



# Guia de Operações do OmniCRM

**OmniCRM** é a solução abrangente de BSS/OSS da Omnitouch para provedores de serviços móveis e fixos. Uma plataforma completa que gerencia tudo, desde a integração de clientes até faturamento, provisionamento e suporte - tudo em um sistema integrado.

---

## Começando

### Para a Equipe de Atendimento ao Cliente

#### Seus primeiros passos:

1. [Aprenda a interface](#) - Familiarize-se com a navegação e pesquisa
2. [Crie um cliente](#) - Criação de cliente passo a passo
3. [Adicione um serviço](#) - Provisione seu primeiro serviço
4. [Processar um pagamento](#) - Gerencie pagamentos de clientes
5. [Recarregar serviços](#) - Adicione crédito às contas dos clientes

#### Tarefas diárias:

- [Gerenciamento de serviços](#) - Gerencie serviços de clientes
- [Ver uso](#) - Verifique saldos e uso
- [Modificar serviços](#) - Altere configurações de serviços
- [Gerar faturas](#) - Crie e envie faturas

### Para Administradores de Sistema

#### Configuração e configuração:

1. [Entenda a arquitetura](#) - Visão geral do sistema
2. [Configure o sistema](#) - Configurações do sistema
3. [Crie produtos](#) - Construa seu catálogo
4. [Escreva playbooks](#) - Automatize o provisionamento
5. [Configure usuários](#) - Crie contas e atribua funções

#### Tópicos avançados:

- [Gerenciamento de inventário](#) - Gerencie ativos
- [Personalização](#) - Adapte às suas necessidades
- [Integração de API](#) - Conecte sistemas externos
- [Configuração de segurança](#) - Configure 2FA e permissões

## Para Clientes

### Usando o Portal de Autoatendimento:

- [Acesse sua conta](#) - Faça login e navegue
  - [Ver serviços](#) - Veja seus serviços ativos
  - [Verificar uso](#) - Monitore dados e saldos
  - [Pagar faturas](#) - Faça pagamentos online
  - [Recarregar serviços](#) - Adicione crédito instantaneamente
- 

## Referência Rápida de Tarefas

Eu quero...	Documentação
Criar um novo cliente	<a href="#">Criar Cliente</a>
Adicionar um serviço a um cliente	<a href="#">Adicionar Serviço</a>
Ver uso de serviços e saldos	<a href="#">Uso de Serviço</a>
Processar um pagamento	<a href="#">Processar Pagamento</a>
Gerar uma fatura	<a href="#">Gerenciamento de Faturas</a>
Recarregar um serviço	<a href="#">Recarregar &amp; Recarregar</a>
Gerenciar inventário	<a href="#">Sistema de Inventário</a>
Criar um produto	<a href="#">Ciclo de Vida do Produto</a>
Escrever um playbook de provisionamento	<a href="#">Playbooks Ansible</a>
Configurar contas de usuário	<a href="#">RBAC</a>
Habilitar 2FA	<a href="#">Autenticação de Dois Fatores</a>
Pesquisar qualquer coisa	<a href="#">Pesquisa Global</a>
Entender o sistema	<a href="#">Arquitetura do Sistema</a>
Usar a API	<a href="#">Documentação da API</a>

---

## Biblioteca de Documentação

### Leitura Essencial

- [Arquitetura do Sistema](#) - Visão técnica completa com diagramas
- [Guia do Ciclo de Vida do Produto](#) - Gestão de produtos de ponta a ponta
- [Guia de Playbooks Ansible](#) - Domine a automação de provisionamento

### Gerenciamento de Clientes

- [Clientes, Contatos & Locais](#) - Modelo de dados e relacionamentos
- [Criando Clientes](#) - Guia passo a passo
- [Tags de Clientes](#) - Organize com tags
- [Atributos de Clientes](#) - Metadados personalizados
- [Registro de Atividades](#) - Rastreie todas as mudanças

## Operações de Serviço

- [Gerenciamento de Serviços](#) - Visão geral
- [Adicionando Serviços](#) - Fluxo de trabalho de provisionamento
- [Atribuindo Planos](#) - Atribuição de planos
- [Modificando Serviços](#) - Fazendo alterações
- [Uso de Serviço & Saldos](#) - Monitore o uso
- [Recarregar & Recarregar](#) - Adicione créditos

## Faturamento & Finanças

- [Visão Geral de Faturamento](#) - Guia completo de faturamento
- [Métodos de Pagamento](#) - Gerenciando métodos de pagamento
- [Transações](#) - Cobranças e créditos
- [Processando Pagamentos](#) - Fluxos de trabalho de pagamento
- [Gerenciamento de Faturas](#) - Gere e gerencie faturas

## Administração

- [Configuração do Sistema](#) - Configure o OmniCRM
- [Personalização](#) - Adapte ao seu negócio
- [Gerenciamento de Inventário](#) - Rastreio de ativos
- [Chaves de API](#) - Gerenciamento de acesso à API

## Segurança & Acesso

- [Fluxos de Autenticação](#) - Como a autenticação funciona
- [Autenticação de Dois Fatores](#) - Configure 2FA
- [Controle de Acesso Baseado em Funções](#) - Usuários, funções e permissões

## Integrações

- [Integração com Stripe](#) - Processamento de pagamentos
- [Integração com Mailjet](#) - Serviço de email
- [Documentação da API](#) - Referência da API REST

## Voltado para o Cliente

- [Portal de Autoatendimento](#) - Guia do portal do cliente
- [Atendimento ao Cliente](#) - Recursos de autoatendimento
- [Sistema de Transmissão de Emergência](#) - Alertas de emergência

## Referência

- [Glossário](#) - Termos e definições
  - [Registro de Alterações](#) - Histórico de versões
-

# O Que Torna o OmniCRM Especial?

O OmniCRM reúne todas as ferramentas necessárias para administrar um negócio de telecomunicações moderno:

## Provisionamento de Serviço Automatizado

Esqueça a configuração manual - o OmniCRM usa [automação Ansible](#) para provisionar serviços em segundos. Seja ativando um cartão SIM, configurando equipamentos de clientes ou configurando um pacote complexo, o sistema lida com isso automaticamente.

### O que ele faz:

- Provisione serviços com um único clique
- Configure equipamentos de rede automaticamente
- Crie contas de faturamento em tempo real
- Envie e-mails de boas-vindas e notificações SMS
- Reverte automaticamente se algo falhar

[Saiba mais sobre Provisionamento](#) | [Veja o Guia de Playbooks Ansible](#)

## Faturamento e Pagamentos Inteligentes

A integração embutida com [CGRateS](#) fornece classificação e cobrança em tempo real, enquanto a [integração com Stripe](#) lida com o processamento de pagamentos sem problemas.

### O que ele faz:

- Rastreamento e classificação de uso em tempo real
- Geração automática de faturas
- Processamento de cartões de crédito via Stripe
- Modelos de faturamento pré-pago e pós-pago
- Preços flexíveis e códigos promocionais
- Cálculo e relatórios de impostos

[Explore Recursos de Faturamento](#) | [Processamento de Pagamentos](#) | [Gerenciamento de Faturas](#)

## Gerenciamento Completo de Clientes

Gerencie clientes, contatos, múltiplos locais e serviços com um modelo de relacionamento abrangente.

### O que ele faz:

- Rastreie clientes, contatos e locais de serviço

- Gerencie múltiplos serviços por cliente
- Armazene atributos e metadados personalizados
- Marque clientes para organização
- Histórico completo de atividades e logs de auditoria
- Integração com Google Maps para geocodificação de locais

[Guia de Gerenciamento de Clientes](#) | [Crie um Cliente](#) | [Tags de Clientes](#)

## **Gerenciamento de Inventário**

Rastreie e gerencie todos os seus ativos físicos e virtuais - de cartões SIM a roteadores e blocos de endereços IP.

### **O que ele faz:**

- Rastreie cartões SIM, equipamentos e acessórios
- Gerencie números de telefone e blocos de endereços IP
- Atribuição automatizada durante o provisionamento
- Capacidades de importação e exportação em lote
- Armazenamento de configuração de equipamentos
- Modelos de inventário personalizados

[Documentação do Sistema de Inventário](#)

## **Portal de Autoatendimento do Cliente**

Empodere seus clientes com um [portal de autoatendimento](#) onde eles podem gerenciar seus próprios serviços.

### **O que os clientes podem fazer:**

- Ver serviços e uso
- Verificar saldos e limites de dados
- Pagar faturas online
- Baixar faturas como PDFs
- Atualizar informações de contato
- [Recarregar seus serviços](#) instantaneamente

[Guia do Portal de Autoatendimento](#) | [Recursos de Atendimento ao Cliente](#)

## **Segurança Empresarial**

Construído com segurança em seu núcleo, apresentando autenticação e autorização abrangentes.

### **Recursos de segurança:**

- [Autenticação baseada em JWT](#)

- [Autenticação de dois fatores \(2FA\)](#) com TOTP e códigos de backup
- [Controle de acesso baseado em funções \(RBAC\)](#) com permissões granulares
- Verificação de e-mail para alterações de conta
- Registro completo de auditoria via log de atividades
- Gerenciamento de sessão e controles de tempo limite

[Guia de Autenticação](#) | [Documentação RBAC](#) | [Configuração de 2FA](#)

## Integrações Poderosas

O OmniCRM se integra perfeitamente com serviços líderes da indústria:

- **CGRateS** - Motor de faturamento e classificação em tempo real para cobrança de nível telecom
- [Stripe](#) - Processamento de pagamentos seguro e armazenamento de cartões
- [Mailjet](#) - Entrega de e-mail profissional com modelos
- **Google Maps** - Validação de endereços e geocodificação para locais de site precisos
- [API RESTful](#) - Integre com seus sistemas existentes

[Documentação da API](#) | [Integração com Stripe](#) | [Integração com Mailjet](#)

## Catálogo de Produtos Flexível

Crie qualquer tipo de oferta de produto - desde serviços simples e independentes até pacotes complexos com múltiplos componentes.

### Tipos de produtos suportados:

- **Independentes** - Serviços únicos (planos móveis, pacotes de internet)
- **Pacotes** - Ofertas combinadas (internet + TV + telefone)
- **Complementos** - Serviços suplementares (recargas de dados, chamadas internacionais)
- **Promoções** - Ofertas especiais e descontos

Cada produto pode ter sua própria automação de provisionamento, regras de preços e lógica de negócios.

[Guia do Ciclo de Vida do Produto](#) | [Conceitos de Produtos e Serviços](#)

## Sistema de Transmissão de Emergência

Para operadores móveis, o OmniCRM inclui um [Sistema de Transmissão de Células](#) para alertas de segurança pública.

### O que ele faz:

- Envia alertas de emergência para áreas geográficas
- Suporte a mensagens em múltiplas línguas
- Segmentação por operador de rede móvel
- Conformidade com padrões de alerta do governo

## [Documentação de Transmissão de Células](#)

### **Pesquisa e Navegação Poderosas**

Encontre qualquer coisa instantaneamente com [pesquisa global](#) em clientes, serviços, faturas e mais.

#### **Recursos de pesquisa:**

- Pesquise em todas as entidades
- Filtre e classifique resultados
- Atalhos de navegação rápida
- Sugestões inteligentes

## [Guia de Navegação](#) | [Pesquisa Global](#)

---

## **Arquitetura do Sistema**

### [Veja a Documentação Completa da Arquitetura](#)

---

## **Conceitos Chave**

### **Entendendo Produtos vs Serviços**

Este é um dos conceitos mais importantes no OmniCRM:

- **Produto** = Um modelo ou oferta em seu catálogo (por exemplo, "Plano Móvel Ilimitado")
- **Serviço** = Uma instância ativa de um produto para um cliente específico (por exemplo, "Plano Móvel Ilimitado do John Smith")

Quando você provisiona um produto, o sistema usa [automação Ansible](#) para criar o(s) serviço(s) real(is). Um produto pode criar múltiplos serviços, nenhum serviço (apenas configuração) ou modificar serviços existentes.

### [Saiba mais sobre Produtos & Serviços](#)

### **A Magia do Provisionamento**

O segredo do OmniCRM é seu [sistema de provisionamento automatizado](#):

O playbook lida com tudo - desde a criação de contas no CGRateS até a configuração de roteadores e o envio de e-mails de boas-vindas. E se algo falhar, ele reverte automaticamente.

[Aprofunde-se: Playbooks Ansible](#) | [Sistema de Provisionamento](#)

## Faturamento Simplificado

Cada serviço gera [transações](#) (cobranças ou créditos). Estas são agrupadas em [faturas](#) e enviadas aos clientes. Com a integração do Stripe, os pagamentos podem ser processados automaticamente.

[Visão Geral de Faturamento](#) | [Processamento de Pagamentos](#)

## Hierarquia de Clientes

```
Cliente (ABC Company)
├── Contatos (John Smith - Faturamento, Jane Doe - Técnico)
├── Locais (Sede, Filial)
├── Serviços
│   ├── Internet - Sede
│   ├── Plano Móvel - John Smith
│   └── VoIP - Filial
```

Tudo é organizado de forma hierárquica, facilitando o gerenciamento de clientes com múltiplas localizações e serviços.

[Clientes, Contatos & Locais](#)

## Por Que Escolher o OmniCRM?

### Plataforma Tudo-em-Um

Tudo o que você precisa em um único sistema integrado - sem necessidade de lidar com múltiplas ferramentas ou fornecedores.

### Automação em Primeiro Lugar

Provisionamento alimentado por Ansible significa que os serviços são implantados em segundos, não em horas. Reduza erros, economize tempo, aumente a satisfação do cliente.

### Flexível e Personalizável

Desde definições de produtos até playbooks de provisionamento e atributos personalizados - adapte o OmniCRM para corresponder exatamente aos seus processos de negócios.



## Construído para Telecom

Projetado especificamente para provedores de serviços com recursos como integração com CGRateS, gerenciamento de inventário e classificação em tempo real.

## Segurança Empresarial

Autenticação JWT, 2FA, RBAC e registro completo de auditoria garantem que seus dados estejam seguros e em conformidade.

## Orientado por API

Uma API REST abrangente significa que você pode integrar o OmniCRM com quaisquer sistemas existentes ou construir ferramentas personalizadas.

---

## Obtenha Suporte

### Recursos de Documentação

- Comece com [Arquitetura do Sistema](#) para uma visão técnica
- Veja [Guias de Introdução](#) para integração específica de funções
- Verifique a [Referência Rápida de Tarefas](#) para operações comuns
- Consulte o [Glossário](#) para terminologia

### Precisa de Mais Ajuda?

Navegue pela biblioteca de documentação completa acima ou use a [pesquisa global](#) para encontrar tópicos específicos.

---

*OmniCRM - BSS/OSS Completo para Provedores de Serviços Modernos*

*Última Atualização: 2025-12-23*



# Autenticação de Dois Fatores (2FA)

## Segurança Aprimorada com Senhas de Uso Único Baseadas em Tempo

OmniCRM suporta **autenticação de dois fatores (2FA)** usando senhas de uso único baseadas em tempo (TOTP). Isso adiciona uma camada extra de segurança ao exigir que os usuários forneçam tanto sua senha quanto um código sensível ao tempo de um aplicativo autenticador.

Veja também: RBAC <rbac> para permissões de gerenciamento de 2FA, Authentication Flows <authentication\_flows> para detalhes do processo de login.

## Propósito

A 2FA fornece:

1. **Segurança Aprimorada** — Protege contas mesmo que as senhas sejam comprometidas.
2. **Conformidade** — Atende aos requisitos de segurança para indústrias regulamentadas.
3. **Escolha do Usuário** — Opcional para os usuários, pode ser aplicada por função ou globalmente.
4. **Padrão da Indústria** — Usa o protocolo TOTP compatível com Google Authenticator, Authy, Microsoft Authenticator e outros aplicativos padrão.

## Como a 2FA Funciona

Quando a 2FA é ativada para um usuário:

1. **Configuração** — O usuário escaneia um código QR com seu aplicativo autenticador durante o registro.
2. **Login** — Após inserir nome de usuário/senha, o usuário fornece o código de 6 dígitos de seu aplicativo.
3. **Verificação** — O sistema valida se o código baseado em tempo corresponde ao valor esperado.
4. **Acesso** — O usuário ganha acesso somente após ambos os fatores serem verificados.

# Habilitando a 2FA

## Para Usuários Individuais

Os usuários podem habilitar a 2FA para sua própria conta:

1. Navegue até **Configurações do Usuário** ou **Perfil**
2. Selecione **Habilitar Autenticação de Dois Fatores**
3. Escaneie o código QR com um aplicativo autenticador
4. Insira o código de verificação para confirmar a configuração
5. Salve os códigos de backup em um local seguro

Configuração de 2FA

## Backup e Recuperação

### Códigos de Backup

Ao habilitar a 2FA, os usuários recebem códigos de backup que podem ser usados se seu dispositivo não estiver disponível:

- Cada código é de uso único
- Armazene os códigos de forma segura offline
- Gere novos códigos se todos forem usados

### Redefinição pelo Administrador

Se um usuário perder o acesso ao seu autenticador e todos os códigos de backup forem esgotados, um administrador com acesso ao banco de dados pode redefinir manualmente a 2FA do usuário limpando o campo `totp_secret` no banco de dados. O usuário pode então reativar a 2FA.

## Integração com API

### Habilitar 2FA para um usuário

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

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

A resposta inclui a URI de provisionamento (para código QR) e códigos de backup.

## Verificar configuração da 2FA

POST /2fa/verify-setup/user/{user\_id}

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

## Verificar 2FA durante o login

POST /2fa/verify/user/{user\_id}

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

Retorna o token de acesso, token de atualização e dados do usuário após verificação bem-sucedida.

Verificação de Login 2FA

## Regenerar códigos de backup

POST /2fa/backup-codes/regenerate/user/{user\_id}

Requer autenticação. Retorna um novo conjunto de códigos de backup.

## Melhores Práticas

- **Códigos de backup primeiro.** Sempre salve os códigos de backup antes de concluir a configuração da 2FA.
- **Eduque os usuários.** Forneça instruções claras para configuração e recuperação.
- **Processo de redefinição seguro.** Verifique a identidade do usuário antes de redefinir manualmente a 2FA no banco de dados.

## Perguntas Frequentes

**Quais aplicativos autenticadores são suportados?** Qualquer aplicativo compatível com TOTP (Google Authenticator, Authy, Microsoft Authenticator, 1Password, etc.).

**E se eu perder meu telefone?** Use um código de backup ou entre em contato com um administrador para redefinir a 2FA.

**Posso usar SMS em vez de um aplicativo?** Atualmente, apenas aplicativos autenticadores TOTP são suportados.

**A 2FA é obrigatória?** Depende da política da sua organização. A 2FA é tipicamente obrigatória para funções administrativas e de suporte, mas é opcional para usuários clientes. O sistema não impõe 2FA para contas de clientes (usuários com função de cliente não veem os prompts de registro de 2FA).

**Por quanto tempo os códigos TOTP são válidos?** Os códigos se atualizam a cada 30 segundos e têm uma pequena janela de tempo para validação (normalmente aceita códigos da janela atual de 30 segundos mais janelas anteriores/próximas para tolerância de desvio de relógio).



# Gerenciamento de Chaves da API

A interface de Gerenciamento de Chaves da API fornece uma **interface web** para criar, monitorar e gerenciar chaves da API usadas para acesso programático à API do OmniCRM.

Nota

Para conceitos gerais de autenticação da API e exemplos de uso, veja `concepts_api`.

## Visão Geral

As chaves da API permitem **autenticação segura e de longa duração** para:

- Integrações servidor a servidor
- Scripts de automação
- Aplicativos de terceiros
- Tarefas agendadas e jobs cron
- Sistemas de monitoramento externos

Ao contrário dos tokens JWT (que expiram após minutos/horas), as chaves da API permanecem válidas até serem revogadas manualmente ou até sua data de expiração.

## Acessando o Gerenciamento de Chaves da API

Navegue para:

Ou diretamente:

**Permissão Necessária:** `MANAGE_API_KEYS` (papel de administrador)

## Visualização da Lista de Chaves da API

A página principal exibe todas as chaves da API em formato de tabela:

**Colunas:**

- **Nome** - Rótulo descritivo para a chave da API (por exemplo, "Sistema de Provisionamento", "Ferramenta de Monitoramento")
- **Criado Por** - Nome de usuário da pessoa que criou a chave
- **Chave da API** - A string da chave real (parcialmente mascarada por

- segurança)
- **Status** - Ativa, Expirada ou Revogada
- **Data de Criação** - Quando a chave foi gerada
- **Data de Expiração** - Quando a chave expirará automaticamente
- **Ações** - Botões Editar, Excluir, Regenerar

### Exibição de Exemplo:

## Widgets do Painel

Na parte superior da página, estatísticas resumidas são exibidas:

- **Total de Chaves da API** - Contagem de todas as chaves da API (ativas e inativas)
- **Chaves Ativas** - Chaves atualmente válidas
- **Expirando em Breve** - Chaves que expiram nos próximos 30 dias
- **Chaves Expiradas** - Chaves que passaram da data de expiração

## Criando uma Chave da API

### Passo 1: Clique em "Adicionar Chave da API"

Clique no botão + **Adicionar** no canto superior direito da lista de Chaves da API.

### Passo 2: Preencha os Detalhes

Um formulário modal aparece solicitando:

Nome: \_\_\_\_\_  
(por exemplo, "Sistema de Provisionamento")

Descrição: \_\_\_\_\_  
(Opcional - propósito desta chave)

Data de Expiração: [Seletor de Data]  
(Opcional - deixe em branco para sem expiração)

Permissões: ☐ Visualizar Clientes ☐ Criar Clientes ☐ Visualizar Serviços ☐ Criar Serviços ☐ Provisionamento ☐ Visualizar Inventário ☒ Admin (todas as permissões)

[Cancelar] [Gerar Chave]

### Diretrizes para os Campos:

**Nome** (obrigatório)

- Identificador curto e descritivo

- Exemplos: "Sistema de Provisionamento", "Integração de Faturamento", "Monitoramento"
- Usado em logs de auditoria e exibido na lista

### **Descrição** (opcional)

- Explicação mais detalhada
- Exemplos: "Usado pelo servidor de provisionamento Ansible", "Sincronização de faturamento de terceiros"
- Ajuda futuros administradores a entender o propósito da chave

### **Data de Expiração** (opcional)

- Se em branco: A chave nunca expira (não recomendado)
- Se definida: A chave se torna automaticamente inválida após esta data
- Recomendado: Definir expiração por segurança (90 dias a 1 ano)

### **Permissões**

- Selecione permissões específicas ou marque "Admin" para acesso total
- Segue o mesmo modelo de permissão baseado em papéis que as contas de usuário
- **Melhor Prática:** Atribua as permissões mínimas necessárias

### **Passo 3: Gerar e Copiar a Chave**

Após clicar em "**Gerar Chave**", o sistema exibe a nova chave da API criada:

△ Copie esta chave agora - ela não será mostrada novamente!

sk\_live\_a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6q7r8s9t0

[Copiar para a Área de Transferência]

[Fechar]

Aviso

### **Salve a chave da API imediatamente!**

Uma vez que você feche este diálogo, a chave completa não poderá ser recuperada novamente. Você só verá uma versão mascarada (sk\_live\_...XYZ) na visualização da lista.

Se você perder a chave, deve **regenerá-la**, o que invalida a chave antiga e pode quebrar integrações existentes.

### **Passo 4: Configure Seu Aplicativo**



Use a chave da API nas solicitações do seu aplicativo:

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

Ou em variáveis de ambiente:

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

## Gerenciando Chaves Existentes

### Visualizando Detalhes da Chave

Clique em qualquer nome de chave da API para ver detalhes completos:

- Nome completo da chave e descrição
- Timestamp de criação
- Nome de usuário do criador
- Permissões associadas
- Estatísticas de uso (se implementadas)
- Logs de acesso recentes

### Regenerando uma Chave da API

Se uma chave da API estiver comprometida ou perdida, regenere-a:

1. Clique no **⋮ (três pontos)** na coluna Ações
2. Selecione **"Regenerar Chave"**
3. Confirme a ação

Aviso

**Regenerar invalida a chave antiga imediatamente.**

Quaisquer aplicativos usando a chave antiga pararão de funcionar. Atualize todas as integrações com a nova chave antes de regenerar.

#### O que Acontece:

- A chave antiga é revogada
- Uma nova chave com as mesmas permissões é gerada
- A nova chave é exibida (copie-a imediatamente)
- Nome, descrição e permissões permanecem inalterados

### Revogando (Excluindo) uma Chave da API

Para remover permanentemente uma chave da API:

1. Clique no **:** (**três pontos**) na coluna Ações
2. Selecione "**Excluir**"
3. Confirme a exclusão

### O que Acontece:

- A chave é imediatamente revogada
- Todas as solicitações usando esta chave retornam 401 Unauthorized
- A chave é removida do banco de dados
- **Não pode ser desfeita** - a chave não pode ser recuperada

### Quando Revogar:

- A integração não é mais necessária
- A chave foi comprometida
- O sistema que usa a chave foi desativado
- Substituindo por uma nova chave com permissões diferentes

## Editando Detalhes da Chave da API

Para modificar os detalhes de uma chave da API:

1. Clique no **:** (**três pontos**) na coluna Ações
2. Selecione "**Editar**"
3. Atualize nome, descrição, expiração ou permissões
4. Clique em "**Salvar Alterações**"

### Campos Editáveis:

- Nome
- Descrição
- Data de expiração
- Permissões

### Não Editável:

- O valor da chave em si (use Regenerar para alterar)
- Data de criação
- Criado por usuário

## Status da Chave da API

As chaves da API podem ter vários status:

### Ativa

- A chave é válida e pode ser usada
- Dentro da data de expiração (ou sem expiração definida)

- Não revogada manualmente
- Exibida com um distintivo verde

### **Expirando em Breve**

- Ativa, mas expirará nos próximos 30 dias
- Exibida com um distintivo laranja/aviso
- Considere rotacionar antes da expiração

### **Expirada**

- Passou da data de expiração
- Não aceita mais autenticação
- Exibida com um distintivo vermelho
- Pode ser excluída ou a expiração estendida

### **Revogada**

- Excluída/desativada manualmente
- Permanentemente inválida
- Não é mais exibida na lista ativa

## **Filtrando e Pesquisando**

A lista de Chaves da API suporta:

### **Pesquisa:**

Pesquise por nome, descrição ou chave parcial:

### **Filtrar por Status:**

Dropdown de filtro para mostrar:

- Todas as Chaves
- Apenas Ativas
- Expirando em Breve (próximos 30 dias)
- Expiradas

### **Ordenar:**

Clique nos cabeçalhos das colunas para ordenar por:

- Nome
- Data de Criação
- Data de Expiração
- Criado Por

# Melhores Práticas de Segurança

## Geração de Chaves da API

- **Comprimento:** As chaves devem ter pelo menos 32 caracteres (o sistema impõe isso)
- **Aleatoriedade:** Geradas usando geradores de números aleatórios criptograficamente seguros
- **Formato:** Normalmente prefixadas (por exemplo, `sk_live_`) para identificação

## Armazenamento de Chaves da API

### No CRM:

- As chaves são hashadas antes do armazenamento (como senhas)
- A chave completa é mostrada apenas uma vez durante a criação
- O banco de dados armazena o hash para verificação
- Mesmo administradores não podem recuperar a chave completa depois

### No Seu Aplicativo:

- Armazene em variáveis de ambiente, não no código
- Use sistemas de gerenciamento de segredos (AWS Secrets Manager, HashiCorp Vault)
- Nunca comite chaves no controle de versão
- Rotacione chaves periodicamente (90-365 dias)

## Gerenciamento de Permissões

- **Princípio do Menor Privilégio** - Conceda apenas as permissões necessárias
- Evite criar chaves de administrador, a menos que absolutamente necessário
- Use chaves separadas para diferentes sistemas/purposes
- Revise permissões regularmente

## Monitoramento e Auditoria

- Monitore o uso da chave da API por meio de logs de atividade
- Configure alertas para padrões de acesso incomuns
- Revise timestamps de "último uso" regularmente
- Remova chaves não utilizadas

## Rotação de Chaves

Estabeleça uma política de rotação de chaves:

1. **Crie uma nova chave** com as mesmas permissões
2. **Atualize aplicativos** para usar a nova chave
3. **Monitore** para garantir que a chave antiga não seja mais usada
4. **Revogue a chave antiga** após o período de carência

## Resolução de Problemas

### "401 Unauthorized" ao usar a chave da API

- **Causa:** Chave inválida, expirada ou incorreta
- **Solução:**
  - Verifique se a chave foi copiada corretamente (sem espaços extras)
  - Verifique o status da chave (Ativa vs. Expirada)
  - Confirme se a chave tem as permissões necessárias
  - Certifique-se de usar o cabeçalho X-API-KEY (não Authorization)

### "Chave da API não encontrada" após a criação

- **Causa:** A chave pode ter sido criada, mas não armazenada corretamente
- **Solução:**
  - Verifique a lista de chaves da API para a nova entrada
  - Se estiver faltando, crie uma nova chave
  - Relate o problema ao administrador

### Chave da API expirando em breve

- **Causa:** Data de expiração se aproximando (dentro de 30 dias)
- **Solução:**
  - Crie uma nova chave com expiração estendida
  - Atualize aplicativos para usar a nova chave
  - Revogue a chave antiga após a migração

### Não é possível excluir a chave da API

- **Causa:** Pode estar protegida ou em uso
- **Solução:**
  - Certifique-se de que você tem permissões de administrador
  - Verifique se a chave está bloqueada/protegida
  - Contate o administrador se o problema persistir

## Endpoints da API (para Gerenciamento Programático)

As chaves da API também podem ser gerenciadas via API (requer permissões de administrador):

### Listar Chaves da API

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

### **Criar Chave da API**

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

```
{
  "name": "Nova Integração",
  "description": "Sincronização de faturamento de terceiros",
  "expiry_date": "2026-01-10",
  "permissions": ["view_customer", "view_service"]
}
```

### **Resposta:**

```
{
  "api_key_id": 123,
  "name": "Nova Integração",
  "api_key": "sk_live_alb2c3d4e5f6g7h8i9j0",
  "status": "active",
  "created": "2025-01-10T10:00:00Z",
  "expiry_date": "2026-01-10T23:59:59Z"
}
```

### **Revogar Chave da API**

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

### **Atualizar Chave da API**

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

```
{
  "name": "Nome Atualizado",
  "expiry_date": "2026-12-31"
}
```

## **Casos de Uso Comuns**

### **Caso de Uso 1: Integração do Sistema de Provisionamento**

Crie uma chave da API para seu servidor de provisionamento Ansible:

1. Navegue para Chaves da API → Adicionar
2. Nome: "Servidor de Provisionamento Ansible"
3. Descrição: "Usado pela automação de provisionamento"
4. Permissões: Provisionamento, Visualizar/Criar Serviços, Visualizar/Atualizar Inventário
5. Expiração: 1 ano
6. Copie a chave e adicione ao `crm_config.yaml` do Ansible

## **Caso de Uso 2: Integração de Faturamento de Terceiros**

Crie uma chave somente leitura para exportação de faturamento:

1. Nome: "Sincronização de Faturamento - QuickBooks"
2. Permissões: Visualizar Clientes, Visualizar Transações, Visualizar Faturas
3. Expiração: 90 dias (rotacionar trimestralmente)
4. Usar no script de exportação agendada

## **Caso de Uso 3: Monitoramento e Alerta**

Crie uma chave para coleta de métricas do Prometheus/Grafana:

1. Nome: "Monitoramento - Grafana"
2. Permissões: Visualizar Serviços, Visualizar Provisionamento
3. Expiração: Nunca (monitoramento precisa de acesso contínuo)
4. Configurar na fonte de dados do Grafana

## **Caso de Uso 4: API do Portal do Cliente**

Crie uma chave para o portal de autoatendimento do cliente:

1. Nome: "Backend do Portal do Cliente"
2. Permissões: Visualizar Próprio Cliente, Visualizar Próprios Serviços, Criar Pagamentos
3. Expiração: 180 dias
4. Usar nas chamadas da API do backend do portal

## **Documentação Relacionada**

- `concepts_api` - Conceitos e exemplos de autenticação da API
- `rbac` - Controle de acesso baseado em papéis e permissões
- `2fa` - Autenticação de dois fatores para segurança adicional



# Atributos do Cliente

Atributos do Cliente são pares flexíveis de chave-valor que podem ser anexados a qualquer registro de cliente para armazenar metadados personalizados, configurações de configuração ou informações específicas do negócio que não se encaixam nos campos padrão do cliente.

Para categorização visual de clientes e links clicáveis, veja Customer Tags <administration\_tags>. Para informações básicas sobre clientes, veja Customers, Contacts, Sites & Services <basics\_customers>.

Ao contrário dos campos de banco de dados fixos, os atributos permitem que você estenda dinamicamente os registros de clientes sem modificar o esquema do banco de dados. Isso os torna ideais para armazenar dados específicos de implementação, parâmetros de integração ou sinalizadores de lógica de negócios personalizados.

## Propósito e Casos de Uso

Casos de uso comuns para Atributos do Cliente incluem:

### 1. Dados de Integração

Armazene identificadores de sistemas externos ou chaves de API específicas para este cliente:

- `external_crm_id` = "SF-12345" (ID do cliente Salesforce)
- `legacy_system_id` = "OLD-CRM-789" (Referência de migração)
- `hubspot_contact_id` = "12345678" (Integração HubSpot)

### 2. Lógica de Negócios Personalizada

Armazene sinalizadores ou configurações que controlam o comportamento específico do cliente:

- `billing_method` = "quarterly" (Substituir a cobrança mensal padrão)
- `auto_provision` = "true" (Habilitar provisionamento automático de serviços)
- `support_tier` = "premium" (Nível de suporte personalizado)
- `credit_limit` = "10000" (Limite de crédito específico do cliente)

### 3. Dados de Conformidade e Regulatórios

Rastreie metadados relacionados à conformidade:



- `gdpr_consent_date = "2025-01-01"` (Consentimento para processamento de dados)
- `tax_exempt = "true"` (Status de isenção fiscal)
- `regulatory_entity = "FCC-123456"` (Identificador regulatório)

#### 4. Metadados Operacionais

Armazene informações operacionais:

- `preferred_contact_method = "email"` (Preferência de comunicação)
- `account_manager = "<john.smith@company.com>"` (Gerente de conta designado)
- `onboarding_date = "2025-01-15"` (Rastreamento do ciclo de vida do cliente)
- `churn_risk_score = "0.23"` (Análise preditiva)

#### 5. Parâmetros de Provisionamento

Armazene configurações específicas de provisionamento:

- `radius_username_format = "email"` (Formato RADIUS personalizado)
- `vlan_id = "100"` (Configuração de rede)
- `ipv6_enabled = "true"` (Sinalizadores de recurso)

## Atributos vs. Campos Padrão

### Use Atributos Quando:

- Os dados são específicos da implementação ou variam por instalação
- Os requisitos mudam com frequência
- Armazenar metadados específicos de integração
- Prototipar novos recursos antes de adicionar campos ao banco de dados
- Os dados não precisam de consultas complexas ou junções

### Use Campos Padrão Quando:

- Os dados são essenciais para o modelo de cliente (nome, email, endereço)
- Busca, filtragem ou relatórios frequentes são necessários
- Os dados têm restrições de integridade referencial
- O desempenho é crítico para consultas em grande escala

## Gerenciando Atributos via a UI

### Visualizando Atributos do Cliente

Para visualizar atributos de um cliente:

1. Navegue até a página de visão geral do cliente
2. Clique na aba **Attributes**
3. Você verá uma tabela de todos os atributos para este cliente, mostrando:
  - Nome do Atributo (chave)
  - Valor do Atributo
  - Data de Criação
  - Data da Última Modificação

## Criando um Novo Atributo

Para criar um novo atributo para um cliente:

1. Navegue até a página de visão geral do cliente
2. Clique na aba **Attributes**
3. Clique no botão **Add Attribute**
4. Preencha os campos obrigatórios:
  - **Attribute Name** (obrigatório): A chave/nome para este atributo (por exemplo, `external_crm_id`)
  - **Attribute Value** (obrigatório): O valor a ser armazenado (por exemplo, SF-12345)
5. Clique em **Create Attribute**

## Convenções de Nomenclatura:

- Use letras minúsculas com underscores: `external_system_id` ✓
- Evite espaços: `external system id` ✗
- Mantenha os nomes descritivos, mas concisos
- Use nomenclatura consistente entre clientes para os mesmos tipos de atributos

## Editando um Atributo

Para editar um atributo existente:

1. Navegue até a página de visão geral do cliente
2. Clique na aba **Attributes**
3. Encontre o atributo que você deseja editar na tabela
4. Clique no botão **Edit** (lápiz)
5. Modifique o nome ou valor do atributo
6. Clique em **Update Attribute**

## Nota

Alterar o nome de um atributo cria um novo par chave-valor. Certifique-se de que isso não quebre integrações que dependem do nome original do atributo.

## Deletando um Atributo

Para deletar um atributo:

1. Navegue até a página de visão geral do cliente
2. Clique na aba **Attributes**
3. Encontre o atributo que você deseja deletar na tabela
4. Clique no botão **Delete** (lixeira)
5. Confirme a exclusão na janela pop-up

### Aviso

Deletar atributos usados por integrações, fluxos de trabalho de provisionamento ou lógica de cobrança pode causar falhas. Verifique as dependências antes da exclusão.

## Referência de Campo de Atributo

### Integração API

Os atributos podem ser gerenciados programaticamente via API:

#### Criar ou Atualizar um Atributo

**Endpoint:** PUT /crm/attribute/

**Permissão Necessária:** create\_customer\_attribute

#### Corpo da Solicitação:

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

#### Resposta:

```
{
  "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"
}
```

## Atualizar um Atributo Existente

**Endpoint:** PATCH /crm/attribute/attribute\_id/{attribute\_id}

**Permissão Necessária:** update\_customer\_attribute

**Corpo da Solicitação:**

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

## Obter Atributo por ID

**Endpoint:** GET /crm/attribute/attribute\_id/{attribute\_id}

**Permissão Necessária:** view\_customer\_attribute

**Resposta:**

```
{
  "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"
}
```

## Obter Todos os Atributos por ID do Cliente

**Endpoint:** GET /crm/attribute/customer\_id/{customer\_id}

**Permissão Necessária:** view\_customer\_attribute

**Resposta:**

```
[
  {
    "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"  
  }  
]
```

## Deletar um Atributo

**Endpoint:** DELETE /crm/attribute/attribute\_id/{attribute\_id}

**Permissão Necessária:** delete\_customer\_attribute

**Resposta:**

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

## Operações em Lote de Atributos

### Gerenciando Múltiplos Atributos

Para definir múltiplos atributos para um cliente de uma só vez (por exemplo, durante a integração ou sincronização):

```
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"}  
    )
```

## Consultando Clientes por Atributo

Embora os atributos não tenham endpoints de busca incorporados, você pode filtrar clientes por atributo usando a API de busca de clientes com filtragem personalizada:

```
# Obter todos os clientes, depois filtrar por atributo no código do aplicativo
customers = requests.get("https://api.example.com/crm/customer/").json()

for customer in customers:
    attributes = requests.get(
        f"https://api.example.com/crm/attribute/customer_id/{customer['customer_id']}"
    ).json()

    # Encontrar clientes com 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 frequentes baseadas em atributos, considere adicionar campos de banco de dados indexados ou implementar um endpoint de busca dedicado.

## Melhores Práticas

### 1. Convenções de Nomenclatura

- Use snake\_case: external\_system\_id ✓
- Seja descritivo: billing\_method ✓ vs method ✗
- Evite palavras-chave reservadas ou caracteres especiais
- Documente os significados dos atributos em seu guia de implementação

### 2. Tipos de Dados

- Atributos armazenam valores como strings (máx. 150 caracteres)
- Para booleanos, use "true"/"false" (minúsculas)
- Para datas, use o formato ISO 8601: "2025-01-04 10:30:00"
- Para grandes dados JSON, considere campos de banco de dados dedicados

### 3. Validação

- Valide os valores dos atributos no código do aplicativo antes de salvar
- Use formatos de valor consistentes entre clientes

- Documente os valores esperados para cada nome de atributo

#### 4. Documentação

- Mantenha um registro dos nomes e propósitos dos atributos
- Documente quais sistemas/integrações dependem de atributos específicos
- Inclua exemplos de valores válidos

#### 5. Migração e Limpeza

- Audite regularmente atributos não utilizados
- Remova atributos obsoletos após migrações de sistema
- Versão os nomes dos atributos ao alterar esquemas (por exemplo, api\_key\_v2)

## Exemplos de Fluxos de Trabalho

### Integração de Onboarding

Ao migrar clientes de um sistema legado:

```
# Armazenar referência do sistema legado para depuração
PUT /crm/attribute/
{
  "customer_id": 123,
  "attribute_name": "legacy_crm_id",
  "attribute_value": "OLD-12345"
}

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

### Regras de Cobrança Personalizadas

Substituir o ciclo de cobrança padrão para um cliente específico:

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

```
# Então, no código de cobrança, verifique o atributo antes de
processar
attributes = GET /crm/attribute/customer_id/123
billing_cycle = next(
    (a['attribute_value'] for a in attributes if a['attribute_name']
    == 'billing_cycle'),
    'monthly' # padrão
)
```

## Sinalizadores de Recursos

Habilitar recursos beta para clientes específicos:

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

## Permissões

As operações de atributo requerem as seguintes permissões:

- view\_customer\_attribute - Visualizar atributos
- create\_customer\_attribute - Criar novos atributos
- update\_customer\_attribute - Modificar atributos existentes
- delete\_customer\_attribute - Remover atributos

Veja rbac para configuração de controle de acesso baseado em funções.

## Solução de Problemas

### Atributo Não Aparecendo na UI

- Verifique se o atributo foi criado (verifique a resposta da API)
- Atualize a página para recarregar os dados do cliente
- Verifique se o usuário tem permissão view\_customer\_attribute

### Não é Possível Atualizar Atributo

- Certifique-se de que você tem permissão update\_customer\_attribute
- Verifique se o attribute\_id está correto
- Verifique se o atributo pertence ao cliente especificado



## **Integração Falhando Após Exclusão de Atributo**

- Restaure o atributo com o valor anterior
- Atualize o código de integração para lidar com atributos ausentes de forma adequada
- Audite as dependências do atributo antes da exclusão

## **Valor do Atributo Truncado**

- Os valores dos atributos têm limite de 150 caracteres
- Para dados mais longos, divida em múltiplos atributos ou use o campo de notas do cliente
- Considere armazenar grandes dados em campos de banco de dados dedicados



# Configuração do Sistema

OmniCRM utiliza dois principais sistemas de configuração: **crm\_config.yaml** para configurações da API de backend e **variáveis de ambiente** para a UI React. Este guia cobre todas as opções de configuração e como modificá-las.

## Visão Geral dos Arquivos de Configuração

### Configuração da API de Backend:

- **Arquivo:** OmniCRM-API/crm\_config.yaml
- **Formato:** YAML
- **Requer:** Reinício da API após alterações
- **Usado para:** Banco de dados, integrações, segurança, provisionamento

### Configuração da UI Frontend:

- **Arquivo:** OmniCRM-UI/.env
- **Formato:** Variáveis de ambiente
- **Requer:** Reconstrução da UI após alterações
- **Usado para:** Branding, recursos, serviços externos

## Configuração de Backend (crm\_config.yaml)

O arquivo `crm_config.yaml` contém todas as configurações do sistema de backend.

### Configuração do Banco de Dados

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

### Campos:

- `username` - Nome de usuário do banco de dados MySQL
- `password` - Senha do banco de dados MySQL
- `server` - Nome do host ou IP do servidor de banco de dados (padrão: localhost)

### Conexão com o Banco de Dados:

- O nome do banco de dados é codificado como omnicon
- Porta padrão: 3306 (padrão do MySQL)
- String de conexão: mysql+pymysql://username:password@server/omnicon

**Nota de Segurança:** Nunca comite este arquivo com credenciais reais para controle de versão. Use configuração específica para o ambiente ou gerenciamento de segredos.

## Tipos de Serviço

```
service_types:  
- omnicon  
- mobile  
- internet  
- iptv  
- voip
```

**Propósito:** Define valores válidos de tipo de serviço para o campo service\_type.

### Tipos Padrão:

- mobile - Serviços móveis/celulares
- internet - Internet fixa (fibra, DSL, sem fio)
- iptv - Serviços de televisão
- voip - Serviços de voz sobre IP
- omnicon - Serviços de cobrança/carregamento

Adicione tipos de serviço personalizados aqui para seus casos de uso específicos.

## Configuração do HSS (Home Subscriber Server)

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

### Campos:

- hss\_peers - Lista de URLs de servidores HSS para gerenciamento de assinantes
- apn\_list - Lista separada por vírgulas de identificadores de APN (Access Point Name)

**Usado para:** Provisionamento de rede móvel e autenticação de assinantes.

## Configuração de Email do Mailjet

```
mailjet:
  api_key: your_mailjet_api_key
  api_secret: your_mailjet_api_secret

  api_crmCommunicationCustomerWelcome:
    from_email: "support@yourcompany.com"
    from_name: "Your Company Support"
    template_id: 5977509
    subject: "Welcome to Your Company"

  api_crmCommunicationCustomerInvoice:
    from_email: "billing@yourcompany.com"
    from_name: "Your Company Billing"
    template_id: 6759851
    subject: "Your Invoice - "
```

### Tipos de Email Configurados:

- api\_crmCommunicationCustomerWelcome - Email de boas-vindas para novos clientes
- api\_crmCommunicationCustomerInvoice - Entrega de fatura
- api\_crmCommunicationCustomerInvoiceReminder - Lembretes de fatura vencida
- api\_crmCommunicationUserWelcome - Boas-vindas para novos usuários da equipe
- api\_crmCommunicationUserPasswordReset - Solicitações de redefinição de senha
- api\_crmCommunicationUserPasswordResetSuccess - Redefinição de senha bem-sucedida
- api\_crmCommunicationUserPasswordChange - Notificações de alteração de senha
- api\_crmCommunicationEmailVerification - Verificação de endereço de email
- api\_crmCommunicationsBalanceExpired - Notificações de expiração de serviço
- api\_crmCommunicationsBalanceLow - Alertas de saldo baixo

### IDs de Template:

Obtenha na sua conta do Mailjet após criar templates de email. Veja `integrations_mailjet` para detalhes.

## Configuração de Provisionamento

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

### **Campos:**

- `failure_list` - Endereços de email para notificar quando o provisionamento falhar

Quando os playbooks do Ansible falham durante o provisionamento, o sistema envia detalhes de erro para esses endereços.

## **Configuração de Fatura**

```
invoice:
  template_filename: 'your_invoice_template.html'
```

### **Campos:**

- `template_filename` - Arquivo de template HTML para geração de faturas

O arquivo de template deve existir no diretório `OmniCRM-API/templates/`.

## **URL Base do CRM**

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

**Propósito:** Usado pelos playbooks do Ansible para fazer chamadas de API de volta ao CRM.

### **Importante:**

- Não inclua barra no final
- Use URL acessível publicamente se os playbooks forem executados em servidores diferentes
- Atualize ao implantar em produção (por exemplo, `https://api.yourcrm.com`)

## **Configuração do OCS (Online Charging System)**

```
ocs:
  ocsApi: 'http://10.179.2.142:8080/api'
  ocsTenant: 'mnc380.mcc313.3gppnetwork.org'
  cgrates: 'localhost:2080'
```

### **Campos:**

- `ocsApi` - URL do endpoint da API REST do OCS
- `ocsTenant` - Identificador do tenant para implantações OCS multi-tenant
- `cgrates` - Endpoint JSON-RPC do CGRateS (host:port)

**Usado para:** Cobrança em tempo real, gerenciamento de saldo, rastreamento de uso.

## Configuração do SMSC (SMS Center)

```
smc:  
  source_msidn: 'YourCompany'  
  smc_url: 'http://10.179.2.216/SMSc/'  
  api_key: 'your_smc_api_key'
```

### Campos:

- source\_msidn - ID do remetente para SMS de saída (nome da empresa ou código curto)
- smc\_url - URL da API do Centro de SMS
- api\_key - Chave de autenticação para a API do SMSC

**Usado para:** Envio de notificações SMS (alertas de saldo, OTPs, etc.)

## Configuração de Transmissão Celular

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

**Propósito:** Endpoint da API do Centro de Transmissão Celular (CBC) para alertas de emergência.

Veja features\_cell\_broadcast para detalhes de uso.

## Chave Secreta JWT

```
jwt_secret:  
'2b93110f723db60172c8e9a1eaa80027a9a9c3f05b44e50dc3fcf38dba68d87e'
```

**Propósito:** Chave secreta para assinar tokens de autenticação JWT.

### Segurança:

- Gere usando: openssl rand -hex 32
- **Nunca compartilhe publicamente**
- Alterar isso invalida todas as sessões de usuário existentes
- Use segredos diferentes para dev/staging/produção

## Configuração de Pagamento Stripe

```
stripe:  
  secret_key: 'sk_test_...'  
  publishable_key: 'pk_test_...'  
  currency: 'usd'
```

```
statement_descriptor_suffix: 'YOURCOMPANY'
```

### Campos:

- `secret_key` - Chave secreta da API do Stripe (começa com `sk_`)
- `publishable_key` - Chave publicável do Stripe (começa com `pk_`)
- `currency` - Código de moeda ISO 4217 (usd, gbp, aud, eur, etc.)
- `statement_descriptor_suffix` - Texto que aparece nos extratos de cartão de crédito dos clientes

### Tipos de Chave:

- Chaves de teste: `sk_test_...` e `pk_test_...` (para desenvolvimento)
- Chaves ao vivo: `sk_live_...` e `pk_live_...` (para produção)

Veja `integrations_stripe` para detalhes de configuração.

### Chaves da API

```
api_keys:  
  "your-secure-api-key-minimum-32-chars":  
    roles: ["admin"]  
    ips: ["127.0.0.1", "::1"]  
  "another-api-key-for-specific-service":  
    roles: ["customer_service_agent_1"]  
    ips: ["10.0.1.50"]
```

### Estrutura:

- **Chave (string):** A chave da API real (mínimo 32 caracteres)
- **roles (lista):** Nomes de função que esta chave tem acesso
- **ips (lista, opcional):** Endereços IP permitidos para usar esta chave

### Gerando Chaves da API:

```
openssl rand -base64 48
```

### Funções:

- `admin` - Acesso total a todos os endpoints
- Funções personalizadas definidas no sistema RBAC

Veja `administration_api_keys` e `concepts_api` para detalhes.

### Lista Branca de IP

```
ip_whitelist:  
  "10.179.2.142":
```

```
roles: ["admin"]
"192.168.1.100":
  roles: ["provisioning"]
```

**Propósito:** Permitir que endereços IP específicos acessem a API sem autenticação.

**Estrutura:**

- **Endereço IP (string):** Endereço IPv4 a ser incluído na lista branca
- **roles (lista):** Funções atribuídas às solicitações deste IP

**Aviso de Segurança:**

- Use apenas para redes internas confiáveis
- Não deve usar IPs de localhost (127.0.0.1, ::1)
- Use chaves da API em vez disso para acesso externo
- Considere regras de firewall como proteção adicional

## Configuração Frontend (.env)

A UI React é configurada através de variáveis de ambiente em OmniCRM-UI/.env.

### Configuração de Branding

```
REACT_APP_COMPANY_NAME="YourCompany"
REACT_APP_PORTAL_NAME="YourPortal"
REACT_APP_SELF_CARE_NAME="YourCare"
REACT_APP_COMPANY_TAGLINE="Your Company Slogan"
```

**Campos:**

- REACT\_APP\_COMPANY\_NAME - Nome da empresa (aparece em cabeçalhos, emails, branding)
- REACT\_APP\_PORTAL\_NAME - Nome do portal de administração (títulos de página, navegação)
- REACT\_APP\_SELF\_CARE\_NAME - Nome do portal do cliente
- REACT\_APP\_COMPANY\_TAGLINE - Slogan de marketing (aparece na página de login)

**Exemplo:**

### Configuração Regional

```
REACT_APP_DEFAULT_LOCATION="London, United Kingdom"
REACT_APP_DEFAULT_COUNTRY="United Kingdom"
REACT_APP_DEFAULT_LANGUAGE="en"
REACT_APP_LOCALE="en-GB"
```



## **Campos:**

- `REACT_APP_DEFAULT_LOCATION` - Localização padrão para mapas e endereços
- `REACT_APP_DEFAULT_COUNTRY` - País padrão para formulários
- `REACT_APP_DEFAULT_LANGUAGE` - Idioma da UI (ar, ch, en, fr, gr, it, ru, sp)
- `REACT_APP_LOCALE` - Localização de formatação de data/número (en-GB, en-US, etc.)

## **Idiomas Suportados:**

- ar - Árabe
- ch - Chinês
- en - Inglês (padrão)
- fr - Francês
- gr - Grego
- it - Italiano
- ru - Russo
- sp - Espanhol

## **Configuração de Moeda**

```
REACT_APP_CURRENCY_CODE="USD"  
REACT_APP_CURRENCY_SYMBOL="$"
```

## **Campos:**

- `REACT_APP_CURRENCY_CODE` - Código de moeda ISO 4217
- `REACT_APP_CURRENCY_SYMBOL` - Símbolo a ser exibido (£, \$, €, etc.)

## **Moedas Comuns:**

- USD - \$ (Dólar Americano)
- GBP - £ (Libra Esterlina)
- EUR - € (Euro)
- AUD - \$ (Dólar Australiano)
- CAD - \$ (Dólar Canadense)

**Nota:** Deve corresponder a `stripe.currency` em `crm_config.yaml`.

## **Configuração de Tema de Cores**

```
REACT_APP_PRIMARY_COLOR=#405189  
REACT_APP_SECONDARY_COLOR=#2bFFcf  
REACT_APP_TERTIARY_COLOR=#1a9fbf
```

## **Cores Disponíveis:**

- REACT\_APP\_PRIMARY\_COLOR - Cor principal da marca (botões, links, destaques)
- REACT\_APP\_SECONDARY\_COLOR - Cor de destaque
- REACT\_APP\_TERTIARY\_COLOR - Destaque adicional
- REACT\_APP\_SUCCESS\_COLOR - Mensagens de sucesso (#28a745)
- REACT\_APP\_INFO\_COLOR - Mensagens de informação (#17a2b8)
- REACT\_APP\_WARNING\_COLOR - Avisos (#ffc107)
- REACT\_APP\_DANGER\_COLOR - Erros (#dc3545)
- REACT\_APP\_LIGHT\_COLOR - Fundos claros (#f8f9fa)
- REACT\_APP\_DARK\_COLOR - Texto escuro (#343a40)
- REACT\_APP\_PRIMARY\_DARK\_COLOR - Variante escura da cor primária (para modo escuro/estados de hover)

**Formato:** Códigos de cores hexadecimais (#RRGGBB)

## Configuração de Fonte

```
REACT_APP_FONT_FAMILY=Quicksand
```

**Propósito:** Define a família de fonte principal para toda a UI.

**Importante:** Todas as fontes são **auto-hospedadas localmente** dentro da aplicação OmniCRM-UI. Isso significa:

- **Sem carregamento de fontes externas** - As fontes são empacotadas com a aplicação
- **Compatível com jardim murado** - Nenhum acesso à internet é necessário para que as fontes funcionem
- **Operação offline** - Funcionalidade completa em ambientes de rede restrita ou isolada
- **Privacidade** - Nenhuma solicitação externa para Google Fonts, Adobe Fonts ou outros CDNs
- **Desempenho** - Carregamento mais rápido sem dependências externas
- **Segurança** - Sem rastreamento de terceiros ou vazamento de dados através de solicitações de fonte

## Opções Disponíveis:

### Fontes Sem Serifa:

- Inter
- Roboto
- Open Sans
- Lato
- Quicksand (padrão)
- Poppins
- Nunito
- Montserrat

- Work Sans
- Source Sans Pro
- Raleway
- Ubuntu
- Josefin Sans
- HKGrotesk

### Fontes Serifadas:

- Merriweather
- Lora
- Playfair Display

### Fontes do Sistema:

- System - Usa fontes nativas do dispositivo para melhor desempenho e menor tamanho do pacote

**Padrão:** Quicksand

### Adicionando Fontes Personalizadas

**Sim, você pode adicionar fontes adicionais!** Todas as fontes são armazenadas localmente na aplicação.

### Para adicionar uma nova fonte personalizada:

1. **Adicione arquivos de fonte** em `OmniCRM-UI/src/assets/fonts/your-font-name/`
  - Use o formato WOFF2 para melhor compressão e suporte a navegadores
  - Inclua múltiplos pesos (300, 400, 500, 600, 700) para renderização adequada
  - Nomeie os arquivos: `your-font-name-300.woff2`, `your-font-name-400.woff2`, etc.
2. **Defina regras @font-face** em `OmniCRM-UI/src/assets/scss/fonts/_google-fonts.scss`

```
//
// Sua Fonte Personalizada - Descrição
//

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

### 3. Defina no arquivo .env:

```
REACT_APP_FONT_FAMILY=Your Font Name
```

#### Diretrizes de Peso da Fonte:

- 300 - Leve (opcional, para títulos sutis)
- 400 - Regular (obrigatório, texto padrão)
- 500 - Médio (opcional, ênfase)
- 600 - Semi-Negrito (opcional, subtítulos)
- 700 - Negrito (obrigatório, títulos e texto forte)

**Nota:** Todas as fontes permanecem auto-hospedadas e funcionam offline. Nenhum CDN externo ou conexão com a internet é necessária.

## Serviços Externos

```
REACT_APP_GOOGLE_API_KEY=your_google_maps_api_key
REACT_APP_STRIPE_PUBLISHABLE_KEY=pk_test_...
```

#### Campos:

- REACT\_APP\_GOOGLE\_API\_KEY - Chave da API do Google Maps (para mapas, geolocalização)
- REACT\_APP\_STRIPE\_PUBLISHABLE\_KEY - Chave publicável do Stripe (para pagamentos)

#### Deve Corresponder:

REACT\_APP\_STRIPE\_PUBLISHABLE\_KEY deve corresponder a stripe.publishable\_key em crm\_config.yaml.

## Links Rápidos para Aplicativos Web

```
REACT_APP_WEB_APP_1_NAME="GitHub"  
REACT_APP_WEB_APP_1_URL="https://github.com"  
REACT_APP_WEB_APP_1_ICON_PATH="resources/webapp_icons/github.png"
```

**Propósito:** Configurar até 6 links de acesso rápido a aplicativos web na navegação da UI.

### Padrão:

- REACT\_APP\_WEB\_APP\_N\_NAME - Nome exibido
- REACT\_APP\_WEB\_APP\_N\_URL - URL de destino
- REACT\_APP\_WEB\_APP\_N\_ICON\_PATH - Caminho do arquivo do ícone (relativo a public/)

**Exemplos de Ícones:** GitHub, Xero, Monday.com, Gmail, MailJet, Slack

## Integração com Grafana

```
REACT_APP_GRAFANA_URLS=http://grafana1.local/d/  
abc,http://grafana2.local/d/xyz  
REACT_APP_GRAFANA_LABELS=Network Monitoring,Service Health  
REACT_APP_GRAFANA_API_KEY=your_grafana_api_key
```

### Campos:

- REACT\_APP\_GRAFANA\_URLS - Lista separada por vírgulas de URLs de dashboards do Grafana
- REACT\_APP\_GRAFANA\_LABELS - Lista separada por vírgulas de nomes de dashboards
- REACT\_APP\_GRAFANA\_API\_KEY - Chave da API do Grafana para incorporação

**Uso:** Incorpora dashboards do Grafana na página de dashboard do OmniCRM.

## URLs de Suporte

```
REACT_APP_FAQS_URL=https://support.yourcompany.com/faqs  
REACT_APP_SUPPORT_URL=https://support.yourcompany.com
```

**Propósito:** Links para recursos de suporte externos mostrados na UI.

## Logins Sociais

```
REACT_APP_ALLOW_SOCIAL_LOGINS=yes
```

### Opções:

- yes - Habilitar botões de login social (Google, Facebook, etc.)
- no - Desabilitar logins sociais

**Nota:** Provedores de login social devem ser configurados separadamente.

## Configuração de Recarga e Recarregamento

```
REACT_APP_TOPUP_PRICE_PER_DAY=10
```

**Propósito:** Define o preço por dia para serviços de recarga/recarregamento no portal de autoatendimento.

### Campos:

- REACT\_APP\_TOPUP\_PRICE\_PER\_DAY - Preço diário para serviços de recarga (valor numérico)

**Exemplo:** Se definido como 10 e a moeda for USD, os clientes pagam \$10 por dia de serviço.

**Nota:** Este valor deve corresponder à configuração de preços do backend. Veja `features_topup_recharge` para detalhes completos de configuração.

## Aplicando Alterações de Configuração

### Backend (crm\_config.yaml)

1. Edite OmniCRM-API/crm\_config.yaml
2. Salve as alterações
3. Reinicie o serviço da API:

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

**As alterações entram em vigor imediatamente** após o reinício.

### Frontend (.env)

1. Edite OmniCRM-UI/.env
2. Salve as alterações
3. Reconstrua a UI:

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

4. Reinicie o serviço da UI ou servidor web

## Modo de Desenvolvimento:

Durante o desenvolvimento com `npm start`, reinicie o servidor de desenvolvimento para aplicar as alterações.

# Melhores Práticas de Configuração

## Segurança

- **Nunca comite segredos** - Use `.gitignore` para arquivos de configuração com credenciais
- **Use configurações específicas para o ambiente** - Separe configurações de dev/staging/produção
- **Rotacione segredos regularmente** - Atualize segredos JWT, chaves da API periodicamente
- **Limite permissões de chaves da API** - Atribua funções mínimas necessárias
- **Use lista branca de IP com moderação** - Prefira chaves da API para melhor segurança

## Manutenção

- **Documente alterações** - Mantenha um changelog de modificações de configuração
- **Faça backup das configurações** - Armazene cópias antes de grandes alterações
- **Teste em staging** - Verifique alterações de configuração antes da produção implantação
- **Controle de versão** - Rastreie templates de configuração (sem segredos) no git

## Desempenho

- **Use banco de dados local** - Evite banco de dados remoto para melhor desempenho
- **Configure cache** - Habilite cache OCS se disponível
- **Otimize Grafana** - Limite o número de dashboards incorporados

## Branding

- **Combine cores** - Certifique-se de que as cores da UI complementam seu logotipo
- **Teste contraste** - Verifique a legibilidade do texto em fundos coloridos
- **Teste em dispositivos móveis** - Verifique o branding em dispositivos móveis
- **Colocação do logotipo** - Coloque logotipos da empresa em `OmniCRM-UI/public/resources/`

# Resolução de Problemas

## Alterações não aplicadas

- **Causa:** Serviço não reiniciado ou UI não reconstruída
- **Correção:** Reinicie os serviços da API/UI após alterações de configuração

## Erros de sintaxe YAML

- **Causa:** Formatação YAML inválida (indentação, aspas, etc.)
- **Correção:** Valide YAML online ou use `yamllint crm_config.yaml`

## Conexão com o banco de dados falhou

- **Causa:** Credenciais erradas ou servidor inacessível
- **Correção:** Verifique se o banco de dados está em execução, se as credenciais estão corretas

## Pagamentos Stripe não funcionando

- **Causa:** Chaves incompatíveis entre backend e frontend
- **Correção:** Certifique-se de que `publishable_key` corresponda em ambos os arquivos

## Emails não enviados

- **Causa:** Credenciais do Mailjet inválidas ou IDs de template
- **Correção:** Verifique a chave/secreto da API do Mailjet, verifique se os IDs de template existem

## Documentação Relacionada

- `administration_api_keys` - Gerenciamento de chaves da API
- `integrations_stripe` - Configuração de pagamento Stripe
- `integrations_mailjet` - Integração de email
- `concepts_api` - Autenticação da API
- `rbac` - Controle de acesso baseado em funções





# Personalização e Configuração

Este guia explica como personalizar e configurar o OmniCRM para corresponder à sua identidade de marca, requisitos operacionais e necessidades de integração.

## Variáveis de Ambiente (.env)

O OmniCRM usa variáveis de ambiente para configurar tanto a API de backend quanto a interface do usuário de frontend. Existem dois arquivos .env separados que controlam diferentes aspectos do sistema.

### Configuração da API de Backend (.env)

Localizado na raiz do diretório do OmniCRM, este arquivo configura conexões de banco de dados e integração com CGRates.

**Localização:** /OmniCRM/.env

#### Configuração do Banco de Dados:

```
# Banco de Dados MySQL/MariaDB (Dados do CRM)
MYSQL_ROOT_PASSWORD=your_secure_password
MYSQL_USER=omnitouch
MYSQL_PASSWORD=your_database_password
MYSQL_DATABASE=crm
DB_SERVER=db

# Banco de Dados PostgreSQL (Dados do CGRates)
POSTGRES_USER=cgrates
POSTGRES_PASSWORD=cgrates_password
POSTGRES_DB=cgrates_db
```

#### Configuração do CGRates:

```
# Credenciais da API do CGRates
CGRATES_API_USER=admin
CGRATES_API_PASS=secret
CGRATES_DB_USER=cgrates
CGRATES_DB_PASS=cgrates_password
CGRATES_DB_NAME=cgrates_db
CGRATES_DB_PORT=5432
```

#### Considerações de Segurança:

- **Nunca comite arquivos .env no controle de versão** - Use .env.example como um modelo
- **Use senhas fortes** - Mínimo de 16 caracteres com letras maiúsculas e minúsculas, números e símbolos
- **Rotacione credenciais regularmente** - Especialmente para implantações em produção
- **Restrinja o acesso ao banco de dados** - Use lista de permissões de IP e regras de firewall

## Configuração da Interface do Usuário de Frontend (.env)

Localizado no diretório OmniCRM-UI, este arquivo controla branding, aparência, integrações e flags de recursos.

**Localização:** /OmniCRM/OmniCRM-UI/.env

### Chaves da API e Integração:

```
# API do Google Maps (para autocomplete de endereço e geocodificação)
REACT_APP_GOOGLE_API_KEY=your_google_api_key
```

```
# Gateway de Pagamento Stripe
REACT_APP_STRIPE_PUBLISHABLE_KEY=pk_test_XXXXX
```

```
# Desativar auto-lançamento do navegador ao iniciar npm
BROWSER=none
```

### Branding e Informações da Empresa:

```
# Branding da Empresa
REACT_APP_COMPANY_NAME="ShellFone"
REACT_APP_PORTAL_NAME="ShellManager"
REACT_APP_SELF_CARE_NAME="ShellCare"
REACT_APP_COMPANY_TAGLINE="Phones with Shells"
```

Esses valores aparecem em toda a interface:

- COMPANY\_NAME - Exibido em títulos de páginas, e-mails e comunicações com clientes
- PORTAL\_NAME - Nome do portal de administração/equipe (por exemplo, "ShellManager")
- SELF\_CARE\_NAME - Nome do portal de autoatendimento do cliente (por exemplo, "ShellCare")
- COMPANY\_TAGLINE - Aparece em telas de login e materiais de marketing

### Localização e Configurações Regionais:

```
# Idioma e Localidade
# Idiomas suportados: ar, ch, en, fr, gr, it, ru, sp
```

```
REACT_APP_DEFAULT_LANGUAGE=en
REACT_APP_LOCALE="en-GB"

# Localização Padrão (para autocomplete de endereço)
REACT_APP_DEFAULT_LOCATION="Sydney, Australia"
REACT_APP_DEFAULT_COUNTRY="Australia"

# Configurações de Moeda
REACT_APP_CURRENCY_CODE="GBP"
REACT_APP_CURRENCY_SYMBOL="£"
```

### **Personalização do Esquema de Cores:**

```
# Cor Primária (cor principal da marca)
REACT_APP_PRIMARY_COLOR=#405189

# Opções de Cores Adicionais (exemplos comentados)
# REACT_APP_SECONDARY_COLOR=#2bFFcf
# REACT_APP_TERTIARY_COLOR=#1a9fbf
# REACT_APP_SUCCESS_COLOR=#28a745
# REACT_APP_INFO_COLOR=#17a2b8
# REACT_APP_WARNING_COLOR=#ffc107
# REACT_APP_DANGER_COLOR=#dc3545
```

A cor primária é aplicada a:

- Cabeçalhos de navegação
- Botões de ação
- Links e destaques
- Estados ativos
- Elementos de marca

### **Integrações de Aplicativos Web:**

Configure até 6 aplicativos web de acesso rápido que aparecem no painel de administração:

```
# Aplicativo Web 1: GitHub
REACT_APP_WEB_APP_1_NAME="GitHub"
REACT_APP_WEB_APP_1_URL="https://github.com"
REACT_APP_WEB_APP_1_ICON_PATH="resources/webapp_icons/github.png"

# Aplicativo Web 2: Xero
REACT_APP_WEB_APP_2_NAME="Xero"
REACT_APP_WEB_APP_2_URL="https://go.xero.com/"
REACT_APP_WEB_APP_2_ICON_PATH="resources/webapp_icons/xero.png"

# Aplicativo Web 3-6: Integrações adicionais
# (Configure de forma semelhante com NAME, URL e ICON_PATH)
```

## Monitoramento e Análise:

```
# Integração do Painel Grafana
REACT_APP_GRAFANA_URLS=url1,url2,url3
REACT_APP_GRAFANA_LABELS=label1,label2,label3
REACT_APP_GRAFANA_API_KEY=your-api-key
```

## Flags de Recursos:

```
# URLs de Suporte e Documentação
REACT_APP_FAQS_URL=https://docs.yourcompany.com/faqs
REACT_APP_SUPPORT_URL=https://support.yourcompany.com

# Login Social (Google, Facebook, etc.)
REACT_APP_ALLOW_SOCIAL_LOGINS=yes
```

## Personalização do Logo e Imagem de Inicialização

O OmniCRM permite que você substitua as imagens de branding padrão pelo logo da sua empresa e telas de inicialização sem modificar o código.

### Arquivos de Logo

Os logos são armazenados em /OmniCRM-UI/src/assets/images/omnitouch/ e usam um sistema de fallback:

**Logos Padrão** (sempre presentes):

- DefaultLogoDark.png - Logo do tema escuro (usado em fundos claros)
- DefaultLogoLight.png - Logo do tema claro (usado em fundos escuros)

**Logos Personalizados** (opcionais, têm precedência quando presentes):

- logoSm.png - Logo pequeno para barra lateral colapsada (recomendado: 100x100px)
- logoDark.png - Logo escuro em tamanho real para cabeçalhos (recomendado: 200x50px)
- logoLight.png - Logo claro em tamanho real para telas de autenticação (recomendado: 200x50px)

### Como Funciona o Fallback do Logo:

O sistema tenta carregar logos personalizados primeiro. Se um arquivo de logo personalizado não existir, ele recorre ao padrão:

```
// De Header.js
```

```
const tryImport = (filename) => {
  try {
    return require(`../assets/images/omnitouch/${filename}`);
  } catch (err) {
    return null; // Recorre ao padrão
  }
};

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

### Onde os Logos Aparecem:

- **logoSm.png** - Barra lateral colapsada, navegação móvel, exibições de cabeçalho pequeno
- **logoDark.png** - Barra de cabeçalho principal (modo claro), cabeçalho do painel de administração
- **logoLight.png** - Telas de login/registro, fundos escuros, carrossel de autenticação

### Substituindo Logos:

#### 1. Crie Seus Arquivos de Logo:

- Use o formato PNG para suporte a transparência
- Correspondam às dimensões recomendadas acima
- Certifique-se de que os logos estejam claros em resoluções regulares e retina

#### 2. Adicione ao OmniCRM:

```
# Copie seus arquivos de logo para o diretório de imagens
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. Reconstrua a UI:

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

#### 4. Verifique as Alterações:

- Verifique o cabeçalho do modo claro (deve mostrar logoDark.png)

- Verifique o cabeçalho do modo escuro (deve mostrar logoLight.png)
- Verifique a barra lateral colapsada (deve mostrar logoSm.png)
- Verifique a tela de login (deve mostrar logoLight.png)

## Melhores Práticas de Design de Logo:

- **Contraste** - Certifique-se de que os logos sejam visíveis em fundos claros e escuros
- **Simplicidade** - Os logos devem ser reconhecíveis em tamanhos pequenos
- **Formato** - Use PNG com fundos transparentes
- **Retina** - Forneça resolução 2x para telas de alta DPI
- **Consistência** - Use as mesmas cores da marca em todas as variantes do logo

## Telas de Inicialização e Fundos de Autenticação

As telas de autenticação (login, registro, redefinição de senha) usam um fundo de carrossel com imagens personalizáveis.

**Localização:** /OmniCRM-UI/src/pages/AuthenticationInner/authCarousel.js

### Configuração Padrão:

```
import logoLight from "../../assets/images/logo-light.png";  
  
// Logo exibido nas telas de autenticação  
<img src={logoLight} alt="" height="18" />
```

### Personalizando Telas de Autenticação:

1. Substitua logo-light.png em /OmniCRM-UI/src/assets/images/
2. Adicione CSS de fundo personalizado à classe .auth-one-bg
3. Modifique as citações do carrossel em authCarousel.js para corresponder à sua voz de marca

### Exemplo de Personalização:

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

# Arquivo de Configuração do CRM (crm\_config.yaml)

O arquivo `crm_config.yaml` é a configuração central para a API do OmniCRM, controlando integrações, provisionamento, modelos de e-mail e serviços externos.

**Localização:** `/OmniCRM/OmniCRM-API/crm_config.yaml`

## Configuração do Banco de Dados

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

**Nota:** Isso deve corresponder às suas credenciais de banco de dados do `.env`. Em implantações em contêiner, o servidor é tipicamente `db` (nome do serviço Docker).

## Tipos de Serviço

Defina tipos de serviço válidos para sua implantação:

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

Estes são usados em todo o sistema para:

- Categorização de produtos
- Filtragem de addons (addons correspondem a tipos de serviço)
- Fluxos de trabalho de provisionamento
- Relatórios e análises

## Integração HSS (Home Subscriber Server)

Para operadores de rede móvel com integração HSS:

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

**Configuração:**

- `hss_peers` - Lista de endpoints HSS para provisionamento de assinantes
- `apn_list` - Lista separada por vírgulas de IDs de APN disponíveis para provisionamento

## Configuração do Modelo de E-mail Mailjet

O OmniCRM usa Mailjet para e-mails transacionais. Cada tipo de e-mail tem sua própria configuração de modelo:

```
mailjet:
  api_key: your_mailjet_api_key
  api_secret: your_mailjet_api_secret

# E-mail de Boas-Vindas ao Cliente
api_crmCommunicationCustomerWelcome:
  from_email: "support@yourcompany.com"
  from_name: "Suporte da Sua Empresa"
  template_id: 5977509
  subject: "Bem-vindo à Sua Empresa"

# E-mail de Fatura do Cliente
api_crmCommunicationCustomerInvoice:
  from_email: "billing@yourcompany.com"
  from_name: "Cobrança da Sua Empresa"
  template_id: 6759851
  subject: "Sua Fatura - "

# Lembrete de Fatura
api_crmCommunicationCustomerInvoiceReminder:
  from_email: "billing@yourcompany.com"
  from_name: "Cobrança da Sua Empresa"
  template_id: 5977570
  subject: "Lembrete de Pagamento de Fatura"

# E-mail de Boas-Vindas ao Usuário (Equipe/Admin)
api_crmCommunicationUserWelcome:
  from_email: "admin@yourcompany.com"
  from_name: "Admin da Sua Empresa"
  template_id: 6118112
  subject: "Bem-vindo à Equipe"

# Solicitação de Redefinição de Senha
api_crmCommunicationUserPasswordReset:
  from_email: "security@yourcompany.com"
  from_name: "Segurança da Sua Empresa"
  template_id: 6735666
  subject: "Solicitação de Redefinição de Senha"
```



```

# Confirmação de Sucesso na Redefinição de Senha
api_crmCommunicationUserPasswordResetSuccess:
  from_email: "security@yourcompany.com"
  from_name: "Segurança da Sua Empresa"
  template_id: 6118378
  subject: "Redefinição de Senha Bem-Sucedida"

# Notificação de Mudança de Senha
api_crmCommunicationUserPasswordChange:
  from_email: "security@yourcompany.com"
  from_name: "Segurança da Sua Empresa"
  template_id: 6118423
  subject: "Senha Alterada"

# Verificação de E-mail
api_crmCommunicationEmailVerification:
  from_email: "verify@yourcompany.com"
  from_name: "Verificação da Sua Empresa"
  template_id: 6267350
  subject: "Verifique Seu Endereço de E-mail"

# Notificação de Saldo Expirado
api_crmCommunicationsBalanceExpired:
  from_email: "alerts@yourcompany.com"
  from_name: "Alertas da Sua Empresa"
  template_id: 7238252
  subject: "Saldo de Serviço Expirado"

# Aviso de Saldo Baixo
api_crmCommunicationsBalanceLow:
  from_email: "alerts@yourcompany.com"
  from_name: "Alertas da Sua Empresa"
  template_id: 7238263
  subject: "Aviso de Saldo Baixo"

```

### Criando Modelos Mailjet:

1. Faça login no painel do Mailjet (<<https://app.mailjet.com>>)
2. Navegue para **Transacional** → **Modelos**
3. Crie um novo modelo ou clone um existente
4. Anote o **ID do Modelo** (valor numérico)
5. Adicione variáveis de modelo correspondentes à estrutura de dados do OmniCRM
6. Atualize `crm_config.yaml` com o ID do modelo

### Variáveis de Modelo Disponíveis:

Cada tipo de e-mail recebe variáveis específicas. Exemplos comuns:

- {{customer\_name}} - Nome do cliente ou usuário
- {{service\_name}} - Nome do serviço ou produto
- {{invoice\_id}} - Número da fatura
- {{invoice\_amount}} - Valor total da fatura
- {{due\_date}} - Data de vencimento do pagamento
- {{reset\_link}} - URL de redefinição de senha
- {{verification\_link}} - URL de verificação de e-mail
- {{balance}} - Saldo atual da conta
- {{expiry\_date}} - Data de expiração do saldo ou serviço

## Configuração de Provisionamento

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

### Propósito:

- failure\_list - Endereços de e-mail notificados quando o provisionamento do Ansible falha
- As notificações incluem o nome do playbook, detalhes do erro e informações do cliente
- Permite que a equipe de operações responda rapidamente a problemas de provisionamento

## Configuração de Fatura

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

### Propósito:

Especifica qual modelo HTML Jinja2 usar para a geração de faturas em PDF.

**Localização do Modelo:** /OmniCRM-API/invoice\_templates/

Veja a seção **Geração de PDF de Fatura** abaixo para detalhes sobre como criar modelos personalizados.

## URL Base do CRM

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

### Propósito:

- Usado pelos playbooks do Ansible para fazer callbacks da API
- Usado em modelos de e-mail para gerar links para o CRM
- Deve ser a URL publicamente acessível da sua API (não nomes internos de

contêiner)

### Exemplos:

- Desenvolvimento: `http://localhost:5000`
- Produção: `https://api.yourcompany.com`
- Docker: `http://omnicrm-api:5000` (comunicação interna de contêiner)

## Configuração OCS e CGRates

```
ocs:  
  ocsApi: 'http://10.179.2.142:8080/api'  
  ocsTenant: 'mnc380.mcc313.3gppnetwork.org'  
  cgrates: 'localhost:2080'
```

### Configuração:

- `ocsApi` - Endpoint da API OCS para gerenciamento de assinantes
- `ocsTenant` - Identificador do tenant para implantações OCS multi-tenant
- `cgrates` - Endpoint da API JSON-RPC do CGRates (host:port)

## Configuração do SMSC (Gateway de SMS)

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

### Propósito:

- Enviar notificações SMS para clientes (saldo baixo, alertas de serviço, códigos 2FA)
- `source_msisdn` - ID do remetente exibido para os destinatários (alfanumérico ou número de telefone)
- `smsc_url` - Endpoint da API do gateway SMSC
- `api_key` - Autenticação para a API do SMSC

## Chave Secreta JWT

```
jwt_secret:  
'2b93110f723db60172c8e9a1eaa80027a9a9c3f05b44e50dc3fcf38dba68d87e'
```

### Segurança:

- Usado para assinar e verificar tokens de autenticação
- **DEVE ser alterado do valor padrão** em produção
- Gere uma string aleatória segura (mínimo 64 caracteres)
- Nunca compartilhe ou comite no controle de versão

## Gerando uma Nova Chave Secreta JWT:

```
# Gere uma chave aleatória criptograficamente segura
python3 -c "import secrets; print(secrets.token_hex(32))"
```

## Configuração de Pagamento Stripe

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

### Configuração:

- `secret_key` - Chave API secreta do Stripe (lado do servidor, mantenha confidencial)
- `publishable_key` - Chave publicável do Stripe (lado do cliente, seguro para expor)
- `currency` - Código de moeda ISO 4217 (aud, usd, gbp, eur, etc.)
- `statement_descriptor_suffix` - Aparece nos extratos de cartão de crédito dos clientes

### Uso do Descritor de Extrato:

- Exibido nos extratos bancários dos clientes como "YOURCOMPANY"
- Máximo de 22 caracteres
- Ajuda os clientes a identificar cobranças
- Também usado nos nomes de arquivos PDF de fatura (por exemplo, YOURCOMPANY\_12345.pdf)

## Chaves da API e Lista de Permissões de IP

Defina chaves da API com acesso baseado em funções e restrições 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 de Permissões de IP (autônoma, sem chave da API)
ip_whitelist:
  "10.179.2.142":
    roles: ["admin"]
```

## **Propósito:**

- Permitir que sistemas externos se autentiquem via chave da API
- Restringir o acesso por endereço IP
- Conceder funções específicas aos consumidores da API
- Útil para integrações (sistemas de faturamento, monitoramento, automação)

## **Melhores Práticas de Segurança:**

- Use chaves da API longas e aleatórias (mínimo 32 caracteres)
- Restringir IPs apenas a fontes conhecidas
- Conceder as funções mínimas necessárias
- Rotacione as chaves da API regularmente
- Monitore o uso da chave da API nos logs

# **Geração de PDF de Fatura**

O OmniCRM gera faturas em PDF profissionais usando modelos HTML Jinja2 e renderização de PDF WeasyPrint.

