Facturación', 'Técnico'

Sitio - Ubicaciones físicas de entrega de servicios

Campo	Tipo	Descripción
site_id	Entero (PK)	Identificador único
customer_id	Entero (FK)	Cliente padre
site_name	Cadena	Nombre de la ubicación
address_line_1	Cadena	Dirección de la calle
city, state, zip_code	Cadena	Detalles de la ubicación
latitude, longitude	Flotante	Coordenadas GPS
google_maps_place_id	Cadena	Referencia de Google Maps
plus_code	Cadena	Código de Ubicación Abierto

Modelos de Servicio y Producto

Servicio - Instancias de servicio activas

Campo	Tipo	Descripción
service_id	Entero (PK)	Identificador único
customer_id	Entero (FK)	Cliente padre
product_id	Entero (FK)	Plantilla de producto
site_id	Entero (FK)	Ubicación del servicio
service_name	Cadena	Nombre para mostrar
service_uuid	Cadena	Identificador del sistema de facturación
service_status	Enum	Estado actual
service_billed	Booleano	Generar transacciones
wholesale_cost	Flotante	Costo del proveedor
retail_cost	Flotante	Precio para el cliente
bundled_parent	Entero (FK)	Servicio padre para paquetes

Producto - Plantillas de ofertas de servicio

Campo	Tipo	Descripción
product_id	Entero (PK)	Identificador único
product_name	Cadena	Nombre para mostrar
product_slug	Cadena	Nombre amigable para URL
category	Enum	'independiente', 'paquete', 'complemento', 'promoción'
provisioning_play	Cadena	Nombre del playbook de Ansible
provisioning_json_vars	JSON	Variables del playbook
inventory_items_list	Cadena	Inventario requerido
retail_cost	Flotante	Precio mensual
retail_setup_cost	Flotante	Tarifa única
enabled	Booleano	Disponible para la venta

Modelos de Facturación

Transacción - Cargos/créditos individuales

Campo	Tipo	Descripción
transaction_id	Entero (PK)	Identificador único
customer_id	Entero (FK)	Cliente padre
invoice_id	Entero (FK)	Factura padre (opcional)
service_id	Entero (FK)	Servicio relacionado
title	Cadena	Descripción de la transacción
retail_cost	Flotante	Monto (negativo = crédito/pago)
payment_type	Cadena	p.ej. `wallet_topup_card`, `wallet_invoice`, `postpaid_deferred_charge`
wallet_ledger_id	Entero (FK)	Enlaces a wallet_ledger cuando la fila refleja movimiento de wallet
void	Booleano	Transacción cancelada

Factura - Transacciones agrupadas para facturación

Campo	Tipo	Descripción
invoice_id	Entero (PK)	Identificador único
customer_id	Entero (FK)	Cliente padre
paid	Booleano	Pago recibido
void	Booleano	Factura cancelada
payment_reference	Cadena	ID de transacción de Stripe
start_date, end_date	Fecha	Período de facturación
due_date	Fecha	Fecha límite de pago
retail_cost	Flotante	Monto total

Modelos de Inventario

Inventario - Activos físicos y virtuales

Campo	Tipo	Descripción
inventory_id	Entero (PK)	Identificador único
customer_id	Entero (FK)	Cliente asignado
service_id	Entero (FK)	Servicio vinculado
inventory_template_id	Entero (FK)	Plantilla de tipo de artículo
item	Cadena	Tipo de artículo (Tarjeta SIM, Router, etc.)
item_state	Enum	'Nuevo', 'Asignado', 'Usado', etc.
itemtext1-20	Cadena	Campos flexibles
management_url	Cadena	URL de administración del equipo
config_content	Texto	Archivo de configuración

Inventory_Template - Define la estructura del artículo de inventario

Campo	Tipo	Descripción
inventory_template_id	Entero (PK)	Identificador único
item	Cadena	Nombre de la plantilla
itemtext1_label	Cadena	Etiqueta para el campo itemtext1
itemtext2_label	Cadena	Etiqueta para el campo itemtext2

Modelos de Aprovisionamiento

Aprovisionar - Seguimiento de trabajos de aprovisionamiento

Campo	Tipo	Descripción
provision_id	Entero (PK)	Identificador único
product_id	Entero (FK)	Producto que se está aprovisionando
customer_id	Entero (FK)	Cliente objetivo
service_id	Entero (FK)	Servicio creado/modificado
provisioning_play	Cadena	Nombre del playbook de Ansible
provisioning_status	Entero	0=Éxito, 1=Ejecutándose, 2=Fallido

Provision_Event - Pasos individuales de aprovisionamiento

Campo	Tipo	Descripción
provision_event_id	Entero (PK)	Identificador único
provision_id	Entero (FK)	Trabajo de aprovisionamiento padre
event_name	Cadena	Nombre de la tarea
event_number	Entero	Número de secuencia
provisioning_status	Entero	0=Éxito, 1=Ejecutándose, 2=Fallido
provisioning_result_json	JSON	Salida completa de la tarea

Modelos de Usuario y Seguridad

Usuario - Cuentas de usuario

Campo	Tipo	Descripción
id	Entero (PK)	Identificador único
username	Cadena	Nombre de usuario para iniciar sesión
email	Cadena	Dirección de correo electrónico
email_verified	Booleano	Correo electrónico confirmado
is_2fa_enabled	Booleano	Autenticación de dos factores habilitada
totp_secret	Cadena	Clave secreta TOTP

Rol - Roles de usuario

Campo	Tipo	Descripción
id	Entero (PK)	Identificador único
name	Cadena	Nombre del rol
description	Cadena	Descripción del rol

Permiso - Permisos granulares

Campo	Tipo	Descripción
id	Entero (PK)	Identificador único
name	Cadena	Nombre del permiso (recurso.acción)
description	Cadena	Descripción del permiso

Relaciones:

- Los usuarios tienen muchos roles (muchos a muchos)
- Los roles tienen muchos permisos (muchos a muchos)
- Los usuarios pueden vincularse a un contacto (para acceso al portal del cliente)

Puntos de Integración

Pasarela de Pago Stripe

- Métodos de pago tokenizados
- Almacenamiento de tarjetas conforme a PCI
- Pago automático de facturas
- Procesamiento de reembolsos
- Notificaciones de tarjetas que expiran

Servicio de Correo Electrónico Mailjet

- Correos electrónicos transaccionales (facturas, bienvenida, notificaciones)
- Sincronización de contactos
- Gestión de plantillas
- Seguimiento de entregas

Google Maps

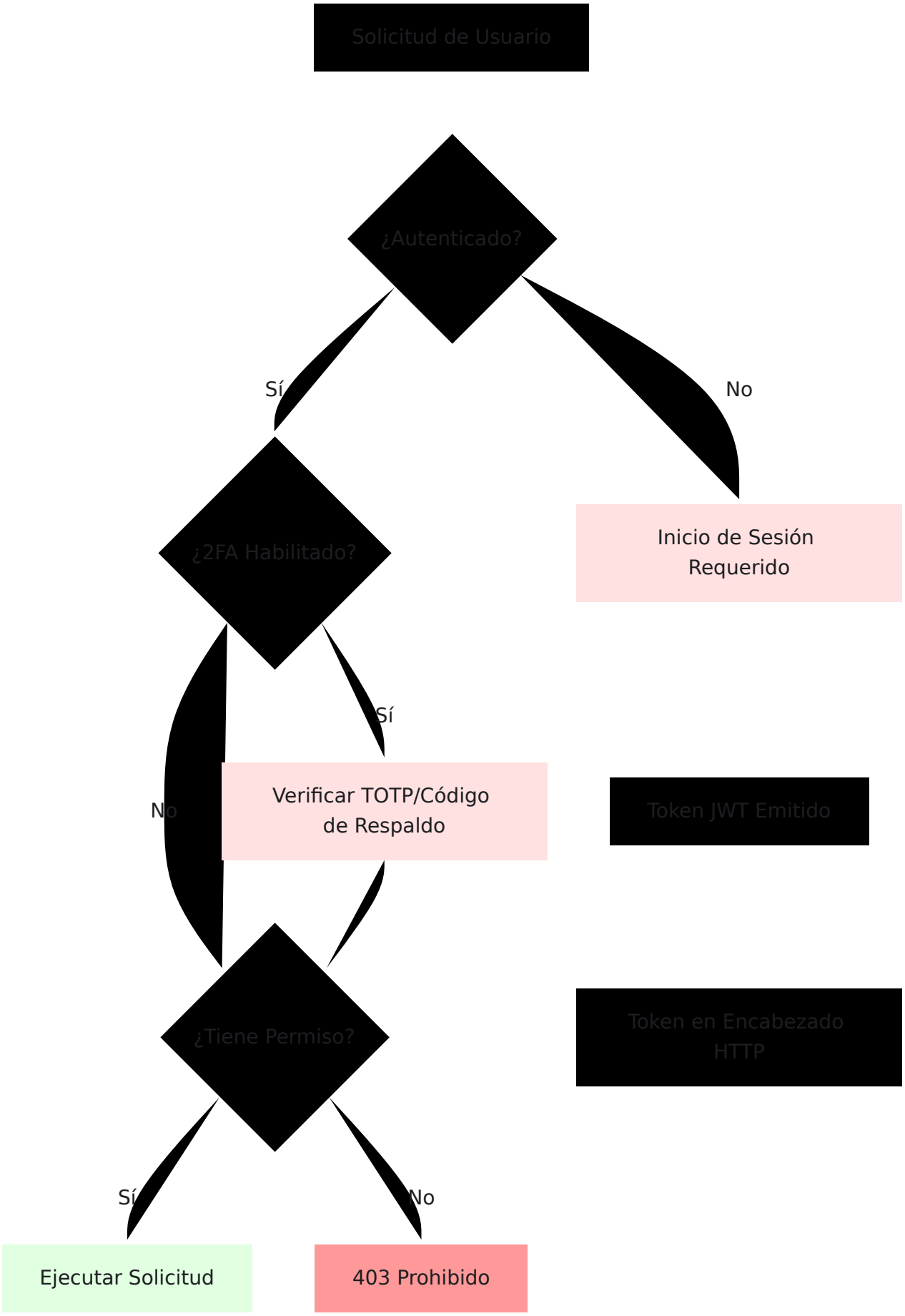
- Autocompletar direcciones
- Geocodificación y geocodificación inversa
- Generación de Plus Code
- Mapeo de ubicación de sitios

OCS/CGRateS

- Aprovisionamiento de cuentas

- Tarificación en tiempo real
- Gestión de saldos
- Procesamiento de CDR
- Planes de acción y horarios

Arquitectura de Seguridad

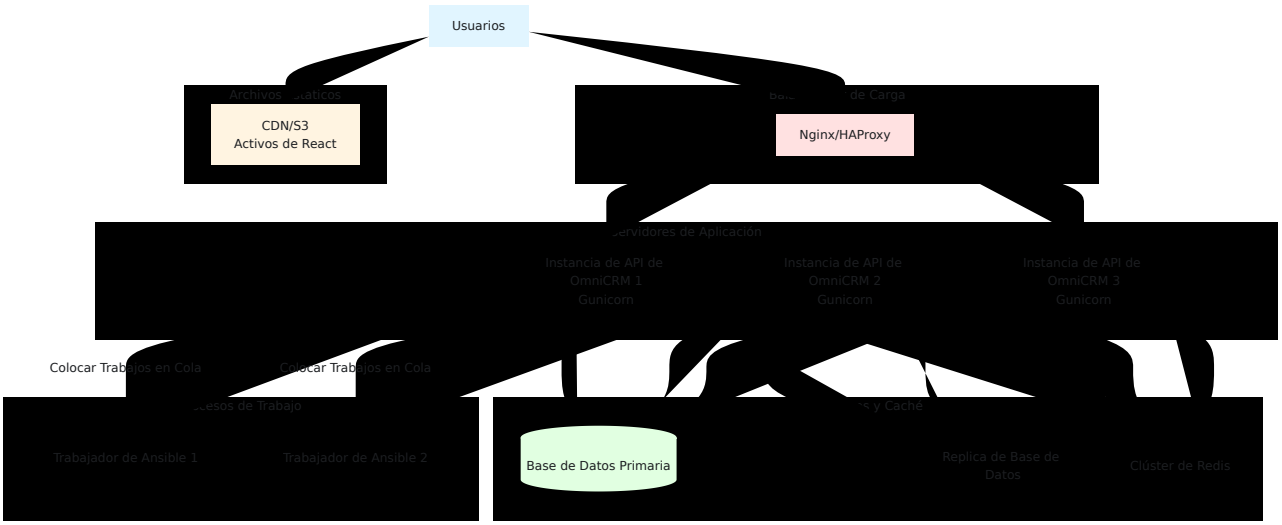


Características de Seguridad:

- Autenticación basada en JWT
- Autenticación de dos factores (TOTP)
- Control de acceso basado en roles (RBAC)
- Acceso a recursos basado en permisos
- Gestión de sesiones con Redis
- Hashing de contraseñas (bcrypt)
- Verificación de correo electrónico
- Registro de auditoría a través del Registro de Actividades

Arquitectura de Despliegue

Configuración de Producción Recomendada:



Stack Tecnológico

Backend:

- Python 3.x
- Framework web Flask
- SQLAlchemy ORM
- Migraciones de Alembic
- Ansible para automatización

Frontend:

- React
- Gestión de estado
- React Router
- Axios para llamadas a la API

Base de Datos:

- Base de datos relacional (almacenamiento de datos primario)
- Redis (caché y sesiones)

Servicios Externos:

- CGRateS (facturación/tarificación)
- Stripe (pagos)
- Mailjet (correo electrónico)
- Google Maps (geocodificación)

Consideraciones de Escalabilidad

Escalabilidad Horizontal:

- Diseño de API sin estado permite múltiples instancias
- Balanceador de carga distribuye solicitudes
- Redis para estado de sesión compartido

Escalabilidad de Base de Datos:

- Réplicas de lectura para consultas de informes
- Agrupamiento de conexiones
- Optimización de consultas e indexación

Aprovisionamiento a Escala:

- Procesamiento de trabajos en segundo plano
- Múltiples trabajadores de Ansible

- Gestión de cola de trabajos
- Lógica de reintento para aprovisionamientos fallidos

Monitoreo y Observabilidad

OmniCRM proporciona métricas integrales basadas en Prometheus para monitorear todos los aspectos del sistema. Para detalles completos, consulte Monitoreo y Métricas.

Métricas Clave:

- Tiempos de respuesta de la API y tasas de solicitud
- Tasas de éxito/fallo de aprovisionamiento y duración de trabajos
- Rendimiento de consultas de base de datos y uso de grupo de conexiones
- Salud de integraciones externas (OCS, Stripe, Mailjet)
- Ejecución de trabajos en segundo plano y errores

Registro:

- Registros de aplicación (Flask)
- Registros de aprovisionamiento (salida de Ansible)
- Registro de actividades (traza de auditoría)
- Seguimiento de errores y alertas

Endpoint de Métricas: Todas las métricas están expuestas en `/crm/metrics` en formato Prometheus. Consulte Monitoreo y Métricas para la configuración de raspado, reglas de alerta y ejemplos de paneles.

Documentación Relacionada

- Playbooks de Ansible - Automatización de aprovisionamiento
- Sistema de Aprovisionamiento - Detalles del flujo de trabajo
- Productos y Servicios - Arquitectura de productos
- Documentación de API - Referencia de API
- RBAC - Seguridad y permisos

Flujos de Autenticación y Controles de Administración

OmniCRM proporciona características de autenticación completas que incluyen inicio de sesión, autenticación de dos factores (2FA), gestión de contraseñas y controles de administración para gestionar la seguridad del usuario. Esta guía se centra en los flujos de trabajo de la interfaz de usuario tanto para los usuarios finales como para los administradores.

Véase también: Portal de Autocuidado para el inicio de sesión de clientes y acceso al portal, RBAC para permisos del personal.

Descripción General

La autenticación de OmniCRM incluye:

- **Inicio de Sesión con Correo Electrónico/Contraseña** - Autenticación estándar basada en credenciales
- **Autenticación de Dos Factores (2FA)** - Segundo factor basado en TOTP opcional
- **Recordarme** - Sesión extendida de hasta 30 días
- **Restablecimiento de Contraseña** - Recuperación de contraseña de autoservicio a través de correo electrónico
- **Controles de Administración** - Herramientas administrativas para restablecer 2FA y contraseñas
- **Inicios de Sesión Sociales** - Integración opcional de Google, Apple, Facebook (si está habilitada)
- **Navegación Basada en Roles** - Enrutamiento automático basado en roles de usuario

Flujo de Inicio de Sesión

La página de inicio de sesión es el punto de entrada para todos los usuarios (personal y clientes).

Inicio de Sesión Estándar

Proceso de Inicio de Sesión:

1. Ingrese **dirección de correo electrónico** (correo del personal o cliente)
2. Ingrese **contraseña**
3. Opcional: Marque **"Recordarme por 30 días"** para sesión extendida
4. Haga clic en **"Iniciar sesión"**

Qué Sucede Después:

- **Sin 2FA:** Usuario inicia sesión inmediatamente, navegado según rol:
  - **Clientes** → portal de Autocuidado (`/self-care`)
  - **Personal/Administradores** → panel de clientes (`/customers`)
  - **Modo CBC** → interfaz de Difusión Celular (`/create-cell-broadcast`)
- **Con 2FA Habilitado:** Redirigido a la pantalla de verificación de 2FA

Función Recordarme:

Cuando está habilitada, la sesión persiste durante **30 días** en lugar de expirar cuando se cierra el navegador. Utiliza cookies seguras solo HTTP.

Mostrar/Ocultar Contraseña:

Haga clic en el **icono de ojo (□)** para alternar la visibilidad de la contraseña.

Inicio de Sesión con 2FA

Si el usuario tiene 2FA habilitado, después de ingresar correo electrónico/contraseña, aparece la pantalla de desafío de 2FA:

Usando Código de Autenticador:

1. Abra la aplicación de autenticador (Google Authenticator, Authy, etc.)
2. Encuentre la entrada de OmniCRM
3. Ingrese el código de 6 dígitos
4. El código se envía automáticamente cuando se ingresan los 6 dígitos
5. Si es válido, el usuario inicia sesión y se navega al panel correspondiente

Usando Código de Recuperación:

Si la aplicación de autenticador no está disponible:

1. Haga clic en la pestaña **"Código de Recuperación"**
2. Ingrese uno de sus códigos de respaldo guardados (por ejemplo, `3fa5b9c2`)

3. Haga clic en **"Verificar"**
4. El código se consume (solo se puede usar una vez)

Cancelar:

Haga clic en **"Cancelar"** para volver a la página de inicio de sesión.

Inicios de Sesión Sociales (Opcional)

Si está habilitado (`VITE_ALLOW_SOCIAL_LOGINS=yes`), aparecen botones de inicio de sesión social:

> [Iniciar sesión con Google] [• Iniciar sesión con Apple] [Iniciar sesión con Facebook]

Haga clic en cualquier botón para autenticarse a través de ese proveedor. Actualmente muestra un mensaje de "próximamente" (implementación de inicio de sesión social en progreso).

Enlace de Olvidé mi Contraseña

Haga clic en el enlace **"¿Olvidó su contraseña?"** para iniciar el flujo de restablecimiento de contraseña.

Configuración de Autenticación de Dos Factores (2FA)

Los usuarios pueden habilitar 2FA para una mayor seguridad de la cuenta. 2FA utiliza TOTP (Contraseña de Un Solo Uso Basada en Tiempo) compatible con aplicaciones de autenticador estándar.

Accediendo a la Configuración de 2FA

Desde el perfil o la configuración del usuario:

Nota para Clientes:

Los usuarios con rol de cliente no ven los avisos de 2FA. 2FA generalmente se requiere solo para el personal y los usuarios administrativos.

Paso 1: Confirmar Contraseña

> Contraseña Actual
>
> [Cancelar] [Continuar]

Ingrese su contraseña actual para continuar. Esto verifica su identidad antes de habilitar 2FA.

Paso 2: Escanear Código QR

> [Cancelar] [Confirmar]

Instrucciones de Configuración:

1. **Descargar Aplicación de Autenticador** (si no tiene una):
   - iOS: Apple App Store → "Google Authenticator"
   - Android: Google Play → "Google Authenticator"
   - Alternativas: Authy, Microsoft Authenticator, 1Password
2. **Escanear Código QR:**
   - Abra la aplicación de autenticador
   - Toque "+" o "Agregar cuenta"
   - Elija "Escanear código QR"
   - Apunte la cámara al código QR en la pantalla
   - La aplicación agrega la entrada "OmniCRM" con el código de 6 dígitos
3. **Guardar Códigos de Respaldo:**
   - **CRÍTICO:** Anote o copie estos 8 códigos
   - Almacene en un lugar seguro (gestor de contraseñas, caja fuerte, etc.)
   - Cada código es de un solo uso
   - Se utiliza si pierde el acceso a la aplicación de autenticador
   - Haga clic en **"Copiar Códigos"** para copiar todos los códigos al portapapeles
4. **Verificar Configuración:**

- Ingrese el código actual de 6 dígitos de la aplicación de autenticador
- Haga clic en **"Confirmar"**
- Si es válido, 2FA ahora está habilitado

Paso 3: 2FA Habilitado

Aparece un mensaje de éxito:

A partir de ahora, el inicio de sesión requiere tanto la contraseña como el código de 2FA.

Flujo de Restablecimiento de Contraseña (Autoservicio)

Los usuarios que olvidan su contraseña pueden restablecerla a través de correo electrónico.

Paso 1: Solicitar Enlace de Restablecimiento

Desde la página de inicio de sesión, haga clic en **"¿Olvidó su contraseña?"**

1. Ingrese **dirección de correo electrónico**
2. Haga clic en **"Enviar Enlace de Restablecimiento"**

Qué Sucede:

- El sistema verifica si el correo existe en la base de datos
- Si se encuentra, envía un correo electrónico de restablecimiento de contraseña a través de Mailjet
- El correo contiene un enlace de restablecimiento limitado en el tiempo (típicamente 1 hora de expiración)
- Aparece un mensaje de éxito: "Las instrucciones de restablecimiento han sido enviadas a su correo electrónico"

Si No Se Encuentra el Correo:

Mensaje de error: "No se encontró ninguna cuenta con esa dirección de correo electrónico"

Paso 2: Revisar Correo Electrónico

El usuario recibe un correo con un asunto como:

> Hola [Nombre],
>
> Ha solicitado un restablecimiento de contraseña para su cuenta de OmniCRM.
>
> Haga clic en el enlace a continuación para restablecer su contraseña: <https://yourcompany.com/reset-password/abc123token456>
>
> Este enlace expira en 1 hora.
>
> Si no solicitó esto, ignore este correo.

Haga clic en el enlace de restablecimiento para continuar.

Paso 3: Establecer Nueva Contraseña

El enlace de restablecimiento abre la página de creación de contraseña:

> □ (icono de candado)
>
> Contraseña
>
> Confirmar Contraseña
>
> [Restablecer Contraseña]

1. Ingrese **nueva contraseña**
2. Vuelva a ingresar en el campo **Confirmar Contraseña**
3. Haga clic en **"Restablecer Contraseña"**

Requisitos de Contraseña:

- Longitud mínima (típicamente 8+ caracteres)
- Las contraseñas deben coincidir

Éxito:

- Mensaje de éxito: "La contraseña ha sido restablecida con éxito"
- Redirigido automáticamente a la página de inicio de sesión
- El usuario ahora puede iniciar sesión con la nueva contraseña

Token Expirado/Inválido:

Si el enlace de restablecimiento está expirado o es inválido:

> [Solicitar Nuevo Enlace de Restablecimiento]

Controles de Administración para la Gestión de Usuarios

Los administradores con los permisos apropiados pueden gestionar la configuración de autenticación de usuarios desde la interfaz de Gestión de Usuarios.

Accediendo a la Gestión de Usuarios

Muestra una tabla de todos los usuarios con botones de acción.

> Nombre Correo Teléfono Acciones John Smith <john@example.com> +44 123... ✏ □ ××× □ Jane Doe <jane@example.com> +44 456... ✏ □ □ Bob Wilson <bob@example.com> +44 789... ✏ □ □ □ ✉

Iconos de Acción:

- ✏ **Editar** - Modificar detalles del usuario, roles, permisos
- □ **Eliminar** - Eliminar cuenta de usuario
- □ **Restablecer Contraseña** - Generar contraseña temporal
- □ **Restablecer 2FA** - Deshabilitar 2FA para el usuario (solo se muestra si 2FA está habilitado)
- ✉ **Enviar Correo de Bienvenida** - Reenviar correo de bienvenida (solo se muestra si el usuario nunca ha iniciado sesión)

Admin: Restablecer Contraseña de Usuario

Cuando el usuario olvida su contraseña y el administrador necesita ayudar:

Paso 1: Hacer Clic en el Icono de Restablecer Contraseña (□)

Aparece un modal de confirmación:

> ¿Está seguro de que desea restablecer la contraseña de:
>
> Usuario: John Smith (<john@example.com>)
>
> Se generará y mostrará una contraseña temporal. El usuario debe cambiar esta contraseña en el próximo inicio de sesión.
>
> [Cancelar] [Restablecer Contraseña]

Paso 2: Confirmar Restablecimiento

Haga clic en **"Restablecer Contraseña"**. El sistema genera una contraseña temporal segura.

Paso 3: Contraseña Temporal Mostrada

> Contraseña temporal para John Smith:
>
> [□ Copiar Contraseña]
>
> ⚠ IMPORTANTE: • Envíe esta contraseña al usuario a través de un canal seguro • No envíe por correo electrónico o mensajería no segura • El usuario deberá cambiar la contraseña en el próximo inicio de sesión
>
> [Cerrar]

Acción del Administrador:

- Copiar la contraseña temporal
- Llamar al usuario o comunicarse por un método seguro
- Proporcionar la contraseña temporal verbalmente
- Indicar al usuario que inicie sesión y cambie la contraseña

Experiencia del Usuario:

Cuando el usuario inicia sesión con la contraseña temporal:

1. El inicio de sesión tiene éxito
2. Redirigido inmediatamente a la pantalla de "Cambiar Contraseña"
3. Debe establecer una nueva contraseña antes de acceder al sistema
4. No puede omitir el cambio de contraseña

Admin: Restablecer 2FA de Usuario

Cuando el usuario pierde el acceso a la aplicación de autenticador y todos los códigos de respaldo:

Paso 1: Hacer Clic en el Icono de Restablecer 2FA (□)

Solo aparece para usuarios con 2FA actualmente habilitado.

Aparece un modal de confirmación:

Paso 2: Confirmar Restablecimiento

Haga clic en **"Restablecer 2FA"**

Paso 3: Confirmación

Mensaje de éxito:

> John Smith ahora puede iniciar sesión solo con su contraseña. Puede volver a habilitar 2FA desde su configuración de usuario.

Experiencia del Usuario:

- El usuario ahora puede iniciar sesión solo con la contraseña (no se requiere código 2FA)
- El icono de escudo 2FA (□) desaparece de la fila del usuario en la tabla de administración
- El usuario puede volver a habilitar 2FA voluntariamente desde su configuración

Nota de Seguridad Importante:

Antes de restablecer 2FA, el administrador debe:

1. Verificar la identidad del usuario a través de medios alternativos:
   - Verificación de ID gubernamental
   - Preguntas de seguridad
   - Verificación de transacciones recientes
   - Verificación en persona (si corresponde)
2. Documentar el restablecimiento en las notas del cliente
3. Informar al usuario que vuelva a habilitar 2FA después de recuperar el acceso

Admin: Enviar Correo de Bienvenida

Para usuarios que no han recibido o han perdido su correo de bienvenida:

Cuando Está Disponible:

El icono de avión de papel (✉) solo aparece para usuarios que **nunca han iniciado sesión** (`login_count = 0`).

Haga clic en el Icono de Enviar Correo de Bienvenida (✉)

> Enviar correo de bienvenida a:

> Usuario: Bob Wilson (<bob@example.com>)
>
> El correo incluirá: • Mensaje de bienvenida • Instrucciones de inicio de sesión • Enlace para establecer la contraseña inicial (si corresponde) • Información de contacto de soporte
>
> [Cancelar] [Enviar Correo]

Haga clic en **"Enviar Correo"**

Mensaje de éxito:

Correo Enviado a través de Mailjet:

Utiliza la plantilla: `api_crmCommunicationUserWelcome`

Admin: Editar Usuario

Haga clic en **Icono de Editar (✏)** para modificar los detalles del usuario:

> Nombre
>
> Apellido
>
> Correo Electrónico
>
> Número de Teléfono
>
> Roles ☑ admin ☐ customer_service_agent_1 ☐ customer
>
> [Cancelar] [Guardar Cambios]

Campos Editables:

- Nombre, correo, teléfono
- **Roles** - Asignar/quitar roles (afecta permisos)
- Estado activo/inactivo

Admin: Eliminar Usuario

Haga clic en **Icono de Eliminar (🗑)** para eliminar al usuario:

> ¿Está seguro de que desea eliminar:
>
> Usuario: John Smith (<john@example.com>)
>
> ⚠ ADVERTENCIA: Esta acción no se puede deshacer.
>
> Esto eliminará permanentemente: • Cuenta de usuario y credenciales • Configuraciones de 2FA • Historial de sesiones
>
> Los datos y transacciones del cliente NO serán eliminados.
>
> [Cancelar] [Eliminar Usuario]

Haga clic en **"Eliminar Usuario"** para confirmar.

Mensaje de éxito:

Mejores Prácticas

Para Usuarios Finales

Seguridad en el Inicio de Sesión:

- Utilice contraseñas fuertes y únicas
- Habilite "Recordarme" solo en dispositivos personales
- Siempre cierre sesión en computadoras compartidas
- Habilite 2FA para mayor seguridad

Gestión de 2FA:

- Guarde los códigos de respaldo inmediatamente después de habilitar 2FA
- Almacene los códigos en un gestor de contraseñas o lugar seguro
- Pruebe un código de respaldo para asegurarse de que funcione
- Regenerar códigos de respaldo si utiliza varios

- Contacte al administrador si pierde tanto el autenticador como los códigos de respaldo

Gestión de Contraseñas:

- Utilice un gestor de contraseñas para generar y almacenar contraseñas
- Nunca comparta contraseñas por correo electrónico o mensajería
- Cambie la contraseña si sospecha que ha sido comprometida
- Use una contraseña única para OmniCRM (no reutilice contraseñas)

Para Administradores

Gestión de Seguridad de Usuarios:

- Verifique la identidad del usuario antes de restablecer 2FA o contraseñas
- Nunca envíe contraseñas temporales por correo electrónico
- Documente todos los restablecimientos de seguridad en las notas del usuario
- Anime al personal a habilitar 2FA
- Monitoree patrones de inicio de sesión inusuales

Restablecimientos de Contraseña:

- Comuníquese con contraseñas temporales solo por teléfono o en persona
- Genere contraseñas temporales fuertes (el sistema lo hace automáticamente)
- Asegúrese de que el usuario cambie la contraseña en el primer inicio de sesión
- No restablezca contraseñas innecesariamente - utilice el flujo de restablecimiento por correo electrónico cuando sea posible

Restablecimientos de 2FA:

- Trate los restablecimientos de 2FA como acciones de alta seguridad
- Verifique la identidad a través de múltiples canales antes de restablecer
- Documente la razón del restablecimiento

- Anime al usuario a volver a habilitar 2FA inmediatamente después de recuperar el acceso
- Considere requerir 2FA para todos los usuarios administrativos

Gestión de Usuarios:

- Revise regularmente la lista de usuarios en busca de cuentas inactivas
- Elimine usuarios que hayan dejado la organización
- Asegúrese de que las asignaciones de roles sean apropiadas
- Monitoree a los usuarios que nunca han iniciado sesión
- Audite los permisos de usuario trimestralmente

Solución de Problemas

Error "Correo electrónico o contraseña inválidos"

- **Causa:** Credenciales incorrectas
- **Solución:**
  - Verifique que la dirección de correo electrónico sea correcta
  - Asegúrese de que el bloqueo de mayúsculas esté desactivado
  - Intente restablecer la contraseña si la olvidó
  - Contacte al administrador si la cuenta está bloqueada

Código 2FA no aceptado

- **Causa:** Problema de sincronización de tiempo o código incorrecto
- **Solución:**
  - Asegúrese de que la hora del dispositivo sea correcta (Configuración → Fecha y Hora → Automático)
  - Espere a que el código se actualice (los códigos cambian cada 30 segundos)
  - Intente con el siguiente código que aparece
  - Use un código de respaldo si el autenticador no funciona
  - Contacte al administrador para restablecer 2FA si todo lo demás falla

"Recordarme" no funciona

- **Causa:** Cookies deshabilitadas o borradas
- **Solución:**
  - Habilite las cookies en la configuración del navegador
  - No borre las cookies cuando cierre el navegador
  - Desactive las extensiones de privacidad para el dominio de OmniCRM
  - Intente con un navegador diferente

Correo de restablecimiento de contraseña no recibido

- **Causa:** Correo no enviado, filtro de spam o correo electrónico incorrecto
- **Solución:**
  - Verifique la carpeta de spam/correo no deseado
  - Verifique que la dirección de correo electrónico sea correcta
  - Espere de 5 a 10 minutos (la entrega del correo puede retrasarse)
  - Verifique que la integración de Mailjet esté funcionando (administrador)
  - Contacte al administrador para un restablecimiento manual de la contraseña

Enlace de restablecimiento de contraseña expirado

- **Causa:** Token expirado (típicamente 1 hora)
- **Solución:**
  - Solicite un nuevo restablecimiento de contraseña
  - Revise el correo y haga clic en el enlace rápidamente
  - Contacte al administrador si hay problemas repetidos

No se puede habilitar 2FA (contraseña incorrecta)

- **Causa:** Contraseña actual ingresada incorrectamente
- **Solución:**
  - Verifique la contraseña actual
  - Restablezca la contraseña primero si no está seguro
  - Contacte al administrador para obtener ayuda

Perdió la aplicación de autenticador y los códigos de respaldo

- **Causa:** Teléfono perdido/restablecido, códigos de respaldo no guardados

- **Solución:**
  - Contacte al administrador de inmediato
  - El administrador verificará la identidad y restablecerá 2FA
  - Inicie sesión solo con la contraseña después del restablecimiento
  - Vuelva a habilitar 2FA y GUARDE los códigos de respaldo esta vez

Admin: "No se pudo restablecer 2FA" error

- **Causa:** Permisos insuficientes
- **Solución:**
  - Asegúrese de tener el rol de administrador
  - Verifique los permisos de la API
  - Contacte al administrador del sistema

Admin: Contraseña temporal no generada

- **Causa:** Error de API o problema de permisos
- **Solución:**
  - Actualice la página e inténtelo de nuevo
  - Verifique los permisos de administrador
  - Revise los registros de la API en busca de errores
  - Asegúrese de que la base de datos sea accesible

Consideraciones de Seguridad

Gestión de Sesiones:

- Las sesiones expiran después de un período de inactividad
- "Recordarme" extiende la sesión a 30 días
- Las sesiones se almacenan como cookies solo HTTP (no accesibles para JavaScript)
- La bandera segura asegura que las cookies solo se envíen a través de HTTPS

Seguridad de Contraseñas:

- Las contraseñas se hash utilizando algoritmos estándar de la industria
- Las contraseñas en texto plano nunca se almacenan
- Las contraseñas temporales expiran automáticamente después del primer uso
- Se rastrean los intentos de inicio de sesión fallidos (posible limitación de tasa)

Seguridad de 2FA:

- Los secretos de TOTP se cifran en la base de datos
- Los códigos QR se generan del lado del cliente cuando es posible
- Los códigos de respaldo se hash antes de su almacenamiento
- Cada código de respaldo es de un solo uso

Acciones de Administrador:

- Los restablecimientos de 2FA se registran en el registro de actividad
- Los restablecimientos de contraseñas crean un rastro de auditoría
- Las acciones de administrador requieren permisos de rol apropiados
- Las direcciones IP se registran para eventos de seguridad

Documentación Relacionada

- `2fa` - Referencia detallada de la API de 2FA (enfocada en API)
- `rbac` - Control de acceso basado en roles y permisos
- `administration_configuration` - Configuración de correo de Mailjet para el restablecimiento de contraseña
- `integrations_mailjet` - Configuración de plantilla de correo
- `customer_care` - Portal de Autocuidado para clientes

Crear un Cliente

0fT52ZvoZBE

Clientes, Contactos, Sitios y Servicios

Tenemos un modelo simple de un **Cliente** que puede tener múltiples **Contactos** y múltiples **Sitios**, **Servicios**, etc.

Un **Cliente** es una empresa o individuo que tiene una relación con nosotros, a quien le enviamos una factura / cuenta.

Un **Contacto** es una persona que trabaja con el cliente; para un individuo, probablemente sea lo mismo que el cliente mismo, una sola persona, pero podemos tener miembros de la familia u otros contactos, y cada contacto tiene un tipo, por ejemplo, un contacto de facturación, un contacto técnico, etc., lo que influye en cómo manejamos el contacto.

Un **Sitio** es una ubicación física donde entregamos servicios; podría ser un hogar, oficina u otra ubicación. Esto nos permite tener múltiples sitios para un solo cliente, por ejemplo, un cliente con múltiples oficinas, y saber qué servicios están asociados con qué sitio.

Un **Servicio** es algo por lo que facturamos a un cliente; podría ser un servicio de internet en casa, servicio móvil o incluso servicios abstractos como arrendar una subred o proporcionar electricidad medida a un rack. Cada servicio está vinculado a un cliente y un sitio, y puede tener múltiples cargos asociados.

Los clientes también tienen un Registro de Actividad, que es un registro de todos los cambios realizados, Etiquetas, Atributos para almacenar metadatos personalizados, Artículos de Inventario e información financiera como Transacciones, Facturas y Métodos de Pago.

Una vez que hemos creado un cliente, podemos agregar un servicio a ese cliente, que es por lo que les facturamos.

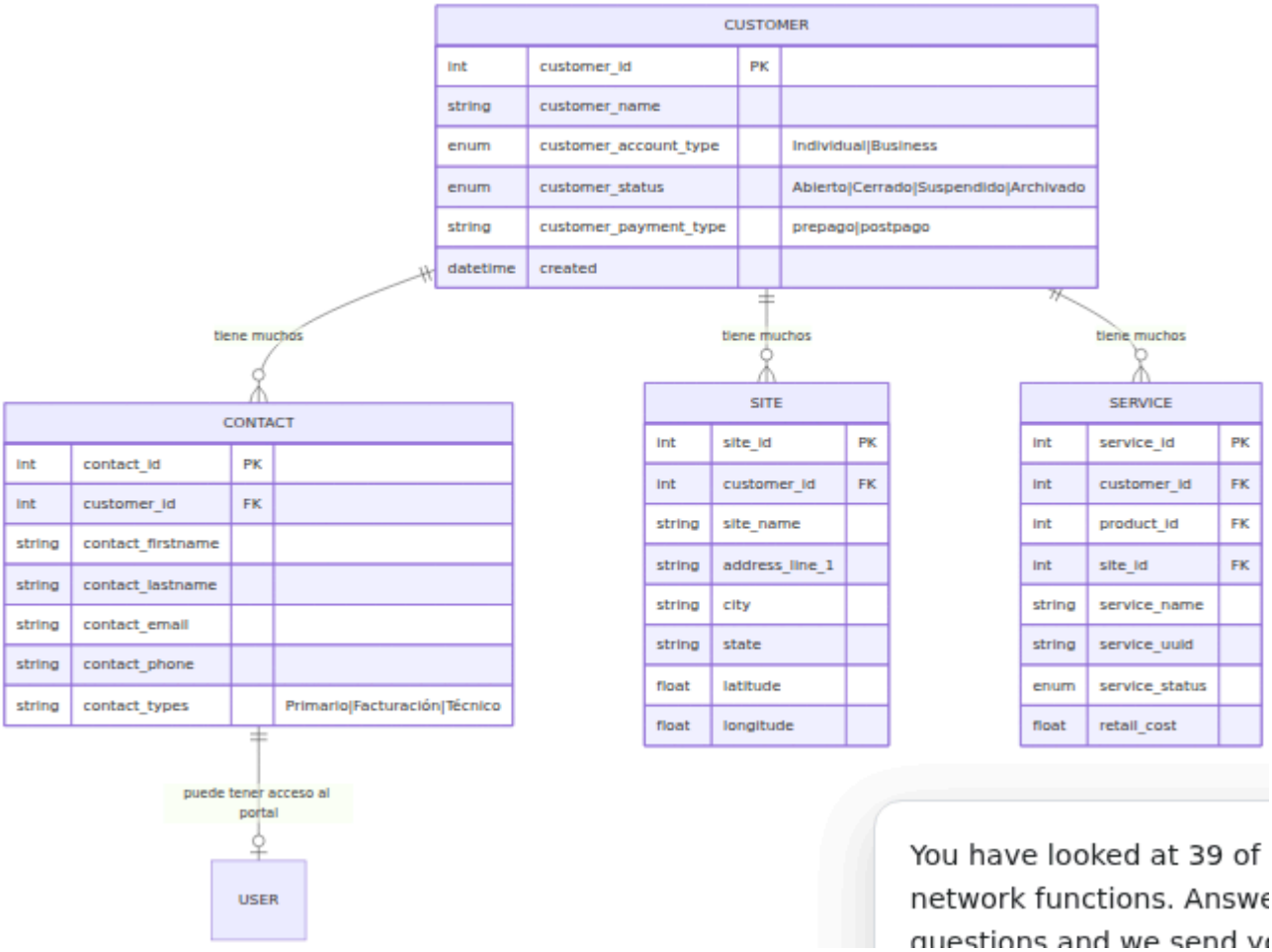
Para obtener información sobre cómo crear un cliente, consulte Crear un Cliente.

Resumen del Modelo de Datos

OmniCRM utiliza un modelo de datos relacional organizado en torno a los clientes y sus servicios. El modelo se desglosa en secciones enfocadas a continuación.

Relaciones Básicas del Cliente

El cliente es la entidad central, con contactos, sitios y servicios relacionados.

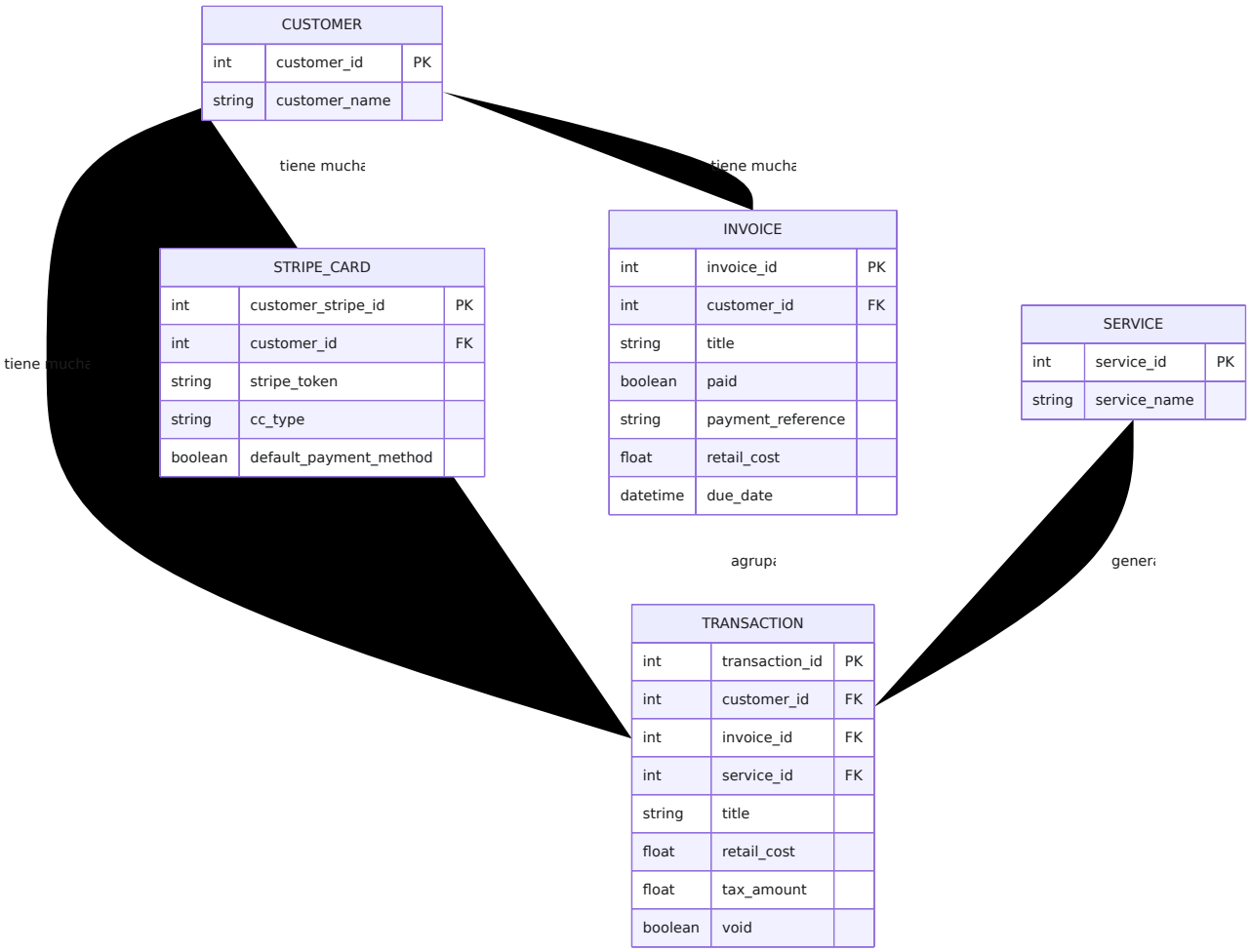


Puntos Clave:

- Un cliente puede tener múltiples contactos (facturación, técnico, etc.)
- Un cliente puede tener múltiples sitios (oficinas sucursales, ubicaciones)
- Los servicios se entregan a los sitios
- Los contactos pueden tener acceso al portal a través de cuentas de usuario vinculadas

Datos de Facturación y Financieros

Las transacciones y facturas rastrean toda la actividad financiera.

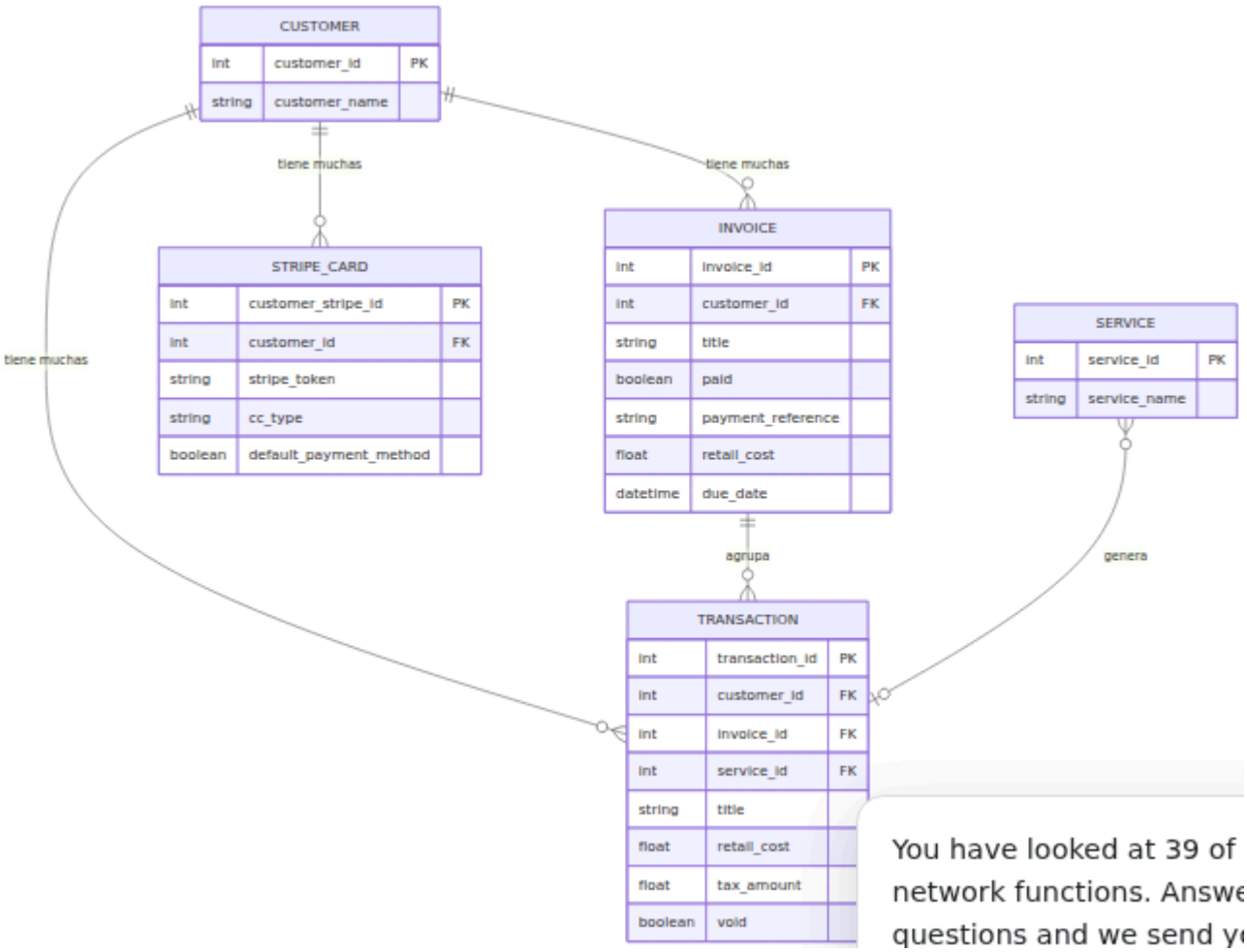


Puntos Clave:

- Las transacciones pueden ser independientes o agruparse en facturas
- Los servicios generan automáticamente transacciones mensuales
- Las tarjetas de Stripe están tokenizadas y almacenadas de forma segura
- Las facturas agrupan múltiples transacciones para la facturación

Producto y Aprovisionamiento

Los productos definen las ofertas de servicio; el aprovisionamiento crea servicios reales.

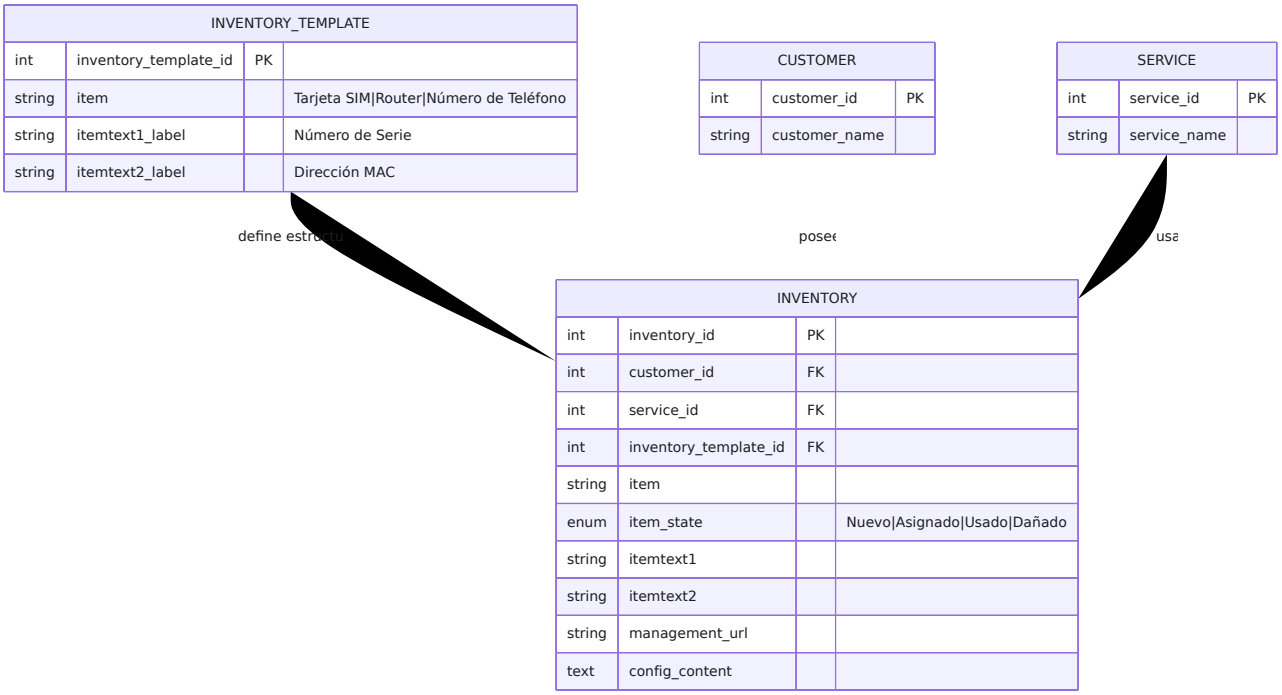


Puntos Clave:

- Los productos son plantillas; los servicios son instancias activas
- El aprovisionamiento crea o modifica servicios a través de playbooks de Ansible
- Cada trabajo de aprovisionamiento tiene múltiples eventos para rastrear el progreso
- Un aprovisionamiento de producto puede crear múltiples servicios (paquetes)

Sistema de Inventario

Rastrear activos físicos y virtuales asignados a los clientes.

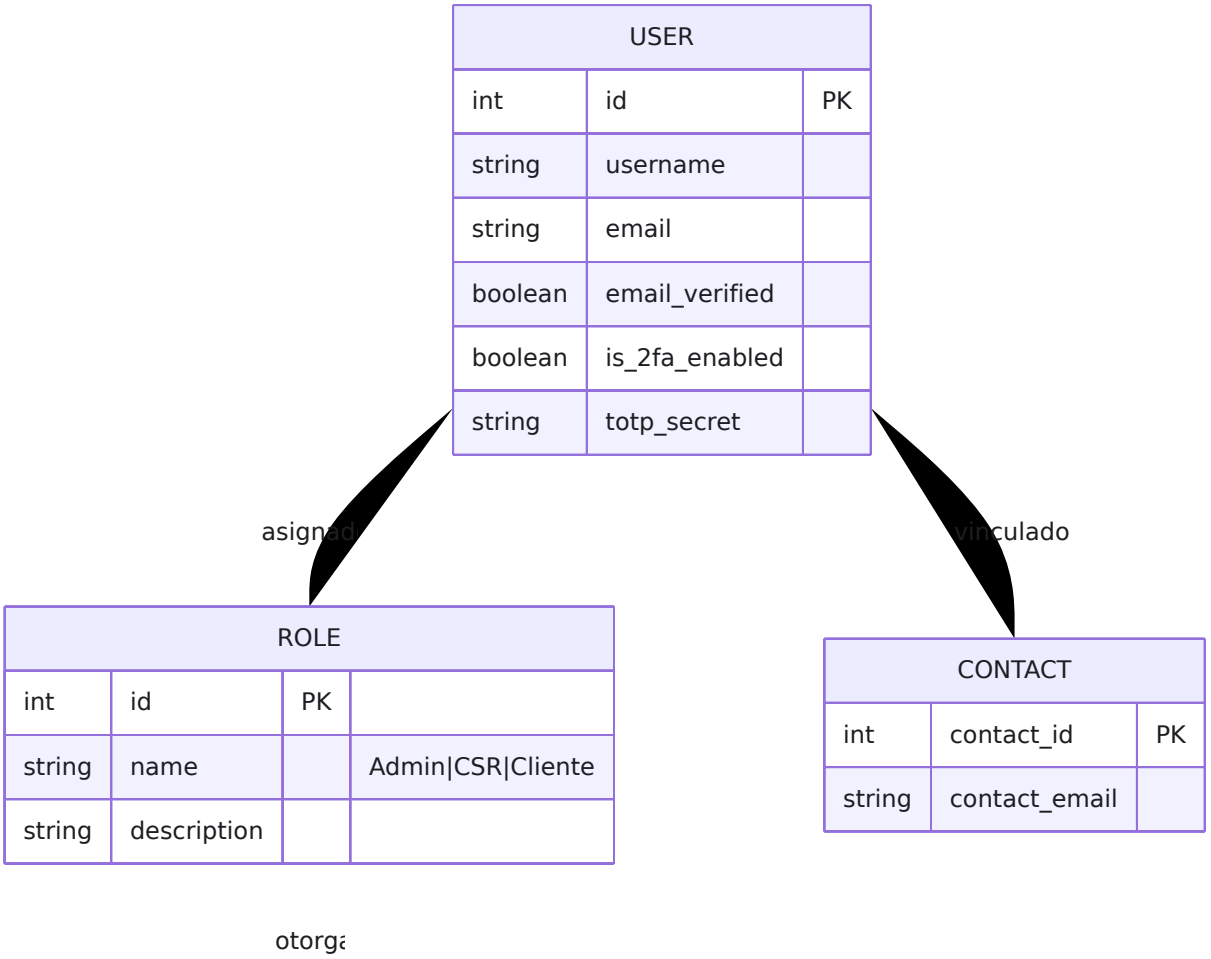


Puntos Clave:

- Las plantillas de inventario definen la estructura (campos) para cada tipo de artículo
- Los campos flexibles itemtext1-20 se adaptan a diferentes tipos de inventario
- Los artículos pueden ser asignados a clientes y vinculados a servicios
- Almacena configuraciones de equipos y credenciales de gestión

Usuario y Seguridad

Cuentas de usuario con control de acceso basado en roles.



Puntos Clave:

- Los usuarios pueden tener múltiples roles; los roles tienen múltiples permisos
- La autenticación de dos factores (2FA) es opcional por usuario
- Los usuarios del personal son independientes; los usuarios clientes se vinculan a contactos
- Permisos granulares controlan el acceso a recursos

Lista de Clientes

La lista de clientes proporciona una tabla buscable y filtrable de todos los clientes en el sistema.

Características:

- **Buscar** - Filtrar clientes por nombre o ID
- **Acciones Masivas** - Seleccionar múltiples clientes para operaciones por lotes
- **Paginación** - Navegar a través de listas grandes de clientes
- **Acciones Rápidas** - Ver o eliminar clientes directamente desde la lista

Cliente - Detalle

El objeto cliente en sí no contiene mucha información, solo es un nombre y una referencia a los contactos y sitios.

Su implementación específica puede incluir campos adicionales o personalizaciones, pero el objeto cliente básico es muy simple.

En la página de resumen también hay un gráfico que muestra el Ingreso Promedio por Usuario (`ARPU`) para el cliente, que es el ingreso total dividido por

el número de servicios, y una comparación de cómo se compara este cliente con el ARPU promedio de todos los clientes en el sistema.

Las opciones de Estado del Cliente están adaptadas según las necesidades específicas de su negocio, pero típicamente incluyen opciones como Activo, Inactivo, Pendiente, etc., con diferentes reglas en cada una que controlan el comportamiento del cliente en el sistema en ese estado.

Eliminar un cliente solo se puede hacer si el cliente no tiene servicios activos, facturas impagas o transacciones no facturadas. Si el cliente tiene alguno de estos, deberá cerrar los servicios activos y asegurarse de que se realicen los pagos antes de poder eliminar al cliente, lo que a su vez archivará al cliente y todos los datos asociados, que pueden ser restaurados más tarde si es necesario.

Sitio - Detalle

Los sitios son ubicaciones físicas donde se entregan servicios y pueden tener múltiples servicios asociados con ellos.

Se utilizan predominantemente para clientes comerciales, donde un solo cliente podría tener múltiples sitios, por ejemplo, un cliente con múltiples oficinas.

Tener múltiples sitios nos permite rastrear qué servicios están asociados con qué sitio, por ejemplo, si un cliente solicita un nuevo servicio para una nueva oficina, necesitamos asegurarnos de que entregamos el servicio correcto a la ubicación correcta. Esto nos permite rastrear servicios por sitio y facturarlos por separado si es necesario.

Integración de Google Maps y Geocodificación

Cada sitio está integrado con **Google Maps** para garantizar datos de dirección precisos y geolocalización. La interfaz de usuario geocodifica automáticamente las direcciones y genera datos de ubicación.

Cómo Funciona la Geocodificación de Direcciones:

Al agregar o editar un sitio, la interfaz proporciona dos métodos para establecer la ubicación:

1. **Búsqueda de Dirección (Recomendado)**
    - Use la barra de búsqueda en la parte superior del formulario
    - Escriba una dirección y Google Maps Autocomplete sugiere coincidencias
    - Seleccione la dirección correcta del menú desplegable
    - El sistema completa automáticamente:
        - **Nombre del Sitio** - Nombre del lugar de Google Maps
        - **Línea de Dirección 1** - Número y nombre de la calle
        - **Línea de Dirección 2** - Subpremisa (número de unidad/suite)
        - **Ciudad** - Localidad
        - **Estado/Región** - Área administrativa
        - **Código Postal** - Código postal
        - **País** - Nombre del país
        - **Latitud y Longitud** - Coordenadas precisas
        - **Código Plus** - Código de Ubicación Abierto de 11 caracteres (por ejemplo, `8C3MFJV8+2F`)
        - **ID de Lugar de Google Maps** - Identificador único del lugar
2. **Marcador de Mapa Arrastrable (Manual)**
    - Arrastre el pin en el mapa a la ubicación exacta

- El sistema realiza **geocodificación inversa** para obtener la dirección de las coordenadas
- Los mismos campos se completan automáticamente según la ubicación del pin
- Útil para áreas rurales o cuando la dirección es imprecisa

Generación de Código Plus:

Los Códigos Plus se generan automáticamente a partir de la latitud/longitud utilizando la biblioteca de Código de Ubicación Abierto. Un Código Plus es un código corto (11 caracteres) que representa una ubicación precisa en cualquier parte del mundo.

Ejemplo:

> Campos autocompletados: ✓ Nombre del Sitio: "123 Main Street" ✓ Línea de Dirección 1: "123 Main Street" ✓ Ciudad: "Londres" ✓ Estado: "Gran Londres" ✓ País: "Reino Unido" ✓ Código Postal: "SW1A 1AA" ✓ Latitud: 51.5074 ✓ Longitud: -0.1278 ✓ Código Plus: "9C3XGPHC+3Q" ✓ ID de Lugar de Google: "ChIJdd4hrwug2EcRmSrV3Vo6llI"

Requisitos de Validación:

Antes de guardar un sitio, el sistema valida:

- **Latitud y Longitud** deben estar configuradas (ya sea a través de búsqueda o arrastrando el marcador)
- **País** debe estar poblado (se establece de forma predeterminada en `VITE_DEFAULT_COUNTRY` si no se establece)
- **Código Plus** debe tener 12 caracteres (11 + 1 para relleno)

Si la validación falla, verá un error:

Retroalimentación Visual:

La interfaz muestra retroalimentación en tiempo real:

o

Cómo se Utiliza la Información de Ubicación

Los datos de ubicación geocodificados (latitud, longitud, Código Plus) se utilizan en todo OmniCRM para:

1. Entrega e Instalación de Servicios

- **Técnicos de Campo** - Acceder al Código Plus para navegar a la ubicación exacta del sitio
- **Programación de Instalaciones** - Asignar técnicos según la proximidad geográfica
- **Despliegue de Equipos** - Asegurar que el equipo correcto se entregue al sitio correcto

2. Notificaciones de Interrupciones

- **Alertas Geocercadas** - Si hay una interrupción de red en un área específica, consultar sitios por radio de lat/long
- **Comunicaciones Dirigidas** - Enviar notificaciones de interrupciones solo a los clientes afectados a través de Mailjet
- **Páginas de Estado** - Mostrar un mapa de interrupciones con los sitios afectados

Ejemplo:

> Consulta: SELECT * FROM Customer_Site
> WHERE distance(latitude, longitude, 51.5074, -0.1278) < 5
>
> Resultado: 47 sitios afectados Acción: Enviar notificación de interrupción a 47 clientes

3. Informes y Análisis

- **Ingresos Geográficos** - Ingresos por ciudad, estado, región
- **Mapas de Densidad de Servicios** - Mapa de calor de ubicaciones de servicios
- **Planificación de Expansión** - Identificar áreas desatendidas

4. Clientes Comerciales de Múltiples Sitios

- **Gestión de Sitios** - Rastrear qué servicios están en qué ubicaciones
- **Facturación Separada** - Facturar por sitio si es necesario
- **Asignación de Servicios** - Vincular servicios a sitios específicos durante el aprovisionamiento

Sitios Rurales y Remotos

Para clientes en áreas rurales donde las direcciones de calle pueden no existir o ser inexactas:

1. **Usar Arrastre de Mapa**
   - Acercarse a la área aproximada
   - Arrastrar el pin a la propiedad/edificio exacto
   - El sistema genera un Código Plus para esa ubicación precisa
2. **Compartir Código Plus**
   - Compartir el Código Plus con el cliente (por ejemplo, `8C3MFJV8+2F`)
   - El cliente puede ingresar esto en Google Maps para ver la ubicación exacta
   - Los técnicos de campo utilizan el Código Plus para la navegación
3. **Notas de Dirección**
   - Usar el campo "Nota de Dirección" para direcciones adicionales
   - Ejemplo: "Gire a la izquierda en el granero rojo, 500 m después de la rejilla de ganado"
   - Notas visibles para los equipos de instalación

TIP

Puede arrastrar el pin en el mapa a la ubicación correcta si la dirección no es precisa. El sistema geocodificará inversamente la ubicación y completará automáticamente todos los campos de dirección.

Contacto - Detalle

Los Contactos son personas asociadas con el cliente. Pueden ser contactos de facturación, contactos técnicos u otros tipos, y cada contacto tiene un tipo que influye en cómo manejamos el contacto.

Podemos tener múltiples contactos para un solo cliente, por ejemplo, un cliente con múltiples contactos de facturación, o un cliente con múltiples contactos técnicos.

Un buen ejemplo sería una empresa con un proveedor de servicios gestionados, que maneja el lado técnico de las cosas, y un contacto de facturación separado que maneja el lado financiero, o una familia donde cada miembro tiene su propio contacto pero no todos están autorizados para hacer cambios.

Asimismo, es posible que solo queramos enviar notificaciones de interrupciones al contacto técnico, o solo enviar facturas al contacto de facturación, y el tipo de contacto nos permite controlar esto.

La lógica exacta de cómo se utilizan los tipos de contacto depende de su negocio, pero la idea básica es que cada contacto tiene un tipo que influye en

cómo los manejamos, y que cada persona asociada con el cliente es un contacto.

Los contactos se sincronizan con la integración de Mailjet, lo que nos permite enviar campañas de correo electrónico dirigidas según el tipo de contacto, ubicación, gasto mensual o servicios adquiridos, y gestionar todas las plantillas de correo electrónico utilizadas para comunicaciones transaccionales.

Navegación

OmniCRM está diseñado desde cero para ser receptivo e intuitivo.

Esta guía te ayudará a navegar por el sistema y encontrar la información que necesitas.

Una vez que inicies sesión, la barra de navegación en el lado izquierdo de la pantalla te mostrará las secciones principales del sistema y las subsecciones dentro de cada una.

En la parte superior derecha de la pantalla, verás el menú de usuario, que te permite cerrar sesión, cambiar tu contraseña o acceder a la configuración de tu

usuario.

Tienes enlaces a todas las aplicaciones web más utilizadas de tu organización en la barra de WebApps (esto se puede adaptar a las necesidades de tu organización).

Cualquier alerta que tengas pendiente es visible en la parte superior derecha de la pantalla, y puedes hacer clic en la alerta para ver más información.

Podemos cambiar el idioma del sistema haciendo clic en el menú desplegable de idioma en la parte superior derecha de la pantalla.

Si eres un príncipe de las tinieblas, puedes cambiar a modo oscuro haciendo clic en el ícono de la luna en la parte superior derecha de la pantalla.

Pagos, Facturas y Transacciones

Transacciones de Clientes

Cualquier cosa que cueste dinero en el sistema se registra como una transacción bajo el cliente.

Cada transacción tiene un monto monetario para el costo mayorista y el costo minorista, y una descripción de para qué es la transacción.

Las transacciones pueden ser generadas automáticamente por el sistema, por ejemplo, cuando se provisiona un servicio, se crea una transacción para el costo de configuración, y cuando se factura un servicio, se crea una transacción para el costo minorista.

Las transacciones también pueden ser creadas manualmente, por ejemplo, si se le otorga un crédito a un cliente, se crea una transacción por el monto del crédito, o si se cobra una tarifa de instalación, se crea una transacción por la tarifa de instalación.

Las transacciones se agrupan para formar Facturas, que se envían al cliente para su pago.

Accediendo a las Transacciones

Las transacciones se pueden ver a nivel del sistema o por cliente:

Vista por Cliente:

1. Navegar a **Clientes → [Seleccionar Cliente]**
2. Hacer clic en la pestaña **Facturación**
3. Ver la lista de transacciones en la primera tarjeta

Vista a Nivel del Sistema:

1. Navegar a **Facturación → Transacciones** (desde el menú principal)
2. Ver todas las transacciones de todos los clientes

Widgets de Estadísticas de Transacciones

En la parte superior de la página de transacciones, cuatro tarjetas de estadísticas muestran resúmenes financieros:

Descripciones de Widgets:

- **Total de Transacciones** - Suma de todos los costos minoristas de transacciones (todo el tiempo)
- **Total de Transacciones No Facturadas** - Suma de transacciones que aún no se han incluido en una factura
- **Total de Transacciones Este Mes** - Suma de transacciones creadas este mes calendario
- **Total de Transacciones El Mes Pasado** - Suma de transacciones creadas el mes pasado

Formato de Valores:

- Valores superiores a 1,000: Mostrar como sufijo "k" (por ejemplo, $1.5k)
- Valores superiores a 1,000,000: Mostrar como sufijo "M" (por ejemplo, $2.3M)
- Valores superiores a 1,000,000,000: Mostrar como sufijo "B" (por ejemplo, $1.1B)

Lista de Transacciones

La tabla de transacciones muestra todas las transacciones con las siguientes columnas:

Descripciones de Columnas:

- **ID** - ID único de la transacción
- **Fecha** - Fecha de creación de la transacción
- **Título** - Nombre corto de la transacción
- **Descripción** - Descripción detallada de para qué es la transacción
- **Monto** - Costo minorista (positivo por cargos, negativo por créditos)
- **Factura** - ID de factura si la transacción ha sido facturada (enlace clicable)
- **Estado** - Marca de verificación si está facturada, guion si aún no está facturada

Acciones Por Fila:

Cada fila tiene un menú de acciones (⋮) con opciones:

- **Ver Detalles** - Abre el modal de detalles de la transacción
- **Descargar PDF de Factura** - Descargar PDF (solo si está facturada)
- **Anular Transacción** - Marcar la transacción como anulada (solo si no está facturada)

Tipos de Transacciones

Las transacciones se dividen en dos categorías principales:

Transacciones de Débito (Cargos)

Montos positivos que aumentan el saldo adeudado por el cliente:

- **Tarifas de Configuración de Servicio** - Cargos únicos cuando se provisiona el servicio
- **Tarifas Mensuales de Servicio** - Cargos recurrentes por servicios
- **Tarifas de Instalación** - Cargos por visitas de técnicos de campo
- **Cargos por Equipos** - Cargos por módems, enrutadores, tarjetas SIM
- **Cargos por Pago Tardío** - Penalizaciones por facturas vencidas
- **Cargos Manuales** - Cargos personalizados añadidos por el personal

Transacciones de Crédito (Pagos/Reembolsos)

Montos negativos que disminuyen el saldo adeudado por el cliente:

- **Pagos en Efectivo** - Cliente pagado en efectivo
- **Pagos con Tarjeta** - Cliente pagado con tarjeta de crédito/débito
- **Pagos por Transferencia Bancaria** - Cliente pagado a través de transferencia bancaria
- **Créditos de Cuenta** - Créditos de buena voluntad, compensación
- **Reembolsos** - Dinero devuelto al cliente
- **Descuentos** - Descuentos promocionales o de lealtad

Agregar una Transacción Manualmente

Haga clic en **"+ Agregar Transacción"** para abrir el modal de agregar transacción.

Transacción de Débito (Cargo):

Transacción de Crédito (Pago/Reembolso):

Descripciones de Campos:

- **Tipo de Transacción** - Seleccionar Débito (cargo) o Crédito (pago/reembolso)
- **Tipo de Crédito** - Si se selecciona Crédito, elegir método de pago (Efectivo, Tarjeta, Transferencia Bancaria)
- **Título** - Nombre corto para la transacción (requerido)
- **Descripción** - Explicación detallada (opcional)
- **Costo Minorista** - Monto que paga el cliente (requerido, número positivo)
- **Costo Mayorista** - Su costo (opcional, para seguimiento de márgenes)
- **Porcentaje de Impuesto** - Tasa de impuesto aplicada a esta transacción (opcional, por defecto a impuesto del producto o 0%)
- **Servicio** - Vincular la transacción a un servicio específico (opcional)
- **Sitio** - Vincular la transacción a un sitio específico (opcional)
- **Fecha de Transacción** - Fecha de la transacción (por defecto a hoy)

Validación:

- El título y el costo minorista son obligatorios
- El costo minorista debe ser un número positivo
- Si se selecciona el tipo de Crédito, se debe elegir un tipo de crédito

Qué Sucede:

1. Transacción creada en la base de datos
2. Aparece en la lista de transacciones del cliente
3. Incluida en el conteo de "Transacciones No Facturadas"
4. Disponible para inclusión en la próxima generación de facturas
5. Entrada de registro de actividad creada

Buscar y Filtrar Transacciones

Buscar

Utilice la barra de búsqueda para encontrar transacciones. Busca en:

- ID de Transacción

- Título
- Descripción
- ID de Factura

Filtros

Aplique filtros para reducir la lista de transacciones:

Filtros Disponibles:

- **Estado de Anulación** - Todos, Anulado, No Anulado
- **Estado de Factura** - Todos, Facturado, No Facturado

Acciones de Filtro:

- **Aplicar Filtros** - Aplicar filtros seleccionados a la lista
- **Restablecer Filtros** - Limpiar todos los filtros y mostrar todas las transacciones

Ordenar

Haga clic en cualquier encabezado de columna para ordenar:

- **ID** - Ordenar por ID de transacción (más nuevo/más antiguo)
- **Fecha** - Ordenar por fecha de transacción
- **Título** - Ordenar alfabéticamente
- **Monto** - Ordenar por costo minorista (más alto/más bajo)
- **Factura** - Ordenar por ID de factura

Haga clic nuevamente para invertir la dirección de orden (ascendente ↔ descendente).

Anular Transacciones

Las transacciones añadidas por error pueden ser **anuladas** (marcadas como eliminadas).

Requisitos:

- La transacción NO debe estar facturada
- Una vez facturadas, las transacciones no pueden ser anuladas (deben ser reembolsadas en su lugar)

Cómo Anular:

1. Localizar la transacción en la lista
2. Hacer clic en el menú de acciones (⋮)
3. Seleccionar **"Anular Transacción"**
4. Confirmar en el modal

Qué Sucede:

- Transacción marcada como `void = true`
- Ya no aparece en la lista de transacciones por defecto
- Excluida de la generación de facturas
- Puede ser vista filtrando por transacciones "Anuladas"
- Deducida del total de "Transacciones No Facturadas"

Transacciones vinculadas a la billetera (`wallet_ledger_id` establecido):

- El saldo de la billetera se revierte antes de anular (débito para deshacer un crédito, crédito para deshacer un débito)

- **Recarga de tarjeta de fondos** (`payment_type=wallet_topup_card`): billetera debitada y tarjeta reembolsada
- **Recarga en efectivo / pago de factura de billetera / otras líneas de billetera:** solo se ajusta la billetera; **sin reembolso de tarjeta**
- **Billetera insuficiente:** la anulación falla si los fondos recargados ya se han gastado (la transacción permanece activa)

Cargos por servicio / provisión de captura híbrida generalmente no tienen `wallet_ledger_id` en la línea de facturación. Anular esas líneas solo establece `void=true` hoy; se planea un crédito completo de billetera para anulación de servicios en la rama de documentos financieros.

API: `DELETE /crm/transaction/transaction_id/{transaction_id}`. Ver guía_del_sistema_de_pagos.md: Anular Transacción.

Nota: Anular NO es lo mismo que reembolsar. Anular significa "esta transacción nunca debió haber existido." Reembolsar significa "revertir una transacción válida."

Impuesto sobre Transacciones

Las transacciones pueden incluir impuestos, que se calculan automáticamente según la configuración de impuestos del producto o se especifican manualmente por transacción.

Comportamiento del Impuesto:

- **Transacciones de Débito (Cargos)** - Se aplica impuesto a los cargos basados en:
  - **Porcentaje de Impuesto del Producto** - Si la transacción está vinculada a un producto, el porcentaje de impuesto del producto se aplica automáticamente
  - **Anulación Manual** - El personal puede anular el porcentaje de impuesto al crear una transacción
  - **Monto del Impuesto** - Calculado como: `retail_cost × (tax_percentage / 100)`
  - **Formato de Visualización** - Se muestra como: `$10.00 (10%)` en listas de transacciones
- **Transacciones de Crédito (Pagos/Reembolsos)** - No se aplica impuesto a los créditos
  - El campo de porcentaje de impuesto está oculto para transacciones de crédito
  - El impuesto se establece automáticamente en 0% para todos los pagos y reembolsos
  - Los créditos reducen el saldo pendiente del cliente sin implicaciones fiscales

Ejemplo de Cálculo de Impuestos:

- Producto: Plan Móvil con 10% de impuesto, $50.00 costo minorista
- Cálculo Automático de Impuestos: $50.00 × 0.10 = $5.00
- Visualización: $5.00 (10%)

Impuesto Cero (NIL/Exento):

- Los productos pueden estar exentos de impuestos estableciendo el porcentaje de impuesto en 0
- El impuesto por defecto es 0% si no se especifica
- Las transacciones exentas de impuestos muestran "-" en la columna de Impuesto

Vista de Detalles de Transacción

Haga clic en una transacción para ver todos los detalles:

Transacciones Facturadas vs No Facturadas

Transacciones No Facturadas:

- Aún no incluidas en ninguna factura
- Disponibles para la próxima generación de facturas
- Pueden ser anuladas
- Cuentan para el total de "Transacciones No Facturadas"
- El estado muestra un guion (-)

Transacciones Facturadas:

- Incluidas en una factura
- No pueden ser anuladas (deben reembolsarse si es necesario)
- ID de factura clicable (enlaces a detalles de la factura)
- El estado muestra una marca de verificación (✓)
- No pueden ser modificadas

Generación de Factura:

Cuando genera una factura para un cliente:

1. El sistema encuentra todas las transacciones no facturadas para ese cliente
2. Opcionalmente filtrar por rango de fechas
3. Transacciones incluidas en la nueva factura
4. Campo `invoice_id` de la transacción poblado
5. Transacción ahora marcada como "facturada"

Ver `payments_invoices` para detalles de generación de facturas.

Flujos de Trabajo Comunes

Flujo de Trabajo 1: Crédito Manual por Interrupción del Servicio

1. El cliente llama: "El servicio estuvo inactivo durante 2 días"
2. El personal decide acreditar £10
3. Navegar a la pestaña **Facturación** del cliente
4. Hacer clic en **"+ Agregar Transacción"**
5. Seleccionar tipo de transacción **Crédito**
6. Seleccionar tipo de crédito **Pago en Efectivo**
7. Ingresar título: "Crédito por Interrupción del Servicio"
8. Ingresar descripción: "Compensación por interrupción de 2 días 8-9 de enero"
9. Ingresar costo minorista: 10.00
10. Seleccionar servicio afectado del desplegable
11. Hacer clic en **"Agregar Transacción"**
12. La transacción aparece con un monto de -£10.00
13. Se incluirá en la próxima factura como crédito

Flujo de Trabajo 2: Tarifa de Instalación Manual

1. El técnico de campo instala el servicio
2. El personal necesita cobrar una tarifa de instalación de £75
3. Navegar a la pestaña **Facturación** del cliente
4. Hacer clic en **"+ Agregar Transacción"**
5. Seleccionar tipo de transacción **Débito**
6. Ingresar título: "Tarifa de Instalación"
7. Ingresar descripción: "Visita del técnico de campo para instalación de fibra"
8. Ingresar costo minorista: 75.00
9. Ingresar costo mayorista: 45.00 (opcional, para seguimiento de márgenes)
10. Seleccionar servicio instalado
11. Seleccionar sitio donde se instaló
12. Hacer clic en **"Agregar Transacción"**
13. La transacción aparece en la lista de no facturadas
14. Se incluirá en la próxima factura

Flujo de Trabajo 3: Anular Transacción Duplicada

1. El personal nota una transacción duplicada
2. Verificar que la transacción NO esté facturada
3. Hacer clic en el menú de acciones (⋮) en la transacción duplicada
4. Seleccionar **"Anular Transacción"**
5. Confirmar en el modal
6. La transacción se elimina de la lista
7. El total no facturado disminuye en consecuencia

Flujo de Trabajo 4: Encontrar Transacciones para Factura

1. Necesita generar una factura mensual
2. Hacer clic en **Filtro de Factura: "No Facturado"**
3. Hacer clic en **Aplicar Filtros**

4. Ver todas las transacciones no facturadas
5. Anotar el monto total de los widgets
6. Navegar para generar factura
7. Seleccionar rango de fechas (por ejemplo, 1-31 de enero)
8. Las transacciones en el rango se incluyen en la factura

Resolución de Problemas

No se puede anular la transacción

- **Causa:** La transacción ya está facturada
- **Solución:** La transacción es parte del historial de facturas. Si se necesita reembolso, crear una transacción de Crédito en su lugar.
- **Causa:** Falló la reversión de recarga de billetera (saldo de billetera insuficiente)
- **Solución:** El cliente ya gastó los fondos recargados. Utilizar un débito/ajuste manual de billetera o flujos de reversión de documentos financieros; no se puede anular la línea de recarga hasta que la billetera cubra el monto.
- **Causa:** Falló el reembolso de tarjeta (solo recarga de fondos)
- **Solución:** La reversión de billetera se revierte automáticamente. Resolver el problema del proveedor de pagos y volver a intentar la anulación.

Aparecen transacciones duplicadas

- **Causa:** Servicio cobrado múltiples veces o error de provisión
- **Solución:** Anular la(s) transacción(es) duplicadas si no están facturadas. Si están facturadas, emitir crédito.

Transacción no aparece en la lista

- **Causa:** Filtros aplicados o transacción anulada

- **Solución:** Hacer clic en "Restablecer Filtros" para mostrar todas las transacciones. Para ver transacciones anuladas, filtrar por "Anular: Anulado".

El total no facturado no coincide con lo esperado

- **Causa:** Algunas transacciones ya están facturadas, o las transacciones anuladas están excluidas
- **Solución:** Aplicar filtro "Factura: No Facturada" para ver solo no facturadas. Verificar transacciones anuladas por separado.

No se puede agregar transacción (campo de cliente deshabilitado)

- **Causa:** Visualizando la página de transacciones específicas del cliente
- **Solución:** El cliente está preseleccionado. Si necesita agregar una transacción para un cliente diferente, ir a la página de Transacciones a Nivel del Sistema.

Documentación Relacionada

- `payments_invoices` - Generación y gestión de facturas
- `payments_process` - Procesamiento de pagos contra facturas
- `basics_payment` - Resumen de métodos de pago
- `csa_activity_log` - Visualización del historial de transacciones en el registro de actividad

Facturas de Clientes

Transacciones se agrupan para formar una factura, que se envía al cliente para su pago.

Las facturas tienen una fecha de inicio y una fecha de finalización, que es el período que cubre la factura, y una fecha de vencimiento, que es la fecha en que la factura debe ser pagada.

Las facturas pueden ser generadas automáticamente por el sistema, por ejemplo, cuando se factura un servicio, se crea una factura para el costo minorista, o pueden ser creadas manualmente, por ejemplo, si un cliente solicita una copia de una factura, o si un cliente es facturado por un cargo único.

Las facturas de clientes están completamente plantilladas con Mailjet y pueden ser personalizadas para incluir el logotipo de la empresa, dirección y detalles de pago, y pueden ser enviadas al cliente por correo electrónico, o descargadas como PDF.

Personalizando Plantillas de Factura

OmniCRM utiliza plantillas HTML con Jinja2 para generar facturas. Puede personalizar completamente el diseño de la factura, la marca, los colores y el diseño.

Ubicación de la Plantilla de Factura

Las plantillas de factura se almacenan en `OmniCRM-API/invoice_templates/`

Plantillas Predeterminadas:

- `sample_invoice_template.html` - Plantilla de factura de muestra
- `cifi_invoice_template.html` - Ejemplo de plantilla alternativa

Configuración:

La plantilla de factura activa se especifica en `OmniCRM-API/crm_config.yaml`:

```
invoice:
  template_filename: 'sample_invoice_template.html'
```

Variables de Plantilla Disponibles

Las plantillas de factura tienen acceso a las siguientes variables Jinja2:

Información de Factura:

- `{{ invoice_number }}` - ID único de la factura (por ejemplo, `INV-2025-001234`)
- `{{ date }}` - Fecha de emisión de la factura (formato ISO: `2025-01-10T12:00:00`)
- `{{ due_date }}` - Fecha de vencimiento del pago (por ejemplo, `2025-02-10`)
- `{{ start_date }}` - Fecha de inicio del período de facturación
- `{{ end_date }}` - Fecha de finalización del período de facturación
- `{{ total_amount }}` - Monto total de la factura antes de impuestos (numérico)
- `{{ total_tax }}` - Monto total de impuestos calculado de todas las transacciones (numérico)

Información del Cliente:

- `{{ client.name }}` - Nombre completo del cliente o nombre de la empresa
- `{{ client.address.address_line_1 }}` - Línea de dirección 1
- `{{ client.address.address_line_2 }}` - Línea de dirección 2
- `{{ client.address.city }}` - Ciudad
- `{{ client.address.state }}` - Estado/provincia
- `{{ client.address.zip_code }}` - Código postal
- `{{ client.address.country }}` - País

Artículos de Línea de Transacción:

Recorrer las transacciones usando:

```
{% for sub_transaction in transactions %}
    <tr>
        <td>{{ sub_transaction.transaction_id }}</td>
        <td>{{ sub_transaction.created.split("T")[0] }}</td>
        <td>{{ sub_transaction.title }}</td>
        <td>{{ sub_transaction.description }}</td>
        <td>${{ "%.2f"|format(sub_transaction.retail_cost) }}</td>
    </tr>
{% endfor %}
```

Campos de Transacción:

- `sub_transaction.transaction_id` - ID de transacción
- `sub_transaction.created` - Fecha/hora de la transacción
- `sub_transaction.title` - Título de la transacción
- `sub_transaction.description` - Descripción detallada
- `sub_transaction.retail_cost` - Monto del artículo
- `sub_transaction.tax_percentage` - Porcentaje de impuesto aplicado (por ejemplo, 10 para 10%)
- `sub_transaction.tax_amount` - Monto de impuesto calculado en dólares

Visualizando Impuestos en Plantillas:

```
<td>
    {% if sub_transaction.tax_amount and
sub_transaction.tax_amount > 0 %}
        ${{ "%.2f"|format(sub_transaction.tax_amount) }} ({{
sub_transaction.tax_percentage }}%)
    {% else %}
        -
    {% endif %}
</td>
```

Creando una Plantilla de Factura Personalizada

Paso 1: Copiar Plantilla Existente

```
cd OmniCRM-API/invoice_templates/
cp sample_invoice_template.html your_company_invoice_template.html
```

Paso 2: Personalizar HTML/CSS

Edite `your_company_invoice_template.html` para que coincida con su marca:

Áreas Clave de Personalización:

1. **Logotipo de la Empresa y Marca**

```
<!-- Reemplace con la URL de su logotipo -->
![Su Empresa](https://yourcompany.com/logo.png)

<!-- Actualice el nombre de la empresa -->
<h1>Nombre de Su Empresa</h1>
```

2. **Esquema de Color**

```
<style>
    /* Color de marca principal */
    .navbar {
        background: linear-gradient(to bottom right, #your-
color-1, #your-color-2);
    }

    /* Encabezados de tabla */
    .table thead th {
        background-color: #your-brand-color !important;
        color: white !important;
    }

    /* Botones y enlaces */
    .btn-primary {
        background-color: #your-brand-color;
    }
</style>
```

3. **Información de la Empresa en el Pie de Página**

```
<footer>
    <p>Nombre de Su Empresa</p>
    <p>Calle 123, Ciudad, País</p>
    <p>Teléfono: +1-555-123-4567 | Correo:
billing@yourcompany.com</p>
    <p>ABN/ID Fiscal: 12345678900</p>
</footer>
```

4. **Instrucciones de Pago**

```
<div class="payment-info">
    <h3>Métodos de Pago</h3>
    <p><strong>En Línea:</strong> Pague en
https://yourcompany.com/pay</p>
    <p><strong>Transferencia Bancaria:</strong></p>
    <ul>
        <li>Nombre de la Cuenta: Su Empresa Ltd</li>
        <li>BSB: 123-456</li>
        <li>Número de Cuenta: 987654321</li>
        <li>Referencia: {{ invoice_number }}</li>
    </ul>
</div>
```

5. **Términos y Condiciones**

```
<div class="terms">
    <h4>Términos de Pago</h4>
    <p>Pago debido dentro de los 30 días a partir de la fecha
de la factura.</p>
    <p>Cargos por pago tardío: 2% por mes sobre saldos
vencidos.</p>
    <p>Para consultas de facturación:
billing@yourcompany.com</p>
</div>
```

Paso 3: Actualizar Configuración

Edite `OmniCRM-API/crm_config.yaml`:

```
invoice:
  template_filename: 'your_company_invoice_template.html'
```

Paso 4: Reiniciar API

```
cd OmniCRM-API
sudo systemctl restart omnicrm-api
```

Paso 5: Probar Generación de Factura

1. Navegar a un cliente con transacciones
2. Generar una factura de prueba
3. Descargar PDF para verificar el formato
4. Enviar la factura por correo electrónico a usted mismo para probar la entrega del correo

Personalización Avanzada

Contenido Condicional:

Utilice condicionales de Jinja2 para mostrar/ocultar contenido:

```
{% if total_amount > 1000 %}
    <div class="high-value-notice">
        <p><strong>Nota:</strong> Saldo grande - Plan de pago
disponible a pedido.</p>
    </div>
{% endif %}

{% if client.address.country == "Australia" %}
    <p>GST Incluido: ${{ "%.2f"|format(total_amount * 0.10) }}</p>
{% endif %}
```

Soporte Multilingüe:

Cree plantillas específicas para cada idioma:

```
invoice_template_en.html
invoice_template_es.html
invoice_template_fr.html
```

Configure según la preferencia de idioma del cliente.

Cálculos Personalizados:

```
<!-- Mostrar subtotal y desglose de impuestos -->
<tr>
    <td colspan="4" class="text-right"><strong>Subtotal:</strong>
</td>
    <td>${{ "%.2f"|format(total_amount) }}</td>
</tr>
<tr>
    <td colspan="4" class="text-right"><strong>Impuesto:</strong>
</td>
    <td>${{ "%.2f"|format(total_tax) }}</td>
</tr>
<tr>
    <td colspan="4" class="text-right"><strong>Total:</strong>
</td>
    <td>${{ "%.2f"|format(total_amount + total_tax) }}</td>
</tr>
```

Nota: La variable `total_tax` se calcula automáticamente sumando el `tax_amount` de todas las transacciones en la factura. El impuesto de cada transacción se calcula en función de su campo `tax_percentage`, que por defecto es el `tax_percentage` del producto o 0% si no se especifica.

Código QR para Pago:

Genere un código QR para el pago móvil:

```
<div class="qr-payment">
    ![Escanear para Pagar](https://api.qrserver.com/v1/create-qr-
code/?size=150x150&data={{ payment_url }})
    <p>Escanee con su teléfono para pagar al instante</p>
</div>
```

Mejores Prácticas de Estilo para PDF

OmniCRM utiliza **WeasyPrint** para convertir HTML a PDF. Siga estas directrices:

CSS Soportado:

- La mayoría de las propiedades CSS 2.1
- CSS3 limitado (flexbox, algunas transformaciones)
- Fuentes web a través de `@font-face`

No Soportado:

- JavaScript
- CSS Grid (use tablas en su lugar)
- Animaciones complejas
- Algunas propiedades CSS modernas

Tamaño de Página y Márgenes:

```
@page {
    size: A4;
    margin: 1cm;
}

body {
    font-family: Arial, sans-serif;
    font-size: 10pt;
}
```

Estilo Específico para Impresión:

```
@media print {
    .no-print {
        display: none;
    }

    .page-break {
        page-break-after: always;
    }
}
```

Diseño de Tabla:

```
.table {
    table-layout: fixed;
    width: 100%;
}

.table th, .table td {
    word-wrap: break-word;
    padding: 4px;
}
```

Incorporación de Fuentes:

Para fuentes personalizadas, use fuentes seguras para la web o incorpore:

```
@font-face {
    font-family: 'YourFont';
    src: url('https://yourcompany.com/fonts/yourfont.woff2')
format('woff2');
}

body {
    font-family: 'YourFont', Arial, sans-serif;
}
```

Pruebas de Plantillas de Factura

Lista de Verificación de Pruebas:

1. **Inspección Visual:**
   - El logo se muestra correctamente
   - Los colores coinciden con las pautas de la marca
   - El texto es legible (no demasiado pequeño)
   - Las tablas se alinean correctamente
   - Todas las secciones están presentes
2. **Precisión de Datos:**
   - Detalles del cliente correctos
   - Los montos de las transacciones suman correctamente
   - Las fechas están formateadas correctamente
   - Todas las variables se sustituyen correctamente
3. **Calidad del PDF:**
   - Tamaño del archivo razonable (<5MB)
   - Imágenes nítidas y claras
   - Sin texto cortado o desbordado
   - Saltos de página en lugares apropiados
4. **Facturas de Varias Páginas:**
   - Los encabezados se repiten en cada página
   - Se muestran los números de página
   - Listas de transacciones largas se paginan correctamente
5. **Entrega por Correo Electrónico:**
   - PDF se adjunta al correo electrónico
   - Tamaño del archivo bajo el límite de Mailjet (15MB)
   - Se renderiza en Gmail, Outlook, Apple Mail

Comando de Prueba (Generación Manual):

Puede probar la generación de facturas a través de la API:

```
curl -X GET "http://localhost:5000/crm/invoice/{invoice_id}/pdf" \
     -H "Authorization: Bearer YOUR_TOKEN" \
     --output test_invoice.pdf
```

Problemas Comunes de Plantillas

Variables no sustituyendo:

- **Causa:** Error tipográfico en el nombre de la variable o datos faltantes
- **Solución:** Verifique la ortografía exactamente (sensible a mayúsculas y minúsculas), verifique que los datos existan en la base de datos

Estilo del PDF roto:

- **Causa:** Propiedad CSS no soportada
- **Solución:** Use propiedades CSS 2.1, pruebe con CSS compatible con WeasyPrint

Imágenes no mostrando:

- **Causa:** URLs relativas o recursos externos bloqueados
- **Solución:** Use URLs absolutas HTTPS, asegúrese de que las imágenes sean accesibles públicamente

Tablas desbordando la página:

- **Causa:** Anchos de columna fijos demasiado anchos
- **Solución:** Use anchos porcentuales, `table-layout: fixed`

Fuentes no renderizando:

- **Causa:** Fuente no incorporada o no disponible
- **Solución:** Use fuentes seguras para la web (Arial, Times New Roman, etc.) o incorpore correctamente fuentes personalizadas

Fallo en la generación de PDF:

- **Causa:** Errores de sintaxis HTML o fallo de WeasyPrint
- **Solución:** Valide HTML, verifique los registros de WeasyPrint, simplifique diseños complejos

Caché de PDF de Factura

Para mejorar el rendimiento y reducir la generación redundante de PDF, OmniCRM incluye un sistema de caché de PDF de factura. Cuando se genera por primera vez un PDF de factura, se almacena en la base de datos para solicitudes posteriores.

Cómo Funciona la Caché de PDF:

1. **Primera Solicitud** - Cuando se solicita un PDF de factura (descarga o correo electrónico), el sistema:
   - Genera el PDF a partir de la plantilla de factura
   - Codifica el PDF como Base64
   - Calcula un hash SHA256 del contenido del PDF
   - Almacena en la tabla `Invoice_PDF_Cache` con:
     - Referencia de ID de factura
     - Datos del PDF (codificados en Base64)
     - Nombre del archivo
     - Hash de contenido (para verificación de integridad)
     - Marca de tiempo de creación
2. **Solicitudes Posteriores** - Cuando se solicita la misma factura nuevamente:
   - El sistema verifica si hay un PDF en caché por invoice_id
   - Si la caché existe y es válida, devuelve el PDF en caché de inmediato
   - Actualiza la marca de tiempo `last_accessed` para rastrear el uso de la caché
3. **Invalidación de Caché** - Los PDFs en caché se invalidan cuando:
   - La factura se modifica (transacciones añadidas/eliminadas, detalles cambiados)
   - La plantilla de factura se actualiza
   - Se activa la limpieza manual de caché

Beneficios:

- **Rendimiento** - Entrega instantánea de PDF para solicitudes repetidas (sin retraso de regeneración)

- **Consistencia** - Mismo PDF para todas las descargas de una factura (a menos que la factura se modifique)
- **Carga del Servidor** - Reduce el uso de CPU por la generación de PDF
- **Experiencia del Usuario** - Un indicador de carga aparece durante la generación inicial, las solicitudes posteriores son instantáneas

Gestión de Caché:

La Caché de PDF de Factura es gestionada automáticamente por el sistema. Las entradas de caché antiguas o no utilizadas pueden ser eliminadas periódicamente en función de:

- Edad (por ejemplo, eliminar entradas de caché de más de 90 días)
- Patrones de acceso (eliminar entradas no accedidas en 30 días)
- Límites de almacenamiento (implementar límites de tamaño de caché si es necesario)

Comportamiento de la API:

Al descargar una factura a través de la API o la interfaz de usuario:

- Primera solicitud: Muestra un indicador de carga mientras se genera el PDF, luego se almacena en caché
- Solicitudes posteriores: Descarga inmediata desde la caché
- El acierto/fallo de caché es transparente para el usuario

Importante: Cuando actualice su plantilla de factura, limpie la caché para asegurar que las nuevas facturas usen el diseño actualizado:

```
-- Limpiar todos los PDFs de factura en caché (ejecutar en MySQL)
DELETE FROM Invoice_PDF_Cache;
```

O actualice `crm_config.yaml` para invalidar automáticamente la caché al cambiar la plantilla.

Accediendo a Facturas

Las facturas se pueden ver a nivel del sistema o por cliente:

Vista por Cliente:

1. Navegue a **Clientes → [Seleccionar Cliente]**
2. Haga clic en la pestaña **Facturación**
3. Vea la lista de facturas en la tercera tarjeta

Vista a Nivel del Sistema:

1. Navegue a **Facturación → Facturas** (desde el menú principal)
2. Vea todas las facturas de todos los clientes

Widgets de Estadísticas de Facturas

En la parte superior de la página de facturas, cuatro tarjetas de estadísticas muestran resúmenes financieros.

Descripciones de Widgets:

- **Total de Facturas** - Suma de todos los costos de venta de facturas (todo el tiempo) y conteo de facturas enviadas
- **Facturas No Pagadas** - Suma de facturas aún no pagadas y conteo de facturas no pagadas
- **Facturas Este Mes** - Suma de facturas creadas este mes calendario con conteo
- **Facturas del Mes Pasado** - Suma de facturas creadas el mes pasado con conteo

Formato de Valores:

- Valores superiores a 1,000: Mostrar como sufijo "k" (por ejemplo, $1.5k)
- Valores superiores a 1,000,000: Mostrar como sufijo "M" (por ejemplo, $2.3M)
- Valores superiores a 1,000,000,000: Mostrar como sufijo "B" (por ejemplo, $1.1B)

Indicadores de Tendencia:

- Widgets para "Este Mes" y "Mes Pasado" muestran cambio porcentual
- Flecha verde hacia arriba: Aumento respecto al período anterior
- Flecha roja hacia abajo: Disminución respecto al período anterior
- Flecha gris hacia la derecha: Sin cambio

Lista de Facturas

La tabla de facturas muestra todas las facturas con las siguientes columnas:

Descripciones de Columnas:

- **ID** - ID único de la factura
- **Título** - Título/descripción de la factura
- **Período** - Período de facturación (fecha de inicio - fecha de finalización) o "N/A" para facturas únicas
- **Fecha de Vencimiento** - Fecha de vencimiento del pago
- **Creado** - Fecha de creación de la factura
- **Monto** - Monto total de la factura (costo de venta)
- **Estado** - Pagado, No Pagado o Reembolsado
- **Acciones** - Acciones disponibles (varía según el estado)

Íconos de Acción:

- ⬇ **(Descargar)** - Descargar PDF de la factura
- □ **(Eliminar)** - Anular factura (solo si no está pagada)
- □ **(Pagar)** - Pagar factura en línea (solo si no está pagada)
- ✉ **(Correo Electrónico)** - Enviar correo electrónico de factura al cliente
- □ **(Reembolso)** - Reembolsar pago de Stripe (solo para facturas de Stripe pagadas)

Generando una Factura

Haga clic en **"+ Generar Factura Proforma"** para crear una nueva factura.

Descripciones de Campos:

- **Buscar Clientes** - Seleccionar cliente (solo se muestra en vista a nivel del sistema, prellenado en vista de cliente)
- **Título** - Título/nombre de la factura (opcional, por defecto "Factura para [Período]")
- **Fecha de Inicio** - Inicio del período de facturación (por defecto hace 14 días)
- **Fecha de Fin** - Fin del período de facturación (por defecto hoy)
- **Fecha de Vencimiento** - Fecha límite de pago (por defecto hoy)
- **Vista Previa de Transacciones** - Muestra todas las transacciones no facturadas en el rango de fechas con la capacidad de incluir/excluir transacciones específicas

Selección de Transacciones:

- ✓ **(Más Verde)** - Haga clic para excluir una transacción de la factura
- × **(X Roja)** - Haga clic para incluir una transacción previamente excluida
- **Seleccionar Todo** - Incluir todas las transacciones mostradas
- **Limpiar Todo** - Excluir todas las transacciones

- Las transacciones excluidas aparecen atenuadas con texto tachado
- Los totales se actualizan en tiempo real a medida que selecciona/deselecciona transacciones

Qué Ocurre:

1. El sistema encuentra todas las transacciones no facturadas para el cliente dentro del rango de fechas
2. Muestra la vista previa de transacciones con la capacidad de incluir/excluir transacciones individuales
3. Muestra el cálculo en tiempo real de subtotal, impuesto y total basado en las transacciones seleccionadas
4. Solo las transacciones seleccionadas (incluidas) se añaden a la factura
5. Genera el PDF de la factura y lo almacena en caché
6. Marca las transacciones seleccionadas como facturadas (`invoice_id` campo poblado)
7. Las transacciones excluidas permanecen no facturadas y disponibles para futuras facturas
8. La factura aparece en la lista con estado "No Pagada"

Ejemplos de Casos de Uso:

Facturación Mensual: Establezca la fecha de inicio al primer día del mes, la fecha de finalización al último día del mes, la vista previa muestra todas las transacciones no facturadas de ese período. Seleccione todas o excluya manualmente las específicas.

Factura Específica de Servicio: Use el mismo rango de fechas, luego excluya manualmente las transacciones no deseadas (por ejemplo, excluya transacciones no móviles para crear una factura solo móvil).

Factura Única: Establezca tanto la fecha de inicio como la fecha de finalización en el mismo día, la vista previa muestra solo las transacciones de esa fecha. Excluya cualquier cargo que no sea relevante para esta factura específica.

Visualizando Detalles de la Factura

Haga clic en cualquier fila de factura en la tabla para ver los detalles completos de la factura, incluyendo todas las transacciones, totales y acciones disponibles.

Modal de Detalles de la Factura:

- **Información de la Factura** - Muestra ID de factura, título, fechas, estado de pago y estado de anulación
- **Lista de Transacciones** - Muestra todas las transacciones incluidas en la factura con:
  - Fecha de transacción
  - Título y descripción
  - Costo de venta
  - Monto y porcentaje de impuesto (formateado como `$10.00 (10%)`)
  - Las transacciones exentas de impuestos muestran "-" en la columna de Impuestos
- **Resumen de Totales** - Cálculo en tiempo real que muestra:
  - Conteo de transacciones
  - Subtotal (suma de todos los costos de venta)
  - Impuesto (suma de todos los montos de impuestos)
  - Total de la Factura (subtotal + impuesto)

- **Botones de Acción** - Mismas acciones disponibles que en la tabla:
  - **Descargar PDF** - Descargar PDF de la factura (siempre disponible)
  - **Enviar Correo Electrónico** - Correo electrónico de la factura al cliente (facturas no anuladas)
  - **Pagar Factura** - Procesar pago (facturas no pagadas, no anuladas solo)
  - **Reembolso** - Reembolsar pago de Stripe (solo facturas de Stripe pagadas)
  - **Eliminar** - Anular factura (solo facturas no pagadas, no anuladas)

Descargando PDFs de Factura

Haga clic en el **ícono de descarga (⬇)** en la tabla o en el botón **"Descargar PDF"** en el modal de detalles de la factura para descargar una factura como PDF.

Proceso de Descarga:

1. Haga clic en el ícono de descarga junto a la factura
2. Un spinner de carga aparece durante la generación (solo la primera vez)
3. El navegador solicita guardar el archivo: `Invoice_01234.pdf`
4. El PDF se abre o se guarda en la carpeta de Descargas

Comportamiento de Caché de PDF:

- **Primera Descarga** - PDF generado a partir de la plantilla, almacenado en caché en la base de datos (puede tardar de 2 a 3 segundos)
- **Descargas Posteriores** - Descarga instantánea desde la caché
- **Invalidación de Caché** - Caché borrada si la factura se modifica o la plantilla se actualiza

Resolución de Problemas de Problemas de Descarga:

- **El spinner nunca se detiene** - Verifique la consola del navegador, la API puede estar caída
- **PDF en blanco o dañado** - Verifique la plantilla de factura en busca de errores de sintaxis

- **La descarga falla** - Verifique la configuración del bloqueador de ventanas emergentes, intente con otro navegador

Pagando Facturas

Haga clic en el **ícono de pago (□)** para pagar una factura en línea.

Proceso de Pago:

1. Haga clic en el ícono de pago en la factura no pagada
2. Se abre el modal de pago mostrando los detalles de la factura
3. Seleccione el método de pago:
    - **Transacción de Stripe** - Cargar tarjeta de crédito guardada (disponible para todos los usuarios)
    - **Efectivo** - Pago manual en efectivo (solo personal)
    - **Reembolso** - Aplicar reembolso como pago (solo personal)
    - **Transacción de POS** - Terminal de punto de venta (solo personal)
    - **Transferencia Bancaria** - Transferencia bancaria manual (solo personal)
4. Si se selecciona Stripe:
    - Seleccione la tarjeta de los métodos de pago guardados
    - La tarjeta predeterminada se preselecciona
    - Haga clic para seleccionar una tarjeta diferente

5. Si se selecciona otro método:
    - Ingrese el número de referencia (opcional)
6. Haga clic en **"Pagar Factura"** para procesar
7. El sistema procesa el pago:
    - **Stripe** - Carga la tarjeta a través de la API de Stripe
    - **Otros métodos** - Crea una transacción negativa por el monto del pago
8. El estado de la factura cambia a "Pagada"
9. Se muestra una notificación de éxito

Autocuidado vs Pago del Personal:

:doc:`Portal de Autocuidado <self_care_portal>` (Clientes):

- Solo disponible el pago de Stripe
- Debe tener un método de pago guardado
- Se muestra una advertencia si no existen métodos de pago
- Se proporciona un enlace para agregar un método de pago

Portal del Personal (Administradores):

- Todos los métodos de pago disponibles
- Puede marcar la factura como pagada manualmente (efectivo, POS, transferencia bancaria)
- Puede ingresar números de referencia para seguimiento

Advertencia de Método de Pago:

Si el cliente no tiene métodos de pago guardados, se muestra una advertencia sugiriendo que agregue un método de pago antes de poder pagar facturas.

Enviando Facturas por Correo Electrónico

Haga clic en el **ícono de correo electrónico (✉)** para enviar la factura al cliente.

Qué Ocurre:

1. Haga clic en el ícono de correo electrónico junto a la factura
2. El sistema recupera el PDF de la factura de la caché (o genera si no está en caché)
3. Envía el correo electrónico a través de Mailjet usando la plantilla `api_crmCommunicationCustomerInvoice`
4. El correo electrónico incluye:
   - PDF de la factura como archivo adjunto
   - Nombre del cliente
   - Número de factura y fecha de vencimiento
   - Monto total a pagar
   - Enlace para pagar la factura en línea
   - Enlace para ver/descargar la factura
5. Notificación de éxito: "Correo electrónico de factura enviado con éxito"

Destinatarios del Correo Electrónico:

El correo electrónico se envía a todos los contactos del cliente con tipo "facturación" o contacto principal si no existe un contacto de facturación.

Variables de Plantilla de Correo Electrónico:

- `{{ var:customer_name }}` - Nombre completo del cliente
- `{{ var:invoice_number }}` - ID de la factura
- `{{ var:invoice_date }}` - Fecha de emisión de la factura
- `{{ var:due_date }}` - Fecha de vencimiento del pago
- `{{ var:total_amount }}` - Monto total a pagar
- `{{ var:invoice_url }}` - Enlace para ver/descargar PDF
- `{{ var:pay_url }}` - Enlace para pagar la factura en línea

Resolución de Problemas de Problemas de Correo Electrónico:

- **Correo electrónico no enviado** - Verifique las credenciales de la API de Mailjet en `crm_config.yaml`
- **Cliente no recibe** - Verifique las direcciones de correo electrónico de contacto del cliente
- **PDF no adjunto** - Verifique que la generación de PDF haya tenido éxito (intente descargar primero)

Anulando Facturas

Haga clic en el **ícono de eliminar (🗑)** para anular una factura.

Requisitos:

- La factura debe estar **No Pagada**
- Las facturas pagadas no pueden ser anuladas (deben ser reembolsadas en su lugar)

Cómo Anular:

1. Localice la factura no pagada en la lista
2. Haga clic en el ícono de eliminar (🗑)
3. Confirme en el modal:

Qué Ocurre:

- La factura se marca como `void = true`
- Todas las transacciones se desvinculan de la factura (`invoice_id` se establece en null)
- Las transacciones se convierten nuevamente en "no facturadas"
- Las transacciones pueden incluirse en una nueva factura
- La factura aparece en la lista con el prefijo "Anulada:" en el título
- Las acciones de la factura están deshabilitadas (sin descarga, pago o correo electrónico)
- Se puede ver filtrando por facturas "Anuladas"

Notas Importantes:

- Anular NO es lo mismo que reembolsar
- **Anular** = "Esta factura nunca debió existir" (error de facturación, duplicado)
- **Reembolso** = "Revertir una factura válida pagada" (devolver dinero al cliente)

Reembolsando Facturas

Haga clic en el **ícono de reembolso (□)** para reembolsar una factura pagada.

Requisitos:

- La factura debe estar **Pagada**
- La factura debe haber sido pagada a través de **Stripe**
- La factura debe tener una `payment_reference` válida (ID de intención de pago de Stripe)
- Solo disponible para usuarios del personal (no Autocuidado)

Cómo Reembolsar:

1. Localice la factura de Stripe pagada
2. Haga clic en el ícono de reembolso (□)
3. Se abre el modal de confirmación de reembolso:

4. Haga clic en **"Confirmar Reembolso"**
5. El sistema procesa el reembolso de Stripe:
    - Llama a la API de Stripe para reembolsar el pago
    - Crea una transacción de reembolso en Stripe
    - Actualiza la factura con `refund_reference`
6. El estado de la factura cambia a "Reembolsada"
7. Se muestra una notificación de éxito

Qué Ocurre Después del Reembolso:

- La factura permanece en el sistema (no anulada)
- El estado muestra "Reembolsada"

- Las transacciones permanecen vinculadas a la factura
- El cliente recibe el reembolso al método de pago original (3-7 días hábiles)
- El panel de Stripe muestra la transacción de reembolso

Restricciones de Reembolso:

- No se pueden reembolsar facturas pagadas en efectivo, POS o transferencia bancaria (se requiere reversión manual)
- No se puede reembolsar parcialmente (solo monto total de la factura)
- No se puede reembolsar dos veces

Buscando y Filtrando Facturas

Buscar

Utilice la barra de búsqueda para encontrar facturas. Busca en:

- ID de factura
- Título de factura
- Nombre del cliente (solo vista a nivel del sistema)

Filtros

Aplique filtros para reducir la lista de facturas:

Filtros Disponibles:

- **Estado de Anulación** - Todos, Anulada, No Anulada
- **Estado de Pago** - Todos, Pagada, No Pagada

Acciones de Filtro:

- **Aplicar Filtros** - Aplicar filtros seleccionados a la lista
- **Restablecer Filtros** - Limpiar todos los filtros y mostrar todas las facturas

Ordenar

Haga clic en cualquier encabezado de columna para ordenar:

- **ID** - Ordenar por ID de factura (más nuevo/más antiguo)
- **Título** - Ordenar alfabéticamente
- **Fecha de Vencimiento** - Ordenar por fecha de vencimiento
- **Creado** - Ordenar por fecha de creación
- **Monto** - Ordenar por costo de venta (más alto/más bajo)
- **Estado** - Ordenar por estado de pago (pagado primero o no pagado primero)

Haga clic nuevamente para invertir la dirección de ordenación (ascendente ↔ descendente).

Paginación

Navegue a través de grandes listas de facturas con controles de página que muestran la página actual, el total de páginas y un selector de elementos por página (10, 25, 50 o 100 elementos).

Flujos de Trabajo Comunes de Facturas

Flujo de Trabajo 1: Facturación Mensual con Vista Previa de Transacción

1. Llega el fin de mes (por ejemplo, 31 de enero)
2. Navegue a **Facturación → Facturas**
3. Haga clic en **"+ Generar Factura Proforma"**
4. Seleccione cliente (o hágalo por cliente si factura a muchos clientes)
5. Establezca las fechas:
    - Fecha de Inicio: 2025-01-01
    - Fecha de Fin: 2025-01-31

    - Fecha de Vencimiento: 2025-02-15 (15 días a partir de ahora)
    - Título: "Servicios de Enero 2025" (opcional)
6. La sección **Vista Previa de Transacciones** aparece mostrando todas las transacciones no facturadas de enero
7. Revise la vista previa:
    - Todas las transacciones están incluidas por defecto
    - Verifique los totales: Subtotal, Impuesto y Total de la Factura
    - Verifique que todos los cargos sean correctos
8. Haga clic en **"Generar Factura"** (el botón muestra el conteo de transacciones, por ejemplo, "Generar Factura (15)")
9. Factura creada con todas las transacciones seleccionadas
10. Haga clic en la fila de la factura para ver detalles y verificar
11. Haga clic en el botón **"Enviar Correo Electrónico"** en el modal de detalles o en el ícono de correo electrónico en la tabla
12. El cliente recibe un correo electrónico de factura con PDF y enlace de pago

Flujo de trabajo 2: Facturación Selectiva de Transacciones

1. El cliente tiene múltiples servicios (Móvil + Internet) y cargos varios
2. Quiere facturas separadas para cada servicio
3. **Generar primera factura (Servicios Móviles):**
    - Haga clic en **"+ Generar Factura Proforma"**
    - Título: "Servicios Móviles - Enero 2025"
    - Inicio/Fin: 1-31 de enero
    - Fecha de vencimiento: 15 de febrero
    - En la vista previa de transacciones, **excluir** todas las transacciones no móviles:
        - Haga clic en el botón **X** junto a las transacciones de Internet
        - Haga clic en el botón **X** junto a los cargos varios
        - Solo permanecen seleccionadas las transacciones de servicios móviles
    - Verifique que los totales reflejen solo servicios móviles

- Haga clic en **"Generar Factura"** (muestra el conteo de transacciones móviles)

4. **Generar segunda factura (Servicios de Internet):**
   - Haga clic en **"+ Generar Factura Proforma"** nuevamente
   - Título: "Servicios de Internet - Enero 2025"
   - Inicio/Fin: 1-31 de enero (mismo período)
   - En la vista previa de transacciones:
     - Las transacciones móviles ya facturadas (no aparecen)
     - Excluir cargos varios usando el botón **X**
     - Solo permanecen las transacciones de servicios de Internet
   - Haga clic en **"Generar Factura"**
5. **Generar tercera factura (Cargos Adicionales):**
   - Haga clic en **"+ Generar Factura Proforma"** nuevamente
   - Título: "Cargos Adicionales - Enero 2025"
   - Solo los cargos varios no facturados aparecen en la vista previa
   - Haga clic en **"Seleccionar Todo"** para incluir todos
   - Haga clic en **"Generar Factura"**
6. Envíe las tres facturas al cliente por correo electrónico

Flujo de trabajo 3: Exclusión de Transacciones Disputadas o Pendientes

1. Llega el final del período de facturación
2. Navegue a la pestaña **Facturación** del cliente
3. Haga clic en **"+ Generar Factura Proforma"**
4. Establezca las fechas del período de facturación
5. La vista previa de transacciones muestra 20 transacciones
6. El cliente ha disputado un cargo y otro está pendiente de investigación
7. En la vista previa de transacciones:
   - Localice la transacción disputada (por ejemplo, "Cargo por exceso de datos")
   - Haga clic en el botón **X** para excluirla
   - Localice la transacción pendiente (por ejemplo, "Cargo por instalación")
   - Haga clic en el botón **X** para excluirla

    - El conteo de transacciones se actualiza: "18 Transacciones seleccionadas"
    - Los totales se recalculan automáticamente
8. Revise los totales actualizados (excluye los montos disputados)
9. Haga clic en **"Generar Factura (18)"**
10. La factura se genera solo con transacciones aprobadas
11. Las transacciones disputadas/pendientes permanecen no facturadas para el próximo ciclo de facturación

Flujo de trabajo 4: Revisión Rápida y Ajuste de Factura

1. El personal genera la factura mensual
2. La vista previa de transacciones muestra un total inesperadamente alto
3. Revise cada transacción en la vista previa:
    - Nota un cargo duplicado por el mismo servicio
    - Haga clic en **X** para excluir el duplicado
    - Nota una transacción de prueba que no debería ser facturada
    - Haga clic en **X** para excluir la transacción de prueba
4. Los totales se actualizan en tiempo real
5. Verifique que el nuevo total coincida con el monto esperado
6. Haga clic en **"Generar Factura"** con las transacciones corregidas
7. Regrese y anule/elimine las transacciones excluidas si es necesario
8. Envíe la factura al cliente con confianza

Flujo de trabajo 5: Factura de Instalación Única

1. El técnico de campo completa la instalación
2. El personal agrega la transacción de instalación manualmente
3. Navegue a la pestaña **Facturación** del cliente
4. Haga clic en **"+ Generar Factura Proforma"**
5. Establezca las fechas:
    - Fecha de inicio: hoy
    - Fecha de finalización: hoy

    - Fecha de vencimiento: hoy + 7 días
    - Título: "Servicios de Instalación"
6. La vista previa de transacciones muestra solo las transacciones de hoy
7. Verifique que el cargo por instalación aparezca
8. Excluya cualquier cargo recurrente usando el botón **X** (si está presente)
9. Haga clic en **"Generar Factura"**
10. Envíe al cliente inmediatamente
11. El cliente paga en línea a través de Stripe

Flujo de trabajo 6: Revisión de Factura Antes del Contacto con el Cliente

1. El cliente llama con una pregunta de facturación
2. El personal navega a la lista de facturas del cliente
3. **Haga clic en la fila de la factura** para abrir el modal de Detalles de la Factura
4. Revise la información de la factura:
    - ID de factura, fechas, estado
    - Todas las transacciones incluidas con descripciones
    - Desglose de impuestos por transacción
    - Subtotal, Impuesto y Totales
5. Responda a las preguntas del cliente con detalles exactos
6. Si el cliente solicita un PDF, haga clic en el botón **"Descargar PDF"** en el modal
7. Si el cliente solicita reenvío de correo electrónico, haga clic en el botón **"Enviar Correo Electrónico"**
8. Cierre el modal cuando haya terminado

Flujo de trabajo 7: Corrección de Error de Facturación

1. El cliente informa un cargo incorrecto
2. El personal hace clic en la fila de la factura para ver los detalles
3. Revisa la lista de transacciones en el modal de Detalles de la Factura

4. Identifica la transacción incorrecta
5. La factura está sin pagar, por lo que puede ser anulada
6. Haga clic en el botón **"Eliminar"** en el pie del modal
7. Confirme la anulación
8. Las transacciones vuelven a ser no facturadas
9. El personal modifica o elimina la transacción incorrecta de la lista de transacciones
10. Genere una nueva factura con las transacciones corregidas:
    - Use la vista previa de transacciones para excluir la transacción corregida si es necesario
    - Incluya solo cargos válidos
11. Envíe la factura corregida al cliente

Flujo de trabajo 8: Procesamiento de Múltiples Pagos

1. El cliente trae efectivo para pagar múltiples facturas
2. Navegue a la pestaña **Facturación** del cliente
3. Vea las facturas no pagadas
4. Haga clic en la primera fila de la factura para ver los detalles
5. Verifique el monto y las transacciones
6. Haga clic en el botón **"Pagar Factura"** en el pie del modal
7. Seleccione el método de pago **"Efectivo"**
8. Ingrese la referencia: "Efectivo pagado 2025-01-15"
9. Haga clic en **"Pagar Factura"**
10. El modal se cierra, la factura se marca como "Pagada"
11. Repita para las facturas restantes
12. Todas las facturas ahora marcadas como "Pagadas"

Flujo de trabajo 9: Manejo de Solicitud de Reembolso

1. El cliente solicita un reembolso por sobrepago
2. El personal verifica que la factura fue pagada a través de Stripe

3. Navegue a la factura en la lista
4. Haga clic en la fila de la factura para ver los detalles
5. Verifique la información de pago y el monto
6. Haga clic en el botón **"Reembolso"** en el pie del modal (solo aparece para facturas de Stripe)
7. Confirme el reembolso
8. El sistema procesa el reembolso de Stripe
9. El estado de la factura cambia a "Reembolsado"
10. El cliente recibe el reembolso en 3-7 días hábiles
11. El personal hace un seguimiento con el cliente para confirmar la recepción

Solución de Problemas

No se puede generar la factura - No se encontraron transacciones

- **Causa:** No hay transacciones no facturadas en el rango de fechas especificado
- **Solución:** Verifique la lista de transacciones, confirme que existan transacciones y que no estén ya facturadas. Ajuste el rango de fechas o elimine el filtro.

La generación de PDF de la factura falla

- **Causa:** Error de sintaxis en la plantilla, fallo de WeasyPrint o falta de datos del cliente
- **Solución:** Verifique el HTML de la plantilla de la factura en busca de errores, confirme que los campos de dirección del cliente estén poblados, revise los registros de la API.

El pago falla con error de Stripe

- **Causa:** Tarjeta rechazada, fondos insuficientes, tarjeta expirada o problema con la API de Stripe
- **Solución:** Intente con un método de pago diferente, confirme que la tarjeta sea válida, verifique el panel de Stripe para conocer el motivo del rechazo.

No se puede anular la factura

- **Causa:** La factura ya está pagada
- **Solución:** Las facturas pagadas no se pueden anular. Si se necesita un reembolso, utilice la función de reembolso para facturas de Stripe o cree una transacción de crédito manualmente.

El correo electrónico de la factura no se envía

- **Causa:** Credenciales de la API de Mailjet inválidas, el cliente no tiene contacto de facturación o falta la plantilla de correo electrónico
- **Solución:** Verifique la configuración de Mailjet en `crm_config.yaml`, revise los contactos del cliente, confirme que exista la plantilla de correo electrónico de la factura.

El botón de reembolso no aparece

- **Causa:** La factura fue pagada en efectivo/POS/transferencia bancaria (no Stripe), o la factura no está pagada
- **Solución:** El botón de reembolso solo aparece para pagos de Stripe. Para otros métodos de pago, cree una transacción de crédito manual.

Descargar PDF muestra el diseño de plantilla antiguo

- **Causa:** PDF en caché antes de la actualización de la plantilla
- **Solución:** Limpiar la caché del PDF de la factura: `DELETE FROM Invoice_PDF_Cache WHERE invoice_id = X;`

El cliente no puede pagar la factura (sin métodos de pago)

- **Causa:** No hay métodos de pago guardados en el portal de Autocuidado
- **Solución:** El cliente debe agregar una tarjeta de crédito en la página de **Métodos de Pago** antes de pagar las facturas.

Se generan múltiples facturas para el mismo período

- **Causa:** El personal generó la factura dos veces o los rangos de fechas se superponen

- **Solución:** Anule la factura duplicada. Ajuste los rangos de fechas para evitar superposiciones. Use la vista previa de transacciones para asegurar conjuntos de transacciones únicos.

La vista previa de transacciones no muestra transacciones

- **Causa:** Todas las transacciones en el rango de fechas ya están facturadas o no existen transacciones
- **Solución:** Verifique que el rango de fechas sea correcto. Revise la lista de transacciones para confirmar que existan transacciones. Filtre las facturas para ver cuál factura contiene las transacciones.

No se puede excluir la transacción de la generación de la factura

- **Causa:** La transacción ya está facturada o hay un problema con el navegador
- **Solución:** Verifique que la transacción aparezca en la vista previa con una marca de verificación. Actualice la página y vuelva a intentarlo. Limpie la caché del navegador si el problema persiste.

El total de la factura no coincide con el monto esperado

- **Causa:** Se incluyeron transacciones inesperadas, el impuesto no se calculó o las transacciones excluidas aún se cuentan
- **Solución:** Revise la vista previa de transacciones cuidadosamente. Verifique el costo minorista y el impuesto de cada transacción. Confirme que las transacciones excluidas estén atenuadas. Verifique la insignia de conteo de transacciones en el botón Generar Factura.

El botón Generar Factura está deshabilitado

- **Causa:** No se seleccionaron transacciones o el rango de fechas es inválido
- **Solución:** Asegúrese de que al menos una transacción esté incluida (no excluida). Verifique que la Fecha de Inicio sea anterior a la Fecha de Fin. Confirme que la Fecha de Vencimiento esté establecida.

El modal de Detalles de la Factura no se abre

- **Causa:** Error de JavaScript o la página no se cargó completamente

- **Solución:** Actualice la página. Verifique la consola del navegador en busca de errores. Pruebe con un navegador diferente. Verifique la conexión a Internet.

El impuesto de la transacción no se muestra en los Detalles de la Factura

- **Causa:** La transacción tiene un impuesto del 0% o el tax_amount es nulo
- **Solución:** Verifique que la transacción tenga el tax_percentage establecido. Confirme que el tax_amount se calculó cuando se creó la transacción. Actualice la transacción si es necesario.

Faltan botones de acción en el modal de Detalles de la Factura

- **Causa:** La factura está anulada o el usuario no tiene permisos
- **Solución:** Las facturas anuladas solo muestran el botón Descargar PDF. Verifique el estado de la factura. Revise el rol y los permisos del usuario.

Documentación Relacionada

- `integrations_mailjet` - Entrega de facturas por correo electrónico y plantillas
- `administration_configuration` - Configuración de la plantilla de factura
- `payments_transaction` - Creación de transacciones que aparecen en las facturas
- `payments_process` - Procesamiento de pagos de facturas
- `payment_system_guide` - Referencia de la API de pagos y configuración del proveedor

Gestión de Métodos de Pago

El sistema de Métodos de Pago de OmniCRM permite a los clientes y al personal gestionar de forma segura las tarjetas de pago utilizando **procesamiento de pagos de múltiples proveedores** (Stripe, PayPal, etc.). Los métodos de pago permiten la facturación automática de servicios, pagos únicos y cargos recurrentes sin almacenar datos sensibles de tarjetas en OmniCRM.

Vea también: Guía del Sistema de Pagos, Resumen de Facturación, Procesamiento de Pagos, Facturas.

Resumen

El sistema de métodos de pago proporciona:

- **Almacenamiento Seguro de Tarjetas** - Tarjetas tokenizadas por proveedores de pagos (Stripe, PayPal), nunca almacenadas en OmniCRM
- **Soporte Multi-Proveedor** - Métodos de pago de Stripe y PayPal soportados
- **Múltiples Tarjetas** - Los clientes pueden almacenar múltiples métodos de pago
- **Selección Predeterminada** - Designar el método de pago preferido para cargos automáticos
- **Seguimiento de Vencimientos** - Monitorear y actualizar tarjetas que están por vencer
- **Autoservicio** - Los clientes pueden gestionar sus propias tarjetas a través del Portal de Autocuidado
- **Gestión del Personal** - El personal de soporte puede agregar/eliminar tarjetas en nombre de los clientes

Métodos de Pago Soportados:

- Tarjetas de Crédito (Visa, Mastercard, American Express, Discover)
- Tarjetas de Débito
- Tarjetas Prepagadas (si son soportadas por la red de tarjetas)

No Almacenado en OmniCRM:

Los detalles de la tarjeta son tokenizados por proveedores de pagos y almacenados de forma segura. OmniCRM solo almacena:

- Proveedor de pago (stripe, paypal)
- Marca de la tarjeta (Visa, Mastercard, etc.)
- Últimos 4 dígitos
- Mes/año de vencimiento
- Nombre/apodo del titular de la tarjeta
- Token de método de pago específico del proveedor

Accediendo a Métodos de Pago

Desde la Página del Cliente:

1. Navegue a **Clientes → [Seleccionar Cliente]**
2. Haga clic en la pestaña **Facturación**
3. Desplácese a la sección **Métodos de Pago**

O directamente:

Desde el Panel de Tarjetas que Vencen:

Vea todos los clientes con tarjetas que están por vencer:

Esto muestra una lista a nivel de sistema de tarjetas que vencen dentro de los próximos 60 días.

Lista de Métodos de Pago

La tabla de métodos de pago muestra todas las tarjetas almacenadas para un cliente:

Descripciones de Columnas:

- **Apodo** - Nombre amigable para la tarjeta (por ejemplo, "Tarjeta Personal", "Visa de Trabajo")
- **Emisor** - Marca de la tarjeta y últimos 4 dígitos
- **Vencimiento** - Mes/año de vencimiento (formato MM/YYYY)
- **Agregado** - Fecha en que se agregó la tarjeta a la cuenta
- **Predeterminado** - Marca de verificación indica el método de pago predeterminado para cargos automáticos

Acciones Por Tarjeta:

Cada fila tiene un menú de acciones (⋮) con opciones:

- **Establecer como Predeterminado** - Hacer de este el método de pago predeterminado
- **Eliminar** - Eliminar la tarjeta de la cuenta

Agregando un Método de Pago

Haga clic en **"Agregar Método de Pago"** para abrir el modal de pago seguro.

Paso 1: Ingresar Detalles de la Tarjeta

El formulario de pago seguro aparece (impulsado por Stripe Elements o PayPal SDK):

Campos Requeridos:

- **Información de la Tarjeta** - Número de tarjeta, vencimiento, CVC (validado por Stripe)
- **Nombre del Titular de la Tarjeta** - Nombre en la tarjeta
- **País/Región** - País de facturación

Campos Opcionales:

- **Apodo de la Tarjeta** - Etiqueta amigable para distinguir entre tarjetas

Seguridad:

- Los detalles de la tarjeta se ingresan directamente en un iframe seguro alojado por el proveedor (Stripe Elements / PayPal SDK)
- OmniCRM nunca ve ni almacena números de tarjeta completos
- Cumplimiento de PCI DSS manejado por el proveedor de pagos
- La validación en tiempo real previene números de tarjeta inválidos

Paso 2: Enviar y Tokenizar

Cuando haga clic en **"Agregar Método de Pago"**:

1. **Validación del Lado del Cliente:**
   - El proveedor de pagos valida el formato del número de tarjeta
   - Verifica que la fecha de vencimiento esté en el futuro
   - Verifica el formato del CVC
2. **Tokenización:**
   - Los detalles de la tarjeta se envían directamente al proveedor de pagos (no a OmniCRM)
   - El proveedor crea un token seguro (por ejemplo, `pm_1A2B3C4D` para Stripe)
   - El token se devuelve a OmniCRM
3. **Procesamiento del Servidor:**
   - OmniCRM guarda el token en el registro del cliente con el identificador del proveedor
   - Almacena los últimos 4 dígitos, la marca, el vencimiento y el nombre del proveedor para su visualización
   - Ningún número de tarjeta completo toca nunca los servidores de OmniCRM

Paso 3: Confirmación

Aparece un mensaje de éxito:

> Su Visa terminada en 1234 ha sido agregada a su cuenta.

La nueva tarjeta aparece en la tabla de métodos de pago.

Selección Automática Predeterminada:

- Si esta es la primera tarjeta del cliente, se establece automáticamente como predeterminada
- Si el cliente ya tiene tarjetas, la nueva tarjeta se agrega como no predeterminada
- El cliente puede cambiar la predeterminada después de agregar

Estableciendo el Método de Pago Predeterminado

El método de pago predeterminado se utiliza para:

- Cargos automáticos recurrentes de servicios
- Pagos de facturas
- Recargas y recargas
- Transacciones únicas (a menos que se especifique lo contrario)

Para Cambiar el Predeterminado:

1. Localice la tarjeta que desea establecer como predeterminada en la tabla de métodos de pago
2. Haga clic en el **menú de acciones (⋮)** junto a la tarjeta
3. Seleccione **"Establecer como Predeterminado"**
4. Aparece la confirmación

   La Visa terminada en 5678 es ahora su método de pago predeterminado.

La marca de verificación se mueve a la tarjeta recién seleccionada.

Indicador Visual:

Las tarjetas predeterminadas muestran:

en la columna Predeterminado, típicamente con una insignia de marca de verificación verde.

Eliminando un Método de Pago

Elimine tarjetas que estén vencidas, perdidas o que ya no se necesiten.

Paso 1: Iniciar Eliminación

1. Encuentre la tarjeta a eliminar en la tabla de métodos de pago
2. Haga clic en el ícono de **basura** en esa fila

Si el cliente tiene solo un método de pago activo, el ícono de basura está **deshabilitado**. Al pasar el cursor sobre él, se muestra un tooltip:

> No se puede eliminar su único método de pago. Agregue otro método de pago primero.

Agregue un segundo método de pago antes de poder eliminar el existente.

Paso 2: Confirmar Eliminación

Cuando se permite la eliminación, aparece un modal de confirmación:

> ¿Está seguro de que desea eliminar este método de pago?
>
> [Cancelar] [Eliminar Método de Pago]

Si el método que se está eliminando es el predeterminado actual y existen otros métodos activos, el modal señala que otro método activo será promovido a predeterminado después de la eliminación.

Haga clic en **"Eliminar Método de Pago"** para confirmar.

Paso 3: Eliminación Completa

Mensaje de éxito:

La tarjeta se elimina de la tabla y se elimina del proveedor de pagos.

Restricciones Importantes:

- **No se puede eliminar el único método de pago activo** - El control de eliminación está deshabilitado (con tooltip). La API también rechaza la solicitud. Agregue otro método de pago primero.
- **Eliminar el predeterminado cuando existen otros** - Otro método de pago activo utilizable se promueve a predeterminado automáticamente.

- **Sin deshacer** - La eliminación es permanente; el cliente debe volver a agregar el método si es necesario.

Gestión de Tarjetas que Vencen

OmniCRM rastrea las fechas de vencimiento de las tarjetas y proporciona herramientas para actualizar proactivamente las tarjetas que están por vencer.

Panel de Tarjetas que Vencen

Navegue a **Facturación → Tarjetas que Vencen** para ver una lista a nivel de sistema:

> Días hasta la Acción de Vencimiento de la Tarjeta del Cliente John Smith Visa ****1234 02/2025 12 días Actualizar Acme Corp MC5678 03/2025 45 días Actualizar Jane Doe Amex**9012 01/2025 VENCIDA Actualizar

Filtros:

- **Rango de Vencimiento** - Próximos 30/60/90 días o ya vencidos
- **Tipo de Cliente** - Individual vs Empresa
- **Tipo de Servicio** - Filtrar por servicio que requiere método de pago

Acciones:

- **Actualizar** - Abre la página de métodos de pago del cliente para agregar una nueva tarjeta
- **Notificar** - Enviar recordatorio por correo electrónico al cliente (si Mailjet está configurado)

Notificaciones de Vencimiento

Si Mailjet está configurado, se envían correos electrónicos automáticos:

- **60 días antes del vencimiento** - Primer recordatorio
- **30 días antes del vencimiento** - Segundo recordatorio
- **7 días antes del vencimiento** - Advertencia final
- **Al vencimiento** - Aviso de que la tarjeta ha vencido

Los clientes pueden hacer clic en un enlace en el correo electrónico para actualizar su método de pago a través del portal de Autocuidado.

Variables de Plantilla de Correo Electrónico:

Las plantillas de Mailjet reciben:

- Nombre del cliente
- Marca de la tarjeta y últimos 4 dígitos
- Fecha de vencimiento
- Enlace a la página de métodos de pago del Autocuidado

Vea `integrations_mailjet` para la configuración de la plantilla de correo electrónico.

Actualizando una Tarjeta que Vence

Flujo de Trabajo Recomendado:

1. El cliente recibe un correo electrónico de notificación de vencimiento
2. El cliente inicia sesión en el portal de Autocuidado
3. Navega a **Facturación → Métodos de Pago**

4. Hace clic en **"Agregar Método de Pago"**
5. Ingresa los detalles de la nueva tarjeta (la misma tarjeta con vencimiento actualizado o tarjeta de reemplazo)
6. Establece la nueva tarjeta como predeterminada
7. Elimina la tarjeta antigua/vencida

Flujo de Trabajo del Personal:

Si el cliente llama al soporte:

1. El personal abre la cuenta del cliente
2. Navega a **Facturación → Métodos de Pago**
3. Agrega una nueva tarjeta en nombre del cliente (el cliente proporciona detalles por teléfono)
4. Establece la nueva tarjeta como predeterminada
5. Elimina la tarjeta vencida
6. Confirma con el cliente

> ⚠ **AVISO**
>
> Nunca pida a los clientes que envíen por correo electrónico o mensaje de texto los detalles de la tarjeta. Siempre use:
>
> - Portal seguro de Autocuidado para autoservicio
> - Teléfono con el personal ingresando detalles directamente en el sistema
> - En persona en la ubicación minorista

Qué Sucede Cuando las Tarjetas Vencen

Cuando una tarjeta de pago alcanza su fecha de vencimiento y no se actualiza:

Efectos Inmediatos:

1. **Los Pagos Automáticos Fallan**
   - El proveedor de pagos rechaza transacciones con tarjetas vencidas
   - Las renovaciones de servicios mensuales no se procesan

    - Las recargas automáticas fallan
    - Los pagos automáticos de facturas fallan
2. **Notificaciones al Cliente**
    - El sistema intenta cargar la tarjeta
    - Se envía una notificación de fallo de pago
    - Se envía un correo electrónico "Actualizar Método de Pago" con un enlace al portal de Autocuidado
3. **Cambios en el Estado del Servicio**
    - **Servicios Postpago** - Pueden continuar temporalmente con saldo pendiente
    - **Servicios Prepago** - Suspensión del servicio cuando se agota el saldo
    - **Servicios de Auto-Renovación** - La renovación falla, el servicio puede expirar

Acciones Posteriores:

Día 1-3 (Período de Gracia):

- El servicio continúa normalmente
- El cliente recibe el primer aviso de fallo de pago
- El sistema intenta reintentar (dependiendo de la configuración)

Día 4-7:

- Segundo intento de pago (si está configurado)
- Se envía un correo electrónico de advertencia
- El servicio al cliente puede contactar al cliente

Día 8-14:

- El servicio puede ser suspendido por falta de pago
- El estado suspendido impide el uso pero preserva la cuenta
- El cliente puede restaurar actualizando el método de pago y pagando el saldo pendiente

Día 15+:

- El servicio puede ser terminado por falta de pago

- Inventario (tarjetas SIM, equipo) marcado para devolución
- Aviso final enviado
- Cuenta referida a cobranzas (si aplica)

Prevención de Interrupción del Servicio:

Para evitar la interrupción del servicio:

- Actualizar tarjetas **30 días antes de la expiración**
- Agregar múltiples métodos de pago para redundancia
- Habilitar alertas de fallo de pago
- Monitorear el panel de Tarjetas que Expiran semanalmente

Restaurar el Servicio Después de la Expiración:

Si el servicio fue suspendido debido a una tarjeta expirada:

1. Agregar un nuevo método de pago válido
2. Establecer como predeterminado
3. Pagar el saldo pendiente (si lo hay)
4. Contactar soporte para reactivar el servicio
5. Servicio restaurado en minutos a horas

Seguridad del Método de Pago

Tokenización

OmniCRM utiliza la tokenización del proveedor para garantizar la seguridad:

1. **El cliente ingresa la tarjeta** → Enviada directamente a los servidores del proveedor de pagos
2. **El proveedor valida y tokeniza** → Crea un token único
3. **Token almacenado en OmniCRM** → El número completo de la tarjeta nunca se almacena
4. **Procesamiento de pagos** → Token enviado al proveedor, el proveedor carga la tarjeta

Lo que OmniCRM Almacena:

```
{
  "vendor": "stripe",
  "vendor_payment_method_id": "pm_1A2B3C4D5E6F",
  "payment_type": "card",
  "brand": "visa",
  "last4": "1234",
  "exp_month": 12,
  "exp_year": 2026,
  "name": "John Smith",
  "nickname": "Personal Card",
  "is_default": true
}
```

Lo que OmniCRM NO Almacena:

- Número completo de la tarjeta
- Código CVV/CVC
- Datos de la banda magnética
- Números de PIN

Cumplimiento de PCI

Al usar formularios de pago alojados por el proveedor:

- **Alcance PCI reducido** - Los datos de la tarjeta nunca tocan los servidores de OmniCRM
- **Campos alojados por el proveedor** - La entrada de la tarjeta ocurre en el iframe seguro del proveedor
- **Sin almacenamiento de tarjetas** - Se utilizan tokens en lugar de datos de tarjeta en bruto
- **Transmisión segura** - Toda la comunicación se realiza a través de HTTPS/TLS

Consulte la Guía del Sistema de Pagos para obtener detalles sobre la seguridad del proveedor de pagos.

Flujos de Trabajo Comunes

Flujo de Trabajo 1: El Cliente Agrega el Primer Método de Pago

Escenario: Nuevo cliente registrándose para el servicio

1. El cliente crea una cuenta
2. Selecciona un plan de servicio
3. Se le solicita agregar un método de pago durante el proceso de pago
4. Ingresa los detalles de la tarjeta en el modal de Stripe
5. La tarjeta es tokenizada y guardada
6. Se establece automáticamente como predeterminada
7. Servicio provisionado
8. Primer cargo procesado

Flujo de Trabajo 2: El Cliente Actualiza la Tarjeta que Está por Expirar

Escenario: Tarjeta de crédito a punto de expirar

1. El cliente recibe una notificación por correo electrónico (60 días antes de la expiración)
2. Inicia sesión en el portal de Autoayuda
3. Navega a **Facturación → Métodos de Pago**
4. Revisa la tarjeta actual que expira en 12/2025
5. Hace clic en **"Agregar Método de Pago"**
6. Ingresa la tarjeta de reemplazo con expiración 12/2028
7. Establece la nueva tarjeta como predeterminada
8. Elimina la tarjeta antigua
9. Se envía un correo electrónico de confirmación

Flujo de Trabajo 3: El Personal Ayuda al Cliente por Teléfono

Escenario: El cliente llama: "Mi tarjeta fue rechazada"

1. El cliente llama al soporte
2. El personal verifica la identidad (preguntas de seguridad)
3. El personal verifica los métodos de pago: Tarjeta expirada 01/2025
4. El personal: "Tu tarjeta ha expirado. ¿Tienes una nueva tarjeta?"
5. El cliente proporciona los detalles de la nueva tarjeta por teléfono
6. El personal navega a **Clientes → [Cliente] → Facturación**
7. Hace clic en **"Agregar Método de Pago"**
8. Ingresa los detalles de la tarjeta mientras el cliente los lee
9. Establece la nueva tarjeta como predeterminada
10. Elimina la tarjeta expirada
11. Vuelve a intentar el pago fallido
12. Confirma con el cliente: "Pago exitoso, servicio restaurado"

Flujo de Trabajo 4: Cliente Empresarial con Múltiples Tarjetas

Escenario: La empresa quiere diferentes tarjetas para diferentes propósitos

1. El cliente empresarial agrega la tarjeta principal (Visa terminando en 1111)
2. Se establece como predeterminada para los cargos mensuales del servicio
3. Agrega tarjeta de respaldo (Mastercard terminando en 2222) para recargas
4. Agrega tarjeta de compra (Amex terminando en 3333) para compras de equipo
5. Al hacer una recarga, selecciona Mastercard manualmente al momento de pagar
6. La Visa predeterminada sigue utilizándose para la facturación automática mensual

Flujo de Trabajo 5: Gestión de Tarjetas que Expiran (Admin)

Escenario: Gestión proactiva de expiraciones

1. El administrador navega a **Facturación → Tarjetas que Expiran**
2. Filtros: "Próximos 30 días"
3. Ve 15 clientes con tarjetas que expiran
4. Selecciona todos → **"Enviar Correos de Recordatorio"**
5. Mailjet envía correos electrónicos personalizados a cada cliente
6. Los clientes actualizan tarjetas a través de Autoayuda
7. El administrador revisa la lista 1 semana después
8. Llama a los clientes restantes que no han actualizado
9. Ayuda con las actualizaciones de tarjetas por teléfono

Solución de Problemas

"Tarjeta rechazada" al agregar método de pago

- **Causa:** Stripe rechazó la tarjeta (fondos insuficientes, prevención de fraude, rechazo del emisor)
- **Solución:**
  - Intenta con una tarjeta diferente
  - Contacta al emisor de la tarjeta para autorizar la transacción
  - Asegúrate de que la tarjeta soporte compras en línea
  - Verifica que la dirección de facturación coincida con la tarjeta en el archivo

"Error al agregar método de pago" (error genérico)

- **Causa:** Error de API del proveedor de pagos o problema de red
- **Solución:**
  - Actualiza la página y vuelve a intentarlo
  - Verifica la conexión a Internet

    - Verifica que la configuración del proveedor de pagos sea correcta en la configuración del sistema
    - Revisa la consola del navegador para un mensaje de error específico
    - Intenta con un navegador diferente (desactiva bloqueadores de anuncios)

No se puede eliminar el método de pago (botón deshabilitado)

- **Causa:** Es el único método de pago activo del cliente
- **Solución:**
    - Agrega un nuevo método de pago primero, luego elimina el antiguo
    - Pasa el cursor sobre el icono de papelera deshabilitado para la explicación del tooltip

La tarjeta aparece como expirada pero no en la lista de "Tarjetas que Expiran"

- **Causa:** La tarjeta expiró recientemente, caché no actualizado
- **Solución:**
    - Actualiza la página
    - Verifica los filtros en el panel de Tarjetas que Expiran
    - Las tarjetas expiradas pueden moverse a una vista diferente

Nueva tarjeta no aparece inmediatamente

- **Causa:** La página no se ha actualizado después de agregar la tarjeta
- **Solución:**
    - La tabla de métodos de pago debería actualizarse automáticamente
    - Si no, actualiza manualmente el navegador
    - Verifica si ocurrió un error durante el proceso de adición

El modal de pago no se carga

- **Causa:** SDK del proveedor de pagos no se carga, problema de clave API o extensión del navegador bloqueando
- **Solución:**
    - Verifica la consola del navegador en busca de errores

- Desactiva bloqueadores de anuncios y protección de seguimiento
- Verifica la configuración del proveedor de pagos en la configuración del sistema
- Asegúrate de que el script del SDK del proveedor se cargue (verifica la pestaña de Red)
- Intenta en modo incógnito/navegación privada

El cliente no recibe notificaciones de expiración

- **Causa:** Mailjet no configurado o plantilla de correo electrónico faltante
- **Solución:**
  - Verifica las credenciales de Mailjet en *crm_config.yaml*
  - Verifica que exista una plantilla de correo electrónico para la expiración de la tarjeta
  - Confirma que la dirección de correo electrónico del cliente sea válida
  - Revisa los registros de Mailjet en busca de fallos de entrega

Mejores Prácticas

Para Clientes:

- Agregar método de pago antes de la activación del servicio para evitar retrasos
- Mantener al menos 2 tarjetas en archivo para redundancia
- Actualizar tarjetas que expiran 30+ días antes de la expiración
- Eliminar tarjetas antiguas/expiradas para evitar confusiones
- Usar apodos descriptivos ("Visa Personal", "Amex de Trabajo")
- Verificar que el método de pago predeterminado sea correcto para la facturación automática

Para el Personal de Soporte:

- Verificar la identidad del cliente antes de acceder a los métodos de pago
- Nunca pedir a los clientes que envíen detalles de la tarjeta por correo electrónico/SMS/chat

- Procesar la adición de tarjetas de inmediato durante las llamadas (no posponer)
- Confirmar que la nueva tarjeta esté establecida como predeterminada después de agregar
- Eliminar tarjetas antiguas solo después de confirmar que la nueva tarjeta funciona
- Probar el pago después de actualizar la tarjeta expirada (procesar autorización de £0.01)

Para Administradores:

- Monitorear el panel de Tarjetas que Expiran semanalmente
- Enviar correos de recordatorio 60/30/7 días antes de la expiración
- Mantener las claves de prueba/vivo del proveedor de pagos separadas para desarrollo y producción
- Asegurarse de que las plantillas de Mailjet estén configuradas para notificaciones de expiración
- Revisar informes de pagos fallidos para identificar tarjetas expiradas
- Capacitar al personal sobre procedimientos seguros de manejo de tarjetas

Mejores Prácticas de Seguridad:

- Usar solo claves publicables del proveedor (nunca claves secretas en el frontend)
- Asegurarse de que todas las páginas de pago se carguen a través de HTTPS
- Revisar regularmente el panel del proveedor de pagos en busca de actividad sospechosa
- Habilitar reglas de prevención de fraude del proveedor
- Requerir CVC para todas las transacciones sin tarjeta presente
- Registrar cambios en métodos de pago en el registro de actividad

Documentación Relacionada

- Guía del Sistema de Pagos - Configuración e integración del proveedor de pagos
- `payments_process` - Procesamiento de pagos con métodos de pago almacenados
- `payments_invoices` - Pago automático de facturas utilizando la tarjeta predeterminada
- `features_topup_recharge` - Sistema de recarga utilizando métodos de pago
- `basics_payment` - Conceptos generales de pago y facturación
- `customer_care` - Portal de Autoayuda para que los clientes gestionen sus propias tarjetas

Procesar Pagos

La mayoría de los pagos serán procesados automáticamente por el sistema, pero hay momentos en los que puede que necesites procesar un pago manualmente.

Para pagar una factura, selecciona la factura no pagada y haz clic en el botón "Pagar".

Esto abrirá un formulario de pago, donde puedes ingresar el método de pago, y hacer clic en "Enviar" para procesar el pago.

El cliente recibirá automáticamente un recibo por el pago, y la factura se marcará como pagada.

Para transferencias bancarias, puedes ingresar la referencia de pago y la fecha en que se realizó el pago (si es diferente de la fecha actual).

Descripción General de la Pestaña de Facturación

La pestaña de Facturación proporciona una vista unificada de toda la información financiera de un cliente, combinando métodos de pago, transacciones y facturas en una sola interfaz para una gestión de facturación eficiente.

Documentación relacionada: Métodos de Pago, Transacciones, Facturas, Procesamiento de Pagos, Libro mayor de billetera vs transacciones de cliente.

Accediendo a la Pestaña de Facturación

Vista por Cliente:

1. Navega a **Clientes → [Seleccionar Cliente]**
2. Haz clic en la pestaña **Facturación**
3. Visualiza las tres secciones: Métodos de Pago, Transacciones y Facturas

Vistas a Nivel de Sistema:

Los datos de facturación a nivel de sistema se pueden acceder por separado:

- **Facturación → Transacciones** - Todas las transacciones de todos los clientes
- **Facturación → Facturas** - Todas las facturas de todos los clientes

Portal de Autoservicio:

Los clientes que acceden al Portal de Autoservicio ven la misma estructura de pestaña de Facturación:

- Ver y gestionar sus métodos de pago
- Ver el historial de transacciones
- Ver y pagar facturas en línea

Estructura de la Pestaña de Facturación

La pestaña de Facturación está organizada en tres secciones principales, mostradas como tarjetas:

Sección 1: Métodos de Pago

Propósito: Gestionar cómo el cliente paga por los servicios

Características Clave:

- Ver todas las tarjetas de crédito guardadas
- Establecer método de pago predeterminado
- Agregar nuevos métodos de pago (a través de Stripe)
- Eliminar tarjetas caducadas o no utilizadas

Documentación: `basics_payment`

Sección 2: Transacciones

Propósito: Rastrear todos los cargos y créditos para el cliente

Características Clave:

- Ver estadísticas de transacciones (Total, No Facturado, Este Mes, Último Mes)
- Listar todas las transacciones con filtrado por estado de anulación/factura
- Agregar transacciones manuales (cargos o créditos)
- Anular transacciones incorrectas
- Ver qué transacciones están facturadas vs no facturadas

Documentación: `payments_transaction`

Sección 3: Facturas

Propósito: Agrupar transacciones en facturas para que el cliente pague

Características Clave:

- Ver estadísticas de facturas (Total, No Pagadas, Este Mes, Último Mes)
- Listar todas las facturas con filtrado por estado pagado/anulado

- Generar nuevas facturas a partir de transacciones no facturadas
- Descargar PDFs de facturas
- Enviar facturas por correo electrónico a los clientes
- Pagar facturas en línea (métodos de pago de Stripe o manuales)

- Anular o reembolsar facturas

Documentación: `payments_invoices`

Flujo de Datos Entre Secciones

Entender cómo fluye la información entre las tres secciones es crucial para una gestión de facturación efectiva.