## **Como Funciona a Geração de PDF**

### **1. Seleção de Modelo:**

- O nome do arquivo do modelo é especificado em `crm_config.yaml` sob `invoice.template_filename`
- O modelo é carregado do diretório `/OmniCRM-API/invoice_templates/`

### **2. Preparação de Dados:**

- Os dados da fatura (ID, datas, valores, status) são buscados no banco de dados
- As informações do cliente (nome, endereço) são recuperadas
- A lista de transações é compilada (todas as cobranças/créditos na fatura)

### **3. Renderização do Modelo:**

- Jinja2 renderiza o modelo HTML com os dados da fatura
- Variáveis como `{{ invoice_number }}`, `{{ total_amount }}`, etc. são substituídas
- O HTML renderizado é salvo em `invoice_templates/rendered/` para depuração

### **4. Geração de PDF:**

- WeasyPrint converte o HTML renderizado em PDF
- O PDF suporta estilização CSS, imagens, quebras de página, cabeçalhos/rodapés
- Os dados binários do PDF são gerados na memória

### **5. Cache:**

- O PDF é codificado em Base64 e armazenado na tabela `Invoice_PDF_Cache`

- Um hash SHA256 é calculado para verificação de integridade
- Solicitações subsequentes retornam o PDF em cache (entrega instantânea)

#### 6. **Invalidação de Cache:**

- O cache é invalidado quando a fatura é modificada, anulada ou reembolsada
- Alterações no modelo não invalidam automaticamente caches existentes

## Estrutura do Modelo de Fatura

Os modelos de fatura são arquivos HTML Jinja2 com variáveis e lógica incorporadas.

**Localização do Modelo:** /OmniCRM-API/invoice\_templates/  
yourcompany\_invoice\_template.html

### Variáveis Disponíveis:

```
{
  'invoice_number': 12345,
  'date': '2025-01-04',
  'client': {
    'name': 'John Smith',
    'address': {
      'address_line_1': '123 Main St',
      'city': 'Sydney',
      'state': 'NSW',
      'zip_code': '2000',
      'country': 'Australia'
    }
  },
  'transaction_list': [
    {
      'transaction_id': 1,
      'title': 'Serviço Móvel - Taxa Mensal',
      'retail_cost': 30.00,
      'wholesale_cost': 10.00,
      'created': '2025-01-01'
    },
    {
      'transaction_id': 2,
      'title': 'Addon de Dados - 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
}

```

### Exemplo de Trecho de Modelo:

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Fatura {{ 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>FATURA</h1>
    </div>

    <div class="invoice-details">
        <p><strong>Número da Fatura:</strong> {{ invoice_number
}}</p>
        <p><strong>Data:</strong> {{ date }}</p>
        <p><strong>Data de Vencimento:</strong> {{ due_date }}</p>
        <p><strong>Período de Cobrança:</strong> {{ start_date }} a
{{ end_date }}</p>
    </div>

    <div class="customer-details">
        <h3>Faturar Para:</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>Descrição</th>
                <th>Data</th>
                <th>Valor</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;">
    PAGO
</div>
{% endif %}

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

```

## Melhores Práticas para Modelos:

- **Use caminhos absolutos para imagens** - file:///absolute/path/to/image.png
- **CSS Inline** - WeasyPrint não carrega folhas de estilo externas de forma confiável
- **Teste com dados de exemplo** - Use invoice\_templates/rendered/ para inspecionar HTML
- **Quebras de página** - Use <div style="page-break-after: always;"></div> para faturas de várias páginas
- **Cabeçalhos e rodapés** - Use regras CSS @page para elementos repetidos
- **Formatação de moeda** - Use filtros Jinja2: {{ "%.2f"|format(amount) }}

## Criando um Modelo de Fatura Personalizado

### 1. Copie o Modelo de Exemplo:

```

cd /OmniCRM/OmniCRM-API/invoice_templates
cp norfone_invoice_template.html
yourcompany_invoice_template.html

```

### 2. Edite o Modelo:

- Substitua o nome da empresa, logo, informações de contato
- Ajuste o estilo (cores, fontes, layout) para corresponder à marca
- Adicione ou remova seções conforme necessário (quebras de impostos, instruções de pagamento, etc.)

### 3. Atualize a Configuração:

Edite `crm_config.yaml`:

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

#### 4. **Teste a Geração de Fatura:**

- Crie uma fatura de teste no CRM
- Baixe o PDF e verifique a formatação
- Verifique `invoice_templates/rendered/{invoice_id}.html` para depuração

#### 5. **Invalidate Caches Antigos (se necessário):**

Se você alterou o modelo e deseja regenerar faturas existentes:

```
-- Limpar todos os PDFs em cache (força a regeneração)
DELETE FROM Invoice_PDF_Cache;
```

## Sistema de Cache de PDF

Para melhorar o desempenho, o OmniCRM armazena em cache PDFs gerados:

### Comportamento do Cache:

- **Primeira Solicitação** - O PDF é gerado, armazenado em cache e retornado
- **Solicitações Subsequentes** - O PDF em cache é retornado imediatamente (sem regeneração)
- **Invalidação de Cache** - Ocorre quando a fatura é modificada, anulada ou reembolsada
- **Limpeza de Cache** - Caches antigos são automaticamente purgados após 30 dias de inatividade

### Armazenamento de Cache:

- PDF codificado em Base64 armazenado na tabela `Invoice_PDF_Cache`
- Hash de conteúdo SHA256 para verificação de integridade
- Inclui nome do arquivo, timestamp de criação, timestamp do último acesso

### Gerenciamento Manual de Cache:

```
# No OmniCRM API ou shell 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()
```

```
# Limpar caches com mais de 30 dias
result = cleanup_old_pdf_cache(session, days_old=30)
print(result) # {'status': 'success', 'deleted_count': 15}

# Invalidar cache de fatura específica
invalidate_invoice_cache(session, invoice_id='12345')
```

## Endpoints da API:

Gerar/download PDF de fatura:

GET /invoice/pdf/{invoice\_id}

**Resposta:** Download do arquivo PDF com nome do descritor de extrato do Stripe

## Cabeçalhos de Cache:

- Primeira solicitação: Resposta mais lenta (tempo de geração)
- Solicitações em cache: Resposta instantânea
- Acerto/falha de cache é transparente para o usuário

## Solução de Problemas

### Falha na Geração de PDF:

- Verifique se o WeasyPrint está instalado: `pip install weasyprint`
- Verifique se o nome do modelo corresponde ao `crm_config.yaml`
- Verifique `invoice_templates/rendered/` para erros de renderização HTML
- Revise os logs da API para erros de modelo Jinja2

### Imagens Não Aparecendo no PDF:

- Use caminhos de arquivo absolutos: `file:///full/path/to/image.png`
- Certifique-se de que os arquivos de imagem existam e sejam legíveis
- Verifique o formato da imagem (PNG e JPEG funcionam melhor)
- Verifique se os caminhos das imagens não contêm caracteres especiais

### Problemas de Estilo:

- Inline todos os CSS (folhas de estilo externas não suportadas)
- Evite recursos CSS complexos (flexbox, grid podem não renderizar corretamente)
- Teste com layouts simples primeiro, adicione complexidade gradualmente
- Use tabelas para layout em vez de divs sempre que possível

### Cache Não Invalida:

- Verifique se `invalidate_invoice_cache()` é chamado quando a fatura é

modificada

- Verifique se atualizações de transações acionam a invalidação do cache
- Exclua manualmente da tabela Invoice\_PDF\_Cache se necessário

## Lista de Verificação de Configuração

Use esta lista de verificação ao implantar o OmniCRM:

### Configuração de Backend

- ☐ Copie .env.example para .env
- ☐ Defina senhas fortes para o banco de dados
- ☐ Configure credenciais do CGRates
- ☐ Atualize crm\_config.yaml com suas configurações:
  - ☐ Conexão com o banco de dados
  - ☐ Tipos de serviço
  - ☐ Chaves da API do Mailjet e IDs de modelo
  - ☐ E-mails de notificação de falha de provisionamento
  - ☐ Nome do arquivo do modelo de fatura
  - ☐ URL base do CRM (publicamente acessível)
  - ☐ Endpoints OCS/CGRates
  - ☐ Configuração do SMSC
  - ☐ Gere uma nova chave secreta JWT
  - ☐ Chaves do Stripe (ao vivo, não teste)
  - ☐ Chaves da API e lista de permissões de IP

### Configuração de Frontend

- ☐ Copie OmniCRM-UI/.env.example para OmniCRM-UI/.env
- ☐ Defina a chave da API do Google Maps
- ☐ Defina a chave publicável do Stripe
- ☐ Atualize o branding da empresa:
  - ☐ Nome da empresa
  - ☐ Nome do portal
  - ☐ Nome de autoatendimento
  - ☐ Slogan da empresa
- ☐ Configure a localização:
  - ☐ Idioma padrão
  - ☐ Localidade
  - ☐ Localização e país padrão
  - ☐ Código e símbolo da moeda
- ☐ Defina a cor primária da marca
- ☐ Configure integrações de aplicativos web (opcional)
- ☐ Adicione URLs de suporte e FAQ (opcional)

## Ativos de Branding

- [ ] Crie arquivos de logo (logoSm.png, logoDark.png, logoLight.png)
- [ ] Faça upload dos logos para OmniCRM-UI/src/assets/images/omnitouch/
- [ ] Crie HTML de modelo de fatura personalizado
- [ ] Faça upload do modelo de fatura para OmniCRM-API/invoice\_templates/
- [ ] Atualize crm\_config.yaml com o nome do arquivo do modelo de fatura
- [ ] Teste a geração de PDF de fatura
- [ ] Reconstrua a UI: npm run build

## Segurança

- [ ] Altere todas as senhas padrão
- [ ] Gere uma chave JWT única
- [ ] Use chaves Stripe de produção (não chaves de teste)
- [ ] Rotacione as chaves da API do Mailjet
- [ ] Habilite regras de firewall
- [ ] Configure a lista de permissões de IP para acesso à API
- [ ] Configure certificados SSL/TLS
- [ ] Habilite HTTPS para todos os endpoints
- [ ] Revise as configurações de CORS
- [ ] Implemente limitação de taxa
- [ ] Configure procedimentos de backup e recuperação

## Testes

- [ ] Teste o fluxo de registro de clientes
- [ ] Teste o provisionamento de serviços de ponta a ponta
- [ ] Verifique se as notificações por e-mail são enviadas corretamente
- [ ] Teste a geração de faturas e download de PDF
- [ ] Verifique o processamento de pagamentos (Stripe)
- [ ] Verifique a autenticação de usuários e 2FA
- [ ] Teste a impersonação e o registro de auditoria
- [ ] Verifique se os dados de uso sincronizam do OCS
- [ ] Teste a criação e renovação de ActionPlan
- [ ] Confirme se a alocação de inventário funciona corretamente

## Implantação

- [ ] Construa imagens Docker ou implante em servidores
- [ ] Inicie contêineres de banco de dados (MySQL, PostgreSQL)
- [ ] Inicie o CGRates
- [ ] Inicie a API do OmniCRM
- [ ] Inicie a UI do OmniCRM
- [ ] Configure o proxy reverso (nginx, traefik)
- [ ] Configure monitoramento (Grafana, Prometheus)

- [ ] Configure agregação de logs
- [ ] Configure backups automatizados
- [ ] Documente a arquitetura de implantação
- [ ] Treine a equipe sobre o uso do sistema

## **Documentação Relacionada**

- RBAC e Gerenciamento de Usuários </rbac>
- Produtos e Serviços </concepts\_products\_and\_services>
- Provisionamento Ansible </concepts\_ansible>
- Gerenciamento de Inventário </administration\_inventory>
- Faturas de Clientes </payments\_invoices>
- Autenticação de Dois Fatores </2fa>
- Atendimento ao Cliente e Impersonação </customer\_care>



# Visão Geral do Inventário no OmniCRM

O sistema de **Inventário** no OmniCRM é projetado para gerenciar e rastrear tanto itens físicos quanto virtuais usados por operadores de rede e clientes.

Isso significa que podemos rastrear todos os tipos de itens, como modems, números de telefone, blocos de endereços IP ou até mesmo hardware físico como ONTs GPON ou CPEs de Wireless Fixo.

Veja também: Atributos do Cliente <administration\_attributes> para armazenar metadados personalizados, e Tags do Cliente <administration\_tags> para categorização visual.

Para apoiar clientes com uma pegada de rede fixa, o sistema de inventário também pode rastrear residências atendidas para um determinado serviço, permitindo que os operadores realizem qualificação de serviço remotamente, e para aqueles que operam uma rede wireless fixa, podemos rastrear os CPEs implantados em campo com suas localizações.

Nota

Os itens de inventário estão vinculados a produtos durante o provisionamento através do campo `inventory_items_list`. Para um guia completo de como o inventário se integra ao provisionamento de produtos, incluindo a interface do usuário do seletor de inventário e a integração do playbook do Ansible, veja Guia Completo do Ciclo de Vida do Produto - Requisitos de Inventário <guide\_product\_lifecycle>.

## Propósito

O Inventário do OmniCRM serve a vários propósitos-chave:

1. **Provisionamento de Serviços:** Quando um cliente se inscreve para um serviço, itens como modems, cartões SIM ou números de telefone podem precisar ser alocados. O sistema de inventário rastreia esses itens e os associa aos clientes.
2. **Gestão de Estoque:** Para estoque físico, como hardware ou outros equipamentos, o inventário ajuda os operadores a manter visibilidade sobre o que está disponível, onde está armazenado e o que foi alocado ou vendido aos clientes.
3. **Alocação de Clientes:** O sistema permite que itens sejam alocados a clientes, seja para uso em um serviço (por exemplo, atribuindo um modem

- à conta de internet de um cliente) ou para venda direta.
4. **Qualificação de Serviço / Pegada de Rede:** Ao armazenar informações sobre a pegada de rede, como cada residência atendida para um serviço GPON, ou cada CPE Wireless Fixo implantado, permite que a equipe realize qualificação de serviço remotamente e veja se há uma pegada de rede em uma área específica.

## Exemplo do Ciclo de Vida do Inventário

Para ilustrar como o sistema de Inventário funciona, considere os seguintes exemplos:

### Exemplo de Cartão SIM

Um lote de 1000 cartões SIM é encomendado da Omnitouch. Primeiramente, um Modelo de Inventário é criado para cartões SIM (se ainda não existir) e os SIMs encomendados são carregados no Inventário no estado Em Trânsito.

Uma vez que os SIMs são recebidos, eles são marcados como Em Estoque, e podem ser alocados para diferentes lojas de varejo, com a Localização do item de inventário atualizada para refletir em qual loja de varejo cada cartão SIM está - Isso é útil para rastrear os níveis de estoque em cada loja e garantir que cada loja tenha estoque suficiente para atender à demanda dos clientes.

Quando um cliente se inscreve para um serviço móvel na loja, um cartão SIM é alocado ao cliente e o status é alterado para Alocado. O item de inventário do cartão SIM é então atribuído ao cliente, e o status é atualizado para Em Uso.

Se o cliente cancelar o serviço ou o serviço ficar inativo, o cartão SIM é marcado como Desativado.

### Exemplo de Residências Atendidas GPON

Para uma rede GPON, o sistema de inventário pode rastrear cada residência atendida para um determinado serviço.

Quando uma nova área é construída, cada endereço atendido pode ser adicionado ao inventário.

Isso permite que os operadores vejam quais residências estão atendidas para um determinado serviço e quais residências ainda não estão atendidas.

Quando um cliente se inscreve para um serviço, o OmniCRM pode executar automaticamente uma qualificação de serviço contra o endereço do cliente, para ver se o endereço está no inventário de residências atendidas e quais serviços podem ser oferecidos.



# Modelos de Inventário

O recurso **InventoryTemplate** permite a criação de qualquer número de tipos de itens com campos predefinidos. Esses modelos atuam como plantas que definem as características essenciais de diferentes itens, como:

- **Modems** com um endereço MAC (itemtext1) e um número de série (itemtext2).
- **Residências Atendidas** para um determinado serviço, com uma localização e status (por exemplo, atendida ou não atendida).
- **Números de Telefone** com um número principal (itemtext1) e uma localização geográfica (itemtext2).
- **Recursos Virtuais** como blocos de endereços IP, com identificadores mapeados através do modelo.

Cada modelo de inventário define até 20 campos de texto personalizáveis (itemtext1 até itemtext20) com rótulos correspondentes (itemtext1\_label até itemtext20\_label) que descrevem o que cada campo representa. Por exemplo, um modelo de Modem pode definir itemtext1\_label como "Endereço MAC" e itemtext2\_label como "Número de Série".

Os operadores podem personalizar os campos para cada tipo de item usando **InventoryTemplates**. Esses modelos permitem que os itens sejam categorizados e gerenciados de maneira estruturada, garantindo consistência em como os itens são rastreados.

## Vinculando a Produtos:

Os nomes dos modelos de inventário são referenciados nas definições de produtos através do campo `inventory_items_list`. Ao provisionar um produto, o sistema exibe um seletor de inventário mostrando apenas itens que correspondem aos tipos de modelo exigidos.

**Exemplo:** Um produto com `inventory_items_list`: `"['SIM Card', 'Mobile Number']"` requer que dois modelos de inventário chamados exatamente "SIM Card" e "Mobile Number" existam. Os nomes dos modelos são sensíveis a maiúsculas e minúsculas.

Para detalhes completos sobre como os modelos de inventário se conectam ao provisionamento de produtos, veja [Ciclo de Vida do Produto - Requisitos de Inventário <guide\\_product\\_lifecycle>](#).

Visualizando modelos no OmniCRM

## Criando Modelos de Inventário via a UI

Para criar um novo modelo de inventário:

1. Navegue até **Inventário** → **Modelos** no menu principal
2. Clique no botão **Adicionar Modelo**
3. Preencha os campos obrigatórios:

#### **Informações Básicas:**

- **Ícone** (opcional): Nome da classe do ícone para identificação visual (por exemplo, fa-solid fa-sim-card)
- **Item** (obrigatório): O nome do modelo (deve corresponder exatamente ao que é usado em `inventory_items_list` para produtos)

#### **Informações de Custo (Obrigatório):**

- **Custo Atacado** (obrigatório): Seu custo para comprar ou provisionar este tipo de item
- **Custo de Varejo** (obrigatório): Preço de varejo padrão se vendido separadamente aos clientes

#### **Nota**

Os custos atacadistas e de varejo definidos aqui servem como valores padrão ao criar novos itens de inventário a partir deste modelo. Itens de inventário individuais podem ter custos diferentes, se necessário.

#### **Rótulos de Campo:**

- **Rótulo do Texto do Item 1** (obrigatório): Rótulo para o primeiro campo personalizável (padrão para "Número do Modelo")
  - Exemplos comuns: "ICCID" para cartões SIM, "Endereço MAC" para modems, "Número de Telefone" para números
- **Rótulo do Texto do Item 2** (obrigatório): Rótulo para o segundo campo personalizável (padrão para "Número de Série")
  - Exemplos comuns: "IMSI" para cartões SIM, "Número de Série" para hardware, "Região Geográfica" para números
- **Rótulos do Texto do Item 3-20** (opcional): Rótulos de campo adicionais conforme necessário
  - Clique em **Adicionar Campo** para adicionar mais campos personalizados
  - Adicione apenas campos que você realmente usará para este tipo de item

#### **Configurações de Visibilidade:**

- **Permitir Dropdown para Funcionários:** Permitir que os funcionários selecionem este tipo de inventário em dropdowns
- **Permitir Dropdown para Clientes:** Permitir que os clientes vejam/

selecionem este tipo de inventário (portal do cliente)

4. Clique em **Salvar** para criar o modelo

## **Editando Modelos de Inventário**

Para editar um modelo existente:

1. Navegue até **Inventário** → **Modelos**
2. Encontre o modelo na lista
3. Clique no botão **Editar**
4. Modifique os campos conforme necessário
5. Clique em **Salvar**

### **Aviso**

Alterar rótulos de campo (por exemplo, `itemtext1_label`) afeta apenas novos itens criados após a alteração. Itens de inventário existentes mantêm seus dados, mas serão exibidos com os novos nomes de rótulo.

### **Cuidado**

Os nomes dos modelos referenciados nos campos `inventory_items_list` de produtos são sensíveis a maiúsculas e minúsculas. Renomear um modelo quebrará o vínculo com produtos que usam o nome antigo.

## **Exemplos Comuns de Modelos**

### **Modelo de Cartão SIM:**

- Item: "SIM Card"
- Custo Atacado: 2.50
- Custo de Varejo: 10.00
- Rótulo do Texto do Item 1: "ICCID"
- Rótulo do Texto do Item 2: "IMSI"
- Rótulo do Texto do Item 3: "Tipo de SIM" (Físico/eSIM)

### **Modelo de Número Móvel:**

- Item: "Mobile Number"
- Custo Atacado: 1.00
- Custo de Varejo: 0.00
- Rótulo do Texto do Item 1: "Número de Telefone"
- Rótulo do Texto do Item 2: "Região Geográfica"
- Rótulo do Texto do Item 3: "Tipo de Número" (Móvel/Fixo)

### **Modelo de CPE Wireless Fixo:**

- Item: "Fixed Wireless CPE"
- Custo Atacado: 250.00
- Custo de Varejo: 450.00
- Rótulo do Texto do Item 1: "Endereço MAC"
- Rótulo do Texto do Item 2: "Número de Série"
- Rótulo do Texto do Item 3: "Versão do Firmware"
- Rótulo do Texto do Item 4: "Fabricante"
- Rótulo do Texto do Item 5: "Modelo"

### Modelo de ONT GPON:

- Item: "GPON ONT"
- Custo Atacado: 45.00
- Custo de Varejo: 0.00 (incluído com o serviço)
- Rótulo do Texto do Item 1: "Número de Série"
- Rótulo do Texto do Item 2: "Endereço MAC"
- Rótulo do Texto do Item 3: "Localização PON"
- Rótulo do Texto do Item 4: "Modelo"

Criando um Item a partir de um Modelo no OmniCRM

## Criando e Gerenciando Itens de Inventário

Uma vez que um **InventoryTemplate** é definido, itens individuais de **Inventário** podem ser criados. Cada item de inventário representa uma instância específica de um tipo de item (por exemplo, um modem ou número de telefone específico) que pode ser:

- **Alocado a Clientes:** Itens são vinculados a clientes para provisionamento de serviços (por exemplo, atribuindo hardware para uma conexão de internet).
- **Rastreados para Estoque:** Os operadores podem monitorar o inventário disponível, como itens não vendidos ou não atribuídos.
- **Vendidos ou Desativados:** Uma vez vendidos, os itens são marcados com timestamps relevantes (por exemplo, `sold_date`) e não podem mais ser considerados estoque disponível.

### Visualização da Lista de Inventário Editar Item de Inventário

Através deste sistema, o OmniCRM facilita a gestão eficiente de estoque, ajuda a alocar recursos para clientes e fornece visibilidade detalhada sobre o status e a história de cada item.

**Serviços** podem ser vinculados a um determinado item de **Inventário**, permitindo um rastreamento fácil de quais itens estão associados a quais clientes ou serviços.

Uma vez que um item de inventário foi atribuído a um cliente, os plays do Ansible

podem atualizar o status e a história do item para refletir a alocação. Isso garante que os operadores tenham um registro preciso de quais itens estão em uso e quais estão disponíveis para alocação, além de saber qual cliente está usando qual item.

Podemos visualizar itens alocados a um cliente a partir da página de perfil do cliente na aba **Inventário**.

### Visualização do Inventário do Cliente

Para itens de Inventário vinculados a um **Serviço**, podemos ver isso editando o serviço, para ver os itens de Inventário vinculados.

Visualizando Itens de Inventário vinculados a um Serviço no OmniCRM

## Campos do Item de Inventário

Cada item de inventário contém informações abrangentes organizadas em várias categorias:

### Informações Básicas do Item

- **inventory\_id** - Identificador único para o item de inventário (gerado automaticamente)
- **item** - Tipo de item (corresponde ao nome do Modelo de Inventário, por exemplo, "SIM Card", "Modem", "Phone Number")
- **inventory\_template\_id** - Link para o Modelo de Inventário que define este tipo de item
- **customer\_id** - Se atribuído a um cliente, o ID do cliente (nulo)
- **service\_id** - Se vinculado a um serviço específico, o ID do serviço (nulo)
- **item\_location** - Localização física ou lógica do item:
  - Para itens físicos: prédio, armazém, localização da prateleira, nome da loja, etc.
  - Para itens virtuais: localização geográfica, localização do bloco IP, região do intervalo de números, etc.
- **item\_state** - Estado atual do item de inventário (valores enumerados):
  - **Novo** - Item novo, não utilizado
  - **Usado** - Anteriormente utilizado, mas funcional
  - **Uso Interno** - Alocado para testes internos ou uso de funcionários
  - **Atribuído** - Atualmente atribuído a um cliente ou serviço
  - **Danificado** - Não funcional, requer reparo ou descarte
  - **Fora de Serviço** - Temporariamente indisponível
  - **Perdido** - Item não pode ser localizado
  - **Roubado** - Item foi roubado

### Campos Personalizáveis do Item (do Modelo)

O sistema de inventário suporta até 20 campos de texto personalizáveis cujo

significado é definido pelo Modelo de Inventário:

- **itemtext1** - Primeiro campo personalizável (obrigatório, rótulo definido pelo `itemtext1_label` do modelo)
  - Exemplo: Para modems, pode ser "Endereço MAC"
  - Exemplo: Para cartões SIM, pode ser "ICCID"
  - Exemplo: Para números de telefone, pode ser "Número de Telefone"
- **itemtext2** até **itemtext20** - Campos personalizáveis adicionais (opcional, rótulos definidos pelo modelo)
  - Exemplo: `itemtext2` para modems pode ser "Número de Série"
  - Exemplo: `itemtext2` para cartões SIM pode ser "IMSI"
  - Exemplo: `itemtext3` para modems pode ser "Versão do Firmware"

Cada Modelo de Inventário define quais desses campos são usados e o que representam através dos campos de rótulo correspondentes (`itemtext1_label`, `itemtext2_label`, etc.).

### Informações de Custo

- **wholesale\_cost** - Seu custo para comprar/provisionar este item (float)
- **retail\_cost** - Preço cobrado ao cliente se vendido separadamente (float)
- **sold\_date** - Timestamp quando o item foi vendido ou atribuído ao cliente

### Endereço Físico (para Equipamentos e Locais de Rede)

Usado para rastrear locais de implantação física, particularmente para equipamentos de rede fixa (CPEs, ONTs, modems) ou residências atendidas:

- **address\_line\_1** - Endereço da rua, número do prédio, número da unidade
- **address\_line\_2** - Informações adicionais do endereço (suíte, apartamento, andar)
- **city** - Cidade ou município
- **state** - Estado, província ou região
- **zip\_code** - Código postal/ZIP
- **country** - Nome do país

### Localização Geográfica (Preenchido Automaticamente pela Web UI)

Ao criar itens de inventário via a Web UI com autocompletar de endereço, esses campos são preenchidos automaticamente:

- **google\_maps\_place\_id** - Google Maps Place ID para o endereço
- **plus\_code** - Google Maps Plus Code (Open Location Code) para localização precisa
- **latitude** - Latitude geográfica (graus decimais)
- **longitude** - Longitude geográfica (graus decimais)

Esses campos possibilitam:

- Mapeamento de locais de inventário em uma visualização de mapa
- Cálculos de proximidade para qualificação de serviço
- Análise de cobertura para planejamento de rede
- Otimização de rotas para despacho de técnicos de campo

## Gerenciamento de Dispositivos e URLs de Acesso

O campo `management_url` fornece acesso rápido às interfaces de dispositivos e URLs de provisionamento:

- **management\_url** - URL de acesso para o item de inventário
  - **Equipamentos de Rede:** URL da interface web (por exemplo, `https://192.168.1.1` para roteadores, switches, ONTs, CPEs)
  - **Perfis de eSIM:** endereço LPA (Local Profile Assistant) para ativação de eSIM (por exemplo, `LPA:1$smgp.example.com$ACTIVATION-CODE-HERE`)
  - **Outros Casos de Uso:** Qualquer URL que precise ser facilmente acessível via dispositivo móvel

## Geração de QR Code

Ao visualizar itens de inventário com um `management_url`, o sistema gera automaticamente um **QR code escaneável**:

- **Visualização de detalhes do item de inventário:** QR code 128x128 exibido ao lado da URL
- **Tabela de inventário de serviços:** QR code 64x64 mostrado para itens atribuídos
- **Formato:** Tanto o QR code quanto o hyperlink clicável exibidos juntos

## Casos de Uso Comuns

- **Técnicos de Rede:** Escaneie o QR code para acessar a interface de gerenciamento do dispositivo sem digitar endereços IP
  - **Ativação de eSIM:** Clientes escaneiam o QR code do CRM para instalar o perfil de eSIM em seu dispositivo
  - **Autoatendimento do Cliente:** Fornecer acesso fácil à configuração do dispositivo ou portais de clientes
- **management\_username** - Nome de usuário admin para acesso ao dispositivo
  - **management\_password** - Senha admin para acesso ao dispositivo (criptografada em repouso)

## Gerenciamento de Configuração

Para dispositivos com arquivos de configuração:

- **config\_content** - Conteúdo completo do arquivo de configuração (armazenado como texto)
  - Útil para backup, versionamento e recuperação de desastres
  - Pode armazenar configurações de roteadores, configurações de switches, configurações de CPEs, etc.
- **config\_file\_path** - Caminho para o arquivo de configuração externo se armazenado separadamente
  - Alternativa para armazenar a configuração completa no banco de dados
  - Caminho para compartilhamento de rede, repositório de controle de versão ou sistema de gerenciamento de configuração

## Notas e Metadados

- **inventory\_notes** - Notas em formato livre sobre o item de inventário
  - Notas de instalação
  - Histórico de manutenção
  - Particularidades ou requisitos especiais de manuseio
  - Informações do fornecedor
  - Detalhes da garantia
- **created** - Timestamp quando o item de inventário foi criado no sistema (definido automaticamente)
- **last\_modified** - Timestamp da última atualização do item de inventário (atualizado automaticamente)

## Exemplos de Uso de Campos

### Exemplo 1: Cartão SIM Móvel

```
{
  "inventory_id": 1001,
  "item": "SIM Card",
  "inventory_template_id": 5,
  "itemtext1": "8961234567890123456",
  "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"
}
```



## Exemplo 2: Perfil de eSIM Móvel

```
{
  "inventory_id": 1002,
  "item": "eSIM",
  "inventory_template_id": 6,
  "itemtext1": "8961234567890123457",
  "itemtext2": "310120123456790",
  "itemtext3": "eSIM",
  "item_location": "Virtual Inventory",
  "item_state": "Assigned",
  "customer_id": 457,
  "service_id": 790,
  "management_url": "LPA:1$sm dp.example.com$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"
}
```

Ao visualizar este item de inventário de eSIM, a interface do usuário exibe um QR code contendo o endereço LPA. Os clientes escaneiam este QR code com seu dispositivo móvel para instalar o perfil de eSIM.

## Exemplo 3: Equipamento de Cliente (CPE) - Wireless Fixo

```
{
  "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°"
}
```

#### Exemplo 4: ONT GPON com Endereço Completo

```
{
  "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

Ao visualizar itens de inventário com um `management_url` (como Exemplos 2, 3 e 4 acima), a interface do usuário exibe automaticamente:

- Um QR code escaneável contendo a URL ou endereço LPA
- Um hyperlink clicável (para URLs da web)

#### Casos de Uso:

- **Ativação de eSIM** (Exemplo 2): Clientes escaneiam o QR code para instalar o perfil de eSIM em seu dispositivo
- **Acesso a Equipamentos de Rede** (Exemplos 3 e 4): Técnicos escaneiam para acessar interfaces de gerenciamento de dispositivos sem digitar manualmente endereços IP

### Exemplo 5: Número de Telefone (Inventário 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 do Item de Inventário Explicados

O campo `item_state` rastreia o ciclo de vida dos itens de inventário:

- **Novo** → **Atribuído** - Fluxo normal ao provisionar para um cliente
- **Atribuído** → **Usado** - Após a desativação do serviço, o item pode ser reutilizado
- **Novo** → **Uso Interno** - Alocado para testes, demonstrações ou uso de funcionários
- **Atribuído** → **Danificado** - Item falhou, requer RMA ou descarte
- **Qualquer Estado** → **Perdido** - Item não pode ser localizado (aciona auditoria)
- **Qualquer Estado** → **Roubado** - Item foi roubado (aciona relatório de segurança)
- **Danificado/Usado** → **Novo** - Após reforma ou reparo

Filtrar e pesquisar inventário por estado permite que os operadores:

- Rastrear estoque disponível (itens Novos)
- Identificar itens atribuídos a clientes (Atribuído)
- Encontrar itens disponíveis para reutilização (Usado)
- Monitorar problemas de equipamentos (Danificado, Fora de Serviço)
- Auditar itens faltantes (Perdido, Roubado)



# Etiquetas de Cliente

As etiquetas são links codificados por cores que podem ser adicionados a um cliente para ajudar a categorizá-los, por exemplo, um cliente pode ter uma etiqueta para "Chamado de Suporte Aberto" ou "Fatura Super Atrasada" ou "Desagradável".

Para armazenar metadados estruturados e dados personalizados de chave-valor, consulte Atributos do Cliente `<administration_attributes>`.

As etiquetas são exibidas como pílulas na página de perfil do cliente, e a cor da pílula é personalizável junto com o link.

Um caso de uso comum é etiquetar clientes que têm um chamado de suporte aberto, para que a equipe de suporte possa facilmente acessar o chamado aberto a partir da página de perfil do cliente.

As etiquetas podem ser criadas no sistema por um administrador através da interface do usuário ou por sistemas de terceiros via API e podem ter datas de início e término, para que possam ser removidas automaticamente após um certo período.

Etiquetas

## Gerenciando Etiquetas via a UI

### Visualizando Etiquetas de Cliente

Para visualizar etiquetas de um cliente:

1. Navegue até a página de visão geral do cliente
2. Clique na aba **Etiquetas**
3. Você verá uma lista de todas as etiquetas ativas para o cliente, mostrando:
  - Visualização da etiqueta com a cor configurada
  - Texto da etiqueta
  - Data de ativação (quando a etiqueta se torna visível)
  - Data de desativação (quando a etiqueta será ocultada)
  - Link (se configurado)

### Criando uma Nova Etiqueta

Para criar uma nova etiqueta para um cliente:

1. Navegue até a página de visão geral do cliente

2. Clique na aba **Etiquetas**
3. Clique no botão **Adicionar Etiqueta**
4. Preencha os campos obrigatórios:
  - **Texto da Etiqueta** (obrigatório): O texto que será exibido na etiqueta
  - **Cor da Etiqueta** (obrigatório): Escolha uma cor usando o seletor de cores ou insira um código hex
  - **Link da Etiqueta** (opcional): URL que será aberta quando a etiqueta for clicada
  - **Data de Ativação** (obrigatório): Data em que a etiqueta deve começar a ser exibida
  - **Data de Desativação** (obrigatório): Data em que a etiqueta deve parar de ser exibida (padrão é 2099-01-01)
5. Visualize sua etiqueta na seção de pré-visualização
6. Clique em **Criar Etiqueta**

Formulário de Criação de Etiqueta de Cliente Exibição de Etiquetas de Cliente

## Editando uma Etiqueta

Para editar uma etiqueta existente:

1. Navegue até a página de visão geral do cliente
2. Clique na aba **Etiquetas**
3. Encontre a etiqueta que você deseja editar na lista
4. Clique no botão **Editar** (lápis)
5. Modifique os campos conforme necessário
6. Clique em **Atualizar Etiqueta**

## Excluindo uma Etiqueta

Para excluir uma etiqueta:

1. Navegue até a página de visão geral do cliente
2. Clique na aba **Etiquetas**
3. Encontre a etiqueta que você deseja excluir na lista
4. Clique no botão **Excluir** (lixeira)
5. Confirme a exclusão na janela pop-up

## Referência de Campo de Etiqueta

## Integração com API

As etiquetas também podem ser gerenciadas programaticamente via API:

**Criar uma 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
}
```

### **Atualizar uma Etiqueta:**

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

### **Obter Etiquetas por Cliente:**

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

### **Excluir uma Etiqueta:**

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



# Arquitetura do Sistema OmniCRM

Este documento fornece uma visão geral da arquitetura do sistema OmniCRM, incluindo relacionamentos entre componentes e fluxo de dados.

## Visão Geral do Sistema em Alto Nível

OmniCRM é uma plataforma abrangente BSS/OSS que integra vários componentes-chave para fornecer gerenciamento completo de serviços para provedores de telecomunicações.

## Componentes Principais

### 1. Aplicações de Frontend

#### OmniCRM Web UI

- Aplicação de página única em React
- Interface para funcionários para gerenciamento de clientes, provisionamento de serviços, faturamento
- Atualizações de status de provisionamento em tempo real
- Controle de acesso baseado em funções

#### Portal de Autoatendimento

- Portal voltado para o cliente
- Gerenciamento de serviços e rastreamento de uso
- Visualização e pagamento de faturas
- Código compartilhado com a interface de funcionários, diferentes visualizações

### 2. Camada de API

#### OmniCRM API (Flask/Python)

- API RESTful para todas as operações
- Documentação OpenAPI/Swagger
- Autenticação baseada em JWT
- Limitação de taxa e cache
- Suporte a WebSocket para atualizações em tempo real

#### Principais Rotas da API:

- /crm/customer/\* - Gerenciamento de clientes
- /crm/service/\* - Operações de serviço
- /crm/product/\* - Catálogo de produtos
- /crm/provision/\* - Operações de provisionamento
- /crm/transaction/\* - Transações de faturamento
- /crm/invoice/\* - Gerenciamento de faturas

### 3. Sistema de Provisionamento

### 4. Faturamento & Classificação

#### Integração OCS/CGRateS

- Cobrança e classificação em tempo real
- Gerenciamento de contas
- Rastreamento de saldo (monetário, dados, voz, SMS)
- Planos de ação para cobranças recorrentes
- Notificações baseadas em limites

#### Fluxo de Trabalho de Faturamento:

### 5. Modelo de Dados

OmniCRM utiliza um banco de dados relacional com os seguintes modelos principais. Para diagramas visuais de relacionamento de entidades, veja [Clientes](#), [Contatos](#), [Locais & Serviços](#).

#### Cliente & Entidades Relacionadas

**Cliente** - Entidade central representando uma empresa ou indivíduo

<b>Campo</b>	<b>Tipo</b>	<b>Descrição</b>
customer_id	Inteiro (PK)	Identificador único
customer_name	String	Nome da empresa ou indivíduo
customer_account_type	Enum	'Individual' ou 'Business'
customer_status	Enum	'Open', 'Closed', 'Suspended', 'Archived'
customer_payment_type	String	'pré-pago' ou 'pós-pago'
customer_enabled	Boolean	A conta está ativa
tax_identifier	String	Número VAT/GST
contract_start_date	DateTime	Início do contrato
contract_end_date	DateTime	Fim do contrato

**Contato** - Pessoas associadas a um cliente

<b>Campo</b>	<b>Tipo</b>	<b>Descrição</b>
contact_id	Inteiro (PK)	Identificador único



<b>Campo</b>	<b>Tipo</b>	<b>Descrição</b>
customer_id	Inteiro (FK)	Cliente pai
contact_firstname	String	Primeiro nome
contact_lastname	String	Último nome
contact_email	String	Endereço de email
contact_phone	String	Número de telefone
contact_types	String	'Primário', 'Faturamento', 'Técnico'

**Local** - Locais físicos de entrega de serviços

<b>Campo</b>	<b>Tipo</b>	<b>Descrição</b>
site_id	Inteiro (PK)	Identificador único
customer_id	Inteiro (FK)	Cliente pai
site_name	String	Nome do local
address_line_1	String	Endereço da rua
city, state, zip_code	String	Detalhes do local
latitude, longitude	Float	Coordenadas GPS
google_maps_place_id	String	Referência do Google Maps
plus_code	String	Código de Localização Aberto

## **Modelos de Serviço & Produto**

**Serviço** - Instâncias de serviço ativas

<b>Campo</b>	<b>Tipo</b>	<b>Descrição</b>
service_id	Inteiro (PK)	Identificador único
customer_id	Inteiro (FK)	Cliente pai
product_id	Inteiro (FK)	Modelo de produto
site_id	Inteiro (FK)	Local do serviço
service_name	String	Nome exibido
service_uuid	String	Identificador do sistema de faturamento
service_status	Enum	Status atual
service_billed	Boolean	Gerar transações
wholesale_cost	Float	Custo do provedor
retail_cost	Float	Preço do cliente
bundled_parent	Inteiro (FK)	Serviço pai para pacotes

**Produto** - Modelos de ofertas de serviço

<b>Campo</b>	<b>Tipo</b>	<b>Descrição</b>
product_id	Inteiro (PK)	Identificador único
product_name	String	Nome exibido
product_slug	String	Nome amigável para URL
category	Enum	'standalone', 'bundle', 'addon', 'promo'

<b>Campo</b>	<b>Tipo</b>	<b>Descrição</b>
provisioning_play	String	Nome do playbook do Ansible
provisioning_json_vars	JSON	Variáveis do playbook
inventory_items_list	String	Inventário necessário
retail_cost	Float	Preço mensal
retail_setup_cost	Float	Taxa única
enabled	Boolean	Disponível para venda

## **Modelos de Faturamento**

### **Transação** - Cobranças/créditos individuais

<b>Campo</b>	<b>Tipo</b>	<b>Descrição</b>
transaction_id	Inteiro (PK)	Identificador único
customer_id	Inteiro (FK)	Cliente pai
invoice_id	Inteiro (FK)	Fatura pai (opcional)
service_id	Inteiro (FK)	Serviço relacionado
title	String	Descrição da transação
retail_cost	Float	Montante
tax_percentage	Float	Taxa de imposto
tax_amount	Float	Imposto calculado
void	Boolean	Transação cancelada

### **Fatura** - Transações agrupadas para faturamento

<b>Campo</b>	<b>Tipo</b>	<b>Descrição</b>
invoice_id	Inteiro (PK)	Identificador único
customer_id	Inteiro (FK)	Cliente pai
paid	Boolean	Pagamento recebido
void	Boolean	Fatura cancelada
payment_reference	String	ID da transação Stripe
start_date, end_date	Date	Período de faturamento
due_date	Date	Prazo de pagamento
retail_cost	Float	Montante total



## **Modelos de Inventário**

### **Inventário** - Ativos físicos e virtuais

<b>Campo</b>	<b>Tipo</b>	<b>Descrição</b>
inventory_id	Inteiro (PK)	Identificador único
customer_id	Inteiro (FK)	Cliente atribuído
service_id	Inteiro (FK)	Serviço vinculado
inventory_template_id	Inteiro (FK)	Modelo de tipo de item

<b>Campo</b>	<b>Tipo</b>	<b>Descrição</b>
item	String	Tipo de item (Cartão SIM, Roteador, etc.)
item_state	Enum	'Novo', 'Atribuído', 'Usado', etc.
itemtext1-20	String	Campos flexíveis
management_url	String	URL de administração do equipamento
config_content	Text	Arquivo de configuração

**Inventory\_Template** - Define a estrutura do item de inventário

<b>Campo</b>	<b>Tipo</b>	<b>Descrição</b>
inventory_template_id	Inteiro (PK)	Identificador   nico
item	String	Nome do modelo
itemtext1_label	String	Rótulo para o campo itemtext1
itemtext2_label	String	Rótulo para o campo itemtext2

## Modelos de Provisionamento

**Provision** - Rastreamento de trabalhos de provisionamento

<b>Campo</b>	<b>Tipo</b>	<b>Descrição</b>
provision_id	Inteiro (PK)	Identificador único
product_id	Inteiro (FK)	Produto sendo provisionado
customer_id	Inteiro (FK)	Cliente alvo
service_id	Inteiro (FK)	Serviço criado/modificado
provisioning_play	String	Nome do playbook do Ansible
provisioning_status	Inteiro	0=Sucesso, 1=Executando, 2=Falhou

**Provision\_Event** - Etapas individuais de provisionamento

<b>Campo</b>	<b>Tipo</b>	<b>Descrição</b>
provision_event_id	Inteiro (PK)	Identificador único
provision_id	Inteiro (FK)	Trabalho de provisionamento pai
event_name	String	Nome da tarefa
event_number	Inteiro	Número da sequência
provisioning_status	Inteiro	0=Sucesso, 1=Executando, 2=Falhou
provisioning_result_json	JSON	Saída completa da tarefa

## Modelos de Usuário & Segurança

**Usuário** - Contas de usuário

<b>Campo</b>	<b>Tipo</b>	<b>Descrição</b>
id	Inteiro (PK)	Identificador único
username	String	Nome de usuário para login

<b>Campo</b>	<b>Tipo</b>	<b>Descrição</b>
email	String	Endereço de email
email_verified	Boolean	Email confirmado
is_2fa_enabled	Boolean	Autenticação de dois fatores habilitada
totp_secret	String	Chave secreta TOTP

### **Função** - Funções de usuário

<b>Campo</b>	<b>Tipo</b>	<b>Descrição</b>
id	Inteiro (PK)	Identificador único
name	String	Nome da função
description	String	Descrição da função

### **Permissão** - Permissões granulares

<b>Campo</b>	<b>Tipo</b>	<b>Descrição</b>
id	Inteiro (PK)	Identificador único
name	String	Nome da permissão (recurso.ação)
description	String	Descrição da permissão

### **Relacionamentos:**

- Usuários têm muitas Funções (muitos-para-muitos)
- Funções têm muitas Permissões (muitos-para-muitos)
- Usuários podem vincular a um Contato (para acesso ao portal do cliente)

## **Pontos de Integração**

### **Gateway de Pagamento Stripe**

- Métodos de pagamento tokenizados
- Armazenamento de cartões em conformidade com PCI
- Pagamento automático de faturas
- Processamento de reembolsos
- Notificações de cartões expirando

### **Serviço de Email Mailjet**

- Emails transacionais (faturas, boas-vindas, notificações)
- Sincronização de contatos
- Gerenciamento de templates
- Rastreamento de entrega

### **Google Maps**

- Autocompletar endereço

- Geocodificação e geocodificação reversa
- Geração de Código Plus
- Mapeamento de localização do site

## **OCS/CGRateS**

- Provisionamento de contas
- Classificação em tempo real
- Gerenciamento de saldo
- Processamento de CDR
- Planos de ação e cronogramas

## **Arquitetura de Segurança**

### **Recursos de Segurança:**

- Autenticação baseada em JWT
- Autenticação de dois fatores (TOTP)
- Controle de acesso baseado em funções (RBAC)
- Acesso a recursos baseado em permissões
- Gerenciamento de sessão com Redis
- Hashing de senhas (bcrypt)
- Verificação de email
- Registro de auditoria via Registro de Atividades

## **Arquitetura de Implantação**

### **Configuração de Produção Recomendada:**

## **Pilha de Tecnologia**

### **Backend:**

- Python 3.x
- Framework web Flask
- SQLAlchemy ORM
- Migrações Alembic
- Ansible para automação

### **Frontend:**

- React
- Gerenciamento de estado
- React Router
- Axios para chamadas de API

### **Banco de Dados:**

- Banco de dados relacional (armazenamento de dados primário)
- Redis (cache e sessões)

### **Serviços Externos:**

- CGRateS (cobrança/classificação)
- Stripe (pagamentos)
- Mailjet (email)
- Google Maps (geocodificação)

## **Considerações sobre Escalabilidade**

### **Escalonamento Horizontal:**

- Design de API sem estado permite múltiplas instâncias
- Balanceador de carga distribui requisições
- Redis para estado de sessão compartilhada

### **Escalonamento de Banco de Dados:**

- Réplicas de leitura para consultas de relatórios
- Pooling de conexões
- Otimização de consultas e indexação

### **Provisionamento em Escala:**

- Processamento de trabalhos em segundo plano
- Múltiplos trabalhadores Ansible
- Gerenciamento de fila de trabalhos
- Lógica de repetição para provisionamentos falhados

## **Monitoramento & Observabilidade**

### **Métricas Chave:**

- Tempos de resposta da API
- Taxas de sucesso/falha de provisionamento
- Uso do pool de conexões do banco de dados
- Taxas de acerto de cache
- Sessões ativas

### **Registro:**

- Logs de aplicação (Flask)
- Logs de provisionamento (saída do Ansible)

- Registro de atividades (trilha de auditoria)
- Rastreamento de erros e alertas

## **Documentação Relacionada**

- [Playbooks do Ansible](#) - Automação de provisionamento
- [Sistema de Provisionamento](#) - Detalhes do fluxo de trabalho
- [Produtos e Serviços](#) - Arquitetura do produto
- [Documentação da API](#) - Referência da API
- [RBAC](#) - Segurança e permissões



# Fluxos de Autenticação e Controles de Administração

OmniCRM oferece recursos abrangentes de autenticação, incluindo login, autenticação de dois fatores (2FA), gerenciamento de senhas e controles administrativos para gerenciar a segurança do usuário. Este guia foca nos fluxos de trabalho da interface do usuário tanto para usuários finais quanto para administradores.

Veja também: Self-Care Portal <self\_care\_portal> para login de clientes e acesso ao portal, RBAC <rbac> para permissões de funcionários.

## Visão Geral

A autenticação do OmniCRM inclui:

- **Login com Email/Senha** - Autenticação padrão baseada em credenciais
- **Autenticação de Dois Fatores (2FA)** - Segundo fator baseado em TOTP opcional
- **Lembrar-me** - Sessão estendida de até 30 dias
- **Redefinição de Senha** - Recuperação de senha self-service via email
- **Controles Administrativos** - Ferramentas administrativas para redefinir 2FA e senhas
- **Logins Sociais** - Integração opcional com Google, Apple, Facebook (se habilitado)
- **Navegação Baseada em Funções** - Roteamento automático com base nas funções dos usuários

## Fluxo de Login

A página de login é o ponto de entrada para todos os usuários (funcionários e clientes).

### Login Padrão

Tela de Login

#### Processo de Login:

1. Insira **endereço de email** (email do funcionário ou cliente)
2. Insira **senha**
3. Opcional: Marque "**Lembrar-me por 30 dias**" para sessão estendida
4. Clique em "**Login**"

#### O que acontece a seguir:

- **Sem 2FA:** Usuário logado imediatamente, navegado com base na função:
  - **Clientes** → portal Self-Care (/self-care)
  - **Funcionários/Admins** → painel de clientes (/customers)
  - **Modo CBC** → interface de Cell Broadcast (/create-cell-broadcast)
- **Com 2FA Habilitado:** Redirecionado para a tela de verificação de 2FA

#### Recurso Lembrar-me:

Quando habilitado, a sessão persiste por **30 dias** em vez de expirar quando o navegador é fechado. Usa cookies seguros apenas HTTP.



### **Mostrar/Ocultar Senha:**

Clique no **ícone de olho (👁)** para alternar a visibilidade da senha.

### **Login com 2FA**

Se o usuário tiver 2FA habilitado, após inserir email/senha, a tela de desafio 2FA aparece:

Tela de Autenticação de Dois Fatores

#### **Usando Código do Autenticador:**

1. Abra o aplicativo autenticador (Google Authenticator, Authy, etc.)
2. Encontre a entrada do OmniCRM
3. Insira o código de 6 dígitos
4. O código é enviado automaticamente quando todos os 6 dígitos são inseridos
5. Se válido, o usuário é logado e navegado para o painel apropriado

#### **Usando Código de Recuperação:**

Se o aplicativo autenticador não estiver disponível:

1. Clique na aba **"Código de Recuperação"**
2. Insira um dos seus códigos de backup salvos (por exemplo, 3fa5b9c2)
3. Clique em **"Verificar"**
4. O código é consumido (pode ser usado apenas uma vez)

#### **Cancelar:**

Clique em **"Cancelar"** para retornar à página de login.

### **Logins Sociais (Opcional)**

Se habilitado (REACT\_APP\_ALLOW\_SOCIAL\_LOGINS=yes), botões de login social aparecem:

[🔗 Entrar com Google] [🍏 Entrar com Apple] [🔗 Entrar com Facebook]

Clique em qualquer botão para autenticar via esse provedor. Atualmente exibe uma mensagem de "em breve" (implementação de login social em andamento).

### **Link de Esqueci a Senha**

Clique no link **"Esqueceu a senha?"** para iniciar o fluxo de redefinição de senha.

## **Configuração da Autenticação de Dois Fatores (2FA)**

Os usuários podem habilitar 2FA para maior segurança da conta. O 2FA usa TOTP (Senha de Uso Único Baseada em Tempo) compatível com aplicativos autenticadores padrão.

### **Acessando a Configuração de 2FA**

A partir do perfil ou configurações do usuário:

#### **Nota para Clientes:**

Usuários com função de cliente não veem os prompts de 2FA. O 2FA é tipicamente exigido apenas para funcionários e usuários administrativos.

## Passo 1: Confirmar Senha

Senha Atual

[Cancelar] [Continuar]

Insira sua senha atual para prosseguir. Isso verifica sua identidade antes de habilitar o 2FA.

## Passo 2: Escanear Código QR

[Cancelar] [Confirmar]

### Instruções de Configuração:

1. **Baixar Aplicativo Autenticador** (se você não tiver um):
  - iOS: Apple App Store → "Google Authenticator"
  - Android: Google Play → "Google Authenticator"
  - Alternativas: Authy, Microsoft Authenticator, 1Password
2. **Escanear Código QR:**
  - Abra o aplicativo autenticador
  - Toque em "+" ou "Adicionar conta"
  - Escolha "Escanear código QR"
  - Aponte a câmera para o código QR na tela
  - O aplicativo adiciona a entrada "OmniCRM" com o código de 6 dígitos
3. **Salvar Códigos de Backup:**
  - **CRÍTICO:** Anote ou copie esses 8 códigos
  - Armazene em local seguro (gerenciador de senhas, cofre, etc.)
  - Cada código é de uso único
  - Usado se você perder o acesso ao aplicativo autenticador
  - Clique em "**Copiar Códigos**" para copiar todos os códigos para a área de transferência
4. **Verificar Configuração:**
  - Insira o código atual de 6 dígitos do aplicativo autenticador
  - Clique em "**Confirmar**"
  - Se válido, o 2FA agora está habilitado

## Passo 3: 2FA Habilitado

Uma mensagem de sucesso aparece:

A partir de agora, o login requer tanto a senha quanto o código 2FA.

Interface de Configuração de 2FA

## Fluxo de Redefinição de Senha (Self-Service)

Usuários que esquecem sua senha podem redefini-la via email.

### Passo 1: Solicitar Link de Redefinição

A partir da página de login, clique em "**Esqueceu a senha?**"

Solicitação de Redefinição de Senha

1. Insira **endereço de email**
2. Clique em "**Enviar Link de Redefinição**"

### O que acontece:

- O sistema verifica se o email existe no banco de dados
- Se encontrado, envia um email de redefinição de senha via Mailjet
- O email contém um link de redefinição com tempo limitado (tipicamente 1 hora de expiração)
- Mensagem de sucesso aparece: "Instruções de redefinição foram enviadas para seu email"

### Se o Email Não For Encontrado:

Mensagem de erro: "Nenhuma conta encontrada com esse endereço de email"

### Passo 2: Verificar Email

O usuário recebe um email com um assunto como:

Olá [Nome],

Você solicitou uma redefinição de senha para sua conta do OmniCRM.

Clique no link abaixo para redefinir sua senha: <<https://yourcompany.com/reset-password/abc123token456>>

Este link expira em 1 hora.

Se você não solicitou isso, ignore este email.

Clique no link de redefinição para prosseguir.

### Passo 3: Definir Nova Senha

O link de redefinição abre a página de criação de senha:

🔒 (ícone de cadeado)

Senha

Confirmar Senha

[Redefinir Senha]

1. Insira **nova senha**
2. Reinsira no campo **Confirmar Senha**
3. Clique em "**Redefinir Senha**"

### Requisitos de Senha:

- Comprimento mínimo (tipicamente 8+ caracteres)
- As senhas devem corresponder

### Sucesso:

- Mensagem de sucesso: "A senha foi redefinida com sucesso"
- Redirecionado automaticamente para a página de login
- O usuário pode agora fazer login com a nova senha

### Token Expirado/Inválido:

Se o link de redefinição estiver expirado ou inválido:

[Solicitar Novo Link de Redefinição]

## Controles Administrativos para Gerenciamento de Usuários

Administradores com permissões apropriadas podem gerenciar as configurações de autenticação do usuário a partir da interface de Gerenciamento de Usuários.

### Acessando o Gerenciamento de Usuários

Exibe uma tabela de todos os usuários com botões de ação.

Nome	Email	Telefone	Ações
John Smith	< <a href="mailto:john@example.com">john@example.com</a> >	+44 123...	
Jane Doe	< <a href="mailto:jane@example.com">jane@example.com</a> >	+44 456...	
Bob Wilson	< <a href="mailto:bob@example.com">bob@example.com</a> >	+44 789...	

#### Ícones de Ação:

- **Editar** - Modificar detalhes do usuário, funções, permissões
- **Excluir** - Remover conta de usuário
- **Redefinir Senha** - Gerar senha temporária
- **Redefinir 2FA** - Desabilitar 2FA para o usuário (apenas exibido se 2FA habilitado)
- **Enviar Email de Boas-Vindas** - Reenviar email de boas-vindas (apenas exibido se o usuário nunca fez login)

### Admin: Redefinir Senha do Usuário

Quando o usuário esquece a senha e o administrador precisa ajudar:

#### Passo 1: Clique no Ícone de Redefinir Senha ()

A modal de confirmação aparece:

Você tem certeza de que deseja redefinir a senha de:

Usuário: John Smith (<[john@example.com](mailto:john@example.com)>)

Uma senha temporária será gerada e exibida. O usuário deve mudar essa senha no próximo login.

[Cancelar] [Redefinir Senha]

#### Passo 2: Confirmar Redefinição

Clique em "**Redefinir Senha**". O sistema gera uma senha temporária segura.

#### Passo 3: Senha Temporária Exibida

Senha temporária para John Smith:

[ Copiar Senha]

⚠ **IMPORTANTE:** • Envie esta senha para o usuário via canal seguro • Não envie por email ou mensagem não segura • O usuário será forçado a mudar a senha no próximo login

[Fechar]

### **Ação do Administrador:**

- Copiar a senha temporária
- Ligar para o usuário ou comunicar via método seguro
- Fornecer a senha temporária verbalmente
- Instruir o usuário a fazer login e mudar a senha

### **Experiência do Usuário:**

Quando o usuário faz login com a senha temporária:

1. O login é bem-sucedido
2. Redirecionado imediatamente para a tela "Mudar Senha"
3. Deve definir uma nova senha antes de acessar o sistema
4. Não pode pular a mudança de senha

### **Admin: Redefinir 2FA do Usuário**

Quando o usuário perde o acesso ao aplicativo autenticador e todos os códigos de backup:

#### **Passo 1: Clique no Ícone de Redefinir 2FA (🔑)**

Aparece apenas para usuários com 2FA atualmente habilitado.

A modal de confirmação aparece:

Confirmação de Redefinição de 2FA do Usuário Admin

#### **Passo 2: Confirmar Redefinição**

Clique em "**Redefinir 2FA**"

#### **Passo 3: Confirmação**

Mensagem de sucesso:

John Smith agora pode fazer login apenas com sua senha. Ele pode reabilitar o 2FA nas configurações do usuário.

### **Experiência do Usuário:**

- O usuário agora pode fazer login apenas com a senha (sem código 2FA necessário)
- O ícone de escudo 2FA (🔑) desaparece da linha do usuário na tabela do administrador
- O usuário pode reabilitar voluntariamente o 2FA em suas configurações

### **Nota de Segurança Importante:**

Antes de redefinir o 2FA, o administrador deve:

1. Verificar a identidade do usuário por meios alternativos:
  - Verificação de ID do governo
  - Perguntas de segurança
  - Verificação de transação recente
  - Verificação presencial (se aplicável)
2. Documentar a redefinição nas notas do cliente
3. Informar o usuário para reabilitar o 2FA após recuperar o acesso

## Admin: Enviar Email de Boas-Vindas

Para usuários que não receberam ou perderam seu email de boas-vindas:

### Quando Disponível:

O ícone de avião de papel (✉) aparece apenas para usuários que **nunca fizeram login** (login\_count = 0).

### Clique no Ícone de Enviar Email de Boas-Vindas (✉)

Enviar email de boas-vindas para:

Usuário: Bob Wilson (<[bob@example.com](mailto:bob@example.com)>)

O email incluirá: • Mensagem de boas-vindas • Instruções de login • Link para definir a senha inicial (se aplicável) • Informações de contato de suporte

[Cancelar] [Enviar Email]

Clique em "**Enviar Email**"

Mensagem de sucesso:

### Email Enviado via Mailjet:

Usa o template: api\_crmCommunicationUserWelcome

## Admin: Editar Usuário

Clique no **ícone Editar** (✎) para modificar os detalhes do usuário:

Primeiro Nome

Último Nome

Email

Número de Telefone

Funções ☒ admin ☐ customer\_service\_agent\_1 ☐ customer

[Cancelar] [Salvar Alterações]

### Campos Editáveis:

- Nome, email, telefone
- **Funções** - Atribuir/remover funções (afeta permissões)
- Status ativo/inativo

## Admin: Excluir Usuário

Clique no **ícone Excluir** (✖) para remover o usuário:

Você tem certeza de que deseja excluir:

Usuário: John Smith (<[john@example.com](mailto:john@example.com)>)

△ AVISO: Esta ação não pode ser desfeita.

Isso excluirá permanentemente: • Conta do usuário e credenciais • Configurações de 2FA  
• Histórico de sessão

Dados e transações do cliente NÃO serão excluídos.

[Cancelar] [Excluir Usuário]

Clique em "**Excluir Usuário**" para confirmar.

Mensagem de sucesso:

## Melhores Práticas

### Para Usuários Finais

#### Segurança do Login:

- Use senhas fortes e únicas
- Habilite "Lembrar-me" apenas em dispositivos pessoais
- Sempre faça logout em computadores compartilhados
- Habilite 2FA para segurança adicional

#### Gerenciamento de 2FA:

- Salve os códigos de backup imediatamente após habilitar o 2FA
- Armazene os códigos em gerenciador de senhas ou local seguro
- Teste um código de backup para garantir que funcionem
- Regerear códigos de backup se você usar vários
- Contate o administrador se você perder tanto o autenticador quanto os códigos de backup

#### Gerenciamento de Senhas:

- Use um gerenciador de senhas para gerar e armazenar senhas
- Nunca compartilhe senhas por email ou mensagem
- Mude a senha se suspeitar de comprometimento
- Use uma senha única para o OmniCRM (não reutilize senhas)

### Para Administradores

#### Gerenciamento de Segurança do Usuário:

- Verifique a identidade do usuário antes de redefinir 2FA ou senhas
- Nunca envie senhas temporárias por email
- Documente todas as redefinições de segurança nas notas do usuário
- Incentive os funcionários a habilitar 2FA
- Monitore padrões de login incomuns

#### Redefinições de Senha:

- Comunique senhas temporárias apenas por telefone ou pessoalmente
- Gere senhas temporárias fortes (o sistema faz isso automaticamente)
- Certifique-se de que o usuário mude a senha no primeiro login
- Não redefina senhas desnecessariamente - use o fluxo de redefinição por email quando possível

## **Redefinições de 2FA:**

- Trate as redefinições de 2FA como ações de alta segurança
- Verifique a identidade por meio de múltiplos canais antes de redefinir
- Documente o motivo da redefinição
- Incentive o usuário a reabilitar 2FA imediatamente após recuperar o acesso
- Considere exigir 2FA para todos os usuários administrativos

## **Gerenciamento de Usuários:**

- Revise regularmente a lista de usuários em busca de contas inativas
- Remova usuários que deixaram a organização
- Garanta atribuições de função apropriadas
- Monitore usuários que nunca fizeram login
- Audite permissões de usuários trimestralmente

# **Solução de Problemas**

## **Erro "Email ou senha inválidos"**

- **Causa:** Credenciais incorretas
- **Solução:**
  - Verifique se o endereço de email está correto
  - Verifique se o caps lock está desligado
  - Tente redefinir a senha se esquecida
  - Contate o administrador se a conta estiver bloqueada

## **Código 2FA não aceito**

- **Causa:** Problema de sincronização de tempo ou código incorreto
- **Solução:**
  - Certifique-se de que a hora do dispositivo está correta (Configurações → Data e Hora → Automático)
  - Aguarde o código atualizar (os códigos mudam a cada 30 segundos)
  - Tente o próximo código que aparecer
  - Use o código de backup se o autenticador não estiver funcionando
  - Contate o administrador para redefinir 2FA se tudo mais falhar

## **"Lembrar-me" não funcionando**

- **Causa:** Cookies desativados ou limpos
- **Solução:**
  - Ative cookies nas configurações do navegador
  - Não limpe cookies ao fechar o navegador
  - Desative extensões de privacidade para o domínio do OmniCRM
  - Tente um navegador diferente

## **Email de redefinição de senha não recebido**

- **Causa:** Email não enviado, filtro de spam ou email errado
- **Solução:**
  - Verifique a pasta de spam/lixo eletrônico
  - Verifique se o endereço de email está correto
  - Aguarde 5-10 minutos (a entrega de email pode ser atrasada)
  - Verifique se a integração com Mailjet está funcionando (administrador)
  - Contate o administrador para redefinição manual de senha



## Link de redefinição de senha expirado

- **Causa:** Token expirado (tipicamente 1 hora)
- **Solução:**
  - Solicite nova redefinição de senha
  - Verifique o email e clique no link prontamente
  - Contate o administrador se houver problemas repetidos

## Não é possível habilitar 2FA (senha incorreta)

- **Causa:** Senha atual inserida incorretamente
- **Solução:**
  - Verifique a senha atual
  - Redefina a senha primeiro se estiver incerto
  - Contate o administrador para assistência

## Perdeu o aplicativo autenticador e os códigos de backup

- **Causa:** Telefone perdido/resetado, códigos de backup não salvos
- **Solução:**
  - Contate o administrador imediatamente
  - O administrador verificará a identidade e redefinirá o 2FA
  - Faça login apenas com a senha após a redefinição
  - Reabilite o 2FA e SALVE os códigos de backup desta vez

## Admin: "Falha ao redefinir 2FA" erro

- **Causa:** Permissões insuficientes
- **Solução:**
  - Certifique-se de que você tem a função de administrador
  - Verifique as permissões da API
  - Contate o administrador do sistema

## Admin: Senha temporária não gerada

- **Causa:** Erro de API ou problema de permissões
- **Solução:**
  - Atualize a página e tente novamente
  - Verifique as permissões do administrador
  - Verifique os logs da API em busca de erros
  - Certifique-se de que o banco de dados está acessível

# Considerações de Segurança

## Gerenciamento de Sessão:

- Sessões expiram após um período de inatividade
- "Lembrar-me" estende a sessão para 30 dias
- Sessões armazenadas como cookies apenas HTTP (não acessíveis ao JavaScript)
- A flag segura garante que os cookies sejam enviados apenas por HTTPS

## Segurança da Senha:

- Senhas criptografadas usando algoritmos padrão da indústria
- Senhas em texto simples nunca são armazenadas
- Senhas temporárias expiram automaticamente após o primeiro uso
- Tentativas de login falhadas são rastreadas (potencial limitação de taxa)

## **Segurança do 2FA:**

- Segredos TOTP criptografados no banco de dados
- Códigos QR gerados do lado do cliente quando possível
- Códigos de backup criptografados antes do armazenamento
- Cada código de backup é de uso único apenas

## **Ações do Administrador:**

- Redefinições de 2FA registradas no log de atividades
- Redefinições de senha criam trilha de auditoria
- Ações administrativas requerem permissões de função apropriadas
- Endereços IP registrados para eventos de segurança

## **Documentação Relacionada**

- 2fa - Referência detalhada da API de 2FA (focada em API)
- rbac - Controle de acesso baseado em função e permissões
- administration\_configuration - Configuração de email Mailjet para redefinição de senha
- integrations\_mailjet - Configuração de template de email
- customer\_care - Portal Self-Care para clientes

Verificação de Login 2FA



# Criar um Cliente

0fT52ZvoZBE



# Clientes, Contatos, Sites e Serviços

Temos um modelo simples de um **Cliente** que pode ter múltiplos **Contatos** e múltiplos **Sites**, **Serviços**, etc.

Um **Cliente** é uma empresa ou indivíduo que tem um relacionamento conosco, para quem enviamos uma fatura / conta.

Um **Contato** é uma pessoa que trabalha com o cliente, para um indivíduo, provavelmente é a mesma pessoa que o próprio cliente, uma única pessoa, mas podemos ter membros da família ou outros contatos, e cada contato tem um tipo, por exemplo, um contato de cobrança, um contato técnico, etc., que influencia como lidamos com o contato.

Um **Site** é uma localização física onde entregamos serviços, pode ser uma casa, escritório ou outra localização. Isso nos permite ter múltiplos sites para um único cliente, por exemplo, um cliente com múltiplos escritórios, e saber quais serviços estão associados a qual site.

Um **Serviço** é algo pelo qual cobramos um cliente, pode ser um serviço de internet residencial, serviço móvel ou até mesmo serviços abstratos como locação de uma sub-rede ou fornecimento de eletricidade medida para um rack. Cada serviço está vinculado a um cliente e a um site, e pode ter múltiplas cobranças associadas a ele.

Os clientes também têm um Registro de Atividades <csa\_activity\_log>, que é um registro de todas as alterações feitas, Tags <administration\_tags>, Atributos <administration\_attributes> para armazenar metadados personalizados, Itens de Inventário <administration\_inventory> e informações financeiras como Transações <payments\_transaction>, Faturas <payments\_invoices> e Métodos de Pagamento <basics\_payment>.

Uma vez que criamos um cliente, podemos então adicionar um serviço <csa\_add\_service> a esse cliente, que é a coisa pela qual cobramos.

Para informações sobre como criar um cliente, veja [Criar um Cliente](#).

## Visão Geral do Modelo de Dados

OmniCRM usa um modelo de dados relacional organizado em torno de clientes e seus serviços. O modelo é dividido em seções focadas abaixo.

### Relações Centrais do Cliente

O cliente é a entidade central, com contatos, sites e serviços relacionados.

#### Pontos Chave:

- Um cliente pode ter múltiplos contatos (cobrança, técnico, etc.)
- Um cliente pode ter múltiplos sites (filiais, localizações)
- Os serviços são entregues aos sites
- Os contatos podem ter acesso ao portal via contas de usuário vinculadas

### Dados de Cobrança e Financeiros

Transações e faturas rastreiam toda a atividade financeira.

### **Pontos Chave:**

- Transações podem ser independentes ou agrupadas em faturas
- Serviços geram automaticamente transações mensais
- Cartões Stripe são tokenizados e armazenados com segurança
- Faturas agrupam múltiplas transações para cobrança

## **Produto e Provisionamento**

Produtos definem ofertas de serviços; o provisionamento cria serviços reais.

### **Pontos Chave:**

- Produtos são modelos; serviços são instâncias ativas
- O provisionamento cria ou modifica serviços via playbooks do Ansible
- Cada trabalho de provisionamento tem múltiplos eventos para rastrear o progresso
- Um provisionamento de produto pode criar múltiplos serviços (pacotes)

## **Sistema de Inventário**

Rastreie ativos físicos e virtuais atribuídos a clientes.

### **Pontos Chave:**

- Modelos de inventário definem a estrutura (campos) para cada tipo de item
- Campos flexíveis itemtext1-20 se adaptam a diferentes tipos de inventário
- Itens podem ser atribuídos a clientes e vinculados a serviços
- Armazena configurações de equipamentos e credenciais de gerenciamento

## **Usuário e Segurança**

Contas de usuário com controle de acesso baseado em funções.

### **Pontos Chave:**

- Usuários podem ter múltiplas funções; funções têm múltiplas permissões
- A autenticação de dois fatores (2FA) é opcional por usuário
- Usuários da equipe são independentes; usuários clientes estão vinculados a contatos
- Permissões granulares controlam o acesso a recursos

## **Lista de Clientes**

A lista de clientes fornece uma tabela pesquisável e filtrável de todos os clientes no sistema.

Visualização da Lista de Clientes

### **Recursos:**

- **Pesquisa** - Filtrar clientes por nome ou ID
- **Ações em Lote** - Selecionar múltiplos clientes para operações em lote
- **Paginação** - Navegar por grandes listas de clientes
- **Ações Rápidas** - Visualizar ou excluir clientes diretamente da lista

## **Cliente - Detalhe**

O objeto cliente em si não contém muitas informações, é apenas um nome e uma referência aos

contatos e sites.

Sua implantação específica pode incluir campos adicionais ou personalizações, mas o objeto cliente básico é muito simples.

Na página de visão geral também há um gráfico mostrando a Receita Média por Usuário (ARPU) para o cliente, que é a receita total dividida pelo número de serviços, e uma comparação de como este cliente se compara à média de ARPU para todos os clientes no sistema.

As opções de Status do Cliente são adaptadas com base nas necessidades específicas do seu negócio, mas geralmente incluem opções como Ativo, Inativo, Pendente, etc., com diferentes regras em cada uma que controlam o comportamento do cliente no sistema nesse estado.

Excluir um cliente só pode ser feito se o cliente não tiver serviços ativos, faturas não pagas ou transações não faturadas. Se o cliente tiver algum desses, você precisará encerrar os serviços ativos e garantir que os pagamentos sejam feitos antes de poder excluir o cliente, o que, por sua vez, arquivará o cliente e todos os dados associados, que podem ser restaurados posteriormente, se necessário.

Cliente - Detalhe

## Site - Detalhe

Os sites são localizações físicas onde os serviços são entregues e podem ter múltiplos serviços associados a eles.

Eles são predominantemente usados para clientes empresariais, onde um único cliente pode ter múltiplos sites, por exemplo, um cliente com múltiplos escritórios.

Ter múltiplos sites nos permite rastrear quais serviços estão associados a qual site, por exemplo, se um cliente solicita um novo serviço para um novo escritório, precisamos garantir que entregamos o serviço correto para a localização correta. Isso nos permite rastrear serviços por site e faturá-los separadamente, se necessário.

## Integração com Google Maps e Geocodificação

Cada site está integrado com o **Google Maps** para garantir dados de endereço precisos e geolocalização. A interface automaticamente geocodifica endereços e gera dados de localização.

### Como Funciona a Geocodificação de Endereço:

Ao adicionar ou editar um site, a interface fornece dois métodos para definir a localização:

#### 1. Pesquisa de Endereço (Recomendado)

- Use a barra de pesquisa na parte superior do formulário
- Digite um endereço e o Google Maps Autocomplete sugere correspondências
- Selecione o endereço correto no dropdown
- O sistema automaticamente preenche:
  - **Nome do Site** - Nome do local do Google Maps
  - **Endereço Linha 1** - Número e nome da rua
  - **Endereço Linha 2** - Subpremissa (número da unidade/sala)
  - **Cidade** - Localidade
  - **Estado/Região** - Área administrativa
  - **Código Postal** - Código postal
  - **País** - Nome do país
  - **Latitude & Longitude** - Coordenadas precisas
  - **Plus Code** - Código de Localização Aberta de 11 caracteres (ex.,

8C3MFJV8+2F)

■ **Google Maps Place ID** - Identificador único do local

2. **Marcador de Mapa Arrastável (Manual)**

- Arraste o pino no mapa para a localização exata
- O sistema realiza **geocodificação reversa** para obter o endereço a partir das coordenadas
- Os mesmos campos são preenchidos automaticamente com base na localização do pino
- Útil para áreas rurais ou quando o endereço é impreciso

**Geração de Plus Code:**

Os Plus Codes são gerados automaticamente a partir da latitude/longitude usando a biblioteca Open Location Code. Um Plus Code é um código curto (11 caracteres) que representa uma localização precisa em qualquer lugar do mundo.

**Exemplo:**

Campos auto-preenchidos: ✓ Nome do Site: "123 Main Street" ✓ Endereço Linha 1: "123 Main Street" ✓ Cidade: "Londres" ✓ Estado: "Grande Londres" ✓ País: "Reino Unido" ✓ Código Postal: "SW1A 1AA" ✓ Latitude: 51.5074 ✓ Longitude: -0.1278 ✓ Plus Code: "9C3XGPHC+3Q" ✓ Google Place ID: "ChIJdd4hrwug2EcRmSrV3Vo6lII"

**Requisitos de Validação:**

Antes de salvar um site, o sistema valida:

- **Latitude & Longitude** devem ser definidos (seja via pesquisa ou arrastando o marcador)
- **País** deve ser preenchido (padrão para REACT\_APP\_DEFAULT\_COUNTRY se não definido)
- **Plus Code** deve ter 12 caracteres (11 + 1 para preenchimento)

Se a validação falhar, você verá um erro:

**Feedback Visual:**

A interface mostra feedback em tempo real:

ou

**Como os Dados de Localização são Usados**

Os dados de localização geocodificados (latitude, longitude, Plus Code) são usados em todo o OmniCRM para:

**1. Entrega e Instalação de Serviços**

- **Técnicos de Campo** - Acessar Plus Code para navegar até a localização exata do site
- **Agendamento de Instalação** - Atribuir técnicos com base na proximidade geográfica
- **Implantação de Equipamentos** - Garantir que o equipamento correto seja entregue no site correto

**2. Notificações de Interrupção**

- **Alertas Geofenced** - Se houver uma interrupção de rede em uma área específica, consultar sites por raio de lat/long
- **Comunicações Direcionadas** - Enviar notificações de interrupção apenas para clientes afetados via Mailjet <integrations\_mailjet>
- **Páginas de Status** - Exibir mapa de interrupção com sites afetados

### Exemplo:

Consulta: SELECT \* FROM Customer\_Site  
WHERE distance(latitude, longitude, 51.5074, -0.1278) < 5

Resultado: 47 sites afetados Ação: Enviar notificação de interrupção para 47 clientes

### 3. Relatórios e Análises

- **Receita Geográfica** - Receita por cidade, estado, região
- **Mapas de Densidade de Serviços** - Mapa de calor das localizações de serviços
- **Planejamento de Expansão** - Identificar áreas mal atendidas

### 4. Clientes Empresariais com Múltiplos Sites

- **Gerenciamento de Sites** - Rastrear quais serviços em quais localizações
- **Cobrança Separada** - Faturar por site, se necessário
- **Atribuição de Serviços** - Vincular serviços a sites específicos durante o provisionamento

### Sites Rurais e Remotos

Para clientes em áreas rurais onde endereços de rua podem não existir ou serem imprecisos:

1. **Usar Arraste no Mapa**
  - Aproxime-se da área aproximada
  - Arraste o pino para a propriedade/edifício exato
  - O sistema gera Plus Code para essa localização precisa
2. **Compartilhamento de Plus Code**
  - Compartilhe o Plus Code com o cliente (ex., 8C3MFJV8+2F)
  - O cliente pode inserir isso no Google Maps para ver a localização exata
  - Técnicos de campo usam o Plus Code para navegação
3. **Notas de Endereço**
  - Use o campo "Nota de Endereço" para direções adicionais
  - Exemplo: "Vire à esquerda no celeiro vermelho, 500m após a grade de gado"
  - Notas visíveis para as equipes de instalação

Site - Adicionar com Google Maps

Dica

Você pode arrastar o pino no mapa para a localização correta se o endereço não for preciso. O sistema fará a geocodificação reversa da localização e preencherá todos os campos de endereço automaticamente.

Site - Visualização de Detalhe

## Contato - Detalhe

Os Contatos são pessoas associadas ao cliente. Eles podem ser contatos de cobrança, contatos técnicos ou outros tipos, e cada contato tem um tipo que influencia como lidamos com o contato.

Lista de Contatos do Cliente

Podemos ter múltiplos contatos para um único cliente, por exemplo, um cliente com múltiplos contatos de cobrança, ou um cliente com múltiplos contatos técnicos.

Um bom exemplo seria uma empresa com um provedor de serviços gerenciados, que lida com a



parte técnica das coisas, e um contato de cobrança separado que lida com a parte financeira, ou uma família onde cada membro tem seu próprio contato, mas nem todos estão autorizados a fazer alterações.

Da mesma forma, podemos querer enviar notificações de interrupção apenas para o contato técnico, ou enviar faturas apenas para o contato de cobrança, e o tipo de contato nos permite controlar isso.

A lógica exata de como os tipos de contato são usados depende do seu negócio, mas a ideia básica é que cada contato tem um tipo que influencia como lidamos com eles, e que cada pessoa que está associada ao cliente é um contato.

Os contatos são sincronizados com a integração Mailjet <integrations\_mailjet>, permitindo-nos enviar campanhas de email direcionadas com base no tipo de contato, localização, gastos mensais ou serviços adquiridos, e gerenciar todos os modelos de email usados para comunicações transacionais.

Contato - Visão Geral Contato - Formulário de Edição Contato - Detalhe



# Navegação

OmniCRM foi projetado desde o início para ser responsivo e intuitivo.

Este guia ajudará você a navegar pelo sistema e encontrar as informações de que precisa.

Uma vez logado, a barra de navegação no lado esquerdo da tela mostrará as principais seções do sistema e as subseções dentro de cada uma.

## Navegação - Barra de Navegação

No canto superior direito da tela, você verá o menu do usuário, que permite sair, alterar sua senha ou acessar suas configurações de usuário.

## Navegação - Menu do Usuário

Você tem links para todos os webapps mais comumente usados de sua organização na barra de WebApp (isso pode ser adaptado às necessidades da sua organização).

## Navegação - WebApps

Quaisquer alertas que você tenha pendentes são visíveis no canto superior direito da tela, e você pode clicar no alerta para ver mais informações.

## Navegação - Alertas

Podemos mudar o idioma do sistema clicando no menu suspenso de idiomas no canto superior direito da tela.

## Navegação - Idioma

Se você é um príncipe das trevas, pode alternar para o modo escuro clicando no ícone da lua no canto superior direito da tela.

## Navegação - Modo Escuro



# Visão Geral da Aba de Faturamento

A aba de Faturamento fornece uma visão unificada de todas as informações financeiras de um cliente, combinando métodos de pagamento, transações e faturas em uma única interface para uma gestão de faturamento eficiente.

Documentação relacionada: Payment Methods <payment\_methods>, Transactions <payments\_transaction>, Invoices <payments\_invoices>, Payment Processing <payments\_process>.

## Acessando a Aba de Faturamento

### Visão por Cliente:

1. Navegue até **Customers** → **[Select Customer]**
2. Clique na aba **Billing**
3. Veja todas as três seções: Métodos de Pagamento, Transações e Faturas

### Visões em Todo o Sistema:

Os dados de faturamento em todo o sistema podem ser acessados separadamente:

- **Billing** → **Transactions** - Todas as transações de todos os clientes
- **Billing** → **Invoices** - Todas as faturas de todos os clientes

### Portal de Autoatendimento:

Clientes que acessam o Self-Care Portal <self\_care\_portal> veem a mesma estrutura da aba de Faturamento:

- Visualizar e gerenciar seus métodos de pagamento
- Visualizar histórico de transações
- Visualizar e pagar faturas online

## Estrutura da Aba de Faturamento

A aba de Faturamento está organizada em três seções principais, exibidas como cartões:

### Seção 1: Métodos de Pagamento

**Propósito:** Gerenciar como o cliente paga pelos serviços

#### Principais Recursos:

- Visualizar todos os cartões de crédito salvos
- Definir método de pagamento padrão
- Adicionar novos métodos de pagamento (via Stripe)
- Remover cartões expirados ou não utilizados

Faturamento do Cliente - Métodos de Pagamento  
Detalhe do Cartão de Métodos de Pagamento

**Documentação:** basics\_payment

## Seção 2: Transações

**Propósito:** Rastrear todas as cobranças e créditos do cliente

**Principais Recursos:**

- Visualizar estatísticas de transações (Total, Não Faturado, Este Mês, Mês Passado)
- Listar todas as transações com filtragem por status de anulação/faturamento
- Adicionar transações manuais (cobranças ou créditos)
- Anular transações incorretas
- Ver quais transações estão faturadas vs não faturadas

**Documentação:** payments\_transaction

## Seção 3: Faturas

**Propósito:** Agrupar transações em faturas para o cliente pagar

**Principais Recursos:**

- Visualizar estatísticas de faturas (Total, Não Pagas, Este Mês, Mês Passado)
- Listar todas as faturas com filtragem por status de pagas/anuladas
- Gerar novas faturas a partir de transações não faturadas
- Baixar PDFs de faturas
- Enviar faturas por e-mail para os clientes
- Pagar faturas online (métodos de pagamento Stripe ou manuais)

Faturamento do Cliente - Transações e Faturas

- Anular ou reembolsar faturas

**Documentação:** payments\_invoices

## Fluxo de Dados Entre Seções

Entender como os dados fluem entre as três seções é crucial para uma gestão de faturamento eficaz.

### Diagrama de Fluxo

#### Relação Transação → Fatura

##### 1. Criação de Transação:

Quando um serviço é provisionado ou uma cobrança manual é adicionada:

- Transação criada na seção **Transactions**
- Status da transação: **Não Faturado**
- Campo invoice\_id da transação é null

**Exemplo:**

##### 2. Geração de Fatura:

Quando a equipe gera uma fatura:

- Fatura criada na seção **Invoices**

- Todas as transações não faturadas dentro do intervalo de datas agrupadas em uma fatura
- Campo `invoice_id` da transação populado
- Status da transação muda para: **Faturado**

#### Exemplo:

### 3. Atualização de Estatísticas de Transação:

- Total de **Transações Não Faturadas** diminui
- Estatística de **Total de Faturas** aumenta
- Total de **Faturas Não Pagas** aumenta

## Relação Fatura → Pagamento

### 1. Pagamento da Fatura:

Quando o cliente paga a fatura:

- Pagamento processado usando **Método de Pagamento** salvo (cartão Stripe)
- Ou método de pagamento manual selecionado (dinheiro, POS, transferência bancária)
- Status da fatura muda para: **Paga**

### 2. Transação de Pagamento Criada:

Para pagamentos manuais (não Stripe):

- Transação negativa criada automaticamente
- Título da transação: "Pagamento pela Fatura #1234"
- Valor da transação: -\$45.00 (negativo, credita o cliente)
- Campo `invoice_id` da transação: Link para a fatura paga

#### Exemplo:

### 3. Atualização de Estatísticas:

- Total de **Faturas Não Pagas** diminui
- **Total de Faturas Este Mês** inalterado (fatura já existia)

## Relação Método de Pagamento → Fatura

### Fluxo de Pagamento Stripe:

1. Cliente adiciona cartão de crédito em **Payment Methods**
2. Cartão tokenizado via Stripe, armazenado com segurança
3. Ao pagar a fatura, cliente seleciona cartão salvo
4. Stripe cobra o cartão
5. Fatura marcada como paga
6. Campo `payment_reference` populado com o ID da intenção de pagamento do Stripe

### Fluxo de Pagamento Manual:

1. Cliente paga via dinheiro/POS/transferência bancária (nenhum método de pagamento necessário)
2. A equipe seleciona o método de pagamento no modal Pagar Fatura
3. A equipe insere o número de referência (opcional)
4. Transação negativa criada para o valor do pagamento
5. Fatura marcada como paga

# Fluxos de Trabalho de Faturamento Completo

Esses fluxos de trabalho demonstram como as três seções trabalham juntas para realizar tarefas comuns.

## Fluxo de Trabalho 1: Configuração de Novo Cliente e Primeira Fatura

**Objetivo:** Configurar faturamento para novo cliente e coletar o primeiro pagamento

- 1. Adicionar Método de Pagamento:**
  - Navegue até o cliente → aba **Billing**
  - Seção **Payment Methods** → Clique em **"Add Payment Method"**
  - Cliente adiciona cartão de crédito via Stripe
  - Cartão salvo como método de pagamento padrão
- 2. Verificar Transações:**
  - Seção **Transactions** mostra transações não faturadas:
    - Taxa de configuração do serviço: \$50.00
    - Serviço do primeiro mês: \$45.00
    - Total Não Faturado: \$95.00
- 3. Gerar Fatura:**
  - Seção **Invoices** → Clique em **"Generate Proforma Invoice"**
  - Defina o intervalo de datas para incluir configuração e primeiro mês
  - Clique em **"Generate Invoice"**
  - Fatura #INV-2025-001234 criada por \$95.00
- 4. Atualização de Transações:**
  - Ambas as transações agora mostram: Fatura #INV-2025-001234
  - Total de **Transações Não Faturadas** agora \$0.00
- 5. Enviar Fatura por E-mail:**
  - Clique no ícone de e-mail ao lado da fatura
  - Cliente recebe e-mail da fatura com PDF e link para pagamento
- 6. Cliente Paga Online:**
  - Cliente clica no link de pagamento no e-mail
  - Redirecionado para o portal de Autoatendimento
  - Clique no botão **"Pay Invoice"**
  - Selecione o método de pagamento padrão
  - Clique em **"Pay Invoice"**
  - Stripe cobra o cartão
- 7. Atualização da Fatura:**
  - Status da fatura muda para **"Paid"**
  - Total de **Faturas Não Pagas** diminui em \$95.00

**Resultado:** Cliente totalmente configurado com método de pagamento, primeira fatura paga.

## Fluxo de Trabalho 2: Faturamento Recorrente Mensal

**Objetivo:** Faturar todos os clientes pelo serviço mensal no final do mês

- 1. Serviços Auto-Cobrança:**
  - Chegada do final do mês (31 de janeiro)
  - O sistema de faturamento cria automaticamente transações para todos os serviços recorrentes
  - Seção **Transactions** mostra novas transações não faturadas
- 2. Revisar Transações Não Faturadas:**
  - Navegue até a seção **Transactions**
  - Filtrar: **Invoice Status: Not Invoiced**
  - Revisar lista de todas as transações prontas para faturamento
  - Verificar se os valores e descrições estão corretos

### 3. Gerar Faturas:

- Navegue até **Billing** → **Invoices** (em todo o sistema)
- Para cada cliente (ou use o processo em lote):
  - Clique em **"Generate Proforma Invoice"**
  - Selecione o cliente
  - Data de Início: 2025-01-01
  - Data de Fim: 2025-01-31
  - Data de Vencimento: 2025-02-15
  - Clique em **"Generate Invoice"**

### 4. Atualização de Transações:

- Todas as transações agora vinculadas às faturas
- Totais de **Transações Não Faturadas** redefinidos para \$0.00

### 5. Enviar Todas as Faturas por E-mail:

- Para cada fatura, clique no ícone de e-mail
- Todos os clientes recebem faturas mensais

### 6. Clientes Pagam:

- Clientes com métodos de pagamento salvos pagam online via Autoatendimento
- A equipe processa pagamentos em dinheiro/POS para clientes que pagam pessoalmente
- Total de **Faturas Não Pagas** diminui à medida que os pagamentos são recebidos

**Resultado:** Todos os clientes faturados por janeiro, faturas enviadas, pagamentos processados.

## Fluxo de Trabalho 3: Tratamento de Crédito por Problema de Serviço

**Objetivo:** Creditar o cliente por interrupção de serviço, aplicar à fatura não paga

### 1. Cliente Relata Problema:

- O serviço ficou fora do ar por 2 dias
- Cliente merece crédito de \$10

### 2. Adicionar Transação de Crédito:

- Navegue até o cliente → aba **Billing** → seção **Transactions**
- Clique em **" + Add Transaction "**
- Tipo de Transação: **Credit**
- Tipo de Crédito: **Cash Payment** (ou tipo apropriado)
- Título: "Crédito por Interrupção de Serviço"
- Descrição: "Compensação por interrupção de 2 dias 8-9 Jan"
- Custo de Varejo: 10.00
- Clique em **"Add Transaction"**

### 3. Transação Criada:

- Transação aparece na lista com valor: **-\$10.00**
- Status da transação: **Não Faturado**
- Total de **Transações Não Faturadas** agora inclui -\$10.00

### 4. Aplicar à Fatura:

- Se o cliente já tiver fatura não paga:
  - Fatura permanece não paga com o valor original
  - Crédito será aplicado à próxima geração de fatura
- Se gerando nova fatura:
  - Seção **Invoices** → Clique em **"Generate Proforma Invoice"**
  - Incluir intervalo de datas com a transação de crédito
  - Fatura gerada com crédito aplicado:

### 5. Cliente Paga:

- Cliente paga valor reduzido: \$35.00
- Fatura marcada como paga

**Resultado:** Cliente creditado pela interrupção, crédito aplicado à próxima fatura, pagamento menor coletado.

## Fluxo de Trabalho 4: Método de Pagamento Expirado - Atualizar e Tentar Novamente

**Objetivo:** O cartão do cliente expirou, causando falha no pagamento - atualizar cartão e tentar pagamento novamente

1. **Notificação de Falha de Pagamento:**
  - Cliente tenta pagar a fatura
  - Stripe retorna erro: "Cartão expirado"
  - Pagamento falha, fatura permanece não paga
2. **Atualizar Método de Pagamento:**
  - Cliente navega até a aba **Billing**
  - Seção **Payment Methods** → Clique em **"Add Payment Method"**
  - Insira os novos dados do cartão (data de expiração atualizada)
  - Novo cartão salvo
3. **Definir como Padrão:**
  - Cliente clica em **"Set as Default"** no novo cartão
  - Cartão antigo removido automaticamente (se desejado)
4. **Tentar Pagamento Novamente:**
  - Navegue até a seção **Invoices**
  - Localize a fatura não paga
  - Clique no ícone **"Pay"**
  - Modal de pagamento abre com novo cartão padrão pré-selecionado
  - Clique em **"Pay Invoice"**
  - Stripe cobra o novo cartão com sucesso
5. **Atualização da Fatura:**
  - Status da fatura muda para **"Paid"**
  - Campo `payment_reference` populado com o novo ID da intenção de pagamento do Stripe

**Resultado:** Cliente atualizou o método de pagamento, fatura paga com sucesso com o novo cartão.

## Fluxo de Trabalho 5: Anulando Fatura Incorreta e Re-Faturamento

**Objetivo:** Fatura gerada pela equipe com transações erradas - anular e regenerar corretamente

1. **Erro Descoberto:**
  - Fatura #INV-2025-001234 gerada com intervalo de datas errado
  - Incluiu transações do mês errado
  - Fatura está não paga
2. **Anular Fatura:**
  - Navegue até a aba **Billing** → seção **Invoices**
  - Localize a fatura incorreta
  - Clique no ícone de excluir (✖)
  - Confirme a anulação
  - Fatura anulada
3. **Transações Liberadas:**
  - Navegue até a seção **Transactions**
  - Todas as transações da fatura anulada agora mostram: **Não Faturado**
  - Total de **Transações Não Faturadas** aumenta
  - Transações disponíveis para nova fatura
4. **Gerar Fatura Correta:**
  - Seção **Invoices** → Clique em **"Generate Proforma Invoice"**
  - Defina o intervalo de datas correto
  - Aplique filtro se necessário (por exemplo, "Mobile" para fatura apenas móvel)
  - Clique em **"Generate Invoice"**



- Nova fatura criada com transações corretas
5. **Verificar e Enviar por E-mail:**
    - Revise os detalhes da nova fatura
    - Verifique se as transações corretas estão incluídas
    - Clique no ícone de e-mail para enviar ao cliente

**Resultado:** Fatura incorreta anulada, transações re-faturadas corretamente, cliente recebe fatura corrigida.

## Fluxo de Trabalho 6: Processando Pagamento em Dinheiro para Múltiplas Faturas

**Objetivo:** Cliente paga várias faturas não pagas com um único pagamento em dinheiro

1. **Cliente Chega com Dinheiro:**
  - Cliente traz \$300 em dinheiro para pagar faturas pendentes
  - Navegue até o cliente → aba **Billing**
2. **Revisar Faturas Não Pagas:**
  - Seção **Invoices** → Filtrar: **Paid: Not yet Paid**
  - Visualizar faturas não pagas:
3. **Pagar Primeira Fatura:**
  - Clique no ícone de pagamento na fatura #1234
  - Modal de pagamento abre
  - Selecione o método de pagamento "**Cash**"
  - Insira a referência: "Dinheiro pago 2025-02-10 - Recibo #001"
  - Clique em "**Pay Invoice**"
  - Fatura #1234 marcada como "**Paid**"
4. **Pagar Faturas Restantes:**
  - Repita o processo para a fatura #1235:
    - Referência: "Dinheiro pago 2025-02-10 - Recibo #001"
  - Repita para a fatura #1236:
    - Referência: "Dinheiro pago 2025-02-10 - Recibo #001"
5. **Verificar Transações:**
  - Navegue até a seção **Transactions**
  - Três novas transações de pagamento criadas:
  - Todas vinculadas às respectivas faturas
6. **Atualizar Estatísticas:**
  - Seção **Invoices** → Total de **Faturas Não Pagas** diminuído em \$300.00
  - Todas as faturas agora mostram status "**Paid**"

**Resultado:** Cliente pagou todas as faturas pendentes em dinheiro, transações de pagamento registradas com referência de recibo.

## Melhores Práticas

### Para Usuários da Equipe

#### Gerenciamento de Transações:

- Adicione transações manuais imediatamente (não atrase)
- Use títulos e descrições descritivas para clareza
- Vincule transações a serviços e locais quando aplicável
- Anule transações incorretas antes que sejam faturadas

#### Geração de Faturas:

- Gere faturas em intervalos consistentes (por exemplo, mensalmente no 1º do mês)

- Use intervalos de datas com cuidado para evitar sobreposições ou lacunas
- Use filtros para criar faturas específicas de serviços quando necessário
- Envie faturas por e-mail imediatamente após a geração
- Revise PDFs de faturas antes de enviar aos clientes

### **Processamento de Pagamentos:**

- Verifique se o método de pagamento é válido antes de tentar a cobrança
- Sempre insira números de referência para pagamentos manuais (dinheiro, POS, transferência bancária)
- Marque faturas como pagas imediatamente após receber o pagamento
- Reembolse apenas via Stripe (crie crédito manual para outros métodos de pagamento)

### **Higiene de Dados:**

- Revise regularmente transações não faturadas
- Investigue transações anuladas para entender erros de faturamento
- Monitore faturas não pagas e siga com os clientes
- Mantenha métodos de pagamento atualizados (remova cartões expirados)

## **Para Clientes (Portal de Autoatendimento)**

### **Métodos de Pagamento:**

- Mantenha pelo menos um método de pagamento válido em arquivo
- Atualize métodos de pagamento antes que os cartões expirem
- Defina seu cartão preferido como padrão

### **Pagamentos de Faturas:**

- Pague faturas antes da data de vencimento para evitar taxas de atraso
- Revise os detalhes da fatura e transações antes de pagar
- Baixe PDFs de faturas para seus registros
- Entre em contato com o suporte imediatamente se a fatura parecer incorreta

### **Revisão de Transações:**

- Revise regularmente o histórico de transações
- Relate quaisquer cobranças inesperadas imediatamente
- Entenda quais transações estão faturadas vs não faturadas

## **Para Administradores**

### **Configuração do Sistema:**

- Configure modelos de e-mail Mailjet para entrega profissional de faturas
- Personalize modelos de PDF de faturas para combinar com a marca
- Configure a integração Stripe para processamento seguro de pagamentos
- Configure termos de pagamento e datas de vencimento

### **Monitoramento e Relatórios:**

- Use widgets de estatísticas para monitorar a saúde do faturamento
- Acompanhe o total de **Transações Não Faturadas** - deve diminuir após o ciclo de faturamento
- Monitore o total de **Faturas Não Pagas** - siga os pagamentos em atraso
- Revise estatísticas de **Este Mês vs Mês Passado** para tendências

### Automação:

- Automatize cobranças de serviços recorrentes via configuração de produtos
- Configure geração automática de faturas para faturamento recorrente (se disponível)
- Configure lembretes por e-mail para faturas em atraso

## Problemas Comuns e Soluções

### Problema: Cliente Não Consegue Pagar a Fatura

#### Sintomas:

- Cliente clica no botão de pagamento, mas nada acontece
- Mensagem de erro: "Nenhum método de pagamento encontrado"

#### Diagnóstico:

1. Navegue até o cliente → aba **Billing** → seção **Payment Methods**
2. Verifique se o cliente tem algum método de pagamento salvo
3. Verifique se os cartões salvos estão expirados

#### Solução:

- O cliente deve adicionar um método de pagamento válido antes de pagar as faturas
- Oriente o cliente para a página **Payment Methods** para adicionar um cartão de crédito
- Verifique se o cartão é aceito (Visa, Mastercard, Amex, etc.)
- Tente o pagamento novamente após adicionar o cartão

### Problema: Fatura Gerada com Transações Erradas

#### Sintomas:

- Fatura inclui transações do período errado
- Fatura faltando transações esperadas
- Total da fatura incorreto

#### Diagnóstico:

1. Abra a fatura na seção **Invoices**
2. Revise as transações incluídas na fatura
3. Verifique as datas das transações vs intervalo de datas da fatura
4. Verifique se um filtro foi aplicado durante a geração

#### Solução:

- **Se a fatura estiver não paga:** Anule a fatura, verifique se as transações estão não faturadas, regenere com o intervalo de datas correto
- **Se a fatura estiver paga:** Não é possível anular - crie uma transação de crédito para o valor incorreto, gere uma fatura corrigida
- **Prevenção:** Sempre revise a seção **Transactions** antes de gerar a fatura para verificar se as transações corretas serão incluídas

### Problema: Total de Transações Não Faturadas Não Está Diminuindo

#### Sintomas:

- Widget de **Transações Não Faturadas** mostra um valor alto

- Lista de transações mostra muitas transações não faturadas
- Faturas mensais já foram geradas

### Diagnóstico:

1. Filtre transações por **Invoice Status: Not Invoiced**
2. Revise a lista de transações não faturadas
3. Verifique as datas das transações - podem ser cobranças recentes após a última geração de fatura
4. Verifique se algumas transações foram anuladas (não devem contar para o total não faturado)

### Solução:

- **Comportamento esperado:** Transações criadas após a última geração de fatura permanecem não faturadas até o próximo ciclo de faturamento
- **Se transações antigas estiverem não faturadas:** Gere fatura para essas transações com o intervalo de datas apropriado
- **Se transações anuladas contadas:** O sistema deve excluir automaticamente transações anuladas - relate um bug se não o fizer

## Problema: Método de Pagamento Continua Sendo Recusado

### Sintomas:

- Pagamento Stripe falha com "Cartão recusado"
- Cliente relata que o cartão deve funcionar
- Múltiplas tentativas de reembolso falham

### Diagnóstico:

1. Verifique o painel do Stripe para o motivo da recusa
2. Motivos comuns de recusa:
  - Fundos insuficientes
  - Cartão expirado
  - Cartão relatado como perdido/roubado
  - Prevenção de fraude bancária
  - Cartão internacional bloqueado

### Solução:

- Peça ao cliente para entrar em contato com o banco para autorizar o pagamento
- Tente um método de pagamento diferente (cartão diferente)
- Para a equipe: Processar como pagamento manual (dinheiro, POS) e marcar fatura como paga
- Verifique se o endereço de cobrança corresponde ao endereço de cobrança do cartão

## Atalhos de Teclado e Dicas

### Navegação:

- Acesse a aba de faturamento rapidamente: Página do cliente → tecla **B** (se os atalhos de teclado estiverem habilitados)

### Filtragem:

- **Transactions:** Clique no filtro "**Not Invoiced**" antes de gerar faturas para ver exatamente

o que será incluído

- **Invoices:** Clique no filtro "**Not yet Paid**" para ver todas as faturas pendentes que requerem acompanhamento

### **Operações em Lote:**

- Selecione várias faturas (caixa de seleção) para excluir/anular várias de uma vez (apenas para a equipe)
- Use a barra de pesquisa para encontrar rapidamente uma transação ou fatura específica pelo ID

### **Ações Rápidas:**

- Clique no título da transação/fatura para ver todos os detalhes no modal
- Menu de ações de clique direito ( : ) para acesso rápido a opções de download/enviar por e-mail/pagar

## **Documentação Relacionada**

- `basics_payment` - Métodos de pagamento e integração Stripe
- `payments_transaction` - Transações em detalhes
- `payments_invoices` - Faturas em detalhes
- `integrations_mailjet` - Entrega de faturas por e-mail
- `csa_activity_log` - Visualizando o histórico de atividades de faturamento



# Ansible Playbooks: Guia Detalhado

Os produtos OmniCRM são provisionados usando **Ansible**, permitindo a gestão automatizada de serviços com base nos requisitos específicos de cada produto e seu inventário associado.

## Como Playbooks e Produtos Trabalham Juntos

**Conceito Crítico:** Playbooks são o que realmente cria serviços em OmniCRM. Quando você atribui um playbook a um produto, está definindo **o que acontece** quando esse produto é provisionado - mas isso pode significar coisas diferentes para produtos diferentes.

### Produtos Disparam Playbooks

Quando um produto é provisionado no OmniCRM:

1. A definição do produto especifica qual playbook executar (via campo `provisioning_play`)
2. O produto passa variáveis para o playbook (via `provisioning_json_vars` e seleções de inventário)
3. O playbook é executado e faz o que foi programado para fazer
4. O playbook determina o que será criado (se houver algo)

### O Que os Playbooks Podem Fazer

Um único playbook de provisionamento pode:

#### Criar Múltiplos Serviços

Um playbook de produto agrupado pode criar:

- Um registro de serviço de internet principal
- Um registro de serviço de addon IPTV
- Um registro de serviço VoIP
- Tudo com uma única ação de provisionamento de produto

#### Criar Nenhum Serviço

Alguns playbooks não criam registros de serviço:

- Um playbook que apenas configura equipamentos CPE
- Um playbook que envia configuração para equipamentos de rede
- Um playbook que atualiza sistemas externos

#### Criar Um Serviço

O padrão mais comum:

- Criar um único registro de serviço para o cliente
- Vincular inventário a esse serviço
- Configurar faturamento para esse serviço

### **Modificar Serviços Existentes**

Playbooks de recarga e addon:

- Não criam novos serviços
- Atualizam registros de serviços existentes (adicionam dados, estendem a expiração, etc.)
- Adicionam saldos a contas de faturamento existentes

### **Executar Ações Sem Registros de Serviço**

Alguns playbooks são puramente operacionais:

- Redefinir saldos de contas
- Trocar itens de inventário entre clientes
- Gerar relatórios ou configurações

### **Exemplo: Comportamentos Diferentes de Playbook**

```
# Produto 1: Serviço de SIM Móvel (cria 1 serviço)
# provisioning_play: play_simple_service
- Cria registro de serviço no CRM
- Cria conta de faturamento no OCS
- Atribui inventário de cartão SIM e número de telefone
- Envia e-mail de boas-vindas

# Produto 2: Pacote de Internet (cria 3 serviços)
# provisioning_play: play_bundle_internet_tv_voice
- Cria registro de serviço de internet
- Cria registro de serviço de IPTV
- Cria registro de serviço VoIP
- Vincula todos ao mesmo cliente
- Conta de faturamento única para o pacote

# Produto 3: Recarga de Dados (cria 0 serviços)
# provisioning_play: play_topup_no_charge
- Encontra serviço existente pelo service_id
- Adiciona saldo de dados à conta OCS existente
- Atualiza data de expiração do serviço
- NENHUM novo serviço criado

# Produto 4: Configuração de CPE (cria 0 serviços)
# provisioning_play: play_prov_cpe_mikrotik
- Gera configuração do roteador
```

- Atualiza registro de inventário com a configuração
- Envia configuração por e-mail para a equipe de suporte
- NENHUM serviço criado (apenas configuração de equipamento)

O ponto chave: **O playbook define o comportamento, o produto é apenas um gatilho.**

## Plays vs Tarefas

Compreender a distinção entre Plays e Tarefas é fundamental para trabalhar com playbooks do OmniCRM.

### Play (Playbook)

Um fluxo de trabalho de provisionamento completo que orquestra várias tarefas para alcançar um objetivo comercial. Plays são os playbooks de nível superior armazenados em OmniCRM-API/Provisioners/plays/ e são referenciados nas definições de produtos.

Exemplos:

- `play_simple_service.yaml` - Provisionar um serviço básico
- `play_topup_no_charge.yaml` - Aplicar uma recarga gratuita a um serviço
- `play_prov_cpe_mikrotik.yaml` - Configurar equipamentos de premissas do cliente

### Tarefa (Componente Reutilizável)

Um conjunto de operações autônomas e reutilizáveis que podem ser incluídas por várias plays. As tarefas são prefixadas com `task_` e vivem no mesmo diretório.

Exemplos:

- `task_welcome_email.yaml` - Enviar um e-mail de boas-vindas a um cliente
- `task_activate_olt.yaml` - Ativar equipamentos OLT
- `task_notify_ocs.yaml` - Enviar notificações para o sistema de faturamento

A relação entre eles:

```
# play_simple_service.yaml (Uma Play)
- name: Play de Provisionamento Simples
  hosts: localhost
  tasks:
    - name: Bloco de provisionamento principal
      block:
        - name: Criar serviço
          uri: ...

        - name: Configurar faturamento
          uri: ...
```



```
# Incluir tarefa reutilizável
- include_tasks: task_welcome_email.yaml

# Incluir tarefas pós-provisionamento
- include_tasks: post_provisioning_tasks.yaml
```

## Estrutura e Anatomia do Playbook

Todos os playbooks do OmniCRM seguem uma estrutura consistente. Compreender essa estrutura é essencial para criar e manter playbooks.

### Estrutura Básica

Todo playbook começa com esses cabeçalhos padrão:

```
- name: Nome Descritivo do Playbook
  hosts: localhost           # Sempre localhost para OmniCRM
  gather_facts: no          # Desativado para desempenho
  become: False             # Não elevar privilégios

  tasks:
    - name: Bloco principal
      block:
        # Tarefas de provisionamento vão aqui

      rescue:
        # Tarefas de rollback/limpeza vão aqui
```

### Explicação do Cabeçalho

#### **name**

Nome descritivo exibido nos logs de provisionamento e na interface do usuário. Isso aparece como `playbook_description` no registro de provisionamento.

#### **hosts: localhost**

Todos os playbooks do OmniCRM são executados em localhost, pois interagem com sistemas remotos via APIs, não SSH.

#### **gather\_facts: no**

A coleta de fatos do Ansible está desativada porque:

- Não precisamos de informações do sistema
- Adiciona sobrecarga desnecessária
- Pode travar navegadores se exibido na saída de depuração

#### **become: False**

Nenhuma elevação de privilégios é necessária, pois estamos fazendo chamadas de

API, não modificando arquivos do sistema.

## Carregamento de Configuração

Todo playbook deve carregar o arquivo de configuração central:

```
tasks:
  - name: Incluir vars do crm_config
    ansible.builtin.include_vars:
      file: "../../crm_config.yaml"
      name: crm_config
```

Isso torna a configuração disponível como `crm_config.ocs.cgrates`, `crm_config.crm.base_url`, etc.

O `crm_config.yaml` geralmente contém:

```
ocs:
  cgrates: "10.0.1.100:2080"
  ocsTenant: "default_tenant"
crm:
  base_url: "https://crm.example.com"
```

## Padrões de Acesso a Variáveis

As variáveis podem vir de várias fontes:

### Da Definição do Produto:

```
- name: Acessar product_id passado pelo OmniCRM
  debug:
    msg: "Provisionando produto {{ product_id }}"
```

### Da Seleção de Inventário:

```
- name: Obter ID do inventário para Cartão SIM
  set_fact:
    sim_card_id: "{{ hostvars[inventory_hostname]['SIM Card'] | int
  }}"
  when: "'SIM Card' in hostvars[inventory_hostname]"
```

### Das Respostas da API:

```
- name: Obter informações do Produto da API do 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 o nome do produto
  debug:
    msg: "O nome do produto é {{
api_response_product.json.product_name }}"

```

## Padrões Comuns de Playbook

### Padrão de Provisionamento de Serviço

Este é o padrão mais comum para criar novos serviços.

```

- name: Playbook de Provisionamento de Serviço
  hosts: localhost
  gather_facts: no
  become: False

  tasks:
    - name: Bloco principal
      block:

        # 1. Carregar configuração
        - name: Incluir vars do crm_config
          ansible.builtin.include_vars:
            file: "../crm_config.yaml"
            name: crm_config

        # 2. Obter informações do produto
        - name: Obter informações do Produto da API do 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. Obter informações do cliente
        - name: Obter informações do Cliente da API do 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. Definir fatos a partir dos dados recuperados
- name: Definir fatos do pacote
  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. Gerar identificadores únicos
- name: Gerar UUID
  set_fact:
    uuid: "{{ 99999999 | random | to_uuid }}"

- name: Gerar UUID do Serviço
  set_fact:
    service_uuid: "Service_{{ uuid[0:8] }}"

# 6. Criar conta no sistema de faturamento
- name: Criar conta no 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 criação da conta OCS
  assert:
    that:
      - ocs_response.status == 200
      - ocs_response.json.result == "OK"

# 7. Adicionar saldo inicial
- name: Adicionar 0 Saldo Monetário
  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. Criar registro de serviço no CRM
- name: Obter data e hora atuais no formato ISO 8601
  command: date --utc +%Y-%m-%dT%H:%M:%S%z
  register: current_date_time

- name: Adicionar Serviço via 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. Adicionar transação de custo de configuração
- name: Adicionar Transação de Custo de Configuração via 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 }} - Custos de Configuração",
        "description": "Custos de configuração para {{
package_comment }}",
        "invoice_id": null,
        "retail_cost": "{{ setup_cost | float }}"
      }
    return_content: yes
    register: transaction_response

# 10. Incluir tarefas pós-provisionamento
- include_tasks: post_provisioning_tasks.yaml

rescue:

```

```

# Seção de rollback/limpeza
- name: Imprimir todas as variáveis para depuração
  debug:
    var: hostvars[inventory_hostname]

- name: Remover conta no 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: Deletar Serviço do CRM se foi criado
  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: Falhar se não for desprovisionamento intencional
  assert:
    that:
      - action == "deprovision"

```

## Padrão de Recarga

Usado para adicionar créditos, dados ou tempo a serviços existentes.

```

- name: Playbook de Recarga de Serviço
  hosts: localhost
  gather_facts: no
  become: False

```

```

tasks:
- name: Incluir vars do crm_config
  ansible.builtin.include_vars:
    file: "../../crm_config.yaml"
    name: crm_config

# 1. Obter informações do serviço
- name: Obter informações do Serviço da API do 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. Obter informações do produto (o que recarregar)
- name: Obter informações do Produto da API do 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. Extrair detalhes do serviço
- name: Definir fatos do serviço
  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. Executar ação no sistema de faturamento (recarga gratuita)
- name: Executar Ação para adicionar 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 se a ação foi executada com sucesso
  assert:
    that:
      - action_response.status == 200
      - action_response.json.result == "OK"

# 5. Redefinir quaisquer limites acionados
- name: Redefinir 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. Atualizar datas do serviço
- name: Calcular nova data de expiração
  command: "date --utc +%Y-%m-%dT%H:%M:%S%z -d '+30 days'"
  register: new_expiry_date

- name: Atualizar Serviço com nova expiração
  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 notificação
- name: Enviar SMS de Notificação
  uri:
    url: "http://sms-gateway/api/send"
    method: POST
    body_format: json
    body:
      {
        "source": "CompanyName",
        "destination": "{{ customer_phone }}",
        "message": "Seu serviço foi recarregado. Nova expiração:
{{ new_expiry_date.stdout }}"
      }
    status_code: 201
    ignore_errors: True
```

## Padrão de Provisionamento de CPE

Usado para configurar equipamentos de premissas do cliente (roteadores, modems, ONTs).

```
- name: Playbook de Provisionamento de CPE
  hosts: localhost
  gather_facts: no
  become: False

  tasks:
    - name: Incluir vars do crm_config
      ansible.builtin.include_vars:
        file: "../../crm_config.yaml"
        name: crm_config

    # 1. Obter item de inventário para CPE
    - name: Definir ID do inventário de CPE a partir de hostvars
      set_fact:
        cpe_inventory_id: "{{ hostvars[inventory_hostname]['WiFi
Router CPE'] | int }}"
      when: "'WiFi Router CPE' in hostvars[inventory_hostname]"

    # 2. Obter detalhes do CPE do inventário
    - name: Obter dados do Inventário 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. Obter informações do site do cliente
- name: Obter informações do Site da 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. Atualizar inventário de CPE com localização
- name: Patch item de inventário de CPE com localização
  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. Gerar credenciais
- name: Definir nome do host do CPE
  set_fact:
    cpe_hostname: "CPE_{{ cpe_inventory_id }}"
    cpe_username: "admin_{{ cpe_inventory_id }}"

- name: Gerar senha aleatória
  set_fact:
    cpe_password: "{{ lookup('pipe', 'cat /dev/urandom | tr -dc
a-zA-Z0-9 | head -c 16') }}"

# 6. Gerar credenciais WiFi
- name: Definir SSID do WiFi
  set_fact:

```

```

    wifi_ssid: "Network_{{ cpe_inventory_id }}"

- name: Gerar senha WiFi
  set_fact:
    word_list:
      - apple
      - cloud
      - river
      - mountain
      - ocean

- name: Criar PSK WiFi
  set_fact:
    random_word: "{{ word_list | random }}"
    random_number: "{{ 99999 | random(start=10000) }}"

- name: Combinar PSK WiFi
  set_fact:
    wifi_psk: "{{ random_word }}{{ random_number }}"

# 7. Gerar arquivo de configuração
- name: Definir nome do arquivo de configuração
  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: Criar configuração a partir do template
  template:
    src: "templates/cpe_router_config.j2"
    dest: "{{ config_dest }}"

# 8. Ler configuração gerada
- name: Ler arquivo de configuração
  ansible.builtin.slurp:
    src: "{{ config_dest }}"
  register: config_content

# 9. Atualizar inventário com informações de provisionamento
- name: Patch CPE inventário com configuração
  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": "Provisionado: {{ lookup('pipe', 'date
+%Y-%m-%d') }}"
}
status_code: 200

# 10. Enviar configuração para a equipe de suporte
- name: E-mail de configuração para suporte
  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 Provisionamento"
          },
          "To": [{
            "Email": "support@example.com",
            "Name": "Equipe de Suporte"
          }],
          "Subject": "Configuração 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

```

## Padrão de Renovação Automática

Configurar cobranças ou renovações recorrentes automáticas usando Planos de Ação do CGRateS.

```
# Parte de um playbook de recarga que configura a renovação automática
```

```
# 1. Normalizar parâmetro auto_renew
```

```
- name: Normalizar auto_renew para booleano
  set_fact:
    auto_renew_bool: "{{ (auto_renew | string | lower) in ['true', '1', 'yes'] }}"
```

```
# 2. Criar ação para renovação automática
```

```
- name: Criar Ação para AutoRenovação
  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. Criar AçãoPlano mensal
```

```

- name: Criar AçãoPlano para Renovação Mensal
  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. Atribuir AçãoPlano à conta

```

- name: Atribuir AçãoPlano à conta
  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. Remover AçãoPlano se a renovação automática estiver desativada
- name: Remover AçãoPlano da conta
  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

```

## Tarefas Reutilizáveis

Tarefas reutilizáveis são pequenos playbooks autônomos que podem ser incluídos por várias plays. Elas promovem a reutilização de código e a consistência.

### Tarefa de E-mail de Boas-Vindas

task\_welcome\_email.yaml - Envia um e-mail de boas-vindas para novos clientes.

```

# Esta tarefa espera que essas variáveis sejam definidas pela play
pai:
# - api_response_customer (detalhes do cliente)
# - package_name (nome do produto)
# - monthly_cost (custo recorrente)
# - setup_cost (custo único)

- name: Definir configuração de e-mail
  set_fact:
    mailjet_api_key: "{{ lookup('env', 'MAILJET_API_KEY') }}"
    mailjet_api_secret: "{{ lookup('env', 'MAILJET_SECRET') }}"
    email_from: "noreply@example.com"
    recipients: []

- name: Definir assunto do e-mail e nome do remetente
  set_fact:
    email_subject: "Bem-vindo ao nosso serviço!"

```



```

    email_from_name: "Equipe de Atendimento ao Cliente"

- name: Preparar lista de destinatários a partir dos contatos do
cliente
  loop: "{{ api_response_customer.json.contacts }}"
  set_fact:
    recipients: "{{ recipients + [{'Email': item.contact_email,
'Name': item.contact_firstname ~ ' ' ~ item.contact_lastname}] }}"

- name: Obter nome do primeiro contato
  set_fact:
    first_contact: "{{
api_response_customer.json.contacts[0].contact_firstname }}"

- name: Enviar e-mail de boas-vindas
  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": "Caro {{ first_contact }}, bem-vindo! Seu
serviço está pronto.",
          "HTMLPart": "Caro {{ first_contact }},<br/><h3>Bem-
vindo!</h3><br/>Seu serviço {{ package_name }} agora está
ativo.<br/>Custo mensal: ${{ monthly_cost }}<br/>Taxa de
configuração: ${{ setup_cost }}<br/>Se você tiver alguma dúvida,
entre em contato com support@example.com"
        }]
      }
    user: "{{ mailjet_api_key }}"
    password: "{{ mailjet_api_secret }}"
    force_basic_auth: true
    status_code: 200
  register: email_response

```

## Tarefas Pós-Provisionamento

post\_provisioning\_tasks.yaml - Limpeza padrão e notificações executadas após

cada provisionamento.

```
# Este arquivo é incluído no final da maioria dos playbooks de
provisionamento
# Ele lida com operações comuns pós-provisionamento
```

```
- include_tasks: task_notify_ocs.yaml
```

O task\_notify\_ocs.yaml pode conter:

```
- name: Notificar OCS da conclusão do provisionamento
  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
```

## Operações Comuns

### Trabalhando com Inventário

Recuperando detalhes do inventário:

```
- name: Obter ID do inventário do Cartão SIM
  set_fact:
    sim_inventory_id: "{{ hostvars[inventory_hostname]['SIM Card'] |
int }}"
  when: "'SIM Card' in hostvars[inventory_hostname]"

- name: Obter detalhes do Cartão 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: Extrair detalhes do SIM
```

```
set_fact:
  iccid: "{{ sim_response.json.iccid }}"
  imsi: "{{ sim_response.json.imsi }}"
  ki: "{{ sim_response.json.ki }}"
```

Atribuindo inventário ao cliente:

```
- name: Atribuir SIM ao 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
```

## Operações de Data e Hora

Obtendo data/hora atuais:

```
- name: Obter data e hora atuais no formato ISO 8601
  command: date --utc +%Y-%m-%dT%H:%M:%S%z
  register: current_date_time

- name: Obter apenas a data de hoje
  set_fact:
    today: "{{ lookup('pipe', 'date +%Y-%m-%d') }}"
```

Calculando datas futuras:

```
- name: Calcular data de expiração 30 dias a partir de agora
  command: "date --utc +%Y-%m-%dT%H:%M:%S%z -d '+30 days'"
  register: expiry_date

- name: Calcular data 90 dias no futuro
  command: "date --utc +%Y-%m-%d -d '+{{ days }} days'"
  register: future_date
  vars:
    days: 90
```

## Gerando Valores Aleatórios

UUIDs e identificadores:

```
- name: Gerar UUID
  set_fact:
    uuid: "{{ 99999999 | random | to_uuid }}"

- name: Gerar identificador do serviço
  set_fact:
    service_uuid: "SVC_{{ uuid[0:8] }}"
```

Senhas aleatórias:

```
- name: Gerar senha segura
  set_fact:
    password: "{{ lookup('pipe', 'cat /dev/urandom | tr -dc a-zA-Z0-9
| head -c 16') }}"
```

Frases de senha memoráveis:

```
- name: Definir lista de palavras
  set_fact:
    words:
      - alpha
      - bravo
      - charlie
      - delta
      - echo

- name: Gerar frase de senha
  set_fact:
    word: "{{ words | random }}"
    number: "{{ 99999 | random(start=10000) }}"

- name: Combinar em frase de senha
  set_fact:
    passphrase: "{{ word }}{{ number }}"
```

## Trabalhando com CGRateS/OCS

Criando contas:

```
- name: Criar conta de faturamento
  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

```

Adicionando saldos:

```

- name: Adicionar saldo de dados
  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": "Pacote de Dados",
            "Value": 10737418240,
            "ExpiryTime": "+720h",
            "Weight": 10
          }
        }]
      }
    status_code: 200

```

Executando ações:

```

- name: Executar ação de cobrança
  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

```

Obtendo informações da conta:

```

- name: Obter detalhes da conta
  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

```

Trabalhando com Perfis de Atributo:

```

- name: Obter 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: Extrair valor do 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 se as variáveis existem:

```
- name: Usar valor personalizado ou padrão
  set_fact:
    monthly_cost: "{{ custom_cost | default(50.00) }}"

- name: Executar apenas se a variável estiver definida
  debug:
    msg: "UUID do Serviço é {{ service_uuid }}"
  when: service_uuid is defined
```

Condições booleanas:

```
- name: Provisionar equipamento
  include_tasks: configure_cpe.yaml
  when: provision_cpe | default(false) | bool

- name: Pular se desprovisionar
  assert:
    that:
      - action != "deprovision"
  when: action is defined
```

Múltiplas condições:

```
- name: Tarefa condicional complexa
  uri:
    url: "{{ endpoint }}"
    method: POST
  when:
    - service_uuid is defined
    - customer_id is defined
    - action != "deprovision"
    - enable_feature | default(true) | bool
```

## Laços e Iteração

Laços simples:

```

- name: Criar múltiplos 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 respostas da API:

```

- name: Obter todos os sites do 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 equipamento em cada site
  debug:
    msg: "Configurando site em {{ item.address_line_1 }}"
  loop: "{{ sites_response.json }}"

```

## Tratamento de Erros

Usando ignore\_errors:

```

- name: Notificação SMS opcional
  uri:
    url: "http://sms-gateway/send"
    method: POST
    body: {...}
  ignore_errors: true

```



Asserções para validação:

```
- name: Verificar resposta da API
  assert:
    that:
      - response.status == 200
      - response.json.result == "OK"
    fail_msg: "Chamada da API falhou: {{ response.json }}"
```

Tratamento condicional de erros:

```
- name: Tentar obter serviço 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: Criar serviço se não existir
  uri:
    url: "{{ crm_config.crm.base_url }}/crm/service/"
    method: PUT
    body: {...}
  when: service_lookup.status == 404
```

## Melhores Práticas

### Nomeação de Variáveis

Use nomes descritivos e consistentes:

```
# Bom
service_uuid: "SVC_12345"
customer_name: "John Smith"
monthly_cost: 49.99

# Ruim
svc: "SVC_12345"
name: "John Smith"
cost: 49.99
```

Prefixe variáveis por origem:

```
api_response_customer: {...}
api_response_product: {...}
```

```
cgr_account_info: {...}
```

## Depuração

Imprima variáveis para solução de problemas:

- `name`: Imprimir todas as variáveis  
`debug`:  
    `var`: `hostvars[inventory_hostname]`
- `name`: Imprimir variável específica  
`debug`:  
    `msg`: "UUID do Serviço: `{{ service_uuid }}`"
- `name`: Imprimir resposta da API  
`debug`:  
    `var`: `api_response_product.json`

## Validação

Sempre valide respostas críticas da API:

- `name`: Criar conta  
`uri`:  
    `url`: "`{{ billing_endpoint }}`"  
    `method`: POST  
    `body`: {...}  
`register`: response
- `name`: Verificar criação da conta  
`assert`:  
    `that`:  
  - `response.status == 200`
  - `response.json.result == "OK"`  
    `fail_msg`: "Falha ao criar conta: `{{ response.json }}`"

## Idempotência

Projete tarefas para serem reexecutáveis com segurança:

```
# Verifique se o recurso existe primeiro  
- name: Verificar se a conta existe  
  uri:  
    url: "{{ ocs_endpoint }}/get_account"  
    method: POST  
    body: {"Account": "{{ service_uuid }}"}  
  register: account_check  
  failed_when: false
```

```
# Apenas crie se não existir
- name: Criar conta
  uri:
    url: "{{ ocs_endpoint }}/create_account"
    method: POST
    body: {...}
  when: account_check.status == 404
```

## Segurança

Nunca codifique credenciais:

```
# Ruim
mailjet_api_key: "abc123def456"

# Bom - use variáveis de ambiente
mailjet_api_key: "{{ lookup('env', 'MAILJET_API_KEY') }}"

# Bom - use arquivo de configuração
mailjet_api_key: "{{ crm_config.email.api_key }}"
```

Sempre use HTTPS e autenticação:

```
- name: Chamar API externa
  uri:
    url: "https://api.example.com/endpoint"
    method: POST
    headers:
      Authorization: "Bearer {{ access_token }}"
    validate_certs: yes
```

## Documentação

Documente lógica complexa:

```
# Calcular cobrança pro-rata para mês parcial
# Se o cliente se inscrever no dia 15 e a cobrança for no dia 1,
# cobrar 50% do custo mensal pelos dias restantes
- name: Calcular dias até o final do mês
  command: "date -d 'last day of this month' +%d"
  register: days_in_month

- name: Obter dia atual
  command: "date +%d"
  register: current_day

- name: Calcular valor pro-rata
```

```
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) }}"
```

## Testando Playbooks

### Abordagem de Teste

1. **Executar Teste Primeiro:** Testar com sistemas não produtivos
2. **Verificar Variáveis:** Usar tarefas de depuração para confirmar que todas as variáveis necessárias estão presentes
3. **Verificar Respostas:** Validar respostas da API antes de prosseguir
4. **Teste de Rollback:** Falhar intencionalmente tarefas para verificar se os blocos de resgate funcionam
5. **Teste de Desprovisionamento:** Testar com action: "deprovision" para verificar limpeza

Exemplo de playbook de teste:

```
- name: Testar Provisionamento de Serviço
  hosts: localhost
  gather_facts: no

  tasks:
    - name: Verificar variáveis necessárias
      assert:
        that:
          - product_id is defined
          - customer_id is defined
          - access_token is defined
        fail_msg: "Variáveis necessárias ausentes"

    - name: Testar conectividade da API
      uri:
        url: "http://localhost:5000/crm/health"
        method: GET
      register: health_check

    - name: Verificar verificação de saúde
      assert:
        that:
          - health_check.status == 200
```

## Armadilhas Comuns

### Faltando conversões de tipo:

```
# Errado - pode ser string
customer_id: "{{ customer_id }}"

# Correto - garantir inteiro
customer_id: {{ customer_id | int }}
```

### Não lidando com variáveis indefinidas:

```
# Errado - falha se não definido
service_uuid: "{{ service_uuid }}"

# Correto - fornecer padrão
service_uuid: "{{ service_uuid | default('') }}"
```

### Esquecendo validação:

```
# Errado - não verifica resposta
- name: Criar conta
  uri: ...
  register: response

# Correto - valida resposta
- name: Criar conta
  uri: ...
  register: response

- name: Verificar criação
  assert:
    that:
      - response.json.result == "OK"
```

## Fluxo de Trabalho de Provisionamento

Geralmente, a equipe da Omnitouch trabalhará com o cliente para:

1. Definir os requisitos do produto
2. Desenvolver os playbooks Ansible necessários para automatizar o processo de provisionamento
3. Testar os playbooks em um ambiente de staging
4. Implantar em produção

Isso garante que cada serviço seja implantado de forma consistente e confiável, reduzindo o risco de erros e garantindo que todas as etapas necessárias sejam concluídas na ordem correta.

# Variáveis Ansible

As variáveis passadas para os playbooks Ansible incluem:

## Variáveis do Produto

Derivadas das configurações do produto OmniCRM e definem como o serviço deve ser configurado.

## Variáveis de Inventário

Selecioneadas do inventário, incluem itens como modems, cartões SIM, blocos de endereços IP ou números de telefone que são necessários para provisionamento.

## Variáveis do Sistema

Adicionadas automaticamente pelo OmniCRM:

- `product_id` - O produto sendo provisionado
- `customer_id` - O cliente recebendo o serviço
- `service_id` - O serviço sendo modificado (para recargas/mudanças)
- `access_token` - JWT para autenticação da API

# Desprovisionamento

Quando um serviço não é mais necessário, os **Playbooks Ansible** também são usados para desprovisionar o serviço usando o padrão de bloco `rescue`. Isso:

- Remove quaisquer configurações
- Libera o inventário de volta para o pool
- Deleta contas de faturamento
- Garante que o sistema permaneça limpo

# Rollback e Tratamento de Erros

O recurso **`block/rescue`** do Ansible é empregado durante o provisionamento e desprovisionamento para lidar com erros de forma elegante. Se uma tarefa falhar em qualquer ponto durante o provisionamento, a seção de resgate reverte automaticamente as alterações para retornar a um estado consistente. Isso garante confiabilidade e reduz o risco de implantações parciais ou falhadas.

Para detalhes completos sobre o sistema de provisionamento, fluxos de trabalho e autenticação, consulte `concepts_provisioning`.



# OmniCRM API

Todas as funções dentro do OmniCRM são acessíveis via API - Não há funcionalidade que esteja disponível apenas na interface do usuário.

Isso permite que você integre o OmniCRM com outros sistemas ou automatize tarefas.

A API é uma API RESTful e é protegida usando múltiplos métodos de autenticação, incluindo tokens JWT, chaves de API e lista de permissões de IP.

A documentação da API é feita usando Swagger, uma ferramenta que permite a fácil leitura, compreensão e teste da funcionalidade da API.

A documentação da API está disponível no seguinte URL:

<<https://yourcrm/crm/docs/>>

Página Swagger do OmniCRM

## Métodos de Autenticação

O OmniCRM suporta três métodos de autenticação, cada um projetado para casos de uso diferentes:

1. **Tokens Bearer JWT** - Para sessões de usuário interativas (UI Web, aplicativos móveis)
2. **Chaves de API** - Para integrações servidor-a-servidor e scripts de automação
3. **Lista de Permissões de IP** - Para sistemas internos confiáveis (servidores de provisionamento, ferramentas de monitoramento)

## Autenticação com Token Bearer JWT

Este é o método de autenticação principal para sessões de usuário. Os usuários fazem login com e-mail e senha, recebem um token JWT e o utilizam para requisições subsequentes.