Diagrama de Flujo

Relación Transacción → Factura

1. Creación de Transacción:

Cuando se proporciona un servicio o se agrega un cargo manual:

- Transacción creada en la sección **Transacciones**
- Estado de la transacción: **No Facturada**
- Campo `invoice_id` de la transacción es `null`

Ejemplo:

2. Generación de Factura:

Cuando el personal genera una factura:

- Factura creada en la sección **Facturas**
- Todas las transacciones no facturadas dentro del rango de fechas agrupadas en la factura
- Campo `invoice_id` de la transacción poblado
- Estado de la transacción cambia a: **Facturada**

Ejemplo:

3. Actualización de Estadísticas de Transacciones:

- Total de **Transacciones No Facturadas** disminuye
- Estadística de **Total de Facturas** aumenta
- Total de **Facturas No Pagadas** aumenta

Relación Factura → Pago

1. Pago de Factura:

Cuando el cliente paga la factura:

- Pago procesado utilizando el **Método de Pago** guardado (tarjeta de Stripe)
- O se selecciona un método de pago manual (efectivo, POS, transferencia bancaria)
- Estado de la factura cambia a: **Pagada**

2. Transacción de Pago Creada:

Para pagos manuales (no Stripe):

- Transacción negativa creada automáticamente
- Título de la transacción: "Pago por Factura #1234"
- Monto de la transacción: -$45.00 (negativo, acredita al cliente)
- Campo `invoice_id` de la transacción: Enlaza a la factura pagada

Ejemplo:

3. Actualización de Estadísticas:

- Total de **Facturas No Pagadas** disminuye
- **Total de Facturas Este Mes** sin cambios (la factura ya existía)

Relación Método de Pago → Factura

Flujo de Pago de Stripe:

1. El cliente agrega una tarjeta de crédito en **Métodos de Pago**
2. Tarjeta tokenizada a través de Stripe, almacenada de forma segura
3. Al pagar la factura, el cliente selecciona la tarjeta guardada
4. Stripe cobra la tarjeta
5. Factura marcada como pagada
6. Campo `payment_reference` poblado con el ID de intención de pago de Stripe

Flujo de Pago Manual:

1. El cliente paga en efectivo/POS/transferencia bancaria (no se necesita método de pago)
2. El personal selecciona el método de pago en el modal Pagar Factura
3. El personal ingresa el número de referencia (opcional)
4. Transacción negativa creada por el monto del pago
5. Factura marcada como pagada

Flujos de Trabajo Completos de Facturación

Estos flujos de trabajo demuestran cómo las tres secciones trabajan juntas para llevar a cabo tareas comunes.

Flujo de Trabajo 1: Configuración de Nuevo Cliente y Primera Factura

Objetivo: Configurar la facturación para un nuevo cliente y recaudar el primer pago

1. **Agregar Método de Pago:**
   - Navegar al cliente → pestaña **Facturación**
   - Sección **Métodos de Pago** → Hacer clic en **"Agregar Método de Pago"**
   - El cliente agrega una tarjeta de crédito a través de Stripe
   - Tarjeta guardada como método de pago predeterminado
2. **Verificar Transacciones:**
   - La sección **Transacciones** muestra transacciones no facturadas:
     - Cargo por configuración de servicio: $50.00
     - Servicio del primer mes: $45.00
     - Total No Facturado: $95.00
3. **Generar Factura:**
   - Sección **Facturas** → Hacer clic en **"Generar Factura Proforma"**
   - Establecer rango de fechas para incluir la configuración y el primer mes
   - Hacer clic en **"Generar Factura"**
   - Factura #INV-2025-001234 creada por $95.00
4. **Actualización de Transacciones:**
   - Ambas transacciones ahora muestran: Factura #INV-2025-001234
   - Total de **Transacciones No Facturadas** ahora $0.00
5. **Enviar Factura por Correo:**
   - Hacer clic en el ícono de correo junto a la factura

    - El cliente recibe un correo electrónico de la factura con PDF y enlace de pago
6. **El Cliente Paga en Línea:**
    - El cliente hace clic en el enlace de pago en el correo
    - Redirigido al portal de Autoservicio
    - Hacer clic en el botón **"Pagar Factura"**
    - Seleccionar el método de pago predeterminado
    - Hacer clic en **"Pagar Factura"**
    - Stripe cobra la tarjeta
7. **Actualización de la Factura:**
    - Estado de la factura cambia a **"Pagada"**
    - Total de **Facturas No Pagadas** disminuye en $95.00

Resultado: Cliente completamente configurado con método de pago, primera factura pagada.

Flujo de Trabajo 2: Facturación Mensual Recurrente

Objetivo: Facturar a todos los clientes por el servicio mensual al final del mes

1. **Servicios de Carga Automática:**
    - Llega el final del mes (31 de enero)
    - El sistema de facturación crea automáticamente transacciones para todos los servicios recurrentes
    - La sección **Transacciones** muestra nuevas transacciones no facturadas
2. **Revisar Transacciones No Facturadas:**
    - Navegar a la sección **Transacciones**
    - Filtrar: **Estado de Factura: No Facturada**
    - Revisar la lista de todas las transacciones listas para facturación
    - Verificar que los montos y descripciones sean correctos
3. **Generar Facturas:**
    - Navegar a **Facturación → Facturas** (a nivel de sistema)
    - Para cada cliente (o usar proceso por lotes):

        - Hacer clic en **"Generar Factura Proforma"**
        - Seleccionar cliente
        - Fecha de Inicio: 2025-01-01
        - Fecha de Fin: 2025-01-31
        - Fecha de Vencimiento: 2025-02-15
        - Hacer clic en **"Generar Factura"**
4. **Actualización de Transacciones:**
    - Todas las transacciones ahora vinculadas a facturas
    - Totales de **Transacciones No Facturadas** restablecidos a $0.00
5. **Enviar Todas las Facturas por Correo:**
    - Para cada factura, hacer clic en el ícono de correo
    - Todos los clientes reciben facturas mensuales
6. **Los Clientes Pagan:**
    - Los clientes con métodos de pago guardados pagan en línea a través de Autoservicio
    - El personal procesa pagos en efectivo/POS para clientes que pagan en persona
    - Total de **Facturas No Pagadas** disminuye a medida que se reciben los pagos

Resultado: Todos los clientes facturados por enero, facturas enviadas, pagos procesados.

Flujo de Trabajo 3: Manejo de Crédito por Problema de Servicio

Objetivo: Acreditar al cliente por una interrupción del servicio, aplicar a factura no pagada

1. **Cliente Informa Problema:**
    - El servicio estuvo fuera de servicio durante 2 días
    - El cliente merece un crédito de $10
2. **Agregar Transacción de Crédito:**
    - Navegar al cliente → pestaña **Facturación** → sección **Transacciones**
    - Hacer clic en **"+ Agregar Transacción"**

- Tipo de Transacción: **Crédito**
- Tipo de Crédito: **Pago en Efectivo** (o tipo apropiado)
- Título: "Crédito por Interrupción de Servicio"
- Descripción: "Compensación por interrupción de 2 días 8-9 de enero"
- Costo Minorista: 10.00
- Hacer clic en **"Agregar Transacción"**

3. **Transacción Creada:**
   - La transacción aparece en la lista con monto: **-$10.00**
   - Estado de la transacción: **No Facturada**
   - Total de **Transacciones No Facturadas** ahora incluye -$10.00
4. **Aplicar a Factura:**
   - Si el cliente ya tiene una factura no pagada:
     - La factura permanece no pagada con el monto original
     - El crédito se aplicará a la próxima generación de factura
   - Si se genera una nueva factura:
     - Sección **Facturas** → Hacer clic en **"Generar Factura Proforma"**
     - Incluir rango de fechas con la transacción de crédito
     - Factura generada con crédito aplicado:
5. **El Cliente Paga:**
   - El cliente paga el monto reducido: $35.00
   - Factura marcada como pagada

Resultado: Cliente acreditado por la interrupción, crédito aplicado a la próxima factura, pago menor recaudado.

Flujo de Trabajo 4: Método de Pago Caducado - Actualizar y Reintentar

Objetivo: La tarjeta del cliente caducó, causando un fallo en el pago - actualizar tarjeta y reintentar el pago

1. **Notificación de Fallo en el Pago:**
   - El cliente intenta pagar la factura
   - Stripe devuelve el error: "Tarjeta caducada"
   - El pago falla, la factura permanece no pagada

2. **Actualizar Método de Pago:**
    - El cliente navega a la pestaña **Facturación**
    - Sección **Métodos de Pago** → Hacer clic en **"Agregar Método de Pago"**
    - Ingresar los nuevos datos de la tarjeta (fecha de caducidad actualizada)
    - Nueva tarjeta guardada
3. **Establecer como Predeterminado:**
    - El cliente hace clic en **"Establecer como Predeterminado"** en la nueva tarjeta
    - La tarjeta antigua se elimina automáticamente (si se desea)
4. **Reintentar Pago:**
    - Navegar a la sección **Facturas**
    - Localizar la factura no pagada
    - Hacer clic en el ícono **"Pagar"**
    - Se abre el modal de pago con la nueva tarjeta predeterminada preseleccionada
    - Hacer clic en **"Pagar Factura"**
    - Stripe cobra la nueva tarjeta con éxito
5. **Actualización de la Factura:**
    - Estado de la factura cambia a **"Pagada"**
    - Campo `payment_reference` poblado con la nueva intención de pago de Stripe

Resultado: Cliente actualizó el método de pago, factura pagada con éxito con la nueva tarjeta.

Flujo de Trabajo 5: Anulación de Factura Incorrecta y Re-Facturación

Objetivo: Factura generada por el personal con transacciones incorrectas - anular y regenerar correctamente

1. **Error Descubierto:**
    - Factura #INV-2025-001234 generada con rango de fechas incorrecto
    - Incluyó transacciones del mes equivocado

- La factura está no pagada

2. **Anular Factura:**
   - Navegar a la pestaña **Facturación** → sección **Facturas**
   - Localizar la factura incorrecta
   - Hacer clic en el ícono de eliminar (🗑)
   - Confirmar anulación
   - Factura anulada
3. **Transacciones Liberadas:**
   - Navegar a la sección **Transacciones**
   - Todas las transacciones de la factura anulada ahora muestran: **No Facturadas**
   - Total de **Transacciones No Facturadas** aumenta
   - Transacciones disponibles para nueva factura
4. **Generar Factura Correcta:**
   - Sección **Facturas** → Hacer clic en **"Generar Factura Proforma"**
   - Establecer el rango de fechas correcto
   - Aplicar filtro si es necesario (por ejemplo, "Móvil" para factura solo de móvil)
   - Hacer clic en **"Generar Factura"**
   - Nueva factura creada con las transacciones correctas
5. **Verificar y Enviar por Correo:**
   - Revisar los detalles de la nueva factura
   - Verificar que se incluyan las transacciones correctas
   - Hacer clic en el ícono de correo para enviar al cliente

Resultado: Factura incorrecta anulada, transacciones re-facturadas correctamente, el cliente recibe la factura corregida.

Flujo de Trabajo 6: Procesamiento de Pago en Efectivo para Múltiples Facturas

Objetivo: El cliente paga múltiples facturas no pagadas con un solo pago en efectivo

1. **Cliente Llega con Efectivo:**

- El cliente trae $300 en efectivo para pagar facturas pendientes
- Navegar al cliente → pestaña **Facturación**

2. **Revisar Facturas No Pagadas:**
   - Sección **Facturas** → Filtrar: **Pagadas: No Pagadas**
   - Ver facturas no pagadas:
3. **Pagar Primera Factura:**
   - Hacer clic en el ícono de pago en la factura #1234
   - Se abre el modal de pago
   - Seleccionar método de pago **"Efectivo"**
   - Ingresar referencia: "Efectivo pagado 2025-02-10 - Recibo #001"
   - Hacer clic en **"Pagar Factura"**
   - Factura #1234 marcada como **"Pagada"**
4. **Pagar Facturas Restantes:**
   - Repetir proceso para la factura #1235:
     - Referencia: "Efectivo pagado 2025-02-10 - Recibo #001"
   - Repetir para la factura #1236:
     - Referencia: "Efectivo pagado 2025-02-10 - Recibo #001"
5. **Verificar Transacciones:**
   - Navegar a la sección **Transacciones**
   - Tres nuevas transacciones de pago creadas:
   - Todas vinculadas a las respectivas facturas
6. **Actualizar Estadísticas:**
   - Sección **Facturas** → Total de **Facturas No Pagadas** disminuido en $300.00
   - Todas las facturas ahora muestran estado **"Pagadas"**

Resultado: Cliente pagó todas las facturas pendientes en efectivo, transacciones de pago registradas con referencia de recibo.

Mejores Prácticas

Para Usuarios del Personal

Gestión de Transacciones:

- Agregar transacciones manuales de inmediato (no retrasar)
- Usar títulos y descripciones descriptivas para mayor claridad
- Vincular transacciones a servicios y sitios cuando sea aplicable
- Anular transacciones incorrectas antes de que sean facturadas

Generación de Facturas:

- Generar facturas a intervalos consistentes (por ejemplo, mensualmente el 1 del mes)
- Usar rangos de fechas cuidadosamente para evitar superposiciones o vacíos
- Usar filtros para crear facturas específicas de servicios cuando sea necesario
- Enviar facturas por correo electrónico inmediatamente después de la generación
- Revisar PDFs de facturas antes de enviarlas a los clientes

Procesamiento de Pagos:

- Verificar que el método de pago sea válido antes de intentar el cargo
- Siempre ingresar números de referencia para pagos manuales (efectivo, POS, transferencia bancaria)
- Marcar las facturas como pagadas inmediatamente después de recibir el pago
- Reembolsar solo a través de Stripe (crear crédito manual para otros métodos de pago)

Higiene de Datos:

- Revisar regularmente las transacciones no facturadas
- Investigar transacciones anuladas para entender errores de facturación
- Monitorear facturas no pagadas y hacer seguimiento con los clientes
- Mantener los métodos de pago actualizados (eliminar tarjetas caducadas)

Para Clientes (Portal de Autoservicio)

Métodos de Pago:

- Mantener al menos un método de pago válido en el archivo
- Actualizar métodos de pago antes de que las tarjetas caduquen
- Establecer su tarjeta preferida como predeterminada

Pagos de Facturas:

- Pagar facturas antes de la fecha de vencimiento para evitar cargos por demora
- Revisar los detalles de la factura y las transacciones antes de pagar
- Descargar PDFs de facturas para sus registros
- Contactar al soporte de inmediato si la factura parece incorrecta

Revisión de Transacciones:

- Revisar regularmente el historial de transacciones
- Informar cualquier cargo inesperado de inmediato
- Entender qué transacciones están facturadas vs no facturadas

Para Administradores

Configuración del Sistema:

- Configurar plantillas de correo electrónico de Mailjet para la entrega profesional de facturas
- Personalizar plantillas de PDF de facturas para que coincidan con la marca
- Configurar integración de proveedores de pago (Stripe, PayPal) para un procesamiento de pagos seguro
- Configurar términos de pago y fechas de vencimiento

Monitoreo e Informes:

- Usar widgets de estadísticas para monitorear la salud de la facturación
- Rastrear el total de **Transacciones No Facturadas** - debería disminuir después del ciclo de facturación
- Monitorear el total de **Facturas No Pagadas** - hacer seguimiento de los pagos atrasados
- Revisar estadísticas de **Este Mes** vs **Último Mes** para tendencias

Automatización:

- Automatizar cargos de servicios recurrentes a través de la configuración del producto
- Configurar generación automática de facturas para facturación recurrente (si está disponible)
- Configurar recordatorios por correo electrónico para facturas vencidas

Problemas Comunes y Soluciones

Problema: El Cliente No Puede Pagar la Factura

Síntomas:

- El cliente hace clic en el botón de pago pero no sucede nada
- Mensaje de error: "No se encontraron métodos de pago"

Diagnóstico:

1. Navegar al cliente → pestaña **Facturación** → sección **Métodos de Pago**
2. Verificar si el cliente tiene métodos de pago guardados
3. Verificar si las tarjetas guardadas están caducadas

Solución:

- El cliente debe agregar un método de pago válido antes de pagar las facturas
- Guiar al cliente a la página de **Métodos de Pago** para agregar una tarjeta de crédito
- Verificar que la tarjeta sea aceptada (Visa, Mastercard, Amex, etc.)
- Reintentar el pago después de agregar la tarjeta

Problema: Factura Generada con Transacciones Incorrectas

Síntomas:

- La factura incluye transacciones del período incorrecto
- La factura falta transacciones esperadas
- Total de la factura incorrecto

Diagnóstico:

1. Abrir la factura en la sección **Facturas**
2. Revisar las transacciones incluidas en la factura
3. Verificar las fechas de las transacciones vs el rango de fechas de la factura
4. Verificar si se aplicó un filtro durante la generación

Solución:

- **Si la factura no está pagada:** Anular la factura, verificar que las transacciones estén no facturadas, regenerar con el rango de fechas correcto
- **Si la factura está pagada:** No se puede anular - crear una transacción de crédito por el monto incorrecto, generar una factura corregida
- **Prevención:** Siempre revisar la sección **Transacciones** antes de generar la factura para verificar que se incluirán las transacciones correctas

Problema: Total de Transacciones No Facturadas No Disminuye

Síntomas:

- El widget de **Transacciones No Facturadas** muestra un monto alto
- La lista de transacciones muestra muchas transacciones no facturadas
- Facturas mensuales ya generadas

Diagnóstico:

1. Filtrar transacciones por **Estado de Factura: No Facturada**
2. Revisar la lista de transacciones no facturadas
3. Verificar las fechas de las transacciones - pueden ser cargos recientes después de la última generación de facturas

4. Verificar si algunas transacciones están anuladas (no deberían contar para el total no facturado)

Solución:

- **Comportamiento esperado:** Las transacciones creadas después de la última generación de facturas permanecen no facturadas hasta el próximo ciclo de facturación
- **Si las transacciones antiguas están no facturadas:** Generar factura para esas transacciones con el rango de fechas apropiado
- **Si las transacciones anuladas se contaron:** El sistema debería excluir automáticamente las transacciones anuladas - informar un error si no es así

Problema: Método de Pago Sigue Siendo Rechazado

Síntomas:

- El pago de Stripe falla con "Tarjeta rechazada"
- El cliente informa que la tarjeta debería funcionar
- Múltiples intentos de reintento fallan

Diagnóstico:

1. Verificar el panel de control de Stripe para la razón del rechazo
2. Razones comunes de rechazo:
    - Fondos insuficientes
    - Tarjeta caducada
    - Tarjeta reportada como perdida/robada
    - Prevención de fraude bancario
    - Tarjeta internacional bloqueada

Solución:

- Pedir al cliente que se comunique con su banco para autorizar el pago
- Intentar con un método de pago diferente (tarjeta diferente)

- Para el personal: Procesar como pago manual (efectivo, POS) y marcar la factura como pagada
- Verificar que la dirección de facturación coincida con la dirección de facturación de la tarjeta

Atajos de Teclado y Consejos

Navegación:

- Acceder rápidamente a la pestaña de facturación: Página del cliente → tecla **B** (si los atajos de teclado están habilitados)

Filtrado:

- **Transacciones:** Hacer clic en el filtro **"No Facturado"** antes de generar facturas para ver exactamente qué se incluirá
- **Facturas:** Hacer clic en el filtro **"No Pagadas"** para ver todas las facturas pendientes que requieren seguimiento

Operaciones Masivas:

- Seleccionar múltiples facturas (casilla de verificación) para eliminar/anular múltiples a la vez (solo personal)
- Usar la barra de búsqueda para encontrar rápidamente una transacción o factura específica por ID

Acciones Rápidas:

- Hacer clic en el título de la transacción/factura para ver todos los detalles en un modal
- Menú de acciones de clic derecho (⋮) para acceso rápido a opciones de descarga/envío/pago

Documentación Relacionada

- `basics_payment` - Métodos de pago e integración de pago multi-proveedor

- `payment_system_guide` - Referencia de API de pago y configuración de proveedores
- `payments_transaction` - Transacciones en detalle
- `payments_invoices` - Facturas en detalle
- `integrations_mailjet` - Entrega de facturas por correo electrónico
- `csa_activity_log` - Visualización del historial de actividad de facturación
- Monitoreo y Métricas - Métricas y monitoreo del sistema de facturación OCS/CGRateS

Movimientos de billetera y transacciones de clientes (alineación del libro mayor)

Descripción general

Las filas `customer_transaction` son el libro mayor estilo facturación / GL para el CRM. `wallet_ledger` sigue siendo el historial operativo de la billetera. Para **recarga de billetera con tarjeta** y **crédito manual en efectivo / billetera administrativa**, la API ahora crea una `customer_transaction` correspondiente con `retail_cost` **negativo** (fondos recibidos), `payment_type` establecido en:

- `wallet_topup_card`: recarga a través de método de pago cargado (`POST /crm/wallet/topup`)
- `wallet_topup_cash`: crédito manual (`POST /crm/wallet/credit`, por ejemplo, efectivo / banco)
- `wallet_adjustment_debit`: débito administrativo de billetera (`POST /crm/wallet/debit`) publica una línea **positiva** (cargo de ajuste)

Cada línea almacena `wallet_ledger_id` que apunta a la fila correspondiente de `wallet_ledger` para la conciliación.

Convención de signos vs GAAP

En este sublibro, **`retail_cost`** **negativo** significa una línea que **reduce** los cargos netos desde la perspectiva del cliente (por ejemplo, recargas de billetera / créditos estilo pago); **positivo** significa una línea tipo **cargo** (por

ejemplo, factura, débito de ajuste de billetera). Esa es una **convención de facturación del CRM**, no un sustituto para la presentación formal de GAAP.

US GAAP no exige este signo particular para tablas internas. El cumplimiento depende de **doble entrada en el libro mayor**, **mapeo de cuentas** (efectivo, AR, ingresos no devengados, etc.), y preparación de **estados financieros**, no de si `customer_transaction.retail_cost` utiliza más/menos. Para los operadores, se prefiere una interfaz de usuario que muestre **monto absoluto + etiqueta** y opcionalmente un **saldo acumulado**, en lugar de depender solo del signo bruto. La tabla de facturación **Todas las Transacciones** formatea `retail_cost` con un **monto en negrita** y un **superíndice Cr** / **Dr** (cargo de crédito / débito; las descripciones emergentes lo explican) más verde/rojo bajo la columna **Adeudado / crédito**, por lo que el tipo no se transmite solo por color. **Dr** aquí significa un **cargo al cliente** (monto adeudado), no “su tarjeta fue debitada.”

Pagar factura desde la billetera

Pagar Factura incluye **Solo billetera** cuando el cliente tiene un saldo positivo en la billetera; llama a `POST /crm/payments/invoice` con `routing_mode: wallet_only` (el monto total debe estar cubierto). **Tarjeta guardada** aún utiliza enrutamiento **híbrido**: primero se aplica la billetera, luego la tarjeta para cualquier resto.

¿Deducción automática de la billetera cuando se crea una factura?

No implementado por defecto. **Prepagado vs postpago** es una regla comercial: anular silenciosamente la billetera cuando se emite una factura puede sorprender a los clientes y complica los pagos parciales, créditos y disputas. Patrón más seguro: **emitir la factura**, luego **pagar explícitamente** (solo billetera, tarjeta, híbrido o tipos manuales de personal). Si el producto necesita aplicar automáticamente más tarde, agregue una configuración clara (por ejemplo, por cliente o por factura) y documente el impacto en el GL.

Postpago

Los créditos de billetera reducen `customer.outstanding_balance` hasta el monto acreditado (para `customer_payment_type = postpaid`).

Facturación

Las líneas de sincronización de billetera no facturadas son **prepagos de clientes**. La vista previa de **Generar Factura** en la lista de facturas **excluye** líneas cuyo `payment_type` comienza con `wallet_` para que las facturas rutinarias no se reduzcan accidentalmente por créditos de prepago. Para aplicar un prepago a una factura específica, use un flujo de trabajo que seleccione `transaction_ids` explícitos (si su herramienta lo admite), o maneje ajustes fuera de esta vista previa predeterminada.

Impuesto: Las nuevas facturas de transacciones pendientes almacenan **`customer_invoice.retail_cost` = suma(línea `retail_cost`) + suma(línea `tax_amount`)** (monto adeudado). Las respuestas de API de lista/detalle también **recalculan** ese total a partir de líneas vinculadas, por lo que las facturas más antiguas sin impuestos incluidos en `retail_cost` aún muestran y cobran el monto correcto. El subtotal/impuesto/pie de página en PDF utiliza **líneas de artículos**, no solo el encabezado de la factura en bruto.

Permisos

- `POST /crm/wallet/topup`: usuarios autenticados que pueden usar la aplicación (los clientes usan recarga con tarjeta).
- `POST /crm/wallet/credit` y `POST /crm/wallet/debit`: requieren **`manage_customer_wallet`** (otorgado a `admin` y `customer_service_agent_2` en las semillas de rol predeterminadas). Vuelva a ejecutar `populate_roles_permissions` (o equivalente) después de la implementación para que el permiso exista y esté asignado.

“Crédito” manual en Agregar Transacción

La interfaz de usuario ya no ofrece entrada de transacción de **crédito** manual, porque no acreditaba la billetera. Use **Facturación → Billetera → Efectivo / manual** (personal) o pagos de facturas en su lugar.

Tableros (Gestión del Trabajo)

OmniCRM **Tableros** proporciona un sistema de gestión del trabajo flexible, estilo kanban, para rastrear tareas, instalaciones, solicitudes de características y cualquier otro trabajo que su equipo necesite coordinar. Los tableros se sitúan junto al Sistema de Tickets de Soporte. Los tickets son impulsados por problemas del cliente. Los tableros son impulsados por trabajo y proyectos. Los tableros ofrecen a los equipos una forma visual de organizar elementos en grupos, moverlos a través de estados, asignar propietarios y colaborar.

Véase también: Tickets de Soporte para el seguimiento de problemas de clientes, Usuarios y Roles para permisos, Clientes, Contactos y Sitios para los registros a los que puede vincular elementos.

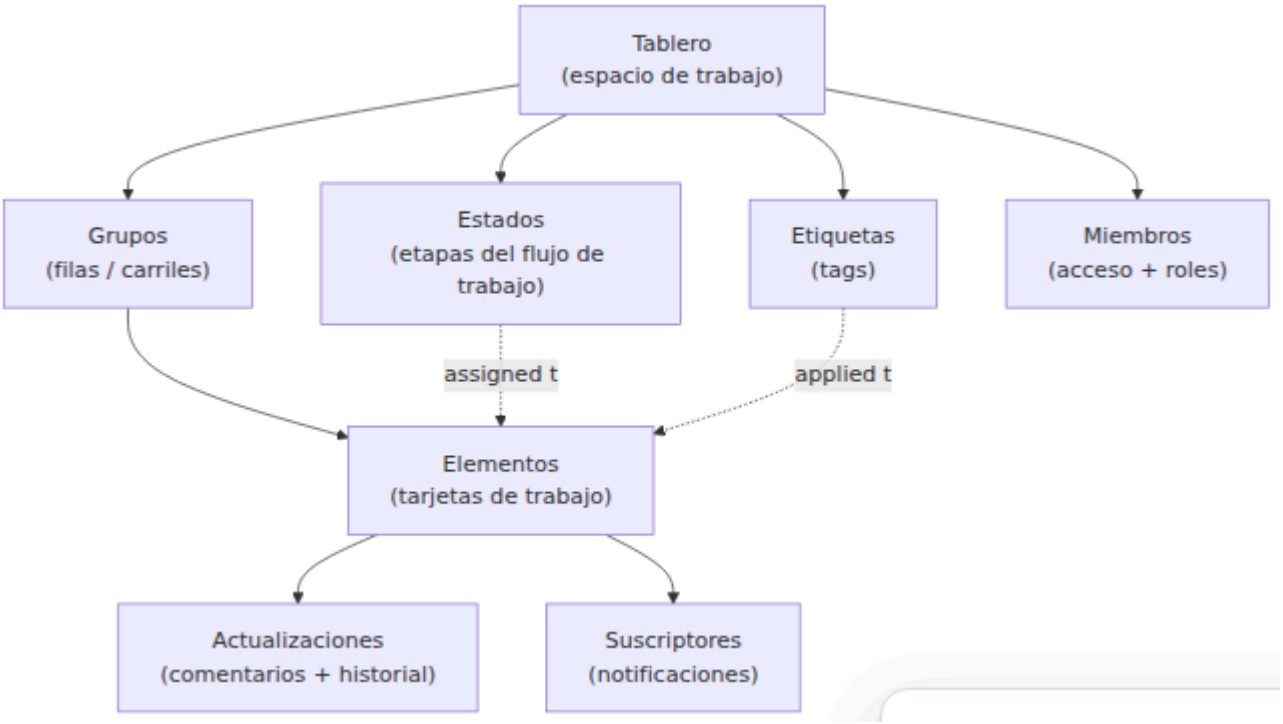
Descripción General

Los tableros permiten:

1. **Seguimiento Visual del Trabajo**: Organizar el trabajo en grupos (filas) y estados (etapas del flujo de trabajo)
2. **Plantillas**: Comenzar rápidamente con diseños preconstruidos para listas de tareas, instalaciones y solicitudes de características
3. **Propiedad**: Asignar elementos al personal, registrar quién los solicitó y rastrear fechas de vencimiento
4. **Vinculación**: Conectar elementos a clientes, servicios y tickets de soporte
5. **Colaboración**: Comentarios, historial de cambios, @menciones y suscripciones
6. **Control de Acceso**: Membresía por tablero con permisos basados en roles
7. **Compartición Pública**: Opcionalmente exponer un tablero de solo lectura a través de un enlace seguro para incrustar

Conceptos Clave

Un tablero se construye a partir de un pequeño conjunto de bloques de construcción. Comprender cómo encajan hace que el resto de esta guía sea sencillo.

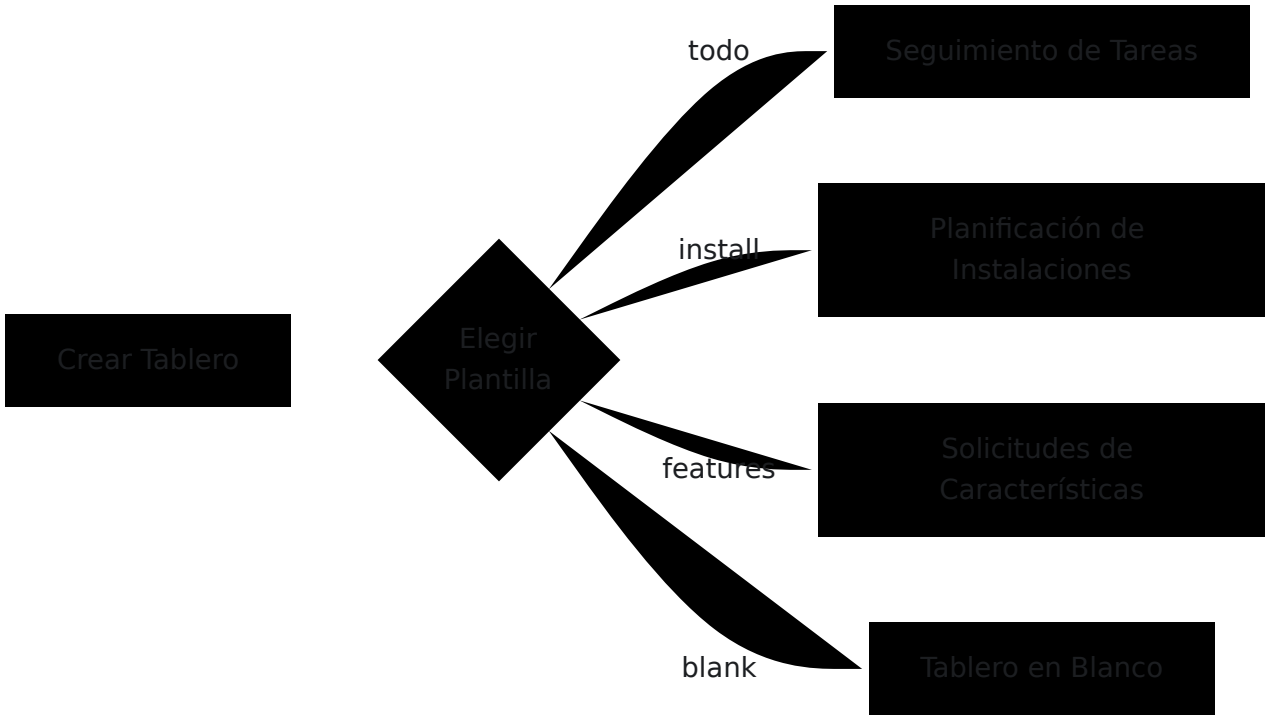


Concepto	Qué es
Tablero	El espacio de trabajo de nivel superior. Contiene sus propios grupos, estados, etiquetas, miembros y elementos.
Grupo	Una fila horizontal (carril) que organiza elementos en secciones, por ejemplo *Esta Semana*, *Próxima Semana*, *Más Tarde*. Los grupos pueden ser colapsados.
Estado	La etapa del flujo de trabajo de un elemento, por ejemplo *Por Hacer*, *En Progreso*, *Hecho*. Un estado es el predeterminado para nuevos elementos. Cualquier estado puede ser marcado como un estado "hecho".
Elemento	Una unidad única de trabajo (una tarjeta). Tiene un título, descripción, prioridad, asignado, fecha de vencimiento, etiquetas y un número de elemento único.
Etiqueta	Una etiqueta codificada por colores aplicada a los elementos para categorización (por ejemplo *Error*, *Característica*). Un elemento puede llevar múltiples etiquetas.
Miembro	Un usuario que ha sido otorgado acceso al tablero, con un rol que controla lo que puede hacer.

Cómo funciona el diseño: Los grupos son las filas del tablero. Dentro de cada grupo, los elementos se enumeran y muestran su estado actual, asignado, fecha de vencimiento y prioridad. Los elementos se mueven *entre grupos* para reorganizarlos, y *entre estados* para progresarlos a través del flujo de trabajo.

Plantillas de Tablero

Al crear un tablero, eliges una plantilla. La plantilla pre-puebla los estados y grupos del tablero para que puedas comenzar a trabajar de inmediato. Puedes agregar, renombrar, recolorear o eliminar estados y grupos después.



Plantilla	Propósito	Estados (predeterminado primero)	Grupo
Seguimiento de Tareas (`todo`)	Listas de tareas generales	Por Hacer · En Progreso · **Hecho**	Principal
Planificación de Instalaciones (`install`)	Programación de instalaciones de campo/red	Programado · En Progreso · **Completado** · En Espera	Esta Sema Próxima Semana · Tarde
Solicitudes de Características (`features`)	Pipeline de producto/característica	Nuevo · En Revisión · Planificado · En Desarrollo · **Liberado**	Solicitudes Caracterís · En Progre Completac
Tablero en Blanco (`blank`)	Punto de partida mínimo	Por Hacer · **Hecho**	Elementos

Los estados mostrados en **negrita** están marcados como estados "hechos". Los elementos en esos estados cuentan como completados. Si no se selecciona ninguna plantilla, se utiliza el diseño de **Tablero en Blanco**.

Numeración de Elementos

Cada elemento se asigna un número único y legible por humanos cuando se crea, para que pueda ser referenciado en conversaciones, tickets y correos electrónicos.

El formato es:

```
{PREFIX}-{BOARD}-{SEQUENCE}
```

Por ejemplo: `BRD-001-00001`

Segmento	Significado
`PREFIX`	El prefijo del elemento del tablero. Por defecto es `BRD` y es configurable por tablero (hasta 10 caracteres, por ejemplo `INST` para un tablero de instalaciones).
`BOARD`	El ID del tablero, rellenado con ceros hasta tres dígitos.
`SEQUENCE`	La secuencia del elemento dentro del tablero, rellenada con ceros hasta cinco dígitos.

Puedes abrir cualquier elemento directamente por su número de elemento desde la vista del tablero o a través de la API.

Creación y Gestión de Tableros

Creación de un Tablero

1. Navega a **Gestión del Trabajo → Tableros**

2. Haz clic en **Crear Tablero**
3. Completa los detalles del tablero:
    - **Nombre del Tablero** (requerido)
    - **Descripción** (opcional)
    - **Plantilla**: *Seguimiento de Tareas*, *Planificación de Instalaciones*, *Solicitudes de Características*, o *Tablero en Blanco*
    - **Prefijo de Elemento**: utilizado en los números de elemento (predeterminado `BRD`, hasta 10 caracteres)
    - **Visibilidad**: *Solo Personal* o *Público (Incrustación API)*
4. Haz clic en **Crear**

El usuario que crea un tablero se convierte en su **propietario**.

Configuraciones del Tablero

Los propietarios y administradores del tablero pueden editar un tablero para cambiar su nombre, descripción, ícono, color, cliente vinculado, prefijo de elemento y visibilidad. Los tableros también pueden ser **archivados** para ocultarlos de la lista activa sin eliminar su contenido, y **eliminados** (una eliminación suave que quita el tablero de la vista mientras retiene los registros subyacentes).

Personalización de Estados, Grupos y Etiquetas

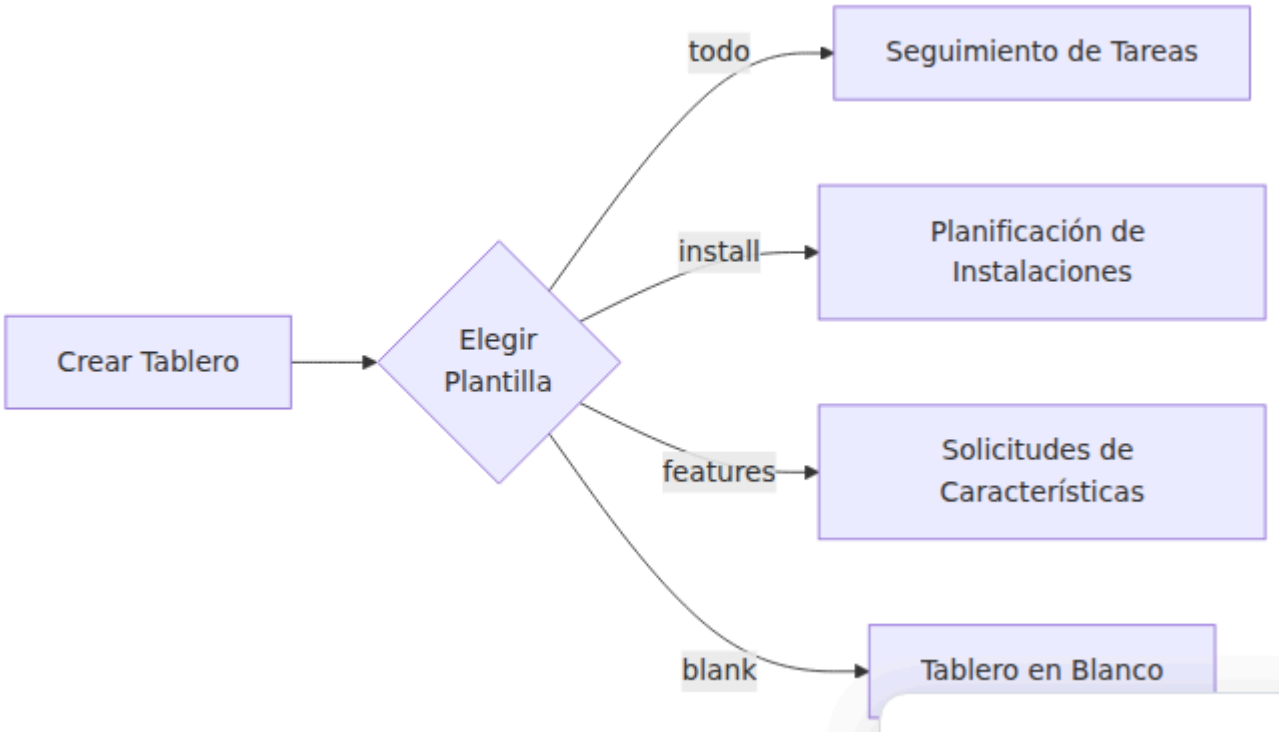
Desde dentro de un tablero puedes:

- **Estados**: agregar, renombrar, recolorear, reordenar, establecer el estado predeterminado para nuevos elementos y marcar estados como "hechos". Si eliminas un estado, sus elementos quedan sin un estado. No se eliminan.
- **Grupos**: agregar, renombrar, recolorear, reordenar y colapsar. Si eliminas un grupo, sus elementos quedan sin agrupar. No se eliminan.
- **Etiquetas**: agregar, renombrar y recolorear. Si eliminas una etiqueta, el sistema la quita de todos los elementos.

Trabajando con Elementos

La Vista del Tablero

La vista del tablero muestra cada grupo como una sección con un encabezado coloreado que muestra el nombre del grupo, el número de elementos y el progreso de finalización (cuántos elementos están en un estado "hecho"). Dentro de cada grupo, los elementos se enumeran con su título y número, asignado, estado, fecha de vencimiento y prioridad. Una fila de adición rápida te permite crear elementos en línea.



Los nombres de estado anteriores son ilustrativos. Cada tablero define sus propios estados.

Creación de un Elemento

Los elementos pueden ser creados desde la fila de adición rápida en cualquier grupo, o a través del botón **Crear Elemento**. Un elemento tiene:

Campo	Descripción
Título	Requerido. Resumen corto del trabajo.
Descripción	Detalle opcional.
Grupo	La fila a la que pertenece el elemento. Por defecto es el primer grupo si no se elige.
Estado	La etapa del flujo de trabajo. Por defecto es el estado predeterminado del tablero.
Prioridad	*Baja*, *Media*, *Alta* o *Urgente* (predeterminado es *Media*).
Fecha de Vencimiento	Fecha límite opcional.
Asignado	El miembro del personal responsable.
Solicitante	Quién solicitó el trabajo: un usuario del personal o un contacto del cliente.
Cliente / Servicio / Ticket	Enlaces opcionales a registros relacionados.
Etiquetas	Una o más etiquetas de categoría.

Moviendo Elementos

Los elementos se mueven para reorganizarlos o progresarlos:

- **Cambiar grupo**: mover el elemento a una fila/carril diferente
- **Cambiar estado**: progresar el elemento a través del flujo de trabajo
- **Reordenar**: cambiar la posición del elemento dentro de su grupo

Cada movimiento se registra en el historial del elemento.

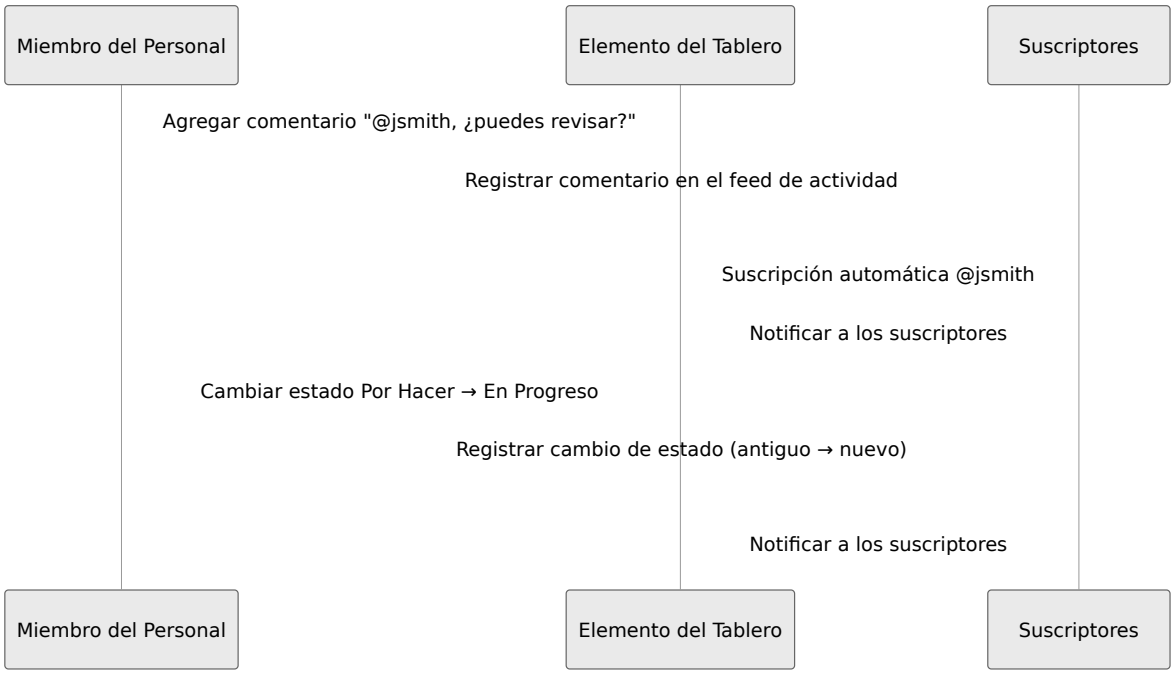
Niveles de Prioridad

Prioridad	Uso Típico
Urgente	Trabajo crítico que requiere atención inmediata
Alta	Trabajo importante, atender pronto
Media	Trabajo estándar (predeterminado)
Baja	Trabajo menor o que sería un buen complemento

Colaboración

Comentarios e Historial de Actividad

Cada elemento tiene un feed de actividad que combina **comentarios** manuales con un **historial de cambios** automático. El sistema registra cambios en el estado, asignado, título, prioridad, fecha de vencimiento y grupo. Captura los valores anteriores y nuevos, por lo que la historia completa de un elemento siempre es visible.



Suscriptores y Notificaciones

Los suscriptores reciben notificaciones cuando un elemento cambia. Las suscripciones se gestionan automáticamente y manualmente:

- **Automático**: el sistema suscribe automáticamente al creador y al asignado del elemento. Cuando reasignas un elemento, el sistema suscribe al nuevo asignado.
- **@Menciones**: menciona a un miembro del personal con `@nombredeusuario` en un comentario para suscribirlo automáticamente.
- **Manual**: suscribir a miembros del personal o contactos de clientes a un elemento directamente, o desuscribirlos.

Los suscriptores añadidos automáticamente aún pueden desuscribirse.

Control de Acceso y Roles

Los tableros son privados para sus miembros. A cada miembro se le otorga un **rol** que gobierna lo que pueden hacer en ese tablero.

Rol	Ver	Crear / Editar Elementos	Gestionar Configuraciones (estados, etiquetas)	Gestionar Miembros	El Ta
Propietario	✓	✓	✓	✓	
Administrador	✓	✓	✓	✗	
Miembro	✓	✓	✗	✗	
Espectador	✓	✗	✗	✗	

Los administradores del sistema pueden acceder y gestionar todos los tableros independientemente de la membresía. Los usuarios no administradores solo ven los tableros de los que son miembros.

> **Nota:** Las acciones del tablero también están gobernadas por permisos a nivel de sistema (ver Usuarios y Roles). Un usuario necesita tanto el permiso del sistema relevante *como* un rol de tablero apropiado. Un tablero debe tener siempre al menos un propietario.

Permisos Relacionados con el Tablero

Permiso	Otorga la capacidad de
`VIEW_BOARD`	Listar y ver tableros
`CREATE_BOARD`	Crear nuevos tableros
`UPDATE_BOARD`	Editar detalles del tablero y gestionar grupos
`DELETE_BOARD`	Eliminar un tablero
`MANAGE_BOARD_SETTINGS`	Crear, editar y eliminar estados y etiquetas
`MANAGE_BOARD_MEMBERS`	Agregar, actualizar y eliminar miembros del tablero
`GENERATE_PUBLIC_TOKEN`	Generar un enlace de compartición pública
`VIEW_BOARD_ITEM`	Ver elementos, comentarios y suscriptores
`CREATE_BOARD_ITEM`	Crear elementos
`UPDATE_BOARD_ITEM`	Editar elementos, mover elementos y agregar comentarios
`DELETE_BOARD_ITEM`	Eliminar elementos

Visibilidad y Compartición Pública

Cada tablero tiene una configuración de visibilidad:

Visibilidad	Comportamiento
Solo Personal	El predeterminado. Solo los miembros del tablero (y los administradores del sistema) pueden acceder al tablero.
Público	Además del acceso de miembros, el tablero puede ser leído a través de un enlace público seguro sin iniciar sesión.

Generar un **token de acceso público** produce un enlace único e impredecible y cambia la visibilidad del tablero a *Público*. El enlace público es **de solo lectura**. Úsalo para incrustar una vista del tablero, por ejemplo, en una página de estado interna o en un panel. Regenerar el token invalida el enlace anterior. Para detener el acceso público, cambia la visibilidad del tablero de nuevo a *Solo Personal*.



Encontrando Elementos

Filtrado y Búsqueda

La vista del tablero y las listas de elementos admiten filtrado y búsqueda para enfocarse en lo que importa:

Filtro	Descripción
Buscar	Coincide con el título del elemento, descripción y número de elemento
Estado	Uno o más estados
Asignado	Un miembro del personal específico
Prioridad	Uno o más niveles de prioridad
Grupo	Un grupo específico
Cliente	Elementos vinculados a un cliente específico
Etiquetas	Una o más etiquetas
Vencido	Elementos cuya fecha de vencimiento ha pasado

Ordenación

Los elementos pueden ser ordenados por posición (el orden manual del tablero), fecha de creación, fecha de vencimiento, prioridad, título o fecha de última modificación, en orden ascendente o descendente. Los tableros largos están paginados.

Mejores Prácticas

Configuración del Tablero

1. **Elige la plantilla más cercana**: comienza desde *Planificación de Instalaciones* o *Solicitudes de Características* en lugar de *En Blanco* cuando se ajusten. Aún puedes personalizar después.
2. **Usa un prefijo de elemento significativo**: por ejemplo `INST` para instalaciones, para que los números de elemento sean auto-descriptivos.

3. **Mantén los estados simples**: un pequeño y claro conjunto de etapas del flujo de trabajo es más fácil de gestionar que muchos superpuestos.
4. **Marca los estados "hechos"**: para que el progreso de finalización se muestre correctamente en los encabezados de los grupos.

Uso Diario

1. **Asigna cada elemento**: la propiedad clara evita que el trabajo se detenga.
2. **Establece fechas de vencimiento**: para que el filtro *Vencido* tenga sentido.
3. **Usa @menciones**: para involucrar a las personas adecuadas en un elemento sin gestionar suscripciones manualmente.
4. **Vincula registros relacionados**: conecta elementos al cliente, servicio o ticket relevante para contexto.
5. **Mueve, no recrees**: progresa los elementos a través de estados para que el historial de cambios cuente la historia completa.

Resolución de Problemas

Problema: Un usuario no puede ver un tablero

- Confirma que el usuario ha sido agregado como miembro del tablero.
- Confirma que el usuario tiene el permiso `VIEW_BOARD` para su rol.
- Recuerda que los no administradores solo ven tableros a los que pertenecen.

Problema: Un usuario no puede cambiar estados o etiquetas

- Gestionar estados y etiquetas requiere el rol de tablero **Propietario** o **Administrador** más el permiso `MANAGE_BOARD_SETTINGS`.

Problema: Los elementos desaparecieron después de eliminar un estado o grupo

- Los elementos no se eliminan. Eliminar un estado deja sus elementos sin un estado; eliminar un grupo deja sus elementos sin agrupar. Asigna un

nuevo estado o grupo para volver a traerlos a la vista.

Problema: El enlace público no muestra el tablero

- Confirma que la visibilidad del tablero está configurada como *Público*.
- Confirma que el enlace es actual. Regenerar el token invalida el enlace anterior.

Problema: No se puede eliminar a un miembro

- Un tablero debe tener siempre al menos un propietario. Asigna otro propietario antes de eliminar al último.

Comportamientos de Acciones y Recargas de CGRateS

Esta guía explica cómo funcionan las Acciones de CGRateS dentro de OmniCRM, enfocándose específicamente en la gestión de saldos, comportamientos de recarga y cómo diferentes tipos de acciones afectan a los productos adicionales.

Resumen

En el Sistema de Carga en Línea (CGRateS) de OmniCRM, **Acciones** son el mecanismo para agregar, modificar o eliminar saldos en las cuentas de los clientes.

Cuando provisionas un producto adicional o de recarga, en realidad estás ejecutando una Acción de CGRateS que manipula los saldos de la cuenta. (También puedes tomar otros enfoques, como agregar saldos a cuentas manualmente desde Playbooks. Este es solo un patrón común que usamos para mantener las cosas limpias.)

Conceptos Clave

Acción - Un conjunto de operaciones a realizar en una cuenta (agregar saldo, deducir saldo, registrar CDR, etc.)

Saldo - Una cantidad de un recurso (datos, voz, SMS, monetario) con un tiempo de expiración y peso

ID de Saldo - Un identificador único para un tipo de saldo (por ejemplo, "Paquete de Datos", "Minutos de Voz")

Peso - Prioridad para el consumo de saldo (peso más alto consumido primero)

Tiempo de Expiración - Cuando el saldo expira (fecha absoluta o relativa como "+5 días")

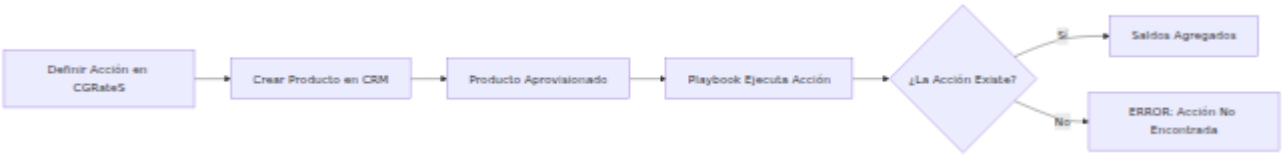
Bloqueador - Una bandera especial de saldo que bloquea todo uso cuando el saldo llega a cero, incluso si existen otros saldos (ver Bloqueadores de Saldo)

Crítico: Las Acciones Deben Definirse Primero

Antes de que puedas ejecutar una Acción en un playbook usando `ExecuteAction`, esa Acción ya debe estar definida en CGRateS. Este es un requisito crítico que a menudo se pasa por alto.

Cuándo se Definen las Acciones

Las acciones se definen típicamente durante la **configuración inicial del sistema** o **configuración del producto**, NO durante el aprovisionamiento. Generalmente se crean a través de scripts de Python que configuran simultáneamente tanto el CRM como el OCS.



Cómo se Enlazan las Acciones a los Productos

Las acciones están vinculadas a productos a través de una convención de nomenclatura:

- **Acción de CGRateS:** `ActionsId = "Action_50gb-data-pack"`
- **Producto CRM:** `product_slug = "50gb-data-pack"`
- **En Playbook:** `cgr_action_name = "Action_" + product_slug`

Cuando se ejecuta el playbook, construye el nombre de la Acción a partir del product_slug y llama a `ExecuteAction`. Si esa Acción no existe en CGRateS, el aprovisionamiento falla.

Dónde Definir Acciones

Las acciones deben definirse en tus scripts de configuración de productos, típicamente:

1. **Durante la Configuración Inicial** - Al configurar tu sistema por primera vez
2. **Al Crear Nuevos Productos** - Define la Acción antes de crear el producto
3. **A través de Scripts de Configuración** - Scripts de Python que configuran tanto OCS como CRM

Ejemplo: Definiendo una Acción antes de crear el producto

```
import cgrateshttpapi

OCS_Obj = cgrateshttpapi.CGRateS("ocs.example.com", "2080")
tenant = "tu_tenant"

# Paso 1: Define la Acción en CGRateS PRIMERO
Action_50GB_Data_Pack = {
    "method": "ApierV1.SetActions",
    "params": [{
        "ActionsId": "Action_50gb-data-pack",
        "Tenant": tenant,
        "Actions": [
            {
                "Identifier": "*topup",
                "BalanceType": "*data",
                "Units": 50 * 1024 * 1024 * 1024,
                "ExpiryTime": "+720h",
                "Weight": 10
            }
        ]
    }]
}

result = OCS_Obj.SendData(Action_50GB_Data_Pack)
assert result['error'] is None or result['error'] == "EXISTS"

# Paso 2: Ahora crea el producto en CRM
# (product_slug = "50gb-data-pack" se vinculará a Action_50gb-
data-pack)
```

Qué Sucede Si la Acción No Existe:

```
# En playbook
- name: Ejecutar Acción
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body:
      {
        "method": "APIerSv1.ExecuteAction",
        "params": [{
          "ActionsId": "Action_50gb-data-pack"
        }]
      }
  register: response

# Resultado si la Acción no está definida:
# response.json.error = "SERVER_ERROR: Acción no encontrada"
# El aprovisionamiento falla, el cliente no recibe saldo
```

Para obtener detalles completos sobre cómo definir Acciones y vincularlas a productos, consulta Definición de Productos.

Independencia de Saldo

Por defecto, los productos adicionales crean **saldos independientes** que funcionan por separado entre sí. Esto significa:

- Puedes tener múltiples addons activos simultáneamente
- Cada addon mantiene su propio saldo y expiración
- Los saldos se consumen según las reglas de peso y expiración

Ejemplo: Múltiples Addons Independientes

```
# El cliente tiene estos saldos activos:
Saldo 1:
  ID: "Data_5GB_5days_uuid_abc123"
  Tipo: *data
  Valor: 5368709120  # 5GB en bytes
  Expiración: 2024-12-29
  Peso: 10

Saldo 2:
  ID: "Data_10GB_30days_uuid_def456"
  Tipo: *data
  Valor: 10737418240  # 10GB en bytes
  Expiración: 2025-01-24
  Peso: 10

# Ambos saldos coexisten de manera independiente
# El sistema consume según el peso; si el peso es igual, el orden
no está garantizado
```

Cuando los saldos tienen el mismo peso y coinciden con el mismo destino, CGRateS no garantiza el orden de consumo basado en la expiración; el orden depende de cómo se almacenan y recuperan los saldos. **Usa diferentes pesos para controlar el orden de consumo**.

Tipos de Acción: *topup vs *topup_reset

El tipo de acción determina cómo los nuevos saldos interactúan con los saldos existentes del mismo ID.

*topup - Comportamiento Aditivo

La acción `*topup` **agrega a los saldos existentes** con el mismo ID de Saldo y **extiende el tiempo de expiración**.

Comportamiento:

- Encuentra saldo existente con ID coincidente
- Agrega nuevo valor al valor existente
- Actualiza la expiración al nuevo tiempo de expiración
- Preserva el saldo existente (rollover)

Ejemplo:

```
# Estado inicial:
Saldo:
  ID: "Data_Package__5368709120"
  Valor: 1073741824  # 1GB restante
  Expiración: 2024-12-24 (1 día restante)

# Ejecutar acción *topup con:
Acción:
  Identifier: "*topup"
  Saldo:
    ID: "Data_Package__5368709120"  # Mismo ID - activa el
rollover
    Valor: 5368709120  # 5GB
    ExpiryTime: "+5d"

# Resultado después de *topup:
Saldo:
  ID: "Data_Package__5368709120"
  Valor: 6442450944  # 6GB (1GB + 5GB rollover)
  Valor_original: 5368709120  # Sigue mostrando los 5GB originales
  Valor_hr: "6 GB"
  Valor_original_hr: "5 GB"
  Restante_hr: "6 GB de 5 GB (1 GB rollover)"
  Expiración: 2024-12-29 (5 días a partir de ahora)
```

Casos de Uso:

- Recompensas de lealtad (agregar datos adicionales al paquete existente)
- Créditos de compensación (agregar al saldo del cliente)
- Paquetes de datos rollover
- Extensiones de período de gracia

*topup_reset - Comportamiento de Reinicio

La acción `*topup_reset` **reemplaza los saldos existentes** con el mismo ID de Saldo.

Comportamiento:

- Encuentra saldo existente con ID coincidente
- Desecha el valor antiguo (sin rollover)
- Establece el saldo al nuevo valor solamente
- Actualiza la expiración al nuevo tiempo de expiración

Ejemplo:

```
# Estado inicial:
Saldo:
  ID: "Data_Package__5368709120"
  Valor: 1073741824  # 1GB restante
  Expiración: 2024-12-24 (1 día restante)

# Ejecutar acción *topup_reset con:
Acción:
  Identifier: "*topup_reset"
  Saldo:
    ID: "Data_Package__5368709120"  # Mismo ID - activa el
reinicio
    Valor: 5368709120  # 5GB
    ExpiryTime: "+5d"

# Resultado después de *topup_reset:
Saldo:
  ID: "Data_Package__5368709120"
  Valor: 5368709120  # 5GB (se descartó el antiguo 1GB)
  Valor_original: 5368709120
  Valor_hr: "5 GB"
  Restante_hr: "5 GB de 5 GB"
  PorcentajeUsado: 0
  Expiración: 2024-12-29 (5 días a partir de ahora)
```

Casos de Uso:

- Paquetes recurrentes mensuales (reiniciar a la cantidad total cada mes)
- Recargas de tamaño fijo (siempre recibir la cantidad exacta)
- Cambios de plan (reemplazar el saldo del plan antiguo con el nuevo plan)
- Prevenir abusos (no se pueden apilar addons ilimitados)

Controlando el Comportamiento del Saldo con IDs de Saldo

El **ID de Saldo** es crucial para determinar si los saldos son independientes o interactúan entre sí.

Convención de Nomenclatura de ID de Saldo y Vistas Legibles por Humanos

OmniCRM utiliza una convención de nomenclatura específica para los IDs de Saldo que codifica tanto el nombre descriptivo como el tamaño original. Esto permite que la API genere automáticamente campos legibles por humanos para la interfaz web.

Patrón de ID de Saldo:

```
{NombreDescriptivo}__{TamañoOriginalEnUnidadesBase}
```

Ejemplos de IDs de Saldo:

```
# Saldo de datos: 100GB
"AU_Data_Domestic__107374182400"
# Se descompone en:
# - Parte descriptiva: "AU_Data_Domestic" (QUÉ es - tipo/destino)
# - Separador: "__" (doble guion bajo)
# - Tamaño original: "107374182400" (100GB en bytes)
# - La UI muestra: "AU Data Domestic - 100 GB"

# Saldo de voz: 3000 minutos
"AU_Voice_Domestic__180000000000000"
# - Parte descriptiva: "AU_Voice_Domestic" (NO
"AU_Voice_Domestic_3000min")
# - Tamaño original: "180000000000000" (3000 minutos en
nanosegundos)
# - La UI muestra: "AU Voice Domestic - 3000 min"

# Saldo de SMS: 3000 mensajes
"AU_SMS_Domestic__3000"
# - Parte descriptiva: "AU_SMS_Domestic"
# - Tamaño original: "3000" (conteo)
# - La UI muestra: "AU SMS Domestic - 3000 msgs"
```

Importante: No incluyas información de tamaño en la parte descriptiva. Es redundante ya que el tamaño está codificado después de `__` y la API lo convierte automáticamente a formato legible por humanos.

Cómo la API Crea Vistas Legibles por Humanos:

Cuando la API de OmniCRM recupera datos de saldo de CGRateS, automáticamente analiza el ID de Saldo y genera campos `_hr` (legibles por humanos):

```
{
  "BalanceMap": {
    "*data": [
      {
        "ID": "AU_Data_Domestic__107374182400",
        "Value": 53687091200,
        "ExpiryTime": "2025-01-25T23:59:59Z",
        "Weight": 1200,

        // Campos legibles por humanos generados automáticamente:
        "ID_hr": "AU Data Domestic",
        "OriginalValue": 107374182400,
        "OriginalValue_hr": "100 GB",
        "Value_hr": "50 GB",
        "Remaining_hr": "50 GB de 100 GB",
        "PercentUsed": 50,
        "ExpiryTime_hr": "25 Ene 2025 (22 días)"
      }
    ]
  }
}
```

Lógica de Procesamiento de la API:

1. **Analizar ID de Saldo:**

```
balance_id = "AU_Data_Domestic__107374182400"
parts = balance_id.split("__")

descriptive_name = parts[0]  # "AU_Data_Domestic"
original_size = int(parts[1]) if len(parts) > 1 else None  # 107374182400
```

2. **Generar Nombre Descriptivo Legible por Humanos:**

```
# Reemplazar guiones bajos por espacios
id_hr = descriptive_name.replace("_", " ")  # "AU Data Domestic"
```

3. **Convertir Tamaño Original a Unidades Humanas:**

```
# Para saldos de datos (bytes)
if balance_type == "*data":
    original_value_hr = convert_bytes_to_gb(original_size)  #
"100 GB"

# Para saldos de voz (nanosegundos)
elif balance_type == "*voice":
    original_value_hr = convert_ns_to_minutes(original_size)  #
"3000 min"

# Para saldos de SMS (conteo)
elif balance_type == "*sms":
    original_value_hr = f"{original_size} msgs"  # "3000 msgs"
```

4. **Calcular Porcentaje de Uso:**

```
if original_size and original_size > 0:
    percent_used = ((original_size - current_value) /
original_size) * 100
```

5. **Formatear Visualización Restante:**

```
remaining_hr = f"{current_value_hr} de {original_value_hr}"
# "50 GB de 100 GB"
```

Visualización en la Interfaz Web:

El frontend utiliza estos campos `_hr` para mostrar información de saldo amigable para el usuario:

```
// En lugar de mostrar valores en bruto:
// ID: "AU_Data_Domestic__107374182400"
// Valor: 53687091200

// Mostrar legible por humanos:
<BalanceCard>
  <Title>{balance.ID_hr}</Title>           {/* "AU Data Domestic" */}
  <Progress value={balance.PercentUsed}>   {/* 50% */}
    {balance.Remaining_hr}                 {/* "50 GB de 100 GB" */}
  </Progress>
  <Expiry>{balance.ExpiryTime_hr}</Expiry> {/* "25 Ene 2025 (22 días)" */}
</BalanceCard>
```

Por Qué Esto Importa:

1. **Seguimiento del Tamaño Original** - Incluso cuando los saldos están parcialmente consumidos, la UI puede mostrar "50 GB de 100 GB" en lugar de solo "50 GB restantes"

2. **Visualización del Progreso** - Los cálculos de porcentaje permiten barras de progreso precisas

3. **Nomenclatura Consistente** - Los nombres descriptivos extraídos de los IDs de Saldo aseguran consistencia entre el backend y el frontend

4. **Visualización de Rollover** - Al usar `*topup` (rollover), si un cliente tiene 70 GB restantes y recarga 100 GB:
   - El ID de Saldo permanece: `"AU_Data_Domestic__107374182400"` (original 100 GB)
   - El valor actual se convierte en: 170 GB
   - La UI muestra: "170 GB (70 GB rollover + 100 GB nuevo)"

Mejor Práctica - Creación de IDs de Saldo:

Siempre incluye el tamaño original después de `__` para una visualización adecuada en la UI. No dupliques la información de tamaño en el nombre

descriptivo:

```
# Bueno - nombre descriptivo + tamaño en unidades base
Action_Data_100GB = {
    "Actions": [
        {
            "BalanceId": f"AU_Data_Domestic__{100 * 1024 * 1024 *
1024}",
            "Units": 100 * 1024 * 1024 * 1024
        }
    ]
}

# Malo - tamaño redundante en el nombre descriptivo
Action_Data_100GB = {
    "Actions": [
        {
            "BalanceId": f"AU_Data_Domestic_100GB__{100 * 1024 *
1024 * 1024}",  # ¡Redundante!
            "Units": 100 * 1024 * 1024 * 1024
        }
    ]
}

# Malo - sin información de tamaño (la UI no puede calcular
porcentaje)
Action_Data_100GB = {
    "Actions": [
        {
            "BalanceId": "AU_Data_Domestic",  # Falta __tamaño
            "Units": 100 * 1024 * 1024 * 1024
        }
    ]
}
```

Caso Especial - Saldos Monetarios:

Los saldos monetarios no suelen incluir tamaño original ya que pueden recargarse a cualquier cantidad:

```
# Saldo monetario sin codificación de tamaño
{
    "BalanceId": "PAYG_Monetary_Balance",
    "BalanceType": "*monetary",
    "Units": 5000  # $50.00
}

# La UI simplemente muestra el saldo actual sin porcentaje
# "Saldo: $50.00"
```

Estrategia 1: IDs Únicos (Saldos Independientes)

Usa IDs únicos (por ejemplo, con UUIDs) para crear saldos completamente independientes que nunca interactúan.

Ejemplo de Implementación:

```
- name: Generar identificador de saldo único
  set_fact:
    uuid: "{{ 99999999 | random | to_uuid }}"
    balance_id: "Data_5days__5368709120_{{ uuid[0:8] }}"

- name: Agregar saldo independiente con *topup
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "BalanceType": "*data",
          "Balance": {
            "ID": "{{ balance_id }}",
            "Value": 5368709120,
            "ExpiryTime": "+5d",
            "Weight": 10
          }
        }]
      }
```

Resultado: Cada addon crea un nuevo saldo separado incluso si el cliente compra el mismo addon varias veces.

```yaml
# El cliente compra "addon de datos de 5 días" tres veces:
Saldo 1:
  ID: "Data_5days__5368709120_a1b2c3d4"
  Valor: 5368709120
  Valor_hr: "5 GB"
  Valor_original_hr: "5 GB"
  Restante_hr: "5 GB de 5 GB"
  Expiración: 2024-12-29

Saldo 2:
  ID: "Data_5days__5368709120_e5f6g7h8"
  Valor: 5368709120
  Valor_hr: "5 GB"
  Restante_hr: "5 GB de 5 GB"
  Expiración: 2024-12-30

Saldo 3:
  ID: "Data_5days__5368709120_i9j0k1l2"
  Valor: 5368709120
  Valor_hr: "5 GB"
  Restante_hr: "5 GB de 5 GB"
  Expiración: 2024-12-31

# Total disponible: 15GB en tres saldos separados
# Cada uno se muestra individualmente en la UI como "Datos 5 días
- 5 GB de 5 GB"
```

Estrategia 2: IDs Compartidos con *topup (Rollover)

Usa el mismo ID de Saldo con la acción `*topup` para permitir el rollover de saldo y la extensión de la expiración.

Ejemplo de Implementación:

```
- name: Establecer ID de saldo fijo
  set_fact:
    balance_id: "Data_5days__5368709120"

- name: Agregar saldo con *topup (rollover)
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "BalanceType": "*data",
          "Balance": {
            "ID": "Data_5days__5368709120",
            "Value": 5368709120,
            "ExpiryTime": "+5d",
            "Weight": 10
          }
        }]
      }
```

Resultado: Compras posteriores se suman al saldo existente y extienden la expiración.

```yaml
# Día 1: El cliente compra "addon de datos de 5 días":
Saldo:
  ID: "Data_5days__5368709120"
  Valor: 5368709120
  Valor_hr: "5 GB"
  Restante_hr: "5 GB de 5 GB"
  PorcentajeUsado: 0
  Expiración: 2024-12-29

# Día 3: El cliente usó 1GB, luego compra el mismo addon
nuevamente:
Saldo:
  ID: "Data_5days__5368709120"
  Valor: 9663676416  # 4GB restantes + 5GB nuevos
  Valor_hr: "9 GB"
  Valor_original_hr: "5 GB"
  Restante_hr: "9 GB (4 GB rollover + 5 GB nuevos)"
  PorcentajeUsado: -80  # Negativo indica rollover
  Expiración: 2024-12-27 (nuevos +5 días a partir de hoy)
```

Estrategia 3: IDs Compartidos con *topup_reset (Cantidad Fija)

Usa el mismo ID de Saldo con la acción `*topup_reset` para restablecer siempre a una cantidad fija.

Ejemplo de Implementación:

```
- name: Establecer ID de saldo fijo
  set_fact:
    balance_id: "Monthly_Plan__32212254720"

- name: Agregar saldo con *topup_reset (sin rollover)
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.SetActions",
        "params": [{
          "ActionsId": "Action_Reset_Monthly_Plan",
          "Actions": [{
            "Identifier": "*topup_reset",
            "BalanceType": "*data",
            "Units": 32212254720,  # 30GB
            "ExpiryTime": "*monthly",
            "DestinationIds": "*any",
            "BalanceId": "Monthly_Plan__32212254720",
            "Weight": 10
          }]
        }]
      }
```

Resultado: Cada mes, el saldo se restablece exactamente a 30GB independientemente de cuánto se haya utilizado.

```yaml
# Mes 1, Día 1:
Saldo:
  ID: "Monthly_Plan__32212254720"
  Valor: 32212254720
  Valor_hr: "30 GB"
  Restante_hr: "30 GB de 30 GB"
  PorcentajeUsado: 0
  Expiración: 2024-12-31

# Mes 1, Día 25: El cliente usó 28GB
Saldo:
  ID: "Monthly_Plan__32212254720"
  Valor: 2147483648
  Valor_hr: "2 GB"
  Restante_hr: "2 GB de 30 GB"
  PorcentajeUsado: 93
  Expiración: 2024-12-31

# Mes 2, Día 1: El ActionPlan ejecuta *topup_reset
Saldo:
  ID: "Monthly_Plan__32212254720"
  Valor: 32212254720  # Restablecido a completo, se pierden 2GB no
utilizados
  Valor_hr: "30 GB"
  Restante_hr: "30 GB de 30 GB"
  PorcentajeUsado: 0
  Expiración: 2025-01-31
```

Bloqueadores de Saldo

Los Bloqueadores de Saldo son una poderosa característica de CGRateS que te permite **bloquear o limitar el uso** incluso cuando existen saldos. Un saldo bloqueador detiene el consumo cuando llega a cero, impidiendo un uso adicional independientemente de otros saldos disponibles.

Cómo Funcionan los Bloqueadores

Cuando un saldo tiene `Blocker: true`:

1. **Mientras el bloqueador tenga valor** → Se permite el uso hasta la cantidad del bloqueador
2. **Cuando el bloqueador llega a cero** → Se detiene todo uso, incluso si existen otros saldos
3. **Error devuelto** → `INSUFFICIENT_CREDIT_BALANCE_BLOCKER` previene la sesión

Características Clave:

- Los saldos bloqueadores son **verificados incluso cuando el valor es cero** (a diferencia de los saldos normales que se omiten)
- Cuando se encuentra un bloqueador con uso restante solicitado, CGRateS **detiene el procesamiento** y devuelve un error
- Los bloqueadores funcionan con todos los tipos de saldo: `*voice`, `*data`, `*sms`, `*monetary`

Casos de Uso para Bloqueadores

1. Suspensión de Cuenta

Bloquear todo uso cuando una cuenta está suspendida (por ejemplo, fallo de pago):

```
Action_Suspend_Account = {
    "id": "0",
    "method": "ApierV1.SetActions",
    "params": [{
        "ActionsId": "Action_suspend-account",
        "Overwrite": True,
        "Tenant": tenant,
        "Actions": [
            # Agregar bloqueador de valor cero para prevenir todo
uso
            {
                "Identifier": "*topup",
                "BalanceId": "Suspension_Blocker",
                "BalanceType": "*monetary",
                "DestinationIDs": "*any",
                "Units": 0,  # Valor cero
                "BalanceWeight": 9999,  # Mayor prioridad -
verificado primero
                "Blocker": True,  # Bloquear todo uso
                "Weight": 10
            }
        ]
    }]
}

result = OCS_Obj.SendData(Action_Suspend_Account)
```

Resultado: Todas las llamadas/datos/SMS bloqueados independientemente de otros saldos.

2. Límites de Gastos

Limitar el gasto máximo para prevenir sorpresas en la factura:

```
Action_Monthly_Plan_With_Cap = {
    "id": "0",
    "method": "ApierV1.SetActions",
    "params": [{
        "ActionsId": "Action_monthly-with-cap",
        "Overwrite": True,
        "Tenant": tenant,
        "Actions": [
            # 10GB de datos incluidos
            {
                "Identifier": "*topup_reset",
                "BalanceId": f"Included_Data__{10 * 1024 * 1024 *
1024}",
                "BalanceType": "*data",
                "DestinationIDs": "Dest_PLMN_OnNet",
                "Units": 10 * 1024 * 1024 * 1024,
                "ExpiryTime": "*month",
                "BalanceWeight": 1200,  # Consumido primero
                "Weight": 95
            },
            # Límite de exceso de $50 (bloqueador)
            {
                "Identifier": "*topup_reset",
                "BalanceId": "Overage_Cap",
                "BalanceType": "*monetary",
                "DestinationIDs": "*any",
                "Units": 5000,  # $50.00 máximo de exceso
                "ExpiryTime": "*month",
                "BalanceWeight": 1000,  # Consumido después de lo
incluido
                "Blocker": True,  # Detener cuando se gaste $50
                "Weight": 90
            }
        ]
    }]
}
```

Flujo:

1. El cliente usa 10GB incluidos → GRATIS (de Included_Data_10GB)
2. El cliente usa 5GB adicionales → Cargado de Overage_Cap (tarifas PAYG)

3. Cuando Overage_Cap llega a $0 → **Todo uso bloqueado** (límite de gasto alcanzado)

3. Prueba Gratuita Limitada en el Tiempo

Proporcionar uso gratuito limitado para cuentas de prueba:

```
Action_Trial_Account = {
    "id": "0",
    "method": "ApierV1.SetActions",
    "params": [{
        "ActionsId": "Action_trial-100-minutes",
        "Overwrite": True,
        "Tenant": tenant,
        "Actions": [
            # 100 minutos gratuitos (bloqueador - se detiene
cuando se agota)
            {
                "Identifier": "*topup",
                "BalanceId": f"Trial_Voice__{100 * 60 *
1000000000}",
                "BalanceType": "*voice",
                "DestinationIDs": "Dest_Domestic_All",
                "Units": 100 * 60 * 1000000000,  # 100 minutos
                "ExpiryTime": "+720h",  # 30 días
                "BalanceWeight": 1200,
                "Blocker": True,  # Sin uso después de 100 minutos
                "Weight": 10
            }
        ]
    }]
}
```

Resultado: El cliente obtiene exactamente 100 minutos gratis. Después de eso, todas las llamadas están bloqueadas (sin cargos automáticos).

4. Bloqueo Específico por Destino

Bloquear destinos específicos mientras se permiten otros:

```
Action_Block_Premium_Numbers = {
    "id": "0",
    "method": "ApierV1.SetActions",
    "params": [{
        "ActionsId": "Action_block-premium",
        "Overwrite": True,
        "Tenant": tenant,
        "Actions": [
            # Uso regular permitido
            {
                "Identifier": "*topup_reset",
                "BalanceId": "Regular_Usage",
                "BalanceType": "*monetary",
                "DestinationIDs": "Dest_Domestic_All",
                "Units": 10000,  # $100
                "ExpiryTime": "*month",
                "BalanceWeight": 1000,
                "Weight": 20
            },
            # Bloquear números premium (0900, etc.)
            {
                "Identifier": "*topup",
                "BalanceId": "Premium_Blocker",
                "BalanceType": "*monetary",
                "DestinationIDs": "Dest_Domestic_Premium",
                "Units": 0,  # Valor cero
                "BalanceWeight": 2000,  # Peso más alto - revisado
primero para premium
                "Blocker": True,
                "Weight": 10
            }
        ]
    }]
}
```

Flujo:

- Llamadas nacionales → Usa el saldo Regular_Usage
- Llamadas a números premium → Coincide con Premium_Blocker (peso 2000 > 1000) → **BLOQUEADO**

Bloqueador vs Saldos Desactivados

No confundir `Blocker` con `Disabled`:

Característica	Bloqueador	Desactivado
Propósito	Detener el uso cuando el saldo se agota	Pausar temporalmente un saldo
Cuando el valor > 0	El saldo se puede usar normalmente	El saldo se omite/ignora
Cuando el valor = 0	**Bloquea todo uso posterior**	El saldo se omite (se prueba el siguiente saldo)
Caso de Uso	Límites de gasto, topes, límites de prueba	Suspender temporalmente un saldo específico

```
# Bloqueador: Permite 10GB, luego bloquea todo
{
    "BalanceId": f"Data_Cap__{10 * 1024 * 1024 * 1024}",
    "Units": 10 * 1024 * 1024 * 1024,
    "Blocker": True  # Detiene el uso a 10GB
}

# Desactivado: Ignora este saldo por completo (pausado
temporalmente)
{
    "BalanceId": f"Bonus_Data__{5 * 1024 * 1024 * 1024}",
    "Units": 5 * 1024 * 1024 * 1024,
    "Disabled": True  # Este saldo no se usará en absoluto
}
```

Ejemplo Práctico: Plan Híbrido con Tope de Seguridad

Combinar saldos unitarios, desbordamiento monetario y un tope bloqueador:

```
Action_Safe_Hybrid_Plan = {
    "id": "0",
    "method": "ApierV1.SetActions",
    "params": [{
        "ActionsId": "Action_safe-hybrid-plan",
        "Overwrite": True,
        "Tenant": tenant,
        "Actions": [
            {
                "Identifier": "*reset_account",
                "Weight": 700
            },
            # 500 minutos nacionales incluidos
            {
                "Identifier": "*topup_reset",
                "BalanceId": f"Domestic_Voice__{500 * 60 *
1000000000}",
                "BalanceType": "*voice",
                "DestinationIDs": "Dest_Domestic_All",
                "Units": 500 * 60 * 1000000000,
                "ExpiryTime": "*month",
                "BalanceWeight": 1200,
                "Weight": 95
            },
            # $20 de límite de exceso
            {
                "Identifier": "*topup_reset",
                "BalanceId": "Overage_Allowance",
                "BalanceType": "*monetary",
                "DestinationIDs": "*any",
                "Units": 2000,  # $20.00
                "ExpiryTime": "*month",
                "BalanceWeight": 1000,
                "Weight": 90
            },
            # LÍMITE DURO DE $50 (bloqueador)
            {
                "Identifier": "*topup_reset",
                "BalanceId": "Hard_Spending_Cap",
                "BalanceType": "*monetary",
                "DestinationIDs": "*any",
                "Units": 5000,  # $50.00 máximo absoluto
                "ExpiryTime": "*month",
```

```
                "BalanceWeight": 500,  # Menor que el exceso - usado al final
                "Blocker": True,  # DETENER cuando se alcance el tope
                "Weight": 85
            }
        ]
    }]
}
```

Viaje del Cliente:

1. **0-500 minutos:** Usa Domestic_Voice_500min (GRATIS)
2. **500-700 minutos:** Usa Overage_Allowance a $0.10/min = $20 (200 min)
3. **700-1200 minutos:** Usa Hard_Spending_Cap a $0.10/min = $50 (500 min)
4. **A 1200 minutos:** Hard_Spending_Cap agotado → **TODO USO BLOQUEADO**

El cliente obtiene un total de 1200 minutos, con un gasto limitado a un máximo de $50 en exceso.

Mejores Prácticas con Bloqueadores

1. **Usar Alto Peso para Bloqueadores**

   ```
   "BalanceWeight": 9999  # Asegurar que el bloqueador sea revisado primero
   ```

2. **Bloqueadores de Valor Cero para Bloqueo Inmediato**

   ```
   "Units": 0,  # Bloquear inmediatamente
   "Blocker": True
   ```

3. **Notificar a los Clientes Antes de la Agotamiento del Bloqueador**

   - Usar ActionTriggers para enviar notificaciones al 80%, 90%, 100% del uso del bloqueador

- Dar a los clientes la opción de aumentar el tope antes de que ocurra el bloqueo

4. **Eliminar Bloqueadores al Desuspender**

```
# Usar *remove_balance para eliminar el bloqueador
{
    "Identifier": "*remove_balance",
    "BalanceId": "Suspension_Blocker"
}
```

5. **Probar el Comportamiento del Bloqueador**

   - Verificar que el bloqueador devuelve el error `INSUFFICIENT_CREDIT_BALANCE_BLOCKER`
   - Confirmar que los CDRs muestran costo = -1.0 cuando está bloqueado
   - Probar que otros saldos NO se usan después del agotamiento del bloqueador

Solución de Problemas de Bloqueadores

Problema: Uso no bloqueado a pesar de que el bloqueador está en cero

Causas Posibles:

1. Peso del bloqueador demasiado bajo (otros saldos revisados primero)
2. DestinationIDs no coinciden con el destino de uso
3. Campo de bloqueador no configurado en `True`

Solución:

```
# Verificar la configuración del bloqueador
OCS_Obj.SendData({
    'method': 'ApierV2.GetAccount',
    'params': [{"Tenant": tenant, "Account": "service_uuid"}]
})

# Comprobar:
# - Blocker: true
# - BalanceWeight es el más alto (por ejemplo, 9999)
# - DestinationIDs incluye el destino de uso
# - Valor = 0
```

Problema: Uso bloqueado inesperadamente

Causa Posible: Bloqueador creado involuntariamente con un valor bajo

Solución: Verificar todos los saldos para `Blocker: true` y verificar que sus valores sean apropiados para su caso de uso.

Reglas de Consumo de Saldos

Regla 1: Prioridad de Precisión de Destino

Los saldos con **mayor precisión de destino** (coincidencia de destino más específica) se consumen primero. Esto se determina por la longitud de coincidencia de prefijo.

```
# El cliente llama a +44-20-1234-5678 (Londres, Reino Unido)

Saldo 1:
  DestinationIDs: "Dest_UK_London"  # Prefijo: "4420" (precisión:
4)
  Valor: 100 minutos
  Peso: 10

Saldo 2:
  DestinationIDs: "Dest_UK_All"     # Prefijo: "44" (precisión: 2)
  Valor: 200 minutos
  Peso: 10

# El saldo 1 se consume primero (precisión 4 > precisión 2)
# La coincidencia de destino más específica gana
```

Caso de Uso: Los saldos específicos de la ciudad o región tienen prioridad sobre los saldos a nivel nacional.

Regla 2: Prioridad de Peso (Misma Precisión)

Cuando la precisión de destino es igual, los saldos con **mayor peso** se consumen primero.

```
Saldo 1:
  ID: "Premium_Data"
  Valor: 5GB
  Peso: 20

Saldo 2:
  ID: "Standard_Data"
  Valor: 10GB
  Peso: 10

# El saldo 1 se consume primero (peso 20 > peso 10)
# A pesar de que el saldo 2 tiene más datos
```

Caso de Uso: Los saldos prioritarios (datos adicionales consumidos antes que los datos regulares).

Regla 3: Primero el Más Antiguo

Cuando el peso coincide, se usa primero el saldo más antiguo.

```
Saldo 1:
  ID: "Data_Package_A"
  Valor: 5GB
  Expiración: 2024-12-25
  Peso: 10

Saldo 2:
  ID: "Data_Package_B"
  Valor: 10GB
  Expiración: 2025-01-15
  Peso: 10
```

Para controlar el orden de consumo, use diferentes pesos:

```
# Enfoque correcto: Use peso para priorizar saldos que están a
punto de expirar
Saldo 1:
  ID: "Data_Package_A"
  Valor: 5GB
  Expiración: 2024-12-25
  Peso: 11  # Peso más alto = consumido primero

Saldo 2:
  ID: "Data_Package_B"
  Valor: 10GB
  Expiración: 2025-01-15
  Peso: 10  # Peso más bajo = consumido segundo

# El saldo 1 se consume primero (peso 11 > peso 10)
```

Mejor Práctica: Si desea asegurarse de que los saldos que están a punto de expirar se consuman primero, asígneles pesos más altos al crearlos.

Ejemplos Prácticos

Ejemplo 1: Complemento de Datos Simple (Independiente)

Escenario: El cliente puede comprar un complemento de "5GB 5 días" varias veces, cada uno creando un saldo separado.

Implementación:

```
- name: Generar UUID para saldo único
  set_fact:
    uuid: "{{ 99999999 | random | to_uuid }}"

- name: Agregar saldo independiente de 5GB
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "BalanceType": "*data",
          "Categories": "*any",
          "Balance": {
            "ID": "Data_5GB_{{ uuid[0:8] }}",
            "Value": 5368709120,
            "ExpiryTime": "+120h",  # 5 días
            "Weight": 10
          }
        }]
      }
```

Experiencia del Cliente:

- Compra el complemento el 24 de diciembre → Obtiene 5GB que expiran el 29 de diciembre

- Compra el complemento el 25 de diciembre → Obtiene 5GB que expiran el 30 de diciembre
- Ambos saldos coexisten; el orden de consumo depende de los pesos de los saldos (ambos tienen peso 10, por lo que el orden no está garantizado)

Ejemplo 2: Paquete de Datos de Rollover

Escenario: "Plan mensual de 50GB" donde los datos no utilizados se transfieren cuando el cliente recarga temprano.

Implementación:

```
- name: Agregar saldo de datos de rollover
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "BalanceType": "*data",
          "Balance": {
            "ID": "Rollover_Monthly_50GB",
            "Value": 53687091200,  # 50GB
            "ExpiryTime": "+720h",  # 30 días
            "Weight": 10
          }
        }]
      }
```

Tipo de Acción: Usa el comportamiento predeterminado de `*topup` (rollover habilitado)

Experiencia del Cliente:

- Día 1: Obtiene 50GB que expiran en 30 días
- Día 20: Usó 30GB, tiene 20GB restantes

- Día 20: Recarga nuevamente → Obtiene 70GB en total (20GB + 50GB), la expiración se extiende a +30 días desde el día 20

Ejemplo 3: Plan Mensual Fijo (Sin Rollover)

Escenario: Plan "Ilimitado de 100GB mensual" que se restablece exactamente a 100GB cada mes, sin rollover.

Implementación:

```
- name: Crear acción de reinicio mensual
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.SetActions",
        "params": [{
          "ActionsId": "Action_Monthly_100GB_Reset",
          "Overwrite": true,
          "Actions": [{
            "Identifier": "*topup_reset",
            "BalanceType": "*data",
            "Units": 107374182400,  # 100GB
            "ExpiryTime": "*monthly",
            "BalanceId": "Monthly_Plan__107374182400",
            "Weight": 10
          }]
        }]
      }

- name: Crear Plan de Acción mensual
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.SetActionPlan",
        "params": [{
          "Id": "ActionPlan_Monthly_100GB",
          "ActionPlan": [{
            "ActionsId": "Action_Monthly_100GB_Reset",
            "Time": "*monthly",
            "Weight": 10
          }],
          "Overwrite": true,
          "ReloadScheduler": true
        }]
      }

- name: Asignar Plan de Acción a la cuenta
```

```
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV2.SetAccount",
        "params": [{
          "Account": "{{ service_uuid }}",
          "ActionPlanIds": ["ActionPlan_Monthly_100GB"],
          "ReloadScheduler": true
        }]
      }
```

Experiencia del Cliente:

- Mes 1: Obtiene 100GB, usa 95GB, tiene 5GB restantes
- Mes 2: El saldo se restablece a 100GB (se pierde 5GB de datos no utilizados)
- Mes 2: Usa 20GB, tiene 80GB restantes
- Mes 3: El saldo se restablece a 100GB (se pierden 80GB de datos no utilizados)

Ejemplo 4: Saldos de Múltiples Niveles con Prioridad de Peso

Escenario: El cliente tiene "Datos Adicionales" (alta prioridad) y "Datos Regulares" (baja prioridad). Los datos adicionales se consumen primero.

Implementación:

```
# Agregar datos adicionales con alto peso
- name: Agregar saldo de datos adicionales
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
          "Account": "{{ service_uuid }}",
          "BalanceType": "*data",
          "Balance": {
            "ID": "Bonus_Data",
            "Value": 5368709120,  # 5GB
            "ExpiryTime": "+240h",  # 10 días
            "Weight": 20  # Mayor prioridad
          }
        }]
      }

# Agregar datos regulares con peso normal
- name: Agregar saldo de datos regulares
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
          "Account": "{{ service_uuid }}",
          "BalanceType": "*data",
          "Balance": {
            "ID": "Regular_Data",
            "Value": 53687091200,  # 50GB
            "ExpiryTime": "+720h",  # 30 días
            "Weight": 10  # Prioridad normal
          }
        }]
      }
```

Experiencia del Cliente:

- Tiene 5GB de datos adicionales (peso 20) + 50GB de datos regulares (peso 10)
- Consume primero todos los 5GB de datos adicionales
- Luego consume del pool de 50GB de datos regulares
- Los datos regulares se preservan por más tiempo

Identificadores de Acción Comunes

CGRateS admite múltiples identificadores de acción para diferentes operaciones:

Manipulación de Saldos

***topup** - Agregar al saldo existente (rollover) ***topup_reset** - Restablecer saldo a nuevo valor (sin rollover) ***debit** - Restar del saldo ***debit_reset** - Establecer saldo a valor negativo ***reset_account** - Eliminar todos los saldos

Diseño de Productos Adicionales

Al diseñar productos adicionales en OmniCRM, considere estas preguntas:

Pregunta 1: ¿Deben apilarse los saldos?

Sí (Independiente) → Use ID de Saldo únicos (con UUID) **No (Reemplazar)** → Use ID de Saldo fijo con `*topup_reset` **Sí (Rollover)** → Use ID de Saldo fijo con `*topup`

Pregunta 2: ¿Qué pasa con el saldo no utilizado?

Rollover → Use acción `*topup` **Perdido** → Use acción `*topup_reset` **Pools separados** → Use ID de Saldo únicos

Pregunta 3: ¿Cómo deben consumirse los saldos?

Primero el más antiguo → Use diferentes pesos (asigne un peso más alto a los saldos más antiguos/próximos a expirar) **Primero el premium** → Diferentes pesos (peso más alto = mayor prioridad) **Orden específico** → Use pesos: 30 (premium), 20 (bonus), 10 (regular) **Aleatorio/Sin preferencia** → Use el mismo peso (el orden de consumo no está garantizado)

Pregunta 4: ¿Cuál es la estrategia de expiración?

Duración fija → Use expiración relativa (`+720h` para 30 días) **Fin de mes** → Use `*monthly` en ActionPlan **Nunca expira** → Use `*unlimited` o duración muy larga

Estructura de Acción de CGRateS en Playbooks

Aquí está la estructura completa para crear una Acción:

```
- name: Crear Acción de CGRateS
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.SetActions",
        "params": [{
          "ActionsId": "Action_Name_Here",
          "Overwrite": true,  # Reemplazar si existe
          "Actions": [
            {
              "Identifier": "*topup",  # o *topup_reset
              "BalanceType": "*data",  # *data, *voice, *sms,
*monetary
              "Units": 5368709120,  # Monto a agregar
              "ExpiryTime": "+120h",  # +Xh, *unlimited, *monthly
              "DestinationIds": "*any",  # Usualmente *any
              "BalanceId": "Balance_Name",  # Único o compartido
              "Weight": 10,  # Prioridad (más alto = consumido
primero)
              "Blocker": false,  # Si es verdadero, evitar saldo
negativo
              "Disabled": false,  # Si es verdadero, el saldo está
desactivado
              "SharedGroups": ""  # Grupo de saldo compartido
(opcional)
            },
            {
              "Identifier": "*cdrlog",  # Registrar esta acción
como CDR
              "BalanceType": "*generic",
              "ExtraParameters": "
{\"Category\":\"^activation\",\"Destination\":\"Addon Name\"}"
            }
          ]
        }]
      }
```

Descripciones de Campos

ActionsId - Identificador único para este conjunto de acciones

Identifier - El tipo de operación (`*topup`, `*topup_reset`, `*cdrlog`, etc.)

BalanceType - Tipo de saldo:

- `*data` - Saldos de datos (bytes)
- `*voice` - Saldos de voz (segundos)
- `*sms` - Saldos de SMS (conteo)
- `*monetary` - Saldos monetarios (unidades de moneda)
- `*generic` - Saldos genéricos

Units - Monto a agregar/restar (en unidades base: bytes para datos, segundos para voz)

ExpiryTime - Cuándo expira el saldo:

- `+Xh` - Relativo (por ejemplo, `+720h` = 30 días)
- `*unlimited` - Nunca expira
- `*monthly` - Fin de mes
- `2024-12-31T23:59:59Z` - Marca de tiempo absoluta

BalanceId - Identificador para este saldo (ID compartido = interactúa, ID único = independiente)

Weight - Prioridad (número más alto = mayor prioridad, consumido primero)

Blocker - Si es verdadero, evita que la cuenta se vuelva negativa

Disabled - Si es verdadero, el saldo existe pero no se puede usar

Definiendo Acciones a través de Python (Configuración Inicial)

Las acciones se definen típicamente durante la configuración inicial del sistema utilizando scripts de Python con la biblioteca `cgrateshttpapi`. Estos ejemplos muestran cómo definir Acciones utilizando `OCS_Obj.SendData()`.

Requisitos Previos

```
import cgrateshttpapi
import time

OCS_Obj = cgrateshttpapi.CGRateS("ocs.example.com", "2080")
tenant = "your_tenant_name"
tpid = str(tenant) + "_" + str(int(time.time()))
```

Definición de Destinos

Antes de crear Acciones, debes definir destinos que especifiquen DÓNDE se pueden usar los saldos.

Los Destinos vienen en dos tipos:

- **Destinos Geográficos** - Prefijos de número para voz/SMS A lugares (por ejemplo, `Dest_International_UK`)
- **Destinos PLMN** - Códigos de red para datos DESDE lugares (por ejemplo, `Dest_PLMN_OnNet`, `Dest_PLMN_US_Verizon`)

Regla Crítica:

- **Saldos de voz/SMS** → Usa destinos geográficos (el número que se llama A)
- **Saldos de datos** → Usa destinos PLMN (la red a la que está conectado el cliente DESDE)

Para la configuración completa de destinos, incluyendo:

- Destinos geográficos (nacionales, internacionales, gratuitos, premium)
- Destinos PLMN (en red, redes de roaming, zonas)
- Reglas de formato PLMN y mejores prácticas
- Solución de problemas de destinos

Consulta: **Configuración de Destinos CGRateS**

Cálculos de Unidades

Entender las conversiones de unidades es crítico para definir los saldos correctamente:

```
# Saldos de datos (en bytes)
1_GB = 1 * 1024 * 1024 * 1024  # 1073741824 bytes
100_GB = 100 * 1024 * 1024 * 1024

# Saldos de voz (en nanosegundos)
1_minute = 60 * 1000000000  # 60 mil millones de nanosegundos
3000_minutes = 3000 * 60 * 1000000000

# Saldos de SMS (en cantidad)
3000_sms = 3000
```

Ejemplo 1: Plan Mensual de Múltiples Saldos (Python)

Un plan mensual integral con saldos de datos, voz, SMS y roaming que se restablecen cada mes.

```
Action_AU_Premium_Plan_1 = {
    "id": "0",
    "method": "ApierV1.SetActions",
    "params": [{
        "ActionsId": "Action_au-premium-plan-1",
        "Overwrite": True,
        "Tenant": str(tenant),
        "Actions": [
            # Primero, restablecer la cuenta para limpiar saldos
antiguos
            {
                "Identifier": "*reset_account",
                "Weight": 700
            },
            # Agregar saldo de datos de 100GB
            # IMPORTANTE: Los saldos de datos utilizan destinos
PLMN (la red a la que está conectado el cliente)
            # NO destinos geográficos. Usa TU PLMN en red.
            {
                "Identifier": "*topup_reset",
                "BalanceId": "AU_Data_Domestic__" + str(100 * 1024
* 1024 * 1024),
                "BalanceType": "*data",
                "DestinationIDs": "Dest_PLMN_OnNet",  # Tu PLMN en
red (mcc505.mnc057)
                "Units": 100 * 1024 * 1024 * 1024,
                "ExpiryTime": "*month",
                "BalanceWeight": 1200,
                "Weight": 90
            },
            # Agregar saldo de voz de 3000 minutos
            {
                "Identifier": "*topup_reset",
                "BalanceId": "AU_Voice_Domestic__" + str(3000 * 60
* 1000000000),
                "BalanceType": "*voice",
                "DestinationIDs":
"Dest_AU_Mobile;Dest_AU_Fixed;Dest_AU_TollFree;",
                "Units": 3000 * 60 * 1000000000,
                "ExpiryTime": "*month",
                "BalanceWeight": 1200,
                "Weight": 89
            },
```

```
            # Agregar saldo de 3000 SMS
            {
                "Identifier": "*topup_reset",
                "BalanceId": "AU_SMS_Domestic__" + str(3000),
                "BalanceType": "*sms",
                "DestinationIDs": "Dest_AU_Mobile;",
                "Units": 3000,
                "ExpiryTime": "*month",
                "BalanceWeight": 1200,
                "Weight": 88
            },
            # Agregar 6GB de datos de roaming
            {
                "Identifier": "*topup_reset",
                "BalanceId": "AU_Roaming_Data__" + str(6 * 1024 *
1024 * 1024),
                "BalanceType": "*data",
                "DestinationIDs": "Dest_Roaming_All",
                "Units": 6 * 1024 * 1024 * 1024,
                "ExpiryTime": "*month",
                "BalanceWeight": 1100,
                "Weight": 87
            },
            # Registrar esta acción como un CDR
            {
                "Identifier": "*cdrlog",
                "BalanceId": "",
                "BalanceUuid": "",
                "BalanceType": "*generic",
                "Directions": "*out",
                "Units": 0,
                "ExpiryTime": "",
                "Filter": "",
                "TimingTags": "",
                "DestinationIds": "",
                "RatingSubject": "",
                "Categories": "",
                "SharedGroups": "",
                "BalanceWeight": 0,
                "ExtraParameters": "
{\"Category\":\"^activation\",\"Destination\":\"AU Premium Plan
1\"}",
                "BalanceBlocker": "false",
                "BalanceDisabled": "false",
```

```
                "Weight": 80
            }
        ]
    }]
}

# Enviar la definición de la acción a CGRateS
result = OCS_Obj.SendData(Action_AU_Premium_Plan_1)
assert result['error'] is None or result['error'] == "EXISTS"
print("Acción creada: Action_au-premium-plan-1")
```

Puntos Clave:

- Utiliza `*reset_account` para limpiar saldos antiguos primero
- Utiliza `*topup_reset` para asignaciones mensuales fijas (sin acumulación)
- BalanceWeight determina el orden de consumo (nacional 1200 > roaming 1100)
- Weight determina el orden de ejecución dentro de la Acción
- Incluye `*cdrlog` para rastrear activaciones

Ejemplo 2: Complemento de Datos Simple (Python)

Un complemento de datos simple de 20GB con acumulación desactivada.

```
Action_AU_Data_Addon_20GB = {
    "id": "0",
    "method": "ApierV1.SetActions",
    "params": [{
        "ActionsId": "Action_au-data-addon-20gb",
        "Overwrite": True,
        "Tenant": str(tenant),
        "Actions": [
            # Restablecer la cuenta primero
            {
                "Identifier": "*reset_account",
                "Weight": 700
            },
            # Agregar 20GB de datos
            # Los saldos de datos utilizan destinos PLMN (en qué
red está el cliente)
            {
                "Identifier": "*topup_reset",
                "BalanceId": "AU_Data_Domestic__" + str(20 * 1024
* 1024 * 1024),
                "BalanceType": "*data",
                "DestinationIDs": "Dest_PLMN_OnNet",  # Tu PLMN en
red (mcc505.mnc057)
                "Units": 20 * 1024 * 1024 * 1024,
                "ExpiryTime": "*month",
                "BalanceWeight": 1200,
                "Weight": 90
            }
        ]
    }]
}

result = OCS_Obj.SendData(Action_AU_Data_Addon_20GB)
assert result['error'] is None or result['error'] == "EXISTS"
print("Acción creada: Action_au-data-addon-20gb")
```

Ejemplo 3: Complemento de Voz Internacional (Python)

Un complemento para minutos de llamadas internacionales.

```
Action_AU_International_Voice_100min = {
    "id": "0",
    "method": "ApierV1.SetActions",
    "params": [{
        "ActionsId": "Action_au-international-voice-100min",
        "Overwrite": True,
        "Tenant": str(tenant),
        "Actions": [
            {
                "Identifier": "*reset_account",
                "Weight": 700
            },
            # Agregar 100 minutos para llamadas internacionales
            {
                "Identifier": "*topup_reset",
                "BalanceId": "AU_Voice_International__" + str(100
* 60 * 1000000000),
                "BalanceType": "*voice",
                "DestinationIDs": "Dest_International_All",
                "Units": 100 * 60 * 1000000000,
                "ExpiryTime": "*month",
                "BalanceWeight": 1000,
                "Weight": 90
            }
        ]
    }]
}

result = OCS_Obj.SendData(Action_AU_International_Voice_100min)
assert result['error'] is None or result['error'] == "EXISTS"
print("Acción creada: Action_au-international-voice-100min")
```

Nota: La voz/SMS utiliza destinos geográficos (número que se llama), mientras que los datos utilizan destinos PLMN (red a la que está conectado el cliente). Consulta Definición de Productos para la configuración de destinos.

Referencia de Campos de Acción en Python

Campos de Definición de Acción:

- **ActionsId** (requerido) - Identificador único para este conjunto de acciones (debe coincidir con la convención product_slug en CRM)
- **Overwrite** - Si es verdadero, reemplaza la acción existente con el mismo ID
- **Tenant** - Nombre del inquilino de CGRateS
- **Actions** - Array de acciones individuales a ejecutar

Campos de Acción Individual:

- **Identifier** - Tipo de acción (`*topup`, `*topup_reset`, `*reset_account`, `*cdrlog`, etc.)
- **BalanceId** - Identificador único para este saldo (debe coincidir en los recargas para que la acumulación funcione)
- **BalanceType** - Tipo de saldo (`*data`, `*voice`, `*sms`, `*monetary`)
- **DestinationIDs** - Controla DÓNDE se puede usar el saldo:
  - Para **voz/SMS**: Usa destinos geográficos (por ejemplo, `"Dest_AU_Mobile"`, `"Dest_International_UK"`)
  - Para **datos**: Usa destinos PLMN (por ejemplo, `"Dest_PLMN_OnNet"`, `"Dest_PLMN_US_Verizon"`)
- **Units** - Cantidad a agregar (bytes para datos, nanosegundos para voz, cantidad para SMS)
- **ExpiryTime** - Cuándo expira el saldo (`*month`, `+720h`, `2024-12-31`, etc.)
- **BalanceWeight** - Prioridad de consumo (más alto = consumido primero)
- **Weight** - Orden de ejecución dentro del conjunto de acciones (más alto = se ejecuta primero)
- **Blocker** (opcional) - Bandera booleana; si es `True`, bloquea todo uso cuando este saldo llega a cero (ver Bloqueadores de Saldo)
- **Disabled** (opcional) - Bandera booleana; si es `True`, este saldo es ignorado/saltado durante el consumo

Ejecución de Acciones

Una vez que se crea una Acción, ejecútala en una cuenta:

```
- name: Ejecutar Acción en Cuenta
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "APIerSv1.ExecuteAction",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "ActionsId": "Action_Name_Here"
        }]
      }
```

Esto ejecuta todas las operaciones definidas en la Acción en la cuenta especificada.

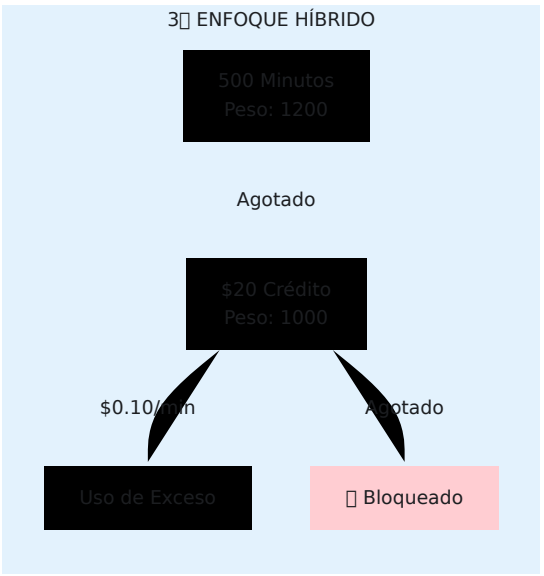
Enfoques de Gestión de Saldos

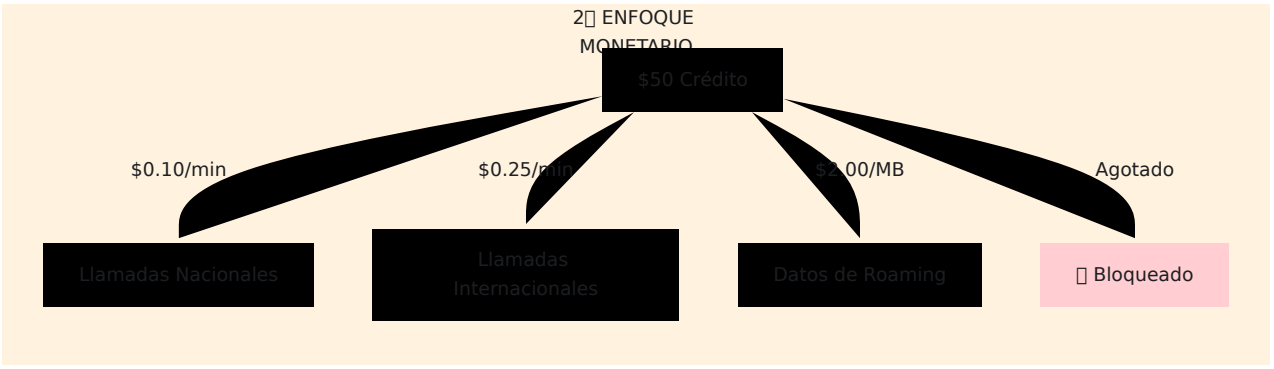
Hay tres enfoques principales para gestionar saldos en CGRateS, cada uno con diferentes compensaciones para la experiencia del cliente, la previsibilidad de la facturación y la complejidad operativa.

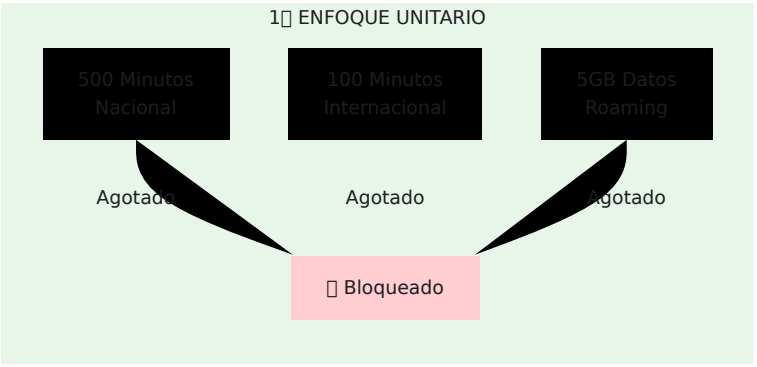
Comparación: Unitario vs Monetario vs Híbrido

Esta tabla resume las compensaciones entre los tres enfoques de saldo:

Característica	Unitario	Monetario (PAYG)	Híbrido
Previsibilidad del Cliente	□ Costo mensual fijo	□ Costos variables	⚠ Mayormente predecible
Flexibilidad de Destino	□ Limitado a destinos incluidos	□ Llamar/utilizar en cualquier lugar	□ Incluido + en cualquier lugar
Manejo de Excesos	□ Corte duro	□ Uso automático	□ Desbordamiento automático
Riesgo de Shock de Factura	□ Bajo (límites duros)	□ Alto (facturación ilimitada)	⚠ Moderado (desbordamiento limitado)
Complejidad de Configuración	⚠ Moderada (muchos saldos)	□ Simple (un saldo)	□ Complejo (ambos)
Optimización de Ingresos	⚠ Menor ARPU	□ Mayor ARPU de usuarios intensivos	□ ARPU equilibrado
Satisfacción del Cliente	□ Alta (sin sorpresas)	□ Baja (shock de factura)	□ Alta (lo mejor de ambos)
Mejor Para	Usuarios predecibles	Usuarios ocasionales	La mayoría de los clientes







Enfoque 1: Saldos Unitarios

Concepto: Proporcionar cantidades específicas para destinos específicos. Cada saldo tiene una cantidad fija (minutos, GB, conteo de SMS) vinculada a destinos específicos. Cuando se agota, el uso se bloquea a menos que haya un respaldo monetario.

Ejemplo: Plan Nacional con Múltiples Saldos

```
Action_Domestic_Plan = {
    "id": "0",
    "method": "ApierV1.SetActions",
    "params": [{
        "ActionsId": "Action_domestic-plan",
        "Overwrite": True,
        "Tenant": tenant,
        "Actions": [
            {
                "Identifier": "*reset_account",
                "Weight": 700
            },
            # 500 minutos de llamadas nacionales
            {
                "Identifier": "*topup_reset",
                "BalanceId": f"Domestic_Voice__{500 * 60 *
1000000000}",
                "BalanceType": "*voice",
                "DestinationIDs": "Dest_Domestic_All",  # SOLO
nacional
                "Units": 500 * 60 * 1000000000,  # 500 minutos en
nanosegundos
                "ExpiryTime": "*month",
                "BalanceWeight": 1200,
                "Weight": 90
            },
            # 1000 SMS nacionales
            {
                "Identifier": "*topup_reset",
                "BalanceId": "Domestic_SMS__1000",
                "BalanceType": "*sms",
                "DestinationIDs": "Dest_Domestic_All",
                "Units": 1000,
                "ExpiryTime": "*month",
                "BalanceWeight": 1200,
                "Weight": 89
            },
            # 10GB de datos nacionales
            {
                "Identifier": "*topup_reset",
                "BalanceId": f"Domestic_Data__{10 * 1024 * 1024 *
1024}",
                "BalanceType": "*data",
```

```
                "DestinationIDs": "Dest_PLMN_OnNet",  # Tu PLMN en
red
                "Units": 10 * 1024 * 1024 * 1024,
                "ExpiryTime": "*month",
                "BalanceWeight": 1200,
                "Weight": 88
            },
            {
                "Identifier": "*cdrlog",
                "BalanceType": "*generic",
                "ExtraParameters": "
{\"Category\":\"^activation\",\"Destination\":\"Plan Nacional\"}",
                "Weight": 80
            }
        ]
    }]
}

result = OCS_Obj.SendData(Action_Domestic_Plan)
assert result['error'] is None or result['error'] == "EXISTS"
```

Cómo Funciona Esto:

- Llamadas nacionales (1-555-1234) → Utiliza saldo de 500 minutos
- Llamadas internacionales (44-20-xxx) → NO hay saldo disponible, bloqueado O utiliza saldo monetario si está disponible
- SMS nacionales → Utiliza saldo de 1000 SMS
- Uso de datos en casa → Utiliza saldo de 10GB
- Datos de roaming → NO hay saldo disponible (necesitaría saldo de datos de roaming separado)

Pros:

- Costos predecibles para los clientes
- Sin shock de factura
- Límites claros

Contras:

- Menos flexible - no se puede utilizar el servicio fuera de los destinos incluidos
- Requiere múltiples saldos para diferentes casos de uso
- El cliente puede sentirse restringido

Enfoque 2: Monetario (PAYG)

Concepto: Proporcionar crédito monetario cobrado a tarifas específicas de destino. Un solo saldo monetario utilizado para TODOS los tipos de uso. CGRateS busca la tarifa para cada destino y deduce el costo del crédito.

Nota: PAYG requiere definir Perfiles de Tarifas para cada destino para establecer el monto en dólares por unidad. Consulta Perfiles de Tarifas para Saldos PAYG/Monetarios para la configuración completa.

Ejemplo: Crédito PAYG de $50

```
Action_PAYG_Credit = {
    "id": "0",
    "method": "ApierV1.SetActions",
    "params": [{
        "ActionsId": "Action_payg-50-credit",
        "Overwrite": True,
        "Tenant": tenant,
        "Actions": [
            # Saldo monetario de $50
            {
                "Identifier": "*topup",
                "BalanceId": "PAYG_Monetary_Balance",
                "BalanceType": "*monetary",
                "DestinationIDs": "*any",  # Funciona para
CUALQUIER destino
                "Units": 5000,  # $50.00 (en centavos)
                "ExpiryTime": "+2160h",  # 90 días
                "BalanceWeight": 1000,  # Menor que los saldos
unitarios
                "Weight": 90
            },
            {
                "Identifier": "*cdrlog",
                "BalanceType": "*generic",
                "ExtraParameters": "
{\"Category\":\"^activation\",\"Destination\":\"$50 Crédito
PAYG\"}",
                "Weight": 80
            }
        ]
    }]
}

result = OCS_Obj.SendData(Action_PAYG_Credit)
```

Cómo Funciona PAYG:

Escenario 1: Llamada nacional, 10 minutos

- Tarifa: $0.10/min
- Cargo: 10 × $0.10 = $1.00

- Restante: $49.00

Escenario 2: Llamada al Reino Unido, 5 minutos

- Tarifa: $0.25/min + $0.05 de tarifa de conexión
- Cargo: (5 × $0.25) + $0.05 = $1.30
- Restante: $47.70

Escenario 3: Roaming en Verizon, 100MB de datos

- Tarifa: $2.00/MB
- Cargo: 100 × $2.00 = $200.00
- Resultado: Fondos insuficientes → Sesión bloqueada a ~$47 de valor (~23MB)

Pros:

- Un saldo para todo - muy flexible
- El cliente puede utilizar el servicio en cualquier lugar donde se definan tarifas
- Configuración simple

Contras:

- Costos impredecibles para los clientes
- Riesgo de shock de factura (especialmente en roaming)
- Puede ser costoso para usuarios intensivos

Enfoque 3: Híbrido (Lo Mejor de Ambos)

Concepto: Combinar saldos unitarios con respaldo monetario. Utilizar saldos unitarios de alto peso para el uso incluido, con un saldo monetario de bajo peso como desbordamiento. La mejor experiencia para el cliente con un costo base predecible y sobrecargas flexibles.

Ejemplo: Plan Híbrido Flexible

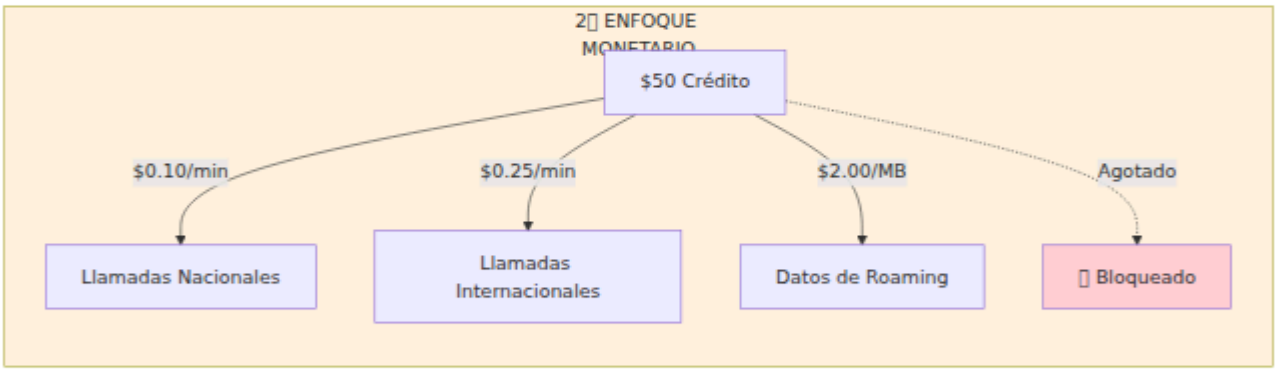
```
Action_Hybrid_Plan = {
    "id": "0",
    "method": "ApierV1.SetActions",
    "params": [{
        "ActionsId": "Action_hybrid-flex-plan",
        "Overwrite": True,
        "Tenant": tenant,
        "Actions": [
            {
                "Identifier": "*reset_account",
                "Weight": 700
            },

            # ============= SALDOS UNITARIOS (Incluidos)
=============
            # 500 minutos de voz nacional
            {
                "Identifier": "*topup_reset",
                "BalanceId": f"Domestic_Voice__{500 * 60 *
1000000000}",
                "BalanceType": "*voice",
                "DestinationIDs": "Dest_Domestic_All",
                "Units": 500 * 60 * 1000000000,
                "ExpiryTime": "*month",
                "BalanceWeight": 1200,  # Consumido PRIMERO para
llamadas nacionales
                "Weight": 95
            },
            # 100 minutos de voz internacional
            {
                "Identifier": "*topup_reset",
                "BalanceId": f"International_Voice__{100 * 60 *
1000000000}",
                "BalanceType": "*voice",
                "DestinationIDs": "Dest_International_All",
                "Units": 100 * 60 * 1000000000,
                "ExpiryTime": "*month",
                "BalanceWeight": 1150,  # Consumido PRIMERO para
internacionales
                "Weight": 94
            },
            # 1000 SMS nacionales
            {
```

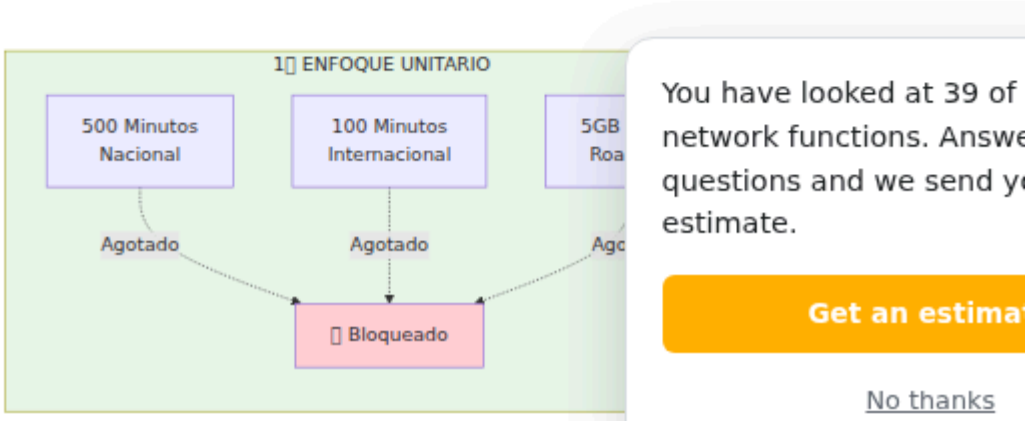
```
                "Identifier": "*topup_reset",
                "BalanceId": "Domestic_SMS__1000",
                "BalanceType": "*sms",
                "DestinationIDs": "Dest_Domestic_All",
                "Units": 1000,
                "ExpiryTime": "*month",
                "BalanceWeight": 1200,
                "Weight": 93
            },
            # 15GB de datos nacionales
            {
                "Identifier": "*topup_reset",
                "BalanceId": f"Domestic_Data__{15 * 1024 * 1024 *
1024}",
                "BalanceType": "*data",
                "DestinationIDs": "Dest_PLMN_OnNet",
                "Units": 15 * 1024 * 1024 * 1024,
                "ExpiryTime": "*month",
                "BalanceWeight": 1200,
                "Weight": 92
            },
            # 2GB de datos de roaming (Zona 1)
            {
                "Identifier": "*topup_reset",
                "BalanceId": f"Roaming_Zone1_Data__{2 * 1024 *
1024 * 1024}",
                "BalanceType": "*data",
                "DestinationIDs": "Dest_PLMN_Zone_NorthAmerica",
                "Units": 2 * 1024 * 1024 * 1024,
                "ExpiryTime": "*month",
                "BalanceWeight": 1100,
                "Weight": 91
            },

            # ============= SALDO MONETARIO (Desbordamiento/PAYG)
=============
            # $20 para sobrecargas
            {
                "Identifier": "*topup",
                "BalanceId": "PAYG_Overflow_Balance",
                "BalanceType": "*monetary",
                "DestinationIDs": "*any",
                "Units": 2000,  # $20.00
                "ExpiryTime": "*month",
```

```
                "BalanceWeight": 1000,  # Consumido ÚLTIMO
(respaldo)
                "Weight": 90
            },

            {
                "Identifier": "*cdrlog",
                "BalanceType": "*generic",
                "ExtraParameters": "
{\"Category\":\"^activation\",\"Destination\":\"Plan Híbrido
Flexible\"}",
                "Weight": 80
            }
        ]
    }]
}

result = OCS_Obj.SendData(Action_Hybrid_Plan)
```

Flujo de Consumo de Balance:







You have looked at 39 of
network functions. Answe
questions and we send y
estimate.
Get an estima
No thanks

Ejemplos de Escenarios:

Escenario 1: 600 minutos nacionales utilizados

1. Primeros 500 minutos → Usa "Domestic_Voice_500min" (peso 1200) - GRATIS
2. Siguientes 100 minutos → "Domestic_Voice_500min" agotado, recurre a "PAYG_Overflow_Balance" (peso 1000)
3. Cargado: 100 min × $0.10 = $10.00 del balance monetario

4. Restante: $10.00 monetarios

Escenario 2: 150 minutos internacionales (a Reino Unido)

1. Primeros 100 minutos → Usa "International_Voice_100min" (peso 1150) - GRATIS
2. Siguientes 50 minutos → Recurre a "PAYG_Overflow_Balance"
3. Cargado: (50 × $0.25) + $0.05 conexión = $12.55
4. Restante: $7.45 monetarios (si comienza con $20)

Ventajas:

- Costo base predecible
- Flexibilidad para sobrecargas ocasionales
- Mejor experiencia para el cliente
- Sin corte abrupto - el servicio continúa

Desventajas:

- Más complejo de configurar
- Más complejo de explicar a los clientes
- Requiere una gestión cuidadosa del peso del balance

Escenarios de Uso en el Mundo Real

Estos escenarios demuestran cómo los conceptos se unen en la práctica. Cada uno muestra el flujo completo desde el evento de uso hasta la deducción o bloqueo del balance.

Escenario 1: Cliente en Casa Llama al Reino Unido

El cliente tiene: Plan Nacional (500 min nacionales, 1000 SMS, 10GB de datos)

Acción: Llama a un número del Reino Unido (44-20-7946-0958) durante 10 minutos

Verificación de Balance:

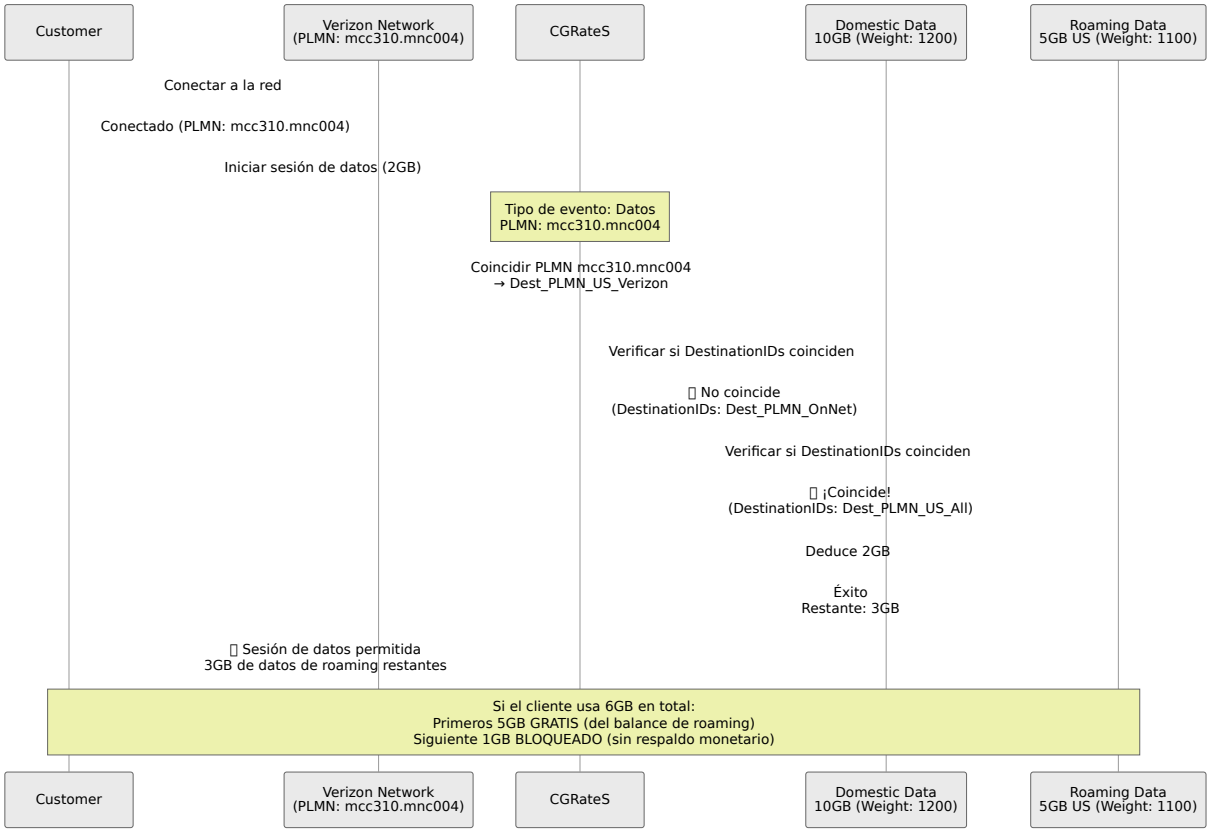
1. CGRateS recibe llamada a `44207946...`
2. Coincide destino: `Dest_International_UK`
3. Verifica balance con `DestinationIDs: "Dest_International_UK"`
4. NO se encontró balance coincidente
5. Verifica balance monetario
6. NO se encontró balance monetario
7. **Resultado:** Llamada BLOQUEADA (saldo insuficiente)

Solución: El cliente necesita un Plan Internacional O crédito PAYG para realizar llamadas al Reino Unido

Escenario 2: Cliente en Roaming en EE. UU. Usa Datos

El cliente tiene: Plan Nacional + complemento de Roaming en EE. UU. de 5GB

Acción: Roaming en Verizon (PLMN mcc310.mnc004), usa 2GB de datos



Verificación de Balance:

1. CGRateS recibe sesión de datos en PLMN `mcc310.mnc004`
2. Coincide destino: `Dest_PLMN_US_Verizon`
3. Verifica balance de datos con `DestinationIDs: "Dest_PLMN_US_All"` (coincidencia más amplia)
4. Encuentra: balance "Roaming_US_Data_5GB" (BalanceWeight: 1100)
5. Deduce 2GB
6. **Restante:** 3GB de datos de roaming

¿Qué pasa si usaron 6GB?

1. Primeros 5GB → Usa balance de roaming (agotado)
2. Siguiente 1GB → Verifica balance monetario
3. NO hay balance monetario → **Sesión bloqueada** en 5GB

Escenario 3: Cliente con Plan Híbrido Excede Minutos Nacionales

El cliente tiene: Plan Híbrido (500 min nacionales + $20 de sobrecarga)

Acción: Realiza 600 minutos de llamadas nacionales

Verificación de Balance:

1. Primeros 500 minutos:
    - Usa "Domestic_Voice_500min" (BalanceWeight: 1200)
    - Sin cargo
2. "Domestic_Voice_500min" agotado
3. Siguientes 100 minutos:
    - Recurre a "PAYG_Overflow_Balance" (BalanceWeight: 1000)
    - Tarifa: $0.10/min
    - Cargo: 100 × $0.10 = $10.00
    - Deducido de $20 de balance monetario
    - **Restante:** $10.00

Facturación:

- El cliente ve: 500 minutos incluidos + 100 minutos de sobrecarga ($10.00)

Escenario 4: Usuario de PAYG Roaming Inesperadamente

El cliente tiene: $50 de crédito PAYG

Acción: Viaja a EE. UU., hace roaming en Verizon, usa 100MB de datos (sin saberlo)

Cálculo de Cargo:

1. 100MB en Verizon
2. Tarifa: $2.00/MB (de tarifas PAYG)
3. Cargo total: 100 × $2.00 = $200.00
4. Disponibles: $50.00
5. **Resultado:** El cliente usa ~25MB antes de que el crédito se agote, sesión bloqueada
6. **¡Shock de factura!** El cliente consumió inesperadamente todo el $50

Mejor enfoque: Recomendar complemento de roaming para evitar shock de factura

Mejores Prácticas

1. Usar IDs de Balance Descriptivos

Buenos IDs de Balance son auto-documentados:

```
# Bueno
"Data_5GB_5days_{{ uuid }}"
"Voice_100min_Monthly"
"Bonus_Data_Loyalty"

# Malo
"balance1"
"data"
"temp"
```

2. Documentar Tu Estrategia de Peso

Definir un esquema de peso consistente en todos los productos:

```
# Esquema de peso:
# 30 = Balances Premium/Promocionales
# 20 = Balances de Bonificación/Loyalty
# 10 = Balances Regulares/Comprados
# 5 = Balances de Respaldo/Fallback
```

3. Incluir Registro de CDR

Siempre registra adiciones de balance para auditorías:

```
{
  "Identifier": "*cdrlog",
  "BalanceType": "*generic",
  "ExtraParameters": "
{\"Category\":\"^activation\",\"Destination\":\"{{ package_name
}}\"}"
}
```

4. Usar ActionPlans para Operaciones Recurrentes

Para reinicios mensuales, no ejecutes la Acción manualmente. Usa ActionPlans:

```
# Crear ActionPlan que se ejecute mensualmente
- name: Crear ActionPlan mensual
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.SetActionPlan",
        "params": [{
          "Id": "ActionPlan_Monthly_Reset",
          "ActionPlan": [{
            "ActionsId": "Action_Monthly_Reset",
            "Time": "*monthly",
            "Weight": 10
          }],
          "ReloadScheduler": true
        }]
      }
```

5. Restablecer ActionTriggers Después de Recarga

Después de ejecutar una acción, restablece los triggers para evitar notificaciones duplicadas:

```
- name: Restablecer ActionTriggers
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "APIerSv1.ResetAccountActionTriggers",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "Executed": false
        }]
      }
```

6. Establecer Pesos de Balance Apropiados

Usar una estrategia de peso consistente en todos los productos:

```
# Prioridad de consumo (de mayor a menor)
BalanceWeight: 1200  # Nacional incluido (usar primero en casa)
BalanceWeight: 1150  # Internacional incluido (usar primero para
intl)
BalanceWeight: 1100  # Roaming incluido (usar primero al hacer
roaming)
BalanceWeight: 1000  # Respaldo monetario (usar al final)
```

Esto asegura un orden de consumo predecible: balances nacionales primero, luego internacionales, luego roaming, y finalmente PAYG monetario como respaldo.

7. Controlar el Uso del Balance con DestinationIDs

Siempre usa DestinationIDs específicos para evitar usos no intencionados:

```
# Bueno - control de destino explícito
{
    "BalanceType": "*voice",
    "DestinationIDs": "Dest_International_UK",  # SOLO Reino Unido
    "Units": 200 * 60 * 1000000000
}

# Malo - uso no intencionado
{
    "BalanceType": "*voice",
    "DestinationIDs": "*any",  # ¡Podría usarse para números
premium!
    "Units": 200 * 60 * 1000000000
}
```

8. Agrupar Destinos Lógicamente

Crear grupos de destinos que tengan sentido para tu oferta de productos:

```
# Bueno - agrupación lógica
"Dest_International_Europe" = Todos los países de la UE (pool
compartido de 500min)

# Malo - demasiado granular
"Dest_Francia", "Dest_Alemania", "Dest_Italia" (pools pequeños
separados)
```

Consulta CGRateS Destinations para ejemplos de agrupación de destinos.

9. Incluir Respaldo Monetario en los Planes

Para la mejor experiencia del cliente, incluye un pequeño balance monetario como sobrecarga:

```
# Recomendado: Siempre incluir pequeño balance monetario
{
    "BalanceType": "*monetary",
    "Units": 1000,  # $10 de protección contra sobrecargas
    "BalanceWeight": 1000  # Menor que los balances unitarios
}
```

Esto previene cortes de servicio abruptos y proporciona flexibilidad para sobrecargas ocasionales.

10. Usar Incrementos de Tarifa Estratégicamente

Elegir incrementos de tarifa apropiados para diferentes tipos de servicio:

```
# Nacional: Amigable para el cliente por segundo
"RateIncrement": "1s"

# Internacional: Bloques estándar de 6 segundos
"RateIncrement": "6s"

# Premium: Bloques de 30 segundos para desalentar abusos
"RateIncrement": "30s"

# Datos: Por KB para precisión
"RateIncrement": "1024"
```

11. Establecer Tiempos de Expiración Razonables

Ajustar los tiempos de expiración a los tipos de productos:

```
# Planes de suscripción mensual
"ExpiryTime": "*month"

# Recargas/complentos únicos
"ExpiryTime": "+720h"  # 30 días

# Crédito PAYG (validez más larga)
"ExpiryTime": "+2160h"  # 90 días

# Complementos de roaming (basados en el viaje)
"ExpiryTime": "+360h"  # 15 días
```

Solución de Problemas

Problema: Balance No Añadido

Síntomas: La acción se ejecuta con éxito pero el balance no aparece

Causas Posibles:

- UUID de cuenta incorrecto
- Desajuste de BalanceType
- La expiración ya ha pasado

Solución: Verifica que la cuenta exista y revisa el tiempo de expiración del balance

Problema: Balance Consumido Incorrectamente

Síntomas: El sistema consume de un balance inesperado

Causas Posibles:

- Configuración de peso incorrecta
- Múltiples balances con el mismo ID

Solución: Revisa los valores de peso y asegúrate de que los IDs de Balance sean únicos si se desea independencia

Problema: Rollover No Funciona

Síntomas: Usando `*topup` pero el balance antiguo no se transfiere

Causas Posibles:

- Usando diferentes IDs de Balance (necesita el mismo ID)
- Acción usando `*topup_reset` en lugar de `*topup`

Solución: Verifica la consistencia del ID de Balance y el tipo de acción

Problema: Balance Expira Inmediatamente

Síntomas: Balance añadido pero muestra como expirado

Causas Posibles:

- ExpiryTime en el pasado
- Usando timestamp absoluto en lugar de relativo

Solución: Usa expiración relativa (`+Xh`) en lugar de timestamps absolutos

Problema: Balance No Está Siendo Consumido

Síntoma: El cliente tiene balance pero aún bloqueado o cargado PAYG

Causas Posibles:

1. Desajuste de DestinationIDs - Los destinos de balance no coinciden con el destino de uso
2. Balance expirado
3. Tipo de balance incorrecto para el uso

Pasos de Depuración:

```
# Verifica los balances de la cuenta
OCS_Obj.SendData({
    'method': 'ApierV2.GetAccount',
    'params': [{"Tenant": tenant, "Account": "service_uuid"}]
})

# Verifica qué se devuelve:
# - ¿El balance existe?
# - ¿DestinationIDs correctos?
# - ¿ExpiryTime en el futuro?
# - ¿Units > 0?
```

Solución: Verifica que el destino coincida y que el balance esté activo. Consulta Escenario 1: Cliente en Casa Llama al Reino Unido para un ejemplo.

Problema: Tarifa Incorrecta Aplicada

Síntoma: Cliente cargado con un monto incorrecto por uso PAYG

Causas Posibles:

1. Superposición de destinos (precedencia incorrecta en RatingPlan)
2. Pesos de enlace del plan de tarifas incorrectos
3. La definición de tarifa tiene valores incorrectos

Solución:

- Asegúrate de que los destinos específicos tengan mayor peso en los enlaces del RatingPlan
- Verifica que la coincidencia de prefijo más largo esté funcionando correctamente
- Verifica que los valores y los incrementos de tarifas sean correctos

Consulta Perfiles de Tarifas para Balances PAYG/Monetarios para una configuración adecuada.

Problema: Balance de Roaming No Usado

Síntoma: Cliente en roaming, tiene balance de roaming, pero cargado PAYG o bloqueado

Causas Posibles:

1. Desajuste de destino PLMN - El balance no incluye el PLMN visitado
2. Cliente conectado a una red incorrecta/inesperada
3. DestinationIDs no coinciden con el PLMN real

Solución:

```
# Verifica en qué PLMN está el cliente (de CDRs o eventos de uso)
# Verifica que el balance incluya ese PLMN:
"DestinationIDs": "Dest_PLMN_US_All"  # Debería incluir mcc310.mnc004
```

Consulta CGRateS Destinations para la configuración de destinos PLMN y Escenario 2: Cliente en Roaming en EE. UU. Usa Datos para un ejemplo funcional.

Problema: Plan Híbrido No Recurre a Monetario

Síntoma: Balance unitario agotado pero no se usa el balance monetario

Causas Posibles:

1. El balance monetario tiene restricción de DestinationIDs (debería ser `*any`)
2. No se definieron tarifas PAYG para ese destino
3. Problema de peso de balance - el peso del balance monetario no es menor que el unitario

Solución:

- Asegúrate de que el balance monetario tenga `DestinationIDs: "*any"`

- Verifica que se definan tarifas PAYG para todos los destinos que el cliente podría usar
- Revisa los pesos de balance: el monetario debe ser el más bajo (por ejemplo, 1000)

Consulta Enfoque 3: Híbrido para una configuración híbrida adecuada.

Problema: Cargos Internacionales Inesperados

Síntoma: Cliente llamó a un número nacional, cargado con tarifas internacionales

Causas Posibles:

1. El número es en realidad internacional (por ejemplo, número de Canadá +1 tratado como internacional)
2. Superposición de prefijos en las definiciones de destino
3. Cliente marcó incorrectamente (agregó dígitos extra)

Solución:

- Verifica los CDRs para el número marcado real
- Verifica que las definiciones de prefijo de destino no se superpongan incorrectamente
- Considera un destino separado para Canadá si los países NANP necesitan tarifas diferentes

Consulta CGRateS Destinations para la configuración de destinos geográficos.

Documentación Relacionada

Conceptos Básicos

- Definición de Productos - Flujo de trabajo completo para crear productos, definir Acciones de CGRateS y enlazarlos
- CGRateS Destinations - Cómo definir destinos geográficos y PLMN para servicios de voz, SMS y datos

Implementación y Operaciones

- Cargos y Pagos desde Playbooks - Flujo de pago de compromiso en dos fases, cálculos prorrateados y cargos a clientes
- Guía de Playbooks de Ansible - Estructura de playbooks, flujos de aprovisionamiento y desaprovisionamiento con bloques de rescate
- Descripción General del Sistema de Aprovisionamiento - Cómo funciona el aprovisionamiento en OmniCRM

Configuración de Destinos de CGRateS

Esta guía explica cómo configurar destinos en CGRateS para OmniCRM. Los destinos definen DÓNDE se pueden usar los saldos, ya sea llamando a números específicos (geográficos) o utilizando datos/voces en redes específicas (PLMN).

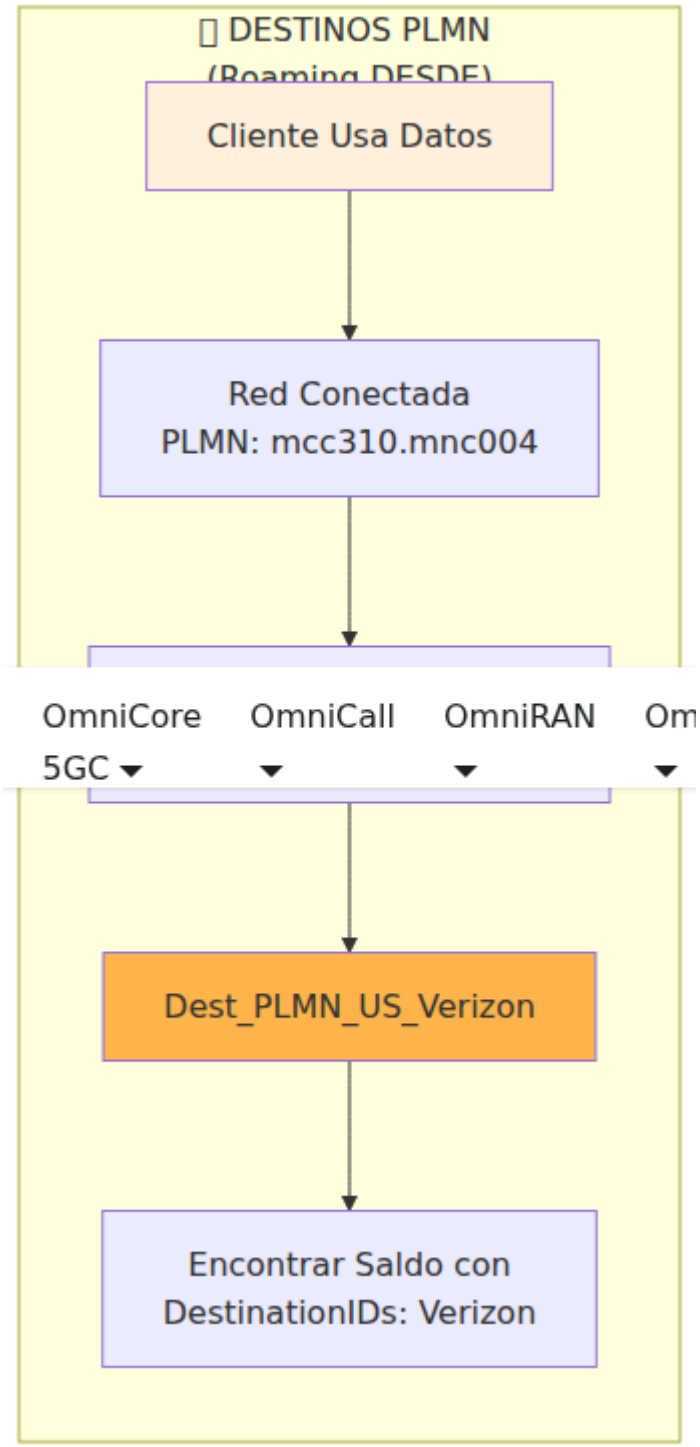
Descripción general

CGRateS utiliza **destinos** para controlar dónde los clientes pueden usar sus saldos:

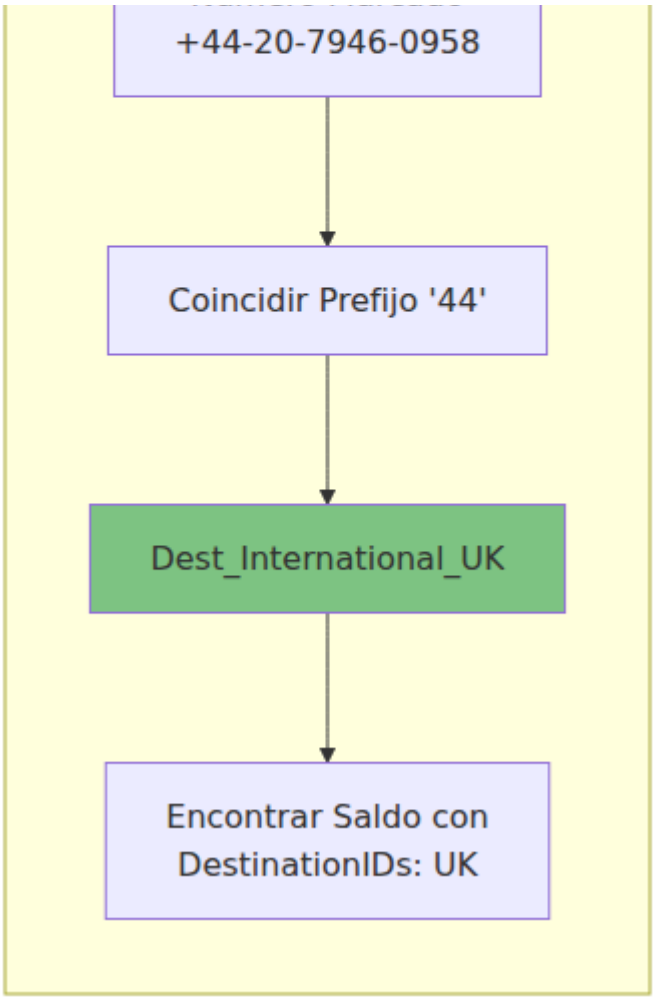
- **Destinos Geográficos** - Prefijos de números para llamadas/SMS A ubicaciones específicas (por ejemplo, números del Reino Unido, números gratuitos de EE. UU.)
- **Destinos PLMN** - Códigos de red para uso de datos y roaming DESDE redes específicas (por ejemplo, Verizon, Vodafone UK)

Cuando los clientes utilizan servicios, CGRateS hace coincidir:

- **Voz/SMS**: El prefijo del número marcado → Destino geográfico
- **Datos**: El código PLMN de la red → Destino PLMN


DESTINOS PLMN
(Roaming DESDE)
Cliente Usa Datos
Red Conectada
PLMN: mcc310.mnc004
OmniCore
5GC
OmniCall
OmniRAN
Om
Dest_PLMN_US_Verizon
Encontrar Saldo con
DestinationIDs: Verizon


DESTINOS
GEOGRÁFICOS
Cliente Realiza Llamada
Número Marcado



Regla Crítica

- **Saldos de Voz/SMS** → Usar destinos geográficos (el número que se llama A)
- **Saldos de Datos** → Usar destinos PLMN (la red a la que está conectado el cliente DESDE)

Requisitos previos

```
import cgrateshttpapi
import time

OCS_Obj = cgrateshttpapi.CGRateS("ocs.example.com", "2080")
tenant = "tu_nombre_de_tenant"
tpid = str(tenant) + "_" + str(int(time.time()))
```

Parte 1: Destinos Geográficos (Llamando A Lugares)

Los destinos geográficos definen prefijos de números para llamadas de voz y SMS a ubicaciones específicas.