### Casos de Uso:

- Autenticação na UI Web
- Autenticação em aplicativos móveis
- Acesso programático de curta duração

## Como Autenticar:

Para fazer login, envie um corpo JSON com a seguinte estrutura para `/crm/auth/login` como uma requisição POST:

```
{
  "email": "youruser@yourdomain.com",
  "password": "yourpassword"
}
```

A API retornará um objeto JSON contendo um campo `token`, que é usado para autenticar todas as requisições futuras. Além disso, a resposta inclui um `refresh_token` que pode ser usado para atualizar o token quando ele expira, juntamente com as permissões e funções do usuário.

Você pode testar isso na página Swagger selecionando o endpoint `/auth/login`, preenchendo seu Nome de Usuário e Senha, e clicando no botão `Try it out`.

Página Swagger do OmniCRM

Para autorizar a sessão, copie o valor do token e clique no botão "Authorize" no canto superior direito da página Swagger. Cole o token no campo "Value", prefixado por `Bearer` e clique em "Authorize".

Autenticação Bearer do OmniCRM

Agora, todas as requisições subsequentes serão autenticadas com este token.

## Autenticação com Chave de API

As chaves de API fornecem autenticação segura e de longa duração para integrações servidor-a-servidor e scripts de automação sem exigir senhas de usuário.

### Casos de Uso:

- Sistemas de provisionamento automatizados
- Ferramentas de monitoramento e alerta
- Integração com sistemas externos
- Tarefas agendadas e cron jobs

### Como as Chaves de API Funcionam:

As chaves de API são configuradas no arquivo `crm_config.yaml` e estão associadas a funções e permissões específicas. Cada chave de API é uma string aleatória segura (mínimo 32 caracteres) que autentica requisições quando passada no cabeçalho `X-API-KEY`.

### Configurando Chaves de API:



As chaves de API devem ser adicionadas ao `crm_config.yaml` por um administrador com acesso ao servidor:

```
api_keys:
  sua-chave-api-segura-aqui-minimo-32-caracteres:
    roles:
      - admin
    description: "Sistema de automação de provisionamento"
  outra-chave-api-para-sistema-de-monitoramento:
    roles:
      - view_customer
      - view_service
    description: "Monitoramento e alerta"
```

### Usando Chaves de API:

Inclua a chave de API no cabeçalho X-API-KEY de suas requisições:

```
curl -X GET "https://yourcrm.com/crm/customers" \
-H "X-API-KEY: sua-chave-api-segura-aqui-minimo-32-caracteres"
```

### Exemplo em Python:

```
import requests

crm_url = 'https://yourcrm.com'
api_key = 'sua-chave-api-segura-aqui-minimo-32-caracteres'

headers = {
    "Content-Type": "application/json",
    "X-API-KEY": api_key
}

# Obter Clientes
response = requests.get(crm_url + '/crm/customers', headers=headers)
for customer in response.json()['data']:
    print(customer)
```

### Melhores Práticas:

- Gere chaves de API usando geradores aleatórios seguros (`openssl rand -base64 48`)
- Use chaves de API diferentes para diferentes sistemas
- Documente o propósito de cada chave de API no campo `description`
- Rotacione as chaves de API periodicamente
- Nunca comite chaves de API no controle de versão
- Atribua permissões mínimas necessárias a cada chave de API

## Autenticação com Lista de Permissões de IP

A lista de permissões de IP permite que endereços IP específicos acessem a API sem autenticação. Isso é útil para sistemas internos confiáveis em redes privadas.

### Casos de Uso:

- Servidores de provisionamento internos
- Sistemas de monitoramento de rede em VLANs de gerenciamento
- Playbooks do Ansible executando em infraestrutura controlada

### Configurando a Lista de Permissões de IP:

Adicione endereços IP confiáveis ao `crm_config.yaml`:

```
ip_whitelist:  
- 192.168.1.100  
- 10.0.0.0/24  
- 172.16.50.10
```

### Considerações de Segurança:

- Use a lista de permissões de IP apenas em redes privadas e seguras
- Nunca adicione endereços IP públicos à lista de permissões
- Use os intervalos de IP mais específicos possíveis
- Documente por que cada IP está na lista de permissões
- Audite regularmente os IPs na lista de permissões

## Exemplos de Chamadas de API com Python

Aqui está um exemplo de como fazer login e recuperar uma lista de clientes usando autenticação com token JWT:

```
import requests  
  
crm_url = 'https://yourcrm.com'  
session = requests.Session()  
  
print("Provisionando dados para o servidor: " + str(crm_url))  
  
headers = {  
    "Content-Type": "application/json"  
}  
  
# Obter Token de Autenticação  
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 no CRM com sucesso")

# Obter Clientes
response = session.get(crm_url + '/crm/customers', headers=headers)
for customer in response.json()['data']:
    print(customer)
```



# Notas de Cobrança de Serviços / Produtos CRM

Nota

Para um guia completo de ponta a ponta cobrindo definição de produtos, provisionamento, complementos e desprovisionamento com exemplos detalhados de Ansible e estratégia de preços, veja Complete Product Lifecycle Guide <guide\_product\_lifecycle>.

## Visão Geral de Produtos e Serviços

### Produto (Item do Menu):

Um Produto é como um prato específico em um menu de restaurante, como um "Spaghetti Carbonara."

Ele tem uma descrição clara, uma lista de ingredientes (como massa, creme, ovos, queijo e bacon) e um preço.

No OmniCRM, um Produto contém de forma semelhante os detalhes do que está incluído — recursos, especificações e preços.

Frequentemente, os clientes podem querer modificações, como "sem cebolas" ou "adicionar queijo extra" ao seu prato. Dentro do OmniCRM, isso corresponde a personalizar um serviço antes da criação. O nível de personalizações ou modificações a um serviço fica a seu critério (do operador) para definir.

No OmniCRM, clientes ou funcionários podem modificar um Produto para melhor atender às necessidades de um cliente específico, como aumentar a velocidade da Internet ou adicionar recursos específicos. Essa personalização é refletida no Serviço específico fornecido.

Um produto é essencialmente uma oferta que os clientes podem escolher para pedir, semelhante a ler e escolher um prato do menu.

Definições de Produto OmniCRM

### Catálogo de Produtos (Menu do Restaurante):

O Catálogo de Produtos é como todo o menu em um restaurante, que lista todos os pratos disponíveis — de aperitivos a sobremesas.

É a coleção completa de tudo que o restaurante (ou, no seu caso, o Provedor de Serviços) tem a oferecer.

No contexto empresarial, o Catálogo de Produtos fornece aos clientes todos os Produtos disponíveis, para que possam escolher o que melhor atende às suas necessidades.

Interface de Gerenciamento de Produtos Visão de Edição do Catálogo de Produtos

## Serviço (Prato Preparado):

Quando um cliente pede um item do menu, o prato é preparado na cozinha. Isso é semelhante a criar um Serviço a partir de um Produto.

No OmniCRM, quando um cliente seleciona um Produto, uma instância desse Produto é criada e entregue como um Serviço.

Ele é personalizado e preparado especificamente para aquele cliente, assim como uma refeição preparada para um cliente.

Por exemplo, quando alguém seleciona o "Plano de Internet Bronze" do Catálogo de Produtos, o sistema de provisionamento "cozinha" cria uma instância desse plano a partir dos ingredientes (endereços IP, Modens e Portas) — ou seja, ativa o plano e o entrega ao cliente específico.

## Produtos Agrupados (Refeições Combinadas):

O Catálogo de Produtos também pode oferecer pacotes, como uma refeição combinada que inclui um aperitivo, prato principal e sobremesa juntos por um preço especial.

No OmniCRM, Produtos agrupados combinam vários Produtos individuais em um pacote conveniente — como um "Pacote de Essenciais para o Lar" que inclui serviços de Internet, TV e telefone a uma tarifa com desconto.

Uma vez selecionado, esse pacote é transformado em vários Serviços adaptados para o cliente.

## Definições de Produto

Um produto é um modelo que é usado para criar um serviço / complemento / desconto / adição, etc.

Dentro da definição, incluímos:

- Informações sobre o produto (recursos, inclusões, T&Cs, duração do contrato, ícone, etc.) que são exibidas para o usuário do CRM (Cliente ou *Funcionário*).
- A lógica de negócios sobre quem pode comprar o produto (*Empresarial* ou *Residencial*), se depende de ter um serviço pai provisionado (como complementos móveis disponíveis apenas para clientes com um serviço móvel), se pode ser solicitado diretamente por um cliente via autoatendimento ou apenas por um agente de atendimento ao cliente, e quando o produto pode ser adquirido (permitindo que um produto esteja disponível apenas por um período definido).
- Quando itens de Inventário devem ser incluídos (como Modens ou Cartões SIM) estes são especificados como Lista de Itens de Inventário, por exemplo, o serviço abaixo requer um Cartão SIM e um Número de Telefone a serem atribuídos:

[ 'SIM Card', 'Phone Number' ] Estes se correlacionam com os Inventory Items <administration\_inventory> definidos no CRM.

- Referenciar um Playbook Ansible para provisionar o serviço Provisioning Play <concepts\_ansible> assim como as variáveis a passar para o Ansible. Essas variáveis a passar são mágicas, pois podem ser variáveis como `service_id` que são definidas pelo produto ao qual estamos adicionando, ou podem ser como ICCID & MSISDN onde temos itens de inventário selecionados que são passados ao atribuir o inventário. O agrupamento é tratado no playbook de provisionamento para conter múltiplos serviços, por exemplo, um

produto de internet residencial, TV e Voz pode provisionar um serviço para cada.

Definições de Produto OmniCRM

## Categorias de Produto e Tipos de Serviço

Os produtos usam dois campos de classificação para ajudar a organizar e filtrar ofertas:

### Categorias de Produto

O campo `category` controla onde os produtos são exibidos na interface do usuário. Os valores comuns incluem:

- **standalone** - Exibido como uma opção de serviço base ao criar um novo serviço
- **addon** - Exibido ao adicionar a um serviço existente
- **bundle** - Exibido como uma opção de serviço agrupado (provisionado como um complemento para serviços existentes)
- **promo** - Ofertas promocionais especiais

Essas categorias são puramente organizacionais e não ditam o que é provisionado. O comportamento real de provisionamento é determinado inteiramente pelo playbook Ansible referenciado em `provisioning_play`.

Por exemplo: - Um produto `standalone` tipicamente cria um novo objeto de serviço

- Um produto `addon` ou `bundle` é tipicamente adicionado a um serviço existente - Mas isso depende do implementador que escreve o playbook - você poderia criar múltiplos objetos de serviço a partir de um complemento, ou modificar serviços existentes a partir de um produto autônomo, se necessário

A categoria simplesmente controla o fluxo da interface do usuário e onde os clientes/funcionários veem a opção do produto.

### Tipos de Serviço

O campo `service_type` categoriza que tipo de serviço está sendo fornecido.

Esses são inteiramente definidos pelo usuário, mas os valores comuns incluem:

- **mobile** - Serviços de telefone móvel com voz, SMS e dados
- **fixed** - Serviços de internet fixa sem fio ou com fio
- **fixed-voice** - Serviços de voz fixa (VoIP, linha fixa)
- **hotspot** - Dispositivos móveis de hotspot ou aluguel
- **dongle** - Serviços de modem USB ou dongle
- **voice** - Serviços somente de voz
- **data** - Serviços somente de dados

Assim como as categorias, os tipos de serviço são personalizáveis com base em suas ofertas. Eles ajudam em:

- Filtrar quais complementos se aplicam a quais serviços base
- Organizar produtos no portal do cliente
- Correspondência de requisitos de inventário
- Determinar fluxos de trabalho de provisionamento

Exemplo: Um cliente com um serviço `mobile` pode ver complementos móveis, enquanto um cliente com um serviço `fixed` vê complementos de linha fixa.

# Gerenciando Produtos

Os produtos são gerenciados através da página de Gerenciamento de Produtos, onde você pode visualizar, pesquisar, filtrar e editar todos os produtos disponíveis.

Página de Lista de Produtos

## Interface Modal de Produto

Clicando em qualquer produto, uma interface de abas aprimorada é aberta que organiza todas as configurações do produto em grupos lógicos para facilitar a navegação e edição.

Modal de Produto - Aba de Informações Básicas

O modal de gerenciamento de produtos possui cinco abas organizadas:

1. **Informações Básicas** - Informações principais do produto (nome, slug, categoria, ícone, recursos, termos)
2. **Preços** - Todos os campos relacionados a custos, incluindo custos recorrentes, custos de configuração e porcentagem de imposto
3. **Configuração** - Configurações de renovação, tipos de clientes e dependências
4. **Provisionamento** - Configuração do playbook Ansible e requisitos de inventário
5. **Disponibilidade** - Intervalos de datas e timestamps do sistema

Modal de Produto - Aba de Preços

### Organização da Aba de Preços:

A aba de Preços agrupa os campos de custo em seções lógicas:

- **Custos Recorrentes** - Custos mensais de varejo e atacado lado a lado
- **Custos de Configuração** - Taxas de ativação únicas para varejo e atacado
- **Imposto** - Configuração da porcentagem de imposto com cálculo automático

### Recursos do Modo de Edição:

- **Selecionador de Ícones** - Pesquisar e selecionar ícones FontAwesome visualmente
- **Selecionador de Itens de Inventário** - Selecionar entre os tipos de itens de inventário disponíveis
- **Selecionador de Data/Hora** - Seleção fácil de janelas de disponibilidade
- **Formatação de Moeda** - Prefixo \$ automático para campos de custo
- **Seletores de Dropdown** - Opções pré-definidas para categorias e campos booleanos

Modal de Produto - Modo de Edição de Informações Básicas

### Selecionador de Ícones:

Ao editar o campo de ícone, uma interface de seleção de ícones pesquisável aparece permitindo que você navegue visualmente e selecione entre milhares de ícones FontAwesome.

Modal de Produto - Selecionador de Ícones

Recursos: \* Pesquisar ícones por palavra-chave (por exemplo, "chave inglesa", "móvel", "wifi") \* Visualizar a aparência do ícone em tempo real \* Mostra o nome da classe do ícone para referência \* Seleção por dropdown para acesso rápido

### Aba de Configuração:

A aba de Configuração organiza as configurações de comportamento do produto em grupos lógicos.

Modal de Produto - Aba de Configuração

### Seções de Configuração:

- **Configurações de Renovação:**
  - Renovação Automática - Comportamento padrão de renovação (Prompt/Sim/Não)
  - Permitir Renovação Automática - Se os clientes podem habilitar a renovação automática
  - Dias de Contrato - Duração mínima do contrato (por exemplo, 30 para mensal, 365 para anual)
- **Tipos de Clientes:**
  - Residencial - Disponível para clientes consumidores
  - Empresarial - Disponível para clientes comerciais
- **Dependências:**
  - Lista de Dependências - IDs de produtos ou tipos de serviço necessários antes que este produto possa ser adicionado
  - Usado para dependências de complementos (por exemplo, complementos móveis requerem serviço móvel ativo)

### Aba de Provisionamento:

A aba de Provisionamento lida com automação Ansible e requisitos de inventário.

Modal de Produto - Aba de Provisionamento

### Campos de Provisionamento:

- **Play de Provisionamento:**
  - Nome do playbook Ansible (sem extensão .yaml)
  - Deve existir no diretório OmniCRM-API/Provisioners/plays/
  - Chamado quando o serviço é criado, atualizado ou desprovisionado
- **Variáveis JSON de Provisionamento:**
  - Variáveis padrão passadas para o playbook Ansible como JSON
  - Podem ser substituídas durante o provisionamento
  - O playbook recebe estas mais customer\_id, product\_id, service\_id, access\_token
- **Lista de Itens de Inventário:**
  - Seleccionador de múltiplas opções mostrando os tipos de itens de inventário disponíveis
  - Exemplos: Cartão SIM, Número de Telefone, Modem Roteador, Endereço IPv4
  - O cliente/funcionário seleciona itens específicos do inventário disponível durante o pedido
  - IDs de inventário selecionados passados para o playbook com o tipo de inventário como nome da variável

### Aba de Disponibilidade:

A aba de Disponibilidade controla quando o produto pode ser adquirido e exibe metadados do sistema.

Modal de Produto - Aba de Disponibilidade

### Configurações de Disponibilidade:

- **Disponível A Partir de:**
  - Data/hora quando o produto se torna disponível para compra



- Deixe em branco para disponibilidade imediata
- Útil para pré-anunciar novos produtos
- **Disponível Até:**
  - Data/hora quando o produto não está mais disponível para compra
  - Deixe em branco para disponibilidade indefinida
  - Perfeito para promoções por tempo limitado ou produtos fora de linha
- **Metadados do Sistema (Somente Leitura):**
  - Criado - Timestamp quando o produto foi criado pela primeira vez
  - Última Modificação - Timestamp da atualização mais recente
  - Mantido automaticamente pelo sistema

### Ações do Modal:

- **Modo de Visualização:**
  - Fechar - Descartar modal
  - Clonar Produto - Criar uma cópia com sufixo "\_clone"
  - Editar Produto - Mudar para modo de edição
- **Modo de Edição/Criar:**
  - Cancelar - Descartar alterações e fechar
  - Salvar Alterações - Criar ou atualizar produto (botão grande para ênfase)

## Campos de Produto

O modelo de Produto contém todas as informações necessárias para definir uma oferta e como ela deve ser provisionada. Esses campos são gerenciados através da interface modal de Gerenciamento de Produtos descrita acima.

### Informações Básicas

- **product\_id** - Identificador único atribuído automaticamente pelo sistema
- **product\_name** - Nome exibido mostrado aos clientes e funcionários na interface do usuário
- **product\_slug** - Identificador único usado em URLs e chamadas de API (minúsculas, sem espaços, use hífen)
- **category** - Controla onde este produto aparece na interface do usuário:
  - standalone - Exibido como uma opção de serviço base ao criar um novo serviço
  - addon - Exibido ao adicionar a um serviço existente
  - bundle - Exibido como uma opção de serviço agrupado
  - promo - Ofertas promocionais especiais
- **service\_type** - Tipo de serviço sendo fornecido (por exemplo, móvel, fixo, voz-fixa, hotspot, dongle, voz, dados). Usado para filtrar quais complementos se aplicam a quais serviços.
- **comment** - Notas internas sobre o produto para referência do funcionário apenas (não exibido para clientes)
- **icon** - Classe de ícone FontAwesome exibida na interface do usuário (por exemplo, fa-solid fa-sim-card)

### Campos de Preço

- **retail\_cost** - Cobrança mensal recorrente faturada ao cliente (defina como 0 para compras únicas ou produtos pré-pagos)
- **wholesale\_cost** - Seu custo mensal para fornecer este serviço (usado para cálculos de margem)
- **retail\_setup\_cost** - Taxa de ativação ou configuração única cobrada ao cliente
- **wholesale\_setup\_cost** - Seu custo único para configurar o serviço
- **tax\_percentage** - Porcentagem de imposto aplicada a este produto (por exemplo, 10 para 10%, 12,5 para 12,5%). Defina como 0 para produtos isentos de impostos. Essa taxa de imposto é aplicada automaticamente a transações criadas a partir deste produto.

## Configuração de Imposto do Produto

### Aplicação de Imposto:

Quando uma transação é criada a partir deste produto, a porcentagem de imposto é copiada automaticamente para a transação e o valor do imposto é calculado. Por exemplo:

- Produto com 10% de imposto, \$50,00 custo de varejo → Transação tem \$5,00 de imposto
- Produto com 0% de imposto (isenção de imposto) → Transação tem \$0,00 de imposto
- Sobrescrita de transação manual → Funcionários podem alterar a porcentagem de imposto por transação

### Visibilidade e Acesso do Cliente

- **enabled** - Se este produto está ativo e disponível para compra (defina como falso para ocultar sem excluir)
- **residential** - Se clientes residenciais (consumidores) podem comprar este produto
- **business** - Se clientes empresariais (comerciais) podem comprar este produto
- **customer\_can\_purchase** - Se os clientes podem se auto-adquirir via o portal (verdadeiro) ou se apenas funcionários podem adicioná-lo (falso)
- **available\_from** - Data/hora quando este produto se torna disponível para compra (opcional)
- **available\_until** - Data/hora quando este produto não está mais disponível para compra (opcional, útil para ofertas por tempo limitado)

### Contrato e Renovação

- **contract\_days** - Duração mínima do contrato em dias (por exemplo, 30 para mensal, 365 para anual, 0 para sem contrato mínimo)
- **auto\_renew** - Comportamento padrão de renovação:
  - **prompt** - Pergunta ao cliente toda vez se deseja renovar
  - **true** - Renova automaticamente sem perguntar
  - **false** - Requer renovação manual
- **allow\_auto\_renew** - Se os clientes podem habilitar a renovação automática (defina como falso para compras únicas)

### Conteúdo Voltado para o Cliente

- **terms** - Termos e condições exibidos aos clientes antes da compra (inclua limitações, regras de expiração, condições de uso)
- **features\_list** - Lista de recursos e inclusões mostradas aos clientes (formato de lista Python: ['Feature 1', 'Feature 2'])

### Configuração de Provisionamento

- **provisioning\_play** - Nome do playbook Ansible que provisiona este serviço (sem extensão .yaml). Deve existir em OmniCRM-API/Provisioners/plays/.
- **provisioning\_json\_vars** - Variáveis padrão passadas para o playbook Ansible como JSON. Estas podem ser sobrescritas ao provisionar. O playbook recebe estas junto com `customer_id`, `product_id`, `service_id` e `access_token`.
- **inventory\_items\_list** - Lista de itens de inventário necessários para este produto (por exemplo, ['SIM Card', 'Mobile Number']). Quando um cliente faz um pedido, eles serão solicitados a selecionar itens específicos do inventário disponível. IDs de inventário selecionados são passados para o playbook de provisionamento com o tipo de inventário como o nome da variável.
- **relies\_on\_list** - Lista de IDs de produtos ou tipos de serviço que devem existir antes que este produto possa ser adicionado. Usado para dependências de complementos (por exemplo, complementos móveis requerem um serviço móvel ativo).

## Metadados do Sistema

- **created** - Timestamp quando o produto foi criado (definido automaticamente)
- **last\_modified** - Timestamp quando o produto foi atualizado pela última vez (atualizado automaticamente)

## Exemplos de Definições de Produto

### Produto Autônomo (SIM Móvel)

Definições de Produto OmniCRM

```
{
  "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 produto autônomo requer dois itens de inventário (Cartão SIM e Número Móvel) e cria um novo serviço quando provisionado.

### Produto Adicional (Plano de Dados Mensal)

```
{
  "product_slug": "norfone-mobile-prepaid-mini",
  "product_name": "Norfone 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 Norfone 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 produto adicional não requer inventário e é aplicado a um serviço existente. Ele cobra o cliente e adiciona créditos/saldos ao seu serviço.

### Produto Agrupado (Pacote para Idosos)

```

{
  "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 produto agrupado provisiona múltiplos serviços (Internet + IPTV + Telefone) usando um único playbook. Ele requer um item de inventário (Modem Roteador).

### Produto Adicional (Recarga Simples)

```

{
  "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 simplesmente adiciona crédito monetário a um serviço existente. Nenhum inventário é necessário, e ele usa `service_id` para identificar qual serviço deve ser recarregado.

## Como as Variáveis São Passadas para o Ansible

Entender como as variáveis fluem da definição do produto através da API para o playbook Ansible é crítico para escrever playbooks de provisionamento eficazes.

### Fontes de Variáveis e Mesclagem

Quando um trabalho de provisionamento é criado, as variáveis vêm de várias fontes e são mescladas nesta ordem (fontes posteriores substituem as anteriores):

1. **Variáveis de provisioning\_json\_vars do Produto** - Variáveis padrão da definição do produto
2. **Corpo da Requisição** - Variáveis passadas na chamada da API (podem substituir os padrões do produto)
3. **Variáveis adicionadas pelo sistema** - Adicionadas automaticamente pelo sistema de provisionamento
4. **Seleções de Inventário** - IDs de itens de inventário selecionados (se `inventory_items_list` não estiver vazio)

### Processo de Mesclagem de Variáveis

O sistema mescla variáveis de todas as fontes, com fontes posteriores substituindo as anteriores. Isso permite uma personalização flexível no momento do provisionamento.

Por exemplo, se seu produto tiver:

```
"provisioning_json_vars": "{\"monthly_cost\": 50, \"data_gb\": 100}"
```

E sua requisição API incluir:

```

{
  "product_id": 10,
  "customer_id": 456,
  "monthly_cost": 45,
  "custom_param": "value"
}

```

As variáveis finais `extra_vars` passadas para o Ansible serão:

```

{
  "monthly_cost": 45,      # Sobrescrito da requisição
  "data_gb": 100,         # Da provisioning_json_vars
  "product_id": 10,       # Da requisição
  "customer_id": 456,     # Da requisição

```

```
"custom_param": "value",    # Da requisição
"access_token": "eyJ..."  # Adicionado pelo sistema
}
```

## Variáveis Adicionadas pelo Sistema

O sistema de provisionamento adiciona automaticamente:

- `access_token` - Token JWT para autenticar chamadas de API de volta ao CRM (de `g.access_token` para autenticação IP/API key, ou gerado a partir de `refresh_token` para autenticação de usuário)
- `initiating_user` - O ID do usuário que acionou o provisionamento (ou primeiro administrador para sistemas automatizados)
- Quaisquer campos do corpo da requisição (`product_id`, `customer_id`, `service_id`, etc.)

## Variáveis de Inventário

Quando um produto requer itens de inventário (por exemplo, `inventory_items_list: ["SIM Card', 'Mobile Number']`"), o processo funciona da seguinte maneira:

1. **UI/API solicita seleção** - O usuário seleciona itens de inventário específicos do estoque disponível
2. **IDs de Inventário são adicionados às variáveis** - Os IDs dos itens de inventário selecionados são adicionados com o tipo de inventário como o nome da variável
3. **O playbook acessa os IDs de inventário** - O playbook de provisionamento pode então recuperar os detalhes completos do inventário da API do CRM

Por exemplo, se um usuário selecionar: - Cartão SIM com `inventory_id: 789` - Número Móvel com `inventory_id: 101`

As variáveis passadas para o playbook incluem: - SIM Card: 789 - Mobile Number: 101

O playbook pode então usar esses IDs para buscar os registros completos de inventário (ICCID, IMSI, MSISDN, etc.) da API do CRM e usar essas informações para provisionar o serviço no equipamento de rede.

## Como o Ansible Recebe Variáveis

O sistema de provisionamento passa todas as variáveis mescladas para o playbook Ansible como extravars. Dentro do playbook, essas variáveis estão disponíveis através do sistema padrão de variáveis do Ansible e podem ser usadas em tarefas.

As variáveis podem ser referenciadas diretamente nas tarefas do playbook usando a sintaxe `{{ variable_name }}`. Por exemplo, `{{ product_id }}`, `{{ customer_id }}`, `{{ monthly_cost }}`, etc.

## Variáveis Passadas para Produtos Adicionais

Quando um produto adicional é provisionado, o sistema passa automaticamente:

- `product_id` - O ID do produto adicional sendo provisionado
- `customer_id` - O cliente que possui o serviço
- `service_id` - O ID do serviço ao qual este complemento está sendo adicionado (crítico para complementos)
- `access_token` - Token de autenticação para chamadas de API
- Quaisquer variáveis de `provisioning_json_vars`
- Quaisquer variáveis adicionais da requisição API

## Exemplo de Fluxo de Provisionamento de Complemento

Quando um cliente adiciona o complemento "Recarga de £5" ao seu serviço móvel (service\_id: 123), o playbook recebe variáveis incluindo:

- product\_id: 45 (o produto de recarga)
- customer\_id: 456 (o cliente)
- service\_id: 123 (o serviço para o qual adicionar crédito)
- access\_token: Token de autenticação
- Além de quaisquer variáveis do provisioning\_json\_vars do produto

O playbook então usa essas variáveis para:

1. **Buscar detalhes do serviço** da API do CRM usando o service\_id
2. **Extrair o UUID do serviço** e outras informações do registro do serviço
3. **Adicionar crédito ao sistema de faturamento** (OCS) usando o UUID do serviço
4. **Registrar a transação** no CRM para fins de faturamento

Esse fluxo permite que o complemento identifique exatamente qual serviço modificar e aplique as alterações adequadamente.

## Diferença Entre Variáveis de Produtos Autônomos e Adicionais

**Produtos Autônomos** recebem:

- product\_id - O produto sendo provisionado
- customer\_id - O cliente que está pedindo o serviço
- IDs de itens de inventário (por exemplo, SIM Card, Mobile Number) se o produto os requer
- access\_token - Para autenticação de API

**Produtos Adicionais** recebem:

- product\_id - O produto adicional sendo provisionado
- customer\_id - O cliente que possui o serviço
- service\_id - **O ID do serviço existente a ser modificado** (esta é a diferença chave)
- access\_token - Para autenticação de API

A diferença chave é service\_id - isso informa ao playbook qual serviço existente modificar ou adicionar a.

## Produtos Agrupados

Os produtos agrupados são provisionados como complementos, mas seu playbook pode criar múltiplos registros de serviço. Eles recebem as mesmas variáveis que complementos, incluindo:

- product\_id - O produto agrupado
- customer\_id - O cliente
- service\_id - Serviço pai (se aplicável)
- IDs de itens de inventário (por exemplo, Modem Router) se necessário
- access\_token - Para autenticação de API

O playbook de agrupamento (por exemplo, play\_seniors\_package) então cria múltiplos serviços relacionados (Internet, IPTV, Telefone) e os vincula.

## Serviços

Um serviço é uma instância de um produto que pertence a um cliente, pelo qual ele é cobrado.

É essencialmente um link para uma conta OCS </glossary> (Sistema de Cobrança Online) que lida com a geração de cobranças e os saldos e usos reais da conta. O OCS é alimentado por CGRateS e gerencia saldos monetários, saldos unitários (dados, voz, SMS), ActionPlans para renovação automática e ThresholdS para limites de gastos.

## Adicionando um Serviço: Seleção e Filtragem de Produtos

Ao adicionar um serviço a um cliente (seja um novo serviço autônomo ou um complemento a um serviço existente), o sistema exibe produtos disponíveis em uma interface de carrossel. Os produtos mostrados são filtrados com base em vários critérios:

### Filtragem de Produtos para Serviços Autônomos

Ao criar um novo serviço para um cliente, a interface do usuário exibe produtos filtrados por:

1. **Tipo de Cliente** - Produtos são categorizados como:
  - **Individual (Residencial)**: Produtos onde `residential = true` ou `business = false`
  - **Empresarial**: Produtos onde `business = true`
2. **Categoria** - Produtos são separados em:
  - **Planos de Serviço**: Produtos com `category = standalone` ou `bundle`
  - **Complementos**: Produtos com `category = addon` (exibidos em carrossel separado)
3. **Disponibilidade** - Produtos são mostrados apenas se:
  - `enabled = true` - Produto está ativo e não desativado
  - A data atual está entre `available_from` e `available_until` - Produto está dentro de sua janela de disponibilidade
  - `customer_can_purchase = true` (se o cliente estiver se auto-adquirindo) - Produto permite compra direta pelo cliente

Nota

**Filtragem em Nível de API:** A API filtra automaticamente produtos por status habilitado e datas de disponibilidade em dois níveis:

- **Endpoints de Compra/Seleção** (`/crm/product/`) - Usado pelo modal de Complementos e PlanList para seleção de produtos. Filtra automaticamente para mostrar APENAS produtos habilitados dentro de seu intervalo de datas de disponibilidade. Isso garante que clientes e funcionários só possam selecionar produtos que estão atualmente disponíveis para compra.
- **Endpoints de Gerenciamento** (`/crm/product/paginated`) - Usado pela página de Gerenciamento de Produtos. Mostra TODOS os produtos, incluindo desativados e fora das datas de disponibilidade, permitindo que administradores gerenciem o catálogo completo de produtos, incluindo produtos inativos.

Passa `include_disabled=true` para o endpoint base de produtos para ignorar a filtragem (apenas para uso administrativo).

A interface do usuário exibe carrosséis separados para:

- **Planos de Serviço Individuais** - Produtos residenciais para clientes consumidores
- **Planos de Serviço Empresariais** - Produtos comerciais para clientes empresariais
- **Complementos Individuais** - Pacotes de complementos residenciais
- **Complementos Empresariais** - Pacotes de complementos comerciais

### Filtragem de Produtos para Serviços Adicionais

Ao adicionar um complemento a um **serviço existente**, filtrações adicionais são aplicadas:



1. **Correspondência de Tipo de Serviço** - Apenas complementos com `service_type` correspondente são mostrados:
  - Se o serviço existente tem `service_type = "mobile"`, apenas complementos com `service_type = "mobile"` são exibidos
  - Isso garante que clientes móveis vejam apenas complementos móveis, clientes de internet vejam apenas complementos de internet, etc.
2. **Verificação de Dependências** - Se um complemento tem uma `relies_on_list`:
  - O sistema verifica se o cliente possui os produtos/serviços necessários
  - Apenas complementos cujas dependências estão satisfeitas são mostrados
3. **Filtro de Mesmo Tipo de Cliente** - Os complementos ainda são filtrados por `residential` vs `business` para corresponder ao tipo de cliente

## Exemplo de Cenário de Filtragem

Para um cliente empresarial com um serviço móvel existente (`service_type = "mobile"`):

- **Produtos Autônomos Exibidos:** Todos os produtos autônomos/agrupados empresariais (`business = true`, `category != "addon"`)
- **Produtos Adicionais Exibidos:** Apenas complementos móveis empresariais (`business = true`, `category = "addon"`, `service_type = "mobile"`)
- **Produtos Ocultos:** Produtos residenciais, complementos para outros tipos de serviço (internet, voz, etc.), produtos desativados

## Campos de Serviço

O modelo de Serviço contém campos que rastreiam a instância de serviço provisionada e sua relação com o cliente, produto e sistema de faturamento.

### Informações Básicas do Serviço

- **service\_id** - Identificador único atribuído automaticamente pelo sistema (somente leitura)
- **customer\_id** - Link para o cliente que possui este serviço (somente leitura após a criação)
- **product\_id** - Link para o produto do qual este serviço foi criado (somente leitura após a criação)
- **service\_name** - Nome exibido mostrado aos clientes (editável)
- **service\_type** - Tipo de serviço: móvel, internet, voip, iptv, pacote, etc. (editável)
- **service\_uuid** - Identificador único usado no OCS/CGRateS para faturamento (somente leitura, gerado automaticamente)
- **icon** - Classe de ícone FontAwesome para exibição no portal de autoatendimento (editável)

### Status e Datas do Serviço

- **service\_status** - Status atual: Ativo, Inativo, Suspenso, etc. (editável)
- **service\_provisioned\_date** - Quando o serviço foi provisionado pela primeira vez (definido automaticamente, somente leitura)
- **service\_active\_date** - Quando o serviço se tornou ativo (editável)
- **service\_deactivate\_date** - Quando o serviço expira ou será desativado (editável)
- **contract\_end\_date** - Data de término do compromisso do contrato (editável)

### Faturamento e Preços

- **retail\_cost** - Cobrança mensal recorrente ao cliente (editável)
- **wholesale\_cost** - Seu custo para fornecer o serviço (editável)
- **service\_billed** - Se este serviço aparece nas faturas (editável, padrão: verdadeiro)
- **service\_taxable** - Se impostos se aplicam a este serviço (editável, padrão: verdadeiro)
- **invoiced** - Se o serviço foi faturado (definido automaticamente pelo sistema de faturamento)

- **promo\_code** - Código promocional usado quando o serviço foi criado (editável)

## Visibilidade do Cliente

- **service\_visible\_to\_customer** - Se o cliente pode ver este serviço no portal de autoatendimento (editável, padrão: verdadeiro)
- **service\_usage\_visible\_to\_customer** - Se o cliente pode visualizar detalhes de uso/saldo (editável, padrão: verdadeiro)

## Configuração de Provisionamento

- **provisioning\_play** - Playbook Ansible usado para provisionar este serviço (herdado do produto, somente leitura)
- **provisioning\_json\_vars** - Variáveis passadas para o playbook de provisionamento (herdadas do produto, somente leitura)
- **deprovisioning\_play** - Playbook Ansible a ser executado quando o serviço é desprovisionado (somente leitura)
- **deprovisioning\_json\_vars** - Variáveis para o playbook de desprovisionamento (somente leitura)

## Relações de Serviço

- **bundled\_parent** - Se este serviço faz parte de um pacote, o service\_id do serviço pai (somente leitura)
- **site\_id** - Link para o local físico onde o serviço é entregue (editável)

## Notas e Metadados

- **service\_notes** - Notas internas sobre o serviço para referência da equipe (editável)
- **created** - Timestamp quando o serviço foi criado (definido automaticamente, somente leitura)
- **last\_modified** - Timestamp da última atualização (atualizado automaticamente, somente leitura)

## Campos Editáveis vs Somente Leitura

### Editáveis via API/UI:

Os serviços podem ser atualizados via PATCH /crm/service/{service\_id} com estes 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

### Somente Leitura (Definidos Automaticamente):

Esses campos não podem ser modificados diretamente após a criação:

- service\_id, customer\_id, product\_id
- service\_uuid (gerado durante o provisionamento)
- service\_provisioned\_date
- provisioning\_play, provisioning\_json\_vars
- deprovisioning\_play, deprovisioning\_json\_vars
- bundled\_parent

- invoiced (gerenciado pelo sistema de faturamento)
- created, last\_modified (gerenciado automaticamente)

## Provisionando Produtos em Serviços

O processo de provisionamento converte um Produto (modelo) em um Serviço (instância específica do cliente) através de uma série de etapas coordenadas envolvendo a UI Web, API e playbooks Ansible.

### Fluxo de Provisionamento de Alto Nível

1. **Configuração Pré-Provisionamento** - Produto criado na API com configuração de provisionamento, e playbooks Ansible correspondentes escritos e testados
2. **Seleção de Serviço** - A partir da Página do Cliente, funcionários ou clientes selecionam "Adicionar Serviço"
3. **Filtragem de Produtos** - Produtos exibidos filtrados com base em:
  - Tipo de cliente (residencial/empresarial)
  - Serviços existentes (para dependências de complementos em `relies_on_list`)
  - Datas de disponibilidade (`available_from/available_until`)
  - Flags `enabled` e `customer_can_purchase`
4. **Personalização** - Opção de sobrescrever variáveis de provisionamento (para ajustes de preço, configurações personalizadas, etc.)
5. **Seleção de Inventário** - Se o produto requer inventário (`inventory_items_list` não está vazio), o usuário seleciona itens específicos (por exemplo, qual cartão SIM, qual número de telefone)
6. **Início do Provisionamento** - Quando o botão "Provisionar" é clicado, a API cria um trabalho de provisionamento

### Fluxo Detalhado de Integração da API e Ansible

Quando um serviço é provisionado, a seguinte sequência ocorre:

#### Passo 1: Criação do Trabalho de Provisionamento (`/routes/service.py`)

A API recebe a requisição de provisionamento e chama `create_provisioning_job()` de `services/provisioning_service.py` com:

- `provisioning_play` - Nome do playbook Ansible (por exemplo, `play_psim_only`)
- `provisioning_json_vars` - String JSON de variáveis do produto ou sobrescritas pela requisição
- `customer_id` - ID do cliente que está pedindo o serviço
- `product_id` - ID do produto sendo provisionado
- `service_id` - (Opcional) ID do serviço existente para complementos
- Seleções de Inventário - IDs dos itens de inventário selecionados

#### Passo 2: Montagem de Variáveis (`services/provisioning_service.py`)

O serviço de provisionamento mescla variáveis de várias fontes nesta ordem:

1. `provisioning_json_vars` do Produto (padrões da definição do produto)
2. Parâmetros do corpo da requisição (podem substituir os padrões do produto)
3. Variáveis adicionadas pelo sistema:
  - `access_token` - Token JWT para autenticação da API de volta ao CRM
  - `initiating_user` - ID do usuário que acionou o provisionamento
  - `customer_id`, `product_id`, `service_id`
4. Seleções de Inventário - Adicionadas como pares `{inventory_type: inventory_id}`

Exemplo de variáveis mescladas:

```
{
  "customer_id": 123,
  "product_id": 456,
  "service_id": 789,           # Somente para complementos
  "SIM Card": 1001,           # Da seleção de inventário
  "Mobile Number": 1002,      # Da seleção de inventário
  "monthly_cost": 30,         # Da provisioning_json_vars
  "data_gb": 50,              # Da provisioning_json_vars
  "access_token": "eyJ...",   # Adicionado pelo sistema para callbacks da API
  "initiating_user": 5        # Adicionado pelo sistema
}
```

### Passo 3: Criação do Registro de Provisionamento (models.py - Modelo de Provisionamento)

Um registro de Provision é criado no banco de dados com:


- provision\_id - Identificador único para rastreamento
- provisioning\_play - Nome do arquivo do playbook
- provisioning\_json\_vars - Variáveis mescladas como string JSON
- task\_count - Número de tarefas no playbook (extraído do YAML)
- provisioning\_status - Código de status (inicialmente definido como 1 = em execução, depois atualizado para 0 = sucesso, 2 = falha, ou pode permanecer 1 se ainda estiver em progresso)
- product\_id, customer\_id, service\_id - Referências de contexto

### Passo 4: Execução do Playbook em Segundo Plano (Provisioners/playbook\_runner\_v2.py)

A API inicia uma thread em segundo plano que:

1. Carrega o YAML do playbook de OmniCRM-API/Provisioners/plays/{playbook\_name}.yaml
2. Chama `ansible_runner.run()` com:
  - `playbook` - Caminho para o arquivo YAML carregado
  - `extravars` - Todas as variáveis mescladas (passadas para o Ansible)
  - `inventory` - Definido como 'localhost,' (execução local)
  - `event_handler` - Manipulador personalizado para capturar eventos de execução de tarefas
3. Monitora a execução do playbook em tempo real

### Passo 5: Captura e Registro de Eventos (ProvisioningEventHandler)

À medida que cada tarefa Ansible  executada, eventos são capturados e armazenados como registros `Provision_Event`:

- event\_name - Nome da tarefa do playbook
- event\_number - Número da sequência
- provisioning\_status - Código de status indicando o resultado da tarefa:
  - **0** = Sucesso - Tarefa concluída com sucesso
  - **1** = Em Execução - Tarefa está atualmente em execução
  - **2** = Falha - Falha crítica que interrompe o provisionamento
  - **3** = Falha (ignorada) - Tarefa falhou, mas erros foram ignorados (`ignore_errors: true` no playbook)
- provisioning\_result\_json - Resultados da tarefa com dados sensíveis redigidos

O manipulador de eventos automaticamente remove senhas, chaves, segredos e outros dados sensíveis dos logs.

## Passo 6: Execução do Playbook Ansible (Provisioners/plays/\*.yaml)

O playbook Ansible é executado localmente e normalmente realiza as seguintes ações:

1. **Buscar Definição do Produto** - Requisição GET para /crm/product/product\_id/{{ product\_id }} usando {{ access\_token }}
2. **Buscar Informações do Cliente** - Requisição GET para /crm/customer/customer\_id/{{ customer\_id }}
3. **Processar Itens de Inventário** (se necessário) - Requisição GET para /crm/inventory/inventory\_id/{{ inventory\_id }} para cada item selecionado para recuperar detalhes completos (ICCID, MSISDN, números de série, etc.)
4. **Configurar Sistemas Externos** - Fazer chamadas de API para:
  - HSS (Home Subscriber Server) para provisionamento de assinantes
  - IMS (IP Multimedia Subsystem) para registro de voz
  - CGRateS/OCS para criação de conta, configuração de cobrança, planos de tarifas
  - Servidores ENUM para mapeamento de números de telefone
  - Equipamento de rede (roteadores, switches, etc.)
5. **Adicionar Custos de Configuração** (se aplicável) - POST para /crm/transaction/ para registrar cobranças únicas
6. **Cobrar o Cliente** - POST para OCS/CGRateS para cobrar retail\_setup\_cost se configurado
7. **Criar Conta OCS** - POST para OCS/CGRateS para criar conta de faturamento com UUID
8. **Configurar Cobranças Recorrentes** - Criar Ações e ActionPlans no OCS/CGRateS para cobranças mensais recorrentes
9. **Criar Registro de Serviço** - PUT/POST para /crm/service/ para criar o registro de serviço no 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. **Atribuir Inventário** - PATCH para /crm/inventory/inventory\_id/{{ inventory\_id }} para marcar o inventário como "Atribuído" ao serviço
11. **Enviar Notificações** (opcional) - Email ou SMS para o cliente com detalhes do serviço

## Passo 7: Conclusão e Atualização de Status

Quando o playbook é concluído:

- **Sucesso:** Provision.provisioning\_status atualizado para **0** (Sucesso)

- **Falha Crítica:** Provision.provisioning\_status atualizado para **2** (Falhou), e email de falha enviado para crm\_config.provisioning.failure\_list
- **Falhas Não Críticas:** Tarefas que falham com ignore\_errors: true são marcadas com status **3** (Falhou, mas ignorado) e não interrompem o provisionamento

O serviço provisionado agora é visível no CRM e ativo para o cliente (se o provisionamento foi bem-sucedido).

## Principais Diferenças: Provisionamento Autônomo vs Adicional vs Agrupado

### Produtos Autônomos (category: standalone):

- Recebem customer\_id e product\_id
- Tipicamente requerem itens de inventário (cartões SIM, números de telefone, modems)
- Criam um registro de serviço **novo** via API PUT /crm/service/
- Provisionam novos recursos em sistemas externos (HSS, OCS, equipamento de rede)
- Exemplo: Ativação de novo SIM móvel, nova conexão de internet

### Produtos Adicionais (category: addon):

- Recebem customer\_id, product\_id e **service\_id** (serviço existente a ser modificado)
- Tipicamente NÃO requerem inventário (ou inventário mínimo)
- **Modificam um serviço existente** ou adicionam cobranças à conta OCS existente
- Podem executar ações no OCS (adicionar pacote de dados, adicionar crédito, habilitar recurso)
- Não criam novos registros de serviço (ou criam registros de serviço filhos vinculados ao pai)
- Exemplo: Recarga de plano de dados mensal, pacote de roaming internacional, crédito extra

### Produtos Agrupados (category: bundle):

- Semelhantes a complementos em termos de variáveis recebidas
- Podem requerer alguns itens de inventário (por exemplo, modem para pacote residencial)
- Criam **múltiplos** registros de serviço relacionados (Internet + TV + Telefone)
- Provisionam múltiplos recursos em diferentes sistemas
- Vinculam serviços no CRM para faturamento/gerenciamento unificado
- Exemplo: Pacote residencial (Internet + IPTV + telefone VoIP)

## Requisitos do Playbook de Provisionamento

Para que um playbook funcione corretamente, ele deve:

1. **Estar localizado em** OmniCRM-API/Provisioners/plays/{playbook\_name}.yaml
2. **Aceitar variáveis** via extravars do Ansible (acessadas como {{ variable\_name }})
3. **Autenticar chamadas de API** usando Authorization: Bearer {{ access\_token }} header
4. **Lidar com falhas de forma adequada** usando blocos rescue e ignore\_errors onde apropriado
5. **Criar registro de serviço** para produtos autônomos, ou modificar serviço existente para complementos
6. **Atribuir inventário** se itens de inventário foram selecionados
7. **Retornar mensagens de erro significativas** via módulo fail quando erros críticos ocorrerem

## Variáveis Comuns Disponíveis em Playbooks

Todo playbook recebe essas variáveis:

- `customer_id` - Inteiro, cliente que está pedindo o serviço
- `product_id` - Inteiro, produto sendo provisionado
- `service_id` - Inteiro (somente complementos/agrupados), serviço existente a modificar
- `access_token` - String, token JWT para autenticação da API do CRM
- `initiating_user` - Inteiro, usuário que acionou o provisionamento
- Além de quaisquer IDs de itens de inventário: `{{ inventory_type }}`: `inventory_id`
- Além de quaisquer variáveis de `provisioning_json_vars`
- Além de quaisquer variáveis passadas na requisição de provisionamento

Os playbooks podem usar essas para:

- Buscar detalhes completos do produto: `GET /crm/product/product_id/{{ product_id }}`
- Buscar detalhes do cliente: `GET /crm/customer/customer_id/{{ customer_id }}`
- Buscar detalhes do inventário: `GET /crm/inventory/inventory_id/{{ SIM_Card }}`
- Criar transações: `POST /crm/transaction/`
- Criar serviços: `PUT /crm/service/`
- Atualizar serviços: `PATCH /crm/service/{{ service_id }}`
- Atribuir inventário: `PATCH /crm/inventory/inventory_id/{{ inventory_id }}`

## Exemplo: Fluxo Simples de Playbook de Complemento

Para um complemento de recarga de dados móveis:

1. O playbook recebe: `customer_id`, `product_id`, `service_id`, `access_token`
2. Buscar detalhes do serviço: `GET /crm/service/{{ service_id }}` para obter `service_uuid`
3. Buscar detalhes do produto: `GET /crm/product/product_id/{{ product_id }}` para obter preços e quantidade de dados
4. Cobrar o cliente no OCS: `POST` para `CGRateS` para deduzir `retail_cost` do saldo
5. Adicionar crédito de dados no OCS: `POST` para `CGRateS` para adicionar saldo de dados com expiração
6. Registrar transação no CRM: `POST /crm/transaction/` com detalhes da cobrança
7. Completar com sucesso

Todo o processo é rastreado nas tabelas `Provision` e `Provision_Event` para fins de depuração e auditoria.

## Envolvimento do OCS

OCS (Sistema de Cobrança Online), implementado via `CGRateS`, lida com toda cobrança em tempo real e rastreamento de uso para serviços. O registro de serviço do CRM atua como um ponteiro para a conta OCS, que gerencia:

- **Cobranças recorrentes** - Taxas mensais, aluguel de DID, cobranças de assinatura
- **Cobrança baseada em uso** - Chamadas de voz por minuto, dados por MB, cobranças por SMS
- **Gerenciamento de saldo** - Saldos monetários (crédito pré-pago) e saldos unitários (dados GB, minutos de voz, contagem de SMS)
- **Conversões de saldo** - Convertendo saldos monetários em saldos unitários (por exemplo, gastar \$30 para obter pacote de dados de 10GB)
- **Estado da conta** - Ativo, suspenso, desativado com base em limites de crédito e thresholds

O registro de serviço do CRM contém metadados e configuração (cliente, produto, preços, visibilidade), enquanto o OCS contém o estado de faturamento ao vivo (saldos, uso, cobranças).

## Recuperando Uso e Saldos do Serviço

As informações de uso do serviço são recuperadas do OCS/CGRateS e exibidas para clientes e funcionários em tempo real.

### Como o Uso é Recuperado

Quando o uso de um serviço é solicitado (via UI ou API), o seguinte fluxo ocorre:

1. **Requisição API** - O frontend chama GET /crm/service/{service\_id} ou visualiza detalhes do serviço na interface do usuário
2. **Busca de Serviço** - A API recupera o registro do serviço do banco de dados, extrai service\_uuid
3. **Chamadas da API CGRateS** - O módulo cgrates\_service.py faz duas chamadas para CGRateS:
  1. **Get\_Balance(service\_uuid)** - Recupera saldo da conta com BalanceMap
    - Retorna saldos organizados por tipo: DADOS, VOZ, SMS, MONETÁRIO, DADOS\_DONGLE
    - Cada saldo inclui: ID, Valor, Data de Expiração, Peso, DestinationIDs
    - O sistema adiciona campos legíveis por humanos: custom\_Name\_hr, custom\_Expiration, custom\_Description\_String
  2. **Get\_ActionPlans(service\_uuid)** - Recupera planos de ação de renovação automática ativos (cobertos na próxima seção)
4. **Mesclagem de Resposta** - Os dados do CGRateS são mesclados na resposta do serviço:

```
{
  "service_id": 123,
  "service_name": "Serviço Móvel",
  "service_uuid": "abc-123-def",
  "cgrates": {
    "BalanceMap": {
      "DATA": [{
        "ID": "DATA_10GB",
        "Value": 5368709120,
        "ExpirationDate": "2025-02-01T00:00:00Z",
        "custom_Name_hr": "Pacote de Dados de 10GB",
        "custom_Expiration": "1 de Fevereiro de 2025",
        "custom_Description_String": "5 GB restantes"
      }],
      "VOICE": [{
        "ID": "VOICE_UNLIMITED",
        "Value": 999999999,
        "custom_Name_hr": "Chamadas Ilimitadas",
        "custom_Description_String": "Minutos ilimitados"
      }],
      "MONETARY": [{
        "ID": "PREPAID_CREDIT",
        "Value": 25.50,
        "custom_Description_String": "Crédito de \"$25.50"
      }]
    }
  }
}
```



```

    },
    "ActionPlans": [...]
  }
}

```

## 5. **Exibição na UI** - Componentes frontend exibem os dados de uso:

- **ServiceUsage.js** - Componente principal de exibição de uso com atualização automática a cada 3 segundos
- **UsageCard.js** - Cartões de resumo para cada tipo de saldo
- **UsageProgress.js** - Barras de progresso mostrando porcentagem usada/restante
- Saldos são codificados por cores e formatados para legibilidade

## **Estrutura de Dados de Uso**

Cada saldo no BalanceMap contém:

### **Campos Nativos do CGRateS:**

- ID - Identificador único para o saldo (por exemplo, "DATA\_10GB\_2025\_01")
- Value - Montante do saldo:
  - Para DADOS: bytes (5368709120 = 5 GB)
  - Para VOZ: segundos (3600 = 1 hora)
  - Para SMS: contagem (100 = 100 mensagens)
  - Para MONETÁRIO: unidades de moeda (25.50 = \$25.50)
- ExpirationDate - Timestamp ISO 8601 quando o saldo expira
- Weight - Prioridade para consumo de saldo (peso mais alto consumido primeiro)
- DestinationIDs - Destinos aos quais este saldo se aplica (por exemplo, ["AU", "INTERNATIONAL"])

### **Campos Legíveis por Humanos (adicionados pelo CRM):**

- custom\_Name\_hr - Nome legível por humanos extraído do ID
- custom\_Expiration - Data de expiração formatada (por exemplo, "15 de Jan, 2025" ou "em 11 dias")
- custom\_Description\_String - Descrição do saldo legível por humanos:
  - DADOS: "5 GB restantes" ou "10 GB totais"
  - VOZ: "60 minutos restantes" ou "ilimitado"
  - SMS: "50 SMS restantes"
  - MONETÁRIO: "\$25.50 de crédito"

## **Controle de Visibilidade de Uso**

A visibilidade do uso do serviço é controlada por dois campos:

- **service\_visible\_to\_customer** - Se falso, o serviço é ocultado totalmente do portal de autoatendimento do cliente
- **service\_usage\_visible\_to\_customer** - Se falso, o serviço é visível mas detalhes de uso/saldo estão ocultos (o cliente pode ver que possui o serviço, mas não quanto usou)

Isso permite que operadores:

- Oculte serviços internos/testes dos clientes
- Mostre que o serviço existe sem revelar o uso (útil para planos ilimitados ou serviços sensíveis)
- Exibição de uso totalmente transparente (padrão)

## Atualizações de Uso em Tempo Real

A interface Web atualiza automaticamente os dados de uso:

- **Intervalo:** A cada 3 segundos (configurável no componente ServiceUsage)
- **Método:** Faz requisições GET /crm/service/{service\_id} que busca dados ao vivo do CGRateS
- **Eficiência:** Apenas visualizações de serviços ativos são atualizadas; visualizações de lista usam dados em cache

Isso garante que clientes e funcionários vejam atualizações de saldo quase em tempo real à medida que o uso ocorre.

## Cobranças Recorrentes / AutoRenovação

Cobranças recorrentes, como uma Cobrança Mensal de Serviço, ou uma Cobrança por DID são primeiro criadas como Ações dentro do OCS e seguem o formato Action\_ServiceUUID\_ServiceName\_WhatitDoes.

Por exemplo, para um serviço GPON de \$60 por mês que inclui 1TB de uso, a Ação seria algo como:

Action\_kj49-adsf-1234-9742\_60\_GPON\_1TB\_MonthlyExpiry

1. Redefinir Saldo Monetário para \$0
2. Enviar um POST HTTP para /simple\_provision no CRM para provisionar algo
3. Adicionar um Crédito para 1TB de Uso expirando em 1 mês

Se quisermos que a MRC seja recorrente (queremos), então criaríamos um ActionPlan chamado ActionPlan\_{{ service\_uuid }}\_Monthly\_Charge que teria o tempo definido para *mensal* para acionar todo mês, e atribuiríamos o ActionPlan à conta.

Podemos definir com base no parâmetro *Ano / Meses / Dias* uma data de expiração para quando a MRC irá parar também, por exemplo, para um serviço fixo de 12 meses que parou após esse ponto.

Como as Ações e ActionPlans são únicas para o serviço, elas não compartilham nada com outros serviços.

Isso significa que podemos provisioná-las com valores ajustados, e isso não impactará outros serviços.

## Complementos e Adições

Complementos / Adições como comprar dados extras, pacotes de roaming, minutos internacionais, etc., são tratados de maneira muito semelhante. Uma Ação é criada para fazer o que é necessário, como cobrar um valor monetário e depois conceder um saldo unitário com uma expiração definida.

Em vez de usar ActionPlans para adicionar isso automaticamente à conta, simplesmente acionamos ExecuteAction para a Ação que acabamos de criar uma vez a partir do Ansible.

## Adicionando Saldos Monetários Pré-Pagos

Para saldos monetários pré-pagos, como um plano PAYG de \$10, isso é adicionado como um saldo monetário, mas com uma prioridade maior.

O limite de crédito nesses serviços para o saldo padrão seria \$0.

## Limites de Crédito / Prevenindo Gastos Excessivos

Thresholds são usados em cada conta para definir o gasto máximo para um dado período de tempo.

Para clientes PAYG / Pré-pagos, isso é \$0.

## Interagindo com OCS via CRM

Para cada Serviço, você pode ver os Balances e ActionPlans, Actions e Thresholds do OCS a partir da API do CRM.

ActionPlans podem ser removidos conforme necessário a partir da API do CRM, acionados via Playbooks Ansible. ActionPlans podem ser adicionados conforme necessário, a partir do CRM, adicionando um Complemento/Serviço e acionados via Playbooks Ansible.

Contas OCS podem ser desativadas, o que impedirá que ActionPlans sejam executados e que serviços possam ser consumidos.

Para Limites de Crédito, um valor de Thresholds é definido de acordo com a política para o produto.

## Visualizando e Gerenciando ActionPlans no CRM

ActionPlans (configurações de renovação automática) são exibidos e gerenciados através da interface do CRM, permitindo que funcionários e clientes vejam renovações automáticas futuras e as gerenciem.

### Como ActionPlans São Recuperados e Exibidos

Ao visualizar um serviço no CRM, ActionPlans são automaticamente buscados e exibidos:

1. **Chamada API** - Quando GET /crm/service/{service\_id} é chamado, a API:
  - Recupera o registro do serviço do banco de dados
  - Extrai o service\_uuid (identificador da conta OCS)
  - Chama get\_cgrates\_action\_plans\_by\_service\_uuid(service\_uuid) de cgrates\_service.py
  - Isso internamente chama ocs.Get\_ActionPlans(service\_uuid) para buscar ActionPlans do CGRateS
2. **Estrutura de Dados do ActionPlan** - Cada ActionPlan retornado contém:

```
{
  "ActionPlanId": "ServiceID_abc-123-def__ProductID_456__...",
  "PlanName": "Plano de Renovação Mensal",
  "NextExecTime": "2025-02-01T00:00:00Z",
  "custom_NextExecTime_hr": "em 11 dias",
  "ActionPlanId_split_dict": {
    "ServiceID": "abc-123-def",
    "ProductID": 456,
    "CustomerID": 789,
    ...
  }
}
```

}

- **ActionPlanId** - Identificador único contendo informações codificadas sobre serviço/produto/cliente
- **PlanName** - Nome do plano de ação (tipicamente o nome do playbook de renovação)
- **NextExecTime** - Timestamp ISO quando o ActionPlan será executado a seguir
- **custom\_NextExecTime\_hr** - Tempo legível por humanos até a execução (por exemplo, "em 11 dias", "amanhã", "1 de Fevereiro de 2025")
- **ActionPlanId\_split\_dict** - Dicionário com componentes analisados do ActionPlanId

### 3. **Exibição na UI** - ActionPlans são mostrados no componente **ActionPlansTable**:

#### **Colunas da Tabela:**

- **Nome do Produto** - Recuperado buscando ProductID do ActionPlanId
- **Custo** - Mostra retail\_cost da definição do produto
- **Data de Renovação** - Exibe custom\_NextExecTime\_hr (legível por humanos)
- **Ações** - Dois botões:
  - **Renovar Agora** - Provisionar imediatamente o complemento/renovação (ignora a espera pela execução programada)
  - **Remover Renovação Automática** - Cancela a renovação automática

#### **Quando Nenhum ActionPlan Existe:**

- A tabela mostra a mensagem: "Nenhuma renovação automática habilitada para este serviço"
- O cliente pode adicionar complementos de renovação automática para habilitar a renovação automática

## **Gerenciando ActionPlans**

Funcionários e clientes podem gerenciar ActionPlans através da interface:

### **Removendo um ActionPlan (Cancelando Renovação Automática):**

1. Clique no botão "Remover Renovação Automática" na ActionPlansTable
2. Modal de confirmação aparece: "Você tem certeza de que deseja remover esta renovação automática?"
3. Ao confirmar, o frontend chama: `DELETE /crm/oam/remove_action_plan/{action_plan_id}`
4. A API remove o ActionPlan do CGRateS via `ocs.Remove_ActionPlan()`
5. A atividade é registrada: "Removed ActionPlan {ActionPlanId} from service {service\_id}"
6. ActionPlan desaparece da tabela

### **Renovando Imediatamente (Renovação Manual):**

1. Clique no botão "Renovar Agora" na ActionPlansTable
2. Modal de confirmação aparece: "Você tem certeza de que deseja renovar isso agora?"
3. Ao confirmar, o sistema:
  - Extrai product\_id do ActionPlanId
  - Cria um novo trabalho de provisionamento para esse produto
  - Provisiona o complemento imediatamente (executa o playbook de provisionamento)
  - O serviço recebe os benefícios do complemento sem esperar pela renovação programada
4. Modal de status de provisionamento mostra progresso
5. Em caso de sucesso, os saldos são imediatamente atualizados

## Adicionando Renovação Automática:

A renovação automática é habilitada provisionando um produto complementar que tem `auto_renew` definido:

- Produtos com `auto_renew = "true"` - Criam automaticamente ActionPlans durante o provisionamento
- Produtos com `auto_renew = "prompt"` - Perguntam ao cliente se ele deseja renovação automática (modal de diálogo)
- Produtos com `auto_renew = "false"` - Nunca criam ActionPlans (compra única)

O playbook de provisionamento cria o ActionPlan no CGRateS com:

- Identificador único ActionPlanId codificando serviço, produto e IDs de cliente
- Cronograma de renovação (mensal, anual, intervalo personalizado)
- Ação a ser executada (tipicamente reprovisionamento do mesmo complemento)
- Data de expiração (se o contrato tiver um prazo fixo)

## Convenção de Nomenclatura de ActionPlan

ActionPlans seguem uma convenção de nomenclatura padronizada para codificar metadados:

### Formato:

```
ServiceID_{service_uuid}__ProductID_{product_id}__CustomerID_{customer_id}__...
```

### Exemplo:

```
ServiceID_abc-123-def__ProductID_456__CustomerID_789__MonthlyRenewal
```

Essa codificação permite que o CRM:

- Identifique a qual serviço o ActionPlan pertence
- Busque detalhes do produto (nome, preços) para exibição
- Rastreie a propriedade do



# Sistema de Provisionamento

OmniCRM usa **Ansible** para automatizar o provisionamento, configuração e desprovisionamento de serviços ao cliente. O sistema de provisionamento é projetado para ser flexível, permitindo fluxos de trabalho complexos enquanto mantém consistência e confiabilidade.

## Eventos Recentes de Provisionamento

### Nota

Para um guia completo sobre a jornada do produto ao serviço com exemplos detalhados de playbooks Ansible, estratégias de preços e cenários do mundo real, veja Guia Completo do Ciclo de Vida do Produto <guide\_product\_lifecycle>.

## Visão Geral

Quando um produto é solicitado ou um serviço precisa ser configurado, o OmniCRM cria um **Trabalho de Provisionamento** que executa um ou mais playbooks Ansible. Esses playbooks interagem com vários sistemas de backend (OCS/CGRateS, equipamentos de rede, APIs, etc.) para provisionar completamente o serviço.

O sistema de provisionamento suporta dois fluxos de trabalho principais:

1. **Provisionamento Padrão** - Acionado por funcionários ou clientes através da UI/API
2. **Provisionamento Simples** - Acionado por sistemas externos como OCS para operações automatizadas

## Valores de Status de Provisionamento

Trabalhos de provisionamento e tarefas individuais podem ter os seguintes status:

- **Status 0 (Sucesso)** - O trabalho de provisionamento foi concluído com sucesso
- **Status 1 (Executando)** - O trabalho ou tarefa de provisionamento está atualmente em execução
- **Status 2 (Falhou - Crítico)** - Ocorreu uma falha crítica que causou a falha do provisionamento
- **Status 3 (Falhou - Ignorado)** - Uma tarefa falhou, mas tinha `ignore_errors: true`, então o provisionamento continuou

Quando um trabalho de provisionamento falha, o OmniCRM envia notificações por e-mail para a lista de notificação de falhas configurada com informações detalhadas sobre o erro.

## Como os Produtos Impulsionam o Provisionamento

A definição de **Produto** é o modelo do que é provisionado e como. Quando um usuário seleciona um produto para provisionar, o sistema lê vários campos-chave da definição do produto para determinar o que fazer.

### Campos do Produto Usados no Provisionamento

Uma definição de produto contém:

- `provisioning_play` - O nome do playbook Ansible a ser executado (sem a extensão `.yaml`)
- `provisioning_json_vars` - String JSON contendo variáveis padrão a serem passadas para o Ansible
- `inventory_items_list` - Lista de tipos de inventário que devem ser atribuídos (por exemplo, `['SIM Card', 'Mobile Number']`)
- `product_id`, `product_name`, campos de preços - Passados automaticamente para o playbook

### Exemplo de Definição de Produto

```
{
  "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
}
```

### Do Produto ao Trabalho de Provisionamento

Quando o provisionamento é iniciado, o sistema:

1. **Carrega o playbook** especificado em `provisioning_play`

O sistema procura por `OmniCRM-API/Provisioners/plays/play_psim_only.yaml`

## 2. **Mescla variáveis** de várias fontes em `extra_vars`:

1. **De provisioning\_json\_vars**: {"iccid": "", "msisdn": ""}
2. **Do corpo da requisição**: Quaisquer variáveis adicionais que o usuário/API fornecer
3. **Dos campos do produto**: `product_id`, `customer_id`, etc.
4. **Da autenticação**: `access_token` ou configuração para `refresh_token`

## 3. **Atribui inventário** com base em `inventory_items_list`

Antes de executar o playbook, a UI/API solicita a seleção de inventário:

- **SIM Card** - O usuário seleciona um SIM disponível do inventário
- **Mobile Number** - O usuário seleciona um número de telefone disponível

Os IDs de inventário selecionados são adicionados a `extra_vars` com o tipo de inventário como a chave:

```
extra_vars = {
    "product_id": 1,
    "customer_id": 456,
    "SIM Card": 789,          # inventory_id do SIM selecionado
    "Mobile Number": 101,    # inventory_id do número de telefone
                              selecionado
    "iccid": "",             # De provisioning_json_vars
    "msisdn": "",            # De provisioning_json_vars
    "access_token": "eyJ..."
}
```

## 4. **Passa tudo para o Ansible** via `hostvars[inventory_hostname]`

Dentro do playbook, as variáveis são acessíveis como:

```
- name: Obter inventory_id para SIM Card
  set_fact:
    inventory_id_sim_card: "{{ hostvars[inventory_hostname]['SIM Card'] | int }}"
  when: "'SIM Card' in hostvars[inventory_hostname]"
```

## Como os Playbooks Usam Variáveis de Inventário

Uma vez que o playbook tem os IDs de inventário, ele recupera os detalhes completos do inventário da API:

```
- name: Obter Detalhes do SIM Card do Inventário
  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: Extrair ICCID e IMSI do inventário
  set_fact:
    iccid: "{{ sim_card_response.json.iccid }}"
    imsi: "{{ sim_card_response.json.imsi }}"

- name: Obter Detalhes do Número de Telefone do Inventário
  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: Extrair MSISDN
  set_fact:
    msisdn: "{{ phone_number_response.json.msisdn }}"

```

O playbook pode então usar esses valores para:

- Provisionar o SIM card no HSS com o IMSI
- Configurar o número de telefone no sistema de cobrança
- Atribuir os itens de inventário ao cliente
- Criar o registro de serviço com esses detalhes

## Exemplo do Mundo Real: Provisionamento de SIM Móvel

Do `play_psim_only.yaml`, veja como ele usa dados de produto e inventário:

```

- name: Obter informações do Produto da API CRM
  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: Definir fatos do pacote a partir do produto

```

```

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: Definir inventory_id_sim_card se o SIM Card foi selecionado
  set_fact:
    inventory_id_sim_card: "{{ hostvars[inventory_hostname]['SIM Card'] | int }}"
  when: "'SIM Card' in hostvars[inventory_hostname]"

- name: Definir inventory_id_phone_number se o Mobile Number foi selecionado
  set_fact:
    inventory_id_phone_number: "{{ hostvars[inventory_hostname]['Mobile Number'] | int }}"
  when: "'Mobile Number' in hostvars[inventory_hostname]"

- name: Obter detalhes do SIM Card do inventário
  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: Obter detalhes do Número de Telefone do inventário
  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: Extrair valores do inventário
  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: Provisionar assinante no 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: Atribuir inventário ao cliente
  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

```

Isso demonstra o fluxo completo:

1. A definição do produto especifica provisioning\_play: "play\_psim\_only"
2. O produto requer inventory\_items\_list: ['SIM Card', 'Mobile Number']
3. O usuário seleciona itens de inventário durante o provisionamento
4. Os IDs de inventário são passados para o playbook como extra\_vars
5. O playbook recupera detalhes completos do inventário da API
6. O playbook usa dados de inventário para configurar equipamentos de rede
7. O playbook marca o inventário como atribuído ao cliente

## Rollback e Limpeza: Padrão de Melhores Práticas

**Melhor Prática Crítica:** O mesmo playbook deve lidar tanto com rollback de provisionamento falhado quanto com desprovisionamento intencional usando a estrutura block e rescue do Ansible.