Destinos Nacionales

```
#
# ============================================================================
# DESTINOS NACIONALES (EE. UU./Canadá - NANP)
#
# ============================================================================

# Móviles y Fijos de EE. UU./Canadá (comparten códigos de área)
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_Domestic_All",
        "Prefixes": ["1"]  # NANP (EE. UU./Canadá)
    }]
})

# Números Gratuitos de EE. UU.
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_Domestic_TollFree",
        "Prefixes": ["1800", "1888", "1877", "1866", "1855",
"1844", "1833"]
    }]
})

# Tarifa Premium de EE. UU.
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_Domestic_Premium",
        "Prefixes": ["1900"]
    }]
})
```

Destinos Internacionales

```
#
# ============================================================================
# DESTINOS INTERNACIONALES (Por País)
#
# ============================================================================

# Reino Unido
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_International_UK",
        "Prefixes": ["44"]
    }]
})

# China
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_International_China",
        "Prefixes": ["86"]
    }]
})

# Australia - Móvil
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_AU_Mobile",
        "Prefixes": ["614"]  # Móviles australianos (04xx marcados
como 614xx)
    }]
})

# Australia - Línea Fija
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
```

```
        "TPid": tpid,
        "ID": "Dest_AU_Fixed",
        "Prefixes": [
            "612",  # NSW (Sídney)
            "613",  # VIC (Melbourne)
            "617",  # TAS (Hobart)
            "618"   # SA, WA, NT (Adelaida, Perth, Darwin)
        ]
    }]
})

# Australia - Números Gratuitos
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_AU_TollFree",
        "Prefixes": [
            "611800",  # Sin cargo
            "611300"   # Tarifa local
        ]
    }]
})

# Australia - Tarifa Premium
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_AU_Premium",
        "Prefixes": ["6119"]  # Servicios de tarifa premium
    }]
})

# Europa (agrupados)
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_International_Europe",
        "Prefixes": [
            "33",  # Francia
            "49",  # Alemania
            "39",  # Italia
```

```
            "34",  # España
            "31",  # Países Bajos
            "32",  # Bélgica
            "41"   # Suiza
        ]
    }]
})

# Internacional - Todos (captura todo, excluye nacional)
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_International_All",
        "Prefixes": ["2", "3", "4", "5", "6", "7", "8", "9"]  # Todos excepto "1" (NANP)
    }]
})
```

Casos de Uso:

- **Planes nacionales**: "Llamadas ilimitadas a números de EE. UU./Canadá" → `Dest_Domestic_All`
- **Complemento de llamadas al Reino Unido**: "100 minutos al Reino Unido" → `Dest_International_UK`
- **Plan australiano**: "3000 minutos a móviles y líneas fijas australianas" → `Dest_AU_Mobile`, `Dest_AU_Fixed`, `Dest_AU_TollFree`
- **Paquete internacional**: "50 minutos a cualquier número internacional" → `Dest_International_All`

Parte 2: Destinos PLMN (Uso DESDE Lugares - Roaming / Uso de Datos)

Los destinos PLMN definen códigos de red para el uso de datos y escenarios de roaming. Utilice el formato `mccXXX.mncYYY` donde:

- **MCC** = Código de País Móvil (3 dígitos)

- **MNC** = Código de Red Móvil (2-3 dígitos)

Reglas de Formato PLMN

- **Red específica**: `"mcc310.mnc004"` → Solo red Verizon 004
- **Todas las redes en el país**: `"mcc310"` → Cualquier red de EE. UU.
- **Ejemplos**: `mcc310.mnc004`, `mcc234.mnc015`, `mcc505.mnc057`

En Red (Tu Red Local)

IMPORTANTE: Define TU red específica donde se provisionan tus SIM.

```
# En Red (Tu Red Local)
# Este es el PLMN de TU operador - la red donde están activas las SIM de TUS clientes
# Todos los saldos de datos nacionales en red deben usar este destino

OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_PLMN_OnNet",
        "Prefixes": ["mcc505.mnc057"]  # Ejemplo: MCC 505, MNC 57 si esta es tu red local
    }]
})
# Reemplaza mcc505.mnc057 con el código PLMN real de TU operador
# Todos los saldos de datos nacionales en red deben usar los prefijos establecidos aquí
# De lo contrario, se consideran roaming
```

Destinos PLMN de Roaming en EE. UU.

```
#
# ============================================================================
# DESTINOS PLMN DE ROAMING EN EE. UU.
#
# ============================================================================

# Verizon Wireless
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_PLMN_US_Verizon",
        "Prefixes": [
            "mcc310.mnc004", "mcc310.mnc010", "mcc310.mnc012",
"mcc310.mnc013",
            "mcc311.mnc480", "mcc311.mnc481", "mcc311.mnc482",
"mcc311.mnc483"
        ]
    }]
})

# Todos los PLMN de EE. UU. (captura todo)
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_PLMN_US_All",
        "Prefixes": ["mcc310", "mcc311", "mcc312", "mcc313",
"mcc316"]
    }]
})
# Nota: Esto coincide con CUALQUIER red con estos MCCs (310, 311,
etc.)
# Cliente en mcc310.mnc004 (Verizon), mcc310.mnc410 (AT&T),
mcc311.mnc580 (US Cellular)
# coincidiría con este destino
```

Destinos PLMN de Roaming en el Reino Unido

```
#
# ============================================================================
# DESTINOS PLMN DE ROAMING EN EL REINO UNIDO
#
# ============================================================================

# Vodafone UK
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_PLMN_UK_Vodafone",
        "Prefixes": ["mcc234.mnc015"]
    }]
})

# EE (Todo en Todas Partes)
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_PLMN_UK_EE",
        "Prefixes": ["mcc234.mnc030", "mcc234.mnc033"]
    }]
})

# Todos los PLMN del Reino Unido
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_PLMN_UK_All",
        "Prefixes": ["mcc234"]
    }]
})
```

Zonas de Roaming (Grupos Multipaís)

Genial para paquetes de roaming regional como "Roaming en Europa" o "Roaming en Asia-Pacífico".

```
#
================================================================
# ZONAS DE ROAMING (Grupos Multipaís)
#
================================================================

# Zona 1: América del Norte
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_PLMN_Zone_NorthAmerica",
        "Prefixes": [
            "mcc310", "mcc311", "mcc312", "mcc313", "mcc316",  #
EE. UU.
            "mcc302", "mcc334"  # Canadá
        ]
    }]
})
# Usar prefijos solo de MCC proporciona cobertura en CUALQUIER red
en estos países
# Genial para paquetes de "roaming en América del Norte" que
funcionan en todos los operadores

# Zona 2: Europa
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_PLMN_Zone_Europe",
        "Prefixes": [
            "mcc234",  # Reino Unido
            "mcc208",  # Francia
            "mcc262",  # Alemania
            "mcc222",  # Italia
            "mcc214",  # España
            "mcc228"   # Suiza
        ]
    }]
})

# Zona 3: Asia-Pacífico
OCS_Obj.SendData({
```

```
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_PLMN_Zone_AsiaPacific",
        "Prefixes": [
            "mcc505",  # Australia
            "mcc460",  # China
            "mcc454",  # Hong Kong
            "mcc440",  # Japón
            "mcc520",  # Tailandia
            "mcc525",  # Singapur
            "mcc530"   # Nueva Zelanda
        ]
    }]
})

# Captura todo para cualquier roaming (usar con precaución)
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_Roaming_All",
        "Prefixes": ["mcc"]  # Coincide con CUALQUIER PLMN (muy
amplio)
    }]
})
```

Casos de Uso:

- **Roaming específico de red**: "5GB solo en Verizon" → `Dest_PLMN_US_Verizon`
- **Roaming en todo el país**: "10GB en EE. UU. (cualquier red)" → `Dest_PLMN_US_All`
- **Roaming regional**: "15GB de roaming en Europa" → `Dest_PLMN_Zone_Europe`
- **Roaming global**: "1GB en cualquier lugar" → `Dest_Roaming_All`

Cuándo Usar Cada Enfoque

Tipo de Destino	Caso de Uso	Ejemplo	Tipos de Saldo
Geográfico	Voz/SMS a países específicos	"100 minutos a números del Reino Unido"	Voz, SMS
PLMN en red	Datos nacionales en TU red	"100GB de datos en red"	Datos
PLMN específico	Roaming premium en redes preferidas	"5GB solo en Verizon"	Datos (y voz/SMS de roaming)
PLMN de todo el país	Roaming en cualquier red del país	"5GB en EE. UU. (cualquier red)"	Datos (y voz/SMS de roaming)
PLMN de zona	Paquetes de roaming regional	"10GB de roaming en Europa"	Datos (y voz/SMS de roaming)

Actualizaciones Dinámicas de Destinos y Acceso al Servicio

Cómo Funcionan las Actualizaciones de Destinos

Concepto Cr��tico: Los destinos en CGRateS son **referenciados por ID**, no copiados. Cuando actualizas los prefijos de un destino o los códigos PLMN y **cargas el Plan de Tarifas en el tiempo de ejecución**, **todos los saldos**

existentes que hacen referencia a ese destino se ven afectados de inmediato.

Esto significa:

- **No es necesario actualizar saldos individuales** al expandir o restringir la cobertura del servicio
- **Cambios a nivel de red** para todos los clientes que utilizan ese destino una vez que se carga el plan de tarifas
- ⚠ **Los cambios entran en vigor cuando cargas el Plan de Tarifas** - puedes modificar destinos múltiples veces antes de cargar, y los cambios solo se activan después de `LoadTariffPlanFromStorDb`

Ejemplo 1: Expansión de Cobertura

Tienes un destino para roaming europeo:

```
# Configuración inicial - Solo 3 países
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_PLMN_Zone_Europe",
        "Prefixes": [
            "mcc262",  # Alemania
            "mcc208",  # Francia
            "mcc222"   # Italia
        ]
    }]
})
```

Clientes con saldos que hacen referencia a `Dest_PLMN_Zone_Europe` pueden hacer roaming en Alemania, Francia e Italia.

Más tarde, agregas España y el Reino Unido:

```
# Configuración actualizada - Ahora 5 países
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_PLMN_Zone_Europe",
        "Prefixes": [
            "mcc262",  # Alemania
            "mcc208",  # Francia
            "mcc222",  # Italia
            "mcc214",  # España (NUEVO)
            "mcc234"   # Reino Unido (NUEVO)
        ]
    }]
})

# Carga el plan de tarifas para hacer los cambios activos
OCS_Obj.SendData({
    'method': 'ApierV1.LoadTariffPlanFromStorDb',
    'params': [{
        "TPid": tpid,
        "DryRun": False,
        "Validate": True
    }]
})

# Recargar la caché de destinos
OCS_Obj.SendData({
    'method': 'CacheSv1.ReloadCache',
    'params': [{
        "DestinationIDs": ["Dest_PLMN_Zone_Europe"]
    }]
})
```

Resultado: Después de cargar el plan de tarifas, todos los clientes existentes con saldos para `Dest_PLMN_Zone_Europe` obtienen acceso para hacer roaming en España y el Reino Unido - no se requieren actualizaciones de saldo individuales.

Ejemplo 2: Restricción de Cobertura

Eliminar un país de un destino bloquea el acceso una vez que se carga el plan de tarifas:

```
# Eliminar Alemania de la zona de Europa
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_PLMN_Zone_Europe",
        "Prefixes": [
            "mcc208",  # Francia
            "mcc222",  # Italia
            "mcc214",  # España
            "mcc234"   # Reino Unido
            # Alemania (mcc262) ELIMINADO
        ]
    }]
})

# Carga el plan de tarifas para hacer los cambios activos
OCS_Obj.SendData({
    'method': 'ApierV1.LoadTariffPlanFromStorDb',
    'params': [{
        "TPid": tpid,
        "DryRun": False,
        "Validate": True
    }]
})

# Recargar caché
OCS_Obj.SendData({
    'method': 'CacheSv1.ReloadCache',
    'params': [{
        "DestinationIDs": ["Dest_PLMN_Zone_Europe"]
    }]
})
```

Resultado: Después de cargar el plan de tarifas, los clientes que intenten hacer roaming en Alemania serán **bloqueados**, incluso si aún tienen saldo

restante para `Dest_PLMN_Zone_Europe`. El saldo existe pero ya no coincide con el destino PLMN.

Ejemplo 3: Destinos Geográficos de Voz/SMS

El mismo principio se aplica a los destinos geográficos para voz y SMS:

```
# Agregar Canadá a "Llamadas Nacionales"
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_Domestic_All",
        "Prefixes": [
            "1",  # Ya incluye EE. UU. (y anteriormente Canadá)
            # No se necesita cambio - el prefijo "1" ya cubre
ambos
        ]
    }]
})

# O crear un destino separado para América del Norte
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_North_America",
        "Prefixes": [
            "1"  # EE. UU. + Canadá
        ]
    }]
})
```

Impacto: Cualquier saldo con `DestinationIDs: "Dest_Domestic_All"` ahora funciona para llamadas tanto a EE. UU. como a Canadá (si se usa el prefijo "1").

Reglas de Acceso al Servicio

Crítico: Si un cliente intenta usar un servicio (voz/SMS/datos) y **no hay destino que coincida**, el servicio será **denegado**, incluso si tienen saldos.

Escenario 1: Sin Destino Coincidente

```
# El cliente tiene saldo:
{
    "BalanceId": "Europe_Data_5GB",
    "BalanceType": "*data",
    "DestinationIDs": "Dest_PLMN_Zone_Europe",  # Solo Francia,
Alemania, Italia
    "Units": 5 * 1024 * 1024 * 1024
}

# El cliente hace roaming en una red en España (mcc214.mnc001)
# España NO está en Dest_PLMN_Zone_Europe
```

Resultado:

- ❌ La sesión de datos **bloqueada** (sin destino coincidente)
- El saldo permanece sin usar
- El cliente recibe un error de "Crédito Insuficiente" o similar

Solución: Agregar España a `Dest_PLMN_Zone_Europe` O asegurarse de que el cliente tenga un saldo monetario con `DestinationIDs: "*any"` como respaldo.

Escenario 2: Sin Respaldo de Wildcard

```
# El cliente tiene dos saldos:
{
    "BalanceId": "UK_Voice_100min",
    "BalanceType": "*voice",
    "DestinationIDs": "Dest_International_UK",  # Solo prefijo
"44"
    "Units": 100 * 60 * 1000000000
}
# NO hay saldo monetario con destino "*any"

# El cliente llama a Australia (+61)
```

Resultado:

- ❌ La llamada **bloqueada** (sin destino que coincida con el prefijo "61")

- El saldo de minutos del Reino Unido no se usó (solo coincide con "44")
- No hay respaldo monetario para cubrir la llamada

Solución: Agregar un saldo monetario con `DestinationIDs: "*any"` O agregar un destino específico de Australia a su saldo.

Escenario 3: Respaldo de Wildcard Funciona

```
# El cliente tiene:
# Saldo 1: minutos del Reino Unido
{
    "BalanceId": "UK_Voice_100min",
    "BalanceType": "*voice",
    "DestinationIDs": "Dest_International_UK",
    "Units": 100 * 60 * 1000000000,
    "BalanceWeight": 1200
}

# Saldo 2: Monetario con wildcard
{
    "BalanceId": "PAYG_Credit",
    "BalanceType": "*monetary",
    "DestinationIDs": "*any",  # Coincide con CUALQUIER COSA
    "Units": 5000,  # $50
    "BalanceWeight": 1000  # Prioridad más baja
}

# El cliente llama a Australia (+61)
```

Resultado:

- □ La llamada **permitida**
- El saldo de minutos del Reino Unido NO se usó (no coincide con "61")
- Se utilizó el saldo de crédito PAYG (coincide con "*any")
- Se cobra a la tarifa PAYG por Australia

Pruebas y Simulación

Los cambios en los destinos se pueden probar y simular en entornos no productivos antes de aplicarse a producción. Esto permite:

- Verificar que los nuevos códigos PLMN sean correctos
- Probar el impacto de la expansión/restricción de cobertura
- Validar que los saldos existentes se comporten como se espera con los destinos actualizados
- Asegurarse de que los mecanismos de respaldo funcionen correctamente

Recomendación: Siempre prueba las actualizaciones de destinos en un entorno de pruebas con cuentas de prueba antes de aplicarlas a producción.

Conclusiones Clave

1. **Los destinos son dinámicos** - Los cambios afectan todos los saldos que los referencian una vez que se carga el plan de tarifas
2. **Cargar el Plan de Tarifas para activar** - Usa `LoadTariffPlanFromStorDb` para hacer activos los cambios de destino en tiempo de ejecución
3. **Sin coincidencia de destino = Sin servicio** - El sistema bloquea a los clientes si su uso no coincide con ningún destino
4. **Wildcard (`*any`) es poderoso** - Úsalo para respaldo monetario y evitar bloqueos inesperados
5. **Recarga de caché requerida** - Siempre recarga la caché de destinos después de cargar el plan de tarifas
6. **Prueba antes de producción** - Los cambios en los destinos impactan a todos los clientes una vez cargados

Perfiles de Tarifas para Saldos PAYG/Monetarios

Al usar saldos monetarios (PAYG - Paga a Medida que Usas), necesitas definir **Perfiles de Tarifas** que especifiquen cuánto cobrar por unidad para diferentes

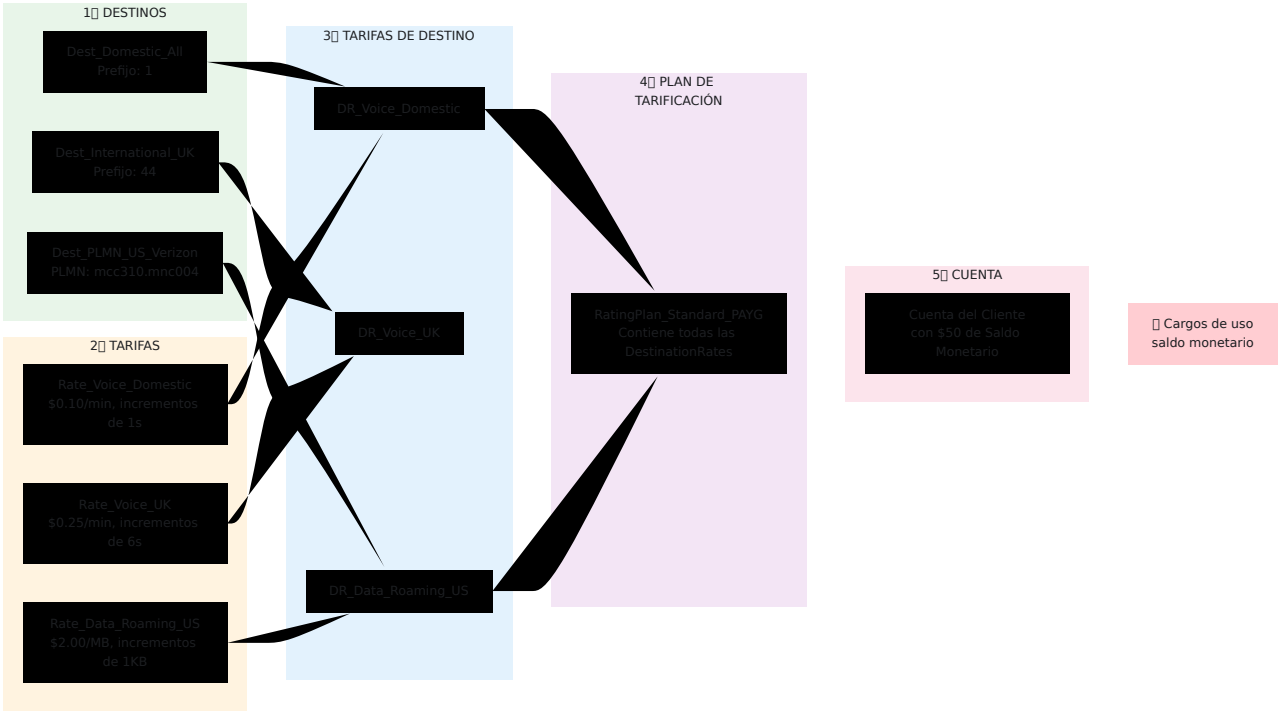
destinos.

Nota Importante sobre la Flexibilidad de CGRateS: CGRateS es extremadamente flexible y puede aplicar diferentes tarifas y planes según muchos factores, incluyendo:

- Desde dónde está llamando el cliente (PLMN - cubierto aquí)
- A qué están llamando (destino geográfico - cubierto aquí)
- Hora del día (tarificación en hora pico/fuera de hora pico)
- Nivel del cliente (VIP, estándar, etc.)
- Nivel de calidad del servicio
- Y muchos más escenarios de tarificación complejos

Esta guía se centra en la **configuración a nivel de CRM** para el roaming basado en PLMN y la tarificación basada en destinos geográficos, que cubre los casos de uso más comunes. Los escenarios de tarificación avanzados (hora del día, niveles de cliente, enrutamiento basado en calidad) son posibles en CGRateS pero están fuera del alcance de esta documentación centrada en CRM.

Cómo Funcionan los Perfiles de Tarifas



Definición de Perfiles de Tarifas

Los perfiles de tarifas definen el costo por unidad para diferentes tipos de uso. Aquí hay ejemplos para voz, SMS y datos:

```
#
================================================================
# TARIFAS DE VOZ (Por Minuto)
#
================================================================

# Voz nacional: $0.10/minuto
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPRate',
    'params': [{
        "TPid": tpid,
        "ID": "Rate_Voice_Domestic",
        "RateSlots": [{
            "ConnectFee": 0,
            "Rate": 0.10,
            "RateUnit": "60s",         # 1 minuto
            "RateIncrement": "1s",     # Facturación por segundo
            "GroupIntervalStart": "0s"
        }]
    }]
})

# Voz del Reino Unido: $0.25/minuto
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPRate',
    'params': [{
        "TPid": tpid,
        "ID": "Rate_Voice_UK",
        "RateSlots": [{
            "ConnectFee": 0.05,        # $0.05 de tarifa de
conexión
            "Rate": 0.25,
            "RateUnit": "60s",
            "RateIncrement": "6s",     # Bloques de 6 segundos
            "GroupIntervalStart": "0s"
        }]
    }]
})

#
================================================================
# TARIFAS DE SMS (Por Mensaje)
#
```

```
# ============================================================================

# SMS nacional: $0.05/mensaje
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPRate',
    'params': [{
        "TPid": tpid,
        "ID": "Rate_SMS_Domestic",
        "RateSlots": [{
            "ConnectFee": 0,
            "Rate": 0.05,
            "RateUnit": "1",          # 1 SMS
            "RateIncrement": "1",     # Por mensaje
            "GroupIntervalStart": "0s"
        }]
    }]
})

#
# ============================================================================
# TARIFAS DE DATOS (Por MB) - Para Roaming
#
# ============================================================================

# Roaming en Redes Premium de EE. UU. (Verizon, AT&T): $2.00/MB
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPRate',
    'params': [{
        "TPid": tpid,
        "ID": "Rate_Data_Roaming_US_Premium",
        "RateSlots": [{
            "ConnectFee": 0,
            "Rate": 2.00,
            "RateUnit": "1048576",    # 1 MB en bytes
            "RateIncrement": "1024",  # Por KB
            "GroupIntervalStart": "0s"
        }]
    }]
})
```

Campos de la Tarifa:

- **ConnectFee** - Cargo único por sesión

- **Rate** - Costo por RateUnit
- **RateUnit** - Tamaño de la unidad de facturación (60s, 1MB, 1 mensaje)
- **RateIncrement** - Unidad mínima cobrable
- **GroupIntervalStart** - Cuándo comienza este nivel de tarifa

Vinculación de Destinos a Tarifas

Después de definir las tarifas, vincúlelas a los destinos utilizando DestinationRates:

```
# Vincular destino nacional a tarifa de voz nacional
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestinationRate',
    'params': [{
        "TPid": tpid,
        "ID": "DR_Voice_Domestic",
        "DestinationRates": [{
            "DestinationId": "Dest_Domestic_All",
            "RateId": "Rate_Voice_Domestic",
            "RoundingMethod": "*up",
            "RoundingDecimals": 4
        }]
    }]
})

# Vincular destino del Reino Unido a tarifa de voz del Reino Unido
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestinationRate',
    'params': [{
        "TPid": tpid,
        "ID": "DR_Voice_UK",
        "DestinationRates": [{
            "DestinationId": "Dest_International_UK",
            "RateId": "Rate_Voice_UK",
            "RoundingMethod": "*up",
            "RoundingDecimals": 4
        }]
    }]
})

# Vincular PLMN de Verizon a tarifa de datos de roaming
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestinationRate',
    'params': [{
        "TPid": tpid,
        "ID": "DR_Data_Roaming_US_Premium",
        "DestinationRates": [{
            "DestinationId": "Dest_PLMN_US_Verizon",
            "RateId": "Rate_Data_Roaming_US_Premium",
            "RoundingMethod": "*up",
            "RoundingDecimals": 4
        }]
```

```
    }]
})
```

Creación del Plan de Tarifas

Combine todas las DestinationRates en un RatingPlan:

```
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPRatingPlan',
    'params': [{
        "TPid": tpid,
        "ID": "RatingPlan_Standard_PAYG",
        "RatingPlanBindings": [
            # Tarifas de voz (mayor peso = más específico)
            {"DestinationRatesId": "DR_Voice_UK", "TimingId":
"*any", "Weight": 40},
            {"DestinationRatesId": "DR_Voice_Domestic",
"TimingId": "*any", "Weight": 20},

            # Tarifas de SMS
            {"DestinationRatesId": "DR_SMS_Domestic", "TimingId":
"*any", "Weight": 20},

            # Tarifas de datos de roaming
            {"DestinationRatesId": "DR_Data_Roaming_US_Premium",
"TimingId": "*any", "Weight": 50}
        ]
    }]
})
```

Carga del Plan de Tarifas

Finalmente, carga el plan de tarifas en CGRateS:

```
# Cargar plan de tarifas
OCS_Obj.SendData({
    'method': 'ApierV1.LoadTariffPlanFromStorDb',
    'params': [{
        "TPid": tpid,
        "DryRun": False,
        "Validate": True
    }]
})

# Recargar caché
OCS_Obj.SendData({
    'method': 'CacheSv1.ReloadCache',
    'params': [{
        "Tenant": tenant,
        "DestinationIDs": ["*all"]
    }]
})
```

Consulta también:

- Definición de Productos - Flujo de trabajo completo para crear productos
- Acciones de CGRateS y Comportamientos de Recarga - Cómo definir Acciones y enfoques de gestión de saldos

Mejores Prácticas

1. Sé Específico con En Red

```
# Bueno - red local específica
"DestinationIDs": "Dest_PLMN_OnNet",  # mcc505.mnc057

# Malo - demasiado amplio, coincide con competidores
"DestinationIDs": "Dest_PLMN_AU_All",  # mcc505 (todos los
operadores de AU)
```

2. Usa Nombres de Destinos Significativos

```
# Bueno - auto-documentado
Dest_PLMN_US_Verizon
Dest_International_UK
Dest_PLMN_Zone_Europe

# Malo - poco claro
Dest_Data_1
Dest_Voice_Package
```

3. Agrupa Destinos Relacionados Lógicamente

```
# Bueno - agrupación lógica
Dest_Domestic_All
Dest_Domestic_TollFree
Dest_Domestic_Premium

# Malo - demasiado granular
Dest_Area_Code_212
Dest_Area_Code_213
Dest_Area_Code_214
```

4. Documenta Fuentes de PLMN

Al definir destinos PLMN, agrega comentarios que indiquen de dónde provienen los códigos MCC/MNC:

```
# Verizon Wireless
# Fuente: https://www.mcc-mnc.com/ (verificado 2024-12)
OCS_Obj.SendData({
    'method': 'ApierV2.SetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_PLMN_US_Verizon",
        "Prefixes": [
            "mcc310.mnc004",  # Verizon - Nacional
            "mcc310.mnc010",  # Verizon - Este
            ...
        ]
    }]
})
```

Solución de Problemas

Problema: Saldo No Coincide con el Destino

Síntoma: El cliente tiene saldo pero no puede usarlo para un destino específico

Verifica:

```
# 1. Verifica que el destino exista
OCS_Obj.SendData({
    'method': 'ApierV2.GetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_PLMN_US_Verizon"
    }]
})

# 2. Verifica el campo DestinationIDs del saldo
# Asegúrate de que incluya el destino o "*any"

# 3. Para PLMN: Verifica que el cliente esté realmente en esa red
# Verifica los mensajes CCR de Diameter para el AVP Visited-PLMN-
Id
```

Problema: Uso de Datos Contado como Roaming

Síntoma: Uso de datos nacionales consumiendo saldo de roaming en lugar de saldo en red

Causa: PLMN del cliente no definido en `Dest_PLMN_OnNet`

Solución:

```
# Verifica en qué PLMN está el cliente (desde Diameter CCR)
# Verifica que el saldo incluya ese PLMN:

OCS_Obj.SendData({
    'method': 'ApierV2.GetTPDestination',
    'params': [{
        "TPid": tpid,
        "ID": "Dest_PLMN_OnNet"
    }]
})
# Asegúrate de que los Prefijos incluyan la red PLMN real del
cliente
# Ejemplo: ["mcc505.mnc057"] debe coincidir con la red de la SIM
del cliente
```

Problema: Llamadas Internacionales No Funcionan

Síntoma: El cliente tiene "minutos internacionales" pero las llamadas fallan

Verifica:

1. **Prefijo de destino**: ¿Coincide el número marcado con el prefijo del destino?
   - Llamando a +44-20-xxxx → Necesita un destino con prefijo "44"
2. **DestinationIDs del saldo**: ¿Incluye el saldo ese destino?
   - Verifica que el saldo tenga `"DestinationIDs": "Dest_International_UK"` o `"Dest_International_All"`
3. **Saldo no expirado**: Verifica `ExpiryTime`
4. **Saldo tiene unidades**: Verifica `Units > 0`

Documentación Relacionada

- Acciones de CGRateS y Comportamientos de Recarga - Cómo usar destinos en Acciones, enfoques de saldo y estrategias de tarificación
- Definición de Productos - Flujo de trabajo completo para la creación de productos

Ansible Playbooks: Guía Detallada

Los productos de OmniCRM se aprovisionan utilizando **Ansible**, lo que permite la gestión automatizada de servicios basada en los requisitos específicos de cada producto y su inventario asociado.

Vea también: Aprovisionamiento de Tarjetas SIM para un ejemplo completo de aprovisionamiento basado en Ansible para servicios móviles, incluyendo SIM físicas y eSIM.

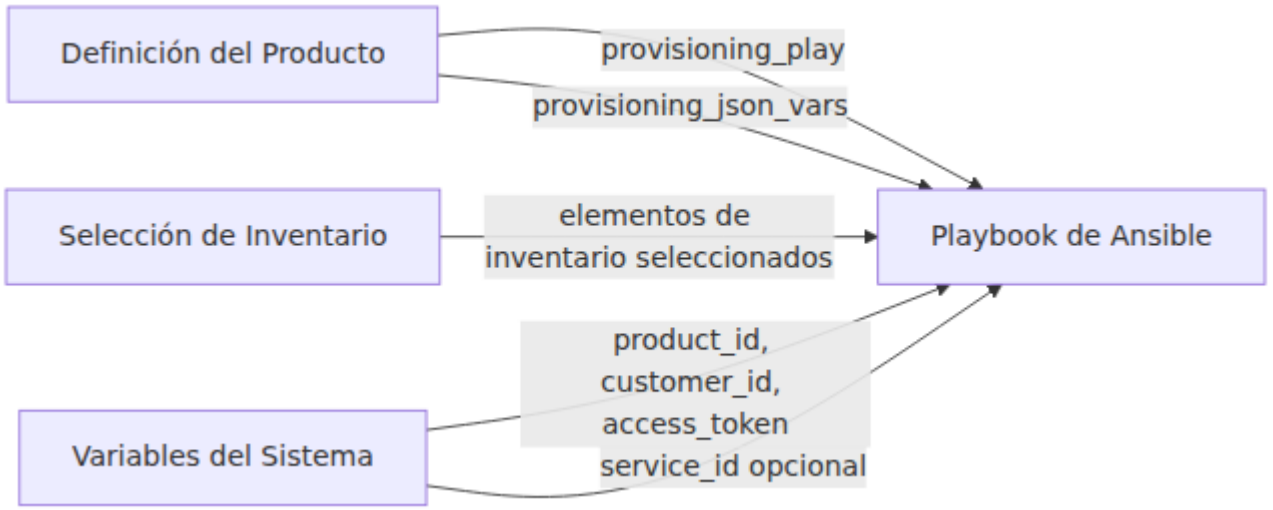
Cómo Funcionan Juntos los Playbooks y los Productos

Concepto Crítico: Los playbooks son los que realmente crean servicios en OmniCRM. Cuando asignas un playbook a un producto, estás definiendo **lo que sucede** cuando se aprovisiona ese producto. Eso puede significar cosas diferentes para diferentes productos.

Los Productos Activan los Playbooks

Cuando un producto se aprovisiona en OmniCRM:

1. La definición del producto especifica qué playbook ejecutar (a través del campo `provisioning_play`)
2. El producto pasa variables al playbook (a través de `provisioning_json_vars` y selecciones de inventario)
3. El playbook se ejecuta y hace lo que está programado para hacer
4. El playbook determina qué se crea (si es que se crea algo)



Lo Que Pueden Hacer los Playbooks

Un solo playbook de aprovisionamiento puede:

Crear Múltiples Servicios
Un playbook de producto agrupado podría crear:

- Un registro de servicio de internet principal
- Un registro de servicio adicional de IPTV
- Un registro de servicio de VoIP
- Todo con una acción de aprovisionamiento de producto

No Crear Servicios
Algunos playbooks no crean registros de servicio en absoluto:

- Un playbook que solo configura equipos CPE
- Un playbook que envía configuraciones a equipos de red
- Un playbook que actualiza sistemas externos

Crear Un Servicio
El patrón más común:

- Crear un único registro de servicio para el cliente
- Vincular inventario a ese servicio
- Configurar la facturación para ese servicio

Modificar Servicios Existentes
Playbooks de recarga y adicionales:

- No crean nuevos servicios
- Actualizan registros de servicio existentes (agregar datos, extender vencimiento, etc.)
- Agregan saldos a cuentas de facturación existentes

Realizar Acciones Sin Registros de Servicio
Algunos playbooks son puramente operacionales:

- Restablecer saldos de cuentas
- Intercambiar elementos de inventario entre clientes
- Generar informes o configuraciones

Ejemplo: Diferentes Comportamientos de Playbooks

```
# Producto 1: Servicio de SIM Móvil (crea 1 servicio)
# provisioning_play: play_simple_service
- Crea registro de servicio en CRM
- Crea cuenta de facturación en OCS
- Asigna inventario de tarjeta SIM y número de teléfono
- Envía correo electrónico de bienvenida

# Producto 2: Paquete de Internet (crea 3 servicios)
# provisioning_play: play_bundle_internet_tv_voice
- Crea registro de servicio de internet
- Crea registro de servicio de IPTV
- Crea registro de servicio de VoIP
- Vincula todo al mismo cliente
- Cuenta de facturación única para el paquete

# Producto 3: Recarga de Datos (crea 0 servicios)
# provisioning_play: play_topup_no_charge
- Encuentra servicio existente por service_id
- Agrega saldo de datos a cuenta OCS existente
- Actualiza fecha de vencimiento del servicio
- NO se crea un nuevo servicio

# Producto 4: Configuración de CPE (crea 0 servicios)
# provisioning_play: play_prov_cpe_mikrotik
- Genera configuración de router
- Actualiza registro de inventario con configuración
- Envía configuración al equipo de soporte
- NO se crea servicio (solo configuración de equipo)
```

El punto clave: **El playbook define el comportamiento, el producto es solo un desencadenante.**

Plays vs Tareas

Entender la distinción entre Plays y Tareas es fundamental para trabajar con los playbooks de OmniCRM.

Play (Playbook)
Un flujo de trabajo de aprovisionamiento completo que orquesta múltiples tareas para lograr un objetivo comercial. Los Plays son los playbooks de nivel superior almacenados en `OmniCRM-API/Provisioners/plays/` y se hacen referencia en las definiciones de productos.

Ejemplos:

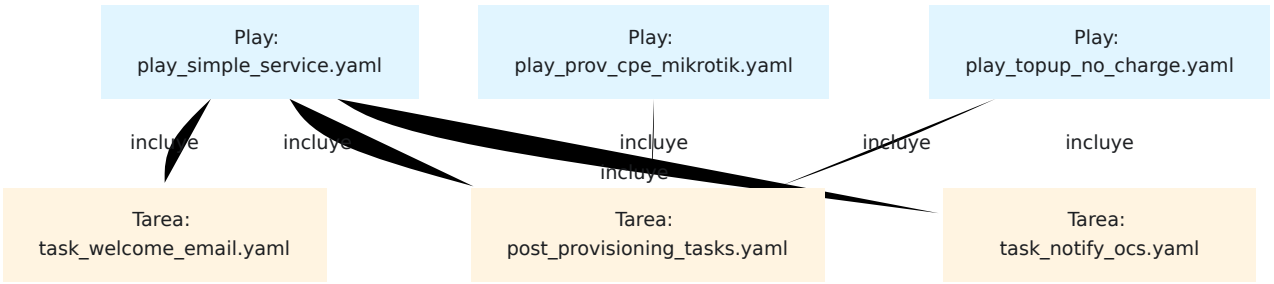
- `play_simple_service.yaml` - Aprovisionar un servicio básico
- `play_topup_no_charge.yaml` - Aplicar una recarga gratuita a un servicio
- `play_prov_cpe_mikrotik.yaml` - Configurar equipos de cliente

Tarea (Componente Reutilizable)
Un conjunto de operaciones autocontenido y reutilizable que puede ser incluido por múltiples plays. Las Tareas están prefijadas con `task_` y viven en el mismo directorio.

Ejemplos:

- `task_welcome_email.yaml` - Enviar un correo electrónico de bienvenida a un cliente
- `task_activate_olt.yaml` - Activar equipos OLT
- `task_notify_ocs.yaml` - Enviar notificaciones al sistema de facturación

La relación entre ellos:



```yaml
# play_simple_service.yaml (Un Play)
- name: Play de Aprovisionamiento Simple
  hosts: localhost
  tasks:
    - name: Bloque de aprovisionamiento principal
      block:
        - name: Crear servicio
          uri: ...

        - name: Configurar facturación
          uri: ...

        # Incluir tarea reutilizable
        - include_tasks: task_welcome_email.yaml

        # Incluir tareas post-aprovisionamiento
        - include_tasks: post_provisioning_tasks.yaml
```

Estructura y Anatomía del Playbook

Todos los playbooks de OmniCRM siguen una estructura consistente. Entender esta estructura es esencial para crear y mantener playbooks.

Estructura Básica

Cada playbook comienza con estos encabezados estándar:

```yaml
- name: Nombre Descriptivo del Playbook
  hosts: localhost           # Siempre localhost para OmniCRM
  gather_facts: no           # Deshabilitado por rendimiento
  become: False              # No escalar privilegios

  tasks:
    - name: Bloque principal
      block:
        # Las tareas de aprovisionamiento van aquí

      rescue:
        # Las tareas de reversión/limpieza van aquí
```

Explicación del Encabezado

name
Nombre descriptivo mostrado en los registros de aprovisionamiento y UI. Esto aparece como `playbook_description` en el registro de aprovisionamiento.

hosts: localhost
Todos los playbooks de OmniCRM se ejecutan en localhost ya que interactúan con sistemas remotos a través de APIs, no SSH.

gather_facts: no
La recopilación de hechos de Ansible está deshabilitada porque:

- No necesitamos información del sistema
- Agrega sobrecarga innecesaria
- Puede bloquear navegadores si se muestra en la salida de depuración

become: False
No se necesita escalada de privilegios ya que estamos haciendo llamadas a la API, no modificando archivos del sistema.

Carga de Configuración

Cada playbook debe cargar el archivo de configuración central:

```
tasks:
  - name: Incluir vars de crm_config
    ansible.builtin.include_vars:
      file: "../../crm_config.yaml"
      name: crm_config
```

Esto hace que la configuración esté disponible como `crm_config.ocs.cgrates`, `crm_config.crm.base_url`, etc.

El `crm_config.yaml` típicamente contiene:

```
ocs:
  cgrates: "10.0.1.100:2080"
  ocsTenant: "default_tenant"
crm:
  base_url: "https://crm.example.com"
```

Patrones de Acceso a Variables

Las variables pueden provenir de varias fuentes:

De la Definición del Producto:

```
- name: Acceder a product_id pasado por OmniCRM
  debug:
    msg: "Aprovisionando producto {{ product_id }}"
```

De la Selección de Inventario:

```
- name: Obtener ID de inventario para Tarjeta SIM
  set_fact:
    sim_card_id: "{{ hostvars[inventory_hostname]['SIM Card'] |
int }}"
  when: "'SIM Card' in hostvars[inventory_hostname]"
```

De Respuestas de API:

```
- name: Obtener información del Producto desde la API de CRM
  uri:
    url: "http://localhost:5000/crm/product/product_id/{{
product_id }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
    return_content: yes
  register: api_response_product

- name: Usar el nombre del producto
  debug:
    msg: "El nombre del producto es {{
api_response_product.json.product_name }}"
```

Patrones Comunes de Playbook

Patrón de Aprovisionamiento de Servicio

Este es el patrón más común para crear nuevos servicios.

```yaml
- name: Playbook de Aprovisionamiento de Servicio
  hosts: localhost
  gather_facts: no
  become: False

  tasks:
    - name: Bloque principal
      block:

      # 1. Cargar configuración
      - name: Incluir vars de crm_config
        ansible.builtin.include_vars:
          file: "../../crm_config.yaml"
          name: crm_config

      # 2. Obtener información del producto
      - name: Obtener información del Producto desde la API de CRM
        uri:
          url: "http://localhost:5000/crm/product/product_id/{{
product_id }}"
          method: GET
          headers:
            Authorization: "Bearer {{ access_token }}"
          return_content: yes
          validate_certs: no
        register: api_response_product

      # 3. Obtener información del cliente
      - name: Obtener información del Cliente desde la API de CRM
        uri:
          url: "http://localhost:5000/crm/customer/customer_id/{{
customer_id }}"
          method: GET
          headers:
            Authorization: "Bearer {{ access_token }}"
          return_content: yes
        register: api_response_customer

      # 4. Establecer hechos a partir de los datos recuperados
      - name: Establecer hechos del paquete
        set_fact:
          package_name: "{{ api_response_product.json.product_name
}}"
```

```
          package_comment: "{{ api_response_product.json.comment
}}"
          setup_cost: "{{
api_response_product.json.retail_setup_cost }}"
          monthly_cost: "{{ api_response_product.json.retail_cost
}}"

      # 5. Generar identificadores únicos
      - name: Generar UUID
        set_fact:
          uuid: "{{ 99999999 | random | to_uuid }}"

      - name: Generar UUID de Servicio
        set_fact:
          service_uuid: "Service_{{ uuid[0:8] }}"

      # 6. Crear cuenta en el sistema de facturación
      - name: Crear cuenta en OCS/CGRateS
        uri:
          url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
          method: POST
          body_format: json
          headers:
            Content-Type: "application/json"
          body:
            {
              "method": "ApierV2.SetAccount",
              "params": [{
                  "Tenant": "{{ crm_config.ocs.ocsTenant }}",
                  "Account": "{{ service_uuid }}",
                  "ActionPlanIds": [],
                  "ActionPlansOverwrite": true,
                  "ExtraOptions": {
                    "AllowNegative": false,
                    "Disabled": false
                  },
                  "ReloadScheduler": true
              }]
            }
          status_code: 200
        register: ocs_response

      - name: Verificar creación de cuenta OCS
        assert:
```

```
      that:
        - ocs_response.status == 200
        - ocs_response.json.result == "OK"

    # 7. Agregar saldo inicial
    - name: Agregar 0 Saldo Monetario
      uri:
        url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
        method: POST
        body_format: json
        body:
          {
            "method": "ApierV1.AddBalance",
            "params": [{
                "Tenant": "{{ crm_config.ocs.ocsTenant }}",
                "Account": "{{ service_uuid }}",
                "BalanceType": "*monetary",
                "Categories": "*any",
                "Balance": {
                    "ID": "Saldo Inicial",
                    "Value": 0,
                    "ExpiryTime": "+4320h",
                    "Weight": 1,
                    "Blocker": true
                }
            }]
          }
        status_code: 200
      register: balance_response

    # 8. Crear registro de servicio en CRM
    - name: Obtener fecha y hora actuales en formato ISO 8601
      command: date --utc +%Y-%m-%dT%H:%M:%S%z
      register: current_date_time

    - name: Agregar Servicio a través de API
      uri:
        url: "http://localhost:5000/crm/service/"
        method: PUT
        body_format: json
        headers:
          Content-Type: "application/json"
          Authorization: "Bearer {{ access_token }}"
        body:
```

```
            {
              "customer_id": "{{ customer_id }}",
              "product_id": "{{ product_id }}",
              "service_name": "{{ package_name }} - {{
service_uuid }}",
              "service_type": "generic",
              "service_uuid": "{{ service_uuid }}",
              "service_billed": true,
              "service_taxable": true,
              "service_provisioned_date": "{{
current_date_time.stdout }}",
              "service_status": "Active",
              "wholesale_cost": "{{
api_response_product.json.wholesale_cost | float }}",
              "retail_cost": "{{ monthly_cost | float }}"
            }
          status_code: 200
        register: service_creation_response

      # 9. Agregar transacción de costo de configuración
      - name: Agregar Transacción de Costo de Configuración a
través de API
        uri:
          url: "http://localhost:5000/crm/transaction/"
          method: PUT
          headers:
            Content-Type: "application/json"
            Authorization: "Bearer {{ access_token }}"
          body_format: json
          body:
            {
              "customer_id": {{ customer_id | int }},
              "service_id": {{
service_creation_response.json.service_id | int }},
              "title": "{{ package_name }} - Costos de
Configuración",
              "description": "Costos de configuración para {{
package_comment }}",
              "invoice_id": null,
              "retail_cost": "{{ setup_cost | float }}"
            }
          return_content: yes
        register: transaction_response
```

```
    # 10. Incluir tareas post-aprovisionamiento
    - include_tasks: post_provisioning_tasks.yaml

    rescue:

    # Sección de reversión/limpieza
    - name: Imprimir todas las vars para depuración
      debug:
        var: hostvars[inventory_hostname]

    - name: Eliminar cuenta en OCS
      uri:
        url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
        method: POST
        body_format: json
        body:
          {
            "method": "ApierV2.RemoveAccount",
            "params": [{
                "Tenant": "{{ crm_config.ocs.ocsTenant }}",
                "Account": "{{ service_uuid }}",
                "ReloadScheduler": true
            }]
          }
        status_code: 200
      ignore_errors: True
      when: service_uuid is defined

    - name: Eliminar Servicio de CRM si fue creado
      uri:
        url: "http://localhost:5000/crm/service/service_id/{{
service_creation_response.json.service_id }}"
        method: DELETE
        headers:
          Authorization: "Bearer {{ access_token }}"
        status_code: 200
      ignore_errors: True
      when: service_creation_response is defined

    - name: Fallar si no es una desprovisión intencional
      assert:
        that:
          - action == "deprovision"
```

Patrón de Recarga/Recarga

Utilizado para agregar créditos, datos o tiempo a servicios existentes.

```yaml
- name: Playbook de Recarga de Servicio
  hosts: localhost
  gather_facts: no
  become: False

  tasks:
    - name: Incluir vars de crm_config
      ansible.builtin.include_vars:
        file: "../../crm_config.yaml"
        name: crm_config

    # 1. Obtener información del servicio
    - name: Obtener información del Servicio desde la API de CRM
      uri:
        url: "http://localhost:5000/crm/service/service_id/{{
service_id }}"
        method: GET
        headers:
          Authorization: "Bearer {{ access_token }}"
        return_content: yes
      register: api_response_service

    # 2. Obtener información del producto (qué recargar)
    - name: Obtener información del Producto desde la API de CRM
      uri:
        url: "http://localhost:5000/crm/product/product_id/{{
product_id }}"
        method: GET
        headers:
          Authorization: "Bearer {{ access_token }}"
        return_content: yes
      register: api_response_product

    # 3. Extraer detalles del servicio
    - name: Establecer hechos del servicio
      set_fact:
        service_uuid: "{{ api_response_service.json.service_uuid
}}"
        customer_id: "{{ api_response_service.json.customer_id }}"
        package_name: "{{ api_response_product.json.product_name
}}"
        topup_value: "{{ api_response_product.json.retail_cost }}"
```

```
    # 4. Ejecutar acción en el sistema de facturación (recarga
gratuita)
    - name: Ejecutar Acción para agregar créditos
      uri:
        url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
        method: POST
        body_format: json
        body:
          {
            "method": "APIerSv1.ExecuteAction",
            "params": [{
                "Tenant": "{{ crm_config.ocs.ocsTenant }}",
                "Account": "{{ service_uuid }}",
                "ActionsId": "Action_Topup_Standard"
            }]
          }
        status_code: 200
      register: action_response

    - name: Verificar que la acción se ejecutó correctamente
      assert:
        that:
          - action_response.status == 200
          - action_response.json.result == "OK"

    # 5. Restablecer cualquier límite activado
    - name: Restablecer ActionTriggers
      uri:
        url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
        method: POST
        body_format: json
        body:
          {
            "method": "APIerSv1.ResetAccountActionTriggers",
            "params": [{
                "Tenant": "{{ crm_config.ocs.ocsTenant }}",
                "Account": "{{ service_uuid }}",
                "Executed": false
            }]
          }
        status_code: 200

    # 6. Actualizar fechas del servicio
    - name: Calcular nueva fecha de vencimiento
```

```
      command: "date --utc +%Y-%m-%dT%H:%M:%S%z -d '+30 days'"
      register: new_expiry_date

    - name: Actualizar Servicio con nueva fecha de vencimiento
      uri:
        url: "http://localhost:5000/crm/service/{{ service_id }}"
        method: PATCH
        headers:
          Authorization: "Bearer {{ access_token }}"
          Content-Type: "application/json"
        body_format: json
        body:
          {
            "service_deactivate_date": "{{ new_expiry_date.stdout
}}",
            "service_status": "Active"
          }

    # 7. Opcional: Enviar notificación
    - name: Enviar SMS de Notificación
      uri:
        url: "http://sms-gateway/api/send"
        method: POST
        body_format: json
        body:
          {
            "source": "CompanyName",
            "destination": "{{ customer_phone }}",
            "message": "Su servicio ha sido recargado. Nueva fecha
de vencimiento: {{ new_expiry_date.stdout }}"
          }
        status_code: 201
      ignore_errors: True
```

Patrón de Aprovisionamiento de CPE

Utilizado para configurar equipos de cliente (routers, módems, ONTs).

```yaml
- name: Playbook de Aprovisionamiento de CPE
  hosts: localhost
  gather_facts: no
  become: False

  tasks:
    - name: Incluir vars de crm_config
      ansible.builtin.include_vars:
        file: "../../crm_config.yaml"
        name: crm_config

    # 1. Obtener elemento de inventario para CPE
    - name: Establecer ID de inventario de CPE desde hostvars
      set_fact:
        cpe_inventory_id: "{{ hostvars[inventory_hostname]['WiFi
Router CPE'] | int }}"
      when: "'WiFi Router CPE' in hostvars[inventory_hostname]"

    # 2. Obtener detalles de CPE del inventario
    - name: Obtener datos de Inventario para CPE
      uri:
        url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ cpe_inventory_id }}"
        method: GET
        headers:
          Authorization: "Bearer {{ access_token }}"
        return_content: yes
      register: api_response_cpe

    # 3. Obtener información del sitio del cliente
    - name: Obtener información del Sitio desde la API
      uri:
        url: "{{ crm_config.crm.base_url
}}/crm/site/customer_id/{{ customer_id }}"
        method: GET
        headers:
          Authorization: "Bearer {{ access_token }}"
        return_content: yes
      register: api_response_site

    # 4. Actualizar inventario de CPE con ubicación
    - name: Parchear elemento de inventario de CPE con ubicación
      uri:
```

```
        url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ cpe_inventory_id }}"
        method: PATCH
        body_format: json
        headers:
          Authorization: "Bearer {{ access_token }}"
        body:
          {
            "address_line_1": "{{
api_response_site.json.0.address_line_1 }}",
            "city": "{{ api_response_site.json.0.city }}",
            "state": "{{ api_response_site.json.0.state }}",
            "latitude": "{{ api_response_site.json.0.latitude }}",
            "longitude": "{{ api_response_site.json.0.longitude
}}"
          }
        status_code: 200

    # 5. Generar credenciales
    - name: Establecer nombre de host de CPE
      set_fact:
        cpe_hostname: "CPE_{{ cpe_inventory_id }}"
        cpe_username: "admin_{{ cpe_inventory_id }}"

    - name: Generar contraseña aleatoria
      set_fact:
        cpe_password: "{{ lookup('pipe', 'cat /dev/urandom | tr -
dc a-zA-Z0-9 | head -c 16') }}"

    # 6. Generar credenciales de WiFi
    - name: Establecer SSID de WiFi
      set_fact:
        wifi_ssid: "Network_{{ cpe_inventory_id }}"

    - name: Generar contraseña de WiFi
      set_fact:
        word_list:
          - apple
          - cloud
          - river
          - mountain
          - ocean

    - name: Crear PSK de WiFi
```

```
      set_fact:
        random_word: "{{ word_list | random }}"
        random_number: "{{ 99999 | random(start=10000) }}"

    - name: Combinar PSK de WiFi
      set_fact:
        wifi_psk: "{{ random_word }}{{ random_number }}"

    # 7. Generar archivo de configuración
    - name: Establecer nombre de archivo de configuración
      set_fact:
        config_name: "{{ cpe_hostname }}_{{ lookup('pipe', 'date
+%Y%m%d%H%M%S') }}.cfg"
        config_dest: "/tmp/{{ cpe_hostname }}_{{ lookup('pipe',
'date +%Y%m%d%H%M%S') }}.cfg"

    - name: Crear configuración desde plantilla
      template:
        src: "templates/cpe_router_config.j2"
        dest: "{{ config_dest }}"

    # 8. Leer configuración generada
    - name: Leer archivo de configuración
      ansible.builtin.slurp:
        src: "{{ config_dest }}"
      register: config_content

    # 9. Actualizar inventario con información de
aprovisionamiento
    - name: Parchear inventario de CPE con configuración
      uri:
        url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ cpe_inventory_id }}"
        method: PATCH
        body_format: json
        headers:
          Authorization: "Bearer {{ access_token }}"
        body:
          {
            "itemtext3": "{{ wifi_ssid }}",
            "itemtext4": "{{ wifi_psk }}",
            "management_url": "{{ cpe_hostname }}",
            "management_username": "{{ cpe_username }}",
            "management_password": "{{ cpe_password }}",
```

```
            "config_content": "{{ config_content.content |
b64decode }}",
            "inventory_notes": "Aprovisionado: {{ lookup('pipe',
'date +%Y-%m-%d') }}"
          }
        status_code: 200

    # 10. Enviar configuración al equipo de soporte
    - name: Enviar configuración por correo electrónico al soporte
      uri:
        url: "https://api.mailjet.com/v3.1/send"
        method: POST
        body_format: json
        headers:
          Content-Type: "application/json"
        body:
          {
            "Messages": [{
              "From": {
                "Email": "provisioning@example.com",
                "Name": "Sistema de Aprovisionamiento"
              },
              "To": [{
                "Email": "support@example.com",
                "Name": "Equipo de Soporte"
              }],
              "Subject": "Configuración de CPE - {{ cpe_hostname
}}",
              "Attachments": [{
                "ContentType": "text/plain",
                "Filename": "{{ config_name }}",
                "Base64Content": "{{ config_content.content }}"
              }]
            }]
          }
        user: "{{ mailjet_api_key }}"
        password: "{{ mailjet_api_secret }}"
        force_basic_auth: true
        status_code: 200
```

Patrón de Renovación Automática

Configurar cargos recurrentes automáticos o renovaciones utilizando Planes de Acción de CGRateS.

```
# Parte de un playbook de recarga que configura la renovación
automática

# 1. Normalizar par??metro auto_renew
- name: Normalizar auto_renew a booleano
  set_fact:
    auto_renew_bool: "{{ (auto_renew | string | lower) in ['true',
'1', 'yes'] }}"

# 2. Crear acción para renovación automática
- name: Crear Acción para AutoRenovación
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.SetActions",
        "params": [{
          "ActionsId": "Action_AutoTopup_{{ service_uuid }}_{{
product_id }}",
          "Overwrite": true,
          "Actions": [
            {
              "Identifier": "*http_post",
              "ExtraParameters": "{{ crm_config.crm.base_url
}}/crm/provision/simple_provision_addon/service_id/{{ service_id
}}/product_id/{{ product_id }}"
            },
            {
              "Identifier": "*cdrlog",
              "BalanceType": "*generic",
              "ExtraParameters": "
{\"Category\":\"^activation\",\"Destination\":\"Auto Renewal\"}"
            }
          ]
        }]
      }
    status_code: 200
  register: action_response
  when: auto_renew_bool

# 3. Crear Plan de Acción mensual
```

```
- name: Crear Plan de Acción para Renovación Mensual
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.SetActionPlan",
        "params": [{
          "Id": "ActionPlan_Monthly_{{ service_uuid }}_{{
product_id }}",
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "ActionPlan": [{
            "ActionsId": "Action_AutoTopup_{{ service_uuid }}_{{
product_id }}",
            "Years": "*any",
            "Months": "*any",
            "MonthDays": "*any",
            "WeekDays": "*any",
            "Time": "*monthly",
            "StartTime": "*now",
            "Weight": 10
          }],
          "Overwrite": true,
          "ReloadScheduler": true
        }]
      }
    status_code: 200
  when: auto_renew_bool

# 4. Asignar Plan de Acción a la cuenta
- name: Asignar Plan de Acción a la cuenta
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV2.SetAccount",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "ActionPlanIds": ["ActionPlan_Monthly_{{ service_uuid
}}_{{ product_id }}"],
```

```
          "ActionPlansOverwrite": true,
          "ReloadScheduler": true
        }]
      }
    status_code: 200
  when: auto_renew_bool

# 5. Eliminar Plan de Acción si la renovación automática está
deshabilitada
- name: Eliminar Plan de Acción de la cuenta
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.RemoveActionPlan",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Id": "ActionPlan_Monthly_{{ service_uuid }}_{{
product_id }}"
        }]
      }
    status_code: 200
  ignore_errors: true
  when: not auto_renew_bool
```

Tareas Reutilizables

Las tareas reutilizables son pequeños playbooks autocontenidos que pueden ser incluidos por múltiples plays. Promueven la reutilización de código y la consistencia.

Tarea de Correo Electrónico de Bienvenida

`task_welcome_email.yaml` - Envía un correo electrónico de bienvenida a nuevos clientes.

```
# Esta tarea espera que estas variables sean establecidas por el
play padre:
# - api_response_customer (detalles del cliente)
# - package_name (nombre del producto)
# - monthly_cost (costo recurrente)
# - setup_cost (costo único)

- name: Establecer configuración de correo electrónico
  set_fact:
    mailjet_api_key: "{{ lookup('env', 'MAILJET_API_KEY') }}"
    mailjet_api_secret: "{{ lookup('env', 'MAILJET_SECRET') }}"
    email_from: "noreply@example.com"
    recipients: []

- name: Establecer asunto del correo y nombre del remitente
  set_fact:
    email_subject: "¡Bienvenido a nuestro servicio!"
    email_from_name: "Equipo de Atención al Cliente"

- name: Preparar lista de destinatarios de los contactos del
cliente
  loop: "{{ api_response_customer.json.contacts }}"
  set_fact:
    recipients: "{{ recipients + [{'Email': item.contact_email,
'Name': item.contact_firstname ~ ' ' ~ item.contact_lastname}] }}"

- name: Obtener nombre del primer contacto
  set_fact:
    first_contact: "{{
api_response_customer.json.contacts[0].contact_firstname }}"

- name: Enviar correo electrónico de bienvenida
  uri:
    url: "https://api.mailjet.com/v3.1/send"
    method: POST
    body_format: json
    headers:
      Content-Type: "application/json"
    body:
      {
        "Messages": [{
          "From": {
            "Email": "{{ email_from }}",
```

```
            "Name": "{{ email_from_name }}"
          },
          "To": "{{ recipients }}",
          "Subject": "{{ email_subject }}",
          "TextPart": "Estimado {{ first_contact }}, ¡bienvenido!
Su servicio está listo.",
          "HTMLPart": "Estimado {{ first_contact }},<br/>
<h3>¡Bienvenido!</h3><br/>Su servicio {{ package_name }} está
ahora activo.<br/>Costo mensual: ${{ monthly_cost }}<br/>Costo de
configuración: ${{ setup_cost }}<br/>Si tiene alguna pregunta,
contacte a support@example.com"
        }]
      }
    user: "{{ mailjet_api_key }}"
    password: "{{ mailjet_api_secret }}"
    force_basic_auth: true
    status_code: 200
  register: email_response
```

Tareas Post Aprovisionamiento

`post_provisioning_tasks.yaml` - Limpieza estándar y notificaciones que se ejecutan después de cada aprovisionamiento.

```
# Este archivo se incluye al final de la mayoría de los playbooks
de aprovisionamiento
# Maneja operaciones comunes post-aprovisionamiento

- include_tasks: task_notify_ocs.yaml
```

El `task_notify_ocs.yaml` podría contener:

```
- name: Notificar a OCS de la finalización del aprovisionamiento
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "APIerSv1.ReloadCache",
        "params": [{
          "ArgsCache": "*all"
        }]
      }
    status_code: 200
  ignore_errors: true
```

Operaciones Comunes

Trabajando con Inventario

Recuperando detalles del inventario:

```
- name: Obtener ID de inventario de la Tarjeta SIM
  set_fact:
    sim_inventory_id: "{{ hostvars[inventory_hostname]['SIM Card']
| int }}"
  when: "'SIM Card' in hostvars[inventory_hostname]"

- name: Obtener detalles de la Tarjeta SIM
  uri:
    url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ sim_inventory_id }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
    return_content: yes
  register: sim_response

- name: Extraer detalles de la SIM
  set_fact:
    iccid: "{{ sim_response.json.iccid }}"
    imsi: "{{ sim_response.json.imsi }}"
    ki: "{{ sim_response.json.ki }}"
```

Asignando inventario al cliente:

```
- name: Asignar SIM al cliente
  uri:
    url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ sim_inventory_id }}"
    method: PATCH
    headers:
      Authorization: "Bearer {{ access_token }}"
    body_format: json
    body:
      {
        "customer_id": {{ customer_id }},
        "service_id": {{ service_id }},
        "item_state": "Assigned"
      }
    status_code: 200
```

Operaciones de Fecha y Hora

Obteniendo la fecha/hora actual:

```
- name: Obtener fecha y hora actuales en formato ISO 8601
  command: date --utc +%Y-%m-%dT%H:%M:%S%z
  register: current_date_time

- name: Obtener solo la fecha de hoy
  set_fact:
    today: "{{ lookup('pipe', 'date +%Y-%m-%d') }}"
```

Calculando fechas futuras:

```
- name: Calcular fecha de vencimiento 30 días a partir de ahora
  command: "date --utc +%Y-%m-%dT%H:%M:%S%z -d '+30 days'"
  register: expiry_date

- name: Calcular fecha 90 días en el futuro
  command: "date --utc +%Y-%m-%d -d '+{{ days }} days'"
  register: future_date
  vars:
    days: 90
```

Generando Valores Aleatorios

UUIDs e identificadores:

```
- name: Generar UUID
  set_fact:
    uuid: "{{ 99999999 | random | to_uuid }}"

- name: Generar identificador de servicio
  set_fact:
    service_uuid: "SVC_{{ uuid[0:8] }}"
```

Contraseñas aleatorias:

```
- name: Generar contraseña segura
  set_fact:
    password: "{{ lookup('pipe', 'cat /dev/urandom | tr -dc a-zA-
Z0-9 | head -c 16') }}"
```

Frases de contraseña memorables:

```
- name: Establecer lista de palabras
  set_fact:
    words:
      - alpha
      - bravo
      - charlie
      - delta
      - echo

- name: Generar frase de contraseña
  set_fact:
    word: "{{ words | random }}"
    number: "{{ 99999 | random(start=10000) }}"

- name: Combinar en frase de contraseña
  set_fact:
    passphrase: "{{ word }}{{ number }}"
```

Trabajando con CGRateS/OCS

Creando cuentas:

```
- name: Crear cuenta de facturación
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV2.SetAccount",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "ActionPlanIds": [],
          "ActionPlansOverwrite": true,
          "ExtraOptions": {
            "AllowNegative": false,
            "Disabled": false
          },
          "ReloadScheduler": true
        }]
      }
    status_code: 200
  register: account_response
```

Agregando saldos:

```
- name: Agregar saldo de datos
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "BalanceType": "*data",
          "Categories": "*any",
          "Balance": {
            "ID": "Paquete de Datos",
            "Value": 10737418240,
            "ExpiryTime": "+720h",
            "Weight": 10
          }
        }]
      }
    status_code: 200
```

Ejecutando acciones:

```
- name: Ejecutar acción de cobro
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "APIerSv1.ExecuteAction",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "ActionsId": "Action_Standard_Charge"
        }]
      }
    status_code: 200
```

Obteniendo información de la cuenta:

```
- name: Obtener detalles de la cuenta
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV2.GetAccount",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}"
        }]
      }
    status_code: 200
  register: account_info
```

Trabajando con Perfiles de Atributos:

```
- name: Obtener Perfil de Atributo
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "APIerSv1.GetAttributeProfile",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "ID": "ATTR_{{ service_uuid }}"
        }]
      }
    return_content: yes
    status_code: 200
  register: attr_response
  ignore_errors: true

- name: Extraer valor de atributo
  set_fact:
    phone_number: "{{ attr_response.json.result.Attributes |
json_query(\"[?Path=='*req.PhoneNumber'].Value[0].Rules\") | first
}}"
  when: attr_response is defined
```

Lógica Condicional

Verificando si las variables existen:

```
- name: Usar valor personalizado o predeterminado
  set_fact:
    monthly_cost: "{{ custom_cost | default(50.00) }}"

- name: Solo ejecutar si la variable está definida
  debug:
    msg: "El UUID del servicio es {{ service_uuid }}"
  when: service_uuid is defined
```

Condiciones booleanas:

```yaml
- name: Aprovisionar equipo
  include_tasks: configure_cpe.yaml
  when: provision_cpe | default(false) | bool

- name: Omitir si es desprovisión
  assert:
    that:
      - action != "deprovision"
  when: action is defined
```

Múltiples condiciones:

```yaml
- name: Tarea condicional compleja
  uri:
    url: "{{ endpoint }}"
    method: POST
  when:
    - service_uuid is defined
    - customer_id is defined
    - action != "deprovision"
    - enable_feature | default(true) | bool
```

Bucles e Iteración

Bucles simples:

```
- name: Crear múltiples saldos
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
          "Account": "{{ service_uuid }}",
          "BalanceType": "{{ item.type }}",
          "Balance": {
            "Value": "{{ item.value }}"
          }
        }]
      }
  loop:
    - { type: "*voice", value: 3600 }
    - { type: "*data", value: 10737418240 }
    - { type: "*sms", value: 100 }
```

Iterando sobre respuestas de API:

```
- name: Obtener todos los sitios del cliente
  uri:
    url: "{{ crm_config.crm.base_url }}/crm/site/customer_id/{{
customer_id }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
  register: sites_response

- name: Configurar equipo en cada sitio
  debug:
    msg: "Configurando sitio en {{ item.address_line_1 }}"
  loop: "{{ sites_response.json }}"
```

Manejo de Errores

Usando ignore_errors:

```yaml
- name: Notificación SMS opcional
  uri:
    url: "http://sms-gateway/send"
    method: POST
    body: {...}
  ignore_errors: true
```

Afirmaciones para validación:

```yaml
- name: Verificar respuesta de API
  assert:
    that:
      - response.status == 200
      - response.json.result == "OK"
    fail_msg: "La llamada a la API falló: {{ response.json }}"
```

Manejo de errores condicional:

```yaml
- name: Intentar obtener servicio existente
  uri:
    url: "{{ crm_config.crm.base_url
}}/crm/service/service_uuid/{{ service_uuid }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
  register: service_lookup
  failed_when: false

- name: Crear servicio si no existe
  uri:
    url: "{{ crm_config.crm.base_url }}/crm/service/"
    method: PUT
    body: {...}
  when: service_lookup.status == 404
```

Mejores Prácticas

Nombres de Variables

Usar nombres descriptivos y consistentes:

```yaml
# Bueno
service_uuid: "SVC_12345"
customer_name: "John Smith"
monthly_cost: 49.99

# Malo
svc: "SVC_12345"
name: "John Smith"
cost: 49.99
```

Prefijar variables por fuente:

```yaml
api_response_customer: {...}
api_response_product: {...}
cgr_account_info: {...}
```

Depuración

Imprimir variables para solucionar problemas:

```yaml
- name: Imprimir todas las variables
  debug:
    var: hostvars[inventory_hostname]

- name: Imprimir variable específica
  debug:
    msg: "UUID del servicio: {{ service_uuid }}"

- name: Imprimir respuesta de API
  debug:
    var: api_response_product.json
```

Validación

Siempre validar respuestas críticas de API:

```
- name: Crear cuenta
  uri:
    url: "{{ billing_endpoint }}"
    method: POST
    body: {...}
  register: response

- name: Verificar creación de cuenta
  assert:
    that:
      - response.status == 200
      - response.json.result == "OK"
    fail_msg: "Fallo al crear cuenta: {{ response.json }}"
```

Idempotencia

Diseñar tareas para que sean seguras de volver a ejecutar:

```
# Verificar si el recurso existe primero
- name: Verificar si la cuenta existe
  uri:
    url: "{{ ocs_endpoint }}/get_account"
    method: POST
    body: {"Account": "{{ service_uuid }}"}
  register: account_check
  failed_when: false

# Solo crear si no existe
- name: Crear cuenta
  uri:
    url: "{{ ocs_endpoint }}/create_account"
    method: POST
    body: {...}
  when: account_check.status == 404
```

Seguridad

Nunca codificar credenciales:

```
# Malo
mailjet_api_key: "abc123def456"

# Bueno - usar variables de entorno
mailjet_api_key: "{{ lookup('env', 'MAILJET_API_KEY') }}"

# Bueno - usar archivo de configuración
mailjet_api_key: "{{ crm_config.email.api_key }}"
```

Siempre usar HTTPS y autenticación:

```
- name: Llamar a API externa
  uri:
    url: "https://api.example.com/endpoint"
    method: POST
    headers:
      Authorization: "Bearer {{ access_token }}"
    validate_certs: yes
```

Documentación

Documentar lógica compleja:

```
# Calcular cargo prorrateado por mes parcial
# Si el cliente se inscribe el día 15 y la facturación es el día
1,
# cobrar el 50% del costo mensual por los días restantes
- name: Calcular días hasta el final del mes
  command: "date -d 'last day of this month' +%d"
  register: days_in_month

- name: Obtener día actual
  command: "date +%d"
  register: current_day

- name: Calcular monto prorrateado
  set_fact:
    days_remaining: "{{ (days_in_month.stdout | int) -
(current_day.stdout | int) }}"
    pro_rata_cost: "{{ (monthly_cost | float) * (days_remaining |
float) / (days_in_month.stdout | float) }}"
```

Pruebas de Playbooks

Enfoque de Pruebas

1. **Ejecutar Simulación Primero**: Probar con sistemas no productivos
2. **Verificar Variables**: Usar tareas de depuración para confirmar que todas las variables requeridas están presentes
3. **Verificar Respuestas**: Validar respuestas de API antes de continuar
4. **Pruebas de Reversión**: Fallar intencionalmente tareas para verificar que los bloques de rescate funcionen
5. **Pruebas de Desprovisión**: Probar con `action: "deprovision"` para verificar limpieza

Ejemplo de playbook de prueba:

```yaml
- name: Probar Aprovisionamiento de Servicio
  hosts: localhost
  gather_facts: no

  tasks:
    - name: Verificar variables requeridas
      assert:
        that:
          - product_id is defined
          - customer_id is defined
          - access_token is defined
        fail_msg: "Faltan variables requeridas"

    - name: Probar conectividad de API
      uri:
        url: "http://localhost:5000/crm/health"
        method: GET
      register: health_check

    - name: Verificar chequeo de salud
      assert:
        that:
          - health_check.status == 200
```

Errores Comunes

Faltan conversiones de tipo:

```yaml
# Incorrecto - puede ser cadena
customer_id: "{{ customer_id }}"

# Correcto - asegurar entero
customer_id: {{ customer_id | int }}
```

No manejar variables indefinidas:

```
# Incorrecto - falla si no está definida
service_uuid: "{{ service_uuid }}"

# Correcto - proporcionar predeterminado
service_uuid: "{{ service_uuid | default('') }}"
```

Olvidar validación:

```
# Incorrecto - no verifica respuesta
- name: Crear cuenta
  uri: ...
  register: response

# Correcto - validar respuesta
- name: Crear cuenta
  uri: ...
  register: response

- name: Verificar creación
  assert:
    that:
      - response.json.result == "OK"
```

Flujo de Trabajo de Aprovisionamiento

Generalmente, el personal de Omnitouch trabajará con el cliente para:

1. Definir los requisitos del producto
2. Desarrollar los playbooks de Ansible necesarios para automatizar el proceso de aprovisionamiento
3. Probar los playbooks en un entorno de pruebas
4. Desplegar en producción

Esto asegura que cada servicio se despliegue de manera consistente y confiable, reduciendo el riesgo de errores y asegurando que todos los pasos necesarios se completen en el orden correcto.

Variables de Ansible

Las variables pasadas a los playbooks de Ansible incluyen:

Variables del Producto
Derivadas de las configuraciones del producto de OmniCRM y definen cómo se debe configurar el servicio.

Variables de Inventario
Seleccionadas del inventario, estas incluyen elementos como módems, tarjetas SIM, bloques de direcciones IP o números de teléfono que son necesarios para el aprovisionamiento.

Variables del Sistema
Agregadas automáticamente por OmniCRM:

- `product_id` - El producto que se está aprovisionando
- `customer_id` - El cliente que recibe el servicio
- `service_id` - El servicio que se está modificando (para recargas/cambios)
- `access_token` - JWT para autenticación de API

Desprovisionamiento

Cuando un servicio ya no es necesario, los **Playbooks de Ansible** también se utilizan para desprovisionar el servicio utilizando el patrón de bloque `rescue`. Esto:

- Elimina cualquier configuración
- Libera el inventario de vuelta al grupo
- Elimina cuentas de facturación
- Asegura que el sistema se mantenga limpio

Métodos de Desprovisionamiento

Hay dos formas principales en que se activa el desprovisionamiento:

1. Reversión Automática (Fallo de Aprovisionamiento) Cuando cualquier tarea en el bloque principal de aprovisionamiento falla, la sección de rescate se ejecuta automáticamente para limpiar los cambios parciales.

2. Desprovisionamiento Manual Establecer `action: "deprovision"` al ejecutar un playbook activa el bloque de rescate intencionalmente para eliminar un servicio.

Patrón de Desprovisionamiento

Todos los playbooks de OmniCRM siguen esta estructura para una limpieza segura:

```yaml
- name: Playbook de Aprovisionamiento de Servicio
  hosts: localhost
  gather_facts: no
  become: False

  tasks:
    - name: Bloque principal
      block:
        # Todas las tareas de aprovisionamiento van aquí
        - name: Crear cuenta OCS
          uri: ...

        - name: Crear registro de servicio
          uri: ...
          register: service_creation_response

        - name: Agregar transacción
          uri: ...

      rescue:
        # Tareas de desprovisionamiento/reversión van aquí
        - name: Imprimir todas las vars para depuración
          debug:
            var: hostvars[inventory_hostname]

        # Las tareas de limpieza se ejecutan en orden inverso
        - name: Eliminar cuenta en OCS
          uri: ...

        - name: Eliminar Servicio de CRM
          uri: ...

        - name: Fallar si no es desprovisión intencional
          assert:
            that:
              - action == "deprovision"
```

Ejemplo Completo de Desprovisionamiento

Aquí hay un ejemplo completo de `play_simple_service.yaml`:

```
rescue:
  # 1. Información de depuración
  - name: Imprimir todas las vars para Desprovisión/Reversión
    debug:
      var: hostvars[inventory_hostname]

  - name: Obtener fecha de hoy
    set_fact:
      today: "{{ lookup('pipe', 'date +%Y-%m-%d') }}"

  # 2. Obtener información del servicio si está disponible
  - name: Intentar obtener información del Servicio desde la API
de CRM para Desprovisionar
    uri:
      url: "http://localhost:5000/crm/service/service_id/{{
service_creation_response.json.service_id }}"
      method: GET
      headers:
        Authorization: "Bearer {{ access_token }}"
      return_content: yes
    register: api_response_service
    ignore_errors: True
    when: service_creation_response is defined and
service_creation_response.json is defined and
service_creation_response.json.service_id is defined

  - name: Imprimir api_response_service
    debug:
      var: api_response_service
    when: api_response_service is defined

  # 3. Establecer hechos de service_uuid para limpieza
  - name: Establecer hechos de service_uuid para Desprovisionar
    set_fact:
      service_uuid: "{{ api_response_service.json.service_uuid }}"
    when:
      - service_uuid is not defined
      - api_response_service is defined
      - api_response_service.json is defined

  # 4. Eliminar cuenta OCS
  - name: Eliminar cuenta en OCS
    uri:
```

```
      url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
      method: POST
      body_format: json
      headers:
        Content-Type: "application/json"
        Authorization: "Bearer {{ access_token }}"
      body:
        {
          "method": "ApierV2.RemoveAccount",
          "params": [{
            "Tenant": "{{ crm_config.ocs.ocsTenant }}",
            "Account": "{{ service_uuid }}",
            "ReloadScheduler": true
          }]
        }
      status_code: 200
    register: response
    ignore_errors: True
    when: service_uuid is defined

  - name: Imprimir respuesta de eliminación de cuenta OCS
    debug:
      var: response
    when: response is defined

  # 5. Eliminar registro de servicio de CRM
  - name: Eliminar Servicio de CRM si fue creado
    uri:
      url: "http://localhost:5000/crm/service/service_id/{{
service_creation_response.json.service_id }}"
      method: DELETE
      headers:
        Authorization: "Bearer {{ access_token }}"
      status_code: 200
    register: delete_service_response
    ignore_errors: True
    when: service_creation_response is defined and
service_creation_response.json is defined and
service_creation_response.json.service_id is defined

  - name: Imprimir respuesta de eliminación de servicio
    debug:
      var: delete_service_response
    when: delete_service_response is defined
```

```
  # 6. Determinar si esto fue intencional o un fallo
  - name: Establecer estado a "Éxito" si desprovisión manual /
Fallar si fallo de aprovisionamiento
    assert:
      that:
        - action == "deprovision"
      fail_msg: "El aprovisionamiento falló y fue revertido"
      success_msg: "Servicio desprovisionado con éxito"
```

Conceptos Clave de Desprovisionamiento

ignore_errors: True La mayoría de las tareas de limpieza utilizan `ignore_errors: True` porque los recursos pueden no existir (por ejemplo, si el aprovisionamiento falló antes de crearlos).

Ejecución Condicional Usar cláusulas `when` para limpiar solo los recursos que fueron creados:

```
when: service_creation_response is defined
```

Orden Inverso La limpieza ocurre en orden inverso a la creación:

1. Eliminar recursos dependientes primero (transacciones, asignaciones de inventario)
2. Eliminar registro de servicio
3. Eliminar cuenta de OCS/facturación
4. Liberar cualquier recurso retenido

Afirmación Final La afirmación final distingue entre:

- **Desprovisión intencional** (`action == "deprovision"`) → Éxito
- **Fallo de aprovisionamiento** (acción no establecida) → Fallo

Desprovisionamiento de Diferentes Tipos de Recursos

Cuentas de OCS/CGRateS:

```
- name: Eliminar cuenta en OCS
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV2.RemoveAccount",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "ReloadScheduler": true
        }]
      }
    status_code: 200
  ignore_errors: True
  when: service_uuid is defined
```

Registros de Servicio de CRM:

```
- name: Eliminar Servicio de CRM
  uri:
    url: "http://localhost:5000/crm/service/service_id/{{
service_id }}"
    method: DELETE
    headers:
      Authorization: "Bearer {{ access_token }}"
    status_code: 200
  ignore_errors: True
  when: service_id is defined
```

Elementos de Inventario (Devolver al Grupo):

```
- name: Liberar tarjeta SIM de vuelta al grupo de inventario
  uri:
    url: "http://localhost:5000/crm/inventory/inventory_id/{{
sim_inventory_id }}"
    method: PATCH
    headers:
      Authorization: "Bearer {{ access_token }}"
    body_format: json
    body:
      {
        "customer_id": null,
        "service_id": null,
        "item_state": "Available"
      }
    status_code: 200
  ignore_errors: True
  when: sim_inventory_id is defined
```

Planes de Acción (Cargos Recurrentes):

```
- name: Eliminar Plan de Acción de la cuenta
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.RemoveActionPlan",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Id": "ActionPlan_Monthly_{{ service_uuid }}_{{
product_id }}"
        }]
      }
    status_code: 200
  ignore_errors: True
```

Autorizaciones de Pago:

```
- name: Liberar autorización de pago
  uri:
    url: "http://localhost:5000/crm/payments/release/{{
authorization_id }}"
    method: POST
    headers:
      Authorization: "Bearer {{ access_token }}"
    body_format: json
    body:
      {
        "metadata": {
          "release_reason": "provisioning_failed"
        }
      }
    return_content: yes
  ignore_errors: True
  when: authorization_id is defined
```

Flujo de Trabajo de Desprovisionamiento Manual

Para desprovisionar un servicio manualmente:

1. Identificar el servicio a desprovisionar
2. Ejecutar el playbook de aprovisionamiento original con `action: "deprovision"`
3. El playbook entra en el bloque de rescate inmediatamente
4. Todas las tareas de limpieza se ejecutan
5. El servicio se elimina de manera limpia

Ejemplo de llamada a API:

```
POST /crm/provision/run_playbook
{
  "product_id": 123,
  "customer_id": 456,
  "service_id": 789,
  "action": "deprovision"
}
```

Escenarios de Limpieza Parcial

Escenario 1: Cuenta OCS Creada, Creación de Servicio Fallida

- La cuenta OCS existe
- El registro de servicio NO existe
- El bloque de rescate elimina la cuenta OCS
- No hay servicio que eliminar (safely skipped with `when` condition)

Escenario 2: Servicio Creado, Transacción Fallida

- La cuenta OCS existe
- El registro de servicio existe
- La transacción NO existe
- El bloque de rescate elimina tanto la cuenta OCS como el servicio
- No hay transacción que anular (nunca se creó)

Escenario 3: Aprovisionamiento Completo, Captura de Pago Fallida

- La cuenta OCS existe
- El registro de servicio existe
- Pago autorizado pero NO capturado
- El bloque de rescate:
  - Libera la autorización de pago (el cliente no es cobrado)
  - Elimina el registro de servicio
  - Elimina la cuenta OCS

Mejores Prácticas para Desprovisionamiento

1. Usar ignore_errors liberalmente La limpieza debe ser indulgente. No fallar si un recurso no existe.

2. Verificar existencia de recursos con cláusulas when Solo intentar limpiar si el recurso fue creado:

```
when: service_creation_response is defined
```

3. Imprimir información de depuración Siempre incluir tareas de depuración para ayudar a solucionar problemas de aprovisionamientos fallidos:

```
- name: Imprimir todas las vars para depuración
  debug:
    var: hostvars[inventory_hostname]
```

4. Limpiar en orden inverso de dependencia Eliminar hijos antes que padres:

- Transacciones antes de servicios
- Servicios antes de cuentas OCS
- Asignaciones de inventario antes de liberaciones de inventario

5. Manejar autorizaciones de pago Siempre liberar autorizaciones de pago en bloques de rescate para evitar cobrar a los clientes por aprovisionamientos fallidos.

6. Recargar programadores después de la limpieza Al eliminar recursos de OCS, incluir `ReloadScheduler: true` para asegurar que CGRateS se actualice de inmediato.

Desprovisionamiento vs Eliminación

Desprovisionamiento (a través de bloque de rescate):

- Elimina el servicio de todos los sistemas

- Libera inventario
- Cancela cargos recurrentes
- Deja un rastro de auditoría
- Enfoque recomendado

Eliminación Directa (a través de API):

- Solo elimina el registro de CRM
- NO limpia la cuenta OCS
- NO libera inventario
- Puede dejar recursos huérfanos
- No recomendado para producción

Reversión y Manejo de Errores

La función de **bloque/rescate** de Ansible se emplea durante el aprovisionamiento y desprovisionamiento para manejar errores de manera elegante. Si una tarea falla en cualquier punto durante el aprovisionamiento, la sección de rescate revierte automáticamente los cambios para volver a un estado consistente. Esto asegura confiabilidad y reduce el riesgo de implementaciones parciales o fallidas.

Ejemplo de Manejo de Errores

```yaml
- name: Bloque principal
  block:
    - name: Paso 1: Crear cuenta
      uri: ...
      register: response

    - name: Verificar paso 1
      assert:
        that:
          - response.status == 200

    - name: Paso 2: Crear servicio
      uri: ...

  rescue:
    # Si alguna afirmación falla o cualquier tarea da error:
    # 1. La ejecución salta al bloque de rescate
    # 2. Se ejecutan tareas de limpieza
    # 3. El playbook falla con un mensaje de error
    - name: Limpiar cambios parciales
      uri: ...
```

Para obtener detalles completos sobre el sistema de aprovisionamiento, flujos de trabajo y autenticación, consulte `concepts_provisioning`.

OmniCRM API

Todas las funciones dentro de OmniCRM son accesibles a través de la API - No hay funcionalidad que esté disponible solo en la interfaz de usuario.

Esto te permite integrar OmniCRM con otros sistemas o automatizar tareas.

La API es una API RESTful y está asegurada utilizando múltiples métodos de autenticación, incluidos tokens JWT, claves API y listas blancas de IP.

La documentación de la API está documentada usando Swagger, una herramienta que permite una fácil lectura, comprensión y prueba de la funcionalidad de la API.

La documentación de la API está disponible en la siguiente URL:

> <https://yourcrm/crm/docs/>

Métodos de Autenticación

OmniCRM soporta tres métodos de autenticación, cada uno diseñado para diferentes casos de uso:

1. **Tokens Bearer JWT** - Para sesiones de usuario interactivas (interfaz web, aplicaciones móviles)
2. **Claves API** - Para integraciones de servidor a servidor y scripts de automatización
3. **Lista Blanca de IP** - Para sistemas internos de confianza (servidores de aprovisionamiento, herramientas de monitoreo)

Autenticación con Token Bearer JWT

Este es el método de autenticación principal para las sesiones de usuario. Los usuarios inician sesión con su correo electrónico y contraseña, reciben un token JWT y lo utilizan para solicitudes posteriores.

Casos de Uso:

- Autenticación de interfaz web
- Autenticación de aplicaciones móviles
- Acceso programático de corta duración

Cómo Autenticarse:

Para iniciar sesión, envía un cuerpo JSON con la siguiente estructura a `/crm/auth/login` como una solicitud POST:

```
{
    "email": "youruser@yourdomain.com",
    "password": "yourpassword"
}
```

La API devolverá un objeto JSON que contiene un campo `token`, que se utiliza para autenticar todas las solicitudes futuras. Además, la respuesta incluye un `refresh_token` que se puede usar para refrescar el token cuando expire, junto con los permisos y roles del usuario.

Puedes probar esto desde la página de Swagger seleccionando el endpoint `/auth/login`, completando tu Nombre de Usuario y Contraseña, y haciendo clic en el botón `Try it out`.

Para autorizar la sesión, copia el valor del token y haz clic en el botón "Authorize" en la parte superior derecha de la página de Swagger. Pega el token en el campo "Value", precedido por `Bearer` y haz clic en "Authorize".

Ahora, todas las solicitudes posteriores estarán autenticadas con este token.

Autenticación con Clave API

Las claves API proporcionan autenticación segura y de larga duración para integraciones de servidor a servidor y scripts de automatización sin requerir contraseñas de usuario.

Casos de Uso:

- Sistemas de aprovisionamiento automatizados
- Herramientas de monitoreo y alerta
- Integración con sistemas externos
- Tareas programadas y trabajos cron

Cómo Funcionan las Claves API:

Las claves API se configuran en el archivo `crm_config.yaml` y están asociadas con roles y permisos específicos. Cada clave API es una cadena aleatoria segura (mínimo 32 caracteres) que autentica las solicitudes cuando se pasa en el encabezado `X-API-KEY`.

Configurando Claves API:

Las claves API deben ser añadidas a `crm_config.yaml` por un administrador con acceso al servidor:

```yaml
api_keys:
  your-secure-api-key-here-minimum-32-chars:
    roles:
      - admin
    description: "Sistema de automatización de aprovisionamiento"
  another-api-key-for-monitoring-system:
    roles:
      - view_customer
      - view_service
    description: "Monitoreo y alerta"
```

Usando Claves API:

Incluye la clave API en el encabezado `X-API-KEY` de tus solicitudes:

```
curl -X GET "https://yourcrm.com/crm/customers" \
  -H "X-API-KEY: your-secure-api-key-here-minimum-32-chars"
```

Ejemplo en Python:

```python
import requests

crm_url = 'https://yourcrm.com'
api_key = 'your-secure-api-key-here-minimum-32-chars'

headers = {
    "Content-Type": "application/json",
    "X-API-KEY": api_key
}

# Obtener Clientes
response = requests.get(crm_url + '/crm/customers',
headers=headers)
for customer in response.json()['data']:
    print(customer)
```

Mejores Prácticas:

- Genera claves API utilizando generadores aleatorios seguros criptográficamente (`openssl rand -base64 48`)
- Usa diferentes claves API para diferentes sistemas
- Documenta el propósito de cada clave API en el campo `description`
- Rota las claves API periódicamente
- Nunca comités claves API en control de versiones
- Asigna los permisos mínimos necesarios a cada clave API

Autenticación con Lista Blanca de IP

La lista blanca de IP permite que direcciones IP específicas accedan a la API sin autenticación. Esto es útil para sistemas internos de confianza en redes privadas.

Casos de Uso:

- Servidores de aprovisionamiento internos
- Sistemas de monitoreo de red en VLANs de gestión
- Playbooks de Ansible ejecutándose en infraestructura controlada

Configurando la Lista Blanca de IP:

Agrega direcciones IP de confianza a `crm_config.yaml`:

```
ip_whitelist:
  - 192.168.1.100
  - 10.0.0.0/24
  - 172.16.50.10
```

Consideraciones de Seguridad:

- Usa la lista blanca de IP solo en redes privadas y seguras
- Nunca incluyas direcciones IP públicas en la lista blanca
- Usa los rangos de IP más específicos posibles
- Documenta por qué cada IP está en la lista blanca

- Audita regularmente las IPs en la lista blanca

Ejemplo de Llamadas a la API con Python

Aquí hay un ejemplo de cómo iniciar sesión y recuperar una lista de clientes utilizando autenticación con token JWT:

```
import requests

crm_url = 'https://yourcrm.com'
session = requests.Session()

print("Aprovisionando datos al servidor: " + str(crm_url))

headers = {
    "Content-Type": "application/json"
}

# Obtener Token de Autenticación
response = session.post(crm_url + '/crm/auth/login', json={
    "email": "youruser@yourdomain.com",
    "password": "yourpassword"
}, headers=headers)

print(response.status_code)
print(response.json())
assert response.status_code == 200

headers['Authorization'] = 'Bearer ' + response.json()['token']
print("Autenticado en CRM exitosamente")

# Obtener Clientes
response = session.get(crm_url + '/crm/customers',
headers=headers)
for customer in response.json()['data']:
    print(customer)
```

Monitoreo del Rendimiento de la API

Todas las solicitudes a la API son rastreadas automáticamente con métricas completas. Para detalles sobre las métricas de rendimiento de la API, incluyendo tasas de solicitud, tiempos de respuesta, tasas de error y estadísticas específicas de endpoints, consulta Monitoreo & Métricas.

El endpoint de métricas está disponible en `/crm/metrics` en formato Prometheus para monitoreo y alerta.

Documentación Relacionada

- Arquitectura del Sistema - Diseño general del sistema y relaciones de componentes
- Monitoreo & Métricas - Referencia completa de métricas incluyendo API, base de datos y métricas de integración
- Flujos de Autenticación - Flujos detallados de autenticación y autorización
- RBAC - Control de acceso basado en roles y permisos

Notas de Carga de Productos / Servicios de CRM

> **NOTA**
>
> Para una guía completa de extremo a extremo que cubre la definición de productos, provisionamiento, complementos y desprovisionamiento con ejemplos detallados de Ansible y estrategia de precios, consulte la Guía Completa del Ciclo de Vida del Producto.
>
> Para obtener información detallada sobre el provisionamiento de servicios móviles con tarjetas SIM, consulte Provisionamiento de Tarjetas SIM.

Visión General de Productos y Servicios

Producto (Elemento del Menú):

Un Producto es como un plato específico en el menú de un restaurante, como un "Espagueti a la Carbonara."

Tiene una descripción clara, una lista de ingredientes (como pasta, crema, huevos, queso y tocino), y un precio.

En OmniCRM, un Producto contiene de manera similar los detalles de lo que está incluido: características, especificaciones y precios.

A menudo, los clientes pueden querer modificaciones, como "sin cebollas" o "agregar queso extra" a su comida. Dentro de OmniCRM, esto corresponde a

personalizar un servicio antes de su creación. El nivel de personalizaciones o modificaciones a un servicio depende de usted (el operador) definir.

En OmniCRM, los clientes o el personal pueden modificar un Producto para adaptarse mejor a las necesidades de un cliente específico, como actualizar su velocidad de Internet o agregar características específicas. Esta personalización se refleja en el Servicio específico proporcionado.

Un producto es esencialmente una oferta de la que los clientes pueden elegir ordenar, similar a leer y elegir un plato del menú.

Catálogo de Productos (Menú del Restaurante):

El Catálogo de Productos es como el menú completo en un restaurante, que lista todos los platos disponibles, desde aperitivos hasta postres.

Es la colección completa de todo lo que el restaurante (o en su caso, el Proveedor de Servicios) tiene para ofrecer.

En el contexto empresarial, el Catálogo de Productos proporciona a los clientes todos los Productos disponibles, para que puedan elegir el que mejor se adapte a sus necesidades.

Servicio (Plato Preparado):

Cuando un cliente ordena un artículo del menú, el plato se prepara en la cocina. Esto es similar a crear un Servicio a partir de un Producto.

En OmniCRM, cuando un cliente selecciona un Producto, se crea una instancia de ese Producto y se entrega como un Servicio.

Se personaliza y prepara específicamente para ese cliente, al igual que una comida preparada para un comensal.

Por ejemplo, cuando alguien selecciona el "Plan de Internet Bronce" del Catálogo de Productos, el sistema de aprovisionamiento "cocina" una instancia de ese plan a partir de los ingredientes (direcciones IP, módems y puertos): es decir, activa el plan y lo entrega al cliente específico.

Productos Agrupados (Comidas Combinadas):

El Catálogo de Productos también puede ofrecer paquetes, como una comida combinada que incluye un aperitivo, plato principal y postre juntos por un precio especial.

En OmniCRM, los Productos agrupados combinan múltiples Productos individuales en un solo paquete conveniente, como un "Paquete de Esenciales para el Hogar" que incluye servicios de Internet, cable y teléfono a un precio con descuento.

Una vez seleccionado, este paquete se convierte en múltiples Servicios adaptados para el cliente.

Definiciones de Producto

Un producto es una plantilla que se utiliza para crear un servicio / complemento / descuento / agregado, etc.

Dentro de la definición incluimos:

- Información sobre el producto (características, inclusiones, Términos y Condiciones, duración del contrato, icono, etc.) que se muestra al usuario del CRM (Cliente o *Personal*).
- La lógica empresarial sobre quién puede comprar el producto (*Empresarial* o *Residencial*), si depende de tener un servicio principal provisionado (como complementos móviles solo disponibles para clientes con un servicio móvil), si puede ser ordenado directamente por un cliente a través del autoservicio o solo por un agente de servicio al cliente, y cuándo se puede

comprar el producto (Permitiendo que un producto esté disponible solo por un tiempo determinado).

- Cuando se deben incluir elementos de inventario (como módems o tarjetas SIM), estos se especifican como Lista de Elementos de Inventario, por ejemplo, el siguiente servicio requiere que se asigne una tarjeta SIM y un número de teléfono:

  `['SIM Card', 'Phone Number']` Estos se correlacionan con los Elementos de Inventario definidos en el CRM.

- Referenciar un Playbook de Ansible para aprovisionar el servicio Provisioning Play así como las variables a pasar a Ansible. Estas variables a pasar son mágicas, en el sentido de que pueden ser variables como `service_id` que están definidas por el producto al que estamos agregando, o pueden ser como `ICCID` & `MSISDN` donde hemos seleccionado elementos de inventario que se pasan al asignar el inventario. El agrupamiento se maneja en el juego de aprovisionamiento para contener múltiples servicios, por ejemplo, un producto de Internet, TV y Voz agrupado, puede aprovisionar un servicio para cada uno.

Categorías de Productos y Tipos de Servicio

Los productos utilizan dos campos de clasificación para ayudar a organizar y filtrar ofertas:

Categorías de Productos

El campo `category` controla dónde se muestran los productos en la interfaz de usuario. Los valores comunes incluyen:

- **standalone** - Se muestra como una opción de servicio base al crear un nuevo servicio
- **addon** - Se muestra al agregar a un servicio existente
- **bundle** - Se muestra como una opción de servicio agrupado (provisionado como un complemento a servicios existentes)
- **promo** - Ofertas promocionales especiales

Estas categorías son puramente organizativas y no dictan lo que se aprovisiona. El comportamiento real de aprovisionamiento está determinado completamente por el playbook de Ansible referenciado en `provisioning_play`.

Por ejemplo: - Un producto `standalone` típicamente crea un nuevo objeto de servicio - Un producto `addon` o `bundle` típicamente se agrega a un servicio existente - Pero esto depende del implementador que escribe el playbook - podría crear múltiples objetos de servicio a partir de un complemento, o modificar servicios existentes a partir de un producto independiente si es necesario

La categoría simplemente controla el flujo de la interfaz de usuario y dónde los clientes/el personal ven la opción del producto.

Tipos de Servicio

El campo `service_type` categoriza qué tipo de servicio se está proporcionando.

Estos son completamente definidos por el usuario, pero los valores comunes incluyen:

- **mobile** - Servicios de teléfono móvil con voz, SMS y datos
- **fixed** - Servicios de internet fijos inalámbricos o por cable
- **fixed-voice** - Servicios de voz de línea fija (VoIP, línea terrestre)
- **hotspot** - Dispositivos móviles de hotspot o de alquiler
- **dongle** - Servicios de módem USB o dongle
- **voice** - Servicios solo de voz
- **data** - Servicios solo de datos

Al igual que las categorías, los tipos de servicio son personalizables según sus ofertas. Ayudan en:

- Filtrar qué complementos se aplican a qué servicios base
- Organizar productos en el portal del cliente
- Coincidir con los requisitos de inventario
- Determinar flujos de trabajo de aprovisionamiento

Ejemplo: Un cliente con un servicio `mobile` puede ver complementos móviles, mientras que un cliente con un servicio `fixed` ve complementos de línea fija.

Gestión de Productos

Los productos se gestionan a través de la página de Gestión de Productos, donde puede ver, buscar, filtrar y editar todos los productos disponibles.

Interfaz Modal de Producto

Hacer clic en cualquier producto abre una interfaz de pestañas mejorada que organiza todas las configuraciones del producto en grupos lógicos para facilitar la navegación y edición.

El modal de gestión de productos cuenta con cinco pestañas organizadas:

1. □ **Información Básica** - Información central del producto (nombre, slug, categoría, icono, características, términos)
2. □ **Precios** - Todos los campos relacionados con costos, incluidos costos recurrentes, costos de configuración y porcentaje de impuestos
3. ⚙ **Configuración** - Configuraciones de renovación, tipos de clientes y dependencias
4. □ **Aprovisionamiento** - Configuración del playbook de Ansible y requisitos de inventario
5. □ **Disponibilidad** - Rangos de fechas y marcas de tiempo del sistema

Organización de la Pestaña de Precios:

La pestaña de Precios agrupa los campos de costo en secciones lógicas:

- **Costos Recurrentes** - Costos minoristas y mayoristas mensuales uno al lado del otro
- **Costos de Configuración** - Tarifas de activación únicas para minoristas y mayoristas
- **Impuesto** - Configuración del porcentaje de impuestos con cálculo automático

Características del Modo de Edición:

- **Selector de Iconos** - Buscar y seleccionar iconos de FontAwesome visualmente
- **Selector de Elementos de Inventario** - Seleccionar entre tipos de elementos de inventario disponibles
- **Selector de Fecha/Hora** - Selección fácil de ventanas de disponibilidad
- **Formato de Moneda** - Prefijo automático $ para campos de costo
- **Selectores Desplegables** - Opciones predefinidas para categorías y campos booleanos

Selector de Iconos:

Al editar el campo de icono, aparece una interfaz de selector de iconos buscable que le permite navegar y seleccionar visualmente entre miles de iconos de FontAwesome.

Características: * Buscar iconos por palabra clave (por ejemplo, "llave", "móvil", "wifi") * Previsualizar la apariencia del icono en tiempo real * Muestra el nombre de la clase del icono como referencia * Selección de desplegable para acceso rápido

Pestaña de Configuración:

La pestaña de Configuración organiza las configuraciones del comportamiento del producto en grupos lógicos.

Secciones de Configuración:

- **Configuraciones de Renovación:**
  - Auto Renovar - Comportamiento de renovación predeterminado (Solicitar/Sí/No)
  - Permitir Auto Renovar - Si los clientes pueden habilitar la renovación automática
  - Días de Contrato - Duración mínima del contrato (por ejemplo, 30 para mensual, 365 para anual)
- **Tipos de Clientes:**
  - Residencial - Disponible para clientes consumidores
  - Empresarial - Disponible para clientes comerciales
- **Dependencias:**
  - Depende de la Lista - IDs de productos o tipos de servicio requeridos antes de que este producto pueda ser agregado
  - Usado para dependencias de complementos (por ejemplo, los complementos móviles requieren un servicio móvil activo)

Pestaña de Aprovisionamiento:

La pestaña de Aprovisionamiento maneja la automatización de Ansible y los requisitos de inventario.

Campos de Aprovisionamiento:

- **Play de Aprovisionamiento:**
  - Nombre del playbook de Ansible (sin extensión .yaml)
  - Debe existir en el directorio OmniCRM-API/Provisioners/plays/
  - Se llama cuando se crea, actualiza o desprovisiona el servicio
- **Variables JSON de Aprovisionamiento:**
  - Variables predeterminadas pasadas al playbook de Ansible como JSON
  - Pueden ser sobrescritas durante el aprovisionamiento
  - El playbook recibe estas más customer_id, product_id, service_id, access_token
- **Lista de Elementos de Inventario:**
  - Selector de múltiples elementos que muestra tipos de elementos de inventario disponibles
  - Ejemplos: Tarjeta SIM, Número de Teléfono, Módem Router, Dirección IPv4
  - El cliente/el personal selecciona elementos específicos del inventario disponible durante el pedido
  - Los IDs de inventario seleccionados se pasan al playbook con el tipo de inventario como nombre de variable

Pestaña de Disponibilidad:

La pestaña de Disponibilidad controla cuándo se puede comprar el producto y muestra metadatos del sistema.

Configuraciones de Disponibilidad:

- **Disponible Desde:**
  - Fecha/hora cuando el producto se vuelve disponible para la compra
  - Dejar vacío para disponibilidad inmediata
  - Útil para pre-anunciar nuevos productos
- **Disponible Hasta:**
  - Fecha/hora cuando el producto ya no está disponible para la compra
  - Dejar vacío para disponibilidad indefinida
  - Perfecto para promociones por tiempo limitado o productos al final de su vida útil
- **Metadatos del Sistema (Solo Lectura):**
  - Creado - Marca de tiempo cuando se creó el producto por primera vez
  - Última Modificación - Marca de tiempo de la actualización más reciente
  - Mantenido automáticamente por el sistema

Acciones del Modal:

- **Modo de Vista:**
  - Cerrar - Descartar modal
  - Clonar Producto - Crear una copia con el sufijo "_clone"
  - Editar Producto - Cambiar a modo de edición
- **Modo de Edición/Creación:**
  - Cancelar - Descartar cambios y cerrar
  - Guardar Cambios - Crear o actualizar producto (botón grande para énfasis)

Campos del Producto

El modelo de Producto contiene toda la información necesaria para definir una oferta y cómo debe ser aprovisionada. Estos campos se gestionan a través de la interfaz modal de gestión de productos descrita anteriormente.

Información Básica

- **product_id** - Identificador único asignado automáticamente por el sistema

- **product_name** - Nombre de visualización mostrado a los clientes y al personal en la interfaz de usuario
- **product_slug** - Identificador único utilizado en URLs y llamadas a la API (minúsculas, sin espacios, usar guiones)
- **category** - Controla dónde aparece este producto en la interfaz de usuario:
  - `standalone` - Se muestra como una opción de servicio base al crear un nuevo servicio
  - `addon` - Se muestra al agregar a un servicio existente
  - `bundle` - Se muestra como una opción de servicio agrupado
  - `promo` - Ofertas promocionales especiales
- **service_type** - Tipo de servicio que se proporciona (por ejemplo, móvil, fijo, voz fija, hotspot, dongle, voz, datos). Se utiliza para filtrar qué complementos se aplican a qué servicios.
- **comment** - Notas internas sobre el producto para referencia del personal solamente (no se muestra a los clientes)
- **icon** - Clase de icono de FontAwesome mostrada en la interfaz de usuario (por ejemplo, `fa-solid fa-sim-card`)

Campos de Precios

- **retail_cost** - Cargo recurrente mensual facturado al cliente (establecer en 0 para compras únicas o productos prepagados)
- **wholesale_cost** - Su costo mensual por proporcionar este servicio (utilizado para cálculos de margen)
- **retail_setup_cost** - Tarifa de activación o configuración única cobrada al cliente
- **wholesale_setup_cost** - Su costo único por configurar el servicio
- **tax_percentage** - Porcentaje de impuestos aplicado a este producto (por ejemplo, 10 para 10%, 12.5 para 12.5%). Establecer en 0 para productos exentos de impuestos. Esta tasa de impuestos se aplica automáticamente a las transacciones creadas a partir de este producto.

Aplicación de Impuestos:

Cuando se crea una transacción a partir de este producto, el porcentaje de impuestos se copia automáticamente a la transacción y se calcula el monto del impuesto. Por ejemplo:

- Producto con 10% de impuesto, $50.00 costo minorista → Transacción tiene $5.00 de impuesto
- Producto con 0% de impuesto (exento de impuestos) → Transacción tiene $0.00 de impuesto
- Anulación manual de transacción → El personal puede cambiar el porcentaje de impuestos por transacción

Visibilidad y Acceso del Cliente

- **enabled** - Si este producto está activo y disponible para compra (establecer en falso para ocultar sin eliminar)
- **residential** - Si los clientes residenciales (consumidores) pueden comprar este producto
- **business** - Si los clientes empresariales (comerciales) pueden comprar este producto
- **customer_can_purchase** - Si los clientes pueden auto-comprar a través del portal (verdadero) o si solo el personal puede agregarlo (falso)
- **available_from** - Fecha/hora cuando este producto se vuelve disponible para compra (opcional)

- **available_until** - Fecha/hora cuando este producto ya no está disponible para compra (opcional, útil para ofertas por tiempo limitado)

Contrato y Renovación

- **contract_days** - Duración mínima del contrato en días (por ejemplo, 30 para mensual, 365 para anual, 0 para sin contrato mínimo)
- **auto_renew** - Comportamiento de renovación predeterminado:
  - `prompt` - Pregunta al cliente cada vez si desea renovar
  - `true` - Renueva automáticamente sin preguntar
  - `false` - Requiere renovación manual
- **allow_auto_renew** - Si los clientes pueden habilitar la renovación automática (establecer en falso para compras únicas)

Contenido Visible para el Cliente

- **terms** - Términos y condiciones mostrados a los clientes antes de la compra (incluir limitaciones, reglas de caducidad, condiciones de uso)
- **features_list** - Lista de características e inclusiones mostradas a los clientes (formato de lista de Python: `['Feature 1', 'Feature 2']`)

Configuración de Aprovisionamiento

- **provisioning_play** - Nombre del playbook de Ansible que aprovisiona este servicio (sin extensión .yaml). Debe existir en `OmniCRM-API/Provisioners/plays/` directorio.
- **provisioning_json_vars** - Variables predeterminadas pasadas al playbook de Ansible como JSON. Estas pueden ser sobrescritas al aprovisionar. El playbook recibe estas junto con `customer_id`, `product_id`, `service_id`, y `access_token`.
- **inventory_items_list** - Lista de elementos de inventario requeridos para este producto (por ejemplo, `['SIM Card', 'Mobile Number']`). Cuando un cliente realiza un pedido, se le pedirá que seleccione elementos específicos del inventario disponible. Los IDs de inventario seleccionados se pasan al playbook con el tipo de inventario como el nombre de la variable.
- **relies_on_list** - Lista de IDs de productos o tipos de servicio que deben existir antes de que este producto pueda ser agregado. Usado para

dependencias de complementos (por ejemplo, los complementos móviles requieren un servicio móvil activo).

Metadatos del Sistema

- **created** - Marca de tiempo cuando se creó el producto (establecido automáticamente)
- **last_modified** - Marca de tiempo cuando se actualizó el producto por última vez (actualizado automáticamente)

Ejemplos de Definiciones de Productos

Producto Independiente (SIM Móvil)

```
{
  "product_id": 1,
  "product_slug": "Mobile-SIM",
  "product_name": "Mobile SIM Only",
  "category": "standalone",
  "service_type": "mobile",
  "provisioning_play": "play_psim_only",
  "provisioning_json_vars": "{\"iccid\": \"\", \"msisdn\": \"\"}",
  "inventory_items_list": "['SIM Card', 'Mobile Number']",
  "retail_cost": 0,
  "retail_setup_cost": 0,
  "wholesale_cost": 3,
  "wholesale_setup_cost": 1,
  "contract_days": 0,
  "residential": true,
  "business": true,
  "enabled": true,
  "customer_can_purchase": true,
  "icon": "fa-solid fa-sim-card",
  "features_list": "['Australian Phone Number (04xxx)', 'Fastest speeds', 'Best coverage', 'Roaming on the Mainland']",
  "terms": "Must be activated within 6 months. All credit lost if service is not used for 12 months.",
  "comment": "Physical SIM card for use with Mobile Phones"
}
```


Este producto independiente requiere dos elementos de inventario (Tarjeta SIM y Número Móvil) y crea un nuevo servicio cuando se aprovisiona.

Producto Complemento (Plan de Datos Mensual)


```
{
  "product_slug": "example-mobile-prepaid-mini",
  "product_name": "Example Mobile Mini Plan",
  "category": "addon",
  "service_type": "mobile",
  "provisioning_play": "play_topup_charge_then_action",
  "provisioning_json_vars": "",
  "inventory_items_list": "[]",
  "retail_cost": 30,
  "retail_setup_cost": 0,
  "wholesale_cost": 5.84,
  "contract_days": 30,
  "residential": true,
  "business": false,
  "enabled": true,
  "customer_can_purchase": true,
  "auto_renew": "prompt",
  "icon": "fa-solid fa-sim-card",
  "features_list": "['8GB of Ultra fast data', 'Unlimited Calls & Texts to Example Mobile users', '100 Minutes of Talk to Australia', '100 SMS to Australia', '30 Day Expiry']",
  "terms": "Credit expires after 30 days. Once data, voice or sms is used up, you will need to top up to continue using the service.",
  "comment": "Our smallest plan for light users"
}
```


Este producto complemento no requiere inventario y se aplica a un servicio existente. Cobra al cliente y agrega créditos/saldos a su servicio.

Producto Agrupado (Paquete para Mayores)


```
{
  "product_slug": "Bundle-Seniors",
  "product_name": "Seniors Bundle",
  "category": "bundle",
  "service_type": "fixed",
  "provisioning_play": "play_seniors_package",
  "provisioning_json_vars": "{\"IPTV_Service_ID\":
\"SeniorBundle\"}",
  "inventory_items_list": "['Modem Router']",
  "retail_cost": 30,
  "retail_setup_cost": 0,
  "wholesale_cost": 10,
  "wholesale_setup_cost": 11,
  "contract_days": 180,
  "residential": true,
  "business": false,
  "enabled": true,
  "icon": "fa-solid fa-person-walking-with-cane",
  "features_list": "['20Mbps Download', '5Mbps Upload', 'Unlimited
Data', 'Home Voice', 'TV: Extra +£5 per month', '£60 Installation
Fee']",
  "terms": "6 Month Contract, must show senior citizen's card to
qualify",
  "comment": "20Mbps/2Mbps GPON Service + IPTV + Phone"
}
```


Este producto agrupado aprovisiona múltiples servicios (Internet + IPTV + Teléfono) utilizando un solo playbook. Requiere un elemento de inventario (Módem Router).

Producto Complemento (Recarga Simple)


```
{
  "product_slug": "Mobile-Topup-5",
  "product_name": "PAYG £5 Topup",
  "category": "addon",
  "service_type": "mobile",
  "provisioning_play": "play_topup_monetary",
  "provisioning_json_vars": "{\"service_id\": \"\"}",
  "inventory_items_list": "[]",
  "retail_cost": 5,
  "retail_setup_cost": 0,
  "wholesale_cost": 0,
  "contract_days": 0,
  "residential": true,
  "business": false,
  "enabled": true,
  "customer_can_purchase": true,
  "icon": "fa-solid fa-coins",
  "features_list": "['£5 credit', 'Valid for 180 days']",
  "terms": "Valid for 180 days or until all credit is used. See
our website for full rates",
  "comment": "£5 to use for Calls, SMS & Data"
}
```


Este complemento simplemente agrega crédito monetario a un servicio existente. No se requiere inventario, y utiliza `service_id` para identificar qué servicio recargar.

Cómo se Pasan las Variables a Ansible

Entender cómo fluyen las variables desde la definición del producto a través de la API hasta el playbook de Ansible es crítico para escribir playbooks de aprovisionamiento efectivos.

Fuentes de Variables y Fusión

Cuando se crea un trabajo de aprovisionamiento, las variables provienen de múltiples fuentes y se fusionan en este orden (las fuentes posteriores sobrescriben las anteriores):

1. **Variables de provisioning_json_vars del Producto** - Variables predeterminadas de la definición del producto
2. **Cuerpo de la Solicitud** - Variables pasadas en la llamada a la API (pueden sobrescribir los valores predeterminados del producto)
3. **Variables Agregadas por el Sistema** - Agregadas automáticamente por el sistema de aprovisionamiento
4. **Selecciones de Inventario** - IDs de elementos de inventario seleccionados (si `inventory_items_list` no está vacío)

Proceso de Fusión de Variables

El sistema fusiona variables de todas las fuentes, con las fuentes posteriores sobrescribiendo las anteriores. Esto permite una personalización flexible en el momento del aprovisionamiento.

Por ejemplo, si su producto tiene:

```
"provisioning_json_vars": "{\"monthly_cost\": 50, \"data_gb\": 100}"
```

Y su solicitud de API incluye:

```
{
  "product_id": 10,
  "customer_id": 456,
  "monthly_cost": 45,
  "custom_param": "value"
}
```

Las `extra_vars` finales pasadas a Ansible serán:

```
{
    "monthly_cost": 45,         # Sobrescrito desde la solicitud
    "data_gb": 100,             # Desde provisioning_json_vars
    "product_id": 10,           # Desde la solicitud
    "customer_id": 456,         # Desde la solicitud
    "custom_param": "value",    # Desde la solicitud
    "access_token": "eyJ..."    # Agregado por el sistema
}
```

Variables Agregadas por el Sistema

El sistema de aprovisionamiento agrega automáticamente:

- `access_token` - Token JWT para autenticar llamadas a la API de vuelta al CRM (proporcionado directamente para autenticación IP/API key, o generado a partir de `refresh_token` para autenticación de usuario)
- `initiating_user` - El ID de usuario que activó el aprovisionamiento (o primer administrador para sistemas automatizados)
- Cualquier campo del cuerpo de la solicitud (`product_id`, `customer_id`, `service_id`, etc.)

Variables de Inventario

Cuando un producto requiere elementos de inventario (por ejemplo, `inventory_items_list: "['SIM Card', 'Mobile Number']"`), el proceso funciona de la siguiente manera:

1. **UI/API solicita selección** - El usuario selecciona elementos de inventario específicos del stock disponible
2. **IDs de Inventario se agregan a las variables** - Los IDs de los elementos de inventario seleccionados se agregan con el tipo de inventario como el nombre de la variable
3. **El playbook accede a los IDs de inventario** - El playbook de aprovisionamiento puede recuperar detalles completos del inventario desde la API del CRM

Por ejemplo, si un usuario selecciona: - Tarjeta SIM con inventory_id: 789 - Número Móvil con inventory_id: 101

Las variables pasadas al playbook incluyen: - `SIM Card`: 789 - `Mobile Number`: 101

El playbook puede entonces usar estos IDs para obtener los registros completos de inventario (ICCID, IMSI, MSISDN, etc.) desde la API del CRM y usar esa información para aprovisionar el servicio en el equipo de red.

Cómo Ansible Recibe Variables

El sistema de aprovisionamiento pasa todas las variables fusionadas al playbook de Ansible como `extravars`. Dentro del playbook, estas variables están disponibles a través del sistema de variables estándar de Ansible y pueden ser usadas en tareas.

Las variables pueden ser referenciadas directamente en las tareas del playbook usando la sintaxis `{{ variable_name }}`. Por ejemplo, `{{ product_id }}`, `{{ customer_id }}`, `{{ monthly_cost }}`, etc.

Variables Pasadas a Productos Complementarios

Cuando se aprovisiona un producto complemento, el sistema pasa automáticamente:

- `product_id` - El ID del producto complemento que se está aprovisionando
- `customer_id` - El cliente que posee el servicio
- `service_id` - El ID del servicio al que se está agregando este complemento (crítico para complementos)
- `access_token` - Token de autenticación para llamadas a la API
- Cualquier variable de `provisioning_json_vars`
- Cualquier variable adicional de la solicitud de API

Ejemplo de Flujo de Aprovisionamiento de Complementos

Cuando un cliente agrega el complemento "Recarga de £5" a su servicio móvil (service_id: 123), el playbook recibe variables que incluyen:

- `product_id`: 45 (el producto de recarga)
- `customer_id`: 456 (el cliente)
- `service_id`: 123 (el servicio al que agregar crédito)
- `access_token`: Token de autenticación
- Además de cualquier variable de `provisioning_json_vars` del producto

El playbook luego usa estas variables para:

1. **Obtener detalles del servicio** desde la API del CRM usando el `service_id`
2. **Extraer el UUID del servicio** y otra información del registro del servicio
3. **Agregar crédito al sistema de facturación** (OCS) usando el UUID del servicio
4. **Registrar la transacción** en el CRM para fines de facturación

Este flujo permite que el complemento identifique exactamente qué servicio modificar y aplicar los cambios adecuadamente.

Diferencia entre Variables de Productos Independientes y Complementos

Productos Independientes reciben:

- `product_id` - El producto que se está aprovisionando
- `customer_id` - El cliente que ordena el servicio
- IDs de elementos de inventario (por ejemplo, `SIM Card`, `Mobile Number`) si el producto los requiere
- `access_token` - Para autenticación de API

Productos Complementos reciben:

- `product_id` - El producto complemento que se está aprovisionando
- `customer_id` - El cliente que posee el servicio
- `service_id` - **El ID del servicio existente a modificar** (esta es la diferencia clave)
- `access_token` - Para autenticación de API

La diferencia clave es `service_id` - esto le dice al playbook qué servicio existente modificar o agregar.

Productos Agrupados

Los productos agrupados se aprovisionan como complementos, pero su playbook puede crear múltiples registros de servicio. Reciben las mismas variables que los complementos, incluyendo:

- `product_id` - El producto agrupado
- `customer_id` - El cliente
- `service_id` - Servicio principal (si corresponde)
- IDs de elementos de inventario (por ejemplo, `Modem Router`) si es necesario
- `access_token` - Para autenticación de API

El playbook de agrupamiento (por ejemplo, `play_seniors_package`) luego crea múltiples servicios relacionados (Internet, IPTV, Teléfono) y los vincula juntos.

Servicios

Un servicio es una instancia de un producto que pertenece a un cliente, por la que se le factura.

Es esencialmente un enlace a una cuenta de `OCS` (Sistema de Carga En Línea) que maneja la generación de cargos y los saldos y usos reales de la cuenta. El OCS es potenciado por CGRateS y gestiona saldos monetarios, saldos unitarios (datos, voz, SMS), Planes de Acción para renovación automática y ThresholdS para límites de gasto.

Agregar un Servicio: Selección y Filtrado de Productos

Al agregar un servicio a un cliente (ya sea un nuevo servicio independiente o un complemento a un servicio existente), el sistema muestra productos disponibles en una interfaz de carrusel. Los productos mostrados se filtran según varios criterios:

Filtrado de Productos para Servicios Independientes

Al crear un nuevo servicio para un cliente, la interfaz de usuario muestra productos filtrados por:

1. **Tipo de Cliente** - Los productos se categorizan como:
   - **Individual (Residencial)**: Productos donde `residential = true` o `business = false`
   - **Empresarial**: Productos donde `business = true`
2. **Categoría** - Los productos se separan en:
   - **Planes de Servicio**: Productos con `category` = `standalone` o `bundle`
   - **Complementos**: Productos con `category` = `addon` (mostrados en un carrusel separado)
3. **Disponibilidad** - Los productos solo se muestran si:
   - `enabled = true` - El producto está activo y no deshabilitado
   - La fecha actual está entre `available_from` y `available_until` - El producto está dentro de su ventana de disponibilidad
   - `customer_can_purchase = true` (si el cliente está auto-comprando) - El producto permite la compra directa por parte del cliente

> ⓘ **NOTA**
>
> **Filtrado a Nivel de API**: La API filtra automáticamente los productos por estado habilitado y fechas de disponibilidad en dos niveles:
>
> - **Puntos finales de Compra/Selección** (`/crm/product/`) - Usado por el modal de Complementos y PlanList para la selección de productos.

Filtra automáticamente para mostrar SOLO productos habilitados dentro de su rango de fechas de disponibilidad. Esto asegura que los clientes y el personal solo puedan seleccionar productos que están actualmente disponibles para la compra.

- **Puntos finales de Gestión** (`/crm/product/paginated`) - Usado por la página de Gestión de Productos. Muestra TODOS los productos, incluyendo deshabilitados y fuera de fechas de disponibilidad, permitiendo a los administradores gestionar el catálogo completo de productos incluyendo productos inactivos.

Pase `include_disabled=true` al punto final base de productos para omitir el filtrado (solo para uso administrativo).

La interfaz de usuario muestra carruseles separados para:

- **Planes de Servicio Individuales** - Productos residenciales para clientes consumidores
- **Planes de Servicio Empresariales** - Productos comerciales para clientes empresariales
- **Complementos Individuales** - Paquetes de complementos residenciales
- **Complementos Empresariales** - Paquetes de complementos comerciales

Filtrado de Productos para Servicios Complementarios

Al agregar un complemento a un **servicio existente**, se aplica un filtrado adicional:

1. **Coincidencia de Tipo de Servicio** - Solo se muestran complementos con `service_type` coincidente:
   - Si el servicio existente tiene `service_type = "mobile"`, solo se muestran complementos con `service_type = "mobile"`
   - Esto asegura que los clientes móviles solo vean complementos móviles, los clientes de internet solo vean complementos de internet, etc.
2. **Verificación de Dependencias** - Si un complemento tiene una `relies_on_list`:
   - El sistema verifica si el cliente tiene los productos/servicios requeridos

    - Solo se muestran complementos cuyas dependencias están satisfechas
3. **Filtro de Mismo Tipo de Cliente** - Los complementos aún se filtran por `residential` vs `business` para coincidir con el tipo de cliente

Ejemplo de Escenario de Filtrado

Para un cliente empresarial con un servicio móvil existente (`service_type = "mobile"`):

- **Productos Independientes Mostrados**: Todos los productos independientes/combinados empresariales (`business = true`, `category != "addon"`)
- **Productos Complementarios Mostrados**: Solo complementos móviles empresariales (`business = true`, `category = "addon"`, `service_type = "mobile"`)
- **Productos Ocultos**: Productos residenciales, complementos para otros tipos de servicio (internet, voz, etc.), productos deshabilitados

Campos del Servicio

El modelo de Servicio contiene campos que rastrean la instancia del servicio provisionado y su relación con el cliente, el producto y el sistema de facturación.

Información Básica del Servicio

- **service_id** - Identificador único asignado automáticamente por el sistema (solo lectura)
- **customer_id** - Enlace al cliente que posee este servicio (solo lectura después de la creación)
- **product_id** - Enlace al producto del que se creó este servicio (solo lectura después de la creación)
- **service_name** - Nombre de visualización mostrado a los clientes (editable)
- **service_type** - Tipo de servicio: móvil, internet, voip, iptv, combinado, etc. (editable)

- **service_uuid** - Identificador único utilizado en OCS/CGRateS para facturación (solo lectura, autogenerado)
- **icon** - Clase de icono de FontAwesome para mostrar en el portal de autoservicio (editable)

Estado y Fechas del Servicio

- **service_status** - Estado actual: Activo, Inactivo, Suspendido, etc. (editable)
- **service_provisioned_date** - Cuando se aprovisionó el servicio por primera vez (establecido automáticamente, solo lectura)
- **service_active_date** - Cuando el servicio se volvió activo (editable)
- **service_deactivate_date** - Cuando el servicio caduca o será desactivado (editable)
- **contract_end_date** - Fecha de finalización del compromiso del contrato (editable)

Facturación y Precios

- **retail_cost** - Cargo recurrente mensual al cliente (editable)
- **wholesale_cost** - Su costo por proporcionar el servicio (editable)
- **service_billed** - Si este servicio aparece en las facturas (editable, predeterminado: verdadero)
- **service_taxable** - Si se aplican impuestos a este servicio (editable, predeterminado: verdadero)
- **invoiced** - Si el servicio ha sido facturado (establecido automáticamente por el sistema de facturación)
- **promo_code** - Código promocional utilizado cuando se creó el servicio (editable)

Visibilidad del Cliente

- **service_visible_to_customer** - Si el cliente puede ver este servicio en el portal de autoservicio (editable, predeterminado: verdadero)
- **service_usage_visible_to_customer** - Si el cliente puede ver detalles de uso/saldo (editable, predeterminado: verdadero)

Configuración de Aprovisionamiento

- **provisioning_play** - Playbook de Ansible utilizado para aprovisionar este servicio (heredado del producto, solo lectura)
- **provisioning_json_vars** - Variables pasadas al playbook de aprovisionamiento (heredadas del producto, solo lectura)
- **deprovisioning_play** - Playbook de Ansible que se ejecuta cuando el servicio es desprovisionado (solo lectura)
- **deprovisioning_json_vars** - Variables para el playbook de desprovisionamiento (solo lectura)

Relaciones del Servicio

- **bundled_parent** - Si este servicio es parte de un paquete, el service_id del servicio principal (solo lectura)
- **site_id** - Enlace al sitio/ubicación física donde se entrega el servicio (editable)

Notas y Metadatos

- **service_notes** - Notas internas sobre el servicio para referencia del personal (editable)
- **created** - Marca de tiempo cuando se creó el servicio (establecido automáticamente, solo lectura)
- **last_modified** - Marca de tiempo de la última actualización (actualizado automáticamente, solo lectura)

Campos Editables vs Solo Lectura

Editables a través de API/UI:

Los servicios pueden ser actualizados a través de `PATCH /crm/service/{service_id}` con estos campos:

- service_name, service_type, service_status
- service_notes
- retail_cost, wholesale_cost
- service_billed, service_taxable

- service_visible_to_customer, service_usage_visible_to_customer
- service_active_date, service_deactivate_date, contract_end_date
- icon, promo_code, site_id

Solo Lectura (Auto-Establecido):

Estos campos no pueden ser modificados directamente después de la creación:

- service_id, customer_id, product_id
- service_uuid (generado durante el aprovisionamiento)
- service_provisioned_date
- provisioning_play, provisioning_json_vars
- deprovisioning_play, deprovisioning_json_vars
- bundled_parent
- invoiced (gestionado por el sistema de facturación)
- created, last_modified (gestionados automáticamente)

Aprovisionamiento de Productos en Servicios

El proceso de aprovisionamiento convierte un Producto (plantilla) en un Servicio (instancia específica del cliente) a través de una serie de pasos coordinados que involucran la Interfaz Web, la API y los playbooks de Ansible.

Flujo de Aprovisionamiento de Alto Nivel

1. **Configuración Pre-Aprovisionamiento** - Producto creado en la API con configuración de aprovisionamiento, y los correspondientes playbooks de Ansible escritos y probados
2. **Selección de Servicio** - Desde la Página del Cliente, el personal o el cliente selecciona "Agregar Servicio"
3. **Filtrado de Productos** - Productos mostrados filtrados en base a:
   - Tipo de cliente (residencial/empresarial)
   - Servicios existentes (para dependencias de complementos en `relies_on_list`)

    - Fechas de disponibilidad (`available_from`/`available_until`)
    - Flags `enabled` y `customer_can_purchase`
4. **Personalización** - Opción para sobrescribir variables de aprovisionamiento (para ajustes de precios, configuraciones personalizadas, etc.)
5. **Selección de Inventario** - Si el producto requiere inventario (`inventory_items_list` no está vacío), el usuario selecciona elementos específicos (por ejemplo, qué tarjeta SIM, qué número de teléfono)
6. **Inicio de Aprovisionamiento** - Cuando se hace clic en el botón "Aprovisionar", la API crea un trabajo de aprovisionamiento

Flujo Detallado de Integración de API y Ansible

Cuando se aprovisiona un servicio, ocurre la siguiente secuencia:

Paso 1: Creación del Trabajo de Aprovisionamiento

La API recibe la solicitud de aprovisionamiento y crea un trabajo de aprovisionamiento con:

- `provisioning_play` - Nombre del playbook de Ansible (por ejemplo, `play_psim_only`)
- `provisioning_json_vars` - Cadena JSON de variables del producto o sobrescritas por la solicitud
- `customer_id` - ID del cliente que ordena el servicio
- `product_id` - ID del producto que se está aprovisionando
- `service_id` - (Opcional) ID del servicio existente para complementos
- Selecciones de inventario - IDs de elementos de inventario seleccionados

Paso 2: Ensamblaje de Variables

El servicio de aprovisionamiento fusiona variables de múltiples fuentes en este orden:

1. `provisioning_json_vars` del Producto (valores predeterminados de la definición del producto)
2. Parámetros del cuerpo de la solicitud (pueden sobrescribir los valores predeterminados del producto)

3. Variables agregadas por el sistema:
    - `access_token` - Token JWT para la autenticación de API de vuelta al CRM
    - `initiating_user` - ID de usuario que activó el aprovisionamiento
    - `customer_id`, `product_id`, `service_id`
4. Selecciones de inventario - Agregadas como pares `{inventory_type: inventory_id}`

Ejemplo de variables fusionadas:

```
{
    "customer_id": 123,
    "product_id": 456,
    "service_id": 789,          # Solo para complementos
    "SIM Card": 1001,           # Desde la selección de inventario
    "Mobile Number": 1002,      # Desde la selección de inventario
    "monthly_cost": 30,         # Desde provisioning_json_vars
    "data_gb": 50,              # Desde provisioning_json_vars
    "access_token": "eyJ...",   # Agregado por el sistema
    "initiating_user": 5        # Agregado por el sistema
}
```

Paso 3: Creación del Registro de Aprovisionamiento

Se crea un registro de `Provision` en la base de datos con:

- `provision_id` - Identificador único para seguimiento
- `provisioning_play` - Nombre del archivo del playbook
- `provisioning_json_vars` - Variables fusionadas como cadena JSON
- `task_count` - Número de tareas en el playbook (extraído de YAML)
- `provisioning_status` - Código de estado (inicialmente establecido en 1 = ejecutándose, luego actualizado a 0 = éxito, 2 = fallido, o puede permanecer en 1 si aún está en progreso)
- `product_id`, `customer_id`, `service_id` - Referencias de contexto

Paso 4: Ejecución del Playbook en Segundo Plano

La API genera un hilo en segundo plano que:

1. Carga el YAML del playbook desde `OmniCRM-API/Provisioners/plays/{playbook_name}.yaml`
2. Llama a `ansible_runner.run()` con:
   - `playbook` - Ruta al archivo YAML cargado
   - `extravars` - Todas las variables fusionadas (pasadas a Ansible)
   - `inventory` - Establecido en `'localhost,'` (ejecución local)
   - `event_handler` - Manejador personalizado para capturar eventos de ejecución de tareas
3. Monitorea la ejecución del playbook en tiempo real

Paso 5: Captura de Eventos y Registro (`ProvisioningEventHandler`)

A medida que se ejecuta cada tarea de Ansible, se capturan eventos y se almacenan como registros de `Provision_Event`:

- `event_name` - Nombre de la tarea del playbook
- `event_number` - Número de secuencia
- `provisioning_status` - Código de estado que indica el resultado de la tarea:
  - **0** = Éxito - Tarea completada con éxito
  - **1** = Ejecutándose - Tarea actualmente en ejecución
  - **2** = Fallido - Fallo crítico que detiene el aprovisionamiento
  - **3** = Fallido (ignorado) - La tarea falló pero se ignoraron los errores (`ignore_errors: true` en el playbook)
- `provisioning_result_json` - Resultados de la tarea con datos sensibles redactados

El manejador de eventos elimina automáticamente contraseñas, claves, secretos y otros datos sensibles de los registros.

Paso 6: Ejecución del Playbook de Ansible (`Provisioners/plays/*.yaml`)

El playbook de Ansible se ejecuta localmente y generalmente realiza estas acciones:

1. **Obtener Definición del Producto** - Solicitud GET a `/crm/product/product_id/{{ product_id }}` usando `{{ access_token }}`

2. **Obtener Información del Cliente** - Solicitud GET a `/crm/customer/customer_id/{{ customer_id }}`

3. **Procesar Elementos de Inventario** (si es necesario) - Solicitud GET a `/crm/inventory/inventory_id/{{ inventory_id }}` para cada elemento seleccionado para recuperar detalles completos (ICCID, MSISDN, números de serie, etc.)

4. **Configurar Sistemas Externos** - Realizar llamadas a la API para:

    - HSS (Servidor de Suscriptores de Hogar) para aprovisionamiento de suscriptores
    - IMS (Subsistema Multimedia IP) para registro de voz
    - CGRateS/OCS para creación de cuentas, configuración de carga, planes de tarifas
    - Servidores ENUM para mapeo de números de teléfono
    - Equipos de red (enrutadores, conmutadores, etc.)

5. **Agregar Costos de Configuración** (si corresponde) - POST a `/crm/transaction/` para registrar cargos únicos

6. **Cargar al Cliente** - POST a OCS/CGRateS para cargar retail_setup_cost si está configurado

7. **Crear Cuenta OCS** - POST a OCS/CGRateS para crear una cuenta de facturación con UUID

8. **Configurar Cargos Recurrentes** - Crear Acciones y Planes de Acción en OCS/CGRateS para cargos recurrentes mensuales

9. **Crear Registro de Servicio** - PUT/POST a `/crm/service/` para crear el registro del servicio en el CRM:

```
{
  "customer_id": 123,
  "product_id": 456,
  "service_name": "Mobile SIM - 0412345678",
  "service_uuid": "generated-uuid-for-ocs",
  "service_status": "Active",
  "service_type": "mobile",
  "retail_cost": 30,
  "wholesale_cost": 5,
  "provisioning_play": "play_psim_only",
  "provisioning_json_vars": "{...}"
}
```

10. **Asignar Inventario** - PATCH a `/crm/inventory/inventory_id/{{ inventory_id }}` para marcar el inventario como "Asignado" al servicio

11. **Enviar Notificaciones** (opcional) - Correo electrónico o SMS al cliente con detalles del servicio

Paso 7: Finalización y Actualización de Estado

Cuando el playbook se completa:

- **Éxito**: `Provision.provisioning_status` actualizado a **0** (Éxito)
- **Fallo Crítico**: `Provision.provisioning_status` actualizado a **2** (Fallido), y correo de fallo enviado a `crm_config.provisioning.failure_list`
- **Fallos No Críticos**: Tareas que fallan con `ignore_errors: true` se marcan con estado **3** (Fallido pero ignorado) y no detienen el aprovisionamiento

El servicio aprovisionado ahora es visible en el CRM y está activo para el cliente (si el aprovisionamiento tuvo éxito).

Diferencias Clave: Aprovisionamiento Independiente vs Complemento vs Agrupado

Productos Independientes (`category: standalone`):

- Reciben `customer_id` y `product_id`

- Típicamente requieren elementos de inventario (tarjetas SIM, números de teléfono, módems)
- Crean un nuevo registro de servicio a través de la API PUT `/crm/service/`
- Aprovisionan nuevos recursos en sistemas externos (HSS, OCS, equipos de red)
- Ejemplo: Nueva activación de SIM móvil, nueva conexión a internet

Productos Complementos (`category: addon`):

- Reciben `customer_id`, `product_id`, y **`service_id`** (servicio existente a modificar)
- Típicamente NO requieren inventario (o inventario mínimo)
- **Modifican un servicio existente** o agregan cargos a una cuenta OCS existente
- Pueden ejecutar acciones en OCS (agregar paquete de datos, agregar crédito, habilitar característica)
- No crean nuevos registros de servicio (o crean registros de servicio secundarios vinculados al principal)
- Ejemplo: Recarga de plan de datos mensual, paquete de roaming internacional, crédito extra

Productos Agrupados (`category: bundle`):

- Similar a los complementos en términos de variables recibidas
- Pueden requerir algunos elementos de inventario (por ejemplo, módem para paquete de hogar)
- Crean **múltiples** registros de servicio relacionados (Internet + TV + Teléfono)
- Aprovisionan múltiples recursos a través de diferentes sistemas
- Vinculan servicios juntos en el CRM para facturación/gestión unificada
- Ejemplo: Paquete de hogar (Internet + IPTV + teléfono VoIP)

Requisitos del Playbook de Aprovisionamiento

Para que un playbook funcione correctamente, debe:

1. **Estar ubicado en** `OmniCRM-API/Provisioners/plays/{playbook_name}.yaml`
2. **Aceptar variables** a través de `extravars` de Ansible (accedidas como `{{ variable_name }}`)
3. **Autenticar llamadas a la API** usando `Authorization: Bearer {{ access_token }}` encabezado
4. **Manejar fallos de manera elegante** usando bloques `rescue` y `ignore_errors` donde sea apropiado
5. **Crear registro de servicio** para productos independientes, o modificar servicio existente para complementos
6. **Asignar inventario** si se seleccionaron elementos de inventario
7. **Devolver mensajes de error significativos** a través del módulo `fail` cuando ocurran errores críticos

Variables Comunes Disponibles en Playbooks

Cada playbook recibe estas variables:

- `customer_id` - Entero, cliente que ordena el servicio
- `product_id` - Entero, producto que se está aprovisionando
- `service_id` - Entero (solo complementos/agrupados), servicio existente a modificar
- `access_token` - Cadena, token JWT para autenticación de API en CRM
- `initiating_user` - Entero, usuario que activó el aprovisionamiento
- Además de cualquier ID de elemento de inventario: `{{ inventory_type }}: inventory_id`
- Además de cualquier variable de `provisioning_json_vars`
- Además de cualquier variable pasada en la solicitud de aprovisionamiento

Los playbooks pueden usar estos para:

- Obtener detalles completos del producto: `GET /crm/product/product_id/{{ product_id }}`
- Obtener detalles del cliente: `GET /crm/customer/customer_id/{{ customer_id }}`

- Obtener detalles del inventario: `GET /crm/inventory/inventory_id/{{ SIM_Card }}`
- Crear transacciones: `POST /crm/transaction/`
- Crear servicios: `PUT /crm/service/`
- Actualizar servicios: `PATCH /crm/service/{{ service_id }}`
- Asignar inventario: `PATCH /crm/inventory/inventory_id/{{ inventory_id }}`

Ejemplo: Flujo de Playbook de Complemento Simple

Para un complemento de recarga de datos móviles:

1. El playbook recibe: `customer_id`, `product_id`, `service_id`, `access_token`
2. Obtener detalles del servicio: `GET /crm/service/{{ service_id }}` para obtener `service_uuid`
3. Obtener detalles del producto: `GET /crm/product/product_id/{{ product_id }}` para obtener precios y cantidad de datos
4. Cargar al cliente en OCS: `POST a CGRateS` para deducir retail_cost del saldo
5. Agregar crédito de datos en OCS: `POST a CGRateS` para agregar saldo de datos con caducidad
6. Registrar transacción en CRM: `POST /crm/transaction/` con detalles de carga
7. Completar con éxito

Todo el proceso se rastrea en las tablas `Provision` y `Provision_Event` para propósitos de depuración y auditoría.

Participación de OCS

`OCS` (Sistema de Carga En Línea), implementado a través de CGRateS, maneja toda la carga en tiempo real y el seguimiento de uso para servicios. El registro de servicio del CRM actúa como un puntero a la cuenta de OCS, que gestiona:

- **Cargos recurrentes** - Tarifas mensuales, alquiler de DID, cargos de suscripción

- **Carga basada en uso** - Llamadas de voz por minuto, datos por MB, cargos por SMS
- **Gestión de saldos** - Saldos monetarios (crédito prepago) y saldos unitarios (GB de datos, minutos de voz, conteo de SMS)
- **Conversiones de saldo** - Convertir saldos monetarios en saldos unitarios (por ejemplo, gastar $30 para obtener un paquete de datos de 10GB)
- **Estado de la cuenta** - Activo, suspendido, deshabilitado según los límites de crédito y umbrales

El registro de servicio del CRM contiene metadatos y configuración (cliente, producto, precios, visibilidad), mientras que OCS contiene el estado de facturación en vivo (saldos, uso, cargos).

Recuperación de Uso y Saldos del Servicio

La información de uso del servicio se recupera de OCS/CGRateS y se muestra a clientes y personal en tiempo real.

Cómo se Recupera el Uso

Cuando se solicita el uso de un servicio (a través de la interfaz de usuario o API), ocurre el siguiente flujo:

1. **Solicitud de API** - El frontend llama a `GET /crm/service/{service_id}` o ve detalles del servicio en la interfaz de usuario

2. **Búsqueda del Servicio** - La API recupera el registro del servicio de la base de datos, extrae `service_uuid`

3. **Llamadas a la API de CGRateS** - El sistema realiza dos llamadas a CGRateS:

    i. **Get_Balance(service_uuid)** - Recupera el saldo de la cuenta con `BalanceMap`
        - Devuelve saldos organizados por tipo: DATOS, VOZ, SMS, MONETARIO, DATOS_DONGLE

- Cada saldo incluye: ID, Valor, Fecha de Expiración, Peso, DestinationIDs
- El sistema agrega campos legibles por humanos: `custom_Name_hr`, `custom_Expiration`, `custom_Description_String`

ii. **Get_ActionPlans(service_uuid)** - Recupera planes de acción de renovación automática activos (cubiertos en la siguiente sección)

4. **Fusión de Respuestas** - Los datos de CGRateS se fusionan en la respuesta del servicio:


```
{
  "service_id": 123,
  "service_name": "Mobile Service",
  "service_uuid": "abc-123-def",
  "cgrates": {
    "BalanceMap": {
      "DATA": [{
        "ID": "DATA_10GB",
        "Value": 5368709120,
        "ExpirationDate": "2025-02-01T00:00:00Z",
        "custom_Name_hr": "10GB Data Pack",
        "custom_Expiration": "Feb 1, 2025",
        "custom_Description_String": "5 GB remaining"
      }],
      "VOICE": [{
        "ID": "VOICE_UNLIMITED",
        "Value": 999999999,
        "custom_Name_hr": "Unlimited Calls",
        "custom_Description_String": "Unlimited minutes"
      }],
      "MONETARY": [{
        "ID": "PREPAID_CREDIT",
        "Value": 25.50,
        "custom_Description_String": "$25.50 credit"
      }]
    },
    "ActionPlans": [...]
  }
}
```

5. **Visualización en la Interfaz de Usuario** - Los componentes del frontend muestran los datos de uso:
    - **ServiceUsage.js** - Componente principal de visualización de uso con actualización automática cada 3 segundos
    - **UsageCard.js** - Tarjetas de resumen para cada tipo de saldo
    - **UsageProgress.js** - Barras de progreso que muestran el porcentaje utilizado/restante
    - Los saldos están codificados por colores y formateados para facilitar la lectura

Estructura de Datos de Uso

Cada saldo en el `BalanceMap` contiene:

Campos Nativos de CGRateS:

- `ID` - Identificador único para el saldo (por ejemplo, "DATA_10GB_2025_01")
- `Value` - Monto del saldo:
    - Para DATOS: bytes (5368709120 = 5 GB)
    - Para VOZ: segundos (3600 = 1 hora)
    - Para SMS: conteo (100 = 100 mensajes)
    - Para MONETARIO: unidades de moneda (25.50 = $25.50)
- `ExpirationDate` - Marca de tiempo ISO 8601 cuando expira el saldo
- `Weight` - Prioridad para el consumo del saldo (peso más alto consumido primero)
- `DestinationIDs` - Destinos a los que se aplica este saldo (por ejemplo, ["AU", "INTERNATIONAL"])

Campos Legibles por Humanos (agregados por CRM):

- `custom_Name_hr` - Nombre legible por humanos extraído del ID
- `custom_Expiration` - Fecha de expiración formateada (por ejemplo, "15 de enero de 2025" o "en 11 días")
- `custom_Description_String` - Descripción del saldo legible por humanos:
    - DATOS: "5 GB restantes" o "10 GB en total"

    - VOZ: "60 minutos restantes" o "Ilimitado"
    - SMS: "50 SMS restantes"
    - MONETARIO: "$25.50 de crédito"

Control de Visibilidad del Uso

La visibilidad del uso del servicio se controla mediante dos campos:

- **service_visible_to_customer** - Si es falso, el servicio se oculta completamente del portal de autoservicio del cliente
- **service_usage_visible_to_customer** - Si es falso, el servicio es visible pero se ocultan los detalles de uso/saldo (el cliente puede ver que tiene el servicio, pero no cuánto ha utilizado)

Esto permite a los operadores:

- Ocultar servicios internos/de prueba de los clientes
- Mostrar que el servicio existe sin revelar el uso (útil para planes ilimitados o servicios sensibles)
- Visualización completamente transparente del uso (predeterminado)

Actualizaciones de Uso en Tiempo Real

La Interfaz Web actualiza automáticamente los datos de uso:

- **Intervalo**: Cada 3 segundos (configurable en el componente ServiceUsage)
- **Método**: Consulta `GET /crm/service/{service_id}` que obtiene datos en vivo de CGRateS
- **Eficiencia**: Solo las vistas de servicio activas se actualizan; las vistas de lista utilizan datos en caché

Esto asegura que los clientes y el personal vean actualizaciones de saldo en casi tiempo real a medida que ocurre el uso.

Cargos Recurrentes / AutoRenovación

Los cargos recurrentes, como un Cargo de Servicio Mensual, o un Cargo por DID, se crean primero como Acciones dentro de `OCS` y toman el formato `Action_ServiceUUID_ServiceName_WhatitDoes`.

Por ejemplo, para un servicio GPON de $60 por mes que incluye 1TB de uso, la Acción se vería algo así:

`Action_kj49-adsf-1234-9742_60_GPON_1TB_MonthlyExpiry`

1. Restablecer Saldo Monetario a $0
2. Enviar un POST HTTP a `/simple_provision` en el CRM para aprovisionar algo
3. Agregar un Crédito por 1TB de Uso que expira en 1 mes

Si queremos que el MRC sea recurrente (lo hacemos), entonces crearíamos un `ActionPlan` llamado `ActionPlan_{{ service_uuid }}_Monthly_Charge` que tendría el tiempo establecido en *mensual* para activarse cada mes, y asignaríamos el `ActionPlan` a la cuenta.

Podemos establecer en función del parámetro *Año / Meses / Días* una fecha de expiración para cuando el MRC se detenga también, por ejemplo, para un servicio fijo de 12 meses que se detenga después de este punto.

Debido a que las Acciones y `ActionPlans` son únicas para el servicio, no comparten nada con otros servicios.

Esto significa que podemos aprovisionarlos con valores ajustados, y no impactará a otros servicios.

Complementos y Adiciones

Los complementos / Adiciones como comprar datos adicionales, paquetes de roaming, minutos internacionales, etc., se manejan de manera similar. Se crea

una Acción para hacer lo que se necesita, como cobrar un valor monetario y luego otorgar un saldo unitario con una fecha de expiración establecida.

En lugar de usar `ActionPlans` para agregar esto automáticamente a la cuenta, simplemente disparamos `ExecuteAction` para la Acción que acabamos de crear una vez desde Ansible.

Agregar Saldos Monetarios Prepagados

Para saldos monetarios prepagados, como un plan PAYG de $10, esto se agrega como un saldo monetario, pero con una prioridad más alta.

El límite de crédito en estos servicios para el saldo predeterminado sería $0.

Límites de Crédito / Prevención de Gastos Excesivos

`ThresholdS` se utilizan en cada cuenta para establecer el gasto máximo para un período de tiempo determinado.

Para clientes PAYG / Prepagados, esto es $0.

Interactuando con OCS a través del CRM

Para cada Servicio, puede ver los `Balances` y `ActionPlans`, `Actions` y `ThresholdS` de `OCS` desde dentro de la API del CRM.

Los `ActionPlans` pueden ser eliminados según sea necesario desde la API del CRM, ejecutados a través de Playbooks de Ansible. Los `ActionPlans` pueden ser agregados según sea necesario, desde el CRM, agregando un Complemento/Servicio y ejecutados a través de Playbooks de Ansible.

Las cuentas de `OCS` pueden ser deshabilitadas, lo que detendrá la ejecución de `ActionPlans` y dejará de poder consumirse servicios.

Para Límites de Crédito, se establece un valor de `ThresholdS` según la política para el producto.

Visualización y Gestión de ActionPlans en el CRM

Los ActionPlans (configuraciones de renovación automática) se muestran y gestionan a través de la interfaz del CRM, permitiendo al personal y a los clientes ver las próximas renovaciones automáticas y gestionarlas.

Cómo se Recuperan y Muestran los ActionPlans

Al ver un servicio en el CRM, los ActionPlans se recuperan automáticamente y se muestran:

1. **Llamada a la API** - Cuando se llama a `GET /crm/service/{service_id}`, la API:
    - Recupera el registro del servicio de la base de datos
    - Extrae el `service_uuid` (identificador de cuenta de OCS)
    - Llama a la API de CGRateS para recuperar ActionPlans por UUID de servicio
    - Esto llama internamente a `Get_ActionPlans(service_uuid)` en CGRateS
2. **Estructura de Datos de ActionPlan** - Cada ActionPlan devuelto contiene:

```
{
    "ActionPlanId": "ServiceID_abc-123-
def__ProductID_456__...",
    "PlanName": "Monthly_Renewal_Plan",
    "NextExecTime": "2025-02-01T00:00:00Z",
    "custom_NextExecTime_hr": "in 11 days",
    "ActionPlanId_split_dict": {
        "ServiceID": "abc-123-def",
        "ProductID": 456,
        "CustomerID": 789,
        ...
    }
}
```

- `ActionPlanId` - Identificador único que contiene información codificada del servicio/producto/cliente
- `PlanName` - Nombre del plan de acción (típicamente el nombre del playbook de renovación)
- `NextExecTime` - Marca de tiempo ISO cuando se ejecutará el ActionPlan a continuación
- `custom_NextExecTime_hr` - Tiempo legible por humanos hasta la ejecución (por ejemplo, "en 11 días", "mañana", "1 de febrero de 2025")
- `ActionPlanId_split_dict` - Diccionario con componentes analizados del ActionPlanId

3. **Visualización en la Interfaz de Usuario** - Los ActionPlans se muestran en el componente **ActionPlansTable**:

   Columnas de la Tabla:

   - **Nombre del Producto** - Recuperado al buscar `ProductID` desde el ActionPlanId
   - **Costo** - Muestra `retail_cost` de la definición del producto
   - **Fecha de Renovación** - Muestra `custom_NextExecTime_hr` (legible por humanos)
   - **Acciones** - Dos botones:

- **Renovar Ahora** - Aprovisionar el complemento/renovación de inmediato (evita esperar la ejecución programada)
- **Eliminar Renovación Automática** - Cancela la renovación automática

Cuando No Existen ActionPlans:

- La tabla muestra el mensaje: "No hay renovación automática habilitada para este servicio"
- El cliente puede agregar complementos de renovación automática para habilitar la renovación automática

Gestión de ActionPlans

El personal y los clientes pueden gestionar ActionPlans a través de la interfaz de usuario:

Eliminar un ActionPlan (Cancelar Renovación Automática):

1. Haga clic en el botón "Eliminar Renovación Automática" en ActionPlansTable
2. Aparece un modal de confirmación: "¿Está seguro de que desea eliminar esta renovación automática?"
3. Al confirmar, el frontend llama a: `DELETE /crm/oam/remove_action_plan/{action_plan_id}`
4. La API elimina el ActionPlan de CGRateS a través de `ocs.Remove_ActionPlan()`
5. Se registra la actividad: "Se eliminó ActionPlan {ActionPlanId} del servicio {service_id}"
6. El ActionPlan desaparece de la tabla

Renovar Inmediatamente (Renovación Manual):

1. Haga clic en el botón "Renovar Ahora" en ActionPlansTable
2. Aparece un modal de confirmación: "¿Está seguro de que desea renovar esto ahora?"
3. Al confirmar, el sistema:
   - Extrae product_id del ActionPlanId

    - Crea un nuevo trabajo de aprovisionamiento para ese producto
    - Aprovisiona el complemento de inmediato (ejecuta el playbook de aprovisionamiento)
    - El servicio recibe los beneficios del complemento sin esperar a la renovación programada
4. El modal de estado de aprovisionamiento muestra el progreso
5. Al éxito, los saldos se actualizan de inmediato

Persistencia de descuentos de complemento a través de renovaciones:

Los descuentos en el punto de venta elegidos al agregar un complemento a un servicio se almacenan por `(service_id, product_id)` en `service_product_discount` y se reaplican en la autorenovación y Renovar Ahora.

- Un descuento positivo actualiza el porcentaje almacenado para futuras renovaciones.
- Elegir **Sin descuento (0%)** en un aprovisionamiento de complemento posterior también actualiza: el porcentaje se establece en `0` para que las renovaciones cobren el precio completo y ya no se basen en una transacción de venta con descuento anterior.
- Mientras no exista una fila persistente (ventas heredadas), las renovaciones pueden aún inferir el descuento de la transacción de venta original o variables de aprovisionamiento.
- Detener la autorenovación por sí solo **no** borra un descuento almacenado; re- aprovisionar sin descuento (o almacenar una fila explícita del 0%) lo hace.

Los descuentos del plan base independiente continúan existiendo en `customer_service.discount_*` y no están relacionados con esta tabla.

Agregar Renovación Automática:

La renovación automática se habilita al aprovisionar un producto complemento que tiene `auto_renew` configurado:

- Productos con `auto_renew = "true"` - Crean automáticamente ActionPlans durante el aprovisionamiento

- Productos con `auto_renew = "prompt"` - Preguntan al cliente si desea renovación automática (diálogo modal)
- Productos con `auto_renew = "false"` - Nunca crean ActionPlans (compra única)

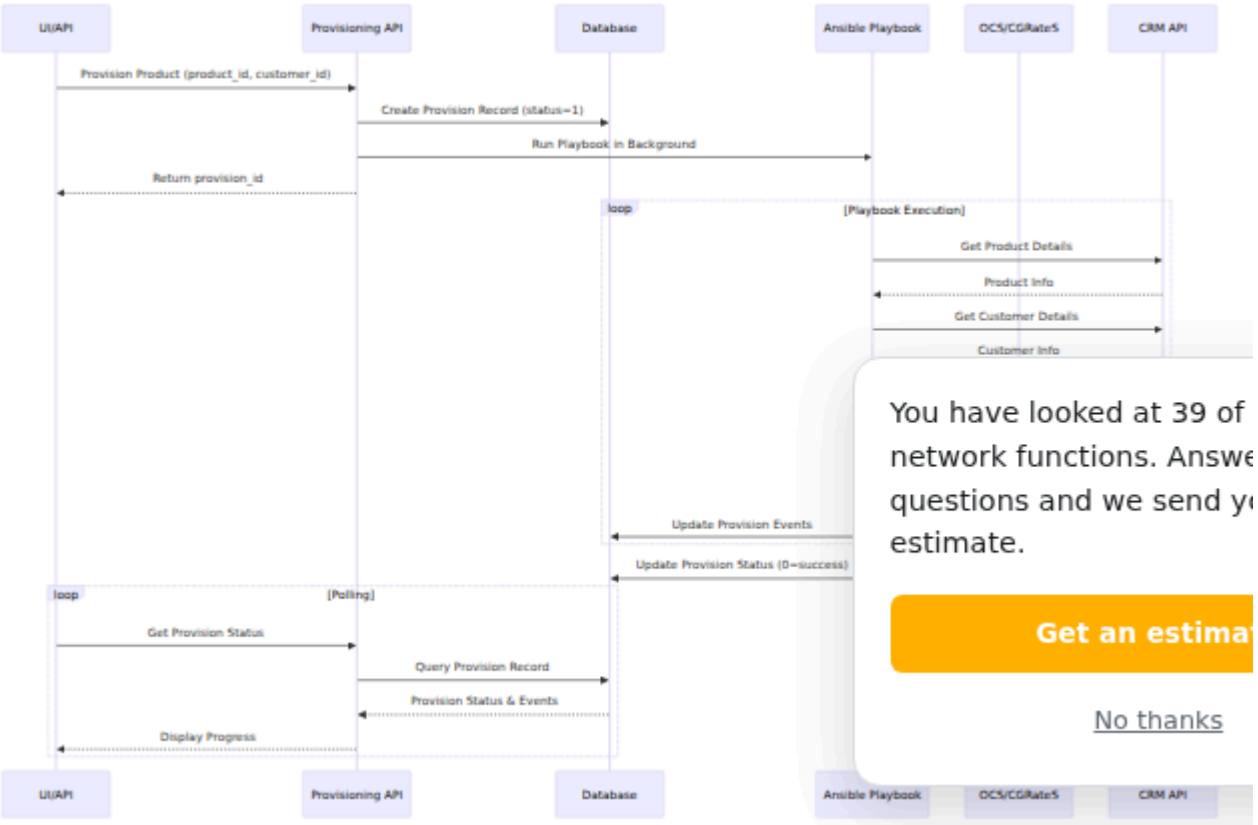
El playbook de aprovisionamiento crea el ActionPlan en CGRateS con:

- ActionPlanId único que codifica los IDs de servicio, producto y cliente
- Horario de renovación (mensual, anual, intervalo personalizado)
- Acción a ejecutar (típicamente reprovisionar el mismo complemento)
- Fecha de expiración (si el contrato

Sistema de Aprovisionamiento

OmniCRM utiliza **Ansible** para automatizar el aprovisionamiento, configuración y desaprovisionamiento de servicios al cliente. El sistema de aprovisionamiento está diseñado para ser flexible, permitiendo flujos de trabajo complejos mientras mantiene la consistencia y la fiabilidad.

Véase también: Aprovisionamiento de Tarjetas SIM para información detallada sobre el aprovisionamiento de SIM móviles, incluyendo tanto SIM físicas como eSIM.



> **NOTA**
>
> Para un recorrido completo del viaje de producto a servicio con ejemplos detallados de playbooks de Ansible, estrategias de precios y escenarios del mundo real, consulte Guía Completa del Ciclo de Vida del Producto.

Visión General

Cuando se ordena un producto o se necesita configurar un servicio, OmniCRM crea un **Trabajo de Aprovisionamiento** que ejecuta uno o más playbooks de Ansible. Estos playbooks interactúan con varios sistemas backend (OCS/CGRateS, equipos de red, APIs, etc.) para aprovisionar completamente el servicio.

El sistema de aprovisionamiento admite dos flujos de trabajo principales:

1. **Aprovisionamiento Estándar** - Activado por el personal o los clientes a través de la UI/API
2. **Aprovisionamiento Simple** - Activado por sistemas externos como OCS para operaciones automatizadas

Valores de Estado de Aprovisionamiento

Los trabajos de aprovisionamiento y las tareas individuales pueden tener los siguientes estados:

- **Estado 0 (Éxito)** - El trabajo de aprovisionamiento se completó con éxito
- **Estado 1 (En Ejecución)** - El trabajo de aprovisionamiento o la tarea se está ejecutando actualmente
- **Estado 2 (Fallido - Crítico)** - Ocurrió una falla crítica que causó que el aprovisionamiento fallara
- **Estado 3 (Fallido - Ignorado)** - Una tarea falló pero tenía `ignore_errors: true`, por lo que el aprovisionamiento continuó

Cuando un trabajo de aprovisionamiento falla, OmniCRM envía notificaciones por correo electrónico a la lista de notificación de fallos configurada con información detallada sobre el error.

Cómo los Productos Impulsan el Aprovisionamiento

La definición de **Producto** es el plano de lo que se aprovisiona y cómo. Cuando un usuario selecciona un producto para aprovisionar, el sistema lee varios campos clave de la definición del producto para determinar qué hacer.

Campos del Producto Utilizados en el Aprovisionamiento

Una definición de producto contiene:

- `provisioning_play` - El nombre del playbook de Ansible a ejecutar (sin la extensión .yaml)
- `provisioning_json_vars` - Cadena JSON que contiene variables predeterminadas para pasar a Ansible

- `inventory_items_list` - Lista de tipos de inventario que deben ser asignados (por ejemplo, `['SIM Card', 'Mobile Number']`)
- `product_id`, `product_name`, campos de precios - Pasados automáticamente al playbook

Ejemplo de Definición de Producto

```
{
  "product_id": 1,
  "product_slug": "Mobile-SIM",
  "product_name": "Mobile SIM Only",
  "provisioning_play": "play_psim_only",
  "provisioning_json_vars": "{\"iccid\": \"\", \"msisdn\": \"\"}",
  "inventory_items_list": "['SIM Card', 'Mobile Number']",
  "retail_cost": 0,
  "retail_setup_cost": 0,
  "wholesale_cost": 3,
  "wholesale_setup_cost": 1
}
```

Del Producto al Trabajo de Aprovisionamiento

Cuando se inicia el aprovisionamiento, el sistema:

1. **Carga el playbook** especificado en `provisioning_play`

   El sistema busca `OmniCRM-API/Provisioners/plays/play_psim_only.yaml`

2. **Fusiona variables** de múltiples fuentes en `extra_vars`:

   i. **Desde provisioning_json_vars**: `{"iccid": "", "msisdn": ""}`
   ii. **Desde el cuerpo de la solicitud**: Cualquier variable adicional que el usuario/API proporcione
   iii. **Desde los campos del producto**: `product_id`, `customer_id`, etc.
   iv. **Desde la autenticación**: `access_token` o configuración para `refresh_token`

3. **Asigna inventario** basado en `inventory_items_list`

Antes de ejecutar el playbook, la UI/API solicita la selección de inventario:

- **SIM Card** - El usuario selecciona una SIM disponible del inventario
- **Mobile Number** - El usuario selecciona un número de teléfono disponible

Los IDs de inventario seleccionados se añaden a `extra_vars` con el tipo de inventario como clave:

```
extra_vars = {
    "product_id": 1,
    "customer_id": 456,
    "SIM Card": 789,        # inventory_id de la SIM
seleccionada
    "Mobile Number": 101,   # inventory_id del número de
teléfono seleccionado
    "iccid": "",            # Desde provisioning_json_vars
    "msisdn": "",           # Desde provisioning_json_vars
    "access_token": "eyJ..."
}
```

4. **Pasa todo a Ansible** a través de `hostvars[inventory_hostname]`

   Dentro del playbook, las variables son accesibles como:

```
- name: Get inventory_id for SIM Card
  set_fact:
    inventory_id_sim_card: "{{ hostvars[inventory_hostname]
['SIM Card'] | int }}"
  when: "'SIM Card' in hostvars[inventory_hostname]"
```

Cómo los Playbooks Usan Variables de Inventario

Una vez que el playbook tiene los IDs de inventario, recupera los detalles completos del inventario desde la API:

```
- name: Get SIM Card Details from Inventory
  uri:
    url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ inventory_id_sim_card }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
    return_content: yes
  register: sim_card_response

- name: Extract ICCID and IMSI from inventory
  set_fact:
    iccid: "{{ sim_card_response.json.iccid }}"
    imsi: "{{ sim_card_response.json.imsi }}"

- name: Get Phone Number Details from Inventory
  uri:
    url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ inventory_id_phone_number }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
    return_content: yes
  register: phone_number_response

- name: Extract MSISDN
  set_fact:
    msisdn: "{{ phone_number_response.json.msisdn }}"
```

El playbook puede entonces usar estos valores para:

- Aprovisionar la tarjeta SIM en el HSS con el IMSI
- Configurar el número de teléfono en el sistema de facturación
- Asignar los elementos de inventario al cliente
- Crear el registro de servicio con estos detalles

Ejemplo del Mundo Real: Aprovisionamiento de SIM Móvil

Desde `play_psim_only.yaml`, aquí se muestra cómo utiliza los datos del producto y del inventario:

```
- name: Get Product information from CRM API
  uri:
    url: "{{ crm_config.crm.base_url }}/crm/product/product_id/{{
product_id }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
    return_content: yes
  register: api_response_product

- name: Set package facts from product
  set_fact:
    package_name: "{{ api_response_product.json.product_name }}"
    package_comment: "{{ api_response_product.json.comment }}"
    setup_cost: "{{ api_response_product.json.retail_setup_cost
}}"
    monthly_cost: "{{ api_response_product.json.retail_cost }}"

- name: Set inventory_id_sim_card if SIM Card was selected
  set_fact:
    inventory_id_sim_card: "{{ hostvars[inventory_hostname]['SIM
Card'] | int }}"
  when: "'SIM Card' in hostvars[inventory_hostname]"

- name: Set inventory_id_phone_number if Mobile Number was
selected
  set_fact:
    inventory_id_phone_number: "{{ hostvars[inventory_hostname]
['Mobile Number'] | int }}"
  when: "'Mobile Number' in hostvars[inventory_hostname]"

- name: Get SIM Card details from inventory
  uri:
    url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ inventory_id_sim_card }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
    return_content: yes
  register: sim_inventory_response

- name: Get Phone Number details from inventory
  uri:
```

```
    url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ inventory_id_phone_number }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
    return_content: yes
  register: phone_inventory_response

- name: Extract values from inventory
  set_fact:
    iccid: "{{ sim_inventory_response.json.iccid }}"
    imsi: "{{ sim_inventory_response.json.imsi }}"
    msisdn: "{{ phone_inventory_response.json.msisdn }}"
    ki: "{{ sim_inventory_response.json.ki }}"
    opc: "{{ sim_inventory_response.json.opc }}"

- name: Provision subscriber on HSS
  uri:
    url: "http://{{ hss_server }}/subscriber/{{ imsi }}"
    method: PUT
    body_format: json
    body:
      {
        "imsi": "{{ imsi }}",
        "msisdn": "{{ msisdn }}",
        "ki": "{{ ki }}",
        "opc": "{{ opc }}",
        "enabled": true
      }
    status_code: 200

- name: Assign inventory to customer
  uri:
    url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ inventory_id_sim_card }}"
    method: PATCH
    headers:
      Authorization: "Bearer {{ access_token }}"
    body_format: json
    body:
      {
        "customer_id": {{ customer_id }},
        "item_state": "Assigned"
```

```
      }
    status_code: 200
```

Esto demuestra el flujo completo:

1. La definición del producto especifica `provisioning_play: "play_psim_only"`
2. El producto requiere `inventory_items_list: ['SIM Card', 'Mobile Number']`
3. El usuario selecciona elementos de inventario durante el aprovisionamiento
4. Los IDs de inventario se pasan al playbook como `extra_vars`
5. El playbook recupera los detalles completos del inventario desde la API
6. El playbook utiliza los datos del inventario para configurar el equipo de red
7. El playbook marca el inventario como asignado al cliente

Rollback y Limpieza: Patrón de Mejores Prácticas

Mejor Práctica Crítica: El mismo playbook debe manejar tanto el rollback de aprovisionamiento fallido como el desaprovisionamiento intencional utilizando la estructura `block` y `rescue` de Ansible.

Estructura del Playbook

Desde `play_psim_only.yaml`:

```yaml
- name: OmniCore Service Provisioning 2024
  hosts: localhost
  gather_facts: no
  become: False

  tasks:
    - name: Main block
      block:
        # --- TAREAS DE APROVISIONAMIENTO ---
        - name: Get Product information
          uri: ...

        - name: Create account in OCS
          uri: ...

        - name: Provision subscriber on HSS
          uri: ...

        - name: Create service record
          uri: ...

        # ... muchas más tareas de aprovisionamiento ...

      rescue:
        # --- TAREAS DE LIMPIEZA ---
        # Esta sección se ejecuta cuando:
        # 1. Cualquier tarea en el bloque falla (rollback)
        # 2. action == "deprovision" (limpieza intencional)

        - name: Get Inventory items linked to this service
          uri:
            url: "{{ crm_config.crm.base_url
}}/crm/inventory/customer_id/{{ customer_id }}"
            method: GET
          register: inventory_api_response
          ignore_errors: True

        - name: Return inventory to pool
          uri:
            url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ item.inventory_id }}"
            method: PATCH
            body_format: json
```

```
      body:
        service_id: null
        customer_id: null
        item_state: "Used"
    with_items: "{{ inventory_api_response.json.data }}"
    ignore_errors: True

  - name: Delete Account from Charging
    uri:
      url: "http://{{ crm_config.ocs.OCS }}/jsonrpc"
      method: POST
      body:
        {
          "method": "ApierV1.RemoveAccount",
          "params": [{
              "Tenant": "{{ crm_config.ocs.ocsTenant }}",
              "Account": "{{ service_uuid }}"
          }]
        }
    ignore_errors: True

  - name: Delete Attribute Profile
    uri:
      url: "http://{{ crm_config.ocs.OCS }}/jsonrpc"
      method: POST
      body:
        {
          "method": "APIerSv1.RemoveAttributeProfile",
          "params": [{
              "ID": "ATTR_ACCOUNT_{{ service_uuid }}"
          }]
        }
    ignore_errors: True

  - name: Remove Resource Profile
    uri: ...
    ignore_errors: True

  - name: Remove Filters
    uri: ...
    ignore_errors: True

  - name: Deprovision Subscriber from HSS
    uri:
```

```
            url: "{{ item.key }}/subscriber/{{
item.value.subscriber_id }}"
            method: DELETE
          loop: "{{ hss_subscriber_data | dict2items }}"
          ignore_errors: True
          when:
            - deprovision_subscriber | bool == true

        - name: Patch Subscriber to Dormant State
          uri:
            url: "{{ item.key }}/subscriber/{{
item.value.subscriber_id }}"
            method: PATCH
            body:
              {
                "enabled": true,
                "msisdn": "9999{{ imsi[-10:] }}",  # Número
ficticio
                "ue_ambr_dl": 9999999,              # Inusualmente
alto
                "ue_ambr_ul": 9999999
              }
          loop: "{{ hss_subscriber_data | dict2items }}"
          when:
            - deprovision_subscriber | default(false) | bool ==
false

        # La afirmación final determina el éxito o el fracaso
        - name: Set status to "Success" if Manual deprovision /
Fail if failed provision
          assert:
            that:
              - action == "deprovision"
```

Por qué Este Patrón es una Mejores Prácticas

1. Sin Duplicación de Código

Las mismas tareas de limpieza manejan ambos escenarios:

- **Fallo de Aprovisionamiento (Rollback)**: Si cualquier tarea en el `block` falla, la sección `rescue` se ejecuta automáticamente

- **Desaprovisionamiento Intencional**: Cuando se llama con `action: "deprovision"`, el playbook salta inmediatamente a `rescue`

2. Limpieza Completa Garantizada

Cuando un aprovisionamiento falla a mitad de camino, la sección de rescate asegura:

- Todas las cuentas de OCS creadas son eliminadas
- Todas las entradas de equipos de red configuradas son removidas
- El inventario asignado se devuelve al pool
- Los suscriptores de HSS son eliminados o establecidos como inactivos
- No queda aprovisionamiento parcial en ningún sistema

Esto previene recursos "huérfanos" que:

- Consumen inventario sin ser rastreados
- Crean cuentas de facturación que no están vinculadas a servicios
- Causan confusión durante la solución de problemas
- Desperdician recursos de red

3. Manejo de Fallos Elegante con ignore_errors

Observe que cada tarea de limpieza utiliza `ignore_errors: True`. Esto es intencional porque:

- Durante el rollback, algunos recursos pueden no haberse creado aún
- Queremos intentar todas las tareas de limpieza incluso si algunas fallan
- La afirmación final determina el éxito/fallo general

Por ejemplo, si el aprovisionamiento falla en "Crear cuenta en OCS", la limpieza intentará:

- Eliminar la cuenta de OCS (fallará, pero se ignorará)
- Remover perfiles de atributos (fallará, pero se ignorará)
- Devolver inventario (éxito)
- Eliminar suscriptor de HSS (puede no existir, ignorado)

4. Distinguir Desaprovisionamiento de Rollback

La afirmación final al final de `rescue` es ingeniosa:

```
- name: Set status to "Success" if Manual deprovision / Fail if
failed provision
  assert:
    that:
      - action == "deprovision"
```

Esto significa:

- **Si ``action == "deprovision"``**: La afirmación pasa, el playbook tiene éxito (estado 0)
- **Si ``action`` no está configurado o != "deprovision"**: La afirmación falla, el playbook falla (estado 2)

Así que el mismo código de limpieza resulta en diferentes estados de trabajos de aprovisionamiento dependiendo de la intención.

5. Limpieza Condicional Basada en el Tipo de Servicio

Algunas tareas de limpieza utilizan condicionales para manejar diferentes escenarios:

```
- name: Deprovision Subscriber from HSS
  uri: ...
  when:
    - deprovision_subscriber | bool == true

- name: Patch Subscriber to Dormant State
  uri: ...
  when:
    - deprovision_subscriber | default(false) | bool == false
```

Esto permite una limpieza flexible:

- **Eliminación completa**: Cuando las SIM están dedicadas a clientes (`deprovision_subscriber: true`)

- **Estado inactivo**: Cuando las SIM son reutilizables y deben permanecer en HSS (`deprovision_subscriber: false`)

Cómo Usar Este Patrón

Para Aprovisionamiento:

```
{
  "product_id": 1,
  "customer_id": 456,
  "provisioning_play": "play_psim_only"
}
```

Si el aprovisionamiento falla, se produce un rollback automático a través de `rescue`.

Para Desaprovisionamiento:

```
{
  "service_id": 123,
  "service_uuid": "Service_abc123",
  "action": "deprovision",
  "provisioning_play": "play_psim_only"
}
```

El playbook salta directamente a la sección `rescue`, ejecuta toda la limpieza y tiene éxito.

Resumen de Beneficios

 Una única fuente de verdad: Un playbook maneja aprovisionamiento y desaprovisionamiento **Operaciones atómicas**: O completamente aprovisionado o completamente limpiado **Sin recursos huérfanos**: Aprovisionamientos fallidos no dejan rastro **Mantenimiento más fácil**: Cambios en la lógica de aprovisionamiento se aplican automáticamente a la limpieza **Errores reducidos**: Sin posibilidad de que el código de aprovisionamiento y desaprovisionamiento se desincronice **Testeable**: Se puede probar la lógica de desaprovisionamiento ejecutando con `action: "deprovision"`

Este patrón debe seguirse en todos los playbooks de aprovisionamiento para asegurar fiabilidad y consistencia.

Sobrescribiendo Variables del Producto

Las `provisioning_json_vars` pueden ser sobrescritas en el momento del aprovisionamiento. Por ejemplo, un producto podría definir:

```
{
  "provisioning_json_vars": "{\"monthly_cost\": 50,
\"data_limit_gb\": 100}"
}
```

Pero al aprovisionar, puedes sobrescribir estos:

```
{
  "product_id": 1,
  "customer_id": 456,
  "monthly_cost": 45,
  "data_limit_gb": 150
}
```

Las `extra_vars` fusionadas utilizarán los valores sobrescritos. Esto permite:

- Precios personalizados para clientes específicos
- Límites de datos diferentes basados en promociones
- Pruebas con diferentes parámetros sin modificar el producto

Productos Sin Inventario

No todos los productos requieren inventario. Por ejemplo, un complemento de datos o un interruptor de función podría tener:

```
{
  "product_id": 10,
  "product_name": "Extra 10GB Data",
  "provisioning_play": "play_local_data_addon",
  "provisioning_json_vars": "{\"data_gb\": 10}",
  "inventory_items_list": "[]"
}
```

En este caso, el playbook recibe:

```
extra_vars = {
    "product_id": 10,
    "customer_id": 456,
    "service_id": 123,  # Servicio al que se añadirá datos
    "data_gb": 10,
    "access_token": "eyJ..."
}
```

El playbook simplemente añade los datos al servicio existente sin necesidad de elementos de inventario.

Flujo de Trabajo de Aprovisionamiento Estándar

El aprovisionamiento estándar se inicia cuando:

- Un miembro del personal añade un servicio a un cliente desde la UI
- Un cliente ordena un servicio a través del portal de autoservicio
- La API se llama directamente con `PUT /crm/provision/`

Cuando Haces Clic en "Aprovisionar"

Aquí está el flujo completo que ocurre cuando un usuario hace clic en el botón "Aprovisionar":

1. La UI Muestra la Selección de Producto

El usuario selecciona un producto del catálogo de productos. El producto contiene:

- `provisioning_play` - Qué playbook de Ansible ejecutar
- `inventory_items_list` - Inventario requerido (por ejemplo, `['SIM Card', 'Mobile Number']`)
- `provisioning_json_vars` - Variables predeterminadas

2. Selector de Inventario (Si es Requerido)

Si `inventory_items_list` no está vacío, aparece un modal mostrando listas desplegables para cada tipo de inventario. El usuario debe seleccionar elementos de inventario disponibles antes de continuar.

3. Botón de Aprovisionamiento Clicado

JavaScript envía la solicitud `PUT /crm/provision/`:

```
PUT /crm/provision/
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
Content-Type: application/json

{
  "product_id": 42,
  "customer_id": 123,
  "SIM Card": 5001,
  "Mobile Number": 5002
}
```

4. La API Recibe la Solicitud

El endpoint de aprovisionamiento:

- Valida la autenticación (token Bearer, clave API o lista blanca de IP)
- Verifica que el usuario tenga permiso `CREATE_PROVISION`
- Extrae `initiating_user` del token
- Carga la definición del producto desde la base de datos
- Recupera la ruta del playbook: `OmniCRM-API/Provisioners/plays/play_psim_only.yaml`

5. Variables Fusionadas

El sistema combina variables de múltiples fuentes:

```
# Desde el producto
product_vars = json.loads(product['provisioning_json_vars'])
# Desde el cuerpo de la solicitud
request_vars = request_body_json
# Agregadas por el sistema
system_vars = {
    'product_id': 42,
    'customer_id': 123,
    'access_token': auth_token,  # Ver sección de autenticación a
continuación
    'initiating_user': 7
}
# Final fusionado
extra_vars = {**product_vars, **request_vars, **system_vars}
```

6. Registro de Aprovisionamiento Creado

Se crea un registro en la base de datos con estado 1 (En Ejecución):

```
provision = {
    'provision_id': 456,
    'customer_id': 123,
    'product_id': 42,
    'provisioning_play': 'play_psim_only',
    'provisioning_json_vars': json.dumps(extra_vars),
    'provisioning_status': 1,  # En Ejecución
    'task_count': 85,
    'initiating_user': 7,
    'created': '2025-01-10T14:30:00Z'
}
```

7. Hilo en Segundo Plano Generado

El sistema genera un hilo en segundo plano para ejecutar el playbook de manera asíncrona:

- Playbook: `play_psim_only.yaml`
- Variables: `extra_vars` fusionadas
- ID de Aprovisionamiento: 456 (para seguimiento de estado)
- Token de actualización: pasado para la actualización de token durante la ejecución

8. La API Retorna Inmediatamente

Se devuelve una respuesta a la UI con `provision_id`:

```
{
  "provision_id": 456,
  "provisioning_status": 1,
  "message": "Provisioning job created"
}
```

9. La UI Realiza Polling para Actualizaciones

La UI comienza a consultar `GET /crm/provision/provision_id/456` cada 3 segundos para verificar el estado. La respuesta incluye:

```
{
  "provision_id": 456,
  "provisioning_status": 1,
  "task_count": 12,
  "provisioning_result_json": [
    {
      "event_number": 1,
      "event_name": "Get Product information from CRM API",
      "provisioning_status": 0,
      "timestamp": "2024-01-15T10:30:05"
    },
    {
      "event_number": 2,
      "event_name": "Assign SIM Card from inventory",
      "provisioning_status": 1,
      "timestamp": "2024-01-15T10:30:07"
    }
  ]
}
```


10. Ansible Se Ejecuta en Segundo Plano

El playbook ejecuta tareas secuencialmente:

- La finalización de cada tarea crea un registro `Provision_Event` en la base de datos
- El evento incluye: nombre de la tarea, estado (0=éxito, 2=fallido, 3=fallido pero ignorado), resultado JSON
- La UI muestra el progreso en tiempo real mostrando tareas completadas y la tarea actualmente en ejecución
- Las tareas fallidas muestran mensajes de error en los detalles del evento

Seguimiento en la UI:

Mientras el aprovisionamiento se está ejecutando (estado 1), los usuarios pueden ver:

- **Página de Detalles del Servicio** - Muestra la insignia de estado de aprovisionamiento (En Ejecución/Éxito/Fallo)

- **Registro de Actividad** - Lista todos los eventos de aprovisionamiento con marcas de tiempo
- **Vista de Detalles de Aprovisionamiento** - Muestra el progreso tarea por tarea con expandir/colapsar para detalles

Ejemplo de visualización:

Estado de Aprovisionamiento: En Ejecución (8 de 12 tareas completadas)

✓ Obtener información del producto desde la API de CRM ✓ Obtener detalles del cliente ✓ Asignar tarjeta SIM del inventario (ICCID: 8991101200003204510) ✓ Asignar número de teléfono (555-0123) ⟳ Crear cuenta en OCS/CGRateS (en progreso...) □ Configurar políticas de red □ Crear registro de servicio ...

11. El Aprovisionamiento Se Completa

Estado final establecido:

- `provisioning_status: 0` - Éxito
- `provisioning_status: 2` - Fallido (error crítico)

La UI deja de consultar y muestra el resultado:

- **Éxito**: Marca de verificación verde, servicio marcado como Activo, el usuario puede ver los detalles del servicio
- **Fallo**: X roja, se muestra un mensaje de error, opción de reintentar o contactar soporte
- **Notificación por correo electrónico**: Si hay fallo, se envía un correo electrónico a `provisioning.failure_list` en la configuración

Autenticación y Autorización

Seguimiento de Usuarios

Cada trabajo de aprovisionamiento rastrea qué usuario lo inició:

- **Iniciado por el usuario**: El campo `initiating_user` se establece en el ID del usuario desde su token JWT

- **Autenticación por clave API**: Utiliza el ID del primer usuario administrador
- **Autenticación por lista blanca de IP**: Utiliza el ID del primer usuario administrador

Verificaciones de Permisos

El sistema verifica los permisos antes de permitir el aprovisionamiento:

- El personal necesita el permiso `CREATE_PROVISION`
- Los clientes solo pueden aprovisionar servicios para su propia cuenta (`VIEW_OWN_PROVISION` permiso)

Cómo Ansible Se Autentica con la API de CRM

Los playbooks de Ansible necesitan hacer llamadas API autenticadas de vuelta al CRM (para obtener detalles del producto, crear servicios, actualizar inventario, etc.). La autenticación se maneja a través de **tokens Bearer** pasados al playbook.

La fuente del `access_token` depende del método de autenticación utilizado para llamar a la API de aprovisionamiento:

Método 1: Inicio de Sesión del Usuario (Token Bearer)

Cuando un usuario inicia sesión a través de la UI web:

1. El usuario se autentica: `POST /crm/auth/login`
2. Recibe un `access_token` JWT (de corta duración, 15-30 min) y un `refresh_token` (de larga duración)
3. Realiza la solicitud de aprovisionamiento con el token Bearer en el encabezado
4. La API de aprovisionamiento extrae el token del encabezado `Authorization: Bearer ...`
5. Almacena el token para su uso durante el aprovisionamiento
6. Pasa a Ansible como variable `access_token`

Flujo de Implementación:

El sistema extrae el token Bearer del encabezado de Autorización, lo valida y decodifica, luego lo almacena para su uso durante el aprovisionamiento. El token se pasa al playbook como una variable adicional para que pueda autenticar las llamadas API subsiguientes.

Método 2: Clave API (Encabezado X-API-KEY)

Para sistemas automatizados que utilizan claves API:

1. El sistema realiza la solicitud: `PUT /crm/provision/` con el encabezado `X-API-KEY: your-api-key...`
2. La API de aprovisionamiento valida la clave API contra `crm_config.yaml`
3. **Genera un nuevo token JWT sobre la marcha** para el primer usuario administrador
4. Almacena el token para el aprovisionamiento
5. Pasa a Ansible

¿Por qué Generar un Token?

Las claves API son cadenas, no JWT. Los playbooks llaman a los endpoints API esperando autenticación JWT. Así que:

- Valida la clave API
- Si es válida y tiene rol de `admin`, genera un JWT temporal
- Utiliza el ID del primer usuario administrador como sujeto del JWT
- El token permite que el playbook haga llamadas API autenticadas

Flujo de Implementación:

El sistema valida la clave API contra las claves configuradas. Si es válida y tiene rol de administrador, genera un JWT temporal utilizando la identidad de un usuario administrador, luego lo almacena para su uso en el aprovisionamiento.

Método 3: Lista Blanca de IP

Para sistemas internos de confianza en redes privadas:

1. El sistema realiza la solicitud desde una IP en la lista blanca (por ejemplo, 192.168.1.100)

2. La API de aprovisionamiento verifica la IP del cliente contra `ip_whitelist` en `crm_config.yaml`
3. Si está en la lista blanca, **genera un nuevo token JWT** para el primer usuario administrador
4. Almacena el token para el aprovisionamiento
5. Pasa a Ansible

Flujo de Implementación:

El sistema verifica la IP del cliente contra la lista blanca. Si está permitido, genera un JWT para un usuario administrador y lo almacena para su uso en el aprovisionamiento.

Usando el Token en los Playbooks

Cada llamada API en el playbook incluye el token:

```
- name: Get Product Details
  uri:
    url: "http://localhost:5000/crm/product/product_id/{{
product_id }}"
    headers:
      Authorization: "Bearer {{ access_token }}"

- name: Create Service Record
  uri:
    url: "http://localhost:5000/crm/service/"
    method: PUT
    headers:
      Authorization: "Bearer {{ access_token }}"
    body:
      customer_id: "{{ customer_id }}"
      service_name: "Mobile Service"
```

Expiración y Actualización del Token

Los playbooks de larga duración (5-10 minutos) pueden superar el `access_token` (expiración de 15-30 min). Para aprovisionamientos iniciados por el usuario, el sistema pasa tanto `access_token` como `refresh_token`:

```
refresh_token = request.cookies.get('refresh_token')
run_playbook(playbook_path, extra_vars, provision_id,
refresh_token=refresh_token)
```

Si el `access_token` expira, el ejecutor del playbook puede:

1. Detectar respuesta 401 No Autorizado
2. Llamar a `POST /crm/auth/refresh` con `refresh_token`
3. Recibir un nuevo `access_token`
4. Reintentar la solicitud fallida

Para autenticación por clave API/lista blanca de IP, los tokens generados pueden tener una expiración más larga (1-2 horas) ya que estos son sistemas automatizados de confianza.

El Proceso de Aprovisionamiento

1. **Creación del Trabajo**

   Cuando se recibe una solicitud de aprovisionamiento, el sistema:

   - Valida la solicitud y verifica permisos
   - Carga el playbook de Ansible especificado en la definición del producto
   - Crea un registro de `Provision` en la base de datos con estado 1 (En Ejecución)
   - Extrae variables de la definición del producto y del cuerpo de la solicitud
   - Captura tokens de autenticación para el acceso a la API

2. **Manejo de Tokens**

   Los playbooks de Ansible necesitan autenticarse con la API de CRM para recuperar datos y hacer cambios. El sistema de aprovisionamiento maneja esto de dos maneras:

   - **Token Bearer (JWT)**: Para aprovisionamientos iniciados por el usuario, el `refresh_token` de la solicitud se utiliza para generar tokens de acceso frescos durante la ejecución del playbook

- **Clave API/Autenticación por IP**: Para sistemas automatizados, un `access_token` se pasa directamente al playbook

3. **Ejecución en Segundo Plano**

   El playbook se ejecuta en un hilo en segundo plano. Esto permite que la API devuelva inmediatamente mientras el aprovisionamiento continúa de manera asíncrona.

   Durante la ejecución:

   - La finalización/fallo de cada tarea crea un registro `Provision_Event`
   - El manejador de eventos monitorea fallos críticos frente a fallos ignorados
   - Actualizaciones de estado en tiempo real se escriben en la base de datos
   - La UI puede consultar actualizaciones a través de `GET /crm/provision/provision_id/<id>`

4. **Ejecución del Playbook**

   El playbook de Ansible generalmente realiza estas operaciones:

   - Recupera información del producto desde la API
   - Recupera información del cliente desde la API
   - Asigna elementos de inventario (tarjetas SIM, direcciones IP, números de teléfono, etc.)
   - Crea cuentas en OCS/OCS
   - Configura equipos de red
   - Crea el registro de servicio en la API de CRM
   - Añade transacciones de costo de configuración
   - Envía correos electrónicos/SMS de bienvenida a los clientes

5. **Manejo de Errores**

   Los playbooks de Ansible utilizan secciones `block` y `rescue` para el rollback:

- Si una tarea crítica falla, la sección de rescate elimina el aprovisionamiento parcial
- Las tareas con `ignore_errors: true` se marcan como estado 3 y no fallan el trabajo
- Errores fatales (errores de sintaxis YAML, fallos de conexión) crean un evento de error especial con información de depuración

Ejemplo: Playbook de Aprovisionamiento Estándar

Aquí hay un ejemplo de `play_simple_service.yaml`:

```
- name: Simple Provisioning Play
  hosts: localhost
  gather_facts: no
  become: False

  tasks:
    - name: Main block
      block:
      - name: Get Product information from CRM API
        uri:
          url: "http://localhost:5000/crm/product/product_id/{{
product_id }}"
          method: GET
          headers:
            Authorization: "Bearer {{ access_token }}"
          return_content: yes
          validate_certs: no
        register: api_response_product

      - name: Set package facts
        set_fact:
          package_name: "{{ api_response_product.json.product_name
}}"
          setup_cost: "{{
api_response_product.json.retail_setup_cost }}"
          monthly_cost: "{{ api_response_product.json.retail_cost
}}"

      - name: Generate Service UUID
        set_fact:
          service_uuid: "Service_{{ 99999999 | random | to_uuid
}}"

      - name: Create account in OCS
        uri:
          url: "http://{{ crm_config.ocs.OCS }}/jsonrpc"
          method: POST
          body_format: json
          body:
            {
              "method": "ApierV2.SetAccount",
              "params": [{
                  "Tenant": "{{ crm_config.ocs.ocsTenant }}",
```

```
                  "Account": "{{ service_uuid }}",
                  "ActionPlanIds": [],
                  "ExtraOptions": { "AllowNegative": false,
"Disabled": false }
                }]
              }
            status_code: 200
          register: response

      - name: Add Service via API
          uri:
            url: "http://localhost:5000/crm/service/"
            method: PUT
            body_format: json
            headers:
              Authorization: "Bearer {{ access_token }}"
            body:
              {
                "customer_id": "{{ customer_id }}",
                "product_id": "{{ product_id }}",
                "service_name": "Service: {{ service_uuid }}",
                "service_uuid": "{{ service_uuid }}",
                "service_status": "Active",
                "retail_cost": "{{ monthly_cost | float }}"
              }
            status_code: 200
          register: service_creation_response

      - name: Add Setup Cost Transaction
          uri:
            url: "http://localhost:5000/crm/transaction/"
            method: PUT
            headers:
              Authorization: "Bearer {{ access_token }}"
            body_format: json
            body:
              {
                "customer_id": {{ customer_id | int }},
                "service_id": {{
service_creation_response.json.service_id | int }},
                "title": "{{ package_name }} - Setup Costs",
                "retail_cost": "{{ setup_cost | float }}"
              }
          register: api_response_transaction
```

```
    rescue:
      - name: Remove account in OCS
        uri:
          url: "http://{{ crm_config.ocs.OCS }}/jsonrpc"
          method: POST
          body_format: json
          body:
            {
              "method": "ApierV2.RemoveAccount",
              "params": [{
                  "Tenant": "{{ crm_config.ocs.ocsTenant }}",
                  "Account": "{{ service_uuid }}"
              }]
            }
          status_code: 200

      - name: Fail the provision
        assert:
          that:
            - false
```

Este playbook demuestra el flujo típico:

1. Recuperar detalles del producto desde la API de CRM
2. Generar un UUID de servicio único
3. Crear la cuenta de facturación en OCS
4. Crear el registro de servicio a través de la API de CRM
5. Añadir transacciones de costo de configuración
6. Si algo falla, la sección `rescue` elimina la cuenta de OCS

Flujo de Trabajo de Aprovisionamiento Simple

El aprovisionamiento simple está diseñado para sistemas automatizados que necesitan activar el aprovisionamiento sin interacción del usuario. El caso de uso más común es el aprovisionamiento de complementos activado por OCS a través de ActionPlans.

Endpoints de Aprovisionamiento Simple

OmniCRM proporciona dos endpoints de aprovisionamiento simple:

- `POST /crm/provision/simple_provision_addon/service_id/<id>/product_id/<id>`

  Para aprovisionamiento automatizado de complementos (por ejemplo, cargos recurrentes, recargas automáticas)

- `POST /crm/provision/simple_provision_addon_recharge/service_id/<id>/product_id/<id>`

  Para operaciones de recarga rápida que necesitan retroalimentación inmediata

Autenticación para Aprovisionamiento Simple

Los endpoints de aprovisionamiento simple utilizan **lista blanca de IP** o **claves API** para autenticación:

- La IP de origen de la solicitud se verifica contra `ip_whitelist` en `crm_config.yaml`
- O se puede proporcionar una clave API de `api_keys` en `crm_config.yaml`
- Se genera un token de acceso y se pasa al playbook

Ejemplo: Callback de ActionPlan de OCS

OCS puede configurarse para llamar al endpoint de aprovisionamiento simple al ejecutar acciones recurrentes:

```
{
  "method": "ApierV1.SetActionPlan",
  "params": [{
    "Id": "ActionPlan_Service123_Monthly_Charge",
    "ActionsId": "Action_Service123_Add_Monthly_Data",
    "Timing": {
      "Years": [],
      "Months": [],
      "MonthDays": [1],
      "Time": "00:00:00Z"
    },
    "Weight": 10,
    "ActionTriggers": [
      {
        "ThresholdType": "*min_event_counter",
        "ThresholdValue": 1,
        "ActionsID": "Action_Service123_HTTP_Callback"
      }
    ]
  }]
}
```

La acción realiza un POST HTTP a:

Esto activa el playbook asociado (por ejemplo, `play_topup_no_charge.yaml`) que añade datos/créditos al servicio.

Ejemplo: Playbook de Recarga Simple

Desde `play_topup_monetary.yaml`:

```yaml
- name: Mobile Topup Monetary - 2024
  hosts: localhost
  gather_facts: no
  become: False

  tasks:
    - name: Get Product information from CRM API
      uri:
        url: "http://localhost:5000/crm/product/product_id/{{
product_id }}"
        method: GET
        headers:
          Authorization: "Bearer {{ access_token }}"
        return_content: yes
      register: api_response_product

    - name: Get Service information from CRM API
      uri:
        url: "http://localhost:5000/crm/service/service_id/{{
service_id }}"
        method: GET
        headers:
          Authorization: "Bearer {{ access_token }}"
        return_content: yes
      register: api_response_service

    - name: Set service facts
      set_fact:
        service_uuid: "{{ api_response_service.json.service_uuid
}}"
        customer_id: "{{ api_response_service.json.customer_id }}"
        package_name: "{{ api_response_product.json.product_name
}}"
        monthly_cost: "{{ api_response_product.json.retail_cost
}}"

    - name: Get Customer Payment Method
      uri:
        url: "http://localhost:5000/api/payments/methods/default?
customer_id={{ customer_id }}"
        method: GET
        headers:
          Authorization: "Bearer {{ access_token }}"
```

```
      return_content: yes
    register: api_response_payment_method

- name: Charge customer
  uri:
    url: "http://localhost:5000/api/payments/charge"
    method: POST
    headers:
      Authorization: "Bearer {{ access_token }}"
    body_format: json
    body:
      {
        "customer_id": "{{ customer_id | int }}",
        "amount": "{{ monthly_cost }}",
        "currency": "USD",
        "metadata": {
          "description": "{{ package_name }} topup",
          "service_id": "{{ service_id | int }}"
        }
      }
  register: api_response_payment

- name: Add monetary balance to OCS
  uri:
    url: "http://{{ crm_config.ocs.OCS }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
            "Tenant": "{{ crm_config.ocs.ocsTenant }}",
            "Account": "{{ service_uuid }}",
            "BalanceType": "*monetary",
            "Balance": {
                "Value": "{{ monthly_cost | float * 100 }}",
                "ExpiryTime": "+4320h"
            }
        }]
      }
    status_code: 200

- name: Add Transaction to CRM
  uri:
```

```
        url: "http://localhost:5000/crm/transaction/"
        method: PUT
        headers:
          Authorization: "Bearer {{ access_token }}"
        body_format: json
        body:
          {
            "customer_id": {{ customer_id | int }},
            "service_id": {{ service_id | int }},
            "title": "{{ package_name }}",
            "retail_cost": "{{ monthly_cost | float }}"
          }

    - name: Send Notification SMS
      uri:
        url: "http://sms-gateway/SMS/plaintext/{{ api_key }}"
        method: POST
        body_format: json
        body:
          {
            "source_msisdn": "YourCompany",
            "destinatination_msisdn": "{{ customer_phone }}",
            "message_body": "Thanks for topping up {{ monthly_cost
}}!"
          }
        status_code: 201
      ignore_errors: True
```

Este playbook:

1. Obtiene detalles del servicio y del producto desde la API
2. Recupera el método de pago del cliente
3. Carga al cliente a través de la API de Stripe
4. Añade saldo monetario a OCS
5. Registra la transacción en el CRM
6. Envía un SMS de confirmación (con `ignore_errors: True` para que los fallos no fallen el trabajo)

Cadenas de Aprovisionamiento

Para productos complejos que requieren múltiples pasos de aprovisionamiento, OmniCRM admite **cadenas de aprovisionamiento**. Una cadena ejecuta múltiples playbooks secuencialmente, pasando contexto entre ellos.

Ejemplo de caso de uso: Un servicio empaquetado que aprovisiona:

1. Servicio de internet base (crea el registro de servicio principal)
2. Complemento IPTV (utiliza el service_id del paso 1)
3. Complemento de IP estática (utiliza el service_id del paso 1)

El servicio de aprovisionamiento automáticamente:

- Consulta la base de datos para el `service_id` creado por el primer playbook
- Lo inyecta en los `extra_vars` para los playbooks subsiguientes
- Rastrear cada playbook como un registro `Provision` separado

Razones de Fallo y Depuración

Cuando el aprovisionamiento falla, el sistema captura información detallada para ayudar a diagnosticar el problema.

Escenarios Comunes de Fallo

Fallos de Tareas Críticas (Estado 2)

Estos causan que todo el trabajo de aprovisionamiento falle:

- Llamadas API que devuelven códigos de estado inesperados
- Fallos de afirmaciones (por ejemplo, `assert: that: response.status == 200`)
- Elementos de inventario requeridos que faltan
- Equipos de red inalcanzables
- Credenciales inválidas o tokens expirados

- Errores de OCS/OCS

Fallos Ignorados (Estado 3)

Estos se registran pero no fallan el trabajo:

- Fallos de notificaciones SMS/correo electrónico opcionales
- Búsquedas de datos no críticas (marcadas con `ignore_errors: True`)
- Operaciones de limpieza durante el desaprovisionamiento

Errores Fatales

Estos impiden que el playbook se ejecute en absoluto:

- Errores de sintaxis YAML en el playbook
- Variables indefinidas en el playbook
- Archivos de playbook faltantes
- Fallos de conexión al controlador de Ansible

Cuando ocurre un error fatal, el sistema crea un evento de error especial que contiene:

- El código de salida de Ansible
- Todo stdout (contiene detalles del error de sintaxis)
- Todo stderr (contiene errores de tiempo de ejecución)
- Una lista de causas comunes para ese tipo de fallo
- Todas las variables pasadas al playbook

Correos Electrónicos de Notificación de Errores

Cuando el aprovisionamiento falla (estado 2), se envía automáticamente un correo electrónico a la lista de notificación de fallos configurada (`provisioning.failure_list` en `crm_config.yaml`).

El correo electrónico incluye:

- Información del cliente
- Detalles del producto/servicio

- Resultados de tareas codificados por colores:
  - **Verde**: Tareas exitosas
  - **Naranja**: Tareas fallidas pero ignoradas
  - **Rojo**: Fallos críticos
- Para fallos críticos: Salida de depuración completa incluyendo cuerpos de solicitud/respuesta
- Para errores fatales: Salida de Ansible, mensajes de error y causas comunes

Monitoreo de Trabajos de Aprovisionamiento

OmniCRM proporciona métricas completas para monitorear el rendimiento del aprovisionamiento. Para detalles completos sobre métricas de aprovisionamiento, incluyendo duración del trabajo, tasas de éxito, conteos de tareas y seguimiento de errores, consulte Monitoreo & Métricas.

API de Estado de Aprovisionamiento

Para verificar el estado de un trabajo de aprovisionamiento:

```
GET /crm/provision/provision_id/<id>
Authorization: Bearer <token>
```

La respuesta incluye:

```
{
  "provision_id": 123,
  "customer_id": 456,
  "customer_name": "John Smith",
  "product_id": 10,
  "provisioning_status": 0,
  "provisioning_play": "play_psim_only",
  "playbook_description": "OmniCore Service Provisioning 2024",
  "task_count": 85,
  "provisioning_result_json": [
    {
      "event_number": 1,
      "event_name": "Get Product information from CRM API",
      "provisioning_status": 1,
      "provisioning_result_json": "{...}"
    },
    {
      "event_number": 2,
      "event_name": "Create account in OCS",
      "provisioning_status": 1,
      "provisioning_result_json": "{...}"
    }
  ]
}
```


Listando Trabajos de Aprovisionamiento

Para obtener una lista paginada de todos los trabajos de aprovisionamiento:

```
GET /crm/provision/?
page=1&per_page=20&sort=provision_id&order=desc
Authorization: Bearer <token>
```

Soporta filtrado:

```
GET /crm/provision/?filters={"provisioning_status":
[2]}&search=Mobile
Authorization: Bearer <token>
```

Esto devuelve solo trabajos fallidos (estado 2) donde la descripción contiene "Mobile".

Mejores Prácticas

Diseño del Playbook

- **Siempre usar block/rescue**: Asegúrese de que el aprovisionamiento parcial pueda ser revertido
- **Usar ignore_errors con juicio**: Solo para operaciones verdaderamente opcionales
- **Registrar variables importantes**: Utilice tareas `debug` para registrar valores clave para la solución de problemas
- **Validar respuestas**: Utilice `assert` para verificar que las respuestas de la API sean como se esperaba
- **Idempotencia**: Diseñe los playbooks para que sean seguros de volver a ejecutar

Autenticación

- **Aprovisionamiento iniciado por el usuario**: Siempre use `refresh_token` para playbooks de larga duración
- **Aprovisionamiento automatizado**: Utilice lista blanca de IP o claves API con tokens de acceso generados
- **Expiración del token**: El `refresh_token` asegura que los tokens de acceso se regeneren según sea necesario

Manejo de Errores

- **Proporcionar contexto**: Incluir `customer_id`, `service_id` y detalles de operación en mensajes de error
- **Notificar adecuadamente**: Fallos críticos desencadenan correos electrónicos, pero no spam para fallos esperados
- **Información de depuración**: Capturar cuerpos de solicitud/respuesta completos en registros `Provision_Event`

Seguridad

- **Validar entradas**: Verificar `customer_id`, `product_id`, `service_id` antes del aprovisionamiento
- **Verificaciones de permisos**: Verificar que los usuarios solo puedan aprovisionar para clientes autorizados
- **Datos sensibles**: Utilizar el sistema de redacción para eliminar contraseñas/claves de los registros
- **Lista blanca de IP**: Restringir endpoints de `simple_provision` a sistemas de confianza solamente

Rendimiento

- **Ejecución en segundo plano**: Nunca bloquear respuestas de API esperando aprovisionamiento
- **Intervalos de polling**: La UI debe consultar actualizaciones de estado cada 2-5 segundos
- **Tareas paralelas**: Utilizar la paralelización nativa de Ansible para operaciones independientes
- **Actualizaciones de base de datos**: El manejador de eventos actualiza la base de datos en tiempo real, no es necesario consultar durante la ejecución

Documentación Relacionada

- `concepts_ansible` - Conceptos generales de aprovisionamiento de Ansible
- `concepts_sim_provisioning` - Aprovisionamiento de tarjetas SIM (físicas y eSIM)
- `concepts_api` - Autenticación y uso de la API de CRM
- `concepts_products_and_services` - Definiciones de productos y servicios
- `administration_inventory` - Gestión de inventario para aprovisionamiento
- Monitoreo & Métricas - Métricas de rendimiento y monitoreo de aprovisionamiento

Aprovisionamiento de Tarjetas SIM

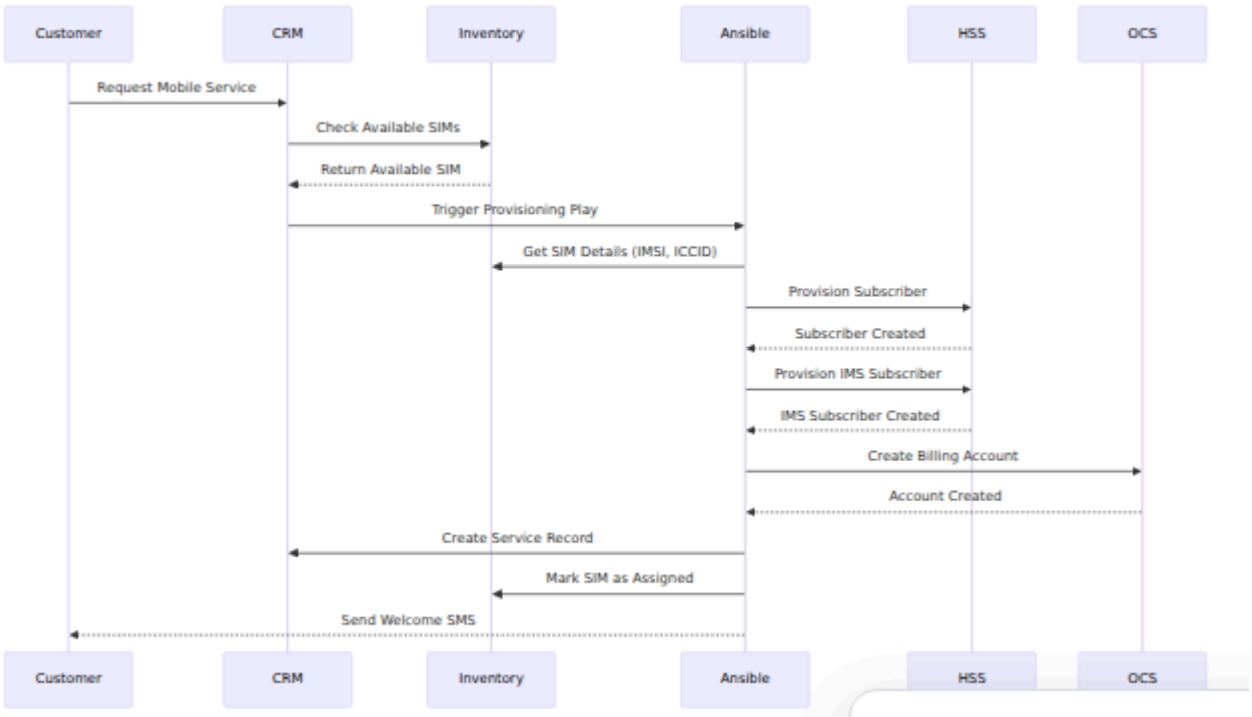
OmniCRM proporciona soporte integral para el aprovisionamiento tanto de **tarjetas SIM físicas** como de **eSIMs** (SIMs embebidas) para operadores de redes móviles y MVNOs. El sistema maneja el ciclo de vida completo desde la gestión de inventario hasta la activación, asignación y desprovisionamiento.

Véase también: Sistema de Aprovisionamiento para conceptos generales de aprovisionamiento, Inventario para la gestión de inventario, Playbooks de Ansible para la automatización del aprovisionamiento.

Resumen

El aprovisionamiento de SIM en OmniCRM implica varios sistemas integrados que trabajan juntos:

1. **Gestión de Inventario** - Rastrear tarjetas SIM disponibles (físicas y perfiles de eSIM)
2. **Integración HSS/IMS** - Aprovisiona credenciales de suscriptor y servicios de voz
3. **Integración OCS** - Configura la facturación y el cobro por el servicio
4. **Automatización de Ansible** - Orquesta el flujo de trabajo de aprovisionamiento
5. **Soporte de Autoregistro** - Permite a los clientes activar sus propias SIM



Aprovisionamiento de Tarjetas SIM Físicas

Las tarjetas SIM físicas son tarjetas SIM removibles tradicionales que los clientes insertan en sus dispositivos. OmniCRM gestiona estas a través del sistema de inventario y las aprovisiona al HSS/IMS para el acceso a la red.

Flujo de Trabajo de SIM Física

1. Configuración de Inventario

Las tarjetas SIM deben ser cargadas primero en el sistema de inventario:

- **ICCID** (Identificador de Circuito Integrado de Tarjeta) - Identificador único de la tarjeta SIM
- **IMSI** (Identidad Internacional de Suscriptor Móvil) - Identidad del suscriptor para la red

Estos se almacenan en los campos de elementos de inventario:

- `itemtext1`: ICCID

- `itemtext2`: IMSI
- `itemtext3`: Tipo de SIM (Física/eSIM) - opcional

Ejemplo de Elemento de Inventario de SIM Física:

```
{
  "inventory_id": 1001,
  "item": "SIM Card",
  "itemtext1": "89612345678901234567",  // ICCID
  "itemtext2": "310120123456789",       // IMSI
  "itemtext3": "Physical",               // SIM Type
  "item_location": "Warehouse A, Shelf 3",
  "item_state": "New",
  "wholesale_cost": 2.50,
  "retail_cost": 10.00
}
```

Para una explicación completa de todos los campos de inventario (itemtext1-20, valores de item_state, campos de dirección, management_url, etc.), consulte Resumen de Inventario - Campos de Elementos de Inventario.

Almacenamiento de Credenciales de Autenticación

Las credenciales de autenticación para las tarjetas SIM se almacenan en el **HSS AuC (Centro de Autenticación)**, no en el inventario de CRM:

- **Ki** (Clave de Autenticación) - Clave secreta para la autenticación (almacenada en HSS)
- **OPC** (Código de Operador) - Parámetro de autenticación específico del operador (almacenado en HSS)
- **PIN1/PIN2** - Códigos PIN del usuario (almacenados en HSS)
- **PUK1/PUK2** - Códigos de desbloqueo de PIN (almacenados en HSS)

El inventario de CRM rastrea qué SIM (por ICCID/IMSI) está asignada a qué cliente, mientras que el HSS maneja la autenticación real de la red utilizando las credenciales Ki/OPC que coinciden con la tarjeta SIM física.

2. Asignación de Servicio

Cuando un cliente solicita un servicio móvil:

1. El personal o el cliente selecciona una tarjeta SIM disponible del inventario
2. Se selecciona un número móvil (MSISDN) del inventario de números de teléfono
3. Se activa el playbook de aprovisionamiento del producto (por ejemplo, `play_psim_only.yaml`)

3. Aprovisionamiento HSS

El playbook de Ansible aprovisiona el suscriptor al Servidor de Suscriptores Local (HSS):

Primero, recupera el **auc_id** (referencia a las credenciales de autenticación ya almacenadas en HSS):

```
- name: Get AuC ID for IMSI
  uri:
    url: "{{ item }}/auc/imsi/{{ imsi }}"
    method: GET
  loop: "{{ hss_peers }}"
  register: auc_lookup

- name: Extract auc_id
  set_fact:
    auc_id: "{{ auc_lookup.results[0].json.auc_id }}"
```

Luego crea el registro de suscriptor haciendo referencia a ese auc_id:

```
- name: Provision subscriber on HSS
  uri:
    url: "{{ item }}/subscriber/"
    method: PUT
    body_format: json
    body:
      enabled: true
      roaming_enabled: true
      auc_id: "{{ auc_id }}"
      msisdn: "{{ phone_number }}"
      imsi: "{{ imsi }}"
      ue_ambr_dl: 9999999
      ue_ambr_ul: 9999999
      apn_list: "1,2,3"
      default_apn: 1
  loop: "{{ hss_peers }}"
```

Las credenciales de autenticación (Ki, OPC) permanecen en el HSS AuC y nunca se exponen al CRM o a los playbooks de aprovisionamiento.

4. Aprovisionamiento IMS (para Servicios de Voz)

Para la capacidad de llamada de voz, se crea un suscriptor IMS:

```
- name: Create IMS subscriber for voice services
  uri:
    url: "{{ item }}/ims_subscriber/"
    method: PUT
    body_format: json
    body:
      imsi: "{{ imsi }}"
      msisdn: "{{ phone_number }}"
      msisdn_list: "{{ phone_number }}"
      ifc_path: "default_ifc.xml"
      sh_profile: "{{ sh_profile_xml }}"
  loop: "{{ hss_peers }}"
```

5. Configuración del Sistema de Facturación

El OCS (Sistema de Cobro en Línea) se configura con:

- Creación de cuentas
- Mapeo IMSI/MSISDN a través de perfiles de atributos
- Reglas de filtro para identificar al suscriptor
- Límites de recursos (sesiones concurrentes)
- Saldo inicial

6. Creación de Servicio y Asignación de Inventario

Finalmente:

- Se crea un registro de servicio en el CRM
- El elemento de inventario de la tarjeta SIM se asigna al cliente (`customer_id` establecido)
- La tarjeta SIM se vincula al servicio (`service_id` establecido)
- El estado del inventario se actualiza a `"Assigned"`

Autenticación de SIM Física

Las SIM físicas utilizan credenciales de AuC (Centro de Autenticación) almacenadas en el HSS:

- **Ki** y **OPC** se utilizan para la autenticación 4G/5G (algoritmo MILENAGE)
- Estas credenciales se cargan previamente en la base de datos AuC del HSS durante la importación de SIM
- La tarjeta SIM física contiene valores coincidentes de Ki/OPC
- Durante el aprovisionamiento, el playbook hace referencia al **auc_id** para vincular al suscriptor con estas credenciales
- La autenticación de la red ocurre a través de un desafío-respuesta entre la SIM y el HSS
- El CRM nunca ve ni maneja los valores reales de Ki/OPC

Aprovisionamiento de eSIM

Las eSIM (SIMs embebidas) son perfiles de SIM basados en software que se pueden descargar a dispositivos compatibles sin tarjetas SIM físicas. OmniCRM

admite el aprovisionamiento de eSIM con códigos de activación LPA (Asistente de Perfil Local).

Características de eSIM

Códigos de Activación LPA

Las eSIM utilizan códigos LPA para la activación:

```
LPA:1$smdp.example.com$ACTIVATION-CODE-ABC123XYZ
```

Donde:

- `LPA:1` - Identificador de versión LPA
- `smdp.example.com` - Dirección del servidor SM-DP+ (Preparación de Datos del Administrador de Suscripciones)
- `ACTIVATION-CODE-ABC123XYZ` - Código de activación único para este perfil de eSIM

Generación de Código QR

OmniCRM genera automáticamente códigos QR a partir de códigos de activación LPA:

- Almacenados en el campo `management_url` del inventario
- Mostrados en la interfaz de usuario como código QR escaneable
- Los clientes escanean con la cámara de su dispositivo para instalar el perfil de eSIM
- No se requiere la escritura manual de largos códigos de activación

Ejemplo de Elemento de Inventario de eSIM:

```
{
  "inventory_id": 1002,
  "item": "eSIM",
  "itemtext1": "89612345678901234567",
  "itemtext2": "310120123456790",
  "itemtext3": "eSIM",
  "management_url": "LPA:1$smdp.example.com$ACTIVATION-CODE-
ABC123XYZ",
  "item_location": "Virtual Inventory",
  "item_state": "New",
  "wholesale_cost": 0.00,
  "retail_cost": 0.00
}
```

Al ver este elemento de inventario en la interfaz de usuario, se genera y muestra automáticamente un código QR para facilitar el escaneo.

Flujo de Trabajo de Aprovisionamiento de eSIM

El flujo de trabajo de aprovisionamiento de eSIM es casi idéntico al aprovisionamiento de SIM física, con algunas diferencias clave:

1. Asignación Automática de eSIM

Si un producto requiere una SIM pero no se selecciona una SIM física, el sistema puede asignar automáticamente una eSIM disponible:

```
# Búsqueda automática de eSIM
search_filters = {
    "customer_id": [None],
    "item_state": ["New"],
    "itemtext3": ["eSIM"]  # Filtrar por tipo eSIM
}
available_esim = search_inventory("SIM Card",
filters=search_filters)
```

2. Aprovisionamiento HSS/IMS

Las eSIM se aprovisionan al HSS/IMS exactamente como las SIM físicas:

- Mismo aprovisionamiento IMSI
- Creación del mismo suscriptor IMS
- Mismas credenciales de autenticación (Ki, OPC)

3. Activación del Cliente

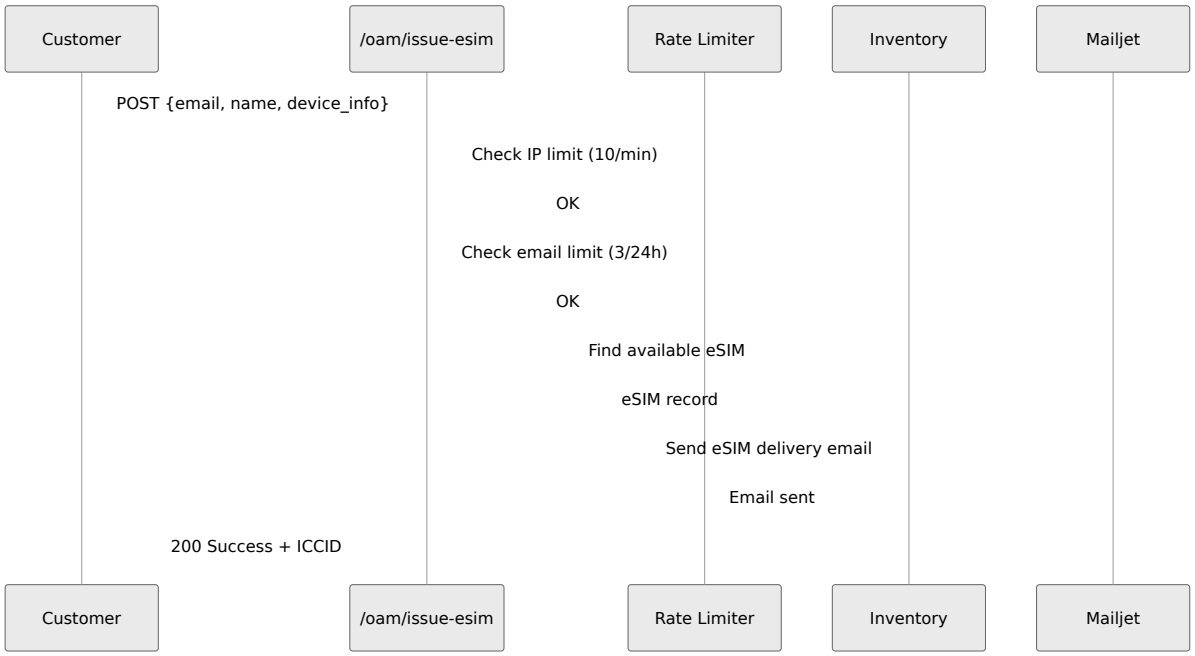
Después del aprovisionamiento:

1. El cliente recibe detalles de activación de eSIM por correo electrónico o en el portal del cliente
2. Se muestra el código QR para escaneo
3. El cliente escanea el código QR con su dispositivo
4. El dispositivo descarga el perfil de eSIM del servidor SM-DP+
5. La eSIM se instala y está lista para usar

API Pública del Emisor de eSIM

OmniCRM proporciona un punto final de API pública para solicitudes de eSIM de autoservicio, permitiendo a los clientes solicitar una eSIM directamente desde un sitio web o aplicación móvil sin intervención del personal.

Punto final: `POST /crm/oam/issue-esim`



Parámetros de Solicitud

Parámetro	Tipo	Requerido	Predetermin
`email`	String	Sí	-
`name`	String	No	Derivado del correo electró
`customer_id`	Integer	No	`null`
`device_info`	Object	No	`{}`

Parámetro	Tipo	Requerido	Predetermir
`device_info.compatibility_status`	String	No	`"unknown"`
`device_info.user_agent`	String	No	Encabezado d solicitud

Limitación de Tasa

El punto final implementa una limitación de tasa dual para prevenir abusos:

Límite	Ventana	Alcance	Descripción
10 solicitudes	1 minuto	Por dirección IP	Límite de tasa general aplicado a través de decorador
3 solicitudes	24 horas	Por dirección de correo electrónico	Previene solicitudes excesivas de eSIM al mismo correo electrónico

Cuando se superan los límites de tasa, el punto final devuelve HTTP 429 con `error_type: "RateLimitExceeded"`.

Formato de Respuesta

Respuesta de Éxito (HTTP 200):

```
{
  "result": "Success",
  "message": "eSIM request received. You will receive your eSIM QR code via email shortly.",
  "esim_id": 8472,
  "iccid": "89442000000001234567",
  "email_sent": true
}
```

Campo	Tipo	Descripción
`result`	String	Siempre `"Success"` para respuestas HTTP 200
`message`	String	Mensaje de confirmación legible para humanos
`esim_id`	Integer	El ID de inventario de la eSIM asignada
`iccid`	String	El ICCID de la eSIM asignada
`email_sent`	Boolean	Si el correo electrónico de entrega fue enviado con éxito

Respuestas de Error

Código HTTP	Tipo de Error	Descripción
400	-	Dirección de correo electrónico inválida o faltante
429	`RateLimitExceeded`	Límite de tasa por correo electrónico o por IP excedido
503	`NoInventory`	No hay eSIMs disponibles en el inventario
500	-	Error interno del servidor

Formato de Respuesta de Error:

```
{
  "result": "Failed",
  "reason": "Error description",
  "error_type": "ErrorType"
}
```

Entrega de Correo Electrónico

El punto final envía un correo electrónico de entrega de eSIM a través de Mailjet que contiene:

- ICCID
- Código QR (si está disponible en `itemtext2`)
- Código de activación/cadena LPA (si está disponible en `itemtext3` o `management_url`)
- Instrucciones de activación paso a paso

Configure la plantilla de Mailjet en `crm_config.yaml`:

```yaml
mailjet:
  api_crmCommunicationEsimDelivery:
    from_email: "support@aimobile.ac"
    from_name: "Example Mobile"
    template_id: 1234567
    subject: "Your eSIM is Ready"
```

Parámetro	Tipo	Requerido	Descripción
`from_email`	String	Sí	Dirección de correo electrónico del remitente
`from_name`	String	Sí	Nombre de visualización del remitente
`template_id`	Integer	No	ID de plantilla de Mailjet. Si no se proporciona, se genera un correo electrónico HTML predeterminado.
`subject`	String	Sí	Línea de asunto del correo electrónico

Si no se configura ninguna plantilla, el sistema genera un correo electrónico HTML con marca que incluye:

- Marca y logotipo de la empresa
- Código QR para la activación de eSIM (si está disponible)
- Instrucciones de código de activación manual
- Guía de configuración específica para dispositivos iOS y Android

Registro de Actividades

Cada solicitud de eSIM crea un registro de actividad con:

- Tipo de actividad: `esim_request`
- Vínculo de ID de cliente (si se proporciona)
- Estado de compatibilidad del dispositivo
- Información del agente de usuario

Esto permite el seguimiento de solicitudes de eSIM y análisis de conversión.

4. Estructura de Datos de eSIM en HSS

Las eSIM en el HSS AuC (Centro de Autenticación) están marcadas con una bandera `esim: true`:

```
{
  "esim": true,
  "lpa": "LPA:1$smdp.example.com$ACTIVATION-CODE-ABC123XYZ",
  "iccid": "89612345678901234567",
  "imsi": "310120123456790",
  "ki": "00112233445566778899AABBCCDDEEFF",
  "opc": "FFEEDDCCBBAA99887766554433221100",
  "pin1": "1234",
  "pin2": "5678",
  "puk1": "12345678",
  "puk2": "87654321",
  "batch_name": "eSIM_Batch_2024_Q1",
  "sim_vendor": "Thales"
}
```

Proceso de Importación de eSIM

Las eSIM se importan típicamente en bloque desde el proveedor de eSIM:

Script de Importación: `/OmniCRM-API/Provisioners/customer_product_creation/import_keys_esim.py`

El proceso de importación:

1. Lee datos de eSIM de archivos de Excel proporcionados por el proveedor
2. Carga códigos de activación LPA desde archivos CSV
3. Actualiza el HSS con las credenciales de eSIM
4. Crea elementos de inventario correspondientes en el CRM
5. Vincula perfiles de eSIM al sistema de inventario

Esto asegura que los perfiles de eSIM estén disponibles para asignación mientras se mantiene un seguimiento adecuado del inventario.

Integración de Inventario

El sistema de inventario es central para el aprovisionamiento de SIM, rastreando tanto recursos físicos como virtuales necesarios para servicios móviles.

Plantilla de Inventario de Tarjeta SIM

Una plantilla típica de inventario de tarjeta SIM define:

```
{
  "item": "SIM Card",
  "itemtext1_label": "ICCID",
  "itemtext2_label": "IMSI",
  "itemtext3_label": "SIM Type",
  "wholesale_cost": 2.50,
  "retail_cost": 10.00
}
```

Nota: Las credenciales de autenticación (Ki, OPC, PIN, PUK) se almacenan en el HSS AuC, no en el inventario de CRM.

Para obtener información detallada sobre cómo crear y gestionar plantillas de inventario, incluyendo cómo definir campos personalizados (itemtext1-20) y vincular plantillas a productos, consulte la sección Gestión de Inventario - Plantillas de Inventario.

Inventario de Números Móviles

Los números de teléfono se gestionan como elementos de inventario separados:


```
{
  "inventory_id": 4001,
  "item": "Mobile Number",
  "itemtext1": "+61412345678",
  "itemtext2": "Melbourne",
  "itemtext3": "Mobile",
  "item_location": "Australia - VIC",
  "item_state": "New",
  "wholesale_cost": 1.00,
  "retail_cost": 0.00
}
```


Estados de Inventario para SIMs

Las tarjetas SIM progresan a través de varios estados de inventario:

- **New** - SIM no utilizada, disponible para asignación
- **Assigned** - Actualmente activa con un cliente
- **Used** - Anteriormente asignada, devuelta al inventario (puede ser reutilizada)
- **Internal Use** - Para pruebas o uso del personal
- **Damaged** - No funcional, requiere reemplazo
- **Lost** - No se puede localizar
- **Stolen** - Reportada como robada

Integración de Productos

Los productos especifican el inventario requerido a través de `inventory_items_list`:

```
{
  "product_id": 1,
  "product_slug": "Mobile-SIM",
  "product_name": "Mobile SIM Only",
  "provisioning_play": "play_psim_only",
  "inventory_items_list": "['SIM Card', 'Mobile Number']"
}
```

Al aprovisionar este producto:

1. La interfaz de usuario muestra un selector de inventario con dos menús desplegables
2. El usuario selecciona la tarjeta SIM disponible (física o eSIM)
3. El usuario selecciona el número móvil disponible
4. Los IDs de inventario se pasan al playbook de Ansible
5. El playbook recupera todos los detalles del inventario a través de la API
6. El aprovisionamiento procede con los recursos seleccionados

Para una explicación completa de cómo los productos impulsan el aprovisionamiento, cómo se definen los requisitos de inventario y cómo se pasan las variables a los playbooks, consulte Sistema de Aprovisionamiento - Cómo los Productos Impulsan el Aprovisionamiento.

Flujo de Asignación de Inventario

El playbook de Ansible maneja la asignación de inventario.

Para más detalles sobre cómo los playbooks se autentican con la API de CRM (tokens Bearer, claves API, tokens de actualización), consulte Sistema de Aprovisionamiento - Autenticación y Autorización.

```
- name: Get SIM Card details from inventory
  uri:
    url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ inventory_id_sim_card }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
  register: sim_inventory_response

- name: Extract SIM identifiers
  set_fact:
    iccid: "{{ sim_inventory_response.json.itemtext1 }}"
    imsi: "{{ sim_inventory_response.json.itemtext2 }}"

- name: Get AuC ID from HSS (reference to authentication
credentials)
  uri:
    url: "{{ item }}/auc/imsi/{{ imsi }}"
    method: GET
  loop: "{{ hss_peers }}"
  register: auc_response

- name: Extract AuC ID
  set_fact:
    auc_id: "{{ auc_response.results[0].json.auc_id }}"

# ... provision to HSS/IMS/OCS ...

- name: Assign SIM to customer
  uri:
    url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ inventory_id_sim_card }}"
    method: PATCH
    headers:
      Authorization: "Bearer {{ access_token }}"
    body_format: json
    body:
      customer_id: "{{ customer_id }}"
      service_id: "{{ service_id }}"
      item_state: "Assigned"
```

Esto asegura:

- La SIM se marca como asignada al cliente
- El ID del servicio está vinculado para seguimiento
- El estado del inventario refleja el estado actual
- Otros sistemas pueden consultar el inventario para encontrar la SIM del cliente

Autoregistro con Aprovisionamiento de SIM

OmniCRM admite el registro de autoservicio del cliente donde los clientes pueden activar sus propias tarjetas SIM o eSIM sin intervención del personal.

Flujo de Autoregistro

1. El Cliente Inicia el Registro

El cliente visita la página de autoregistro y comienza la creación de la cuenta:

- Proporciona información personal
- Selecciona un plan de servicio (producto)
- Ingresa detalles de pago

2. Validación del Número de SIM

El cliente puede proporcionar su propio número de SIM (opcional):

```
GET /auth/self-signup/validate-itemtext1/{sim_number}
```

Este punto final verifica si:

- La SIM existe en el inventario
- La SIM está disponible (no ya asignada)
- La SIM está en el estado correcto ("New" o "Used")

3. Asignación Automática de SIM (Si No Se Proporciona SIM)

Si el cliente no tiene una SIM, el sistema asigna automáticamente una disponible:

```
def _populate_base_plan_inventory(sim_number, plan):
    if sim_number:
        # El cliente proporcionó SIM - verificar que existe
        sim_inventory = get_inventory_by_itemtext1(session,
sim_number, None)
        if not sim_inventory:
            # Verificar si hay un servicio inactivo con este
número
            service = get_inactive_service_by_sim(sim_number)
            if service:
                sim_inventory =
get_sim_for_service(service.service_id)
    else:
        # Asignar automáticamente una SIM disponible del
inventario
        sim_inventory =
get_first_available_inventory_by_item_name(
            session,
            "SIM Card",
            filters={"item_state": ["New"], "customer_id": [None]}
        )

    # Obtener ICCID y MSISDN
    iccid = sim_inventory.itemtext1

    # Asignar automáticamente el número de teléfono
    number_inventory = get_first_available_inventory_by_item_name(
        session,
        "Mobile Number"
    )
    msisdn = number_inventory.itemtext1

    return {
        "SIM Card": sim_inventory.inventory_id,
        "Mobile Number": number_inventory.inventory_id
    }
```

4. Creación de Cuenta y Aprovisionamiento

Una vez que se autoriza el pago:

```
def _post_signup_tasks(customer_id, plan, payment_method):
    # 1. Guardar método de pago
    save_stripe_payment_method(customer_id, payment_method)

    # 2. Cargar la tarjeta del cliente
    charge_customer_card(customer_id, plan.retail_setup_cost)

    # 3. Iniciar sesión del usuario (obtener tokens JWT)
    tokens = login_customer(customer_id)

    # 4. Llenar inventario (SIM y número de teléfono)
    inventory_vars = _populate_base_plan_inventory(sim_number,
plan)

    # 5. Crear trabajo de aprovisionamiento
    provision_vars = {
        "product_id": plan.product_id,
        "customer_id": customer_id,
        **inventory_vars  # Incluir IDs de SIM Card y Mobile
Number
    }

    # Opcional: Encadenar productos adicionales
    if plan.addon_chain:
        provision_vars["addon_chain"] = plan.addon_chain

    provision_id = create_provisioning_job(provision_vars)

    # 6. Devolver éxito con tokens
    return {
        "access_token": tokens.access_token,
        "refresh_token": tokens.refresh_token,
        "customer_id": customer_id,
        "provision_id": provision_id
    }
```

5. El Cliente Recibe Detalles de Activación

Después de que se complete el aprovisionamiento:

- **SIM Física**: Correo electrónico de bienvenida con número de teléfono e instrucciones de activación
- **eSIM**: El correo electrónico incluye un código QR para la descarga del perfil de eSIM, o el cliente puede ver el código QR en el portal del cliente

6. Permisos Especiales para Autoregistro

Durante el registro, el token JWT incluye un reclamo especial:

```
{
  "sub": "customer_123",
  "signup_process": true,
  "permissions": ["VIEW_OWN_PROVISION", "UPDATE_OWN_INVENTORY"]
}
```

Esto permite:

- Al cliente leer/actualizar el inventario durante el registro
- Restringido solo a su propia cuenta de cliente
- Previene la asignación de SIMs que pertenecen a otros clientes

Seguridad de Autoregistro

Verificaciones de Validación:

- La SIM debe existir en el inventario
- La SIM no debe estar asignada a otro cliente
- La SIM debe estar en un estado aceptable ("New", "Used")
- El cliente solo puede asignar SIMs a su propia cuenta
- El pago debe ser autorizado antes del aprovisionamiento

Salvaguardas de Inventario:

- Las transacciones de base de datos aseguran una asignación atómica
- Se previenen intentos de registro concurrentes mediante bloqueos de base de datos
- Los cambios en el estado del inventario se registran para auditoría

Aprovisionamiento Basado en Ansible

Todo el aprovisionamiento de SIM se orquesta a través de playbooks de Ansible, proporcionando un proceso consistente, auditable y repetible.

Para documentación completa sobre la estructura del playbook de Ansible, el patrón block/rescue, cómo interactúan los playbooks con los productos y mejores prácticas, consulte Playbooks de Ansible: Guía Detallada.

Playbook Principal de Aprovisionamiento de SIM

Archivo: `/OmniCRM-API/Provisioners/plays/play_psim_only.yaml`

Este es el playbook principal para el aprovisionamiento de SIM móviles (tanto físicas como eSIM). Maneja:

- Servicios de voz y datos móviles
- Servicios solo de datos
- Suscriptores IMS solo de voz

Estructura del Playbook (1,561 líneas):

Para una explicación detallada de los encabezados del playbook, las directivas hosts/gather_facts/become y la estructura general, consulte Playbooks de Ansible - Estructura y Anatomía del Playbook.

```yaml
- name: OmniCore Service Provisioning 2024
  hosts: localhost
  gather_facts: no
  become: False

  tasks:
    - name: Main provisioning block
      block:
        # --- PRE-PROVISIONING (Lines 240-396) ---
        - name: Get SIM information from inventory
          # Retrieves IMSI, ICCID, etc.

        - name: Validate IMSI length
          # Must be exactly 15 digits

        - name: Check if IMSI exists in HSS
          # Prevents duplicate provisioning

        - name: Get AuC ID for IMSI
          # Retrieves auc_id reference from HSS

        # --- HSS PROVISIONING (Lines 398-534) ---
        - name: Create/Update HSS Subscriber
          # Provisions to all HSS peers

        - name: Create IMS Subscriber
          # Enables voice calling

        # --- OCS SETUP (Lines 536-837) ---
        - name: Create ENUM entry
          # E.164 number mapping

        - name: Create FilterS rule
          # Account identification

        - name: Create AttributeS profile
          # IMSI/MSISDN mapping

        - name: Create ResourceS profile
          # Concurrent session limits

        - name: Create OCS account
          # Billing account creation
```

```
    # --- SERVICE & INVENTORY (Lines 865-946) ---
    - name: Create service record
      # In CRM database

    - name: Assign SIM to service
      # Update inventory

    - name: Send welcome SMS
      # Customer notification

  rescue:
    # --- CLEANUP/DEPROVISIONING (Lines 975-1562) ---
    - name: Return inventory to pool
      # Reset customer_id, service_id to null

    - name: Delete OCS account
      # Remove billing account

    - name: Remove HSS subscribers
      # Or set to dormant state

    - name: Determine success/failure
      assert:
        that:
          - action == "deprovision"
```

Pasos Clave de Aprovisionamiento

1. Validación Pre-Aprovisionamiento

```
- name: Get SIM information from CRM inventory
  uri:
    url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ inventory_id_sim_card }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
  register: api_response_sim

- name: Extract and validate IMSI
  set_fact:
    imsi: "{{ api_response_sim.json.itemtext2 }}"

- name: Validate IMSI format
  assert:
    that:
      - imsi | length == 15
      - imsi is match('^[0-9]+$')
    fail_msg: "IMSI must be exactly 15 digits"
```

2. Recuperar AuC ID del HSS

```
- name: Get AuC ID for IMSI from HSS
  uri:
    url: "{{ item }}/auc/imsi/{{ imsi }}"
    method: GET
  loop: "{{ crm_config.hss.hss_peers }}"
  register: auc_lookup

- name: Extract auc_id from first HSS response
  set_fact:
    auc_id: "{{ auc_lookup.results[0].json.auc_id }}"
```

3. Aprovisionamiento de Suscriptor HSS

```yaml
- name: Create subscriber in each HSS peer
  uri:
    url: "{{ item }}/subscriber/"
    method: PUT
    body_format: json
    body:
      enabled: true
      roaming_enabled: true
      auc_id: "{{ auc_id }}"
      msisdn: "{{ phone_number }}"
      imsi: "{{ imsi }}"
      ue_ambr_dl: 9999999  # Límite de velocidad de descarga
      ue_ambr_ul: 9999999  # Límite de velocidad de subida
      apn_list: "1,2,3"
      default_apn: 1
    status_code: [200, 201]
  loop: "{{ crm_config.hss.hss_peers }}"
  register: hss_responses
```

4. Aprovisionamiento de Suscriptor IMS (Voz)

```
- name: Create IMS subscriber for voice services
  uri:
    url: "{{ item }}/ims_subscriber/"
    method: PUT
    body_format: json
    body:
      imsi: "{{ imsi }}"
      msisdn: "{{ phone_number }}"
      msisdn_list: "{{ phone_number }}"
      ifc_path: "default_ifc.xml"
      sh_profile: |
        <?xml version="1.0"?>
        <IMSSubscription>
          <PrivateID>{{ imsi }}@ims.mnc{{ mnc }}.mcc{{ mcc
}}.3gppnetwork.org</PrivateID>
          <ServiceProfile>
            <PublicIdentity>
              <Identity>sip:{{ phone_number }}@ims.mnc{{ mnc
}}.mcc{{ mcc }}.3gppnetwork.org</Identity>
            </PublicIdentity>
          </ServiceProfile>
        </IMSSubscription>
    status_code: [200, 201]
  loop: "{{ crm_config.hss.hss_peers }}"
```

5. Configuración de Facturación OCS

```yaml
# Crear filtros de cuenta (identificar suscriptor por IMSI/MSISDN)
- name: Create FilterS for IMSI identification
  uri:
    url: "{{ crm_config.ocs.ocsApi }}/jsonrpc"
    method: POST
    body_format: json
    body:
      method: "ApierV1.SetFilter"
      params:
        - ID: "FILTER_IMSI_{{ service_uuid }}"
          Type: "*string"
          Element: "~*req.OriginHost"
          Values: ["{{ imsi }}"]

# Crear perfil de atributos (mapear IMSI a cuenta)
- name: Create AttributeS profile
  uri:
    url: "{{ crm_config.ocs.ocsApi }}/jsonrpc"
    method: POST
    body_format: json
    body:
      method: "ApierV1.SetAttributeProfile"
      params:
        - ID: "ATTR_ACCOUNT_{{ service_uuid }}"
          FilterIDs: ["FILTER_IMSI_{{ service_uuid }}"]
          Attributes:
            - Path: "*req.Account"
              Value: "{{ service_uuid }}"

# Crear cuenta de facturación
- name: Create OCS account
  uri:
    url: "{{ crm_config.ocs.ocsApi }}/jsonrpc"
    method: POST
    body_format: json
    body:
      method: "ApierV2.SetAccount"
      params:
        - Tenant: "{{ crm_config.ocs.ocsTenant }}"
          Account: "{{ service_uuid }}"
          ActionPlanIds: []
          ExtraOptions:
            AllowNegative: false
```

```
            Disabled: false

# Agregar saldo inicial
- name: Add initial monetary balance
  uri:
    url: "{{ crm_config.ocs.ocsApi }}/jsonrpc"
    method: POST
    body_format: json
    body:
      method: "ApierV1.AddBalance"
      params:
        - Tenant: "{{ crm_config.ocs.ocsTenant }}"
          Account: "{{ service_uuid }}"
          BalanceType: "*monetary"
          Balance:
            Value: 0
            ExpiryTime: "+8760h"  # 1 año
```

6. Creación de Servicio

```
- name: Create service record in CRM
  uri:
    url: "{{ crm_config.crm.base_url }}/crm/service/"
    method: PUT
    headers:
      Authorization: "Bearer {{ access_token }}"
    body_format: json
    body:
      customer_id: "{{ customer_id }}"
      product_id: "{{ product_id }}"
      service_name: "Mobile Service - {{ phone_number }}"
      service_uuid: "{{ service_uuid }}"
      service_status: "Active"
      retail_cost: "{{ monthly_cost }}"
    status_code: 200
  register: service_creation_response

- name: Extract service_id
  set_fact:
    service_id: "{{ service_creation_response.json.service_id }}"
```

7. Asignación de Inventario

```
- name: Assign SIM card to customer and service
  uri:
    url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ inventory_id_sim_card }}"
    method: PATCH
    headers:
      Authorization: "Bearer {{ access_token }}"
    body_format: json
    body:
      customer_id: "{{ customer_id }}"
      service_id: "{{ service_id }}"
      item_state: "Assigned"
      sold_date: "{{ ansible_date_time.iso8601 }}"

- name: Assign phone number to customer and service
  uri:
    url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ inventory_id_phone_number }}"
    method: PATCH
    headers:
      Authorization: "Bearer {{ access_token }}"
    body_format: json
    body:
      customer_id: "{{ customer_id }}"
      service_id: "{{ service_id }}"
      item_state: "Assigned"
      sold_date: "{{ ansible_date_time.iso8601 }}"
```

Desprovisionamiento y Reversión

El mismo playbook maneja tanto la reversión de aprovisionamiento fallido como el desprovisionamiento intencional utilizando el bloque `rescue` de Ansible.

Para una explicación detallada del patrón block/rescue y por qué es una mejor práctica, consulte Sistema de Aprovisionamiento - Reversión y Limpieza: Patrón de Mejor Práctica.

```yaml
rescue:
  # Esta sección se ejecuta cuando:
  # 1. Cualquier tarea en el bloque falla (reversión automática)
  # 2. action == "deprovision" (limpieza intencional)

  - name: Get inventory items linked to service
    uri:
      url: "{{ crm_config.crm.base_url
}}/crm/inventory/customer_id/{{ customer_id }}"
      method: GET
      headers:
        Authorization: "Bearer {{ access_token }}"
    register: inventory_items
    ignore_errors: true

  - name: Return inventory to available pool
    uri:
      url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ item.inventory_id }}"
      method: PATCH
      headers:
        Authorization: "Bearer {{ access_token }}"
      body_format: json
      body:
        customer_id: null
        service_id: null
        item_state: "Used"
    loop: "{{ inventory_items.json.data }}"
    when: inventory_items.json.data is defined
    ignore_errors: true

  - name: Delete OCS account
    uri:
      url: "{{ crm_config.ocs.ocsApi }}/jsonrpc"
      method: POST
      body_format: json
      body:
        method: "ApierV1.RemoveAccount"
        params:
          - Tenant: "{{ crm_config.ocs.ocsTenant }}"
            Account: "{{ service_uuid }}"
    ignore_errors: true
```

```
  - name: Delete or disable HSS subscriber
    uri:
      url: "{{ item.key }}/subscriber/{{ item.value.subscriber_id
}}"
      method: DELETE
    loop: "{{ hss_subscriber_data | dict2items }}"
    when: deprovision_subscriber | default(false) | bool
    ignore_errors: true

  # La afirmación final determina el éxito o el fracaso
  - name: Determine if this was intentional deprovision or failed
provision
    assert:
      that:
        - action == "deprovision"
      fail_msg: "Provisioning failed and rollback completed"
      success_msg: "Deprovisioning completed successfully"
```

Puntos Clave:

- `ignore_errors: true` en todas las tareas de limpieza asegura que se realicen todos los intentos de limpieza
- Si `action == "deprovision"`, la afirmación pasa (estado 0 = éxito)
- Si `action` no está configurado, la afirmación falla (estado 2 = fallo de aprovisionamiento)
- Esto asegura operaciones atómicas: o se aprovisiona completamente o se limpia completamente

Otros Playbooks Relacionados con SIM

Playbook de Desarrollo Local: `play_local_mobile_sim.yaml` - Versión de desarrollo que omite dependencias externas de HSS/OCS

Intercambio de MSISDN: `play_swap_msisdn.yaml` - Intercambia números de teléfono entre servicios existentes

Playbooks de Recarga/Recarga:

- `play_topup_monetary.yaml` - Agregar saldo monetario
- `play_topup_no_charge.yaml` - Agregar datos/minutos gratuitos

- `play_topup_dongle.yaml` - Recarga para dispositivos IoT/datos

Integración HSS e IMS

OmniCRM se integra con un Servidor de Suscriptores Local (HSS) externo para la gestión y autenticación de suscriptores.

Funciones de HSS

Archivo: `/OmniCRM-API/Provisioners/hss.py`

Funciones Clave:

```
def ProvisionMobileSIM(json_data, hss_urls):
    """
    Aprovisiona una SIM móvil a HSS e IMS

    Args:
        json_data: Dict con IMSI, MSISDN, credenciales de
autenticación
        hss_urls: Lista de URLs de pares HSS

    Returns:
        Dict con IDs de suscriptor de cada par HSS
    """
    # Crear suscriptor HSS (para datos)
    # Crear suscriptor IMS (para voz)
    # Devolver IDs de suscriptor para seguimiento

def get_subscriber_info_msisdn(msisdn, hss_urls):
    """
    Recupera detalles del suscriptor por número de teléfono

    Args:
        msisdn: Número de teléfono a buscar
        hss_urls: Lista de URLs de pares HSS

    Returns:
        Información del suscriptor desde HSS
    """
```

Soporte Multi-HSS

OmniCRM admite el aprovisionamiento a múltiples pares HSS para redundancia:

```
# crm_config.yaml
hss:
  hss_peers:
    - "http://10.12.64.140:8080"
    - "http://10.12.64.141:8080"
  apn_list: "1,2,3"
  default_apn: 1
```

El aprovisionamiento ocurre en todas las instancias HSS configuradas para asegurar:

- Redundancia geográfica
- Balanceo de carga
- Capacidad de failover
- Datos de suscriptor sincronizados

Centro de Autenticación (AuC)

El AuC almacena las credenciales de autenticación del suscriptor. Durante el aprovisionamiento, solo se recupera el **auc_id**:

Obtener AuC ID para IMSI:

```
GET /auc/imsi/{imsi}
```

Respuesta:

```
{
  "auc_id": 12345,
  "imsi": "310120123456789",
  "iccid": "89612345678901234 56",
  "esim": false,
  "lpa": null
}
```

Para eSIMs, la bandera `esim` es `true` y `lpa` contiene el código de activación para mostrar a los clientes.

Nota: Las credenciales de autenticación reales (Ki, OPC, PIN, PUK) permanecen en el HSS y nunca se exponen a través de la API. El sistema de aprovisionamiento solo necesita el `auc_id` para hacer referencia a estas credenciales al crear suscriptores.

Integración OCS

El Sistema de Cobro en Línea (OCS) maneja el cobro y la calificación en tiempo real para servicios móviles. OmniCRM utiliza CGRateS como la plataforma OCS.

Configuración de OCS

Archivo: `/OmniCRM-API/Provisioners/ocs.py`

Configuración:

```yaml
# crm_config.yaml
ocs:
  cgrates: "10.64.12.160:2080"
  ocsTenant: "mnc001.mcc001.3gppnetwork.org"
  ocsApi: "http://10.64.12.160:2080"
```

Componentes OCS para Servicios SIM

1. Filtros - Identificación del Suscriptor

Los filtros identifican al suscriptor a partir de las solicitudes de red:

```
{
  "ID": "FILTER_IMSI_Service_abc123",
  "Type": "*string",
  "Element": "~*req.OriginHost",
  "Values": ["310120123456789"]
}
```

2. Atributos - Mapeo de Cuenta

Los atributos mapean el IMSI a la cuenta de facturación:

```
{
  "ID": "ATTR_ACCOUNT_Service_abc123",
  "FilterIDs": ["FILTER_IMSI_Service_abc123"],
  "Attributes": [
    {
      "Path": "*req.Account",
      "Value": "Service_abc123"
    },
    {
      "Path": "*req.Bandwidth",
      "Value": "100000000"  // 100 Mbps
    }
  ]
}
```

3. Recursos - Límites de Sesiones

Los recursos controlan los límites de sesiones concurrentes:

```
{
  "ID": "RES_SESSIONS_Service_abc123",
  "FilterIDs": ["FILTER_IMSI_Service_abc123"],
  "Limit": 5,  // Máx 5 sesiones concurrentes
  "UsageTTL": "-1"
}
```

4. Estadísticas - Seguimiento de Uso

Las colas de estadísticas rastrean el uso para análisis:

```
{
  "ID": "STATS_Service_abc123",
  "FilterIDs": ["FILTER_IMSI_Service_abc123"],
  "Metrics": [
    "*sum#~*req.Usage",
    "*tcc"  // Total de conteo de llamadas
  ]
}
```

5. Planes de Acción - Cargos Recurrentes

Los planes de acción manejan operaciones recurrentes como cargos mensuales:

```
{
  "Id": "ActionPlan_Service_abc123_Monthly",
  "ActionsId": "Action_Add_Monthly_Data",
  "Timing": {
    "MonthDays": [1],  // 1 de cada mes
    "Time": "00:00:00Z"
  }
}
```

Funciones de API de OCS

Funciones Clave de ocs.py:

```
async def async_get_balance(account, server, tenant):
    """Obtener saldo actual de la cuenta"""

async def async_topup_dongle(tenant, account, server, days):
    """Agregar asignación de datos basada en tiempo"""

def Get_Account_Status(account, server, tenant):
    """Obtener información completa de la cuenta"""

def CGRateS_API_Call(method, params, server, tenant):
    """Llamada API genérica de CGRateS JSON-RPC"""
```

Reconciliación de Inventario de SIM

OmniCRM incluye herramientas para asegurar la consistencia entre el inventario de CRM y el HSS.

Script de Reconciliación

Archivo: `/OmniCRM-API/Provisioners/hss_reconcile.py`

Propósito:

- Identificar SIMs en HSS pero no en inventario (huérfanas)
- Encontrar SIMs asignadas no aprovisionadas en HSS
- Validar formatos de MSISDN
- Generar informes HTML

Verificaciones Realizadas:

1. **SIMs Vendidas en HSS:**
    - Consultar todas las SIMs asignadas desde el inventario
    - Verificar si cada IMSI existe en HSS
    - Informar SIMs asignadas a clientes pero no en HSS
2. **SIMs Aprovisionadas en Inventario:**
    - Consultar todos los suscriptores desde HSS
    - Verificar si cada IMSI existe en inventario
    - Informar entradas huérfanas de HSS
3. **Validación de MSISDN:**
    - Validar formatos de números de teléfono
    - Verificar cumplimiento de números australianos
    - Informar números mal formados

Ejecutar Reconciliación:

```
python hss_reconcile.py
```

Salida:

- Informe HTML con resultados codificados por colores

- Listas de discrepancias para investigación
- Recomendaciones para limpieza

Mejores Prácticas:

- Ejecutar reconciliación semanalmente
- Investigar discrepancias de inmediato
- Utilizar informes para identificar fallos de aprovisionamiento
- Limpiar entradas huérfanas para liberar recursos

Solución de Problemas de Aprovisionamiento de SIM

Problemas Comunes y Soluciones

Problema: El aprovisionamiento de SIM falla con "IMSI ya existe en HSS"

- **Causa:** IMSI ya está aprovisionada de un intento anterior
- **Solución:** Verificar HSS para suscriptor existente, ya sea:
  - Eliminar el suscriptor existente si está huérfano
  - Actualizar el suscriptor existente en lugar de crear uno nuevo
  - Ejecutar reconciliación para identificar el conflicto

Problema: Código QR de eSIM no se genera

- **Causa:** Campo `management_url` no poblado con código LPA
- **Solución:** Asegurarse de que el script de importación de eSIM establezca `management_url` correctamente
- **Formato:** Debe ser `LPA:1$server$activation-code`

Problema: Elemento de inventario muestra como "Disponible" pero está asignado

- **Causa:** El aprovisionamiento falló antes de la asignación del inventario
- **Solución:** Verificar eventos de aprovisionamiento para la razón del fallo

- **Solución:** Completar el aprovisionamiento o devolver el inventario al grupo

Problema: El cliente no puede hacer llamadas (los datos funcionan)

- **Causa:** Suscriptor IMS no aprovisionado
- **Solución:** Verificar si se completó el paso de aprovisionamiento IMS
- **Solución:** Aprovisionar manualmente el suscriptor IMS o volver a ejecutar el playbook

Problema: Fallos de autenticación (SIM no puede conectarse)

- **Causa:** Desajuste de Ki/OPC entre la tarjeta SIM y el HSS
- **Solución:** Verificar que las credenciales de AuC coincidan con la SIM física
- **Solución:** Actualizar el HSS AuC con las credenciales correctas

Problema: El autoregistro falla al asignar SIM

- **Causa:** No hay SIMs disponibles en el inventario
- **Solución:** Verificar el inventario para SIMs con estado "New" y customer_id NULL
- **Solución:** Importar un nuevo lote de SIM o marcar las SIMs devueltas como "Used"

Consejos de Depuración

1. Verificar Eventos de Aprovisionamiento:

```
GET /crm/provision/provision_id/{id}
```

Buscar tareas fallidas (estado 2) y revisar mensajes de error.

2. Verificar Suscriptor HSS:

```
GET {hss_url}/subscriber/msisdn/{phone_number}
```

Verificar si el suscriptor existe y tiene la configuración correcta.

3. Verificar Cuenta OCS:

```
{
  "method": "ApierV2.GetAccount",
  "params": [
    {
      "Tenant": "mnc001.mcc001.3gppnetwork.org",
      "Account": "Service_abc123"
    }
  ]
}
```

4. Consultar Estado del Inventario:

```
GET /crm/inventory/?filters={"itemtext2":["310120123456789"]}
```

Buscar por IMSI para encontrar el elemento de inventario y verificar su estado.

5. Revisar Registros de Ansible:

Verificar `/var/log/ansible/` o detalles de eventos de aprovisionamiento para la salida completa del playbook.

Mejores Prácticas

Gestión de Inventario de SIM

1. **Importación Masiva:** Utilizar scripts de importación para grandes lotes de SIM
2. **Reconciliación Regular:** Ejecutar reconciliación semanal de HSS/inventario
3. **Gestión de Estados:** Mantener los estados de inventario precisos (New, Assigned, Used)
4. **Seguimiento de Proveedores:** Registrar información de proveedor y lote de SIM
5. **Seguimiento de Costos:** Mantener costos mayoristas/minoristas precisos

Seguridad

1. **Separación de Credenciales:** Las credenciales de autenticación (Ki/OPC) permanecen solo en HSS, nunca en CRM
2. **Control de Acceso:** Limitar quién puede asignar/modificar el inventario de SIM
3. **Registro de Auditoría:** Rastrear todas las asignaciones y cambios de SIM
4. **Gestión de PIN/PUK:** Códigos PIN/PUK almacenados en HSS, proporcionados a los clientes a través de canales seguros
5. **Proceso de Perdido/Robo:** Suspender inmediatamente el servicio en HSS y marcar el estado del inventario

Aprovisionamiento

1. **Validación:** Siempre validar el formato y la unicidad del IMSI
2. **Reversión:** Utilizar block/rescue para limpieza automática en caso de fallo
3. **Notificaciones:** Enviar instrucciones de activación claras a los clientes
4. **Pruebas:** Probar el aprovisionamiento en desarrollo antes de producción
5. **Monitoreo:** Rastrear tasas de éxito de aprovisionamiento y razones de fallo

Específico de eSIM

1. **Formato LPA:** Validar el formato del código LPA antes de almacenarlo
2. **Pruebas de QR:** Probar códigos QR en múltiples dispositivos
3. **Instrucciones de Activación:** Proporcionar guías paso a paso claras
4. **Gestión de Perfiles:** Rastrear descargas e instalaciones de perfiles de eSIM
5. **Códigos de Respaldo:** Proporcionar códigos LPA manuales en caso de que falle el escaneo del QR

Documentación Relacionada

- Sistema de Aprovisionamiento - Detalles completos del flujo de trabajo de aprovisionamiento

- Playbooks de Ansible - Estructura de playbook y mejores prácticas
- Gestión de Inventario - Resumen del sistema de inventario
- Productos y Servicios - Configuración de productos
- Atención al Cliente - Apoyando a los clientes con problemas de SIM
- Recarga y Recarga - Agregar saldo a servicios móviles

Registro de Actividad del Cliente

El Registro de Actividad registra cada cambio realizado en un cliente, contacto, sitio, servicio y parámetros financieros como Transacciones, Facturas y Métodos de Pago.

Esto nos permite rastrear los cambios realizados en el sistema, quién los realizó y cuándo los realizó. El registro es útil para auditar cambios y para rastrear problemas. Por ejemplo, si un cliente dice que nunca recibió una factura, podemos verificar el Registro de Actividad para ver si el sistema la envió. Si un contacto fue eliminado, podemos ver quién lo eliminó y cuándo.

El Registro de Actividad es una lista cronológica de cambios. Los cambios más recientes aparecen en la parte superior, y los cambios más antiguos aparecen más abajo en la lista. Puedes filtrar la lista desde las pestañas.

No puedes eliminar registros del registro de actividad. Puedes filtrarlos y puedes ver los detalles para ver qué se cambió y por quién.

Agregar un Servicio

Los servicios son las cosas por las que facturamos al cliente. Pueden ser servicios de internet en casa, servicios móviles o incluso servicios abstractos como arrendar una subred o proporcionar electricidad medida a un rack.

Un Servicio es solo una instancia de un Producto para un cliente dado. El sistema lo selecciona del catálogo de productos y lo provisiona para el cliente.

Si aún no has creado un cliente, necesitas hacerlo primero, porque los servicios están vinculados a los clientes. También necesitas definir un método de pago para el cliente, porque los servicios generan cargos que deben ser pagados.

Los clientes pueden provisionar sus propios servicios (si les permitimos hacerlo). El personal de Atención al Cliente también puede provisionar servicios para el cliente.

La definición del producto establece reglas sobre quién puede comprar un producto. Por ejemplo, una regla puede permitir que solo los clientes comerciales comprar un producto comercial. Una regla también puede permitir que solo los clientes con un servicio móvil compren un complemento móvil.

Los servicios también pueden tener componentes de uso, como uso de datos, minutos de llamada, u otros cargos basados en el uso. Un servicio puede tener múltiples cargos, como cargos mensuales, cargos únicos o cargos por uso. Vemos estos desde el botón "Uso".

Muchos servicios admiten Complementos. Por ejemplo, un servicio móvil tiene todas las recargas disponibles en el menú de Complementos. Un servicio de internet en casa podría tener una dirección IP estática o datos adicionales disponibles como un complemento. El catálogo de productos define todo esto.

Flujo de Trabajo para Asignar Planes

La función Asignar Plan permite al personal **proveer servicios para clientes** seleccionando productos del catálogo e iniciando el proceso de aprovisionamiento. Este es el método principal para crear nuevos servicios cuando los clientes no se aprovisionan por sí mismos.

Descripción General

Asignar un plan implica:

1. Seleccionar un cliente
2. Elegir un producto del catálogo
3. Configurar los requisitos de inventario (tarjetas SIM, equipos, etc.)
4. Establecer parámetros del servicio (renovación automática, campos personalizados)
5. Iniciar el aprovisionamiento
6. Monitorear el progreso del aprovisionamiento

Este flujo de trabajo se utiliza para todos los tipos de servicio: móvil, internet, IPTV y VoIP.

Accediendo a Asignar Plan

Desde el Catálogo de Productos:

Desde la Página del Cliente:

Desde la Página de Complementos:

La interfaz de asignar plan se abre en un modal o página dedicada.

Flujo de Trabajo Paso a Paso

Paso 1: Navegar por el Catálogo de Productos

El catálogo de productos muestra los productos disponibles agrupados por categoría y tipo de cliente.

Alternar Vistas:

> Categorías: • Planes de Servicio • Complementos

Alternar entre filtros Individual y Empresarial filtra los productos para mostrar solo aquellos disponibles para el tipo de cliente seleccionado.

Tarjetas de Producto:

Los productos se muestran en un carrusel o cuadrícula:

Haz clic en **"Asignar a Cliente"** para continuar.

Paso 2: Seleccionar Cliente

Si no estás ya en una página de cliente, el sistema te pedirá que selecciones un cliente.

Búsqueda de Clientes:

> Buscar Clientes: [John ▼]
>
> Clientes coincidentes: • John Smith (ID: 123) • John Doe (ID: 456) • Johnson Enterprises (ID: 789)

Escribe para buscar por:

- Nombre del cliente
- ID del cliente
- Dirección de correo electrónico
- Número de teléfono

Selecciona el cliente del menú desplegable.

> ⚠ **AVISO**
> Asegúrate de haber seleccionado el cliente correcto antes de continuar. Si asignas un plan al cliente equivocado, requerirá intervención manual para corregirlo.

Paso 3: Configurar Inventario (si es necesario)

Si el producto requiere artículos de inventario (definidos en `inventory_items_list`), aparecerán seleccionadores de inventario.

Ejemplo de Selector de Inventario:

> Artículos de Inventario Requeridos:
>
> Tarjeta SIM *
>
> Tarjetas SIM Disponibles: • SIM-00123 - ICCID: 8944...0001 (Nueva) • SIM-00124 - ICCID: 8944...0002 (Nueva) • SIM-00125 - ICCID: 8944...0003 (Nueva)
>
> Número Móvil *
>
> Números Disponibles: • +44 7700 900123 (Reservado) • +44 7700 900124 (Disponible) • +44 7700 900125 (Disponible)

Reglas de Selección de Inventario:

- El asterisco (*) indica un campo requerido

- Solo se muestran los artículos de inventario disponibles (estado: "En Stock" o "Nuevo")
- Los menús desplegables se cargan dinámicamente según los nombres de las plantillas de inventario
- Una vez seleccionados, los artículos se reservan temporalmente

Qué Ocurre:

- Los artículos de inventario seleccionados se pasan como variables al libro de jugadas de aprovisionamiento
- Durante el aprovisionamiento, los artículos se asignan al servicio y al cliente
- El estado del artículo cambia de "En Stock" a "Asignado"

Paso 4: Configurar Renovación Automática (opcional)

Para servicios recurrentes, se te puede solicitar que configures la renovación automática:

> ¿Te gustaría habilitar la renovación automática para este servicio?
>
> Cuando esté habilitado, este servicio se renovará automáticamente al final de cada período de facturación y se cargará al método de pago predeterminado del cliente.
>
> [No] [Sí]

Comportamiento de Renovación Automática:

- **Sí**: El servicio se renueva automáticamente, se cobra al cliente mensualmente
- **No**: El servicio expira al final del período del contrato, se requiere renovación manual

Mejor Práctica: Predeterminar "Sí" para servicios de consumo, "No" para servicios únicos o cuando el cliente solicita control manual.

Paso 5: Revisar y Confirmar

La pantalla de revisión muestra todas las selecciones antes del aprovisionamiento:

> Cliente: John Smith (ID: 123) Producto: Móvil - Plan de 20GB
>
> Inventario: • Tarjeta SIM: SIM-00123 (ICCID: 8944...0001) • Número Móvil: +44 7700 900123
>
> Precios: • Cuota de Instalación: £0.00 • Costo Mensual: £15.00
>
> Renovación Automática: Sí
>
> [Cancelar] [Confirmar y Aprovisionar]

Haz clic en **"Confirmar y Aprovisionar"** para iniciar el proceso de aprovisionamiento.

Paso 6: Progreso del Aprovisionamiento

El modal de aprovisionamiento muestra el progreso en tiempo real:

> ✓ Validando cuenta del cliente ✓ Asignando Tarjeta SIM (ICCID: 8944...0001) ✓ Asignando Número Móvil (+44 7700 900123) ⟳ Configurando cuenta OCS (en progreso...) □ Creando registro de servicio □ Enviando correo electrónico de bienvenida
>
> Progreso: 3 de 6 tareas completadas

El modal consulta la API de aprovisionamiento cada 0.2 segundos para actualizaciones de estado.

Indicadores de Progreso:

☑ Completado con éxito
⟳ Actualmente en ejecución
□ Pendiente (no iniciado)
✗ Fallido (si ocurren errores)

Paso 7: Finalización

Éxito:

> Servicio aprovisionado con éxito para John Smith
>
> ID del Servicio: 789 Nombre del Servicio: Móvil - +44 7700 900123 Estado: Activo
>
> [Ver Servicio] [Cerrar]

Haz clic en **"Ver Servicio"** para abrir la página de detalles del servicio.

Fallo:

Si el aprovisionamiento falla:

> Error: No se pudo conectar a OCS
>
> El registro del servicio ha sido creado pero el aprovisionamiento no se completó. Por favor, revisa el error y vuelve a intentarlo.
>
> ID de Aprovisionamiento: 456
>
> [Ver Registros] [Reintentar] [Cerrar]

- **Ver Registros**: Abre detalles de aprovisionamiento con mensajes de error
- **Reintentar**: Intenta aprovisionar nuevamente
- **Cerrar**: Sale del modal (el registro del servicio permanece pero no está activado)

Casos Especiales

Agregar Complementos a un Servicio Existente

Al asignar un complemento (categoría: "addon") a un cliente que ya tiene un servicio:

1. **Detección Automática de Servicio:**

    - El sistema encuentra los servicios existentes del cliente
    - Filtra por service_type (el complemento móvil solo se muestra para servicios móviles)
    - Si el cliente tiene múltiples servicios coincidentes, solicita seleccionar cuál
2. **No Se Crea un Nuevo Servicio:**
    - El complemento se aprovisiona contra el service_id existente
    - Utiliza la cuenta OCS del servicio existente
    - El inventario (si lo hay) se asigna al servicio existente
3. **Libro de Jugadas de Aprovisionamiento:**
    - Libro de jugadas diferente al de servicios independientes
    - Típicamente añade saldo, características o equipos a la cuenta existente

Ejemplo:

> Servicios Móviles Existentes: • Móvil - +44 7700 900123 (ID: 789) • Móvil - +44 7700 900456 (ID: 790)
>
> ¿Qué servicio debería recibir este complemento? [Móvil - +44 7700 900123 ▼]
>
> [Cancelar] [Continuar]

Aprovisionamiento para Clientes Empresariales

Los clientes empresariales pueden tener requisitos adicionales:

- **Selección de Sitio** - Elegir qué ubicación empresarial recibe el servicio
- **Asignación de Contacto** - Designar contactos de facturación/técnicos
- **Campos Personalizados** - Números de cuenta, centros de costo, números de orden de compra

Ejemplo de Flujo Empresarial:

> Seleccionar Sitio de Instalación: [Oficina de Londres - 123 Main St ▼]
>
> Contacto de Facturación: [Jane Doe - <jane@acme.com> ▼]

> Contacto Técnico: [Bob Smith - <bob@acme.com> ▼]
>
> Número de Orden de Compra: [PO-2025-001234_____]
>
> [Cancelar] [Continuar]

Asignación de Servicios en Masa

Para asignar el mismo plan a múltiples clientes (por ejemplo, migraciones masivas):

1. Usar importación CSV (si está disponible)
2. O asignar individualmente con configuraciones de plantilla
3. El inventario debe estar disponible en masa
4. Monitorear la cola de aprovisionamiento para evitar sobrecarga

Flujos de Trabajo Comunes

Flujo de Trabajo 1: Nuevo Servicio Móvil

1. El cliente entra en la tienda queriendo un servicio móvil
2. El personal abre **Productos → Planes**
3. Cambia a tipo de cliente **Individual**
4. Selecciona **"Móvil - Plan de 20GB"**
5. Haz clic en **"Asignar a Cliente"**
6. Busca al cliente por teléfono: "+1234567890"
7. Selecciona a **John Smith** de los resultados
8. Elige la Tarjeta SIM del selector de inventario
9. Elige un número móvil disponible
10. Habilita **renovación automática**
11. Confirma y aprovisiona
12. Observa el progreso hasta completar
13. Entrega la SIM al cliente con el paquete de bienvenida

Flujo de Trabajo 2: Agregar Internet a un Cliente Existente

1. Navega a la página del cliente: **Clientes → John Smith**
2. Haz clic en la pestaña **Servicios**
3. Haz clic en el botón **"Agregar Servicio"**
4. Navega por los planes de internet
5. Selecciona **"Fibra - 100Mbps"**
6. Selecciona el sitio de instalación (si es empresarial/múltiples sitios)
7. Elige el módem CPE del inventario
8. Establece la fecha de instalación (si es necesario)
9. Aprovisiona el servicio
10. Crea un ticket de instalación

Flujo de Trabajo 3: Asignar Complemento de Recarga de Datos

1. El cliente llama: "Necesito más datos"
2. El personal busca al cliente en la búsqueda global
3. Abre la pestaña de servicios del cliente
4. Haz clic en "Complementos" junto al servicio móvil
5. Selecciona "Aumento de Datos de 5GB"
6. Pago autorizado
7. El complemento se aprovisiona inmediatamente
8. El cliente recibe un aumento de datos instantáneo

Solución de Problemas

"No hay productos disponibles"

- **Causa:** Los filtros excluyen todos los productos
- **Solución:**
  - Cambiar el tipo de cliente (Individual vs Empresarial)
  - Verificar que el catálogo de productos tenga productos habilitados

    - Confirmar que los productos coincidan con la elegibilidad del cliente

"No hay inventario disponible"

- **Causa:** Artículos de inventario agotados o todos asignados
- **Solución:**
    - Agregar más artículos de inventario al sistema
    - Verificar el estado del artículo (debería ser "Nuevo" o "En Stock")
    - Confirmar que los nombres de las plantillas de inventario coincidan con los requisitos del producto

"Cliente no encontrado"

- **Causa:** El cliente no existe o el término de búsqueda es incorrecto
- **Solución:**
    - Crear el cliente primero
    - Probar diferentes términos de búsqueda (ID, correo electrónico, teléfono)
    - Verificar errores tipográficos

El aprovisionamiento se queda atascado

- **Causa:** Error en el libro de jugadas o sistema externo inalcanzable
- **Solución:**
    - Esperar el tiempo de espera (típicamente 5 minutos)
    - Revisar los registros de aprovisionamiento para errores específicos
    - Verificar que OCS y los sistemas de red estén en línea
    - Reintentar el aprovisionamiento después de solucionar el problema

El aprovisionamiento tiene éxito pero el servicio no funciona

- **Causa:** Cuenta OCS creada pero la red no se actualizó
- **Solución:**
    - Verificar que OCS tenga la cuenta
    - Confirmar que la SIM esté activada en HLR/HSS
    - Verificar el aprovisionamiento de la red (RADIUS, DPI, etc.)
    - Revisar las tareas del libro de jugadas para pasos omitidos

Mejores Prácticas

Antes de Asignar:

- Verificar que el cliente tenga un método de pago válido en el archivo
- Confirmar la elegibilidad del cliente para el producto (residencial vs empresarial)
- Asegurarse de que el inventario requerido esté disponible
- Revisar los términos y precios del producto con el cliente

Durante la Asignación:

- Verificar la selección del cliente antes de confirmar
- Seleccionar los artículos de inventario correctos (verificar números de serie)
- Habilitar la renovación automática por conveniencia (a menos que el cliente objete)
- Monitorear el progreso del aprovisionamiento hasta la finalización

Después de la Asignación:

- Verificar que el servicio aparezca en la lista de servicios del cliente
- Confirmar que el estado del servicio sea "Activo"
- Asegurarse de que el inventario esté asignado correctamente
- Enviar un correo electrónico de bienvenida o instrucciones al cliente
- Probar el servicio si es posible (hacer una llamada de prueba, verificar datos)

Para Complementos:

- Confirmar que el complemento sea compatible con el servicio existente
- Explicar la facturación (única vs recurrente)
- Verificar el pago antes de aprovisionar
- Comprobar que el saldo se actualice inmediatamente después del aprovisionamiento del complemento

Documentación Relacionada

- `csa_add_service` - Descripción general de servicios
- `concepts_products_and_services` - Conceptos del catálogo de productos
- `administration_inventory` - Gestión de artículos de inventario
- `concepts_provisioning` - Detalles del sistema de aprovisionamiento
- `guide_product_lifecycle` - Ciclo de vida completo del producto, incluyendo aprovisionamiento

Registros de Detalle de Llamadas (CDRs)

Los Registros de Detalle de Llamadas muestran las **llamadas, mensajes SMS y sesiones de datos** individuales para un servicio. Donde Uso del Servicio muestra cuánto saldo queda, los Registros de Llamadas muestran los *eventos* que lo utilizaron: quién fue llamado, cuándo, por cuánto tiempo y cuánto costó.

Abrir Registros de Llamadas

Los Registros de Llamadas se encuentran en la ventana de **Uso del Servicio**, junto a la vista de Saldo.

Desde un servicio:

1. Haz clic en el botón **Saldo** (o **Uso**) junto al servicio
2. Selecciona la pestaña **Registros de Llamadas (CDRs)**

También puedes abrirlo desde el botón **CDRs** en la pantalla de Editar Servicio. Para los clientes, ábrelo desde **Ver Registros de Llamadas** en la página de uso del Portal de Autocuidado.

Leer la Lista

Los registros se enumeran con los **más recientes primero**. Cada fila muestra:

Columna	Lo que muestra
Hora de Configuración	Cuando comenzó la llamada, mensaje o sesión
Destino	El número llamado, el destinatario del SMS o el destino de datos
Categoría	Si fue un evento de **Llamada**, **SMS** o **Datos**
Uso	Duración de la llamada (`HH:MM:SS`), cantidad de datos (MB/GB) o conteo de SMS
Costo	Lo que se cobró por el evento

Usa **Anterior** y **Siguiente** en la parte inferior para navegar a través de registros más antiguos.

Haz clic en cualquier fila para abrir los detalles completos de ese registro, incluyendo el tiempo de respuesta y la información adicional registrada en su contra.

Encontrar Registros Específicos

Usa la barra de filtro en la parte superior para reducir la lista, luego haz clic en **Aplicar**:

- **Botones rápidos (1h / 24h / 7d / 30d)**: muestran registros de la última hora, día, semana o mes.
- **Desde / Hasta**: elige tu propia fecha y hora de inicio y fin.
- **Destino comienza con**: escribe los primeros dígitos de un número para mostrar solo las llamadas coincidentes (por ejemplo, `44` para números del Reino Unido).
- **Categoría**: muestra solo **Llamadas**, **SMS** o **Datos** (o **Todos**).

Restablecer borra los filtros y muestra nuevamente los registros más recientes.

> **¿No ves nada?** La vista se abre en los registros más recientes, pero si seleccionaste un rango de fechas, el servicio puede simplemente no haber tenido actividad en ese período. Intenta **Restablecer**, o amplía las fechas de **Desde** / **Hasta**.

Descargar Registros

Haz clic en **Exportar CSV** para descargar los registros que se muestran actualmente (después de tus filtros) como un archivo de hoja de cálculo. Esto es útil para compartir con un cliente o mantener un registro de una consulta de facturación.

Quién Puede Ver los Registros de Llamadas

- **El personal** necesita el permiso de registros de llamadas. Si no ves la pestaña **Registros de Llamadas**, pide a tu administrador que lo habilite para tu cuenta.
- **Los clientes** pueden ver los registros de **sus propios servicios únicamente** a través del Portal de Autocuidado.

Los Registros de Llamadas son **solo de visualización**. No se pueden editar ni eliminar desde aquí.

Modificación de un Servicio

Los servicios pueden ser modificados por el cliente final a través del Portal de Autocuidado, o por un administrador a través del portal de administración.

Una vez que un servicio está provisionado, puedes modificar sus parámetros, agregar mejoras o cambiar configuraciones.

Editando Parámetros del Servicio

Los parámetros básicos del servicio pueden ser modificados haciendo clic en el botón **Editar** en la página de detalles del servicio.

Campos Editables:

- Nombre del Servicio
- Estado del Servicio (Activo, Inactivo, Suspendido)
- Notas del Servicio
- Configuraciones de Renovación Automática
- Artículos de inventario vinculados
- Campos personalizados

Los cambios se guardan de inmediato y se reflejan en el portal de autocuidado del cliente.

Agregando Complementos al Servicio

Los complementos mejoran los servicios existentes con características adicionales, asignaciones de datos o hardware. Los tipos de complementos comunes incluyen:

- **Recargas de datos** - Asignación de datos adicional (por ejemplo, "Aumento de 5GB de Datos")
- **Actualizaciones de características** - Capacidades adicionales (por ejemplo, "Llamadas Internacionales")
- **Alquiler de hardware** - Adiciones de equipo (por ejemplo, "Módem WiFi 6")
- **Servicios premium** - Características mejoradas (por ejemplo, "Dirección IP Estática")

Accediendo al Catálogo de Complementos

Desde una página de detalles del servicio:

1. Navega al servicio que deseas mejorar
2. Haz clic en el botón **"Agregar Complemento"** o **"Explorar Complementos"**
3. Se abre el catálogo de complementos, filtrado para mostrar solo complementos compatibles

Filtrado Automático:

El sistema filtra automáticamente los complementos según:

- **Tipo de Servicio** - Solo muestra complementos que coinciden con el tipo de servicio (móvil, internet, voip, etc.)
- **Tipo de Cliente** - Filtra por cliente residencial vs. cliente empresarial
- **Compatibilidad del Servicio** - Verifica si el servicio cumple con los requisitos del complemento
- **Disponibilidad** - Solo muestra productos habilitados

Por ejemplo, si ves un servicio móvil residencial, solo verás complementos móviles marcados para clientes residenciales.

Visualización de Complementos

Los complementos se muestran en un carrusel interactivo que muestra:

Visualización de Tarjeta de Producto:

Información Mostrada:

- Icono del producto

- Nombre del producto
- Lista de características (puntos de viñeta)
- Costo de instalación
- Costo mensual/recurrido
- Enlace a términos y condiciones
- Botón "Agregar al Servicio"

Provisionando un Complemento

Paso 1: Seleccionar Complemento

Haz clic en la tarjeta del complemento deseado, luego haz clic en **"Agregar al Servicio"**

Paso 2: Seleccionar Inventario (si es necesario)

Si el complemento requiere inventario físico (por ejemplo, alquiler de hardware), aparece un selector de inventario:

> Módems Disponibles: • Modem-12345 - TP-Link AX1800 (Nuevo) • Modem-12346 - TP-Link AX1800 (Nuevo) • Modem-12347 - Netgear RAX40 (Nuevo)

Selecciona el artículo de inventario específico para asignar a este servicio.

Paso 3: Configurar Renovación Automática (opcional)

Para complementos recurrentes, se te puede preguntar:

> ¿Te gustaría habilitar la renovación automática para este complemento?
>
> [No] [Sí]

- **Sí** - El complemento se renueva automáticamente cada período de facturación
- **No** - Compra única, el cliente debe renovar manualmente

Paso 4: Confirmar y Provisionar

Revisa los detalles del complemento y haz clic en **"Confirmar"**

El modal de provisión aparece mostrando el progreso en tiempo real:

> ✓ Validando pago ✓ Asignando inventario ⟳ Configurando servicio (en progreso...) □ Creando transacción □ Enviando correo electrónico de confirmación

Una vez completado, el complemento aparece en la lista de complementos del servicio.

Provisionamiento Detrás de Escena:

Cuando agregas un complemento, el sistema:

1. Valida que el cliente pueda comprar el complemento
2. Ejecuta el playbook de Ansible del complemento (`provisioning_play`)
3. Asigna cualquier artículo de inventario requerido al servicio
4. Crea un registro de transacción para la facturación
5. Actualiza la configuración del servicio (OCS, sistemas de red, etc.)
6. Envía una notificación de confirmación al cliente

Para detalles técnicos sobre el aprovisionamiento de complementos, consulta Guía Completa del Ciclo de Vida del Producto - Agregando Complementos.

Visualizando Complementos Activos

Los complementos activos aparecen en la página de detalles del servicio en la sección **"Complementos"**:

Ejemplo de Visualización:

> □ Aumento de 5GB de Datos
> Agregado: 10 de enero de 2025 Expira: 17 de enero de 2025 Estado: Activo Costo: £5.00
>
> □ Alquiler de Módem WiFi 6
> Agregado: 01 de diciembre de 2024 Inventario: Modem-12345 Renovación Automática: Sí Mensual: £10.00

Eliminando Complementos

Para eliminar un complemento de un servicio:

1. Navega a la página de detalles del servicio
2. Localiza el complemento en la sección **"Complementos Activos"**
3. Haz clic en el **"Eliminar"** o en el **icono de papelera** junto al complemento
4. Confirma la eliminación

Qué Ocurre:

- **Complementos de hardware** - Se ejecuta el playbook de deprovisionamiento, el inventario se marca para devolución
- **Complementos virtuales** - Los beneficios se eliminan de inmediato
- **Complementos de renovación automática** - Cancela renovaciones futuras
- **Transacción creada** - Se acredita cualquier monto prorrateado si es aplicable

> ⚠ **AVISO**
>
> Eliminar complementos de hardware (módems, CPE, etc.) generalmente requiere que el equipo sea devuelto. El sistema marcará el inventario para devolución y puede enviar instrucciones de devolución al cliente.

Escenarios Comunes de Complementos

Escenario 1: Cliente Quedándose Sin Datos

1. El cliente contacta al soporte: "He utilizado mi asignación mensual"
2. El personal navega al servicio móvil del cliente
3. Hace clic en "Agregar Complemento"
4. Selecciona "Aumento de 5GB de Datos"
5. Provisión inmediata

6. El cliente recibe una recarga de datos instantánea

Escenario 2: Negocio Necesita IP Estática

1. El cliente empresarial solicita una IP estática para VPN
2. El personal abre el servicio de internet del cliente
3. Navega por los complementos, selecciona "Dirección IP Estática"
4. El sistema provisiona la IP del grupo disponible (inventario)
5. Configura el enrutamiento en el equipo de red
6. El negocio recibe los detalles de configuración de IP

Escenario 3: Alquiler de Equipos para Servicio de Fibra

1. El cliente se inscribe para internet de fibra
2. Durante la provisión, el cliente opta por el alquiler de un módem
3. El personal agrega el complemento "Módem WiFi 6"
4. Selecciona el módem disponible del inventario
5. El módem se envía al cliente
6. Se agrega un cargo recurrente de £10/mes a la cuenta

Solución de Problemas

"No hay complementos compatibles disponibles"

- **Causa:** No hay complementos que coincidan con el tipo de servicio o tipo de cliente
- **Solución:** Verifica que el catálogo de productos tenga complementos con configuraciones de `service_type` y `residential`/`business` coincidentes

Fallo en la provisión del complemento

- **Causa:** Error en el playbook de provisión o inventario no disponible
- **Solución:**
  - Revisa los registros de provisión para un error específico
  - Verifica que los artículos de inventario estén en stock (si es necesario)
  - Revisa los registros del playbook en el sistema de provisión

El selector de inventario no muestra artículos

- **Causa:** No hay artículos de inventario disponibles del tipo requerido
- **Solución:**
  - Agrega artículos de inventario al sistema
  - Verifica que los artículos existentes no estén todos asignados o dañados
  - Verifica que el nombre de la plantilla de inventario coincida exactamente con `inventory_items_list`

La renovación automática no funciona

- **Causa:** La bandera de renovación automática no está configurada o el método de pago ha expirado
- **Solución:**
  - Verifica que el servicio tenga `auto_renew: true`
  - Verifica que el cliente tenga un método de pago válido en el archivo
  - Revisa los trabajos programados en el sistema de facturación

Documentación Relacionada

- `guide_product_lifecycle` - Flujo completo de provisión de complementos
- `csa_add_service` - Creación de nuevos servicios
- `concepts_products_and_services` - Conceptos de productos y servicios
- `administration_inventory` - Gestión de inventario para complementos de hardware

Gestión de Servicios

La interfaz de Gestión de Servicios proporciona herramientas completas para visualizar, filtrar y gestionar los servicios de los clientes en toda su organización.

Documentación relacionada: Añadiendo Servicios, Uso del Servicio, Modificando Servicios, Productos y Servicios.

Accediendo a la Lista de Servicios

Navegue a:

O directamente:

Esto muestra todos los servicios con capacidades de filtrado, ordenamiento y gestión masiva.

Visión General de la Lista de Servicios

La lista de servicios muestra los servicios en una tabla con:

Columnas:

- **ID del Servicio** - Identificador único
- **Nombre del Servicio** - Nombre descriptivo (por ejemplo, "Móvil - +44 7700 900123")
- **Nombre del Cliente** - Cliente asociado (enlace clicable)
- **Tipo de Servicio** - móvil, iptv, internet, voip
- **Estado** - Activo, Inactivo, Suspendido
- **Fecha de Inicio** - Cuando comenzó el servicio
- **Fecha de Finalización** - Cuando expira/termina el servicio
- **Costo Mensual** - Precio de venta
- **Acciones** - Ver, Editar, Complementos, Uso, Eliminar

Filtrando Servicios

Pestañas de Estado

Filtre rápidamente los servicios por estado utilizando la barra de pestañas:

- **Todos los Resultados** - Muestra todos los servicios independientemente del estado
- **Activos** - Servicios actualmente activos (vista predeterminada)
- **Inactivos** - Servicios cancelados o expirados
- **Suspendidos** - Servicios deshabilitados temporalmente (por falta de pago, fraude, etc.)

Hacer clic en una pestaña actualiza la lista instantáneamente. La pestaña activa está resaltada.

Filtro por Tipo de Servicio

Filtre por tipo de servicio:

Seleccione uno o varios tipos para mostrar solo los servicios que coincidan.

Casos de Uso:

- **Móvil** - Ver todos los servicios de SIM móvil
- **Internet** - Mostrar fibra, DSL y wireless fijo
- **IPTV** - Mostrar servicios de suscripción de TV
- **VoIP** - Listar servicios de voz sobre IP

Filtro por Cliente

Busque y filtre por cliente:

Comience a escribir el nombre de un cliente para ver resultados coincidentes:

Seleccione un cliente para mostrar solo sus servicios.

Caso de Uso: Ver rápidamente todos los servicios de un cliente específico.

Búsqueda

Búsqueda global en todos los campos de servicio:

Busca:

- Nombre del servicio
- ID del servicio
- Nombre del cliente
- Números de teléfono (para servicios móviles)
- UUID del servicio

Ejemplo: Busque "0770" para encontrar todos los servicios con números de teléfono que contengan "0770".

Ordenando Servicios

Haga clic en cualquier encabezado de columna para ordenar por ese campo:

Columnas Ordenables:

- ID del Servicio (predeterminado: más reciente primero)

- Nombre del Servicio (alfabético)
- Nombre del Cliente (alfabético)
- Tipo de Servicio (alfabético)
- Fecha de Inicio (cronológico)
- Fecha de Finalización (cronológico)
- Costo Mensual (numérico)

Dirección de Ordenación:

- Haga clic una vez: Ascendente (A-Z, más antiguo-más nuevo, más bajo-más alto)
- Haga clic dos veces: Descendente (Z-A, más nuevo-más antiguo, más alto-más bajo)
- La columna de ordenación activa muestra el indicador ▲ o ▼

Desplegable de Ordenación:

El desplegable en la parte superior derecha proporciona preajustes de ordenación rápida:

Visualizando Detalles del Servicio

Haga clic en cualquier nombre de servicio para abrir la vista detallada del servicio.

Pesta��as de Detalle del Servicio:

1. **Resumen** - Resumen del servicio, estado, fechas, precios
2. **Inventario** - Equipos asignados (tarjetas SIM, módems, etc.)
3. **Transacciones** - Cargos, créditos, pagos
4. **Uso** - Estadísticas de uso de datos/voz/SMS
5. **Complementos** - Complementos activos y disponibles
6. **Registro de Actividad** - Historial de cambios

Acciones Rápidas

Desde la lista de servicios, haga clic en el menú de acciones (⋮) para operaciones rápidas:

- **Ver** - Abre los detalles del servicio
- **Editar** - Modificar parámetros del servicio
- **Complementos** - Navegar y añadir mejoras al servicio
- **Uso** - Ver uso y saldo actuales
- **Eliminar** - Cancelar/eliminar servicio

Operaciones Masivas

Seleccione múltiples servicios utilizando casillas de verificación para realizar acciones masivas.

Seleccionando Servicios

Selección Individual:

Haga clic en la casilla de verificación junto a cada servicio que desea seleccionar.

Seleccionar Todos:

Haga clic en la casilla de verificación en el encabezado de la tabla para seleccionar todos los servicios visibles.

> ☑ Servicio 1 ☑ Servicio 2 ☑ Servicio 3

Contador de Selección:

La interfaz muestra cuántos servicios están seleccionados:

Acciones Masivas

Una vez seleccionados los servicios, aparecen botones de acción masiva:

Eliminar Múltiples Servicios:

1. Seleccione los servicios a eliminar
2. Haga clic en el botón **"Eliminar Seleccionados"**
3. Confirme la eliminación en el modal

> ⚠ **AVISO**
>
> La eliminación masiva es permanente y:
>
> - Cancelará todos los servicios seleccionados
> - Marcará el inventario como no asignado
> - Detendrá todos los cargos recurrentes
> - Creará entradas en el registro de actividad

Casos de Uso:

- Limpiar servicios de prueba
- Cancelar servicios para cuentas de clientes cerradas
- Eliminar entradas duplicadas o erróneas

Mejor Práctica: Utilice filtros para reducir la lista antes de realizar operaciones masivas para evitar eliminaciones accidentales.

Gestión del Estado del Servicio

Tipos de Estado

Los servicios pueden tener tres estados:

Activo

- El servicio está operativo
- Se aplican cargos
- El cliente puede usar el servicio
- Se muestra con una insignia verde

Inactivo

- El servicio ha sido cancelado o ha expirado
- No se aplican cargos
- El cliente no puede usar el servicio
- Se muestra con una insignia gris
- Inventario marcado para devolución/refurbishment

Suspendido

- Servicio deshabilitado temporalmente
- Puede o no cobrar (configurable)
- El cliente no puede usar el servicio
- Se muestra con una insignia naranja/amarilla
- Razones comunes: Falta de pago, investigación de fraude, solicitud del cliente

Cambiando el Estado del Servicio

Para cambiar el estado de un servicio:

1. Abra los detalles del servicio
2. Haga clic en el botón **"Editar"**
3. Cambie el desplegable **"Estado del Servicio"**
4. Haga clic en **"Guardar"**

Qué Sucede:

- **Activo → Inactivo:** Activa el libro de jugadas de desprovisionamiento (si está configurado)
- **Activo → Suspendido:** Desactiva el servicio en OCS/red pero mantiene el registro
- **Suspendido → Activo:** Reactiva el servicio, reanuda la facturación
- **Inactivo → Activo:** Puede activar el reprovisionamiento (usar con precaución)

Vista de Uso del Servicio

Haga clic en **"Uso"** en el menú de acciones para abrir el modal de uso.

Información Mostrada:

- **Saldo** - Crédito restante o valor prepago
- **Uso de Datos** - Usado vs. asignado (barra de progreso)
- **Uso de Voz** - Minutos usados vs. asignación del plan
- **Uso de SMS** - Mensajes enviados vs. asignación
- **Fecha de Expiración** - Cuando expira el saldo/plan actual

Acciones:

- **Recargar** - Añadir crédito o datos (para servicios prepago)
- **Ver Detalles** - Ver desglose detallado del uso
- **Exportar** - Descargar informe de uso (si está disponible)

Próximas Renovaciones Automáticas

La vista de Próximas Renovaciones Automáticas proporciona una interfaz centralizada para monitorear y gestionar todas las renovaciones de servicios programadas en toda su organización.

Accediendo a Renovaciones Automáticas

Navegue a:

O directamente:

Esto muestra todos los servicios con renovación automática programada, ordenados por la próxima fecha de renovación.

Visión General de Renovaciones Automáticas

La lista de renovaciones automáticas muestra las renovaciones programadas en una tabla con:

Columnas:

- **Cliente** - Nombre del cliente (enlace clicable a la vista general del cliente)
- **Servicio** - Nombre del servicio (enlace clicable a la vista general del cliente)
- **Producto** - Nombre del producto/plan que se renueva
- **Costo** - Costo de renovación (del precio de venta del producto)
- **Renueva El** - Fecha y hora de la próxima renovación con formato legible por humanos
- **Estado** - Estado del servicio (Activo, Suspendido, etc.)
- **Acciones** - Renovar ahora o eliminar la renovación automática

Ejemplo de Visualización:

Cómo Funciona la Renovación Automática

La renovación automática se programa en CGRateS (el sistema de facturación) cuando se crea o modifica un servicio. El sistema:

1. **Programa Acción** - Crea un ActionPlan en CGRateS con la fecha de renovación
2. **Monitorea Saldo** - Verifica si el cliente tiene saldo suficiente antes de la renovación
3. **Ejecuta Renovación** - En la fecha programada, renueva automáticamente el servicio
4. **Actualiza Registros** - Crea transacciones, actualiza fechas de servicio, y registra la actividad

Fuente de Datos:

La vista de Próximas Renovaciones Automáticas consulta CGRateS directamente utilizando el `ApierV1.GetScheduledActions` API y enriquece los datos con información de cliente y servicio de la base de datos CRM.

Renovar Ahora

Para activar manualmente una renovación antes de la fecha programada:

1. Haga clic en el botón **(Renovar Ahora)** para el servicio
2. Revise los detalles de la renovación en el modal de confirmación:

   > Cliente: Acme Corp Servicio: FixedWireless_75628fa5 Producto: Internet Hogar Mega Costo: $89.99
   >
   > Próxima renovación programada: 11 de Noviembre de 2025 10:45 AM
   >
   > [Cancelar] [Confirmar Renovación]

3. Haga clic en **"Confirmar Renovación"** para procesar inmediatamente
4. Se crea y ejecuta un trabajo de aprovisionamiento
5. Monitoree el progreso en el modal de estado de aprovisionamiento

Qué Sucede:

- El servicio se recarga con las asignaciones del producto
- El saldo se actualiza en CGRateS
- Se crea un registro de transacción
- La fecha de finalización del servicio se extiende
- Se crea una entrada en el registro de actividad

Casos de Uso:

- El cliente solicita una renovación anticipada
- Resolver problemas de servicio restableciendo asignaciones
- Probar procesos de renovación
- El cliente desea usar el servicio antes de la renovación programada

Eliminar Renovación Automática

Para cancelar la renovación automática de un servicio:

1. Haga clic en el botón **(Eliminar Renovación Automática)** para el servicio
2. Confirme la eliminación en el modal:

> Esto eliminará la renovación automática programada para:
>
> Cliente: Acme Corp Servicio: FixedWireless_75628fa5 Próxima Renovación: 11 de Noviembre de 2025 10:45 AM
>
> El servicio no se renovará automáticamente. Deberá renovar manualmente o el servicio expirará en la fecha de finalización.
>
> [Cancelar] [Eliminar Renovación Automática]

3. Haga clic en **"Eliminar Renovación Automática"** para confirmar
4. El ActionPlan se elimina de CGRateS
5. El servicio expirará naturalmente a menos que se renueve manualmente

Qué Sucede:

- El ActionPlan se elimina de CGRateS
- El estado del servicio permanece sin cambios
- El servicio expirará en su fecha de finalización actual
- El cliente debe renovar manualmente o el servicio se detendrá

Casos de Uso:

- El cliente está cancelando el servicio
- Cambiando a un proceso de renovación manual
- El plan de servicio está cambiando
- El cliente solicitó detener los cargos automáticos

> ⚠ **AVISO**
>
> Eliminar la renovación automática significa que el servicio expirará a menos que se renueve manualmente. Los clientes no recibirán la continuación automática del servicio.

Entendiendo la Visualización

Columna de Cliente:

- Muestra el nombre del cliente como un enlace clicable
- Enlaces a la página de vista general del cliente
- Muestra "N/A" si el servicio no está vinculado a un cliente (servicio huérfano)

Columna de Servicio:

- Muestra el nombre/UUID del servicio
- Enlaces a la página de vista general del cliente
- Muestra el UUID del servicio si no se establece el nombre
- Muestra "N/A" si no se puede encontrar el servicio en la base de datos

Columna de Producto:

- Nombre del producto del ActionPlan de CGRateS
- Extraído de los metadatos de la acción programada
- Siempre muestra el nombre del producto incluso si no se encuentra el servicio

Columna de Costo:

- Costo de venta de la tabla de Productos en CRM
- Monto que se cobrará en la renovación
- Muestra "N/A" si no se encuentra el producto en la base de datos

Columna de Renueva El:

- Fecha y hora exactas de la próxima renovación
- Tiempo relativo legible por humanos (por ejemplo, "3 horas a partir de ahora", "2 días a partir de ahora")
- Consciente de la zona horaria (utiliza la zona horaria del servidor CGRateS)

Columna de Estado:

- Estado actual del servicio de la base de datos CRM
- **Activo** (Verde) - El servicio está operativo
- **Suspendido** (Naranja) - El servicio está deshabilitado temporalmente
- **Desconocido** (Gris) - Servicio no encontrado en la base de datos o estado no establecido

Solución de Problemas

Servicios que muestran "N/A" para Cliente/Servicio

- **Causa:** El UUID del servicio en CGRateS no coincide con el service_uuid en la base de datos CRM
- **Solución:**
  - Verifique que el servicio exista en la base de datos
  - Verifique que el formato de service_uuid coincida: `ServiceType_UUID` (por ejemplo, `FixedWireless_75628fa5`)

    - El servicio puede haber sido eliminado de CRM pero aún programado en CGRateS

Costo mostrando "N/A"

- **Causa:** El ID del producto en CGRateS no existe en la tabla de Productos de CRM
- **Solución:**
    - Verifique que el producto exista en la base de datos
    - Verifique que el product_id en el ActionPlan coincida con un producto en CRM
    - El producto puede haber sido eliminado

"Renueva El" no muestra fecha

- **Causa:** NextRunTime no presente en la respuesta de CGRateS
- **Solución:**
    - Verifique la configuración del ActionPlan de CGRateS
    - Verifique que ActionTiming esté configurado correctamente
    - Verifique los registros de CGRateS en busca de errores

"Renovar Ahora" falla

- **Causa:** Varios errores de aprovisionamiento
- **Solución:**
    - Verifique el modal de estado de aprovisionamiento para detalles de errores
    - Verifique que los libros de jugadas de Ansible estén configurados correctamente
    - Verifique la conectividad de OCS
    - Revise los registros de aprovisionamiento

Renovación automática eliminada pero aún mostrando

- **Causa:** Retraso en la caché o problema de sincronización de CGRateS
- **Solución:**
    - Haga clic en el botón **"Actualizar"** para recargar los datos

- Verifique que el ActionPlan se haya eliminado realmente en CGRateS
- Verifique la conectividad de la API de CGRateS

Paginación

Los servicios se muestran en páginas para mejorar el rendimiento:

> ◄ Anterior 1 [2] 3 4 5 Siguiente ►
>
> Elementos por página: [10 ▼]
>
> - 10
> - 25
> - 50
> - 100

Controles:

- **Anterior/Siguiente** - Navegar por las páginas
- **Números de Página** - Saltar a una página específica
- **Elementos por Página** - Ajustar cuántos servicios se muestran por página

Consejo de Rendimiento: Utilice filtros para reducir los resultados totales en lugar de aumentar los elementos por página.

Insignias e Indicadores de Servicio

Los indicadores visuales ayudan a identificar rápidamente los estados del servicio:

Insignias de Estado:

Indicador de Renovación Automática:

Los servicios con renovación automática habilitada muestran:

Expirando Pronto:

Los servicios que expiran en 7 días muestran:

Atrasado:

Los servicios con saldo pendiente muestran:

Integración CGRateS (Avanzada)

Para servicios integrados con CGRateS (el motor de facturación y calificación), los administradores pueden gestionar configuraciones avanzadas directamente desde la vista del servicio. Esto incluye atributos, filtros y visualización de sesiones activas.

> ⓘ **NOTA**
>
> Las características de integración de CGRateS requieren el permiso **cgrates_api_access**. Solo los administradores tienen acceso a estas características por defecto. Consulte `rbac` para la configuración de permisos.

Aprovisionamiento Automático vs. Gestión Manual

En operación normal, los atributos y filtros de CGRateS se aprovisionan automáticamente por Ansible durante el flujo de trabajo de aprovisionamiento inicial del servicio. Cuando se crea un nuevo servicio, los libros de jugadas de aprovisionamiento:

- Crean la cuenta de servicio en CGRateS
- Configuran atributos (IMSI, MSISDN, identificadores de cuenta, perfiles de velocidad, etc.)
- Configuran filtros para asegurar la calificación correcta
- Aplican los planes de calificación apropiados

Sin embargo, hay casos en los que puede necesitar modificar estas configuraciones después de que el servicio ha sido aprovisionado:

- **Cambios en el Perfil de Velocidad** - El cliente actualiza/disminuye el ancho de banda (MaxBitrateDL/UL)

- **Ajustes de Políticas** - Cambiar políticas de QoS o reglas de modelado de tráfico (PcefPolicyName)
- **Cambios en el Número de Teléfono** - Actualizar MSISDN u otros identificadores
- **Solución de Problemas** - Corregir configuraciones incorrectas o probar diferentes ajustes
- **Configuraciones Especiales** - Aplicar atributos personalizados que no forman parte de aprovisionamiento estándar

La interfaz de gestión manual permite a los administradores realizar estos cambios directamente sin volver a ejecutar todo el flujo de trabajo de aprovisionamiento. Esto es particularmente útil para:

- **Cambios Rápidos** - Modificar un solo atributo sin esperar el aprovisionamiento
- **Pruebas** - Experimentar con diferentes configuraciones
- **Soporte al Cliente** - Resolver problemas sobre la marcha durante las llamadas de soporte
- **Configuraciones Personalizadas** - Aplicar ajustes específicos del servicio que no están cubiertos por plantillas

> ⚠ **AVISO**
>
> Los cambios manuales en la configuración de CGRateS evitan el flujo de trabajo de aprovisionamiento estándar. Asegúrese de entender el impacto de sus cambios, ya que configuraciones incorrectas pueden afectar la facturación y la funcionalidad del servicio. Todos los cambios se registran en el feed de actividad del cliente para fines de auditoría.

Accediendo a las Características de CGRateS

Al ver o editar un servicio que está aprovisionado en CGRateS, aparecen tres secciones colapsables en la parte inferior del formulario del servicio:

- **Atributos de CGRateS** - Configurar atributos específicos del servicio
- **Filtros de CGRateS** - Definir reglas de filtrado para el servicio
- **Sesiones Activas** - Ver sesiones activas en tiempo real

Cada sección está colapsada por defecto para mantener la interfaz limpia. Haga clic en el encabezado de la sección para expandir y ver/editar la configuración.

Las secciones colapsadas muestran insignias de conteo que indican cuántos atributos, filtros o sesiones activas existen para el servicio.

Atributos de CGRateS

Los atributos le permiten definir campos y transformaciones personalizados que se aplican a los eventos de calificación para este servicio específico.

Formato del ID de Atributo: `ATTR_ACCOUNT_{service_uuid}`

Ejemplo: Para el servicio con UUID `Mobile_SIM_c2880638`, el ID del perfil de atributo es `ATTR_ACCOUNT_Mobile_SIM_c2880638`

Gestionando Atributos:

1. Abra la vista de edición del servicio
2. Expanda la sección **"Atributos de CGRateS"**
3. Haga clic en el botón **"Editar Atributos"**
4. Agregue/modifique/elimine atributos según sea necesario
5. Haga clic en **"Guardar Atributos"**

Campos de Atributo:

- **Ruta** - El campo a modificar (por ejemplo, `*req.Account`, `*req.IMSI`)
- **Tipo** - Cómo se establece el valor:

    - `*constant` - Establece un valor fijo
    - `*variable` - Captura el valor de los campos de evento utilizando RSRParser
    - `*composed` - Añade el valor en lugar de sobrescribir
    - `*usage_difference` - Calcula la duración entre dos campos
    - `*sum` - Suma múltiples valores
    - `*value_exponent` - Calcula el exponente de un campo
- **Reglas** - El/los valor(es) a aplicar (puede tener múltiples reglas por atributo)

Ejemplo de Configuración de Atributo:

```
{
  "Path": "*req.Account",
  "Type": "*constant",
  "Value": [{"Rules": "Mobile_SIM_474a380a"}]
}
```

Casos de Uso Comunes:

- Establecer identificador de cuenta para calificación
- Mapear IMSI/MSISDN al servicio
- Configurar límites de ancho de banda (MaxBitrateDL/UL)
- Establecer nombres de políticas (PcefPolicyName)
- Transformar o enriquecer eventos de calificación

Registro de Actividad:

Todas las modificaciones de atributos se registran en el feed de actividad del cliente con todos los detalles de los cambios realizados.

Filtros de CGRateS

Los filtros definen reglas de coincidencia que determinan cuándo debe aplicarse la configuración de este servicio durante la calificación.

Formato del ID de Filtro: `FLTR_ACCOUNT_{service_uuid}`

Ejemplo: Para el servicio con UUID `Mobile_SIM_c2880638`, el ID del filtro es `FLTR_ACCOUNT_Mobile_SIM_c2880638`

Gestionando Filtros:

1. Abra la vista de edición del servicio
2. Expanda la sección **"Filtros de CGRateS"**
3. Haga clic en el botón **"Editar Filtros"**
4. Agregue/modifique/elimine reglas de filtro
5. Haga clic en **"Guardar Filtros"**

Campos de Regla de Filtro:

- **Elemento** - El campo contra el que coincidir (por ejemplo, `~*req.Account`, `~*req.Destination`)
- **Tipo** - Tipo de coincidencia:
  - `*string` - Coincidencia exacta de cadena
  - `*prefix` - Comienza con el valor especificado
  - `*suffix` - Termina con el valor especificado
  - `*empty` - El campo está vacío
  - `*exists` - El campo existe
  - `*notexists` - El campo no existe
  - `*timings` - Coincidir patrones de tiempo/fecha
  - `*destinations` - Coincidir patrones de destino
  - `*rsr` - Coincidencia de campo RSR
  - `*gt` / `*gte` / `*lt` / `*lte` - Comparaciones numéricas
- **Valores** - El/los valor(es) a coincidir (puede tener múltiples valores por regla)

Ejemplo de Configuración de Filtro:

```
{
  "Element": "~*req.Account",
  "Type": "*string",
  "Values": ["Mobile_SIM_474a380a"]
}
```

Casos de Uso Comunes:

- Asegurar que los atributos solo se apliquen a una cuenta específica
- Filtrar por destino (nacional vs. internacional)
- Filtrado basado en tiempo (horas pico vs. fuera de pico)
- Filtrar por tipo o categoría de servicio

Registro de Actividad:

Todas las modificaciones de filtros se registran en el feed de actividad del cliente.