## Estrutura do Playbook

Do play\_psim\_only.yaml:

```
- name: Provisionamento de Serviço OmniCore 2024
  hosts: localhost
  gather_facts: no
  become: False

  tasks:
    - name: Bloco principal
      block:
        # --- TAREFAS DE PROVISIONAMENTO ---
        - name: Obter informações do Produto
          uri: ...

        - name: Criar conta no OCS
          uri: ...

        - name: Provisionar assinante no HSS
          uri: ...

        - name: Criar registro de serviço
          uri: ...

        # ... muitas mais tarefas de provisionamento ...

      rescue:
        # --- TAREFAS DE LIMPEZA ---
        # Esta seção é executada quando:
        # 1. Qualquer tarefa no bloco falha (rollback)
        # 2. action == "deprovision" (limpeza intencional)

        - name: Obter itens de inventário vinculados a este serviço
          uri:
            url: "{{ crm_config.crm.base_url }}/crm/inventory/customer_id/{{ customer_id }}"
            method: GET
            register: inventory_api_response
            ignore_errors: True

        - name: Retornar inventário para o 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: Excluir Conta da Cobrança
  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: Excluir Perfil de Atributo
  uri:
    url: "http://{{ crm_config.ocs.OCS }}/jsonrpc"
    method: POST
    body:
      {
        "method": "APIerSv1.RemoveAttributeProfile",
        "params": [{
          "ID": "ATTR_ACCOUNT_{{ service_uuid }}"
        }]
      }
    ignore_errors: True

- name: Remover Perfil de Recurso
  uri: ...
  ignore_errors: True

- name: Remover Filtros
  uri: ...
  ignore_errors: True

- name: Desprovisionar Assinante do 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: Atualizar Assinante para Estado Dormante
  uri:
    url: "{{ item.key }}/subscriber/{{
item.value.subscriber_id }}"
    method: PATCH
    body:
      {
        "enabled": true,
        "msisdn": "9999{{ imsi[-10:] }}", # Número fictício
        "ue_ambr_dl": 9999999,           # Altamente
elevado
        "ue_ambr_ul": 9999999
      }
  loop: "{{ hss_subscriber_data | dict2items }}"
when:
  - deprovision_subscriber | default(false) | bool == false

# A asserção final determina sucesso ou falha
- name: Definir status como "Sucesso" se desprovisionamento
manual / Falhar se provisionamento falhou
  assert:
    that:
      - action == "deprovision"

```

## Por que Este Padrão é uma Melhor Prática

### 1. Sem Duplicação de Código

As mesmas tarefas de limpeza lidam com ambos os cenários:

- **Provisionamento Falhado (Rollback):** Se qualquer tarefa no block falhar, a seção rescue é executada automaticamente
- **Desprovisionamento Intencional:** Quando chamado com action: "deprovision", o playbook salta imediatamente para rescue

### 2. Limpeza Completa Garantida

Quando um provisionamento falha parcialmente, a seção de resgate garante:

- Todas as contas OCS criadas são excluídas
- Todas as entradas de equipamentos de rede configuradas são removidas
- O inventário atribuído é retornado ao pool
- Assinantes HSS são excluídos ou definidos como dormentes
- Nenhum provisionamento parcial permanece em nenhum sistema

Isso previne recursos "órfãos" que:

- Consomem inventário sem serem rastreados
- Criam contas de cobrança que não estão vinculadas a serviços
- Causam confusão durante a solução de problemas
- Desperdiçam recursos de rede

### 3. Tratamento de Falhas Elegante com `ignore_errors`

Observe que cada tarefa de limpeza usa `ignore_errors: True`. Isso é intencional porque:

- Durante o rollback, alguns recursos podem não ter sido criados ainda
- Queremos tentar todas as tarefas de limpeza mesmo que algumas falhem
- A asserção final determina o sucesso/falha geral

Por exemplo, se o provisionamento falhar em "Criar conta no OCS", a limpeza tentará:

- Excluir a conta OCS (falhará, mas será ignorada)
- Remover perfis de atributos (falhará, mas será ignorada)
- Retornar inventário (sucede)
- Excluir assinante HSS (pode não existir, ignorado)

### 4. Distinguindo Desprovisionamento de Rollback

A asserção final no final de `rescue` é inteligente:

```
- name: Definir status como "Sucesso" se desprovisionamento manual /  
Falhar se provisionamento falhou  
  assert:  
    that:  
      - action == "deprovision"
```

Isso significa:

- **Se ```action == "deprovision"```:** A asserção passa, o playbook tem sucesso (status 0)
- **Se ```action`` não estiver definido ou != "deprovision"`:** A asserção falha, o playbook falha (status 2)

Assim, o mesmo código de limpeza resulta em diferentes status de trabalho de provisionamento dependendo da intenção.

### 5. Limpeza Condicional Baseada no Tipo de Serviço

Algumas tarefas de limpeza usam condicionais para lidar com diferentes cenários:

```
- name: Desprovisionar Assinante do HSS
```

```

uri: ...
when:
  - deprovision_subscriber | bool == true

- name: Atualizar Assinante para Estado Dormante
  uri: ...
  when:
    - deprovision_subscriber | default(false) | bool == false

```

Isso permite uma limpeza flexível:

- **Exclusão completa:** Quando SIMs são dedicados a clientes (deprovision\_subscriber: true)
- **Estado dormente:** Quando SIMs são reutilizáveis e devem permanecer no HSS (deprovision\_subscriber: false)

## Como Usar Este Padrão

### Para Provisionamento:

```

{
  "product_id": 1,
  "customer_id": 456,
  "provisioning_play": "play_psim_only"
}

```

Se o provisionamento falhar, o rollback automático ocorre via rescue.

### Para Desprovisionamento:

```

{
  "service_id": 123,
  "service_uuid": "Service_abc123",
  "action": "deprovision",
  "provisioning_play": "play_psim_only"
}

```

O playbook salta diretamente para a seção rescue, executa toda a limpeza e sucede.

## Resumo dos Benefícios

◇ **Fonte única de verdade:** Um playbook lida com provisionamento e desprovisionamento  
 ◇ **Operações atômicas:** Ou totalmente provisionado ou totalmente limpo  
 ◇ **Sem recursos órfãos:** Provisionamentos falhados não deixam rastros  
 ◇ **Manutenção mais fácil:** Alterações na lógica de provisionamento aplicam-se automaticamente à limpeza  
 ◇ **Redução de erros:** Sem chance de código de provisionamento e desprovisionamento ficarem fora de



sincronia ♦ **Testável:** Pode testar a lógica de desprovisionamento executando com action: "deprovision"

Este padrão deve ser seguido em todos os playbooks de provisionamento para garantir confiabilidade e consistência.

## Sobrescrevendo Variáveis do Produto

As provisioning\_json\_vars podem ser sobrescritas no momento do provisionamento. Por exemplo, um produto pode definir:

```
{
  "provisioning_json_vars": "{\\"monthly_cost\\": 50,
  \\"data_limit_gb\\": 100}"
}
```

Mas ao provisionar, você pode sobrescrever isso:

```
{
  "product_id": 1,
  "customer_id": 456,
  "monthly_cost": 45,
  "data_limit_gb": 150
}
```

As extra\_vars mescladas usarão os valores sobrescritos. Isso permite:

- Preços personalizados para clientes específicos
- Limites de dados diferentes com base em promoções
- Testes com diferentes parâmetros sem modificar o produto

## Produtos Sem Inventário

Nem todos os produtos requerem inventário. Por exemplo, um complemento de dados ou uma alternância de recurso pode ter:

```
{
  "product_id": 10,
  "product_name": "Extra 10GB Data",
  "provisioning_play": "play_local_data_addon",
  "provisioning_json_vars": "{\\"data_gb\\": 10}",
  "inventory_items_list": "[]"
}
```

Neste caso, o playbook recebe:

```
extra_vars = {
  "product_id": 10,
```

```
"customer_id": 456,  
"service_id": 123, # Serviço para adicionar dados  
"data_gb": 10,  
"access_token": "eyJ..."  
}
```

O playbook simplesmente adiciona os dados ao serviço existente sem necessidade de itens de inventário.

## Fluxo de Trabalho de Provisionamento Padrão

O provisionamento padrão é iniciado quando:

- Um membro da equipe adiciona um serviço a um cliente pela UI
- Um cliente solicita um serviço através do portal de autoatendimento
- A API é chamada diretamente com PUT /crm/provision/

### Quando Você Clica em "Provisionar"

Aqui está o fluxo completo que ocorre quando um usuário clica no botão "Provisionar":

#### 1. UI Exibe Seleção de Produto

O usuário seleciona um produto do catálogo de produtos. O produto contém:

- provisioning\_play - Qual playbook Ansible executar
- inventory\_items\_list - Inventário necessário (por exemplo, ['SIM Card', 'Mobile Number'])
- provisioning\_json\_vars - Variáveis padrão

#### 2. Seletor de Inventário (Se Necessário)

Se inventory\_items\_list não estiver vazio, um modal aparece mostrando dropdowns para cada tipo de inventário. O usuário deve selecionar itens de inventário disponíveis antes de prosseguir.

#### 3. Botão de Provisionamento Clicado

JavaScript envia a requisição 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. API Recebe a Requisição

O endpoint de provisionamento (routes/provisioning.py):

- Valida a autenticação (token Bearer, chave API ou lista de IPs permitidos)
- Verifica se o usuário tem permissão CREATE\_PROVISION
- Extrai `initiating_user` do token
- Carrega a definição do produto do banco de dados
- Recupera o caminho do playbook: `OmniCRM-API/Provisioners/plays/play_psim_only.yaml`

#### 5. Variáveis Mescladas

O sistema combina variáveis de várias fontes:

```
# Do produto  
product_vars = json.loads(product['provisioning_json_vars'])  
# Do corpo da requisição  
request_vars = request.json  
# Adicionadas pelo sistema  
system_vars = {  
    'product_id': 42,  
    'customer_id': 123,  
    'access_token': g.access_token, # Veja a seção de autenticação  
    'initiating_user': 7  
}  
# Final mesclado  
extra_vars = {**product_vars, **request_vars, **system_vars}
```

#### 6. Registro de Provisionamento Criado

Registro no banco de dados criado com status 1 (Executando):

```
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, # Executando  
    'task_count': 85,  
    'initiating_user': 7,  
    'created': '2025-01-10T14:30:00Z'  
}
```

## 7. Thread em Segundo Plano Gerada

```
run_playbook_in_background(  
    playbook='plays/play_psim_only.yaml',  
    extra_vars=extra_vars,  
    provision_id=456,  
    refresh_token=refresh_token # Para atualização de token durante  
a execução  
)
```

## 8. API Retorna Imediatamente

Resposta retornada à UI com provision\_id:

```
{  
  "provision_id": 456,  
  "provisioning_status": 1,  
  "message": "Trabalho de provisionamento criado"  
}
```

## 9. UI Faz Polling para Atualizações

A UI começa a fazer polling em GET /crm/provision/provision\_id/456 a cada 3 segundos para verificar o status. A resposta inclui:

```
{  
  "provision_id": 456,  
  "provisioning_status": 1,  
  "task_count": 12,  
  "provisioning_result_json": [  
    {  
      "event_number": 1,  
      "event_name": "Obter informações do Produto da API CRM",  
      "provisioning_status": 0,  
      "timestamp": "2024-01-15T10:30:05"  
    },  
    {  
      "event_number": 2,  
      "event_name": "Atribuir SIM Card do inventário",  
      "provisioning_status": 1,  
      "timestamp": "2024-01-15T10:30:07"  
    }  
  ]  
}
```

## 10. Ansible Executa em Segundo Plano

O playbook executa tarefas sequencialmente:

- Cada conclusão de tarefa cria um registro `Provision_Event` no banco de dados
- O evento inclui: nome da tarefa, status (0=sucesso, 2=falhou, 3=falhou mas ignorado), resultado JSON
- A UI exibe progresso em tempo real mostrando tarefas concluídas e a tarefa atualmente em execução
- Tarefas falhadas mostram mensagens de erro nos detalhes do evento

## Rastreamento na UI:

Enquanto o provisionamento está em execução (status 1), os usuários podem visualizar:

- **Página de Detalhes do Serviço** - Mostra o badge de status de provisionamento (Executando/Sucesso/Falhou)
- **Registro de Atividades** - Lista todos os eventos de provisionamento com timestamps
- **Visualização de Detalhes de Provisionamento** - Mostra progresso tarefa por tarefa com expandir/colapsar para detalhes

Exemplo de exibição:

Status de Provisionamento: Executando (8 de 12 tarefas concluídas)

✓ Obter informações do Produto da API CRM ✓ Obter detalhes do Cliente ✓  
 Atribuir SIM Card do inventário (ICCID: 8991101200003204510) ✓ Atribuir  
 Número de Telefone (555-0123) ⌘ Criar conta no OCS/CGRateS (em progresso...)  
 ⌘ Configurar políticas de rede ⌘ Criar registro de serviço ...

## 11. Provisionamento Completo

Status final definido:

- `provisioning_status: 0` - Sucesso
- `provisioning_status: 2` - Falhou (erro crítico)

A UI para de fazer polling e exibe o resultado:

- **Sucesso:** Marca de verificação verde, serviço marcado como Ativo, usuário pode visualizar detalhes do serviço
- **Falha:** X vermelho, mensagem de erro exibida, opção de tentar novamente ou entrar em contato com o suporte
- **Notificação por e-mail:** Se falhar, e-mail enviado para `provisioning.failure_list` na configuração

## Autenticação e Autorização

### Rastreamento de Usuário

Cada trabalho de provisionamento rastreia qual usuário o iniciou:

- **Iniciado pelo usuário:** O campo `initiating_user` é definido como o ID do usuário de seu token JWT
- **Autenticação por chave API:** Usa o ID do primeiro usuário administrador
- **Autenticação por lista de IPs:** Usa o ID do primeiro usuário administrador

## Verificações de Permissão

O sistema verifica permissões antes de permitir o provisionamento:

- Funcionários precisam da permissão `CREATE_PROVISION`
- Clientes só podem provisionar serviços para sua própria conta (`VIEW_OWN_PROVISION` permissão)

## Como o Ansible se Autentica com a API CRM

Os playbooks Ansible precisam fazer chamadas de API autenticadas de volta ao CRM (para buscar detalhes do produto, criar serviços, atualizar inventário, etc.). A autenticação é tratada através de **tokens Bearer** passados para o playbook.

A origem do `access_token` depende do método de autenticação usado para chamar a API de provisionamento:

### Método 1: Login do Usuário (Token Bearer)

Quando um usuário faz login via UI web:

1. O usuário se autentica: `POST /crm/auth/login`
2. Recebe JWT `access_token` (de curta duração, 15-30 min) e `refresh_token` (de longa duração)
3. Faz a requisição de provisionamento com o token Bearer no cabeçalho
4. A API de provisionamento extrai o token do cabeçalho `Authorization: Bearer ...`
5. Armazena em `g.access_token` (contexto de requisição Flask)
6. Passa para o Ansible como variável `access_token`

### Código (permissions.py):

```
# Extrair token Bearer do cabeçalho
auth_header = request.headers.get('Authorization', '')
if auth_header.startswith('Bearer '):
    bearer_token = auth_header[7:]
    # Validar e decodificar
    decoded_token = jwt.decode(bearer_token, secret_key,
    algorithms=['HS256'])
    # Armazenar para provisionamento
    g.access_token = bearer_token
```

### Código (provisioning.py):

```
if "access_token" in g:
    json_data['access_token'] = g.access_token
run_playbook(playbook_path, extra_vars=json_data,
provision_id=provision_id)
```

### Método 2: Chave API (Cabeçalho X-API-KEY)

Para sistemas automatizados que usam chaves API:

1. O sistema faz a requisição: PUT /crm/provision/ com X-API-KEY: sua-chave-api... cabeçalho
2. A API de provisionamento valida a chave API contra crm\_config.yaml
3. **Gera um novo token JWT em tempo real** para o primeiro usuário administrador
4. Armazena em g.access\_token
5. Passa para o Ansible

### Por que Gerar um Token?

As chaves API são strings, não JWTs. Os playbooks chamam endpoints de API esperando autenticação JWT. Então:

- Valide a chave API
- Se válida e tem papel de admin, gere um JWT temporário
- Use o ID do primeiro usuário administrador como sujeito do JWT
- O token permite que o playbook faça chamadas de API autenticadas

### Código (permissions.py):

```
def handle_api_key_auth(f, api_key, *args, **kwargs):
    if not secure_compare_api_key(api_key):
        return {'message': 'Chave API inválida'}, 401

    API_KEYS = yaml_config.get('api_keys', {})
    if api_key in API_KEYS:
        if 'admin' in API_KEYS[api_key].get('roles', []):
            admin_user_id = retrieve_first_admin_user_id()
            access_token =
create_access_token(identity=str(admin_user_id))
            g.access_token = access_token
```

### Método 3: Lista de IPs Permitidos

Para sistemas internos confiáveis em redes privadas:

1. O sistema faz a requisição de um IP permitido (por exemplo, 192.168.1.100)

2. A API de provisionamento verifica o IP do cliente contra `ip_whitelist` em `crm_config.yaml`
3. Se permitido, **gera um novo token JWT** para o primeiro usuário administrador
4. Armazena em `g.access_token`
5. Passa para o Ansible

### Código (permissions.py):

```
def handle_ip_auth(f, *args, **kwargs):
    client_ip = get_real_client_ip()
    if not is_ip_whitelisted(client_ip):
        return {'message': 'Acesso negado'}, 403

    admin_user_id = retrieve_first_admin_user_id()
    access_token = create_access_token(identity=str(admin_user_id))
    g.access_token = access_token
```

### Usando o Token nos Playbooks

Cada chamada de API no playbook inclui o token:

```
- name: Obter Detalhes do Produto
  uri:
    url: "http://localhost:5000/crm/product/product_id/{{ product_id
  }}"
  headers:
    Authorization: "Bearer {{ access_token }}"

- name: Criar Registro de Serviço
  uri:
    url: "http://localhost:5000/crm/service/"
    method: PUT
    headers:
      Authorization: "Bearer {{ access_token }}"
    body:
      customer_id: "{{ customer_id }}"
      service_name: "Serviço Móvel"
```

### Expiração e Atualização do Token

Playbooks de longa duração (5-10 minutos) podem ultrapassar o `access_token` (15-30 min de expiração). Para provisionamentos iniciados pelo usuário, o sistema passa tanto `access_token` quanto `refresh_token`:

```
refresh_token = request.cookies.get('refresh_token')
run_playbook(playbook_path, extra_vars, provision_id,
refresh_token=refresh_token)
```



Se o `access_token` expirar, o executor do playbook pode:

1. Detectar resposta 401 Não Autorizado
2. Chamar `POST /crm/auth/refresh` com `refresh_token`
3. Receber novo `access_token`
4. Tentar novamente a requisição falhada

Para autenticação por chave API/lista de IPs, os tokens gerados podem ter expiração mais longa (1-2 horas) uma vez que estes são sistemas automatizados confiáveis.

## O Processo de Provisionamento

### 1. Criação do Trabalho

Quando uma requisição de provisionamento é recebida, o sistema:

- Valida a requisição e verifica permissões
- Carrega o playbook Ansible especificado na definição do produto
- Cria um registro `Provision` no banco de dados com status 1 (Executando)
- Extrai variáveis da definição do produto e do corpo da requisição
- Captura tokens de autenticação para acesso à API

### 2. Tratamento de Tokens

Os playbooks Ansible precisam se autenticar com a API CRM para recuperar dados e fazer alterações. O sistema de provisionamento trata isso de duas maneiras:

- **Token Bearer (JWT):** Para provisionamento iniciado pelo usuário, o `refresh_token` da requisição é usado para gerar tokens de acesso frescos durante a execução do playbook
- **Autenticação por Chave API/IP:** Para sistemas automatizados, um `access_token` é passado diretamente para o playbook via `g.access_token`

### 3. Execução em Segundo Plano

O playbook é executado em uma thread de segundo plano usando `playbook_runner_v2`. Isso permite que a API retorne imediatamente enquanto o provisionamento continua de forma assíncrona.

Durante a execução:

- Cada conclusão/falha de tarefa cria um registro `Provision_Event`
- O manipulador de eventos monitora falhas críticas vs. ignoradas
- Atualizações de status em tempo real são escritas no banco de dados
- A UI pode fazer polling para atualizações via `GET /crm/provision/`

provision\_id/<id>

#### 4. Execução do Playbook

O playbook Ansible normalmente realiza estas operações:

- Recupera informações do produto da API
- Recupera informações do cliente da API
- Atribui itens de inventário (cartões SIM, endereços IP, números de telefone, etc.)
- Cria contas no OCS/OCS
- Configura equipamentos de rede
- Cria o registro de serviço na API CRM
- Adiciona transações de custo de configuração
- Envia e-mails/SMS de boas-vindas para os clientes

#### 5. Tratamento de Erros

Os playbooks Ansible usam seções block e rescue para rollback:

- Se uma tarefa crítica falhar, a seção de resgate remove o provisionamento parcial
- Tarefas com ignore\_errors: true são marcadas como status 3 e não falham o trabalho
- Erros fatais (sintaxe YAML, falhas de conexão) criam um evento de erro especial com informações de depuração

### Exemplo: Playbook de Provisionamento Padrão

Aqui está um exemplo do play\_simple\_service.yaml:

```
- name: Play de Provisionamento Simples
  hosts: localhost
  gather_facts: no
  become: False

  tasks:
    - name: Bloco principal
      block:
        - name: Obter informações do Produto da API 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: Definir fatos do pacote
  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: Gerar UUID do Serviço
  set_fact:
    service_uuid: "Service_{{ 99999999 | random | to_uuid }}"

- name: Criar conta no 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: Adicionar Serviço 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": "Serviço: {{ service_uuid }}",
        "service_uuid": "{{ service_uuid }}",
        "service_status": "Active",
        "retail_cost": "{{ monthly_cost | float }}"
      }

```

```

    status_code: 200
    register: service_creation_response

- name: Adicionar Transação de Custo de Configuração
  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 }} - Custos de Configuração",
        "retail_cost": "{{ setup_cost | float }}"
      }
    register: api_response_transaction

rescue:
- name: Remover conta no 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: Falhar o provisionamento
  assert:
    that:
      - false

```

Este playbook demonstra o fluxo típico:

1. Recuperar detalhes do produto da API CRM
2. Gerar um UUID de serviço exclusivo
3. Criar a conta de cobrança no OCS
4. Criar o registro de serviço via API CRM
5. Adicionar transações de custo de configuração
6. Se algo falhar, a seção rescue remove a conta do OCS

# Fluxo de Trabalho de Provisionamento Simples

O provisionamento simples é projetado para sistemas automatizados que precisam acionar o provisionamento sem interação do usuário. O caso de uso mais comum é o OCS acionando o provisionamento de complementos via ActionPlans.

## Endpoints de Provisionamento Simples

OmniCRM fornece dois endpoints de provisionamento simples:

- POST /crm/provision/simple\_provision\_addon/  
service\_id/<id>/product\_id/<id>

Para provisionamento automatizado de complementos (por exemplo, cobranças recorrentes, recargas automáticas)

- POST /crm/provision/simple\_provision\_addon\_recharge/  
service\_id/<id>/product\_id/<id>

Para operações de recarga rápida que precisam de feedback imediato

## Autenticação para Provisionamento Simples

Os endpoints de provisionamento simples usam **listas de IPs permitidos** ou **chaves API** para autenticação:

- O IP de origem da requisição é verificado contra ip\_whitelist em crm\_config.yaml
- Ou uma chave API de api\_keys em crm\_config.yaml pode ser fornecida
- Um token de acesso é gerado e passado para o playbook via g.access\_token

## Exemplo: Callback de ActionPlan do OCS

O OCS pode ser configurado para chamar o endpoint de provisionamento simples ao executar ações recorrentes:

```
{
  "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"
      }
    ]
  }
}

```

A ação faz um POST HTTP para:

Isso aciona o playbook associado (por exemplo, `play_topup_no_charge.yaml`) que adiciona dados/créditos ao serviço.

## Exemplo: Playbook de Recarga Simples

Do `play_topup_monetary.yaml`:

```

- name: Recarga Monetária Móvel - 2024
  hosts: localhost
  gather_facts: no
  become: False

  tasks:
    - name: Obter informações do Produto da API 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: Obter informações do Serviço da API 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

    - name: Definir fatos do serviço
      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: Obter Método de Pagamento do Cliente

```
uri:
  url: "http://localhost:5000/crm/stripe/customer_id/{{
customer_id }}"
  method: GET
  headers:
    Authorization: "Bearer {{ access_token }}"
  return_content: yes
  register: api_response_stripe
```

- name: Cobrar cliente

```
uri:
  url: "http://localhost:5000/crm/stripe/charge_card/{{
customer_stripe_id }}"
  method: POST
  headers:
    Authorization: "Bearer {{ access_token }}"
  body_format: json
  body:
    {
      "retail_cost": "{{ monthly_cost }}",
      "description": "{{ package_name }} recarga",
      "customer_id": "{{ customer_id | int }}",
      "service_id": "{{ service_id | int }}"
    }
  register: api_response_stripe
```

- name: Adicionar saldo monetário ao 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: Adicionar Transação ao 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: Enviar SMS de Notificação
  uri:
    url: "http://sms-gateway/SMS/plaintext/{{ api_key }}"
    method: POST
    body_format: json
    body:
      {
        "source_msisdn": "YourCompany",
        "destination_msisdn": "{{ customer_phone }}",
        "message_body": "Obrigado por recarregar {{ monthly_cost
}}!"
      }
  status_code: 201
  ignore_errors: True

```

Este playbook:

1. Obtém detalhes do serviço e do produto da API
2. Recupera o método de pagamento do cliente
3. Cobra o cliente via API Stripe
4. Adiciona saldo monetário ao OCS
5. Registra a transação no CRM
6. Envia um SMS de confirmação (com `ignore_errors: True` para que falhas não falhem o trabalho)

## Cadeias de Provisionamento

Para produtos complexos que requerem várias etapas de provisionamento, o OmniCRM suporta **cadeias de provisionamento**. Uma cadeia executa múltiplos



playbooks sequencialmente, passando contexto entre eles.

Exemplo de caso de uso: Um serviço agrupado que provisiona:

1. Serviço de internet base (cria o registro de serviço primário)
2. Complemento IPTV (usa o `service_id` da etapa 1)
3. Complemento IP Estático (usa o `service_id` da etapa 1)

O serviço de provisionamento automaticamente:

- Consulta o banco de dados pelo `service_id` criado pelo primeiro playbook
- Injeta-o nas `extra_vars` para os playbooks subsequentes
- Rastreia cada playbook como um registro Provision separado

## Razões de Falha e Depuração

Quando o provisionamento falha, o sistema captura informações detalhadas para ajudar a diagnosticar o problema.

### Cenários Comuns de Falha

#### Falhas Críticas de Tarefa (Status 2)

Essas causam a falha de todo o trabalho de provisionamento:

- Chamadas de API retornando códigos de status inesperados
- Asserções falhando (por exemplo, `assert: that: response.status == 200`)
- Itens de inventário obrigatórios ausentes
- Equipamentos de rede inacessíveis
- Credenciais inválidas ou tokens expirados
- Erros OCS/OCS

#### Falhas Ignoradas (Status 3)

Essas são registradas, mas não falham o trabalho:

- Notificações SMS/e-mail opcionais falhando
- Consultas de dados não críticas (marcadas com `ignore_errors: True`)
- Operações de limpeza durante o desprovisionamento

#### Erros Fatais

Esses impedem a execução do playbook:

- Erros de sintaxe YAML no playbook
- Variáveis indefinidas no playbook
- Arquivos de playbook ausentes

- Falhas de conexão com o controlador Ansible

Quando ocorre um erro fatal, o sistema cria um evento de erro especial contendo:

- O código de saída do Ansible
- Todo o stdout (contém detalhes do erro de sintaxe)
- Todo o stderr (contém erros de tempo de execução)
- Uma lista de causas comuns para esse tipo de falha
- Todas as variáveis passadas para o playbook

## E-mails de Notificação de Erro

Quando o provisionamento falha (status 2), um e-mail é enviado automaticamente para a lista de notificação de falhas configurada (provisioning.failure\_list em crm\_config.yaml).

O e-mail inclui:

- Informações do cliente
- Detalhes do produto/serviço
- Resultados de tarefas codificados por cores:
  - **Verde:** Tarefas bem-sucedidas
  - **Laranja:** Tarefas falhadas mas ignoradas
  - **Vermelho:** Falhas críticas
- Para falhas críticas: Saída de depuração completa incluindo corpos de requisição/resposta
- Para erros fatais: Saída do Ansible, mensagens de erro e causas comuns

## Monitoramento de Trabalhos de Provisionamento

### API de Status de Provisionamento

Para verificar o status de um trabalho de provisionamento:

```
GET /crm/provision/provision_id/<id>  
Authorization: Bearer <token>
```

A resposta inclui:

```
{  
  "provision_id": 123,  
  "customer_id": 456,  
  "customer_name": "John Smith",  
  "product_id": 10,  
  "provisioning_status": 0,  
  "provisioning_play": "play_psim_only",  
  "playbook_description": "Provisionamento de Serviço OmniCore 2024",  
  "task_count": 85,  
}
```

```

"provisioning_result_json": [
  {
    "event_number": 1,
    "event_name": "Obter informações do Produto da API CRM",
    "provisioning_status": 1,
    "provisioning_result_json": "{...}"
  },
  {
    "event_number": 2,
    "event_name": "Criar conta no OCS",
    "provisioning_status": 1,
    "provisioning_result_json": "{...}"
  }
]
}

```

## Listando Trabalhos de Provisionamento

Para obter uma lista paginada de todos os trabalhos de provisionamento:

```

GET /crm/provision/?page=1&per_page=20&sort=provision_id&order=desc
Authorization: Bearer <token>

```

Suporta filtragem:

```

GET /crm/provision/?filters={"provisioning_status": [2]}&search=Mobile
Authorization: Bearer <token>

```

Isso retorna apenas trabalhos falhados (status 2) onde a descrição contém "Mobile".

## Melhores Práticas

### Design de Playbook

- **Sempre use block/rescue:** Garanta que o provisionamento parcial possa ser revertido
- **Use ignore\_errors com sabedoria:** Apenas para operações realmente opcionais
- **Registre variáveis importantes:** Use tarefas debug para registrar valores-chave para solução de problemas
- **Valide respostas:** Use assert para verificar se as respostas da API são como esperado
- **Idempotência:** Projete playbooks para serem seguros para reexecução

## Autenticação

- **Provisionamento iniciado pelo usuário:** Sempre use refresh\_token para playbooks de longa duração
- **Provisionamento automatizado:** Use lista de IPs permitidos ou chaves API com tokens de acesso gerados
- **Expiração do token:** O refresh\_token garante que os tokens de acesso sejam regenerados conforme necessário

## Tratamento de Erros

- **Forneça contexto:** Inclua customer\_id, service\_id e detalhes da operação nas mensagens de erro
- **Notifique adequadamente:** Falhas críticas acionam e-mails, mas não spam para falhas esperadas
- **Informações de depuração:** Capture corpos de requisição/resposta completos nos registros Provision\_Event

## Segurança

- **Valide entradas:** Verifique customer\_id, product\_id, service\_id antes do provisionamento
- **Verificações de permissão:** Verifique se os usuários podem apenas provisionar para clientes autorizados
- **Dados sensíveis:** Use o sistema de redacção para remover senhas/chaves dos logs
- **Lista de IPs permitidos:** Restrinja endpoints de provisionamento simples a sistemas confiáveis apenas

## Desempenho

- **Execução em segundo plano:** Nunca bloqueie respostas da API aguardando provisionamento
- **Intervalos de polling:** A UI deve fazer polling para atualizações de status a cada 2-5 segundos
- **Tarefas paralelas:** Use a paralelização nativa do Ansible para operações independentes
- **Atualizações de banco de dados:** O manipulador de eventos atualiza o banco de dados em tempo real, não há necessidade de consultar durante a execução

## Documentação Relacionada

- concepts\_ansible - Conceitos gerais de provisionamento Ansible
- concepts\_api - Autenticação e uso da API CRM
- concepts\_products\_and\_services - Definições de produtos e serviços
- administration\_inventory - Gerenciamento de inventário para

provisionamento



# Registro de Atividade do Cliente

Toda alteração feita em um cliente, contato, site, serviço e parâmetros financeiros, como Transações, Faturas e Métodos de Pagamento, é registrada no Registro de Atividade.

Isso nos permite rastrear as alterações feitas no sistema, quem as fez e quando foram feitas, e é útil para auditar mudanças e rastrear problemas. Por exemplo, se um cliente disser que nunca recebeu uma fatura, podemos verificar o Registro de Atividade para ver se foi enviada, ou se um contato foi removido, podemos ver quem o removeu e quando.

O Registro de Atividade é uma lista cronológica de alterações, com as alterações mais recentes no topo e as mais antigas mais abaixo na lista, que podem ser filtradas a partir das abas.

Os registros do log de atividade não podem ser excluídos, mas podem ser filtrados, e os detalhes podem ser visualizados para ver o que foi alterado e por quem.

Registro de Atividade do Cliente



# Adicionando um Serviço

Serviços são as coisas pelas quais cobramos o cliente, eles podem ser serviços de internet residencial, serviços móveis ou até mesmo serviços abstratos como alugar uma sub-rede ou fornecer eletricidade medida para um rack.

Um Serviço é apenas uma instância de um Product `<concepts_products_and_services>` para um determinado cliente, que é selecionada do catálogo de produtos e provisionada para o cliente.

Se você ainda não `created a customer <basics_create_customer>`, você precisará fazer isso primeiro, já que os serviços estão vinculados aos clientes, você também precisará definir um método de pagamento para o cliente, pois os serviços geram cobranças que precisam ser pagas.

Os clientes podem provisionar seus próprios serviços (se permitirmos), ou a equipe de Atendimento ao Cliente pode provisionar serviços para o cliente.

Com base na definição do produto, existem regras sobre quem pode comprar um produto, como permitir apenas que clientes empresariais comprem um produto empresarial, ou permitir apenas que clientes com um serviço móvel comprem um complemento móvel.

Os serviços também podem ter componentes de uso, como uso de dados, minutos de chamada, ou outras cobranças baseadas em uso, e podem ter várias cobranças associadas a eles, como cobranças mensais, cobranças únicas ou cobranças de uso, podemos visualizar isso a partir do botão "Uso".

Muitos serviços suportam Complementos, por exemplo, um serviço móvel tem todos os recargas disponíveis no menu de Complementos, e um serviço de internet residencial pode ter um endereço IP estático ou dados extras disponíveis como um complemento, novamente tudo isso é definido no product catalog `<concepts_products_and_services>`.

Serviço - Provisionamento e Cobranças



# Fluxo de Trabalho de Atribuição de Planos

O recurso Atribuir Plano permite que a equipe **provisione serviços para clientes** selecionando produtos do catálogo e iniciando o processo de provisionamento. Este é o método principal para criar novos serviços quando os clientes não se auto-provisionam.

## Visão Geral

A atribuição de um plano envolve:

1. Selecionar um cliente
2. Escolher um produto do catálogo
3. Configurar requisitos de inventário (cartões SIM, equipamentos, etc.)
4. Definir parâmetros de serviço (renovação automática, campos personalizados)
5. Iniciar o provisionamento
6. Monitorar o progresso do provisionamento

Este fluxo de trabalho é utilizado para todos os tipos de serviço: móvel, internet, IPTV e VoIP.

## Acessando Atribuir Plano

**Do Catálogo de Produtos:**

**Da Página do Cliente:**

**Da Página de Adicionais:**

A interface de atribuição de plano é aberta em um modal ou página dedicada.

## Fluxo de Trabalho Passo a Passo

### Passo 1: Navegar pelo Catálogo de Produtos

O catálogo de produtos exibe os produtos disponíveis agrupados por categoria e tipo de cliente.

Catálogo de Produtos - Visão do Cliente

Modal do Carrossel de Adicionais

**Alternâncias de Visualização:**

Categorias: • Planos de Serviço • Adicionais

Alternar entre os filtros Individual e Empresarial filtra os produtos para mostrar apenas aqueles disponíveis para o tipo de cliente selecionado.

**Cartões de Produto:**

Os produtos são exibidos em um carrossel ou grade:

Clique "**Atribuir ao Cliente**" para prosseguir.



## Passo 2: Selecionar Cliente

Se você não estiver na página de um cliente, será solicitado que selecione um cliente.

### Pesquisa de Clientes:

Pesquisar Clientes: [John ▼]

Clientes correspondentes: • John Smith (ID: 123) • John Doe (ID: 456) • Johnson Enterprises (ID: 789)

Digite para pesquisar por:

- Nome do cliente
- ID do cliente
- Endereço de e-mail
- Número de telefone

Selecione o cliente no menu suspenso.

Aviso

Certifique-se de ter selecionado o cliente correto antes de prosseguir. Atribuir um plano ao cliente errado requer intervenção manual para corrigir.

## Passo 3: Configurar Inventário (se necessário)

Se o produto exigir itens de inventário (definidos em `inventory_items_list`), os seletores de inventário aparecerão.

Selecionador de Inventário de Serviço

### Exemplo de Selecionador de Inventário:

Itens de Inventário Necessários:

Cartão SIM \*

Cartões SIM Disponíveis: • SIM-00123 - ICCID: 8944...0001 (Novo) • SIM-00124 - ICCID: 8944...0002 (Novo) • SIM-00125 - ICCID: 8944...0003 (Novo)

Número de Telefone Móvel \*

Números Disponíveis: • +44 7700 900123 (Reservado) • +44 7700 900124 (Disponível) • +44 7700 900125 (Disponível)

### Regras de Seleção de Inventário:

- Asterisco (\*) indica campo obrigatório
- Apenas itens de inventário disponíveis são mostrados (status: "Em Estoque" ou "Novo")
- Os menus suspensos carregam dinamicamente com base nos nomes dos modelos de inventário
- Uma vez selecionados, os itens são reservados temporariamente

### O que Acontece:

- Os itens de inventário selecionados são passados como variáveis para o playbook de provisionamento

- Durante o provisionamento, os itens são atribuídos ao serviço e ao cliente
- O status do item muda de "Em Estoque" para "Atribuído"

## Passo 4: Configurar Renovação Automática (opcional)

Para serviços recorrentes, pode ser solicitado que você defina a renovação automática:

Você gostaria de habilitar a renovação automática para este serviço?

Quando habilitado, este serviço será renovado automaticamente no final de cada período de cobrança e cobrará o método de pagamento padrão do cliente.

[ Não ] [ Sim ]

### Comportamento da Renovação Automática:

- **Sim:** O serviço renova automaticamente, o cliente é cobrado mensalmente
- **Não:** O serviço expira ao final do período do contrato, renovação manual necessária

**Melhor Prática:** Defina como "Sim" para serviços de consumo, "Não" para serviços únicos ou quando o cliente solicita controle manual.

## Passo 5: Revisar e Confirmar

A tela de revisão mostra todas as seleções antes do provisionamento:

Cliente: John Smith (ID: 123) Produto: Móvel - Plano de 20GB

Inventário: • Cartão SIM: SIM-00123 (ICCID: 8944...0001) • Número de Telefone: +44 7700 900123

Preços: • Taxa de Configuração: £0.00 • Custo Mensal: £15.00

Renovação Automática: Sim

[Cancelar] [Confirmar & Provisionar]

Clique "**Confirmar & Provisionar**" para iniciar o processo de provisionamento.

## Passo 6: Progresso do Provisionamento

O modal de provisionamento exibe o progresso em tempo real:

✓ Validando conta do cliente ✓ Atribuindo Cartão SIM (ICCID: 8944...0001) ✓ Atribuindo Número Móvel (+44 7700 900123) ⌂ Configurando conta OCS (em progresso...) ⬢ Criando registro de serviço ⬢ Enviando e-mail de boas-vindas

Progresso: 3 de 6 tarefas concluídas

Progresso do Provisionamento - Visão Detalhada

O modal consulta a API de provisionamento a cada 0,2 segundos para atualizações de status.

### Indicadores de Progresso:

- ☒ Concluído com sucesso
- ⌂ Atualmente em execução
- ⬢ Pendente (não iniciado)

- ✗ Falhou (se ocorrerem erros)

Exemplo de Provisionamento Falhado

## Passo 7: Conclusão

### Sucesso:

Serviço provisionado com sucesso para John Smith

ID do Serviço: 789 Nome do Serviço: Móvel - +44 7700 900123 Status: Ativo

[Ver Serviço] [Fechar]

Clique "**Ver Serviço**" para abrir a página de detalhes do serviço.

### Falha:

Se o provisionamento falhar:

Erro: Não foi possível conectar ao OCS

O registro de serviço foi criado, mas o provisionamento não foi concluído. Por favor, revise o erro e tente novamente.

ID de Provisionamento: 456

[Ver Logs] [Tentar Novamente] [Fechar]

- **Ver Logs:** Abre detalhes do provisionamento com mensagens de erro
- **Tentar Novamente:** Tenta o provisionamento novamente
- **Fechar:** Sai do modal (o registro de serviço permanece, mas não ativado)

## Casos Especiais

### Adicionando Adicionais a um Serviço Existente

Ao atribuir um adicional (categoria: "addon") a um cliente que já possui um serviço:

1. **Deteção Automática de Serviço:**
  - O sistema encontra os serviços existentes do cliente
  - Filtra por service\_type (adicional móvel só mostra para serviços móveis)
  - Se o cliente tiver vários serviços correspondentes, solicita que selecione qual deles
2. **Nenhum Novo Serviço Criado:**
  - O adicional é provisionado contra o service\_id existente
  - Usa a conta OCS do serviço existente
  - Inventário (se houver) atribuído ao serviço existente
3. **Playbook de Provisionamento:**
  - Playbook diferente do de serviços autônomos
  - Normalmente adiciona saldo, recursos ou equipamentos à conta existente

### Exemplo:

Serviços Móveis Existentes: • Móvel - +44 7700 900123 (ID: 789) • Móvel - +44 7700 900456 (ID: 790)

Qual serviço deve receber este adicional? [Móvel - +44 7700 900123 ▼]

[Cancelar] [Continuar]

## Provisionamento para Clientes Empresariais

Clientes empresariais podem ter requisitos adicionais:

- **Seleção de Local** - Escolha qual local de negócios recebe o serviço
- **Atribuição de Contato** - Designar contatos de cobrança/técnicos
- **Campos Personalizados** - Números de conta, centros de custo, números de PO

### Exemplo de Fluxo Empresarial:

Selecionar Local de Instalação: [Escritório de Londres - 123 Main St ▼]

Contato de Cobrança: [Jane Doe - <jane@acme.com> ▼]

Contato Técnico: [Bob Smith - <bob@acme.com> ▼]

Número do Pedido de Compra: [PO-2025-001234\_\_\_\_\_]

[Cancelar] [Continuar]

## Atribuição de Serviço em Lote

Para atribuir o mesmo plano a vários clientes (por exemplo, migrações em massa):

1. Use importação CSV (se disponível)
2. Ou atribua individualmente com configurações de modelo
3. O inventário deve estar disponível em lote
4. Monitore a fila de provisionamento para evitar sobrecarga

## Fluxos de Trabalho Comuns

### Fluxo de Trabalho 1: Novo Serviço Móvel

1. O cliente entra na loja querendo serviço móvel
2. A equipe abre **Produtos → Planos**
3. Alterna para o tipo de cliente **Individual**
4. Seleciona "**Móvel - Plano de 20GB**"
5. Clica em "**Atribuir ao Cliente**"
6. Pesquisa o cliente pelo telefone: "+1234567890"
7. Seleciona **John Smith** dos resultados
8. Escolhe o Cartão SIM do seletor de inventário
9. Escolhe o número móvel disponível
10. Habilita **renovação automática**
11. Confirma e provisiona
12. Acompanha o progresso até a conclusão
13. Entrega o SIM ao cliente com o pacote de boas-vindas

### Fluxo de Trabalho 2: Adicionando Internet a um Cliente Existente

1. Navega até a página do cliente: **Clientes → John Smith**
2. Clica na aba **Serviços**
3. Clica no botão "**Adicionar Serviço**"
4. Navega pelos planos de internet
5. Seleciona "**Fibra - 100Mbps**"
6. Seleciona o local de instalação (se empresarial/múltiplos locais)

7. Escolhe o modem CPE do inventário
8. Define a data de instalação (se necessário)
9. Provisione o serviço
10. Cria o ticket de instalação

### **Fluxo de Trabalho 3: Atribuindo Adicional de Recarga de Dados**

1. O cliente liga: "Preciso de mais dados"
2. A equipe pesquisa o cliente na busca global
3. Abre a aba de serviços do cliente
4. Clica em "Adicionais" ao lado do serviço móvel
5. Seleciona "Aumento de Dados de 5GB"
6. Pagamento autorizado
7. Adicional é provisionado imediatamente
8. O cliente recebe um aumento instantâneo de dados

## **Solução de Problemas**

### **"Nenhum produto disponível"**

- **Causa:** Filtros excluem todos os produtos
- **Correção:**
  - Alternar tipo de cliente (Individual vs Empresarial)
  - Verificar se o catálogo de produtos tem produtos habilitados
  - Verificar se os produtos correspondem à elegibilidade do cliente

### **"Nenhum inventário disponível"**

- **Causa:** Itens de inventário fora de estoque ou todos atribuídos
- **Correção:**
  - Adicionar mais itens de inventário ao sistema
  - Verificar o status do item (deve ser "Novo" ou "Em Estoque")
  - Verificar se os nomes dos modelos de inventário correspondem aos requisitos do produto

### **"Cliente não encontrado"**

- **Causa:** O cliente não existe ou o termo de pesquisa está incorreto
- **Correção:**
  - Criar o cliente primeiro
  - Tentar diferentes termos de pesquisa (ID, e-mail, telefone)
  - Verificar se há erros de digitação

### **Provisionamento fica preso**

- **Causa:** Erro no playbook ou sistema externo inacessível
- **Correção:**
  - Aguardar o tempo limite (normalmente 5 minutos)
  - Verificar os logs de provisionamento para erro específico
  - Verificar se OCS e sistemas de rede estão online
  - Tentar o provisionamento novamente após corrigir o problema

### **Provisionamento é bem-sucedido, mas o serviço não funciona**

- **Causa:** Conta OCS criada, mas rede não atualizada
- **Correção:**
  - Verificar se OCS tem conta

- Verificar se o SIM está ativado no HLR/HSS
- Verificar o provisionamento da rede (RADIUS, DPI, etc.)
- Revisar tarefas do playbook para etapas perdidas

## Melhores Práticas

### Antes de Atribuir:

- Verificar se o cliente tem um método de pagamento válido registrado
- Confirmar a elegibilidade do cliente para o produto (residencial vs empresarial)
- Garantir que o inventário necessário esteja disponível
- Revisar os termos e preços do produto com o cliente

### Durante a Atribuição:

- Verificar a seleção do cliente antes de confirmar
- Selecionar os itens de inventário corretos (verificar números de série)
- Habilitar renovação automática para conveniência (a menos que o cliente se oponha)
- Monitorar o progresso do provisionamento até a conclusão

### Após a Atribuição:

- Verificar se o serviço aparece na lista de serviços do cliente
- Verificar se o status do serviço está "Ativo"
- Confirmar se o inventário foi atribuído corretamente
- Enviar e-mail de boas-vindas ou instruções ao cliente
- Testar o serviço, se possível (fazer chamada de teste, verificar dados)

### Para Adicionais:

- Confirmar se o adicional é compatível com o serviço existente
- Explicar a cobrança (uma vez vs recorrente)
- Verificar o pagamento antes do provisionamento
- Verificar se o saldo foi atualizado imediatamente após o provisionamento do adicional

## Documentação Relacionada

- csa\_add\_service - Visão geral dos serviços
- concepts\_products\_and\_services - Conceitos do catálogo de produtos
- administration\_inventory - Gerenciando itens de inventário
- concepts\_provisioning - Detalhes do sistema de provisionamento
- guide\_product\_lifecycle - Ciclo de vida completo do produto, incluindo provisionamento



# Modificando um Serviço

Os serviços podem ser modificados pelo cliente final através do Self-Care Portal <self\_care\_portal>, ou por um administrador via o portal de administração.

Uma vez que um serviço é provisionado, você pode modificar seus parâmetros, adicionar melhorias ou alterar configurações.

## Editando Parâmetros do Serviço

Parâmetros básicos do serviço podem ser modificados clicando no botão **Editar** na página de detalhes do serviço.

Modal de Edição de Serviço

### Campos Editáveis:

- Nome do Serviço
- Status do Serviço (Ativo, Inativo, Suspenso)
- Notas do Serviço
- Configurações de Auto-Renovação
- Itens de inventário vinculados
- Campos personalizados

As alterações são salvas imediatamente e refletidas no portal de autoatendimento do cliente.

## Adicionando Complementos ao Serviço

Complementos melhoram serviços existentes com recursos adicionais, limites de dados ou hardware. Tipos comuns de complementos incluem:

- **Recargas de dados** - Limite de dados adicional (por exemplo, "5GB Data Boost")
- **Atualizações de recursos** - Capacidades extras (por exemplo, "Chamadas Internacionais")
- **Aluguel de hardware** - Adições de equipamentos (por exemplo, "Modem WiFi 6")
- **Serviços premium** - Recursos aprimorados (por exemplo, "Endereço IP Estático")

## Acessando o Catálogo de Complementos

A partir de uma página de detalhes do serviço:

1. Navegue até o serviço que você deseja melhorar
2. Clique no botão **"Adicionar Complemento"** ou **"Navegar Complementos"**
3. O catálogo de complementos é aberto, filtrado para mostrar apenas complementos compatíveis

### Filtragem Automática:

O sistema filtra automaticamente os complementos com base em:

- **Tipo de Serviço** - Mostra apenas complementos que correspondem ao tipo de serviço (móvel, internet, voip, etc.)
- **Tipo de Cliente** - Filtra por cliente residencial vs. empresarial
- **Compatibilidade do Serviço** - Verifica se o serviço atende aos requisitos do complemento
- **Disponibilidade** - Mostra apenas produtos habilitados

Por exemplo, se você estiver visualizando um serviço móvel residencial, verá apenas complementos móveis marcados para clientes residenciais.

## Exibição de Complementos

Os complementos são exibidos em um carrossel interativo mostrando:

Catálogo de Complementos de Serviço

### Exibição do Cartão do Produto:

#### Informações Exibidas:

- Ícone do produto
- Nome do produto
- Lista de recursos (itens em destaque)
- Custo de configuração
- Custo mensal/recorrente
- Link para termos e condições
- Botão "Adicionar ao Serviço"

## Provisionando um Complemento

### Passo 1: Selecionar Complemento

Clique no cartão do complemento desejado, em seguida clique em **"Adicionar ao Serviço"**



## Passo 2: Selecionar Inventário (se necessário)

Se o complemento requer inventário físico (por exemplo, aluguel de hardware), um seletor de inventário aparece:

Modens Disponíveis: • Modem-12345 - TP-Link AX1800 (Novo) • Modem-12346 - TP-Link AX1800 (Novo) • Modem-12347 - Netgear RAX40 (Novo)

Selecione o item de inventário específico a ser atribuído a este serviço.

## Passo 3: Configurar Auto-Renovação (opcional)

Para complementos recorrentes, você pode ser solicitado:

Você gostaria de habilitar a auto-renovação para este complemento?

[ Não ] [ Sim ]

- **Sim** - O complemento renova automaticamente a cada período de cobrança
- **Não** - Compra única, o cliente deve renovar manualmente

## Passo 4: Confirmar e Provisionar

Revise os detalhes do complemento e clique em "**Confirmar**"

O modal de provisionamento aparece mostrando o progresso em tempo real:

✓ Validando pagamento ✓ Atribuindo inventário ⌂ Configurando serviço (em progresso...) ⌂ Criando transação ⌂ Enviando e-mail de confirmação

Uma vez completo, o complemento aparece na lista de complementos do serviço.

## Provisionamento nos Bastidores:

Quando você adiciona um complemento, o sistema:

1. Valida se o cliente pode comprar o complemento
2. Executa o playbook Ansible do complemento (provisioning\_play)
3. Atribui quaisquer itens de inventário necessários ao serviço
4. Cria um registro de transação para cobrança
5. Atualiza a configuração do serviço (OCS, sistemas de rede, etc.)
6. Envia notificação de confirmação ao cliente

Para detalhes técnicos sobre o provisionamento de complementos, consulte Complete Product Lifecycle Guide - Adding Addons <guide\_product\_lifecycle>.

## Visualizando Complementos Ativos

Complementos ativos aparecem na página de detalhes do serviço na seção **"Complementos"**:

### Exemplo de Exibição:

◇ 5GB Data Boost

Adicionado: 10 Jan 2025 Expira: 17 Jan 2025 Status: Ativo Custo: £5.00

◇ Aluguel de Modem WiFi 6

Adicionado: 01 Dec 2024 Inventário: Modem-12345 Auto-Renov: Sim Mensal: £10.00

## Removendo Complementos

Para remover um complemento de um serviço:

1. Navegue até a página de detalhes do serviço
2. Localize o complemento na seção **"Complementos Ativos"**
3. Clique no **"Remover"** ou no **ícone de lixeira** ao lado do complemento
4. Confirme a remoção

### O que Acontece:

- **Complementos de hardware** - O playbook de desprovisionamento é executado, o inventário é marcado para retorno
- **Complementos virtuais** - Benefícios removidos imediatamente
- **Complementos com auto-renovação** - Cancela renovações futuras
- **Transação criada** - Credita qualquer valor proporcional, se aplicável

### Aviso

Remover complementos de hardware (modems, CPE, etc.) geralmente requer que o equipamento seja retornado. O sistema marcará o inventário para retorno e pode enviar instruções de retorno ao cliente.

## Cenários Comuns de Complementos

### Cenário 1: Cliente Faltando Dados

1. Cliente contata o suporte: "Eu usei minha cota mensal"
2. Funcionário navega até o serviço móvel do cliente
3. Clica em "Adicionar Complemento"
4. Seleciona "5GB Data Boost"
5. Provisiona imediatamente
6. Cliente recebe recarga de dados instantânea

## Cenário 2: Empresa Precisa de IP Estático

1. Cliente empresarial solicita IP estático para VPN
2. Funcionário abre o serviço de internet do cliente
3. Navega pelos complementos, seleciona "Endereço IP Estático"
4. O sistema provisiona IP do pool disponível (inventário)
5. Configura o roteamento no equipamento de rede
6. Empresa recebe detalhes de configuração do IP

## Cenário 3: Aluguel de Equipamento para Serviço de Fibra

1. Cliente se inscreve para internet de fibra
2. Durante o provisionamento, o cliente opta pelo aluguel de modem
3. Funcionário adiciona o complemento "Modem WiFi 6"
4. Seleciona o modem disponível do inventário
5. Modem enviado ao cliente
6. Custo recorrente de £10/mês adicionado à conta

# Solução de Problemas

## "Nenhum complemento compatível disponível"

- **Causa:** Nenhum complemento corresponde ao tipo de serviço ou tipo de cliente
- **Solução:** Verifique se o catálogo de produtos possui complementos com configurações de `service_type` e `residential/business` correspondentes

## Falha no provisionamento de complemento

- **Causa:** Erro no playbook de provisionamento ou inventário indisponível
- **Solução:**
  - Verifique os logs de provisionamento para erro específico
  - Verifique se os itens de inventário estão em estoque (se necessário)
  - Revise os logs do playbook no sistema de provisionamento

## Seletor de inventário não mostra itens

- **Causa:** Nenhum item de inventário disponível do tipo requerido
- **Solução:**
  - Adicione itens de inventário ao sistema
  - Verifique se os itens existentes não estão todos atribuídos ou danificados
  - Verifique se o nome do modelo de inventário corresponde exatamente a `inventory_items_list`

## Auto-renovação não funcionando

- **Causa:** Sinal de auto-renovação não definido ou método de pagamento

expirado

- **Solução:**

- Verifique se o serviço tem `auto_renew: true`
- Verifique se o cliente tem um método de pagamento válido registrado
- Revise trabalhos agendados no sistema de cobrança

## **Documentação Relacionada**

- `guide_product_lifecycle` - Fluxo completo de provisionamento de complementos
- `csa_add_service` - Criando novos serviços
- `concepts_products_and_services` - Conceitos de produtos e serviços
- `administration_inventory` - Gestão de inventário para complementos de hardware



# Gerenciamento de Serviços

A interface de Gerenciamento de Serviços fornece ferramentas abrangentes para visualizar, filtrar e gerenciar serviços ao cliente em toda a sua organização.

Documentação relacionada: Adicionando Serviços <csa\_add\_service>, Uso de Serviços <csa\_service\_usage>, Modificando Serviços <csa\_modify>, Produtos e Serviços <concepts\_products\_and\_services>.

## Acessando a Lista de Serviços

Navegue para:

Ou diretamente:

Isso exibe todos os serviços com capacidades de filtragem, ordenação e gerenciamento em massa.

## Visão Geral da Lista de Serviços

A lista de serviços exibe os serviços em uma tabela com:

Visualização da Lista de Serviços

### Colunas:

- **ID do Serviço** - Identificador único
- **Nome do Serviço** - Nome descritivo (por exemplo, "Móvel - +44 7700 900123")
- **Nome do Cliente** - Cliente associado (link clicável)
- **Tipo de Serviço** - móvel, iptv, internet, voip
- **Status** - Ativo, Inativo, Suspenso
- **Data de Início** - Quando o serviço começou
- **Data de Término** - Quando o serviço expira/termina
- **Custo Mensal** - Preço de varejo
- **Ações** - Visualizar, Editar, Adicionais, Uso, Excluir

## Filtrando Serviços

### Abas de Status

Filtre rapidamente os serviços por status usando a barra de abas:

- **Todos os Resultados** - Mostra todos os serviços, independentemente do status
- **Ativos** - Serviços atualmente ativos (visualização padrão)
- **Inativos** - Serviços cancelados ou expirados
- **Suspensos** - Serviços temporariamente desativados (não pagamento, fraude, etc.)

Clicar em uma aba atualiza a lista instantaneamente. A aba ativa é destacada.

### Filtro por Tipo de Serviço

Filtre por tipo de serviço:

Selecione um ou vários tipos para mostrar apenas os serviços correspondentes.

#### Casos de Uso:

- **Móvel** - Visualizar todos os serviços de SIM móvel
- **Internet** - Mostrar fibra, DSL e wireless fixo
- **IPTV** - Exibir serviços de assinatura de TV
- **VoIP** - Listar serviços de voz sobre IP

#### Filtro por Cliente

Pesquise e filtre por cliente:

Comece a digitar o nome de um cliente para ver resultados correspondentes:

Selecione um cliente para mostrar apenas seus serviços.

**Caso de Uso:** Visualizar rapidamente todos os serviços de um cliente específico.

#### Pesquisa

Pesquisa global em todos os campos de serviço:

Pesquisa:

- Nome do serviço
- ID do serviço
- Nome do cliente
- Números de telefone (para serviços móveis)
- UUID do serviço

**Exemplo:** Pesquise "0770" para encontrar todos os serviços com números de telefone contendo "0770".

## Ordenando Serviços

Clique em qualquer cabeçalho de coluna para ordenar por esse campo:

#### Colunas Ordenáveis:

- ID do Serviço (padrão: mais recente primeiro)
- Nome do Serviço (alfabético)
- Nome do Cliente (alfabético)
- Tipo de Serviço (alfabético)
- Data de Início (cronológico)
- Data de Término (cronológico)
- Custo Mensal (numérico)

#### Direção da Ordenação:

- Clique uma vez: Ascendente (A-Z, mais antigo-mais recente, mais baixo-mais alto)
- Clique duas vezes: Descendente (Z-A, mais recente-mais antigo, mais alto-mais baixo)
- A coluna de ordenação ativa mostra o indicador ▲ ou ▼

#### Dropdown de Ordenação:

O dropdown no canto superior direito fornece predefinições rápidas de ordenação:

## Visualizando Detalhes do Serviço

Clique em qualquer nome de serviço para abrir a visualização detalhada do serviço.

### Abas de Detalhes do Serviço:

1. **Visão Geral** - Resumo do serviço, status, datas, preços
2. **Inventário** - Equipamentos atribuídos (cartões SIM, modems, etc.)
3. **Transações** - Cobranças, créditos, pagamentos
4. **Uso** - Estatísticas de uso de dados/voz/SMS
5. **Adicionais** - Adicionais ativos e disponíveis
6. **Registro de Atividades** - Histórico de alterações

### Ações Rápidas

Na lista de serviços, clique no menu de ações ( : ) para operações rápidas:

- **Visualizar** - Abre os detalhes do serviço
- **Editar** - Modifica os parâmetros do serviço
- **Adicionais** - Navegar e adicionar melhorias ao serviço
- **Uso** - Visualizar uso atual e saldo
- **Excluir** - Cancelar/remover serviço

## Operações em Massa

Selecione vários serviços usando caixas de seleção para realizar ações em massa.

### Selecionando Serviços

#### Seleção Individual:

Clique na caixa de seleção ao lado de cada serviço que deseja selecionar.

#### Selecionar Todos:

Clique na caixa de seleção no cabeçalho da tabela para selecionar todos os serviços visíveis.

☒ Serviço 1 ☒ Serviço 2 ☒ Serviço 3

#### Contador de Seleção:

A interface mostra quantos serviços estão selecionados:

### Ações em Massa

Uma vez que os serviços estão selecionados, os botões de ação em massa aparecem:

#### Excluir Vários Serviços:

1. Selecione os serviços a serem excluídos
2. Clique no botão "**Excluir Selecionados**"
3. Confirme a exclusão no modal

Aviso

A exclusão em massa é permanente e irá:

- Cancelar todos os serviços selecionados
- Marcar o inventário como não atribuído
- Parar todas as cobranças recorrentes
- Criar entradas no registro de atividades

#### Casos de Uso:

- Limpar serviços de teste
- Cancelar serviços de contas de clientes fechadas
- Remover entradas duplicadas ou errôneas

**Melhor Prática:** Use filtros para restringir a lista antes de operações em massa para evitar exclusões acidentais.

## Gerenciamento de Status do Serviço

### Tipos de Status

Os serviços podem ter três status:

#### Ativo

- O serviço está operacional
- Cobranças se aplicam
- O cliente pode usar o serviço
- Exibido com um distintivo verde

#### Inativo

- O serviço foi cancelado ou expirou
- Nenhuma cobrança aplicada
- O cliente não pode usar o serviço
- Exibido com um distintivo cinza
- Inventário marcado para retorno/refabricação

#### Suspenso

- Serviço temporariamente desativado
- Pode ou não cobrar (configurável)
- O cliente não pode usar o serviço
- Exibido com um distintivo laranja/amarelo
- Razões comuns: Não pagamento, investigação de fraude, solicitação do cliente

### Mudando o Status do Serviço

Para mudar o status de um serviço:

1. Abra os detalhes do serviço
2. Clique no botão **"Editar"**
3. Altere o dropdown **"Status do Serviço"**
4. Clique em **"Salvar"**

#### O que Acontece:

- **Ativo → Inativo:** Aciona o playbook de desprovisionamento (se configurado)
- **Ativo → Suspenso:** Desativa o serviço na OCS/rede, mas mantém o registro
- **Suspenso → Ativo:** Reativa o serviço, retoma a cobrança



- **Inativo → Ativo:** Pode acionar o re-provisionamento (use com cautela)

## Visualização de Uso do Serviço

Clique em "**Uso**" no menu de ações para abrir o modal de uso.

Modal de Uso do Serviço

### Informações Exibidas:

- **Saldo** - Crédito restante ou valor pré-pago
- **Uso de Dados** - Usado vs. alocado (barra de progresso)
- **Uso de Voz** - Minutos usados vs. limite do plano
- **Uso de SMS** - Mensagens enviadas vs. limite
- **Data de Expiração** - Quando o saldo/plano atual expira

### Ações:

- **Recarregar** - Adicionar crédito ou dados (para serviços pré-pagos)
- **Ver Detalhes** - Ver detalhamento do uso
- **Exportar** - Baixar relatório de uso (se disponível)

## Próximas Renovações Automáticas

A visualização de Próximas Renovações Automáticas fornece uma interface centralizada para monitorar e gerenciar todas as renovações de serviços agendadas em sua organização.

### Acessando Renovações Automáticas

Navegue para:

Ou diretamente:

Isso exibe todos os serviços com renovação automática agendada, ordenados pela próxima data de renovação.

### Visão Geral das Renovações Automáticas

A lista de renovações automáticas exibe renovações agendadas em uma tabela com:

#### Colunas:

- **Cliente** - Nome do cliente (link clicável para a visão geral do cliente)
- **Serviço** - Nome do serviço (link clicável para a visão geral do cliente)
- **Produto** - Nome do produto/plano sendo renovado
- **Custo** - Custo da renovação (do preço de varejo do produto)
- **Renova Em** - Data e hora da próxima renovação com formato legível por humanos
- **Status** - Status do serviço (Ativo, Suspenso, etc.)
- **Ações** - Renovar agora ou remover renovação automática

#### Exibição de Exemplo:

Painel de Próximas Renovações Automáticas

## Como Funciona a Renovação Automática

A renovação automática é agendada no CGRateS (o sistema de cobrança) quando um serviço é criado ou modificado. O sistema:


1. **Agenda Ação** - Cria um ActionPlan no CGRateS com a data de renovação
2. **Monitora Saldo** - Verifica se o cliente tem saldo suficiente antes da renovação
3. **Executa Renovação** - Na data agendada, renova automaticamente o serviço
4. **Atualiza Registros** - Cria transações, atualiza datas de serviço, e registra atividades

### Fonte de Dados:

A visualização de Próximas Renovações Automáticas consulta diretamente o CGRateS usando a `Apierv1.GetScheduledActions` API e enriquece os dados com informações de cliente e serviço do banco de dados CRM.

## Renovar Agora

Para acionar manualmente uma renovação antes da data agendada:

1. Clique no botão  (**Renovar Agora**) para o serviço
2. Revise os detalhes da renovação no modal de confirmação:

Cliente: Acme Corp Serviço: FixedWireless\_75628fa5 Produto: Internet Mega Residencial Custo: \$89.99

Próxima renovação agendada: 11 de Nov, 2025 10:45 AM

[Cancelar] [Confirmar Renovação]

3. Clique em "**Confirmar Renovação**" para processar imediatamente
4. Um trabalho de provisionamento é criado e executado
5. Monitore o progresso no modal de status de provisionamento

### O que Acontece:

- O serviço é recarregado com as permissões do produto
- O saldo é atualizado no CGRateS
- Um registro de transação é criado
- A data de término do serviço é estendida
- Uma entrada no registro de atividades é criada

### Casos de Uso:

- O cliente solicita renovação antecipada
- Resolver problemas de serviço redefinindo permissões
- Testar processos de renovação
- O cliente deseja usar o serviço antes da renovação agendada

## Remover Renovação Automática

Para cancelar a renovação automática de um serviço:

1. Clique no botão  (**Remover Renovação Automática**) para o serviço

2. Confirme a exclusão no modal:

Isso removerá a renovação automática agendada para:

Cliente: Acme Corp Serviço: FixedWireless\_75628fa5 Próxima Renovação: 11 de Nov, 2025 10:45 AM

O serviço não será renovado automaticamente. Você precisará renovar manualmente ou o serviço expirará na data de término.

[Cancelar] [Remover Renovação Automática]

3. Clique em **"Remover Renovação Automática"** para confirmar

4. O ActionPlan é removido do CGRateS

5. O serviço expirará naturalmente, a menos que renovado manualmente

**O que Acontece:**

- O ActionPlan é excluído do CGRateS
- O status do serviço permanece inalterado
- O serviço expirará na sua data de término atual
- O cliente deve renovar manualmente ou o serviço será interrompido

**Casos de Uso:**

- O cliente está cancelando o serviço
- Mudando para o processo de renovação manual
- O plano de serviço está mudando
- O cliente solicitou parar cobranças automáticas

Aviso

Remover a renovação automática significa que o serviço expirará, a menos que renovado manualmente. Os clientes não receberão continuidade automática do serviço.

**Entendendo a Exibição**

**Coluna Cliente:**

- Mostra o nome do cliente como um link clicável
- Links para a página de visão geral do cliente
- Mostra "N/A" se o serviço não estiver vinculado a um cliente (serviço órfão)

**Coluna Serviço:**

- Mostra o nome/UUID do serviço
- Links para a página de visão geral do cliente
- Mostra o UUID do serviço se o nome não estiver definido
- Mostra "N/A" se o serviço não puder ser encontrado no banco de dados

**Coluna Produto:**

- Nome do produto do ActionPlan do CGRateS
- Extraído dos metadados da ação agendada
- Sempre mostra o nome do produto, mesmo que o serviço não seja encontrado

### Coluna Custo:

- Custo de varejo da tabela de Produtos no CRM
- Valor que será cobrado na renovação
- Mostra "N/A" se o produto não for encontrado no banco de dados

### Coluna Renova Em:

- Data e hora exatas da próxima renovação
- Tempo relativo legível por humanos (por exemplo, "3 horas a partir de agora", "2 dias a partir de agora")
- Consciente do fuso horário (usa o fuso horário do servidor CGRateS)

### Coluna Status:

- Status atual do serviço do banco de dados CRM
- **Ativo** (Verde) - O serviço está operacional
- **Suspenso** (Laranja) - O serviço está temporariamente desativado
- **Desconhecido** (Cinza) - Serviço não encontrado no banco de dados ou status não definido

## Resolução de Problemas

### Serviços mostrando "N/A" para Cliente/Serviço

- **Causa:** UUID do serviço no CGRateS não corresponde ao service\_uuid no banco de dados CRM
- **Solução:**
  - Verifique se o serviço existe no banco de dados
  - Verifique se o formato do service\_uuid corresponde: ServiceType\_UUID (por exemplo, FixedWireless\_75628fa5)
  - O serviço pode ter sido excluído do CRM, mas ainda está agendado no CGRateS

### Custo mostrando "N/A"

- **Causa:** ID do produto no CGRateS não existe na tabela de Produtos do CRM
- **Solução:**
  - Verifique se o produto existe no banco de dados
  - Verifique se o product\_id no ActionPlan corresponde a um produto no CRM
  - O produto pode ter sido excluído

### "Renova Em" não mostrando data

- **Causa:** NextRunTime não presente na resposta do CGRateS
- **Solução:**
  - Verifique a configuração do ActionPlan do CGRateS
  - Verifique se o ActionTiming está configurado corretamente
  - Verifique os logs do CGRateS em busca de erros

### "Renovar Agora" falha

- **Causa:** Vários erros de provisionamento
- **Resolução de Problemas:**
  - Verifique o modal de status de provisionamento para detalhes do erro
  - Verifique se os playbooks do Ansible estão configurados corretamente
  - Verifique a conectividade com a OCS
  - Revise os logs de provisionamento

### Renovação automática removida, mas ainda mostrando

- **Causa:** Atraso de cache ou problema de sincronização do CGRateS
- **Solução:**
  - Clique no botão "**Atualizar**" para recarregar os dados
  - Verifique se o ActionPlan foi realmente removido no CGRateS
  - Verifique a conectividade da API do CGRateS

## Paginação

Os serviços são exibidos em páginas para desempenho:

◀ Anterior 1 [2] 3 4 5 Próximo ▶

Itens por página: [10 ▼]

- 10
- 25
- 50
- 100

### Controles:

- **Anterior/Próximo** - Navegar pelas páginas
- **Números de Página** - Ir para uma página específica
- **Itens por Página** - Ajustar quantos serviços são exibidos por página

**Dica de Desempenho:** Use filtros para reduzir o total de resultados em vez de aumentar itens por página.

## Distintivos e Indicadores de Serviço

Indicadores visuais ajudam a identificar rapidamente os estados do serviço:

### Distintivos de Status:

#### Indicador de Renovação Automática:

Serviços com renovação automática ativada mostram:

#### Expirando em Breve:

Serviços que expiram dentro de 7 dias mostram:

#### Atrasado:

Serviços com saldo pendente mostram:

## Integração CGRateS (Avançado)

Para serviços integrados com CGRateS (o motor de cobrança e tarifação), administradores podem gerenciar configurações avançadas diretamente da visualização do serviço. Isso inclui atributos, filtros e visualização de sessões ativas.

Nota

Os recursos de integração do CGRateS requerem a permissão **cgrates\_api\_access**. Apenas administradores têm acesso a esses recursos por padrão. Consulte rbac para configuração de

permissões.

## Provisionamento Automático vs. Gerenciamento Manual

Na operação normal, atributos e filtros do CGRateS são provisionados automaticamente pelo Ansible durante o fluxo de trabalho inicial de provisionamento do serviço. Quando um novo serviço é criado, os playbooks de provisionamento:

- Criam a conta de serviço no CGRateS
- Configuram atributos (IMSI, MSISDN, identificadores de conta, perfis de velocidade, etc.)
- Configuram filtros para garantir a tarifação correta
- Aplicam os planos de tarifação apropriados

No entanto, há casos em que você pode precisar modificar essas configurações após o serviço ter sido provisionado:

- **Mudanças no Perfil de Velocidade** - O cliente atualiza/reduz a largura de banda (MaxBitrateDL/UL)
- **Ajustes de Política** - Alterar políticas de QoS ou regras de modelagem de tráfego (PcefPolicyName)
- **Mudanças no Número de Telefone** - Atualizar MSISDN ou outros identificadores
- **Resolução de Problemas** - Corrigir configurações incorretas ou testar diferentes configurações
- **Configurações Especiais** - Aplicar atributos personalizados que não fazem parte do provisionamento padrão

A interface de gerenciamento manual permite que os administradores façam essas mudanças diretamente sem reexecutar todo o fluxo de trabalho de provisionamento. Isso é particularmente útil para:

- **Mudanças Rápidas** - Modificar um único atributo sem esperar pelo provisionamento
- **Testes** - Experimentar diferentes configurações
- **Suporte ao Cliente** - Resolver problemas rapidamente durante chamadas de suporte
- **Configurações Personalizadas** - Aplicar configurações específicas do serviço que não são cobertas por modelos

### Aviso

Mudanças manuais na configuração do CGRateS ignoram o fluxo de trabalho padrão de provisionamento. Certifique-se de entender o impacto de suas mudanças, pois configurações incorretas podem afetar a cobrança e a funcionalidade do serviço. Todas as mudanças são registradas no feed de atividades do cliente para fins de auditoria.

## Acessando Recursos do CGRateS

Ao visualizar ou editar um serviço que está provisionado no CGRateS, três seções colapsáveis aparecem na parte inferior do formulário de serviço:

- **Atributos do CGRateS** - Configurar atributos específicos do serviço
- **Filtros do CGRateS** - Definir regras de filtragem para o serviço
- **Sessões Ativas** - Visualizar sessões ativas em tempo real

Cada seção é colapsada por padrão para manter a interface limpa. Clique no cabeçalho da seção para expandir e visualizar/editar a configuração.

### Visualização Colapsada das Seções do CGRateS

As seções colapsadas mostram distintivos de contagem indicando quantos atributos, filtros ou

sessões ativas existem para o serviço.

## Atributos do CGRateS

Os atributos permitem definir campos e transformações personalizados que são aplicados a eventos de tarifação para este serviço específico.

**Formato do ID do Atributo:** ATTR\_ACCOUNT\_{service\_uuid}

**Exemplo:** Para o serviço com UUID Mobile\_SIM\_c2880638, o ID do perfil de atributo é ATTR\_ACCOUNT\_Mobile\_SIM\_c2880638

### Gerenciando Atributos:

1. Abra a visualização de edição do serviço
2. Expanda a seção "**Atributos do CGRateS**"
3. Clique no botão "**Editar Atributos**"
4. Adicione/modifique/remova atributos conforme necessário
5. Clique em "**Salvar Atributos**"

Visualização de Edição de Atributos do CGRateS

### Campos de Atributo:

- **Caminho** - O campo a ser modificado (por exemplo, \*req.Account, \*req.IMSI)
- **Tipo** - Como o valor é definido:
  - \*constant - Define um valor fixo
  - \*variable - Captura valor de campos de evento usando RSRParser
  - \*composed - Anexa valor em vez de sobrescrever
  - \*usage\_difference - Calcula a duração entre dois campos
  - \*sum - Soma múltiplos valores
  - \*value\_exponent - Computa o expoente de um campo
- **Regras** - O(s) valor(es) a serem aplicados (pode ter várias regras por atributo)

### Exemplo de Configuração de Atributo:

```
{
  "Path": "*req.Account",
  "Type": "*constant",
  "Value": [{"Rules": "Mobile_SIM_474a380a"}]
}
```

### Casos de Uso Comuns:

- Definir identificador de conta para tarifação
- Mapear IMSI/MSISDN para serviço
- Configurar limites de largura de banda (MaxBitrateDL/UL)
- Definir nomes de políticas (PcefPolicyName)
- Transformar ou enriquecer eventos de tarifação

### Registro de Atividades:

Todas as modificações de atributos são registradas no feed de atividades do cliente com todos os detalhes das mudanças feitas.

## Filtros do CGRateS

Filtros definem regras de correspondência que determinam quando a configuração deste serviço

deve ser aplicada durante a tarifação.

**Formato do ID do Filtro:** FLTR\_ACCOUNT\_{service\_uuid}

**Exemplo:** Para o serviço com UUID Mobile\_SIM\_c2880638, o ID do filtro é FLTR\_ACCOUNT\_Mobile\_SIM\_c2880638

### Gerenciando Filtros:

1. Abra a visualização de edição do serviço
2. Expanda a seção "**Filtros do CGRateS**"
3. Clique no botão "**Editar Filtros**"
4. Adicione/modifique/remova regras de filtro
5. Clique em "**Salvar Filtros**"

### Campos de Regra de Filtro:

- **Elemento** - O campo a ser correspondido (por exemplo, ~\*req.Account, ~\*req.Destination)
- **Tipo** - Tipo de correspondência:
  - \*string - Correspondência exata de string
  - \*prefix - Começa com o valor especificado
  - \*suffix - Termina com o valor especificado
  - \*empty - O campo está vazio
  - \*exists - O campo existe
  - \*notexists - O campo não existe
  - \*timings - Correspondência de padrões de hora/data
  - \*destinations - Correspondência de padrões de destino
  - \*rsr - Correspondência de campo RSR
  - \*gt / \*gte / \*lt / \*lte - Comparações numéricas
- **Valores** - O(s) valor(es) a serem correspondidos (pode ter múltiplos valores por regra)

### Exemplo de Configuração de Filtro:

```
{
  "Element": "~*req.Account",
  "Type": "*string",
  "Values": ["Mobile_SIM_474a380a"]
}
```

### Casos de Uso Comuns:

- Garantir que atributos se apliquem apenas a contas específicas
- Filtrar por destino (nacional vs. internacional)
- Filtragem baseada em tempo (horário de pico vs. fora de pico)
- Filtrar por tipo ou categoria de serviço

### Registro de Atividades:

Todas as modificações de filtros são registradas no feed de atividades do cliente.

### Sessões Ativas

Visualize sessões ativas em tempo real para este serviço. Isso mostra chamadas em andamento, sessões de dados ou outros eventos faturáveis atualmente em progresso.

### Visualizando Sessões Ativas:



1. Abra a visualização de edição do serviço
2. Expanda a seção "**Sessões Ativas**"
3. Veja a lista de sessões ativas
4. Clique em "**Ver Detalhes**" em qualquer sessão para ver os dados completos da sessão
5. Clique em "**Atualizar**" para recarregar a lista de sessões

Visualização de Sessões Ativas do CGRateS

#### **Informações da Sessão Exibidas:**

- **Hora de Configuração** - Quando a sessão começou
- **Uso** - Duração atual da sessão (em segundos)
- **Destino** - Número chamado ou destino

#### **Modal de Detalhes da Sessão:**

Clicar em "Ver Detalhes" abre um modal mostrando:

- **Informações Básicas:**
  - CGRID (ID da sessão)
  - Conta (UUID do serviço)
  - Hora de configuração
  - Uso/duração atual
  - Destino
  - Categoria
- **Dados Completos da Sessão:**
  - Representação JSON completa da sessão
  - Todos os campos da sessão do CGRateS
  - Estado da sessão em tempo real
  - Visualizador JSON rolável para inspeção

Modal de Detalhes da Sessão do CGRateS

#### **Casos de Uso:**

- Monitorar chamadas ativas ou sessões de dados
- Resolver problemas de faturamento
- Verificar se a sessão está sendo tarifada corretamente
- Verificar atributos e valores da sessão
- Auditar o uso ativo do serviço

#### **Taxa de Atualização:**

As sessões são buscadas sob demanda quando você expande a seção. Clique em "Atualizar" para obter os dados mais recentes da sessão.

Nota

Apenas sessões que correspondem à conta deste serviço (UUID do serviço) são exibidas. O filtro `*string:~*req.Account:{service_uuid}` é aplicado automaticamente.

#### **Proxy da API do CGRateS**

Todas as operações do CGRateS (atributos, filtros, sessões) usam o endpoint do proxy da API do OmniCRM:

**Endpoint:** POST /crm/ocs/proxy

## Campos Obrigatórios:

- **method** - Método da API do CGRateS (por exemplo, `APIerSv1.GetAttributeProfile`)
- **params** - Array de parâmetros para o método
- **customer\_id** - ID do cliente (para registro de atividades)
- **service\_id** - ID do serviço (para registro de atividades)

## Campos Opcionais:

- **tenant** - Inquilino do CGRateS (padrão para valor de configuração)

## Exemplo de Solicitação:

```
{
  "method": "APIerSv1.GetAttributeProfile",
  "params": [{"ID": "ATTR_ACCOUNT_Mobile_SIM_c2880638"}],
  "customer_id": 123,
  "service_id": 456
}
```

## Configuração do Inquilino:

O inquilino é definido automaticamente a partir do arquivo de configuração do OmniCRM (`crm_config.yaml`) sob `ocs.ocsTenant`. Isso garante que todas as operações do CGRateS usem o inquilino correto sem codificar valores no frontend.

## Requisito de Permissão:

A permissão `cgrates_api_access` é necessária. Esta permissão é concedida ao papel `admin` por padrão.

## Registro de Atividades:

Todas as operações da API do CGRateS que não são GET são registradas automaticamente no feed de atividades do cliente, incluindo:

- Método da API chamado
- Inquilino utilizado
- Todos os parâmetros enviados
- ID do serviço em que a operação foi realizada
- Usuário que realizou a operação
- Timestamp

Isso cria uma trilha de auditoria completa de todas as mudanças de configuração do CGRateS.

## Resolução de Problemas da Integração do CGRateS

### "Permissão Negada" ao acessar recursos do CGRateS

- **Causa:** O usuário não possui a permissão `cgrates_api_access`
- **Solução:** Conceda permissão ao papel do usuário (normalmente recurso exclusivo para administradores)

### Atributos ou Filtros não carregando

- **Causa:** Problema de conectividade com o CGRateS ou perfil não existe
- **Solução:**
  - Verifique a conectividade do servidor CGRateS na configuração

- Verifique se a configuração do inquilino está correta
- Verifique o console do navegador em busca de erros de API
- O perfil pode não existir ainda (mostrará formulário vazio)

### **Mudanças não salvando**

- **Causa:** Erro de validação ou erro da API do CGRateS
- **Solução:**
  - Verifique se os campos obrigatórios estão preenchidos (Caminho, Tipo, Elemento)
  - Verifique se o formato JSON está correto
  - Verifique o registro de atividades para detalhes do erro
  - Revise os logs do CGRateS

### **Nenhuma sessão ativa sendo exibida**

- **Causa:** Nenhuma sessão atualmente ativa para este serviço
- **Solução:**
  - Isso é normal se o serviço não estiver em uso
  - Tente atualizar após iniciar uma sessão (chamada, dados, etc.)
  - Verifique se o UUID do serviço corresponde à conta no CGRateS

### **Detalhes da sessão não atualizando em tempo real**

- **Causa:** Os dados da sessão são buscados sob demanda, não ao vivo
- **Solução:** Clique no botão "Atualizar" para obter os dados mais recentes da sessão

### **Registro de atividades não mostrando mudanças do CGRateS**

- **Causa:** Apenas operações que não são GET são registradas (leituras não são registradas)
- **Solução:** Isso é intencional - apenas gravações/modificações criam entradas de atividade

## **Fluxos de Trabalho Comuns**

### **Fluxo de Trabalho 1: Encontrar Serviços do Cliente**

1. Clique no **filtro de Tipo de Serviço** (opcional)
2. Clique no **filtro de Cliente**
3. Digite o nome do cliente
4. Selecione o cliente no dropdown
5. Revise os serviços do cliente

### **Fluxo de Trabalho 2: Identificar Serviços que Estão Prestes a Expirar**

1. Clique na aba "**Ativos**"
2. Ordene por "**Data de Término**" (crescente)
3. Serviços que expiram mais cedo aparecem primeiro
4. Contate os clientes para renovação

### **Fluxo de Trabalho 3: Limpar Serviços de Teste**

1. Pesquise por "teste" na caixa de pesquisa
2. Revise os resultados para confirmar que são dados de teste
3. Selecione todos os serviços de teste
4. Clique em "**Excluir Seleccionados**"
5. Confirme a exclusão

## Fluxo de Trabalho 4: Suspender Cliente Não Pagante

1. Navegue até a conta do cliente
2. Veja a aba de serviços
3. Selecione todos os serviços ativos
4. Altere o status para "Suspendido"
5. Salve as alterações

## Fluxo de Trabalho 5: Visualizar Uso do Serviço Móvel

1. Filtre por **Tipo de Serviço: Móvel**
2. Clique no nome do serviço para abrir os detalhes
3. Clique na aba **"Uso"**
4. Revise o consumo de dados/voz/SMS
5. Identifique usuários pesados ou excessos

## Resolução de Problemas

### Serviços não aparecendo na lista

- **Causa:** Filtro de status ocultando resultados
- **Solução:** Clique na aba "Todos os Resultados" para mostrar todos os status

### Não é possível encontrar serviço pela pesquisa

- **Causa:** Termo de pesquisa não corresponde aos dados armazenados
- **Solução:**
  - Tente uma pesquisa parcial (por exemplo, "0770" em vez do número completo)
  - Use o filtro de cliente em vez disso
  - Verifique se há erros de digitação

### Botão de exclusão em massa desativado

- **Causa:** Nenhum serviço selecionado
- **Solução:** Marque as caixas ao lado dos serviços que deseja excluir

### Ordenação não funcionando

- **Causa:** Coluna não é ordenável
- **Solução:** Apenas colunas com ícones ▲▼ são ordenáveis

### Página carrega lentamente

- **Causa:** Muitos serviços para exibir
- **Solução:**
  - Aplique filtros para reduzir o conjunto de resultados
  - Reduza itens por página
  - Use a pesquisa para restringir resultados

## Documentação Relacionada

- csa\_add\_service - Adicionando novos serviços
- csa\_modify - Modificando serviços e adicionando adicionais
- concepts\_products\_and\_services - Conceitos de produtos e serviços
- basics\_customers - Gerenciamento de clientes



# Uso do Serviço e Rastreamento de Saldo

O sistema de Uso do Serviço fornece **monitoramento em tempo real** do consumo dos clientes para dados, voz, SMS e saldos monetários. Este recurso se integra com o OCS (Sistema de Cobrança Online) para exibir o uso atual, as permissões restantes e as informações de expiração do saldo tanto para clientes (via Self-Care Portal <self\_care\_portal>) quanto para a equipe.

## Visão Geral

O rastreamento de uso permite:

- **Exibição de Saldo em Tempo Real** - Veja o uso atual e as permissões restantes
- **Múltiplos Tipos de Saldo** - Rastreie dados, voz, SMS e saldos monetários simultaneamente
- **Monitoramento de Expiração** - Veja quando os saldos expiram
- **Detalhamento do Saldo** - Visão detalhada dos diferentes buckets de saldo
- **Atualização Automática** - Atualizações de uso a cada 3 segundos automaticamente

## Acessando o Uso do Serviço

### Da Lista de Serviços:

1. Navegue até **Serviços** → **Lista de Serviços**
2. Clique no **menu de ações ( : )** ao lado de um serviço
3. Selecione **"Uso"**

### Dos Detalhes do Serviço:

1. Abra a página de detalhes de um serviço
2. Clique na aba **"Uso"**

### Da Página do Cliente:

1. Abra a visão geral do cliente
2. Navegue até a aba **Serviços**
3. Clique em **"Ver Uso"** ao lado de qualquer serviço

O modal ou página de uso se abre mostrando dados de consumo em tempo real.

## Exibição de Uso

A interface de uso mostra cartões de resumo e barras de progresso detalhadas para cada tipo de saldo.

Exibição de Uso do Serviço  
Modal de Saldo do Serviço

### Cartões de Resumo

A linha superior exibe cartões de visualização rápida para cada tipo de saldo:

### Informações do Cartão:

- **Tipo de Saldo** - Ícone e rótulo (Dados, Voz, SMS, Monetário)
- **Quantidade Restante** - Saldo atual nas unidades apropriadas
- **Hora de Expiração** - Dias/horas até o saldo expirar
- **Botão Mais Informações** - Clique para expandir o detalhamento

## Barras de Progresso

Abaixo dos cartões, as barras de progresso mostram o consumo visualmente, com porções preenchidas indicando o saldo restante.

### Recursos da Barra de Progresso:

- **Indicador Visual** - Porção preenchida mostra o saldo restante
- **Porcentagem** - Porcentagem numérica do saldo restante
- **Valores Absolutos** - Mostra usado vs. total (ex: "12.5GB / 20GB")
- **Codificação de Cores:**
  - Verde: >50% restante
  - Amarelo: 20-50% restante
  - Vermelho: <20% restante
- **Clicável** - Clique para expandir o detalhamento

## Tipos de Saldo

### Saldo de Dados

Rastreia o consumo de dados da internet.

**Unidades:** Gigabytes (GB) ou Megabytes (MB)

### Formato de Exibição:

Progresso: 12.5GB / 20GB (62%)

### Cenários Comuns:

- **Planos de dados móveis** - 5GB, 10GB, 20GB de permissões mensais
- **Wireless fixo** - Ilimitado ou limitado a altos valores (500GB, 1TB)
- **Recargas** - Dados adicionais comprados no meio do ciclo
- **Serviços de dongle** - Dados pré-pagos para dispositivos hotspot

### Múltiplos Buckets:

Os serviços frequentemente têm múltiplos saldos de dados:

- Permissão mensal (expira mensalmente)
- Dados bônus (expira após o período da campanha)
- Dados de recarga (expiração mais curta, consumidos primeiro)

### Saldo de Voz

Rastreia minutos de chamadas telefônicas.

**Unidades:** Minutos (min)

### Formato de Exibição:

Progresso: 125 min / 500 min (25%)

## Rastreamento de Duração de Chamadas:

- Chamadas recebidas (se cobradas)
- Chamadas feitas
- Chamadas internacionais (bucket separado, se aplicável)
- Números premium

## Cálculo:

O uso de voz é calculado pela duração da chamada em nanosegundos internamente, convertido em minutos para exibição.

## Saldo de SMS

Rastreia o uso de mensagens de texto.

**Unidades:** Mensagens (msgs)

## Formato de Exibição:

Progresso: 45 / 250 (18%)

## Tipos de Mensagens:

- SMS padrão (160 caracteres)
- SMS longo (múltiplos segmentos)
- MMS (se rastreado separadamente)

## Saldo Monetário

Rastreia crédito pré-pago ou saldo da conta.

**Unidades:** Moeda (£, \$, €, etc.)

## Formato de Exibição:

Progresso: £15.50 / £20.00 (77%)

## Uso:

- Contas pré-pagas usam saldo monetário para pagar pelo uso
- O crédito diminui à medida que o cliente usa os serviços
- Pode ser recarregado via pagamento ou voucher
- Pode expirar se não utilizado dentro do período de validade

## Detalhamento do Saldo

Clique em "**Mais Informações**" em qualquer cartão ou clique em uma barra de progresso para expandir o detalhamento.

## Visão Expandida:

◆ Permissão Mensal 20GB

Restante: 12.5 GB Usado: 7.5 GB Expira: 25 Jan 2025 (15 dias) Peso: 10

◆ Dados Bônus 5GB

Restante: 5.0 GB Usado: 0 GB Expira: 31 Jan 2025 (21 dias) Peso: 20

◇ Dados de Recarga 3GB

Restante: 0 GB Usado: 3.0 GB Expira: 18 Jan 2025 (EXPIRADO) Peso: 30

Total Restante: 17.5 GB

### Campos do Bucket de Saldo:

- **ID/Nome** - Identificador para o bucket de saldo
- **Restante** - Quantidade restante neste bucket específico
- **Usado** - Quantidade consumida deste bucket
- **Data de Expiração** - Quando este saldo expira
- **Peso** - Ordem de prioridade (peso mais alto consumido primeiro)

### Sistema de Peso

Os saldos têm um valor de **peso** que determina a ordem de consumo:

- **Peso mais alto = consumido primeiro**
- **Peso mais baixo = consumido por último**

### Exemplo de Pesos:

- Dados de recarga: Peso 30 (consumido primeiro, expiração mais curta)
- Dados bônus: Peso 20 (consumido em segundo)
- Permissão mensal: Peso 10 (consumido por último, expiração mais longa)

Isso garante que os saldos que estão prestes a expirar sejam usados antes dos que duram mais.

## Atualizações em Tempo Real

Os dados de uso são atualizados automaticamente a cada **3 segundos** via polling.

### O que Atualiza:

- Quantidades atuais de saldo
- Barras de progresso de uso
- Temporizadores de expiração
- Detalhes de buckets individuais

### Experiência do Usuário:

- Nenhum recarregamento de página necessário
- Atualizações suaves sem flicker
- Sobreposição de carregamento durante a atualização
- Insígnia de status mostra o estado atual do serviço

### Casos de Uso:

- Monitorar o uso do cliente durante a chamada
- Assistir à diminuição do saldo em tempo real enquanto o cliente usa o serviço
- Verificar a recarga imediatamente após a compra



# Uso em Diferentes Tipos de Serviço

## Serviços Móveis

Exibe todos os quatro tipos de saldo:

- Dados (GB)
- Voz (minutos)
- SMS (mensagens)
- Monetário (moeda)

### Exemplo:

DADOS: 12.5GB restantes VOZ: 125 min restantes SMS: 45 msgs restantes MONETÁRIO: £15.50 restantes

## Wireless Fixo / Internet

Normalmente mostra apenas:

- Dados (GB ou TB)
- Monetário (se pré-pago)

### Exemplo:

DADOS: 450GB / 500GB restantes MONETÁRIO: £45.00 (crédito pré-pago)

## Serviços de Hotspot / Dongle

Mostra rastreamento de dados específico de dongle:

- Dados (consumidos vs. pré-pagos)
- Monetário (saldo pré-pago)

### Modo de Exibição:

Quando `dongle=true`, o componente oculta voz e SMS, mostrando apenas dados relevantes e saldos monetários.

# Resolução de Problemas

## Uso mostrando como 0 / 0

- **Causa:** Serviço não integrado com OCS ou CGRateS
- **Solução:**
  - Verifique se o serviço está provisionado no OCS
  - Verifique a conectividade da API do OCS
  - Revise o mapeamento do UUID do serviço

## Uso não atualizando

- **Causa:** Polling parado ou OCS inacessível
- **Solução:**
  - Atualize a página
  - Verifique o console do navegador em busca de erros
  - Verifique se a API do OCS está online

## Saldos mostram valores incorretos

- **Causa:** Desconexão de dados do OCS ou problema de cache
- **Solução:**
  - Forçar atualização do saldo do OCS
  - Verifique se há transações pendentes
  - Verifique a configuração do OCS

## Datas de expiração ausentes

- **Causa:** Saldo sem data de expiração definida
- **Solução:**
  - Alguns saldos estão configurados para nunca expirar (validade ilimitada)
  - Verifique a configuração do saldo no OCS

## Múltiplos saldos confusos

- **Causa:** Múltiplas recargas ou dados bônus adicionados
- **Solução:**
  - Use a visão de detalhamento para ver todos os buckets
  - Classifique por peso para ver a ordem de consumo
  - Revise as datas de expiração individuais

# Integração com OCS/CGRateS

Os dados de uso vêm do **OCS (Sistema de Cobrança Online)**, tipicamente CGRateS.

### Fluxo de Dados:

1. O usuário abre a visualização de uso
2. OmniCRM chama GET /crm/service/{service\_id}
3. A API consulta o OCS via UUID do serviço
4. O OCS retorna o 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. A UI converte os valores para unidades de exibição (bytes → GB, nanosegundos → minutos)
6. Barras de progresso e cartões renderizados
7. O polling continua a cada 3 segundos

### Mapeamento de Tipos de Saldo do OCS:

O OCS retorna dados de saldo com prefixos de tipo que mapeiam para a exibição da UI:

- \*data → **CARTÃO DE DADOS** (uso da internet)
- \*voice → **CARTÃO DE VOZ** (minutos de chamadas)
- \*sms → **CARTÃO DE SMS** (mensagens de texto)
- \*monetary → **CARTÃO MONETÁRIO** (crédito pré-pago)

Cada tipo de saldo pode ter múltiplos buckets (ex: permissão mensal + dados bônus + dados de recarga), todos exibidos na visão de detalhamento.

### Conversões de Valores de Saldo:

- **Dados:** Bytes → GB (dividir por  $1024^3$ )
- **Voz:** Nanosegundos → Minutos (dividir por  $60 \times 10^9$ )
- **SMS:** Contagem (sem conversão)
- **Monetário:** Centavos → Moeda (dividir por 100)

## Auto-Renovação e ActionPlans

Serviços com auto-renovação habilitada têm **ActionPlans** agendados no OCS.

### O que são ActionPlans?

ActionPlans são tarefas agendadas no CGRateS que são executadas automaticamente em horários específicos para:

- Adicionar saldo a uma conta (recarga automática)
- Renovar permissões mensais
- Aplicar cobranças recorrentes
- Expirar saldos antigos

### Como Funciona a Auto-Renovação:

#### 1. Provisionamento do Serviço:

- Quando o serviço é criado com `auto_renew = true`
- O playbook de provisionamento cria um ActionPlan no OCS
- ActionPlan configurado para rodar mensalmente (ou por ciclo de faturamento)

## 2. Configuração do ActionPlan:

ActionPlan contém:

- **ID da Conta** - UUID do serviço
- **Ações** - O que fazer (adicionar dados, voz, SMS, saldo monetário)
- **Agendamento** - Quando executar (ex: mensalmente no dia 1 às 00:00 UTC)
- **Quantidade** - Quanto saldo adicionar

## 3. Execução Automática:

- O OCS executa o ActionPlan no horário agendado
- Adiciona saldo à conta (ex: 20GB de dados, 500 minutos de voz)
- Define a data de expiração para o novo saldo (ex: 30 dias)
- O cliente é cobrado via método de pagamento registrado

## 4. Visualizando ActionPlans:

- Navegue até os detalhes do serviço na visualização do OCS
- ActionPlans listados com o próximo horário de execução
- Mostra: Nome do plano, data da próxima execução, detalhes da ação

### Exemplo de ActionPlan:

```
{
  "ActionPlanId":
  "ProductID_MonthlyPlan__ProductName_20GB_Mobile__ActionPlan_Monthly_Renew",
  "NextExecTime": "2025-02-01T00:00:00+00:00",
  "ActionName_hr": "Renovação Mensal",
  "PlanName": "20GB Móvel",
  "ActionFrequency_hr": "Todo Plano Mensal",
  "custom_NextExecTime_hr": "em 22 dias"
}
```

### Gerenciando Auto-Renovação:

- **Habilitar** - Definir durante a criação ou modificação do serviço
- **Desabilitar** - Remover ActionPlan do OCS (o serviço mantém o saldo existente, mas não renova automaticamente)
- **Modificar** - Alterar a quantidade ou frequência de renovação via modificação do serviço

### Renovação Manual:

Se a auto-renovação estiver desabilitada, o cliente deve manualmente:

- Recarregar antes que o saldo expire
- Ou o serviço será suspenso quando o saldo se esgotar

### Visualizando na UI:

A aba de serviços mostra o status de auto-renovação:

Próxima Renovação: 1 Feb 2025 (em 22 dias) Valor da Renovação: £15.00

## Melhores Práticas

### Para a Equipe de Suporte:

- Verifique o uso antes de responder chamadas "Por que meu serviço está lento?"
- Verifique o saldo após recargas para confirmar sucesso
- Use o detalhamento para identificar buckets expirados
- Monitore clientes de alto uso para prevenir excessos

#### **Para Clientes (Autoatendimento):**

- Verifique o uso regularmente para evitar ficar sem saldo
- Recarregue antes que o saldo expire
- Entenda o sistema de peso para saber qual saldo é consumido primeiro
- Entre em contato com o suporte se o uso parecer incorreto

#### **Para Administradores:**

- Configure as expirações de saldo apropriadas
- Defina valores de peso para priorizar saldos que estão prestes a expirar
- Monitore a conectividade do OCS para relatórios precisos
- Revise as configurações de saldo para corresponder às ofertas de produtos

## **Documentação Relacionada**

- `features_topup_recharge` - Sistema de recarga para adicionar saldo
- `csa_service_management` - Gerenciando serviços
- `csa_modify` - Adicionando complementos para aumentar permissões
- `concepts_products_and_services` - Configuração de produtos



# Atendimento ao Cliente

## Impersonação de Usuário para Suporte e Solução de Problemas

A impersonação permite que funcionários autorizados façam login temporariamente como outro usuário para solucionar problemas, verificar configurações ou ver exatamente o que o usuário está experienciando. Este recurso é essencial para o suporte ao cliente, mas requer permissões apropriadas e é totalmente auditado.

Ao impersonar um cliente, a equipe acessa o Self-Care Portal <self\_care\_portal> exatamente como o cliente vê, permitindo uma solução de problemas e suporte precisos.

Veja também: RBAC <rbac> para configuração de permissões, Customers <basics\_customers> para gerenciamento de clientes, Self-Care Portal <self\_care\_portal> para recursos do portal do cliente.

## Propósito

A impersonação de usuários fornece:

1. **Solução de Problemas** — Veja exatamente o que o cliente vê para diagnosticar problemas
2. **Verificação** — Confirme se as configurações de serviço e permissões funcionam corretamente
3. **Treinamento** — Demonstre recursos da perspectiva do cliente
4. **Suporte** — Ajude os clientes a navegar pelo sistema sem exigir compartilhamento de tela
5. **Registro de Auditoria** — Todas as sessões de impersonação são registradas para segurança e conformidade

## Permissões Necessárias

Para impersonar usuários, você deve ter uma das seguintes permissões:

- `can_impersonate` — Permissão dedicada de impersonação para a equipe de suporte
- `admin` — Acesso administrativo total (inclui direitos de impersonação)

Usuários sem essas permissões não podem acessar o recurso de impersonação.

# Como Impersonar um Usuário

## Via Interface Web:

1. **Navegar até o Cliente** — Encontre o cliente no CRM
2. **Selecionar Contatos** — Veja a lista de contatos do cliente
3. **Clique em "Login como Usuário"** — O botão aparece ao lado de cada contato que possui uma conta de usuário
4. **Confirmar Impersonação** — O sistema pode solicitar confirmação
5. **Sessão Iniciada** — Você agora está logado como esse usuário

Fazendo login como um Usuário no OmniCRM

## Via API:

Iniciar sessão de impersonação:

**Endpoint:** POST /auth/impersonate

**Permissão Necessária:** can\_impersonate ou admin

## Requisição:

```
{  
  "user_id": 42  
}
```

## Resposta:

```
{  
  "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"  
}
```

Os tokens retornados são para a sessão do usuário impersonado.

## O Que Acontece Durante a Impersonação

Quando você impersona um usuário:

- **Contexto Completo** — Você vê o sistema exatamente como o usuário alvo vê:
  - O painel e a navegação dele
  - Os dados do cliente dele (se usuário cliente)

- Suas permissões e controles de acesso
- Seus serviços, faturas e uso
- **Rastreamento de Sessão** — O sistema rastreia ambas as identidades:
  - `impersonating_user_id` — Seu verdadeiro ID de usuário
  - `target_user_id` — O usuário que você está impersonando
  - `impersonation_start` — Quando a impersonação começou
- **Indicador** — A interface exibe um banner mostrando que você está impersonando alguém:
  - "Você está atualmente logado como [Nome de Usuário]"
  - "Clique aqui para parar a impersonação"
- **Registro de Auditoria** — Todas as ações são registradas com ambos os IDs de usuário:
  - Ações aparecem como realizadas pelo usuário alvo
  - Registros de auditoria registram quem estava realmente realizando-as (usuário impersonador)
  - Registro completo de auditoria mantido na tabela `ImpersonationLog`

## Parando a Impersonação

### Via Interface Web:

1. **Clique no Banner** — Clique no banner de impersonação na parte superior da página
2. **Ou Navegue** — Vá ao menu do usuário e selecione "Parar Impersonação"
3. **Confirmação** — A sessão termina e você retorna à sua própria conta

### Via API:

**Endpoint:** `POST /auth/stop_impersonation`

**Requisição:** Nenhum corpo necessário (requisição autenticada)

### Resposta:

```
{
  "success": true,
  "impersonation_end": "2025-01-04T15:45:00Z",
  "duration_seconds": 900,
  "access_token": "your_original_token",
  "refresh_token": "your_original_refresh_token"
}
```

Sua sessão original é restaurada.

## Registro de Auditoria de Impersonação

Todas as sessões de impersonação são registradas na tabela `ImpersonationLog`



com:

- **impersonating\_user\_id** — Membro da equipe que realizou a impersonação
- **target\_user\_id** — Cliente ou usuário que foi impersonado
- **impersonation\_start** — Timestamp de início
- **impersonation\_end** — Timestamp de término (quando a sessão foi parada)
- **impersonation\_duration** — Duração em segundos

Isso fornece total responsabilidade por todas as sessões de impersonação e permite:

- **Auditorias de segurança** — Revisar quem impersonou quem e quando
- **Relatórios de conformidade** — Demonstrar uso adequado de acesso elevado
- **Investigação** — Rastrear ações durante sessões de impersonação
- **Monitoramento** — Identificar padrões de impersonação incomuns

### Visualizando Registros de Impersonação:

Administradores podem consultar registros de impersonação via API:

```
GET /auth/impersonation_logs?user_id={user_id}&start_date={date}&end_date={date}
```

Filtrar por:

- Usuário impersonador (quem realizou a impersonação)
- Usuário alvo (quem foi impersonado)
- Intervalo de datas
- Duração

## Melhores Práticas

1. **Minimizar Duração** — Impersonar apenas pelo tempo necessário para resolver o problema
2. **Documentar Propósito** — Anotar por que a impersonação foi necessária no log de atividades do cliente
3. **Informar Clientes** — Avisar os clientes que você pode precisar ver a conta deles (política de privacidade)
4. **Verificar Identidade** — Confirmar a identidade do usuário antes de impersonar via conta dele
5. **Revisar Registros** — Auditar regularmente os registros de impersonação em busca de padrões incomuns
6. **Limitar Permissões** — Conceder `can_impersonate` apenas à equipe de suporte que precisa
7. **Treinamento** — Garantir que a equipe entenda a responsabilidade e as implicações de auditoria

# Considerações de Segurança

- **Acesso Completo** — A impersonação concede acesso completo à conta do usuário alvo
- **Nenhuma Senha Necessária** — A impersonação contorna a autenticação (apenas baseada em permissões)
- **Ações Registradas** — Todas as ações durante a impersonação são atribuídas ao usuário alvo nos registros do aplicativo (mas os registros de auditoria mostram o verdadeiro ator)
- **Isolamento de Sessão** — A impersonação cria uma nova sessão; não afeta as sessões ativas do usuário alvo
- **Limites de Tempo** — Sessões de impersonação devem ter limite de tempo (configurável)
- **Contorno de MFA** — A impersonação contorna os requisitos de 2FA (usa a autenticação do impersonador)

## Restrições

- **Não é possível Impersonar Administradores** — Dependendo da configuração, pode não ser possível impersonar outros administradores
- **Mesmas Permissões** — Você obtém as permissões do usuário alvo, não a união das permissões de ambos os usuários
- **Limites de Sessão** — Apenas uma sessão de impersonação por membro da equipe por vez
- **Requisitos de Auditoria** — Não é possível desativar ou ocultar o registro de impersonação

## Solucionando Problemas Comuns

### Problema: O botão "Login como Usuário" não aparece

- Solução: Verifique se você tem a permissão `can_impersonate` ou `admin`
- Solução: Confirme se o contato tem uma conta de usuário vinculada

### Problema: A impersonação falha com erro de permissão

- Solução: Verifique se o usuário alvo é um administrador (pode estar restrito)
- Solução: Verifique se sua permissão de impersonação está ativa

### Problema: Não é possível parar a impersonação

- Solução: Use o endpoint da API para parar a impersonação: `POST /auth/stop_impersonation`
- Solução: Limpe os cookies do navegador e faça login novamente com suas credenciais

**Problema: Ações não estão registrando corretamente**

- Solução: Verifique se a sessão de impersonação está ativa (verifique o banner)
- Solução: Revise os registros de auditoria - as ações são registradas com ambos os IDs de usuário



# Sistema de Transmissão de Células

O Sistema de Transmissão de Células no OmniCRM permite que operadores de redes móveis enviem alertas de emergência e avisos públicos para dispositivos móveis dentro de áreas geográficas específicas. A Transmissão de Células é um recurso crítico de segurança pública usado para alertas AMBER, avisos meteorológicos, alertas de tsunamis e outras notificações de emergência.

**Vantagem Principal:** Ao contrário das mensagens SMS padrão, as mensagens de Transmissão de Células alertarão audivelmente em telefones que estão no modo silencioso, sem crédito ou em roaming. Como esta é uma mensagem de transmissão, é possível enviar um alerta para cada membro da população que possui um telefone móvel em questão de segundos.

## Link de Aviso Omnitouch (OWL)

A plataforma Link de Aviso Omnitouch (OWL) fornece uma solução abrangente para profissionais de gestão de desastres e operadores de redes móveis:

- **Entidade de Transmissão de Células (CBE)** - Aplicação web segura para usuários autorizados criarem e transmitirem mensagens de aviso de emergência
- **Centro de Transmissão de Células (CBC)** - Componente de integração de rede compatível com padrões que se conecta a redes celulares (2G/3G/4G/5G) para distribuir mensagens

O OWL é projetado para uso em qualquer dispositivo com um navegador web (Chrome/Firefox/Safari/Edge), como computadores, laptops, tablets ou telefones móveis.

## Visão Geral

A Transmissão de Células (também conhecida como Sistema de Aviso Público ou PWS) permite que os operadores:

- **Enviar Alertas de Emergência** - Distribuir informações críticas de segurança para todos os dispositivos em uma área
- **Direcionar Regiões Geográficas** - Transmitir para áreas de rastreamento específicas ou células de rede controlando quais torres de celular transmitem mensagens
- **Suportar Múltiplas Línguas** - Fornecer mensagens de alerta em várias línguas simultaneamente (até 500 caracteres por língua)
- **Gerenciar o Ciclo de Vida do Alerta** - Criar, atualizar, aprovar, monitorar

- e excluir mensagens de transmissão
- **Integrar com Sistemas Externos** - Conectar-se à infraestrutura do CBC (Centro de Transmissão de Células) através de múltiplas interfaces de rede celular
- **Autenticação de Dois Fatores** - Processo de aprovação seguro usando Senhas de Uso Único Baseadas em Tempo (TOTP)
- **Regra de Duas Pessoas** - Requisito opcional para aprovação de uma segunda pessoa antes da transmissão da mensagem

Ao contrário do SMS, a Transmissão de Células não requer listas de assinantes e pode alcançar todos os dispositivos capazes em uma área geográfica instantaneamente, tornando-a ideal para alertas de segurança pública críticos em termos de tempo. Na maioria das redes, uma transmissão para todos os dispositivos leva menos de 10 segundos.

## Casos de Uso

A Transmissão de Células é usada para:

- **Avisos de Emergência** - Desastres naturais (terremotos, tsunamis, inundações, incêndios)
- **Alertas AMBER** - Notificações de sequestro de crianças
- **Alertas Meteorológicos** - Avisos de clima severo, alertas de tornados
- **Segurança Pública** - Ameaças terroristas, vazamentos químicos, evacuações
- **Mensagens de Teste** - Testes de sistema e campanhas de conscientização pública (Mensagens de Teste Mensais não alertam os usuários, mas serão recebidas)

O direcionamento geográfico é frequentemente de suma importância em situações de emergência. Por exemplo, alertas de Tsunami aconselhando aqueles próximos a áreas costeiras a buscarem terrenos mais altos não devem ser enviados para pessoas em áreas interiores distantes do perigo. Ao controlar quais torres de celular transmitem mensagens de aviso de emergência, o escopo da transmissão pode ser limitado à área geográfica apropriada.

## Ciclo de Vida da Mensagem de Aviso de Emergência

Transmitir uma mensagem de Aviso de Emergência requer ação rápida enquanto fornece informações precisas e autenticação para garantir a validade da mensagem.

O ciclo de vida da mensagem consiste em quatro etapas:

1. **Definição da Mensagem** - Tipo de mensagem, conteúdo da mensagem, configurações de expiração

2. **Direcionamento da Mensagem** - Áreas geográficas e torres de celular para transmitir
3. **Aprovação da Mensagem** - Autorização/verificação da identidade do operador e aprovação da segunda pessoa (se necessário)
4. **Revisão da Mensagem** - Confirmação final antes da transmissão

Após a transmissão, as mensagens podem ser monitoradas, atualizadas à medida que as situações evoluem e interrompidas quando o perigo imediato diminui.

## **Etapa 1: Definição da Mensagem**

Esta etapa define os parâmetros básicos da mensagem de aviso de emergência a ser transmitida:

### **Identificador da Mensagem**

Diferentes tipos de mensagens têm identificadores diferentes, que são tratados de maneira diferente pelos telefones receptores. Por exemplo, uma Mensagem de Teste Mensal não deve alertar usuários reais, mas eles ainda receberão a mensagem.

**Nota:** Cada modelo de mensagem disponível no sistema OWL já possui o Identificador da Mensagem (MI) apropriado embutido, portanto, os usuários do sistema não precisam selecionar isso manualmente ao criar uma mensagem de alerta.

### **Texto da Mensagem**

Corpo do texto limitado a 500 caracteres contendo a mensagem a ser exibida para os usuários finais. As mensagens podem ser fornecidas em **múltiplas línguas** adicionando o texto da segunda língua abaixo da primeira. Lembre-se de que o limite de 500 caracteres se aplica ao total da mensagem, incluindo todas as línguas.

### **Modelos de Mensagem**

Mensagens de alerta de perigo podem ser pré-definidas com antecedência como "modelos" para diferentes cenários previstos, como:

- Inundações
- Tsunamis
- Terremotos
- Testes periódicos
- Outros cenários de desastre

Os modelos economizam tempo durante emergências. Esses modelos podem ser modificados conforme necessário ao definir a mensagem, ou as mensagens podem ser escritas do zero.

### **Expiração da Mensagem e Repetições**

Mensagens de emergência têm uma vida útil finita para as quais são relevantes. Ao definir a mensagem:

- **Expira (minutos)** - Quanto tempo a mensagem continuará a ser transmitida
- **Repetições da Mensagem** - Quantas vezes será retransmitida

Cada telefone mostrará a mensagem ao usuário apenas uma vez. No entanto, os sites de celular continuarão a transmitir mensagens até que o tempo de expiração seja alcançado para garantir que as pessoas que entram na área de cobertura de fora recebam a mensagem.

## **Etapa 2: Direcionamento**

As mensagens de Transmissão de Células são enviadas no nível da torre de celular, e o alcance geográfico pode ser limitado selecionando quais torres transmitem a mensagem.

### **Direcionamento Opcional**

Esta etapa é opcional. Não inserir nenhuma informação de direcionamento significa que todas as torres de celular transmitirão a mensagem de aviso de emergência.

### **Áreas de Alvo Pré-definidas**

O sistema OWL possui um banco de dados de todas as torres de celular e pode definir áreas de alvo em um mapa. As áreas podem ser direcionadas usando zonas pré-definidas (determinadas com antecedência para seleção rápida) ou desenhando áreas personalizadas no mapa.

### **Ferramentas de Desenho no Mapa**

Áreas de alvo personalizadas podem ser criadas usando:

- Ferramenta de Polígono - Desenhar limites de cobertura precisos
- Ferramenta de Círculo - Alertas rápidos baseados em raio
- Ferramenta de Retângulo - Cobertura alinhada à grade

O recurso "Adicionar Nova Zona" permite definir áreas de alvo personalizadas que podem ser salvas para uso futuro.

## **Etapa 3: Aprovação**

Um processo de aprovação válida que a pessoa emitindo a mensagem de Aviso de Emergência está autorizada a fazê-lo.

### **Autenticação de Dois Fatores**

Usa Senhas de Uso Único Baseadas em Tempo (TOTP) via:

- **Token físico** (como um RSA SecurID)
- **Solução baseada em aplicativo** (Google Authenticator, Authy, Microsoft Authenticator ou outros aplicativos compatíveis com TOTP)

Usuários que criarão ou aprovarão rascunhos de mensagens de alerta devem ter um aplicativo autenticador em seu smartphone para gerar o código de autorização que o sistema solicitará.

### **Configurando 2FA**

Ao configurar o 2FA pela primeira vez:

1. Instale um aplicativo autenticador em seu smartphone (Google Authenticator, Authy, Microsoft Authenticator, etc.)
2. Navegue até as configurações da sua conta OWL e escaneie o código QR com seu aplicativo autenticador
3. Insira o código de verificação para confirmar a configuração
4. Salve os códigos de backup em um local seguro
5. Teste a geração de códigos antes das emergências

Para instruções detalhadas de configuração do 2FA, veja [Autenticação de Dois Fatores <2fa>](#).

Se você trocar de dispositivos móveis ou o aplicativo parar de sincronizar com sua conta OWL, entre em contato com o Administrador do Sistema para obter ajuda. Administradores podem redefinir tokens de 2FA na página **Usuários e Funções** → **Usuários**.

### **Regra de Duas Pessoas**

Quando o processo requer aprovação de uma segunda pessoa, a pessoa emitindo o alerta deve inserir a Senha de Uso Único Baseada em Tempo da outra pessoa antes que o processo possa prosseguir. Isso fornece supervisão e minimiza o risco de uso indevido.

### **Funções de Usuário Granulares**

Funções de usuário individuais podem ser configuradas para:

- Permitir que apenas certos usuários enviem mensagens pré-definidas
- Restringir o direcionamento da mensagem a regiões específicas
- Exigir fluxos de aprovação adicionais

## **Etapas 4: Revisão**

Uma vez que as etapas de Definição da Mensagem, Direcionamento e Aprovação estejam completas, o operador deve revisar a mensagem antes da transmissão final. Uma vez satisfeito com os detalhes da mensagem, ele pode transmitir a mensagem.

**Velocidade de Transmissão:** Na maioria das redes, uma transmissão para todos os dispositivos na rede leva menos de 10 segundos.



## Etapa 5: Monitoramento e Atualizações

Uma vez que a transmissão da mensagem é iniciada, os operadores podem monitorar e gerenciar as mensagens transmitidas.

### Feedback da Rede

As redes celulares retornam informações sobre os sites de celular que transmitiram a mensagem. Se um site de celular estiver offline ou indisponível, isso será relatado de volta ao operador.

### Retransmissão Automática

Se algum site de celular offline se tornar disponível novamente enquanto o Aviso de Emergência ainda estiver ativo, todos os telefones conectados a essa célula receberão a mensagem.

### Atualizações em Voo

Uma vez transmitida, a mensagem pode ser:

- Atualizada à medida que a situação evolui
- Modificada com novo conteúdo do corpo da mensagem
- Revogada/parada a qualquer momento

### Registros Históricos

Todas as informações sobre mensagens históricas podem ser visualizadas e revisadas para fins de auditoria.

## Estrutura da Mensagem

Cada mensagem de Transmissão de Células consiste em:

### Configuração da Mensagem

- **Identificador da Mensagem** - Identificador único para o tipo de alerta (por exemplo, 4370 para ETWS Terremoto, 4371 para ETWS Tsunami)
- **Categoria** - Categoria do alerta (normal, emergência, alta, extrema)
- **Período de Repetição** - Segundos entre repetições de transmissão
- **Número de Transmissões** - Quantas vezes transmitir a mensagem
- **Período de Aviso** - Duração em segundos que o aviso é válido
- **Indicador de Canal** - Tipo de canal usado para transmissão

### Mensagens Localizadas

Cada mensagem do CBC pode incluir múltiplas variantes de idioma:

- **Idioma** - Código de idioma ISO (en, es, fr, zh, etc.)
- **Corpo da Mensagem** - Texto do alerta nesse idioma (até 1395 caracteres)

O sistema transmite automaticamente todas as variantes de idioma, permitindo

que os destinatários visualizem alertas em seu idioma preferido.

## Áreas de Rastreamento

Define o direcionamento geográfico para o alerta:

- **Área de Rastreamento** - Identificador geográfico (ID da célula, código da área de rastreamento)
- **Operador** - Código do operador de rede móvel (MCC-MNC)
- **Tipo de RAT** - Tecnologia de Acesso Rádio (LTE, 5G, UMTS, GSM)

Múltiplas áreas de rastreamento podem ser especificadas para cobrir regiões maiores ou múltiplos operadores.

## Criando uma Mensagem de Transmissão de Células

### Via UI Web:

1. **Navegar para Transmissão de Células** - Acesse a interface de gerenciamento do CBC a partir da navegação principal
2. **Clique em "Criar Alerta"** - Abre o formulário de criação de mensagem
3. **Configurar Parâmetros da Mensagem:**
  - Identificador da Mensagem (por exemplo, 4370 para alertas de terremoto)
  - Categoria (normal, alta, extrema)
  - Período de Repetição (tipicamente 5-60 segundos)
  - Número de Transmissões (999 para contínuo, ou contagem específica)
  - Período de Aviso (duração em segundos)
  - Indicador de Canal (tipicamente "básico")

Definição da Mensagem de Transmissão de Células

4. **Adicionar Mensagens Localizadas:**
  - Clique em "Adicionar Idioma"
  - Selecione o idioma no menu suspenso
  - Insira o texto da mensagem (máx 1395 caracteres para GSM7, menos para Unicode)
  - Repita para idiomas adicionais
5. **Definir Áreas de Rastreamento:**
  - Clique em "Adicionar Área de Rastreamento"
  - Insira o código da área de rastreamento
  - Selecione o operador (combinação MCC-MNC)
  - Escolha o tipo de RAT (LTE, 5G, etc.)
  - Repita para áreas geográficas adicionais

Seleção da Área de Rastreamento da Transmissão de Células

## 6. **Revisar e Criar** - Verifique todos os detalhes e clique em "Criar Alerta"

Revisão da Mensagem de Transmissão de Células

**Via API:**

**Endpoint:** PUT /crm/cbc/

**Permissão Necessária:** CREATE\_CBC\_MESSAGE

**Corpo da Solicitação:**

```
{
  "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"
    }
  ]
}
```

**Resposta:**

```
{
  "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": [...]
}
```

A mensagem é imediatamente enviada ao Centro de Transmissão de Células para transmissão.

## Gerenciando Mensagens Existentes

### Ver Todas as Mensagens

**Endpoint:** GET /crm/cbc/

**Permissão Necessária:** VIEW\_CBC\_MESSAGE

Retorna a lista de todas as mensagens do CBC com seu status, timestamps e configuração.

### Atualizar uma Mensagem

**Endpoint:** PATCH /crm/cbc/{cbc\_message\_id}

**Permissão Necessária:** UPDATE\_CBC\_MESSAGE

Atualiza o conteúdo da mensagem, áreas de rastreamento ou parâmetros de transmissão. Mensagens atualizadas são reenviadas ao CBC.

### Corpo da Solicitação:

```
{
  "cbc_message_id": 123,
  "numberOfBroadcasts": 500,
  "localized_messages": [
    {
      "language": "en",
      "messageBody": "UPDATED: Earthquake warning still in effect.
Aftershocks possible."
    }
  ]
}
```

```
]
}
```

## Excluir uma Mensagem

**Endpoint:** DELETE /crm/cbc/{cbc\_message\_id}

**Permissão Necessária:** DELETE\_CBC\_MESSAGE

Remove a mensagem do banco de dados e tenta cancelá-la no CBC.

## Fluxo de Aprovação

Mensagens de Transmissão de Células suportam um fluxo de aprovação opcional para alertas de alto risco:

1. **Usuário Iniciador** - Membro da equipe que cria o alerta (initiating\_user field)
2. **Usuário Aprovador** - Gerente que aprova o alerta antes da transmissão (approving\_user field)

Se a aprovação for necessária:

- A mensagem é criada com approving\_user = null
- A mensagem é mantida em estado "pendente de aprovação"
- O usuário aprovador revisa a mensagem e aprova ou rejeita
- Após a aprovação, approving\_user é definido e a mensagem é transmitida

Esse fluxo de trabalho é configurável com base na política da organização.

## Identificadores de Mensagem

Identificadores de mensagem padrão seguem 3GPP TS 23.041:

### Sistema de Aviso de Terremoto e Tsunami (ETWS):

- **4370** - Aviso de Terremoto ETWS
- **4371** - Aviso de Tsunami ETWS
- **4372** - Aviso Combinado de Terremoto e Tsunami ETWS
- **4373-4378** - Outros Tipos de Emergência ETWS
- **4379** - Mensagem de Teste ETWS

### Sistema de Alerta Móvel Comercial (CMAS) / Alertas de Emergência Sem Fio (WEA):

- **4352** - Alerta Presidencial
- **4353-4355** - Alertas Extremos

- **4356-4359** - Alertas Severos
- **4360-4363** - Alertas AMBER
- **4364-4367** - Mensagens de Segurança Pública
- **4368-4369** - Testes Estaduais/Locais
- **4380-4381** - Mensagens de Teste

### **Faixas Personalizadas:**

- **0-999** - Reservado para alertas específicos do operador
- **1000-4095** - Tipos de mensagens personalizadas

## **Integração com o Centro de Transmissão de Células**

A Entidade de Transmissão de Células (CBE) precisa de um mecanismo para entregar mensagens a redes celulares individuais. O Centro de Transmissão de Células (CBC) do OWL se conecta a cada rede celular para enviar mensagens de Aviso de Emergência ao público.

### **Redundância Multi-Rede**

No caso de um operador de rede celular estar indisponível (queda ou sem cobertura), se outra rede celular operante estiver disponível, os usuários ainda receberão mensagens de Aviso de Emergência através das outras redes disponíveis.

### **Instâncias CBC por Operador**

Para garantir que não haja conexão entre redes concorrentes, o OWL executa uma instância de CBC separada para cada MNO; não é compartilhada entre operadores.

## **Configuração do OWL CBC**

A URL do CBC é configurada em `crm_config.yaml`:

```
cbc_url: "http://cbc.example.com:8080"
```

### **Transmissão de Mensagens:**

Quando uma mensagem do CBC é criada ou atualizada:

1. O OmniCRM CBE armazena a mensagem em seu banco de dados
2. A mensagem é formatada para a API do CBC
3. Solicitação HTTP POST enviada para `{cbc_url}/alerts/send`
4. O CBC se conecta à rede celular através da interface apropriada (CBSP, SBc-AP ou N50)
5. O CBC reconhece o recebimento e começa a transmissão
6. Dispositivos nas áreas de rastreamento alvo recebem o alerta

## **Exclusão de Mensagem:**

Quando uma mensagem é excluída, o CBE tenta cancelá-la no CBC para interromper transmissões em andamento.

## **Pontos de Integração da Rede Celular**

Diferentes gerações de redes celulares (2G/3G/4G/5G) têm interfaces únicas para se conectar ao Centro de Transmissão de Células. Com base nas tecnologias usadas por cada rede celular, a interface correta deve ser configurada.

O Centro de Transmissão de Células do OWL suporta interfaces de Transmissão de Células 2G, 3G, 4G e 5G e integrou-se a vários componentes de rede celular comumente usados.

### **CBSP - 2G/3G - Controlador de Estação Base (BSC)**

A interface do Protocolo de Serviço de Transmissão de Células (CBSP) conecta o CBC ao Controlador de Estação Base (BSC) que controla estações base 2G (GSM).

- Usado para mensagens de Transmissão de Células 2G e 3G com implantações combinadas de Controlador de Rede de Rádio/Controlador de Estação Base
- Pode ser configurado como um arranjo de cliente ou servidor, dependendo do fornecedor do BSC
- Uma conexão deve ser estabelecida entre todos os BSCs na rede e o OWL CBC
- As interfaces são continuamente monitoradas com alertas para indicar se um link CBSP caiu

Nota: 3GPP definiu o Protocolo de Transmissão de Área de Serviço (SABP) para uso em implantações RNC autônomas. Isso pode ser usado se necessário quando o CBSP não é suportado para células 3G em um RNC/BSC combinado, no entanto, testes adicionais e suporte do fornecedor do RNC podem ser necessários.

### **SBc-AP - 4G/5G Não Autônomo - MME/IWF**

A interface SBc-AP conecta o OWL CBC ao MME (Entidade de Gerenciamento de Mobilidade) que atende 4G e 5G eNodeB/gNodeBs.

- Usado para redes 4G LTE
- Também usado para 5G Não Autônomo (a maioria das implantações em 2025)
- Uma conexão deve ser estabelecida entre todos os MMEs na rede e o OWL CBC
- As interfaces são continuamente monitoradas com alertas para indicar se um link SBc-AP caiu

### **N50 - 5G Autônomo - AMF**

Para redes 5G autônomas, a interface N50 conecta o OWL CBC ao AMF (Função de Gerenciamento de Acesso e Mobilidade) que atende 5G gNodeBs.

- A interface está presente no OWL CBC
- Não foi extensivamente testada com AMFs de terceiros devido ao pequeno número de redes 5G SA comercialmente disponíveis em 2025
- Será totalmente suportada à medida que as implantações 5G SA se tornem mais comuns

## **Requisitos de Rede do MNO**

A rede deve estar em vigor entre o OWL CBC e a rede do Operador de Rede Móvel para alcançar as interfaces descritas acima.

Isso é tratado caso a caso, mas geralmente requer:

- Conexão cruzada/fibra dedicada entre o CBC e a rede do MNO
- Cada interface logicamente separada
- Conectividade para cada ponto de integração (MME, RNC, BSC) na rede celular

## **Equipamentos de Rede Suportados**

O OWL CBC foi testado e integra-se a componentes de rede celular comumente usados de grandes fornecedores:

### **Integração de Dados do Site de Celular**

O OWL suporta a coleta automática de dados de:

- **Nokia NetAct**
- **Huawei U2000 / U2020**
- **ZTE NetNumen / ZXPOS**
- **Ericsson ENM**

Alternativamente, os dados do site de celular podem ser fornecidos à equipe de operações do Omnitouch periodicamente via e-mail.

## **Gerenciamento de Usuários e Controle de Acesso**

### **Controle de Acesso Baseado em Funções (RBAC)**

O sistema OWL utiliza controle de acesso baseado em funções (RBAC): as pessoas (Usuários Registrados) são atribuídas a uma ou mais Funções, e cada Função é um conjunto de Permissões. As Permissões são a menor unidade de acesso (por exemplo, criar alerta de mensagem em rascunho). O acesso efetivo de um Usuário Registrado é a união das Permissões de todas as Funções atribuídas.



## Componentes do RBAC:

- **Usuários** - Pessoas reais que fazem login no sistema OWL
- **Permissões** - Micro capacidades (por exemplo, aprovar rascunho de mensagem, criar mensagem, visualizar relatórios)
- **Funções** - Conjuntos nomeados de permissões (por exemplo, Aprovadores de Mensagens, Criadores de Mensagens)
- **Atribuição** - Usuários recebem uma ou mais Funções; permissões se agregam

## Benefícios do RBAC:

1. **Proteção de Dados** - Usuários veem e fazem apenas o que estão autorizados
2. **Adequação Operacional** - Funções refletem funções de trabalho (Admin, Criador de Mensagens, Aprovador de Mensagens)
3. **Administração Simples** - Conceder acesso atribuindo funções; evitar microgerenciamento por usuário

## Permissões do Sistema

As permissões do sistema geralmente seguem padrões CRUD com quatro opções:

- **Visualizar** - Ler ou navegar por mensagens e relatórios
- **Criar** - Criar ou adicionar um alerta de mensagem
- **Atualizar** - Editar ou modificar um alerta de mensagem em rascunho
- **Excluir** - Excluir ou remover um alerta de mensagem em rascunho

## Permissões Básicas do CBC:

- `CREATE_CBC_MESSAGE` - Criar novas mensagens de transmissão
- `VIEW_CBC_MESSAGE` - Visualizar mensagens existentes e seu status
- `UPDATE_CBC_MESSAGE` - Modificar o conteúdo da mensagem ou parâmetros de transmissão
- `DELETE_CBC_MESSAGE` - Excluir mensagens e cancelar transmissões

Atribua essas permissões às funções com base nas responsabilidades de segurança pública da sua organização.

Para informações abrangentes sobre gerenciamento de funções e permissões de usuários, veja `Usuários` e `Funções <rbac>`.

## Gerenciamento de Contas de Usuário

### Primeiro Login

Quando um novo usuário é configurado no sistema, ele recebe credenciais de login pelo Administrador do Sistema:

1. Navegue até a URL de login do OWL fornecida pelo seu Administrador do Sistema
2. Insira seu nome de usuário e senha
3. Se o 2FA estiver habilitado para sua conta, você será solicitado a configurá-lo (veja a seção 2FA acima)

**Melhor Prática:** Os usuários devem alterar sua senha inicial imediatamente após o primeiro login por segurança. Navegue até as configurações do seu perfil de usuário para alterar sua senha.

Para mais detalhes sobre fluxos de autenticação, veja Fluxos de Autenticação e Controles de Admin <authentication\_flows>.

## **Redefinição de Senha**

Os usuários podem redefinir suas próprias senhas:

1. Na página de login do sistema, selecione "Esqueci a Senha"
2. Siga o processo de verificação por e-mail
3. Defina uma nova senha

## **Redefinição de Senha do Admin:**

Os Administradores do Sistema podem redefinir senhas para usuários na página **Usuários e Funções → Usuários**:

1. Navegue até **Usuários e Funções → Usuários**
2. Selecione o usuário que precisa de redefinição de senha
3. Clique em **Redefinir Senha**
4. O sistema gera uma senha temporária segura e aleatória (12 caracteres)
5. A senha temporária é exibida para o admin
6. Forneça a senha temporária ao usuário por meio de um canal seguro (telefone ou pessoalmente, nunca por e-mail)
7. O usuário pode fazer login com a senha temporária
8. **Importante:** O usuário deve alterar essa senha imediatamente após fazer login

Para mais detalhes, veja Usuários e Funções <rbac>.

## **Adicionando Novos Usuários ao Sistema**

Os Administradores do Sistema podem adicionar novos usuários navegando até **Usuários e Funções → Usuários**:

Informações necessárias para cada novo Usuário Registrado:

- Primeiro Nome
- Último Nome
- E-mail

- Senha (temporária)
- Número de Celular
- Função(s)

**Melhor Prática:** Implemente um Formulário de Solicitação de Novo Usuário do Sistema que esteja alinhado com as políticas e procedimentos existentes da sua organização. Adicione usuários apenas após a devida aprovação.

## **Editando Perfis de Usuário**

Os Administradores do Sistema podem alterar permissões e acesso para usuários existentes:

1. Navegue até **Usuários e Funções → Usuários**
2. Selecione o usuário a ser editado
3. Modifique as funções atribuídas conforme necessário
4. Salve as alterações

## **Excluindo Usuários do Sistema**

Os Administradores do Sistema podem excluir usuários de **Usuários e Funções → Usuários**.

**Aviso:** Uma vez que um usuário foi excluído do sistema OWL, a ação não pode ser desfeita. Tenha muito cuidado antes de executar essa tarefa.

## **Visualizando Todos os Usuários do Sistema**

A qualquer momento, o Administrador do Sistema pode visualizar uma lista de todos os Usuários Registrados e ver quais permissões e funções cada um possui navegando até **Usuários e Funções → Usuários**.

# **Histórico de Mensagens de Alerta**

O sistema OWL mantém um registro de todas as mensagens de alerta enviadas para fins de auditoria e para apoiar a revisão operacional.

## **Acessando o Histórico de Mensagens:**

Navegue até **Gerenciar Transmissões de Células** para visualizar:

- Todas as mensagens enviadas anteriormente
- Status da mensagem (ativa, expirada, cancelada)
- Timestamp da transmissão
- Usuários iniciadores e aprovadores
- Detalhes do direcionamento geográfico
- Conteúdo e parâmetros da mensagem

Isso ajuda com:

- Auditoria do sistema
- Revisão dos procedimentos operacionais usados pelo NDMO e outras agências de alerta
- Compreensão da eficácia da mensagem
- Requisitos de conformidade e relatórios

## Visualização do Mapa de Transmissão de Células

O Mapa de Transmissão de Células fornece uma **interface visual** para planejar áreas de cobertura geográfica e identificar quais torres de celular irão transmitir alertas.

### Acessando o Mapa

Navegue até:

Ou diretamente:

**Permissão Necessária:** VIEW\_CBC\_MESSAGE ou CREATE\_CBC\_MESSAGE

### Recursos do Mapa

O mapa exibe:

#### Localizações das Torres de Celular:

- **Ícones de Torres** - Posições dos sites de celular exibidas como ícones de torres de celular
- **Indicadores de Setor** - Setas direcionais mostrando a orientação do setor da antena
- **Nomes das Torres** - Identificadores e nomes do site
- **Dados de Cobertura** - Carregados do arquivo `cellSites.txt`

#### Ferramentas de Desenho:

- **Ferramenta de Polígono** - Desenhar áreas de cobertura personalizadas
- **Ferramenta de Círculo** - Criar zonas de transmissão circulares
- **Ferramenta de Retângulo** - Definir áreas de cobertura retangulares

#### Análise de Cobertura:

- **Lista de Sites** - Tabela mostrando todos os sites de celular dentro da área desenhada
- **Gerenciamento de Zonas** - Salvar e carregar zonas de cobertura pré-definidas
- **Contagem de Sites** - Contagem em tempo real de torres na área de

cobertura

## Usando o Mapa

### Passo 1: Carregar Dados do Site de Celular

O mapa carrega automaticamente as localizações das torres de celular do `/cellSites.txt` ao carregar a página. Este arquivo contém:

- **lat/lng** - Coordenadas da torre
- **site\_name** - Identificador
- **sectors** - Direções da antena em graus

### Passo 2: Desenhar Área de Cobertura

Selecione uma ferramenta de desenho:

1. Clique em **Polígono** para desenhar formas personalizadas
  - Clique para adicionar pontos
  - Clique duas vezes para concluir
  - Cria limites de cobertura precisos
2. Clique em **Círculo** para desenhar áreas circulares
  - Clique no ponto central
  - Arraste para definir o raio
  - Útil para alertas rápidos baseados em raio
3. Clique em **Retângulo** para desenhar áreas retangulares
  - Clique em um canto
  - Arraste para o canto oposto
  - Eficiente para cobertura alinhada à grade

### Passo 3: Revisar Sites Cobertos

Após desenhar, o sistema:

- Calcula quais sites de celular caem dentro da área
- Exibe a lista na tabela abaixo do mapa
- Mostra nomes e coordenadas dos sites
- Fornece contagem de torres afetadas

### Tabela de Cobertura de Exemplo:

### Passo 4: Exportar Dados de Cobertura

Use os dados de cobertura para:

- **Identificar códigos de área de rastreamento** para a transmissão
- **Estimar alcance do alerta** com base na contagem de torres
- **Planejar resposta a emergências** entendendo áreas afetadas
- **Salvar definições de zona** para uso futuro

## Integração de Fluxo de Trabalho

A visualização do mapa integra-se com a criação de transmissões:

### Planejando um Alerta:

1. **Abrir Mapa do CBC** para visualizar a área afetada
2. **Desenhar zona de cobertura** usando a ferramenta de polígono
3. **Revisar sites cobertos** e ajustar limites
4. **Anotar códigos de área de rastreamento** para sites de celular afetados
5. **Criar mensagem do CBC** com direcionamento apropriado

### Exemplo: Alerta de Tsunami Costeiro

## Gerenciamento de Zonas

### Salvando Zonas:

Zonas de cobertura pré-definidas podem ser salvas para cenários comuns:

- **Região Costeira** - Para alertas de tsunami/onda de tempestade
- **Zonas de Incêndio Florestal** - Para alertas de evacuação
- **Centros Urbanos** - Para alertas AMBER em áreas populosas
- **Corredores de Rodovia** - Para alertas de tráfego/clima

### Formato de Dados da Zona:

Zonas são armazenadas em `site_data.json`:

```
[
  {
    "area": "Região Costeira",
    "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}
        ]
      }
    ]
  }
]
```

### Carregando Zonas Salvas:

1. Selecione a zona no menu suspenso

2. O polígono é exibido automaticamente no mapa
3. A lista de sites é atualizada para mostrar torres cobertas
4. Modifique o polígono se necessário

## Configuração do Mapa

### API do Google Maps:

O mapa requer uma chave da API do Google Maps configurada no ambiente:

```
REACT_APP_GOOGLE_API_KEY=your_google_maps_api_key_here
```

### Dados do Site de Celular:

Atualize `public/cellSites.txt` com as localizações das torres de celular da sua rede:

```
# Formato: lat,lng,nome,setor1,setor2,setor3
-33.8688,151.2093,SYD_CENTRAL,0,120,240
-33.8650,151.2070,SYD_HARBOUR,45,165,285
```

Obtenha as coordenadas do site de celular de:

- Ferramentas de planejamento de rede
- Registros de instalação de torres
- Pesquisas de campo com GPS
- Configuração do controlador de rede de rádio (RNC)

### Estilização do Mapa:

Estilos de mapa personalizados podem ser configurados em `mapStyles.js` para:

- Destacar serviços de emergência
- Mostrar características topográficas
- Enfatizar centros populacionais
- Combinar com a marca da organização

## Casos de Uso

### Planejamento de Emergência:

- Pré-planejar zonas de cobertura para áreas de risco conhecidas
- Testar diferentes estratégias de direcionamento
- Estimar o alcance populacional
- Coordenar com serviços de emergência

### Verificação de Alertas:

- Confirmar que a mensagem alcançará a área pretendida

- Identificar lacunas de cobertura
- Evitar sobre-alertas em regiões adjacentes
- Validar códigos de área de rastreamento

### **Análise de Rede:**

- Visualizar a distribuição de torres de celular
- Identificar sobreposições de cobertura
- Planejar implantações de torres para melhor cobertura de alertas
- Otimizar a eficiência da transmissão

## **Melhores Práticas**

Com base na experiência com clientes ao redor do mundo, a Omnitouch recomenda as seguintes melhores práticas para todas as implantações do OWL.

### **Conteúdo da Mensagem:**

- Mantenha as mensagens concisas e claras (menos de 360 caracteres para alertas de uma página)
- Use todas as letras maiúsculas para alertas de emergência (melhora a legibilidade)
- Inclua itens de ação específicos ("Evacue imediatamente", "Busque abrigo")
- Evite jargões técnicos
- Teste mensagens com dispositivos reais antes das emergências

### **Suporte a Idiomas:**

- Sempre forneça mensagens no idioma principal da região
- Inclua idiomas adicionais para áreas multiculturais
- Garanta que as traduções sejam culturalmente apropriadas
- Teste caracteres especiais e suporte a Unicode

### **Direcionamento Geográfico:**

- Use as menores áreas de rastreamento necessárias para evitar fadiga de alertas
- Considere a densidade populacional ao definir o período de repetição
- Teste o direcionamento geográfico antes das emergências
- Mantenha documentação precisa das áreas de rastreamento

### **Testes:**

- Use identificadores de mensagem de teste (4379, 4380-4381) para simulações
- Programe testes regulares do sistema
- Verifique se a integração do CBC está funcionando
- Treine a equipe sobre procedimentos de emergência



## **Fadiga de Alertas:**

- Use apenas para emergências genuínas
- Evite sobre-transmissões
- Defina períodos de repetição apropriados
- Use níveis de severidade de forma adequada

## **Segurança da Conta do Usuário**

### **Segurança do Token 2FA:**

- Proteja tokens 2FA físicos como cartões de acesso a edifícios
- Relate tokens perdidos ou roubados imediatamente
- Teste o 2FA regularmente para garantir que esteja funcionando
- Salve códigos de backup ao configurar o 2FA e armazene-os de forma segura offline
- Entre em contato com o Administrador do Sistema se precisar redefinir o 2FA

Para mais informações sobre configuração e recuperação do 2FA, veja Autenticação de Dois Fatores <2fa>.

## **Manutenção de Dados**

### **Áreas de Alvo Pré-definidas:**

À medida que os limites geográficos mudam, o desenvolvimento ocorre e as áreas de risco mudam, há necessidade de revisar as Áreas de Alvo Pré-definidas. A Omnitouch sugere que esses dados sejam revisados **anualmente pelo NDMO**, com apoio de outras agências de alerta, quando aplicável.

Atualizações aos limites podem ser definidas usando várias plataformas GIS comuns ou Google Earth, e depois fornecidas via e-mail à equipe de operações do Omnitouch, que fará as alterações no sistema.