Sesiones Activas

Vea sesiones activas en tiempo real para este servicio. Esto muestra llamadas en curso, sesiones de datos u otros eventos facturables que están actualmente en progreso.

Visualizando Sesiones Activas:

1. Abra la vista de edición del servicio
2. Expanda la sección **"Sesiones Activas"**
3. Vea la lista de sesiones activas
4. Haga clic en **"Ver Detalles"** en cualquier sesión para ver los datos completos de la sesión
5. Haga clic en **"Actualizar"** para recargar la lista de sesiones

Información de Sesión Mostrada:

- **Hora de Configuración** - Cuando comenzó la sesión
- **Uso** - Duración actual de la sesión (en segundos)
- **Destino** - Número llamado o destino

Modal de Detalles de Sesión:

Hacer clic en "Ver Detalles" abre un modal que muestra:

- **Información Básica:**
  - CGRID (ID de sesión)
  - Cuenta (UUID del servicio)
  - Hora de configuración
  - Uso/duración actual
  - Destino
  - Categoría
- **Datos Completos de la Sesión:**
  - Representación JSON completa de la sesión
  - Todos los campos de sesión de CGRateS
  - Estado de sesión en tiempo real
  - Visor JSON desplazable para inspección

Casos de Uso:

- Monitorear llamadas activas o sesiones de datos
- Solucionar problemas de facturación

- Verificar que la sesión se esté calificando correctamente
- Comprobar atributos y valores de la sesión
- Auditar el uso activo del servicio

Frecuencia de Actualización:

Las sesiones se obtienen bajo demanda cuando expande la sección. Haga clic en "Actualizar" para obtener los datos de sesión más recientes.

> ⓘ **NOTA**
>
> Solo se muestran las sesiones que coinciden con la cuenta de este servicio (UUID del servicio). El filtro `*string:~*req.Account:{service_uuid}` se aplica automáticamente.

Proxy de API de CGRateS

Todas las operaciones de CGRateS (atributos, filtros, sesiones) utilizan el endpoint de proxy de la API de OmniCRM:

Endpoint: `POST /crm/ocs/proxy`

Campos Requeridos:

- `method` - Método de API de CGRateS (por ejemplo, `APIerSv1.GetAttributeProfile`)
- `params` - Array de parámetros para el método
- `customer_id` - ID del cliente (para registro de actividad)
- `service_id` - ID del servicio (para registro de actividad)

Campos Opcionales:

- `tenant` - Inquilino de CGRateS (se establece en el valor de configuración por defecto)

Ejemplo de Solicitud:

```
{
  "method": "APIerSv1.GetAttributeProfile",
  "params": [{"ID": "ATTR_ACCOUNT_Mobile_SIM_c2880638"}],
  "customer_id": 123,
  "service_id": 456
}
```

Configuración de Inquilino:

El inquilino se establece automáticamente desde el archivo de configuración de OmniCRM (`crm_config.yaml`) bajo `ocs.ocsTenant`. Esto asegura que todas las operaciones de CGRateS utilicen el inquilino correcto sin codificar valores en el frontend.

Requisito de Permiso:

Se requiere el permiso `cgrates_api_access`. Este permiso se otorga al rol `admin` por defecto.

Registro de Actividad:

Todas las operaciones de API de CGRateS que no son GET se registran automáticamente en el feed de actividad del cliente, incluyendo:

- Método de API llamado
- Inquilino utilizado
- Parámetros completos enviados
- ID del servicio sobre el que se realizó la operación
- Usuario que realizó la operación
- Marca de tiempo

Esto crea una auditoría completa de todos los cambios de configuración de CGRateS.

Solución de Problemas de Integración de CGRateS

"Permiso Denegado" al acceder a características de CGRateS

- **Causa:** El usuario carece del permiso `cgrates_api_access`
- **Solución:** Otorgar permiso al rol del usuario (típicamente característica solo para administradores)

Atributos o Filtros no cargando

- **Causa:** Problema de conectividad de CGRateS o el perfil no existe
- **Solución:**
  - Verifique la conectividad del servidor CGRateS en la configuración
  - Verifique que la configuración del inquilino sea correcta
  - Verifique la consola del navegador en busca de errores de API
  - El perfil puede no existir aún (mostrará un formulario vacío)

Cambios no guardados

- **Causa:** Error de validación o error de API de CGRateS
- **Solución:**
  - Verifique los campos requeridos (Ruta, Tipo, Elemento)
  - Verifique que el formato JSON sea correcto
  - Verifique el registro de actividad para detalles de errores
  - Revise los registros de CGRateS

No se muestran sesiones activas

- **Causa:** No hay sesiones actualmente activas para este servicio
- **Solución:**
  - Esto es normal si el servicio no está en uso
  - Intente actualizar después de iniciar una sesión (llamada, datos, etc.)
  - Verifique que el UUID del servicio coincida con la cuenta en CGRateS

Detalles de sesión no actualizándose en tiempo real

- **Causa:** Los datos de sesión se obtienen bajo demanda, no en vivo
- **Solución:** Haga clic en el botón "Actualizar" para obtener los datos de sesión más recientes

El registro de actividad no muestra cambios de CGRateS

- **Causa:** Solo se registran operaciones que no son GET (las lecturas no se registran)
- **Solución:** Esto es por diseño - solo las escrituras/modificaciones crean entradas de actividad

Flujos de Trabajo Comunes

Flujo de Trabajo 1: Encontrar Servicios del Cliente

1. Haga clic en el **filtro de Tipo de Servicio** (opcional)
2. Haga clic en el **filtro de Cliente**
3. Escriba el nombre del cliente
4. Seleccione al cliente del desplegable
5. Revise los servicios del cliente

Flujo de Trabajo 2: Identificar Servicios que Expiran

1. Haga clic en la pestaña **"Activos"**
2. Ordene por **"Fecha de Finalización"** (ascendente)
3. Los servicios que expiran primero aparecen primero
4. Contacte a los clientes para la renovación

Flujo de Trabajo 3: Limpiar Servicios de Prueba

1. Busque "test" en el cuadro de búsqueda
2. Revise los resultados para confirmar que son datos de prueba
3. Seleccione todos los servicios de prueba
4. Haga clic en **"Eliminar Seleccionados"**
5. Confirme la eliminación

Flujo de Trabajo 4: Suspender Cliente Moroso

1. Navegue a la cuenta del cliente
2. Vea la pestaña de servicios
3. Seleccione todos los servicios activos
4. Cambie el estado a "Suspendido"
5. Guarde los cambios

Flujo de Trabajo 5: Ver Uso del Servicio Móvil

1. Filtre por **Tipo de Servicio: Móvil**
2. Haga clic en el nombre del servicio para abrir los detalles
3. Haga clic en la pestaña **"Uso"**
4. Revise el consumo de datos/voz/SMS
5. Identifique a los usuarios intensivos o sobrecargas

Solución de Problemas

Servicios no apareciendo en la lista

- **Causa:** Filtro de estado ocultando resultados
- **Solución:** Haga clic en la pestaña "Todos los Resultados" para mostrar todos los estados

No se puede encontrar el servicio por búsqueda

- **Causa:** El término de búsqueda no coincide con los datos almacenados
- **Solución:**
  - Intente una búsqueda parcial (por ejemplo, "0770" en lugar del número completo)
  - Utilice el filtro de cliente en su lugar
  - Verifique si hay errores tipográficos

Botón de eliminación masiva deshabilitado

- **Causa:** No se han seleccionado servicios

- **Solución:** Marque las casillas junto a los servicios que desea eliminar

Ordenar no funciona

- **Causa:** La columna no es ordenable
- **Solución:** Solo las columnas con iconos ▲▼ son ordenables

La página carga lentamente

- **Causa:** Demasiados servicios para mostrar
- **Solución:**
  - Aplique filtros para reducir el conjunto de resultados
  - Reduzca los elementos por página
  - Utilice la búsqueda para reducir resultados

Documentación Relacionada

- `csa_add_service` - Añadiendo nuevos servicios
- `csa_modify` - Modificando servicios y añadiendo complementos
- `concepts_products_and_services` - Conceptos de productos y servicios
- `basics_customers` - Gestión de clientes

Seguimiento del Uso del Servicio y del Saldo

El sistema de Uso del Servicio proporciona **monitoreo en tiempo real** del consumo de los clientes para datos, voz, SMS y saldos monetarios. Esta función se integra con el OCS (Sistema de Carga en Línea) para mostrar el uso actual, las asignaciones restantes y la información de caducidad del saldo tanto a los clientes (a través del Portal de Autocuidado) como al personal.

Descripción General

El seguimiento del uso permite:

- **Visualización del Saldo en Tiempo Real** - Ver el uso actual y las asignaciones restantes
- **Múltiples Tipos de Saldo** - Rastrear datos, voz, SMS y saldos monetarios simultáneamente
- **Monitoreo de Caducidad** - Ver cuándo caducan los saldos
- **Desglose del Saldo** - Vista detallada de los diferentes buckets de saldo
- **Actualización Automática** - Actualizaciones de uso cada 3 segundos automáticamente

Accediendo al Uso del Servicio

Desde la Lista de Servicios:

1. Navega a **Servicios → Lista de Servicios**
2. Haz clic en el **menú de acciones (⋮)** junto a un servicio
3. Selecciona **"Uso"**

Desde los Detalles del Servicio:

1. Abre la página de detalles de un servicio

2. Haz clic en la pestaña **"Uso"**

Desde la Página del Cliente:

1. Abre la vista general del cliente
2. Navega a la pestaña **Servicios**
3. Haz clic en **"Ver Uso"** junto a cualquier servicio

Se abre el modal o la página de uso mostrando datos de consumo en tiempo real. Para inspeccionar las llamadas individuales, SMS y sesiones de datos detrás de estos saldos, cambia a la pestaña **Registros de Llamadas (CDRs)**. Consulta Registros de Detalle de Llamadas.

Visualización del Uso

La interfaz de uso muestra tarjetas resumen y barras de progreso detalladas para cada tipo de saldo.

Tarjetas Resumen

La fila superior muestra tarjetas de vista rápida para cada tipo de saldo:

Información de la Tarjeta:

- **Tipo de Saldo** - Icono y etiqueta (Datos, Voz, SMS, Monetario)
- **Cantidad Restante** - Saldo actual en las unidades apropiadas
- **Tiempo de Caducidad** - Días/horas hasta que el saldo caduque
- **Botón de Más Información** - Haz clic para expandir el desglose detallado

Barras de Progreso

Debajo de las tarjetas, las barras de progreso muestran el consumo visualmente con porciones llenas que indican el saldo restante.

Características de la Barra de Progreso:

- **Indicador Visual** - La porción llena muestra el saldo restante
- **Porcentaje** - Porcentaje numérico de saldo restante
- **Valores Absolutos** - Muestra usado vs. total (por ejemplo, "12.5GB / 20GB")
- **Codificación de Color**:
  - Verde: >50% restante
  - Amarillo: 20-50% restante

    - Rojo: <20% restante
- **Clickable** - Haz clic para expandir el desglose detallado

Tipos de Saldo

Saldo de Datos

Rastrea el consumo de datos de internet.

Unidades: Gigabytes (GB) o Megabytes (MB)

Formato de Visualización:

> Progreso: 12.5GB / 20GB (62%)

Escenarios Comunes:

- **Planes de datos móviles** - Asignaciones mensuales de 5GB, 10GB, 20GB
- **Inalámbrico fijo** - Ilimitado o limitado a altas cantidades (500GB, 1TB)
- **Recargas** - Datos adicionales comprados a mitad de ciclo
- **Servicios de dongle** - Datos prepagados para dispositivos hotspot

Múltiples Buckets:

Los servicios a menudo tienen múltiples saldos de datos:

- Asignación mensual (caduca mensualmente)
- Datos de bonificación (caduca después del período de campaña)
- Datos de recarga (caducidad más corta, consumidos primero)

Saldo de Voz

Rastrea minutos de llamadas telefónicas.

Unidades: Minutos (min)

Formato de Visualización:

> Progreso: 125 min / 500 min (25%)

Seguimiento de la Duración de Llamadas:

- Llamadas entrantes (si se cobran)
- Llamadas salientes
- Llamadas internacionales (bucket separado si corresponde)
- Números premium

Cálculo:

El uso de voz se calcula por la duración de la llamada en nanosegundos internamente, convertido a minutos para la visualización.

Saldo de SMS

Rastrea el uso de mensajes de texto.

Unidades: Mensajes (msgs)

Formato de Visualización:

> Progreso: 45 / 250 (18%)

Tipos de Mensajes:

- SMS estándar (160 caracteres)
- SMS largos (múltiples segmentos)
- MMS (si se rastrean por separado)

Saldo Monetario

Rastrea el crédito prepago o el saldo de la cuenta.

Unidades: Moneda (£, $, €, etc.)

Formato de Visualización:

> Progreso: £15.50 / £20.00 (77%)

Uso:

- Las cuentas prepagas utilizan el saldo monetario para pagar por el uso
- El crédito se decrementa a medida que el cliente utiliza los servicios
- Puede recargarse a través de pago o vale
- Puede caducar si no se utiliza dentro del período de validez

Desglose Detallado del Saldo

Haz clic en **"Más Información"** en cualquier tarjeta o haz clic en una barra de progreso para expandir el desglose detallado.

Vista Expandida:

> □ Asignación Mensual 20GB
> Restante: 12.5 GB Usado: 7.5 GB Caduca: 25 Ene 2025 (15 días) Peso: 10
>
> □ Datos de Bonificación 5GB
> Restante: 5.0 GB Usado: 0 GB Caduca: 31 Ene 2025 (21 días) Peso: 20
>
> □ Datos de Recarga 3GB
> Restante: 0 GB Usado: 3.0 GB Caduca: 18 Ene 2025 (CADUCADO) Peso: 30
>
> Total Restante: 17.5 GB

Campos del Bucket de Saldo:

- **ID/Nombre** - Identificador para el bucket de saldo
- **Restante** - Cantidad restante en este bucket específico
- **Usado** - Cantidad consumida de este bucket
- **Fecha de Caducidad** - Cuándo caduca este saldo
- **Peso** - Orden de prioridad (peso más alto consumido primero)

Sistema de Peso

Los saldos tienen un valor de **peso** que determina el orden de consumo:

- **Peso más alto = consumido primero**

- **Peso más bajo = consumido al final**

Ejemplo de Pesos:

- Datos de recarga: Peso 30 (consumido primero, caducidad más corta)
- Datos de bonificación: Peso 20 (consumido segundo)
- Asignación mensual: Peso 10 (consumido al final, caducidad más larga)

Esto asegura que los saldos que están a punto de caducar se utilicen antes que los que duran más.

Para obtener detalles completos sobre las reglas de consumo de saldo, estrategias de peso y cómo CGRateS prioriza los saldos, consulta Acciones de CGRateS y Comportamientos de Recarga.

Actualizaciones en Tiempo Real

Los datos de uso se actualizan automáticamente cada **3 segundos** mediante polling.

Qué se Actualiza:

- Cantidades actuales de saldo
- Barras de progreso de uso
- Temporizadores de caducidad
- Detalles de cada bucket

Experiencia del Usuario:

- No se requiere recarga de página
- Actualizaciones suaves sin parpadeo
- Superposición de carga durante la actualización
- La insignia de estado muestra el estado actual del servicio

Casos de Uso:

- Monitorear el uso del cliente durante la llamada

- Observar la disminución del saldo en tiempo real a medida que el cliente utiliza el servicio
- Verificar la recarga inmediatamente después de la compra

Uso en Diferentes Tipos de Servicios

Servicios Móviles

Muestra los cuatro tipos de saldo:

- Datos (GB)
- Voz (minutos)
- SMS (mensajes)
- Monetario (moneda)

Ejemplo:

> DATOS: 12.5GB restantes VOZ: 125 min restantes SMS: 45 msgs restantes MONETARIO: £15.50 restantes

Inalámbrico Fijo / Internet

Normalmente muestra solo:

- Datos (GB o TB)
- Monetario (si es prepago)

Ejemplo:

> DATOS: 450GB / 500GB restantes MONETARIO: £45.00 (crédito prepago)

Servicios de Hotspot / Dongle

Muestra el seguimiento de datos específico del dongle:

- Datos (consumidos vs. prepagados)
- Monetario (saldo prepago)

Modo de Visualización:

Cuando `dongle=true`, el componente oculta voz y SMS, mostrando solo los saldos de datos y monetarios relevantes.

Solución de Problemas

Uso mostrando como 0 / 0

- **Causa:** Servicio no integrado con OCS o CGRateS
- **Solución:**
  - Verifica que el servicio esté provisionado en OCS
  - Verifica la conectividad de la API de OCS
  - Revisa el mapeo del UUID del servicio

Uso no actualizándose

- **Causa:** Polling detenido o OCS inalcanzable
- **Solución:**
  - Actualiza la página
  - Verifica la consola del navegador en busca de errores
  - Verifica que la API de OCS esté en línea

Los saldos muestran cantidades incorrectas

- **Causa:** Desajuste de datos de OCS o problema de caché
- **Solución:**
  - Fuerza la actualización del saldo de OCS
  - Verifica si hay transacciones pendientes
  - Verifica la configuración de OCS

Fechas de caducidad faltantes

- **Causa:** El saldo no tiene fecha de caducidad establecida

- **Solución:**
  - Algunos saldos están configurados para nunca caducar (validez ilimitada)
  - Verifica la configuración del saldo en OCS

Múltiples saldos confusos

- **Causa:** Se han agregado múltiples recargas o datos de bonificación
- **Solución:**
  - Usa la vista de desglose detallado para ver todos los buckets
  - Ordena por peso para ver el orden de consumo
  - Revisa las fechas de caducidad individuales

Integración con OCS/CGRateS

Los datos de uso provienen del **OCS (Sistema de Carga en Línea)**, típicamente CGRateS. Para obtener información completa sobre cómo funcionan los saldos, incluidos los tipos de saldo, las reglas de consumo, el comportamiento de rollover y las configuraciones de Acción, consulta Acciones de CGRateS y Comportamientos de Recarga.

Flujo de Datos:

1. El usuario abre la vista de uso
2. OmniCRM llama a `GET /crm/service/{service_id}`
3. La API consulta OCS a través del UUID del servicio
4. OCS devuelve el mapa de saldo:

```
{
  "BalanceMap": {
    "*data": [
      {
        "ID": "monthly_data_20GB",
        "Value": 13421772800,
        "ExpiryTime": "2025-01-25T23:59:59Z",
        "Weight": 10
      }
    ],
    "*voice": [
      {
        "ID": "monthly_voice_500min",
        "Value": 7500000000000,
        "ExpiryTime": "2025-01-25T23:59:59Z",
        "Weight": 10
      }
    ],
    "*sms": [
      {
        "ID": "monthly_sms_250",
        "Value": 250,
        "ExpiryTime": "2025-01-25T23:59:59Z",
        "Weight": 10
      }
    ],
    "*monetary": [
      {
        "ID": "prepaid_credit",
        "Value": 1550,
        "ExpiryTime": "2025-02-25T23:59:59Z",
        "Weight": 10
      }
    ]
  }
}
```


5. La UI convierte los valores a unidades de visualización (bytes → GB, nanosegundos → minutos)
6. Se renderizan las barras de progreso y las tarjetas
7. El polling continúa cada 3 segundos

Mapeo de Tipos de Saldo de OCS:

El OCS devuelve datos de saldo con prefijos de tipo que se mapean a la visualización de la UI:

- `*data` → **TARJETA DE DATOS** (uso de internet)
- `*voice` → **TARJETA DE VOZ** (minutos de llamada)
- `*sms` → **TARJETA DE SMS** (mensajes de texto)
- `*monetary` → **TARJETA MONETARIA** (crédito prepago)

Cada tipo de saldo puede tener múltiples buckets (por ejemplo, asignación mensual + datos de bonificación + datos de recarga), todos mostrados en la vista de desglose detallado.

Conversiones de Valores de Saldo:

- **Datos:** Bytes → GB (dividir por 1024^3)
- **Voz:** Nanosegundos → Minutos (dividir por 60×10^9)
- **SMS:** Conteo (sin conversión)
- **Monetario:** Centavos → Moneda (dividir por 100)

Auto-Renovación y Planes de Acción

Los servicios con auto-renovación habilitada tienen **Planes de Acción** programados en el OCS. Para obtener información detallada sobre cómo funcionan las Acciones, la manipulación de saldos y la configuración de Planes de Acción, consulta Acciones de CGRateS y Comportamientos de Recarga.

¿Qué son los Planes de Acción?

Los Planes de Acción son tareas programadas en CGRateS que se ejecutan automáticamente en momentos específicos para:

- Agregar saldo a una cuenta (recarga automática)
- Renovar asignaciones mensuales
- Aplicar cargos recurrentes
- Caducar saldos antiguos

Cómo Funciona la Auto-Renovación:

1. **Provisionamiento del Servicio:**
   - Cuando se crea el servicio con `auto_renew = true`
   - El libro de provisión crea un Plan de Acción en OCS
   - El Plan de Acción está configurado para ejecutarse mensualmente (o por ciclo de facturación)
2. **Configuración del Plan de Acción:**

   El Plan de Acción contiene:

   - **ID de Cuenta** - UUID del servicio
   - **Acciones** - Qué hacer (agregar datos, voz, SMS, saldo monetario)
   - **Horario** - Cuándo ejecutar (por ejemplo, mensualmente el 1 a las 00:00 UTC)
   - **Cantidad** - Cuánto saldo agregar
3. **Ejecución Automática:**
   - OCS ejecuta el Plan de Acción en el momento programado
   - Agrega saldo a la cuenta (por ejemplo, 20GB de datos, 500 minutos de voz)
   - Establece la fecha de caducidad para el nuevo saldo (por ejemplo, 30 días)
   - Se cobra al cliente a través del método de pago registrado
4. **Visualización de Planes de Acción:**
   - Navega a los detalles del servicio en la vista de OCS
   - Los Planes de Acción se enumeran con el próximo tiempo de ejecución
   - Muestra: Nombre del plan, fecha de próxima ejecución, detalles de la acción

Ejemplo de Plan de Acción:

```
{
  "ActionPlanId":
"ProductID_MonthlyPlan__ProductName_20GB_Mobile__ActionPlan_Monthly_F
  "NextExecTime": "2025-02-01T00:00:00+00:00",
  "ActionName_hr": "Renovación Mensual",
  "PlanName": "20GB Móvil",
  "ActionFrequency_hr": "Cada Plan Mensual",
  "custom_NextExecTime_hr": "en 22 días"
}
```


Gestión de la Auto-Renovación:

- **Habilitar** - Configurado durante la creación o modificación del servicio
- **Deshabilitar** - Eliminar el Plan de Acción de OCS (el servicio mantiene el saldo existente pero no se renovará automáticamente)
- **Modificar** - Cambiar la cantidad de renovación o la frecuencia a través de la modificación del servicio

Renovación Manual:

Si la auto-renovación está deshabilitada, el cliente debe manualmente:

- Recargar antes de que el saldo caduque
- O el servicio se suspende cuando se agota el saldo

Visualización en la UI:

La pestaña de servicios muestra el estado de auto-renovación:

> Próxima Renovación: 1 Feb 2025 (en 22 días) Monto de Renovación: £15.00

Mejores Prácticas

Para el Personal de Soporte:

- Verifica el uso antes de responder a llamadas de "¿Por qué mi servicio es lento?"
- Verifica el saldo después de las recargas para confirmar el éxito

- Usa el desglose detallado para identificar buckets caducados
- Monitorea a los clientes de alto uso para prevenir sobrecargos

Para los Clientes (Autocuidado):

- Verifica el uso regularmente para evitar quedarte sin saldo
- Recarga antes de que el saldo caduque
- Comprende el sistema de peso para saber qué saldo se consume primero
- Contacta al soporte si el uso parece incorrecto

Para los Administradores:

- Configura las caducidades de saldo apropiadas
- Establece valores de peso para priorizar los saldos que caducan
- Monitorea la conectividad de OCS para informes precisos
- Revisa que las configuraciones de saldo coincidan con las ofertas de productos

Documentación Relacionada

- Acciones de CGRateS y Comportamientos de Recarga - Guía completa sobre tipos de saldo, reglas de consumo, pesos, comportamiento de rollover y configuraciones de Acción
- Destinos de CGRateS - Configuración de destinos para restricciones de saldo geográficas y basadas en PLMN
- `features_topup_recharge` - Sistema de recarga para agregar saldo
- `csa_service_management` - Gestión de servicios
- `csa_modify` - Agregar complementos para aumentar las asignaciones
- `concepts_products_and_services` - Configuración de productos

Atención al Cliente

Suplantación de Usuario para Soporte y Solución de Problemas

La suplantación permite al personal autorizado iniciar sesión temporalmente como otro usuario. El personal puede solucionar problemas, verificar configuraciones o ver exactamente lo que experimenta el usuario. Esta función es esencial para el soporte al cliente. Requiere permisos apropiados y está completamente auditada.

Cuando suplantas a un cliente, accedes al Portal de Autocuidado exactamente como lo ve el cliente. Esto proporciona una solución de problemas y soporte precisos.

Consulta también: RBAC para la configuración de permisos, Clientes para la gestión de clientes, Portal de Autocuidado para las características del portal del cliente.

Propósito

La suplantación de usuario proporciona:

1. **Solución de Problemas**: Ver exactamente lo que ve el cliente para diagnosticar problemas
2. **Verificación**: Confirmar que las configuraciones y permisos del servicio funcionan correctamente
3. **Capacitación**: Demostrar características desde la perspectiva del cliente
4. **Soporte**: Ayudar a los clientes a navegar por el sistema sin compartir pantalla
5. **Registro de Auditoría**: El sistema registra todas las sesiones de suplantación para seguridad y cumplimiento

Permisos Requeridos

Para suplantar usuarios, debes tener uno de los siguientes permisos:

- `can_impersonate`: Permiso de suplantación dedicado para el personal de soporte
- `admin`: Acceso administrativo completo (incluye derechos de suplantación)

Los usuarios sin estos permisos no pueden acceder a la función de suplantación.

Cómo Suplantar a un Usuario

A través de la Interfaz Web:

1. **Navegar al Cliente**: Encuentra al cliente en el CRM
2. **Seleccionar Contactos**: Ver la lista de contactos del cliente
3. **Hacer clic en "Iniciar sesión como Usuario"**: El botón aparece junto a cada contacto que tiene una cuenta de usuario
4. **Confirmar Suplantación**: El sistema puede solicitar confirmación
5. **Inicio de Sesión**: Ahora has iniciado sesión como ese usuario

A través de la API:

Iniciar sesión de suplantación:

Endpoint: `POST /auth/impersonate`

Permiso Requerido: `can_impersonate` o `admin`

Solicitud:

```
{
  "user_id": 42
}
```

Respuesta:

```
{
  "success": true,
  "impersonating_user_id": 1,
  "target_user_id": 42,
  "impersonation_start": "2025-01-04T15:30:00Z",
  "access_token": "new_token_for_impersonated_user",
  "refresh_token": "new_refresh_token"
}
```

Los tokens devueltos son para la sesión del usuario suplantado.

Qué Ocurre Durante la Suplantación

Cuando suplantas a un usuario:

- **Contexto Completo**: Ves el sistema exactamente como lo ve el usuario objetivo:
  - Su panel de control y navegación
  - Sus datos de cliente (si es un usuario cliente)
  - Sus permisos y controles de acceso
  - Sus servicios, facturas y uso

- **Seguimiento de Sesiones**: El sistema rastrea ambas identidades:
  - `impersonating_user_id`: Tu ID de usuario real
  - `target_user_id`: El usuario que estás suplantando
  - `impersonation_start`: Cuándo comenzó la suplantación
- **Indicador**: La interfaz muestra un banner que indica que estás suplantando a alguien:
  - "Actualmente has iniciado sesión como [Nombre de Usuario]"
  - "Haz clic aquí para detener la suplantación"
- **Registro de Auditoría**: El sistema registra todas las acciones con ambos IDs de usuario:
  - Las acciones aparecen como realizadas por el usuario objetivo
  - Los registros de auditoría registran quién las realizó realmente (usuario suplantador)
  - Se mantiene un registro de auditoría completo en la tabla `ImpersonationLog`

Detener la Suplantación

A través de la Interfaz Web:

1. **Hacer clic en el Banner**: Haz clic en el banner de suplantación en la parte superior de la página
2. **O Navegar**: Ve al menú de usuario y selecciona "Detener Suplantación"
3. **Confirmación**: La sesión termina y regresas a tu propia cuenta

A través de la API:

Endpoint: `POST /auth/stop_impersonation`

Solicitud: No se requiere cuerpo (solicitud autenticada)

Respuesta:

```
{
  "success": true,
  "impersonation_end": "2025-01-04T15:45:00Z",
  "duration_seconds": 900,
  "access_token": "your_original_token",
  "refresh_token": "your_original_refresh_token"
}
```

Tu sesión original se restaura.

Registro de Auditoría de Suplantación

Todas las sesiones de suplantación se registran en la tabla `ImpersonationLog` con:

- **impersonating_user_id**: Miembro del personal que realizó la suplantación
- **target_user_id**: Cliente o usuario que fue suplantado
- **impersonation_start**: Marca de tiempo de inicio
- **impersonation_end**: Marca de tiempo de finalización (cuando se detuvo la sesión)
- **impersonation_duration**: Duración en segundos

Esto proporciona una responsabilidad completa para todas las sesiones de suplantación y permite:

- **Auditorías de seguridad**: Revisar quién suplantó a quién y cuándo
- **Informes de cumplimiento**: Demostrar el uso adecuado de acceso elevado
- **Investigación**: Rastrear acciones durante las sesiones de suplantación
- **Monitoreo**: Identificar patrones inusuales de suplantación

Visualización de Registros de Suplantación:

Los administradores pueden consultar los registros de suplantación a través de la API:

`GET /auth/impersonation_logs?user_id={user_id}&start_date={date}&end_date={date}`

Filtrar por:

- Usuario suplantador (quien realizó la suplantación)
- Usuario objetivo (quien fue suplantado)
- Rango de fechas
- Duración

Mejores Prácticas

1. **Minimizar Duración**: Suplantar solo el tiempo necesario para resolver el problema
2. **Documentar Propósito**: Anotar por qué se necesitó la suplantación en el registro de actividad del cliente
3. **Informar a los Clientes**: Hacer saber a los clientes que puede que necesites ver su cuenta (política de privacidad)
4. **Verificar Identidad**: Confirmar la identidad del usuario antes de suplantar su cuenta
5. **Revisar Registros**: Auditar regularmente los registros de suplantación en busca de patrones inusuales
6. **Limitar Permisos**: Conceder `can_impersonate` solo al personal de soporte que lo necesite
7. **Capacitación**: Asegurarse de que el personal entienda la responsabilidad y las implicaciones de auditoría

Consideraciones de Seguridad

- **Acceso Completo**: La suplantación otorga acceso completo a la cuenta del usuario objetivo
- **No Se Requiere Contraseña**: La suplantación elude la autenticación (solo basada en permisos)

- **Acciones Registradas**: Los registros de la aplicación atribuyen todas las acciones durante la suplantación al usuario objetivo (pero los registros de auditoría muestran al actor real)
- **Aislamiento de Sesiones**: La suplantación crea una nueva sesión. No afecta las sesiones activas del usuario objetivo
- **Límites de Tiempo**: Las sesiones de suplantación deben tener un límite de tiempo (configurable)
- **Elusión de MFA**: La suplantación elude los requisitos de 2FA (utiliza la autenticación del suplantador)

Restricciones

- **No Se Pueden Suplantar Administradores**: Dependiendo de la configuración, es posible que no puedas suplantar a otros administradores
- **Mismos Permisos**: Obtienes los permisos del usuario objetivo, no la unión de los permisos de ambos usuarios
- **Límites de Sesión**: Solo se permite una sesión de suplantación por miembro del personal a la vez
- **Requisitos de Auditoría**: No puedes desactivar o ocultar el registro de suplantación

Solución de Problemas Comunes

Problema: El botón "Iniciar sesión como Usuario" no aparece

- Solución: Verifica que tengas el permiso `can_impersonate` o `admin`
- Solución: Confirma que el contacto tenga una cuenta de usuario vinculada

Problema: La suplantación falla con un error de permiso

- Solución: Verifica si el usuario objetivo es un administrador (puede estar restringido)
- Solución: Verifica que tu permiso de suplantación esté activo

Problema: No se puede detener la suplantación

- Solución: Utiliza el endpoint de la API para detener la suplantación: `POST /auth/stop_impersonation`
- Solución: Borra las cookies del navegador e inicia sesión nuevamente con tus credenciales

Problema: Las acciones no se registran correctamente

- Solución: Verifica que la sesión de suplantación esté activa (verifica el banner)
- Solución: Revisa los registros de auditoría: las acciones se registran con ambos IDs de usuario

Definición de Productos y Configuración de CGRateS

Esta guía explica cómo definir productos en OmniCRM junto con su configuración de facturación asociada de CGRateS.

Los productos y sus Acciones de CGRateS se definen típicamente juntos en scripts de Python que configuran tanto el CRM como el OCS (Sistema de Carga en Línea).

Visión General

Definir un producto completo implica configuración en **dos sistemas**:

1. **CGRateS/OCS** - Definir el comportamiento de facturación (Acciones, Tarifas, Destinos)
2. **OmniCRM** - Definir los metadatos del producto (precio, nombre, características)

Estos están intencionadamente poco vinculados, lo que te permite interactuar con CGRateS a través de los Ansible Plays que defines.

Importante: Capacidades de CGRateS

CGRateS es un motor de calificación extremadamente poderoso y flexible. Esta guía muestra **un patrón de configuración común** para productos de OmniCRM, pero CGRateS puede hacer mucho más:

- **Escenarios de calificación avanzados**: Tarifas según la hora del día, niveles de clientes, enrutamiento basado en la calidad de la llamada,

facturación por ráfagas

- **Lógica de roaming compleja**: Tarifas diferentes para llamadas/SMS de MO vs MT cuando se está en roaming, precios específicos de la red
- **Soporte de protocolos heredados**: CAMEL para roaming 2G/3G (SMS MO-MT, USSD), Diameter para LTE/5G
- **Carga sofisticada**: Carga basada en reservas, control de crédito, retroceso en múltiples niveles
- **Enrutamiento dinámico**: Enrutamiento basado en costos, selección de proveedores, LCR (Enrutamiento de Menor Costo)

Esta es solo la guía del CRM - se centra en patrones de definición de productos simples que funcionan bien con el sistema de aprovisionamiento de OmniCRM. Para configuraciones avanzadas de CGRateS, consulta la documentación de CGRateS o consulta Acciones de CGRateS y Comportamientos de Recarga para enfoques de saldo y estrategias de calificación.

Flujo de Trabajo Completo para la Definición de Productos

Paso 1: Autenticarse en la API de CRM

```
import requests
import json

crm_url = 'https://crm.example.com/'
session = requests.Session()

headers = {
    "Content-Type": "application/json"
}

# Obtener token de autenticación
response = session.post(crm_url + '/crm/auth/login', json={
    "email": "admin@example.com",
    "password": "your_password"
}, headers=headers)

assert response.status_code == 200
headers['Authorization'] = 'Bearer ' + response.json()['token']
print("Autenticación en CRM OK")
```

Paso 2: Definir Plantillas de Inventario (Opcional)

Las plantillas de inventario definen los tipos de artículos físicos (tarjetas SIM, enrutadores, etc.) que se pueden asignar durante el aprovisionamiento.

```
inventory_list_new = []

# Definir plantilla de inventario de tarjeta SIM
inventory_list_new.append({
    "item": "SIM Card",
    "itemtext1_label": "ICCID",
    "itemtext2_label": "IMSI",
    "wholesale_cost": 0.2,
    "retail_cost": 1,
    "allow_dropdown_staff": True,
    "allow_dropdown_customer": True,
    "icon": "fa-solid fa-sim-card"
})

# Definir plantilla de inventario de Número Móvil
inventory_list_new.append({
    "item": "Mobile Number",
    "itemtext1_label": "E.164 Mobile Number",
    "itemtext2_label": "Type",
    "wholesale_cost": 0.2,
    "retail_cost": 3,
    "icon": "fa-solid fa-phone"
})

# Verificar si la plantilla de inventario existe, crear o
actualizar
for inventory_template in inventory_list_new:
    # Obtener plantillas existentes
    inventory_list_existing = session.get(
        crm_url + '/crm/inventory/item_template/',
        headers=headers
    )

    # Verificar si la plantilla ya existe
    if inventory_template['item'] in [x['item'] for x in
inventory_list_existing.json()]:
        # Actualizar plantilla existente
        inventory_template_id = [
            x['inventory_template_id']
            for x in inventory_list_existing.json()
            if x['item'] == inventory_template['item']
        ][0]
```

```
        response = session.patch(
            crm_url + '/crm/inventory/item_template/' +
str(inventory_template_id),
            json=inventory_template,
            headers=headers
        )
        assert response.status_code == 200
        print(f"Plantilla de inventario actualizada:
{inventory_template['item']}")
    else:
        # Crear nueva plantilla
        response = session.put(
            crm_url + '/crm/inventory/item_template',
            json=inventory_template,
            headers=headers
        )
        assert response.status_code == 200
        print(f"Plantilla de inventario creada:
{inventory_template['item']}")
```

Campos de Plantilla de Inventario:

- **item** (requerido) - Nombre de la plantilla
- **itemtext1_label** - Etiqueta para el primer campo personalizado
- **itemtext2_label** - Etiqueta para el segundo campo personalizado
- **wholesale_cost** - Tu costo por este artículo
- **retail_cost** - Costo para el cliente
- **allow_dropdown_staff** - Mostrar en los menús desplegables del personal
- **allow_dropdown_customer** - Mostrar en los menús desplegables del cliente
- **icon** - Clase de icono de Font Awesome

Paso 3: Conectar a CGRateS

```
import cgrateshttpapi
import time

OCS_Obj = cgrateshttpapi.CGRateS("ocs.example.com", "2080")

tenant = "your_tenant_name"
tpid = str(tenant) + "_" + str(int(time.time()))
```

Paso 4: Definir Destinos en CGRateS

Los destinos definen DÓNDE se pueden usar los saldos. Deben definirse antes de crear Acciones que los hagan referencia.

Hay dos tipos:

1. **Destinos Geográficos** - Prefijos de número para voz/SMS A lugares específicos (por ejemplo, `Dest_AU_Mobile`, `Dest_International_UK`)
2. **Destinos PLMN** - Códigos de red para uso de datos y roaming (por ejemplo, `Dest_PLMN_OnNet`, `Dest_PLMN_US_Verizon`)

Regla Crítica:

- **Saldos de Voz/SMS** → Usar destinos geográficos (el número que se llama A)
- **Saldos de Datos** → Usar destinos PLMN (la red a la que el cliente está conectado DESDE)

Para ejemplos completos de definición de destinos que incluyen:

- Destinos geográficos (móvil, línea fija, gratuito, internacional)
- Destinos PLMN (en red, roaming, zonas)
- Reglas de formato PLMN (`mccXXX.mncYYY`)
- Cuándo usar cada enfoque

Consulta: Acciones de CGRateS - Definición de Destinos

Paso 5: Definir Acciones de CGRateS

Las Acciones son el mecanismo para agregar saldos a las cuentas de los clientes. Antes de que puedas vincular un producto de CRM a una Acción, la Acción debe estar definida en CGRateS.

Crítico: Las Acciones se definen durante la configuración inicial del sistema (no durante el aprovisionamiento). Especifican qué saldos agregar, tiempos de expiración, comportamiento de rollover, etc.

Consulta Acciones de CGRateS y Comportamientos de Recarga para la documentación completa sobre:

- Cómo definir Acciones usando Python (`OCS_Obj.SendData`)
- Entender `*topup` vs `*topup_reset` (comportamiento de rollover)
- Cálculos de unidades (bytes para datos, nanosegundos para voz, conteo para SMS)
- Referencia de campos (BalanceId, BalanceType, DestinationIDs, Units, ExpiryTime, Weight)
- Ejemplos completos de trabajo (planes de múltiples saldos, complementos de datos, complementos de voz)
- Reglas de consumo de saldo y prioridad

Referencia Rápida de Rollover:

- **`*topup`** - Agrega al saldo existente (habilita rollover). Para: complementos prepagados, paquetes de datos
- **`*topup_reset`** - Reemplaza el saldo existente (sin rollover). Para: planes mensuales con asignaciones fijas

Paso 6: Crear Productos en CRM

Ahora que las Acciones están definidas en CGRateS, crea los productos correspondientes en el CRM.

Ejemplo: Servicio SIM Independiente

```
product_list_new = []

product_list_new.append({
    "category": "standalone",
    "provisioning_play": "play_psim_only",
    "relies_on_list": "",
    "contract_days": 0,
    "retail_cost": 0,
    "retail_setup_cost": 0,
    "product_slug": "Mobile-SIM",
    "product_name": "Solo SIM Móvil",
    "comment": "Tarjeta SIM física para uso con teléfonos
móviles",
    "provisioning_json_vars": "{\"iccid\" : \"\", \"msisdn\" :
\"\"}",
    "terms": "Debe ser activado dentro de 6 meses. Todo el crédito
se pierde si el servicio no se usa durante 12 meses.",
    "service_type": "mobile",
    "residential": True,
    "business": True,
    "enabled": True,
    "inventory_items_list": "['SIM Card', 'Mobile Number']",
    "icon": "fa-solid fa-sim-card",
    "features_list": "['Número de Teléfono Australiano (61xxx)',
'Las velocidades más rápidas', 'Mejor cobertura']",
    "wholesale_cost": 3,
    "wholesale_setup_cost": 1,
    "customer_can_purchase": True
})
```

Ejemplo: Complemento de Plan Mensual

```
product_list_new.append({
    "category": "addon",
    "provisioning_play": "play_topup_charge_then_action",
    "relies_on_list": "",
    "contract_days": 30,
    "retail_cost": 59.00,
    "retail_setup_cost": 0,
    "product_slug": "au-premium-plan-1",  # Vincula a Action_au-
premium-plan-1
    "product_name": "Plan Premium AU 1",
    "comment": "100GB de Datos en Australia, 3000 Minutos en
Australia, 3000 SMS en Australia, 6GB de Datos en Roaming",
    "provisioning_json_vars": "",
    "terms": "Plan postpago. Se renueva automáticamente cada 30
días.",
    "service_type": "mobile",
    "residential": True,
    "business": True,
    "enabled": True,
    "icon": "fa-solid fa-mobile",
    "features_list": "['100GB de Datos en Australia', '3000
Minutos en Australia', '3000 SMS en Australia', '6GB de Datos en
Roaming']",
    "wholesale_cost": 45.00,
    "wholesale_setup_cost": 0,
    "auto_renew": "True",
    "customer_can_purchase": True
})
```

Ejemplo: Complemento de Recarga Única

```
product_list_new.append({
    "category": "addon",
    "provisioning_play": "play_topup_charge_then_action",
    "relies_on_list": "",
    "contract_days": 0,
    "retail_cost": 30,
    "retail_setup_cost": 0,
    "product_slug": "au-data-addon-20gb",  # Vincula a Action_au-data-addon-20gb
    "product_name": "Complemento de Datos AU 20GB",
    "comment": "20GB de Datos en Australia",
    "provisioning_json_vars": "",
    "terms": "Recarga prepaga. Carga inmediata.",
    "service_type": "mobile",
    "residential": True,
    "business": True,
    "enabled": True,
    "icon": "fa-solid fa-mobile",
    "features_list": "['20GB de Datos en Australia']",
    "wholesale_cost": 22,
    "wholesale_setup_cost": 0,
    "auto_renew": "False",
    "customer_can_purchase": True
})
```

Descripciones de Campos del Producto:

- **category** (requerido) - Categoría del producto:
  - `standalone` - Servicio completo (crea un nuevo registro de servicio)
  - `addon` - Agregado a un servicio existente
  - `bundle` - Paquete de múltiples servicios
- **provisioning_play** (requerido) - Playbook de Ansible a ejecutar (por ejemplo, `play_topup_charge_then_action`, `play_simple_service`)
- **relies_on_list** - Lista separada por comas de product_slugs de los que depende este producto
- **contract_days** - Duración del contrato en días (0 = sin contrato, 30 = mensual)
- **retail_cost** (requerido) - Costo mensual/recurrido para el cliente
- **retail_setup_cost** - Tarifa de configuración única cobrada al cliente

- **product_slug** (requerido) - **Identificador único que vincula a la Acción de CGRateS** (convención: `Action_{product_slug}`)
- **product_name** (requerido) - Nombre para mostrar
- **comment** - Descripción corta
- **provisioning_json_vars** - Cadena JSON de variables pasadas al playbook
- **terms** - Texto de términos y condiciones
- **service_type** - Clasificación del servicio (móvil, dongle, fijo, etc.)
- **residential** - Disponible para clientes residenciales
- **business** - Disponible para clientes comerciales
- **enabled** - El producto está activo y se puede comprar
- **inventory_items_list** - Cadena de lista de Python de artículos de inventario requeridos
- **icon** - Clase de icono de Font Awesome
- **features_list** - Cadena de lista de Python de características
- **wholesale_cost** - Tu costo recurrente
- **wholesale_setup_cost** - Tu costo único
- **auto_renew** - Cadena "True" o "False" para renovación automática
- **customer_can_purchase** - El cliente puede comprar a través del portal de autoservicio

Consulta también: Ciclo de Vida del Producto

Paso 7: Crear o Actualizar Productos a través

de la API

```
# Obtener productos existentes
product_list_existing = session.get(crm_url + '/crm/product',
headers=headers)
existing_products = product_list_existing.json()

# Procesar cada producto
for product in product_list_new:
    print(f"Procesando producto: {product['product_slug']}")

    # Verificar si el producto ya existe
    if product['product_slug'] in [x['product_slug'] for x in
existing_products]:
        print(f"El producto ya existe: {product['product_slug']}")

        # Obtener product_id del producto existente
        product_id = [
            x['product_id']
            for x in existing_products
            if x['product_slug'] == product['product_slug']
        ][0]

        product['product_id'] = product_id

        # Actualizar producto existente
        response = session.patch(
            crm_url + '/crm/product/' + str(product_id),
            json=product,
            headers=headers
        )
        print(f"Estado: {response.status_code}")
        assert response.status_code == 200
        print(f"Producto actualizado: {product['product_slug']}")
    else:
        # Crear nuevo producto
        print(f"Creando nuevo producto:
{product['product_slug']}")
        response = session.put(
            crm_url + '/crm/product',
            json=product,
            headers=headers
        )
```

```
        print(f"Estado: {response.status_code}")
        assert response.status_code == 200
        print(f"Producto creado: {product['product_slug']}")
```

Vinculación de Productos a Acciones

El vínculo crítico entre los productos de CRM y las Acciones de CGRateS es la **convención de nomenclatura**:

```
# En CGRateS
ActionsId = "Action_au-premium-plan-1"

# En CRM
product_slug = "au-premium-plan-1"

# En el playbook (construido automáticamente)
cgr_action_name = "Action_" + product_slug
# Resultado: "Action_au-premium-plan-1"
```

Cómo los Playbooks Ejecutan Acciones

Nota: Usar `ExecuteAction` es solo un patrón para agregar saldos. También podrías llamar directamente a `SetBalance` u otras API de CGRateS dentro de un playbook. Sin embargo, `OmniCRM-API/Provisioners/plays/play_topup_charge_then_action.yaml` utiliza el patrón ExecuteAction ya que es un enfoque común y reutilizable que separa la lógica de saldo (Acciones) de la lógica de aprovisionamiento (Playbooks).

Cuando un producto se aprovisiona utilizando este patrón, el playbook construye el nombre de la Acción a partir del product_slug:

```
# En play_topup_charge_then_action.yaml
- name: Establecer el hecho cgr_action_name
  set_fact:
    cgr_action_name: "Action_{{
api_response_product.json.product_slug }}"

# Ejecutar la acción
- name: Ejecutar acción de CGRateS
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "APIerSv1.ExecuteAction",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "ActionsId": "{{ cgr_action_name }}"
        }]
      }
```

Al usar este patrón: La Acción **debe existir en CGRateS** antes de que se ejecute el playbook, de lo contrario, la ejecución fallará con el error "Acción no encontrada".

Consulta también: Guía de Playbooks de Ansible

Ejemplo Completo: Definición de un Nuevo Producto

Esta sección explica cómo crear un nuevo producto completo "Paquete de Datos de 50GB":

1. Definir la Acción en CGRateS

Consulta el ejemplo de Complemento de Datos Simple en el Paso 5 anterior para ver cómo definir una Acción de complemento de datos. Para nuestro

ejemplo de 50GB, la Acción seguiría el mismo patrón pero con diferentes unidades:

- **ActionsId**: `"Action_50gb-data-pack"` (coincide con product_slug a continuación)
- **BalanceType**: `"*data"`
- **DestinationIDs**: `"Dest_PLMN_OnNet"` (para uso de datos en red)
- **Units**: `50 * 1024 * 1024 * 1024` (50GB en bytes)
- **Identifier**: `"*topup"` (habilita rollover)

2. Crear el Producto en CRM

```
product_50gb_data = {
    "category": "addon",
    "provisioning_play": "play_topup_charge_then_action",
    "relies_on_list": "",
    "contract_days": 0,
    "retail_cost": 25.00,
    "retail_setup_cost": 0,
    "product_slug": "50gb-data-pack",  # Debe coincidir con
ActionsId sin "Action_"
    "product_name": "Paquete de Datos de 50GB",
    "comment": "50GB de datos de alta velocidad válidos por 30
días",
    "provisioning_json_vars": "",
    "terms": "Los datos expiran después de 30 días. Los datos no
utilizados se acumulan si se recarga nuevamente antes de la
expiración.",
    "service_type": "mobile",
    "residential": True,
    "business": True,
    "enabled": True,
    "icon": "fa-solid fa-database",
    "features_list": "['50GB de Datos de Alta Velocidad', 'Validez
de 30 Días', 'Rollover Habilitado']",
    "wholesale_cost": 20.00,
    "wholesale_setup_cost": 0,
    "auto_renew": "False",
    "customer_can_purchase": True
}

# Crear en CRM
response = session.put(
    crm_url + '/crm/product',
    json=product_50gb_data,
    headers=headers
)
assert response.status_code == 200
print("Producto creado: 50gb-data-pack")
```

3. Verificar el Vínculo

El producto ahora se puede aprovisionar:

1. El cliente compra "Paquete de Datos de 50GB" (product_slug: `50gb-data-pack`)
2. El playbook `play_topup_charge_then_action` se ejecuta
3. El playbook construye el nombre de la Acción: `Action_50gb-data-pack`
4. El playbook llama a `APIerSv1.ExecuteAction` con ActionsId `Action_50gb-data-pack`
5. CGRateS encuentra la Acción y la ejecuta
6. El cliente recibe un saldo de 50GB

Consulta también: Cargos y Pagos desde Playbooks

Patrones Comunes de Productos

Estos patrones muestran configuraciones típicas de productos en CRM y cómo se vinculan a las Acciones de CGRateS. Para las definiciones de Acción correspondientes, consulta Acciones de CGRateS y Comportamientos de Recarga.

Patrón 1: Plan Mensual de Múltiples Saldos

Caso de Uso: Plan mensual integral con datos, voz, SMS, roaming

Producto CRM:

```
{
    "product_slug": "premium-monthly-plan",  # Vincula a
Action_premium-monthly-plan
    "product_name": "Plan Mensual Premium",
    "category": "addon",
    "retail_cost": 59.00,
    "auto_renew": "True",
    "contract_days": 30,
    "provisioning_play": "play_topup_charge_then_action",
    "features_list": "['100GB de Datos', '5000 Minutos', '5000
SMS', 'Roaming Incluido']"
}
```

Acción: Usa `*topup_reset` con múltiples tipos de saldo. Consulta Ejemplo 1: Plan Mensual de Múltiples Saldos (Python).

Patrón 2: Complemento de Datos Simple (Rollover)

Caso de Uso: Compra de datos única que se acumula si se recarga nuevamente

Producto CRM:

```
{
    "product_slug": "10gb-addon",  # Vincula a Action_10gb-addon
    "product_name": "Complemento de Datos de 10GB",
    "category": "addon",
    "retail_cost": 15.00,
    "auto_renew": "False",  # Compra única
    "contract_days": 0,
    "provisioning_play": "play_topup_charge_then_action",
    "features_list": "['10GB de Datos con Rollover', 'Validez de
30 Días']"
}
```

Acción: Usa `*topup` para comportamiento de rollover. Consulta Acciones de CGRateS - Rollover.

Patrón 3: Plan Mensual Fijo (Sin Rollover)

Caso de Uso: Plan mensual que siempre se restablece a la misma cantidad

Producto CRM:

```
{
    "product_slug": "fixed-50gb-monthly",  # Vincula a
Action_fixed-50gb-monthly
    "product_name": "Mensual Fijo de 50GB",
    "category": "addon",
    "retail_cost": 35.00,
    "auto_renew": "True",
    "contract_days": 30,
    "provisioning_play": "play_topup_charge_then_action",
    "features_list": "['50GB de Datos', 'Se Restablece
Mensualmente', 'Sin Rollover']"
}
```

Acción: Usa `*topup_reset` para prevenir rollover. Consulta Acciones de CGRateS - Comportamiento de Restablecimiento.

Avanzado: Cargadores y Planes de Calificación

Importante: Esta sección cubre la calificación básica de pago por uso. CGRateS admite escenarios de calificación increíblemente sofisticados, incluyendo:

- **Tarifas de roaming diferenciales**: Precios diferentes para llamadas/SMS de MO vs MT cuando se está en roaming
- **Carga específica de protocolos**: CAMEL para 2G/3G (SMS MO-MT, USSD), Diameter para 4G/5G
- **Precios contextuales**: Tarifas basadas en el tiempo, ubicación, segmento de cliente, calidad de red
- **Calificación multidimensional**: Combinando tipo de red, estado de roaming, destino y hora del día

Los ejemplos a continuación muestran configuraciones PAYG simples y de tarifa plana adecuadas para la mayoría de las implementaciones de OmniCRM.

Para la facturación de pago por uso (donde el uso más allá de los saldos incluidos cuesta dinero), necesitas definir Planes de Calificación que especifiquen montos en dólares a cobrar.

Cómo Funciona la Calificación

Cuando un cliente utiliza un servicio (datos, voz, SMS):

1. CGRateS verifica los saldos unitarios (por ejemplo, GB de datos incluidos)
2. Si existe saldo unitario, se deduce el uso de este
3. Si no hay saldo unitario o el saldo se agota, CGRateS recurre al saldo monetario
4. **El Plan de Calificación** determina el precio (por ejemplo, $0.10 por MB)
5. El costo calculado se deduce del saldo monetario del cliente

Ejemplo: Configuración de Cargos de Datos de

Pago por Uso

```
# 1. Definir una Tarifa de Destino (el precio)
OCS_Obj.SendData({
    "method": "ApierV2.SetTPDestinationRate",
    "params": [{
        "TPid": tpid,
        "ID": "DR_DATA_PAYG",
        "DestinationRates": [{
            "DestinationId": "Dest_PLMN_OnNet",
            "RateId": "RT_DATA_$0_10_PER_MB",
            "RoundingMethod": "*up",
            "RoundingDecimals": 4,
            "MaxCost": 0,
            "MaxCostStrategy": ""
        }]
    }]
})

# 2. Definir la tarifa real
OCS_Obj.SendData({
    "method": "ApierV2.SetTPRate",
    "params": [{
        "TPid": tpid,
        "ID": "RT_DATA_$0_10_PER_MB",
        "RateSlots": [{
            "ConnectFee": 0,          # Sin cargo de conexión
            "Rate": 0.10,             # $0.10 por unidad
            "RateUnit": "1048576",    # 1 MB (1024 * 1024 bytes)
            "RateIncrement": "1048576", # Facturar por MB
            "GroupIntervalStart": "0s"
        }]
    }]
})

# 3. Definir un Plan de Calificación que use esta tarifa
OCS_Obj.SendData({
    "method": "ApierV2.SetTPRatingPlan",
    "params": [{
        "TPid": tpid,
        "ID": "RP_PAYG_DATA",
        "RatingPlanBindings": [{
            "DestinationRatesId": "DR_DATA_PAYG",
```

```
            "TimingId": "*any",
            "Weight": 10
        }]
    }]
})

# 4. Vincular el Plan de Calificación a un Perfil de Calificación (para cuentas específicas)
OCS_Obj.SendData({
    "method": "ApierV2.SetTPRatingProfile",
    "params": [{
        "TPid": tpid,
        "Tenant": tenant,
        "Category": "data",
        "Subject": "*any",  # O identificador de cuenta específico
        "RatingPlanActivations": [{
            "ActivationTime": "2024-01-01T00:00:00Z",
            "RatingPlanId": "RP_PAYG_DATA",
            "FallbackSubjects": ""
        }]
    }]
})

# 5. Cargar el plan tarifario
OCS_Obj.SendData({
    "method": "APIerSv1.LoadTariffPlanFromStorDb",
    "params": [{
        "TPid": tpid,
        "DryRun": False,
        "Validate": True,
        "APIOpts": {}
    }]
})
```

Lo que esto hace:

- Cuando un cliente no tiene saldo de datos (o se queda sin él), el uso de datos cuesta **$0.10 por MB**
- El cargo se deduce de su saldo monetario (`*monetary`)
- Los incrementos de facturación son de 1 MB (redondea hacia arriba)

Ejemplo: Tarifas de Llamadas de Voz

```
# Definir diferentes tarifas para diferentes destinos
OCS_Obj.SendData({
    "method": "ApierV2.SetTPRate",
    "params": [{
        "TPid": tpid,
        "ID": "RT_VOICE_DOMESTIC_$0_30_PER_MIN",
        "RateSlots": [{
            "ConnectFee": 0.15,           # $0.15 de cargo de
conexión
            "Rate": 0.30,                 # $0.30 por minuto
            "RateUnit": "60s",            # Facturar por minuto
            "RateIncrement": "60s",       # Redondear a minutos
completos
            "GroupIntervalStart": "0s"
        }]
    }]
})

OCS_Obj.SendData({
    "method": "ApierV2.SetTPRate",
    "params": [{
        "TPid": tpid,
        "ID": "RT_VOICE_INTL_$1_50_PER_MIN",
        "RateSlots": [{
            "ConnectFee": 0.25,           # $0.25 de cargo de
conexión
            "Rate": 1.50,                 # $1.50 por minuto
            "RateUnit": "60s",
            "RateIncrement": "1s",        # Facturar por segundo
(más preciso)
            "GroupIntervalStart": "0s"
        }]
    }]
})
```

Perfiles de Cargadores (Opcional)

Los cargadores aplican la calificación a tipos específicos de eventos. Para la mayoría de los casos de uso, el cargador predeterminado es suficiente:

```
OCS_Obj.SendData({
    "method": "APIerSv1.SetChargerProfile",
    "params": [{
        "Tenant": tenant,
        "ID": "Charger_Default",
        "FilterIDs": [],
        "AttributeIDs": ["*constant:*req.RequestType:*none"],
        "Weight": 999
    }]
})
```

Consulta también:

- Acciones de CGRateS y Comportamientos de Recarga - Enfoques de gestión de saldos (unitario, monetario, híbrido) y configuración de calificación
- Destinos de CGRateS - Cómo definir destinos geográficos y PLMN

Mejores Prácticas

1. Usar Convenciones de Nomenclatura Consistentes

```
# Bueno - relación clara
ActionsId = "Action_premium-plan-100gb"
product_slug = "premium-plan-100gb"

# Malo - sin relación clara
ActionsId = "Action_Plan_A"
product_slug = "premium_100"
```

2. Definir Acciones Antes de Productos

Siempre crea Acciones de CGRateS antes de crear productos de CRM. El playbook de aprovisionamiento fallará si intenta ejecutar una Acción inexistente.

3. Incluir Registro CDR

Siempre agrega la acción `*cdrlog` para rastrear las adiciones de saldo:

```
{
    "Identifier": "*cdrlog",
    "BalanceType": "*generic",
    "ExtraParameters": "
{\"Category\":\"^activation\",\"Destination\":\"Nombre del
Producto\"}",
    "Weight": 80
}
```

4. Usar IDs de Saldo Descriptivos

```
# Bueno
"BalanceId": "Premium_Data_100GB"
"BalanceId": "AU_Voice_Domestic__" + str(units)

# Malo
"BalanceId": "data1"
"BalanceId": "balance"
```

5. Documentar Unidades Claramente

```
# Bueno - muestra el cálculo
Units = 100 * 1024 * 1024 * 1024  # 100GB en bytes
Units = 3000 * 60 * 1000000000    # 3000 minutos en nanosegundos

# Malo - número mágico
Units = 107374182400
```

6. Usar Peso Estratégicamente

Mayor peso = mayor prioridad tanto para la ejecución como para el consumo:

```
# Orden de ejecución (mayor peso = se ejecuta primero)
{"Identifier": "*reset_account", "Weight": 700}
{"Identifier": "*topup_reset", "Weight": 90}
{"Identifier": "*cdrlog", "Weight": 80}

# Consumo de saldo (mayor peso = se consume primero)
"BalanceWeight": 1200  # Saldos premium/nacionales
"BalanceWeight": 1100  # Saldos de roaming
"BalanceWeight": 1000  # Saldos internacionales
```

7. Probar Primero con Valores Pequeños

Al definir nuevos productos, prueba primero con saldos pequeños:

```
# Pruebas
Units = 100 * 1024 * 1024  # 100MB para pruebas

# Producción
Units = 100 * 1024 * 1024 * 1024  # 100GB
```

Solución de Problemas

Error: "Acción no encontrada"

Síntoma: El playbook falla con "Acción no encontrada" o "ActionsId no existe"

Causa: Acción no definida en CGRateS, o desajuste de nombres

Solución:

1. Verifica que la Acción exista: Consulta CGRateS por la Acción
2. Verifica la nomenclatura: Asegúrate de que `ActionsId = "Action_" + product_slug`
3. Define la Acción si falta

Error: "Destino no encontrado"

Síntoma: La Acción se ejecuta pero no se crean saldos

Causa: DestinationIDs hace referencia a un destino inexistente

Solución:

1. Define el destino en CGRateS primero
2. Usa `*any` para destinos universales
3. Verifica la ortografía de los IDs de destino

Producto Creado Pero No se Agregan Saldos

Síntoma: El producto se aprovisiona con éxito pero el cliente no tiene saldo

Causa: La Acción existe pero no tiene operaciones para agregar saldo

Solución:

1. Verifica que la Acción tenga operaciones `*topup` o `*topup_reset`
2. Verifica que el valor de Units sea correcto (no 0)
3. Verifica que ExpiryTime no haya pasado

Saldos No se Acumulan

Síntoma: Usando rollover pero el saldo no utilizado desaparece

Causa: Usando `*topup_reset` en lugar de `*topup`

Solución:

- Cambia `Identifier` de `*topup_reset` a `*topup`
- Asegúrate de que BalanceId sea consistente en todas las recargas

Consulta también: Acciones de CGRateS y Comportamientos de Recarga

Resumen

Esta guía demuestra un enfoque común para configurar productos en OmniCRM con CGRateS. Los patrones mostrados aquí funcionan bien para casos de uso típicos de operadores de red móvil virtual (MVNO) con planes prepagos/postpagos, paquetes de datos y roaming.

Recuerda: CGRateS es capaz de escenarios mucho más sofisticados de los cubiertos aquí. Si necesitas características avanzadas como:

- Precios diferenciales de MO/MT para voz y SMS en roaming
- Carga basada en CAMEL para dispositivos heredados 2G/3G
- Variaciones de tarifas según la hora del día o estacionales
- Precios basados en niveles de clientes (bronce/plata/oro)
- Cargos basados en calidad de red o QoS
- Enrutamiento de menor costo con múltiples proveedores

...consulta la documentación de CGRateS directamente y trabaja con la configuración de tu plan de calificación independientemente de las definiciones de productos de CRM.

Documentación Relacionada

- Acciones de CGRateS y Comportamientos de Recarga - Guía detallada sobre tipos de Acción y comportamiento de saldo
- Guía de Playbooks de Ansible - Cómo los playbooks ejecutan Acciones
- Cargos y Pagos desde Playbooks - Procesamiento de pagos durante el aprovisionamiento
- Ciclo de Vida del Producto - Gestión de productos a través de su ciclo de vida
- Conceptos: Productos y Servicios - Fundamentos del producto
- Documentación Oficial de CGRateS - Referencia completa de CGRateS

Campañas de Email y SMS (EDM)

El módulo de campañas masivas de OmniCRM (EDM: Correo Directo Electrónico) permite a los operadores construir segmentos de audiencia, diseñar plantillas y enviar campañas masivas a los clientes a través de **dos canales: email (vía SMTP) y SMS (vía OmniMessage)**. Una campaña tiene como objetivo un segmento con una plantilla elegida en un canal seleccionado, con personalización por destinatario, manejo de exclusiones y seguimiento. Esta guía cubre ambos canales.

Véase también: Mailjet para la integración transaccional de Mailjet, API de Notificaciones para notificaciones del sistema, Configuración y Personalización para la configuración general.

> **Nota de alcance:** Esto describe solo la ruta de envío de **campañas masivas** (email y SMS). El correo electrónico transaccional (bienvenida, facturas, restablecimientos de contraseña, entrega de eSIM, actualizaciones de tickets) se entrega por separado a través de la API REST de Mailjet y no se ve afectado por la configuración a continuación.

Descripción general

El módulo tiene estos bloques de construcción:

Concepto	Descripción
Segmento	Una audiencia objetivo, definida por criterios de filtro (tipo de facturación, tipo de contacto, estado, saldo, …) o una lista de direcciones estáticas.
Plantilla	Contenido reutilizable para un canal: un **cuerpo HTML subido por el operador** (email) o un **mensaje SMS corto**, ambos con personalización por destinatario.
Campaña	Un envío: una plantilla a un segmento en un **canal** elegido, con seguimiento por destinatario.
Canal	Cómo se entrega una campaña: **email** (reenvío SMTP) o **SMS** (OmniMessage).

Canales

El `canal` de una campaña decide todo lo que sigue: transporte de entrega, formato de plantilla y qué exclusión se aplica:

Canal	Entregado a través de	Personalización	Exclusión	Detalles
Email	Reenvío SMTP (Mailjet, Mailgun, SES, …)	Plantillas HTML Jinja2	preferencias de categoría `email` + darse de baja + lista de supresión	Entrega de Email
SMS	OmniMessage	sustitución de `var1`–`var4`	preferencias de categoría `sms` + OmniMessage STOP	Entrega de SMS (OmniMessage

Plantillas de email y personalización

Las plantillas almacenan HTML en bruto que **subes o pegas** en el editor de plantillas. La personalización utiliza la sintaxis de **Jinja2**, por lo que las plantillas pueden contener variables y lógica simple:

- `{{ first_name }}`, `{{ last_name }}`, `{{ customer_name }}`, `{{ email }}`
- condicionales y bucles, por ejemplo, `{% if first_name %}Hola {{ first_name }}{% else %}Hola{% endif %}`

Cualquier campo adicional proporcionado por el segmento para un destinatario también está disponible para la plantilla por nombre. El email de cada destinatario se renderiza individualmente antes de enviarlo.

Uso de campañas masivas

Este recorrido cubre la construcción de un segmento, la creación de una campaña de email o SMS, su envío y seguimiento, además de lo que ven los destinatarios. Todo se encuentra en la sección de **Marketing** de la barra lateral (**Segmentos**, **Plantillas**, **Campañas**). (Orientado al operador; escrito desde la implementación actual.)

Construir un segmento

Un **segmento** es la audiencia a la que se dirige una campaña. Desde **Marketing → Segmentos** puedes buscar, actualizar y crear segmentos. Cada fila muestra los criterios del segmento, el conteo actual de destinatarios y cuándo fue actualizado por última vez; haz clic en una fila para abrirla.

La lista de Segmentos. Los conteos de destinatarios se actualizan a demanda; hacer clic en una fila abre el segmento.

Crear Segmento (o hacer clic en una fila para editar) abre el editor de segmentos, donde defines la audiencia a partir de atributos de cliente: *Clientes sin servicios*, *Clientes con saldo cero*, *Tipos de Facturación*, *Tipos de Contacto* y *Estado del Cliente*. Guarda primero el segmento. **Generar Segmento** luego calcula la lista de destinatarios, y **Exportar CSV** la descarga.

El editor de segmentos: criterios a la izquierda, un conteo de destinatarios en vivo y exportación CSV en la parte inferior.

> Los segmentos son agnósticos al canal: el mismo segmento puede ser utilizado para una campaña de email o SMS. Las campañas de email se dirigen a contactos con una dirección de email. Las campañas de SMS se dirigen a contactos con un número de teléfono.

Subir una plantilla

Las campañas de email envían una plantilla HTML. Desde **Marketing → Plantillas**, crea o edita una plantilla y usa **Subir .html** para cargar un archivo HTML directamente en el cuerpo (o pégalo). Las plantillas soportan personalización Jinja2, y el panel de **Variables Disponibles** inserta `{{customer_name}}`, `{{contact_firstname}}`, `{{contact_lastname}}` y `{{contact_email}}` en el cursor.

Las campañas de SMS no utilizan una plantilla. Escribes el mensaje en línea cuando compones la campaña.

Crear una campaña

Desde **Marketing → Campañas → Crear Campaña**, un breve asistente te guía a través de la campaña. El interruptor de **Canal** (Email / SMS) en la parte superior del asistente determina qué pasos y campos aparecen.

1. **Seleccionar Segmento**: elige la audiencia. El segmento seleccionado muestra su conteo de destinatarios.

Paso 1: elige el segmento cuyos contactos recibirán la campaña. El interruptor de canal se encuentra por encima de los pasos.

2. **Plantilla / Mensaje**: para una campaña de **email**, elige una plantilla HTML (haz clic en una tarjeta de plantilla para seleccionarla; **Vista previa** muestra el HTML renderizado). Para una campaña de **SMS**, el asistente omite este paso. Muestra un aviso, y escribes el mensaje en el siguiente paso.

Para SMS, un aviso reemplaza el paso de plantilla. Continúas escribiendo el mensaje.

3. **Detalles de la Campaña**: la configuración específica del canal:
    - **Categoría**: la categoría de comunicación a la que pertenece la campaña (cargada desde `edm.communication_categories`). El sistema excluye a los destinatarios que se dieron de baja de esta categoría para este canal.
    - **Email**: *Nombre de la Campaña*, *Asunto*, *Email del Remitente*, *Nombre del Remitente*.
    - **SMS**: *Nombre de la Campaña*, *ID del Remitente* y el *Mensaje SMS* (personaliza con `{{ first_name }}`, `{{ last_name }}`, `{{ customer_name }}`).

Detalles de la campaña de email: campos de asunto y remitente, además del selector de categoría.

Detalles de la campaña de SMS: el formulario cambia a ID del remitente y un cuerpo de Mensaje SMS.

4. **Revisar y Guardar**: confirma canal, categoría, segmento y contenido, luego guarda como un **borrador**. **Ver destinatarios** lista los contactos del segmento (nombre, email o teléfono, cliente) antes de que te comprometas.

Revisión para una campaña de email.

*Revisión para una campaña de SMS, con el enlace **Ver destinatarios** y la nota de guardar como borrador.*

La campaña siempre se guarda primero como un **borrador**. La envías desde la página de detalles de la campaña (a continuación).

Lista de campañas

Marketing → Campañas lista todas las campañas con su **Canal** (Email / SMS), estado, conteo de destinatarios y, una vez enviadas, tasas de apertura y clics. Filtra por **Todas / Borrador / Enviadas**, y haz clic en cualquier fila para abrir la campaña.

La lista de Campañas. La columna de Canal distingue email de SMS de un vistazo. Las campañas enviadas muestran sus tasas de apertura y clics en línea.

Enviar y rastrear una campaña

Haz clic en una campaña para abrir su página de detalles:

- Un **borrador** muestra un botón de **Enviar Campaña**. El botón se habilita una vez que la campaña tiene un segmento y, para email, una plantilla, o para SMS, un mensaje. Si falta algo, la página lista lo que se requiere.
- Una vez **enviada**, la página de detalles muestra:
  - **Estadísticas de la Campaña**: para *email*, entregado, tasa de apertura, tasa de clics, abiertos/clicados/rebotados. Para *SMS*, entregado / fallido, con un botón de **Sincronizar** que obtiene los últimos conteos de entrega de OmniMessage.
  - **Destinatarios**: una lista por destinatario con estado (`enviado`, `entregado`, `abierto`, `clicado`, `rebotado`, `fallido`). La columna de dirección muestra la dirección de email (email) o número de teléfono (SMS).

*La página de detalles de la campaña para una campaña de SMS en borrador, lista para enviar. Una vez enviada, esta página muestra estadísticas de entrega/compromiso y estado por destinatario; para SMS, **Sincronizar** actualiza los conteos de OmniMessage.*

Una vez enviada, la misma página muestra el panel de **Estadísticas de la Campaña** y la lista por destinatario:

Una campaña de email enviada: conteo de entregas, tasa de apertura/clic y totales de abiertos/clicados/rebotados, junto a la lista por destinatario y el estado de cada destinatario.

> Debido a que el seguimiento es agnóstico al proveedor, **Entregado** se deriva del compromiso. Una apertura o clic prueba que el mensaje llegó. La tasa de apertura se mide, por lo tanto, contra los destinatarios entregados (comprometidos) en lugar del total enviado.

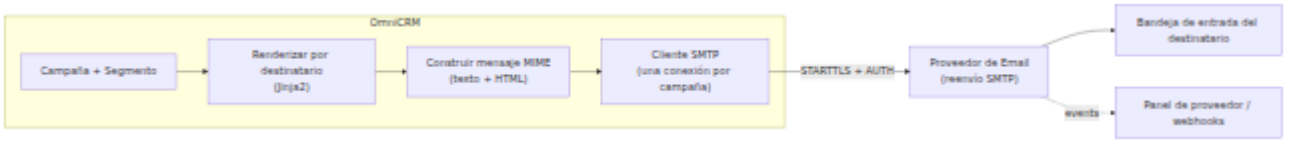
Lo que ven los destinatarios (darse de baja)

Cada email lleva un enlace de darse de baja que abre un **centro de preferencias**: las categorías de email del contacto con su estado actual de opt-in, una acción de **Guardar preferencias**, y una opción de **Darse de baja de todos los emails**. La opción de darse de baja de un cliente de correo con un solo clic excluye al contacto solo de la categoría de esa campaña.

> Los archivos de imagen anteriores se encuentran en `docs/images/` (nombres de archivo con prefijo `edm-`).

Entrega de Email

El email de la campaña se entrega a través de **SMTP** a un reenvío configurado. Esto hace que la ruta de envío sea independiente del proveedor: la misma configuración de OmniCRM puede apuntar a Mailjet, Mailgun, Amazon SES, o al propio servidor SMTP de un cliente.



Para cada destinatario, OmniCRM:

1. Renderiza el HTML y las partes de texto de la plantilla, y el asunto, con las variables del destinatario.
2. Construye un mensaje MIME `multipart/alternative` (fallback de texto plano + HTML).
3. Lo envía a través de una única conexión SMTP reutilizada para toda la campaña.
4. Registra un resultado por destinatario (`enviado` cuando el reenvío acepta el mensaje, `fallido` en caso de rechazo).

Cada mensaje también lleva un encabezado de correlación, `X-MJ-CustomID`, en la forma `campaign_{campaign_id}_recipient_{recipient_id}`. Mailjet lo devuelve en sus webhooks de eventos; otros reenvíos simplemente ignoran el encabezado desconocido.

Configuración del Reenvío SMTP

El reenvío se configura bajo el bloque `mailjet.smtp` en `crm_config.yaml`. A pesar del nombre de la sección, este es el reenvío SMTP genérico utilizado para todas las campañas EDM y puede apuntar a cualquier proveedor.

```yaml
mailjet:
  api_key: "your-api-key"          # utilizado por email
transaccional + como el nombre de usuario SMTP por defecto
  api_secret: "your-api-secret"    # utilizado por email
transaccional + como la contraseña SMTP por defecto
  smtp:
    host: in-v3.mailjet.com        # nombre del host del reenvío
SMTP
    port: 587                      # puerto de envío
    use_tls: true                  # actualizar la conexión con
STARTTLS
    use_auth: true                 # autenticar con nombre de
usuario/contraseña
    username: ''                   # en blanco -> usar mailjet
api_key
    password: ''                   # en blanco -> usar mailjet
api_secret
    timeout: 30                    # tiempo de espera de
conexión/comando en segundos
```

Parámetros

Parámetro	Tipo	Requerido	Por defecto	Descripción
`host`	Cadena	Sí	`in-v3.mailjet.com`	Nombre del host reenvío SMTP.
`port`	Entero	No	`587`	Puerto de envío. `587` para STARTT `465` para TLS implícito; muchos proveedores también ofrecen `2525`/`25`.
`use_tls`	Booleano	No	`true`	Cuando es `true`, conexión se actualiza con STARTTLS despué de conectarse. Establecer `false` solo para un servidor de prueb local.
`use_auth`	Booleano	No	`true`	Cuando es `true`, OmniCRM se autentica con `username`/`passwo` Establecer `false` solo para un servidor de prueb no autenticado.
`username`	Cadena	No	*(vacío)*	Nombre de usuar SMTP. **Cuando es en blanco,**

Parámetro	Tipo	Requerido	Por defecto	Descripción
				OmniCRM recur a la `api_key` de Mailjet. Para cualquier reenvío que no sea Mailje debes establecer esto explícitamer
`password`	Cadena	No	*(vacío)*	Contraseña SMTP **Cuando está en blanco, OmniCR recurre a la `api_secret` de Mailjet.** Para cualquier reenvío que no sea Mailje debes establecer esto explícitamer
`timeout`	Entero	No	`30`	Tiempo de espera de conexión y comando, en segundos.

> **Importante:** `username`/`password` se establecen por defecto en las credenciales de la API de Mailjet únicamente por conveniencia. Cuando apuntas a un proveedor diferente, **siempre establece ambos explícitamente**. De lo contrario, OmniCRM intentará autenticarse con ese proveedor utilizando las credenciales de Mailjet y fallará.

Referencia de Configuración del Proveedor

Las tablas a continuación proporcionan la configuración de conexión SMTP y el mecanismo de seguimiento nativo para proveedores populares. Los nombres de host, puertos y convenciones de credenciales pueden cambiar o variar según la región/plan. Confírmalo contra la documentación actual de tu proveedor. Las notas de **Seguimiento** son *informativas*: describen lo que cada reenvío ofrece en su propio panel si lo usas. OmniCRM solo envía SMTP y no detecta automáticamente ni muestra estadísticas nativas del proveedor.

Mailjet

Configuración	Valor
Host	`in-v3.mailjet.com`
Puertos	`587` (STARTTLS), `465` (TLS), `25` / `2525`
Nombre de usuario	Clave API (Cuenta → Gestión de Claves API)
Contraseña	Clave Secreta

Seguimiento: El seguimiento de aperturas y clics es aplicado por Mailjet (por defecto de cuenta, o por mensaje a través de los encabezados `X-Mailjet-TrackOpen` / `X-Mailjet-TrackClick`). Los eventos de compromiso están disponibles en el **panel de Mailjet**, a través de la **API de Estadísticas**, y como **webhooks de la API de Eventos**. OmniCRM ya incluye un manejador de webhook de Mailjet; correlaciona eventos con destinatarios utilizando el valor `X-MJ-CustomID` que OmniCRM envía.

Mailgun

Configuración	Valor
Host	`smtp.mailgun.org` (EE. UU.) · `smtp.eu.mailgun.org` (UE)
Puertos	`587` / `465` / `25` / `2525`
Nombre de usuario	`postmaster@tu-dominio-de-envío` (o un inicio de sesión SMTP personalizado creado en el dominio)
Contraseña	La contraseña de credenciales SMTP para ese dominio

Seguimiento: Habilita el seguimiento de aperturas/clics por dominio, o por mensaje a través de `X-Mailgun-Track`, `X-Mailgun-Track-Opens`, `X-Mailgun-Track-Clicks`. Adjunta una etiqueta de campaña con `X-Mailgun-Tag` y datos de correlación personalizados con `X-Mailgun-Variables` (JSON). Las estadísticas aparecen en el **panel de Mailgun** y **API de Eventos/Estadísticas**; el compromiso puede ser enviado a OmniCRM a través de **webhooks de Mailgun**.

Amazon SES

Configuración	Valor
Host	`email-smtp.<región>.amazonaws.com` (por ejemplo, `email-smtp.us-east-1.amazonaws.com`)
Puertos	`587` / `2587` (STARTTLS), `465` / `2465` (TLS), `25`
Nombre de usuario	Nombre de usuario **SMTP** de SES (generado en la consola de SES)
Contraseña	Contraseña **SMTP** de SES (distinta de tus claves de acceso de AWS)

Seguimiento: El seguimiento de aperturas y clics se habilita a través de un **Conjunto de Configuración**, adjunto por mensaje con el encabezado `X-SES-CONFIGURATION-SET`; agrega dimensiones de campaña con `X-SES-MESSAGE-TAGS`. Los eventos (envío, entrega, apertura, clic, rebote, queja) se entregan a un destino de eventos (SNS, Kinesis Firehose o EventBridge) y se muestran en **CloudWatch**. No hay un panel de compromiso incorporado, así que planifica un destino de eventos si deseas estadísticas.

SendGrid (Twilio)

Configuración	Valor
Host	`smtp.sendgrid.net`
Puertos	`587` / `465` / `25` / `2525`
Nombre de usuario	la cadena literal `apikey`
Contraseña	una clave API de SendGrid con permiso de **Enviar Email**

Seguimiento: El seguimiento de aperturas/clics se establece en **Configuraciones de Seguimiento** (cuenta) o por mensaje a través del encabezado `X-SMTPAPI` (JSON `filters` como `opentrack`/`clicktrack`, además de `category` y `unique_args` para correlación). Las estadísticas están en el **panel de SendGrid** y **API de Estadísticas**; el compromiso llega a OmniCRM a través del **Webhook de Eventos**.

Postmark

Configuración	Valor
Host	`smtp.postmarkapp.com`
Puertos	`587` / `2525` / `25`
Nombre de usuario	tu **Token API del Servidor**
Contraseña	tu **Token API del Servidor** (mismo valor que el nombre de usuario)

Seguimiento: Habilita aperturas con `X-PM-Track-Opens: true` y enlaces con `X-PM-Track-Links: HtmlAndText`. Etiqueta campañas con `X-PM-Tag` y añade correlación con los encabezados `X-PM-Metadata-<key>`. Usa un **flujo de mensajes de difusión** para el envío masivo/EDM. Las estadísticas están en el **panel de Postmark** y **API de Estadísticas**; el compromiso llega a OmniCRM a través de **webhooks**.

Mailchimp Transactional (Mandrill)

Configuración	Valor
Host	`smtp.mandrillapp.com`
Puertos	`587` / `465` / `25`
Nombre de usuario	tu nombre de usuario de Mailchimp/Mandrill (generalmente el email de la cuenta)
Contraseña	cualquier clave API válida de Mandrill

Seguimiento: Por defecto de cuenta, o por mensaje a través de `X-MC-Track` (por ejemplo, `opens,clicks_all`). Etiqueta con `X-MC-Tags` y añade correlación

con `X-MC-Metadata` (JSON). Las estadísticas están en el **panel de Mandrill** y API; el compromiso llega a OmniCRM a través de **webhooks**.

> **Nota:** El producto estándar **Mailchimp Marketing** *no* es un reenvío SMTP. Solo **Mailchimp Transactional (Mandrill)** ofrece envío SMTP.

Brevo (anteriormente Sendinblue)

Configuración	Valor
Host	`smtp-relay.brevo.com`
Puertos	`587` / `465` / `25` / `2525`
Nombre de usuario	tu inicio de sesión de cuenta de Brevo (email)
Contraseña	tu **clave SMTP** (Brevo → SMTP y API → SMTP)

Seguimiento: Seguimiento de aperturas/clics a nivel de cuenta; estadísticas en el **panel de Brevo**; el compromiso llega a OmniCRM a través de **webhooks**.

SMTP genérico / autoalojado

Configuración	Valor
Host	Según lo proporcionado (por ejemplo, un reenvío Postfix, o un reenvío de Microsoft 365 / Google Workspace del cliente)
Puertos	Típicamente `587` (STARTTLS) o `465` (TLS)
Nombre de usuario / Contraseña	Según lo proporcionado por el operador

Seguimiento: Un servidor SMTP simple **no proporciona seguimiento de compromiso nativo**. Usa el seguimiento nativo de OmniCRM (a continuación) para aperturas y clics, y el procesamiento de rebotes (DSN) para fallos de entrega.

Entrega de SMS (OmniMessage)

Las campañas masivas también pueden ser **SMS**, entregados a través del servicio separado de OmniMessage. El `canal` de una campaña es `email` (por defecto) o `sms`; para SMS, establece `channel: sms` y un `sms_message`.

Destinatarios son los contactos del segmento que tienen un **número de teléfono**, filtrados por la categoría de la campaña en el canal **`sms`**.

Envío sigue el flujo de campañas masivas de OmniMessage (OmniCRM realiza los tres pasos):

1. `POST /api/campaign_lists`: subir destinatarios como una lista nombrada (`msisdn` + `var1`–`var4`).
2. `POST /api/campaigns`: crear la campaña (`message`, `list_id`, `source_msisdn`, `source_smsc`, `drip_tps`).
3. `POST /api/campaigns/control` `{action: "start"}`: iniciar el goteo.

Personalización: Las plantillas de OmniMessage solo soportan `{{ var1 }}`–`{{ var4 }}` (más `{{ msisdn }}`). Escribe los marcadores de posición nombrados familiares y OmniCRM los mapea automáticamente: `first_name`→`var1`, `last_name`→`var2`, `customer_name`→`var3`, `email`→`var4`.

Estado: OmniMessage **no tiene webhook de entrega**. Consulta `GET /api/campaigns/:id` para sus `stats`. `POST /crm/email-campaign/campaigns/<id>/sms-sync` obtiene las últimas estadísticas en la campaña (`total_delivered` / `total_bounced` / `total_sent`). Ejecútalo en un horario para obtener números en vivo.

Configuración (`smsc` bloque en `crm_config.yaml`):

```
smsc:
  campaigns_url: 'https://smsc:8443/api'   # base de la API de
campañas masivas
  source_msisdn: 'YourBrand'                 # remitente por
defecto
  source_smsc: 'IMS_SMSC'                    # ruta SMSC
  drip_tps: 50                               # tasa de envío por
campaña
  verify_tls: false                          # OmniMessage utiliza
un certificado autofirmado
  timeout: 30
```

> OmniMessage mantiene su propia supresión STOP/exclusión; OmniCRM además aplica las preferencias de categoría `sms` del contacto al construir la lista de destinatarios.

Estadísticas y Seguimiento

Hay dos formas independientes de medir una campaña. No son mutuamente excluyentes.





Lo que cada modelo te ofrece

Métrica	Registrada al enviar (cada reenvío)	En la consola propia de tu proveedor [1]	Seguimiento nativo de OmniCRM (cualquier reenvío)
Enviado / en cola	✓	✓	✓
Fallos de envío (rechazados en el reenvío)	✓	✓	✓
Entregado	–	✓	✓ (provenido por una apertura/clic; de lo contrario *asumido* = enviado − rebotado)
Aperturas	–	✓	✓ (píxel de seguimiento)
Clics	–	✓	✓ (redirección de enlace)
Rebotes	–	✓	✓ (procesamiento de DSN)
Quejas de spam	–	✓	solo bucle de retroalimentación del proveedor
Darse de baja	–	✓	✓ (enlace de darse de baja + `List-Unsubscribe`)

[1] **OmniCRM no detecta qué proveedor usas.** Se configura solo con un host SMTP y credenciales, que lucen idénticas para cada reenvío. La columna del medio es lo que tu reenvío ofrezca, visto en **la consola propia de ese proveedor** (no en OmniCRM), a menos que conectes su webhook (Modelo 1) o uses el seguimiento nativo de OmniCRM (Modelo 2).

Modelo 1: Seguimiento nativo del proveedor

La mayoría de los reenvíos pueden aplicar su propio píxel de apertura y seguimiento de enlaces y registrar los eventos. Debido a que OmniCRM solo tiene un host SMTP y credenciales, **no puede decir qué proveedor es**, y no muestra estos números. Los lees al iniciar sesión en **la consola propia de ese proveedor** tú mismo. Esto no necesita configuración de OmniCRM. Es simplemente lo que tu reenvío elegido ofrece.

Para llevar esos eventos **a las analíticas de OmniCRM** en su lugar, OmniCRM debe ingerir el **webhook** del proveedor y hacer coincidir cada evento con un destinatario utilizando el campo de correlación por mensaje:

Proveedor	Campo de correlación que OmniCRM establece / utiliza
Mailjet	`X-MJ-CustomID` (manejador ya presente)
Mailgun	`X-Mailgun-Variables`
Amazon SES	`X-SES-MESSAGE-TAGS`
SendGrid	`unique_args` (a través de `X-SMTPAPI`)
Postmark	`X-PM-Metadata-*`
Mandrill	`X-MC-Metadata`

Cada formato de evento de proveedor difiere, por lo que ingerir eventos de un nuevo proveedor requiere un manejador de webhook específico para ese

proveedor.

Modelo 2: Seguimiento nativo de OmniCRM (portable)

Debido a que el seguimiento nativo del proveedor solo existe mientras permanezcas en ese proveedor, OmniCRM mide el compromiso por sí mismo, de manera idéntica en **cualquier** reenvío. Esto está **implementado y habilitado por defecto**:

- **Aperturas**: OmniCRM inyecta un píxel de seguimiento 1×1 en el HTML. La solicitud registra la apertura contra el destinatario.
- **Clics**: OmniCRM reescribe enlaces para redirigir a través de sí mismo. Registra el clic y reenvía al destino.
- **Entregado**: una apertura o clic es prueba de entrega. OmniCRM marca al destinatario como **entregado** en el primer hit de píxel/redirección, sin requerir webhook del proveedor. Los destinatarios que fueron `enviados` y nunca rebotan se cuentan como *asumidos entregados*.
- **Rebotes**: las campañas utilizan una ruta de retorno que enruta las notificaciones de rebote (DSNs) al servidor de correo entrante de OmniCRM para su procesamiento.
- **Darse de baja**: un encabezado `List-Unsubscribe` y un enlace de darse de baja en el cuerpo alimentan la lista de supresión de darse de baja existente.

Con este modelo, todas las estadísticas residen en OmniCRM y sobreviven a un cambio de proveedor de email.

Estado de soporte actual

- **Activo (por defecto):** Envío de campañas SMTP y **seguimiento nativo de OmniCRM en cualquier reenvío**. Esto proporciona `enviado`/`fallido` al enviar, además de **entregado, abierto, clicado, rebotado** detectados por OmniCRM mismo (píxel, redirección de clic, rebote VERP). No se requiere webhook del proveedor. Consulta Configuración de seguimiento.
- **Opcional:** si usas Mailjet, su webhook de eventos también puede enviar compromiso (correlacionado a través de `X-MJ-CustomID`). OmniCRM no

depende de ello.

Configuración de seguimiento

El seguimiento nativo de OmniCRM se configura bajo el bloque `edm` en `crm_config.yaml`:

```
edm:
  enable_open_tracking: true     # inyectar el píxel de apertura
en el HTML de la campaña
  enable_click_tracking: true    # reescribir enlaces a través de
la redirección de clics
  enable_unsubscribe: true       # agregar encabezado List-
Unsubscribe + enlace de darse de baja en el cuerpo
  tracking_base_url: ''          # URL pública que los
destinatarios pueden alcanzar; recae en crm.base_url
  tracking_secret: ''            # secreto que firma los tokens de
seguimiento; recae en mailjet.api_secret
  bounce_domain: ''              # dominio cuyo MX apunta al
servidor de correo entrante (habilita rebotes VERP)
  bounce_prefix: 'bounce'        # prefijo de la parte local VERP
-> bounce+<token>@<bounce_domain>
```

Parámetro	Tipo	Por defecto	Descripció
`enable_open_tracking`	Booleano	`true`	Inyectar un píxe seguimiento 1× el cuerpo HTML.
`enable_click_tracking`	Booleano	`true`	Reescribir enlac absolutos `http(` para redirigir a través de OmniC para que los clic registren.
`enable_unsubscribe`	Booleano	`true`	Agregar un encabezado `Lis Unsubscribe` (+ clic RFC 8058) y enlace de darse baja en el cuerp
`tracking_base_url`	Cadena	*(crm.base_url)*	URL base públic para enlaces de seguimiento. **Lo clientes de cor de los destinatarios deben poder alcanzarla, así no uses** `localh` Recae en `crm.base_url` cuando está en blanco.
`tracking_secret`	Cadena	*(mailjet.api_secret)*	Secreto utilizad para firmar toke de seguimiento

Parámetro	Tipo	Por defecto	Descripció
			HMAC. Recae en `mailjet.api_se` cuando está en blanco.
`bounce_domain`	Cadena	*(vacío)*	Dominio para direcciones de rebote VERP (`bounce+<token>@<domai` Su MX debe apu al servidor de cc entrante de OmniCRM. En bl desactiva la ruta retorno VERP.
`bounce_prefix`	Cadena	`bounce`	Prefijo de la part local para direcciones de rebote VERP.

Los tokens están firmados con HMAC como pares `(campaign_id, recipient_id)`, por lo que el seguimiento no necesita ninguna columna de base de datos adicional y los enlaces no pueden ser forjados o enumerados sin el secreto.

> **Nota sobre tokens:** el token son los IDs de campaña/destinatario *firmados*, no cifrados. Los IDs están codificados en base64, por lo que un destinatario que decodifica una URL de seguimiento puede leer los números de campaña y destinatario (secuenciales). Por lo tanto, los tokens son a prueba de manipulaciones pero no opacos. Si los IDs no deben ser expuestos, cambia a tokens aleatorios por destinatario almacenados en `EmailCampaignRecipient` (agrega una columna) o un payload de token cifrado.

Los puntos finales de seguimiento públicos (sin autenticación) son servidos por la API:

Punto final	Propósito
`GET /crm/email-campaign/edm/open/<token>`	Registra una apertura; devuelve un GIF transparente 1×1.
`GET /crm/email-campaign/edm/click/<token>?u=<url>`	Registra un clic, luego redirige 302 a `<url>` (solo http/https).
`GET` o `POST /crm/email-campaign/edm/unsubscribe/<token>`	Agrega al destinatario a la lista de supresión **y registra un opt-out de `marketing`/`email` en las preferencias de comunicación del contacto** (respetado en envíos futuros).

> **Importante:** `tracking_base_url` debe ser una URL que los destinatarios puedan alcanzar. La captura de rebotes requiere además `bounce_domain` con un registro MX que apunte al servidor de correo entrante de OmniCRM, que reconoce a los destinatarios VERP y registra el rebote en lugar de crear un ticket.

Preferencias de comunicación

Los opt-outs por contacto se almacenan en un campo JSON `communication_preferences` en el contacto (sin tabla adicional), estructurado como `{category: {channel: opted_in}}`. Las categorías son `marketing`, `newsletter`, `product_updates`, `surveys`, y los canales son `email`, `sms`. Utiliza un modelo de opt-out: una categoría/canal que falta significa que el contacto está optado, por lo que nuevas categorías o canales no necesitan un cambio de esquema.

Un darse de baja de campaña registra un opt-out de `marketing`/`email`, y la selección de destinatarios de campaña excluye cualquier contacto que se haya dado de baja de `marketing`/`email` (además de la lista de supresión global). Las

campañas EDM se tratan como la categoría `marketing`. Nota: el filtro de opt-out utiliza `JSON_EXTRACT` y no está indexado, lo cual está bien en volúmenes de contacto típicos.

Categorías de comunicación

Las categorías son **definidas por el operador** en `crm_config.yaml` bajo `edm.communication_categories` (sin tabla), como una lista de `{key, name, channel, default_opt_in, active}`:

```yaml
edm:
  communication_categories:
    # Tipos de mensajes de marketing por email (opt-out: default_opt_in true = suscrito hasta que se den de baja)
    - {key: marketing,       name: Marketing,           channel: email, default_opt_in: true}
    - {key: newsletter,      name: Newsletter,          channel: email, default_opt_in: true}
    - {key: promos,          name: Promociones y Ofertas, channel: email, default_opt_in: true}
    - {key: product_updates, name: Actualizaciones de Productos, channel: email, default_opt_in: true}
    - {key: surveys,         name: Encuestas y Comentarios, channel: email, default_opt_in: false}
    - {key: events,          name: Eventos y Webinars,   channel: email, default_opt_in: false}
    # Tipos de mensajes de marketing por SMS (opt-in: default_opt_in false = excluido hasta que opten)
    - {key: marketing,       name: Marketing SMS,       channel: sms,   default_opt_in: false}
    - {key: promos,          name: Promociones SMS,      channel: sms,   default_opt_in: false}
    - {key: service_alerts,  name: Alertas de Servicio, channel: sms,   default_opt_in: true}
```

Cada categoría pertenece a un **canal** (`email` o `sms`) y lleva su propio **opt-in por defecto**, por lo que un operador puede ejecutar, por ejemplo, una categoría de opt-out *Promociones (email)* junto a una categoría de *Alertas SMS (sms)* que requiere opt-in.

El opt-in efectivo para un contacto se resuelve como: una **preferencia** de contacto **explícita** gana; de lo contrario, se aplica el `default_opt_in` de la categoría.

Punto final	Propósito
`GET /crm/email-campaign/communication-categories`	Lista el catálogo de categorías (solo lectura; definido en `crm_config.yaml`).
`GET /crm/email-campaign/contacts/<id>/communication-preferences`	Opt-in resuelto por categoría activa para un contacto.
`PUT /crm/email-campaign/contacts/<id>/communication-preferences`	Establece una preferencia (cuerpo `{category, channel, opted_in}`).

Cada **campaña especifica una** `categoría` (por defecto `marketing`). La selección de destinatarios excluye contactos que se hayan dado de baja de esa categoría/canal, aplicando el `default_opt_in` de la categoría para contactos sin preferencia explícita.

El **enlace de darse de baja es un centro de preferencias**: muestra las categorías de email del contacto con su estado actual de opt-in (la categoría de la campaña resaltada), les permite guardar cambios y ofrece *Darse de baja de todos los emails* (una supresión global). Un darse de baja con un solo clic (RFC 8058 `List-Unsubscribe-Post`) excluye al contacto solo de la categoría de esa campaña.

Entregabilidad

Independientemente del proveedor, configura la autenticación para tu dominio de envío para que las campañas lleguen a la bandeja de entrada:

Mecanismo	Propósito
SPF	Autoriza a los servidores del proveedor a enviar en nombre de tu dominio.
DKIM	Firma criptográficamente los mensajes; cada proveedor proporciona el/los registro(s) DKIM para publicar.
DMARC	Alinea SPF/DKIM y establece una política para fallos; mejora la colocación y el informe.
Return-Path	Establecido para que las notificaciones de rebote (DSNs) sean enrutadas donde OmniCRM (o el proveedor) pueda procesarlas.

Pruebas

Para desarrollo y pruebas, apunta el reenvío a un servidor SMTP de depuración local que captura mensajes en lugar de entregarlos:

```
mailjet:
  smtp:
    host: 127.0.0.1
    port: 1025
    use_tls: false
    use_auth: false
```

Se proporciona un servidor de depuración para este propósito; imprime los destinatarios, asunto y ID de correlación de cada mensaje (y, en modo detallado, el cuerpo completo renderizado), por lo que puedes confirmar el renderizado de Jinja2 y la estructura del mensaje sin enviar email en vivo.

Solución de problemas

Campaña atascada o destinatarios marcados como `fallidos` inmediatamente

- Verifica `host`/`port` y que `username`/`password` estén configurados para el proveedor elegido (se establecen en las credenciales de Mailjet cuando están en blanco).
- Confirma que la dirección del remitente sea un remitente verificado/autorizado en el proveedor.
- Verifica que la autenticación del dominio del proveedor (SPF/DKIM) esté completa.

El email se envía pero la personalización muestra en blanco o literal `{{ ... }}`

- Confirma que el nombre de la variable exista para el segmento (por ejemplo, `first_name`).
- Verifica que la plantilla use sintaxis Jinja2 válida; un error de plantilla falla a ese destinatario y se registra en su mensaje de error.

El panel del proveedor muestra aperturas/clics pero las analíticas de OmniCRM no

- Esto es esperado a menos que OmniCRM ingiera el webhook de ese proveedor (Modelo 1) o el seguimiento nativo de OmniCRM esté habilitado (Modelo 2). La ingestión de eventos de Mailjet está disponible hoy; otros proveedores requieren un manejador específico para el proveedor.

No hay datos de apertura/clic en ninguna parte

- El seguimiento de aperturas/clics debe estar habilitado en el proveedor (configuración de cuenta o encabezados por mensaje). En un reenvío simple/autoalojado no hay seguimiento nativo, así que usa el seguimiento nativo de OmniCRM.

Gestión de Inventario de eSIM

Esta guía explica cómo configurar el inventario de eSIM para diferentes canales de distribución: digital (en línea) frente a impreso (tarjetas físicas).

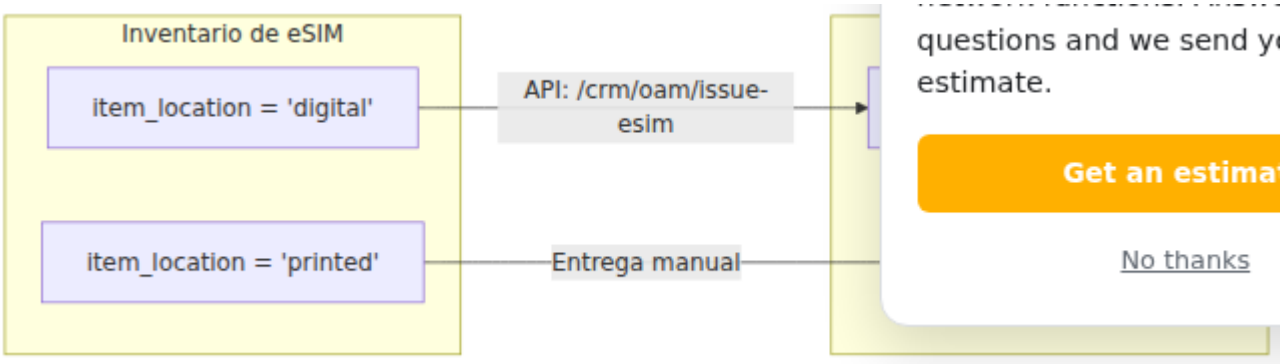
Ver también: Inventario para la gestión general de inventario, Provisionamiento de Tarjetas SIM para flujos de trabajo de aprovisionamiento y la API del emisor de eSIM.

Descripción General

Las eSIM pueden ser distribuidas a través de dos canales:

1. **Distribución Digital**: Emitidas instantáneamente a través del portal de auto-registro, entregadas por correo electrónico con código QR
2. **Distribución Impresa**: Preimpresas en tarjetas de cartón para entrega física o venta al por menor

El campo `item_location` en la tabla de Inventario controla para qué canal está disponible una eSIM.



Configuración del Canal de Distribución

El campo `item_location` determina cómo se puede distribuir una eSIM:

Valor	Descripción	Disponible A Través De
`digital`	Disponible para emisión en línea/inmediata	Portal de auto-registro, API
`printed`	Reservado para distribución de tarjetas físicas	Solo asignación manual
Otros valores	No disponible para emisión automática	Solo asignación manual

Configuración de eSIMs Digitales

Para hacer que las eSIM estén disponibles para distribución en línea:

1. Navega a **Inventario** en OmniCRM
2. Selecciona los elementos de inventario de eSIM destinados a la distribución digital
3. Establece `item_location` a `digital`
4. Guarda los cambios

Configuración de eSIMs Impresas

Para eSIMs que han sido impresas en tarjetas físicas:

1. Navega a **Inventario** en OmniCRM
2. Selecciona los elementos de inventario de eSIM que están impresos
3. Establece `item_location` a `printed` (o cualquier valor diferente de `digital`)

4. Guarda los cambios

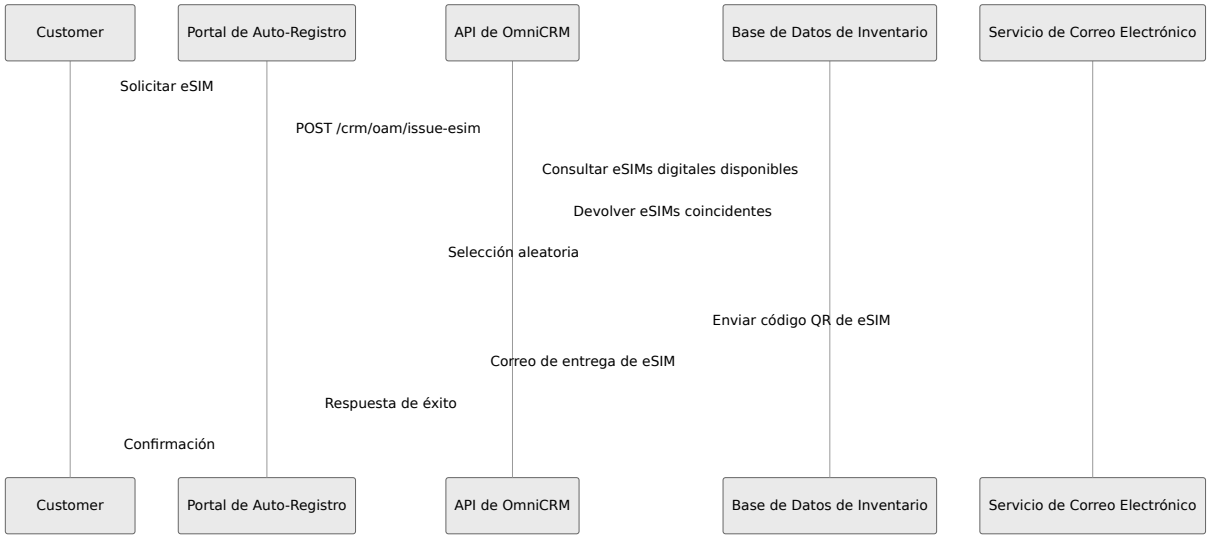
Cómo Funciona la Emisión Digital

Cuando un cliente solicita una eSIM a través del portal de auto-registro mediante `POST /crm/oam/issue-esim`:

1. La API consulta el inventario en busca de eSIMs disponibles
2. Solo se consideran las eSIMs que coinciden con **todos** los criterios a continuación
3. Se selecciona aleatoriamente una eSIM del grupo disponible
4. Los detalles de la eSIM se envían al cliente por correo electrónico

Criterios de Selección de Inventario

Campo	Condición	Propósito
`item`	Contiene "esim" (sin distinción entre mayúsculas y minúsculas)	Identifica elementos de inventario de eSIM
`item_location`	Es igual a `digital`	Restringe al canal de distribución digital
`customer_id`	Es nulo	Aún no asignado a un cliente
`service_id`	Es nulo	Aún no asignado a un servicio



Para obtener detalles completos sobre el endpoint issue-esim, incluidos los parámetros de solicitud, limitación de tasa y respuestas de error, consulta Provisionamiento de Tarjetas SIM - API Pública del Emisor de eSIM.

Manejo de Errores

Si no hay eSIMs digitales disponibles, la API devuelve:

Estado HTTP	Tipo de Error	Mensaje
503	`NoInventory`	"No hay eSIMs disponibles actualmente. Por favor, inténtalo de nuevo más tarde o elige una SIM física."

Campos Clave de Inventario para eSIMs

Campo	Tipo	Descripción
`item`	String	Identificador de tipo de elemento (debe contener "esim" para elementos de eSIM)
`item_location`	String	Canal de distribución: `digital` para en línea, otros valores para físico
`itemtext1`	String	ICCID: el identificador único de la eSIM
`management_url`	String	Cadena de activación LPA (por ejemplo, `LPA:1$rsp-eu.simlessly.com$...`)
`customer_id`	Integer	Cliente asignado (nulo si no asignado)
`service_id`	Integer	Servicio asignado (nulo si no asignado)

Para una referencia completa de todos los campos de elementos de inventario, consulta Inventario - Campos de Elementos de Inventario.

Importación Automática desde HSS

Cuando se importan SIMs desde el HSS utilizando el script de aprovisionamiento, dos campos se establecen automáticamente en función del registro del HSS:

Canal de Distribución (`item_location`):

- Si el nombre del lote contiene **"digital"** (sin distinción entre mayúsculas y minúsculas), la eSIM se marca como `item_location = 'digital'` para distribución en línea

- De lo contrario, la eSIM se establece en la ubicación física predeterminada (por ejemplo, "Oficina Central")

Cadena de Activación LPA (`management_url`):

- El campo `lpa` del registro de SIM del HSS se importa automáticamente como `management_url`
- Esto permite la generación de códigos QR en el CRM para la activación de eSIM
- Si el registro del HSS no tiene un valor `lpa`, la eSIM se importa sin una cadena de activación

Esto significa que el canal de distribución y los datos LPA pueden ser controlados a nivel de HSS:

- Nombra los lotes de manera apropiada (por ejemplo, "eSIMs Digitales Q1 2026", "Lote de Tarjetas Impresas 3")
- Asegúrate de que las eSIMs tengan su campo `lpa` poblado en el HSS antes de la importación

Mejores Prácticas

Segregación de Inventario

- Mantén separados los grupos de eSIM digitales e impresas utilizando `item_location`
- Monitorea los niveles del grupo digital para asegurar disponibilidad para los clientes de auto-registro
- Considera establecer alertas cuando el inventario de eSIM digital esté bajo

Actualizaciones Manuales por Lote

Al importar o actualizar eSIMs manualmente:

1. **eSIMs Digitales**: Establece `item_location = 'digital'` durante la importación

2. **eSIMs Impresas**: Establece `item_location = 'printed'` o la ubicación física (por ejemplo, "Oficina de Georgetown")

Conversión Entre Canales

Para mover eSIMs entre canales:

- **De Impreso a Digital**: Actualiza `item_location` de `printed` a `digital`
- **De Digital a Impreso**: Actualiza `item_location` de `digital` a `printed` (solo si aún no se ha emitido)

Solución de Problemas

No hay eSIMs Disponibles para Emisión en Línea

Síntomas: Los clientes ven "No hay eSIMs disponibles actualmente" al solicitar una eSIM en línea.

Causas posibles:

- No hay eSIMs con `item_location` establecido en `digital`
- Todas las eSIMs digitales ya están asignadas (tanto `customer_id` como `service_id` son no nulos)
- Los elementos de eSIM no tienen "esim" en el nombre del elemento

Resolución:

1. Verifica el inventario en busca de eSIMs con `item_location = 'digital'`
2. Verifica que haya eSIMs no asignadas (tanto `customer_id` como `service_id` son nulos)
3. Agrega más eSIMs al grupo digital o convierte las impresas si están disponibles

eSIMs Impresas Siendo Emitidas en Línea

Síntomas: eSIMs destinadas a tarjetas físicas están siendo emitidas a través del portal.

Causas posibles:

- `item_location` no establecido correctamente en eSIMs impresas

Resolución:

1. Actualiza todas las eSIMs impresas para tener `item_location = 'printed'`
2. Asegúrate de que las nuevas importaciones de eSIM impresas establezcan el valor correcto de `item_location`

Sistema de Difusión Celular

El Sistema de Difusión Celular en OmniCRM permite a los operadores de redes móviles enviar alertas de emergencia y advertencias públicas a dispositivos móviles dentro de áreas geográficas específicas. La Difusión Celular es una característica crítica de seguridad pública utilizada para alertas AMBER, advertencias meteorológicas, alertas de tsunamis y otras notificaciones de emergencia.

Ventaja Clave: A diferencia de los mensajes SMS estándar, los mensajes de Difusión Celular alertarán audiblemente en teléfonos que están en silencio, sin crédito o en roaming. Como este es un mensaje de difusión, es posible enviar una alerta a cada miembro de la población que tenga un teléfono móvil en cuestión de segundos.

Enlace de Advertencia de Omnitouch (OWL)

La plataforma Enlace de Advertencia de Omnitouch (OWL) proporciona una solución integral para profesionales de gestión de desastres y operadores de redes móviles:

- **Entidad de Difusión Celular (CBE)** - Aplicación web segura para que usuarios autorizados creen y difundan mensajes de advertencia de emergencia.
- **Centro de Difusión Celular (CBC)** - Componente de integración de red conforme a estándares que se conecta a redes celulares (2G/3G/4G/5G) para distribuir mensajes.

OWL está diseñado para usarse en cualquier dispositivo con un navegador web (Chrome/Firefox/Safari/Edge), como computadoras, laptops, tabletas o teléfonos móviles.

Visión General

La Difusión Celular (también conocida como Sistema de Advertencia Pública o PWS) permite a los operadores:

- **Enviar Alertas de Emergencia** - Distribuir información crítica de seguridad a todos los dispositivos en un área.
- **Dirigir Regiones Geográficas** - Difundir a áreas de seguimiento específicas o celdas de red controlando qué torres celulares difunden mensajes.
- **Soportar Múltiples Idiomas** - Proporcionar mensajes de alerta en múltiples idiomas simultáneamente (hasta 500 caracteres por idioma).
- **Gestionar el Ciclo de Vida de la Alerta** - Crear, actualizar, aprobar, monitorear y eliminar mensajes de difusión.
- **Integrarse con Sistemas Externos** - Conectar con la infraestructura del CBC (Centro de Difusión Celular) a través de múltiples interfaces de red celular.
- **Autenticación de Dos Factores** - Proceso de aprobación seguro utilizando Contraseñas de Un Solo Uso Basadas en Tiempo (TOTP).
- **Regla de Dos Personas** - Requisito opcional de aprobación de una segunda persona antes de la transmisión del mensaje.

A diferencia de los SMS, la Difusión Celular no requiere listas de suscriptores y puede alcanzar todos los dispositivos capaces en un área geográfica instantáneamente, lo que la hace ideal para alertas de seguridad pública críticas en tiempo. En la mayoría de las redes, una difusión a todos los dispositivos toma menos de 10 segundos.

Casos de Uso

La Difusión Celular se utiliza para:

- **Advertencias de Emergencia** - Desastres naturales (terremotos, tsunamis, inundaciones, incendios).
- **Alertas AMBER** - Notificaciones de secuestro de niños.

- **Alertas Meteorológicas** - Advertencias de clima severo, alertas de tornados.
- **Seguridad Pública** - Amenazas terroristas, derrames químicos, evacuaciones.
- **Mensajes de Prueba** - Pruebas del sistema y campañas de concienciación pública (los Mensajes de Prueba Mensuales no alertan a los usuarios pero serán recibidos).

El direccionamiento geográfico es a menudo de suma importancia en situaciones de emergencia. Por ejemplo, las alertas de Tsunami que aconsejan a aquellos cerca de áreas costeras buscar terrenos más altos no deben enviarse a personas en áreas interiores lejos del peligro. Al controlar qué torres celulares difunden mensajes de advertencia de emergencia, el alcance de la transmisión puede limitarse al ��rea geográfica apropiada.

Ciclo de Vida del Mensaje de Advertencia de Emergencia

Transmitir un mensaje de Advertencia de Emergencia requiere acción rápida mientras se proporciona información precisa y autenticación para garantizar la validez del mensaje.

El ciclo de vida del mensaje consta de cuatro etapas:

1. **Definición del Mensaje** - Tipo de mensaje, contenido del mensaje, configuraciones de expiración.
2. **Dirección del Mensaje** - Áreas geográficas y torres celulares a las que se difundirán.
3. **Aprobación del Mensaje** - Autorización/verificación de la identidad del operador y aprobación de la segunda persona (si es necesario).
4. **Revisión del Mensaje** - Confirmación final antes de la transmisión.

Después de la difusión, los mensajes pueden ser monitoreados, actualizados a medida que las situaciones evolucionan y detenidos cuando el peligro inmediato disminuye.

Etapa 1: Definición del Mensaje

Este paso define los parámetros básicos del mensaje de advertencia de emergencia a ser difundido:

Identificador del Mensaje
Diferentes tipos de mensajes tienen diferentes identificadores, que son tratados de manera diferente por los teléfonos receptores. Por ejemplo, un Mensaje de Prueba Mensual no debería alertar a usuarios reales, pero aún así recibirán el mensaje.

Nota: Cada plantilla de mensaje disponible en el sistema OWL ya tiene el Identificador de Mensaje (MI) apropiado incrustado, por lo que los usuarios del sistema no necesitan seleccionarlo manualmente al crear un mensaje de alerta.

Texto del Mensaje
Cuerpo de texto limitado a 500 caracteres que contiene el mensaje a ser mostrado a los usuarios finales. Los mensajes pueden ser proporcionados en **múltiples idiomas** agregando el texto del segundo idioma debajo del primero. Recuerde que el límite de 500 caracteres se aplica al mensaje total, incluyendo todos los idiomas.

Plantillas de Mensaje
Los mensajes de alerta de peligro pueden ser predefinidos con anticipación como "plantillas" para diferentes escenarios previstos, tales como:

- Inundaciones
- Tsunamis
- Terremotos
- Pruebas periódicas
- Otros escenarios de desastre

Las plantillas ahorran tiempo durante emergencias. Estas plantillas pueden ser modificadas según sea necesario al definir el mensaje, o los mensajes pueden ser escritos desde cero.

Expiración del Mensaje y Repeticiones
Los mensajes de emergencia tienen una vida útil finita para la cual son

relevantes. Al definir el mensaje:

- **Expira (minutos)** - Cuánto tiempo continuará siendo difundido el mensaje.
- **Repeticiones del Mensaje** - Cuántas veces será retransmitido.

Cada teléfono solo mostrará el mensaje al usuario una vez. Sin embargo, los sitios celulares continuarán transmitiendo mensajes hasta que se alcance el tiempo de expiración para asegurar que las personas que ingresen al área de cobertura desde afuera recibirán el mensaje.

Etapa 2: Direccionamiento

Los mensajes de Difusión Celular se envían a nivel de torre celular, y el alcance geográfico puede ser limitado seleccionando qué torres difunden el mensaje.

Direccionamiento Opcional
Este paso es opcional. No ingresar ninguna información de direccionamiento significa que todas las torres celulares transmitirán el mensaje de advertencia de emergencia.

Áreas de Objetivo Predefinidas
El sistema OWL tiene una base de datos de todas las torres celulares y puede definir áreas de objetivo en un mapa. Las áreas pueden ser dirigidas utilizando zonas predefinidas (determinadas con anticipación para una selección rápida) o dibujando áreas personalizadas en el mapa.

Herramientas de Dibujo en el Mapa
Se pueden crear áreas de objetivo personalizadas utilizando:

- Herramienta de Polígono - Dibujar límites de cobertura precisos.
- Herramienta de Círculo - Alertas rápidas basadas en radio.
- Herramienta de Rectángulo - Cobertura alineada a la cuadrícula.

La función "Agregar Nueva Zona" permite definir áreas de objetivo personalizadas que pueden ser guardadas para uso futuro.

Etapa 3: Aprobación

Un proceso de aprobación valida que la persona que emite el mensaje de Advertencia de Emergencia est�� autorizada para hacerlo.

Autenticación de Dos Factores
Utiliza Contraseñas de Un Solo Uso Basadas en Tiempo (TOTP) a través de:

- **Token físico** (como un RSA SecurID).
- **Solución basada en app** (Google Authenticator, Authy, Microsoft Authenticator, o otras apps compatibles con TOTP).

Los usuarios que crearán o aprobarán borradores de alertas de mensaje deben tener una app de autenticador en su smartphone para generar el código de autorización que el sistema solicitará.

Configuración de 2FA
Al configurar por primera vez 2FA:

1. Instale una app de autenticador en su smartphone (Google Authenticator, Authy, Microsoft Authenticator, etc.).
2. Navegue a la configuración de su cuenta OWL y escanee el código QR con su app de autenticador.
3. Ingrese el código de verificación para confirmar la configuración.
4. Guarde los códigos de respaldo en un lugar seguro.
5. Pruebe la generación de códigos antes de emergencias.

Para instrucciones detalladas de configuración de 2FA, consulte Autenticación de Dos Factores.

Si cambia de dispositivos móviles o la app deja de sincronizarse con su cuenta OWL, comuníquese con su Administrador del Sistema para obtener ayuda. Los administradores pueden restablecer los tokens de 2FA desde la página **Usuarios y Roles → Usuarios**.

Regla de Dos Personas
Donde el proceso requiere aprobación de una segunda persona, la persona que emite la alerta debe ingresar la Contraseña de Un Solo Uso Basada en Tiempo

de la otra persona antes de que se permita que el proceso continúe. Esto proporciona supervisión y minimiza el riesgo de uso indebido.

Roles de Usuario Granulares
Los roles de usuario individuales pueden configurarse para:

- Permitir que solo ciertos usuarios envíen mensajes predefinidos.
- Restringir el direccionamiento de mensajes a regiones específicas.
- Requerir flujos de trabajo de aprobación adicionales.

Etapa 4: Revisión

Una vez que las etapas de Definición del Mensaje, Direccionamiento y Aprobación están completas, el operador debe revisar el mensaje antes de la difusión final. Una vez satisfecho con los detalles del mensaje, puede transmitir el mensaje.

Velocidad de Transmisión: En la mayoría de las redes, una difusión a todos los dispositivos en la red toma menos de 10 segundos.

Etapa 5: Monitoreo y Actualizaciones

Una vez que se inicia la difusión del mensaje, los operadores pueden monitorear y gestionar los mensajes transmitidos.

Retroalimentación de la Red
Las redes celulares devuelven información sobre los sitios celulares que han difundido el mensaje. Si un sitio celular está fuera de l��nea o no disponible, esto se informará al operador.

Retransmisión Automática
Si algún sitio celular fuera de línea vuelve a estar disponible mientras la Advertencia de Emergencia sigue activa, todos los teléfonos conectados a esa celda recibirán el mensaje.

Actualizaciones en Vuelo
Una vez difundido, el mensaje puede ser:

- Actualizado a medida que la situación evoluciona.

- Modificado con nuevo contenido del cuerpo del mensaje.
- Recordado/detenido en cualquier momento.

Registros Históricos
Toda la información sobre mensajes históricos puede ser vista y revisada para fines de auditoría.

Estructura del Mensaje

Cada mensaje de Difusión Celular consiste en:

Configuración del Mensaje

- **Identificador del Mensaje** - Identificador único para el tipo de alerta (por ejemplo, 4370 para ETWS Terremoto, 4371 para ETWS Tsunami).
- **Categoría** - Categoría de alerta (normal, emergencia, alta, extrema).
- **Período de Repetición** - Segundos entre repeticiones de difusión.
- **Número de Difusiones** - Cuántas veces difundir el mensaje.
- **Período de Advertencia** - Duración en segundos durante la cual la advertencia es válida.
- **Indicador de Canal** - Tipo de canal utilizado para la difusión.

Mensajes Localizados

Cada mensaje del CBC puede incluir múltiples variantes de idioma:

- **Idioma** - Código de idioma ISO (en, es, fr, zh, etc.).
- **Cuerpo del Mensaje** - Texto de alerta en ese idioma (hasta 1395 caracteres).

El sistema difunde automáticamente todas las variantes de idioma, permitiendo a los destinatarios ver alertas en su idioma preferido.

Áreas de Seguimiento

Define el direccionamiento geográfico para la alerta:

- **Área de Seguimiento** - Identificador geográfico (ID de celda, código de área de seguimiento).
- **Operador** - Código del operador de red móvil (MCC-MNC).
- **Tipo de RAT** - Tecnología de Acceso Radio (LTE, 5G, UMTS, GSM).

Se pueden especificar múltiples áreas de seguimiento para cubrir regiones más grandes o múltiples operadores.

Creando un Mensaje de Difusión Celular

A través de la Interfaz Web:

1. **Navegar a Difusión Celular** - Acceder a la interfaz de gestión del CBC desde la navegación principal.
2. **Hacer clic en "Crear Alerta"** - Abre el formulario de creación de mensajes.
3. **Configurar Parámetros del Mensaje:**
    - Identificador del Mensaje (por ejemplo, 4370 para alertas de terremoto).
    - Categoría (normal, alta, extrema).
    - Período de Repetición (típicamente 5-60 segundos).
    - Número de Difusiones (999 para continuo, o conteo específico).
    - Período de Advertencia (duración en segundos).
    - Indicador de Canal (típicamente "básico").

4. **Agregar Mensajes Localizados:**
    - Hacer clic en "Agregar Idioma".
    - Seleccionar idioma del menú desplegable.
    - Ingresar texto del mensaje (máx. 1395 caracteres para GSM7, menos para Unicode).
    - Repetir para idiomas adicionales.
5. **Definir Áreas de Seguimiento:**
    - Hacer clic en "Agregar Área de Seguimiento".
    - Ingresar código de área de seguimiento.
    - Seleccionar operador (combinación MCC-MNC).
    - Elegir tipo de RAT (LTE, 5G, etc.).
    - Repetir para áreas geográficas adicionales.

6. **Revisar y Crear** - Verificar todos los detalles y hacer clic en "Crear Alerta".

A través de la API:

Endpoint: `PUT /crm/cbc/`

Permiso Requerido: `CREATE_CBC_MESSAGE`

Cuerpo de la Solicitud:

```
{
  "messageIdentifier": "4370",
  "category": "emergency",
  "repetitionPeriod": 10,
  "numberOfBroadcasts": 999,
  "warningPeriodSec": 3600,
  "channelIndicator": "basic",
  "localized_messages": [
    {
      "language": "en",
      "messageBody": "EARTHQUAKE WARNING: Magnitude 6.5 earthquake
detected. Take cover immediately. Drop, Cover, Hold On."
    },
    {
      "language": "es",
      "messageBody": "ADVERTENCIA DE TERREMOTO: Terremoto de
magnitud 6.5 detectado. Cúbrase inmediatamente. Agáchese, Cúbrase,
Agárrese."
    }
  ],
  "tracking_areas": [
    {
      "tracking_area": "12345",
      "operator": "310-410",
      "rat_type": "LTE"
    },
    {
      "tracking_area": "12346",
      "operator": "310-410",
      "rat_type": "5G"
    }
  ]
}
```


Respuesta:

```
{
  "cbc_message_id": 123,
  "cbc_unique_id": "550e8400-e29b-41d4-a716-446655440000",
  "messageIdentifier": "4370",
  "category": "emergency",
  "repetitionPeriod": 10,
  "numberOfBroadcasts": 999,
  "warningPeriodSec": 3600,
  "channelIndicator": "basic",
  "initiating_user": 5,
  "approving_user": null,
  "created": "2025-01-10T14:30:00Z",
  "localized_messages": [...],
  "tracking_areas": [...]
}
```


El mensaje se envía inmediatamente al Centro de Difusión Celular para su transmisión.

Gestión de Mensajes Existentes

Ver Todos los Mensajes

Endpoint: `GET /crm/cbc/`

Permiso Requerido: `VIEW_CBC_MESSAGE`

Devuelve una lista de todos los mensajes del CBC con su estado, marcas de tiempo y configuración.

Actualizar un Mensaje

Endpoint: `PATCH /crm/cbc/{cbc_message_id}`

Permiso Requerido: `UPDATE_CBC_MESSAGE`

Actualiza el contenido del mensaje, áreas de seguimiento o parámetros de difusión. Los mensajes actualizados se reenvían al CBC.

Cuerpo de la Solicitud:

```
{
  "cbc_message_id": 123,
  "numberOfBroadcasts": 500,
  "localized_messages": [
    {
      "language": "en",
      "messageBody": "UPDATED: Earthquake warning still in effect.
Aftershocks possible."
    }
  ]
}
```

Eliminar un Mensaje

Endpoint: `DELETE /crm/cbc/{cbc_message_id}`

Permiso Requerido: `DELETE_CBC_MESSAGE`

Elimina el mensaje de la base de datos e intenta cancelarlo en el CBC.

Flujo de Trabajo de Aprobación

Los mensajes de Difusión Celular soportan un flujo de trabajo de aprobación opcional para alertas de alto riesgo:

1. **Usuario Iniciador** - Miembro del personal que crea la alerta (`initiating_user` field).
2. **Usuario Aprobador** - Gerente que aprueba la alerta antes de la difusión (`approving_user` field).

Si se requiere aprobación:

- El mensaje se crea con `approving_user = null`.
- El mensaje se mantiene en estado de "aprobación pendiente".
- El usuario aprobador revisa el mensaje y lo aprueba o rechaza.
- Con la aprobación, se establece `approving_user` y el mensaje se difunde.

Este flujo de trabajo es configurable según la política de la organización.

Identificadores de Mensaje

Los identificadores de mensaje estándar siguen 3GPP TS 23.041:

Sistema de Advertencia de Terremoto y Tsunami (ETWS):

- **4370** - Advertencia de Terremoto ETWS.
- **4371** - Advertencia de Tsunami ETWS.
- **4372** - Advertencia Combinada de Terremoto y Tsunami ETWS.
- **4373-4378** - Otros Tipos de Emergencia ETWS.
- **4379** - Mensaje de Prueba ETWS.

Sistema de Alertas Móviles Comerciales (CMAS) / Alertas de Emergencia Inalámbricas (WEA):

- **4352** - Alerta Presidencial.
- **4353-4355** - Alertas Extremas.
- **4356-4359** - Alertas Severas.
- **4360-4363** - Alertas AMBER.
- **4364-4367** - Mensajes de Seguridad Pública.
- **4368-4369** - Pruebas Estatales/Locales.
- **4380-4381** - Mensajes de Prueba.

Rangos Personalizados:

- **0-999** - Reservado para alertas específicas del operador.
- **1000-4095** - Tipos de mensajes personalizados.

Integración con el Centro de Difusión Celular

La Entidad de Difusión Celular (CBE) necesita un mecanismo para entregar mensajes a redes celulares individuales. El Centro de Difusión Celular (CBC) de

OWL se conecta a cada red celular para enviar mensajes de Advertencia de Emergencia al público.

Redundancia Multi-Network
En caso de que un operador de red celular no esté disponible (corte o sin cobertura), si hay otra red celular operativa disponible, los usuarios seguirán recibiendo mensajes de Advertencia de Emergencia a través de las otras redes disponibles.

Instancias de CBC por Operador
Para asegurar que no haya conexión entre redes competidoras, OWL ejecuta una instancia de CBC separada para cada MNO; no se comparte entre operadores.

Configuración del CBC de OWL

La URL del CBC se configura en `crm_config.yaml`:

```
cbc_url: "http://cbc.example.com:8080"
```

Transmisión de Mensajes:

Cuando se crea o actualiza un mensaje del CBC:

1. OmniCRM CBE almacena el mensaje en su base de datos.
2. El mensaje se formatea para la API del CBC.
3. Se envía una solicitud HTTP POST a `{cbc_url}/alerts/send`.
4. El CBC se conecta a la red celular a través de la interfaz apropiada (CBSP, SBc-AP, o N50).
5. El CBC reconoce la recepción y comienza la difusión.
6. Los dispositivos en las áreas de seguimiento objetivo reciben la alerta.

Eliminación de Mensajes:

Cuando se elimina un mensaje, el CBE intenta cancelarlo en el CBC para detener las difusiones en curso.

Puntos de Integración de la Red Celular

Diferentes generaciones de redes celulares (2G/3G/4G/5G) tienen interfaces únicas para conectarse desde/hacia el Centro de Difusión Celular. Según las tecnologías utilizadas por cada red celular, se debe configurar la interfaz correcta.

El Centro de Difusión Celular de OWL soporta interfaces de Difusión Celular de 2G, 3G, 4G y 5G y se ha integrado con numerosos componentes de red celular comúnmente utilizados.

CBSP - 2G/3G - Controlador de Estación Base (BSC)

La interfaz del Protocolo de Servicio de Difusión Celular (CBSP) conecta el CBC al Controlador de Estación Base (BSC) que controla estaciones base 2G (GSM).

- Utilizado para mensajería de Difusión Celular de 2G y 3G con despliegues combinados de Controlador de Red de Radio/Controlador de Estación Base.
- Puede configurarse como cliente o servidor dependiendo del proveedor del BSC.
- Se debe establecer una conexión entre todos los BSC en la red y el CBC de OWL.
- Las interfaces son monitoreadas continuamente con alertas para indicar si un enlace CBSP ha caído.

Nota: 3GPP definió el Protocolo de Difusión de Área de Servicio (SABP) para uso en despliegues RNC independientes. Esto puede utilizarse si es necesario cuando CBSP no es soportado para celdas 3G en un RNC/BSC combinado, sin embargo, pueden ser requeridas pruebas adicionales y soporte del proveedor del RNC.

SBc-AP - 4G/5G No Independiente - MME/IWF

La interfaz SBc-AP conecta el CBC de OWL al MME (Entidad de Gestión de Movilidad) que sirve a eNodeB/gNodeBs de 4G y 5G.

- Utilizado para redes LTE de 4G.
- También utilizado para 5G No Independiente (la mayoría de los despliegues a partir de 2025).

- Se debe establecer una conexión entre todos los MMEs en la red y el CBC de OWL.
- Las interfaces son monitoreadas continuamente con alertas para indicar si un enlace SBc-AP ha caído.

N50 - 5G Independiente - AMF

Para redes 5G independientes, la interfaz N50 conecta el CBC de OWL al AMF (Función de Gestión de Acceso y Movilidad) que sirve a gNodeBs de 5G.

- La interfaz está presente en el CBC de OWL.
- No se ha probado extensivamente con AMFs de terceros debido al pequeño número de redes 5G SA comercialmente disponibles en 2025.
- Será completamente soportada a medida que los despliegues de 5G SA se vuelvan más comunes.

Requisitos de Redes de MNO

Se debe establecer una red entre el CBC de OWL y la red del Operador de Red Móvil para alcanzar las interfaces descritas anteriormente.

Esto se maneja caso por caso, pero generalmente requiere:

- Conexión cruzada/fibra dedicada entre el CBC y la red MNO.
- Cada interfaz lógicamente separada.
- Conectividad a cada punto de integración (MME, RNC, BSC) en la red celular.

Equipos de Red Soportados

El CBC de OWL ha sido probado e integrado con componentes de red celular comúnmente utilizados de proveedores importantes:

Integración de Datos del Sitio Celular

OWL soporta la recopilación automática de datos de:

- **Nokia NetAct**
- **Huawei U2000 / U2020**

- **ZTE NetNumen / ZXPOS**
- **Ericsson ENM**

Alternativamente, los datos del sitio celular pueden ser proporcionados al equipo de operaciones de Omnitouch periódicamente a través de correo electrónico.

Gestión de Usuarios y Control de Acceso

Control de Acceso Basado en Roles (RBAC)

El sistema OWL utiliza control de acceso basado en roles (RBAC): las personas (Usuarios Registrados) se asignan uno o más Roles, y cada Rol es un conjunto de Permisos. Los Permisos son la unidad más pequeña de acceso (por ejemplo, crear alerta de mensaje en borrador). El acceso efectivo de un Usuario Registrado es la unión de los Permisos de todos los Roles asignados.

Componentes de RBAC:

- **Usuarios** - Personas reales que inician sesión en el sistema OWL.
- **Permisos** - Micro capacidades (por ejemplo, aprobar borrador de mensaje, crear mensaje, ver informes).
- **Roles** - Conjuntos nombrados de permisos (por ejemplo, Aprobadores de Mensajes, Creadores de Mensajes).
- **Asignación** - Los usuarios reciben uno o más Roles; los permisos se agregan.

Beneficios de RBAC:

1. **Protección de Datos** - Los usuarios solo ven y hacen lo que se les permite.
2. **Ajuste Operativo** - Los roles reflejan funciones laborales (Admin, Creador de Mensajes, Aprobador de Mensajes).
3. **Administración Simple** - Conceder acceso asignando roles; evitar la microgestión por usuario.

Permisos del Sistema

Los permisos del sistema generalmente siguen patrones CRUD con cuatro opciones:

- **Ver** - Leer o navegar por mensajes e informes.
- **Crear** - Crear o agregar una alerta de mensaje.
- **Actualizar** - Editar o modificar una alerta de mensaje en borrador.
- **Eliminar** - Eliminar o quitar una alerta de mensaje en borrador.

Permisos Básicos del CBC:

- `CREATE_CBC_MESSAGE` - Crear nuevos mensajes de difusión.
- `VIEW_CBC_MESSAGE` - Ver mensajes existentes y su estado.
- `UPDATE_CBC_MESSAGE` - Modificar el contenido del mensaje o parámetros de difusión.
- `DELETE_CBC_MESSAGE` - Eliminar mensajes y cancelar difusiones.

Asigne estos permisos a roles según las responsabilidades de seguridad pública de su organización.

Para información completa sobre la gestión de roles y permisos de usuario, consulte Usuarios y Roles.

Gestión de Cuentas de Usuario

Inicio de Sesión por Primera Vez

Cuando se configura un nuevo usuario en el sistema, se le proporcionan credenciales de inicio de sesión por parte del Administrador del Sistema:

1. Navegue a la URL de inicio de sesión de OWL proporcionada por su Administrador del Sistema.
2. Ingrese su nombre de usuario y contraseña.
3. Si 2FA está habilitado para su cuenta, se le pedirá que lo configure (consulte la sección 2FA anterior).

Mejor Práctica: Los usuarios deben cambiar su contraseña inicial inmediatamente después del primer inicio de sesión por seguridad. Navegue a la configuración de su perfil de usuario para cambiar su contraseña.

Para más detalles sobre flujos de autenticación, consulte Flujos de Autenticación y Controles de Admin.

Restablecimiento de Contraseña

Los usuarios pueden restablecer sus propias contraseñas:

1. Desde la página de inicio de sesión del sistema, seleccione "Olvidé mi Contraseña".
2. Siga el proceso de verificación por correo electrónico.
3. Establezca una nueva contraseña.

Restablecimiento de Contraseña del Administrador:

Los Administradores del Sistema pueden restablecer contraseñas para usuarios desde la página **Usuarios y Roles → Usuarios**:

1. Navegue a **Usuarios y Roles → Usuarios**.
2. Seleccione el usuario que requiere restablecimiento de contraseña.
3. Haga clic en **Restablecer Contraseña**.
4. El sistema genera una contraseña temporal aleatoria segura (12 caracteres).
5. La contraseña temporal se muestra al administrador.
6. Proporcione la contraseña temporal al usuario a través de un canal seguro (teléfono o en persona, nunca por correo electrónico).
7. El usuario puede iniciar sesión con la contraseña temporal.
8. **Importante:** El usuario debe cambiar esta contraseña inmediatamente después de iniciar sesión.

Para más detalles, consulte Usuarios y Roles.

Agregar Nuevos Usuarios al Sistema

Los Administradores del Sistema pueden agregar nuevos usuarios navegando a **Usuarios y Roles → Usuarios**:

Información requerida para cada nuevo Usuario Registrado:

- Nombre
- Apellido
- Correo Electrónico
- Contraseña (temporal)
- Número de Móvil
- Rol(es)

Mejor Práctica: Implementar un Formulario de Solicitud de Nuevo Usuario del Sistema que se alinee con las políticas y procedimientos existentes de su organización. Solo agregue usuarios después de la aprobación adecuada.

Editar Perfiles de Usuario

Los Administradores del Sistema pueden cambiar permisos y acceso para usuarios existentes:

1. Navegue a **Usuarios y Roles → Usuarios**.
2. Seleccione el usuario a editar.
3. Modifique los roles asignados según sea necesario.
4. Guarde los cambios.

Eliminar Usuarios del Sistema

Los Administradores del Sistema pueden eliminar usuarios de **Usuarios y Roles → Usuarios**.

Advertencia: Una vez que un usuario ha sido eliminado del sistema OWL, la acción no puede deshacerse. Tenga mucho cuidado antes de ejecutar esta tarea.

Ver Todos los Usuarios del Sistema

En cualquier momento, el Administrador del Sistema puede ver una lista de todos los Usuarios Registrados y ver qué permisos y roles tiene cada uno navegando a **Usuarios y Roles → Usuarios**.

Historial de Mensajes de Alerta

El sistema OWL mantiene un registro de todos los mensajes de alerta enviados para fines de auditoría y para apoyar la revisión operativa.

Accediendo al Historial de Mensajes:

Navegue a **Gestionar Difusiones Celulares** para ver:

- Todos los mensajes enviados anteriormente.
- Estado del mensaje (activo, expirado, cancelado).
- Marca de tiempo de la transmisión.
- Usuarios iniciadores y aprobadores.
- Detalles de direccionamiento geográfico.
- Contenido y parámetros del mensaje.

Esto ayuda con:

- Auditoría del sistema.
- Revisión de procedimientos operativos utilizados por el NDMO y otras agencias de alerta.
- Comprender la efectividad del mensaje.
- Requisitos de cumplimiento e informes.

Visualización del Mapa de Difusión Celular

El Mapa de Difusión Celular proporciona una **interfaz visual** para planificar áreas de cobertura geográfica e identificar qué torres celulares difundirán alertas.

Accediendo al Mapa

Navegue a:

O directamente:

Permiso Requerido: `VIEW_CBC_MESSAGE` o `CREATE_CBC_MESSAGE`

Características del Mapa

El mapa muestra:

Ubicaciones de Torres Celulares:

- **Íconos de Torres** - Posiciones de sitios celulares mostradas como íconos de torres celulares.
- **Indicadores de Sector** - Flechas direccionales que muestran la orientación del sector de la antena.
- **Nombres de Torres** - Identificadores y nombres de sitios.
- **Datos de Cobertura** - Cargados desde el archivo `cellSites.txt`.

Herramientas de Dibujo:

- **Herramienta de Polígono** - Dibujar áreas de cobertura personalizadas.
- **Herramienta de Círculo** - Crear zonas de difusión circulares.
- **Herramienta de Rectángulo** - Definir áreas de cobertura rectangulares.

Análisis de Cobertura:

- **Lista de Sitios** - Tabla que muestra todos los sitios celulares dentro del área dibujada.
- **Gestión de Zonas** - Guardar y cargar zonas de cobertura predefinidas.
- **Conteo de Sitios** - Conteo en tiempo real de torres en el área de cobertura.

Usando el Mapa

Paso 1: Cargar Datos del Sitio Celular

El mapa carga automáticamente las ubicaciones de las torres celulares desde `/cellSites.txt` al cargar la página. Este archivo contiene:

- **lat/lng** - Coordenadas de la torre.
- **site_name** - Identificador.

- **sectors** - Direcciones de antena en grados.

Paso 2: Dibujar Área de Cobertura

Seleccione una herramienta de dibujo:

1. Haga clic en **Polígono** para dibujar formas personalizadas.
    - Haga clic para agregar puntos.
    - Doble clic para completar.
    - Crea límites de cobertura precisos.
2. Haga clic en **Círculo** para dibujar áreas circulares.
    - Haga clic en el punto central.
    - Arrastre para establecer el radio.
    - Útil para alertas rápidas basadas en radio.
3. Haga clic en **Rectángulo** para dibujar áreas rectangulares.
    - Haga clic en una esquina.
    - Arrastre hasta la esquina opuesta.
    - Eficiente para cobertura alineada a la cuadrícula.

Paso 3: Revisar Sitios Cubiertos

Después de dibujar, el sistema:

- Calcula qué sitios celulares caen dentro del área.
- Muestra la lista en la tabla debajo del mapa.
- Muestra nombres de sitios y coordenadas.
- Proporciona conteo de torres afectadas.

Tabla de Cobertura de Ejemplo:

Paso 4: Exportar Datos de Cobertura

Utilice los datos de cobertura para:

- **Identificar códigos de área de seguimiento** para la difusión.
- **Estimar el alcance de la alerta** basado en el conteo de torres.
- **Planificar la respuesta de emergencia** al comprender las áreas afectadas.

- **Guardar definiciones de zona** para uso futuro.

Integración del Flujo de Trabajo

La visualización del mapa se integra con la creación de difusión:

Planificando una Alerta:

1. **Abrir Mapa del CBC** para visualizar el área afectada.
2. **Dibujar zona de cobertura** usando la herramienta de polígono.
3. **Revisar sitios cubiertos** y ajustar límites.
4. **Anotar códigos de área de seguimiento** para los sitios celulares afectados.
5. **Crear mensaje del CBC** con el direccionamiento apropiado.

Ejemplo: Advertencia de Tsunami Costero

Gestión de Zonas

Guardando Zonas:

Las zonas de cobertura predefinidas pueden ser guardadas para escenarios comunes:

- **Región Costera** - Para advertencias de tsunami/surge de tormenta.
- **Zonas de Incendios Forestales** - Para alertas de evacuación.
- **Centros Urbanos** - Para alertas AMBER en áreas pobladas.
- **Corredores de Autopistas** - Para alertas de tráfico/clima.

Formato de Datos de Zona:

Las zonas se almacenan en `site_data.json`:

```
[
  {
    "area": "Región Costera",
    "polygon": [
      {
        "coord": [
          {"lat": -33.8688, "lng": 151.2093},
          {"lat": -33.8650, "lng": 151.2070},
          {"lat": -33.8600, "lng": 151.2150},
          {"lat": -33.8688, "lng": 151.2093}
        ]
      }
    ]
  }
]
```

Cargando Zonas Guardadas:

1. Seleccione la zona del menú desplegable.
2. El polígono se muestra automáticamente en el mapa.
3. La lista de sitios se actualiza para mostrar torres cubiertas.
4. Modifique el polígono si es necesario.

Configuración del Mapa

API de Google Maps:

El mapa requiere una clave de API de Google Maps configurada en el entorno:

```
VITE_GOOGLE_API_KEY=your_google_maps_api_key_here
```

Datos del Sitio Celular:

Actualice `public/cellSites.txt` con las ubicaciones de las torres celulares de su red:

```
# Formato: lat,lng,nombre,sector1,sector2,sector3
-33.8688,151.2093,SYD_CENTRAL,0,120,240
-33.8650,151.2070,SYD_HARBOUR,45,165,285
```

Obtenga las coordenadas del sitio celular de:

- Herramientas de planificación de red.
- Registros de instalación de torres.
- Encuestas de campo con GPS.
- Configuración del controlador de red de radio (RNC).

Estilo del Mapa:

Los estilos de mapa personalizados pueden configurarse en `mapStyles.js` para:

- Resaltar servicios de emergencia.
- Mostrar características topográficas.
- Enfatizar centros de población.
- Coincidir con la marca de la organización.

Casos de Uso

Planificación de Emergencias:

- Pre-planificar zonas de cobertura para áreas de riesgo conocidas.
- Probar diferentes estrategias de direccionamiento.
- Estimar el alcance de la población.
- Coordinar con servicios de emergencia.

Verificación de Alertas:

- Confirmar que el mensaje alcanzará el área prevista.
- Identificar brechas de cobertura.
- Evitar alertas excesivas en regiones adyacentes.
- Validar códigos de área de seguimiento.

Análisis de Red:

- Visualizar la distribución de torres celulares.
- Identificar superposiciones de cobertura.
- Planificar despliegues de torres para mejor cobertura de alertas.
- Optimizar la eficiencia de difusión.

Mejores Prácticas

Basado en la experiencia con clientes de todo el mundo, Omnitouch recomienda las siguientes mejores prácticas para todas las implementaciones de OWL.

Contenido del Mensaje:

- Mantenga los mensajes concisos y claros (menos de 360 caracteres para alertas de una sola página).
- Use letras mayúsculas para alertas de emergencia (mejora la legibilidad).
- Incluya elementos de acción específicos ("Evacuar inmediatamente", "Buscar refugio").
- Evite la jerga técnica.
- Pruebe los mensajes con dispositivos reales antes de emergencias.

Soporte de Idiomas:

- Siempre proporcione mensajes en el idioma principal de la región.
- Incluya idiomas adicionales para áreas multiculturales.
- Asegúrese de que las traducciones sean culturalmente apropiadas.
- Pruebe caracteres especiales y soporte Unicode.

Direccionamiento Geográfico:

- Use las áreas de seguimiento más pequeñas necesarias para evitar la fatiga de alertas.
- Considere la densidad de población al establecer el período de repetición.
- Pruebe el direccionamiento geográfico antes de emergencias.

- Mantenga documentación precisa del área de seguimiento.

Pruebas:

- Use identificadores de mensajes de prueba (4379, 4380-4381) para simulacros.
- Programe pruebas regulares del sistema.
- Verifique que la integración del CBC esté funcionando.
- Capacite al personal sobre procedimientos de emergencia.

Fatiga de Alertas:

- Úselo solo para emergencias genuinas.
- Evite la sobre-difusión.
- Establezca períodos de repetición apropiados.
- Use niveles de severidad de manera apropiada.

Seguridad de la Cuenta de Usuario

Seguridad de Tokens de 2FA:

- Proteja los tokens de 2FA físicos como tarjetas de acceso a edificios.
- Informe de inmediato sobre tokens perdidos o robados.
- Pruebe 2FA regularmente para asegurarse de que esté funcionando.
- Guarde los códigos de respaldo al configurar 2FA y almacénelos de forma segura sin conexión.
- Comuníquese con el Administrador del Sistema si necesita un restablecimiento de 2FA.

Para más información sobre la configuración y recuperación de 2FA, consulte Autenticación de Dos Factores.

Mantenimiento de Datos

Áreas de Objetivo Predefinidas:

A medida que cambian los límites geográficos, se producen desarrollos y cambian las áreas de riesgo, hay una necesidad de revisar las Áreas de Objetivo Predefinidas. Omnitouch sugiere que estos datos se revisen **anualmente por el NDMO**, con el apoyo de otras agencias de alerta donde sea aplicable.

Las actualizaciones de los límites pueden definirse utilizando varias plataformas GIS comunes o Google Earth, y luego proporcionarse por correo electrónico al equipo de operaciones de Omnitouch, que realizará los cambios en el sistema.

Plantillas de Mensaje Predefinidas:

Las plantillas de mensaje predefinidas deben revisarse **al menos anualmente por el NDMO**, con el apoyo de otras agencias de alerta donde sea aplicable, para asegurar:

- Que el contenido del mensaje aún refleja con precisión el peligro y la llamada a la acción.
- Que la información de contacto y las instrucciones estén actualizadas.
- Que las traducciones de idiomas sigan siendo precisas.
- Que el tono y la urgencia del mensaje sean apropiados.
- Que las referencias a servicios o procedimientos de emergencia estén actualizadas.

Las actualizaciones pueden proporcionarse por correo electrónico al equipo de operaciones de Omnitouch, que realizará los cambios en el sistema.

Datos del Sitio Celular:

- Revise y actualice los datos del sitio celular siempre que ocurran cambios en la red.
- Frecuencia típica de actualización: mensual o trimestral.
- Coordinar con los equipos de planificación de red de MNO.
- Verificar la precisión después de actualizaciones o expansiones importantes de la red.

Arquitectura del Sistema

Todos los productos de Omnitouch están diseñados para soportar implementaciones geográficamente distribuidas.

Opciones de Implementación

Todos los componentes pueden ejecutarse como:

- **Contenedores (K8s)** - Implementaciones de contenedores orquestadas por Kubernetes.
- **Máquinas Virtuales** - VMware, Proxmox, HyperV.
- **Nube Privada** - Infraestructura de nube local.
- **Nube Pública** - AWS, GCP.
- **Hardware Dedicado** - Implementación directa en hardware.

Arquitectura Distribuida

La arquitectura distribuida permite:

- A las Oficinas Locales de Gestión de Desastres acceder al CBE de OWL y distribuir mensajes incluso si una región se aísla de la red nacional.
- Múltiples Entidades de Difusión Celular y múltiples Centros de Difusión Celular por operador/país.
- A las agencias locales de respuesta a desastres (Gobierno Municipal, Policía, Bomberos, etc.) emitir alertas a su región incluso si la principal NDMO (Oficina Nacional de Gestión de Desastres) pierde acceso.

Esto es particularmente valioso cuando los MNO han distribuido su red celular con recursos locales de BSC/MME.

Redes CBE y CBC

Para asegurar que no haya conexión entre redes competidoras, cada MNO tiene una instancia de CBC separada (no compartida).

Requisitos de red entre CBE y CBC:

- Todo el tráfico está cifrado entre CBE y CBC.

- Autenticación basada en certificados mutuos.
- Conectividad desde CBE a CBC a través de TLS en el puerto TCP 443.
- Coordinación requerida entre MNO y el NDMO o la agencia que aloja el CBE.

Consideraciones de Acceso

La agencia que aloja el CBE necesitará definir procedimientos de acceso para los usuarios finales (es decir, Citrix, VPN, etc.), teniendo en cuenta que el sistema debe ser accesible en escenarios no ideales como:

- Cortes a gran escala de redes de energía pública.
- Fallas en la red de telecomunicaciones.
- Desastres naturales que afectan la infraestructura.

Requisitos de Implementación

Requisitos de VM del CBE (NDMO / Gobierno)

3x Máquinas Virtuales:

- **2x VMs de CBE** (Idealmente en diferentes centros de datos/zonas de disponibilidad).
- **1x VM de Monitoreo**.