### **Modelos de Mensagem Pré-definidos:**

Os modelos de mensagem pré-definidos devem ser revisados **pelo menos anualmente pelo NDMO**, com apoio de outras agências de alerta, quando aplicável, para garantir:

- O conteúdo da mensagem ainda reflete com precisão o perigo e a chamada à ação
- As informações de contato e instruções estão atualizadas
- As traduções de idiomas permanecem precisas
- O tom e a urgência da mensagem são apropriados
- Referências a serviços de emergência ou procedimentos estão atualizadas

Atualizações podem ser fornecidas via e-mail à equipe de operações do Omnitouch, que fará as alterações no sistema.

### **Dados do Site de Celular:**

- Revise e atualize os dados do site de celular sempre que ocorrerem mudanças na rede
- Frequência típica de atualização: mensal ou trimestral
- Coordene com as equipes de planejamento de rede do MNO
- Verifique a precisão após grandes atualizações ou expansões da rede

## **Arquitetura do Sistema**

Todos os produtos da Omnitouch são projetados para suportar implantações geograficamente distribuídas.

### **Opções de Implantação**

Todos os componentes podem ser executados como:

- **Contêineres (K8s)** - Implantações de contêineres orquestradas pelo Kubernetes
- **Máquinas Virtuais** - VMware, Proxmox, HyperV
- **Nuvem Privada** - Infraestrutura de nuvem local
- **Nuvem Pública** - AWS, GCP
- **Bare Metal** - Implantação direta em hardware

### **Arquitetura Distribuída**

A arquitetura distribuída permite:

- Escritórios Locais de Gestão de Desastres acessarem o OWL CBE e distribuírem mensagens mesmo se uma região se tornar isolada da rede nacional
- Múltiplas Entidades de Transmissão de Células e múltiplos Centros de Transmissão de Células por operador/país
- Agências locais de resposta a desastres (Governo Municipal, Polícia, Bombeiros, etc.) emitirem alertas para sua região mesmo se o principal NDMO (Escritório Nacional de Gestão de Desastres) perder acesso

Isso é particularmente valioso quando os MNOs distribuíram sua rede celular com recursos locais de BSC/MME.

### **Networking do CBE e CBC**

Para garantir que não haja conexão entre redes concorrentes, cada MNO tem uma instância de CBC separada (não compartilhada).

Requisitos de rede entre CBE e CBC:

- Todo o tráfego é criptografado entre CBE e CBC
- Autenticação baseada em certificados mútuos
- Conectividade do CBE ao CBC via TLS na porta TCP 443
- Coordenação necessária entre MNOs e o NDMO ou agência que hospeda o CBE

### **Considerações de Acesso**

A agência que hospeda o CBE precisará definir procedimentos de acesso para usuários finais (ou seja, Citrix, VPN, etc.), tendo em mente que o sistema deve ser acessível em cenários não ideais, como:

- Quedas em larga escala de redes de energia pública
- Falhas na rede de telecomunicações
- Desastres naturais afetando a infraestrutura

## **Requisitos de Implantação**

### **Requisitos de VM do CBE (NDMO / Governo)**

#### **3x Máquinas Virtuais:**

- **2x VMs do CBE** (Idealmente em diferentes datacenters/zones de disponibilidade)
- **1x VM de Monitoramento**

Cada VM requer:

- **Armazenamento:** 50GB
- **CPU:** 2x CPU Virtual
- **RAM:** 8GB
- **OS:** Sistema Operacional Base fornecido pela Omnitouch
- **Rede:** Permitir tráfego para VMs do CBC na porta TCP 443 para tráfego TLS para controlar os CBCs

### **Requisitos de VM do CBC (MNO)**

#### **3x Máquinas Virtuais:**

- **2x VMs do CBC** (Idealmente em diferentes datacenters/zones de disponibilidade)
- **1x VM de Monitoramento**

Cada VM requer:

- **Armazenamento:** 50GB

- **CPU:** 2x CPU Virtual
- **RAM:** 8GB
- **OS:** Sistema Operacional Base fornecido pela Omnitouch
- **Conectividade:** Para cada um dos Pontos de Integração na rede celular (MME, RNC, BSC)
- **Rede:** Permitir tráfego das VMs do CBE na porta TCP 443 para tráfego TLS a ser controlado pelos CBEs

## Etapas de Integração

Implantar o OWL envolve as seguintes etapas:

1. **Fonte de novo(s) operador(es)** - Identificar operadores de rede móvel participantes
2. **Administração do projeto (configuração)** - Estabelecer governança e administração do projeto
3. **Selecionar local de hospedagem para o CBE** - Determinar onde o CBE será hospedado
4. **Definir Usuários & Fluxos/Procedimentos de Mensagens** - Estabelecer fluxos de aprovação e funções de usuários
5. **Definir Polígonos para Áreas de Alvo e Modelos de Mensagem** - Pré-configurar cenários comuns
6. **Alocações de Endereço IP** - Alocar endereços IP para NDMO e MNOs
7. **Configurar VPN site-a-site** - Estabelecer conexão segura com a equipe da Omnitouch
8. **Implantar VMs do CBE para NDMO** - Instalar Entidade de Transmissão de Células
9. **Implantar VMs do CBC para MNO** - Instalar Centro de Transmissão de Células em cada operador
10. **Configurar Elementos de Rede** - Configurar conectividade ao CBC a partir do equipamento de rede celular
11. **Networking entre VMs do CBE e CBC** - Estabelecer comunicação segura
12. **Networking entre VMs do CBC e Elementos de Rede** - Conectar ao BSC/MME/AMF
13. **Configurar acesso à API para NMS** - Configurar integração de dados do site de celular
14. **Configuração de monitoramento & testada** - Verificar monitoramento e alertas
15. **Verificação/Teste de alertas de teste** - Realizar testes do sistema
16. **Teste de aviso público** - Realizar teste público de ponta a ponta

## Limites de Caracteres

As mensagens de Transmissão de Células têm limites de caracteres rigorosos com base na codificação:

### **Codificação GSM de 7 bits (inglês, caracteres latinos básicos):**

- Página única: 93 caracteres
- Múltiplas páginas: 15 páginas × 93 = 1395 caracteres no máximo

### **Codificação Unicode UCS-2 (scripts não latinos, emojis):**

- Página única: 41 caracteres
- Múltiplas páginas: 15 páginas × 41 = 615 caracteres no máximo

### **Plataforma OWL:**

- Texto da mensagem limitado a 500 caracteres
- UI Web exibe a contagem de caracteres restantes e avisa ao se aproximar dos limites

## **Monitoramento e Logs**

A atividade de Transmissão de Células é registrada para fins de auditoria:

- Eventos de criação, atualização e exclusão de mensagens
- Usuários iniciadores e aprovadores
- Timestamps e identificadores de mensagens
- Respostas e erros da API do CBC
- Detalhes de direcionamento geográfico

Acesse logs via o Registro de Atividades ou consultas ao banco de dados:

```
SELECT * FROM cbc  
WHERE created >= '2025-01-01'  
ORDER BY created DESC;
```

## **Integração com Dispositivos Móveis**

As mensagens de Transmissão de Células são recebidas por dispositivos móveis compatíveis:

### **Suporte a Dispositivos:**

- A maioria dos smartphones a partir de 2015 suporta Transmissão de Células
- Telefones básicos podem ter suporte limitado
- O dispositivo deve estar conectado à rede (sem créditos de dados/SMS necessários)
- Funciona mesmo durante congestionamento da rede quando o SMS falha

### **Experiência do Usuário:**

- O alerta é exibido como uma notificação em tela cheia
- Um tom de alerta único toca
- O alerta persiste até ser reconhecido
- Nenhuma assinatura de usuário é necessária
- Não pode ser bloqueado por usuários para alertas presidenciais/extremos

### **Testando a Recepção do Dispositivo:**

Para verificar se os dispositivos podem receber alertas:

1. Envie uma mensagem de teste (identificador 4379 ou 4380-4381)
2. Certifique-se de que o dispositivo esteja na área de rastreamento alvo
3. Verifique se o dispositivo tem a Transmissão de Células ativada nas configurações
4. Verifique com vários modelos de dispositivos e versões de SO

## **Funcionalidade Adicional**

A plataforma OWL pode ser estendida com recursos opcionais para complementar a transmissão de mensagens de Células:

### **SMS em Massa**

Envio de SMS regulares para indivíduos para notificações suplementares.

- Frequentemente usado para alertas rebaixados para informar as pessoas que a ameaça imediata passou, de uma maneira menos intrusiva
- Muito mais lento que a Transmissão de Células, mas pode incluir um mecanismo de confirmação para verificar se a mensagem foi recebida
- Útil para comunicações de acompanhamento direcionadas

### **Chamadas de Voz para Linha Fixa**

Ligar automaticamente para números de linha fixa e reproduzir mensagens de aviso de emergência.

- Reprodução de mensagem pré-gravada
- Reprodução de mensagem de aviso de emergência em texto para fala
- Alcança populações sem telefones móveis
- Pode verificar a entrega da mensagem através da conclusão da chamada

### **Postagem Cruzada em Mídias Sociais**

Publicar automaticamente mensagens de Aviso de Emergência em canais oficiais de mídias sociais.

- Estende o alcance além da rede celular

- Fornece referência para aqueles que perderam o alerta inicial
- Permite mensagens estendidas além dos limites de caracteres

## **Transmissão Automática de Rádio/TV**

Transmitir automaticamente mensagens de emergência via rádio e televisão.

- Reprodução de mensagem pré-gravada
- Reprodução de mensagem de aviso de emergência em texto para fala
- Alcança populações durante quedas de rede
- Complementa a Transmissão de Células para cobertura abrangente

## **Dispositivos de Alerta Externos**

O OWL CBC pode se conectar a uma variedade de fontes externas:

- Plataformas de mídias sociais
- APIs públicas
- Sistemas de chamadas de voz
- Sistemas de transmissão de rádio
- Alarmes físicos e sirenes
- Sinalização eletrônica

Opções de personalização podem ser exploradas como parte da fase de design.

## **Personalização e Manutenção**

### **Testes Periódicos**

Testes periódicos da solução devem ser realizados em intervalos regulares para garantir:

- A solução e todos os componentes estão funcionando corretamente
- Todos os funcionários estão familiarizados com os processos e procedimentos necessários para emitir Mensagens de Aviso de Emergência
- Os pontos de integração permanecem operacionais
- Os modelos de mensagem estão atualizados e eficazes

### **Testes Recomendados:**

- Mensagens de teste mensais usando identificadores 4379, 4380-4381
- Testes completos do sistema trimestrais, incluindo fluxos de aprovação
- Testes anuais de aviso público com notificação antecipada
- Sessões de treinamento regulares para usuários autorizados

## **Manutenção de Dados do Site de Celular**

Quando os operadores adicionam ou removem sites de celular, ou alteram Áreas/ Identificadores de Rastreamento de sites de celular, essas informações devem ser compartilhadas com a equipe do Omnitouch para garantir que os dados da ferramenta de mapeamento permaneçam precisos.

### **Integração Automática de Dados**

O OWL suporta a coleta automática de dados de:

- Nokia NetAct
- Huawei U2000 / U2020
- ZTE NetNumen / ZXPOS
- Ericsson ENM

### **Atualizações Manuais de Dados**

Alternativamente, os dados do site de celular podem ser fornecidos à equipe de operações do Omnitouch periodicamente via e-mail em vários formatos.

**Frequência de Atualização:** Revise e atualize os dados do site de celular sempre que ocorrerem mudanças na rede, tipicamente mensal ou trimestral.

### **Áreas de Alvo Pré-definidas**

À medida que os limites geográficos mudam, o desenvolvimento ocorre e as áreas de risco mudam, há necessidade de revisar as Áreas de Alvo Pré-definidas usadas na etapa de Direcionamento.

**Revisão Anual:** As áreas de alvo pré-definidas devem ser revisadas anualmente pelo NDMO (Escritório Nacional de Gestão de Desastres).

**Processo de Atualização:** Atualizações aos limites podem ser definidas usando várias plataformas GIS comuns ou Google Earth, e depois fornecidas à equipe de operações do Omnitouch.

### **Modelos de Mensagem Pré-definidos**

**Revisão Anual:** Os modelos de mensagem pré-definidos devem ser revisados anualmente pelo NDMO para garantir:

- O conteúdo da mensagem reflete os procedimentos de emergência atuais
- As traduções de idiomas são precisas
- Os identificadores de mensagem são apropriados
- As informações de contato e instruções estão atualizadas

**Processo de Atualização:** Atualizações podem ser fornecidas via e-mail à equipe



de operações do Omnitouch.

## **Fluxos de Aprovação de Mensagens**

Diferentes regiões têm diferentes requisitos em relação à emissão de mensagens e fluxos de aprovação.

**Regra de Duas Pessoas:** O uso da regra de duas pessoas é defendido sempre que prático para garantir supervisão na submissão de mensagens.

**Funções de Usuário Granulares:** Funções de usuário individuais podem ser configuradas para:

- Permitir que apenas certos usuários enviem mensagens pré-definidas
- Restringir o direcionamento a regiões específicas
- Exigir etapas de aprovação adicionais
- Minimizar o risco de uso indevido

## **Manutenção de 2FA / Segurança**

A equipe de operações do Omnitouch pode apoiar com:

- Redefinição de tokens de 2FA
- Reemissão de tokens perdidos/danificados/expirados
- Auditoria de segurança e gerenciamento de tokens

### **Procedimentos de Segurança do Token:**

Quando os tokens são emitidos, um procedimento é detalhado para os passos a serem tomados caso um token se torne perdido ou não contabilizado. A pessoa aceitando o token deve seguir esses procedimentos para garantir que o sistema não seja abusado.

### **Ações Imediatas para Tokens Perdidos:**

1. Relate o token perdido imediatamente ao pessoal autorizado
2. O token é desativado no sistema
3. Revisão de segurança conduzida
4. Novo token emitido após verificação de segurança

## **Integração de API Externa**

Um conjunto completo de APIs está disponível para a Entidade de Transmissão de Células para permitir que sistemas de terceiros integrem e interajam com o CBE.

### **Capacidades da API:**

- **Relatórios/Monitoramento** - Verificar status e alcance das mensagens

transmitidas

- **Criação de Mensagens** - Criar e transmitir novas mensagens programaticamente
- **Verificações de Saúde** - Verificar periodicamente a saúde do sistema com geração de tráfego de mensagens de teste
- **Consultas de Status** - Recuperar status de mensagens, estatísticas de entrega e métricas do sistema

Veja a seção de Referência da API abaixo para documentação detalhada dos endpoints.

## Referência da API

Todos os endpoints do CBC requerem autenticação e permissões apropriadas.

### **Criar Mensagem:**

PUT /crm/cbc/

### **Obter Todas as Mensagens:**

GET /crm/cbc/

### **Atualizar Mensagem:**

PATCH /crm/cbc/{cbc\_message\_id}

### **Excluir Mensagem:**

DELETE /crm/cbc/{cbc\_message\_id}

Veja a documentação do Swagger em /crm/docs/ para especificações detalhadas da API.



# Pesquisa Global

O recurso de Pesquisa Global fornece uma **interface de pesquisa unificada** para encontrar rapidamente clientes, contatos, serviços, inventário e sites em todo o banco de dados do OmniCRM.

Veja também: Customers <basics\_customers>, Inventory <administration\_inventory>, Service Management <csa\_service\_management>.

## Acessar a Pesquisa Global

### De qualquer lugar no CRM:

Clique no ícone de pesquisa na barra de navegação superior ou navegue para:

A página de pesquisa global aparece com uma grande caixa de pesquisa e opções de filtro.

## Como Funciona

A pesquisa global realiza uma **pesquisa entre entidades** em cinco tipos de dados:

### O que é Pesquisado:

1. **Clientes** - Nome do cliente
2. **Contatos** - Primeiro nome, sobrenome, endereço de e-mail, número de telefone
3. **Sites** - Nome do site
4. **Inventário** - Números de série, ICCIDs, identificadores (itemtext1, itemtext2)
5. **Serviços** - Nome do serviço, UUID do serviço

### Comportamento da Pesquisa:

- **Correspondência parcial** - Pesquisa por termos que contenham sua consulta (por exemplo, "Smith" corresponde a "John Smith" e "Smithson")
- **Sem distinção entre maiúsculas e minúsculas** - "john" corresponde a "John", "JOHN" e "john"
- **Múltiplas entidades** - Uma única pesquisa retorna resultados de todos os tipos de entidade
- **Resultados paginados** - Mostra 10 resultados por página por padrão

## Realizando uma Pesquisa

### Pesquisa Básica

1. Digite seu termo de pesquisa na caixa de pesquisa
2. Clique em "**Pesquisar**" ou pressione Enter

Resultados da Pesquisa Global

### Exemplos de termos de pesquisa:

- Nome do cliente: "Acme Corp"
- Número de telefone: "+1234567890" ou "1234567890"

- E-mail: "john@example.com" ou "john"
- Número de série: "ICCID8944" ou apenas "8944"
- UUID do serviço: "123e4567-e89b"

## Incluir Contas Fechadas

Por padrão, a pesquisa retorna apenas resultados de contas de clientes **Abertas**.

Para pesquisar em todas as contas, incluindo as fechadas:

1. Marque a caixa de seleção **"Incluir Contas Fechadas"**
2. Clique em **"Pesquisar"** novamente

Isso irá pesquisar:

- Clientes com `customer_status = "Closed"`
- Contatos, serviços, sites e inventário vinculados a clientes fechados

## Casos de uso para pesquisa de contas fechadas:

- Encontrar registros históricos de clientes
- Localizar equipamentos de serviços desprovisionados
- Consultar números de telefone ou serviços antigos
- Recuperar dados de clientes para reativação

# Entendendo os Resultados da Pesquisa

## Formato de Exibição dos Resultados

Os resultados são exibidos em uma lista rolável mostrando:

John Smith ID do Cliente: 123 Tipo: cliente

John Smith (Contato) ID do Cliente: 123 Tipo: contato

Móvel - +44 7700 900123 ID do Cliente: 123 Tipo: serviço

## Cada resultado mostra:

- **Nome/Título** - O identificador principal (link clicável)
- **ID do Cliente** - O cliente pai ao qual pertence
- **Tipo** - O tipo de entidade (cliente, contato, site, inventário, serviço)

## Tipos de Resultados Explicados

### Resultados de Clientes:

Clicar abre a página de visão geral do cliente mostrando todos os detalhes, serviços, contatos, etc.

### Resultados de Contatos:

Clicar abre a página do cliente com a aba Contatos ativa, rolando para o contato específico.

### Resultados de Sites:

Clicar abre a página do cliente com a aba Sites ativa.

## Resultados de Inventário:

Clicar abre a página do cliente com a aba Inventário ativa. Se o inventário não estiver atribuído (sem `customer_id`), ele vincula à lista principal de inventário.

## Resultados de Serviços:

Clicar abre a página do cliente com a aba Serviços ativa, destacando o serviço específico.

## Navegação a partir dos Resultados

Todos os resultados da pesquisa são **links clicáveis** que navegam diretamente para a página relevante:

### Padrão de Link:

- `/customers/{customer_id}` - Registros de clientes
- `/customers/{customer_id}#4` - Contatos (aba 4)
- `/customers/{customer_id}#2` - Sites (aba 2)
- `/customers/{customer_id}#8` - Inventário (aba 8)
- `/customers/{customer_id}#3` - Serviços (aba 3)
- `/inventory-items-list` - Inventário não atribuído

O fragmento de hash (#) seleciona automaticamente a aba correta quando a página do cliente é carregada.

## Paginação

Os resultados são paginados com 10 itens por página:

Mostrando resultados 11-20 de 47

Navegue pelas páginas usando:

- Botões **Anterior/Próximo**
- **Números de página** - Clique na página específica
- **Teclado** - Setas esquerda/direita (se implementado)

## Cenários Comuns de Pesquisa

### Cenário 1: Encontrar Cliente pelo Telefone

O usuário liga, fornece o número de telefone.

Resultados: • John Smith (Contato) - ID do Cliente: 123 • Móvel - 555-0123 (Serviço) - ID do Cliente: 123

Clique em qualquer resultado para acessar a conta do cliente.

### Cenário 2: Localizar Cartão SIM

O técnico precisa encontrar qual cliente possui um SIM específico.

Resultados: • 8944538000000001234 (Inventário) - ID do Cliente: 456

Clique no resultado para ver a atribuição do SIM, detalhes do cliente.

### Cenário 3: Encontrar Cliente Inativo

É necessário localizar um cliente que fechou sua conta há 6 meses.

☒ Incluir Contas Fechadas

Resultados: • Acme Corporation (cliente) - ID do Cliente: 789

### Cenário 4: Pesquisar por E-mail

O cliente envia um e-mail para o suporte, a equipe precisa encontrar sua conta.

Resultados: • John Smith (Contato) - ID do Cliente: 123

### Cenário 5: Encontrar Serviço pelo UUID

O log de provisionamento mostra o UUID do serviço, é necessário encontrar qual cliente.

Resultados: • Móvel - +44 7700 900123 (Serviço) - ID do Cliente: 456

## Dicas de Pesquisa

### Para Melhores Resultados:

- **Use termos parciais** - "Smith" é melhor do que "John Smith" para resultados mais amplos
- **Tente variações** - Se "John" não funcionar, tente telefone ou e-mail
- **Inclua contas fechadas** - Ao pesquisar dados históricos
- **Seja específico para equipamentos** - Use números de série completos para inventário
- **Pesquise UUID do serviço** - Quando outros identificadores não são conhecidos

### O que é Pesquisado (por Entidade):

#### Clientes:

- Apenas nome do cliente (não endereço, notas ou outros campos)

#### Contatos:

- Primeiro nome
- Sobrenome
- Endereço de e-mail
- Número de telefone

#### Sites:

- Apenas nome do site

#### Inventário:

- itemtext1 (tipicamente ICCID, número de série, endereço MAC)
- itemtext2 (tipicamente IMSI, identificador secundário)
- *Nota: Não pesquisa itemtext3-20 ou notas de inventário*

#### Serviços:

- Nome do serviço
- UUID do serviço

## O que Não é Pesquisado:

- Endereços de clientes
- Notas de clientes
- Descrições de transações
- Detalhes de faturas
- Logs de provisionamento
- Entradas de log de atividade
- Notas de inventário (além de itemtext1/2)

## Referência da API

### Endpoint de Pesquisa Global

```
GET /utilities/  
search_everything?search=Smith&page=1&per_page=10&search_closed_records=false  
Authorization: Bearer <token>
```

### Parâmetros de Consulta:

- search (obrigatório) - O termo de pesquisa
- page (opcional) - Número da página (padrão: 1)
- per\_page (opcional) - Resultados por página (padrão: 10)
- search\_closed\_records (opcional) - Incluir contas fechadas (padrão: false)

### Resposta:

```
{  
  "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 Pesquisa (Backend):

O backend realiza um SQL UNION em todas as tabelas de entidade:

```
-- 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

-- Contatos
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

-- Sites
SELECT site_id AS id,
       site_name AS name,
       customer_id,
       'site' AS type
FROM customer_site
WHERE site_name LIKE '%Smith%'

UNION ALL

-- Inventário
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

-- Serviços
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%'
```

Os resultados são então paginados e retornados.



# Considerações de Desempenho

## Desempenho da Pesquisa:

- As pesquisas usam consultas LIKE com curingas (%term%)
- Nenhum índice de texto completo atualmente implementado
- Bancos de dados grandes (>100k clientes) podem experimentar pesquisas mais lentas
- Resultados limitados a 10 por página para desempenho

## Dicas de Otimização:

- Seja específico com os termos de pesquisa para reduzir o conjunto de resultados
- Use o filtro de contas fechadas para reduzir o escopo da pesquisa
- Considere adicionar índices de banco de dados em campos frequentemente pesquisados

# Solução de Problemas

## Nenhum resultado encontrado (mas o registro existe)

- **Causa:** O termo de pesquisa não corresponde ao formato de dados armazenados
- **Exemplos:**
  - Telefone armazenado como "+44 7700 900123", pesquisar "07700900123" não corresponderá
  - E-mail armazenado como "<[john.smith@example.com](mailto:john.smith@example.com)>", pesquisar "john" não corresponderá
- **Correção:** Tente variações, use correspondências parciais que definitivamente existem

## Pesquisa muito lenta

- **Causa:** Banco de dados grande, consulta complexa em várias tabelas
- **Correção:**
  - Use termos de pesquisa mais específicos
  - Limite apenas a contas abertas (desmarque contas fechadas)
  - Contate o administrador sobre indexação de banco de dados

## Resultados vinculam ao cliente errado

- **Causa:** Múltiplos clientes/contatos com o mesmo nome
- **Correção:** Use o ID do Cliente para diferenciar, ou pesquise por identificador único (e-mail, telefone)

## Contas fechadas não aparecem

- **Causa:** Caixa de seleção "Incluir Contas Fechadas" não marcada
- **Correção:** Marque a caixa e pesquise novamente

# Documentação Relacionada

- basics\_customers - Gestão de clientes
- basics\_navigation - Navegação geral
- administration\_inventory - Pesquisas de inventário



# Sistema de Recarregamento e Top-Up

O sistema de Top-Up do OmniCRM fornece um **portal de recarga pré-paga de autoatendimento** para os clientes adicionarem crédito ou estenderem a validade do serviço através do Self-Care Portal <self\_care\_portal>. Este recurso é comumente utilizado para:

- **Serviços de dados móveis** - Cartões SIM pré-pagos e serviços apenas de dados
- **Serviços de hotspot** - Dongles de hotspot WiFi e dispositivos de internet portáteis
- **Serviços fixos sem fio** - Acesso à internet pré-pago

## Visão Geral

O sistema de top-up permite que os clientes comprem dias adicionais de serviço através de um processo de checkout simplificado e em várias etapas com processamento de pagamento integrado via Stripe.

### Principais Recursos:

- Portal de autoatendimento para clientes (sem intervenção de funcionários necessária)
- Seleção de duração flexível (1-30 dias)
- Exibição de uso em tempo real antes da compra
- Processamento de pagamento seguro com Stripe
- Reembolsos automáticos se a recarga falhar
- Geração de faturas e transações
- Integração do sistema de provisionamento para ativação do serviço

## Acessar o Portal de Top-Up

O portal de top-up é acessado através de uma **URL pública** que os clientes podem visitar sem fazer login no CRM:

### Como os Clientes Acessam:

- Link direto enviado via SMS quando o saldo está baixo
- Código QR em materiais impressos
- Link no portal de autoatendimento
- Compartilhado via suporte ao cliente

O portal detecta automaticamente o serviço do cliente com base no endereço IP ou IMSI solicitado.

## Processo de Top-Up

O fluxo de top-up consiste em **4 etapas**:

### Etapa 1: Seleção de Dados

Os clientes selecionam quantos dias de serviço desejam comprar.

#### Interface:

- **Controle deslizante** - Selecione de 1 a 30 dias
- **Cálculo de preço ao vivo** - Mostra o custo total com base na seleção
- **Exibição da data de validade** - Calcula e mostra quando o serviço irá expirar
- **Exibição do uso atual** - Mostra saldo restante/validade antes do top-up

#### Exemplo de Exibição:

#### Configuração de Preços:

- O preço por dia é configurado através da variável de ambiente `REACT_APP_TOPUP_PRICE_PER_DAY`
- Padrão: \$10 USD por dia
- A moeda é definida através de `REACT_APP_CURRENCY_CODE`

### Etapa 2: Informações de Cobrança

Os clientes fornecem seus dados de contato para a transação:

- **Primeiro Nome**
- **Último Nome**
- **Endereço de E-mail**

Essas informações são usadas para:

- Geração de fatura
- E-mail de recibo de pagamento
- Registros de transação
- Processamento de reembolso (se necessário)

### Etapa 3: Pagamento

Processamento de pagamento seguro via **Stripe Elements**.

## Métodos de Pagamento Suportados:

- Cartões de crédito (Visa, Mastercard, Amex)
- Cartões de débito
- Carteiras digitais (Apple Pay, Google Pay) *se habilitadas no Stripe*

## Recursos de Segurança:

- Integração Stripe em conformidade com PCI
- Nenhum detalhe do cartão armazenado no OmniCRM
- Suporte à autenticação 3D Secure
- Transmissão de pagamento criptografada

## Fluxo de Pagamento:

1. Formulário Stripe Elements exibido com entrada do cartão
2. O cliente insere os detalhes do pagamento
3. Intent de pagamento criada para o valor exato
4. Cartão cobrado imediatamente
5. Sucesso/falha do pagamento tratado

### Nota

Se o pagamento for bem-sucedido, mas o provisionamento do top-up falhar (por exemplo, erro de rede, OCS inacessível), o sistema inicia automaticamente um **reembolso total** para o método de pagamento do cliente.

## Etapa 4: Conclusão

### Tela de Sucesso:

Seu serviço foi estendido. Nova data de validade: 17 Jan 2025

Recibo enviado para: <[customer@example.com](mailto:customer@example.com)> ID da Transação:  
TXN-123456

### Tela de Falha:

Se o top-up falhar, o sistema exibe um erro e processa automaticamente um reembolso:

Não conseguimos completar seu top-up. Seu pagamento foi reembolsado.

Erro: Não foi possível conectar ao sistema de cobrança

Por favor, tente novamente ou entre em contato com o suporte.

# Processamento de Backend

Quando um cliente completa o pagamento, o seguinte acontece automaticamente:

## 1. Validação do Pagamento

O sistema valida:

- O status da Intent de pagamento é succeeded
- O valor do pagamento corresponde aos dias selecionados ( $\text{days} \times \text{price\_per\_day}$ )
- A Intent de pagamento não foi processada antes (previne top-up duplicado)

## 2. Operação de Top-Up

```
- API endpoint: POST /oam/topup_dongle
- Valida service_uuid e IMSI
- Chama OCS/CGRateS para adicionar saldo
- Cria trabalho de provisionamento (play_topup_hotspot)
```

## 3. Criação de Registro

O sistema cria múltiplos registros no banco de dados:

- **Registro HotspotTopup** - Rastreia a transação de top-up
  - payment\_intent\_id
  - service\_uuid
  - imsi
  - dias comprados
  - topup\_amount
  - status (Sucesso/Falhou/Reembolsado)
- **Registro de Transação** - Transação financeira
  - Título: "Hotspot Topup - 7 Dias"
  - Montante: topup\_amount (positivo)
  - Vinculado a service\_id e customer\_id
- **Registro de Fatura** - Fatura de pagamento
  - Contém a transação de top-up
  - Marcada como paga imediatamente
  - Referência de pagamento: Stripe payment\_intent\_id
- **Transação de pagamento** - Transação de crédito compensatória
  - Título: "Pagamento por [Título da Fatura]"
  - Montante: topup\_amount (negativo - crédito)
  - Vincula o pagamento da fatura à conta do cliente

## 4. Trabalho de Provisionamento

Um trabalho de provisionamento é criado com o playbook play\_topup\_hotspot que:

- Conecta-se à API OCS/CGRateS
- Adiciona saldo à conta
- Estende a data de validade
- Cria entrada de log de atividade
- Envia notificação de confirmação (se configurada)

A API aguarda a conclusão do provisionamento (polling com intervalos de 0,2s, máximo 25 iterações) antes de retornar sucesso ao cliente.

## 5. Reembolso Automático em Caso de Falha

Se qualquer etapa falhar após o pagamento:

```
if topup_provisioning_failed:
    refund = stripe.Refund.create(
        payment_intent=payment_intent_id,
        reason='requested_by_customer' # Reembolso automático do
sistema
    )
    status_message = "Topup Falhou. Reembolsando pagamento..."
```

O reembolso é processado automaticamente e o cliente é notificado na tela.

## Endpoints da API

### Endpoint de Top-Up

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
}
```

### Resposta (Sucesso):

```
{
  "result": "OK",
  "status": 200,
  "provision_id": 456,
  "payment_intent_id": "pi_1234567890abcdef",
  "service_uuid": "123e4567-e89b-12d3-a456-426614174000",
  "invoice_id": 789
}
```

### Resposta (Falha):

```
{
  "result": "Failed",
  "Reason": "Timeout de conexão OCS",
  "status": 500
}
```

### Verificações de Validação:

- Todos os campos obrigatórios presentes (service\_uuid, imsi, days, payment\_intent\_id, topup\_amount)
- topup\_amount corresponde aos dias:  $\text{topup\_amount} \times 100 == \text{days} \times 1000$  (em centavos)
- Intent de pagamento existe no Stripe
- O valor da Intent de pagamento corresponde:  $\text{payment\_intent.amount} == \text{topup\_amount} \times 100$
- O status da Intent de pagamento é succeeded
- A Intent de pagamento não foi processada anteriormente (verifica a tabela HotspotTopup)

### Endpoint de Uso

Recupera informações de uso e serviço atuais para o cliente:

GET /oam/usage

### Resposta:

```
{
  "imsi": "310120123456789",
  "service": {
    "service_uuid": "123e4567-e89b-12d3-a456-426614174000",
    "service_name": "Dados Móveis - 0412345678",
    "service_status": "Ativo"
  },
  "balance": {
    "expiry": "2025-01-10T23:59:59Z",
    "unlimited": true
  },
  "requestingIp": "203.0.113.45"
}
```

Este endpoint usa o endereço IP solicitado para identificar automaticamente o serviço do cliente.

# Configuração

## Variáveis de Ambiente

Configure estas no arquivo `.env` do OmniCRM-UI:

```
# Configuração do Portal de Top-Up
REACT_APP_TOPUP_PRICE_PER_DAY=10
REACT_APP_CURRENCY_CODE=AUD
REACT_APP_SELF_CARE_NAME="SuaEmpresa"

# Configuração do Stripe
REACT_APP_STRIPE_PUBLISHABLE_KEY=pk_live_...
```

## Configuração do Stripe

O sistema de top-up utiliza Intents de Pagamento do Stripe:

1. **Ative as Intents de Pagamento** no seu Painel do Stripe
2. **Configure Webhook** para receber atualizações de status de pagamento (opcional mas recomendado)
3. **Configure métodos de pagamento** (cartões, carteiras, etc.)
4. **Modo de teste** - Use chaves de teste para desenvolvimento

```
# Desenvolvimento
REACT_APP_STRIPE_PUBLISHABLE_KEY=pk_test_...

# Produção
REACT_APP_STRIPE_PUBLISHABLE_KEY=pk_live_...
```

## Configuração do Playbook

O playbook de provisionamento `play_topup_hotspot.yaml` deve ser configurado para:

- Aceitar a variável `days`
- Conectar-se à API OCS/CGRateS
- Adicionar saldo à conta
- Atualizar a data de validade do serviço

Estrutura de exemplo do playbook:

```
- name: Top up hotspot service
  hosts: localhost
  tasks:
    - name: Add balance to OCS
      uri:
        url: "{{ ocs_api_url }}/add_balance"
```



```
method: POST
body:
  imsi: "{{ imsi }}"
  days: "{{ days }}"
  service_uuid: "{{ service_uuid }}"
```

## Notificações de Saldo Baixo

O sistema pode enviar notificações automáticas quando o saldo do cliente está baixo:

### Notificações por SMS:

Quando acionadas por eventos do OCS (Action\_Balance\_Low, Action\_Balance\_Out, Action\_Balance\_Expired):

### Notificações por E-mail:

Configuradas nos planos de ação do OCS/CGRateS para enviar alertas de saldo.

### Gatilhos de Notificação:

- Action\_Balance\_Low - Saldo abaixo do limite (por exemplo, 2 dias restantes)
- Action\_Balance\_Out - Saldo esgotado
- Action\_Balance\_Expired - Serviço expirado

Cada notificação inclui o link do portal de top-up para fácil acesso do cliente.

## Solução de Problemas

### Problemas Comuns

#### "Sistema de pagamento indisponível"

- **Causa:** A biblioteca Stripe falhou ao carregar ou chave publicável inválida
- **Solução:**
  - Verifique se REACT\_APP\_STRIPE\_PUBLISHABLE\_KEY está configurado corretamente
  - Verifique se a conta Stripe está ativa
  - Verifique o console do navegador para erros de JavaScript

#### "Top-up falhou. Reembolsando pagamento..."

- **Causa:** O trabalho de provisionamento falhou (OCS inacessível, erro de playbook, etc.)
- **Solução:**

- Verifique os logs de provisionamento: GET /crm/provision/provision\_id/<id>
- Verifique se a API OCS/CGRateS é acessível
- Revise o playbook play\_topup\_hotspot.yaml em busca de erros
- Verifique os logs do Ansible

### "Intent de pagamento já processada"

- **Causa:** Cliente tentando reutilizar o mesmo pagamento (por exemplo, atualizar após sucesso)
- **Solução:** Isso está funcionando conforme o planejado para evitar cobranças duplicadas. O cliente deve iniciar um novo top-up, se necessário.

### "O valor da intent de pagamento não corresponde"

- **Causa:** Desvio entre o cálculo da UI e a validação do backend
- **Solução:**
  - Verifique se REACT\_APP\_TOPUP\_PRICE\_PER\_DAY corresponde à expectativa do backend (padrão \$10)
  - Verifique se a configuração da moeda é consistente
  - Limpe o cache do navegador e tente novamente

## Monitorando Top-Ups

### Visualizar Registros de Top-Up:

Consulta a tabela HotspotTopup para ver todas as tentativas de top-up:

```
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 Status de Provisionamento:

```
GET /crm/provision/provision_id/<provision_id>
```

Mostra o status detalhado do trabalho de provisionamento de top-up.

### Painel do Stripe:

Monitore pagamentos, reembolsos e transações falhadas no seu Painel do Stripe

em <<https://dashboard.stripe.com>>

## Considerações de Segurança

### Segurança de Pagamento:

- Todos os dados do cartão são tratados pelo Stripe (conforme PCI Nível 1)
- Nenhum dado sensível de pagamento armazenado no banco de dados do OmniCRM
- Intents de pagamento previnem cobranças não autorizadas
- Validação de montante em ambos os lados, cliente e servidor

### Prevenção de Fraude:

- Detecção de Intent de pagamento duplicada previne cobranças duplicadas
- Rastreamento de endereço IP para correlação de uso
- Validação de IMSI garante que o top-up vá para o serviço correto
- Reembolsos automáticos limitam a exposição financeira

### Controle de Acesso:

- O portal de top-up é público (por design - os clientes precisam de acesso)
- O endpoint de uso requer identificação de serviço válida (IP ou IMSI)
- A validação de backend previne top-ups arbitrários de serviços
- O administrador pode visualizar todos os registros de top-up através da interface do CRM

## Melhores Práticas

### Para Operadores:

1. **Teste o fluxo de reembolso** - Teste regularmente cenários de falha para garantir que os reembolsos funcionem
2. **Monitore top-ups falhados** - Configure alertas para altas taxas de falha
3. **Mantenha os playbooks simples** - Os playbooks de top-up devem ser rápidos e confiáveis
4. **Verifique a conectividade do OCS** - Garanta que a API OCS esteja sempre acessível
5. **Revise os preços** - Atualize REACT\_APP\_TOPUP\_PRICE\_PER\_DAY conforme necessário

### Para Clientes:

1. **Adicione o URL de top-up aos favoritos** - Acesso rápido quando necessário
2. **Salve notificações de saldo baixo** - O SMS contém um link direto
3. **Mantenha o e-mail atualizado** - Recibos enviados para o e-mail

registrado

4. **Verifique a validade antes de viajar** - Faça o top-up antes de sair da área de cobertura

### **Para Desenvolvedores:**

1. **Gerencie webhooks do Stripe** - Implemente manipuladores de webhook para atualizações de status de pagamento
2. **Implemente idempotência** - Sempre verifique payment\_intent\_id antes de processar
3. **Registre extensivamente** - Falhas de top-up precisam de informações detalhadas de solução de problemas
4. **Teste caminhos de erro** - Verifique se a automação de reembolso funciona corretamente
5. **Monitore o desempenho** - O polling de provisionamento deve ser concluído em <5 segundos

## **Documentação Relacionada**

- payments\_process - Processamento de pagamento geral
- concepts\_provisioning - Visão geral do sistema de provisionamento
- integrations\_stripe - Detalhes da integração com Stripe
- payments\_transaction - Gestão de transações
- payments\_invoices - Manipulação de faturas



# Glossário de Termos

OmniCRM Solução OmniTouch CRM para provedores de serviços móveis e fixos

**OmniCharge**

Motor de cobrança usado no OmniCRM

**Serviço**

Uma instância de um produto provisionado para um cliente e vinculado a uma conta OCS para faturamento e cobrança.

**Provisionamento**

O processo de transformar um produto em um serviço, incluindo a atribuição de inventário, configuração de recursos de rede e geração de cobranças através de playbooks Ansible.

**Addon**

Um serviço ou recurso adicional, como dados extras ou minutos internacionais, que pode ser adicionado a um serviço existente.

**Bolt-On**

Um tipo de addon que estende as capacidades de um serviço existente, muitas vezes de forma pontual.

**Item de Inventário**

Recursos físicos ou virtuais, como cartões SIM, modems, números de telefone ou equipamentos de rede necessários para provisionar certos serviços.

**Ansible Playbook**

Um conjunto de instruções de automação (arquivos YAML) usados para provisionar serviços a partir de uma definição de produto, configurar equipamentos de rede e integrar com sistemas externos.

**ARPU**

Receita Média Por Usuário, uma métrica chave para provedores de serviços que mede a receita média gerada por cada cliente.

**OCS**

Sistema de Cobrança Online. O motor de cobrança em tempo real (baseado em CGRateS) que lida com cobranças monetárias e recorrentes, gerencia saldos (monetários e unitários), provisiona ActionPlans para renovação automática e impõe limites de gastos via ThresholdS. Os serviços no CRM estão vinculados a contas OCS via service\_uuid.

**CGRateS**

A plataforma de sistema de cobrança em tempo real de código aberto na qual o OCS do OmniCRM é baseado. Fornece APIs para gerenciamento de saldo, execução de ações e decisões de classificação/roteamento.

#### Ação

Uma tarefa no OCS/CGRateS, como uma cobrança, redefinição de saldo ou concessão de crédito, que se aplica a um serviço. Ações podem ser executadas imediatamente ou agendadas via ActionPlans.

#### ActionPlan

Um cronograma recorrente de ações no OCS/CGRateS, como cobranças mensais ou renovações automáticas, aplicadas a um serviço. Cada serviço pode ter múltiplos ActionPlans para diferentes operações recorrentes.

#### ThresholdS

Um limite definido em uma conta para evitar gastos excessivos por um cliente, como um limite de crédito em serviços pré-pagos.

#### Cliente

Uma empresa ou indivíduo que tem um relacionamento com o provedor de serviços.

#### Contato

Uma pessoa associada a um cliente, como um contato de faturamento ou técnico, que influencia como o cliente é gerenciado.

#### Site

Um local físico onde os serviços são entregues, associado a um cliente.



# Guia Completo do Ciclo de Vida do Produto

Este guia fornece um passo a passo do ciclo de vida do produto no OmniCRM, desde a criação de uma definição de produto até a provisão de serviços, adição de complementos e desprovisionamento. Abordaremos a estratégia de preços, a integração com o Ansible e forneceremos exemplos do mundo real ao longo do texto.

## Visão Geral: A Jornada do Produto ao Serviço

O ciclo de vida de um produto no OmniCRM segue estas etapas:

1. **Definição do Produto** - O administrador cria um modelo de produto com regras de preços e provisão.
2. **Criação do Serviço** - O cliente solicita o produto, o sistema provisiona a instância do serviço.
3. **Ciclo de Vida do Serviço** - O cliente utiliza o serviço, adiciona complementos/top-ups, modifica o serviço.
4. **Desprovisionamento** - O serviço é encerrado, os recursos são liberados.

## Compreendendo Preços: Atacado vs Varejo

Todo produto e serviço no OmniCRM possui duas dimensões de preços: **atacado** e **varejo**.

### Custo de Atacado

O custo de atacado representa o custo real para entregar o serviço:

- Custos de infraestrutura e largura de banda
- Taxas de licenciamento
- Custos de equipamentos
- Despesas operacionais

### Custo de Varejo

O custo de varejo é o valor cobrado ao cliente.

### Custos de Configuração

Tanto o atacado quanto o varejo têm variantes de custo de configuração para

cobranças de provisão únicas:

- `wholesale_setup_cost` - Seu custo para provisionar
- `retail_setup_cost` - Valor cobrado ao cliente pela ativação

### Exemplo:

```
{
  "retail_cost": 15.00,
  "wholesale_cost": 5.00,
  "retail_setup_cost": 0.00,
  "wholesale_setup_cost": 1.00
}
```

## Etapa 1: Criando uma Definição de Produto

Os produtos são modelos que definem o que é provisionado e como os clientes são cobrados.

### Criando um Produto de SIM Móvel

Vamos criar um produto de SIM móvel pré-pago com 20GB de dados por mês.

#### Passo 1: Navegar até a Gestão de Produtos

Na interface de administração, vá para **Produtos** → **Criar Produto**.

#### Passo 2: Definir Informações Básicas

```
{
  "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"
}
```

#### Explicações dos Campos:

- `product_name` - Nome voltado para o cliente exibido no catálogo.
- `product_slug` - Identificador seguro para URL usado em chamadas de API e links.
- `category` - "standalone" significa que isso cria um novo serviço (vs complemento/pacote).
- `service_type` - Agrupa produtos relacionados, usado para filtragem de



complementos.

- `enabled` - Deve ser verdadeiro para que o produto possa ser solicitado.
- `icon` - Ícone do FontAwesome exibido na interface.
- `comment` - Notas internas para referência da equipe.

### Passo 3: Definir Preços

```
{  
  "retail_cost": 15.00,  
  "wholesale_cost": 5.00,  
  "retail_setup_cost": 0.00,  
  "wholesale_setup_cost": 1.00,  
  "contract_days": 30  
}
```

#### Detalhamento de Preços:

- Receita mensal por cliente: £15.00
- Custo mensal para entrega: £5.00
- Margem de lucro mensal: £10.00 (200% de markup, 67% de margem)
- Lucro de configuração: -£1.00 (subvencionado para atrair clientes)
- Duração do contrato: 30 dias (renovação mensal)

### Passo 4: Definir Elegibilidade do Cliente

```
{  
  "residential": true,  
  "business": false,  
  "customer_can_purchase": true,  
  "available_from": "2025-01-01T00:00:00Z",  
  "available_until": null  
}
```

- Clientes residenciais podem solicitar.
- Clientes empresariais não podem (linha de produto diferente).
- Compra de autoatendimento habilitada.
- Disponível a partir de 1º de janeiro de 2025.
- Sem data de término (oferta contínua).

### Passo 5: Configurar Renovação Automática

```
{  
  "auto_renew": "prompt",  
  "allow_auto_renew": true  
}
```

- `"prompt"` - Perguntar ao cliente se deseja renovação automática na compra.
- `"true"` - Renovar automaticamente sem perguntar.

- "false" - Nunca renovar automaticamente (apenas recarga manual).
- allow\_auto\_renew: true - O cliente pode habilitar/desabilitar a renovação automática posteriormente.

## Passo 6: Especificar Requisitos de Inventário

Os requisitos de inventário definem quais recursos físicos ou virtuais devem ser alocados ao provisionar este produto. Esta é uma etapa crítica que conecta seu catálogo de produtos ao seu Sistema de Gestão de Inventário <administration\_inventory>.

```
{
  "inventory_items_list": ["SIM Card", "Mobile Number"]
}
```

## O que são Itens de Inventário?

Itens de inventário são recursos rastreáveis armazenados no sistema de inventário do OmniCRM. Cada item possui:

- **Tipo** - Definido pelo Modelo de Inventário (por exemplo, "SIM Card", "Mobile Number", "Modem").
- **Atributos únicos** - Números de série, endereços MAC, números de telefone, etc.
- **Estado** - Em Estoque, Atribuído, Descomissionado, etc.
- **Localização** - Localização física ou lógica.

## Como Funcionam os Requisitos de Inventário:

A inventory\_items\_list é uma lista Python (como uma string) contendo nomes de tipos de inventário. Cada nome deve corresponder exatamente a um nome existente de Modelo de Inventário <administration\_inventory>.

## Exemplo de Requisitos de Inventário:

```
# Produto SIM Móvel
inventory_items_list: ["SIM Card", "Mobile Number"]

# Serviço de internet fixa
inventory_items_list: ["Modem Router", "Static IP Address"]

# Serviço digital (sem itens físicos)
inventory_items_list: []

# Wireless fixo com CPE
inventory_items_list: ["Fixed Wireless CPE", "IPv4 Address", "IPv6 Prefix"]
```

## O Processo de Seleção de Inventário

Quando um usuário provisiona um produto com requisitos de inventário, o sistema impõe um processo de seleção obrigatório:

## 1. Botão de Provisão Clicado

Após selecionar o produto, o usuário clica em "Provisionar". Em vez de provisionar imediatamente, o sistema verifica a `inventory_items_list`.

## 2. Modal de Seleção de Inventário Aparece

Se o inventário for necessário, um diálogo modal aparece com um dropdown separado para cada tipo de inventário:

## 3. Filtrando Inventário Disponível

O dropdown para cada tipo de inventário mostra apenas itens que são:

- **Tipo Correto** - Corresponde exatamente ao nome do modelo de inventário.
- **Status Disponível** - `item_state` é "Novo" ou "Em Estoque" (não "Atribuído" ou "Danificado").
- **Não Atribuído** - `service_id` e `customer_id` são NULL.
- **Em Estoque na Localização** - Opcionalmente filtrado pela localização do armazém/loja.

## Exemplo de Opções de Dropdown:

Para o tipo de inventário "SIM Card", o dropdown pode mostrar:

Cada opção exibe:

- ID do inventário ou número de referência.
- Identificador principal (`itemtext1` - por exemplo, ICCID para SIM, número para telefone).
- Localização atual (`item_location`).

## 4. Seleção Necessária para Prosseguir

**Regra Crítica:** O provisionamento NÃO pode prosseguir sem selecionar todos os itens de inventário necessários.

- O botão "Continuar" é desativado até que todos os dropdowns tenham seleções.
- O usuário deve selecionar um item para cada tipo de inventário.
- O sistema valida as seleções antes de prosseguir.

## 5. Inventário Selecionado Passado para Ansible

Uma vez que o usuário clica em "Continuar", os IDs de inventário selecionados são passados para o playbook do Ansible como variáveis:

```
# Usuário selecionou:
# - ID do inventário do SIM Card: 5001
# - ID do inventário do Mobile Number: 5002

# Variáveis passadas para o Ansible:
{
    "product_id": 42,
    "customer_id": 123,
    "SIM Card": 5001,           # ID do Inventário
    "Mobile Number": 5002,     # ID do Inventário
    "access_token": "eyJ..."
}
```

**Nota:** O nome da variável corresponde exatamente ao tipo de inventário. O playbook usa `hostvars[inventory_hostname]['SIM Card']` para acessar o ID do inventário.

## 6. Playbook Busca Detalhes Completos do Inventário

O playbook do Ansible usa o ID do inventário para buscar detalhes 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 }}"
```

Agora o playbook tem todos os detalhes do SIM (ICCID, IMSI, etc.) para provisionar o assinante no HSS.

## 7. Estado do Inventário Alterado para "Atribuído"

Após o registro do serviço ser criado, o playbook atualiza o inventário para vinculá-lo ao serviço:

```
- 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:** A atribuição do inventário acontece **durante a execução do playbook** como uma tarefa específica, NÃO quando o botão de provisão é clicado. Isso significa:

- **Risco de Dupla Atribuição:** Entre clicar em "Provisionar" e o inventário ser atribuído, outro usuário poderia teoricamente selecionar o mesmo item de inventário.
- **Melhor Prática:** Para operações de alto volume, implemente bloqueio de inventário ou use transações de banco de dados.
- **Rollback em Caso de Falha:** Se o playbook falhar antes da atribuição do inventário, o inventário permanece não atribuído e disponível para reutilização.

### Por que não atribuir mais cedo?

O inventário não é atribuído quando "Provisionar" é clicado porque:

1. **ID do Serviço Necessário:** O service\_id não existe até que o serviço seja criado no playbook.
2. **Simplicidade de Rollback:** Se o provisionamento falhar cedo (por exemplo, a criação da conta OCS falhar), o inventário não precisa de limpeza.
3. **Flexibilidade:** O playbook pode decidir não atribuir inventário com base em lógica condicional.

### Tratando Provisionamentos Falhados:

Quando um provisionamento falha após o inventário ser atribuído, o bloco de resgate deve liberar o inventário:

```

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 # Apenas se o serviço foi criado
```

Isso garante que o inventário não fique em um estado "Atribuído" para um serviço inexistente ou falhado.

## Quando a Lista de Inventário está Vazia

Se `inventory_items_list: []` (lista vazia), o seletor de inventário é totalmente ignorado e o provisionamento prossegue imediatamente. Isso é comum para:

- **Produtos digitais** - Licenças de software, contas de VPN.
- **Complementos de serviço** - Recargas de dados que não precisam de novo hardware.
- **Serviços virtuais** - Que não consomem recursos rastreáveis.

**Exemplo:** Um complemento "5GB Data Boost" tem `inventory_items_list: []` porque apenas adiciona saldo a um serviço existente sem precisar de novo hardware.

## Configuração do Modelo de Inventário

Antes de usar um tipo de inventário em `inventory_items_list`, você deve criar o Modelo de Inventário:

1. Navegue até **Administração** → **Inventário** → **Modelos**.
2. Crie um modelo com o nome exato (por exemplo, "SIM Card").
3. Defina os campos:
  - `itemtext1_label`: "ICCID".
  - `itemtext2_label`: "IMSI".
  - `itemtext3_label`: "PUK Code".
4. Adicione itens de inventário desse tipo ao estoque.

Para detalhes completos sobre como criar e gerenciar modelos de inventário, consulte [Gestão de Inventário <administration\\_inventory>](#).

## Múltiplos Itens do Mesmo Tipo

Embora a `inventory_items_list` seja um array, ter tipos duplicados (por exemplo, `['SIM Card', 'SIM Card']`) **não é recomendado** pois pode causar confusão na interface e na nomeação de variáveis do playbook.

Para cenários que exigem múltiplos itens semelhantes:

### Opção 1: Criar nomes de modelos de inventário distintos

```
# Serviço de telefone Dual-SIM
inventory_items_list: ['Primary SIM Card', 'Secondary SIM Card',
'Mobile Number']
```

Crie modelos separados: "Primary SIM Card" e "Secondary SIM Card" com os mesmos campos, mas nomes diferentes.

## Opção 2: Usar um único item de inventário agrupado

```
# Kit Dual-SIM
inventory_items_list: ["Dual SIM Kit", 'Mobile Number']
```

Onde o modelo de inventário "Dual SIM Kit" tem campos para ambos os SIMs (itemtext1: ICCID Primário, itemtext2: ICCID Secundário, etc.).

## Cenários Comuns de Inventário

### Serviço Móvel:

```
inventory_items_list: ["SIM Card", 'Mobile Number']
```

- SIM Card: Físico ou eSIM com ICCID/IMSI.
- Mobile Number: Número de telefone (MSISDN).

### Internet Fixa:

```
inventory_items_list: ["Modem Router", 'Static IP Address']
```

- Modem Router: Dispositivo CPE com endereço MAC.
- Static IP Address: IPv4 do pool de endereços.

### Wireless Fixo:

```
inventory_items_list: ["Fixed Wireless CPE", 'IPv4 Address', 'IPv6 Prefix']
```

- CPE: Equipamento nas instalações do cliente (antena, modem).
- IPv4: Endereço IP público.
- IPv6 Prefix: prefixo /56 ou /64.

**Nota:** Compromissos e agendamentos **não** são itens de inventário. Use sistemas de agendamento/calendário separados para compromissos de instalação.

### Serviço VoIP:

```
inventory_items_list: ["DID Number"]
```

- DID Number: Número de telefone de Discagem Direta.

**Nota:** Nomes de usuário SIP, senhas e configurações de conta são **gerados programaticamente** pelo playbook de provisionamento, não selecionados do inventário.

## GPON/Fibra:

```
inventory_items_list: ["'ONT Device'", 'GPON Port', 'IPv4 Address',  
'Fiber Drop Cable']"
```

- ONT Device: Terminal de Rede Óptica com número de série.
- GPON Port: Porta específica no OLT com conexão de fibra.
- IPv4 Address: IP público ou privado.
- Fiber Drop Cable: Cabo de fibra física da rua até as instalações (rastreado para gestão de ativos).

## Aluguel de Equipamento:

```
inventory_items_list: ["'Rental Modem'"]"
```

- Rastreia qual modem está com qual cliente.
- Importante para recuperar equipamentos em caso de cancelamento.

## Por que os Requisitos de Inventário Importam

### 1. Prevenir Dupla Atribuição

Sem rastreamento de inventário, você poderia acidentalmente:

- Atribuir o mesmo cartão SIM a dois clientes.
- Alocar o mesmo endereço IP a vários serviços.
- Enviar o mesmo número de série de equipamento para diferentes locais.

O seletor de inventário garante que cada item seja atribuído a exatamente um serviço.

### 2. Rastro de Auditoria

A atribuição de inventário cria um rastro de auditoria completo:

- Qual cartão SIM está com qual cliente.
- Quando foi atribuído.
- Qual serviço está usando qual número de telefone.
- Histórico de equipamentos (quem teve, quando, para qual serviço).

### 3. Planejamento de Recursos

Rastreie os níveis de inventário:

- Alerta quando os cartões SIM estão baixos.
- Reordene antes de esgotar o estoque.
- Planeje as agendas dos técnicos com base na disponibilidade do CPE.
- Gerencie a alocação de espaço de endereço IP.



## 4. Rastreio de Custos

Vincule o custo de atacado a um item específico:

- Rastreie o custo de cada cartão SIM.
- Calcule a depreciação do equipamento.
- Identifique itens perdidos ou roubados.
- COGS (Custo das Mercadorias Vendidas) preciso.

## 5. Desprovisionamento

Quando o serviço é cancelado, o inventário pode ser:

- Liberado de volta ao estoque (cartões SIM, modems).
- Retirado (equipamento danificado).
- Devolvido ao fornecedor (equipamento alugado).
- Mantido por um período de carência (números de telefone antes da liberação).

## Resolvendo Problemas com o Seletor de Inventário

**Problema:** Mensagem "Nenhum inventário disponível" aparece.

### Causas:

- Nenhum item de inventário do tipo necessário existe no banco de dados.
- Todos os itens já estão "Atribuídos" a outros serviços.
- Itens estão marcados como "Danificados" ou "Fora de Serviço".
- O nome do modelo de inventário não corresponde exatamente (sensível a maiúsculas e minúsculas).

### Solução:

1. Verifique se o modelo de inventário existe: **Administração → Inventário → Modelos**.
2. Verifique se o nome do modelo corresponde exatamente (incluindo espaços, maiúsculas e minúsculas).
3. Adicione itens de inventário desse tipo: **Administração → Inventário → Adicionar Item**.
4. Verifique se os itens estão no estado "Novo" ou "Em Estoque".
5. Verifique se os itens não estão já atribuídos (`service_id` deve ser NULL).

**Problema:** O seletor de inventário não aparece.

### Causas:

- `inventory_items_list` está vazio: `[]`.
- `inventory_items_list` é NULL ou não está definido.
- A categoria do produto é "complemento" e herda o inventário do serviço

pai.

### **Solução:**

- Se o inventário for necessário, defina `inventory_items_list`: `"['Type1', 'Type2']"`.
- Verifique se a definição do produto foi salva corretamente.
- Verifique a resposta da API para o produto inclui `inventory_items_list`.

**Problema:** O playbook falha com "inventário não encontrado".

### **Causas:**

- O playbook faz referência ao nome da variável errada.
- O ID do inventário não foi passado corretamente.
- O inventário foi excluído entre a seleção e o provisionamento.

### **Solução:**

- Verifique se o playbook usa a variável correta: `hostvars[inventory_hostname]['SIM Card']`.
- Verifique se a variável é um inteiro: `{{ hostvars[inventory_hostname]['SIM Card'] | int }}`.
- Adicione tratamento de erro no playbook para inventário ausente.

Consulte [Gestão de Inventário <administration\\_inventory>](#) para detalhes completos sobre como criar modelos, adicionar itens e gerenciar níveis de estoque.

## **Passo 7: Definir Recursos e Termos**

Recursos e termos são conteúdos de marketing e legais voltados para o cliente que ajudam os clientes a entender o que estão comprando e as obrigações envolvidas.

```
{
  "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 e Valor Comercial**

### **Lista de Recursos - Marketing e Vendas:**

A lista de recursos serve a várias funções comerciais críticas:

1. **Diferenciação de Produtos** - Ajuda os clientes a comparar rapidamente produtos e escolher o correto.
  - "Prepaid Mobile 20GB" vs "Prepaid Mobile 50GB" - os recursos mostram claramente a diferença.
  - Sem recursos, os clientes veem apenas o preço, perdendo a proposta de valor.
2. **Comunicação de Marketing** - Pontos de venda chave exibidos de forma proeminente.
  - "EU Roaming Included" atrai viajantes internacionais.
  - "No Contract" atrai clientes avessos a compromissos.
  - Recursos impulsionam decisões de compra.
3. **Expectativas do Cliente** - Define expectativas claras sobre o que está incluído.
  - Reduz chamadas de suporte ("Isso inclui chamadas?" → claramente listado).
  - Previne mal-entendidos e pedidos de reembolso.
  - Constrói confiança através da transparência.
4. **Autoatendimento** - Permite que os clientes selecionem produtos apropriados por conta própria.
  - O cliente lê os recursos, entende a oferta, faz uma escolha informada.
  - Reduz a necessidade de explicação da equipe de vendas.
  - Acelera o processo de compra.
5. **SEO e Descobribilidade** - Recursos podem ser indexados para pesquisa.
  - O cliente pesquisa "plano móvel de chamadas ilimitadas" → o produto aparece.
  - Melhora a pesquisabilidade do catálogo de produtos.

## **Termos e Condições - Legal e Conformidade:**

Os termos servem a propósitos legais e operacionais:

1. **Proteção Legal** - Protege o negócio de disputas e responsabilidade.
  - "O crédito expira após 30 dias" - o cliente não pode exigir reembolso após 31 dias.
  - "A política de uso justo se aplica" - previne abusos (tethering de todo o escritório em um plano móvel).
  - Cria um acordo vinculativo.
2. **Gestão de Expectativas** - Previne insatisfação do cliente.
  - "Válido apenas dentro do período de validade" - o cliente conhece o prazo de uso.
  - "Não pode ser reembolsado" (para complementos) - previne compras fraudulentas.
  - Reduz chargebacks e reclamações.
3. **Conformidade Regulatória** - Atende a requisitos legais.
  - Leis de proteção ao consumidor exigem termos claros.
  - Regulamentações de telecomunicações exigem divulgação.
  - Termos de GDPR/privacidade podem ser referenciados.

4. **Limites Operacionais** - Define o escopo e limitações do serviço.
  - "Sujeito à cobertura da rede" - não é responsável por zonas mortas.
  - "A velocidade pode variar" - gerencia expectativas sobre velocidades "até".
  - "O equipamento deve ser devolvido" - garante a recuperação de equipamentos alugados.
5. **Rastro de Auditoria** - Prova que o cliente foi informado.
  - O cliente aceitou os termos na compra.
  - O sistema registra o timestamp de aceitação.
  - Defensável em disputas ou processos legais.

### Exemplo do Mundo Real:

O cliente compra um plano "Chamadas e Textos Ilimitados", e depois o usa para telemarketing (10.000 chamadas/dia). Sem termos:

- Cliente: "Você disse ilimitado!"
- Provedor: "Queríamos dizer uso pessoal..."
- Cliente: "Isso não é o que você anunciou!"
- Resultado: Disputa, potencial reclamação do regulador, dano à marca.

Com os termos: "A política de uso justo se aplica. O serviço é para uso pessoal apenas. Uso comercial é proibido."

- Provedor: Aponta para os termos que o cliente aceitou.
- O cliente não pode alegar ignorância.
- Base legal para suspender o serviço.
- Disputa resolvida a favor do provedor.

### Formato da Lista de Recursos:

Compreender o formato correto é crítico porque **formatação inadequada quebra a exibição da interface**. Os recursos podem aparecer como uma longa string em vez de pontos de lista, ou não serem exibidos de forma alguma.

O campo `features_list` pode ser formatado de duas maneiras:

#### Opção 1: String Separada por Ponto (Recomendado)

Os recursos são separados por um ponto e espaço (". "). A interface divide neste delimitador e renderiza cada recurso como um ponto de lista.

#### Por que esse formato?

- Simples de editar - basta digitar os recursos com pontos entre eles.
- Sem caracteres especiais para escapar.
- Funciona de forma confiável em todos os componentes da interface.
- Fácil de atualizar sem quebrar a sintaxe JSON.

## Correto vs Incorreto:

### Opção 2: String de Array JSON

```
"['20GB High-Speed Data', 'Unlimited Calls & Texts', 'EU Roaming Included']"
```

A interface também pode analisar arrays JSON. Note que esta é uma **string contendo JSON**, não um array JSON real no banco de dados.

### Por que esse formato existe?

- Permite recursos com pontos neles (por exemplo, "Até 100Mbps. Sujeito à disponibilidade.").
- A geração programática a partir de scripts/API é mais fácil.
- Importado de catálogos de produtos externos que usam arrays.

**Importante:** Isso deve ser uma sintaxe de lista Python válida como uma string. Aspas simples ao redor de cada item, aspas duplas ao redor da string inteira.

### Qual Formato Usar?

- **Separado por ponto** - Para criação manual de produtos na interface (mais simples, menos propenso a erros).
- **Array JSON** - Para criação de produtos baseada em API/script (mais robusto para recursos complexos).

Ambos os formatos produzem saída idêntica na interface - eles apenas afetam como você insere os dados.

### Onde os Recursos Aparecem na Interface:

#### 1. Catálogo de Produtos (Visão do Cliente)

Quando os clientes navegam pelos produtos disponíveis, os recursos são exibidos como pontos de lista em cada cartão de produto:

#### 2. Página de Detalhes do Produto

Clicando em "Ver Detalhes" mostra todas as informações do produto, incluindo:

- Nome do produto e ícone.
- Preços (custo mensal, custo de configuração).
- Lista completa de recursos (pontos de lista).
- Termos e condições (veja abaixo).
- Disponibilidade e elegibilidade.

#### 3. Confirmação de Provisão

Durante a provisão, os recursos são mostrados para o usuário revisar antes de confirmar:

Recursos: • 20GB High-Speed Data • Unlimited Calls & Texts • EU Roaming Included • No Contract • 30-Day Expiry

Custo: £15.00/mês Configuração: £0.00

[Cancelar] [Confirmar & Provisionar]

#### 4. Detalhes do Serviço (Após Provisão)

Após o serviço estar ativo, os recursos são exibidos na página de detalhes do serviço para referência do cliente.

##### Formato dos Termos e Condições:

O campo terms é texto simples que pode incluir quebras de linha:

##### Onde os Termos Aparecem na Interface:

###### 1. Página de Detalhes do Produto

Os termos são exibidos em uma seção colapsada que se expande quando clicada:

###### 2. Confirmação do Pedido

Durante a provisão, uma caixa de seleção exige que o usuário aceite os termos:

O botão [Provisionar] está desativado até que seja marcado.

###### 3. Faturas

Os termos do serviço podem ser incluídos nas faturas como notas de rodapé para clareza.

##### Melhores Práticas:

- **Recursos:** Mantenha concisos (menos de 50 caracteres cada), concentre-se nos principais benefícios.
- **Termos:** Inclua requisitos legais críticos, políticas de expiração, políticas de uso justo.
- **Ambos:** Atualize quando o produto mudar para manter os clientes informados.

#### Passo 8: Vincular Playbook de Provisionamento do Ansible

```
{  
  "provisioning_play": "play_local_mobile_sim",  
}
```

```

"provisioning_json_vars": "{
  \"days\": 30,
  \"data_gb\": 20,
  \"voice_minutes\": \"unlimited\",
  \"sms_count\": \"unlimited\"
}"
}

```

- provisioning\_play - Nome do playbook do Ansible (sem a extensão .yaml).
- provisioning\_json\_vars - Variáveis padrão passadas para o playbook.
- O playbook deve existir em: OmniCRM-API/Provisioners/plays/play\_local\_mobile\_sim.yaml.

## Definição Completa do Produto

```

{
  "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\"
}"
}

```

## Criando um Produto Adicional

Os complementos melhoram ou modificam serviços existentes. Eles vêm em dois tipos: **complementos virtuais** (sem recursos físicos) e **complementos de hardware** (exigem inventário).

### Exemplo 1: Complemento Virtual (5GB Data Boost)

Um complemento digital que adiciona dados a um serviço móvel 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
    }"
  }
}

```

## Exemplo 2: Complemento de Hardware (Aluguel de Modem)

Um complemento que fornece equipamento físico para um serviço 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  
  }  
}
```

### Principais Diferenças para Complementos:

- **category:** "addon" - Aplicado a serviço existente, não autônomo.
- **contract\_days:** 0 (virtual) ou 30 (aluguel recorrente) - Frequência de cobrança.
- **inventory\_items\_list:** "[]" (virtual) ou "['Rental Modem']" (hardware) - Recursos físicos.
- **auto\_renew:** "false" (uma vez) ou "true" (aluguel) - Comportamento recorrente.
- **relies\_on\_list:** "" - Vazio significa que se aplica a qualquer serviço do tipo service\_type correspondente.

### Por que os Complementos de Hardware Precisam de Inventário:

Os complementos de hardware exigem inventory\_items\_list porque:

1. **Rastrear Equipamento** - Saber qual modem está com qual cliente.
2. **Prevenir Esgotamento de Estoque** - Não é possível provisionar complemento se não houver modems em estoque.
3. **Recuperação** - Quando o cliente cancela, saber qual equipamento recuperar.
4. **Rastreio de Custos** - Vincular custo de atacado a número de série específico.
5. **Depreciação** - Rastrear o valor do equipamento durante o período de aluguel.
6. **Garantia** - Identificar unidades defeituosas pelo número de série.

### Fluxo de Provisionamento de Complementos com Inventário:

Quando um cliente adiciona "Aluguel de Modem WiFi 6" ao seu serviço de fibra:

1. **Complemento Selecionado** - O cliente clica em "Adicionar ao Serviço".
2. **Selecionador de Inventário Aparece** - Assim como serviços autônomos:
3. **Pagamento Processado** - £10.00 de aluguel mensal cobrado.
4. **Modem Atribuído** - Inventário atualizado:
  - service\_id: Vinculado ao serviço de fibra.
  - customer\_id: Vinculado ao cliente.
  - item\_state: "Assigned".
5. **Envio Acionado** - Sistema de atendimento notificado para enviar o modem.
6. **Instalação** - O cliente recebe o modem, conecta ao ONT.
7. **Cobrança Recorrente** - £10/mês cobrado até que o complemento seja cancelado.

### Desprovisionamento de Complementos de Hardware:

Quando o cliente cancela o aluguel do modem:

1. **Cancelamento Iniciado** - O cliente clica em "Remover Complemento".
2. **Processo de Devolução Iniciado:**
  - E-mail enviado com instruções de devolução.
  - Etiqueta de envio pré-paga gerada.
  - Período de carência de 14 dias antes da penalidade.
3. **Equipamento Devolvido:**
  - Inventário atualizado: `item_state = "In Stock"` (após reforma).
  - Ou `item_state = "Damaged"` (se defeituoso).
  - Vinculado ao próximo cliente uma vez reformado.
4. **Sem Devolução:**
  - Após 14 dias, taxa de substituição de £150 cobrada.
  - Inventário marcado: `item_state = "Lost"`.
  - Custo de atacado (£45) + valor de substituição recuperados.

### **Precificação para Complementos:**

Os complementos podem ser precificados de forma diferente dos serviços autônomos:

- Complementos virtuais normalmente não têm custos de configuração.
- Complementos de hardware podem ter custos de configuração de atacado para equipamentos.
- Complementos de aluguel recorrente usam `contract_days` para frequência de cobrança.

## **Etapas 2: O Processo de Provisionamento**

Quando um cliente solicita o produto "Prepaid Mobile 20GB", o OmniCRM orquestra o provisionamento através do Ansible.