Cada VM requiere:

- **Almacenamiento:** 50GB.
- **CPU:** 2x CPU Virtual.
- **RAM:** 8GB.
- **SO:** SO base proporcionado por Omnitouch.
- **Redes:** Permitir tráfico a las VMs del CBC en el puerto TCP 443 para tráfico TLS para controlar los CBC.

Requisitos de VM del CBC (MNO)

3x Máquinas Virtuales:

- **2x VMs de CBC** (Idealmente en diferentes centros de datos/zonas de disponibilidad).
- **1x VM de Monitoreo**.

Cada VM requiere:

- **Almacenamiento:** 50GB.
- **CPU:** 2x CPU Virtual.
- **RAM:** 8GB.
- **SO:** SO base proporcionado por Omnitouch.
- **Conectividad:** A cada uno de los Puntos de Integración en la red celular (MME, RNC, BSC).
- **Redes:** Permitir tráfico desde las VMs de CBE en el puerto TCP 443 para que el tráfico TLS sea controlado por los CBE.

Pasos de Integración

Implementar OWL implica los siguientes pasos:

1. **Fuente de nuevo(s) operador(es)** - Identificar operadores de red móvil participantes.
2. **Admin del proyecto (configuración)** - Establecer gobernanza y administración del proyecto.
3. **Seleccionar ubicación de alojamiento para el CBE** - Determinar dónde se alojará el CBE.
4. **Definir Usuarios y Flujos/Procedimientos de Mensajes** - Establecer flujos de trabajo de aprobación y roles de usuario.
5. **Definir Polígonos para Áreas de Objetivo y Plantillas de Mensaje** - Preconfigurar escenarios comunes.
6. **Asignaciones de Direcciones IP** - Asignar direcciones IP para el NDMO y los MNO.
7. **Configurar VPN de sitio a sitio** - Establecer conexión segura con el equipo de Omnitouch.
8. **Implementar VMs de CBE en el NDMO** - Instalar la Entidad de Difusión Celular.

9. **Implementar VMs de CBC en el MNO** - Instalar el Centro de Difusión Celular en cada operador.
10. **Configurar Elementos de Red** - Establecer conectividad al CBC desde el equipo de red celular.
11. **Redes entre VMs de CBE y CBC** - Establecer comunicación segura.
12. **Redes entre VMs de CBC y Elementos de Red** - Conectar a BSC/MME/AMF.
13. **Configurar acceso a API para NMS** - Configurar integración de datos del sitio celular.
14. **Configuración de monitoreo y pruebas** - Verificar monitoreo y alertas.
15. **Verificación/Pruebas de alertas de prueba** - Realizar pruebas del sistema.
16. **Prueba de advertencia pública** - Realizar prueba pública de extremo a extremo.

Límites de Caracteres

Los mensajes de Difusión Celular tienen límites de caracteres estrictos según la codificación:

Codificación GSM de 7 bits (inglés, caracteres latinos básicos):

- Página única: 93 caracteres.
- Múltiples páginas: 15 páginas × 93 = 1395 caracteres como máximo.

Codificación Unicode UCS-2 (escritos no latinos, emojis):

- Página única: 41 caracteres.
- Múltiples páginas: 15 páginas × 41 = 615 caracteres como máximo.

Plataforma OWL:

- Texto del mensaje limitado a 500 caracteres.
- La interfaz web muestra el conteo de caracteres restantes y advierte cuando se acerca a los límites.

Monitoreo y Registros

La actividad de Difusión Celular se registra para fines de auditoría:

- Eventos de creación, actualización y eliminación de mensajes.
- Usuarios iniciadores y aprobadores.
- Marcas de tiempo e identificadores de mensajes.
- Respuestas y errores de la API del CBC.
- Detalles de direccionamiento geográfico.

Acceda a los registros a través del Registro de Actividad o consultas a la base de datos:

```
SELECT * FROM cbc
WHERE created >= '2025-01-01'
ORDER BY created DESC;
```

Integración con Dispositivos Móviles

Los mensajes de Difusión Celular son recibidos por dispositivos móviles compatibles:

Soporte de Dispositivos:

- La mayoría de los smartphones a partir de 2015 soportan Difusión Celular.
- Los teléfonos básicos pueden tener soporte limitado.
- El dispositivo debe estar conectado a la red (no se requieren créditos de datos/SMS).
- Funciona incluso durante congestión de red cuando los SMS fallan.

Experiencia del Usuario:

- La alerta se muestra como una notificación de pantalla completa.
- Suena un tono de alerta único.

- La alerta persiste hasta que se reconoce.
- No se requiere suscripción del usuario.
- No puede ser bloqueada por los usuarios para alertas presidenciales/extremas.

Prueba de Recepción de Dispositivos:

Para verificar que los dispositivos pueden recibir alertas:

1. Envíe un mensaje de prueba (identificador 4379 o 4380-4381).
2. Asegúrese de que el dispositivo esté en el área de seguimiento objetivo.
3. Verifique que el dispositivo tenga habilitada la Difusión Celular en la configuración.
4. Verifique con múltiples modelos de dispositivos y versiones de SO.

Funcionalidad Adicional

La plataforma OWL puede ser extendida con características opcionales para complementar la mensajería de Difusión Celular:

Mensaje Masivo / SMS

Enviar SMS regulares a individuos para notificaciones suplementarias.

- A menudo utilizado para alertas degradadas para informar a las personas que la amenaza inmediata ha pasado, de manera menos intrusiva.
- Mucho más lento que la Difusión Celular pero puede incluir un mecanismo de confirmación para verificar que el mensaje fue recibido.
- Útil para comunicaciones de seguimiento dirigidas.

Llamadas de Voz para Línea Fija

Llamar automáticamente a números de línea fija y reproducir mensajes de advertencia de emergencia.

- Reproducción de mensajes pregrabados.

- Rendición de texto a voz del mensaje de advertencia de emergencia.
- Llega a poblaciones sin teléfonos móviles.
- Puede verificar la entrega del mensaje a través de la finalización de la llamada.

Publicación Cruzada en Redes Sociales

Publicar automáticamente mensajes de Advertencia de Emergencia en canales oficiales de redes sociales.

- Extiende el alcance más allá de la red celular.
- Proporciona referencia para aquellos que se perdieron la alerta inicial.
- Permite mensajes extendidos más allá de los l��mites de caracteres.

Transmisión Automática de Radio/TV

Transmitir automáticamente mensajes de emergencia a través de radio y televisión.

- Reproducción de mensajes pregrabados.
- Rendición de texto a voz del mensaje de advertencia de emergencia.
- Llega a poblaciones durante cortes de red.
- Complementa la Difusión Celular para una cobertura integral.

Dispositivos de Advertencia Externos

El CBC de OWL puede conectarse a una variedad de fuentes externas:

- Plataformas de redes sociales.
- APIs públicas.
- Sistemas de llamadas de voz.
- Sistemas de transmisión de radio.
- Alarmas físicas y sirenas.
- Señalización electrónica.

Se pueden explorar opciones de personalización como parte de la fase de diseño.

Personalización y Mantenimiento

Pruebas Periódicas

Se deben realizar pruebas periódicas de la solución a intervalos regulares para asegurar:

- Que la solución y todos los componentes estén funcionando correctamente.
- Que todo el personal esté familiarizado con los procesos y procedimientos requeridos para emitir Mensajes de Advertencia de Emergencia.
- Que los puntos de integración sigan operativos.
- Que las plantillas de mensajes estén actualizadas y sean efectivas.

Pruebas Recomendadas:

- Mensajes de prueba mensuales utilizando identificadores 4379, 4380-4381.
- Pruebas completas del sistema trimestrales que incluyan flujos de trabajo de aprobación.
- Pruebas de advertencia pública anuales con notificación anticipada.
- Sesiones de capacitación regulares para usuarios autorizados.

Mantenimiento de Datos del Sitio Celular

Cuando los operadores agregan o eliminan sitios celulares, o cambian Áreas/Identificadores de Seguimiento de los sitios celulares, esta información debe compartirse con el equipo de Omnitouch para asegurar que los datos de la herramienta de mapeo permanezcan precisos.

Integración Automática de Datos

OWL soporta la recopilación automática de datos de:

- Nokia NetAct.

- Huawei U2000 / U2020.
- ZTE NetNumen / ZXPOS.
- Ericsson ENM.

Actualizaciones Manuales de Datos

Alternativamente, los datos del sitio celular pueden ser proporcionados al equipo de operaciones de Omnitouch periódicamente a través de correo electrónico en varios formatos.

Frecuencia de Actualización: Revise y actualice los datos del sitio celular siempre que ocurran cambios en la red, típicamente mensualmente o trimestralmente.

Áreas de Objetivo Predefinidas

A medida que cambian los límites geográficos, se producen desarrollos y cambian las áreas de riesgo, hay una necesidad de revisar las Áreas de Objetivo Predefinidas utilizadas en la etapa de Direccionamiento.

Revisión Anual: Las áreas de objetivo predefinidas deben revisarse anualmente por el NDMO (Oficina Nacional de Gestión de Desastres).

Proceso de Actualización: Las actualizaciones de los límites pueden definirse utilizando varias plataformas GIS comunes o Google Earth, y luego proporcionarse al equipo de operaciones de Omnitouch.

Plantillas de Mensaje Predefinidas

Revisión Anual: Las plantillas de mensaje predefinidas deben revisarse anualmente por el NDMO para asegurar:

- Que el contenido del mensaje refleje los procedimientos de emergencia actuales.
- Que las traducciones de idiomas sean precisas.
- Que los identificadores de mensaje sean apropiados.
- Que la información de contacto y las instrucciones estén actualizadas.

Proceso de Actualización: Las actualizaciones pueden proporcionarse por correo electrónico al equipo de operaciones de Omnitouch.

Flujos de Aprobación de Mensajes

Diferentes regiones tienen diferentes requisitos sobre cómo emiten mensajes y flujos de aprobación.

Regla de Dos Personas: Se aboga por el uso de la regla de dos personas siempre que sea práctico para asegurar supervisión en la presentación de mensajes.

Roles de Usuario Granulares: Los roles de usuario individuales pueden configurarse para:

- Permitir que solo ciertos usuarios envíen mensajes predefinidos.
- Restringir el direccionamiento a regiones específicas.
- Requerir pasos de aprobación adicionales.
- Minimizar el riesgo de uso indebido.

Mantenimiento de Seguridad / 2FA

El equipo de operaciones de Omnitouch puede apoyar con:

- Restablecimiento de tokens de 2FA.
- Reemisión de tokens perdidos/dañados/expirados.
- Auditoría de seguridad y gestión de tokens.

Procedimientos de Seguridad de Tokens:

Cuando se emiten tokens, se detalla un procedimiento para los pasos a seguir si un token se pierde o no se puede localizar. La persona que acepta el token debe seguir estos procedimientos para asegurar que el sistema no sea abusado.

Acciones Inmediatas para Tokens Perdidos:

1. Informe de inmediato sobre el token perdido al personal autorizado.

2. El token se desactiva en el sistema.
3. Se realiza una revisión de seguridad.
4. Se emite un nuevo token tras la verificación de seguridad.

Integración de API Externas

Se encuentra disponible un conjunto completo de APIs para la Entidad de Difusión Celular que permite a sistemas de terceros integrarse e interactuar con el CBE.

Capacidades de API:

- **Informes/Monitoreo** - Verificar el estado y alcance de los mensajes transmitidos.
- **Creación de Mensajes** - Crear y difundir nuevos mensajes programáticamente.
- **Verificaciones de Salud** - Verificar periódicamente la salud del sistema con generación de tráfico de mensajes de prueba.
- **Consultas de Estado** - Recuperar el estado del mensaje, estadísticas de entrega y métricas del sistema.

Consulte la sección de Referencia de API a continuación para la documentación detallada de los endpoints.

Referencia de API

Todos los endpoints del CBC requieren autenticación y permisos apropiados.

Crear Mensaje:

`PUT /crm/cbc/`

Obtener Todos los Mensajes:

`GET /crm/cbc/`

Actualizar Mensaje:

`PATCH /crm/cbc/{cbc_message_id}`

Eliminar Mensaje:

`DELETE /crm/cbc/{cbc_message_id}`

Consulte la documentación de Swagger en `/crm/docs/` para especificaciones detalladas de la API.

Búsqueda Global

La función de Búsqueda Global proporciona una **interfaz de búsqueda unificada** para encontrar rápidamente clientes, contactos, servicios, inventario y sitios en toda la base de datos de OmniCRM.

Ver también: Clientes, Inventario, Gestión de Servicios.

Acceder a la Búsqueda Global

Desde cualquier lugar en el CRM:

Haz clic en el ícono de búsqueda en la barra de navegación superior o navega a:

La página de búsqueda global aparece con un gran cuadro de búsqueda y opciones de filtro.

Cómo Funciona

La búsqueda global realiza una **búsqueda cruzada** a través de cinco tipos de datos:

Qué se Busca:

1. **Clientes** - Nombre del cliente
2. **Contactos** - Nombre, apellido, dirección de correo electrónico, número de teléfono
3. **Sitios** - Nombre del sitio
4. **Inventario** - Números de serie, ICCIDs, identificadores (itemtext1, itemtext2)
5. **Servicios** - Nombre del servicio, UUID del servicio

Comportamiento de Búsqueda:

- **Coincidencia parcial** - Busca términos que contengan tu consulta (por ejemplo, "Smith" coincide con "John Smith" y "Smithson")
- **Sin distinción entre mayúsculas y minúsculas** - "john" coincide con "John", "JOHN" y "john"
- **Múltiples entidades** - Una sola búsqueda devuelve resultados de todos los tipos de entidad
- **Resultados paginados** - Muestra 10 resultados por página de forma predeterminada

Realizando una Búsqueda

Búsqueda Básica

1. Ingresa tu término de búsqueda en el cuadro de búsqueda
2. Haz clic en **"Buscar"** o presiona Enter

Ejemplos de términos de búsqueda:

- Nombre del cliente: `"Acme Corp"`
- Número de teléfono: `"+1234567890"` o `"1234567890"`
- Correo electrónico: `"john@example.com"` o `"john"`
- Número de serie: `"ICCID8944"` o solo `"8944"`
- UUID del servicio: `"123e4567-e89b"`

Incluir Cuentas Cerradas

Por defecto, la búsqueda solo devuelve resultados de cuentas de clientes **Abiertas**.

Para buscar en todas las cuentas, incluidas las cerradas:

1. Marca la casilla **"Incluir Cuentas Cerradas"**
2. Haz clic en **"Buscar"** nuevamente

Esto buscará:

- Clientes con `customer_status = "Closed"`
- Contactos, servicios, sitios e inventario vinculados a clientes cerrados

Casos de uso para la búsqueda de cuentas cerradas:

- Encontrar registros históricos de clientes
- Localizar equipos de servicios desprovisionados
- Buscar números de teléfono o servicios antiguos
- Recuperar datos de clientes para reactivación

Entendiendo los Resultados de Búsqueda

Formato de Visualización de Resultados

Los resultados se muestran en una lista desplazable que muestra:

> John Smith ID de Cliente: 123 Tipo: cliente
>
> John Smith (Contacto) ID de Cliente: 123 Tipo: contacto
>
> Móvil - +44 7700 900123 ID de Cliente: 123 Tipo: servicio

Cada resultado muestra:

- **Nombre/Título** - El identificador principal (enlace clicable)

- **ID de Cliente** - El cliente padre al que pertenece
- **Tipo** - El tipo de entidad (cliente, contacto, sitio, inventario, servicio)

Tipos de Resultados Explicados

Resultados de Clientes:

Hacer clic abre la página de resumen del cliente mostrando todos los detalles, servicios, contactos, etc.

Resultados de Contactos:

Hacer clic abre la página del cliente con la pestaña Contactos activa, desplazándose al contacto específico.

Resultados de Sitios:

Hacer clic abre la página del cliente con la pestaña Sitios activa.

Resultados de Inventario:

Hacer clic abre la página del cliente con la pestaña Inventario activa. Si el inventario no está asignado (sin customer_id), se vincula a la lista principal de inventario en su lugar.

Resultados de Servicios:

Hacer clic abre la página del cliente con la pestaña Servicios activa, destacando el servicio específico.

Navegación desde Resultados

Todos los resultados de búsqueda son **enlaces clicables** que navegan directamente a la página relevante:

Patrón de Enlace:

- `/customers/{customer_id}` - Registros de clientes
- `/customers/{customer_id}#4` - Contactos (pestaña 4)

- `/customers/{customer_id}#2` - Sitios (pestaña 2)
- `/customers/{customer_id}#8` - Inventario (pestaña 8)
- `/customers/{customer_id}#3` - Servicios (pestaña 3)
- `/inventory-items-list` - Inventario no asignado

El fragmento de hash (`#`) selecciona automáticamente la pestaña correcta cuando se carga la página del cliente.

Paginación

Los resultados están paginados con 10 elementos por página:

> Mostrando resultados 11-20 de 47

Navega a través de las páginas utilizando:

- Botones **Anterior/Siguiente**
- **Números de página** - Haz clic en una página específica
- **Teclado** - Flechas izquierda/derecha (si está implementado)

Escenarios Comunes de Búsqueda

Escenario 1: Encontrar Cliente por Teléfono

El usuario llama y proporciona el número de teléfono.

> Resultados: • John Smith (Contacto) - ID de Cliente: 123 • Móvil - 555-0123 (Servicio) - ID de Cliente: 123

Haz clic en cualquiera de los resultados para acceder a la cuenta del cliente.

Escenario 2: Localizar Tarjeta SIM

El técnico necesita encontrar qué cliente tiene una SIM específica.

> Resultados: • 89445380000000001234 (Inventario) - ID de Cliente: 456

Haz clic en el resultado para ver la asignación de la SIM y los detalles del cliente.

Escenario 3: Encontrar Cliente Inactivo

Necesitas localizar un cliente que cerró su cuenta hace 6 meses.

> ☑ Incluir Cuentas Cerradas
>
> Resultados: • Acme Corporation (cliente) - ID de Cliente: 789

Escenario 4: Buscar por Correo Electrónico

El cliente envía un correo electrónico al soporte, el personal necesita encontrar su cuenta.

> Resultados: • John Smith (Contacto) - ID de Cliente: 123

Escenario 5: Encontrar Servicio por UUID

El registro de aprovisionamiento muestra el UUID del servicio, necesitas encontrar qué cliente.

> Resultados: • Móvil - +44 7700 900123 (Servicio) - ID de Cliente: 456

Consejos de Búsqueda

Para Mejores Resultados:

- **Usa términos parciales** - "Smith" es mejor que "John Smith" para resultados más amplios
- **Prueba variaciones** - Si "John" no funciona, prueba con teléfono o correo electrónico
- **Incluye cuentas cerradas** - Al buscar datos históricos
- **Sé específico para equipos** - Usa números de serie completos para inventario
- **Busca UUID de servicio** - Cuando no se conocen otros identificadores

Qué se Busca (por Entidad):

Clientes:

- Solo el nombre del cliente (no dirección, notas u otros campos)

Contactos:

- Nombre
- Apellido
- Dirección de correo electrónico
- Número de teléfono

Sitios:

- Solo el nombre del sitio

Inventario:

- itemtext1 (típicamente ICCID, número de serie, dirección MAC)
- itemtext2 (típicamente IMSI, identificador secundario)
- *Nota: No busca itemtext3-20 o notas de inventario*

Servicios:

- Nombre del servicio
- UUID del servicio

Qué No Se Busca:

- Direcciones de clientes
- Notas de clientes
- Descripciones de transacciones
- Detalles de facturas
- Registros de aprovisionamiento
- Entradas de registro de actividad
- Notas de inventario (más allá de itemtext1/2)

Referencia de API

Endpoint de Búsqueda Global

```
GET /utilities/search_everything?
search=Smith&page=1&per_page=10&search_closed_records=false
Authorization: Bearer <token>
```

Parámetros de Consulta:

- `search` (requerido) - El término de búsqueda
- `page` (opcional) - Número de página (predeterminado: 1)
- `per_page` (opcional) - Resultados por página (predeterminado: 10)
- `search_closed_records` (opcional) - Incluir cuentas cerradas (predeterminado: false)

Respuesta:

```
{
  "data": [
    {
      "id": 123,
      "name": "John Smith",
      "customer_id": 123,
      "type": "customer"
    },
    {
      "id": 456,
      "name": "John Smith",
      "customer_id": 123,
      "type": "contact"
    },
    {
      "id": 789,
      "name": "Mobile - +44 7700 900123",
      "customer_id": 123,
      "type": "service"
    }
  ],
  "pagination": {
    "current_page": 1,
    "per_page": 10,
    "total_pages": 5,
    "total_items": 47
  }
}
```


Lógica de Búsqueda (Backend):

El backend realiza una unión SQL a través de todas las tablas de entidades:

```sql
-- Clientes
SELECT customer_id AS id,
       customer_name AS name,
       customer_id,
       'customer' AS type
FROM customer
WHERE customer_name LIKE '%Smith%'
  AND customer_status = 'Open'

UNION ALL

-- Contactos
SELECT contact_id AS id,
       CONCAT(contact_firstname, ' ', contact_lastname) AS name,
       customer_id,
       'contact' AS type
FROM customer_contact
WHERE (contact_firstname LIKE '%Smith%' OR
       contact_lastname LIKE '%Smith%' OR
       contact_email LIKE '%Smith%' OR
       contact_phone LIKE '%Smith%')

UNION ALL

-- Sitios
SELECT site_id AS id,
       site_name AS name,
       customer_id,
       'site' AS type
FROM customer_site
WHERE site_name LIKE '%Smith%'

UNION ALL

-- Inventario
SELECT inventory_id AS id,
       itemtext1 AS name,
       customer_id,
       'inventory' AS type
FROM inventory
WHERE itemtext1 LIKE '%Smith%' OR
      itemtext2 LIKE '%Smith%'
```

```
UNION ALL

-- Servicios
SELECT service_id AS id,
       service_name AS name,
       customer_id,
       'service' AS type
FROM customer_service
WHERE service_name LIKE '%Smith%' OR
      service_uuid LIKE '%Smith%'
```

Los resultados se paginan y se devuelven.

Consideraciones de Rendimiento

Rendimiento de Búsqueda:

- Las búsquedas utilizan consultas LIKE con comodines (`%term%`)
- Actualmente no se ha implementado indexación de texto completo
- Bases de datos grandes (>100k clientes) pueden experimentar búsquedas más lentas
- Resultados limitados a 10 por página para mejorar el rendimiento

Consejos de Optimización:

- Sé específico con los términos de búsqueda para reducir el conjunto de resultados
- Usa el filtro de cuentas cerradas para reducir el alcance de búsqueda
- Considera agregar índices de base de datos en campos que se buscan con frecuencia

Solución de Problemas

No se encontraron resultados (pero el registro existe)

- **Causa:** El término de búsqueda no coincide con el formato de datos almacenados

- **Ejemplos:**
  - Teléfono almacenado como "+44 7700 900123", buscar "07700900123" no coincidirá
  - Correo electrónico almacenado como "<john.smith@example.com>", buscar "john" no coincidirá
- **Solución:** Prueba variaciones, usa coincidencias parciales que definitivamente existan

Búsqueda demasiado lenta

- **Causa:** Base de datos grande, consulta compleja a través de múltiples tablas
- **Solución:**
  - Usa términos de búsqueda más específicos
  - Limita a cuentas abiertas solamente (desmarca cuentas cerradas)
  - Contacta al administrador sobre la indexación de la base de datos

Resultados enlazan al cliente incorrecto

- **Causa:** Múltiples clientes/contactos con el mismo nombre
- **Solución:** Usa el ID de Cliente para diferenciar, o busca por un identificador único (correo electrónico, teléfono)

Cuentas cerradas no aparecen

- **Causa:** Casilla "Incluir Cuentas Cerradas" no marcada
- **Solución:** Marca la casilla y busca nuevamente

Documentación Relacionada

- `basics_customers` - Gestión de clientes
- `basics_navigation` - Navegación general
- `administration_inventory` - Búsquedas de inventario

Sistema de Recarga y Recarga

El sistema de recarga de OmniCRM proporciona un **portal de recarga prepago de autoservicio** para que los clientes agreguen crédito o extiendan la validez del servicio a través del Portal de Autoservicio. Esta función se utiliza comúnmente para:

- **Servicios de datos móviles** - Tarjetas SIM prepago y servicios solo de datos
- **Servicios de hotspot** - Dongles de hotspot WiFi y dispositivos de internet portátiles
- **Servicios inalámbricos fijos** - Acceso a internet prepago

Vea también: Provisionamiento de Tarjetas SIM para obtener información sobre el aprovisionamiento de servicios móviles y la gestión del inventario de SIM.

Descripción General

El sistema de recarga permite a los clientes comprar días adicionales de servicio a través de un proceso de pago simplificado y de múltiples pasos con procesamiento de pagos integrado de Stripe.

Características Clave:

- Portal de autoservicio para clientes (sin intervención del personal requerida)
- Selección de duración flexible (1-30 días)
- Visualización de uso en tiempo real antes de la compra
- Procesamiento de pagos seguro impulsado por Stripe
- Reembolsos automáticos si la recarga falla
- Generación de facturas y transacciones
- Integración del sistema de aprovisionamiento para la activación del servicio

Acceso al Portal de Recarga

El portal de recarga se accede a través de una **URL pública** que los clientes pueden visitar sin iniciar sesión en el CRM:

Cómo Acceden los Clientes:

- Enlace directo enviado por SMS cuando el saldo es bajo
- Código QR en materiales impresos
- Enlace en el portal de autoservicio
- Compartido a través del soporte al cliente

El portal detecta automáticamente el servicio del cliente basado en su dirección IP de solicitud o IMSI.

Proceso de Recarga

El flujo de recarga consta de **4 pasos**:

Paso 1: Selección de Datos

Los clientes seleccionan cuántos días de servicio desean comprar.

Interfaz:

- **Control deslizante** - Seleccionar de 1 a 30 días
- **Cálculo de precio en vivo** - Muestra el costo total basado en la selección
- **Visualización de fecha de caducidad** - Calcula y muestra cuándo expirará el servicio
- **Visualización de uso actual** - Muestra el saldo restante/caducidad antes de la recarga

Ejemplo de Visualización:

Configuración de Precios:

- El precio por día se configura a través de la variable de entorno `VITE_TOPUP_PRICE_PER_DAY`
- Predeterminado: $10 USD por día
- La moneda se establece a través de `VITE_CURRENCY_CODE`

Paso 2: Información de Facturación

Los clientes proporcionan sus datos de contacto para la transacción:

- **Nombre**
- **Apellido**
- **Dirección de Correo Electrónico**

Esta información se utiliza para:

- Generación de facturas
- Correo electrónico de recibo de pago
- Registros de transacciones
- Procesamiento de reembolsos (si es necesario)

Paso 3: Pago

Procesamiento de pagos seguro a través de **formularios de pago alojados por el proveedor** (Stripe Elements, PayPal SDK).

Métodos de Pago Soportados:

- Tarjetas de crédito (Visa, Mastercard, Amex)
- Tarjetas de débito
- Monederos digitales (Apple Pay, Google Pay, PayPal) *si están habilitados por el proveedor de pagos*

Características de Seguridad:

- Integración de proveedor de pagos conforme a PCI
- No se almacenan detalles de la tarjeta en OmniCRM
- Soporte para autenticación 3D Secure

- Transmisión de pagos cifrada

Flujo de Pago:

1. Se muestra el formulario de pago seguro con entrada de tarjeta
2. El cliente ingresa los detalles de pago
3. El pago se procesa a través del proveedor configurado
4. La tarjeta se carga inmediatamente
5. Éxito/fallo del pago manejado

> **NOTA**
>
> Si el pago tiene éxito pero el aprovisionamiento de la recarga falla (por ejemplo, error de red, OCS inalcanzable), el sistema inicia automáticamente un **reembolso completo** al método de pago del cliente.

Paso 4: Finalización

Pantalla de Éxito:

> Su servicio ha sido extendido. Nueva fecha de caducidad: 17 de enero de 2025
>
> Recibo enviado a: <customer@example.com> ID de Transacción: TXN-123456

Pantalla de Fallo:

Si la recarga falla, el sistema muestra un error y procesa automáticamente un reembolso:

> No pudimos completar su recarga. Su pago ha sido reembolsado.
>
> Error: No se puede conectar al sistema de facturación
>
> Por favor, inténtelo de nuevo o contacte con soporte.

Procesamiento en el Backend

Cuando un cliente completa el pago, lo siguiente ocurre automáticamente:

1. Validación de Pago

El sistema valida:

- El estado de la Intención de Pago es `succeeded`
- El monto del pago coincide con los días seleccionados (`days × price_per_day`)
- La Intención de Pago no se ha procesado antes (previene recargas dobles)

2. Operación de Recarga

```
- API endpoint: POST /oam/topup_dongle
- Valida service_uuid e IMSI
- Llama a OCS/CGRateS para agregar saldo
- Crea trabajo de aprovisionamiento (play_topup_hotspot)
```

3. Creación de Registros

El sistema crea múltiples registros en la base de datos:

- **Registro HotspotTopup** - Rastrea la transacción de recarga
  - payment_intent_id
  - service_uuid
  - imsi
  - días comprados
  - topup_amount
  - estado (Éxito/Fallido/Reembolsado)
- **Registro de Transacción** - Transacción financiera
  - Título: "Recarga de Hotspot - 7 Días"
  - Monto: topup_amount (positivo)
  - Vinculado a service_id y customer_id
- **Registro de Factura** - Factura de pago
  - Contiene la transacción de recarga

    - Marcada como pagada inmediatamente
    - Referencia de pago: Stripe payment_intent_id
- **Transacción de Pago** - Transacción de crédito compensatorio
    - Título: "Pago por [Título de la Factura]"
    - Monto: topup_amount (negativo - crédito)
    - Vincula el pago de la factura a la cuenta del cliente

4. Trabajo de Aprovisionamiento

Se crea un trabajo de aprovisionamiento con el playbook `play_topup_hotspot` que:

- Se conecta a la API de OCS/CGRateS
- Agrega saldo a la cuenta
- Extiende la fecha de caducidad
- Crea una entrada en el registro de actividad
- Envía una notificación de confirmación (si está configurada)

La API espera a que se complete el aprovisionamiento (polling con intervalos de 0.2s, máximo 25 iteraciones) antes de devolver el éxito al cliente.

5. Reembolso Automático en Caso de Fallo

Si algún paso falla después del pago:

```
if topup_provisioning_failed:
    refund = stripe.Refund.create(
        payment_intent=payment_intent_id,
        reason='requested_by_customer'  # Reembolso automático del sistema
    )
    status_message = "Recarga Fallida. Reembolsando el pago..."
```

El reembolso se procesa automáticamente y se notifica al cliente en pantalla.

Puntos de API

Endpoint de Recarga

```
POST /oam/topup_dongle
Content-Type: application/json

{
  "service_uuid": "123e4567-e89b-12d3-a456-426614174000",
  "imsi": "310120123456789",
  "days": 7,
  "payment_intent_id": "pi_1234567890abcdef",
  "topup_amount": 70.00
}
```

Respuesta (Éxito):

```
{
  "result": "OK",
  "status": 200,
  "provision_id": 456,
  "payment_intent_id": "pi_1234567890abcdef",
  "service_uuid": "123e4567-e89b-12d3-a456-426614174000",
  "invoice_id": 789
}
```

Respuesta (Fallo):

```
{
  "result": "Failed",
  "Reason": "Tiempo de espera de conexión OCS",
  "status": 500
}
```

Verificaciones de Validación:

- Todos los campos requeridos presentes (service_uuid, imsi, days, payment_intent_id, topup_amount)

- topup_amount coincide con days: `topup_amount × 100 == days × 1000` (en centavos)
- La Intención de Pago existe en Stripe
- El monto de la Intención de Pago coincide: `payment_intent.amount == topup_amount × 100`
- El estado de la Intención de Pago es `succeeded`
- La Intención de Pago no ha sido procesada anteriormente (verifica la tabla `HotspotTopup`)

Endpoint de Uso

Recupera el uso actual y la información del servicio para el cliente:

```
GET /oam/usage
```

Respuesta:


```
{
  "imsi": "310120123456789",
  "service": {
    "service_uuid": "123e4567-e89b-12d3-a456-426614174000",
    "service_name": "Datos Móviles - 0412345678",
    "service_status": "Activo"
  },
  "balance": {
    "expiry": "2025-01-10T23:59:59Z",
    "unlimited": true
  },
  "requestingIp": "203.0.113.45"
}
```


Este endpoint utiliza la dirección IP de solicitud para identificar automáticamente el servicio del cliente.

Configuración

Variables de Entorno

Configure estas en el archivo `.env` de OmniCRM-UI:

```
# Configuración del Portal de Recarga
VITE_TOPUP_PRICE_PER_DAY=10
VITE_CURRENCY_CODE=AUD
VITE_SELF_CARE_NAME="YourCompany"

# Configuración de Pagos
VITE_PAYMENT_VENDORS="stripe"
VITE_STRIPE_PUBLISHABLE_KEY=pk_live_...
```

Configuración de Stripe

El sistema de recarga utiliza Intenciones de Pago de Stripe:

1. **Habilitar Intenciones de Pago** en su Panel de Stripe
2. **Configurar Webhook** para recibir actualizaciones de estado de pago (opcional pero recomendado)
3. **Configurar métodos de pago** (tarjetas, monederos, etc.)
4. **Modo de prueba** - Utilizar claves de prueba para desarrollo

```
# Desarrollo
VITE_STRIPE_PUBLISHABLE_KEY=pk_test_...

# Producción
VITE_STRIPE_PUBLISHABLE_KEY=pk_live_...
```

Configuración del Playbook

El playbook de aprovisionamiento `play_topup_hotspot.yaml` debe configurarse para:

- Aceptar la variable `days`
- Conectarse a la API de OCS/CGRateS
- Agregar saldo a la cuenta
- Actualizar la fecha de caducidad del servicio

Ejemplo de estructura de playbook:

```
- name: Recargar servicio de hotspot
  hosts: localhost
  tasks:
    - name: Agregar saldo a OCS
      uri:
        url: "{{ ocs_api_url }}/add_balance"
        method: POST
        body:
          imsi: "{{ imsi }}"
          days: "{{ days }}"
          service_uuid: "{{ service_uuid }}"
```

Notificaciones de Saldo Bajo

El sistema puede enviar notificaciones automáticas cuando el saldo del cliente es bajo:

Notificaciones por SMS:

Cuando son activadas por eventos de OCS (`Action_Balance_Low`, `Action_Balance_Out`, `Action_Balance_Expired`):

Notificaciones por Correo Electrónico:

Configuradas en los planes de acción de OCS/CGRateS para enviar alertas de saldo.

Disparadores de Notificación:

- `Action_Balance_Low` - Saldo por debajo del umbral (por ejemplo, 2 días restantes)

- `Action_Balance_Out` - Saldo agotado
- `Action_Balance_Expired` - Servicio expirado

Cada notificación incluye el enlace al portal de recarga para un fácil acceso del cliente.

Solución de Problemas

Problemas Comunes

"Sistema de pago no disponible"

- **Causa:** La biblioteca de Stripe no se pudo cargar o clave publicable inválida
- **Solución:**
  - Verifique que `VITE_STRIPE_PUBLISHABLE_KEY` esté configurado correctamente
  - Verifique que la cuenta de Stripe esté activa
  - Revise la consola del navegador en busca de errores de JavaScript

"Recarga fallida. Reembolsando el pago..."

- **Causa:** Falló el trabajo de aprovisionamiento (OCS inalcanzable, error de playbook, etc.)
- **Solución:**
  - Revise los registros de aprovisionamiento: `GET /crm/provision/provision_id/<id>`
  - Verifique que la API de OCS/CGRateS sea accesible
  - Revise el playbook `play_topup_hotspot.yaml` en busca de errores
  - Verifique los registros de Ansible

"Intención de pago ya procesada"

- **Causa:** El cliente intenta reutilizar el mismo pago (por ejemplo, actualizar después del éxito)

- **Solución:** Esto funciona como se diseñó para prevenir la doble facturación. El cliente debe iniciar una nueva recarga si es necesario.

"El monto de la intención de pago no coincide"

- **Causa:** Desajuste entre el cálculo de la UI y la validación del backend
- **Solución:**
  - Verifique que `VITE_TOPUP_PRICE_PER_DAY` coincida con la expectativa del backend (predeterminado $10)
  - Verifique que la configuración de la moneda sea consistente
  - Limpie la caché del navegador y vuelva a intentar

Monitoreo de Recargas

Ver Registros de Recarga:

Consulta la tabla `HotspotTopup` para ver todos los intentos de recarga:

```
SELECT
  hotspot_topup_id,
  service_uuid,
  days,
  topup_amount,
  status,
  payment_intent_id,
  created
FROM hotspot_topup
WHERE status = 'Failed'
ORDER BY created DESC;
```

Verificar Estado de Aprovisionamiento:

```
GET /crm/provision/provision_id/<provision_id>
```

Muestra el estado detallado del trabajo de aprovisionamiento de recarga.

Panel de Stripe:

Monitoree pagos, reembolsos y transacciones fallidas en su Panel de Stripe en <https://dashboard.stripe.com>

Consideraciones de Seguridad

Seguridad de Pagos:

- Todos los datos de tarjetas manejados por Stripe (conforme a PCI Nivel 1)
- No se almacenan datos de pago sensibles en la base de datos de OmniCRM
- Las Intenciones de Pago previenen cargos no autorizados
- Validación de monto tanto en el lado del cliente como en el servidor

Prevención de Fraude:

- La detección de Intenciones de pago duplicadas previene la doble facturación
- Seguimiento de dirección IP para correlación de uso
- La validación de IMSI asegura que la recarga vaya al servicio correcto
- Los reembolsos automáticos limitan la exposición financiera

Control de Acceso:

- El portal de recarga es público (por diseño - los clientes necesitan acceso)
- El endpoint de uso requiere identificación de servicio válida (IP o IMSI)
- La validación del backend previene recargas arbitrarias de servicios
- El administrador puede ver todos los registros de recarga a través de la interfaz del CRM

Mejores Prácticas

Para Operadores:

1. **Probar el flujo de reembolso** - Pruebe regularmente escenarios de fallo para asegurar que los reembolsos funcionen

2. **Monitorear recargas fallidas** - Configure alertas para altas tasas de fallos
3. **Mantener los playbooks simples** - Los playbooks de recarga deben ser rápidos y confiables
4. **Verificar conectividad de OCS** - Asegúrese de que la API de OCS sea siempre accesible
5. **Revisar precios** - Actualice `VITE_TOPUP_PRICE_PER_DAY` según sea necesario

Para Clientes:

1. **Marcar la URL de recarga** - Acceso rápido cuando sea necesario
2. **Guardar notificaciones de saldo bajo** - El SMS contiene un enlace directo
3. **Mantener el correo electrónico actualizado** - Los recibos se envían al correo electrónico registrado
4. **Verificar la caducidad antes de viajar** - Recargar antes de salir del área de cobertura

Para Desarrolladores:

1. **Manejar webhooks de Stripe** - Implementar controladores de webhook para actualizaciones de estado de pago
2. **Implementar idempotencia** - Siempre verifique payment_intent_id antes de procesar
3. **Registrar extensivamente** - Las fallas de recarga necesitan información detallada de solución de problemas
4. **Probar caminos de error** - Verifique que la automatización de reembolsos funcione correctamente
5. **Monitorear rendimiento** - El polling de aprovisionamiento debe completarse en <5 segundos

Documentación Relacionada

- `payments_process` - Procesamiento de pagos general

- `concepts_provisioning` - Descripción general del sistema de aprovisionamiento
- Guía del Sistema de Pagos - Detalles de integración del proveedor de pagos
- `payments_transaction` - Gestión de transacciones
- `payments_invoices` - Manejo de facturas

Glosario de Términos

Definiciones de la terminología utilizada en toda la documentación de OmniCRM. Los términos se enumeran alfabéticamente. Donde un concepto tiene su propia guía, se incluye un enlace a la página relevante.

A

Addon - Un extra opcional adjunto a un servicio existente, como un impulsor de datos o una característica de valor añadido. Los addons se facturan junto con el servicio principal en lugar de venderse como un producto independiente.

Ansible - El motor de automatización que utiliza OmniCRM para aprovisionar servicios. Las definiciones de productos activan los playbooks de Ansible que llevan a cabo la configuración subyacente. Ver Provisionamiento Basado en Ansible.

AuC (Centro de Autenticación) - La función que almacena la clave secreta material para una SIM y calcula los vectores de autenticación utilizados para verificar a un suscriptor en la red móvil. En OmniCRM, el AuC se llena como parte del aprovisionamiento de SIM al HSS.

C

CGRateS - El motor de tarifas y cargos en tiempo real integrado con OmniCRM. Califica el uso (voz, SMS, datos) contra destinos y aplica saldos, y se configura a través de perfiles de atributos, filtros de cuentas y tarifas de destino.

Cell Broadcast - Una capacidad de mensajería de uno a muchos utilizada para enviar alertas de emergencia e información pública a todos los dispositivos en un área geográfica definida. Ver Sistema de Cell Broadcast.

E

eSIM - Una SIM embebida que se aprovisiona por aire en lugar de insertar una tarjeta física. OmniCRM soporta la emisión e importación de eSIM. Ver Aprovisionamiento de Tarjetas SIM.

H

HSS (Servidor de Suscriptores en el Hogar) - La base de datos central de suscriptores de una red móvil. Contiene datos de suscripción y autenticación y responde a las solicitudes de la red cuando un suscriptor se conecta. OmniCRM escribe registros de suscriptores en el HSS durante el aprovisionamiento de SIM y soporta múltiples backends de HSS.

I

IMS (Subsistema Multimedia IP) - El marco de red central que entrega servicios de voz (VoLTE) y multimedia basados en IP. Los suscriptores se aprovisionan en IMS junto con el HSS para que los servicios de voz funcionen.

IMSI (Identidad Internacional de Suscriptor Móvil) - El identificador único vinculado a una SIM que la red utiliza para identificar a un suscriptor. OmniCRM mapea el IMSI a una cuenta de facturación durante el aprovisionamiento.

Inventario - El conjunto de recursos físicos y lógicos (tarjetas SIM, números móviles y equipos) que pueden ser asignados a los clientes. Los elementos se mueven a través de estados definidos a medida que son almacenados, asignados y activados. Ver Gestión de Inventario.

M

MSISDN (Número ISDN de Estación Móvil) - El número de teléfono marcable asociado con un suscriptor. El IMSI de una SIM es la identidad de la red, mientras que el MSISDN es el número que otras personas llaman.

O

OCS (Sistema de Carga en Línea) - El sistema responsable del control de crédito en tiempo real, autorizando y cobrando el uso a medida que ocurre. La integración de OCS permite a OmniCRM aplicar saldos y prevenir el gasto excesivo en servicios prepagos.

P

Playbook - Una rutina de automatización de Ansible nombrada que realiza una tarea de aprovisionamiento, como activar una SIM o configurar el equipo del cliente. El playbook de aprovisionamiento de un producto se ejecuta cuando se crea el servicio asociado. Ver Playbooks de Ansible.

PLMN (Red Pública Móvil Terrestre) - La red de un operador de red móvil, identificada por un código. La carga de datos utiliza destinos PLMN (la red a la que está conectado el cliente), mientras que la voz y SMS utilizan destinos geográficos.

Producto - Una definición vendible en el catálogo. Un producto especifica precios, comportamiento de aprovisionamiento y el/los servicio(s) que crea al ser comprado. Ver Productos y Servicios.

Factura Proforma - Una factura preliminar emitida antes de que el pago sea debido o antes de que se finalice una venta. Indica lo que se cobrará pero no es en sí misma una demanda de pago. Ver Facturas.

Aprovisionamiento - El proceso de convertir un producto comprado en un servicio funcionante realizando la configuración de red y sistema requerida, típicamente a través de playbooks de Ansible. Ver Aprovisionamiento.

R

RBAC (Control de Acceso Basado en Roles) - El modelo de permisos que gobierna lo que cada usuario puede ver y hacer en OmniCRM. Los permisos se

agrupan en roles, y los roles se asignan a los usuarios. Ver Control de Acceso Basado en Roles.

S

Portal de Autocuidado - El portal orientado al cliente donde los suscriptores ven su cuenta, uso, facturas y servicios sin contactar al soporte. Ver Portal de Autocuidado.

Auto-Registro - Un flujo que permite a un nuevo cliente registrarse y aprovisionar un servicio por sí mismo, incluyendo el aprovisionamiento de SIM, sin intervención del personal.

Servicio - Una instancia activa y aprovisionada de un producto en la cuenta de un cliente. Comprar un producto crea uno o más servicios, cada uno de los cuales puede ser gestionado, modificado o suspendido. Ver Productos y Servicios.

SIM (Módulo de Identidad del Suscriptor) - La tarjeta o módulo embebido que identifica a un suscriptor en la red móvil y almacena la clave de autenticación. Ver Aprovisionamiento de Tarjetas SIM.

T

Etiqueta - Una etiqueta aplicada a clientes o registros para agrupar y filtrarlos, por ejemplo, por segmento o campaña. Ver Etiquetas.

Transacción - Un único movimiento financiero en una cuenta, como un cargo, pago, crédito o reembolso. Las transacciones constituyen el historial de facturación de la cuenta. Ver Transacciones.

Números

2FA (Autenticación de Dos Factores) - Un paso de inicio de sesión adicional que requiere un código de un solo uso junto con la contraseña, protegiendo las

cuentas del personal contra credenciales comprometidas. Ver Autenticación de Dos Factores.

Guía Completa del Ciclo de Vida del Producto

Esta guía proporciona un recorrido completo del ciclo de vida del producto en OmniCRM, desde la creación de una definición de producto hasta la provisión de servicios, la adición de complementos y la desprovisión. Cubriremos la estrategia de precios, la integración de Ansible y proporcionaremos ejemplos del mundo real a lo largo del documento.

Visión General: El Viaje del Producto al Servicio

El ciclo de vida de un producto en OmniCRM sigue estas etapas:

1. **Definición del Producto** - El administrador crea una plantilla de producto con reglas de precios y provisión.
2. **Creación del Servicio** - El cliente ordena el producto, el sistema provisiona la instancia del servicio.
3. **Ciclo de Vida del Servicio** - El cliente utiliza el servicio, añade complementos/recargas, modifica el servicio.
4. **Desprovisión** - El servicio es terminado, los recursos son liberados.

Entendiendo los Precios: Mayorista vs Minorista

Cada producto y servicio en OmniCRM tiene dos dimensiones de precios: **mayorista** y **minorista**.

Costo Mayorista

El costo mayorista representa el costo real para entregar el servicio:

- Costos de infraestructura y ancho de banda
- Tarifas de licencia
- Costos de equipo
- Gastos operativos

Costo Minorista

El costo minorista es el monto cobrado al cliente.

Costos de Configuración

Tanto el mayorista como el minorista tienen variantes de costos de configuración para cargos de provisión únicos:

- `wholesale_setup_cost` - Su costo para provisionar
- `retail_setup_cost` - Monto cobrado al cliente por la activación

Ejemplo:

```
{
  "retail_cost": 15.00,
  "wholesale_cost": 5.00,
  "retail_setup_cost": 0.00,
  "wholesale_setup_cost": 1.00
}
```

Etapa 1: Creando una Definición de Producto

Los productos son plantillas que definen lo que se provisiona y cómo se cobra a los clientes.

Creando un Producto de SIM Móvil

Cree un producto de SIM móvil prepago con 20GB de datos por mes.

Paso 1: Navegar a la Gestión de Productos

Desde la interfaz de administración, vaya a **Productos → Crear Producto**.

Paso 2: Definir Información Básica

```
{
  "product_name": "Prepaid Mobile 20GB",
  "product_slug": "prepaid-mobile-20gb",
  "category": "standalone",
  "service_type": "mobile",
  "enabled": true,
  "icon": "fa-solid fa-sim-card",
  "comment": "Prepaid mobile SIM with 20GB data, unlimited calls & texts"
}
```

Explicaciones de Campos:

- `product_name` - Nombre visible para el cliente que se muestra en el catálogo.
- `product_slug` - Identificador seguro para URL utilizado en llamadas API y enlaces.
- `category` - "standalone" significa que esto crea un nuevo servicio (vs complemento/paquete).
- `service_type` - Agrupa productos relacionados, utilizado para filtrar complementos.
- `enabled` - Debe ser verdadero para que el producto sea ordenable.
- `icon` - Icono de FontAwesome mostrado en la interfaz de usuario.
- `comment` - Notas internas para referencia del personal.

Paso 3: Establecer Precios

```
{
  "retail_cost": 15.00,
  "wholesale_cost": 5.00,
  "retail_setup_cost": 0.00,
  "wholesale_setup_cost": 1.00,
  "contract_days": 30
}
```

Desglose de Precios:

- Ingreso mensual por cliente: £15.00
- Costo mensual para entregar: £5.00
- Margen de beneficio mensual: £10.00 (200% de margen, 67% de margen)
- Beneficio de configuración: -£1.00 (subvencionado para atraer clientes)
- Duración del contrato: 30 días (renovación mensual)

Paso 4: Definir Elegibilidad del Cliente

```
{
  "residential": true,
  "business": false,
  "customer_can_purchase": true,
  "available_from": "2025-01-01T00:00:00Z",
  "available_until": null
}
```

- Los clientes residenciales pueden ordenar.
- Los clientes comerciales no pueden (línea de producto diferente).
- Compra de autoservicio habilitada.
- Disponible desde el 1 de enero de 2025 en adelante.
- Sin fecha de finalización (oferta continua).

Paso 5: Configurar Renovación Automática

```
{
  "auto_renew": "prompt",
  "allow_auto_renew": true
}
```

- `"prompt"` - Preguntar al cliente si desea renovación automática al realizar la compra.
- `"true"` - Renovar automáticamente sin preguntar.
- `"false"` - Nunca renovar automáticamente (solo recarga manual).

- `allow_auto_renew: true` - El cliente puede habilitar/deshabilitar la renovación automática más tarde.

Paso 6: Especificar Requisitos de Inventario

Los requisitos de inventario definen qué recursos físicos o virtuales deben ser asignados al provisionar este producto. Este es un paso crítico que conecta su catálogo de productos con su Sistema de Gestión de Inventario.

```
{
  "inventory_items_list": "['SIM Card', 'Mobile Number']"
}
```

¿Qué Son los Artículos de Inventario?

Los artículos de inventario son recursos rastreables almacenados en el sistema de inventario de OmniCRM. Cada artículo tiene:

- **Tipo** - Definido por la Plantilla de Inventario (por ejemplo, "SIM Card", "Mobile Number", "Modem").
- **Atributos únicos** - Números de serie, direcciones MAC, números de teléfono, etc.
- **Estado** - En Stock, Asignado, Desactivado, etc.
- **Ubicación** - Ubicación física o lógica.

Cómo Funcionan los Requisitos de Inventario:

La `inventory_items_list` es una lista de Python (como una cadena) que contiene nombres de tipos de inventario. Cada nombre debe coincidir exactamente con un nombre de Plantilla de Inventario existente.

Ejemplo de Requisitos de Inventario:

```
# Producto de SIM móvil
inventory_items_list: "['SIM Card', 'Mobile Number']"

# Servicio de internet fijo
inventory_items_list: "['Modem Router', 'Static IP Address']"

# Servicio digital (sin artículos físicos)
inventory_items_list: "[]"

# Inalámbrico fijo con CPE
inventory_items_list: "['Fixed Wireless CPE', 'IPv4 Address',
'IPv6 Prefix']"
```

El Proceso del Selector de Inventario

Cuando un usuario provisiona un producto con requisitos de inventario, el sistema impone un proceso de selección obligatorio:

1. Botón de Provisión Clicado

Después de seleccionar el producto, el usuario hace clic en "Provisionar". En lugar de provisionar inmediatamente, el sistema verifica `inventory_items_list`.

2. Modal del Selector de Inventario Aparece

Si se requiere inventario, aparece un cuadro de diálogo modal con un menú desplegable separado para cada tipo de inventario:

3. Filtrando Inventario Disponible

El menú desplegable para cada tipo de inventario solo muestra artículos que son:

- **Tipo Correcto** - Coincide exactamente con el nombre de la plantilla de inventario.
- **Estado Disponible** - `item_state` es "Nuevo" o "En Stock" (no "Asignado" o "Dañado").
- **No Asignado** - `service_id` y `customer_id` son NULL.

- **En Stock en Ubicación** - Opcionalmente filtrado por ubicación de almacén/tienda.

Ejemplo de Opciones del Menú Desplegable:

Para el tipo de inventario "SIM Card", el menú desplegable podría mostrar:

Cada opción muestra:

- ID de inventario o número de referencia.
- Identificador principal (`itemtext1` - por ejemplo, ICCID para SIM, número para teléfono).
- Ubicación actual (`item_location`).

4. Selección Requerida para Proceder

Regla Crítica: La provisión NO puede proceder sin seleccionar todos los artículos de inventario requeridos.

- El botón "Continuar" está deshabilitado hasta que todos los menús desplegables tengan selecciones.
- El usuario debe seleccionar un artículo para cada tipo de inventario.
- El sistema valida las selecciones antes de proceder.

5. Inventario Seleccionado Pasado a Ansible

Una vez que el usuario hace clic en "Continuar", los IDs de inventario seleccionados se pasan al playbook de Ansible como variables:

```
# Usuario seleccionado:
# - ID de inventario de SIM Card: 5001
# - ID de inventario de Mobile Number: 5002

# Variables pasadas a Ansible:
{
    "product_id": 42,
    "customer_id": 123,
    "SIM Card": 5001,           # ID de Inventario
    "Mobile Number": 5002,       # ID de Inventario
    "access_token": "eyJ..."
}
```

Nota: El nombre de la variable coincide exactamente con el tipo de inventario. El playbook usa `hostvars[inventory_hostname]['SIM Card']` para acceder al ID de inventario.

6. Playbook Recupera Detalles Completos del Inventario

El playbook de Ansible utiliza el ID de inventario para recuperar detalles completos:

```
- name: Get SIM Card details from inventory
  uri:
    url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ hostvars[inventory_hostname]['SIM
Card'] }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
  register: api_response_sim

- name: Extract ICCID and IMSI
  set_fact:
    iccid: "{{ api_response_sim.json.itemtext1 }}"
    imsi: "{{ api_response_sim.json.itemtext2 }}"
```

Ahora el playbook tiene todos los detalles de la SIM (ICCID, IMSI, etc.) para provisionar al suscriptor en el HSS.

7. Estado del Inventario Cambiado a "Asignado"

Después de que se crea el registro del servicio, el playbook actualiza el inventario para vincularlo al servicio:

```
- name: Assign SIM Card to Service
  uri:
    url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ hostvars[inventory_hostname]['SIM
Card'] }}"
    method: PATCH
    body:
      {
        "service_id": "{{
service_creation_response.json.service_id }}",
        "customer_id": "{{ customer_id }}",
        "item_state": "Assigned"
      }
```

Importante: La asignación de inventario ocurre **durante la ejecución del playbook** como una tarea específica, NO cuando se hace clic en el botón de provisión. Esto significa:

- **Riesgo de Doble Asignación**: Entre hacer clic en "Provisionar" y que el inventario sea asignado, otro usuario podría teóricamente seleccionar el mismo artículo de inventario.
- **Mejor Práctica**: Para operaciones de alto volumen, implemente bloqueo de inventario o use transacciones de base de datos.
- **Rollback en Caso de Fallo**: Si el playbook falla antes de la asignación de inventario, el inventario permanece no asignado y disponible para reutilización.

¿Por Qué No Asignar Antes?

El inventario no se asigna cuando se hace clic en "Provisionar" porque:

1. **ID de Servicio Necesario**: El `service_id` no existe hasta que el servicio es creado en el playbook.

2. **Simplicidad de Rollback**: Si la provisión falla temprano (por ejemplo, falla en la creación de la cuenta OCS), no es necesario limpiar el inventario.
3. **Flexibilidad**: El playbook puede decidir no asignar inventario basado en lógica condicional.

Manejo de Provisiones Fallidas:

Cuando una provisión falla después de que el inventario ha sido asignado, el bloque de rescate debe liberar el inventario:

```
rescue:
  - name: Release inventory on failure
    uri:
      url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ hostvars[inventory_hostname]['SIM
Card'] }}"
      method: PATCH
      body:
        {
          "service_id": null,
          "customer_id": null,
          "item_state": "In Stock"
        }
    when: service_id is defined  # Solo si el servicio fue creado
```

Esto asegura que el inventario no quede en un estado "Asignado" para un servicio inexistente o fallido.

Cuando la Lista de Inventario Está Vacía

Si `inventory_items_list: "[]"` (lista vacía), el selector de inventario se omite por completo y la provisión procede inmediatamente. Esto es común para:

- **Productos digitales** - Licencias de software, cuentas de VPN.
- **Complementos de servicio** - Recargas de datos que no necesitan nuevo hardware.
- **Servicios virtuales** - Que no consumen recursos rastreables.

Ejemplo: Un complemento de "5GB Data Boost" tiene `inventory_items_list: "[]"` porque solo agrega saldo a un servicio existente sin necesidad de nuevo

hardware.

Configuración de Plantilla de Inventario

Antes de usar un tipo de inventario en `inventory_items_list`, debe crear la Plantilla de Inventario:

1. Navegue a **Administración → Inventario → Plantillas**.
2. Cree una plantilla con el nombre exacto (por ejemplo, "SIM Card").
3. Defina los campos:
   - `itemtext1_label`: "ICCID".
   - `itemtext2_label`: "IMSI".
   - `itemtext3_label`: "PUK Code".
4. Agregue artículos de inventario de este tipo al stock.

Para obtener detalles completos sobre cómo crear y gestionar plantillas de inventario, consulte Gestión de Inventario.

Múltiples Artículos del Mismo Tipo

Si bien `inventory_items_list` es un array, tener tipos duplicados (por ejemplo, `"['SIM Card', 'SIM Card']"`) **no se recomienda** ya que puede causar confusión en la interfaz de usuario y en la nomenclatura de variables del playbook.

Para escenarios que requieren múltiples artículos similares:

Opción 1: Crear nombres de plantillas de inventario distintos

```
# Servicio de teléfono Dual-SIM
inventory_items_list: "['Primary SIM Card', 'Secondary SIM Card',
'Mobile Number']"
```

Cree plantillas separadas: "Primary SIM Card" y "Secondary SIM Card" con los mismos campos pero diferentes nombres.

Opción 2: Usar un solo artículo de inventario agrupado

```
# Kit Dual-SIM
inventory_items_list: "['Dual SIM Kit', 'Mobile Number']"
```

Donde la plantilla de inventario "Dual SIM Kit" tiene campos para ambas SIMs (`itemtext1`: ICCID Primario, `itemtext2`: ICCID Secundario, etc.).

Escenarios Comunes de Inventario

Servicio Móvil:

```
inventory_items_list: "['SIM Card', 'Mobile Number']"
```

- SIM Card: Física o eSIM con ICCID/IMSI.
- Mobile Number: Número de teléfono (MSISDN).

Internet Fijo:

```
inventory_items_list: "['Modem Router', 'Static IP Address']"
```

- Modem Router: Dispositivo CPE con dirección MAC.
- Static IP Address: IPv4 del grupo de direcciones.

Inalámbrico Fijo:

```
inventory_items_list: "['Fixed Wireless CPE', 'IPv4 Address',
'IPv6 Prefix']"
```

- CPE: Equipo de las instalaciones del cliente (antena, módem).
- IPv4: Dirección IP pública.
- IPv6 Prefix: Prefijo /56 o /64.

Nota: Las citas y la programación **no** son artículos de inventario. Utilice sistemas de programación/calendario separados para las citas de instalación.

Servicio VoIP:

```
inventory_items_list: "['DID Number']"
```

- DID Number: Número de teléfono de Marcación Directa.

Nota: Los nombres de usuario SIP, contraseñas y configuraciones de cuenta son **generados programáticamente** por el playbook de provisión, no seleccionados del inventario.

GPON/Fibra:

```
inventory_items_list: "['ONT Device', 'GPON Port', 'IPv4 Address',
'Fiber Drop Cable']"
```

- ONT Device: Terminal de Red Óptica con número de serie.
- GPON Port: Puerto específico en OLT con conexión de fibra.
- IPv4 Address: IP pública o privada.
- Fiber Drop Cable: Cable de fibra física de la calle a las instalaciones (rastreado para gestión de activos).

Alquiler de Equipos:

```
inventory_items_list: "['Rental Modem']"
```

- Rastrear qué módem está con qué cliente.
- Importante para recuperar equipos en caso de cancelación.

Por Qué Importan los Requisitos de Inventario

1. Prevenir Doble Asignación

Sin seguimiento de inventario, podría accidentalmente:

- Asignar la misma tarjeta SIM a dos clientes.
- Asignar la misma dirección IP a múltiples servicios.
- Enviar el mismo número de serie de equipo a diferentes ubicaciones.

El selector de inventario asegura que cada artículo esté asignado a exactamente un servicio.

2. Registro de Auditoría

La asignación de inventario crea un registro de auditoría completo:

- Qué tarjeta SIM está con qué cliente.
- Cuándo fue asignada.
- Qué servicio está utilizando qué número de teléfono.
- Historial de equipos (quién lo tuvo, cuándo, para qué servicio).

3. Planificación de Recursos

Rastrear niveles de inventario:

- Alertar cuando las tarjetas SIM se estén agotando.
- Reordenar antes de que se agote el stock.
- Planificar horarios de técnicos basados en la disponibilidad de CPE.
- Gestionar la asignación de espacio de direcciones IP.

4. Seguimiento de Costos

Vincular el costo mayorista a un artículo específico:

- Rastrear el costo de cada tarjeta SIM.
- Calcular la depreciación del equipo.
- Identificar artículos perdidos o robados.
- COGS (Costo de Bienes Vendidos) preciso.

5. Desprovisión

Cuando se cancela el servicio, el inventario puede ser:

- Liberado de nuevo al stock (tarjetas SIM, módems).
- Retirado (equipos dañados).
- Devuelto al proveedor (equipos de alquiler).

- Conservado por un período de gracia (números de teléfono antes de la liberación).

Resolución de Problemas con el Selector de Inventario

Problema: Aparece el mensaje "No hay inventario disponible".

Causas:

- No existen artículos de inventario del tipo requerido en la base de datos.
- Todos los artículos ya están "Asignados" a otros servicios.
- Los artículos están marcados como "Dañados" o "Fuera de Servicio".
- El nombre de la plantilla de inventario no coincide exactamente (sensible a mayúsculas y minúsculas).

Solución:

1. Verifique que la plantilla de inventario exista: **Administración** → **Inventario** → **Plantillas**.
2. Verifique que el nombre de la plantilla coincida exactamente (incluyendo espacios, mayúsculas y minúsculas).
3. Agregue artículos de inventario de este tipo: **Administración** → **Inventario** → **Agregar Artículo**.
4. Verifique que los artículos estén en estado "Nuevo" o "En Stock".
5. Verifique que los artículos no estén ya asignados (`service_id` debe ser NULL).

Problema: El selector de inventario no aparece.

Causas:

- `inventory_items_list` está vacío: `"[]"`.
- `inventory_items_list` es NULL o no está configurado.
- La categoría del producto es "complemento" y hereda el inventario del servicio principal.

Solución:

- Si se necesita inventario, configure `inventory_items_list: "['Type1', 'Type2']"`.
- Verifique que la definición del producto se haya guardado correctamente.
- Verifique la respuesta de la API para que el producto incluya `inventory_items_list`.

Problema: El playbook falla con "inventario no encontrado".

Causas:

- El playbook hace referencia al nombre de variable incorrecto.
- El ID de inventario no se pasó correctamente.
- El inventario fue eliminado entre la selección y la provisión.

Solución:

- Verifique que el playbook use la variable correcta: `hostvars[inventory_hostname]['SIM Card']`.
- Verifique que la variable sea un entero: `{{ hostvars[inventory_hostname]['SIM Card'] | int }}`.
- Agregue manejo de errores en el playbook para inventario faltante.

Consulte Gestión de Inventario para obtener detalles completos sobre cómo crear plantillas, agregar artículos y gestionar niveles de stock.

Paso 7: Definir Características y Términos

Las características y términos son contenido de marketing y legal visible para el cliente que ayuda a los clientes a entender lo que están comprando y las obligaciones involucradas.

```
{
  "features_list": "20GB High-Speed Data. Unlimited Calls & Texts. EU Roaming Included. No Contract. 30-Day Expiry",
  "terms": "Credit expires after 30 days. Data, calls, and texts valid only within expiry period. Fair use policy applies. See website for full terms."
}
```

Propósito y Valor Comercial

Lista de Características - Marketing y Ventas:

La lista de características cumple múltiples funciones comerciales críticas:

1. **Diferenciación del Producto** - Ayuda a los clientes a comparar rápidamente productos y elegir el adecuado.
    - "Prepaid Mobile 20GB" vs "Prepaid Mobile 50GB" - las características muestran claramente la diferencia.
    - Sin características, los clientes solo ven el precio, perdiendo la propuesta de valor.
2. **Comunicación de Marketing** - Puntos de venta clave mostrados de manera prominente.
    - "EU Roaming Included" atrae a viajeros internacionales.
    - "No Contract" atrae a clientes que evitan compromisos.
    - Las características impulsan las decisiones de compra.
3. **Expectativas del Cliente** - Establece expectativas claras sobre lo que está incluido.
    - Reduce las llamadas de soporte ("¿Esto incluye llamadas?" → claramente listado).
    - Previene malentendidos y solicitudes de reembolso.
    - Genera confianza a través de la transparencia.
4. **Autoservicio** - Permite a los clientes seleccionar productos apropiados por sí mismos.
    - El cliente lee las características, entiende la oferta, toma una decisión informada.
    - Reduce la necesidad de explicación por parte del personal de ventas.
    - Acelera el proceso de compra.
5. **SEO y Descubribilidad** - Las características pueden ser indexadas para búsqueda.
    - El cliente busca "plan móvil de llamadas ilimitadas" → el producto aparece.
    - Mejora la capacidad de búsqueda del catálogo de productos.

Términos y Condiciones - Legal y Cumplimiento:

Los términos cumplen propósitos legales y operativos:

1. **Protección Legal** - Protege a la empresa de disputas y responsabilidad.
    - "El crédito expira después de 30 días" - el cliente no puede exigir un reembolso a los 31 días.
    - "Se aplica política de uso justo" - previene abusos (compartir toda la oficina en un plan móvil).
    - Crea un acuerdo vinculante.
2. **Gestión de Expectativas** - Previene la insatisfacción del cliente.
    - "Válido solo dentro del período de caducidad" - el cliente conoce la fecha límite de uso.
    - "No puede ser reembolsado" (para complementos) - previene compras fraudulentas.
    - Reduce contracargos y quejas.
3. **Cumplimiento Regulatorio** - Cumple con requisitos legales.
    - Las leyes de protección al consumidor requieren términos claros.
    - Las regulaciones de telecomunicaciones exigen divulgación.
    - Se pueden hacer referencia a términos de GDPR/privacidad.
4. **Límites Operativos** - Define el alcance y las limitaciones del servicio.
    - "Sujeto a cobertura de red" - no es responsable por zonas muertas.
    - "La velocidad puede variar" - gestiona expectativas sobre velocidades "hasta".
    - "El equipo debe ser devuelto" - asegura la recuperación del equipo de alquiler.
5. **Registro de Auditoría** - Prueba que el cliente fue informado.
    - El cliente aceptó los términos al realizar la compra.
    - El sistema registra la marca de tiempo de aceptación.
    - Defendible en disputas o procedimientos legales.

Ejemplo del Mundo Real:

El cliente compra un plan de "Llamadas y Mensajes Ilimitados" y luego lo usa para telemarketing (10,000 llamadas/día). Sin términos:

- Cliente: "¡Dijiste ilimitado!"
- Proveedor: "Nos referíamos a uso personal..."

- Cliente: "¡Eso no es lo que publicitaste!"
- Resultado: Disputa, posible queja ante el regulador, daño a la marca.

Con términos: "Se aplica política de uso justo. El servicio es solo para uso personal. Uso comercial prohibido."

- Proveedor: Señala los términos que el cliente aceptó.
- El cliente no puede alegar ignorancia.
- Base legal para suspender el servicio.
- Disputa resuelta a favor del proveedor.

Formato de la Lista de Características:

Entender el formato correcto es crítico porque **el formato incorrecto rompe la visualización de la interfaz de usuario**. Las características pueden aparecer como una larga cadena en lugar de puntos de viñeta, o no mostrarse en absoluto.

El campo `features_list` se puede formatear de dos maneras:

Opción 1: Cadena Separada por Puntos (Recomendada)

Las características están separadas por un punto y un espacio (". "). La interfaz de usuario se divide en este delimitador y renderiza cada característica como un punto de viñeta.

¿Por qué este formato?

- Simple de editar - solo escriba características con puntos entre ellas.
- Sin caracteres especiales que escapar.
- Funciona de manera confiable en todos los componentes de la interfaz de usuario.
- Fácil de actualizar sin romper la sintaxis JSON.

Correcto vs Incorrecto:

Opción 2: Cadena de Array JSON

```
"['20GB High-Speed Data', 'Unlimited Calls & Texts', 'EU Roaming Included']"
```

La interfaz de usuario también puede analizar arrays JSON. Tenga en cuenta que esto es una **cadena que contiene JSON**, no un array JSON real en la base de datos.

¿Por qué existe este formato?

- Permite características con puntos en ellas (por ejemplo, "Hasta 100Mbps. Sujeto a disponibilidad.").
- La generación programática desde scripts/API es más fácil.
- Importado de catálogos de productos externos que utilizan arrays.

Importante: Esto debe ser sintaxis válida de lista de Python como cadena. Comillas simples alrededor de cada elemento, comillas dobles alrededor de toda la cadena.

¿Qué Formato Usar?

- **Separado por puntos** - Para la creación manual de productos en la interfaz de usuario (más simple, menos propenso a errores).
- **Array JSON** - Para la creación de productos basada en API/script (más robusto para características complejas).

Ambos formatos producen una salida idéntica en la interfaz de usuario; solo afectan cómo ingresa los datos.

Dónde Aparecen las Características en la Interfaz de Usuario:

1. Catálogo de Productos (Vista del Cliente)

Cuando los clientes navegan por los productos disponibles, las características se muestran como puntos de viñeta en cada tarjeta de producto:

2. Página de Detalles del Producto

Hacer clic en "Ver Detalles" muestra toda la información del producto, incluyendo:

- Nombre del producto e icono.
- Precios (costo mensual, costo de configuración).
- Lista completa de características (puntos de viñeta).
- Términos y condiciones (ver más abajo).
- Disponibilidad y elegibilidad.

3. Confirmación de Provisión

Durante la provisión, las características se muestran para que el usuario las revise antes de confirmar:

> Características: • 20GB High-Speed Data • Unlimited Calls & Texts • EU Roaming Included • No Contract • 30-Day Expiry
>
> Costo: £15.00/mes Configuración: £0.00
>
> [Cancelar] [Confirmar y Proveer]

4. Detalles del Servicio (Después de la Provisión)

Después de que el servicio esté activo, las características se muestran en la página de detalles del servicio para referencia del cliente.

Formato de Términos y Condiciones:

El campo `terms` es texto plano que puede incluir saltos de línea:

Dónde Aparecen los Términos en la Interfaz de Usuario:

1. Página de Detalles del Producto

Los términos se muestran en una sección colapsada que se expande al hacer clic:

2. Confirmación de Pedido

Durante la provisión, se requiere que el usuario acepte los términos mediante una casilla de verificación:

> El botón [Provisionar] está deshabilitado hasta que se marque.

3. Facturas

Los términos del servicio pueden incluirse en las facturas como notas al pie para mayor claridad.

Mejores Prácticas:

- **Características:** Mantener concisas (menos de 50 caracteres cada una), centrarse en los beneficios clave.
- **Términos:** Incluir requisitos legales críticos, políticas de expiración, políticas de uso justo.
- **Ambos:** Actualizar cuando cambie el producto para mantener informados a los clientes.

Paso 8: Vincular el Playbook de Provisión de Ansible

```
{
  "provisioning_play": "play_local_mobile_sim",
  "provisioning_json_vars": "{
    \"days\": 30,
    \"data_gb\": 20,
    \"voice_minutes\": \"unlimited\",
    \"sms_count\": \"unlimited\"
  }"
}
```

- `provisioning_play` - Nombre del playbook de Ansible (sin la extensión .yaml).
- `provisioning_json_vars` - Variables predeterminadas pasadas al playbook.
- El playbook debe existir en: `OmniCRM-API/Provisioners/plays/play_local_mobile_sim.yaml`.

Definición Completa del Producto

```
{
  "product_name": "Prepaid Mobile 20GB",
  "product_slug": "prepaid-mobile-20gb",
  "category": "standalone",
  "service_type": "mobile",
  "enabled": true,
  "icon": "fa-solid fa-sim-card",
  "comment": "Prepaid mobile SIM with 20GB data, unlimited calls & texts",

  "retail_cost": 15.00,
  "wholesale_cost": 5.00,
  "retail_setup_cost": 0.00,
  "wholesale_setup_cost": 1.00,
  "contract_days": 30,

  "residential": true,
  "business": false,
  "customer_can_purchase": true,
  "available_from": "2025-01-01T00:00:00Z",
  "available_until": null,

  "auto_renew": "prompt",
  "allow_auto_renew": true,

  "inventory_items_list": "['SIM Card', 'Mobile Number']",

  "features_list": "[
    '20GB High-Speed Data',
    'Unlimited Calls & Texts',
    'EU Roaming Included',
    'No Contract',
    '30-Day Expiry'
  ]",
  "terms": "Credit expires after 30 days. Data, calls, and texts valid only within expiry period. Fair use policy applies.",

  "provisioning_play": "play_local_mobile_sim",
  "provisioning_json_vars": "{
    \"days\": 30,
    \"data_gb\": 20,
    \"voice_minutes\": \"unlimited\",
    \"sms_count\": \"unlimited\"
```

```
  }"
}
```

Creando un Producto Complementario

Los complementos mejoran o modifican servicios existentes. Vienen en dos tipos: **complementos virtuales** (sin recursos físicos) y **complementos de hardware** (requieren inventario).

Ejemplo 1: Complemento Virtual (5GB Data Boost)

Un complemento digital que agrega datos a un servicio móvil existente:

```
{
  "product_name": "5GB Data Boost",
  "product_slug": "5gb-data-boost",
  "category": "addon",
  "service_type": "mobile",
  "enabled": true,
  "icon": "fa-solid fa-plus",
  "comment": "Add 5GB extra data to existing mobile service",

  "retail_cost": 5.00,
  "wholesale_cost": 1.50,
  "retail_setup_cost": 0.00,
  "wholesale_setup_cost": 0.00,
  "contract_days": 0,

  "residential": true,
  "business": true,
  "customer_can_purchase": true,

  "auto_renew": "false",
  "allow_auto_renew": false,

  "inventory_items_list": "[]",
  "relies_on_list": "",

  "features_list": "5GB High-Speed Data. Valid for 7 Days",
  "terms": "Data expires after 7 days or when exhausted. Cannot be
refunded.",

  "provisioning_play": "play_topup_charge_then_action",
  "provisioning_json_vars": "{
    \"data_gb\": 5,
    \"days\": 7
  }"
}
```

Ejemplo 2: Complemento de Hardware (Alquiler de Módem)

Un complemento que proporciona equipo físico para un servicio de fibra existente:

```
{
  "product_name": "WiFi 6 Modem Rental",
  "product_slug": "wifi6-modem-rental",
  "category": "addon",
  "service_type": "internet",
  "enabled": true,
  "icon": "fa-solid fa-router",
  "comment": "Add WiFi 6 modem to fiber service - rental",

  "retail_cost": 10.00,
  "wholesale_cost": 3.00,
  "retail_setup_cost": 0.00,
  "wholesale_setup_cost": 45.00,
  "contract_days": 30,

  "residential": true,
  "business": true,
  "customer_can_purchase": true,

  "auto_renew": "true",
  "allow_auto_renew": true,

  "inventory_items_list": "['Rental Modem']",
  "relies_on_list": "",

  "features_list": "WiFi 6 (802.11ax). Dual-band 2.4GHz + 5GHz. Up
to 40 devices. Parental controls",
  "terms": "Equipment rental. Must be returned on service
cancellation or £150 replacement fee applies. Equipment remains
property of provider.",

  "provisioning_play": "play_addon_assign_modem",
  "provisioning_json_vars": "{
    \"device_type\": \"modem_router\",
    \"requires_configuration\": true
  }"
}
```

Diferencias Clave para Complementos:

- `category: "addon"` - Aplicado a un servicio existente, no independiente.

- `contract_days: 0` (virtual) o `30` (alquiler recurrente) - Frecuencia de facturación.
- `inventory_items_list: "[]"` (virtual) o `"['Rental Modem']"` (hardware) - Recursos físicos.
- `auto_renew: "false"` (una sola vez) o `"true"` (alquiler) - Comportamiento recurrente.
- `relies_on_list: ""` - Vacío significa que se aplica a cualquier servicio de tipo de servicio coincidente.

Por Qué los Complementos de Hardware Necesitan Inventario:

Los complementos de hardware requieren `inventory_items_list` porque:

1. **Rastrear Equipos** - Saber qué módem está con qué cliente.
2. **Prevenir Agotamientos de Stock** - No se puede provisionar el complemento si no hay módems en stock.
3. **Recuperación** - Cuando el cliente cancela, saber qué equipo recuperar.
4. **Seguimiento de Costos** - Vincular el costo mayorista a un número de serie específico.
5. **Depreciación** - Rastrear el valor del equipo durante el período de alquiler.
6. **Garantía** - Identificar unidades defectuosas por número de serie.

Flujo de Provisión de Complementos con Inventario:

Cuando un cliente agrega "Alquiler de Módem WiFi 6" a su servicio de fibra:

1. **Complemento Seleccionado** - El cliente hace clic en "Agregar al Servicio".
2. **Aparece el Selector de Inventario** - Al igual que los servicios independientes:
3. **Pago Procesado** - Se cobra £10.00 de alquiler mensual.
4. **Módem Asignado** - Inventario actualizado:
   - `service_id`: Vinculado al servicio de fibra.
   - `customer_id`: Vinculado al cliente.
   - `item_state`: "Asignado".
5. **Envío Activado** - El sistema de cumplimiento es notificado para enviar el módem.

6. **Instalación** - El cliente recibe el módem, lo conecta al ONT.
7. **Facturación Recurrente** - Se cobra £10/mes hasta que se cancele el complemento.

Desprovisión de Complementos de Hardware:

Cuando el cliente cancela el alquiler del módem:

1. **Cancelación Iniciada** - El cliente hace clic en "Eliminar Complemento".
2. **Proceso de Devolución Iniciado**:
   - Se envía un correo electrónico con instrucciones de devolución.
   - Se genera una etiqueta de envío prepagada.
   - Período de gracia de 14 días antes de la penalización.
3. **Equipo Devuelto**:
   - Inventario actualizado: `item_state` = "En Stock" (después de la renovación).
   - O `item_state` = "Dañado" (si es defectuoso).
   - Vinculado al siguiente cliente una vez renovado.
4. **Sin Devolución**:
   - Después de 14 días, se cobra una tarifa de reemplazo de £150.
   - Inventario marcado: `item_state` = "Perdido".
   - Costo mayorista (£45) + valor de reemplazo recuperado.

Precios para Complementos:

Los complementos pueden tener precios diferentes a los servicios independientes:

- Los complementos virtuales típicamente no tienen costos de configuración.
- Los complementos de hardware pueden tener costos de configuración mayorista para el equipo.
- Los complementos de alquiler recurrente usan `contract_days` para la frecuencia de facturación.

Etapa 2: El Proceso de Provisión

Cuando un cliente ordena el producto "Prepaid Mobile 20GB", OmniCRM orquesta la provisión a través de Ansible.

Diagrama de Flujo de Provisión

El Cliente Ordena → Selección de Inventario → Trabajo de Provisión Creado ↓ ↓ Pago Autorizado ← Variables Ensambladas ← Playbook de Ansible Ejecutado ↓ ↓ Registro de Servicio Creado → Configuración de Cuenta OCS → Inventario Asignado → Servicio Activo

Flujo de Provisión Paso a Paso

1. El Cliente Inicia el Pedido

Desde la página del cliente:

- El personal hace clic en "Agregar Servicio".
- Selecciona "Prepaid Mobile 20GB" del carrusel de productos.
- Se muestran los detalles del producto y los precios.

2. Selección de Inventario

El sistema solicita el inventario requerido:

- **SIM Card** - El menú desplegable muestra las SIM disponibles en stock.
  - Ejemplo: "SIM-00123 - ICCID: 8944...".
- **Mobile Number** - El menú desplegable muestra los números de teléfono disponibles.
  - Ejemplo: "+44 7700 900123".

El personal o el cliente selecciona artículos del inventario disponible.

3. Confirmación de Precios

El sistema muestra el precio final:

- Costo de configuración: £0.00 (activación gratuita).
- Costo mensual: £15.00.
- Total a pagar hoy: £15.00 (primer mes).
- Fecha de renovación: 30 días a partir de hoy.

Si se habilita la solicitud de renovación automática, el cliente elige:

☐ Renovar automáticamente este servicio cada 30 días.

4. Botón de Provisión Clicado

Cuando se hace clic en "Provisionar", la API:

- Crea un registro de `Provision` con estado "Ejecutando" (status=1).
- Fusiona variables del producto + solicitud + selecciones de inventario.
- Genera un hilo en segundo plano para ejecutar el playbook de Ansible.
- Devuelve `provision_id` a la interfaz de usuario para el seguimiento del estado.

5. Variables Ensambladas

El sistema fusiona variables de múltiples fuentes:

Del Producto:

```
{
  "days": 30,
  "data_gb": 20,
  "voice_minutes": "unlimited",
  "sms_count": "unlimited"
}
```

De la Solicitud:

```
{
  "product_id": 42,
  "customer_id": 123,
  "SIM Card": 5001,
  "Mobile Number": 5002
}
```

Agregado por el Sistema:

```
{
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "initiating_user": 7
}
```

Variables Finales Pasadas a Ansible:

```
{
    "product_id": 42,
    "customer_id": 123,
    "SIM Card": 5001,
    "Mobile Number": 5002,
    "days": 30,
    "data_gb": 20,
    "voice_minutes": "unlimited",
    "sms_count": "unlimited",
    "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
    "initiating_user": 7
}
```

6. Ejecución del Playbook de Ansible

El playbook `play_local_mobile_sim.yaml` se ejecuta con estas variables.

Entendiendo el Playbook de Provisión de Ansible

Examine un playbook de provisión real para entender qué sucede detrás de escena.

Ejemplo de Playbook de Provisión de SIM Móvil

Ubicación: `OmniCRM-API/Provisioners/plays/play_local_mobile_sim.yaml`

Estructura de Alto Nivel:

```yaml
- name: Mobile SIM Provisioning
  hosts: localhost
  gather_facts: no
  become: False

  tasks:
    - name: Main block
      block:
        # 1. Load configuration
        # 2. Fetch product details from API
        # 3. Fetch customer details from API
        # 4. Fetch inventory details from API
        # 5. Create account in OCS (CGRateS)
        # 6. Add balances and allowances to OCS
        # 7. Create service record in CRM
        # 8. Assign inventory to service
        # 9. Record transactions
        # 10. Send welcome notifications

      rescue:
        # Rollback on failure
        # - Remove OCS account
        # - Release inventory
        # - Log error
```

Recorrido Detallado del Playbook:

Tarea 1: Cargar Configuración

```
- name: Include vars of crm_config
  ansible.builtin.include_vars:
    file: "../../crm_config.yaml"
    name: crm_config
```

Carga la configuración del sistema, incluyendo:

- URL y credenciales de OCS/CGRateS.
- URL base de CRM.
- Configuración del inquilino.

Tarea 2: Obtener Detalles del Producto

```
- name: Get Product information from CRM API
  uri:
    url: "{{ crm_config.crm.base_url }}/crm/product/product_id/{{
product_id }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
    return_content: yes
  register: api_response_product
```

Qué Hace Esto:

- Llama a `GET /crm/product/product_id/42`.
- Recupera la definición completa del producto.
- Almacena en la variable `api_response_product`.

Por Qué: Aunque tenemos `provisioning_json_vars` del producto, recuperamos el producto completo para obtener:

- Precios más recientes (pueden haber cambiado desde que se inició el pedido).
- Nombre del producto para nombrar el servicio.
- Lista de características para documentación.

- Costos mayoristas para seguimiento de márgenes.

Tarea 3: Establecer Hechos del Paquete

```
- name: Set package facts
  set_fact:
    package_name: "{{ api_response_product.json.product_name }}"
    monthly_cost: "{{ api_response_product.json.retail_cost }}"
    setup_cost: "{{ api_response_product.json.retail_setup_cost
}}"
```

Extrae valores de uso común en variables simples para mejorar la legibilidad.

Tarea 4: Obtener Detalles del Inventario

```
- name: Get SIM information from CRM API
  uri:
    url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ hostvars[inventory_hostname]['SIM
Card'] }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
  register: api_response_sim

- name: Set IMSI from Inventory response
  set_fact:
    imsi: "{{ api_response_sim.json.itemtext2 }}"
    iccid: "{{ api_response_sim.json.itemtext1 }}"
```

Qué Hace Esto:

- Busca el ID de inventario de la tarjeta SIM 5001.
- Recupera detalles de la SIM:
  - `itemtext1` = ICCID (número de tarjeta SIM).
  - `itemtext2` = IMSI (identidad del suscriptor).
- Hace lo mismo para el número de teléfono móvil (recupera el número de teléfono).

Por Qué Esto Importa:

- IMSI es necesario para provisionar al suscriptor en el HSS (Servidor de Suscriptores Local).
- ICCID se registra en las notas del servicio para solucionar problemas.
- El número de teléfono (MSISDN) se muestra al cliente y se utiliza para el enrutamiento.

Tarea 5: Generar UUID del Servicio

```
- name: Generate UUID Fact
  set_fact:
    uuid: "{{ 99999999 | random | to_uuid }}"

- name: Set Service UUID
  set_fact:
    service_uuid: "Local_Mobile_SIM_{{ uuid[0:8] }}"
```

Qué Hace Esto:

- Genera un UUID aleatorio.
- Crea un service_uuid como `Local_Mobile_SIM_a3f2c1d8`.

Por Qué:

- El UUID del servicio es el identificador único en OCS/CGRateS.
- Se utiliza para todas las operaciones de carga.
- Debe ser globalmente único entre todos los servicios.

Tarea 6: Crear Cuenta en OCS

```
- name: Create account in OCS
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    headers:
      Content-Type: "application/json"
    body:
      {
        "method": "ApierV2.SetAccount",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "ActionPlanIds": [],
          "ExtraOptions": {
            "AllowNegative": false,
            "Disabled": false
          },
          "ReloadScheduler": true
        }]
      }
  register: ocs_create_response
```

Qué Hace Esto:

- Llama a la API JSON-RPC de CGRateS.
- Crea una nueva cuenta con service_uuid.
- Establece la cuenta como activa (no deshabilitada).
- Previene saldo negativo (modo prepago).

Por Qué:

- La cuenta OCS es donde ocurre toda la carga.
- Los saldos (datos, voz, SMS, dinero) se almacenan aquí.
- El uso se rastrea y se califica en tiempo real.

Tarea 7: Agregar Saldo de Datos

```
- name: Add 20GB Data Balance
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
          "Tenant": "{{ crm_config.ocs.ocsTenant }}",
          "Account": "{{ service_uuid }}",
          "BalanceType": "*data",
          "Balance": {
            "ID": "DATA_20GB_Monthly",
            "Value": 21474836480,
            "ExpiryTime": "+720h",
            "Weight": 10,
            "DestinationIDs": "*any"
          }
        }]
      }
```

Qué Hace Esto:

- Agrega 20GB de saldo de datos a la cuenta.
- Valor: 21474836480 bytes (20 * 1024 * 1024 * 1024).
- Expira en 720 horas (30 días).
- Peso 10 (peso más alto consumido primero).

Tarea 8: Agregar Voz y SMS Ilimitados

```
- name: Add Unlimited Voice
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
          "Account": "{{ service_uuid }}",
          "BalanceType": "*voice",
          "Balance": {
            "ID": "VOICE_Unlimited",
            "Value": 999999999,
            "ExpiryTime": "+720h"
          }
        }]
      }
```

- Agrega 999,999,999 segundos de voz (esencialmente ilimitados).
- Expira en 30 días.

Tarea 9: Crear Registro de Servicio en CRM

```
- name: Add Service via API
  uri:
    url: "{{ crm_config.crm.base_url }}/crm/service/"
    method: PUT
    body_format: json
    headers:
      Authorization: "Bearer {{ access_token }}"
    body:
      {
        "customer_id": "{{ customer_id }}",
        "product_id": "{{ product_id }}",
        "service_name": "Mobile - {{ phone_number }}",
        "service_type": "mobile",
        "service_uuid": "{{ service_uuid }}",
        "service_status": "Active",
        "service_provisioned_date": "{{ provision_datetime }}",
        "retail_cost": "{{ monthly_cost }}",
        "wholesale_cost": "{{
api_response_product.json.wholesale_cost }}",
        "icon": "fa-solid fa-sim-card"
      }
  register: service_creation_response
```

Qué Crea Esto:

- Registro de servicio vinculado al cliente.
- Vincula a OCS a través de `service_uuid`.
- Almacena costos minoristas y mayoristas.
- Establece el estado como "Activo".
- Devuelve `service_id` para operaciones posteriores.

Tarea 10: Asignar Inventario al Servicio

```
- name: Assign SIM Card to Service
  uri:
    url: "{{ crm_config.crm.base_url
}}/crm/inventory/inventory_id/{{ hostvars[inventory_hostname]['SIM
Card'] }}"
    method: PATCH
    body_format: json
    headers:
      Authorization: "Bearer {{ access_token }}"
    body:
      {
        "service_id": "{{
service_creation_response.json.service_id }}",
        "customer_id": "{{ customer_id }}",
        "item_state": "Assigned"
      }
```

Qué Hace Esto:

- Actualiza el registro de inventario de la tarjeta SIM.
- Establece `service_id` para vincular la SIM al servicio.
- Cambia el estado de "En Stock" a "Asignado".
- Se repite para el inventario del número de teléfono móvil.

Por Qué:

- Rastrear qué SIM está asignada a qué cliente.
- Prevenir la doble asignación de inventario.
- Habilitar informes y auditorías de inventario.

Tarea 11: Registrar Transacción de Costo de Configuración

```
- name: Add Setup Cost Transaction
  uri:
    url: "{{ crm_config.crm.base_url }}/crm/transaction/"
    method: PUT
    body_format: json
    headers:
      Authorization: "Bearer {{ access_token }}"
    body:
      {
        "customer_id": "{{ customer_id }}",
        "service_id": "{{
service_creation_response.json.service_id }}",
        "title": "{{ package_name }} - Setup",
        "description": "Activation fee",
        "retail_cost": "{{ setup_cost }}",
        "wholesale_cost": "{{
api_response_product.json.wholesale_setup_cost }}"
      }
```

Qué Hace Esto:

- Registra un cargo de £0.00 de configuración al cliente (minorista).
- Registra un costo mayorista de £1.00.
- Crea un registro de transacción para la facturación.

Tarea 12: Bloque de Rescate (Manejo de Errores)

```
rescue:
  - name: Remove account in OCS on failure
    uri:
      url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
      method: POST
      body:
        {
          "method": "ApierV2.RemoveAccount",
          "params": [{
            "Account": "{{ service_uuid }}"
          }]
        }

  - name: Fail the provisioning
    fail:
      msg: "Provisioning failed, rolled back OCS account"
```

Qué Hace Esto:

- Si alguna tarea falla, se ejecuta el bloque de rescate.
- Elimina la cuenta OCS que fue parcialmente creada.
- Libera el inventario de nuevo a "En Stock".
- Falla el trabajo de provisión con un mensaje de error.

Por Qué:

- Previene cuentas huérfanas en OCS.
- Asegura un rollback limpio en errores.
- Mantiene la consistencia de datos.

Provisión Completa: Qué Se Creó

Después de una provisión exitosa, el sistema tiene:

1. Cuenta OCS (CGRateS):

- ID de cuenta: `Local_Mobile_SIM_a3f2c1d8`.
- Saldos:
  - 20GB de datos (expira en 30 días).

    - Voz ilimitada (999M segundos, expira en 30 días).
    - SMS ilimitados (999M mensajes, expira en 30 días).

2. Registro de Servicio en CRM:

- ID de servicio: 1234.
- Cliente: John Doe (customer_id: 123).
- Producto: Prepaid Mobile 20GB (product_id: 42).
- Nombre del servicio: "Mobile - +44 7700 900123".
- UUID del servicio: `Local_Mobile_SIM_a3f2c1d8`.
- Estado: Activo.
- Costo Mensual: £15.00 (minorista), £5.00 (mayorista).
- Beneficio: £10.00/mes.

3. Asignaciones de Inventario:

- SIM Card 5001: Asignada al servicio 1234, cliente 123.
- Mobile Number 5002: Asignado al servicio 1234, cliente 123.

4. Registros de Transacción:

- Se creó un registro de transacción de costo de configuración.
- Se registró el cargo del primer mes.

5. El Cliente Ahora Puede:

- Ver el servicio en el portal de autoservicio.
- Ver el saldo de 20GB de datos.
- Realizar llamadas y enviar SMS.
- Recargar o agregar complementos.
- Ver el uso en tiempo real.

Etapa 3: Agregar Complementos y Recargas

Después de que un servicio esté activo, los clientes pueden comprar complementos para mejorar su servicio.

Flujo de Provisión de Complementos

Suponga que el cliente ha utilizado 18GB de su asignación de 20GB y desea comprar el complemento "5GB Data Boost".

1. El Cliente Navega al Servicio

- Abre la página del servicio "Mobile - +44 7700 900123".
- Ve el uso actual: 18GB de 20GB utilizados (90%).
- Hace clic en "Agregar Complemento" o "Recargar".

2. El Sistema Filtra Complementos Disponibles

Solo muestra complementos donde:

- `category = "addon"`.
- `service_type = "mobile"` (coincide con el tipo de servicio).
- `residential = true` (si el cliente es residencial).
- `enabled = true`.

El cliente ve: "5GB Data Boost - £5.00".

3. El Cliente Selecciona el Complemento

- Hace clic en "5GB Data Boost".
- Confirma la compra por £5.00.
- El sistema captura la autorización de pago.

4. Se Inicia la Provisión del Complemento

El sistema llama a `play_topup_charge_then_action.yaml` con variables:

```
{
    "product_id": 43,  # Producto 5GB Data Boost
    "customer_id": 123,
    "service_id": 1234,  # Servicio existente
    "access_token": "eyJ...",
    "data_gb": 5,  # De provisioning_json_vars
    "days": 7      # De provisioning_json_vars
}
```

Diferencia Clave con Servicios Independientes:

- `service_id` está incluido (servicio existente a modificar).
- No se requiere inventario.
- No se crea un servicio (modifica el existente).

Recorrido del Playbook de Provisión de Complementos

Tarea 1: Obtener Detalles del Servicio

```
- name: Get Service information from CRM API
  uri:
    url: "http://localhost:5000/crm/service/service_id/{{
service_id }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
  register: api_response_service

- name: Set service facts
  set_fact:
    service_uuid: "{{ api_response_service.json.service_uuid }}"
    customer_id: "{{ api_response_service.json.customer_id }}"
```

Por Qué:

- Necesita `service_uuid` para agregar saldo a la cuenta OCS correcta.
- Verifica que el servicio exista y esté activo.

- Asegura que el servicio pertenezca al cliente.

Tarea 2: Cobrar al Cliente

```yaml
- name: Get Customer's Default Payment Method
  uri:
    url: "http://localhost:5000/api/payments/methods/default?
customer_id={{ customer_id }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
  register: api_response_payment_method

- name: Get default payment method ID
  set_fact:
    payment_method_id: "{{
api_response_payment_method.json.data.payment_method_id }}"

- name: Charge customer
  uri:
    url: "http://localhost:5000/api/payments/charge"
    method: POST
    body_format: json
    headers:
      Authorization: "Bearer {{ access_token }}"
    body:
      {
        "customer_id": "{{ customer_id | int }}",
        "amount": 5.00,
        "currency": "USD",
        "payment_method_id": "{{ payment_method_id }}",
        "metadata": {
          "description": "5GB Data Boost",
          "service_id": "{{ service_id | int }}",
          "product_id": "{{ product_id | int }}",
          "invoice": true
        }
      }
  register: charge_response

- name: Assert payment successful
  assert:
    that:
      - charge_response.json.success == true
```

Qué Hace Esto:

- Encuentra el método de pago predeterminado del cliente.
- Cobra £5.00 a la tarjeta.
- Registra el costo mayorista de £1.50 para seguimiento de márgenes.
- Crea una transacción vinculada al servicio.
- Agrega a la próxima factura.
- Falla la provisión si el pago falla.

Por Qué Cobrar Primero:

- No se entrega crédito hasta que se confirme el pago.
- Previene fraudes.
- Coincide el pago con la provisión del complemento.

Tarea 3: Agregar Saldo de Datos a OCS

```
- name: Add 5GB Data Balance
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body_format: json
    body:
      {
        "method": "ApierV1.AddBalance",
        "params": [{
          "Account": "{{ service_uuid }}",
          "BalanceType": "*data",
          "Balance": {
            "ID": "DATA_5GB_Boost_{{ uuid }}",
            "Value": 5368709120,
            "ExpiryTime": "+168h",
            "Weight": 20
          }
        }]
      }
```

Qué Hace Esto:

- Agrega 5GB (5368709120 bytes) a la cuenta.
- Expira en 168 horas (7 días).

- Peso 20 (peso más alto consumido primero - boost antes de la asignación mensual).

Saldo del Cliente Después del Complemento:

- Mensual original: 2GB restantes (expira en 25 días).
- Nuevo boost: 5GB (expira en 7 días).
- Total disponible: 7GB.
- Orden de uso: Boost consumido primero, luego mensual.

Tarea 4: Registrar Transacción

```
- name: Add Addon Transaction
  uri:
    url: "http://localhost:5000/crm/transaction/"
    method: PUT
    body_format: json
    headers:
      Authorization: "Bearer {{ access_token }}"
    body:
      {
        "customer_id": "{{ customer_id }}",
        "service_id": "{{ service_id }}",
        "title": "5GB Data Boost",
        "description": "Additional 5GB data valid for 7 days",
        "retail_cost": 5.00,
        "wholesale_cost": 1.50
      }
```

Qué Hace Esto:

- Registra un cargo de £5.00 al cliente.
- Registra un costo mayorista de £1.50.
- Vincula la transacción al servicio para informes.

Resumen Completo del Flujo de Complementos

1. El cliente selecciona el complemento de la lista filtrada.
2. Pago autorizado y cobrado.

3. Saldo de datos agregado a la cuenta OCS.
4. Transacción registrada en CRM.
5. El cliente ve inmediatamente el saldo actualizado: 7GB disponibles.

Seguimiento Financiero:

- Cargo mensual del servicio: £15 minorista, £5 mayorista.
- Compra del complemento: £5 minorista, £1.50 mayorista.

Renovación Automática: Complementos Recurrentes

Algunos complementos pueden configurarse para renovarse automáticamente (planes de datos mensuales, suscripciones, etc.).

Configuración del Producto:

```
{
  "product_name": "Monthly 10GB Data Plan",
  "category": "addon",
  "retail_cost": 10.00,
  "contract_days": 30,
  "auto_renew": "true",
  "provisioning_play": "play_topup_charge_then_action"
}
```

La Provisión Crea un Plan de Acción:

```
- name: Create ActionPlan for Auto-Renewal
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body:
      {
        "method": "ApierV1.SetActionPlan",
        "params": [{
          "Id": "ServiceID_{{ service_uuid }}__ProductID_{{
product_id }}__MonthlyRenewal",
          "ActionPlan": [{
            "ActionsId": "Action_{{ product_slug }}",
            "Years": "*any",
            "Months": "*any",
            "MonthDays": "*any",
            "WeekDays": "*any",
            "Time": "00:00:00",
            "Weight": 10
          }],
          "Overwrite": false,
          "ReloadScheduler": true
        }]
      }
```

Qué Hace Esto:

- Crea una tarea programada en OCS.
- Ejecuta `Action_{{ product_slug }}` cada 30 días.
- La acción cobra al cliente y vuelve a aplicar el saldo de datos.
- Continúa hasta que el cliente cancele.

Gestión del Cliente:

- El cliente ve "Próxima Renovación: 1 de febrero de 2025 - £10.00" en la vista del servicio.
- Puede hacer clic en "Cancelar Renovación Automática" para detener futuros cargos.
- Puede hacer clic en "Renovar Ahora" para aplicar inmediatamente la asignación del próximo mes.

Etapa 4: Desprovisión de Servicios

Cuando un cliente cancela el servicio, el sistema debe eliminar todos los recursos de manera limpia.

Disparadores de Desprovisión

La desprovisión puede ser desencadenada por:

1. **Cancelación del cliente** - El cliente hace clic en "Cancelar Servicio".
2. **Acción administrativa** - El personal marca el servicio para desactivación.
3. **Falta de pago** - El servicio expira debido a la falta de renovación.
4. **Fin de contrato** - El contrato a plazo fijo llega a su fecha de finalización.

Flujo de Desprovisión

1. El Cliente Inicia la Cancelación

- Navega al servicio.
- Hace clic en "Cancelar Servicio".
- El sistema pregunta: "¿Está seguro? Cualquier saldo restante será perdido."
- El cliente confirma.

2. Período de Gracia (Opcional)

Algunos operadores implementan un período de gracia:

- El servicio se marca como "Cancelación Pendiente".
- Permanece activo durante 7-30 días.
- El cliente puede revertir la cancelación durante el período de gracia.
- Desprovisión automática después del período de gracia.

3. Trabajo de Desprovisión Creado

El sistema crea un trabajo de provisión con:

```
{
    "action": "deprovision",
    "service_id": 1234,
    "customer_id": 123,
    "service_uuid": "Local_Mobile_SIM_a3f2c1d8"
}
```

Llama al playbook especificado en `service.deprovisioning_play` o al bloque de rescate del playbook original.

4. Playbook de Desprovisión de Ansible

```
- name: Deprovision Mobile Service
  hosts: localhost
  tasks:
    - name: Disable OCS Account
      uri:
        url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
        method: POST
        body:
          {
            "method": "ApierV2.SetAccount",
            "params": [{
              "Account": "{{ service_uuid }}",
              "ExtraOptions": { "Disabled": true }
            }]
          }

    - name: Remove ActionPlans (stop auto-renewals)
      uri:
        url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
        method: POST
        body:
          {
            "method": "ApierV1.RemoveActionPlan",
            "params": [{
              "Id": "ServiceID_{{ service_uuid }}__*"
            }]
          }

    - name: Update Service Status in CRM
      uri:
        url: "http://localhost:5000/crm/service/{{ service_id }}"
        method: PATCH
        body:
          {
            "service_status": "Deactivated",
            "service_deactivate_date": "{{ current_datetime }}"
          }

    - name: Release Inventory to Stock
      uri:
        url: "http://localhost:5000/crm/inventory/inventory_id/{{
sim_card_id }}"
        method: PATCH
```

```
        body:
          {
            "service_id": null,
            "customer_id": null,
            "item_state": "Decommissioned"
          }
```

Qué Hace Esto:

1. **Desactiva la cuenta OCS** - Detiene toda la carga, el uso se bloquea.
2. **Elimina los Planes de Acción** - Cancela las renovaciones automáticas.
3. **Actualiza el servicio en CRM** - Estado "Desactivado", fecha registrada.
4. **Libera el inventario** - SIM marcada como "Desactivada", disponible para reutilización (después de renovación).

5. Post-Desprovisión

El sistema realiza la limpieza:

- El cliente ya no ve el servicio en el portal de autoservicio.
- El servicio permanece en CRM para informes históricos.
- Las transacciones y facturas se preservan para contabilidad.
- El inventario puede ser renovado y reutilizado.
- La cuenta OCS puede ser archivada después del período de retención.

Desprovisión Parcial vs Completa

Desprovisión Parcial (Suspensión):

- Utilizada para falta de pago o suspensión temporal.
- Cuenta OCS desactivada pero no eliminada.
- Saldos preservados.
- Puede reactivarse cuando se reciba el pago.

```
- name: Suspend Service
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body:
      {
        "method": "ApierV2.SetAccount",
        "params": [{
          "Account": "{{ service_uuid }}",
          "ExtraOptions": { "Disabled": true }
        }]
      }
```

Desprovisión Completa (Cancelación Permanente):

- Utilizada para cancelación permanente.
- Cuenta OCS eliminada por completo.
- Saldos perdidos.
- No puede reactivarse.

```
- name: Remove OCS Account
  uri:
    url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
    method: POST
    body:
      {
        "method": "ApierV2.RemoveAccount",
        "params": [{
          "Account": "{{ service_uuid }}"
        }]
      }
```

Mejores Prácticas para la Gestión de Productos

Gestión del Ciclo de Vida del Producto

Estados del Producto:

- `enabled: true` - Producto disponible para nuevos pedidos.
- `enabled: false` - Producto deshabilitado, los servicios existentes continúan.

Deshabilitar Productos:

- Marque el producto como `enabled: false` para prevenir nuevos pedidos.
- Los servicios existentes permanecen activos.
- Los clientes aún pueden renovar/modificar servicios existentes.
- Útil para descontinuar productos antiguos.

Gestión de Inventario

Estados del Inventario:

- `New` - Stock nuevo, listo para asignar.
- `In Stock` - Disponible para provisión.
- `Assigned` - Vinculado al servicio del cliente.
- `Decommissioned` - Puede ser renovado y reutilizado.
- `Damaged` - Necesita reparación o eliminación.

Reutilización de Inventario:

Después de la desprovisión:

- Tarjetas SIM: Renovar y marcar como "En Stock".
- Números de teléfono: Liberar después del período de portabilidad (30 días).
- Equipos: Probar, renovar, marcar como "Usado".

Métricas de Provisión

Monitorear:

- Tasa de éxito de provisión.
- Tiempo promedio de provisión.
- Puntos de fallo comunes.
- Rotación de inventario.

Integración de Mailjet con OmniCRM

OmniCRM se integra con **Mailjet** para gestionar toda la comunicación por correo electrónico con clientes y personal, asegurando una entrega de correos electrónicos profesional, con marca y confiable tanto para correos electrónicos transaccionales como para campañas de marketing.

Descripción general

La integración de Mailjet proporciona:

- **Correos Electrónicos Transaccionales Automatizados** - Restablecimientos de contraseña, facturas, correos de bienvenida, notificaciones
- **Consentimiento de Marketing (doble opt-in)** - Opt-in/opt-out auditable para correos electrónicos de marketing, incluyendo flujos de confirmación y re-permiso
- **Plantillas de Correo Electrónico** - Más de 10 tipos de correos electrónicos preconfigurados con plantillas personalizables de Mailjet
- **Campañas de Marketing** - Campañas de correo electrónico segmentadas basadas en datos de clientes (los segmentos pueden construirse a partir de datos de CRM o carga de CSV)
- **Entrega Confiable** - Infraestructura de correo electrónico profesional con seguimiento y análisis

OmniCRM **no** mantiene una sincronización de lista de contactos de Mailjet en vivo. Utilice **Clientes → Todos los Contactos** en la interfaz de administración (o `GET /crm/contact/export`) para exportar contactos para la gestión de listas externas cuando sea necesario.

Configuración

Mailjet se configura en `OmniCRM-API/crm_config.yaml` bajo la sección `mailjet`.

Configuración Básica

```yaml
mailjet:
  api_key: your_mailjet_api_key
  api_secret: your_mailjet_api_secret
```

Obtención de Credenciales de API:

1. Cree una cuenta en <https://www.mailjet.com>
2. Navegue a **Configuración de Cuenta → Claves de API**
3. Copie **Clave de API** y **Clave Secreta**
4. Pegue en `crm_config.yaml`

Configuración de Plantillas de Correo Electrónico

OmniCRM utiliza **10 tipos distintos de plantillas de correo electrónico** para comunicaciones automatizadas. Cada plantilla se configura con:

- **from_email** - Dirección de correo electrónico del remitente
- **from_name** - Nombre de visualización del remitente
- **template_id** - ID de plantilla de Mailjet (numérico)
- **subject** - Línea de asunto del correo electrónico

Tipos de Plantillas y Configuración

Correo Electrónico de Bienvenida al Cliente

Enviado cuando se crea una nueva cuenta de cliente.

```
api_crmCommunicationCustomerWelcome:
  from_email: "support@yourcompany.com"
  from_name: "Soporte de Su Empresa"
  template_id: 5977509
  subject: "Bienvenido a Su Empresa"
```

Cuándo se Envía:

- Nuevo cliente se registra a través del portal de Autocuidado
- El personal crea una nueva cuenta de cliente
- El cliente activa el servicio por primera vez

Variables de Plantilla Disponibles:

- `{{ var:customer_name }}` - Nombre completo del cliente
- `{{ var:email }}` - Dirección de correo electrónico del cliente
- `{{ var:company_name }}` - Nombre de su empresa
- `{{ var:login_url }}` - Enlace al portal de Autocuidado
- `{{ var:support_url }}` - Enlace a la página de soporte

Correo Electrónico de Factura del Cliente

Enviado cuando se genera una factura y está lista para el pago.

```
api_crmCommunicationCustomerInvoice:
  from_email: "billing@yourcompany.com"
  from_name: "Facturación de Su Empresa"
  template_id: 6759851
  subject: "Su Factura - "
```

Cuándo se Envía:

- Factura generada automáticamente para el período de facturación
- Factura manual creada por el personal
- El cliente solicita una copia de la factura

Variables de Plantilla Disponibles:

- `{{ var:customer_name }}` - Nombre completo del cliente
- `{{ var:invoice_number }}` - ID/número de la factura
- `{{ var:invoice_date }}` - Fecha de emisión de la factura
- `{{ var:due_date }}` - Fecha de vencimiento del pago
- `{{ var:total_amount }}` - Monto total a pagar
- `{{ var:invoice_url }}` - Enlace para ver/descargar el PDF de la factura
- `{{ var:pay_url }}` - Enlace para pagar la factura en línea

Adjunto de Factura:

El PDF de la factura se adjunta automáticamente al correo electrónico.

Recordatorio de Factura del Cliente

Enviado para recordar a los clientes sobre facturas vencidas.

```
api_crmCommunicationCustomerInvoiceReminder:
  from_email: "billing@yourcompany.com"
  from_name: "Facturación de Su Empresa"
  template_id: 6759852
  subject: "Recordatorio de Pago - Factura Vencida"
```

Cuándo se Envía:

- La factura está X días vencida (configurable)
- Recordatorio manual activado por el personal
- Flujo de recordatorio automatizado (si está configurado)

Variables de Plantilla Disponibles:

- `{{ var:customer_name }}`
- `{{ var:invoice_number }}`
- `{{ var:due_date }}`
- `{{ var:days_overdue }}`
- `{{ var:total_amount }}`
- `{{ var:pay_url }}`

Correo Electrónico de Bienvenida al Usuario del Personal

Enviado cuando se crea una nueva cuenta de usuario del personal.

```
api_crmCommunicationUserWelcome:
  from_email: "admin@yourcompany.com"
  from_name: "Administrador de Su Empresa"
  template_id: 5977510
  subject: "Bienvenido al Equipo"
```

Cuándo se Envía:

- El administrador crea un nuevo usuario del personal
- Se hace clic en el botón "Enviar Correo de Bienvenida" en la gestión de usuarios

Variables de Plantilla Disponibles:

- `{{ var:user_name }}` - Nombre completo del usuario del personal
- `{{ var:email }}` - Correo electrónico del usuario del personal
- `{{ var:role }}` - Rol(es) asignado(s)
- `{{ var:login_url }}` - Enlace para iniciar sesión en el portal de administración
- `{{ var:temp_password }}` - Contraseña temporal (si aplica)
- `{{ var:support_email }}` - Contacto de soporte técnico

Restablecimiento de Contraseña del Usuario

Enviado cuando un usuario solicita restablecer su contraseña.

```
api_crmCommunicationUserPasswordReset:
  from_email: "noreply@yourcompany.com"
  from_name: "Seguridad de Su Empresa"
  template_id: 5977511
  subject: "Solicitud de Restablecimiento de Contraseña"
```

Cuándo se Envía:

- El usuario hace clic en "Olvidé mi Contraseña" en la página de inicio de sesión
- El usuario envía una solicitud de restablecimiento de contraseña

Variables de Plantilla Disponibles:

- `{{ var:user_name }}`
- `{{ var:reset_url }}` - Enlace de restablecimiento de contraseña con límite de tiempo (típicamente 1 hora)
- `{{ var:expiry_time }}` - Cuándo expira el enlace de restablecimiento

Nota de Seguridad:

Los enlaces de restablecimiento expiran después del período de tiempo configurado (predeterminado 1 hora).

Éxito en el Restablecimiento de Contraseña del Usuario

Enviado para confirmar que la contraseña se cambió con éxito.

```
api_crmCommunicationUserPasswordResetSuccess:
  from_email: "noreply@yourcompany.com"
  from_name: "Seguridad de Su Empresa"
  template_id: 5977512
  subject: "Contraseña Cambiada con Éxito"
```

Cuándo se Envía:

- El usuario completa con éxito el restablecimiento de la contraseña
- Inmediatamente después de establecer la nueva contraseña

Variables de Plantilla Disponibles:

- `{{ var:user_name }}`
- `{{ var:change_date }}` - Fecha/hora en que se cambió la contraseña
- `{{ var:ip_address }}` - Dirección IP del cambio (opcional)
- `{{ var:support_email }}` - Contacto si el cambio fue no autorizado

Cambio de Contraseña del Usuario

Enviado cuando un usuario cambia su contraseña desde la configuración.

```
api_crmCommunicationUserPasswordChange:
  from_email: "noreply@yourcompany.com"
  from_name: "Seguridad de Su Empresa"
  template_id: 5977513
  subject: "Notificación de Cambio de Contraseña"
```

Cuándo se Envía:

- El usuario cambia la contraseña desde su perfil/configuración
- El administrador restablece la contraseña del usuario

Variables de Plantilla Disponibles:

- `{{ var:user_name }}`
- `{{ var:change_date }}`
- `{{ var:changed_by }}` - "Auto" o nombre del administrador
- `{{ var:support_email }}`

Verificación de Correo Electrónico

Enviado para verificar la dirección de correo electrónico de un usuario.

```
api_crmCommunicationEmailVerification:
  from_email: "noreply@yourcompany.com"
  from_name: "Su Empresa"
  template_id: 5977514
  subject: "Verifique Su Dirección de Correo Electrónico"
```

Cuándo se Envía:

- Se crea una nueva cuenta (cliente o personal)
- El usuario cambia la dirección de correo electrónico
- Se requiere verificación de correo electrónico por motivos de seguridad

Variables de Plantilla Disponibles:

- `{{ var:user_name }}`
- `{{ var:verification_url }}` - Enlace para verificar el correo electrónico
- `{{ var:verification_code }}` - Código para ingresar manualmente (alternativa al enlace)

Notificación de Balance Expirado

Enviado cuando el saldo/permiso de servicio de un cliente ha expirado.

```
api_crmCommunicationsBalanceExpired:
  from_email: "support@yourcompany.com"
  from_name: "Soporte de Su Empresa"
  template_id: 5977515
  subject: "Su Saldo de Servicio Ha Expirado"
```

Cuándo se Envía:

- El saldo prepago expira
- La fecha de renovación de la asignación mensual ha pasado
- Se alcanza la fecha de expiración del servicio

Variables de Plantilla Disponibles:

- `{{ var:customer_name }}`
- `{{ var:service_name }}` - Nombre del servicio expirado
- `{{ var:expiry_date }}`
- `{{ var:balance_type }}` - "Datos", "Voz", "Monetario", etc.
- `{{ var:renewal_url }}` - Enlace para renovar/cargar saldo

Alerta de Bajo Balance

Enviado cuando el saldo de un cliente cae por debajo del umbral configurado.

```
api_crmCommunicationsBalanceLow:
  from_email: "support@yourcompany.com"
  from_name: "Soporte de Su Empresa"
  template_id: 5977516
  subject: "Alerta de Bajo Balance"
```

Cuándo se Envía:

- El saldo cae por debajo del umbral (por ejemplo, 20% restante)
- Configurado en el plan de servicio o OCS
- Monitoreo en tiempo real activa la alerta

Variables de Plantilla Disponibles:

- `{{ var:customer_name }}`
- `{{ var:service_name }}`
- `{{ var:current_balance }}`
- `{{ var:threshold }}`
- `{{ var:balance_type }}`
- `{{ var:topup_url }}` - Enlace para agregar saldo

Correo Electrónico de Entrega de eSIM

Enviado cuando se provisiona una eSIM y está lista para activación.

```
api_crmCommunicationEsimDelivery:
  from_email: "support@yourcompany.com"
  from_name: "Soporte de Su Empresa"
  template_id: 0
  subject: "Su eSIM Está Lista"
```

Cuándo se Envía:

- eSIM provisionada durante el auto-registro
- eSIM asignada a un servicio al cliente

Variables de Plantilla Disponibles:

- `{{ var:customer_name }}` - Nombre completo del cliente
- `{{ var:iccid }}` - ICCID de la eSIM
- `{{ var:activation_code }}` - Cadena de código de activación LPA
- `{{ var:has_qr_code }}` - Booleano, verdadero si la imagen del código QR está adjunta
- `{{ var:has_activation_code }}` - Booleano, verdadero si el código de activación está disponible

Fallback Incorporado:

Cuando `template_id` es `0` (o la clave de plantilla falta en la configuración), OmniCRM utiliza un correo electrónico HTML con marca incorporado con el código QR de la eSIM renderizado en línea. Los colores y el nombre de la marca son controlados por la sección `branding` en `crm_config.yaml`. Consulte la Guía de Configuración para más detalles.

Creando Plantillas de Correo Electrónico en Mailjet

Para cada tipo de correo electrónico, necesita crear una plantilla correspondiente en Mailjet.

Paso 1: Crear Plantilla en Mailjet

1. Inicie sesión en el panel de Mailjet
2. Navegue a **Plantillas de Correo Electrónico → Plantillas Transaccionales**
3. Haga clic en **Crear una Nueva Plantilla**
4. Elija **Codifique su propia plantilla** (para usuarios avanzados) o **Utilice el generador de plantillas**

Paso 2: Diseñar Plantilla

Utilice el generador de arrastrar y soltar de Mailjet o el editor HTML para diseñar su correo electrónico.

Elementos Esenciales:

- **Encabezado** - Logo de la empresa y marca
- **Saludo** - Personalizado con `{{ var:customer_name }}` o `{{ var:user_name }}`
- **Contenido** - Cuerpo del mensaje principal
- **Variables** - Inserte variables de plantilla de la lista anterior
- **Llamada a la Acción** - Botones/enlaces para acciones del usuario
- **Pie de Página** - Enlace para darse de baja, dirección de la empresa, contacto de soporte

Ejemplo de Plantilla (Restablecimiento de Contraseña):

```
<!DOCTYPE html>
<html>
<head>
    <style>
        body { font-family: Arial, sans-serif; }
        .button { background-color: #4CAF50; color: white;
padding: 14px 28px; }
    </style>
</head>
<body>
    ![Logo](https://yourcompany.com/logo.png)

    <h2>Solicitud de Restablecimiento de Contraseña</h2>

    <p>Hola {{ var:user_name }},</p>

    <p>Recibimos una solicitud para restablecer su contraseña.
Haga clic en el botón a continuación para crear una nueva
contraseña:</p>

    <a href="{{ var:reset_url }}" class="button">Restablecer
Contraseña</a>

    <p>Este enlace expira en {{ var:expiry_time }}.</p>

    <p>Si no solicitó esto, por favor ignore este correo
electrónico.</p>

    <hr>
    <p style="font-size: 12px; color: #666;">
        Su Empresa | support@yourcompany.com<br/>
        123 Business St, Ciudad, País
    </p>
</body>
</html>
```

Paso 3: Obtener ID de Plantilla

1. Guarde la plantilla en Mailjet
2. Anote el **ID de Plantilla** (numérico, por ejemplo, `5977509`)
3. Copie este ID a `crm_config.yaml`

Paso 4: Probar Plantilla

1. En Mailjet, utilice la función **Correo de Prueba**
2. Proporcione valores de muestra para todas las variables
3. Envíe un correo de prueba a usted mismo
4. Verifique el formato, los enlaces y la marca

Paso 5: Configurar en OmniCRM

Agregue la configuración de la plantilla a `crm_config.yaml`:

```
mailjet:
  api_key: your_api_key
  api_secret: your_secret

  api_crmCommunicationUserPasswordReset:
    from_email: "noreply@yourcompany.com"
    from_name: "Seguridad de Su Empresa"
    template_id: 5977511
    subject: "Solicitud de Restablecimiento de Contraseña"
```

Reinicie la API de OmniCRM para que los cambios surtan efecto:

```
cd OmniCRM-API
sudo systemctl restart omnicrm-api
```

Exportación de Contactos (CRM)

Para informes, cumplimiento o importación manual en Mailjet (u otro ESP), los administradores pueden navegar y exportar contactos desde el CRM sin una sincronización de lista de Mailjet en vivo.

Interfaz de Administración

Clientes → Todos los Contactos (`/contacts`) proporciona una tabla paginada con búsqueda, filtros (`marketing_consent_status`, `customer_status`), y

Exportar CSV. La exportación utiliza los mismos filtros y búsqueda que la tabla.

API

Ambos puntos finales requieren el permiso `VIEW_CUSTOMER_CONTACT`.

Método	Ruta	Propósito
`GET`	`/crm/contact/search`	Lista de contactos paginada con `search`, `page`, `per_page`, `sort`, `order`, `filters` (JSON)
`GET`	`/crm/contact/export`	Descarga CSV con los mismos parámetros de consulta (máx. 50,000 filas)

Columnas CSV: `contact_id`, `customer_id`, `customer_name`, `customer_status`, nombre, correo electrónico, teléfono, título del trabajo, `contact_types`, `marketing_consent_status`, `marketing_consent_granted_at`, `created`, `last_modified`.

Filtre por `marketing_consent_status=granted` al preparar una lista para uso de marketing. Importe el CSV en Mailjet (o su ESP) utilizando su flujo de importación de contactos.

Implementación: `OmniCRM-API/services/contact_service.py` (`search_contacts`, `export_contacts_csv`).

Consentimiento de Marketing (doble opt-in)

Mailjet se niega a aprobar una importación de lista de contactos a menos que el remitente pueda demostrar que cada destinatario consentió explícitamente recibir correos electrónicos de marketing. Para cumplir con ese requisito, OmniCRM almacena el consentimiento de marketing por contacto y expone un

flujo de doble opt-in ("correo electrónico de confirmación") que produce un rastro de consentimiento auditable.

La implementación se encuentra en `OmniCRM-API/services/marketing_consent_service.py`. Solo los contactos con `marketing_consent_status == 'granted'` deben ser utilizados para envíos de marketing o importaciones de listas externas.

Modelo de datos

Se involucran dos tablas:

- `contact.marketing_consent_status` (`Enum: none | pending | granted | revoked`) y `contact.marketing_consent_granted_at` - el estado actual desnormalizado utilizado para un filtrado rápido en la tabla de contactos y exportación.
- `marketing_consent` - un **registro de auditoría solo de anexado**. Cada transición de estado (marcador de registro, clic de confirmación, alternancia de autocuidado, revocación impulsada por webhook, re-permiso de administrador) escribe una nueva fila capturando el ID de contacto, correo electrónico en el momento del evento, fuente, versión del texto de consentimiento, dirección IP, agente de usuario, token firmado y marcas de tiempo de confirmación/revocación. El registro es la fuente de verdad si Mailjet alguna vez solicita prueba de consentimiento para un destinatario dado.

La "versión del texto de consentimiento" se configura bajo `marketing_consent.consent_text_version` en `crm_config.yaml` y se estampa en cada fila de auditoría, por lo que cambiar la redacción del texto de consentimiento en el futuro no es destructivo: las filas existentes continúan apuntando a la versión que el usuario realmente vio.

Configuración

Agregue el bloque `marketing_consent` a `crm_config.yaml`:

```yaml
marketing_consent:
  consent_text_version: "v1-2026-05"
  confirmation_url_path: "/marketing/confirm"
  token_expiry_days: 30
  re_permission_batch_size: 200
  re_permission_rate_per_second: 5
  re_permission_skip_pending_within_hours: 72
```

Opcionalmente, también configure una plantilla de Mailjet para el correo electrónico de confirmación (de lo contrario, se utiliza un cuerpo HTML/texto codificado):

```yaml
mailjet:
  api_crmCommunicationMarketingConfirm:
    from_email: "support@example.com"
    from_name: "Soporte de Ejemplo Móvil"
    template_id: 0       # 0 = usar cuerpo HTML incorporado
    subject: "Por favor confirme su suscripción"
```

Variables de plantilla de Mailjet expuestas a una plantilla personalizada: `firstname`, `confirmation_link`, `consent_text_version`.

Flujo A: Nuevas inscripciones

El formulario de auto-inscripción muestra una casilla de verificación "Inscríbase a las noticias de Ejemplo Móvil para las últimas noticias, eventos y promociones". Cuando se marca:

1. `routes/auth.py /self-signup-partial` extrae `customer.subscribe_to_newsletter` del cuerpo de la solicitud.
2. Después de que se crea el contacto, se llama a `marketing_consent_service.request_consent` con `source='signup'`, la IP de la solicitud y el User-Agent.
3. Se escribe una fila de auditoría `marketing_consent` con `granted=True`, `confirmed_at=NULL`, y un token seguro para URL. El `marketing_consent_status` del contacto se establece en `pending`.

4. Se envía un correo electrónico transaccional de Mailjet que contiene `${public_url}${confirmation_url_path}?token=...`.
5. Cuando el usuario hace clic en el enlace, `GET /crm/marketing-consent/confirm` canjea el token: `confirmed_at` se estampa en la fila de auditoría y `marketing_consent_status` cambia a `granted`.

Flujo B: Autocuidado (usuarios registrados)

El autocuidado presenta una página de "Preferencias" (`/self-care/preferences`) que llama:

Método	Ruta	Propósito
`GET`	`/crm/marketing-consent/me`	Estado actual + historial completo de auditoría
`POST`	`/crm/marketing-consent/me/subscribe`	(Re)iniciar el flujo de doble opt-in
`POST`	`/crm/marketing-consent/me/unsubscribe`	Revocar el consentimiento de marketing de inmediato

La revocación escribe una fila de auditoría `granted=False` y establece el estado en `revoked`. El contacto debe ser eliminado de cualquier lista de marketing externa mantenida fuera de OmniCRM.

Flujo C: Re-permiso de base existente (administrador)

Los contactos heredados creados antes de que se implementara esta función no tienen un registro de consentimiento y, por lo tanto, están en `marketing_consent_status='none'`. Los administradores pueden pedirles que opten por participar a través de la campaña de re-permiso:

Método	Ruta	Propósito
`GET`	`/crm/oam/marketing-consent/stats`	Conteos por estado + lista de candidatos + muestras
`POST`	`/crm/oam/marketing-consent/re-permission?dry_run=true`	Plan (sin correo enviado)
`POST`	`/crm/oam/marketing-consent/re-permission?dry_run=false`	Ejecutar. Limitado y controlado por configuración.

El trabajo escanea cada contacto cuyo estado es `none`, `revoked`, o `pending` mayor que `re_permission_skip_pending_within_hours`, tiene un correo electrónico válido, no está dado de baja, y cuyo estado de cliente no es `Closed`. Para cada candidato, se llama al mismo `request_consent` utilizado por el flujo de inscripción, por lo que el formato de la fila de auditoría y la semántica del token son idénticos independientemente del origen.

El envío está limitado por tasa a través de un token-bucket del tamaño de `re_permission_rate_per_second`, manteniéndonos cómodamente por debajo del límite de rendimiento transaccional de Mailjet. Un parámetro de consulta opcional `limit` limita el conteo de envíos por ejecución para que la primera ejecución pueda ser una pequeña prueba de humo.

Implementación recomendada para el primer despliegue

1. Confirme que la migración de base de datos para `marketing_consent` se ha ejecutado.
2. Visite **Email Marketing → Consentimiento de Marketing** en la interfaz de administración (`/email-marketing/marketing-consent`). Revise el desglose por estado. La mayoría de los contactos heredados estarán en `none`.
3. Ejecute **Re-permiso de prueba** e inspeccione los conteos.

4. **Envíe correos electrónicos de confirmación** con el parámetro de consulta `limit` establecido en un pequeño lote (por ejemplo, 50) para validar la entregabilidad y el volumen de bandeja de entrada.
5. Ejecute nuevamente sin `limit` para cubrir el resto de la base. El trabajo es idempotente: los contactos con una solicitud pendiente reciente se omiten.
6. Después de que las inscripciones se estabilicen, utilice **Clientes → Todos los Contactos**, filtre `marketing_consent_status = granted`, y **Exportar CSV** para importar en Mailjet (o su ESP) si mantiene una lista de marketing allí.

Métricas de Prometheus

Métrica	Etiquetas	Nota
`omnicrm_marketing_consent_events_total`	`event`	`confirmation_se` `confirmation_fa` `revoked`, `token_r` `token_invalid`, `t`
`omnicrm_marketing_consent_status_total`	`status`	Medidor por `none|pending|gra` Refrescado por `get_global_stat`

Las métricas de la API transaccional de Mailjet (`omnicrm_mailjet_api_*`, `omnicrm_mailjet_emails_sent_*`) están documentadas en Monitoreo & Métricas.

Solución de Problemas

Correo electrónico no enviado

- **Causa:** Credenciales de API inválidas, cuenta de Mailjet suspendida o ID de plantilla incorrecto
- **Solución:**
  - Verifique `api_key` y `api_secret` en `crm_config.yaml`

- Verifique el estado de la cuenta de Mailjet y la facturación
- Verifique que el ID de plantilla exista en Mailjet
- Verifique los registros de API en busca de errores

Las variables de plantilla no se sustituyen

- **Causa:** Desajuste en el nombre de la variable o datos faltantes en OmniCRM
- **Solución:**
  - Verifique que los nombres de las variables coincidan exactamente (sensible a mayúsculas)
  - Utilice el formato `{{ var:variable_name }}`
  - Verifique que OmniCRM esté pasando los datos de la variable en la llamada a la API
  - Pruebe con datos de muestra en Mailjet

El PDF de la factura no se adjunta

- **Causa:** Fallo en la generación del PDF o tamaño de archivo demasiado grande
- **Solución:**
  - Verifique los registros de generación de facturas
  - Verifique que la plantilla de factura se renderice correctamente
  - Asegúrese de que el PDF esté por debajo de 15MB (límite de Mailjet)
  - Pruebe la generación del PDF de la factura por separado

Necesita una lista de contactos de marketing en Mailjet

- **Causa:** OmniCRM ya no envía contactos a Mailjet automáticamente
- **Solución:**
  - Recoja opt-ins a través del flujo de consentimiento de marketing (`granted` estado)
  - Exportar desde **Clientes → Todos los Contactos** con filtro de consentimiento, o `GET /crm/contact/export?filters={"marketing_consent_status":["granted"]}`
  - Importe el CSV en Mailjet utilizando su interfaz de importación de contactos

Documentación Relacionada

- `administration_configuration` - Referencia completa de configuración de Mailjet
- `payments_invoices` - Generación de facturas y entrega de correos electrónicos
- `authentication_flows` - Correos electrónicos de restablecimiento de contraseña y verificación
- `customer_care` - Correos electrónicos de bienvenida del portal de Autocuidado
- Monitoreo & Métricas - Métricas de API de Mailjet, seguimiento de entrega de correos electrónicos y monitoreo de rendimiento

Lectura Adicional

- Documentación de Mailjet: <https://dev.mailjet.com/>
- Referencia de API de Mailjet: <https://dev.mailjet.com/email/reference/>

API de Notificaciones para Sistemas Externos

La API de Notificaciones de OmniCRM permite a sistemas externos e integraciones de terceros enviar notificaciones dentro de la aplicación a los usuarios. Esto permite que alertas automatizadas, actualizaciones de estado e información importante se entreguen directamente a los usuarios dentro de la interfaz de OmniCRM.

Véase también: Uso de la API para métodos de autenticación, Sistema de Tickets de Soporte para notificaciones relacionadas con tickets, Monitoreo y Métricas para el monitoreo del sistema.

Descripción General

La API de Notificaciones proporciona una forma flexible para que cualquier sistema envíe notificaciones a los usuarios de OmniCRM. Las notificaciones aparecen en el menú desplegable de notificaciones del usuario en la interfaz de OmniCRM y pueden enlazar opcionalmente a páginas o recursos específicos.

Características Clave

1. **Integración Flexible**: Cualquier sistema puede enviar notificaciones a través de la API
2. **Múltiples Tipos de Notificación**: Soporte para varios tipos de eventos (provisionamiento, facturación, tickets, personalizados)
3. **Metadatos Ricos**: Incluir datos estructurados con las notificaciones
4. **Navegación por Enlace**: Las notificaciones pueden enlazar a páginas específicas
5. **Autenticación por Clave API**: Integración segura de servidor a servidor
6. **Entrega en Tiempo Real**: Las notificaciones aparecen inmediatamente en la interfaz de usuario

7. **Almacenamiento Persistente**: El sistema almacena notificaciones y las hace recuperables

Casos de Uso Comunes

Sistemas de Provisionamiento:

- Notificar a los usuarios cuando se completa la activación del servicio
- Alertar sobre fallos en el provisionamiento
- Actualizar a los usuarios sobre tareas de provisionamiento de larga duración

Sistemas de Facturación y Pago:

- Alertar a los usuarios cuando se generan nuevas facturas
- Confirmar el procesamiento de pagos
- Advertir sobre próximas expiraciones de servicio

Monitoreo y Alertas:

- Enviar alertas de salud del sistema a los administradores
- Notificar sobre violaciones de umbrales
- Alertar sobre eventos críticos

Integraciones Personalizadas:

- Sistemas de flujo de trabajo externos
- Integraciones de servicios de terceros
- Sistemas de informes automatizados

Autenticación

La API de Notificaciones soporta dos métodos de autenticación:

Autenticación por Clave API (Recomendada para Sistemas Externos)

Las claves API proporcionan autenticación segura y de larga duración para integraciones de servidor a servidor. Este es el método recomendado para sistemas externos.

Configurar Clave API:

Agrega tu clave API a `crm_config.yaml`:

```
api_keys:
  your-secure-api-key-minimum-32-characters:
    roles:
      - admin
    description: "Sistema de notificación de provisionamiento"
```

Usar Clave API:

Incluye la clave API en el encabezado `X-API-KEY`:

```
curl -X POST "https://yourcrm.com/crm/notification/send" \
  -H "X-API-KEY: your-secure-api-key-minimum-32-characters" \
  -H "Content-Type: application/json" \
  -d '{
    "user_id": 123,
    "type": "provision_complete",
    "title": "Servicio Activado",
    "message": "Tu servicio móvil ha sido activado con éxito",
    "link": "/services/167"
  }'
```

Autenticación por Token Bearer

Para sistemas interactivos o notificaciones iniciadas por el usuario, utiliza la autenticación por token Bearer.

Consulta Uso de la API para detalles sobre cómo obtener tokens Bearer.

Endpoint de la API

Enviar Notificación

Endpoint: `POST /crm/notification/send`

Autenticación: Clave API (X-API-KEY) o Token Bearer

Descripción: Enviar una notificación a un usuario específico

Cuerpo de la Solicitud

```
{
  "user_id": 123,
  "type": "provision_complete",
  "title": "Servicio Activado",
  "message": "Tu servicio móvil ha sido activado con éxito",
  "link": "/services/167",
  "metadata": {
    "service_id": 167,
    "provision_job_id": "prov-2026-00042",
    "duration_seconds": 45
  },
  "related_ticket_id": 42,
  "related_user_id": 5
}
```

Parámetros de Solicitud

Campo	Tipo	Requerido	Descripción
`user_id`	entero	**Sí**	ID del usuario a notificar
`type`	cadena	**Sí**	Tipo de notificación (ver tipos a continuación)
`title`	cadena	**Sí**	Título de la notificación (máx. 255 caracteres)
`message`	cadena	No	Mensaje detallado de la notificación
`link`	cadena	No	URL/ruta a navegar al hacer clic
`metadata`	objeto	No	Datos estructurados adicionales (JSON)
`related_ticket_id`	entero	No	ID del ticket asociado
`related_user_id`	entero	No	ID del usuario relacionado (por ejemplo, quien causó el evento)

Tipos de Notificación

Tipo	Descripción	Caso de Uso Común
`provision_complete`	Provisionamiento completado con éxito	Playbook de Ansible finalizado
`provision_failed`	Fallo en el provisionamiento	Error de activación de servicio
`ticket_mention`	Usuario @mencionado en ticket	Alerta de colaboración
`ticket_assigned`	Ticket asignado al usuario	Flujo de trabajo de soporte
`ticket_updated`	Ticket actualizado	Notificación de suscriptor de ticket
`customer_reply`	Cliente respondió al ticket	Alerta de soporte
`system_alert`	Alerta a nivel de sistema	Evento crítico del sistema
`billing_invoice`	Nueva factura generada	Integración del sistema de facturación
`billing_payment`	Pago procesado	Webhook de pasarela de pago
`service_expiring`	Servicio por expirar pronto	Monitoreo automatizado
`custom`	Tipo de notificación personalizada	Cualquier otro propósito

Respuesta de Éxito

Estado: `201 Created`


```
{
  "result": "Success",
  "message": "Notificación enviada con éxito",
  "notification": {
    "notification_id": 1234,
    "user_id": 123,
    "type": "provision_complete",
    "title": "Servicio Activado",
    "message": "Tu servicio móvil ha sido activado con éxito",
    "link": "/services/167",
    "metadata": {
      "service_id": 167,
      "provision_job_id": "prov-2026-00042",
      "duration_seconds": 45
    },
    "is_read": false,
    "created_at": "2026-01-11T14:30:00Z"
  }
}
```


Respuestas de Error

Campo Requerido Faltante (400 Bad Request):


```
{
  "result": "Failed",
  "reason": "user_id es requerido"
}
```


Usuario No Encontrado (404 Not Found):


```
{
  "result": "Failed",
  "reason": "Usuario 999 no encontrado"
}
```


Autenticación Fallida (401 Unauthorized):

```
{
  "message": "Clave API inválida"
}
```

Ejemplos de Código

Ejemplo en Python

```python
import requests

# Configuración
CRM_URL = 'https://yourcrm.com'
API_KEY = 'your-secure-api-key-minimum-32-characters'

def send_notification(user_id, notification_type, title,
message=None, link=None, metadata=None):
    """Enviar una notificación a un usuario a través de la API de
OmniCRM"""

    headers = {
        "Content-Type": "application/json",
        "X-API-KEY": API_KEY
    }

    payload = {
        "user_id": user_id,
        "type": notification_type,
        "title": title
    }

    # Agregar campos opcionales
    if message:
        payload["message"] = message
    if link:
        payload["link"] = link
    if metadata:
        payload["metadata"] = metadata

    response = requests.post(
        f"{CRM_URL}/crm/notification/send",
        headers=headers,
        json=payload
    )

    if response.status_code == 201:
        print(f"✓ Notificación enviada con éxito:
```

```
{response.json()}")
        return response.json()
    else:
        print(f"✗ Falló al enviar la notificación:
{response.text}")
        response.raise_for_status()

# Ejemplo de uso
if __name__ == "__main__":
    # Notificar al usuario sobre la finalización del
provisionamiento
    send_notification(
        user_id=123,
        notification_type="provision_complete",
        title="Servicio Móvil Activado",
        message="El ID del servicio 167 ha sido provisionado y
activado con éxito",
        link="/services/167",
        metadata={
            "service_id": 167,
            "provision_job_id": "prov-2026-00042",
            "duration_seconds": 45,
            "playbook": "activacion_servicio_movil"
        }
    )
```

Ejemplo en Bash/cURL

```
#!/bin/bash

# Configuración
CRM_URL="https://yourcrm.com"
API_KEY="your-secure-api-key-minimum-32-characters"

# Enviar notificación
curl -X POST "${CRM_URL}/crm/notification/send" \
  -H "X-API-KEY: ${API_KEY}" \
  -H "Content-Type: application/json" \
  -d '{
    "user_id": 123,
    "type": "provision_complete",
    "title": "Servicio Activado",
    "message": "Tu servicio móvil ha sido activado con éxito",
    "link": "/services/167",
    "metadata": {
      "service_id": 167,
      "provision_job_id": "prov-2026-00042",
      "duration_seconds": 45
    }
  }'
```

Ejemplo en Node.js

```
const axios = require('axios');

const CRM_URL = 'https://yourcrm.com';
const API_KEY = 'your-secure-api-key-minimum-32-characters';

async function sendNotification(userId, type, title, options = {})
{
    try {
        const response = await axios.post(
            `${CRM_URL}/crm/notification/send`,
            {
                user_id: userId,
                type: type,
                title: title,
                message: options.message,
                link: options.link,
                metadata: options.metadata,
                related_ticket_id: options.relatedTicketId,
                related_user_id: options.relatedUserId
            },
            {
                headers: {
                    'X-API-KEY': API_KEY,
                    'Content-Type': 'application/json'
                }
            }
        );

        console.log('✓ Notificación enviada:', response.data);
        return response.data;
    } catch (error) {
        console.error('✗ Falló al enviar la notificación:',
error.response?.data || error.message);
        throw error;
    }
}

// Ejemplo de uso
sendNotification(
    123,
    'provision_complete',
```

```
    'Servicio Activado',
    {
        message: 'Tu servicio móvil ha sido activado con éxito',
        link: '/services/167',
        metadata: {
            service_id: 167,
            provision_job_id: 'prov-2026-00042',
            duration_seconds: 45
        }
    }
);
```

Escenarios de Integración

Playbook de Provisionamiento de Ansible

Notificar a los usuarios cuando los playbooks de Ansible se completen:

```python
# Al final de tu plugin de callback de Ansible o script post-playbook
import requests

def notify_provisioning_complete(service_id, user_id, success=True):
    """Notificar al usuario sobre el resultado del provisionamiento"""

    notification_type = "provision_complete" if success else "provision_failed"
    title = "Servicio Activado" if success else "Fallo en la Activación del Servicio"
    message = f"El servicio {service_id} se ha {'completado con éxito' if success else 'encontrado con un error'}"

    requests.post(
        "https://yourcrm.com/crm/notification/send",
        headers={"X-API-KEY": "your-api-key"},
        json={
            "user_id": user_id,
            "type": notification_type,
            "title": title,
            "message": message,
            "link": f"/services/{service_id}",
            "metadata": {
                "service_id": service_id,
                "playbook": "activacion_servicio_movil",
                "success": success
            }
        }
    )
```

Integración del Sistema de Facturación

Enviar notificaciones de facturas desde el sistema de facturación:

```
def notify_invoice_generated(customer_id, invoice_id, amount,
due_date):
    """Notificar al contacto del cliente sobre la nueva factura"""

    # Encontrar al usuario asociado con el cliente
    user_id = get_customer_primary_contact_user_id(customer_id)

    if user_id:
        requests.post(
            "https://yourcrm.com/crm/notification/send",
            headers={"X-API-KEY": "your-billing-api-key"},
            json={
                "user_id": user_id,
                "type": "billing_invoice",
                "title": f"Nueva Factura #{invoice_id}",
                "message": f"La factura por ${amount} ya está
disponible. Fecha de vencimiento: {due_date}",
                "link": f"/invoices/{invoice_id}",
                "metadata": {
                    "invoice_id": invoice_id,
                    "amount": amount,
                    "due_date": due_date,
                    "customer_id": customer_id
                }
            }
        )
```

Integración del Sistema de Monitoreo

Enviar alertas desde sistemas de monitoreo:

```
def send_monitoring_alert(admin_user_ids, alert_title,
alert_message, severity="warning"):
    """Enviar alerta del sistema a los administradores"""

    for user_id in admin_user_ids:
        requests.post(
            "https://yourcrm.com/crm/notification/send",
            headers={"X-API-KEY": "your-monitoring-api-key"},
            json={
                "user_id": user_id,
                "type": "system_alert",
                "title": alert_title,
                "message": alert_message,
                "metadata": {
                    "severity": severity,
                    "alert_time": datetime.now().isoformat(),
                    "source": "prometheus_alertmanager"
                }
            }
        )

# Ejemplo: Alerta de espacio en disco
send_monitoring_alert(
    admin_user_ids=[1, 2, 3],
    alert_title="⚠ Advertencia de Espacio en Disco",
    alert_message="El uso del disco del servidor de base de datos
ha superado el 85%",
    severity="warning"
)
```

Webhook de Pasarela de Pago

Notificar a los usuarios cuando se procesen los pagos:

```
from flask import Flask, request

app = Flask(__name__)

@app.route('/webhook/payment', methods=['POST'])
def payment_webhook():
    """Manejar el webhook de la pasarela de pago"""

    payment_data = request.json

    if payment_data['status'] == 'succeeded':
        # Notificar al usuario sobre el pago exitoso
        requests.post(
            "https://yourcrm.com/crm/notification/send",
            headers={"X-API-KEY": "your-payment-webhook-api-key"},
            json={
                "user_id": payment_data['user_id'],
                "type": "billing_payment",
                "title": "Pago Recibido",
                "message": f"El pago de ${payment_data['amount']}
ha sido procesado con éxito",
                "link": f"/payments/{payment_data['payment_id']}",
                "metadata": {
                    "payment_id": payment_data['payment_id'],
                    "amount": payment_data['amount'],
                    "payment_method":
payment_data['payment_method']
                }
            }
        )

    return {"status": "ok"}
```

Monitor de Expiración de Servicios

Script automatizado para notificar a los usuarios sobre servicios que están por expirar:

```
import datetime
from sqlalchemy import create_engine, text

def check_expiring_services():
    """Verificar servicios que expiran en los próximos 7 días y
notificar a los usuarios"""

    engine = create_engine("postgresql://...")

    # Consultar servicios que expiran pronto
    query = text("""
        SELECT s.service_id, s.end_date, u.id as user_id,
s.service_name
        FROM services s
        JOIN customer_contacts cc ON s.customer_id =
cc.customer_id
        JOIN users u ON cc.contact_id = u.contact_id
        WHERE s.end_date BETWEEN CURRENT_DATE AND CURRENT_DATE +
INTERVAL '7 days'
        AND s.deleted = false
    """)

    with engine.connect() as conn:
        results = conn.execute(query)

        for row in results:
            days_until_expiry = (row.end_date -
datetime.date.today()).days

            requests.post(
                "https://yourcrm.com/crm/notification/send",
                headers={"X-API-KEY": "your-automation-api-key"},
                json={
                    "user_id": row.user_id,
                    "type": "service_expiring",
                    "title": f"Servicio por Expirar en
{days_until_expiry} Días",
                    "message": f"Tu {row.service_name} expirará el
{row.end_date}",
                    "link": f"/services/{row.service_id}",
                    "metadata": {
                        "service_id": row.service_id,
                        "days_remaining": days_until_expiry,
```

```
                        "expiry_date": str(row.end_date)
                    }
                }
            )

# Ejecutar diariamente a través de cron
if __name__ == "__main__":
    check_expiring_services()
```

Mejores Prácticas

Seguridad de la Clave API

1. **Generar Claves Fuertes**: Utilizar generadores aleatorios criptográficamente seguros:

   ```
   openssl rand -base64 48
   ```

2. **Separar Claves por Sistema**: Utilizar diferentes claves API para diferentes sistemas externos

3. **Documentar el Propósito de la Clave**: Siempre incluir una descripción en `crm_config.yaml`

4. **Rotar Claves Regularmente**: Cambiar las claves API periódicamente (por ejemplo, trimestralmente)

5. **Nunca Comprometer Claves**: Mantener las claves API fuera del control de versiones

6. **Permisos Mínimos**: Solo otorgar roles necesarios a cada clave API

Diseño de Notificaciones

1. **Títulos Claros**: Hacer que los títulos sean descriptivos y accionables (por ejemplo, "Servicio Activado" no "Actualización")

2. **Mensajes Concisos**: Mantener los mensajes breves pero informativos
3. **Incluir Enlaces**: Siempre proporcionar un enlace para que los usuarios obtengan más detalles
4. **Usar Metadatos**: Almacenar datos estructurados para referencia futura y depuración
5. **Elegir el Tipo Correcto**: Usar tipos de notificación apropiados para la categorización
6. **Probar Notificaciones**: Probar con usuarios reales antes de implementar en producción

Manejo de Errores

1. **Manejar Fallos de Manera Elegante**: No permitir que los fallos de notificación rompan los flujos de trabajo principales
2. **Registrar Todos los Intentos**: Registrar intentos de notificación para depuración
3. **Reintentar en Errores Transitorios**: Implementar lógica de reintento para problemas de red
4. **Validar IDs de Usuario**: Verificar que el usuario exista antes de enviar
5. **Verificar Respuestas de la API**: Siempre verificar los códigos de estado de respuesta

Rendimiento

1. **Agrupar Cuando Sea Posible**: Para múltiples usuarios, considerar enviar en paralelo
2. **Procesamiento Asíncrono**: Enviar notificaciones de manera asíncrona para evitar bloqueos

3. **Limitación de Tasa**: Implementar limitación de tasa para prevenir abusos de la API

4. **Agrupamiento de Conexiones**: Reutilizar conexiones HTTP para un mejor rendimiento

Pruebas

Probar Configuración de Clave API

Antes de implementar, verifica que tu clave API funcione:

```
# Probar autenticación
curl -X POST "https://yourcrm.com/crm/notification/send" \
  -H "X-API-KEY: your-api-key" \
  -H "Content-Type: application/json" \
  -d '{
    "user_id": 1,
    "type": "custom",
    "title": "Notificación de Prueba",
    "message": "Esta es una notificación de prueba para verificar
la autenticación de la clave API"
  }'
```

Respuesta esperada: `201 Created` con detalles de la notificación

Probar en Desarrollo

Usar un ID de usuario de prueba para verificar que las notificaciones aparezcan:

```python
# Enviar notificación de prueba
response = send_notification(
    user_id=1,  # Tu usuario de prueba
    notification_type="custom",
    title="Prueba de Integración",
    message="Probando la integración de la API de notificaciones"
)

print(f"ID de Notificación: {response['notification']
['notification_id']}")
```

Luego inicia sesión en OmniCRM como ese usuario y verifica que la notificación aparezca en el menú desplegable de notificaciones.

Solución de Problemas

Problema: 401 Unauthorized

- **Causa:** Clave API inválida o faltante
- **Solución:** Verificar la clave API en `crm_config.yaml` y en el encabezado de la solicitud
- **Verificar:** Asegurarse de que la clave API tenga al menos 32 caracteres

Problema: 404 Usuario No Encontrado

- **Causa:** `user_id` inválido
- **Solución:** Verificar que el usuario exista en la base de datos
- **Verificar:** Consultar la tabla de usuarios: `SELECT id FROM users WHERE id = ?`

Problema: 400 Bad Request - "user_id es requerido"

- **Causa:** Campo requerido faltante en la solicitud
- **Solución:** Incluir todos los campos requeridos: user_id, type, title
- **Verificar:** Validar la estructura del payload JSON

Problema: Notificación enviada pero no aparece

- **Causa:** Usuario no conectado o caché del navegador
- **Solución:** Actualizar la página o limpiar la caché
- **Verificar:** Consultar la base de datos: `SELECT * FROM notifications WHERE user_id = ? ORDER BY created_at DESC`

Problema: 500 Internal Server Error

- **Causa:** Error del lado del servidor
- **Solución:** Verificar los registros del servidor CRM para más detalles
- **Verificar:** Revisar `/var/log/omnicrm/api.log` en busca de errores

Resumen de Referencia de la API

Endpoints

Método	Endpoint	Descripción	Auth
POST	`/crm/notification/send`	Enviar notificación a usuario	Clave API o Bearer
GET	`/crm/notification/`	Obtener notificaciones del usuario actual	Bearer
GET	`/crm/notification/unread-count`	Obtener conteo de no leídas	Bearer
POST	`/crm/notification/{id}/mark-read`	Marcar notificación como leída	Bearer
POST	`/crm/notification/mark-all-read`	Marcar todas como leídas	Bearer
DELETE	`/crm/notification/{id}`	Eliminar notificación	Bearer

Permisos Requeridos

- **Enviar Notificación:** Requiere clave API válida o rol de administrador
- **Ver Notificaciones:** El usuario solo puede ver sus propias notificaciones
- **Gestionar Notificaciones:** El usuario solo puede gestionar sus propias notificaciones

Documentación Relacionada

- Uso de la API: Guía completa de autenticación de la API
- Sistema de Tickets de Soporte: Notificaciones de tickets y @menciones
- Sistema de Provisionamiento: Integración de Ansible
- Monitoreo y Métricas: Monitoreo y alertas del sistema
- RBAC: Roles y permisos

Registro de cambios

Esto contiene los últimos 50 cambios realizados en el stack de software de OmniCRM o sus dependencias.

Nota: Esto no rastrea cambios en configuraciones individuales de clientes.

Monitoreo y Métricas

La API de OmniCRM proporciona métricas completas basadas en Prometheus para monitorear el rendimiento de la aplicación, las operaciones comerciales y las integraciones externas. Todas las métricas se exponen en el endpoint `/crm/metrics` en formato de exposición de Prometheus.

Descripción General

El sistema de métricas rastrea:

- **Operaciones de Aprovisionamiento** - Ejecución de trabajos, duración, tasas de éxito/fallo
- **Rendimiento de la Base de Datos** - Tiempos de consulta, salud del pool de conexiones
- **Integraciones Externas** - Llamadas a la API de OCS/CGRateS, Stripe, Mailjet
- **Trabajos en Segundo Plano** - Ejecución y rendimiento de tareas asíncronas
- **Solicitudes HTTP** - Uso de endpoints de API y tiempos de respuesta (generados automáticamente)

Endpoint de Métricas

URL: `http://your-omnicrm-api:5000/crm/metrics`

Formato: Formato de exposición de Prometheus

Autenticación: Prometheus puede raspar el endpoint de métricas, que es accesible públicamente. En producción, restrinja el acceso con reglas de firewall o autenticación de proxy inverso.

Categorías de Métricas

1. Métricas de Aprovisionamiento

Las métricas de aprovisionamiento rastrean la ejecución de playbooks de Ansible que aprovisionan servicios, gestionan inventarios y configuran sistemas externos. Consulte Sistema de Aprovisionamiento y Playbooks de Ansible para más detalles.

omnicrm_provision_jobs_total

Tipo: Contador

Etiquetas:

- `status` - Estado de finalización del trabajo: `success`, `failed`, `running`

Descripción: Número total de trabajos de aprovisionamiento creados. Se incrementa cuando los trabajos de aprovisionamiento se completan con su estado final.

omnicrm_provision_job_duration_seconds

Tipo: Histograma

Etiquetas:

- `playbook` - Nombre del playbook de Ansible ejecutado
- `status` - Estado de finalización del trabajo: `success`, `failed`

Buckets: `[1, 5, 10, 30, 60, 120, 180, 300, 600]` segundos (1s a 10min)

Descripción: Tiempo tomado para completar los trabajos de aprovisionamiento. Registra la duración de la ejecución completa del playbook desde el inicio hasta el final.

omnicrm_provision_jobs_active

Tipo: Medidor

Descripción: Número de trabajos de aprovisionamiento que se están ejecutando actualmente. Se incrementa cuando un trabajo comienza, se decrementa cuando se completa.

omnicrm_provision_tasks_total

Tipo: Contador

Etiquetas:

- `playbook` - Nombre del playbook de Ansible
- `status` - Resultado de la tarea: `ok`, `failed`, `ignored`

Descripción: Número total de tareas de Ansible ejecutadas dentro de los playbooks. Se incrementa por cada tarea individual que se completa (éxito o fallo).

omnicrm_provision_errors_total

Tipo: Contador

Etiquetas:

- `error_type` - Tipo de error: `fatal`, `task_failed`, `timeout`
- `playbook` - Nombre del playbook de Ansible

Descripción: Número total de errores de aprovisionamiento por tipo. Se incrementa cuando las tareas de aprovisionamiento fallan o ocurren errores fatales durante la ejecución.

---

2. Métricas de Base de Datos

Las métricas de base de datos monitorean el rendimiento de las consultas y la salud del pool de conexiones. OmniCRM utiliza SQLAlchemy con instrumentación automática. Consulte Arquitectura del Sistema para detalles del modelo de datos.

omnicrm_db_query_duration_seconds

Tipo: Histograma

Etiquetas:

- `operation` - Tipo de operación SQL: `SELECT`, `INSERT`, `UPDATE`, `DELETE`
- `table` - Nombre de la tabla de la base de datos

Buckets: `[0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1.0, 5.0, 10.0]` segundos

Descripción: Tiempo de ejecución de consultas a la base de datos. Se rastrea automáticamente a través de los oyentes de eventos de SQLAlchemy en cada consulta.

omnicrm_db_pool_size

Tipo: Medidor

Descripción: Tamaño actual del pool de conexiones de la base de datos. El número total de conexiones en el pool (tanto prestadas como disponibles).

omnicrm_db_pool_overflow

Tipo: Medidor

Descripción: Desbordamiento actual del pool de conexiones de la base de datos. El número de conexiones creadas más allá del límite normal del tamaño del pool.

omnicrm_db_pool_connections_checked_out

Tipo: Medidor

Descripción: Número de conexiones actualmente prestadas del pool y en uso por el código de la aplicación.

omnicrm_db_errors_total

Tipo: Contador

Etiquetas:

- `error_type` - Tipo de error de base de datos
- `operation` - Operación que causó el error: `connection_error`, `query_error`, etc.

Descripción: Número total de errores de base de datos.

Estado: Definido pero no utilizado activamente en el código actual (reservado para uso futuro)

---

3. Métricas de OCS/CGRateS

Las métricas de OCS (Sistema de Carga en Línea) rastrean las interacciones con el motor de facturación CGRateS. Consulte Descripción General de Facturación y Integración de CGRateS para más detalles.

omnicrm_ocs_api_calls_total

Tipo: Contador

Etiquetas:

- `method` - Método de la API de OCS: `GetBalance`, `SetBalance`, `SetAccount`, `RemoveAccount`, etc.
- `status` - Resultado de la llamada: `success`, `failed`

Descripción: Número total de llamadas a la API de OCS/CGRateS. Se incrementa por cada llamada a la API asíncrona al sistema OCS.

omnicrm_ocs_api_duration_seconds

Tipo: Histograma

Etiquetas:

- `method` - Método de la API de OCS: `GetBalance`, `SetBalance`, `GetAccount`, etc.

Buckets: `[0.1, 0.25, 0.5, 1.0, 2.0, 5.0, 10.0, 30.0]` segundos

Descripción: Duración de la llamada a la API de OCS/CGRateS. Registra el tiempo tomado para cada llamada a la API, incluyendo la latencia de la red.

omnicrm_ocs_api_errors_total

Tipo: Contador

Etiquetas:

- `method` - Método de la API de OCS que falló
- `error_type` - Categoría de error: `timeout`, `connection_error`, `json_error`, `http_error`, etc.

Descripción: Número total de errores de la API de OCS/CGRateS. Se incrementa cuando las llamadas a la API de OCS fallan con tipos de error específicos.

omnicrm_ocs_balance_queries_total

Tipo: Contador

Etiquetas:

- `query_type` - Tipo de consulta de saldo: `single_account`, `multiple_accounts`

Descripción: Número total de consultas de saldo a OCS. Se utiliza para rastrear el uso de operaciones de consulta de saldo.

omnicrm_ocs_action_plan_operations_total

Tipo: Contador

Etiquetas:

- `operation` - Operación del plan de acción: `create`, `remove`, `query`

Descripción: Número total de operaciones del plan de acción. Rastrear la creación, eliminación y consulta de planes de acción de CGRateS para cargos recurrentes.

---

4. Métricas de Pago de Stripe

Las métricas de Stripe rastrean las operaciones de procesamiento de pagos. Consulte Guía del Sistema de Pagos y Métodos de Pago para detalles de integración.

omnicrm_stripe_api_calls_total

Tipo: Contador

Etiquetas:

- `operation` - Operación de Stripe: `create_customer`, `charge`, `refund`, `update_payment_method`, etc.
- `status` - Resultado de la operación: `success`, `failed`

Descripción: Número total de llamadas a la API de Stripe. Se incrementa por cada operación de procesamiento de pagos.

omnicrm_stripe_api_duration_seconds

Tipo: Histograma

Etiquetas:

- `operation` - Tipo de operación de Stripe

Buckets: `[0.1, 0.5, 1.0, 2.0, 5.0, 10.0, 30.0]` segundos

Descripción: Duración de la llamada a la API de Stripe, incluyendo la latencia de la red.

omnicrm_stripe_api_errors_total

Tipo: Contador

Etiquetas:

- `operation` - Operación de Stripe que falló
- `error_type` - Categoría de error: `card_declined`, `network_error`, `api_error`, etc.

Descripción: Número total de errores de la API de Stripe. Se incrementa cuando las operaciones de pago fallan.

omnicrm_stripe_payment_amount_cents

Tipo: Contador

Etiquetas:

- `payment_type` - Dirección del pago: `charge`, `refund`

Descripción: Monto total de pago procesado a través de Stripe en centavos. Útil para rastrear el volumen de transacciones y los ingresos.

omnicrm_stripe_payment_failures_total

Tipo: Contador

Etiquetas:

- `reason` - Razón del fallo: `card_declined`, `insufficient_funds`, `expired_card`, etc.

Descripción: Número total de fallos de pago de Stripe categorizados por código de rechazo.

---

5. Métricas de Correo Electrónico de Mailjet

Las métricas de Mailjet rastrean las operaciones de entrega de correos electrónicos. Consulte Integración de Mailjet para detalles de configuración.

omnicrm_mailjet_api_calls_total

Tipo: Contador

Etiquetas:

- `email_type` - Tipo de plantilla de correo electrónico: `welcome`, `user_welcome`, `invoice`, `notification`

- `status` - Resultado de la entrega: `success`, `failed`

Descripción: Número total de llamadas a la API de Mailjet. Rastreado a través del decorador `@track_mailjet_call`.

omnicrm_mailjet_api_duration_seconds

Tipo: Histograma

Etiquetas:

- `email_type` - Tipo de plantilla de correo electrónico

Buckets: `[0.1, 0.5, 1.0, 2.0, 5.0, 10.0, 30.0]` segundos

Descripción: Duración de la llamada a la API de Mailjet. Tiempo tomado para enviar el correo electrónico a través de la API de Mailjet (no el tiempo de entrega).

omnicrm_mailjet_api_errors_total

Tipo: Contador

Etiquetas:

- `email_type` - Tipo de plantilla de correo electrónico
- `error_type` - Categoría de error

Descripción: Número total de errores de la API de Mailjet. Se incrementa cuando los envíos de correos electrónicos fallan.

omnicrm_mailjet_emails_sent_total

Tipo: Contador

Etiquetas:

- `email_type` - Tipo de plantilla de correo electrónico
- `template_id` - ID de plantilla de Mailjet

Descripción: Número total de correos electrónicos enviados con éxito a través de Mailjet. Distinto de las llamadas a la API, ya que una llamada puede enviar a múltiples destinatarios.

omnicrm_mailjet_email_recipients_total

Tipo: Contador

Etiquetas:

- `email_type` - Tipo de plantilla de correo electrónico

Descripción: Número total de destinatarios de correo electrónico en todos los correos enviados.

---

6. Métricas de Trabajos en Segundo Plano

Las métricas de trabajos en segundo plano rastrean operaciones asíncronas como cadenas de playbooks y tareas programadas. Consulte Sistema de Aprovisionamiento para detalles sobre trabajos en segundo plano.

omnicrm_background_jobs_total

Tipo: Contador

Etiquetas:

- `job_type` - Categoría de trabajo: `playbook_single`, `playbook_chain`, `periodic_task`

Descripción: Número total de trabajos en segundo plano iniciados. Rastreado a través del administrador de contexto `BackgroundJobTimer`.

omnicrm_background_jobs_active

Tipo: Medidor

Etiquetas:

- `job_type` - Categoría de trabajo

Descripción: Número de trabajos en segundo plano que se están ejecutando actualmente. Se incrementa al inicio del trabajo, se decrementa al completar el trabajo.

omnicrm_background_job_duration_seconds

Tipo: Histograma

Etiquetas:

- `job_type` - Categoría de trabajo
- `status` - Resultado del trabajo: `success`, `failed`

Buckets: `[1, 5, 10, 30, 60, 120, 180, 300, 600, 1800, 3600]` segundos (1s a 1hr)

Descripción: Tiempo de ejecución del trabajo en segundo plano. Incluye la duración total de operaciones de múltiples pasos.

omnicrm_background_job_errors_total

Tipo: Contador

Etiquetas:

- `job_type` - Categoría de trabajo
- `error_type` - Categoría de error

Descripción: Número total de errores de trabajos en segundo plano. Se incrementa cuando los trabajos en segundo plano fallan con excepciones.

---

7. Métricas HTTP de Flask (Generadas Automáticamente)

Estas métricas son generadas automáticamente por la biblioteca `prometheus-flask-exporter` y rastrean todas las solicitudes HTTP a la API.

flask_http_request_duration_seconds

Tipo: Histograma

Etiquetas:

- `method` - Método HTTP: `GET`, `POST`, `PUT`, `DELETE`, etc.
- `endpoint` - Nombre de la ruta de Flask
- `status` - Código de estado HTTP

Descripción: Duración de la solicitud HTTP para todos los endpoints de la API. Instrumentado automáticamente.

flask_http_request_total

Tipo: Contador

Etiquetas:

- `method` - Método HTTP
- `endpoint` - Nombre de la ruta de Flask
- `status` - Código de estado HTTP

Descripción: Número total de solicitudes HTTP por endpoint, método y código de estado.

flask_http_request_exceptions_total

Tipo: Contador

Etiquetas:

- `method` - Método HTTP
- `endpoint` - Nombre de la ruta de Flask

Descripción: Número total de excepciones no manejadas en solicitudes HTTP. Indica errores o fallos inesperados.

---

8. Métricas de Errores de API (Reservadas)

Estas métricas están definidas pero no están actualmente instrumentadas. Están reservadas para uso futuro.

omnicrm_api_errors_total

Tipo: Contador

Etiquetas:

- `endpoint` - Endpoint de la API
- `error_type` - Categoría de error
- `status_code` - Código de estado HTTP

Estado: Definido pero no utilizado activamente

omnicrm_api_auth_failures_total

Tipo: Contador

Etiquetas:

- `auth_method` - Método de autenticación: `jwt`, `api_key`, `ip_whitelist`
- `reason` - Razón del fallo

Estado: Definido pero no utilizado activamente

---

9. Métrica de Información de la Aplicación

omnicrm_api

Tipo: Información

Etiquetas:

- `version` - Cadena de versión de la API
- `environment` - Nombre del entorno: `production`, `staging`, `development`

Descripción: Métrica de información de la API de OmniCRM. Se establece una vez al inicio de la aplicación con información de versión y entorno.

Actualizaciones Periódicas

update_db_pool_metrics(engine)

Se llama automáticamente cada 30 segundos para actualizar los medidores del pool de conexiones de la base de datos.

Configuración de Prometheus

Configuración de Raspeo

Agregue OmniCRM a su `prometheus.yml`:

```yaml
scrape_configs:
  - job_name: 'omnicrm-api'
    scrape_interval: 15s
    scrape_timeout: 10s
    static_configs:
      - targets: ['omnicrm-api:5000']
    metrics_path: '/crm/metrics'
```

Ejemplo de Alertas

Alta Tasa de Fallos de Aprovisionamiento

```yaml
- alert: HighProvisionFailureRate
  expr: |
    (
      rate(omnicrm_provision_jobs_total{status="failed"}[5m]) /
      rate(omnicrm_provision_jobs_total[5m])
    ) > 0.1
  for: 5m
  labels:
    severity: warning
  annotations:
    summary: "Alta tasa de fallos en trabajos de
aprovisionamiento"
    description: "{{ $value | humanizePercentage }} de los
trabajos de aprovisionamiento están fallando"
```

Pool de Conexiones de Base de Datos Agotado

```yaml
- alert: DatabasePoolExhausted
  expr: omnicrm_db_pool_overflow > 0
  for: 2m
  labels:
    severity: critical
  annotations:
    summary: "Desbordamiento del pool de conexiones de la base de
datos detectado"
    description: "El pool de conexiones está utilizando conexiones
de desbordamiento, puede indicar que el tamaño del pool es
demasiado pequeño"
```

Consultas de Base de Datos Lentas

```yaml
- alert: SlowDatabaseQueries
  expr: |
    histogram_quantile(0.99,
      rate(omnicrm_db_query_duration_seconds_bucket[5m])
    ) > 1.0
  for: 5m
  labels:
    severity: warning
  annotations:
    summary: "Consultas de base de datos lentas detectadas"
    description: "El tiempo de consulta del percentil 99 es {{
$value }}s"
```

API de OCS Caída

```yaml
- alert: OCSAPIDown
  expr: |
    (
      rate(omnicrm_ocs_api_calls_total{status="failed"}[5m]) /
      rate(omnicrm_ocs_api_calls_total[5m])
    ) > 0.5
  for: 2m
  labels:
    severity: critical
  annotations:
    summary: "Tasa de fallos de la API de OCS crítica"
    description: "{{ $value | humanizePercentage }} de las
llamadas a la API de OCS están fallando"
```

Problemas de Pago de Stripe

```yaml
- alert: StripePaymentFailures
  expr: rate(omnicrm_stripe_payment_failures_total[5m]) > 5
  for: 5m
  labels:
    severity: warning
  annotations:
    summary: "Fallos de pago de Stripe elevados"
    description: "{{ $value }} fallos de pago por segundo"
```

Mejores Prácticas

Estrategia de Monitoreo

1. **Configurar Alertas Clave** - Configure alertas para métricas críticas:
    - Tasa de fallos de aprovisionamiento > 10%
    - Agotamiento del pool de conexiones de la base de datos
    - Fallos en la API de OCS/CGRateS
    - Errores en el procesamiento de pagos de Stripe
2. **Rastrear Métricas Comerciales** - Monitorear KPIs operativos:
    - Ingresos procesados a través de Stripe
    - Rendimiento de aprovisionamiento
    - Tasas de entrega de correos electrónicos a clientes
3. **Monitoreo de Rendimiento** - Vigilar la degradación:
    - Percentiles de tiempo de respuesta de la API
    - Rendimiento de consultas de base de datos
    - Latencia de API externas
4. **Planificación de Capacidad** - Utilizar métricas para pronosticar:
    - Dimensionamiento del pool de conexiones de la base de datos
    - Escalado de trabajadores de trabajos en segundo plano
    - Capacidad del servidor de API

Retención de Métricas

Recomendaciones de Retención de Prometheus:

- **15 días** - Métricas de alta resolución (intervalo de raspeo de 15s)
- **90 días** - Métricas muestreadas (agregación de 5m)
- **1 año** - Métricas agregadas a largo plazo (agregación de 1h)

Utilice Thanos, Cortex o VictoriaMetrics para almacenamiento a largo plazo y consulta global.

Solución de Problemas

Métricas No Aparecen

Por defecto, el endpoint `/metrics` solo se expone a fuentes internas (no en espacio de direcciones públicas). Puede cambiar esto en su configuración de nginx si es necesario.

Verifique el endpoint de métricas:

```
curl http://localhost:5000/crm/metrics
```

Verifique que Prometheus pueda raspar:

```
# Verifique la página de objetivos de Prometheus
http://prometheus:9090/targets
```

Faltan Métricas Específicas

Algunas métricas solo se crean después de su primer uso:

- Las métricas de aprovisionamiento aparecen después del primer trabajo de aprovisionamiento
- Las métricas de Stripe aparecen después del primer pago
- Las métricas de OCS aparecen después de la primera operación de facturación

Problemas de Alta Cardinalidad

Si Prometheus es lento o consume memoria excesiva, verifique las etiquetas de alta cardinalidad:

```
# Contar series de tiempo únicas por métrica
count by (__name__) ({__name__=~".+"})
```

Las métricas con más de 10,000 series pueden indicar un problema de cardinalidad.

Resumen de Métricas

Total de Métricas: 34 (31 personalizadas + 3 métricas de Flask generadas automáticamente)

Por Tipo:

- **Contadores:** 20
- **Medidores:** 5
- **Histogramas:** 8
- **Información:** 1

Por Categoría:

- Aprovisionamiento: 5 métricas
- Base de Datos: 5 métricas
- OCS/CGRateS: 5 métricas
- Stripe: 5 métricas
- Mailjet: 5 métricas
- Trabajos en Segundo Plano: 4 métricas
- HTTP/Flask: 3 métricas
- Información de la Aplicación: 1 métrica
- Reservadas (uso futuro): 2 métricas

Documentación Relacionada

- Arquitectura del Sistema - Diseño general del sistema y relaciones de componentes
- Sistema de Aprovisionamiento - Flujo de trabajo de aprovisionamiento y ejecución de trabajos
- Playbooks de Ansible - Estructura y ejecución de playbooks
- Documentación de la API - Endpoints de la API y autenticación
- Descripción General de Facturación - Sistema de facturación y cobro
- Guía del Sistema de Pagos - Integración de Stripe y procesamiento de pagos
- Integración de Mailjet - Configuración del servicio de correo electrónico
- Configuración de Administración - Opciones de configuración del sistema

Gestión de Métodos de Pago

El sistema de Métodos de Pago de OmniCRM permite a los clientes y al personal gestionar de manera segura las tarjetas de pago utilizando **procesamiento de pagos de múltiples proveedores** (Stripe, PayPal, etc.). Los métodos de pago permiten la facturación automática por servicios, pagos únicos y cargos recurrentes sin almacenar datos sensibles de tarjetas en OmniCRM.

Véase también: Guía del Sistema de Pagos, Descripción General de Facturación, Procesamiento de Pagos, Facturas.

Descripción General

El sistema de métodos de pago proporciona:

- **Almacenamiento Seguro de Tarjetas** - Tarjetas tokenizadas por proveedores de pago (Stripe, PayPal), nunca almacenadas en OmniCRM
- **Soporte de Múltiples Proveedores** - Métodos de pago de Stripe y PayPal soportados
- **Múltiples Tarjetas** - Los clientes pueden almacenar múltiples métodos de pago
- **Selección Predeterminada** - Designar el método de pago preferido para cargos automáticos
- **Seguimiento de Vencimientos** - Monitorear y actualizar tarjetas que están por vencer
- **Autoservicio** - Los clientes pueden gestionar sus propias tarjetas a través del Portal de Autoservicio
- **Gestión del Personal** - El personal de soporte puede agregar/eliminar tarjetas en nombre de los clientes

Métodos de Pago Soportados:

- **Tarjetas** (a través de Stripe o PayPal)
  - Tarjetas de Crédito (Visa, Mastercard, American Express, Discover)
  - Tarjetas de Débito
  - Tarjetas Prepagadas (si son soportadas por la red de tarjetas)
- **Cuentas de PayPal** (a través de la integración de PayPal)

No Almacenado en OmniCRM:

Los detalles de la tarjeta son tokenizados por los proveedores de pago y almacenados de manera segura. OmniCRM solo almacena:

- Proveedor de pago (stripe, paypal)
- Marca de la tarjeta (Visa, Mastercard, etc.)
- Últimos 4 dígitos
- Mes/año de vencimiento
- Nombre/apodo del titular de la tarjeta
- Token de método de pago específico del proveedor

Accediendo a Métodos de Pago

Desde la Página del Cliente:

1. Navegar a **Clientes → [Seleccionar Cliente]**
2. Hacer clic en la pestaña **Facturación**
3. Desplazarse a la sección **Métodos de Pago**

O directamente:

Desde el Tablero de Tarjetas que Vencen:

Ver todos los clientes con tarjetas que están por vencer:

Esto muestra una lista a nivel de sistema de tarjetas que vencen en los próximos 60 días.

Lista de Métodos de Pago

La tabla de métodos de pago muestra todas las tarjetas almacenadas para un cliente:

Descripciones de Columnas:

- **Apodo** - Nombre amigable para la tarjeta (por ejemplo, "Tarjeta Personal", "Visa de Trabajo")
- **Emisor** - Marca de la tarjeta y últimos 4 dígitos
- **Vencimiento** - Mes/año de vencimiento (formato MM/YYYY)
- **Agregado** - Fecha en que se agregó la tarjeta a la cuenta
- **Predeterminado** - Marca de verificación indica el método de pago predeterminado para cargos automáticos

Acciones Por Tarjeta:

Cada fila tiene un menú de acciones (⋮) con opciones:

- **Establecer como Predeterminado** - Hacer de este el método de pago predeterminado
- **Eliminar** - Eliminar la tarjeta de la cuenta

Agregando un Método de Pago

Haga clic en **"Agregar Método de Pago"** para abrir el modal de pago seguro.

Paso 1: Ingrese los Detalles de la Tarjeta

El formulario de pago seguro aparece (impulsado por Stripe Elements o PayPal SDK):

Campos Requeridos:

- **Información de la Tarjeta** - Número de tarjeta, vencimiento, CVC (validado por Stripe)
- **Nombre del Titular de la Tarjeta** - Nombre en la tarjeta
- **País/Región** - País de facturación

Campos Opcionales:

- **Apodo de la Tarjeta** - Etiqueta amigable para distinguir entre tarjetas

Seguridad:

- Los detalles de la tarjeta se ingresan directamente en un iframe seguro alojado por el proveedor (Stripe Elements / PayPal SDK)
- OmniCRM nunca ve ni almacena números de tarjeta completos

- Cumplimiento de PCI DSS manejado por el proveedor de pago
- La validación en tiempo real previene números de tarjeta inválidos

Paso 2: Enviar y Tokenizar

Cuando haga clic en **"Agregar Método de Pago"**:

1. **Validación del Lado del Cliente:**
   - El proveedor de pago valida el formato del número de tarjeta
   - Verifica que la fecha de vencimiento esté en el futuro
   - Verifica el formato del CVC
2. **Tokenización:**
   - Los detalles de la tarjeta se envían directamente al proveedor de pago (no a OmniCRM)
   - El proveedor crea un token seguro (por ejemplo, `pm_1A2B3C4D` para Stripe)
   - Token devuelto a OmniCRM
3. **Procesamiento en el Servidor:**
   - OmniCRM guarda el token en el registro del cliente con el identificador del proveedor
   - Almacena los últimos 4 dígitos, marca, vencimiento y nombre del proveedor para mostrar
   - Ningún número de tarjeta completo toca los servidores de OmniCRM

Paso 3: Confirmación

Aparece un mensaje de éxito:

> Su Visa terminada en 1234 ha sido agregada a su cuenta.

La nueva tarjeta aparece en la tabla de métodos de pago.

Selección Automática Predeterminada:

- Si esta es la primera tarjeta del cliente, el sistema la establece automáticamente como predeterminada

- Si el cliente ya tiene tarjetas, la nueva tarjeta se agrega como no predeterminada
- El cliente puede cambiar la predeterminada después de agregar

Estableciendo el Método de Pago Predeterminado

El método de pago predeterminado se utiliza para:

- Cargos automáticos recurrentes por servicios
- Pagos de facturas
- Recargas y recargas
- Transacciones únicas (a menos que se especifique lo contrario)

Para Cambiar el Predeterminado:

1. Localice la tarjeta que desea establecer como predeterminada en la tabla de métodos de pago
2. Haga clic en el **menú de acciones (⋮)** junto a la tarjeta
3. Seleccione **"Establecer como Predeterminado"**
4. Aparece la confirmación

   La Visa terminada en 5678 ahora es su método de pago predeterminado.

La marca de verificación se mueve a la tarjeta reci��n seleccionada.

Indicador Visual:

Las tarjetas predeterminadas muestran:

en la columna Predeterminado, típicamente con una insignia de marca de verificación verde.

Eliminando un Método de Pago

Elimine tarjetas que estén vencidas, perdidas o que ya no se necesiten.

Paso 1: Iniciar Eliminación

1. Encuentre la tarjeta a eliminar en la tabla de métodos de pago
2. Haga clic en el ícono de **basura** en esa fila

Si el cliente tiene solo un método de pago activo, el ícono de basura está **deshabilitado**. Al pasar el cursor sobre él, se muestra un tooltip:

> No se puede eliminar su único método de pago. Agregue otro método de pago primero.

Agregue un segundo método de pago antes de poder eliminar el existente.

Paso 2: Confirmar Eliminación

Cuando se permite la eliminación, aparece un modal de confirmación:

> ¿Está seguro de que desea eliminar este método de pago?
>
> [Cancelar] [Eliminar Método de Pago]

Si el método que se está eliminando es el predeterminado actual y existen otros métodos activos, el modal indica que otro método activo será promovido a predeterminado después de la eliminación.

Haga clic en **"Eliminar Método de Pago"** para confirmar.

Paso 3: Eliminación Completa

Mensaje de éxito:

La tarjeta es eliminada de la tabla y borrada del proveedor de pago.

Restricciones Importantes:

- **No se puede eliminar el único método de pago activo** - El control de eliminación está deshabilitado (con tooltip). La API también rechaza la solicitud. Agregue otro método de pago primero.
- **Eliminar el predeterminado cuando existen otros** - Otro método de pago activo utilizable es promovido a predeterminado automáticamente.
- **Sin deshacer** - La eliminación es permanente; el cliente debe volver a agregar el método si es necesario.

Gestionando Tarjetas que Vencen

OmniCRM rastrea las fechas de vencimiento de las tarjetas y proporciona herramientas para actualizar proactivamente las tarjetas que están por vencer.

Tablero de Tarjetas que Vencen

Navegue a **Facturación → Tarjetas que Vencen** para ver una lista a nivel de sistema:

> Cliente Tarjeta Vencimiento Días Hasta la Acción John Smith Visa ****1234 02/2025 12 días Actualizar Acme Corp MC5678 03/2025 45 días Actualizar Jane Doe Amex**9012 01/2025 VENCIDA Actualizar

Filtros:

- **Rango de Vencimiento** - Próximos 30/60/90 días o ya vencidos
- **Tipo de Cliente** - Individual vs Empresa
- **Tipo de Servicio** - Filtrar por servicio que requiere método de pago

Acciones:

- **Actualizar** - Abre la página de métodos de pago del cliente para agregar una nueva tarjeta
- **Notificar** - Enviar recordatorio por correo electrónico al cliente (si Mailjet está configurado)

Notificaciones de Vencimiento

Si Mailjet está configurado, se envían correos electrónicos automáticos:

- **60 días antes del vencimiento** - Primer recordatorio
- **30 días antes del vencimiento** - Segundo recordatorio
- **7 días antes del vencimiento** - Advertencia final
- **Al vencimiento** - Notificación de que la tarjeta ha vencido

Los clientes pueden hacer clic en un enlace en el correo electrónico para actualizar su método de pago a través del portal de Autoservicio.

Variables de Plantilla de Correo Electrónico:

Las plantillas de Mailjet reciben:

- Nombre del cliente
- Marca de la tarjeta y últimos 4 dígitos
- Fecha de vencimiento
- Enlace a la página de métodos de pago del Autoservicio

Vea `integrations_mailjet` para la configuración de la plantilla de correo electrónico.

Actualizando una Tarjeta que Vence

Flujo de Trabajo Recomendado:

1. El cliente recibe un correo electrónico de notificación de vencimiento
2. El cliente inicia sesión en el portal de Autoservicio
3. Navega a **Facturación → Métodos de Pago**
4. Hace clic en **"Agregar Método de Pago"**
5. Ingresa los nuevos detalles de la tarjeta (la misma tarjeta con vencimiento actualizado, o tarjeta de reemplazo)
6. Establece la nueva tarjeta como predeterminada
7. Elimina la tarjeta antigua/vencida

Flujo de Trabajo del Personal:

Si el cliente llama al soporte:

1. El personal abre la cuenta del cliente
2. Navega a **Facturación → Métodos de Pago**
3. Agrega una nueva tarjeta en nombre del cliente (el cliente proporciona detalles por teléfono)
4. Establece la nueva tarjeta como predeterminada
5. Elimina la tarjeta vencida
6. Confirma con el cliente

> ⚠ **AVISO**
>
> Nunca pida a los clientes que envíen detalles de la tarjeta por correo electrónico o mensaje de texto. Siempre use:
>
> - Portal seguro de Autoservicio para autoservicio
> - Teléfono con el personal ingresando detalles directamente en el sistema
> - En persona en la ubicación minorista

Qué Ocurre Cuando las Tarjetas Vencen

Cuando una tarjeta de pago alcanza su fecha de vencimiento y no se actualiza:

Efectos Inmediatos:

1. **Los Pagos Automáticos Fallan**
   - El proveedor de pago rechaza transacciones con tarjetas vencidas
   - Las renovaciones de servicios mensuales no se procesan
   - Las recargas automáticas fallan
   - Los pagos automáticos de facturas fallan
2. **Notificaciones al Cliente**
   - El sistema intenta cargar la tarjeta
   - Se envía una notificación de fallo de pago
   - Se envía un correo electrónico "Actualizar Método de Pago" con enlace al portal de Autoservicio
3. **Cambios en el Estado del Servicio**
   - **Servicios Postpago** - Pueden continuar temporalmente con saldo pendiente
   - **Servicios Prepago** - Suspensión del servicio cuando se agota el saldo
   - **Servicios de Renovación Automática** - La renovación falla, el servicio puede expirar

Acciones Posteriores:

Día 1-3 (Período de Gracia):

- El servicio continúa normalmente
- El cliente recibe el primer aviso de fallo de pago
- El sistema intenta reintentar (dependiendo de la configuración)

Día 4-7:

- Segundo intento de pago (si está configurado)
- Correo electrónico de advertencia enviado
- El servicio al cliente puede contactar al cliente

Día 8-14:

- El servicio puede ser suspendido por falta de pago
- El estado suspendido previene el uso pero preserva la cuenta
- El cliente puede restaurar actualizando el método de pago y pagando el saldo pendiente

Día 15+:

- El servicio puede ser terminado por falta de pago
- Inventario (tarjetas SIM, equipos) marcado para devolución
- Se envía un aviso final
- La cuenta se remite a cobranzas (si aplica)

Previniendo la Interrupción del Servicio:

Para evitar la interrupción del servicio:

- Actualice las tarjetas **30 días antes del vencimiento**
- Agregue múltiples métodos de pago para redundancia
- Habilite alertas de fallo de pago
- Monitoree el tablero de Tarjetas que Vencen semanalmente

Restaurando el Servicio Después del Vencimiento:

Si el servicio se suspendió debido a una tarjeta vencida:

1. Agregue un nuevo método de pago válido
2. Establezca como predeterminado
3. Pague el saldo pendiente (si lo hay)
4. Contacte al soporte para reactivar el servicio
5. Servicio restaurado en minutos a horas

Seguridad del Método de Pago

Tokenización

OmniCRM utiliza la tokenización del proveedor para garantizar la seguridad:

1. **El cliente ingresa la tarjeta** → Enviada directamente a los servidores del proveedor de pago
2. **El proveedor valida y tokeniza** → Crea un token único
3. **Token almacenado en OmniCRM** → El número completo de la tarjeta nunca se almacena
4. **Procesamiento de pagos** → Token enviado al proveedor, el proveedor carga la tarjeta

Lo que OmniCRM Almacena:

```
{
  "vendor": "stripe",
  "vendor_payment_method_id": "pm_1A2B3C4D5E6F",
  "payment_type": "card",
  "brand": "visa",
  "last4": "1234",
  "exp_month": 12,
  "exp_year": 2026,
  "name": "John Smith",
  "nickname": "Personal Card",
  "is_default": true
}
```

Lo que OmniCRM NO Almacena:

- Número completo de la tarjeta
- Código CVV/CVC
- Datos de la banda magnética
- Números PIN

Cumplimiento de PCI

Al utilizar formularios de pago alojados por el proveedor:

- **Alcance PCI reducido** - Los datos de la tarjeta nunca tocan los servidores de OmniCRM
- **Campos alojados por el proveedor** - La entrada de la tarjeta ocurre en el iframe seguro del proveedor
- **Sin almacenamiento de tarjetas** - Se utilizan tokens en lugar de datos de tarjeta en bruto
- **Transmisión segura** - Toda la comunicación a través de HTTPS/TLS

Vea Guía del Sistema de Pagos para detalles de seguridad del proveedor de pagos.

Flujos de Trabajo Comunes

Flujo de Trabajo 1: El Cliente Agrega el Primer Método de Pago

Escenario: Nuevo cliente registrándose para un servicio

1. El cliente crea una cuenta
2. Selecciona un plan de servicio
3. Se le solicita agregar un método de pago durante el proceso de pago
4. Ingresa los detalles de la tarjeta en el modal de Stripe
5. La tarjeta es tokenizada y guardada
6. Se establece automáticamente como predeterminada
7. Se provisiona el servicio
8. Se procesa el primer cargo

Flujo de Trabajo 2: El Cliente Actualiza una Tarjeta que Vence

Escenario: Tarjeta de crédito a punto de vencer

1. El cliente recibe una notificación por correo electrónico (60 días antes del vencimiento)
2. Inicia sesión en el portal de Autoservicio
3. Navega a **Facturación → Métodos de Pago**
4. Revisa la tarjeta actual que vence en 12/2025
5. Hace clic en **"Agregar Método de Pago"**
6. Ingresa la tarjeta de reemplazo con vencimiento 12/2028
7. Establece la nueva tarjeta como predeterminada
8. Elimina la tarjeta antigua
9. Se envía un correo electrónico de confirmación

Flujo de Trabajo 3: El Personal Ayuda al Cliente por Teléfono

Escenario: El cliente llama: "Mi tarjeta fue rechazada"

1. El cliente llama al soporte
2. El personal verifica la identidad (preguntas de seguridad)
3. El personal revisa los métodos de pago: Tarjeta vencida 01/2025
4. El personal: "Su tarjeta ha vencido. ¿Tiene una nueva tarjeta?"
5. El cliente proporciona los nuevos detalles de la tarjeta por teléfono
6. El personal navega a **Clientes → [Cliente] → Facturación**
7. Hace clic en **"Agregar Método de Pago"**
8. Ingresa los detalles de la tarjeta mientras el cliente los lee
9. Establece la nueva tarjeta como predeterminada
10. Elimina la tarjeta vencida
11. Reintenta el pago fallido
12. Confirma con el cliente: "Pago exitoso, servicio restaurado"

Flujo de Trabajo 4: Cliente Empresarial con Múltiples Tarjetas

Escenario: La empresa quiere diferentes tarjetas para diferentes propósitos

1. El cliente empresarial agrega la tarjeta principal (Visa terminada en 1111)
2. Se establece como predeterminada para cargos mensuales por servicios
3. Agrega tarjeta de respaldo (Mastercard terminada en 2222) para recargas
4. Agrega tarjeta de compras (Amex terminada en 3333) para compras de equipos
5. Al hacer una recarga, selecciona Mastercard manualmente al pagar
6. La Visa predeterminada sigue utilizándose para la facturación automática mensual

Flujo de Trabajo 5: Gestión de Tarjetas que Vencen (Admin)

Escenario: Gestión proactiva de vencimientos

1. El administrador navega a **Facturación → Tarjetas que Vencen**
2. Filtros: "Próximos 30 días"
3. Ve 15 clientes con tarjetas que están por vencer
4. Selecciona todos → **"Enviar Correos Electrónicos de Recordatorio"**
5. Mailjet envía correos electrónicos personalizados a cada cliente
6. Los clientes actualizan tarjetas a través del Autoservicio
7. El administrador revisa la lista una semana después
8. Llama a los clientes restantes que no han actualizado
9. Asiste con actualizaciones de tarjetas por teléfono

Solución de Problemas

"Tarjeta rechazada" al agregar método de pago

- **Causa:** El proveedor de pago rechazó la tarjeta (fondos insuficientes, prevención de fraude, rechazo del emisor)
- **Solución:**
  - Intente con una tarjeta diferente
  - Contacte al emisor de la tarjeta para autorizar la transacción
  - Asegúrese de que la tarjeta soporte compras en línea

    - Verifique que la dirección de facturación coincida con la tarjeta en archivo

"Error al agregar método de pago" (error genérico)

- **Causa:** Error de API del proveedor de pago o problema de red
- **Solución:**
    - Actualice la página e intente nuevamente
    - Verifique la conexión a Internet
    - Verifique que la configuración del proveedor de pago sea correcta en la configuración del sistema
    - Verifique la consola del navegador para obtener un mensaje de error específico
    - Intente con un navegador diferente (desactive los bloqueadores de anuncios)

No se puede eliminar el método de pago (botón deshabilitado)

- **Causa:** Es el único método de pago activo del cliente
- **Solución:**
    - Agregue un nuevo método de pago primero, luego elimine el antiguo
    - Pase el cursor sobre el ícono de basura deshabilitado para la explicación del tooltip

La tarjeta aparece como vencida pero no en la lista de "Tarjetas que Vencen"

- **Causa:** La tarjeta venció recientemente, la caché no se actualizó
- **Solución:**
    - Actualice la página
    - Verifique los filtros en el tablero de Tarjetas que Vencen
    - Las tarjetas vencidas pueden moverse a una vista diferente

La nueva tarjeta no aparece inmediatamente

- **Causa:** La página no se ha actualizado después de agregar la tarjeta
- **Solución:**
    - La tabla de métodos de pago debería actualizarse automáticamente

    - Si no, actualice manualmente el navegador
    - Verifique si ocurrió un error durante el proceso de adición

El modal de pago no se carga

- **Causa:** SDK del proveedor de pago no se carga, problema con la clave API o extensión del navegador bloqueando
- **Solución:**
    - Verifique la consola del navegador en busca de errores
    - Desactive los bloqueadores de anuncios y la protección de seguimiento
    - Verifique la configuración del proveedor de pago en la configuración del sistema
    - Asegúrese de que el script del SDK del proveedor se cargue (verifique la pestaña de Red)
    - Intente en modo incógnito/navegación privada

El cliente no recibe notificaciones de vencimiento

- **Causa:** Mailjet no configurado o plantilla de correo electrónico faltante
- **Solución:**
    - Verifique las credenciales de Mailjet en *crm_config.yaml*
    - Verifique que exista una plantilla de correo electrónico para el vencimiento de la tarjeta
    - Confirme que la dirección de correo electrónico del cliente sea válida
    - Verifique los registros de Mailjet para fallos de entrega

Mejores Prácticas

Para Clientes:

- Agregue un método de pago antes de la activación del servicio para evitar retrasos
- Mantenga al menos 2 tarjetas en archivo para redundancia
- Actualice las tarjetas que están por vencer 30+ días antes del vencimiento
- Elimine tarjetas antiguas/vencidas para evitar confusiones
- Use apodos descriptivos ("Visa Personal", "Amex de Trabajo")

- Verifique que el método de pago predeterminado sea correcto para la facturación automática

Para el Personal de Soporte:

- Verifique la identidad del cliente antes de acceder a los métodos de pago
- Nunca pida a los clientes que envíen detalles de la tarjeta por correo electrónico/SMS/chat
- Procese las adiciones de tarjetas inmediatamente durante las llamadas (no las posponga)
- Confirme que la nueva tarjeta esté establecida como predeterminada después de agregar
- Elimine tarjetas antiguas solo después de confirmar que la nueva tarjeta funciona
- Pruebe el pago después de actualizar la tarjeta vencida (procese una autorización de £0.01)

Para Administradores:

- Monitoree el tablero de Tarjetas que Vencen semanalmente
- Envíe correos electrónicos de recordatorio 60/30/7 días antes del vencimiento
- Mantenga las claves de prueba/producción del proveedor de pago separadas para desarrollo vs producción
- Asegúrese de que las plantillas de Mailjet estén configuradas para notificaciones de vencimiento
- Revise los informes de fallos de pago para identificar tarjetas vencidas
- Capacite al personal sobre procedimientos seguros de manejo de tarjetas

Documentación Relacionada

- Guía del Sistema de Pagos - Referencia completa de la API de pagos y arquitectura
- `payments_process` - Procesamiento de pagos con métodos de pago almacenados

- `payments_invoices` - Pago automático de facturas utilizando la tarjeta predeterminada
- `features_topup_recharge` - Sistema de recarga utilizando métodos de pago
- `basics_payment` - Conceptos generales de pago y facturación
- `customer_care` - Portal de Autoservicio para que los clientes gestionen sus propias tarjetas

Guía de la API del Sistema de Pagos de OmniCRM

Descripción general

El sistema de pagos de OmniCRM proporciona una infraestructura de procesamiento de pagos **agnóstica al proveedor** y completa. Hoy en día, admite Stripe y PayPal, pero la arquitectura modular permite la integración con cualquier proveedor de pagos (Square, Adyen, Braintree, etc.) sin cambiar el código de la aplicación.

Este documento cubre todas las API de pagos y flujos de trabajo disponibles en el sistema.

> **Integración de Ansible Playbook**: Para implementar estas API de pagos en playbooks de aprovisionamiento, consulte **Cargos y Pagos desde Playbooks**

Tabla de Contenidos

1. Arquitectura Modular
2. Conceptos Básicos
3. Documentos Financieros
4. API de Métodos de Pago
5. API de Flujos de Pago
6. API de Billetera
7. Resumen de Referencia de API
8. Casos de Uso Comunes

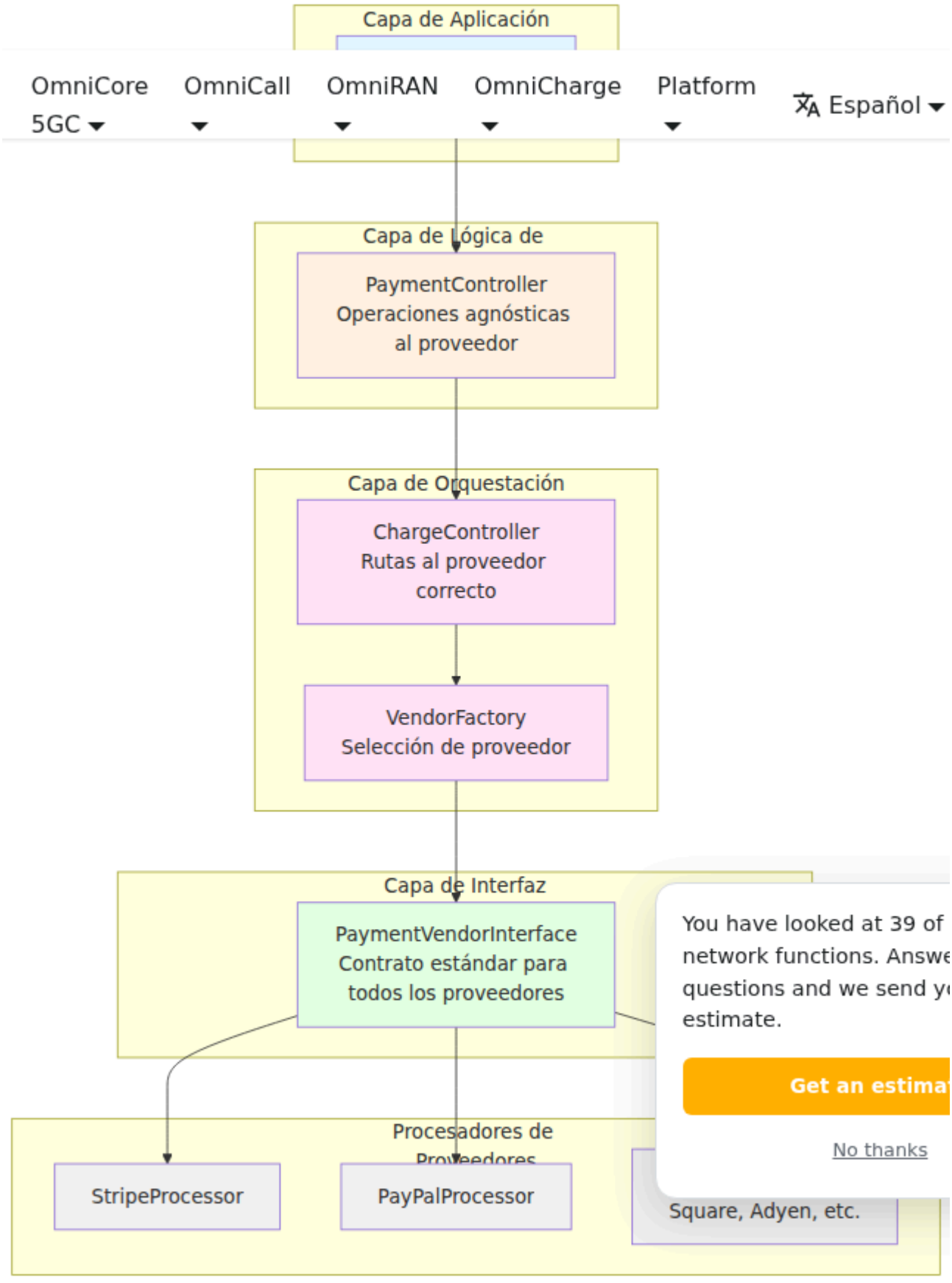
Arquitectura Modular

¿Por qué Agnóstico al Proveedor?

El sistema utiliza **capas de abstracción** para separar la lógica de negocio de las especificaciones del proveedor de pagos. Esto significa:

- □ Agregar nuevos proveedores de pagos sin tocar el código de la aplicación
- □ Cambiar de proveedores sin migraciones de base de datos
- □ Soportar múltiples proveedores simultáneamente
- □ API consistente independientemente del proveedor de backend

Capas de Arquitectura



Agregar un Nuevo Proveedor de Pagos

Para agregar un nuevo proveedor (por ejemplo, Square, Adyen, Braintree), comuníquese con su equipo técnico de OmniCRM.

Para obtener detalles de configuración sobre proveedores existentes (Stripe, PayPal), consulte **Configuración del Proveedor**.

Descripción del Proceso:



Pasos:

1. **Implementar clase de procesador** con `PaymentVendorInterface` estándar (autorizar, capturar, cobrar, reembolsar, liberar)
2. **Registrar procesador** en VendorFactory con el nombre del proveedor
3. **Probar integración** con el entorno sandbox del proveedor
4. **Desplegar** - Todas las API existentes soportan automáticamente el nuevo proveedor

Resultado: Una vez desplegado, el nuevo proveedor funciona sin problemas:

```
# Agregar método de pago de Square
curl -X POST /api/payments/methods \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "vendor": "square",
    "payment_token": "sq_xxxxx"
  }'
```

Todas las API de pagos soportan autom��ticamente el nuevo proveedor sin necesidad de cambios en el código de la aplicación.

Tiempo de Integración: Típicamente 1-2 días para un nuevo procesador de proveedor.

Configuración del Proveedor

Habilitación de Proveedores de Pago

La interfaz de usuario controla qué proveedores de pago están disponibles para los clientes. Esto se establece a través de la variable `VITE_PAYMENT_VENDORS` en el archivo `.env` de la interfaz de usuario:

```
# Habilitar solo Stripe (predeterminado)
VITE_PAYMENT_VENDORS="stripe"

# Habilitar tanto Stripe como PayPal
VITE_PAYMENT_VENDORS="stripe,paypal"

# Habilitar solo PayPal
VITE_PAYMENT_VENDORS="paypal"
```

Importante: `VITE_PAYMENT_VENDORS` es una variable **de tiempo de construcción**. Cambiarla requiere reconstruir y reiniciar la interfaz de usuario.

Configuración de Stripe

Stripe es el proveedor de pagos principal, proporcionando procesamiento de tarjetas con soporte para 3D Secure.

1. Obtener Claves API de Stripe

- Regístrese en https://stripe.com
- Navegue a Desarrolladores → Claves API
- Copie su **Clave Publicable** (pk_live_... o pk_test_...)
- Copie su **Clave Secreta** (sk_live_... o sk_test_...)

Importante:

- Use **claves de prueba** (pk_test_/sk_test_) para desarrollo
- Use **claves en vivo** (pk_live_/sk_live_) solo para producción
- **Nunca** comprometa las claves API en el control de versiones

2. Configurar Backend

Agregue a `crm_config.yaml` (nivel raíz, no anidado bajo `payment_vendors`):

```yaml
stripe:
  secret_key: 'sk_live_YOUR_SECRET_KEY_HERE'
  publishable_key: 'pk_live_YOUR_PUBLISHABLE_KEY_HERE'
  currency: 'aud'
  statement_descriptor_suffix: 'YOUR_COMPANY'
```

La API lee las credenciales de Stripe desde `crm_config.yaml` al inicio. Cambiar estos valores requiere reiniciar la API.

3. Configurar Frontend

Agregue al archivo `.env` de la interfaz de usuario:

```
VITE_STRIPE_PUBLISHABLE_KEY=pk_live_YOUR_PUBLISHABLE_KEY_HERE
```

Nota de Seguridad: Solo la clave publicable va en el frontend. La clave secreta **nunca** debe ser expuesta al navegador.

Configuración de PayPal

PayPal proporciona almacenamiento de tarjetas y pagos de cuentas de PayPal.

1. Obtener Credenciales API de PayPal

- Cree una aplicación REST API en https://developer.paypal.com/dashboard/
- Copie su **ID de Cliente** y **Secreto de Cliente**
- Para pruebas, use credenciales de **Sandbox**; para producción, use credenciales **en vivo**

2. Configurar Backend

Agregue a `crm_config.yaml` (nivel raíz):

```
paypal:
  client_id: 'YOUR_PAYPAL_CLIENT_ID'
  client_secret: 'YOUR_PAYPAL_CLIENT_SECRET'
  mode: 'sandbox'  # 'sandbox' para pruebas, 'live' para
producción
```

La API lee las credenciales de PayPal desde `crm_config.yaml` al inicio. Cambiar estos valores requiere reiniciar la API.

3. Configurar Frontend

Agregue al archivo `.env` de la interfaz de usuario:

```
VITE_PAYMENT_VENDORS="stripe,paypal"
VITE_PAYPAL_CLIENT_ID=YOUR_PAYPAL_CLIENT_ID
```

`VITE_PAYPAL_CLIENT_ID` debe coincidir con el valor de `paypal.client_id` en `crm_config.yaml`.

Sandbox vs En Vivo

	Sandbox	En Vivo
API de Stripe	`sk_test_` / `pk_test_`	`sk_live_` / `pk_live_`
API de PayPal	`api-m.sandbox.paypal.com`	`api-m.paypal.com`
PayPal `mode`	`sandbox`	`live`
Dinero	Falso	Real

Al cambiar de sandbox a en vivo, actualice **ambos** `crm_config.yaml` (credenciales y modo) y el archivo `.env` de la interfaz de usuario (clave publicable / ID de cliente), luego reconstruya la interfaz de usuario y reinicie la API.

Cumplimiento de PCI

Cómo OmniCRM Mantiene el Cumplimiento de PCI:

- Los datos de la tarjeta se ingresan directamente en iframes alojados por el proveedor (Stripe Elements, Campos de Tarjeta de PayPal)
- OmniCRM **nunca ve ni almacena** números de tarjeta completos
- Solo se almacenan métodos de pago tokenizados en la base de datos
- Alcance reducido de cumplimiento de PCI para su negocio

Métricas de Procesamiento de Pagos

OmniCRM proporciona métricas completas para monitorear las operaciones de procesamiento de pagos. Para obtener detalles completos sobre las métricas de pagos de Stripe, incluidas las métricas de seguimiento de llamadas API, volúmenes de pago, tasas de fallos y tiempos de respuesta, consulte Monitoreo y Métricas.

El Esquema de Base de Datos es Agnóstico al Proveedor

El esquema de la base de datos soporta cualquier proveedor de pagos sin migraciones:



Ejemplo - Guardando una tarjeta de Square:

```
{
  "vendor": "square",
  "vendor_payment_method_id": "sq_card_abc123",
  "payment_type": "card"
}
```

Sin cambios en el código. Sin migraciones. Simplemente funciona.

Conceptos Básicos

Modelos de Datos

PaymentMethod

Almacenamiento agnóstico al proveedor de los métodos de pago del cliente.

```
{
  "payment_method_id": 789,
  "customer_id": 123,
  "vendor": "stripe",                    // 'stripe', 'paypal', o
cualquier proveedor agregado
  "vendor_payment_method_id": "pm_xxx", // ID interno del
proveedor
  "payment_type": "card",                // 'card', 'paypal',
'ach', etc.
  "is_default": true,
  "card_brand": "visa",
  "card_last4": "4242",
  "card_exp_month": 12,
  "card_exp_year": 2025,
  "card_nickname": "My Visa Card",
  "status": "active"
}
```

PaymentAuthorization

Registros de autorización de compromiso en dos fases (retener fondos).

```
{
  "authorization_id": 301,
  "customer_id": 123,
  "payment_method_id": 789,
  "vendor": "stripe",                    // Qué proveedor autorizó
  "vendor_authorization_id": "auth_xxx", // ID de autorización del
proveedor
  "amount": 200.00,
  "currency": "USD",
  "status": "authorized",                // 'authorized',
'captured', 'released'
  "authorized_at": "2025-12-27T10:00:00Z",
  "expires_at": "2026-01-03T10:00:00Z",
  "meta": {}
}
```

PaymentCapture

Pagos capturados/completados.

```
{
  "capture_id": 103,
  "authorization_id": 301,
  "customer_id": 123,
  "payment_method_id": 789,
  "vendor": "stripe",
  "vendor_transaction_id": "ch_xxx",     // ID de transacción del
proveedor
  "amount": 200.00,
  "currency": "USD",
  "status": "succeeded",                 // 'succeeded', 'failed',
'refunded'
  "captured_at": "2025-12-27T10:30:00Z",
  "vendor_response": {},                 // Respuesta completa del
proveedor
  "meta": {}
}
```

WalletAccount

Billetera del cliente con seguimiento de saldo (1-a-1 con el cliente).

```
{
  "wallet_account_id": 456,
  "customer_id": 123,
  "balance": 150.50,
  "currency": "USD",
  "auto_recharge_enabled": true,
  "auto_recharge_amount": 100.00,
  "auto_recharge_threshold": 10.00,
  "low_balance_warning_threshold": 10.00
}
```

WalletLedger

Registro completo de todas las transacciones de la billetera.

```
{
  "ledger_id": 501,
  "customer_id": 123,
  "wallet_account_id": 456,
  "transaction_type": "credit",          // 'credit', 'debit',
'refund', 'adjustment'
  "amount": 100.00,
  "balance_before": 150.50,
  "balance_after": 250.50,
  "currency": "USD",
  "description": "Recarga de tarjeta",
  "reference_type": "payment_capture",   // Enlace al objeto
relacionado
  "reference_id": 103,
  "meta": {},
  "created_at": "2025-12-27T10:35:00Z"
}
```

Documentos Financieros

Para la creación y personalización de facturas, consulte **Facturas de Clientes**.

Facturas

Definición: Una factura es un documento que contiene una lista de **transacciones de CARGO** (cargos). Cuando se llama a la API para crear una factura, se debe proporcionar un array de IDs de transacciones de débito. Esas transacciones están "vinculadas" a la factura configurando su campo `invoice_id`.

Para detalles sobre la gestión de transacciones, consulte **Transacciones**.

Campos Clave:

```
{
  "invoice_id": 12345,
  "invoice_number": "INV-2025-000001",   // Generado
automáticamente: INV-YYYY-NNNNNN
  "customer_id": 123,
  "title": "Factura de Servicios Mensuales",
  "paid": true,                          // Estado de pago
  "void": false,                         // Estado de anulación
  "payment_reference": "ch_xxxxx",       // Último/ID de pago
principal
  "payment_type": "stripe_capture",      // Último tipo de pago
  "payment_time": "2025-12-27T10:30:00",
  "start_date": "2025-12-01",
  "end_date": "2025-12-31",
  "due_date": "2026-01-15",
  "retail_cost": 500.00,                 // Monto total de la
factura
  "wholesale_cost": 250.00
}
```

Generación de Facturas: Los números de factura se generan automáticamente en el formato `INV-YYYY-NNNNNN`, secuencial dentro del año calendario, reiniciándose cada 1 de enero.

Pagos en Facturas

Cómo Funcionan los Pagos: Una vez que se crea una factura a partir de transacciones de débito, se puede aplicar un **PAGO** creando una **transacción**

de crédito (costo minorista negativo) vinculada a la factura.

Los pagos deben:

- Estar listados en la factura claramente marcados como "PAGOS"
- Mostrar la fecha de pago relevante (que puede diferir de la fecha de creación de la factura)
- Soportar múltiples pagos por factura
- Netear contra los débitos para calcular el saldo de la factura

Estado de la Factura:

- **PAGADA**: Los pagos totales (créditos) son iguales o superan los débitos totales
- **PARCIALMENTE PAGADA**: Algunos pagos aplicados pero queda un saldo
- **SOBREPAGADA**: Actualmente no manejado - requiere división de crédito (función futura)

Contabilidad de Doble Entrada: El sistema implementa una contabilidad adecuada donde cada cargo tiene un pago compensatorio:

```
// 1. Transacción de débito (cargo)
{
  "transaction_id": 7001,
  "invoice_id": 12345,
  "retail_cost": 100.00,                // Positivo = el cliente
debe
  "title": "Cargo por Servicio"
}

// 2. Transacción de crédito (pago)
{
  "transaction_id": 7002,
  "invoice_id": 12345,
  "retail_cost": -100.00,               // Negativo = pago
recibido
  "title": "Pago por Factura: Cargo por Servicio (12345)",
  "payment_type": "stripe_capture",
  "payment_reference": "ch_xxxxx"
}

// Resultado neto: saldo de $0 → factura marcada como PAGADA
```

Metadatos Suplementarios de la Factura: Además de las transacciones de crédito, la factura también almacena campos de resumen (`payment_reference`, `payment_type`, `payment_time`) para búsquedas rápidas. Sin embargo:

- **Método primario**: Transacciones de crédito vinculadas a través de `invoice_id` (especificación de Hayden)
- **Metadatos secundarios**: Campos de factura almacenan un resumen del último/pago principal
- **Para múltiples pagos**: Consultar transacciones de crédito para el historial completo

Estados de Cuenta

Definición: Un estado de cuenta muestra todos los **DÉBITOS y CRÉDITOS** de las transacciones de un cliente durante un período específico. Este es el único tipo de documento donde tanto los débitos como los créditos se muestran como elementos de línea juntos, como un estado de cuenta bancario.

Notas de Crédito

Propósito: Solo se aplican a facturas con pagos ya aplicados. Debe hacerse como parte del proceso de anulación de la factura.

Proceso:

1. Se anula una factura (junto con sus transacciones de DÉBITO asociadas)
2. Cualquier PAGO ya aplicado se vincula a la NOTA DE CRÉDITO creada
3. El saldo del cliente pasa a crédito por el monto equivalente
4. El saldo de la nota de crédito puede luego ser:
    - Aplicado a otra factura como un PAGO, O
    - REEMBOLSADO al cliente

Lógica de Reembolso: Si se elige el reembolso, se invoca en función de las transacciones de CRÉDITO asociadas y cómo se pagaron originalmente (por ejemplo, reembolso de Stripe si el tipo de transacción de crédito fue Stripe).

API de Métodos de Pago

Todos los puntos finales utilizan la URL base: `/api/payments/`

Para la gestión detallada de métodos de pago y manejo de tarjetas, consulte **Métodos de Pago**.

Agregar Método de Pago

Guarde un nuevo método de pago para un cliente.

Endpoint: `POST /api/payments/methods`

Solicitud:

```
curl -X POST https://your-domain.com/api/payments/methods \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "vendor": "stripe",
    "payment_token": "pm_xxxxx",
    "is_default": false,
    "card_nickname": "My Visa Card"
  }'
```

Cuerpo de la Solicitud:

```
{
  "customer_id": 123,
  "vendor": "stripe",                    // 'stripe', 'paypal', o
cualquier proveedor agregado
  "payment_token": "pm_xxxxx",           // Token único del SDK
del frontend
  "is_default": false,                   // ¿Establecer como
método de pago predeterminado?
  "card_nickname": "My Visa Card"        // Nombre amigable
opcional
}
```

Respuesta (201 Creado):

```
{
  "success": true,
  "message": "Método de pago agregado con éxito",
  "data": {
    "payment_method_id": 789,
    "customer_id": 123,
    "vendor": "stripe",
    "payment_type": "card",
    "card_brand": "visa",
    "card_last4": "4242",
    "card_exp_month": 12,
    "card_exp_year": 2025,
    "card_nickname": "My Visa Card",
    "is_default": false,
    "status": "active"
  }
}
```


Obtener Métodos de Pago

Recuperar todos los métodos de pago para un cliente.

Endpoint: `GET /api/payments/methods?customer_id={id}`

Solicitud:

```
curl -X GET "https://your-domain.com/api/payments/methods?
customer_id=123" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

Respuesta (200 OK):

```
{
  "success": true,
  "data": [
    {
      "payment_method_id": 789,
      "customer_id": 123,
      "vendor": "stripe",
      "payment_type": "card",
      "card_brand": "visa",
      "card_last4": "4242",
      "is_default": true
    },
    {
      "payment_method_id": 790,
      "customer_id": 123,
      "vendor": "paypal",
      "payment_type": "paypal",
      "paypal_email": "user@example.com",
      "is_default": false
    }
  ]
}
```


Obtener Método de Pago Predeterminado

Obtener el método de pago predeterminado para un cliente.

Endpoint: `GET /api/payments/methods/default?customer_id={id}`

Solicitud:

```
curl -X GET "https://your-domain.com/api/payments/methods/default?
customer_id=123" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

Respuesta (200 OK):

```
{
  "success": true,
  "data": {
    "payment_method_id": 789,
    "vendor": "stripe",
    "payment_type": "card",
    "card_brand": "visa",
    "card_last4": "4242",
    "is_default": true
  }
}
```

Establecer Método de Pago Predeterminado

Establecer un método de pago como predeterminado.

Endpoint: `PUT /api/payments/methods/set-default`

Solicitud:

```
curl -X PUT https://your-domain.com/api/payments/methods/set-
default \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "payment_method_id": 790
  }'
```

Respuesta (200 OK):

```
{
  "success": true,
  "message": "Método de pago predeterminado actualizado",
  "data": {
    "payment_method_id": 790,
    "is_default": true
  }
}
```

Eliminar Método de Pago

Eliminar un método de pago guardado.

Endpoint: `DELETE /api/payments/methods/{payment_method_id}?customer_id={id}`

Solicitud:

```
curl -X DELETE "https://your-domain.com/api/payments/methods/789?
customer_id=123" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

Respuesta (200 OK):

```
{
  "success": true,
  "message": "Método de pago eliminado con éxito"
}
```

API de Flujos de Pago

El sistema admite varios flujos de pago dependiendo de su caso de uso.

1. Pago Directo (Cargo Simple)

Caso de Uso: Pago simple de un paso sin aprovisionamiento de servicio.

Endpoint: `POST /api/payments/charge`

Solicitud:

```
curl -X POST https://your-domain.com/api/payments/charge \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "amount": 50.00,
    "currency": "USD",
    "payment_method_id": 789,
    "metadata": {
      "order_id": "12345"
    }
  }'
```

Cuerpo de la Solicitud:

```
{
  "customer_id": 123,
  "amount": 50.00,
  "currency": "USD",                    // Predeterminado: USD
  "payment_method_id": 789,             // Opcional - usa el
predeterminado si se omite
  "vendor": "stripe",                   // Requerido si se usa
payment_token
  "payment_token": "pm_xxxxx",          // Opcional - token único
  "save_method": false,                 // ¿Guardar método de
pago para uso futuro?
  "metadata": {
    "order_id": "12345"
  }
}
```

Respuesta (200 OK):

```
{
  "success": true,
  "message": "Pago exitoso",
  "data": {
    "transaction_id": "ch_xxxxx",
    "capture_id": 101,
    "amount": 50.00,
    "currency": "USD",
    "status": "succeeded"
  }
}
```

2. Pago de Factura (Enrutamiento Primero a la Billetera)

Caso de Uso: Pagar una factura usando el saldo de la billetera primero, luego la tarjeta para el resto.

Endpoint: `POST /api/payments/invoice`

Flujo:

1. Verificar saldo de la billetera
2. Usar fondos de la billetera primero
3. Cargar la tarjeta por el déficit (si lo hay)
4. Acreditar la billetera con el monto de la tarjeta
5. Debitar la billetera por el monto total de la factura

Solicitud:

```
curl -X POST https://your-domain.com/api/payments/invoice \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "amount": 200.00,
    "payment_method_id": 789,
    "metadata": {
      "invoice_id": 12345,
      "description": "Pago de factura"
    }
  }'
```

Respuesta (200 OK):

```
{
  "success": true,
  "message": "Pago de factura procesado",
  "data": {
    "customer_id": 123,
    "service_amount": 200.00,
    "routing_mode": "hybrid",
    "initial_balance": 150.00,
    "wallet_portion_used": 150.00,       // La billetera cubrió
esta cantidad
    "card_portion_used": 50.00,          // Se cargó la tarjeta
por el resto
    "charged_amount": 50.00,
    "wallet_credited": 50.00,
    "wallet_debited": 200.00,
    "final_balance": 0.00,
    "payment_method_used": true
  }
}
```

3. Retención de Autorización (Reservar Fondos)

Caso de Uso: Retener fondos para captura posterior (por ejemplo, reservas de hotel, alquileres).

> **Implementación de Playbook**: Para ejemplos de playbook de Ansible de flujos de pago en dos fases, consulte Patrón de Compromiso en Dos Fases

Paso 1: Crear Retención de Autorización

Endpoint: `POST /api/payments/authorize/hold`

Flujo Primero a la Billetera:

1. Verificar saldo de la billetera
2. Calcular déficit (monto - wallet_balance)
3. Autorizar tarjeta solo por el déficit
4. El débito de la billetera ocurre en el momento de la captura, no aquí

Solicitud:

```
curl -X POST https://your-domain.com/api/payments/authorize/hold \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "amount": 200.00,
    "payment_method_id": 789,
    "use_wallet": true,
    "metadata": {
      "reservation_id": "RES-001"
    }
  }'
```

Cuerpo de la Solicitud:

```
{
  "customer_id": 123,
  "amount": 200.00,
  "currency": "USD",
  "payment_method_id": 789,              // Opcional - usa el
predeterminado si se omite
  "vendor": "stripe",                    // Requerido si se usa
payment_token
  "payment_token": "pm_xxxxx",           // Opcional - token único
  "save_method": false,
  "use_wallet": true,                    // Habilitar enrutamiento
primero a la billetera (predeterminado: true)
  "metadata": {
    "reservation_id": "RES-001"
  }
}
```

Respuesta (200 OK):

```
{
  "success": true,
  "message": "Pago autorizado (retención creada)",
  "data": {
    "authorization_id": 301,
    "vendor_authorization_id": "auth_xxxxx",
    "amount": 200.00,
    "currency": "USD",
    "status": "authorized",
    "wallet_balance": 150.00,
    "wallet_to_use": 150.00,             // La billetera cubrirá
esta cantidad
    "card_amount": 50.00,                // Tarjeta autorizada por
esta cantidad
    "message": "Tarjeta autorizada por $50 (recarga de billetera).
El débito de la billetera de $200 ocurrirá en la captura."
  }
}
```

Paso 2: Capturar Autorización

Endpoint: `POST /api/payments/capture/{authorization_id}`

Flujo de Captura Primero a la Billetera:

1. Capturar tarjeta (si está autorizada) - recarga la billetera
2. Acreditar billetera con el monto capturado de la tarjeta
3. Debitar billetera por el monto total del servicio
4. Crear facturas/transacciones (si se solicita)

Solicitud:

```
curl -X POST https://your-domain.com/api/payments/capture/301 \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "amount": 200.00,
    "metadata": {
      "invoice": true,
      "title": "Reserva de Hotel",
      "description": "Estancia de 3 noches",
      "wholesale_cost": 100.00,
      "contract_days": 3,
      "send_email": true
    }
  }'
```

Cuerpo de la Solicitud:

```
{
  "amount": 200.00,                    // Opcional - captura el
monto total si se omite
  "metadata": {
    "invoice": true,                   // ¿Crear factura y
transacciones?
    "create_transaction": true,        // ¿Crear registro de
transacción?
    "title": "Reserva de Hotel",
    "description": "Estancia de 3 noches",
    "wholesale_cost": 100.00,
    "contract_days": 3,
    "send_email": true                 // ¿Enviar correo
electrónico de factura?
  }
}
```

Respuesta (200 OK):

```
{
  "success": true,
  "message": "Autorización capturada",
  "data": {
    "capture_id": 103,
    "transaction_id": "ch_xxxxx",
    "authorization_id": 301,
    "amount": 200.00,
    "currency": "USD",
    "status": "succeeded",
    "wallet_credit": {                    // Tarjeta recargó la
billetera
      "ledger_id": 401,
      "amount": 50.00
    },
    "wallet_debit": {                     // Servicio cargado desde
la billetera
      "ledger_id": 402,
      "amount": 200.00
    },
    "transaction": {                      // Creado si invoice=true
      "transaction_id": 7001
    },
    "invoice": {                          // Creado si invoice=true
      "invoice_id": 12345,
      "invoice_number": "INV-2025-000001"
    }
  }
}
```

Paso 3: Liberar Autorización (Cancelar)

Endpoint: `POST /api/payments/release/{authorization_id}`

Caso de Uso: Cancelar reserva, fallo en el aprovisionamiento o el cliente cambió de opinión.

Solicitud:

```
curl -X POST https://your-domain.com/api/payments/release/301 \
  -H "Authorization: Bearer YOUR_API_KEY"
```

Respuesta (200 OK):

```
{
  "success": true,
  "message": "Autorización liberada",
  "data": {
    "authorization_id": 301,
    "vendor_authorization_id": "auth_xxxxx",
    "status": "released",
    "released_at": "2025-12-27T10:45:00Z"
  }
}
```

Nota: Con el flujo primero a la billetera, no se necesita reembolso de la billetera ya que esta no se debita hasta el momento de la captura.

4. Pago de Recarga (Dos Fases con Aprovisionamiento)

Caso de Uso: Procesar pago para recarga de servicio que requiere aprovisionamiento (por ejemplo, activación de hotspot/dongle). Si el aprovisionamiento falla, se libera la autorización.

Endpoint: `POST /api/payments/topup`

Flujo: Autorizar → Aprovisionar servicio → Capturar

Solicitud:

```
curl -X POST https://your-domain.com/api/payments/topup \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "amount": 30.00,
    "payment_method_id": 789,
    "service_uuid": "svc-uuid-123",
    "imsi": "123456789012345",
    "days": 30,
    "metadata": {
      "is_rental": false
    }
  }'
```

Para Alquiler/Hotspot Anónimo (método de pago NO guardado):

```
curl -X POST https://your-domain.com/api/payments/topup \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 1,
    "amount": 5.00,
    "vendor": "stripe",
    "payment_token": "pm_xxxxx",
    "service_uuid": "hotspot-uuid",
    "imsi": "123456789012345",
    "days": 1,
    "metadata": {
      "is_rental": true,
      "billing_email": "user@example.com"
    }
  }'
```

Respuesta (200 OK):

```
{
  "success": true,
  "message": "Pago de recarga procesado con éxito",
  "data": {
    "transaction_id": "ch_xxxxx",
    "authorization_id": 302,
    "capture_id": 104,
    "amount": 30.00,
    "status": "succeeded",
    "provision_result": {
      "success": true,
      "topup_result": {...},
      "service_uuid": "svc-uuid-123",
      "imsi": "123456789012345",
      "days": 30
    },
    "payment_method_saved": false        // Falso para
anónimo/alquiler
  }
}
```

5. Pago de Alquiler (Tercero)

Caso de Uso: Cargar la tarjeta de un cliente para pagar el servicio de otro cliente.

Endpoint: `POST /api/payments/rental`

Solicitud:

```
curl -X POST https://your-domain.com/api/payments/rental \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "beneficiary_customer_id": 456,
    "charge_customer_id": 123,
    "amount": 75.00,
    "payment_method_id": 789,
    "service_description": "Pago por servicio de alquiler",
    "metadata": {
      "rental_agreement_id": "RA-001"
    }
  }'
```

Para Alquiler Anónimo (tarjeta del arrendatario NO guardada):

```
curl -X POST https://your-domain.com/api/payments/rental \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "beneficiary_customer_id": 456,
    "amount": 75.00,
    "vendor": "stripe",
    "payment_token": "pm_xxxxx",
    "service_description": "Pago de alquiler anónimo",
    "metadata": {
      "billing_email": "renter@example.com"
    }
  }'
```

Respuesta (200 OK):

```
{
  "success": true,
  "message": "Pago de alquiler procesado",
  "data": {
    "transaction_id": "ch_xxxxx",
    "amount": 75.00,
    "payment_method_saved": false        // Anónimo: no método guardado
  }
}
```

6. Reembolsar Pago

Endpoint: `POST /api/payments/refund`

Solicitud:

```
curl -X POST https://your-domain.com/api/payments/refund \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "transaction_id": "ch_xxxxx",
    "vendor": "stripe",
    "amount": 50.00,
    "reason": "customer_request"
  }'
```

Cuerpo de la Solicitud:

```
{
  "transaction_id": "ch_xxxxx",          // ID de transacción del proveedor
  "vendor": "stripe",                    // 'stripe', 'paypal', etc.
  "amount": 50.00,                       // Opcional - reembolso completo si se omite
  "reason": "customer_request"           // Razón de reembolso opcional
}
```

Respuesta (200 OK):

```
{
  "success": true,
  "message": "Reembolso procesado",
  "data": {
    "refund_id": "re_xxxxx",
    "amount": 50.00,
    "status": "succeeded"
  }
}
```

API de Billetera

Todos los puntos finales de la billetera utilizan la URL base: `/api/wallet/`

Obtener Saldo de la Billetera

Endpoint: `GET /api/wallet/balance?customer_id={id}`

Solicitud:

```
curl -X GET "https://your-domain.com/api/wallet/balance?
customer_id=123" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

Respuesta (200 OK):

```
{
  "customer_id": 123,
  "balance": 150.50,
  "currency": "USD"
}
```

Obtener Información de la Billetera

Obtener información completa de la billetera y créditos, incluidos los ajustes automáticos.

Endpoint: `GET /api/wallet/info?customer_id={id}`

Solicitud:

```
curl -X GET "https://your-domain.com/api/wallet/info?
customer_id=123" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

Respuesta (200 OK):

```
{
  "customer_id": 123,
  "wallet": {
    "wallet_account_id": 456,
    "balance": 150.50,
    "currency": "USD",
    "auto_recharge_enabled": true,
    "auto_recharge_amount": 100.00,
    "auto_recharge_threshold": 10.00,
    "low_balance_warning_threshold": 10.00
  }
}
```

Recarga de Billetera

Recargar la billetera cargando un método de pago.

Endpoint: `POST /api/wallet/topup`

Flujo: Cargar tarjeta/PayPal → Acreditar billetera

Solicitud:

```
curl -X POST https://your-domain.com/api/wallet/topup \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "amount": 100.00,
    "payment_method_id": 789
  }'
```

Cuerpo de la Solicitud:

```
{
  "customer_id": 123,
  "amount": 100.00,
  "currency": "USD",                    // Predeterminado: USD
  "payment_method_id": 789              // Opcional - usa el
predeterminado si se omite
}
```

Respuesta (200 OK):

```
{
  "success": true,
  "payment": {
    "transaction_id": "ch_xxxxx",
    "amount": 100.00
  },
  "wallet": {
    "ledger_id": 503,
    "balance_after": 250.50,
    "customer_transaction_id": 7001
  },
  "message": "Billetera recargada con 100.00 USD"
}
```

Crea una **línea de crédito de facturación** (`customer_transaction` con `payment_type=wallet_topup_card`, costo minorista negativo, `wallet_ledger_id` establecido). Esto es distinto de la aprovisionamiento

híbrido, donde los déficits de tarjeta acreditan el libro mayor de la billetera como `card_capture_topup` sin una línea de facturación de recarga.

Anulación de recargas de fondos: `DELETE /crm/transaction/transaction_id/{transaction_id}` debita la billetera y reembolsa la tarjeta. Consulte Anular Transacción a continuación.

Obtener Transacciones de la Billetera

Obtener el historial de transacciones de la billetera (libro mayor).

Endpoint: `GET /api/wallet/transactions?customer_id={id}&limit={n}&offset={n}&type={type}`

Solicitud:

```
curl -X GET "https://your-domain.com/api/wallet/transactions?
customer_id=123&limit=50&offset=0&type=credit" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

Parámetros de Consulta:

- `customer_id` (requerido): ID del cliente
- `limit` (opcional): Número de registros (predeterminado: 50)
- `offset` (opcional): Desplazamiento de paginación (predeterminado: 0)
- `type` (opcional): Filtrar por tipo de transacción ('credit', 'debit', 'refund', 'adjustment')

Respuesta (200 OK):

```
{
  "customer_id": 123,
  "count": 2,
  "transactions": [
    {
      "ledger_id": 501,
      "transaction_type": "credit",
      "amount": 100.00,
      "balance_before": 150.50,
      "balance_after": 250.50,
      "description": "Recarga de tarjeta",
      "reference_type": "payment_capture",
      "reference_id": 103,
      "created_at": "2025-12-27T10:35:00Z"
    },
    {
      "ledger_id": 502,
      "transaction_type": "debit",
      "amount": 50.00,
      "balance_before": 250.50,
      "balance_after": 200.50,
      "description": "Cargo por servicio",
      "reference_type": "service_charge",
      "reference_id": 789,
      "created_at": "2025-12-27T11:00:00Z"
    }
  ]
}
```

Anular Transacción

Anular suavemente una transacción de cliente no facturada. Requiere `delete_customer_transaction`.

Endpoint: `DELETE /crm/transaction/transaction_id/{transaction_id}`

Solicitud:

```
curl -X DELETE "https://your-
domain.com/api/transaction/transaction_id/7001" \
  -H "Authorization: Bearer YOUR_API_KEY"
```

Éxito (200 OK): cargo manual simple (sin enlace a billetera):

```
{
  "result": "Success"
}
```

Éxito (200 OK): línea vinculada a billetera (solo reversión):

```
{
  "result": "Success",
  "wallet_reversal": {
    "wallet_reversed": true,
    "wallet_amount": 50.00,
    "wallet_ledger_id": 503
  }
}
```

Éxito (200 OK): recarga de fondos explícita de tarjeta (`payment_type=wallet_topup_card`):

```
{
  "result": "Success",
  "wallet_reversal": {
    "wallet_reversed": true,
    "wallet_amount": 100.00,
    "wallet_ledger_id": 503,
    "refund": {
      "capture_id": 88,
      "vendor": "stripe",
      "vendor_transaction_id": "ch_xxxxx",
      "refund_id": "re_xxxxx",
      "status": "succeeded",
      "amount": 100.00
    }
  }
}
```


Política de Anulación / Reembolso

Fuente de transacción	`wallet_ledger_id`	Billetera en anulación	Reembolso de tarjeta en anulación
Cargo/crédito manual	No	No	No
Recarga de Fondos (`POST /wallet/topup`)	Sí	Debitar billetera	**Sí** (`wallet_topup_card`
Crédito manual de billetera	Sí	Debitar billetera	No
Factura pagada desde billetera	Sí	Acreditar billetera	No
Servicio / aprovisionamiento (captura híbrida)	Generalmente **no** en el cargo CT	No*	No

*Los flujos híbridos dirigen tarjeta → billetera → débito de servicio internamente (`card_capture_topup` + filas de libro mayor de `service_charge`). La línea de facturación visible es el cargo por servicio, no una línea de recarga. Se planea una reversión más rica para esos cargos en la rama de documentos financieros.

Errores (500):

```
{
  "result": "Failed",
  "Reason": "Saldo insuficiente en la billetera. Disponible: 0.00,
Requerido: 20.00"
}
```

- **Transacción facturada:** `"No se puede eliminar una transacción que está asociada con una factura"`

- **Saldo insuficiente:** el cliente ya gastó fondos recargados; se rechaza la anulación (sin anulación parcial)
- **Fallo en reembolso de tarjeta:** la reversión de billetera se revierte; la transacción permanece no anulada

Nota: La anulación marca `void=true` (excluida de facturación/estadísticas). No es lo mismo que un reembolso de factura.

Resumen de Referencia de API

Puntos finales de Métodos de Pago

Método	Endpoint	Descripción
POST	`/api/payments/methods`	Agregar método de pago
GET	`/api/payments/methods?customer_id={id}`	Obtener todos los métodos de pago
GET	`/api/payments/methods/default?customer_id={id}`	Obtener método de pago predeterminado
PUT	`/api/payments/methods/set-default`	Establecer método de pago predeterminado
DELETE	`/api/payments/methods/{id}?customer_id={id}`	Eliminar método de pago

Puntos finales de Flujos de Pago

Método	Endpoint	Descripción
POST	`/api/payments/charge`	Pago directo (un paso)
POST	`/api/payments/invoice`	Pago de factura (primero a billetera)
POST	`/api/payments/topup`	Recarga con aprovisionamiento
POST	`/api/payments/authorize/hold`	Crear retención de autorización
POST	`/api/payments/capture/{id}`	Capturar autorización
POST	`/api/payments/release/{id}`	Liberar autorización
POST	`/api/payments/rental`	Pago de alquiler/pago de terceros
POST	`/api/payments/refund`	Reembolsar pago

Puntos finales de Billetera

Método	Endpoint	Descripción
GET	`/api/wallet/balance?customer_id={id}`	Obtener saldo de billetera
GET	`/api/wallet/info?customer_id={id}`	Obtener información de billetera + configuraciones
POST	`/api/wallet/topup`	Recargar billetera a través de método de pago (crea línea de facturación `wallet_topup_card`)
GET	`/api/wallet/transactions?customer_id={id}`	Obtener historial de transacciones

Puntos finales de Transacción

Método	Endpoint	Descripción
PUT	`/api/transaction/`	Crear transacción de cliente
GET	`/api/transaction/`	Listar transacciones (paginadas)
DELETE	`/api/transaction/transaction_id/{id}`	Anular transacción; reversión de billetera; reembolso de tarjeta solo para recargas de fondos

Puntos finales Específicos de PayPal

Método	Endpoint	Descripción
POST	`/api/payments/paypal/vault/setup-token`	Crear token de configuración de PayPal para SDK de Campos de Tarjeta
POST	`/api/payments/paypal/vault/finalize`	Finalizar el almacenamiento de PayPal y guardar método de pago
POST	`/api/payments/paypal/vault/update-setup-token`	Actualizar token de configuración para manejo de 3DS/SCA

Casos de Uso Comunes

Caso de Uso 1: Cargar al Cliente por Servicio (Prepagado)

Escenario: Cliente pagando por 30 días de servicio usando enrutamiento primero a la billetera.

```
# Pago de factura (primero a billetera, tarjeta para déficit)
curl -X POST https://your-domain.com/api/payments/invoice \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "amount": 99.99,
    "metadata": {
      "service_id": 456,
      "contract_days": 30,
      "description": "Servicio Premium de 30 días"
    }
  }'
```

Respuesta:

```
{
  "success": true,
  "message": "Pago de factura procesado",
  "data": {
    "service_amount": 99.99,
    "wallet_portion_used": 50.00,        // La billetera tenía $50
    "card_portion_used": 49.99,          // Se cargó la tarjeta $49.99
    "final_balance": 0.00
  }
}
```

Caso de Uso 2: Pago en Dos Fases con Aprovisionamiento de Servicio

Escenario: Aprovisionar un servicio solo después de que se autorice el pago. Si el aprovisionamiento falla, no se realiza el cargo.

```
# Pago de recarga con aprovisionamiento automático
curl -X POST https://your-domain.com/api/payments/topup \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "amount": 149.99,
    "payment_method_id": 789,
    "service_uuid": "fiber-svc-uuid",
    "imsi": "123456789012345",
    "days": 30,
    "metadata": {
      "service_type": "fiber_internet"
    }
  }'
```

Flujo:

1. Autorizar $149.99 en la tarjeta
2. Aprovisionar servicio de internet de fibra
3. Si el aprovisionamiento tiene éxito → capturar pago
4. Si el aprovisionamiento falla → liberar autorización (sin cargo)

Caso de Uso 3: Pago Anónimo de Hotspot

Escenario: Usuario anónimo paga por acceso a WiFi hotspot. Sin registro de cliente, sin método de pago guardado.

```
# Pago anónimo de hotspot
curl -X POST https://your-domain.com/api/payments/topup \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 1,
    "amount": 5.00,
    "vendor": "stripe",
    "payment_token": "pm_xxxxx",
    "service_uuid": "hotspot-downtown",
    "imsi": "999999999999999",
    "days": 1,
    "metadata": {
      "is_rental": true,
      "billing_email": "user@example.com",
      "hotspot_location": "Café Downtown"
    }
  }'
```

Resultado:

- Pago procesado
- Hotspot activado
- Método de pago NO guardado
- No se creó registro de cliente

Caso de Uso 4: Reserva de Hotel (Retención + Captura)

Escenario: Retener fondos para reserva de hotel, capturar al hacer el check-in, liberar si se cancela.

```
# Paso 1: Check-in - retener fondos
curl -X POST https://your-domain.com/api/payments/authorize/hold \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "amount": 500.00,
    "use_wallet": true,
    "metadata": {
      "reservation_id": "RES-2025-001"
    }
  }'

# Respuesta: {"authorization_id": 301, "status": "authorized"}

# Paso 2a: El cliente hace check-in - capturar
curl -X POST https://your-domain.com/api/payments/capture/301 \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "metadata": {
      "invoice": true,
      "title": "Estancia en Hotel - 3 Noches",
      "description": "Habitación 302, 27-30 de diciembre",
      "send_email": true
    }
  }'

# Paso 2b: El cliente cancela - liberar retención
curl -X POST https://your-domain.com/api/payments/release/301 \
  -H "Authorization: Bearer YOUR_API_KEY"
```

Caso de Uso 5: Agregar Método de Pago (Stripe)

Escenario: El cliente agrega una nueva tarjeta Visa a su cuenta.

```
# Paso 1: El frontend crea un token de Stripe a través de
Stripe.js
# const {token} = await stripe.createToken(cardElement);

# Paso 2: El backend guarda el método de pago
curl -X POST https://your-domain.com/api/payments/methods \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "vendor": "stripe",
    "payment_token": "pm_xxxxx",
    "is_default": true,
    "card_nickname": "Visa de Trabajo"
  }'
```

Caso de Uso 6: Recargar Billetera

Escenario: El cliente recarga su billetera con $100 usando un método de pago guardado.

```
curl -X POST https://your-domain.com/api/wallet/topup \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "amount": 100.00
  }'
```

Respuesta:

```
{
  "success": true,
  "payment": {
    "transaction_id": "ch_xxxxx",
    "amount": 100.00
  },
  "wallet": {
    "balance_after": 250.50
  },
  "message": "Billetera recargada con 100.00 USD"
}
```

Manejo de Errores

Respuestas de Error Comunes

Fondos Insuficientes (400 Bad Request):


```
{
  "error": "Saldo insuficiente en la billetera. Disponible: 50.00,
Requerido: 200.00"
}
```


Fallo en el Pago (400 Bad Request):


```
{
  "error": "El pago falló. Por favor, inténtelo de nuevo."
}
```


Error de Validación (400 Bad Request):


```
{
  "error": "customer_id y amount son requeridos"
}
```

No Encontrado (404 Not Found):

```
{
  "error": "Autorizaci��n 999 no encontrada"
}
```

Error del Servidor (500 Internal Server Error):

```
{
  "error": "Ocurrió un error al procesar su pago"
}
```

Reembolsos y Enrutamiento Primero a la Billetera

Opciones de Reembolso

El sistema soporta **dos tipos de reembolsos** dependiendo de las necesidades de su negocio:

1. Reembolso a la Fuente de Pago (Stripe/PayPal)

Caso de Uso: El cliente solicita un reembolso completo por un servicio cancelado o un producto defectuoso.

Flujo: El dinero regresa al método de pago original (tarjeta, cuenta de PayPal, etc.)

Endpoint: `POST /api/payments/refund`

Ejemplo:

```
curl -X POST https://your-domain.com/api/payments/refund \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "transaction_id": "ch_xxxxx",
    "vendor": "stripe",
    "amount": 100.00,
    "reason": "customer_request"
  }'
```

Resultado:

- Stripe/PayPal procesa el reembolso al método de pago original
- El cliente ve el crédito en su tarjeta/cuenta de PayPal dentro de 5-10 días hábiles
- El dinero NO va a la billetera
- Se mantiene un registro completo en la tabla `PaymentCapture`

Cuándo Usar:

- □ El cliente canceló el pedido
- □ Servicio no entregado
- □ Error de facturación
- □ El cliente solicita explícitamente un reembolso a la tarjeta

2. Crédito a la Billetera

Caso de Uso: Reembolso parcial, crédito por servicio o mantener fondos para futuras compras.

Flujo: El dinero se acredita al saldo de la billetera del cliente para uso inmediato.

Nota: Los créditos de billetera generalmente son manejados internamente por el sistema durante escenarios de error. Para créditos manuales de billetera, comuníquese con el soporte o use herramientas administrativas.

Resultado:

- Fondos inmediatamente disponibles en la billetera
- Sin período de espera
- Pueden usarse para futuras compras
- Registro completo en la tabla `WalletLedger`

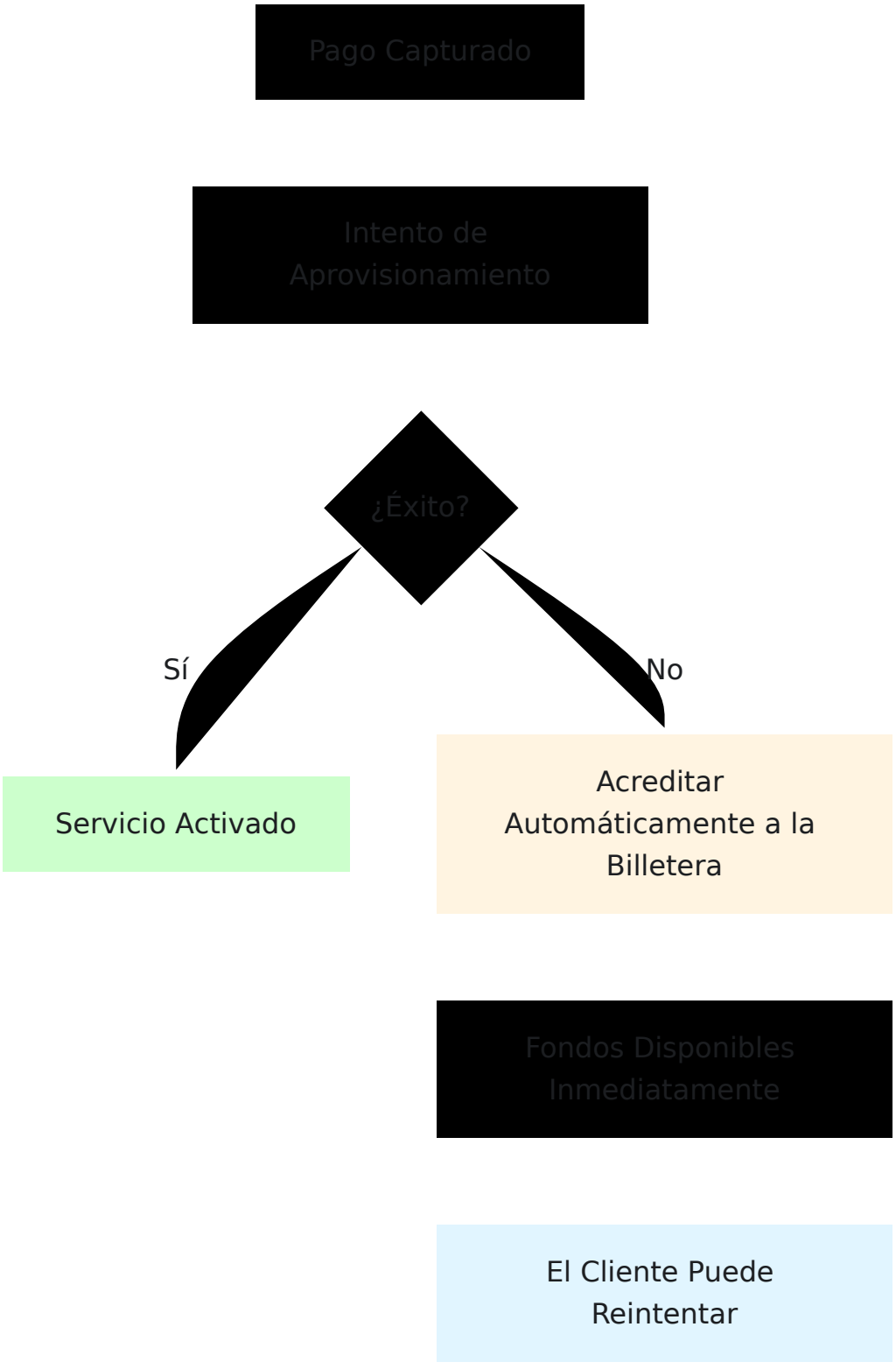
Cuándo Usar:

- □ Créditos por servicio (por ejemplo, compensación por tiempo de inactividad)
- □ Reembolsos parciales donde el cliente volverá a comprar
- □ Compensación por error (fallo en el aprovisionamiento)
- □ Créditos promocionales

Estrategia de Reembolso Híbrido

Mejor Práctica: Para escenarios de error durante el flujo de pago, el sistema acredita automáticamente la billetera en lugar de reembolsar la tarjeta.

Recuperación Automática de Errores:



Escenario de Ejemplo:

1. El cliente paga $100 por un servicio
2. La tarjeta se carga con éxito
3. El aprovisionamiento falla

4. En lugar de reembolsar $100 a la tarjeta (5-10 días + tarifas de reembolso):
    - Acreditar $100 a la billetera de inmediato
    - El cliente puede volver a intentar la compra al instante
    - Sin tarifas de reembolso
    - Mejor experiencia de usuario

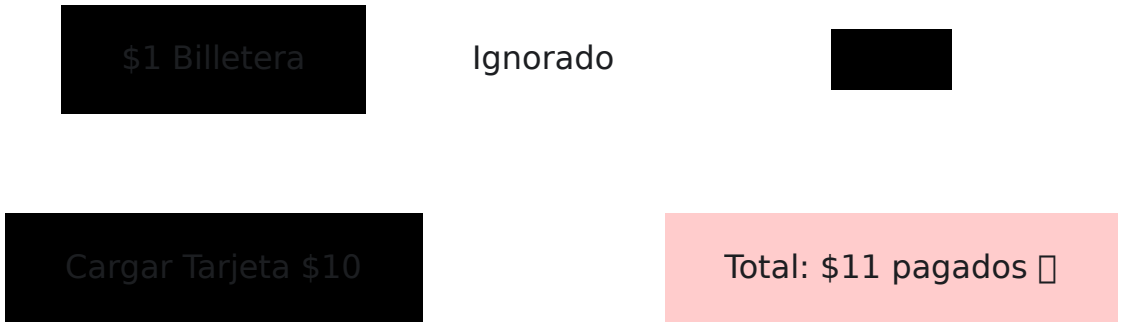
Enrutamiento Primero a la Billetera: Cargos de Tarjeta Optimizados

El sistema **solo carga su tarjeta por el déficit**, no por el monto total, cuando tiene saldo en la billetera.

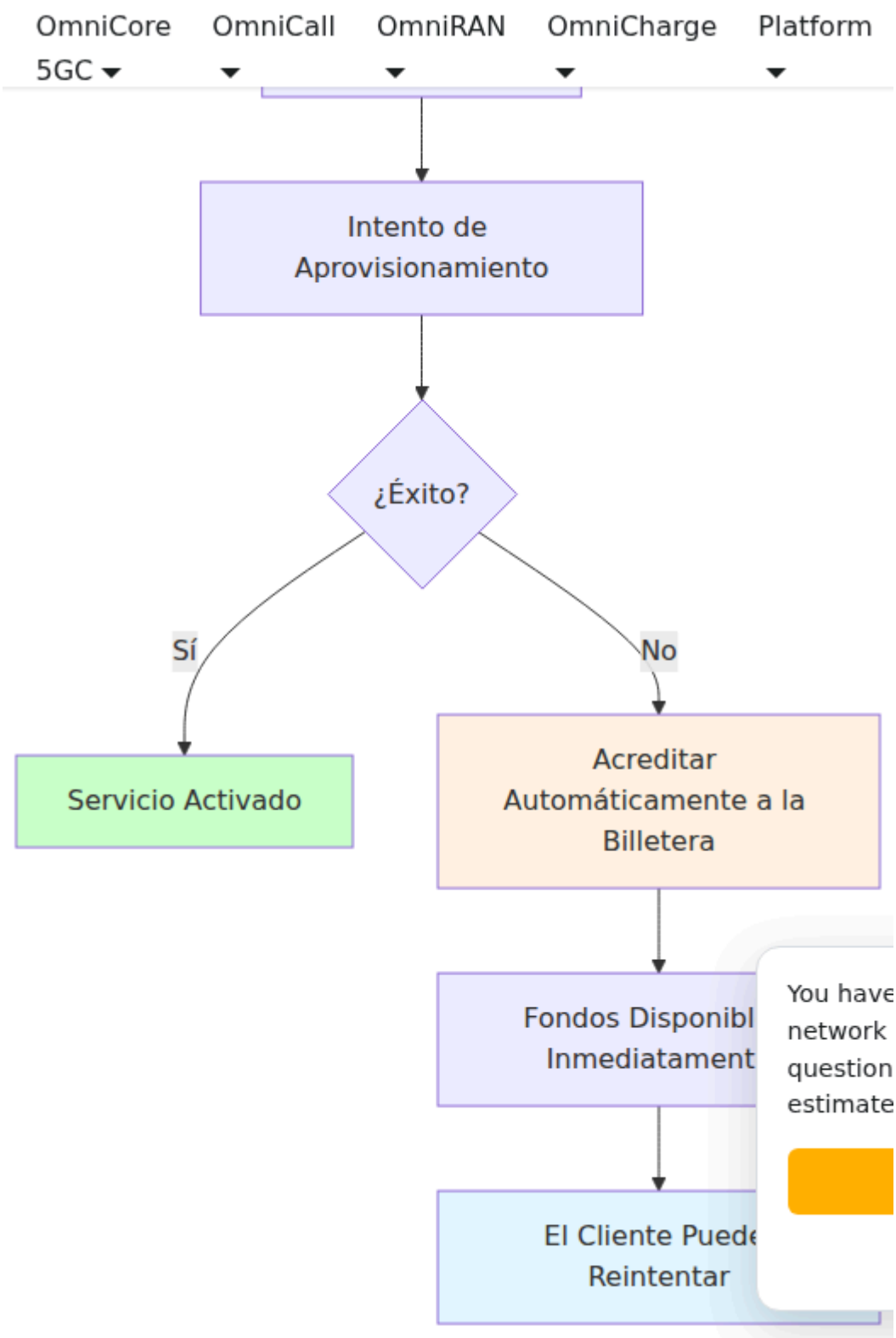
Ejemplo 1: Saldo de Billetera de $1 + Compra de $10

Escenario: Tiene $1 en su billetera y desea comprar un complemento de $10.

Flujo de Pago Tradicional (NO es como funciona este sistema):



Flujo Primero a la Billetera de OmniCRM:



Solicitud API:

```
curl -X POST https://your-domain.com/api/payments/invoice \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "amount": 10.00,
    "metadata": {
      "addon_id": 456
    }
  }'
```

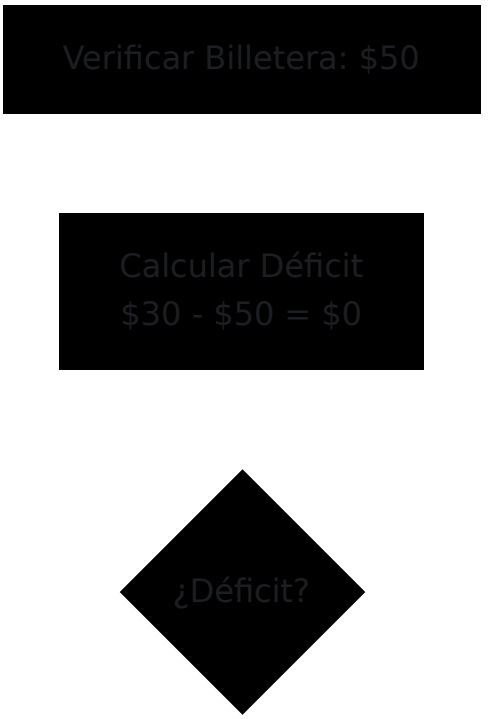
Respuesta API:

```
{
  "success": true,
  "message": "Pago de factura procesado",
  "data": {
    "customer_id": 123,
    "service_amount": 10.00,
    "routing_mode": "hybrid",
    "initial_balance": 1.00,
    "wallet_portion_used": 1.00,     // Usó el $1 existente
    "card_portion_used": 9.00,       // Solo se cargó $9 de
déficit
    "charged_amount": 9.00,          // ← Cargo a la tarjeta
    "wallet_credited": 9.00,         // ← Recargó la billetera
    "wallet_debited": 10.00,         // ← Cargo por servicio
    "final_balance": 0.00
  }
}
```

Ejemplo 2: Saldo de Billetera de $50 + Compra de $30

Escenario: Tiene $50 en la billetera y el costo de la compra es $30.

Flujo Primero a la Billetera:



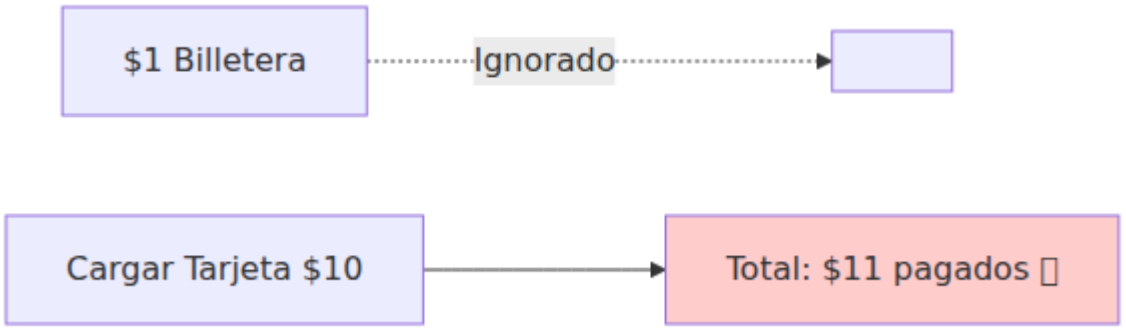
Respuesta API:

```
{
  "success": true,
  "data": {
    "service_amount": 30.00,
    "initial_balance": 50.00,
    "charged_amount": 0,              // ← Sin cargo a la tarjeta
    "wallet_debited": 30.00,
    "final_balance": 20.00,
    "payment_method_used": false      // ← Tarjeta no utilizada
  }
}
```

Ejemplo 3: Autorización de Retención con Saldo de Billetera

Escenario: Hotel retiene $500 de autorización con $150 de saldo en la billetera.

Paso 1: Autorización (`POST /api/payments/authorize/hold`):



Respuesta API:

```
{
  "authorization_id": 301,
  "amount": 500.00,
  "status": "authorized",
  "wallet_balance": 150.00,
  "wallet_to_use": 150.00,            // La billetera cubrirá
esta cantidad
  "card_amount": 350.00,
  "message": "Tarjeta autorizada por $350 (recarga de billetera).
El débito de la billetera de $500 ocurrirá en la captura."
}
```

Paso 2: Captura (`POST /api/payments/capture/301`):

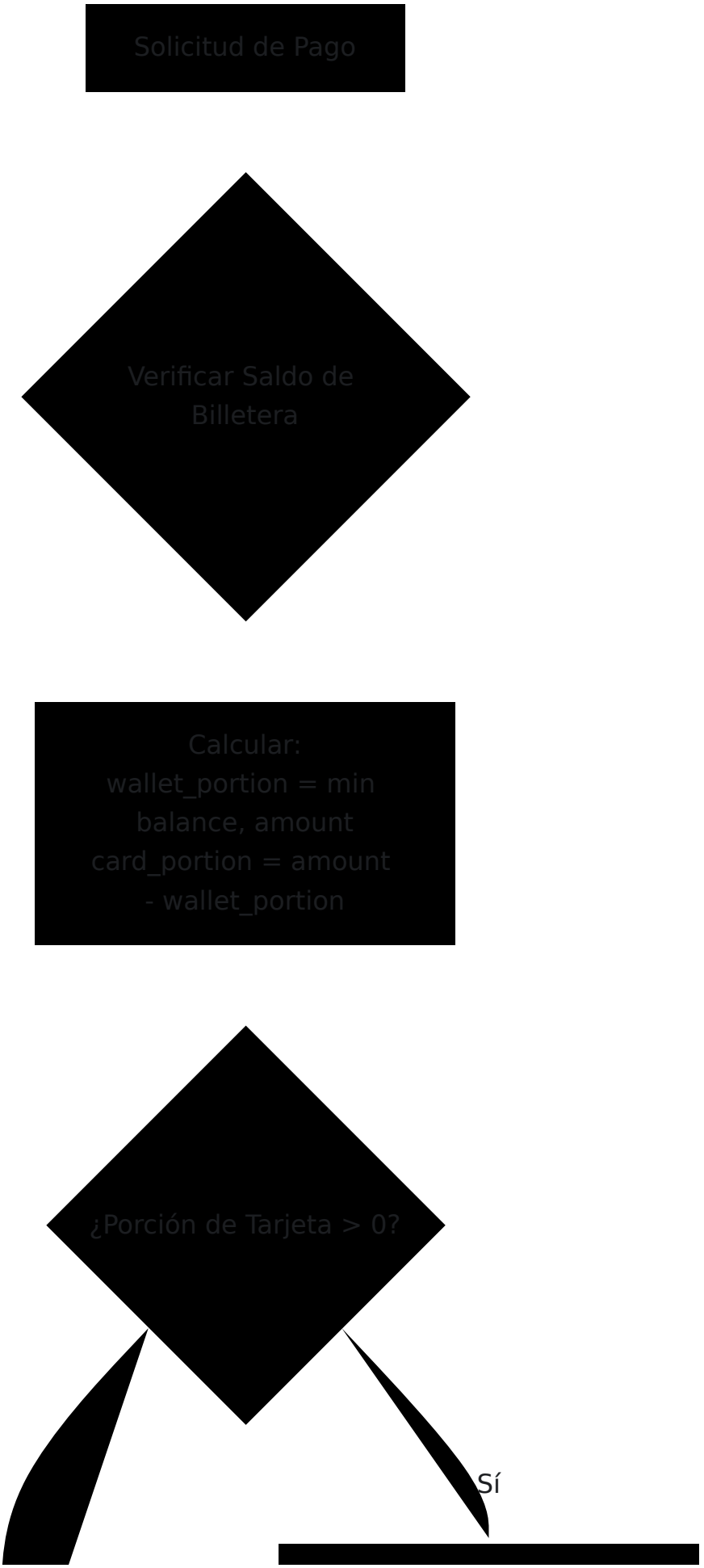


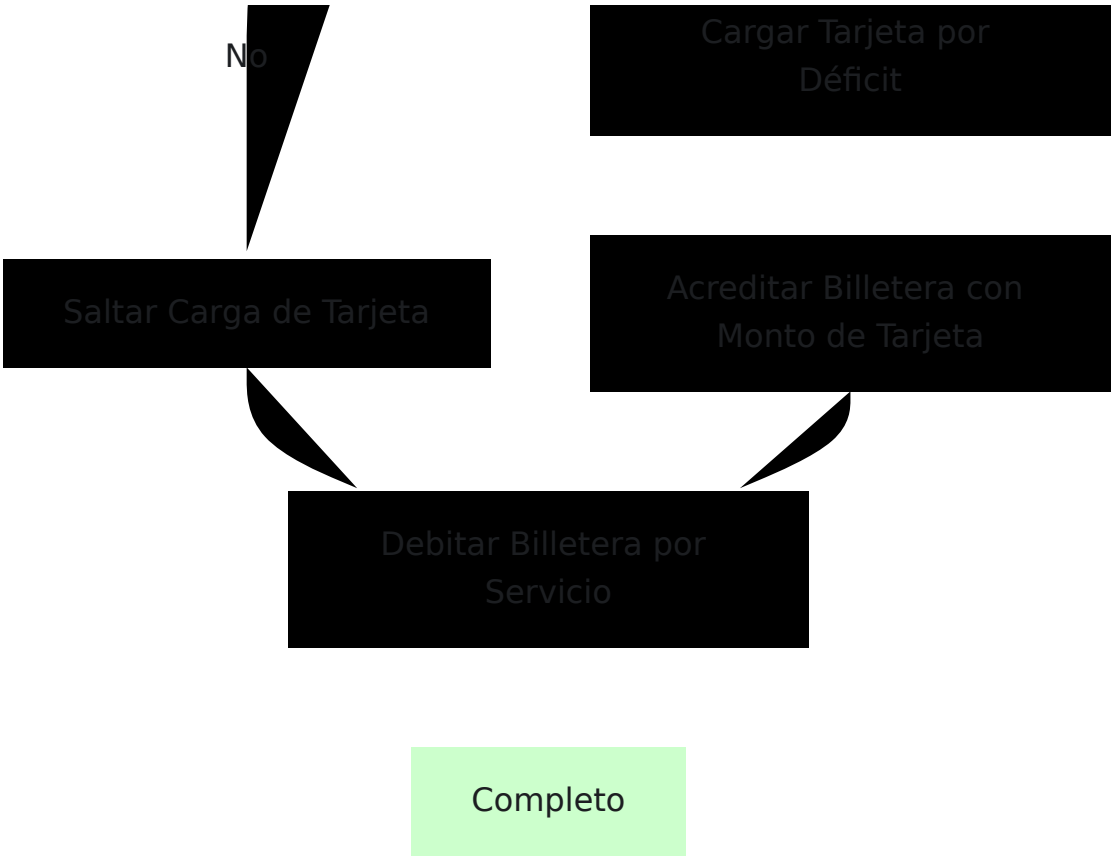
Por qué Esto Importa:

- □ El cliente solo tiene $350 "retenidos" en su tarjeta, no $500
- □ Reduce el impacto en el crédito disponible del cliente
- □ Montos de autorización más precisos
- □ Mejor experiencia para el cliente

Detalles de Implementación

Lógica de Enrutamiento Primero a la Billetera:


Solicitud de Pago
Verificar Saldo de Billetera
Calcular:
wallet_portion = min balance, amount
card_portion = amount - wallet_portion
¿Porción de Tarjeta > 0?
Sí



Cómo Funciona:

1. El sistema verifica el saldo actual de la billetera
2. Calcula el déficit: `amount - wallet_balance`
3. Si el déficit > 0, carga la tarjeta solo por el déficit
4. Acredita la billetera con el pago de la tarjeta
5. Debita la billetera por el monto total del servicio
6. Resultado: El cliente solo se carga por lo que la billetera no pudo cubrir

Anulación de Modo de Enrutamiento

Puede anular el comportamiento primero a la billetera si es necesario:

Modo de Bypass - Siempre cargar tarjeta incluso si la billetera tiene fondos:

```
curl -X POST https://your-domain.com/api/payments/charge \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_API_KEY" \
  -d '{
    "customer_id": 123,
    "amount": 100.00,
    "metadata": {
      "routing_mode": "bypass"
    }
  }'
```

Resultado: Tarjeta cargada $100, billetera acreditada $100, billetera debita $100 (neto: $0)

Caso de Uso: El cliente quiere usar la tarjeta por recompensas/puntos incluso con saldo en la billetera.

Mejores Prácticas

1. **Siempre use el enrutamiento primero a la billetera** para los pagos de los clientes para habilitar la funcionalidad prepagada
2. **Utilice pagos en dos fases** (punto final de recarga) al aprovisionar servicios para evitar cargos si falla el aprovisionamiento
3. **Establezca metadatos** para todos los pagos para mantener un registro de auditoría
4. **Utilice pagos anónimos** solo para usuarios verdaderamente anónimos (hotspots, alquileres)
5. **Maneje errores con gracia** y proporcione comentarios claros al usuario
6. **Pruebe los flujos de pago** en escenarios de éxito y fallo
7. **Monitoree las expiraciones de autorización** (típicamente 7 días para tarjetas)
8. **Implemente recargas automáticas** para servicios recurrentes para mejorar la experiencia del cliente

Mejores Prácticas Específicas para Playbook

Al implementar flujos de pago en playbooks de Ansible:

1. **Siempre use el patrón block/rescue** - Envolver aprovisionamiento en try/catch para retroceso automático
2. **Almacene authorization_id** - Guarde para operaciones de captura/liberación
3. **Valide las respuestas de la API** - Asegúrese de que sea exitosa antes de continuar
4. **Redondee valores de moneda** - Siempre use 2 decimales
5. **Verifique métodos de pago** - Verifique que el cliente tenga un método de pago predeterminado antes de la autorización

Consulte Mejores Prácticas de Playbook para obtener detalles completos y ejemplos.

Autenticación

Todos los puntos finales de la API requieren autenticación a través de la clave API en el encabezado `Authorization`:

```
Authorization: Bearer YOUR_API_KEY
```

Comuníquese con su administrador del sistema para la provisión de la clave API.

Soporte de Proveedores

Actualmente Soportados

- **Stripe** - Soporte completo (tarjetas, ACH)

- **PayPal** - Soporte completo (cuentas de PayPal, tarjetas a través de SDK de Campos de Tarjeta)

Agregar Nuevos Proveedores

La arquitectura modular hace que sea trivial agregar nuevos proveedores de pagos. Consulte la sección Arquitectura Modular para obtener detalles.

Documentación Relacionada

Guías de Implementación

- **Cargos y Pagos desde Playbooks** - Implementación de playbook de Ansible
  - Patrón de Compromiso en Dos Fases
  - Ejemplos de Autorización/Captura
  - Creación de Transacciones Directas
  - Cálculos Pro-Rata

Guías Específicas de Funciones

Facturas de Clientes

> **Referencia de API**: Para las API y flujos de trabajo de pago de facturas, consulta la **Guía de API del Sistema de Pagos**

Transacciones se agrupan para formar una factura, que se envía al cliente para su pago.

Las facturas tienen una fecha de inicio y una fecha de finalización, que es el período que cubre la factura, y una fecha de vencimiento, que es la fecha en que la factura debe ser pagada.

Las facturas pueden ser generadas automáticamente por el sistema, por ejemplo, cuando se factura un servicio, se crea una factura por el costo minorista, o pueden ser creadas manualmente, por ejemplo, si un cliente solicita una copia de una factura, o si un cliente es facturado por un cargo único.

Las facturas de clientes están completamente plantilladas con Mailjet y pueden ser personalizadas para incluir el logotipo de la empresa, dirección y detalles de pago, y pueden ser enviadas al cliente por correo electrónico, o descargadas como un PDF.

Personalizando Plantillas de Facturas

OmniCRM utiliza plantillas HTML con Jinja2 para generar facturas. Puedes personalizar completamente el diseño de la factura, la marca, los colores y el diseño.

Ubicación de la Plantilla de Factura

Las plantillas de factura se almacenan en `OmniCRM-API/invoice_templates/`

Plantillas Predeterminadas:

- `sample_invoice_template.html` - Plantilla de factura de muestra
- `cifi_invoice_template.html` - Ejemplo de plantilla alternativa

Configuración:

La plantilla de factura activa se especifica en `OmniCRM-API/crm_config.yaml`:

```
invoice:
  template_filename: 'sample_invoice_template.html'
```

Variables de Plantilla Disponibles

Las plantillas de factura tienen acceso a las siguientes variables de Jinja2:

Información de la Factura:

- `{{ invoice_number }}` - ID de factura único (por ejemplo, `INV-2025-001234`)
- `{{ date }}` - Fecha de emisión de la factura (formato ISO: `2025-01-10T12:00:00`)
- `{{ due_date }}` - Fecha de vencimiento del pago (por ejemplo, `2025-02-10`)
- `{{ start_date }}` - Fecha de inicio del período de facturación
- `{{ end_date }}` - Fecha de finalización del período de facturación
- `{{ total_amount }}` - Monto total de la factura antes de impuestos (numérico)
- `{{ total_tax }}` - Monto total de impuestos calculado a partir de todas las transacciones (numérico)

Información del Cliente:

- `{{ client.name }}` - Nombre completo del cliente o nombre de la empresa
- `{{ client.address.address_line_1 }}` - Línea de dirección 1
- `{{ client.address.address_line_2 }}` - Línea de dirección 2
- `{{ client.address.city }}` - Ciudad
- `{{ client.address.state }}` - Estado/provincia
- `{{ client.address.zip_code }}` - Código postal
- `{{ client.address.country }}` - País

Artículos de Línea de Transacción:

Recorre las transacciones usando:

```
{% for sub_transaction in transactions %}
    <tr>
        <td>{{ sub_transaction.transaction_id }}</td>
        <td>{{ sub_transaction.created.split("T")[0] }}</td>
        <td>{{ sub_transaction.title }}</td>
        <td>{{ sub_transaction.description }}</td>
        <td>${{ "%.2f"|format(sub_transaction.retail_cost) }}</td>
    </tr>
{% endfor %}
```

Campos de Transacción:

- `sub_transaction.transaction_id` - ID de transacción
- `sub_transaction.created` - Fecha/hora de la transacción
- `sub_transaction.title` - Título de la transacción
- `sub_transaction.description` - Descripción detallada
- `sub_transaction.retail_cost` - Monto del artículo
- `sub_transaction.tax_percentage` - Porcentaje de impuestos aplicado (por ejemplo, 10 para 10%)
- `sub_transaction.tax_amount` - Monto de impuestos calculado en dólares

Mostrando Impuestos en Plantillas:

```
<td>
    {% if sub_transaction.tax_amount and
sub_transaction.tax_amount > 0 %}
        ${{ "%.2f"|format(sub_transaction.tax_amount) }} ({{
sub_transaction.tax_percentage }}%)
    {% else %}
        -
    {% endif %}
</td>
```

Creando una Plantilla de Factura Personalizada

Paso 1: Copiar Plantilla Existente

```
cd OmniCRM-API/invoice_templates/
cp sample_invoice_template.html your_company_invoice_template.html
```

Paso 2: Personalizar HTML/CSS

Edita `your_company_invoice_template.html` para que coincida con tu marca:

Áreas Clave de Personalización:

1. **Logotipo de la Empresa y Marca**

```
<!-- Reemplaza con la URL de tu logotipo -->
![Tu Empresa](https://yourcompany.com/logo.png)

<!-- Actualiza el nombre de la empresa -->
<h1>Nombre de Tu Empresa</h1>
```

2. **Esquema de Color**

```
<style>
    /* Color de marca principal */
    .navbar {
        background: linear-gradient(to bottom right, #your-
color-1, #your-color-2);
    }

    /* Encabezados de tabla */
    .table thead th {
        background-color: #your-brand-color !important;
        color: white !important;
    }

    /* Botones y enlaces */
    .btn-primary {
        background-color: #your-brand-color;
    }
</style>
```

3. **Información de la Empresa en el Pie de Página**

```
<footer>
    <p>Nombre de Tu Empresa</p>
    <p>Calle 123, Ciudad, País</p>
    <p>Teléfono: +1-555-123-4567 | Correo:
billing@yourcompany.com</p>
    <p>ABN/ID Fiscal: 12345678900</p>
</footer>
```

4. **Instrucciones de Pago**

```
<div class="payment-info">
    <h3>Métodos de Pago</h3>
    <p><strong>En Línea:</strong> Paga en
https://yourcompany.com/pay</p>
    <p><strong>Transferencia Bancaria:</strong></p>
    <ul>
        <li>Nombre de la Cuenta: Tu Empresa Ltd</li>
        <li>BSB: 123-456</li>
        <li>Número de Cuenta: 987654321</li>
        <li>Referencia: {{ invoice_number }}</li>
    </ul>
</div>
```

5. **Términos y Condiciones**

```
<div class="terms">
    <h4>Términos de Pago</h4>
    <p>Pago debido dentro de los 30 días a partir de la fecha
de la factura.</p>
    <p>Recargos por pago tardío: 2% por mes sobre saldos
vencidos.</p>
    <p>Para consultas de facturación:
billing@yourcompany.com</p>
</div>
```

Paso 3: Actualizar Configuración

Edita `OmniCRM-API/crm_config.yaml`:

```
invoice:
  template_filename: 'your_company_invoice_template.html'
```

Paso 4: Reiniciar API

```
cd OmniCRM-API
sudo systemctl restart omnicrm-api
```

Paso 5: Probar la Generación de Facturas

1. Navega a un cliente con transacciones
2. Genera una factura de prueba
3. Descarga el PDF para verificar el formato
4. Envía la factura a ti mismo para probar la entrega por correo electrónico

Personalización Avanzada

Contenido Condicional:

Usa condicionales de Jinja2 para mostrar/ocultar contenido:

```
{% if total_amount > 1000 %}
    <div class="high-value-notice">
        <p><strong>Nota:</strong> Saldo alto - Plan de pago
disponible a solicitud.</p>
    </div>
{% endif %}

{% if client.address.country == "Australia" %}
    <p>GST Incluido: ${{ "%.2f"|format(total_amount * 0.10) }}</p>
{% endif %}
```

Soporte Multilenguaje:

Crea plantillas específicas para cada idioma:

```
invoice_template_en.html
invoice_template_es.html
invoice_template_fr.html
```

Configura según la preferencia de idioma del cliente.

Cálculos Personalizados:

```
<!-- Mostrar subtotal y desglose de impuestos -->
<tr>
    <td colspan="4" class="text-right"><strong>Subtotal:</strong>
</td>
    <td>${{ "%.2f"|format(total_amount) }}</td>
</tr>
<tr>
    <td colspan="4" class="text-right"><strong>Impuesto:</strong>
</td>
    <td>${{ "%.2f"|format(total_tax) }}</td>
</tr>
<tr>
    <td colspan="4" class="text-right"><strong>Total:</strong>
</td>
    <td>${{ "%.2f"|format(total_amount + total_tax) }}</td>
</tr>
```

Nota: La variable `total_tax` se calcula automáticamente sumando el `tax_amount` de todas las transacciones en la factura. El impuesto de cada transacción se calcula en función de su campo `tax_percentage`, que por defecto es el `tax_percentage` del producto o 0% si no se especifica.

Código QR para Pago:

Genera un código QR para el pago móvil:

```
<div class="qr-payment">
    ![Escanea para Pagar](https://api.qrserver.com/v1/create-qr-
code/?size=150x150&data={{ payment_url }})
    <p>Escanea con tu teléfono para pagar al instante</p>
</div>
```

Mejores Prácticas de Estilo para PDF

OmniCRM utiliza **WeasyPrint** para convertir HTML a PDF. Sigue estas directrices:

CSS Soportado:

- La mayoría de las propiedades CSS 2.1
- CSS3 limitado (flexbox, algunas transformaciones)
- Fuentes web a través de `@font-face`

No Soportado:

- JavaScript
- CSS Grid (usa tablas en su lugar)
- Animaciones complejas
- Algunas propiedades CSS modernas

Tamaño de Página y Márgenes:

```
@page {
    size: A4;
    margin: 1cm;
}

body {
    font-family: Arial, sans-serif;
    font-size: 10pt;
}
```

Estilo Específico para Impresión:

```
@media print {
    .no-print {
        display: none;
    }

    .page-break {
        page-break-after: always;
    }
}
```

Diseño de Tabla:

```
.table {
    table-layout: fixed;
    width: 100%;
}

.table th, .table td {
    word-wrap: break-word;
    padding: 4px;
}
```

Incorporación de Fuentes:

Para fuentes personalizadas, usa fuentes seguras para la web o incrusta:

```
@font-face {
    font-family: 'YourFont';
    src: url('https://yourcompany.com/fonts/yourfont.woff2')
format('woff2');
}

body {
    font-family: 'YourFont', Arial, sans-serif;
}
```

Pruebas de Plantillas de Facturas

Lista de Verificación de Pruebas:

1. **Inspección Visual:**
   - El logotipo se muestra correctamente
   - Los colores coinciden con las pautas de la marca
   - El texto es legible (no demasiado pequeño)
   - Las tablas están alineadas correctamente
   - Todas las secciones están presentes
2. **Precisión de Datos:**
   - Detalles del cliente correctos
   - Los montos de las transacciones suman correctamente
   - Fechas formateadas correctamente
   - Todas las variables sustituyen correctamente
3. **Calidad del PDF:**
   - Tamaño de archivo razonable (<5MB)
   - Imágenes nítidas y claras
   - Sin texto cortado o desbordado
   - Saltos de página en lugares apropiados
4. **Facturas de Múltiples Páginas:**
   - Encabezados se repiten en cada página
   - Números de página se muestran
   - Listas de transacciones largas se paginan correctamente
5. **Entrega por Correo Electrónico:**
   - El PDF se adjunta al correo electrónico
   - Tamaño de archivo bajo el límite de Mailjet (15MB)
   - Se renderiza en Gmail, Outlook, Apple Mail

Comando de Prueba (Generación Manual):

Puedes probar la generación de facturas a través de la API:

```
curl -X GET "http://localhost:5000/crm/invoice/{invoice_id}/pdf" \
     -H "Authorization: Bearer YOUR_TOKEN" \
     --output test_invoice.pdf
```

Problemas Comunes de Plantillas

Variables no sustituyendo:

- **Causa:** Error tipográfico en el nombre de la variable o datos faltantes
- **Solución:** Verifica la ortografía exactamente (sensible a mayúsculas y minúsculas), verifica que los datos existan en la base de datos

Estilo de PDF roto:

- **Causa:** Propiedad CSS no soportada
- **Solución:** Usa propiedades CSS 2.1, prueba con CSS compatible con WeasyPrint

Imágenes no se muestran:

- **Causa:** URLs relativas o recursos externos bloqueados
- **Solución:** Usa URLs HTTPS absolutas, asegúrate de que las imágenes sean accesibles públicamente

Tablas desbordando la página:

- **Causa:** Anchos de columna fijos demasiado anchos
- **Solución:** Usa anchos porcentuales, `table-layout: fixed`

Fuentes no se renderizan:

- **Causa:** Fuente no incrustada o no disponible
- **Solución:** Usa fuentes seguras para la web (Arial, Times New Roman, etc.) o incrusta correctamente fuentes personalizadas

La generación de PDF falla:

- **Causa:** Errores de sintaxis HTML o fallo de WeasyPrint
- **Solución:** Valida el HTML, verifica los registros de WeasyPrint, simplifica diseños complejos

Caché de PDF de Factura

Para mejorar el rendimiento y reducir la generación redundante de PDF, OmniCRM incluye un sistema de caché de PDF de facturas. Cuando se genera por primera vez un PDF de factura, se almacena en la base de datos para solicitudes posteriores.

Cómo Funciona la Caché de PDF:

1. **Primera Solicitud** - Cuando se solicita un PDF de factura (descarga o correo electrónico), el sistema:
   - Genera el PDF a partir de la plantilla de factura
   - Codifica el PDF como Base64
   - Calcula un hash SHA256 del contenido del PDF
   - Almacena en la tabla `Invoice_PDF_Cache` con:
     - Referencia de ID de factura
     - Datos del PDF (codificados en Base64)
     - Nombre del archivo
     - Hash de contenido (para verificación de integridad)
     - Marca de tiempo de creación
2. **Solicitudes Posteriores** - Cuando se solicita nuevamente la misma factura:
   - El sistema verifica si hay un PDF en caché por invoice_id
   - Si la caché existe y es válida, devuelve el PDF en caché de inmediato
   - Actualiza la marca de tiempo `last_accessed` para rastrear el uso de la caché
3. **Invalidación de Caché** - Los PDFs en caché se invalidan cuando:
   - La factura se modifica (transacciones añadidas/eliminadas, detalles cambiados)
   - La plantilla de factura se actualiza
   - Se activa la limpieza manual de caché

Beneficios:

- **Rendimiento** - Entrega instantánea de PDF para solicitudes repetidas (sin demora de regeneración)

- **Consistencia** - El mismo PDF para todas las descargas de una factura (a menos que la factura se modifique)
- **Carga del Servidor** - Reduce el uso de CPU por la generación de PDF
- **Experiencia del Usuario** - Un indicador de carga aparece durante la generación inicial, las solicitudes posteriores son instantáneas

Gestión de Caché:

La Caché de PDF de Factura es gestionada automáticamente por el sistema. Las entradas de caché antiguas o no utilizadas pueden ser purgadas periódicamente en función de:

- Edad (por ejemplo, eliminar entradas de caché de más de 90 días)
- Patrones de acceso (eliminar entradas no accedidas en 30 días)
- Límites de almacenamiento (implementar límites de tamaño de caché si es necesario)

Comportamiento de la API:

Al descargar una factura a través de la API o UI:

- Primera solicitud: Muestra un indicador de carga mientras se genera el PDF, luego se almacena en caché
- Solicitudes posteriores: Descarga inmediata desde la caché
- El acierto/fallo de caché es transparente para el usuario

Importante: Cuando actualices tu plantilla de factura, limpia la caché para asegurar que las nuevas facturas utilicen el diseño actualizado:

```
-- Limpiar todos los PDFs de factura en caché (ejecutar en MySQL)
DELETE FROM Invoice_PDF_Cache;
```

O actualiza `crm_config.yaml` para invalidar automáticamente la caché al cambiar la plantilla.

Accediendo a Facturas

Las facturas se pueden ver a nivel del sistema o por cliente:

Vista por Cliente:

1. Navega a **Clientes → [Seleccionar Cliente]**
2. Haz clic en la pestaña **Facturación**
3. Ver lista de facturas en la tercera tarjeta

Vista a Nivel del Sistema:

1. Navega a **Facturación → Facturas** (desde el menú principal)
2. Ver todas las facturas de todos los clientes

Widgets de Estadísticas de Facturas

En la parte superior de la página de facturas, cuatro tarjetas de estadísticas muestran resúmenes financieros.

Descripciones de Widgets:

- **Total de Facturas** - Suma de todos los costos minoristas de facturas (todo el tiempo) y conteo de facturas enviadas
- **Facturas No Pagadas** - Suma de facturas aún no pagadas y conteo de facturas no pagadas
- **Facturas Este Mes** - Suma de facturas creadas este mes calendario con conteo
- **Facturas El Mes Pasado** - Suma de facturas creadas el mes pasado con conteo

Formato de Valores:

- Valores superiores a 1,000: Se muestran como sufijo "k" (por ejemplo, $1.5k)
- Valores superiores a 1,000,000: Se muestran como sufijo "M" (por ejemplo, $2.3M)
- Valores superiores a 1,000,000,000: Se muestran como sufijo "B" (por ejemplo, $1.1B)

Indicadores de Tendencia:

- Widgets para "Este Mes" y "El Mes Pasado" muestran cambio porcentual
- Flecha verde hacia arriba: Aumento respecto al período anterior
- Flecha roja hacia abajo: Disminución respecto al período anterior
- Flecha gris hacia la derecha: Sin cambio

Lista de Facturas

La tabla de facturas muestra todas las facturas con las siguientes columnas:

Descripciones de Columnas:

- **ID** - ID de factura único
- **Título** - Título/descripción de la factura
- **Período** - Período de facturación (fecha de inicio - fecha de finalización) o "N/A" para facturas únicas
- **Fecha de Vencimiento** - Fecha de vencimiento del pago
- **Creado** - Fecha de creación de la factura
- **Monto** - Monto total de la factura (costo minorista)
- **Estado** - Pagado, No Pagado o Reembolsado
- **Acciones** - Acciones disponibles (varía según el estado)

Iconos de Acción:

- ⬇ **(Descargar)** - Descargar PDF de factura
- □ **(Eliminar)** - Anular factura (solo si no está pagada)
- □ **(Pagar)** - Pagar factura en línea (solo si no está pagada)
- ✉ **(Correo)** - Enviar correo electrónico de factura al cliente
- □ **(Reembolso)** - Reembolsar pago de Stripe (solo para facturas de Stripe pagadas)

Generando una Factura

Haz clic en **"+ Generar Factura Proforma"** para crear una nueva factura.

Descripciones de Campos:

- **Buscar Clientes** - Seleccionar cliente (solo se muestra en vista a nivel del sistema, prellenado en vista de cliente)
- **Título** - Título/nombre de la factura (opcional, predeterminado a "Factura para [Período]")
- **Fecha de Inicio** - Inicio del período de facturación (predeterminado a 14 días atrás)
- **Fecha de Finalización** - Fin del período de facturación (predeterminado a hoy)
- **Fecha de Vencimiento** - Fecha límite de pago (predeterminado a hoy)
- **Vista Previa de Transacciones** - Muestra todas las transacciones no facturadas en el rango de fechas con la capacidad de incluir/excluir transacciones específicas

Selección de Transacciones:

- ✓ **(Más Verde)** - Haz clic para excluir una transacción de la factura
- × **(X Roja)** - Haz clic para incluir una transacción previamente excluida
- **Seleccionar Todo** - Incluir todas las transacciones mostradas

- **Limpiar Todo** - Excluir todas las transacciones
- Las transacciones excluidas aparecen atenuadas con texto tachado
- Los totales se actualizan en tiempo real a medida que seleccionas/deseleccionas transacciones

Lo Que Ocurre:

1. El sistema encuentra todas las transacciones no facturadas para el cliente dentro del rango de fechas
2. Muestra la vista previa de transacciones con la capacidad de incluir/excluir transacciones individuales
3. Muestra el cálculo en tiempo real de subtotal, impuestos y total basado en las transacciones seleccionadas
4. Solo las transacciones seleccionadas (incluidas) se añaden a la factura
5. Genera el PDF de la factura y lo almacena en caché
6. Marca las transacciones seleccionadas como facturadas (`invoice_id` campo poblado)
7. Las transacciones excluidas permanecen no facturadas y disponibles para futuras facturas
8. La factura aparece en la lista con estado "No Pagada"

Ejemplos de Casos de Uso:

Facturación Mensual: Establecer la fecha de inicio al primer día del mes, la fecha de finalización al último día del mes, la vista previa muestra todas las transacciones no facturadas de ese período. Seleccionar todo o excluir manualmente transacciones específicas.

Factura Específica de Servicio: Usar el mismo rango de fechas, luego excluir manualmente transacciones no deseadas (por ejemplo, excluir transacciones no móviles para crear una factura solo para móviles).

Factura Única: Establecer tanto la fecha de inicio como la fecha de finalización en el mismo día, la vista previa muestra solo transacciones de esa fecha. Excluir cualquier cargo no relevante para esta factura específica.

Visualizando Detalles de la Factura

Haz clic en cualquier fila de factura en la tabla para ver los detalles completos de la factura, incluyendo todas las transacciones, totales y acciones disponibles.

Modal de Detalles de la Factura:

- **Información de la Factura** - Muestra el ID de la factura, título, fechas, estado de pago y estado de anulación
- **Lista de Transacciones** - Muestra todas las transacciones incluidas en la factura con:
  - Fecha de transacción
  - Título y descripción
  - Costo minorista
  - Monto de impuestos y porcentaje (formateado como `$10.00 (10%)`)
  - Las transacciones exentas de impuestos muestran "-" en la columna de impuestos
- **Resumen de Totales** - Cálculo en tiempo real que muestra:
  - Conteo de transacciones
  - Subtotal (suma de todos los costos minoristas)
  - Impuesto (suma de todos los montos de impuestos)
  - Total de la Factura (subtotal + impuestos)

- **Botones de Acción** - Las mismas acciones disponibles que en la tabla:
  - **Descargar PDF** - Descargar PDF de la factura (siempre disponible)
  - **Enviar Correo** - Enviar factura al cliente (facturas no anuladas)
  - **Pagar Factura** - Procesar pago (facturas no pagadas, no anuladas)
  - **Reembolsar** - Reembolsar pago de Stripe (solo facturas de Stripe pagadas)
  - **Eliminar** - Anular factura (facturas no pagadas, no anuladas)

Descargando PDFs de Facturas

Haz clic en el **icono de descarga (⬇)** en la tabla o en el botón **"Descargar PDF"** en el modal de detalles de la factura para descargar una factura como PDF.

Proceso de Descarga:

1. Haz clic en el icono de descarga junto a la factura
2. Un spinner de carga aparece durante la generación (solo la primera vez)
3. El navegador solicita guardar el archivo: `Invoice_01234.pdf`
4. El PDF se abre o se guarda en la carpeta de Descargas

Comportamiento de Caché de PDF:

- **Primera Descarga** - PDF generado a partir de la plantilla, almacenado en la base de datos (puede tardar 2-3 segundos)
- **Descargas Posteriores** - Descarga instantánea desde la caché
- **Invalidación de Caché** - Caché borrada si la factura se modifica o la plantilla se actualiza

Solucionando Problemas de Descarga:

- **El spinner nunca se detiene** - Verifica la consola del navegador, la API puede estar caída
- **PDF en blanco o dañado** - Verifica la plantilla de la factura en busca de errores de sintaxis

- **La descarga falla** - Verifica la configuración del bloqueador de ventanas emergentes, prueba con otro navegador

Pagando Facturas

Haz clic en el **icono de pago (□)** para pagar una factura en línea.

Proceso de Pago:

1. Haz clic en el icono de pago en la factura no pagada
2. Se abre el modal de pago mostrando los detalles de la factura
3. Selecciona el método de pago:
    - **Transacción de Stripe** - Cargar tarjeta de crédito guardada (disponible para todos los usuarios)
    - **Efectivo** - Pago manual en efectivo (solo personal)
    - **Reembolso** - Aplicar reembolso como pago (solo personal)
    - **Transacción de POS** - Terminal de punto de venta (solo personal)
    - **Transferencia Bancaria** - Transferencia bancaria manual (solo personal)
4. Si se selecciona Stripe:
    - Selecciona tarjeta de los métodos de pago guardados
    - La tarjeta predeterminada se preselecciona
    - Haz clic para seleccionar una tarjeta diferente

5. Si se selecciona otro método:
    - Ingresa el número de referencia (opcional)
6. Haz clic en **"Pagar Factura"** para procesar
7. El sistema procesa el pago:
    - **Stripe** - Carga la tarjeta a través de la API de Stripe
    - **Otros métodos** - Crea una transacción negativa por el monto del pago
8. El estado de la factura cambia a "Pagada"
9. Se muestra una notificación de éxito

Auto-Cuidado vs Pago del Personal:

:doc:`Portal de Auto-Cuidado <self_care_portal>` (Clientes):

- Solo disponible el pago de Stripe
- Debe tener un método de pago guardado
- Se muestra una advertencia si no existen métodos de pago
- Se proporciona un enlace para agregar un método de pago

Portal del Personal (Administradores):

- Todos los métodos de pago disponibles
- Puede marcar la factura como pagada manualmente (efectivo, POS, transferencia bancaria)
- Puede ingresar números de referencia para seguimiento

Advertencia de Método de Pago:

Si el cliente no tiene métodos de pago guardados, se muestra una advertencia sugiriendo que agregue un método de pago antes de poder pagar facturas.

Enviando Facturas por Correo Electrónico

Haz clic en el **icono de correo (✉)** para enviar la factura al cliente.

Lo Que Ocurre:

1. Haz clic en el icono de correo junto a la factura
2. El sistema recupera el PDF de la factura de la caché (o genera si no está en caché)
3. Envía el correo electrónico a través de Mailjet usando la plantilla `api_crmCommunicationCustomerInvoice`
4. El correo electrónico incluye:
   - PDF de la factura como archivo adjunto
   - Nombre del cliente
   - Número de factura y fecha de vencimiento
   - Monto total a pagar
   - Enlace para pagar la factura en línea
   - Enlace para ver/descargar la factura
5. Notificación de éxito: "Correo electrónico de factura enviado con éxito"

Destinatarios del Correo Electrónico:

El correo se envía a todos los contactos del cliente con tipo "facturación" o contacto principal si no existe contacto de facturación.

Variables de Plantilla de Correo Electrónico:

- `{{ var:customer_name }}` - Nombre completo del cliente
- `{{ var:invoice_number }}` - ID de la factura
- `{{ var:invoice_date }}` - Fecha de emisión de la factura
- `{{ var:due_date }}` - Fecha de vencimiento del pago
- `{{ var:total_amount }}` - Monto total a pagar
- `{{ var:invoice_url }}` - Enlace para ver/descargar PDF
- `{{ var:pay_url }}` - Enlace para pagar la factura en línea

Solucionando Problemas de Correo Electrónico:

- **Correo no enviado** - Verifica las credenciales de la API de Mailjet en `crm_config.yaml`
- **Cliente no recibe** - Verifica las direcciones de correo electrónico de contacto del cliente
- **PDF no adjunto** - Verifica que la generación del PDF haya tenido éxito (intenta descargar primero)

Anulando Facturas

Haz clic en el **icono de eliminar (🗑)** para anular una factura.

Requisitos:

- La factura debe estar **No Pagada**
- Las facturas pagadas no pueden ser anuladas (deben ser reembolsadas en su lugar)

Cómo Anular:

1. Localiza la factura no pagada en la lista
2. Haz clic en el icono de eliminar (🗑)
3. Confirma en el modal:

Lo Que Ocurre:

- La factura se marca como `void = true`
- Todas las transacciones se desvinculan de la factura (`invoice_id` se establece en nulo)
- Las transacciones se convierten nuevamente en "no facturadas"
- Las transacciones pueden incluirse en una nueva factura
- La factura aparece en la lista con el prefijo "Anulada:" en el título
- Las acciones de la factura se deshabilitan (sin descarga, pago o correo)
- Se puede ver filtrando por facturas "Anuladas"

Notas Importantes:

- Anular NO es lo mismo que reembolsar
- **Anular** = "Esta factura nunca debió existir" (error de facturación, duplicado)
- **Reembolsar** = "Revertir una factura válida pagada" (devolver dinero al cliente)

Reembolsando Facturas

Haz clic en el **icono de reembolso (□)** para reembolsar una factura pagada.

Requisitos:

- La factura debe estar **Pagada**
- La factura debe haber sido pagada a través de **Stripe**
- La factura debe tener una `payment_reference` válida (ID de intención de pago de Stripe)
- Solo disponible para usuarios del personal (no Auto-Cuidado)

Cómo Reembolsar:

1. Localiza la factura de Stripe pagada
2. Haz clic en el icono de reembolso (□)
3. Se abre el modal de confirmación de reembolso:

4. Haz clic en **"Confirmar Reembolso"**
5. El sistema procesa el reembolso de Stripe:
   - Llama a la API de Stripe para reembolsar el pago
   - Crea una transacción de reembolso en Stripe
   - Actualiza la factura con `refund_reference`
6. El estado de la factura cambia a "Reembolsada"
7. Se muestra una notificación de éxito

Lo Que Ocurre Después del Reembolso:

- La factura permanece en el sistema (no anulada)
- El estado muestra "Reembolsada"
- Las transacciones permanecen vinculadas a la factura

- El cliente recibe el reembolso al método de pago original (3-7 días hábiles)
- El panel de Stripe muestra la transacción de reembolso

Restricciones de Reembolso:

- No se pueden reembolsar facturas pagadas en efectivo, POS o transferencia bancaria (se requiere reversión manual)
- No se puede reembolsar parcialmente (solo el monto total de la factura)
- No se puede reembolsar dos veces

Buscando y Filtrando Facturas

Búsqueda

Usa la barra de búsqueda para encontrar facturas. Busca en:

- ID de factura
- Título de factura
- Nombre del cliente (solo vista a nivel del sistema)

Filtros

Aplica filtros para reducir la lista de facturas:

Filtros Disponibles:

- **Estado de Anulación** - Todos, Anulado, No Anulado
- **Estado Pagado** - Todos, Pagado, No Pagado

Acciones de Filtro:

- **Aplicar Filtros** - Aplicar filtros seleccionados a la lista
- **Restablecer Filtros** - Limpiar todos los filtros y mostrar todas las facturas

Ordenación

Haz clic en cualquier encabezado de columna para ordenar:

- **ID** - Ordenar por ID de factura (más nuevo/más antiguo)
- **Título** - Ordenar alfabéticamente
- **Fecha de Vencimiento** - Ordenar por fecha de vencimiento
- **Creado** - Ordenar por fecha de creación
- **Monto** - Ordenar por costo minorista (más alto/más bajo)
- **Estado** - Ordenar por estado pagado (pagado primero o no pagado primero)

Haz clic nuevamente para invertir la dirección de ordenación (ascendente ↔ descendente).

Paginación

Navega a través de listas grandes de facturas con controles de página que muestran la página actual, el total de páginas y un selector de elementos por página (10, 25, 50 o 100 elementos).

Flujos de Trabajo Comunes de Facturas

Flujo de Trabajo 1: Facturación Mensual con Vista Previa de Transacciones

1. Llega el fin de mes (por ejemplo, 31 de enero)
2. Navega a **Facturación → Facturas**
3. Haz clic en **"+ Generar Factura Proforma"**
4. Selecciona cliente (o hazlo por cliente si facturas a muchos clientes)
5. Establece fechas:
    - Fecha de Inicio: 2025-01-01
    - Fecha de Finalización: 2025-01-31

    - Fecha de Vencimiento: 2025-02-15 (15 días a partir de ahora)
    - Título: "Servicios de Enero 2025" (opcional)
6. La sección de **Vista Previa de Transacciones** aparece mostrando todas las transacciones no facturadas de enero
7. Revisa la vista previa:
    - Todas las transacciones están incluidas por defecto
    - Verifica totales: Subtotal, Impuesto y Total de la Factura
    - Verifica que todos los cargos sean correctos
8. Haz clic en **"Generar Factura"** (el botón muestra el conteo de transacciones, por ejemplo, "Generar Factura (15)")
9. Factura creada con todas las transacciones seleccionadas
10. Haz clic en la fila de la factura para ver detalles y verificar
11. Haz clic en el botón **"Enviar Correo"** en el modal de detalles o en el icono de correo en la tabla
12. El cliente recibe un correo electrónico de factura con PDF y enlace de pago

Flujo de Trabajo 2: Facturación Selectiva de Transacciones

1. El cliente tiene múltiples servicios (Móvil + Internet) y cargos diversos
2. Quiere facturas separadas para cada servicio
3. **Generar la primera factura (Servicios Móviles):**
    - Haz clic en **"+ Generar Factura Proforma"**
    - Título: "Servicios Móviles - Enero 2025"
    - Inicio/Fin: 1-31 de enero
    - Fecha de Vencimiento: 15 de febrero
    - En la vista previa de transacciones, **excluir** todas las transacciones no móviles:
        - Haz clic en el botón **X** junto a las transacciones de Internet
        - Haz clic en el botón **X** junto a los cargos diversos
        - Solo permanecen seleccionadas las transacciones de servicios móviles
    - Verifica que los totales reflejen solo los servicios móviles

    - Haz clic en **"Generar Factura"** (muestra el conteo de transacciones móviles)
4. **Generar la segunda factura (Servicios de Internet):**
    - Haz clic en **"+ Generar Factura Proforma"** nuevamente
    - Título: "Servicios de Internet - Enero 2025"
    - Inicio/Fin: 1-31 de enero (mismo período)
    - En la vista previa de transacciones:
        - Las transacciones móviles ya facturadas (no aparecen)
        - Excluye cargos diversos usando el botón **X**
        - Solo permanecen las transacciones de servicios de Internet
    - Haz clic en **"Generar Factura"**
5. **Generar la tercera factura (Cargos Adicionales):**
    - Haz clic en **"+ Generar Factura Proforma"** nuevamente
    - Título: "Cargos Adicionales - Enero 2025"
    - Solo aparecen en la vista previa los cargos diversos no facturados
    - Haz clic en **"Seleccionar Todo"** para incluir todos
    - Haz clic en **"Generar Factura"**
6. Envía las tres facturas al cliente

Flujo de Trabajo 3: Excluyendo Transacciones Disputadas o Pendientes

1. Llega el fin del período de facturación
2. Navega a la pestaña de **Facturación** del cliente
3. Haz clic en **"+ Generar Factura Proforma"**
4. Establece las fechas del período de facturación
5. La vista previa de transacciones muestra 20 transacciones
6. El cliente ha disputado un cargo y otro está pendiente de investigación
7. En la vista previa de transacciones:
    - Localiza la transacción disputada (por ejemplo, "Cargo por exceso de datos")
    - Haz clic en el botón **X** para excluirla
    - Localiza la transacción pendiente (por ejemplo, "Cargo de instalación")
    - Haz clic en el botón **X** para excluirla

    - El conteo de transacciones se actualiza: "18 Transacciones seleccionadas"
    - Los totales se recalculan automáticamente
8. Revisa los totales actualizados (excluye los montos disputados)
9. Haz clic en **"Generar Factura (18)"**
10. Factura generada con solo transacciones aprobadas
11. Las transacciones disputadas/pendientes permanecen no facturadas para el próximo ciclo de facturación

Flujo de Trabajo 4: Revisión Rápida de Factura y Ajuste

1. El personal genera la factura mensual
2. La vista previa de transacciones muestra un total inesperadamente alto
3. Revisa cada transacción en la vista previa:
    - Nota un cargo duplicado por el mismo servicio
    - Haz clic en **X** para excluir el duplicado
    - Nota una transacción de prueba que no debería ser facturada
    - Haz clic en **X** para excluir la transacción de prueba
4. Los totales se actualizan en tiempo real
5. Verifica que el nuevo total coincida con el monto esperado
6. Haz clic en **"Generar Factura"** con las transacciones corregidas
7. Vuelve y anula/elimina las transacciones excluidas si es necesario
8. Envía la factura al cliente con confianza

Flujo de Trabajo 5: Factura de Instalación Única

1. El técnico de campo completa la instalación
2. El personal añade manualmente la transacción de instalación
3. Navega a la pestaña de **Facturación** del cliente
4. Haz clic en **"+ Generar Factura Proforma"**
5. Establece fechas:
    - Fecha de Inicio: hoy

    - Fecha de Finalización: hoy
    - Fecha de Vencimiento: hoy + 7 días
    - Título: "Servicios de Instalación"
6. La vista previa de transacciones muestra solo las transacciones de hoy
7. Verifica que el cargo de instalación aparezca
8. Excluye cualquier cargo recurrente usando el botón **X** (si está presente)
9. Haz clic en **"Generar Factura"**
10. Envía inmediatamente al cliente
11. El cliente paga en línea a través de Stripe

Flujo de Trabajo 6: Revisando la Factura Antes del Contacto con el Cliente

1. El cliente llama con una pregunta de facturación
2. El personal navega a la lista de facturas del cliente
3. **Haz clic en la fila de la factura** para abrir el modal de Detalles de la Factura
4. Revisa la información de la factura:
    - ID de factura, fechas, estado
    - Todas las transacciones incluidas con descripciones
    - Desglose de impuestos por transacción
    - Subtotal, Impuesto y Montos Totales
5. Responde a las preguntas del cliente con detalles exactos
6. Si el cliente solicita el PDF, haz clic en el botón **"Descargar PDF"** en el modal
7. Si el cliente solicita reenvío de correo, haz clic en el botón **"Enviar Correo"**
8. Cierra el modal cuando termines

Flujo de Trabajo 7: Corrigiendo un Error de Facturación

1. El cliente informa un cargo incorrecto
2. El personal hace clic en la fila de la factura para ver detalles
3. Revisa la lista de transacciones en el modal de Detalles de la Factura

4. Identifica la transacción incorrecta
5. La factura está no pagada, por lo que puede ser anulada
6. Haz clic en el botón **"Eliminar"** en el pie del modal
7. Confirma la anulación
8. Las transacciones se convierten nuevamente en no facturadas
9. El personal modifica o elimina la transacción incorrecta de la lista de transacciones
10. Genera una nueva factura con las transacciones corregidas:
    - Usa la vista previa de transacciones para excluir la transacción corregida si es necesario
    - Incluye solo cargos válidos
11. Envía la factura corregida al cliente

Flujo de Trabajo 8: Procesando Múltiples Pagos

1. El cliente trae efectivo para pagar múltiples facturas
2. Navega a la pestaña de **Facturación** del cliente
3. Ver facturas no pagadas
4. Haz clic en la primera fila de factura para ver detalles
5. Verifica monto y transacciones
6. Haz clic en el botón **"Pagar Factura"** en el pie del modal
7. Selecciona el método de pago **"Efectivo"**
8. Ingresa referencia: "Efectivo pagado 2025-01-15"
9. Haz clic en **"Pagar Factura"**
10. El modal se cierra, la factura se marca como "Pagada"
11. Repite para las facturas restantes
12. Todas las facturas ahora marcadas como "Pagadas"

Flujo de Trabajo 9: Manejo de Solicitud de Reembolso

1. El cliente solicita un reembolso por sobrepago
2. El personal verifica que la factura fue pagada a través de Stripe
3. Navega a la factura en la lista

4. Haz clic en la fila de la factura para ver detalles
5. Verifica la información de pago y el monto
6. Haz clic en el botón **"Reembolsar"** en el pie del modal (solo aparece para facturas de Stripe)
7. Confirma el reembolso
8. El sistema procesa el reembolso de Stripe
9. El estado de la factura cambia a "Reembolsada"
10. El cliente recibe el reembolso en 3-7 días hábiles
11. El personal sigue con el cliente para confirmar la recepción

Solucionando Problemas

No se puede generar factura - No se encontraron transacciones

- **Causa:** No hay transacciones no facturadas en el rango de fechas especificado
- **Solución:** Verifica la lista de transacciones, verifica que existan transacciones y que no estén ya facturadas. Ajusta el rango de fechas o elimina el filtro.

La generación de PDF de factura falla

- **Causa:** Error de sintaxis en la plantilla, fallo de WeasyPrint o datos de cliente faltantes
- **Solución:** Verifica el HTML de la plantilla de factura en busca de errores, verifica que los campos de dirección del cliente estén poblados, revisa los registros de la API.

El pago falla con error de Stripe

- **Causa:** Tarjeta rechazada, fondos insuficientes, tarjeta expirada o problema con la API de Stripe
- **Solución:** Prueba con un método de pago diferente, verifica que la tarjeta sea válida, revisa el panel de Stripe para el motivo del rechazo.

No se puede anular la factura

- **Causa:** La factura ya está pagada
- **Solución:** Las facturas pagadas no pueden ser anuladas. Si se necesita un reembolso, utiliza la función de reembolso para facturas de Stripe o crea una transacción de crédito manualmente.

El correo electrónico de la factura no se envía

- **Causa:** Credenciales de API de Mailjet no válidas, el cliente no tiene contacto de facturación o falta la plantilla de correo electrónico
- **Solución:** Verifica la configuración de Mailjet en `crm_config.yaml`, verifica los contactos del cliente, verifica que exista la plantilla de correo electrónico de la factura.

El botón de reembolso no aparece

- **Causa:** Factura pagada en efectivo/POS/transferencia bancaria (no Stripe), o factura no pagada
- **Solución:** El botón de reembolso solo aparece para pagos de Stripe. Para otros métodos de pago, crea una transacción de crédito manual.

La descarga del PDF muestra el diseño antiguo de la plantilla

- **Causa:** PDF almacenado en caché antes de la actualización de la plantilla
- **Solución:** Limpia la caché de PDF de la factura: `DELETE FROM Invoice_PDF_Cache WHERE invoice_id = X;`

El cliente no puede pagar la factura (sin métodos de pago)

- **Causa:** No hay métodos de pago guardados en el portal de Auto-Cuidado
- **Solución:** El cliente debe agregar una tarjeta de crédito en la página de **Métodos de Pago** antes de pagar facturas.

Se generaron múltiples facturas para el mismo período

- **Causa:** El personal generó la factura dos veces, o los rangos de fechas se superponen
- **Solución:** Anula la factura duplicada. Ajusta los rangos de fechas para evitar superposiciones. Usa la vista previa de transacciones para asegurar conjuntos de transacciones únicos.

La vista previa de transacciones no muestra transacciones

- **Causa:** Todas las transacciones en el rango de fechas ya están facturadas o no existen transacciones
- **Solución:** Verifica que el rango de fechas sea correcto. Verifica la lista de transacciones para confirmar que existan transacciones. Filtra las facturas para ver qué factura contiene las transacciones.

No se puede excluir la transacción de la generación de la factura

- **Causa:** Transacción ya facturada o problema del navegador
- **Solución:** Verifica que la transacción se muestre en la vista previa con una marca de verificación. Actualiza la página y vuelve a intentarlo. Limpia la caché del navegador si el problema persiste.

El total de la factura no coincide con el monto esperado

- **Causa:** Transacciones inesperadas incluidas, impuestos no calculados o transacciones excluidas aún contadas
- **Solución:** Revisa la vista previa de transacciones cuidadosamente. Verifica cada costo minorista y el impuesto de la transacción. Verifica que las transacciones excluidas estén atenuadas. Verifica el distintivo de conteo de transacciones en el botón Generar Factura.

El botón Generar Factura está deshabilitado

- **Causa:** No se seleccionó ninguna transacción o rango de fechas no válido
- **Solución:** Asegúrate de que al menos una transacción esté incluida (no excluida). Verifica que la Fecha de Inicio sea anterior a la Fecha de Finalización. Verifica que la Fecha de Vencimiento esté establecida.

El modal de Detalles de la Factura no se abre

- **Causa:** Error de JavaScript o página no completamente cargada
- **Solución:** Actualiza la página. Verifica la consola del navegador en busca de errores. Prueba con otro navegador. Verifica la conexión a Internet.

El impuesto de la transacción no se muestra en los Detalles de la Factura

- **Causa:** La transacción tiene un impuesto del 0% o tax_amount es nulo
- **Solución:** Verifica que la transacción tenga tax_percentage establecido. Verifica que tax_amount se haya calculado cuando se creó la transacción. Actualiza la transacción si es necesario.

Faltan botones de acción en el modal de Detalles de la Factura

- **Causa:** La factura está anulada o el usuario no tiene permisos
- **Solución:** Las facturas anuladas solo muestran el botón Descargar PDF. Verifica el estado de la factura. Verifica el rol y los permisos del usuario.

Documentación Relacionada

- `integrations_mailjet` - Entrega de correo electrónico de factura y plantillas
- `administration_configuration` - Configuración de plantilla de factura
- `payments_transaction` - Creación de transacciones que aparecen en facturas
- `payments_process` - Procesamiento de pagos de facturas
- `basics_payment` - Gestión de métodos de pago
- `payment_system_guide` - Referencia de API de pagos y configuración de proveedores

Procesar Pagos

El sistema procesa la mayoría de los pagos automáticamente. A veces, debes procesar un pago manualmente.

Para pagar una factura, selecciona la factura no pagada y haz clic en el botón "Pagar".

Esto abrirá un formulario de pago, donde puedes ingresar el método de pago y hacer clic en "Enviar" para procesar el pago.

El cliente recibirá automáticamente un recibo por el pago, y la factura será marcada como pagada.

Para transferencias bancarias, ingresa la referencia de pago. También ingresa la fecha en que se realizó el pago, si es diferente de la fecha actual.

Transacciones de Clientes

Cualquier cosa que cueste dinero en el sistema se registra como una transacción bajo el cliente.

Cada transacción tiene un monto monetario para el costo mayorista y el costo minorista, y una descripción de para qué es la transacción.

Las transacciones pueden ser generadas automáticamente por el sistema, por ejemplo, cuando se provisiona un servicio, se crea una transacción para el costo de configuración, y cuando se factura un servicio, se crea una transacción para el costo minorista.

Las transacciones también pueden ser creadas manualmente, por ejemplo, si se le otorga un crédito a un cliente, se crea una transacción por el monto del crédito, o si se cobra una tarifa de instalación, se crea una transacción por la tarifa de instalación.

Las transacciones se agrupan para formar Facturas, que se envían al cliente para su pago.

Accediendo a las Transacciones

Las transacciones se pueden ver a nivel de sistema o por cliente:

Vista por Cliente:

1. Navegar a **Clientes → [Seleccionar Cliente]**
2. Hacer clic en la pestaña **Facturación**
3. Ver la lista de transacciones en la primera tarjeta

Vista a Nivel del Sistema:

1. Navegar a **Facturación → Transacciones** (desde el menú principal)
2. Ver todas las transacciones de todos los clientes

Widgets de Estadísticas de Transacciones

En la parte superior de la página de transacciones, cuatro tarjetas de estadísticas muestran resúmenes financieros:

Descripciones de Widgets:

- **Total de Transacciones** - Suma de todos los costos minoristas de las transacciones (todo el tiempo)
- **Total de Transacciones No Facturadas** - Suma de transacciones que aún no se han incluido en una factura
- **Total de Transacciones Este Mes** - Suma de transacciones creadas este mes calendario
- **Total de Transacciones El Mes Pasado** - Suma de transacciones creadas el mes pasado

Formato de Valores:

- Valores superiores a 1,000: Mostrar como sufijo "k" (por ejemplo, $1.5k)
- Valores superiores a 1,000,000: Mostrar como sufijo "M" (por ejemplo, $2.3M)
- Valores superiores a 1,000,000,000: Mostrar como sufijo "B" (por ejemplo, $1.1B)

Lista de Transacciones

La tabla de transacciones muestra todas las transacciones con las siguientes columnas:

Descripciones de Columnas:

- **ID** - ID único de la transacción
- **Fecha** - Fecha de creación de la transacción
- **Título** - Nombre corto de la transacción
- **Descripción** - Descripción detallada de para qué es la transacción
- **Monto** - Costo minorista (positivo para cargos, negativo para créditos)
- **Factura** - ID de la factura si la transacción ha sido facturada (enlace clicable)
- **Estado** - Marca de verificación si está facturada, guion si aún no está facturada

Acciones por Fila:

Cada fila tiene un menú de acciones (⋮) con opciones:

- **Ver Detalles** - Abre el modal de detalles de la transacción
- **Descargar PDF de Factura** - Descargar PDF (solo si está facturada)
- **Anular Transacción** - Marcar la transacción como anulada (solo si no está facturada)

Tipos de Transacciones

Las transacciones se dividen en dos categorías principales:

Transacciones de Débito (Cargos)

Montos positivos que aumentan el saldo adeudado por el cliente:

- **Tarifas de Configuración de Servicio** - Cargos únicos cuando se provisiona el servicio
- **Tarifas Mensuales de Servicio** - Cargos recurrentes por servicios
- **Tarifas de Instalación** - Cargos por visitas de técnicos de campo
- **Cargos por Equipos** - Cargos por módems, enrutadores, tarjetas SIM
- **Cargos por Pago Tardío** - Penalizaciones por facturas vencidas
- **Cargos Manuales** - Cargos personalizados añadidos por el personal

Transacciones de Crédito (Pagos/Reembolsos)

Montos negativos que disminuyen el saldo adeudado por el cliente:

- **Pagos en Efectivo** - Cliente pagó en efectivo
- **Pagos con Tarjeta** - Cliente pagó con tarjeta de crédito/débito
- **Pagos por Transferencia Bancaria** - Cliente pagó mediante transferencia bancaria
- **Créditos de Cuenta** - Créditos de buena voluntad, compensación
- **Reembolsos** - Dinero devuelto al cliente
- **Descuentos** - Descuentos promocionales o de lealtad

Agregar una Transacción Manualmente

Haga clic en **"+ Agregar Transacción"** para abrir el modal de agregar transacción.

Transacción de Débito (Cargo):

Transacción de Crédito (Pago/Reembolso):

Descripciones de Campos:

- **Tipo de Transacción** - Seleccionar Débito (cargo) o Crédito (pago/reembolso)
- **Tipo de Crédito** - Si se selecciona Crédito, elegir método de pago (Efectivo, Tarjeta, Transferencia Bancaria)
- **Título** - Nombre corto para la transacción (requerido)
- **Descripción** - Explicación detallada (opcional)
- **Costo Minorista** - Monto que paga el cliente (requerido, número positivo)
- **Costo Mayorista** - Su costo (opcional, para seguimiento de márgenes)
- **Porcentaje de Impuesto** - Tasa de impuesto aplicada a esta transacción (opcional, por defecto es el impuesto del producto o 0%)
- **Servicio** - Vincular la transacción a un servicio específico (opcional)
- **Sitio** - Vincular la transacción a un sitio específico (opcional)
- **Fecha de Transacción** - Fecha de la transacción (por defecto es hoy)

Validación:

- El título y el costo minorista son obligatorios
- El costo minorista debe ser un número positivo
- Si se selecciona el tipo de Crédito, debe elegirse un tipo de crédito

Qué Ocurre:

1. Transacción creada en la base de datos
2. Aparece en la lista de transacciones del cliente
3. Incluida en el conteo de "Transacciones No Facturadas"
4. Disponible para inclusión en la próxima generación de facturas
5. Entrada en el registro de actividad creada

Buscar y Filtrar Transacciones

Buscar

Utilice la barra de búsqueda para encontrar transacciones. Busca en:

- ID de transacción

- Título
- Descripción
- ID de factura

Filtros

Aplique filtros para reducir la lista de transacciones:

Filtros Disponibles:

- **Estado de Anulación** - Todos, Anulado, No Anulado
- **Estado de Factura** - Todos, Facturado, No Facturado

Acciones de Filtro:

- **Aplicar Filtros** - Aplicar filtros seleccionados a la lista
- **Restablecer Filtros** - Limpiar todos los filtros y mostrar todas las transacciones

Ordenar

Haga clic en cualquier encabezado de columna para ordenar:

- **ID** - Ordenar por ID de transacción (más nuevo/más antiguo)
- **Fecha** - Ordenar por fecha de transacción
- **Título** - Ordenar alfabéticamente
- **Monto** - Ordenar por costo minorista (más alto/más bajo)
- **Factura** - Ordenar por ID de factura

Haga clic nuevamente para invertir la dirección de ordenación (ascendente ↔ descendente).

Anular Transacciones

Las transacciones añadidas por error pueden ser **anuladas** (marcadas como eliminadas).

Requisitos:

- La transacción NO debe estar facturada
- Una vez facturadas, las transacciones no pueden ser anuladas (deben ser reembolsadas en su lugar)

Cómo Anular:

1. Localizar la transacción en la lista
2. Hacer clic en el menú de acciones (⋮)
3. Seleccionar **"Anular Transacción"**
4. Confirmar en el modal

Qué Ocurre:

- Transacción marcada como `void = true`
- Ya no aparece en la lista de transacciones por defecto
- Excluida de la generación de facturas
- Puede ser vista filtrando por transacciones "Anuladas"
- Deducida del total de "Transacciones No Facturadas"

Transacciones vinculadas a la billetera (`wallet_ledger_id` establecido):

- El saldo de la billetera se revierte antes de anular (débito para deshacer un crédito, crédito para deshacer un débito)

- **Recarga de tarjeta de fondos** (`payment_type=wallet_topup_card`): se debita la billetera y se reembolsa la tarjeta
- **Recarga en efectivo / pago de factura de billetera / otras líneas de billetera:** solo se ajusta la billetera; **sin reembolso de tarjeta**
- **Billetera insuficiente:** la anulación falla si los fondos recargados ya fueron gastados (la transacción permanece activa)

Cargos de servicio / provisión de captura híbrida generalmente no tienen **ningún** `wallet_ledger_id` en la línea de facturación (la tarjeta → billetera → débito ocurre internamente). Anular esas líneas solo establece `void=true` hoy; se planea un crédito completo de billetera para las anulaciones de servicio en la rama de documentos financieros.

API: `DELETE /crm/transaction/transaction_id/{transaction_id}`. Ver guía_del_sistema_de_pago.md: Anular Transacción.

Nota: Anular NO es lo mismo que reembolsar. Anular significa "esta transacción nunca debió haber existido". Reembolsar significa "revertir una transacción válida".

Impuesto sobre Transacciones

Las transacciones pueden incluir impuestos, que se calculan automáticamente según la configuración de impuestos del producto o se especifican manualmente por transacción.

Comportamiento del Impuesto:

- **Transacciones de Débito (Cargos)** - El impuesto se aplica a los cargos según:
  - **Porcentaje de Impuesto del Producto** - Si la transacción está vinculada a un producto, se aplica automáticamente el porcentaje de impuesto del producto
  - **Anulación Manual** - El personal puede anular el porcentaje de impuesto al crear una transacción
  - **Monto del Impuesto** - Calculado como: `retail_cost × (tax_percentage / 100)`
  - **Formato de Visualización** - Mostrado como: `$10.00 (10%)` en listas de transacciones
- **Transacciones de Crédito (Pagos/Reembolsos)** - No se aplica impuesto a los créditos
  - El campo de porcentaje de impuesto está oculto para las transacciones de crédito
  - El impuesto se establece automáticamente en 0% para todos los pagos y reembolsos
  - Los créditos reducen el saldo pendiente del cliente sin implicaciones fiscales

Ejemplo de Cálculo de Impuestos:

- Producto: Plan Móvil con 10% de impuesto, $50.00 costo minorista
- Cálculo Automático de Impuestos: $50.00 × 0.10 = $5.00
- Visualización: $5.00 (10%)

Impuesto Cero (NIL/Exento):

- Los productos pueden estar exentos de impuestos estableciendo el porcentaje de impuesto en 0
- El impuesto por defecto es 0% si no se especifica
- Las transacciones exentas de impuestos muestran "-" en la columna de Impuesto

Vista de Detalles de Transacción

Haga clic en una transacción para ver todos los detalles:

Transacciones Facturadas vs No Facturadas

Transacciones No Facturadas:

- Aún no incluidas en ninguna factura

- Disponibles para la próxima generación de facturas
- Pueden ser anuladas
- Cuentan para el total de "Transacciones No Facturadas"
- El estado muestra un guion (-)

Transacciones Facturadas:

- Incluidas en una factura
- No pueden ser anuladas (deben reembolsarse si es necesario)
- ID de factura clicable (enlaces a detalles de la factura)
- El estado muestra una marca de verificación (✓)
- No pueden ser modificadas

Generación de Factura:

Cuando genera una factura para un cliente:

1. El sistema encuentra todas las transacciones no facturadas para ese cliente
2. Opcionalmente filtrar por rango de fechas
3. Transacciones incluidas en la nueva factura
4. Campo `invoice_id` de la transacción poblado
5. Transacción ahora marcada como "facturada"

Consulte `payments_invoices` para obtener detalles sobre la generación de facturas.

Flujos de Trabajo Comunes

Flujo de Trabajo 1: Crédito Manual por Interrupción del Servicio

1. El cliente llama: "El servicio estuvo inactivo durante 2 días"
2. El personal decide acreditar £10
3. Navegar a la pestaña **Facturación** del cliente

4. Hacer clic en **"+ Agregar Transacción"**
5. Seleccionar tipo de transacción **Crédito**
6. Seleccionar tipo de crédito **Pago en Efectivo**
7. Ingresar título: "Crédito por Interrupción del Servicio"
8. Ingresar descripción: "Compensación por interrupción de 2 días 8-9 de enero"
9. Ingresar costo minorista: 10.00
10. Seleccionar servicio afectado del menú desplegable
11. Hacer clic en **"Agregar Transacción"**
12. La transacción aparece con un monto de -£10.00
13. Se incluirá en la próxima factura como crédito

Flujo de Trabajo 2: Tarifa de Instalación Manual

1. El técnico de campo instala el servicio
2. El personal necesita cobrar una tarifa de instalación de £75
3. Navegar a la pestaña **Facturación** del cliente
4. Hacer clic en **"+ Agregar Transacción"**
5. Seleccionar tipo de transacción **Débito**
6. Ingresar título: "Tarifa de Instalación"
7. Ingresar descripción: "Visita del técnico de campo para instalación de fibra"
8. Ingresar costo minorista: 75.00
9. Ingresar costo mayorista: 45.00 (opcional, para seguimiento de márgenes)
10. Seleccionar servicio instalado
11. Seleccionar sitio donde se instaló
12. Hacer clic en **"Agregar Transacción"**
13. La transacción aparece en la lista no facturada
14. Se incluirá en la próxima factura

Flujo de Trabajo 3: Anulando Transacción Duplicada

1. El personal nota una transacción duplicada
2. Verificar que la transacción NO esté facturada aún

3. Hacer clic en el menú de acciones (⋮) en la transacción duplicada
4. Seleccionar **"Anular Transacción"**
5. Confirmar en el modal
6. La transacción se elimina de la lista
7. El total no facturado disminuye en consecuencia

Flujo de Trabajo 4: Encontrar Transacciones para Factura

1. Necesita generar factura mensual
2. Hacer clic en **Filtro de Factura: "No Facturado"**
3. Hacer clic en **Aplicar Filtros**
4. Ver todas las transacciones no facturadas
5. Anotar el monto total de los widgets
6. Navegar para generar factura
7. Seleccionar rango de fechas (por ejemplo, 1-31 de enero)
8. Transacciones en el rango incluidas en la factura

Solución de Problemas

No se puede anular la transacción

- **Causa:** La transacción ya está facturada
- **Solución:** La transacción es parte del historial de facturas. Si se necesita un reembolso, crear una transacción de Crédito en su lugar.
- **Causa:** Falló la reversión de recarga de billetera (saldo insuficiente)
- **Solución:** El cliente ya gastó los fondos recargados. Utilice un débito/ajuste manual de billetera o flujos de reversión de documentos financieros; no se puede anular la línea de recarga hasta que la billetera cubra el monto.
- **Causa:** Falló el reembolso de tarjeta (solo recarga de fondos)

- **Solución:** La reversión de billetera se revierte automáticamente. Resuelva el problema del proveedor de pagos y vuelva a intentar anular.

Aparecen transacciones duplicadas

- **Causa:** Servicio cobrado múltiples veces o error de provisión
- **Solución:** Anule la(s) transacción(es) duplicada(s) si no están facturadas. Si están facturadas, emita un crédito.

Transacción no aparece en la lista

- **Causa:** Filtros aplicados o transacción anulada
- **Solución:** Haga clic en "Restablecer Filtros" para mostrar todas las transacciones. Para ver transacciones anuladas, filtre por "Anular: Anulado".

El total no facturado no coincide con lo esperado

- **Causa:** Algunas transacciones ya están facturadas, o las transacciones anuladas están excluidas
- **Solución:** Aplique el filtro "Factura: No Facturada" para ver solo las no facturadas. Verifique las transacciones anuladas por separado.

No se puede agregar transacción (campo de cliente deshabilitado)

- **Causa:** Visualizando la página de transacciones específicas del cliente
- **Solución:** El cliente está preseleccionado. Si necesita agregar una transacción para un cliente diferente, vaya a la página de Transacciones a Nivel del Sistema.

Documentación Relacionada

- `payments_invoices` - Generación y gestión de facturas
- `payments_process` - Procesamiento de pagos contra facturas
- `basics_payment` - Resumen de métodos de pago
- `csa_activity_log` - Visualización del historial de transacciones en el registro de actividad

Cargos y Pagos desde Playbooks

Esta guía explica cómo implementar el procesamiento de cargos y pagos dentro de los playbooks de Ansible para flujos de trabajo de aprovisionamiento de OmniCRM.

> **Referencia Completa de API**: Para obtener detalles completos sobre las API de pagos, enrutamiento de billeteras, reembolsos y arquitectura independiente del proveedor, consulte la **Guía de API del Sistema de Pagos**

Visión General

Los Playbooks de Ansible de OmniCRM pueden manejar el procesamiento de pagos de varias maneras diferentes, pero todas ellas simplemente realizan llamadas a la API en el CRM para cobrar.

1. **Flujo de Pago de Compromiso en Dos Fases** - Para servicios pagos que requieren autorización y captura de pago inmediatas (es decir, Servicios Prepagados)
2. **Creación de Transacciones Directas** - Para agregar cargos/créditos sin procesamiento de pago inmediato (por ejemplo, tarifas de configuración, créditos manuales) que pueden ser procesados más tarde (es decir, servicios Postpad)

Estos enfoques aseguran transacciones atómicas donde los clientes solo son cobrados si el aprovisionamiento tiene éxito, con retroceso automático si algún paso falla.

Flexibilidad de Precios: Playbooks como Motor de Scripting

Importante: Los valores en dólares definidos en productos y servicios (por ejemplo, `retail_cost`, `wholesale_cost`, `retail_setup_cost`) son simplemente **valores predeterminados** almacenados en la base de datos. No dictan lo que debes cobrar. El playbook tiene control total sobre el precio final.

El playbook de Ansible es esencialmente un **motor de scripting** que puedes adaptar para cumplir con cualquier requisito de lógica empresarial. Puedes:

- **Usar los precios almacenados tal como están** - Simplemente referencia `api_response_product.json.retail_cost` en tu autorización
- **Sobrescribir los precios completamente** - Cobrar una cantidad diferente independientemente de lo que esté en la definición del producto
- **Aplicar descuentos dinámicos** - Calcular porcentajes de descuento basados en el nivel del cliente, promociones o lealtad
- **Implementar precios escalonados** - Cobrar diferentes tarifas según la cantidad, niveles de uso o términos del contrato
- **Precios agrupados** - Combinar múltiples productos con descuentos personalizados por paquete
- **Precios basados en el tiempo** - Ajustar precios según la hora del día, la temporada o períodos promocionales
- **Precios específicos para el cliente** - Consultar tarifas negociadas de contratos de clientes

Ejemplo: Sobrescribiendo el Precio del Producto

```
# El retail_cost del producto es $99.00, pero queremos cobrar
$79.00 por una promoción
- name: Establecer precio promocional (ignorando retail_cost del
producto)
  set_fact:
    charge_amount: 79.00

- name: Autorizar pago promocional
  uri:
    url: "http://localhost:5000/crm/payments/authorize/hold"
    method: POST
    body_format: json
    body:
      customer_id: "{{ customer_id | int }}"
      amount: "{{ charge_amount | float }}"  # Usando nuestra
sobrescritura, no retail_cost
      # ...
```

Ejemplo: Descuento por Nivel de Cliente

```
# Aplicar descuento basado en el nivel del cliente
- name: Obtener nivel del cliente
  uri:
    url: "http://localhost:5000/crm/customer/{{ customer_id }}"
    # ...
  register: customer_info

- name: Calcular descuento por nivel
  set_fact:
    base_price: "{{ api_response_product.json.retail_cost | float
}}"
    discount_percent: >-
      {% if customer_info.json.tier == 'gold' %}20
      {% elif customer_info.json.tier == 'silver' %}10
      {% else %}0{% endif %}

- name: Aplicar descuento al precio final
  set_fact:
    final_price: "{{ (base_price | float) * (1 - (discount_percent
| float / 100)) | round(2) }}"
```

Ejemplo: Precios para Socios Mayoristas

```
# Los socios mayoristas pagan wholesale_cost en lugar de
retail_cost
- name: Determinar precio basado en el tipo de cliente
  set_fact:
    charge_amount: >-
      {% if customer_info.json.is_wholesale_partner %}
      {{ api_response_product.json.wholesale_cost | float }}
      {% else %}
      {{ api_response_product.json.retail_cost | float }}
      {% endif %}
```

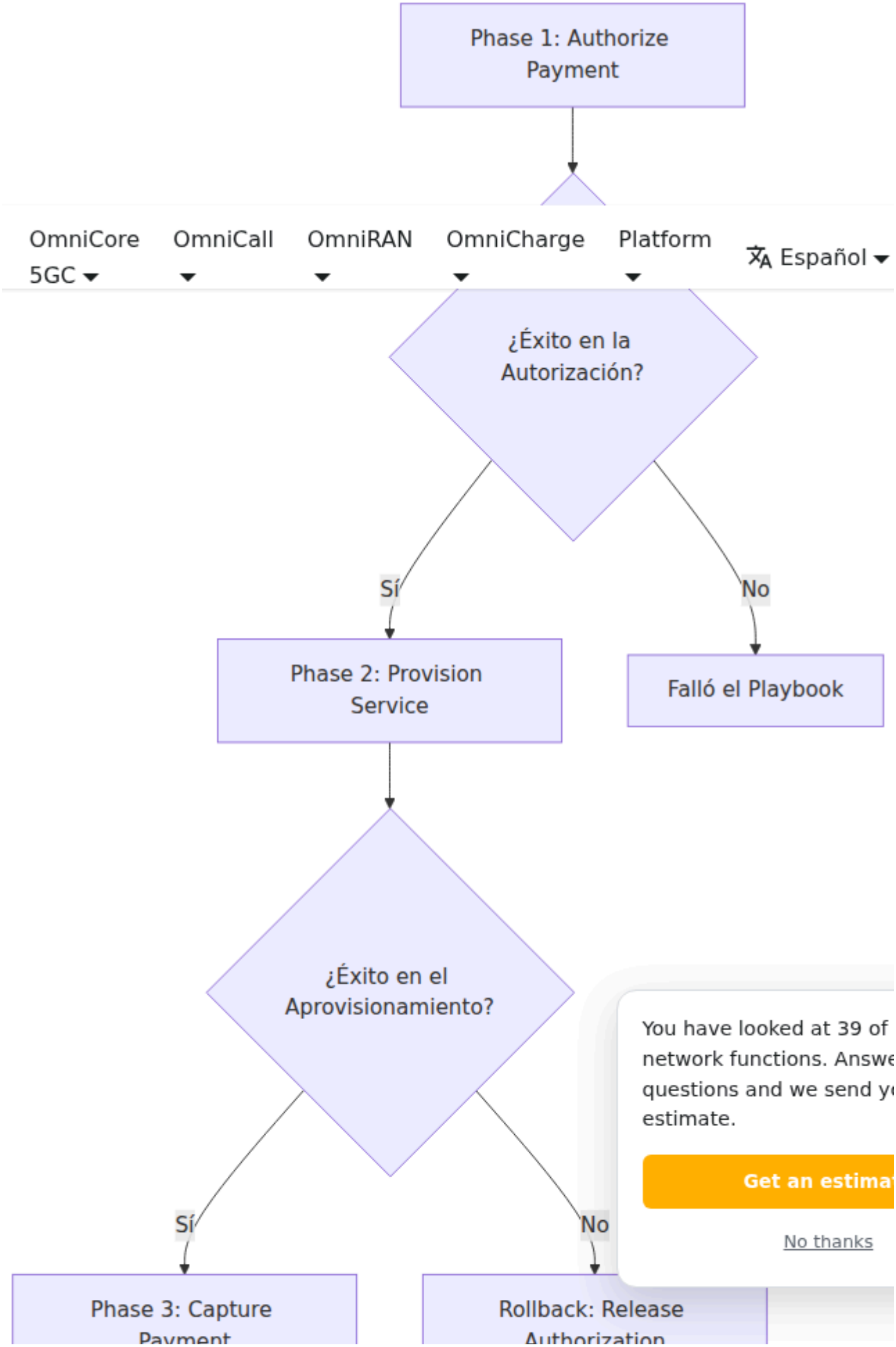
La conclusión clave es que **no estás atado a ningún modelo de precios**. El playbook te da control programático total para implementar cualquier regla comercial que requiera tu organización. Los precios de productos/servicios en el

CRM son solo valores predeterminados convenientes que los playbooks pueden usar o ignorar según sea necesario.

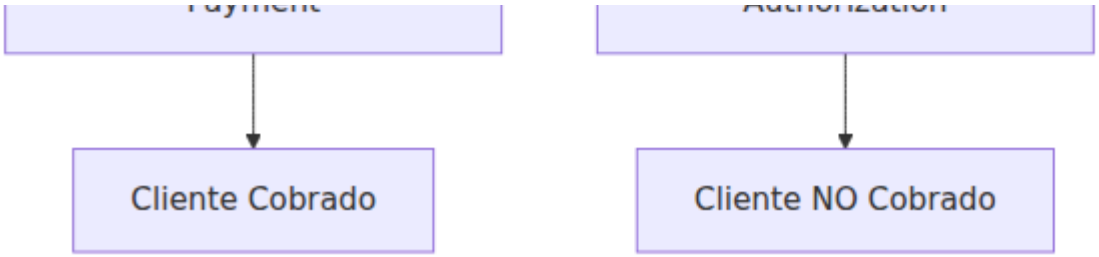
Flujo de Pago de Compromiso en Dos Fases

El patrón de compromiso en dos fases se utiliza cuando necesitas cobrar un método de pago del cliente durante el aprovisionamiento. Esto asegura que el cliente solo sea cobrado si el aprovisionamiento se completa con éxito.

Visión General del Flujo



OmniCore 5GC
OmniCall
OmniRAN
OmniCharge
Platform
Español
You have looked at 39 of
network functions. Answe
questions and we send y
estimate.
Get an estima
No thanks



Patrón de Implementación

El patrón sigue estas fases:

Fase 1: Autorización - Retener fondos en el método de pago
Fase 2: Aprovisionamiento - Ejecutar actualizaciones de saldo/acción de OCS
Fase 3: Captura - Finalizar el pago tras un aprovisionamiento exitoso
Rollback - Liberar la autorización si el aprovisionamiento falla

Ejemplo Completo

Aquí hay un ejemplo completo de `play_topup_charge_then_action.yaml`:

```yaml
- name: Jugar Topup - Cargar tarjeta y luego realizar la acción
del Topup
  hosts: localhost
  gather_facts: no
  become: False

  tasks:
    - name: Incluir vars de crm_config
      ansible.builtin.include_vars:
        file: "../../crm_config.yaml"
        name: crm_config

    # Obtener información del producto y servicio
    - name: Obtener información del Producto desde la API de CRM
      uri:
        url: "http://localhost:5000/crm/product/product_id//{{
product_id }}"
        method: GET
        headers:
          Authorization: "Bearer {{ access_token }}"
        return_content: yes
        validate_certs: no
      register: api_response_product

    - name: Obtener información del Servicio desde la API de CRM
      uri:
        url: "http://localhost:5000/crm/service/service_id/{{
service_id }}"
        method: GET
        headers:
          Authorization: "Bearer {{ access_token }}"
        return_content: yes
        validate_certs: no
      register: api_response_service

    - name: Establecer hechos de servicio y paquete
      set_fact:
        service_uuid: "{{ api_response_service.json.service_uuid
}}"
        customer_id: "{{ api_response_service.json.customer_id }}"
        package_name: "{{ api_response_product.json.product_name
}}"
        monthly_cost: "{{ api_response_product.json.retail_cost
```

```
}}"
        wholesale_cost: "{{
api_response_product.json.wholesale_cost }}"

    # Obtener el método de pago predeterminado del cliente
    - name: Obtener los Métodos de Pago del Cliente desde el
Controlador de Pagos
      uri:
        url: "http://localhost:5000/crm/payments/methods?
customer_id={{ customer_id }}"
        method: GET
        headers:
          Authorization: "Bearer {{ access_token }}"
        return_content: yes
        validate_certs: no
      register: api_response_payment_methods

    - name: Obtener el payment_method_id predeterminado de la
respuesta
      set_fact:
        payment_method_id: "{{ api_response_payment_methods.json |
json_query(query) }}"
      vars:
        query: "data[?is_default==`true`].payment_method_id | [0]"

    # ============================================================
    # FLUJO DE PAGO DE COMPROMISO EN DOS FASES
    # ============================================================

    - name: "Fase 1: Autorizar pago (retener fondos)"
      uri:
        url: "http://localhost:5000/crm/payments/authorize/hold"
        method: POST
        headers:
          Content-Type: "application/json"
          Authorization: "Bearer {{ access_token }}"
        body_format: json
        body:
          {
            "customer_id": "{{ customer_id | int }}",
            "amount": "{{ monthly_cost | float }}",
            "currency": "{{ crm_config.currency | default('AUD')
}}",
            "payment_method_id": "{{ payment_method_id | int }}",
```

```
            "metadata": {
              "description": "{{ package_name }} en {{
api_response_service.json.service_name }}",
              "service_id": "{{
api_response_service.json.service_id | int }}",
              "site_id": "{{ api_response_service.json.site_id |
int }}",
              "product_id": "{{ product_id }}",
              "user_id": "{{ (initiating_user | int) if
(initiating_user is defined and initiating_user is not none) else
omit }}",
              "title": "{{ package_name }}",
              "wholesale_cost": "{{ wholesale_cost | float }}",
              "invoice": true,
              "contract_days": "{{ days | int }}",
              "send_email": true
            }
          }
        return_content: yes
      register: api_response_authorization

    - name: Afirmar que la autorización fue exitosa
      assert:
        that:
          - api_response_authorization.status == 200
          - api_response_authorization.json.success == true

    - name: Almacenar authorization_id para captura/liberación
      set_fact:
        authorization_id: "{{
api_response_authorization.json.data.authorization_id }}"

    # Fase 2: Aprovisionamiento de OCS (envuelto en bloque para
retroceso transaccional)
    - block:
        - name: "Fase 2: Ejecutar acción de CGRateS"
          uri:
            url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
            method: POST
            body_format: json
            headers:
              Content-Type: "application/json"
              Authorization: "Bearer {{ access_token }}"
            body:
```

```
              {
                "id": "{{ 999999999 | random }}",
                "method": "APIerSv1.ExecuteAction",
                "params": [{
                  "Tenant": "{{ crm_config.ocs.ocsTenant }}",
                  "Account": "{{ service_uuid }}",
                  "ActionsId": "{{ cgr_action_name }}"
                }]
              }
            status_code: 200
          register: action_execute_response

        - name: Afirmar que la respuesta de la acción es OK
          assert:
            that:
              - action_execute_response.status == 200
              - action_execute_response.json.result == "OK"

        # Fase 3: Captura de Pago - Finalizar transacción después
de un aprovisionamiento exitoso
        - name: "Fase 3: Capturar pago autorizado"
          uri:
            url: "http://localhost:5000/crm/payments/capture/{{
authorization_id }}"
            method: POST
            headers:
              Content-Type: "application/json"
              Authorization: "Bearer {{ access_token }}"
            body_format: json
            body:
              {
                "metadata": {
                  "provisioning_status": "success",
                  "cgr_action": "{{ cgr_action_name }}"
                }
              }
            return_content: yes
          register: api_response_capture

        - name: Afirmar que la captura fue exitosa
          assert:
            that:
              - api_response_capture.status == 200
              - api_response_capture.json.success == true
```

```
      rescue:
        # Retroceso de Transacción: Anular autorización para
liberar fondos retenidos
        - name: "Rollback: Liberar autorización de pago"
          uri:
            url: "http://localhost:5000/crm/payments/release/{{
authorization_id }}"
            method: POST
            headers:
              Content-Type: "application/json"
              Authorization: "Bearer {{ access_token }}"
            body_format: json
            body:
              {
                "metadata": {
                  "release_reason": "provisioning_failed",
                  "cgr_action": "{{ cgr_action_name }}"
                }
              }
            return_content: yes
          register: api_response_release

        - name: Terminar la ejecución del playbook después del
retroceso
          fail:
            msg: "El aprovisionamiento de OCS falló. Autorización
de pago {{ authorization_id }} anulada. Cliente no cobrado."
```

Puntos de API de Pago

> **Enrutamiento de Billetera Primero**: El sistema de pago optimiza automáticamente los cargos a la tarjeta utilizando primero el saldo de la billetera. Si un cliente tiene $1 en su billetera y compra un complemento de $10, solo se cargan $9 a su tarjeta. Consulte la sección de Enrutamiento de Billetera Primero para más detalles.

1. Autorizar/Reterner Pago

Endpoint: `POST /crm/payments/authorize/hold`

Este endpoint coloca una retención en el método de pago del cliente por la cantidad especificada. Los fondos están reservados pero aún no capturados.

Comportamiento de Billetera Primero: La autorización calcula automáticamente el déficit después de verificar el saldo de la billetera. Si el saldo de la billetera es de $150 y la autorización es por $500, solo se autorizan $350 en la tarjeta. La billetera se debita en el momento de la captura.

Cuerpo de la Solicitud:

```
{
  "customer_id": 123,
  "amount": 49.99,
  "currency": "AUD",
  "payment_method_id": 456,
  "metadata": {
    "description": "Nombre y descripción del paquete",
    "service_id": 789,
    "site_id": 12,
    "product_id": "34",
    "user_id": 5,
    "title": "Nombre del Paquete",
    "wholesale_cost": 25.00,
    "invoice": true,
    "contract_days": 30,
    "send_email": true
  }
}
```

Descripciones de Campos:

- **customer_id** (requerido) - ID del cliente que se está cobrando
- **amount** (requerido) - Monto a autorizar en formato decimal
- **currency** (opcional) - Código de moneda (predeterminado al sistema)
- **payment_method_id** (requerido) - Método de pago a cobrar
- **metadata** (opcional) - Datos adicionales para la creación de la transacción:
  - **description** - Descripción de la transacción
  - **service_id** - Servicio por el que se cobra
  - **site_id** - Sitio asociado con el servicio

- **product_id** - Producto que se está aprovisionando
- **user_id** - Usuario que inicia el cargo (opcional)
- **title** - Título de la transacción
- **wholesale_cost** - Tu costo (para seguimiento de márgenes)
- **invoice** - Si es `true`, crear automáticamente la transacción cuando se capture
- **contract_days** - Duración del contrato
- **send_email** - Si es `true`, enviar notificación por correo electrónico

Respuesta:

```
{
  "success": true,
  "message": "Pago autorizado (retención creada)",
  "data": {
    "authorization_id": 301,
    "vendor_authorization_id": "auth_xxxxx",
    "amount": 500.00,
    "currency": "USD",
    "status": "authorized",
    "wallet_balance": 150.00,
    "wallet_to_use": 150.00,
    "card_amount": 350.00,
    "message": "Tarjeta autorizada por $350 (recarga de
billetera). Se realizará un débito de billetera de $500 en la
captura."
  }
}
```

Importante:

- Guarda el `authorization_id` para usar en llamadas de captura o liberación
- Ten en cuenta que el `card_amount` muestra que solo se autorizó el déficit
- El saldo de la billetera se verifica pero NO se debita hasta la captura

2. Capturar Pago

Endpoint: `POST /crm/payments/capture/{authorization_id}`

Este endpoint finaliza la autorización de pago y cobra al cliente. Si se estableció `invoice: true` en los metadatos de autorización, se crea automáticamente una transacción.

Cuerpo de la Solicitud:

```
{
  "metadata": {
    "provisioning_status": "success",
    "cgr_action": "Action_Topup_Standard",
    "additional_info": "Cualquier otro dato relevante"
  }
}
```

Respuesta:

```
{
  "success": true,
  "data": {
    "payment_id": "pay_xyz789",
    "transaction_id": 1234
  }
}
```

Qué Sucede:

1. **Tarjeta capturada** por el monto del déficit (si la tarjeta fue autorizada)
2. **Billetera acreditada** con el monto capturado de la tarjeta
3. **Billetera debitada** por el monto total del servicio
4. Si `invoice: true` estaba en los metadatos de autorización:
   - **Transacción de débito** creada (monto positivo = cargo)
   - **Factura** creada y vinculada a la transacción de débito
   - **Transacción de crédito** creada (monto negativo = pago recibido)
   - Factura marcada como PAGADA (débito y crédito netos a cero)
   - La transacción aparece en la facturación del cliente
5. Se crea un registro de pago en la base de datos
6. Si `send_email: true`, el cliente recibe un correo electrónico de factura

Ejemplo: Autorización de $500 con saldo de billetera de $150:

- Captura de tarjeta: $350
- Billetera acreditada: +$350 (billetera ahora $500)
- Billetera debitada: -$500 (cargo por servicio)
- Billetera final: $0

Consulte Ejemplos de Enrutamiento de Billetera Primero para un flujo detallado.

3. Liberar Autorización de Pago

Endpoint: `POST /crm/payments/release/{authorization_id}`

Este endpoint cancela una autorización de pago y libera los fondos retenidos. Utiliza esto en escenarios de rescate/retroceso cuando el aprovisionamiento falla.

Cuerpo de la Solicitud:

```
{
  "metadata": {
    "release_reason": "provisioning_failed",
    "error_details": "La creación de la cuenta OCS falló"
  }
}
```

Respuesta:

```
{
  "success": true,
  "message": "Autorización liberada"
}
```

Qué Sucede:

1. **Autorización de tarjeta liberada** (si la tarjeta fue autorizada)
2. Fondos retenidos devueltos al crédito disponible del cliente
3. **Billetera NO debitada** (ya que el débito solo ocurre en la captura)

4. El cliente NO es cobrado
5. No se crea ninguna transacción

Nota: Con el enrutamiento de billetera primero, no se necesita un reembolso de billetera ya que la billetera no se debita hasta el momento de la captura.

Creación Directa de Transacciones

Para cargos que no requieren procesamiento de pago (tarifas de configuración, créditos manuales, ajustes), puedes crear transacciones directamente a través de la API.

Agregar una Transacción a través de la API

Endpoint: `PUT /crm/transaction/`

Ejemplo de `play_simple_service.yaml`:

```
- name: Agregar Transacción de Costo de Configuración a través de
la API
  uri:
    url: "http://localhost:5000/crm/transaction/"
    method: PUT
    headers:
      Content-Type: "application/json"
      Authorization: "Bearer {{ access_token }}"
    body_format: json
    body:
      {
        "customer_id": {{ customer_id | int }},
        "service_id": {{ service_creation_response.json.service_id
| int }},
        "title": "{{ package_name }} - Costos de Configuración",
        "description": "Costos de Configuración para {{
package_comment }}",
        "invoice_id": null,
        "wholesale_cost": {{
api_response_product.json.wholesale_setup_cost | float }},
        "retail_cost": "{{
api_response_product.json.retail_setup_cost | float }}"
      }
    return_content: yes
  register: api_response_transaction

- name: Afirmar que la respuesta de la transacción es OK
  assert:
    that:
      - api_response_transaction.status == 200
```

Campos del Cuerpo de Solicitud:

- **customer_id** (requerido) - ID del cliente
- **service_id** (opcional) - ID del servicio para vincular la transacción
- **title** (requerido) - Nombre corto de la transacción
- **description** (opcional) - Descripción detallada
- **invoice_id** (opcional) - ID de factura si ya se ha facturado (generalmente nulo)
- **wholesale_cost** (opcional) - Tu costo

- **retail_cost** (requerido) - Costo para el cliente
- **site_id** (opcional) - ID de sitio para vincular la transacción
- **tax_percentage** (opcional) - Porcentaje de tasa impositiva

Casos de Uso:

- Tarifas de configuración durante el aprovisionamiento del servicio
- Cargos de instalación
- Créditos o ajustes manuales
- Cargos por equipo
- Tarifas administrativas

Nota: La creación directa de transacciones NO procesa el pago; solo crea un registro de facturación. La transacción aparecerá como no facturada y puede incluirse en futuras facturas.

Cálculo de Cargos Prorrateados

El cobro prorrateado te permite cobrar a los clientes proporcionalmente según períodos de facturación parciales. Esto es común cuando:

- El cliente se inscribe a mitad de mes
- El servicio se actualiza/reduce a mitad del ciclo
- La facturación se calcula en función de los días utilizados

Fórmula de Cálculo Prorrateado

```
pro_rata_charge = (monthly_cost × days_remaining) / days_in_month
```

Ejemplo de Implementación

Aquí se muestra cómo calcular un cargo prorrateado en un playbook:

```yaml
# Calcular cargo prorrateado para mes parcial
# Si el cliente se inscribe el día 15 y la facturación es el día
1,
# cobrar proporcionalmente por los días restantes

- name: Obtener el día actual del mes
  command: "date +%d"
  register: current_day

- name: Obtener el total de días en el mes actual
  command: "date -d 'last day of this month' +%d"
  register: days_in_month

- name: Obtener el último día del mes
  command: "date -d 'last day of this month' +%Y-%m-%d"
  register: last_day_of_month

- name: Calcular días restantes en el mes
  set_fact:
    days_remaining: "{{ (days_in_month.stdout | int) -
(current_day.stdout | int) + 1 }}"

- name: Calcular costo prorrateado
  set_fact:
    pro_rata_cost: "{{ ((monthly_cost | float) * (days_remaining |
float) / (days_in_month.stdout | float)) | round(2) }}"

- name: Mostrar detalles del cálculo
  debug:
    msg:
      - "Costo mensual: ${{ monthly_cost }}"
      - "Días en el mes: {{ days_in_month.stdout }}"
      - "Días restantes: {{ days_remaining }}"
      - "Cargo prorrateado: ${{ pro_rata_cost }}"

# Usar pro_rata_cost en autorización o creación de transacción
- name: "Autorizar pago prorrateado"
  uri:
    url: "http://localhost:5000/crm/payments/authorize/hold"
    method: POST
    headers:
      Content-Type: "application/json"
      Authorization: "Bearer {{ access_token }}"
```

```
    body_format: json
    body:
      {
        "customer_id": "{{ customer_id | int }}",
        "amount": "{{ pro_rata_cost | float }}",
        "currency": "{{ crm_config.currency | default('AUD') }}",
        "payment_method_id": "{{ payment_method_id | int }}",
        "metadata": {
          "title": "{{ package_name }} (Prorrateado {{
days_remaining }} días)",
          "description": "Cargo prorrateado por {{ days_remaining
}}/{{ days_in_month.stdout }} días de {{ package_name }}",
          "service_id": "{{ service_id | int }}",
          "invoice": true
        }
      }
```

Prorrateo desde una Fecha de Inicio Personalizada

Si necesitas calcular el prorrateo desde una fecha de inicio específica hasta una fecha de facturación:

```yaml
- name: Establecer fecha de inicio personalizada y fecha de
facturación
  set_fact:
    service_start_date: "2024-01-15"
    next_billing_date: "2024-02-01"

- name: Calcular días entre fechas
  shell: |
    echo $(( ( $(date -d "{{ next_billing_date }}" +%s) - $(date -
d "{{ service_start_date }}" +%s) ) / 86400 ))
  register: days_until_billing

- name: Obtener días en el período de facturación (generalmente
30)
  set_fact:
    billing_period_days: 30

- name: Calcular costo prorrateado
  set_fact:
    pro_rata_cost: "{{ ((monthly_cost | float) *
(days_until_billing.stdout | float) / (billing_period_days |
float)) | round(2) }}"

- name: Mostrar cálculo
  debug:
    msg:
      - "Fecha de inicio: {{ service_start_date }}"
      - "Próxima facturación: {{ next_billing_date }}"
      - "Días hasta la facturación: {{ days_until_billing.stdout
}}"
      - "Cargo prorrateado: ${{ pro_rata_cost }}"
```

Ejemplos de Escenarios Prorrateados

Escenario 1: Inscripción a Mitad de Mes

- El cliente se inscribe el 15 de enero
- Costo mensual: $60.00
- Días en enero: 31
- Días restantes: 17 (del 15 al 31 inclusive)

- Cargo prorrateado: `$60.00 × 17 ÷ 31 = $32.90`

Escenario 2: Actualización de Servicio

- El cliente actualiza el día 10 del ciclo de facturación
- Plan antiguo: $30/mes
- Nuevo plan: $50/mes
- Días en el ciclo: 30
- Días restantes: 21
- Diferencia: $20/mes
- Cargo prorrateado: `$20.00 × 21 ÷ 30 = $14.00`

Mejores Prácticas

1. Siempre Usa el Patrón de Bloque/Rescate

Envuelve la captura de pagos y el aprovisionamiento en un bloque/rescate para asegurar el retroceso:

```
- block:
    # Tareas de aprovisionamiento
    - name: Aprovisionar servicio
      uri: ...

    # Capturar pago solo después del éxito
    - name: Capturar pago
      uri: ...

  rescue:
    # Liberar autorización si algo falla
    - name: Liberar autorización de pago
      uri: ...

    - name: Fallar ejecución del playbook
      fail:
        msg: "El aprovisionamiento falló, el cliente no fue
cobrado"
```

2. Valida Todas las Respuestas de la API

Siempre afirma que las operaciones críticas tuvieron éxito:

```
- name: Autorizar pago
  uri: ...
  register: api_response_authorization

- name: Afirmar que la autorización fue exitosa
  assert:
    that:
      - api_response_authorization.status == 200
      - api_response_authorization.json.success == true
    fail_msg: "La autorización de pago falló: {{
api_response_authorization.json }}"
```

3. Almacena el ID de Autorización

Siempre guarda el authorization_id para usar en captura/liberación:

```
- name: Almacenar authorization_id para captura/liberación
  set_fact:
    authorization_id: "{{
api_response_authorization.json.data.authorization_id }}"
```

4. Usa Metadatos de Manera Efectiva

Incluye metadatos completos en las solicitudes de autorización:

```
metadata:
  description: "Descripción clara de lo que se está cobrando al
cliente"
  service_id: "{{ service_id | int }}"
  product_id: "{{ product_id }}"
  user_id: "{{ initiating_user | int }}"
  title: "Título corto para la transacción"
  wholesale_cost: "{{ wholesale_cost | float }}"
  invoice: true  # Crear automáticamente la transacción en la
captura
  send_email: true  # Enviar notificación al cliente
```

5. Redondea Valores Monetarios

Siempre redondea los cálculos monetarios a 2 decimales:

```
- name: Calcular costo con redondeo
  set_fact:
    final_cost: "{{ (base_cost | float * multiplier | float) |
round(2) }}"
```

6. Maneja Métodos de Pago Faltantes

Verifica si el cliente tiene un método de pago predeterminado antes de intentar la autorización:

```yaml
- name: Obtener método de pago predeterminado
  set_fact:
    payment_method_id: "{{ api_response_payment_methods.json |
json_query(query) }}"
  vars:
    query: "data[?is_default==`true`].payment_method_id | [0]"

- name: Verificar que el método de pago exista
  assert:
    that:
      - payment_method_id is defined
      - payment_method_id != ""
      - payment_method_id != None
    fail_msg: "No se encontró ningún método de pago predeterminado
para el cliente {{ customer_id }}"
```

Patrones Comunes

Patrón 1: Aprovisionamiento de Servicio Pagado

Para servicios que requieren pago inmediato:

1. Obtener el método de pago del cliente
2. Autorizar el pago
3. Aprovisionar el servicio en OCS/CGRateS
4. Crear el registro del servicio en CRM
5. Capturar el pago
6. En caso de fallo: Liberar la autorización y retroceder

Consulte `play_topup_charge_then_action.yaml` para un ejemplo completo.

Patrón 2: Servicio Gratuito con Costo de Configuración

Para servicios que son gratuitos pero tienen un costo de configuración único:

1. Aprovisionar el servicio
2. Crear el registro del servicio
3. Agregar la transacción de tarifa de configuración directamente (sin procesamiento de pago)
4. La tarifa de configuración aparece en la próxima factura

Consulte `play_simple_service.yaml` en las líneas 202-232 para un ejemplo completo.

Patrón 3: Topup/Addon Gratuito

Para topups gratuitos que no requieren pago:

1. Obtener información del servicio
2. Ejecutar acción de CGRateS
3. Actualizar fechas del servicio
4. No se necesita pago ni creación de transacción

Patrón 4: Cargos Recurrentes a través de ActionPlan

Para cargos recurrentes automáticos:

1. Crear Acción con `*http_post` al endpoint de aprovisionamiento
2. Crear ActionPlan con temporización `*monthly`
3. Asignar ActionPlan a la cuenta
4. CGRateS llamará automáticamente al endpoint mensualmente
5. El playbook del endpoint maneja el procesamiento de pagos

Solución de Problemas

Falla de Autorización

Síntoma: El endpoint de autorización devuelve un error

Causas Comunes:

- El método de pago no existe o es inválido
- Fondos insuficientes
- Método de pago expirado
- Desajuste en el ID del cliente

Solución: Verifica el estado del método de pago y el saldo del cliente.

La Captura Falla Después del Aprovisionamiento Exitoso

Síntoma: Servicio aprovisionado pero la captura de pago falla

Problema: Este es un estado crítico de falla: el servicio está activo pero el cliente no fue cobrado

Solución:

- La autorización puede haber expirado (típicamente 7 días)
- Verifica que la autorizaci��n aún sea válida antes de intentar la captura
- Implementa monitoreo para capturas fallidas
- Puede ser necesaria intervención manual

Transacción No Creada Después de la Captura

Síntoma: Pago capturado pero no hay transacción en la facturación

Causa: `invoice: true` no se estableció en los metadatos de autorización

Solución: O bien:

- Establece `invoice: true` en los metadatos de autorización, O
- Crea manualmente la transacción después de la captura exitosa

Cálculo Prorrateado Incorrecto

Síntoma: Los cargos prorrateados no coinciden con los valores esperados

Problemas Comunes:

- Errores de uno en el conteo de días (incluir/excluir fechas de inicio/fin)
- Mes incorrecto utilizado para el conteo de días
- Errores de redondeo

Solución:

- Usa rangos de fechas inclusivos (incluye tanto el día de inicio como el de fin)
- Siempre redondea a 2 decimales
- Prueba los cálculos con valores conocidos
- Documenta qué fechas se incluyen en los cálculos

Reembolsos y Manejo de Errores

Opciones de Reembolso

El sistema de pagos admite dos tipos de reembolsos:

1. Reembolso al Método de Pago - Dinero devuelto a la tarjeta/PayPal original

```
- name: Reembolsar pago a la tarjeta del cliente
  uri:
    url: "http://localhost:5000/crm/payments/refund"
    method: POST
    headers:
      Content-Type: "application/json"
      Authorization: "Bearer {{ access_token }}"
    body_format: json
    body:
      {
        "transaction_id": "{{ vendor_transaction_id }}",
        "vendor": "stripe",
        "amount": "{{ refund_amount | float }}",
        "reason": "customer_request"
      }
```

2. Crédito a la Billetera - Saldo inmediato para futuras compras (manejado automáticamente para escenarios de error)

Para fallos de aprovisionamiento, el sistema acredita automáticamente la billetera en lugar de reembolsar la tarjeta para:

- Evitar tarifas de reembolso
- Proporcionar disponibilidad inmediata para reintentos
- Mejorar la experiencia del cliente

Consulte Opciones de Reembolso para obtener detalles completos.

Soporte de Proveedores

El sistema de pagos es **independiente del proveedor** y actualmente admite:

- □ Stripe (tarjetas, ACH)
- □ PayPal (cuentas de PayPal, tarjetas)

Se pueden agregar nuevos proveedores de pago (Square, Adyen, Braintree, etc.) sin cambiar los playbooks.

Consulte también:

- Arquitectura Modular - Cómo funciona la abstracción del proveedor
- Configuración del Proveedor - Configuración de Stripe/PayPal y claves de API

Documentación Relacionada

Guías Específicas de Playbook

- `concepts_ansible.md` - Patrones y estructura general de playbooks
- `concepts_provisioning.md` - Visión general del sistema de aprovisionamiento

Documentación del Sistema de Pagos

- **Guía de API del Sistema de Pagos** - Referencia completa de API, enrutamiento de billetera, reembolsos
  - APIs de Métodos de Pago
  - APIs de Flujo de Pago
  - Enrutamiento de Billetera Primero
  - Opciones de Reembolso
  - Arquitectura Modular
- `payments_transaction.md` - Gestión de transacciones y cargos manuales
- `payments_process.md` - Procesamiento de pagos y facturas
- `basics_payment.md` - Métodos de pago y facturación de clientes

Control de Acceso Basado en Reglas

Roles, Permisos y Usuarios en OmniCRM

OmniCRM utiliza **control de acceso basado en roles (RBAC)**: las personas (Usuarios) son asignadas a uno o más Roles, y cada Rol es un conjunto de Permisos. Los Permisos son la unidad más pequeña de acceso (por ejemplo, *view_customer*, *create_inventory*). El acceso efectivo de un usuario es la **unión** de los permisos de todos los roles asignados.

Propósito

RBAC permite:

1. **Protección de Datos**: Los usuarios solo ven y hacen lo que se les permite.
2. **Ajuste Operacional**: Los roles reflejan funciones laborales (Admin, Soporte, Finanzas, Administrador de Clientes).
3. **Administración Simple**: Conceder acceso asignando roles. Evitar la micromanagement por usuario.
4. **Aislamiento de Inquilinos**: Los permisos de “ver propio …” limitan la visibilidad a los datos de cliente/inquilino de un usuario.

Cómo se Relacionan Usuarios, Roles y Permisos

- **Usuarios**: Personas reales que inician sesión en OmniCRM.
- **Permisos**: Capacidades atómicas (por ejemplo, *view_customer*, *delete_product*).
- **Roles**: Conjuntos nombrados de permisos (por ejemplo, *Admin*, *Support*, *Finance*).
- **Asignación**: Los usuarios reciben uno o más roles. Los permisos se agregan.

La autenticación prueba *quién eres* (JWT, clave API o IP en lista blanca). La autorización (roles/permisos) decide *qué puedes hacer*.

Gestión de Usuarios

El sistema de gestión de usuarios de OmniCRM permite a los administradores crear y gestionar usuarios del personal (administradores, agentes de servicio al cliente), ver y modificar roles de usuario, restablecer contraseñas, gestionar la autenticación de dos factores y controlar el acceso de los usuarios.

Tipos de Usuarios

Usuarios Clientes - Creado a través de auto-registro o por administradores. Asignado automáticamente el rol "Cliente". Estos usuarios acceden al portal de auto-cuidado para gestionar sus servicios, ver uso, pagar facturas, etc.

Usuarios del Personal - Creado por administradores con permisos apropiados. Pueden ser asignados roles como Admin, Soporte, Finanzas, etc. Estos usuarios acceden a la interfaz del CRM para gestionar clientes, provisionar servicios, manejar facturación, etc.

Usuarios Administrativos - Usuarios con el permiso `admin`. Tienen acceso completo al sistema, incluida la gestión de usuarios, gestión de roles y todos los puntos finales protegidos.

El usuario administrativo inicial es creado por el equipo de Omnitouch cuando se despliega el sistema.

Agregar Nuevos Usuarios (Administradores y CSAs)

Los administradores pueden crear nuevos usuarios del personal a través de la interfaz web o API.

A través de la Interfaz Web

1. **Navegar a Usuarios y Roles** - Acceder a la interfaz de gestión de usuarios desde el menú de administración.
2. **Hacer clic en "Agregar Usuario"** - Abre el formulario de creación de usuario.

3. **Completar los Detalles del Usuario:**
    - **Nombre de Usuario** - Nombre de usuario único para iniciar sesión (requerido).
    - **Correo Electrónico** - Dirección de correo electrónico del usuario (requerido, debe ser único).

    - **Contraseña** - Contraseña temporal (requerido, el usuario debe cambiarla en el primer inicio de sesión).
    - **Nombre** - Primer nombre del usuario (requerido).
    - **Segundo Nombre** - Segundo nombre del usuario (opcional).
    - **Apellido** - Apellido del usuario (requerido).
    - **Número de Teléfono** - Número de teléfono de contacto (opcional).
    - **Rol** - Seleccionar uno o más roles para asignar (requerido).
    - **Contacto del Cliente** - Opcionalmente vincular usuario a un registro de contacto de cliente (para usuarios clientes).
4. **Hacer clic en "Crear Usuario"** - El usuario es creado y puede iniciar sesión inmediatamente con las credenciales proporcionadas.
5. **El usuario recibe notificación** - Opcionalmente enviar correo electrónico de bienvenida con instrucciones de inicio de sesión.

Mejores Prácticas:

- Utilizar una contraseña temporal como `TempP@ssw0rd!` y requerir que el usuario la cambie en el primer inicio de sesión.
- Asignar roles apropiados según la función laboral (ver Diseños de Roles Típicos a continuación).
- Habilitar 2FA para todo el personal administrativo y de soporte.
- Vincular usuarios clientes a su registro de contacto de cliente para un correcto alcance de datos.

A través de la API

Crear un usuario programáticamente:

Endpoint: `POST /auth/users`

Permiso Requerido: `admin`

Cuerpo de la Solicitud:

```
{
  "username": "john.smith",
  "email": "john.smith@company.com",
  "password": "TempP@ssw0rd!",
  "first_name": "John",
  "middle_name": "D",
  "last_name": "Smith",
  "phone_number": "+61412345678",
  "role": "Support"
}
```

Respuesta:

```
{
  "id": 123,
  "username": "john.smith",
  "email": "john.smith@company.com",
  "first_name": "John",
  "last_name": "Smith",
  "roles": ["Support"],
  "created": "2025-01-04T10:30:00Z"
}
```

Asignación de Múltiples Roles:

Los usuarios pueden tener múltiples roles. Los permisos son aditivos (unión de todos los permisos de roles asignados).

Para asignar múltiples roles, inclúyelos en la solicitud:

```
{
  "username": "jane.doe",
  "email": "jane.doe@company.com",
  "password": "TempP@ssw0rd!",
  "first_name": "Jane",
  "last_name": "Doe",
  "role": "Support,Finance"
}
```

O utiliza el endpoint de asignación de roles después de la creación del usuario:

```
POST /auth/roles/{role_id}/users/{user_id}
```

Ver y Buscar Usuarios

Listar Todos los Usuarios (Admin):

`GET /auth/users`

Devuelve una lista paginada de todos los usuarios con sus roles e información básica.

Buscar Usuarios:

`GET /auth/users/search?search={query}&filters={"role":["Support"]}&page=1&per_page=50`

Filtrar por:

- Nombre del rol
- Dominio de correo electrónico
- Estado activo/eliminado
- Estado de 2FA habilitado
- Fecha del último inicio de sesión

Obtener Usuario Específico:

`GET /auth/users/{user_id}`

Devuelve detalles completos del usuario, incluyendo:

- Información personal
- Roles asignados y permisos efectivos
- Estado de 2FA
- Último inicio de sesión e información de sesión
- Contacto de cliente vinculado (si aplica)

Creación y Gestión de Roles

Los roles son colecciones de permisos que pueden ser asignados a usuarios. En lugar de asignar permisos individualmente a cada usuario, creas roles que agrupan permisos relacionados y asignas esos roles a los usuarios.

Creando un Nuevo Rol

A través de la Interfaz Web:

1. Navegar a **Usuarios y Roles** → pestaña **Roles**.
2. Hacer clic en **"Crear Rol"**.
3. Ingresar detalles del rol:
   - **Nombre** - Nombre corto y descriptivo (por ejemplo, "Tier2_Support").
   - **Descripción** - Explicar el propósito y responsabilidades del rol.
4. Hacer clic en **"Crear"**.
5. El rol es creado sin permisos; agregar permisos en el siguiente paso.

A través de la API:

Endpoint: `POST /auth/roles`

Permiso Requerido: `admin`

Solicitud:

```
{
  "name": "Tier2_Support",
  "description": "Equipo de soporte de nivel 2 con acceso elevado
a aprovisionamiento"
}
```

Respuesta:

```
{
  "id": 45,
  "name": "Tier2_Support",
  "description": "Equipo de soporte de nivel 2 con acceso elevado
a aprovisionamiento",
  "permissions": [],
  "users": []
}
```


Agregar Permisos a un Rol

Una vez que se crea un rol, asigna permisos para definir lo que los usuarios con ese rol pueden hacer.

A través de la Interfaz Web:

1. Navegar a **Usuarios y Roles** → pestaña **Roles**.
2. Hacer clic en el nombre del rol para ver detalles.
3. En la sección **Permisos**, hacer clic en **"Agregar Permiso"**.
4. Seleccionar uno o más permisos de la lista.
5. Hacer clic en **"Agregar"** - Los permisos se asignan inmediatamente.

A través de la API:

Endpoint: `POST /auth/roles/{role_id}/permissions`

Solicitud:

```
{
  "permission_id": 123
}
```

O agregar múltiples permisos:

```
{
  "permission_ids": [123, 124, 125]
}
```

Ejemplo: Creando un Rol de "Especialista en Aprovisionamiento"

Este rol puede ver clientes, gestionar servicios y aprovisionar:

1. Crear el rol:

```
POST /auth/roles
{
  "name": "Provisioning_Specialist",
  "description": "Puede aprovisionar servicios y gestionar
servicios de clientes"
}
```

2. Agregar permisos:

```
POST /auth/roles/45/permissions
{
  "permission_ids": [
    1,   # view_customer
    20,  # view_customer_service
    21,  # create_customer_service
    22,  # update_customer_service
    30,  # view_provision
    31,  # create_provision
    40,  # view_inventory
    50,  # view_product
  ]
}
```

Eliminar Permisos de un Rol

A través de la Interfaz Web:

1. Navegar a los detalles del rol.
2. En la lista de **Permisos**, hacer clic en la **"X"** o en el botón **"Eliminar"** junto al permiso.
3. Confirmar la eliminación.

A través de la API:

Endpoint: `DELETE /auth/roles/{role_id}/permissions/{permission_id}`

Ejemplo:

```
DELETE /auth/roles/45/permissions/31
```

Esto elimina el permiso `create_provision` del rol.

Editar Detalles del Rol

Actualizar el nombre o la descripción del rol:

A través de la Interfaz Web:

1. Navegar a **Usuarios y Roles** → pestaña **Roles**.
2. Hacer clic en el rol para editar.
3. Modificar el nombre o la descripción del rol.
4. Hacer clic en **"Guardar"**.

A través de la API:

Endpoint: `PUT /auth/roles/{role_id}`

```
{
  "name": "Senior_Support",
  "description": "Equipo de soporte senior con acceso completo a clientes"
}
```

Eliminar un Rol

Advertencia: Eliminar un rol lo elimina de todos los usuarios asignados. Asegúrate de que los usuarios tengan roles alternativos o perderán acceso.

A través de la API:

`DELETE /auth/roles/{role_id}`

Mejor Práctica: En lugar de eliminar, considera archivar o renombrar roles que ya no son necesarios.

Asignar Roles a Usuarios

Durante la Creación de Usuarios:

Incluir rol en la solicitud de creación de usuario (ver "Agregar Nuevos Usuarios" arriba).

Para Usuarios Existentes:

A través de la Interfaz Web:

1. Navegar a **Usuarios y Roles** → pestaña **Usuarios**.
2. Hacer clic en el usuario para editar.
3. En la sección **Roles**, seleccionar/deseleccionar roles.
4. Hacer clic en **"Guardar"**.

A través de la API:

Actualizar los roles del usuario:

Endpoint: `PUT /auth/users/{user_id}`

```
{
  "role": "Support,Finance"
}
```

O asignar un solo rol a un usuario a través del endpoint de rol:

Endpoint: `POST /auth/roles/{role_id}/users/{user_id}`

Ver Asignaciones de Roles

Todos los usuarios en un rol:

`GET /auth/roles/{role_id}/users`

Devuelve una lista de todos los usuarios asignados a ese rol.

Todos los roles para un usuario:

`GET /auth/users/{user_id}`

La respuesta incluye un array `roles` con todos los roles asignados.

Gestión de Contraseñas de Usuario

OmniCRM proporciona múltiples métodos para la gestión de contraseñas dependiendo del contexto.

Restablecimiento de Contraseña de Autoservicio del Usuario

Los usuarios que olvidaron su contraseña pueden restablecerla ellos mismos a través de la página de inicio de sesión:

1. **Hacer clic en "Olvidé mi Contraseña"** en la página de inicio de sesión.
2. **Ingresar dirección de correo electrónico** - El sistema envía un correo electrónico de restablecimiento de contraseña.
3. **Revisar correo electrónico** - El correo contiene un enlace de restablecimiento seguro con token (válido por 1 hora).
4. **Hacer clic en el enlace** - Abre el formulario de restablecimiento de contraseña.
5. **Ingresar nueva contraseña** - Debe cumplir con los requisitos de complejidad de contraseña:
   - Mínimo 8 caracteres.
   - Al menos una letra mayúscula.
   - Al menos una letra minúscula.
   - Al menos un número.
   - Al menos un carácter especial.
6. **Enviar** - La contraseña se actualiza inmediatamente; el usuario puede iniciar sesión con la nueva contraseña.

Flujo de API:

1. **Solicitar restablecimiento:**

   Endpoint: `POST /auth/forgot_password`

```
{
  "email": "user@example.com"
}
```

El sistema genera un token de restablecimiento y envía un correo electrónico.

2. **Restablecer con token:**

   Endpoint: `POST /auth/reset_password`

```
{
  "token": "abc123...",
  "new_password": "NewSecureP@ssw0rd!"
}
```

Restablecimiento de Contraseña por Administrador

Los administradores pueden restablecer la contraseña de un usuario sin requerir verificación por correo electrónico. Esto establece una contraseña temporal que el usuario debe cambiar en el próximo inicio de sesión.

A través de la Interfaz Web:

1. Navegar a **Usuarios y Roles** → pestaña **Usuarios**.
2. Encontrar al usuario y hacer clic en el botón **"Restablecer Contraseña"**.
3. Ingresar una contraseña temporal.
4. Hacer clic en **"Restablecer"**.
5. Notificar al usuario de su contraseña temporal (a través de un canal seguro).
6. El usuario debe cambiar la contraseña en el próximo inicio de sesión.

A través de la API:

Endpoint: `POST /auth/users/{user_id}/admin_reset_password`

Permiso Requerido: `admin`

Solicitud:

```
{
  "new_password": "TempP@ssw0rd!",
  "force_change": true
}
```

Parámetros:

- `new_password` - Contraseña temporal a establecer.
- `force_change` (opcional) - Si es verdadero, el usuario debe cambiar la contraseña en el próximo inicio de sesión.

Cambio de Contraseña por el Usuario

Los usuarios autenticados pueden cambiar su propia contraseña desde su perfil:

Endpoint: `POST /auth/change_password`

Solicitud:

```
{
  "current_password": "OldP@ssw0rd!",
  "new_password": "NewSecureP@ssw0rd!"
}
```

El sistema valida la contraseña actual antes de permitir el cambio.

Seguridad de Contraseña

- Las contraseñas se hash utilizan bcrypt (seguridad de werkzeug).
- Nunca se almacenan en texto plano.
- Los tokens de restablecimiento expiran después de 1 hora.

- Los intentos de inicio de sesión fallidos pueden activar el bloqueo de la cuenta (configurable).
- El seguimiento del historial de contraseñas previene la reutilización (configurable).
- Se imponen requisitos de complejidad.

Gestión de Autenticación de Dos Factores (2FA)

OmniCRM admite autenticación de dos factores basada en TOTP para mayor seguridad. Los administradores pueden habilitar, deshabilitar y restablecer 2FA para los usuarios.

Habilitar 2FA para un Usuario

A través de la Interfaz Web:

1. Navegar a **Usuarios y Roles** → pestaña **Usuarios**.
2. Hacer clic en el usuario para ver detalles.

3. En la sección **Seguridad**, hacer clic en **"Habilitar 2FA"**.
4. El sistema genera:
    - Secreto TOTP (código QR mostrado).
    - 10 códigos de respaldo (uso único).
5. El usuario escanea el código QR con la aplicación de autenticador (Google Authenticator, Authy, etc.).
6. El usuario ingresa el código de verificación de la aplicación para confirmar la configuración.
7. El usuario guarda los códigos de respaldo en un lugar seguro.
8. 2FA ahora está habilitado; requerido para todos los futuros inicios de sesión.

A través de la API:

1. **Generar secreto TOTP:**

   Endpoint: `POST /2fa/enable/user/{user_id}`

   Respuesta:

   ```
   {
     "totp_secret": "JBSWY3DPEHPK3PXP",
     "qr_code_url": "otpauth://totp/OmniCRM:user@example.com?
   secret=JBSWY3DPEHPK3PXP&issuer=OmniCRM",
     "backup_codes": [
       "12345678",
       "23456789",
       "34567890",
       ...
     ]
   }
   ```

2. **Verificar configuración:**

   Endpoint: `POST /2fa/verify-setup/user/{user_id}`

   ```
   {
     "code": "123456"
   }
   ```

   Devuelve `{"verified": true}` en caso de éxito.

Flujo de Inicio de Sesión 2FA

Una vez habilitado 2FA, el proceso de inicio de sesión cambia:

1. El usuario ingresa nombre de usuario y contraseña.
2. El sistema valida las credenciales.
3. Si son válidas, solicita el código 2FA.
4. El usuario ingresa el código de la aplicación de autenticador O el código de respaldo.
5. El sistema verifica el código.

6. En caso de éxito, el usuario inicia sesión.

Códigos de Respaldo:

- 10 códigos generados durante la configuración de 2FA.
- Solo de un solo uso (consumidos después de su uso).
- Usados si la aplicación de autenticador no está disponible.
- Pueden ser regenerados por el usuario o el administrador.

Verificación del Código 2FA

Endpoint: `POST /2fa/verify/user/{user_id}`

```
{
  "code": "123456"
}
```

Acepta ambos:

- **Código TOTP** (6 dígitos de la aplicación de autenticador).
- **Código de respaldo** (8 dígitos de la lista de códigos de respaldo).

Regenerar Códigos de Respaldo

Si un usuario agota los códigos de respaldo o los pierde, genera nuevos:

A través de la Interfaz Web:

1. Navegar a los detalles del usuario.
2. Hacer clic en **"Regenerar Códigos de Respaldo"**.
3. Mostrar o enviar nuevos códigos al usuario.
4. Los códigos antiguos se invalidan.

A través de la API:

Endpoint: `POST /2fa/backup-codes/regenerate/user/{user_id}`

Respuesta:

```
{
  "backup_codes": [
    "98765432",
    "87654321",
    "76543210",
    ...
  ]
}
```

Restablecimiento 2FA por Administrador

Si un usuario pierde el acceso a su aplicación de autenticador y todos los códigos de respaldo, un administrador puede deshabilitar y re-habilitar 2FA.

A través de la Interfaz Web:

1. Navegar a **Usuarios y Roles** → pestaña **Usuarios**.
2. Hacer clic en el usuario.
3. Hacer clic en el botón **"Restablecer 2FA"**.
4. Confirmar el restablecimiento.

5. 2FA está deshabilitado; el usuario puede iniciar sesión solo con la contraseña.
6. Guiar al usuario para que configure 2FA nuevamente con un nuevo secreto.

A través de la API:

Endpoint: `POST /2fa/admin/disable/user/{user_id}`

Permiso Requerido: `admin`

Esto deshabilita completamente 2FA para el usuario:

- Se borra el secreto TOTP.
- Se borran los códigos de respaldo.
- Se establece la bandera `is_2fa_enabled` en falso.

El usuario puede luego re-habilitar 2FA para obtener un nuevo secreto y códigos de respaldo.

Restablecimiento 2FA de Autoservicio del Usuario (Nuevo Dispositivo)

Si un usuario obtiene un nuevo dispositivo pero aún tiene acceso a los códigos de respaldo:

Endpoint: `POST /2fa/reset-for-new-device/user/{user_id}`

```
{
  "backup_code": "12345678"
}
```

El sistema valida el código de respaldo, luego genera un nuevo secreto TOTP y códigos de respaldo. El usuario puede configurar la aplicación de autenticador en el nuevo dispositivo.

Mejores Prácticas para 2FA

- **Requerir 2FA para todo el personal administrativo y de soporte.**
- **Almacenar códigos de respaldo de forma segura** (administrador de contraseñas o nota segura).
- **Regenerar códigos de respaldo** después de usar varios.
- **Usar aplicaciones de autenticador de buena reputación** (Google Authenticator, Authy, Microsoft Authenticator).
- **Documentar los procedimientos de restablecimiento de 2FA** para el personal de soporte.
- **Auditar el uso de 2FA** - monitorear qué usuarios tienen 2FA habilitado.

Actualización de Información del Usuario

Los administradores pueden actualizar los detalles del usuario en cualquier momento.

A través de la Interfaz Web:

1. Navegar a **Usuarios y Roles** → pestaña **Usuarios**.
2. Hacer clic en el usuario para editar.
3. Modificar cualquier campo editable:
    - Nombre, segundo nombre, apellido.
    - Dirección de correo electrónico (requiere verificación).
    - Número de teléfono.
    - Roles.
    - Vinculación de contacto de cliente.
4. Hacer clic en **"Guardar"**.

A través de la API:

Endpoint: `PUT /auth/users/{user_id}`

```
{
  "first_name": "Jane",
  "last_name": "Doe-Smith",
  "email": "jane.doesmith@newcompany.com",
  "phone_number": "+61498765432",
  "role": "Support,Finance"
}
```

Cambios de Correo Electrónico:

Cuando se cambia el correo electrónico, el nuevo correo se marca como pendiente hasta que se verifique:

- El campo `pending_email` almacena el nuevo correo.
- Se envía un correo electrónico de verificación a la nueva dirección.
- El usuario hace clic en el enlace para verificar.
- El campo `email` se actualiza al nuevo valor.
- La bandera `email_verified` se establece en verdadero.

Eliminación de Usuarios

OmniCRM utiliza **eliminaciones suaves** para usuarios: se marcan como eliminados pero no se eliminan de la base de datos. Esto preserva las auditorías y los datos históricos.

Eliminar un Usuario

A través de la Interfaz Web:

1. Navegar a **Usuarios y Roles** → pestaña **Usuarios**.
2. Encontrar el usuario a eliminar.
3. Hacer clic en el botón **"Eliminar"**.
4. Confirmar la eliminación.
5. El usuario es inmediatamente desconectado y no puede volver a iniciar sesión.

A través de la API:

Endpoint: `DELETE /auth/users/{user_id}`

Permiso Requerido: `admin`

Qué Ocurre:

- La bandera `deleted` se establece en `True`.
- Se registra la marca de tiempo `deleted_at`.
- El usuario no puede iniciar sesión.
- Todas las sesiones activas se invalidan.
- El usuario aún aparece en los registros de auditoría y registros históricos.
- Los datos vinculados (contactos de clientes, actividades) se preservan.

Ver Usuarios Eliminados

Filtrar por usuarios eliminados:

`GET /auth/users/search?filters={"deleted":[true]}`

Restaurar un Usuario Eliminado

Si un usuario fue eliminado por error, los administradores pueden restaurarlo:

Endpoint: `PUT /auth/users/{user_id}`

```
{
  "deleted": false
}
```

Esto borra la bandera `deleted` y permite que el usuario inicie sesión nuevamente.

Nota: La contraseña del usuario permanece sin cambios, por lo que puede usar su contraseña anterior.

Eliminación Permanente de un Usuario

Advertencia: Esto es irreversible y elimina todos los datos del usuario de la base de datos.

No expuesto a través de la interfaz. Solo disponible a través de acceso directo a la base de datos por razones de cumplimiento (por ejemplo, solicitudes de eliminación de datos GDPR).

Mejores Prácticas para la Eliminación de Usuarios

- **Eliminar suavemente por defecto** - Preserva las auditorías.
- **Documentar razones de eliminación** - Agregar nota en el registro de actividad antes de eliminar.
- **Transferir propiedad** - Reasignar tickets abiertos y tareas del usuario antes de eliminar.
- **Revisar acceso** - Asegurarse de que no haya procesos críticos que dependan del usuario.
- **Archivar datos** - Exportar el historial de trabajo del usuario si es necesario.
- **Notificar a equipos relevantes** - Informar a gerentes/compañeros sobre la eliminación.

Catálogo de Permisos

Los permisos generalmente siguen patrones CRUD:

- *view_**: leer/navegar.
- *create_**: crear/agregar.
- *update_**: editar/modificar.
- *delete_**: eliminar/quitar.

Algunas entidades también incluyen variantes de **“ver propio …”** que restringen la visibilidad a los datos del cliente/inquilino **propio** del usuario.

Global / Administrativo

- `admin`: Acceso administrativo completo (gestionar usuarios, roles y permisos; acceder a todos los puntos finales protegidos).
- `can_impersonate`: Actuar temporalmente como otro usuario (auditado; para soporte/troubleshooting).
- `grafana_access`: Acceso a paneles de análisis de Grafana para crear y ver informes y visualizaciones personalizadas. Consulta la sección de Análisis de Grafana a continuación para más detalles.

Clientes y Registros Relacionados

- **Cliente**
  - *view_customer*, *create_customer*, *update_customer*, *delete_customer*
  - **Alcance propio:** *view own customer*
- **Sitio del Cliente**
  - *view_customer_site*, *create_customer_site*, *update_customer_site*, *delete_customer_site*
  - **Alcance propio:** *view own customer site*
- **Contacto del Cliente**

    - *view_customer_contact*, *create_customer_contact*, *update_customer_contact*, *delete_customer_contact*
    - **Alcance propio:** *view own customer contact*
- **Atributo del Cliente** (ver Atributos del Cliente)
    - *view_customer_attribute*, *create_customer_attribute*, *update_customer_attribute*, *delete_customer_attribute*
    - **Alcance propio:** *view own customer attribute*
- **Etiqueta del Cliente** (ver Etiquetas del Cliente)
    - *view_customer_tag*, *create_customer_tag*, *update_customer_tag*, *delete_customer_tag*
    - **Alcance propio:** *view own customer tag*
- **Servicio del Cliente**
    - *view_customer_service*, *create_customer_service*, *update_customer_service*, *delete_customer_service*
    - **Alcance propio:** *view own customer service*
- **Actividad del Cliente**
    - *view_customer_activity*, *create_customer_activity*, *update_customer_activity*, *delete_customer_activity*
    - **Alcance propio:** *view own customer activity*

Facturación

- **Tarjeta de Stripe**
    - `view_customer_stripe_card`, `create_customer_stripe_card`, `update_customer_stripe_card`, `delete_customer_stripe_card`
    - **Alcance propio:** *view own customer stripe card*
- **Transacciones**
    - `view_customer_transaction`, `create_customer_transaction`, `update_customer_transaction`, `delete_customer_transaction`
    - **Alcance propio:** *view own customer transaction*
- **Facturas**
    - `view_customer_invoice`, `create_customer_invoice`, `update_customer_invoice`, `delete_customer_invoice`
    - **Alcance propio:** *view own customer invoice*

Comunicaciones

- *view_communication*, *create_communication*, *update_communication*, *delete_communication*
- **Alcance propio:** *view own communication*

Inventario y Plantillas

- **Inventario**
  - `view_inventory`, `create_inventory`, `update_inventory`, `delete_inventory`
  - **Alcance propio:** *view own inventory*
- **Plantilla de Inventario**
  - `view_inventory_template`, `create_inventory_template`, `update_inventory_template`, `delete_inventory_template`
  - **Alcance propio:** *view own inventory template*

Productos

- `view_product`, `create_product`, `update_product`, `delete_product`

Difusión Celular (CBC)

- `view_cbc_message`, `create_cbc_message`, `update_cbc_message`, `delete_cbc_message`

Aprovisionamiento

- **Aprovisionar**
  - `view_provision`, `create_provision`, `update_provision`, `delete_provision`
  - **Alcance propio:** *view own provision*
- **Evento de Aprovisionamiento**
  - `view_provision_event`, `create_provision_event`, `update_provision_event`, `delete_provision_event`

Acceso "Ver Propio"

Los permisos “ver propio …” limitan las lecturas (y opcionalmente las ediciones, donde se implementen) a los datos asociados con el **propio cliente/inquilino** del usuario. Por ejemplo, un rol de *Administrador de Clientes* puede gestionar los contactos, sitios, facturas y servicios de *su* inquilino, pero no puede ver otros inquilinos.

Diseños de Roles Típicos

Rol	Permisos Típicos	Notas
Administrador del Sistema	`admin`, opcionalmente `can_impersonate`; más CRUD amplio según sea necesario	Control total sobre usuarios/roles/permisos
Soporte	`view_customer`, `view_customer_service`, `view_communication`, `view_provision`; actualizaciones opcionales	Agregar `can_impersonate` si se permite
Finanzas	`view_customer_invoice`, `view_customer_transaction`, `view_product`; opcional `create_customer_invoice`	Enfoque en lectura; escritura limitada
Administrador de Clientes (inquilino)	"view own ..." en cliente, sitios, contactos, servicios, inventario, facturas, transacciones, comunicaciones, aprovisionamiento	Gestión con alcance de inquilino
Auditor Solo Lectura	Amplio `view_*` solamente	Sin crear/actualizar/eliminar

Ejemplo de Roles y Permisos Incluidos (resumen)

Gestión de Roles y Permisos a través de la API

Todos los endpoints requieren permiso `admin`.

Listar permisos

Endpoint: `GET /auth/permissions`

Crear un permiso (raro)

Endpoint: `POST /auth/permissions`

Cuerpo de la Solicitud:

```
{
    "name": "view_example",
    "description": "Acceso solo de lectura a objetos de ejemplo"
}
```

Listar roles

Endpoint: `GET /auth/roles`

Crear un rol

Endpoint: `POST /auth/roles`

Cuerpo de la Solicitud:

```
{
"name": "Support",
"description": "Equipo de soporte de nivel 1"
}
```

Agregar un permiso a un rol

Endpoint: `POST /auth/roles/{role_id}/permissions`

Cuerpo de la Solicitud:

```
{
  "permission_id": 123
}
```

Eliminar un permiso de un rol

Endpoint: `DELETE /auth/roles/{role_id}/permissions/{permission_id}`

Asignación de Roles a Usuarios

Crear un usuario con rol

Endpoint: `POST /auth/users`

Cuerpo de la Solicitud:

```
{
  "username": "sara",
  "email": "sara@example.com",
  "password": "TempP@ssw0rd!",
  "first_name": "Sara",
  "last_name": "Ng",
  "phone_number": "+61...",
  "role": "Support"
}
```

Actualizar el rol de un usuario

Endpoint: `PUT /auth/users/{user_id}`

Cuerpo de la Solicitud:

```
{
  "role": "Finance"
}
```

Listar usuarios (solo Admin)

Endpoint: `GET /auth/users`

Suplantación (Controlada)

- **Requerido:** `can_impersonate` o `admin`

Iniciar suplantación

Endpoint: `POST /auth/impersonate`

Cuerpo de la Solicitud:

```
{ "user_id": 42 }
```

Detener suplantación

Endpoint: `POST /auth/stop_impersonation`

Mejores Prácticas

- **Principio de menor privilegio primero.** Comienza con roles mínimos; agrega permisos según sea necesario.
- **Preferir "ver propio ...".** Utiliza permisos con alcance de inquilino para roles orientados al cliente.
- **Mantener roles estables.** Actualiza los permisos de rol cuando cambien las funciones. No edites a cada usuario.
- **Auditar regularmente.** Revisa quién tiene `admin` o `can_impersonate`.

Análisis y Paneles de Grafana

OmniCRM se integra con Grafana para proporcionar potentes capacidades de análisis y visualización. Los usuarios con el permiso `grafana_access` pueden acceder a Grafana para crear paneles, informes y visualizaciones personalizadas.

Accediendo a Grafana

Si tu cuenta de usuario ha sido otorgada el permiso `grafana_access`, puedes acceder a Grafana navegando a `/grafana` en tu navegador. Serás autenticado automáticamente utilizando tus credenciales de OmniCRM - no se requiere un inicio de sesión separado en Grafana.

Desde la interfaz de Grafana, puedes:

- Crear paneles personalizados con gráficos, tablas y gráficos.
- Construir consultas complejas para analizar datos de clientes, tendencias de ingresos y métricas de servicio.
- Configurar alertas basadas en indicadores clave de rendimiento.
- Compartir paneles con otros miembros del equipo.

Incrustar Paneles en OmniCRM

Una vez que hayas creado paneles en Grafana, puedes incrustarlos directamente en la interfaz de OmniCRM. Esto te permite ver tus métricas más importantes sin salir del CRM, manteniendo la navegación de OmniCRM intacta.

Para incrustar un panel:

1. **Crea tu panel en Grafana** - Navega a `/grafana` y diseña tu panel con las visualizaciones que necesitas.

2. **Anota el ID del panel** - Al ver tu panel en Grafana, verifica la URL. Se verá como `/grafana/d/abc123/nombre-del-panel`. El ID es la parte después de `/d/` (en este ejemplo, `abc123`).

3. **Configura el panel en tu entorno** - Contacta a tu administrador del sistema para agregar el ID del panel y un nombre de visualización a la configuración del sistema. Los paneles se configuran a través de variables de entorno.

4. **Accede desde la barra lateral** - Una vez configurado, tu panel aparecerá automáticamente en la barra lateral de OmniCRM bajo el menú "Paneles". Puedes hacer clic en él para ver el panel completo mientras mantienes visible la navegación de OmniCRM.

Configuración del Panel

Los paneles son configurados por tu administrador del sistema utilizando variables de entorno. Cada panel requiere:

- **ID del Panel** - El identificador único de la URL de Grafana.
- **Etiqueta del Panel** - El nombre amigable que se muestra en la barra lateral de OmniCRM.

Se pueden configurar múltiples paneles y aparecerán como elementos de menú en el orden en que se definen. Consulta la documentación de Configuración de Administración para obtener detalles sobre la configuración de variables de entorno.

¿Quién Puede Acceder a Grafana?

Solo los usuarios con el permiso `grafana_access` pueden acceder a Grafana y ver paneles incrustados. Este permiso se otorga típicamente a:

- Administradores que necesitan visibilidad total sobre las métricas del sistema.
- Analistas de negocios que crean informes y visualizaciones.
- Equipos de finanzas que rastrean métricas de ingresos y facturación.
- Equipos de operaciones que monitorean el rendimiento del servicio.

Preguntas Frecuentes

¿Puede un usuario tener múltiples roles? Sí. Los permisos son aditivos.

¿Necesito permisos personalizados? Generalmente no. El catálogo incorporado cubre la mayoría de las necesidades.

¿Cómo saben las reglas "ver propio …" qué es mío? Evalúan el vínculo entre tu usuario/contacto y tu cliente (inquilino).

¿Puedo crear mis propios paneles de Grafana? Sí, si tienes el permiso `grafana_access`. Navega a `/grafana` para acceder a la interfaz completa de Grafana.

Portal de Autocuidado

El Portal de Autocuidado es una interfaz orientada al cliente que permite a los usuarios finales gestionar sus propias cuentas, ver el uso, pagar facturas y modificar servicios sin necesidad de asistencia del personal de atención al cliente.

Métodos de Acceso:

- Inicio de sesión directo a través de credenciales del cliente
- Impersonación de personal para resolución de problemas (desde Clientes → Contactos → "Iniciar sesión como usuario")

Vea también: Atención al Cliente para detalles de impersonación, Flujos de Autenticación para el proceso de inicio de sesión.

Propósito

El Portal de Autocuidado proporciona a los clientes:

1. **Gestión de Cuentas** - Ver y actualizar información personal, contactos y direcciones
2. **Resumen de Servicios** - Ver todos los servicios activos, uso y fechas de caducidad
3. **Seguimiento de Uso** - Monitorear datos, voz, SMS y saldos monetarios
4. **Acceso a Facturación** - Ver y pagar facturas, gestionar métodos de pago
5. **Modificaciones de Servicios** - Agregar recargas, comprar complementos, modificar servicios
6. **Disponibilidad 24/7** - Acceder a la información de la cuenta en cualquier momento sin llamar al soporte

Resumen del Portal

Cuando los clientes inician sesión en el Portal de Autocuidado, ven un panel personalizado con:

Secciones de Navegación:

- **Panel** - Resumen rápido de servicios y actividad reciente
- **Cuenta** - Detalles personales, contactos, sitios
- **Servicios** - Lista de todos los servicios con estado y detalles
- **Uso** - Información sobre el consumo de saldo y fechas de caducidad
- **Facturación** - Facturas, transacciones, métodos de pago
- **Recarga** - Comprar crédito de datos, voz, SMS o crédito monetario

Detalles de la Cuenta

La sección de Cuenta muestra la información del cliente y permite actualizaciones de autoservicio limitadas.

Campos Editables:

Los clientes pueden actualizar:

- Dirección de correo electrónico (requiere verificación)
- Número de teléfono
- Contraseña
- Preferencias de notificación

Información Solo de Visualización:

- ID del cliente
- Fecha de creación de la cuenta
- Tipo de cliente (Individual/Empresa)
- Sitios (direcciones)
- Contactos vinculados

Actualización de Información de la Cuenta:

1. Navegar a **Cuenta** → **Detalles**
2. Hacer clic en **"Editar"** junto al campo que se desea actualizar
3. Ingresar nueva información
4. Hacer clic en **"Guardar Cambios"**
5. Para cambios de correo electrónico, verificar a través del enlace enviado a la nueva dirección

Características de Seguridad:

- Los cambios de contraseña requieren la contraseña actual
- Los cambios de correo electrónico requieren verificación
- La actividad se registra para el seguimiento de auditoría
- Configuraciones de 2FA (si están habilitadas)

Resumen de Servicios

La sección de Servicios muestra todos los servicios activos e inactivos del cliente.

Visualización de la Tarjeta de Servicio:

Cada servicio muestra:

- **Nombre del Servicio** - Identificador legible por humanos (por ejemplo, "Móvil - +44 7700 900123")
- **Producto** - Nombre del plan o producto
- **Estado** - Activo, Suspendido, Expirado, Cancelado
- **Fecha de Creación** - Fecha de activación del servicio
- **Fecha de Caducidad** - Cuándo expira el servicio (si aplica)
- **Costo Mensual** - Cargo recurrente
- **Renovación Automática** - Indicador Habilitado/Deshabilitado

Acciones del Servicio:

- **Ver Uso** - Ver consumo de saldo (Datos, Voz, SMS, Monetario)
- **Recargar** - Agregar crédito o datos
- **Complementos** - Comprar características adicionales
- **Modificar** - Cambiar parámetros del servicio (si está permitido)
- **Ver Detalles** - Ver la configuración completa del servicio

Indicadores de Estado del Servicio:

- **Activo** - El servicio está operativo
- **Expirando Pronto** - Se renueva o expira en <7 días
- **Suspendido** - Servicio temporalmente deshabilitado (problema de pago, suspensión manual)
- **Expirado** - El servicio ya no está activo

Seguimiento de Uso

Los clientes pueden monitorear su uso en todos los tipos de saldo en tiempo real.

Uso de Datos

Ver el consumo de datos con desgloses detallados por bucket y caducidad.

Visualización del Uso de Datos:

- **Saldo Total** - Todos los buckets de datos combinados
- **Usado Este Período** - Consumo desde la última renovación

- **Barra de Progreso** - Representación visual del consumo
- **Información de Caducidad** - Cuándo expira cada bucket
- **Desglose de Buckets** - Múltiples buckets de datos con orden de prioridad

Prioridad de Buckets:

Cuando existen múltiples buckets de datos (por ejemplo, plan base + recargas), se consumen en orden de peso:

- **Peso 10** - Consumido primero (típicamente datos promocionales/bono)
- **Peso 20** - Consumido segundo (típicamente datos de recarga)
- **Peso 30** - Consumido al final (típicamente datos del plan base)

Uso de Voz

Rastrear minutos de llamada restantes en todos los buckets de voz.

Visualización del Uso de Voz:

- **Minutos Restantes** - Saldo total de voz
- **Minutos Usados** - Consumo este período
- **Historial de Llamadas** - Llamadas recientes (si está habilitado)

- **Fechas de Caducidad** - Cuándo expiran los buckets de voz
- **Minutos Internacionales** - Seguimiento separado (si aplica)

Desglose de Uso:

- Llamadas On-Net - Llamadas dentro de la misma red
- Llamadas Off-Net - Llamadas a otras redes
- Llamadas Internacionales - Llamadas fuera del país
- Números Premium - Números de tarifa especial

Uso de SMS

Monitorear las asignaciones y el consumo de mensajes SMS.

Visualización del Uso de SMS:

- **Mensajes Restantes** - Saldo de SMS
- **Usados Este Mes** - Mensajes enviados
- **MMS Incluidos** - Si los MMS cuentan para el saldo
- **SMS Internacionales** - Seguimiento separado (si aplica)

Saldo Monetario

Ver el saldo de crédito prepago para servicios de pago por uso.

Visualización Monetaria:

- **Saldo Actual** - Crédito disponible
- **Última Recarga** - Monto y fecha de recarga más reciente
- **Fecha de Caducidad** - Cuándo expira el saldo (si aplica)
- **Recarga Automática** - Estado Habilitado/Deshabilitado

Gestión de Facturación

Los clientes pueden ver facturas, transacciones y gestionar métodos de pago.

Facturas

Ver y pagar facturas pendientes directamente desde el portal.

Visualización de la Lista de Facturas:

- **Número de Factura** - Identificador único
- **Fecha** - Fecha de creación de la factura
- **Fecha de Vencimiento** - Plazo de pago
- **Monto** - Monto total de la factura
- **Estado** - Pagada, No Pagada, Atrasada
- **Acciones** - Descargar PDF, Pagar en Línea

Pago de una Factura:

1. Navegar a **Facturación** → **Facturas**
2. Encontrar la factura no pagada en la lista
3. Hacer clic en el botón **"Pagar Ahora"**
4. Seleccionar método de pago (tarjeta guardada o nueva tarjeta)
5. Confirmar el pago
6. Recibir correo electrónico de confirmación

Descargando Facturas:

1. Hacer clic en el ícono **"Descargar"** junto a la factura
2. PDF se descarga con todos los detalles de la factura
3. Guardado para registros fiscales y documentación

Transacciones

Ver el historial completo de transacciones, incluidos cargos, créditos y pagos.

Visualización de Transacciones:

- **Fecha** - Fecha de creación de la transacción
- **Descripción** - Para qué es el cargo/crédito
- **Monto** - Cargo (positivo) o crédito (negativo)
- **Factura** - Qué factura incluye esta transacción
- **Estado** - Facturado o No Facturado

Métodos de Pago

Gestionar tarjetas de crédito guardadas para facturación automática y pagos en línea.

Gestión de Métodos de Pago:

- **Agregar Tarjeta** - Agregar de forma segura una nueva tarjeta de crédito/débito a través de Stripe
- **Establecer Predeterminado** - Elegir método de pago principal
- **Eliminar Tarjeta** - Eliminar tarjetas caducadas o no utilizadas
- **Detalles de la Tarjeta** - Ver los últimos 4 dígitos, caducidad, marca de la tarjeta

Vea Métodos de Pago para documentación detallada sobre gestión de pagos.

Recarga / Recarga

Comprar datos adicionales, voz, SMS o crédito monetario al instante.

Proceso de Recarga:

1. Navegar a **Servicios** → Seleccionar servicio → **"Recargar"**
2. Elegir producto de recarga del catálogo
3. Seleccionar monto (opciones preestablecidas o personalizadas)
4. Revisar costo e información de caducidad
5. Seleccionar método de pago
6. Confirmar compra
7. Saldo actualizado inmediatamente

Recargas Disponibles:

- **Recargas de Datos** - Opciones de 1GB, 5GB, 10GB, 20GB, 50GB
- **Recargas de Voz** - Minutos de llamada adicionales
- **Paquetes de SMS** - Paquetes de mensajes
- **Crédito Monetario** - Saldo prepago (£5, £10, £20, £50, £100)

Vea Recarga y Recarga para flujos de trabajo de recarga detallados.

Complementos de Servicio

Comprar características y mejoras adicionales para servicios existentes.

Complementos Disponibles:

- **Roaming Internacional** - Habilitar servicio en el extranjero
- **Dirección IP Estática** - IP fija para internet en casa
- **Contenido Premium** - Canales IPTV, servicios de streaming
- **Mejoras de Hardware** - Alquiler de módem, decodificadores
- **Aumentos de Velocidad** - Aumentos temporales de ancho de banda

Compra de Complementos:

1. Navegar a **Servicios** → Seleccionar servicio → **"Complementos"**
2. Navegar por los complementos disponibles para este tipo de servicio
3. Hacer clic en **"Agregar al Servicio"** en el complemento deseado
4. Revisar costo (único + recurrente)
5. Confirmar compra
6. Complemento provisionado automáticamente

Vea Modificación de Servicios - Complementos para detalles sobre la gestión de complementos.

Notificaciones y Alertas

Los clientes reciben notificaciones automatizadas para eventos importantes:

Notificaciones por Correo Electrónico:

- Factura generada y lista para pago
- Confirmación de pago recibido
- Advertencias de caducidad del servicio (7 días, 3 días, 1 día)
- Alertas de saldo bajo (datos, voz, monetario)
- Activación/desactivación del servicio
- Solicitudes de restablecimiento de contraseña
- Alertas de seguridad (inicio de sesión desde un nuevo dispositivo)

Alertas en el Portal:

- Facturas no pagadas
- Servicios que expiran
- Advertencias de datos bajos (10% restantes)
- Método de pago que expira
- Acciones requeridas (verificar correo electrónico, actualizar método de pago)

Preferencias de Notificación:

Los clientes pueden configurar:

- Frecuencia de notificación por correo electrónico
- Alertas por SMS (si están habilitadas)
- Umbrales de alerta (por ejemplo, notificar cuando <20% de datos restantes)
- Categorías de notificación (facturación, uso, servicio)

Limitaciones de Autoservicio

Algunas operaciones requieren asistencia del personal:

Requiere Atención al Cliente:

- Cambiar el tipo de cliente (Individual ↔ Empresa)
- Transferir servicios entre clientes
- Cancelar servicios (puede ser autoservicio si está habilitado)
- Disputar facturas
- Solicitar reembolsos
- Cambiar contacto principal
- Problemas complejos de aprovisionamiento

Restricciones de Seguridad:

- No se puede ver ni modificar las cuentas de otros usuarios
- Limitado a los propios datos del cliente (aislamiento de inquilinos)
- No se puede acceder a funciones administrativas
- No se pueden anular facturas o transacciones
- No se puede modificar la configuración del servicio (solo complementos/recargas)

Acceso de Impersonación del Personal

El personal de soporte puede acceder al Portal de Autocuidado como un cliente para la resolución de problemas.

Proceso de Impersonación:

1. Navegar a **Clientes** → Seleccionar cliente
2. Ir a la pestaña **Contactos**
3. Encontrar el contacto vinculado a la cuenta de usuario
4. Hacer clic en el botón **"Iniciar sesión como Usuario"**
5. Se abre una nueva pestaña con la vista del Portal de Autocuidado del cliente
6. Todas las acciones se registran y se atribuyen al usuario impersonado
7. El personal ve exactamente lo que ve el cliente

Casos de Uso:

- **Resolución de Problemas** - Reproducir problemas reportados por el cliente
- **Verificación** - Confirmar que las configuraciones de servicio aparecen correctamente
- **Capacitación** - Demostrar características del portal
- **Soporte** - Guiar al cliente a través del portal mientras se ve su pantalla

Seguridad y Auditoría:

- Requiere permiso `can_impersonate` o `admin`
- Todas las acciones se registran en el seguimiento de auditoría

- El cliente ve la impersonación en el registro de actividad
- Tiempo de espera de sesión después de la inactividad
- No se puede cambiar la contraseña del cliente mientras se está impersonando

Vea Atención al Cliente - Impersonación de Usuario para la documentación completa de impersonación.

Responsividad Móvil

El Portal de Autocuidado es completamente responsivo y está optimizado para dispositivos móviles.

Características Móviles:

- Navegación optimizada para táctil
- Diseños simplificados para pantallas pequeñas
- Gestos de deslizamiento para navegación en carrusel
- Formularios y entradas amigables para móviles
- Escaneo de códigos QR para aprovisionamiento de eSIM
- Llamadas telefónicas con un toque
- Integración de GPS para autocompletar direcciones

Aplicación Web Progresiva (PWA):

- Instalar como aplicación en la pantalla de inicio
- Visualización fuera de línea de datos recientes
- Notificaciones push (si están habilitadas)
- Carga rápida con trabajadores de servicio

Restablecimiento de Contraseña y Recuperación de Cuenta

Los clientes pueden restablecer contraseñas olvidadas sin llamar al soporte.

Restablecimiento de Contraseña de Autoservicio:

1. Hacer clic en **"Olvidé mi Contraseña"** en la página de inicio de sesión
2. Ingresar dirección de correo electrónico
3. Recibir correo electrónico de restablecimiento de contraseña (válido por 1 hora)
4. Hacer clic en el enlace en el correo electrónico
5. Ingresar nueva contraseña (debe cumplir con los requisitos de complejidad)
6. Enviar e iniciar sesión con la nueva contraseña

Requisitos de Contraseña:

- Mínimo 8 caracteres
- Al menos una letra mayúscula
- Al menos una letra minúscula
- Al menos un número
- Al menos un carácter especial (!@#$%^&*)

Bloqueo de Cuenta:

Después de 5 intentos fallidos de inicio de sesión:

- Cuenta bloqueada durante 30 minutos
- Correo electrónico de restablecimiento de contraseña enviado automáticamente
- Notificación de seguridad enviada al correo electrónico registrado

Mejores Prácticas para Clientes

Recomendaciones de Seguridad:

1. Habilitar 2FA para mayor seguridad
2. Usar una contraseña única y fuerte
3. Mantener actualizada la dirección de correo electrónico para notificaciones
4. Configurar el método de pago predeterminado para renovaciones automáticas

5. Monitorear el uso regularmente para evitar cargos por exceso
6. Guardar códigos de respaldo en un lugar seguro (si 2FA está habilitado)
7. Cerrar sesión después de usar computadoras compartidas/públicas

Gestión de Uso:

1. Habilitar alertas de saldo bajo (10-20% restantes)
2. Recargar antes de que los saldos expiren para evitar interrupciones del servicio
3. Revisar facturas mensuales en busca de cargos inesperados
4. Actualizar métodos de pago antes de que las tarjetas expiren
5. Monitorear el uso de datos durante el mes para evitar limitaciones

Escalación de Soporte:

Si el autoservicio no resuelve el problema:

1. Consultar la base de conocimientos / artículos de ayuda (si están disponibles)
2. Revisar el registro de actividad para cambios recientes
3. Contactar al servicio de atención al cliente por teléfono, correo electrónico o chat
4. Proporcionar ID de cliente y detalles del servicio para una resolución más rápida

Acceso a la API

Los clientes con requisitos técnicos pueden usar la API directamente.

Generación de Clave API:

Disponible para clientes comerciales o bajo solicitud:

1. Navegar a **Cuenta** → **Acceso a la API**
2. Hacer clic en **"Generar Clave API"**
3. Establecer permisos (solo lectura o lectura-escritura)
4. Establecer fecha de caducidad

5. Guardar la clave API de forma segura (se muestra solo una vez)

Casos de Uso de la API:

- Monitoreo automatizado del uso
- Integración con sistemas de facturación internos
- Recargas programáticas
- Aprovisionamiento de servicios a través de scripts
- Exportaciones de datos para análisis

Vea Documentación de la API para detalles de los puntos finales y ejemplos.

Preguntas Frecuentes

P: ¿Por qué no puedo ver todos mis servicios?

R: Asegúrese de haber iniciado sesión con la cuenta correcta. Si tiene múltiples cuentas de cliente, cada una tiene servicios separados. Contacte al soporte para fusionar cuentas si es necesario.

P: Mi pago falló pero me cobraron. ¿Qué debo hacer?

R: Verifique su estado de cuenta bancario en busca de cargos pendientes. Si el cargo aparece pero la factura sigue mostrando como no pagada, contacte al soporte con el número de referencia de la transacción.

P: ¿Cómo cancelo un servicio?

R: Navegue a los detalles del servicio y haga clic en "Cancelar Servicio" (si el autoservicio está habilitado). De lo contrario, contacte al servicio de atención al cliente para procesar la cancelación.

P: ¿Puedo transferir un servicio a otra persona?

R: No, las transferencias de servicio requieren asistencia del servicio de atención al cliente por razones de seguridad y cumplimiento.

P: ¿Por qué mi saldo de datos disminuye más rápido de lo esperado?

R: Verifique las actualizaciones de aplicaciones en segundo plano, la calidad de transmisión de video y las copias de seguridad automáticas en la nube. Revise el desglose de uso en el portal para un consumo detallado.

P: Perdí mi dispositivo 2FA. ¿Cómo recupero el acceso?

R: Use los códigos de respaldo para iniciar sesión, luego desactive y vuelva a habilitar 2FA. Si no tiene códigos de respaldo, contacte al soporte para un restablecimiento de 2FA (requiere verificación de identidad).

P: ¿Puedo pagar una factura sin iniciar sesión?

R: El pago de facturas como invitado puede estar disponible a través de un enlace directo de la factura. De lo contrario, se requiere inicio de sesión por razones de seguridad.

P: ¿Cómo descargo todas mis facturas a la vez?

R: Actualmente se requiere descargar cada factura individualmente. Para descargas masivas, contacte al soporte o use la API si está disponible.

Documentación Relacionada

- Atención al Cliente - Impersonación de Usuario - Acceso de personal para resolución de problemas
- Métodos de Pago - Gestión de tarjetas y pagos
- Recarga y Recarga - Compra de crédito adicional
- Uso del Servicio - Comprender el seguimiento de saldos
- Resumen de Facturación - Conceptos de facturación y facturas
- Flujos de Autenticación - Inicio de sesión y seguridad
- 2FA - Configuración de autenticación de dos factores
- Modificaciones de Servicios - Agregar características y complementos

Sistema de Base de Conocimientos de Preguntas Frecuentes (FAQ)

El sistema de FAQ (Preguntas Frecuentes) proporciona una base de conocimientos de autoservicio tanto para clientes como para el personal, reduciendo el volumen de tickets de soporte y permitiendo una resolución más rápida de problemas.

Véase también: Tickets de Soporte para la gestión de tickets, Atención al Cliente para funciones de soporte al cliente.

Descripción General

El sistema de FAQ permite:

1. **Autoservicio**: Los clientes encuentran respuestas sin crear tickets
2. **Base de Conocimientos**: Repositorio centralizado de soluciones
3. **Recurso para el Personal**: Referencia rápida para el personal de soporte
4. **Mejora Continua**: Rastrear cuáles FAQs son más útiles
5. **Multi-Categoría**: Organizar FAQs por tema y etiquetas
6. **Control de Visibilidad**: FAQs públicas vs internas
7. **Analíticas**: Conteos de vistas y calificaciones de utilidad

Características Clave

Gestión de FAQs

- **Crear FAQs**: Escribir preguntas y respuestas detalladas

- **Contenido Rico**: Formato Markdown, imágenes, bloques de código
- **Categorización**: Organizar por categoría y etiquetas
- **Visibilidad**: Controlar acceso solo para clientes o personal
- **Búsqueda**: Búsqueda de texto completo en títulos y contenido
- **Versionado**: Rastrear cambios y actualizaciones a lo largo del tiempo

Analíticas

- **Seguimiento de Vistas**: Contar cuántas veces se visualiza la FAQ
- **Utilidad**: Los clientes califican si la FAQ fue útil
- **FAQs Populares**: Preguntas más vistas
- **Más Útiles**: Soluciones mejor calificadas
- **Bajo Rendimiento**: FAQs marcadas como no útiles (necesitan mejora)

Organización

- **Categorías**: Agrupar FAQs relacionadas (Facturación, Técnica, Cuenta, etc.)
- **Etiquetas**: Múltiples etiquetas por FAQ para la categorización cruzada
- **Búsqueda**: Búsqueda rápida por palabra clave
- **Filtrado**: Filtrar por categoría, etiquetas, visibilidad
- **Ordenación**: Por fecha, vistas, utilidad, alfabéticamente

Estructura de la FAQ

Campos Requeridos

- **Pregunta**: Pregunta clara y concisa (el título de la FAQ)
- **Respuesta**: Respuesta detallada con instrucciones paso a paso
- **Categoría**: Categorización principal
- **Visibilidad**: Visible para clientes o solo para personal

Campos Opcionales

- **Etiquetas**: Palabras clave de categorización adicionales

- **FAQs Relacionadas**: Enlaces a preguntas similares (función futura)
- **Adjuntos**: Capturas de pantalla, documentos, videos (función futura)
- **Autor**: Miembro del personal que creó la FAQ
- **Última Actualización**: Cuándo se modificó por última vez la FAQ

Ejemplo de FAQ

Pregunta: ¿Cómo puedo verificar mi uso de datos?

Categoría: Gestión de Cuenta

Etiquetas: datos, uso, saldo, autoservicio

Respuesta:

```
Puedes verificar tu uso actual de datos de varias maneras:

## A través del Portal de Autoservicio

1. Inicia sesión en tu cuenta en portal.example.com
2. Ve a la pestaña **Servicios**
3. Haz clic en tu servicio móvil
4. Visualiza la sección **Saldo**
5. Tu uso de datos se muestra como "Datos Usados / Datos Totales"

## A través de la Aplicación Móvil

1. Abre la aplicación móvil
2. Toca **Mis Servicios**
3. Selecciona tu línea móvil
4. El uso de datos se muestra en la pantalla de resumen

## A través de SMS

Envía **BAL** al 123456 para recibir tu saldo actual por SMS.

## ¿Necesitas Ayuda?

Si no puedes ver tu uso, por favor contacta al soporte en
support@example.com
```

Visibilidad: Visible para Clientes

Visualización y Edición de FAQs

Al visualizar una FAQ, todo el formato Markdown se renderiza como HTML enriquecido con el estilo adecuado para encabezados, listas, bloques de código y enlaces.

Figura 1: Vista detallada de FAQ mostrando contenido Markdown renderizado con encabezados, listas y bloques de código formateados

La página de detalle de la FAQ proporciona:

- **Contenido Renderizado**: Markdown convertido automáticamente a HTML formateado
- **Estadísticas**: Conteo de vistas y calificaciones de utilidad
- **Modo de Edición**: Edición en línea con soporte para markdown
- **Alternar Visibilidad**: Mostrar/ocultar a los clientes

- **Acciones Rápidas**: Copiar enlace, eliminar FAQ

Vista del Cliente

Los clientes ven una vista limpia y formateada de las FAQs sin capacidades de edición:

Figura 2: Vista de FAQ orientada al cliente con calificación de utilidad y formato limpio

Creación de FAQs

A través de la Interfaz Web

1. Navega a **Soporte → FAQs**
2. Haz clic en **Crear FAQ**
3. Completa los detalles de la FAQ:
    - Pregunta (título)

- Respuesta (contenido con markdown)
- Categoría
- Etiquetas (separadas por comas)
- Visibilidad (casilla de verificación visible para clientes)

4. Haz clic en **Crear**

A través de la API

```
POST /crm/faq/
```

Permiso Requerido: `create_faq`

Solicitud:

```
{
  "question": "¿Cómo restablezco mi contraseña?",
  "answer": "Para restablecer tu contraseña:\n1. Ve a la página de
inicio de sesión\n2. Haz clic en 'Olvidé mi Contraseña'\n3.
Ingresa tu correo electrónico\n4. Revisa tu bandeja de entrada
para el enlace de restablecimiento\n5. Haz clic en el enlace y
establece una nueva contraseña",
  "category": "Cuenta",
  "tags": "contraseña,inicio de sesión,cuenta,restablecer",
  "is_visible_to_customer": true
}
```

Respuesta:

```
{
  "faq_id": 42,
  "question": "¿Cómo restablezco mi contraseña?",
  "answer": "Para restablecer tu contraseña:\n1. Ve a la página de
inicio de sesión...",
  "category": "Cuenta",
  "tags": "contraseña,inicio de sesión,cuenta,restablecer",
  "is_visible_to_customer": true,
  "view_count": 0,
  "helpful_count": 0,
  "not_helpful_count": 0,
  "created": "2026-01-07T10:00:00Z",
  "last_modified": "2026-01-07T10:00:00Z",
  "created_by_user": {
    "id": 1,
    "username": "admin",
    "first_name": "Admin",
    "last_name": "User"
  }
}
```


Edición de FAQs

Actualización de Contenido

1. Abre la FAQ de la lista
2. Haz clic en **Editar**
3. Modifica la pregunta, respuesta o metadatos
4. Haz clic en **Guardar**

Los cambios se rastrean con:

- Marca de tiempo de última modificación
- Último modificado por usuario
- Historial de cambios (registro de auditoría)

Alternar Visibilidad

Controla si los clientes pueden ver una FAQ:

1. Abre la FAQ
2. Haz clic en el interruptor de **Visibilidad**
3. Elige:
    - **Visible para Clientes**: Aparece en el portal del cliente
    - **Solo Personal**: Solo visible para el personal de soporte
4. Guarda los cambios

Casos de Uso para Solo Personal:

- Procedimientos y flujos de trabajo internos
- Guías de solución de problemas con información sensible
- Borradores de FAQs a la espera de aprobación
- Soluciones obsoletas mantenidas para referencia

Categorías de FAQs

Categorías Estándar

Organiza las FAQs en grupos lógicos:

- **Gestión de Cuenta**: Inicio de sesión, contraseñas, perfiles
- **Facturación**: Facturas, pagos, cargos
- **Soporte Técnico**: Problemas de servicio, solución de problemas
- **Características**: Cómo usar características específicas
- **Servicios Móviles**: Tarjetas SIM, datos, roaming
- **Servicios de Internet**: Conectividad, velocidades, equipos
- **Servicios VoIP**: Llamadas, buzón de voz, características
- **Servicios IPTV**: Canales, streaming, dispositivos

Creación de Categorías

Las categorías se crean automáticamente cuando se utilizan en una FAQ. Para estandarizar:

1. Usa nombres de categoría consistentes
2. Sigue convenciones de nomenclatura
3. Revisa la lista de categorías periódicamente
4. Fusiona categorías duplicadas/similares

Gestión de Categorías

Ver Todas las Categorías:

```
GET /crm/faq/categories
```

Devuelve la lista de todas las categorías con conteos de FAQs.

Vista del Cliente:

```
GET /crm/faq/customer/categories
```

Devuelve solo las categorías con FAQs visibles.

Etiquetas

Propósito

Las etiquetas proporcionan organización transversal:

- Una sola FAQ puede tener múltiples etiquetas
- Las etiquetas complementan la estructura de categorías
- Permiten filtrado y búsqueda flexibles
- Ayudan a los clientes a encontrar contenido relacionado

Ejemplos de Etiquetas

- **Técnico:** solución de problemas, error, bug, arreglo, restablecer
- **Cuenta:** inicio de sesión, contraseña, perfil, configuraciones, seguridad
- **Facturación:** factura, pago, cargo, reembolso, disputa
- **Servicio:** activación, desactivación, actualización, cambio
- **Plataforma:** móvil, web, app, portal, autoservicio

Mejores Prácticas

1. **Nomenclatura Consistente**: Usa minúsculas, forma singular
2. **Etiquetas Específicas**: "restablecimiento-de-contraseña" no solo "contraseña"
3. **Evitar Redundancia**: No duplicar la categoría en etiquetas
4. **Limitar Conteo**: 3-5 etiquetas por FAQ
5. **Estandarizar**: Mantener un glosario de etiquetas

Portal de FAQs para Clientes

Acceso a FAQs

Los clientes acceden a las FAQs a través de:

1. **Portal de Autoservicio**: Sección de FAQ dedicada
2. **Durante la Creación de Tickets**: Sugerencias "Verifica estas FAQs primero"
3. **Búsqueda**: La búsqueda global incluye contenido de FAQs
4. **Correo Electrónico**: Enlaces en respuestas automatizadas

Visualización de FAQs

La vista del cliente muestra:

- **Pregunta**: Título/pregunta de la FAQ
- **Respuesta**: Contenido completo formateado

- **Categoría**: Para navegar por FAQs relacionadas
- **Etiquetas**: Navegación adicional
- **Calificación Útil**: Retroalimentación de pulgar arriba/abajo
- **FAQs Relacionadas**: Preguntas similares (futuro)

Búsqueda y Filtrado

Los clientes pueden:

- **Buscar**: Búsqueda de texto completo en todas las FAQs visibles
- **Filtrar por Categoría**: Ver FAQs en una categoría específica
- **Filtrar por Etiqueta**: Encontrar FAQs con etiquetas específicas
- **Ordenar**: Por popularidad, recencia o alfabéticamente
- **Navegar**: Explorar todas las FAQs o listados de categorías

Calificación de FAQs

Los clientes proporcionan retroalimentación:

1. Lee la FAQ
2. Desplázate hasta el final
3. Haz clic en **¿Fue útil?**
4. Selecciona **Sí** o **No**
5. Calificación registrada

Las calificaciones ayudan a identificar:

- **FAQs Útiles**: Altas votaciones de sí
- **Necesita Mejora**: Altas votaciones de no
- **Contenido Obsoleto**: Disminución de utilidad
- **Temas Populares**: Altas vistas + altas calificaciones

Herramientas de FAQ para el Personal

FAQs Internas

FAQs solo para personal sobre:

- **Procedimientos**: Flujos de trabajo y procesos internos
- **Solución de Problemas**: Pasos de diagnóstico avanzados
- **Escalación**: Cuándo y cómo escalar problemas
- **Herramientas**: Cómo usar sistemas internos
- **Políticas**: Políticas y directrices de la empresa

Referencia Rápida

El personal puede:

- **Buscar Todas las FAQs**: Incluyendo contenido solo para personal
- **Ver Analíticas**: Ver cuáles FAQs son más útiles
- **Copiar Enlaces**: Compartir enlaces de FAQs con clientes
- **Sugerir Ediciones**: Proponer mejoras a las FAQs
- **Crear desde Ticket**: Convertir la resolución de tickets en FAQ

Analíticas de FAQ

Rastrear el rendimiento de las FAQs:

Conteo de Vistas:

- Vistas totales por FAQ
- Vistas a lo largo del tiempo
- FAQs más populares
- FAQs en tendencia (picos repentinos)

Utilidad:

- Votos útiles (pulgar arriba)
- Votos no útiles (pulgar abajo)
- Proporción de utilidad
- FAQs más/menos útiles

Puntos finales de la API:

```
GET /crm/faq/popular?limit=10
GET /crm/faq/helpful?limit=10
```

Mejora Continua

Usar analíticas para:

1. **Identificar Brechas**: Búsquedas comunes sin resultados
2. **Actualizar Contenido**: Mejorar FAQs con bajas calificaciones
3. **Retirar Obsoletas**: Eliminar o actualizar FAQs obsoletas
4. **Promover Mejores**: Destacar FAQs altamente calificadas
5. **Crear Nuevas**: Agregar FAQs para tickets recurrentes

Formato Markdown

Las respuestas de FAQ admiten markdown para un formato rico:

Formato de Texto

```
**Texto en negrita** para énfasis
*Texto en cursiva* para énfasis sutil
~~Tachado~~ para información obsoleta
`código en línea` para términos técnicos
```

Encabezados

```
## Encabezado de Sección
### Encabezado de Subsección
#### Encabezado Menor
```

Listas

```
Lista Ordenada:
1. Primer paso
2. Segundo paso
3. Tercer paso

Lista No Ordenada:
- Punto uno
- Punto dos
- Punto tres
```

Enlaces

```
[Texto del enlace](https://example.com)
[Correo electrónico de soporte](mailto:support@example.com)
```

Bloques de Código

````
```bash
comando --con-banderas
```

```json
{
  "example": "data"
}
```
````

Imágenes

```
![Texto alternativo](https://example.com/image.png)
```

Tablas

```
| Columna 1 | Columna 2 |
|-----------|-----------|
| Datos 1   | Datos 2   |
```

Citas

```
> Nota o consejo importante
> Continúa en múltiples líneas
```

Referencia de API

Obtener Todas las FAQs (Personal)

```
GET /crm/faq/
```

Permiso Requerido: `view_faq`

Parámetros de Consulta:

- `page`: Número de página (predeterminado: 1)
- `per_page`: Resultados por página (predeterminado: 20)
- `sort`: Campo de ordenación (creado, conteo_de_vistas, conteo_de_utilidad)
- `order`: Orden de clasificación (asc, desc)
- `search`: Consulta de búsqueda
- `category`: Filtrar por categoría
- `tags`: Filtrar por etiquetas (separadas por comas)

- `is_visible_to_customer`: Filtrar por visibilidad (true/false)

Obtener FAQs del Cliente

```
GET /crm/faq/customer
```

Autenticación Requerida: Sí **Permiso:** Cualquier usuario autenticado

Devuelve solo FAQs visibles para clientes.

Obtener FAQ Única

```
GET /crm/faq/faq_id/{faq_id}
```

Vista del Personal: No incrementa el conteo de vistas

```
GET /crm/faq/customer/faq_id/{faq_id}
```

Vista del Cliente: Incrementa el conteo de vistas

Actualizar FAQ

```
PATCH /crm/faq/faq_id/{faq_id}
```

Permiso Requerido: `update_faq`

Eliminar FAQ

```
DELETE /crm/faq/faq_id/{faq_id}
```

Permiso Requerido: `delete_faq`

Realiza una eliminación suave (marca como eliminada, preserva datos).

Alternar Visibilidad

```
POST /crm/faq/faq_id/{faq_id}/visibility
```

Permiso Requerido: `manage_faq_visibility`

Solicitud:

```
{
  "is_visible_to_customer": true
}
```

Marcar FAQ como Útil

```
POST /crm/faq/faq_id/{faq_id}/helpful
```

Autenticación Requerida: Sí **Permiso:** Cualquier usuario autenticado

Solicitud:

```
{
  "is_helpful": true
}
```

Obtener FAQs Populares

```
GET /crm/faq/popular?limit=10
```

Devuelve FAQs ordenadas por conteo de vistas.

Obtener FAQs Más Útiles

```
GET /crm/faq/helpful?limit=10
```

Devuelve FAQs ordenadas por conteo de utilidad.

Mejores Prácticas

Redacción de Contenido

1. **Preguntas Claras**: Usa el lenguaje natural que los clientes usarían
2. **Respuestas Completas**: Incluye todos los pasos, sin suposiciones
3. **Capturas de Pantalla**: Agrega ayudas visuales para procedimientos complejos
4. **Enlaces**: Referencia FAQs y recursos relacionados
5. **Pruebas**: Verifica que las instrucciones sean precisas

Organización

1. **Categorías Consistentes**: Usa nombres de categoría estandarizados
2. **Etiquetas Específicas**: Usa etiquetas descriptivas y buscables
3. **Evitar Duplicación**: Fusiona FAQs similares
4. **Referencia Cruzada**: Enlaza FAQs relacionadas entre sí
5. **Revisión Regular**: Actualiza contenido obsoleto trimestralmente

Mantenimiento

1. **Monitorear Analíticas**: Rastrear vistas y utilidad
2. **Actualizar Regularmente**: Mantener el contenido actual
3. **Retirar FAQs Antiguas**: Eliminar información obsoleta
4. **Incorporar Retroalimentación**: Mejorar según calificaciones
5. **Crear desde Tickets**: Convertir problemas comunes en FAQs

Solución de Problemas

Problema: FAQ no aparece para los clientes

- Solución: Verifica que la configuración "Visible para Clientes" esté habilitada
- Solución: Verifica que la FAQ no esté marcada como eliminada
- Solución: Limpia la caché del portal del cliente

Problema: La búsqueda no encuentra la FAQ

- Solución: Verifica la ortografía y prueba con diferentes palabras clave
- Solución: Verifica que la FAQ esté en la categoría correcta
- Solución: Agrega más etiquetas relevantes a la FAQ

Problema: Las analíticas no se actualizan

- Solución: Espera a que se actualice la caché (puede tardar unos minutos)
- Solución: Verifica que el seguimiento de vistas esté habilitado
- Solución: Verifica que el cliente esté autenticado al visualizar

Problema: No se puede editar la FAQ

- Solución: Verifica que tengas el permiso `update_faq`
- Solución: Verifica que la FAQ no esté bloqueada por otro usuario
- Solución: Confirma que la FAQ exista y no esté eliminada

Sistema de Tickets de Soporte

El Sistema de Tickets de Soporte de OmniCRM proporciona una plataforma integral para gestionar solicitudes de soporte al cliente, rastrear problemas y mantener un historial de comunicación detallado con los clientes.

Vea también: Preguntas Frecuentes para la base de conocimientos de autoservicio, Atención al Cliente para suplantación de identidad y solución de problemas, Clientes para la gestión de clientes.

Figura 1: Lista de tickets de soporte que muestra todos los tickets con filtrado, búsqueda, estado, prioridad y personal asignado

Descripción General

El sistema de tickets permite:

1. **Soporte Centralizado**: Rastrear todos los problemas de los clientes en un solo lugar

2. **Integración de Correo Electrónico**: Crear tickets automáticamente a partir de correos electrónicos de clientes
3. **Colaboración**: Varios miembros del personal pueden trabajar en tickets juntos
4. **Base de Conocimientos**: Vincular preguntas frecuentes a problemas comunes para una resolución más rápida
5. **Seguimiento de SLA**: Monitorear la antigüedad del ticket y los tiempos de respuesta
6. **Portal del Cliente**: Los clientes pueden ver y responder a sus tickets
7. **Registro de Auditoría**: Historial completo de todas las interacciones de tickets

Características Clave

Gestión de Tickets

- **Crear Tickets**: Creación manual o automática desde correo electrónico
- **Actualizar Tickets**: Agregar comentarios, cambiar estado, reasignar
- **Adjuntos**: Subir archivos (imágenes, documentos, registros)
- **Etiquetas**: Categorizar tickets para informes y filtrado
- **Búsqueda**: Búsqueda de texto completo en tickets, actualizaciones y adjuntos
- **Filtrado**: Filtrar por estado, prioridad, asignado, cliente, servicio
- **Vistas en Tiempo Real**: Rastrear quién está viendo actualmente cada ticket
- **Última Actividad**: Ver cuándo se vieron o actualizaron los tickets por última vez

Comunicación

- **Notificaciones por Correo Electrónico**: Notificaciones automáticas sobre actualizaciones de tickets
- **Notificaciones en la Aplicación**: Notificaciones en tiempo real en la interfaz de OmniCRM

- **@Menciones**: Mencionar a miembros del personal con @nombredeusuario para notificarles
- **Suscriptores**: Agregar personal y contactos a las notificaciones de tickets
- **Visibilidad del Cliente**: Controlar qué actualizaciones pueden ver los clientes
- **Notas Internas**: Agregar notas privadas no visibles para los clientes
- **Hilos de Correo Electrónico**: Seguimiento automático de hilos de correo electrónico
- **Visualizadores Activos**: Ver quién está viendo actualmente el ticket en tiempo real

Integración de Servicios

- **Vincular a Servicios**: Asociar tickets con servicios específicos del cliente
- **Contexto del Servicio**: Ver detalles del servicio, uso e historial desde el ticket
- **Acciones Rápidas**: Recargar, ver complementos, editar servicio directamente desde el ticket

Ciclo de Vida del Ticket

1. Creación de Ticket

A través de la Interfaz Web:

1. Navegar a **Soporte → Tickets**
2. Hacer clic en **Crear Ticket**
3. Completar los detalles del ticket:
    - Título (requerido)
    - Descripción (requerido)
    - Cliente (opcional, puede asignarse más tarde)
    - Servicio (opcional, vincula a un servicio específico del cliente)
    - Prioridad (Baja, Media, Alta, Urgente)
    - Estado (Abierto, En Progreso, En Espera, Resuelto, Cerrado)

    - Etiquetas (para categorización)
4. Hacer clic en **Crear**

A través de Correo Electrónico:

Los clientes pueden crear tickets enviando un correo a su dirección de soporte. El sistema automáticamente:

- Crea un ticket a partir del asunto y cuerpo del correo
- Vincula al cliente si el correo del remitente coincide con un contacto
- Agrega los archivos adjuntos del correo al ticket
- Envía una confirmación al cliente

A través de API:

```
POST /crm/ticket/
```

Solicitud:

```
{
  "title": "No se puede acceder al correo de voz",
  "description": "El cliente informa que el correo de voz no
funciona desde ayer",
  "customer_id": 290,
  "service_id": 167,
  "priority": "High",
  "status": "Open",
  "tags": "technical,voicemail"
}
```

2. Asignación de Ticket

Los tickets pueden ser asignados al personal de soporte para su manejo:

1. **Asignación Manual**: Hacer clic en "Asignar" y seleccionar al miembro del personal
2. **Asignación Automática**: Asignar automáticamente según reglas (característica futura)

3. **Reasignación**: Transferir tickets entre miembros del personal
4. **Sin Asignar**: Los tickets pueden permanecer sin asignar en la cola

Cuando se asigna un ticket:

- El personal asignado recibe una notificación por correo electrónico
- El personal se suscribe automáticamente a las actualizaciones del ticket
- El ticket aparece en su vista de "Mis Tickets"

3. Trabajando en Tickets

Agregando Actualizaciones:

1. Abrir el ticket
2. Hacer clic en **Agregar Actualización** o responder al ticket
3. Ingresar su mensaje
4. Usar **@nombredeusuario** para mencionar y notificar a miembros específicos del personal (por ejemplo, "@john, ¿puedes ayudar con esto?")
5. Elegir visibilidad:
   - **Visible para el Cliente**: El cliente verá esta actualización
   - **Solo Interno**: Nota privada para el personal
6. Hacer clic en **Enviar**

@Menciones:

Cuando mencionas a un miembro del personal usando @nombredeusuario en una actualización de ticket:

- Reciben una notificación en la aplicación
- Se suscriben automáticamente al ticket
- La notificación incluye un enlace directamente al ticket
- Funciona tanto en actualizaciones visibles para el cliente como internas

Adjuntando Archivos:

1. Hacer clic en el botón **Adjuntar Archivos**
2. Seleccionar archivos (máx. 10MB cada uno)

3. Los archivos se suben y se vinculan al ticket
4. Formatos soportados: PDF, imágenes, documentos, archivos comprimidos

Cambiando Estado:

- **Abierto**: Nuevo ticket, esperando respuesta inicial
- **En Progreso**: El personal está trabajando activamente en el problema
- **En Espera**: Esperando respuesta del cliente o dependencia externa
- **Resuelto**: El problema ha sido solucionado, esperando confirmación
- **Cerrado**: El ticket está completo y archivado

Estableciendo Prioridad:

- **Urgente**: Problema crítico que afecta el servicio (SLA: 1 hora)
- **Alto**: Problema importante, servicio degradado (SLA: 4 horas)
- **Medio**: Problema estándar (SLA: 24 horas)
- **Bajo**: Problema menor o consulta (SLA: 48 horas)

4. Resolución de Ticket

Al resolver un ticket:

1. Agregar la actualización final explicando la resolución
2. Cambiar el estado a **Resuelto**
3. El cliente recibe una notificación
4. El cliente puede reabrir si el problema persiste
5. Después de la confirmación, cambiar a **Cerrado**

Vista de Detalles del Ticket

Al ver un ticket, se ve el hilo de conversación completo con todas las actualizaciones, adjuntos y una barra lateral integral que muestra información del cliente y del servicio.

Figura 2: Página de detalles del ticket que muestra el hilo de conversación, actualizaciones y metadatos del ticket

Barra Lateral de Detalles del Ticket

Cada ticket muestra una barra lateral integral con información del cliente, detalles del servicio y acciones rápidas:

Figura 3: Barra lateral del ticket con información del cliente, detalles del servicio, personal asignado y botones de acción rápida

La barra lateral incluye:

Información del Cliente

- **Nombre del Cliente**: Con enlace al perfil del cliente
- **Correo Electrónico de Contacto**: Correo electrónico principal donde se envían las respuestas
- **Editar Cliente**: Edición rápida de la asignación del cliente

Información del Servicio

Cuando un servicio está vinculado:

- **ID del Servicio**: Identificador único
- **Nombre del Servicio**: Nombre descriptivo
- **Fechas del Servicio**: Fechas de inicio y fin

- **Auto-Renovación**: Estado de renovación
- **Estado del Servicio**: Activo, Inactivo, Suspendido
- **Tipo de Servicio**: Móvil, IPTV, Internet, VoIP
- **Acciones Rápidas:**
  - **Agregar Complemento**: Agregar paquetes/características al servicio
  - **Eliminar**: Eliminar servicio (con confirmación)
  - **Editar Servicio**: Modal de gestión completa del servicio

Metadatos del Ticket

- **Número de Ticket**: Referencia única (por ejemplo, TKT-2026-00042)
- **Fecha de Creación**: Cuando se abrió el ticket
- **Última Actualización**: Marca de tiempo de la actividad más reciente
- **Asignado a**: Propietario actual del ticket
- **Nivel de Prioridad**: Con codificación de colores
- **Estado Actual**: Con insignia de estado
- **Etiquetas**: Etiquetas de categorización

Suscriptores

Lista de usuarios y contactos que reciben notificaciones:

- **Suscritos Automáticamente**: Creador y asignado
- **Suscripciones Manuales**: Otros miembros del personal o contactos
- **Agregar/Eliminar**: Gestionar lista de suscriptores

Integración de Correo Electrónico

Procesamiento de Correo Electrónico Entrante

El sistema procesa correos electrónicos de soporte entrantes:

1. **Correo Recibido**: El correo llega a la dirección de soporte
2. **Analizar Correo**: Extraer remitente, asunto, cuerpo, adjuntos

3. **Identificar Cliente**: Coincidir el correo del remitente con un contacto
4. **Crear o Actualizar**: Crear un nuevo ticket o agregar a un hilo existente
5. **Notificar al Personal**: Alertar al personal o equipo asignado
6. **Enviar Confirmación**: Confirmar recepción al cliente

Encabezados de Correo Usados:

- `From`: Dirección de correo electrónico del cliente
- `Subject`: Título del ticket (con número de ticket si es respuesta)
- `In-Reply-To`: Seguimiento de hilo para respuestas
- `Message-ID`: Identificador único del mensaje

Notificaciones Salientes

Las actualizaciones del personal generan notificaciones por correo electrónico a:

- **Cliente**: Si la actualización está marcada como visible para el cliente
- **Suscriptores**: Todo el personal y contactos suscritos
- **Asignado**: Cuando se asigna o reasigna un ticket

Variables de Plantilla de Correo Electrónico:

- `{{ticket_number}}`: Referencia del ticket
- `{{customer_name}}`: Nombre del cliente
- `{{update_text}}`: Contenido de la actualización
- `{{staff_name}}`: Nombre del personal que agregó la actualización
- `{{ticket_url}}`: Enlace directo al ticket

Acceso al Portal del Cliente

Los clientes pueden ver sus tickets a través del portal de autoservicio:

1. **Iniciar Sesión**: Autenticarse en el portal del cliente
2. **Navegar**: Ir a "Soporte" o "Mis Tickets"
3. **Ver Tickets**: Ver todos sus tickets y estado

4. **Ver Detalles**: Leer el historial completo del ticket (solo actualizaciones visibles para el cliente)
5. **Responder**: Agregar respuestas a tickets abiertos
6. **Adjuntos**: Ver y descargar adjuntos
7. **Estado**: Ver el estado actual del ticket y prioridad

Restricciones de Vista del Cliente:

- Los clientes solo ven sus propios tickets
- Las actualizaciones solo internas están ocultas
- Los adjuntos solo para el personal no son accesibles
- Los cambios del sistema (asignaciones, estado) se resumen

Suscriptores y Notificaciones

Suscripción Automática

Suscritos automáticamente cuando:

- **Creando un ticket**: Creador del ticket
- **Asignado a un ticket**: Asignado actual
- **Mencionado**: Personal mencionado en actualizaciones usando la sintaxis @nombredeusuario

Suscripción Manual

Agregar suscriptores:

1. Abrir ticket
2. Hacer clic en la sección **Suscriptores**
3. Hacer clic en **Agregar Suscriptor**
4. Seleccionar usuario o contacto
5. Hacer clic en **Agregar**

Eliminar suscriptores:

1. Hacer clic en X junto al nombre del suscriptor
2. Confirmar eliminación
3. Ya no recibirán notificaciones

Preferencias de Notificación

El personal recibe notificaciones a través de múltiples canales:

- **Notificaciones en la Aplicación**: Notificaciones en tiempo real en el menú desplegable de notificaciones
- **Notificaciones por Correo Electrónico**: Notificaciones por correo electrónico configurables para actualizaciones de tickets
- **Notificaciones de @Mención**: Alertas inmediatas cuando se les menciona en actualizaciones de tickets
- **Frecuencia**: Inmediata, resumen horario, resumen diario
- **Tipos**: Qué eventos desencadenan notificaciones (nuevos tickets, actualizaciones, asignaciones, menciones)

Búsqueda y Filtrado

Búsqueda Rápida

La barra de búsqueda global busca:

- Títulos y descripciones de tickets
- Números de tickets
- Nombres de clientes
- Contenido de actualizaciones
- Nombres de adjuntos

Filtrado Avanzado

Filtrar tickets por:

- **Estado**: Abierto, En Progreso, Resuelto, etc.

- **Prioridad**: Urgente, Alta, Media, Baja
- **Asignado a**: Miembro específico del personal o sin asignar
- **Cliente**: Cliente específico
- **Servicio**: Servicio específico
- **Etiquetas**: Una o más etiquetas
- **Rango de Fechas**: Creado o actualizado dentro de fechas
- **Tiene Adjuntos**: Tickets con/sin archivos

Filtros Guardados

Crear y guardar combinaciones de filtros personalizadas:

1. **Aplicar Filtros**: Establecer sus filtros deseados
2. **Guardar Filtro**: Hacer clic en el botón "Guardar Filtro"
3. **Nombrar Filtro**: Darle un nombre descriptivo
4. **Acceso Rápido**: Aparece en el menú desplegable de filtros

Filtros Guardados Comunes:

- "Mis Tickets Abiertos": Asignados a mí, estado Abierto o En Progreso
- "Cola Urgente": Prioridad Urgente, estado Abierto
- "Necesita Respuesta": Estado En Espera, esperando al cliente
- "Recientemente Cerrados": Cerrados en los últimos 7 días

Informes y Análisis

Métricas de Tickets

Rastrear el rendimiento de soporte:

- **Total de Tickets**: Creado, resuelto, cerrado
- **Tiempo Promedio de Resolución**: Tiempo para resolver por prioridad
- **Tiempo de Primera Respuesta**: Tiempo hasta la primera actualización del personal
- **Tickets por Estado**: Distribución entre estados

- **Tickets por Prioridad**: Urgente vs Alta vs Media vs Baja
- **Tickets por Personal**: Carga de trabajo por miembro del personal
- **Tickets por Etiqueta**: Categorías de problemas comunes

Cumplimiento de SLA

Monitorear la adherencia al acuerdo de nivel de servicio:

- **Dentro del SLA**: Tickets resueltos dentro del objetivo
- **SLA Violado**: Tickets que exceden los tiempos objetivos
- **En Riesgo**: Tickets que se acercan a la fecha límite del SLA
- **Respuesta Promedio**: Tiempos de respuesta reales vs objetivos

Perspectivas del Cliente

Analizar patrones de soporte al cliente:

- **Principales Creadores de Tickets**: Clientes con más tickets
- **Problemas Recurrentes**: Clientes con problemas recurrentes
- **Relacionados con el Servicio**: Tickets por tipo de servicio
- **Éxito en la Resolución**: Tasa de resolución en el primer contacto

Integración con Servicios

Vinculación de Servicios a Tickets

Cuando un ticket se relaciona con un servicio específico:

1. **Seleccionar Servicio**: Elegir de los servicios del cliente
2. **Contexto del Servicio**: Ver detalles del servicio en la barra lateral
3. **Acciones R��pidas**: Gestionar el servicio desde el ticket
4. **Historial del Servicio**: Ver tickets y cambios relacionados

Acciones Rápidas del Servicio

Desde la barra lateral del ticket:

Complementos:

1. Hacer clic en el botón **Agregar Complemento**
2. Se abre el modal de complementos
3. Seleccionar paquetes para agregar
4. Confirmar y aplicar
5. Servicio actualizado automáticamente

Eliminar Servicio:

1. Hacer clic en el botón **Eliminar**
2. Aparece el cuadro de diálogo de confirmación
3. Confirmar eliminación
4. Servicio eliminado
5. Ticket actualizado con la acción

Editar Servicio:

1. Hacer clic en el botón **Editar Servicio**
2. Se abre el modal completo del servicio
3. Realizar cambios en la configuración del servicio
4. Guardar cambios
5. Volver al ticket

Mejores Prácticas

Creación de Tickets

1. **Títulos Descriptivos**: Resumen claro y conciso del problema
2. **Descripción Detallada**: Incluir toda la información relevante
3. **Prioridad Correcta**: Establecer según el impacto real
4. **Vincular Servicio**: Siempre vincular al servicio afectado si es aplicable

5. **Agregar Etiquetas**: Usar etiquetas consistentes para la categorización

Gestión de Tickets

1. **Actualizaciones Oportunas**: Actualizar tickets regularmente, incluso si "sin progreso"
2. **Precisión del Estado**: Mantener el estado actual (no dejar tickets resueltos abiertos)
3. **Comunicación con el Cliente**: Mantener informados a los clientes sobre el progreso
4. **Notas Internas**: Documentar pasos de solución de problemas internamente
5. **Documentación de Resolución**: Indicar claramente qué solucionó el problema

Transferencia de Conocimientos

1. **Vincular Preguntas Frecuentes**: Referenciar o crear preguntas frecuentes para problemas comunes
2. **Documentar Soluciones**: Agregar detalles de resolución para referencia futura
3. **Etiquetar Apropiadamente**: Facilita encontrar tickets similares
4. **Compartir Aprendizajes**: Discutir resoluciones complejas con el equipo

Experiencia del Cliente

1. **Primer Contacto**: Responder rápidamente para reconocer la recepción
2. **Establecer Expectativas**: Informar al cliente cuándo esperar la resolución
3. **Tono Profesional**: Mantener cortesía y profesionalismo
4. **Lenguaje Claro**: Evitar jerga técnica con los clientes
5. **Confirmar Resolución**: Asegurarse de que el cliente esté satisfecho antes de cerrar

Integración del Sistema de Notificaciones:

- Se agregó soporte de notificación en la aplicación para eventos de tickets

- Funcionalidad de @mención para notificar a miembros específicos del personal
- Integración con la API de Notificaciones para sistemas externos
- Entrega de notificaciones en tiempo real

Colaboración en Tiempo Real:

- Seguimiento de visualizadores activos: ver quién está viendo actualmente cada ticket
- Marcas de tiempo de última actividad para tickets
- Actualizaciones en tiempo real cuando varios usuarios trabajan en el mismo ticket

Comunicación Mejorada:

- Las @menciones suscriben automáticamente a los usuarios mencionados
- Mejores preferencias de notificación con múltiples canales
- Mejor integración entre correo electrónico y notificaciones en la aplicación

Vea también: API de Notificaciones para detalles sobre el envío de notificaciones desde sistemas externos.

Referencia de API

Crear Ticket

```
POST /crm/ticket/
```

Permiso Requerido: `create_ticket`

Obtener Ticket

```
GET /crm/ticket/ticket_id/{ticket_id}
```

Permiso Requerido: `view_ticket`

Actualizar Ticket

```
PATCH /crm/ticket/ticket_id/{ticket_id}
```

Permiso Requerido: `update_ticket`

Eliminar Ticket

```
DELETE /crm/ticket/ticket_id/{ticket_id}
```

Permiso Requerido: `delete_ticket`

Agregar Actualización de Ticket

```
POST /crm/ticket/ticket_id/{ticket_id}/updates
```

Subir Adjunto

```
POST /crm/ticket/ticket_id/{ticket_id}/attachments
```

Permiso Requerido: `upload_attachment`

Suscribirse a Ticket

```
POST /crm/ticket/ticket_id/{ticket_id}/subscribe
```

Permiso Requerido: `subscribe_ticket`

Solución de Problemas

Problema: Correos electrónicos no crean tickets

- Solución: Verificar la configuración de integración de correo electrónico
- Solución: Verificar que el correo del remitente coincida con un contacto del cliente
- Solución: Revisar los registros de procesamiento de correos electrónicos

Problema: El cliente no recibe notificaciones

- Solución: Verificar que el correo de contacto sea correcto
- Solución: Comprobar que la actualización esté marcada como "visible para el cliente"
- Solución: Confirmar que el correo no esté bloqueado/spam

Problema: No se pueden subir adjuntos

- Solución: Verificar el tamaño del archivo (máx. 10MB)
- Solución: Verificar que el tipo de archivo esté permitido
- Solución: Comprobar los permisos del directorio de subidas

Problema: Acciones rápidas del servicio no funcionan

- Solución: Verificar que el servicio esté correctamente vinculado al ticket
- Solución: Comprobar que el usuario tenga los permisos requeridos para el servicio
- Solución: Confirmar que los datos del servicio se hayan cargado correctamente