### **Diagrama de Fluxo de Provisionamento**

Cliente Solicita → Seleção de Inventário → Trabalho de Provisionamento Criado ↓  
↓ Pagamento Autorizado ← Variáveis Montadas ← Playbook do Ansible Executado  
↓ ↓ Registro de Serviço Criado → Configuração da Conta OCS → Inventário Atribuído → Serviço Ativo

### **Fluxo de Provisionamento Passo a Passo**

#### **1. Cliente Inicia o Pedido**

Na página do cliente:

- A equipe clica em "Adicionar Serviço".
- Seleciona "Prepaid Mobile 20GB" no carrossel de produtos.
- Detalhes do produto e preços exibidos.

## 2. Seleção de Inventário

O sistema solicita o inventário necessário:

- **SIM Card** - O dropdown mostra os cartões SIM disponíveis em estoque.
  - Exemplo: "SIM-00123 - ICCID: 8944..."
- **Mobile Number** - O dropdown mostra os números de telefone disponíveis.
  - Exemplo: "+44 7700 900123".

A equipe ou o cliente seleciona itens do inventário disponível.

## 3. Confirmação de Preços

O sistema exibe o preço final:

- Custo de configuração: £0.00 (ativação gratuita).
- Custo mensal: £15.00.
- Devido hoje: £15.00 (primeiro mês).
- Data de renovação: 30 dias a partir de hoje.

Se a solicitação de renovação automática estiver habilitada, o cliente escolhe:

- ☐ Renovar automaticamente este serviço a cada 30 dias.

## 4. Botão de Provisão Clicado

Quando "Provisionar" é clicado, a API:

- Cria um registro de Provision com status "Running" (status=1).
- Mescla variáveis do produto + solicitação + seleções de inventário.
- Cria uma thread em segundo plano para executar o playbook do Ansible.
- Retorna provision\_id para a interface para rastreamento de status.

## 5. Variáveis Montadas

O sistema mescla variáveis de várias fontes:

### Do Produto:

```
{
  "days": 30,
  "data_gb": 20,
  "voice_minutes": "unlimited",
  "sms_count": "unlimited"
}
```

### Da Solicitação:

```
{
```

```
"product_id": 42,  
"customer_id": 123,  
"SIM Card": 5001,  
"Mobile Number": 5002  
}
```

### Adicionado pelo Sistema:

```
{  
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",  
  "initiating_user": 7  
}
```

### Variáveis Finais Passadas para o 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. Execução do Playbook do Ansible

O playbook `play_local_mobile_sim.yaml` é executado com essas variáveis.

## Compreendendo o Playbook de Provisionamento do Ansible

Vamos examinar um playbook de provisionamento real para entender o que acontece nos bastidores.

### Exemplo de Playbook de Provisionamento de SIM Móvel

**Localização:** `OmniCRM-API/Provisioners/plays/play_local_mobile_sim.yaml`

#### Estrutura de Alto Nível:

```
- 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

```

## Passo a Passo Detalhado do Playbook:

### Tarefa 1: Carregar Configuração

```

- name: Include vars of crm_config
  ansible.builtin.include_vars:
    file: "../..//crm_config.yaml"
    name: crm_config

```

Carrega a configuração do sistema, incluindo:

- URL e credenciais do OCS/CGRateS.
- URL base do CRM.
- Configuração do inquilino.

### Tarefa 2: Buscar Detalhes do Produto

```

- 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

```

### O que isso faz:

- Chama GET /crm/product/product\_id/42.
- Recupera a definição completa do produto.
- Armazena na variável `api_response_product`.

**Por que:** Mesmo que tenhamos `provisioning_json_vars` do produto, buscamos o produto completo para obter:

- Preços mais recentes (podem ter mudado desde que o pedido começou).
- Nome do produto para nomeação do serviço.
- Lista de recursos para documentação.
- Custos de atacado para rastreamento de margem.

### Tarefa 3: Definir Fatos do Pacote

```
- 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 }}"
```

Extraí valores comumente usados em variáveis simples para legibilidade.

### Tarefa 4: Buscar Detalhes do Inventário

```
- 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 }}"
```

### O que isso faz:

- Busca o ID do inventário do cartão SIM 5001.
- Recupera detalhes do SIM:
  - `itemtext1` = ICCID (número do cartão SIM).
  - `itemtext2` = IMSI (identidade do assinante).
- Faz o mesmo para o inventário do número móvel (recupera o número de telefone).

### Por que isso importa:

- IMSI é necessário para provisionar o assinante no HSS (Servidor de Assinantes).
- ICCID é registrado nas notas do serviço para solução de problemas.
- O número de telefone (MSISDN) é exibido para o cliente e usado para roteamento.

### Tarefa 5: Gerar UUID do Serviço

```
- 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] }}"
```

### O que isso faz:

- Gera um UUID aleatório.
- Cria um service\_uuid como Local\_Mobile\_SIM\_a3f2c1d8.

### Por que:

- O UUID do serviço é o identificador único no OCS/CGRateS.
- Usado para todas as operações de cobrança.
- Deve ser globalmente único entre todos os serviços.

### Tarefa 6: Criar Conta no 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

```

### O que isso faz:

- Chama a API JSON-RPC do CGRateS.
- Cria uma nova conta com service\_uuid.
- Define a conta como ativa (não desabilitada).
- Previne saldo negativo (modo pré-pago).

### Por que:

- A conta OCS é onde toda a cobrança acontece.
- Saldo (dados, voz, SMS, dinheiro) é armazenado aqui.
- Uso é rastreado e avaliado em tempo real.

## Tarefa 7: Adicionar Saldo de Dados

```

- 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"
          }
        }]
      }

```

### O que isso faz:

- Adiciona 20GB de saldo de dados à conta.
- Valor: 21474836480 bytes (20 \* 1024 \* 1024 \* 1024).
- Expira em 720 horas (30 dias).
- Peso 10 (peso mais alto consumido primeiro).

## Tarefa 8: Adicionar Voz e 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"
          }
        }]
      }
  }
```

- Adiciona 999.999.999 segundos de voz (essencialmente ilimitado).
- Expira em 30 dias.

## Tarefa 9: Criar Registro de Serviço no 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
```

### O que isso cria:

- Registro de serviço vinculado ao cliente.
- Vincula ao OCS através de `service_uuid`.
- Armazena custos de varejo e atacado.
- Define o status como "Ativo".
- Retorna `service_id` para operações subsequentes.

### Tarefa 10: Atribuir Inventário ao Serviço

```
- 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"
      }
```

### O que isso faz:

- Atualiza o registro do inventário do cartão SIM.
- Define `service_id` para vincular o SIM ao serviço.
- Altera o estado de "Em Estoque" para "Atribuído".
- Repete para o inventário do número móvel.

### Por que:

- Rastreia qual SIM está atribuído a qual cliente.
- Previne dupla alocação de inventário.
- Permite relatórios e auditoria de inventário.

### Tarefa 11: Registrar Transação de Custo de Configuração

```
- 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 }}"
    }

```

### O que isso faz:

- Registra a cobrança de £0.00 de configuração ao cliente (varejo).
- Registra o custo de atacado de £1.00.
- Cria um registro de transação para faturamento.

## Tarefa 12: Bloco de Resgate (Tratamento de Erros)

```

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"

```

### O que isso faz:

- Se qualquer tarefa falhar, o bloco de resgate é executado.
- Exclui a conta OCS que foi parcialmente criada.
- Libera o inventário de volta para "Em Estoque".
- Falha o trabalho de provisão com uma mensagem de erro.

### Por que:

- Previne contas órfãs no OCS.
- Garante rollback limpo em erros.
- Mantém a consistência dos dados.

## Provisionamento Completo: O que foi Criado

Após o provisionamento bem-sucedido, o sistema possui:

### 1. Conta OCS (CGRateS):

- ID da Conta: Local\_Mobile\_SIM\_a3f2c1d8.
- Saldos:
  - 20GB de dados (expira em 30 dias).
  - Voz ilimitada (999M segundos, expira em 30 dias).
  - SMS ilimitados (999M mensagens, expira em 30 dias).

### 2. Registro de Serviço no CRM:

- ID do Serviço: 1234.
- Cliente: John Doe (customer\_id: 123).
- Produto: Prepaid Mobile 20GB (product\_id: 42).
- Nome do Serviço: "Mobile - +44 7700 900123".
- UUID do Serviço: Local\_Mobile\_SIM\_a3f2c1d8.
- Status: Ativo.
- Custo Mensal: £15.00 (varejo), £5.00 (atacado).
- Lucro: £10.00/mês.

### 3. Atribuições de Inventário:

- Cartão SIM 5001: Atribuído ao serviço 1234, cliente 123.
- Número Móvel 5002: Atribuído ao serviço 1234, cliente 123.

### 4. Registros de Transação:

- Transação de custo de configuração criada.
- Cobrança do primeiro mês registrada.

### 5. O Cliente Agora Pode:

- Visualizar o serviço no portal de autoatendimento.
- Ver saldo de 20GB de dados.
- Fazer chamadas e enviar SMS.
- Recarregar ou adicionar complementos.
- Visualizar uso em tempo real.

## Etapa 3: Adicionando Complementos e Recargas

Após um serviço estar ativo, os clientes podem comprar complementos para aprimorar seu serviço.

## Fluxo de Provisionamento de Complementos

Vamos supor que o cliente usou 18GB de sua cota de 20GB e deseja comprar o complemento "5GB Data Boost".

### 1. Cliente Navega até o Serviço

- Abre a página "Mobile - +44 7700 900123".
- Vê uso atual: 18GB de 20GB usados (90%).
- Clica em "Adicionar Complemento" ou "Recarregar".

### 2. O Sistema Filtra os Complementos Disponíveis

Mostra apenas complementos onde:

- `category = "addon"`.
- `service_type = "mobile"` (corresponde ao tipo de serviço).
- `residential = true` (se o cliente for residencial).
- `enabled = true`.

O cliente vê: "5GB Data Boost - £5.00".

### 3. Cliente Seleciona o Complemento

- Clica em "5GB Data Boost".
- Confirma a compra por £5.00.
- O sistema captura a autorização de pagamento.

### 4. Provisionamento do Complemento Iniciado

O sistema chama `play_topup_charge_then_action.yaml` com variáveis:

```
{
  "product_id": 43, # Produto 5GB Data Boost
  "customer_id": 123,
  "service_id": 1234, # Serviço existente
  "access_token": "eyJ...",
  "data_gb": 5, # De provisioning_json_vars
  "days": 7 # De provisioning_json_vars
}
```

### Diferença Chave em Relação ao Autônomo:

- `service_id` está incluído (serviço existente a ser modificado).
- Nenhum inventário necessário.
- Nenhuma criação de serviço (modifica o existente).

# Análise do Playbook de Provisionamento de Complementos

## Tarefa 1: Buscar Detalhes do Serviço

```
- 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 que:

- Necessário service\_uuid para adicionar saldo à conta OCS correta.
- Verifica se o serviço existe e está ativo.
- Garante que o serviço pertença ao cliente.

## Tarefa 2: Cobrar o Cliente

```
- name: Get Customer's Default Payment Method
  uri:
    url: "http://localhost:5000/crm/stripe/customer_id/{{ customer_id }}"
    method: GET
    headers:
      Authorization: "Bearer {{ access_token }}"
  register: api_response_stripe

- name: Get default card ID
  set_fact:
    customer_stripe_id: "{{ api_response_stripe.json |
json_query(query) }}"
  vars:
    query: "data[?default_payment_method==`true`].customer_stripe_id
| [0]"

- name: Charge card
  uri:
    url: "http://localhost:5000/crm/stripe/charge_card/{{
customer_stripe_id }}"
    method: POST
```

```

body_format: json
headers:
  Authorization: "Bearer {{ access_token }}"
body:
  {
    "retail_cost": 5.00,
    "description": "5GB Data Boost",
    "customer_id": "{{ customer_id }}",
    "service_id": "{{ service_id }}",
    "product_id": "{{ product_id }}",
    "wholesale_cost": 1.50,
    "invoice": true
  }
register: charge_response
- name: Assert payment successful
  assert:
    that:
      - charge_response.status == 200

```

### O que isso faz:

- Encontra o método de pagamento padrão do cliente no Stripe.
- Cobra £5.00 no cartão.
- Registra o custo de atacado de £1.50 para rastreamento de margem.
- Cria transação vinculada ao serviço.
- Adiciona à próxima fatura.
- Falha o provisionamento se o pagamento falhar.

### Por que cobrar primeiro:

- Nenhum crédito entregue até que o pagamento seja confirmado.
- Previne fraudes.
- Combina pagamento com provisionamento do complemento.

### Tarefa 3: Adicionar Saldo de Dados à Conta 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
    }
}
}

```

#### O que isso faz:

- Adiciona 5GB (5368709120 bytes) à conta.
- Expira em 168 horas (7 dias).
- Peso 20 (peso mais alto consumido primeiro - impulso antes da cota mensal).

#### Saldo do Cliente Após o Complemento:

- Mensal original: 2GB restantes (expira em 25 dias).
- Novo impulso: 5GB (expira em 7 dias).
- Total disponível: 7GB.
- Ordem de uso: impulso consumido primeiro, depois mensal.

#### Tarefa 4: Registrar Transação

```

- 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
      }

```

#### O que isso faz:

- Registra a cobrança de £5.00 ao cliente.
- Registra o custo de atacado de £1.50.
- Vincula a transação ao serviço para relatórios.

## Resumo Completo do Fluxo de Complemento

1. O cliente seleciona o complemento da lista filtrada.
2. Pagamento autorizado e cobrado.
3. Saldo de dados adicionado à conta OCS.
4. Transação registrada no CRM.
5. O cliente imediatamente vê o saldo atualizado: 7GB disponíveis.

### Rastreamento Financeiro:

- Cobrança mensal do serviço: £15 varejo, £5 atacado.
- Compra do complemento: £5 varejo, £1.50 atacado.

## Renovação Automática: Complementos Recorrentes

Alguns complementos podem ser configurados para renovação automática (planos de dados mensais, assinaturas, etc.).

### Configuração do Produto:

```
{
  "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"
}
```

### Provisionamento Cria ActionPlan:

```
- 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
  }]
```

### O que isso faz:

- Cria uma tarefa agendada no OCS.
- Executa `Action_{{ product_slug }}` a cada 30 dias.
- A ação cobra o cliente e reaplica o saldo de dados.
- Continua até que o cliente cancele.

### Gestão do Cliente:

- O cliente vê "Próxima Renovação: 1 de fevereiro de 2025 - £10.00" na visualização do serviço.
- Pode clicar em "Cancelar Renovação Automática" para parar cobranças futuras.
- Pode clicar em "Renovar Agora" para aplicar imediatamente a cota do próximo mês.

## Etapa 4: Desprovisionamento de Serviços

Quando um cliente cancela o serviço, o sistema deve remover todos os recursos de forma limpa.

### Gatilhos de Desprovisionamento

O desprovisionamento pode ser acionado por:

1. **Cancelamento do Cliente** - O cliente clica em "Cancelar Serviço".
2. **Ação Administrativa** - A equipe marca o serviço para desativação.
3. **Não pagamento** - O serviço expira devido à falta de renovação.
4. **Fim do contrato** - O contrato de prazo fixo chega à data final.

### Fluxo de Desprovisionamento

#### 1. Cliente Inicia o Cancelamento

- Navega até o serviço.
- Clica em "Cancelar Serviço".
- O sistema solicita: "Você tem certeza? Qualquer saldo restante será perdido."
- O cliente confirma.

#### 2. Período de Carência (Opcional)

Alguns operadores implementam um período de carência:

- Serviço marcado como "Cancelamento Pendente".
- Permanece ativo por 7-30 dias.
- O cliente pode reverter o cancelamento durante o período de carência.
- Desprovisionamento automático após o período de carência.

### 3. Trabalho de Desprovisionamento Criado

O sistema cria um trabalho de provisão com:

```
{
  "action": "deprovision",
  "service_id": 1234,
  "customer_id": 123,
  "service_uuid": "Local_Mobile_SIM_a3f2c1d8"
}
```

Chama o playbook especificado em `service.deprovisioning_play` ou bloco de resgate do playbook original.

### 4. Playbook de Desprovisionamento do 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"
      }

```

### O que isso faz:

1. **Desabilita a conta OCS** - Para todas as cobranças, uso bloqueado.
2. **Remove ActionPlans** - Cancela renovações automáticas.
3. **Atualiza o serviço no CRM** - Status "Desativado", data registrada.
4. **Libera o inventário** - SIM marcado como "Descomissionado", disponível para reutilização (após reforma).

## 5. Pós-Desprovisionamento

O sistema realiza a limpeza:

- O cliente não vê mais o serviço no portal de autoatendimento.
- O serviço permanece no CRM para relatórios históricos.
- Transações e faturas preservadas para contabilidade.
- O inventário pode ser reformado e reutilizado.
- A conta OCS pode ser arquivada após o período de retenção.

## Desprovisionamento Parcial vs Total

### Desprovisionamento Parcial (Suspensão):

- Usado para não pagamento ou suspensão temporária.
- Conta OCS desabilitada, mas não excluída.

- Saldos preservados.
- Pode ser reativada quando o pagamento for recebido.

```
- name: Suspend Service
uri:
  url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
  method: POST
  body:
    {
      "method": "ApierV2.SetAccount",
      "params": [{
        "Account": "{{ service_uuid }}",
        "ExtraOptions": { "Disabled": true }
      }]
    }
}
```

### **Desprovisionamento Total (Cancelamento Permanente):**

- Usado para cancelamento permanente.
- Conta OCS excluída completamente.
- Saldos perdidos.
- Não pode ser reativada.

```
- name: Remove OCS Account
uri:
  url: "http://{{ crm_config.ocs.cgrates }}/jsonrpc"
  method: POST
  body:
    {
      "method": "ApierV2.RemoveAccount",
      "params": [{
        "Account": "{{ service_uuid }}"
      }]
    }
}
```

## **Melhores Práticas para Gestão de Produtos**

### **Gestão do Ciclo de Vida do Produto**

#### **Estados do Produto:**

- enabled: true - Produto disponível para novos pedidos.
- enabled: false - Produto desabilitado, serviços existentes continuam.

#### **Desabilitando Produtos:**

- Marque o produto como enabled: false para prevenir novos pedidos.
- Serviços existentes permanecem ativos.

- Os clientes ainda podem renovar/modificar serviços existentes.
- Útil para descontinuar produtos antigos.

## **Gestão de Inventário**

### **Estados do Inventário:**

- New - Estoque novo, pronto para atribuição.
- In Stock - Disponível para provisionamento.
- Assigned - Vinculado ao serviço do cliente.
- Decommissioned - Pode ser reformado e reutilizado.
- Damaged - Precisa de reparo ou descarte.

### **Reutilizando Inventário:**

Após o desprovisionamento:

- Cartões SIM: Reformar e marcar como "Em Estoque".
- Números de telefone: Liberar após o período de portabilidade (30 dias).
- Equipamentos: Testar, reformar, marcar como "Usado".

## **Métricas de Provisionamento**

### **Monitorar:**

- Taxa de sucesso de provisionamento.
- Tempo médio de provisionamento.
- Pontos de falha comuns.
- Rotatividade de inventário.



# Integração do Mailjet com o OmniCRM

O OmniCRM integra-se ao **Mailjet** para gerenciar toda a comunicação por e-mail com clientes e funcionários, garantindo a entrega de e-mails profissional, com marca e confiável, tanto para e-mails transacionais quanto para campanhas de marketing.

## Visão Geral

A integração do Mailjet fornece:

- **E-mails Transacionais Automatizados** - Redefinições de senha, faturas, e-mails de boas-vindas, notificações
- **Sincronização de Contatos** - Contatos de clientes sincronizados automaticamente com o Mailjet para campanhas
- **Modelos de E-mail** - Mais de 10 tipos de e-mails pré-configurados com modelos personalizáveis do Mailjet
- **Campanhas de Marketing** - Campanhas de e-mail segmentadas com base em dados de clientes
- **Entrega Confiável** - Infraestrutura de e-mail profissional com rastreamento e análises

## Configuração

O Mailjet é configurado em `OmniCRM-API/crm_config.yaml` na seção `mailjet`.

### Configuração Básica

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

### Obtendo Credenciais da API:

1. Crie uma conta em <<https://www.mailjet.com>>
2. Navegue até **Configurações da Conta** → **Chaves da API**
3. Copie **Chave da API** e **Chave Secreta**
4. Cole em `crm_config.yaml`



# Configuração do Modelo de E-mail

O OmniCRM usa **10 tipos distintos de modelos de e-mail** para comunicações automatizadas. Cada modelo é configurado com:

- **from\_email** - Endereço de e-mail do remetente
- **from\_name** - Nome de exibição do remetente
- **template\_id** - ID do modelo Mailjet (numérico)
- **subject** - Linha de assunto do e-mail

## Tipos de Modelo e Configuração

### E-mail de Boas-Vindas ao Cliente

Enviado quando uma nova conta de cliente é criada.

```
api_crmCommunicationCustomerWelcome:  
  from_email: "support@yourcompany.com"  
  from_name: "Suporte da Sua Empresa"  
  template_id: 5977509  
  subject: "Bem-vindo à Sua Empresa"
```

#### Quando Enviado:

- Novo cliente se inscreve pelo portal de Autoatendimento
- Funcionário cria nova conta de cliente
- Cliente ativa o serviço pela primeira vez

#### Variáveis do Modelo Disponíveis:

- {{ var:customer\_name }} - Nome completo do cliente
- {{ var:email }} - Endereço de e-mail do cliente
- {{ var:company\_name }} - Nome da sua empresa
- {{ var:login\_url }} - Link para o portal de Autoatendimento
- {{ var:support\_url }} - Link para a página de suporte

### E-mail de Fatura do Cliente

Enviado quando uma fatura é gerada e está pronta para pagamento.

```
api_crmCommunicationCustomerInvoice:  
  from_email: "billing@yourcompany.com"  
  from_name: "Cobrança da Sua Empresa"  
  template_id: 6759851  
  subject: "Sua Fatura - "
```

#### Quando Enviado:

- Fatura gerada automaticamente para o período de cobrança
- Fatura manual criada por funcionário
- Cliente solicita cópia da fatura

### **Variáveis do Modelo Disponíveis:**

- {{ var:customer\_name }} - Nome completo do cliente
- {{ var:invoice\_number }} - ID/número da fatura
- {{ var:invoice\_date }} - Data de emissão da fatura
- {{ var:due\_date }} - Data de vencimento do pagamento
- {{ var:total\_amount }} - Valor total devido
- {{ var:invoice\_url }} - Link para visualizar/download do PDF da fatura
- {{ var:pay\_url }} - Link para pagar a fatura online

### **Anexo da Fatura:**

O PDF da fatura é automaticamente anexado ao e-mail.

### **Lembrete de Fatura do Cliente**

Enviado para lembrar os clientes sobre faturas vencidas.

```
api_crmCommunicationCustomerInvoiceReminder:
  from_email: "billing@yourcompany.com"
  from_name: "Cobrança da Sua Empresa"
  template_id: 6759852
  subject: "Lembrete de Pagamento - Fatura Vencida"
```

### **Quando Enviado:**

- Fatura está X dias atrasada (configurável)
- Lembrete manual acionado por funcionário
- Fluxo de lembrete automatizado (se configurado)

### **Variáveis do Modelo Disponíveis:**

- {{ var:customer\_name }}
- {{ var:invoice\_number }}
- {{ var:due\_date }}
- {{ var:days\_overdue }}
- {{ var:total\_amount }}
- {{ var:pay\_url }}

### **E-mail de Boas-Vindas ao Usuário do Staff**

Enviado quando uma nova conta de usuário do staff é criada.

```
api_crmCommunicationUserWelcome:
```

```
from_email: "admin@yourcompany.com"
from_name: "Admin da Sua Empresa"
template_id: 5977510
subject: "Bem-vindo à Equipe"
```

### Quando Enviado:

- Admin cria novo usuário do staff
- Botão "Enviar E-mail de Boas-Vindas" clicado na gestão de usuários

### Variáveis do Modelo Disponíveis:

- {{ var:user\_name }} - Nome completo do usuário do staff
- {{ var:email }} - E-mail do usuário do staff
- {{ var:role }} - Função(s) atribuída(s)
- {{ var:login\_url }} - Link para login no portal de admin
- {{ var:temp\_password }} - Senha temporária (se aplicável)
- {{ var:support\_email }} - Contato de suporte de TI

### Redefinição de Senha do Usuário

Enviado quando um usuário solicita redefinir sua senha.

```
api_crmCommunicationUserPasswordReset:
from_email: "noreply@yourcompany.com"
from_name: "Segurança da Sua Empresa"
template_id: 5977511
subject: "Solicitação de Redefinição de Senha"
```

### Quando Enviado:

- Usuário clica em "Esqueci a Senha" na página de login
- Usuário envia solicitação de redefinição de senha

### Variáveis do Modelo Disponíveis:

- {{ var:user\_name }}
- {{ var:reset\_url }} - Link de redefinição de senha com tempo limitado (normalmente 1 hora)
- {{ var:expiry\_time }} - Quando o link de redefinição expira

### Nota de Segurança:

Os links de redefinição expiram após o período de tempo configurado (padrão 1 hora).

## Sucesso na Redefinição de Senha do Usuário

Enviado para confirmar que a senha foi alterada com sucesso.

```
api_crmCommunicationUserPasswordResetSuccess:  
  from_email: "noreply@yourcompany.com"  
  from_name: "Segurança da Sua Empresa"  
  template_id: 5977512  
  subject: "Senha Alterada com Sucesso"
```

### Quando Enviado:

- Usuário completa com sucesso a redefinição de senha
- Imediatamente após a nova senha ser definida

### Variáveis do Modelo Disponíveis:

- {{ var:user\_name }}
- {{ var:change\_date }} - Data/hora em que a senha foi alterada
- {{ var:ip\_address }} - Endereço IP da alteração (opcional)
- {{ var:support\_email }} - Contato se a alteração foi não autorizada

## Alteração de Senha do Usuário

Enviado quando um usuário altera sua senha nas configurações.

```
api_crmCommunicationUserPasswordChange:  
  from_email: "noreply@yourcompany.com"  
  from_name: "Segurança da Sua Empresa"  
  template_id: 5977513  
  subject: "Notificação de Alteração de Senha"
```

### Quando Enviado:

- Usuário altera a senha no perfil/configurações
- Admin redefine a senha do usuário

### Variáveis do Modelo Disponíveis:

- {{ var:user\_name }}
- {{ var:change\_date }}
- {{ var:changed\_by }} - "Próprio" ou nome do admin
- {{ var:support\_email }}

## Verificação de E-mail

Enviado para verificar o endereço de e-mail de um usuário.

```
api_crmCommunicationEmailVerification:
  from_email: "noreply@yourcompany.com"
  from_name: "Sua Empresa"
  template_id: 5977514
  subject: "Verifique Seu Endereço de E-mail"
```

### **Quando Enviado:**

- Nova conta criada (cliente ou funcionário)
- Usuário altera o endereço de e-mail
- Verificação de e-mail necessária por motivos de segurança

### **Variáveis do Modelo Disponíveis:**

- {{ var:user\_name }}
- {{ var:verification\_url }} - Link para verificar o e-mail
- {{ var:verification\_code }} - Código para inserir manualmente (alternativa ao link)

### **Notificação de Saldo Expirado**

Enviado quando o saldo/limite de serviço de um cliente expira.

```
api_crmCommunicationsBalanceExpired:
  from_email: "support@yourcompany.com"
  from_name: "Suporte da Sua Empresa"
  template_id: 5977515
  subject: "Seu Saldo de Serviço Expirou"
```

### **Quando Enviado:**

- Saldo pré-pago expira
- Data de renovação do limite mensal passou
- Data de expiração do serviço alcançada

### **Variáveis do Modelo Disponíveis:**

- {{ var:customer\_name }}
- {{ var:service\_name }} - Nome do serviço expirado
- {{ var:expiry\_date }}
- {{ var:balance\_type }} - "Dados", "Voz", "Monetário", etc.
- {{ var:renewal\_url }} - Link para renovar/recarregar

### **Alerta de Baixo Saldo**

Enviado quando o saldo de um cliente cai abaixo do limite configurado.

```
api_crmCommunicationsBalanceLow:
```

```
from_email: "support@yourcompany.com"
from_name: "Suporte da Sua Empresa"
template_id: 5977516
subject: "Alerta de Baixo Saldo"
```

### Quando Enviado:

- Saldo cai abaixo do limite (por exemplo, 20% restante)
- Configurado no plano de serviço ou OCS
- Monitoramento em tempo real aciona alerta

### Variáveis do Modelo Disponíveis:

- {{ var:customer\_name }}
- {{ var:service\_name }}
- {{ var:current\_balance }}
- {{ var:threshold }}
- {{ var:balance\_type }}
- {{ var:topup\_url }} - Link para adicionar saldo

## Criando Modelos de E-mail Mailjet

Para cada tipo de e-mail, você precisa criar um modelo correspondente no Mailjet.

### Passo 1: Criar Modelo no Mailjet

1. Faça login no painel do Mailjet
2. Navegue até **Modelos de E-mail → Modelos Transacionais**
3. Clique em **Criar um Novo Modelo**
4. Escolha **Codifique seu próprio modelo** (para usuários avançados) ou **Use o construtor de modelos**

### Passo 2: Projetar o Modelo

Use o construtor de arrastar e soltar do Mailjet ou o editor HTML para projetar seu e-mail.

### Elementos Essenciais:

- **Cabeçalho** - Logotipo e marca da empresa
- **Saudação** - Personalizada com {{ var:customer\_name }} ou {{ var:user\_name }}
- **Conteúdo** - Corpo da mensagem principal
- **Variáveis** - Insira variáveis do modelo da lista acima
- **Chamada para Ação** - Botões/links para ações do usuário
- **Rodapé** - Link para cancelar a inscrição, endereço da empresa, contato de suporte

## Exemplo de Modelo (Redefinição de Senha):

```
<!DOCTYPE html>
<html>
<head>
  <style>
    body { font-family: Arial, sans-serif; }
    .button { background-color: #4CAF50; color: white; padding:
14px 28px; }
  </style>
</head>
<body>
  

  <h2>Solicitação de Redefinição de Senha</h2>

  <p>Olá {{ var:user_name }},</p>

  <p>Recebemos uma solicitação para redefinir sua senha. Clique no
botão abaixo para criar uma nova senha:</p>

  <a href="{{ var:reset_url }}" class="button">Redefinir Senha</a>

  <p>Este link expira em {{ var:expiry_time }}.</p>

  <p>Se você não solicitou isso, por favor, ignore este e-mail.</p>

  <hr>
  <p style="font-size: 12px; color: #666;">
    Sua Empresa | support@yourcompany.com<br/>
    123 Business St, Cidade, País
  </p>
</body>
</html>
```

## Passo 3: Obter ID do Modelo

1. Salve o modelo no Mailjet
2. Anote o **ID do Modelo** (numérico, por exemplo, 5977509)
3. Copie este ID para `crm_config.yaml`

## Passo 4: Testar o Modelo

1. No Mailjet, use o recurso **E-mail de Teste**
2. Forneça valores de exemplo para todas as variáveis
3. Envie um e-mail de teste para você mesmo
4. Verifique a formatação, links e marca

## Passo 5: Configurar no OmniCRM

Adicione a configuração do modelo a `crm_config.yaml`:

```
mailjet:
  api_key: your_api_key
  api_secret: your_secret

  api_crmCommunicationUserPasswordReset:
    from_email: "noreply@yourcompany.com"
    from_name: "Segurança da Sua Empresa"
    template_id: 5977511
    subject: "Solicitação de Redefinição de Senha"
```

Reinicie a API do OmniCRM para que as alterações tenham efeito:

```
cd OmniCRM-API
sudo systemctl restart omnicrm-api
```

## Sincronização de Contatos

Todos os contatos de clientes no OmniCRM são sincronizados automaticamente com o Mailjet.

### O que é Sincronizado:

- Nome do contato
- Endereço de e-mail
- Tipo de contato (cobrança, técnico, etc.)
- Localização do cliente
- Campos personalizados

### Frequência de Sincronização:

Os contatos são sincronizados em tempo real quando:

- Novo cliente criado
- Contato adicionado/atualizado
- Detalhes do cliente modificados

## Solução de Problemas

### E-mail não enviado

- **Causa:** Credenciais da API inválidas, conta do Mailjet suspensa ou ID do modelo errado
- **Solução:**



- Verifique `api_key` e `api_secret` em `crm_config.yaml`
- Verifique o status e a cobrança da conta do Mailjet
- Verifique se o ID do modelo existe no Mailjet
- Verifique os logs da API em busca de erros

## Variáveis do modelo não substituídas

- **Causa:** Desvio no nome da variável ou dados ausentes no OmniCRM
- **Solução:**
  - Verifique se os nomes das variáveis correspondem exatamente (sensível a maiúsculas)
  - Use o formato `{{ var:variable_name }}`
  - Verifique se o OmniCRM está passando os dados da variável na chamada da API
  - Teste com dados de exemplo no Mailjet

## PDF da fatura não anexado

- **Causa:** Falha na geração do PDF ou tamanho do arquivo muito grande
- **Solução:**
  - Verifique os logs de geração da fatura
  - Verifique se o modelo da fatura é renderizado corretamente
  - Certifique-se de que o PDF tenha menos de 15MB (limite do Mailjet)
  - Teste a geração do PDF da fatura separadamente

## Contatos não sincronizados com o Mailjet

- **Causa:** Limite de taxa da API excedido ou serviço de sincronização não está em execução
- **Solução:**
  - Verifique os limites de taxa da API do Mailjet (200 chamadas/minuto)
  - Verifique se o serviço OmniCRM-API está em execução
  - Revise os logs de sincronização em busca de erros
  - Acione manualmente a sincronização para teste

## Documentação Relacionada

- `administration_configuration` - Referência completa da configuração do Mailjet
- `payments_invoices` - Geração de faturas e entrega de e-mails
- `authentication_flows` - E-mails de redefinição de senha e verificação
- `customer_care` - E-mails de boas-vindas do portal de Autoatendimento

## Leitura Adicional

- Documentação do Mailjet: <<https://dev.mailjet.com/>>
- Referência da API do Mailjet: <<https://dev.mailjet.com/email/reference/>>



# Processamento de Pagamentos com Stripe

OmniCRM integra-se com **Stripe** para lidar com o processamento de pagamentos eletrônicos, garantindo transações seguras e eficientes para os clientes.

Veja também: Métodos de Pagamento <payment\_methods>, Visão Geral de Faturamento <billing\_overview>, Faturas <payments\_invoices>, Transações <payments\_transaction>.

## Gerenciamento de Métodos de Pagamento

O sistema de Métodos de Pagamento do OmniCRM permite que clientes e funcionários gerenciem cartões de pagamento de forma segura usando a **integração com Stripe**. Os métodos de pagamento possibilitam faturamento automático para serviços, pagamentos únicos e cobranças recorrentes sem armazenar dados sensíveis do cartão no OmniCRM.

Veja também: Integração com Stripe <integrations\_stripe>, Visão Geral de Faturamento <billing\_overview>, Processamento de Pagamentos <payments\_process>, Faturas <payments\_invoices>.

## Visão Geral

O sistema de métodos de pagamento fornece:

- **Armazenamento Seguro de Cartões** - Cartões tokenizados pelo Stripe, nunca armazenados no OmniCRM
- **Múltiplos Cartões** - Clientes podem armazenar vários métodos de pagamento
- **Seleção Padrão** - Designar método de pagamento preferido para cobranças automáticas
- **Rastreamento de Validade** - Monitorar e atualizar cartões que estão prestes a expirar
- **Autoatendimento** - Clientes podem gerenciar seus próprios cartões através do Portal de Autoatendimento <self\_care\_portal>
- **Gerenciamento de Funcionários** - Funcionários de suporte podem adicionar/remover cartões em nome dos clientes

### Métodos de Pagamento Suportados:

- Cartões de Crédito (Visa, Mastercard, American Express, Discover)
- Cartões de Débito
- Cartões Pré-pagos (se suportados pela rede de cartões)

### Não Armazenado no OmniCRM:

Os detalhes do cartão são tokenizados pelo Stripe e armazenados de forma segura. O OmniCRM armazena apenas:

- Marca do cartão (Visa, Mastercard, etc.)
- Últimos 4 dígitos
- Mês/ano de validade
- Nome/nome de usuário do titular do cartão
- Referência do token do Stripe

# Acessando Métodos de Pagamento

## A partir da Página do Cliente:

1. Navegue até **Clientes** → [Selecionar Cliente]
2. Clique na aba **Faturamento**
3. Role até a seção **Métodos de Pagamento**

Ou diretamente:

## A partir do Painel de Cartões que Estão Prestes a Expirar:

Visualize todos os clientes com cartões que estão prestes a expirar:

Isso mostra uma lista de cartões em todo o sistema que expiram nos próximos 60 dias.

# Lista de Métodos de Pagamento

A tabela de métodos de pagamento exibe todos os cartões armazenados para um cliente:

Lista de Métodos de Pagamento

## Descrições das Colunas:

- **Nome de Usuário** - Nome amigável para o cartão (ex: "Cartão Pessoal", "Visa do Trabalho")
- **Emissor** - Marca do cartão e últimos 4 dígitos
- **Validade** - Mês/ano de expiração (formato MM/AAAA)
- **Adicionado** - Data em que o cartão foi adicionado à conta
- **Padrão** - Marca de seleção indica o método de pagamento padrão para cobranças automáticas

## Ações por Cartão:

Cada linha tem um menu de ações ( : ) com opções:

- **Definir como Padrão** - Faça deste o método de pagamento padrão
- **Excluir** - Remover cartão da conta

# Adicionando um Método de Pagamento

Clique em "**Adicionar Método de Pagamento**" para abrir o modal de pagamento do Stripe.

## Etapa 1: Inserir Detalhes do Cartão

O modal do Stripe Elements aparece:

Modal de Adicionar Método de Pagamento

## Campos Obrigatórios:

- **Informações do Cartão** - Número do cartão, validade, CVC (validado pelo Stripe)
- **Nome do Titular do Cartão** - Nome no cartão
- **País/Região** - País de faturamento

## Campos Opcionais:

- **Nome de Usuário do Cartão** - Rótulo amigável para distinguir entre cartões

### Segurança do Stripe Elements:

- Detalhes do cartão inseridos diretamente no iframe seguro do Stripe
- O OmniCRM nunca vê ou armazena números completos de cartões
- Conformidade com PCI DSS gerenciada pelo Stripe
- Validação em tempo real evita números de cartão inválidos

## Etapa 2: Enviar e Tokenizar

Quando você clica em "**Adicionar Método de Pagamento**":

1. **Validação do Lado do Cliente:**
  - O Stripe valida o formato do número do cartão
  - Verifica se a data de validade está no futuro
  - Verifica o formato do CVC
2. **Tokenização:**
  - Detalhes do cartão enviados diretamente ao Stripe (não ao OmniCRM)
  - O Stripe cria um token seguro (ex: pm\_1A2B3C4D)
  - Token retornado ao OmniCRM
3. **Processamento no Servidor:**
  - O OmniCRM salva o token no registro do cliente
  - Armazena os últimos 4 dígitos, marca, validade para exibição
  - Nenhum número completo do cartão toca nos servidores do OmniCRM

## Etapa 3: Confirmação

Mensagem de sucesso aparece:

Seu Visa terminando em 1234 foi adicionado à sua conta.

O novo cartão aparece na tabela de métodos de pagamento.

### Seleção Padrão Automática:

- Se este for o primeiro cartão do cliente, ele é automaticamente definido como padrão
- Se o cliente já tiver cartões, o novo cartão é adicionado como não padrão
- O cliente pode alterar o padrão após a adição

## Definindo o Método de Pagamento Padrão

O método de pagamento padrão é usado para:

- Cobranças automáticas de serviços recorrentes
- Pagamentos de faturas
- Recargas e reabastecimentos
- Transações únicas (a menos que especificado de outra forma)

### Para Alterar o Padrão:

1. Localize o cartão que você deseja definir como padrão na tabela de métodos de pagamento
2. Clique no **menu de ações ( ⋮ )** ao lado do cartão
3. Selecione "**Definir como Padrão**"

#### 4. A confirmação aparece

O Visa terminando em 5678 agora é seu método de pagamento padrão.

A marca de seleção se move para o cartão recém-selecionado.

#### **Indicador Visual:**

Cartões padrão mostram:

na coluna Padrão, tipicamente com um distintivo de marca de seleção verde.

## **Excluindo um Método de Pagamento**

Remova cartões que estão expirados, perdidos ou que não são mais necessários.

### **Etapas 1: Iniciar Exclusão**

1. Encontre o cartão a ser excluído na tabela de métodos de pagamento
2. Clique no **menu de ações ( ⋮ )**
3. Selecione **"Excluir"**

### **Etapas 2: Confirmar Exclusão**

Um modal de confirmação aparece:

Tem certeza de que deseja excluir este método de pagamento?

Cartão: Visa terminando em 1234 Validade: 12/2026

⚠ Aviso: Se este for seu único método de pagamento, você precisará adicionar um novo para continuar usando serviços que exigem faturamento automático.

[Cancelar] [Excluir Método de Pagamento]

Clique em **"Excluir Método de Pagamento"** para confirmar.

### **Etapas 3: Exclusão Completa**

Mensagem de sucesso:

O cartão é removido da tabela e excluído do Stripe.

#### **Restrições Importantes:**

- **Não é possível excluir o padrão se outros cartões existirem** - Defina um cartão diferente como padrão primeiro
- **Aviso se excluir o último cartão** - Serviços que exigem pagamento podem ser suspensos
- **Sem desfazer** - A exclusão é permanente; o cliente deve re-adicionar o cartão se necessário

## **Gerenciando Cartões que Estão Prestes a Expirar**

O OmniCRM rastreia as datas de validade dos cartões e fornece ferramentas para atualizar proativamente os cartões que estão prestes a expirar.

## Painel de Cartões que Estão Prestes a Expirar

Navegue até **Faturamento → Cartões que Estão Prestes a Expirar** para ver uma lista em todo o sistema:

Cliente Cartão Validade Dias Até Ação John Smith Visa **\*\*1234 02/2025 12 dias**  
**Atualizar Acme Corp MC5678 03/2025 45 dias** Atualizar Jane Doe Amex**\*\*9012 01/**  
2025 EXPIRADO Atualizar

Painel de Cartões que Estão Prestes a Expirar

### Filtros:

- **Intervalo de Validade** - Próximos 30/60/90 dias ou já expirados
- **Tipo de Cliente** - Individual vs Empresa
- **Tipo de Serviço** - Filtrar por serviço que exige método de pagamento

### Ações:

- **Atualizar** - Abre a página de métodos de pagamento do cliente para adicionar um novo cartão
- **Notificar** - Enviar lembrete por e-mail ao cliente (se Mailjet configurado)

## Notificações de Validade

Se o Mailjet estiver configurado, e-mails automáticos são enviados:

- **60 dias antes da validade** - Primeiro lembrete
- **30 dias antes da validade** - Segundo lembrete
- **7 dias antes da validade** - Aviso final
- **Na validade** - Aviso de que o cartão expirou

Os clientes podem clicar em um link no e-mail para atualizar seu método de pagamento através do portal de Autoatendimento.

### Variáveis do Modelo de E-mail:

Os modelos do Mailjet recebem:

- Nome do cliente
- Marca do cartão e últimos 4 dígitos
- Data de validade
- Link para a página de métodos de pagamento do Autoatendimento

Veja `integrations_mailjet` para configuração do modelo de e-mail.

## Atualizando um Cartão que Está Prestes a Expirar

### Fluxo de Trabalho Recomendado:

1. O cliente recebe um e-mail de notificação de validade
2. O cliente faz login no portal de Autoatendimento
3. Navega até **Faturamento → Métodos de Pagamento**
4. Clica em **"Adicionar Método de Pagamento"**
5. Insere os novos detalhes do cartão (mesmo cartão com validade atualizada ou cartão de substituição)
6. Define o novo cartão como padrão
7. Exclui o cartão antigo/expirado

## Fluxo de Trabalho do Funcionário:

Se o cliente ligar para o suporte:

1. O funcionário abre a conta do cliente
2. Navega até **Faturamento → Métodos de Pagamento**
3. Adiciona um novo cartão em nome do cliente (o cliente fornece os detalhes pelo telefone)
4. Define o novo cartão como padrão
5. Exclui o cartão expirado
6. Confirma com o cliente

Aviso

Nunca peça aos clientes para enviar detalhes do cartão por e-mail ou mensagem. Sempre use:

- Portal seguro de Autoatendimento para autoatendimento
- Telefone com funcionários inserindo detalhes diretamente no sistema
- Pessoalmente em local de varejo

## O que Acontece Quando os Cartões Expiram

Quando um cartão de pagamento atinge sua data de validade e não é atualizado:

### Efeitos Imediatos:

1. **Pagamentos Automáticos Falham**
  - O Stripe rejeita transações com cartões expirados
  - Renovações mensais de serviços falham ao processar
  - Recargas automáticas falham
  - Pagamentos automáticos de faturas falham
2. **Notificações ao Cliente**
  - O sistema tenta cobrar o cartão
  - Notificação de falha de pagamento enviada
  - E-mail "Atualizar Método de Pagamento" enviado com link para o portal de Autoatendimento
3. **Mudanças no Status do Serviço**
  - **Serviços Pós-Pagos** - Podem continuar temporariamente com saldo pendente
  - **Serviços Pré-Pagos** - Suspensão do serviço quando o saldo se esgota
  - **Serviços de Auto-Renovação** - A renovação falha, o serviço pode expirar

### Ações Subsequentes:

#### Dia 1-3 (Período de Graça):

- O serviço continua normalmente
- O cliente recebe o primeiro aviso de falha de pagamento
- O sistema tenta uma nova tentativa (dependendo da configuração)

#### Dia 4-7:

- Segunda tentativa de pagamento (se configurada)
- E-mail de aviso enviado
- O atendimento ao cliente pode entrar em contato com o cliente

#### Dia 8-14:

- O serviço pode ser suspenso por falta de pagamento
- O status suspenso impede o uso, mas preserva a conta

- O cliente pode restaurar atualizando o método de pagamento e pagando o saldo pendente

#### **Dia 15+:**

- O serviço pode ser encerrado por falta de pagamento
- Inventário (cartões SIM, equipamentos) marcado para devolução
- Aviso final enviado
- Conta encaminhada para cobrança (se aplicável)

#### **Prevenindo Interrupção do Serviço:**

Para evitar interrupção do serviço:

- Atualize os cartões **30 dias antes da validade**
- Adicione múltiplos métodos de pagamento para redundância
- Habilite alertas de falha de pagamento
- Monitore o painel de Cartões que Estão Prestes a Expirar semanalmente

#### **Restaurando o Serviço Após a Validade:**

Se o serviço foi suspenso devido a um cartão expirado:

1. Adicione um novo método de pagamento válido
2. Defina como padrão
3. Pague o saldo pendente (se houver)
4. Entre em contato com o suporte para reativar o serviço
5. Serviço restaurado em minutos a horas

## **Segurança do Método de Pagamento**

### **Tokenização**

O OmniCRM usa a tokenização do Stripe para garantir segurança:

1. **Cliente insere o cartão** → Enviado diretamente aos servidores do Stripe
2. **O Stripe valida e tokeniza** → Cria um token único
3. **Token armazenado no OmniCRM** → Número completo do cartão nunca armazenado
4. **Processamento de pagamento** → Token enviado ao Stripe, o Stripe cobra o cartão

#### **O que o OmniCRM Armazena:**

```
{
  "customer_stripe_id": "pm_1A2B3C4D5E6F",
  "brand": "visa",
  "last4": "1234",
  "exp_month": 12,
  "exp_year": 2026,
  "name": "John Smith",
  "nickname": "Cartão Pessoal",
  "is_default": true
}
```

#### **O que o OmniCRM NÃO Armazena:**

- Número completo do cartão
- Código CVV/CVC
- Dados da faixa magnética



- Números PIN

## Conformidade com PCI

Ao usar o Stripe Elements:

- **Escopo PCI reduzido** - Dados do cartão nunca tocam os servidores do OmniCRM
- **Campos hospedados pelo Stripe** - A entrada do cartão acontece no iframe do Stripe
- **Sem armazenamento de cartões** - Tokens usados em vez de dados brutos do cartão
- **Transmissão segura** - Toda comunicação via HTTPS/TLS

Veja `integrations_stripe` para detalhes de segurança do Stripe.

## Fluxos de Trabalho Comuns

### Fluxo de Trabalho 1: Cliente Adiciona Primeiro Método de Pagamento

**Cenário:** Novo cliente se inscrevendo para o serviço

1. O cliente cria uma conta
2. Seleciona um plano de serviço
3. Solicitado a adicionar um método de pagamento durante o checkout
4. Insere os detalhes do cartão no modal do Stripe
5. Cartão tokenizado e salvo
6. Definido automaticamente como padrão
7. Serviço provisionado
8. Primeiro pagamento processado

### Fluxo de Trabalho 2: Cliente Atualiza Cartão que Está Prestes a Expirar

**Cenário:** Cartão de crédito prestes a expirar

1. O cliente recebe notificação por e-mail (60 dias antes da validade)
2. Faz login no portal de Autoatendimento
3. Navega até **Faturamento → Métodos de Pagamento**
4. Revê o cartão atual que expira em 12/2025
5. Clica em **"Adicionar Método de Pagamento"**
6. Insere o cartão de substituição com validade 12/2028
7. Define o novo cartão como padrão
8. Exclui o cartão antigo
9. E-mail de confirmação enviado

### Fluxo de Trabalho 3: Funcionário Ajuda Cliente pelo Telefone

**Cenário:** Cliente liga: "Meu cartão foi recusado"

1. O cliente liga para o suporte
2. O funcionário verifica a identidade (perguntas de segurança)
3. O funcionário verifica os métodos de pagamento: Cartão expirado 01/2025
4. Funcionário: "Seu cartão expirou. Você tem um novo cartão?"
5. O cliente fornece os novos detalhes do cartão pelo telefone
6. O funcionário navega até **Clientes → [Cliente] → Faturamento**
7. Clica em **"Adicionar Método de Pagamento"**
8. Insere os detalhes do cartão enquanto o cliente os lê
9. Define o novo cartão como padrão
10. Exclui o cartão expirado

11. Tenta novamente o pagamento falhado
12. Confirma com o cliente: "Pagamento bem-sucedido, serviço restaurado"

## Fluxo de Trabalho 4: Cliente Empresarial com Múltiplos Cartões

**Cenário:** A empresa quer cartões diferentes para diferentes propósitos

1. O cliente empresarial adiciona o cartão principal (Visa terminando em 1111)
2. Define como padrão para cobranças mensais de serviços
3. Adiciona cartão de backup (Mastercard terminando em 2222) para recargas
4. Adiciona cartão de compras (Amex terminando em 3333) para compras de equipamentos
5. Ao fazer uma recarga, seleciona Mastercard manualmente no checkout
6. O Visa padrão ainda é usado para faturamento automático mensal

## Fluxo de Trabalho 5: Gerenciando Cartões que Estão Prestes a Expirar (Admin)

**Cenário:** Gerenciamento proativo de validade

1. O administrador navega até **Faturamento → Cartões que Estão Prestes a Expirar**
2. Filtros: "Próximos 30 dias"
3. Vê 15 clientes com cartões que estão prestes a expirar
4. Seleciona todos → **"Enviar E-mails de Lembrete"**
5. O Mailjet envia e-mails personalizados para cada cliente
6. Os clientes atualizam os cartões via Autoatendimento
7. O administrador revisa a lista 1 semana depois
8. Liga para os clientes restantes que não atualizaram
9. Auxilia com atualizações de cartões pelo telefone

## Solução de Problemas

### "Cartão recusado" ao adicionar método de pagamento

- **Causa:** O Stripe rejeitou o cartão (fundos insuficientes, prevenção de fraudes, recusa do emissor)
- **Solução:**
  - Tente um cartão diferente
  - Entre em contato com o emissor do cartão para autorizar a transação
  - Certifique-se de que o cartão suporta compras online
  - Verifique se o endereço de cobrança corresponde ao cartão registrado

### "Erro ao adicionar método de pagamento" (erro genérico)

- **Causa:** Erro da API do Stripe ou problema de rede
- **Solução:**
  - Atualize a página e tente novamente
  - Verifique a conexão com a internet
  - Verifique se a chave publicável do Stripe está correta em .env
  - Verifique o console do navegador para mensagens de erro específicas
  - Tente um navegador diferente (desative bloqueadores de anúncios)

### Não é possível excluir método de pagamento (botão desativado)

- **Causa:** Tentando excluir o cartão padrão ou é o único cartão
- **Solução:**
  - Defina um cartão diferente como padrão primeiro
  - Se for o único cartão, adicione um novo cartão antes de excluir

## **Cartão aparece como expirado, mas não na lista de "Cartões que Estão Prestes a Expirar"**

- **Causa:** O cartão expirou recentemente, cache não atualizado
- **Solução:**
  - Atualize a página
  - Verifique os filtros no painel de Cartões que Estão Prestes a Expirar
  - Cartões expirados podem se mover para uma visualização diferente

## **Novo cartão não aparece imediatamente**

- **Causa:** A página não foi atualizada após a adição do cartão
- **Solução:**
  - A tabela de métodos de pagamento deve atualizar automaticamente
  - Se não, atualize manualmente o navegador
  - Verifique se ocorreu um erro durante o processo de adição

## **Modal do Stripe não carrega**

- **Causa:** Biblioteca Stripe.js não carregando, problema com a chave da API ou extensão do navegador bloqueando
- **Solução:**
  - Verifique o console do navegador para erros
  - Desative bloqueadores de anúncios e proteção contra rastreamento
  - Verifique REACT\_APP\_STRIPE\_PUBLISHABLE\_KEY em .env
  - Certifique-se de que o script Stripe.js carrega (verifique a aba de Rede)
  - Tente no modo de navegação privada/incógnita

## **Cliente não recebe notificações de validade**

- **Causa:** Mailjet não configurado ou modelo de e-mail ausente
- **Solução:**
  - Verifique as credenciais do Mailjet em crm\_config.yaml
  - Verifique se o modelo de e-mail existe para validade do cartão
  - Confirme se o endereço de e-mail do cliente é válido
  - Verifique os logs do Mailjet para falhas de entrega

# **Melhores Práticas**

### **Para Clientes:**

- Adicione o método de pagamento antes da ativação do serviço para evitar atrasos
- Mantenha pelo menos 2 cartões registrados para redundância
- Atualize cartões que estão prestes a expirar com 30+ dias de antecedência
- Exclua cartões antigos/expirados para evitar confusão
- Use nomes de usuário descritivos ("Visa Pessoal", "Amex do Trabalho")
- Verifique se o método de pagamento padrão está correto para faturamento automático

### **Para Funcionários de Suporte:**

- Verifique a identidade do cliente antes de acessar métodos de pagamento
- Nunca peça aos clientes para enviar detalhes do cartão por e-mail/SMS/chat
- Processe adições de cartões imediatamente durante as chamadas (não adie)
- Confirme se o novo cartão está definido como padrão após a adição
- Exclua cartões antigos apenas após confirmar que o novo cartão funciona
- Teste o pagamento após atualizar o cartão expirado (processo de autorização de £0.01)

## Para Administradores:

- Monitore o painel de Cartões que Estão Prestes a Expirar semanalmente
- Envie e-mails de lembrete 60/30/7 dias antes da validade
- Mantenha as chaves de teste/produção do Stripe separadas para dev vs produção
- Certifique-se de que os modelos do Mailjet estão configurados para notificações de validade
- Revise relatórios de falhas de pagamento para identificar cartões expirados
- Treine a equipe sobre procedimentos seguros de manuseio de cartões

## Melhores Práticas de Segurança:

- Use apenas chaves públicas do Stripe (nunca chaves secretas no frontend)
- Certifique-se de que todas as páginas de pagamento carreguem via HTTPS
- Revise regularmente o painel do Stripe para atividade suspeita
- Habilite regras de prevenção de fraudes do Stripe
- Exija CVC para todas as transações sem cartão presente
- Registre alterações no método de pagamento no log de atividades

## Documentação Relacionada

- `integrations_stripe` - Configuração e configuração da integração com Stripe
- `payments_process` - Processamento de pagamentos com métodos de pagamento armazenados
- `payments_invoices` - Pagamento automático de faturas usando o cartão padrão
- `features_topup_recharge` - Sistema de recarga usando métodos de pagamento
- `basics_payment` - Conceitos gerais de pagamento e faturamento
- `customer_care` - Portal de Autoatendimento para clientes gerenciarem seus próprios cartões

### Interface de Gerenciamento de Métodos de Pagamento

O Stripe é usado para gerenciar tanto o armazenamento de cartões quanto as compras, enquanto métricas financeiras detalhadas estão disponíveis diretamente através do painel do Stripe.

Painel do Stripe

## Armazenamento de Cartões e Pagamentos

No OmniCRM, os cartões dos clientes são armazenados como **Tokens do Stripe**. Esses tokens permitem o armazenamento seguro e o manuseio de informações de pagamento sensíveis sem armazenar detalhes reais do cartão dentro do OmniCRM. Isso fornece uma camada adicional de segurança, já que os dados sensíveis do cartão são gerenciados inteiramente pelo Stripe.

Quando um cliente faz uma compra, o OmniCRM usa esses tokens do Stripe para processar pagamentos de forma contínua através da plataforma Stripe, garantindo que todas as transações sejam seguras e estejam em conformidade com os padrões da indústria.

Ao acionar transações via API `</concepts_api>` ou pela interface do OmniCRM, o sistema processará automaticamente os pagamentos usando os tokens do Stripe armazenados. Isso permite um processamento de pagamento rápido e seguro, garantindo uma experiência tranquila tanto para os operadores quanto para os clientes.

Painel do Stripe

# Métricas Financeiras

Para análises e relatórios financeiros detalhados, os operadores podem acessar o **Painel do Stripe**. Este painel fornece insights sobre transações, receita, taxas e outras métricas importantes relacionadas ao processamento de pagamentos.

Painel do Stripe

Algumas das principais métricas disponíveis no Painel do Stripe incluem:

- **Receita e Pagamentos:** Visualize a receita total, pagamentos bem-sucedidos e valores reembolsados.
- **Taxas de Transação:** Relatórios detalhados sobre taxas aplicadas a cada transação.
- **Disputas e Chargebacks:** Métricas sobre quaisquer disputas ou chargebacks, permitindo que você rastreie e gerencie riscos financeiros.

Os operadores podem usar o painel do Stripe para obter uma visão abrangente de seu desempenho financeiro, facilitando o monitoramento de vendas, gerenciamento de pagamentos e acompanhamento da saúde geral dos negócios.

## Acessando o Painel do Stripe

Para relatórios financeiros mais detalhados, navegue até o Painel do Stripe em:

<<https://dashboard.stripe.com/>>

Este painel fornece informações financeiras em tempo real, garantindo total transparência sobre todas as transações processadas através do Stripe.



# Registro de Alterações

Isso contém as últimas 50 alterações feitas na pilha de software OmniCRM ou em suas dependências.

Nota: Isso não rastreia alterações em configurações individuais de clientes.



# Gerenciamento de Métodos de Pagamento

O sistema de Métodos de Pagamento do OmniCRM permite que clientes e funcionários gerenciem cartões de pagamento de forma segura usando **integração com Stripe**. Os métodos de pagamento permitem cobrança automática por serviços, pagamentos únicos e cobranças recorrentes sem armazenar dados sensíveis do cartão no OmniCRM.

Veja também: Stripe Integration <integrations\_stripe>, Billing Overview <billing\_overview>, Payment Processing <payments\_process>, Invoices <payments\_invoices>.

## Visão Geral

O sistema de métodos de pagamento fornece:

- **Armazenamento Seguro de Cartões** - Cartões tokenizados pelo Stripe, nunca armazenados no OmniCRM
- **Múltiplos Cartões** - Clientes podem armazenar múltiplos métodos de pagamento
- **Seleção Padrão** - Designar método de pagamento preferido para cobranças automáticas
- **Monitoramento de Vencimento** - Monitorar e atualizar cartões que estão prestes a vencer
- **Autoatendimento** - Clientes podem gerenciar seus próprios cartões através do Self-Care Portal <self\_care\_portal>
- **Gerenciamento de Funcionários** - Funcionários de suporte podem adicionar/remover cartões em nome dos clientes

### Métodos de Pagamento Suportados:

- Cartões de Crédito (Visa, Mastercard, American Express, Discover)
- Cartões de Débito
- Cartões Pré-pagos (se suportados pela rede de cartões)

### Não Armazenado no OmniCRM:

Os detalhes do cartão são tokenizados pelo Stripe e armazenados de forma segura. O OmniCRM armazena apenas:

- Marca do cartão (Visa, Mastercard, etc.)
- Últimos 4 dígitos
- Mês/ano de validade
- Nome/nome de usuário do titular do cartão
- Referência do token do Stripe

## Acessando Métodos de Pagamento

### A partir da Página do Cliente:

1. Navegue até **Clientes** → [Selecionar Cliente]
2. Clique na aba **Faturamento**
3. Role até a seção **Métodos de Pagamento**

Ou diretamente:

## A partir do Painel de Cartões Vencendo:

Veja todos os clientes com cartões prestes a vencer:

Isso mostra uma lista de cartões que estão vencendo nos próximos 60 dias.

## Lista de Métodos de Pagamento

A tabela de métodos de pagamento exibe todos os cartões armazenados para um cliente:

Lista de Métodos de Pagamento

### Descrições das Colunas:

- **Nome de Usuário** - Nome amigável para o cartão (ex: "Cartão Pessoal", "Visa do Trabalho")
- **Emissor** - Marca do cartão e últimos 4 dígitos
- **Vencimento** - Mês/ano de validade (formato MM/AAAA)
- **Adicionado** - Data em que o cartão foi adicionado à conta
- **Padrão** - Marca de seleção indica o método de pagamento padrão para cobranças automáticas

### Ações por Cartão:

Cada linha tem um menu de ações ( : ) com opções:

- **Definir como Padrão** - Tornar este o método de pagamento padrão
- **Excluir** - Remover cartão da conta

## Adicionando um Método de Pagamento

Clique em "**Adicionar Método de Pagamento**" para abrir o modal de pagamento do Stripe.

### Passo 1: Inserir Detalhes do Cartão

O modal do Stripe Elements aparece:

Modal de Adicionar Método de Pagamento

### Campos Obrigatórios:

- **Informações do Cartão** - Número do cartão, validade, CVC (validado pelo Stripe)
- **Nome do Titular do Cartão** - Nome no cartão
- **País/Região** - País de faturamento

### Campos Opcionais:

- **Nome de Usuário do Cartão** - Rótulo amigável para distinguir entre cartões

### Segurança do Stripe Elements:

- Detalhes do cartão inseridos diretamente no iframe seguro do Stripe
- O OmniCRM nunca vê ou armazena números de cartão completos
- Conformidade com PCI DSS gerenciada pelo Stripe
- Validação em tempo real evita números de cartão inválidos



## Passo 2: Enviar e Tokenizar

Quando você clica em "**Adicionar Método de Pagamento**":

1. **Validação do Lado do Cliente:**
  - O Stripe valida o formato do número do cartão
  - Verifica se a data de validade está no futuro
  - Verifica o formato do CVC
2. **Tokenização:**
  - Detalhes do cartão enviados diretamente para o Stripe (não para o OmniCRM)
  - O Stripe cria um token seguro (ex: pm\_1A2B3C4D)
  - Token retornado ao OmniCRM
3. **Processamento no Servidor:**
  - O OmniCRM salva o token no registro do cliente
  - Armazena os últimos 4 dígitos, marca, validade para exibição
  - Nenhum número de cartão completo toca os servidores do OmniCRM

## Passo 3: Confirmação

Mensagem de sucesso aparece:

Seu Visa terminando em 1234 foi adicionado à sua conta.

O novo cartão aparece na tabela de métodos de pagamento.

### Seleção Automática Padrão:

- Se este for o primeiro cartão do cliente, ele é automaticamente definido como padrão
- Se o cliente já tiver cartões, o novo cartão é adicionado como não padrão
- O cliente pode mudar o padrão após adicionar

## Definindo o Método de Pagamento Padrão

O método de pagamento padrão é usado para:

- Cobranças automáticas de serviços recorrentes
- Pagamentos de faturas
- Recargas e reabastecimentos
- Transações únicas (a menos que especificado de outra forma)

### Para Mudar o Padrão:

1. Localize o cartão que você deseja definir como padrão na tabela de métodos de pagamento
2. Clique no **menu de ações ( ⋮ )** ao lado do cartão
3. Selecione "**Definir como Padrão**"
4. A confirmação aparece

Visa terminando em 5678 é agora seu método de pagamento padrão.

A marca de seleção se move para o cartão recém-selecionado.

### Indicador Visual:

Cartões padrão mostram:

na coluna Padrão, geralmente com um distintivo de marca de seleção verde.

## Excluindo um Método de Pagamento

Remova cartões que estão vencidos, perdidos ou que não são mais necessários.

### Passo 1: Iniciar Exclusão

1. Encontre o cartão a ser excluído na tabela de métodos de pagamento
2. Clique no **menu de ações ( : )**
3. Selecione **"Excluir"**

### Passo 2: Confirmar Exclusão

Um modal de confirmação aparece:

Você tem certeza de que deseja excluir este método de pagamento?

Cartão: Visa terminando em 1234 Vencimento: 12/2026

⚠ Aviso: Se este for seu único método de pagamento, você precisará adicionar um novo para continuar usando serviços que requerem cobrança automática.

[Cancelar] [Excluir Método de Pagamento]

Clique em **"Excluir Método de Pagamento"** para confirmar.

### Passo 3: Exclusão Completa

Mensagem de sucesso:

O cartão é removido da tabela e excluído do Stripe.

#### Restrições Importantes:

- **Não é possível excluir o padrão se outros cartões existirem** - Defina um cartão diferente como padrão primeiro
- **Aviso ao excluir o último cartão** - Serviços que requerem pagamento podem ser suspensos
- **Sem desfazer** - A exclusão é permanente; o cliente deve re-adicionar o cartão se necessário

## Gerenciando Cartões Vencendo

O OmniCRM rastreia datas de vencimento dos cartões e fornece ferramentas para atualizar proativamente os cartões que estão prestes a vencer.

### Painel de Cartões Vencendo

Navegue até **Faturamento → Cartões Vencendo** para ver uma lista em todo o sistema:

Cliente Cartão Vencimento Dias Até Ação John Smith Visa **\*\*1234 02/2025 12 dias**  
**Atualizar Acme Corp MC5678 03/2025 45 dias Atualizar Jane Doe Amex\*\*9012 01/**  
2025 VENCIDO Atualizar

Painel de Cartões Vencendo

## Filtros:

- **Intervalo de Vencimento** - Próximos 30/60/90 dias ou já expirados
- **Tipo de Cliente** - Individual vs Empresa
- **Tipo de Serviço** - Filtrar por serviço que requer método de pagamento

## Ações:

- **Atualizar** - Abre a página de métodos de pagamento do cliente para adicionar um novo cartão
- **Notificar** - Enviar lembrete por e-mail ao cliente (se Mailjet configurado)

## Notificações de Vencimento

Se o Mailjet estiver configurado, e-mails automáticos são enviados:

- **60 dias antes do vencimento** - Primeiro lembrete
- **30 dias antes do vencimento** - Segundo lembrete
- **7 dias antes do vencimento** - Aviso final
- **No vencimento** - Aviso de que o cartão expirou

Os clientes podem clicar em um link no e-mail para atualizar seu método de pagamento através do portal de autoatendimento.

## Variáveis de Template de E-mail:

Os templates do Mailjet recebem:

- Nome do cliente
- Marca do cartão e últimos 4 dígitos
- Data de vencimento
- Link para a página de métodos de pagamento do Self-Care

Veja `integrations_mailjet` para configuração do template de e-mail.

## Atualizando um Cartão Vencendo

### Fluxo de Trabalho Recomendado:

1. O cliente recebe e-mail de notificação de vencimento
2. O cliente faz login no portal de autoatendimento
3. Navega até **Faturamento → Métodos de Pagamento**
4. Clica em "**Adicionar Método de Pagamento**"
5. Insere os novos detalhes do cartão (mesmo cartão com vencimento atualizado, ou cartão de substituição)
6. Define o novo cartão como padrão
7. Exclui o cartão antigo/vencido

### Fluxo de Trabalho do Funcionário:

Se o cliente ligar para o suporte:

1. O funcionário abre a conta do cliente
2. Navega até **Faturamento → Métodos de Pagamento**
3. Adiciona um novo cartão em nome do cliente (o cliente fornece os detalhes por telefone)
4. Define o novo cartão como padrão
5. Exclui o cartão vencido
6. Confirma com o cliente

## Aviso

Nunca peça aos clientes para enviar detalhes do cartão por e-mail ou mensagem. Sempre use:

- Portal seguro de autoatendimento para autoatendimento
- Telefone com funcionários inserindo detalhes diretamente no sistema
- Pessoalmente em local de varejo

## O Que Acontece Quando os Cartões Expiram

Quando um cartão de pagamento atinge sua data de vencimento e não é atualizado:

### Efeitos Imediatos:

#### 1. Pagamentos Automáticos Falham

- O Stripe rejeita transações com cartões expirados
- Renovações de serviços mensais falham ao processar
- Recargas automáticas falham
- Pagamentos automáticos de faturas falham

#### 2. Notificações ao Cliente

- O sistema tenta cobrar o cartão
- Notificação de falha de pagamento enviada
- E-mail "Atualizar Método de Pagamento" enviado com link para o portal de autoatendimento

#### 3. Mudanças no Status do Serviço

- **Serviços Pós-Pagos** - Podem continuar temporariamente com saldo pendente
- **Serviços Pré-Pagos** - Suspensão do serviço quando o saldo se esgota
- **Serviços de Renovação Automática** - A renovação falha, o serviço pode expirar

### Ações Subsequentes:

#### Dia 1-3 (Período de Graça):

- O serviço continua normalmente
- O cliente recebe o primeiro aviso de falha de pagamento
- O sistema tenta uma nova tentativa (dependendo da configuração)

#### Dia 4-7:

- Segunda tentativa de pagamento (se configurado)
- E-mail de aviso enviado
- O atendimento ao cliente pode entrar em contato com o cliente

#### Dia 8-14:

- O serviço pode ser suspenso por falta de pagamento
- O status suspenso impede o uso, mas preserva a conta
- O cliente pode restaurar atualizando o método de pagamento e pagando o saldo pendente

#### Dia 15+:

- O serviço pode ser encerrado por falta de pagamento
- Inventário (cartões SIM, equipamentos) marcado para devolução
- Aviso final enviado
- Conta encaminhada para cobrança (se aplicável)

### Prevenindo Interrupção do Serviço:

Para evitar interrupção do serviço:

- Atualize cartões **30 dias antes do vencimento**
- Adicione múltiplos métodos de pagamento para redundância
- Habilite alertas de falha de pagamento
- Monitore o painel de Cartões Vencendo semanalmente

### Restaurando o Serviço Após o Vencimento:

Se o serviço foi suspenso devido a um cartão expirado:

1. Adicione um novo método de pagamento válido
2. Defina como padrão
3. Pague o saldo pendente (se houver)
4. Entre em contato com o suporte para reativar o serviço
5. Serviço restaurado em minutos a horas

## Segurança do Método de Pagamento

### Tokenização

O OmniCRM usa a tokenização do Stripe para garantir segurança:

1. **O cliente insere o cartão** → Enviado diretamente para os servidores do Stripe
2. **O Stripe valida e tokeniza** → Cria um token único
3. **Token armazenado no OmniCRM** → Número completo do cartão nunca é armazenado
4. **Processamento de pagamento** → Token enviado ao Stripe, Stripe cobra o cartão

### O Que o OmniCRM Armazena:

```
{
  "customer_stripe_id": "pm_1A2B3C4D5E6F",
  "brand": "visa",
  "last4": "1234",
  "exp_month": 12,
  "exp_year": 2026,
  "name": "John Smith",
  "nickname": "Personal Card",
  "is_default": true
}
```

### O Que o OmniCRM NÃO Armazena:

- Número completo do cartão
- Código CVV/CVC
- Dados da faixa magnética
- Números PIN

## Conformidade com PCI

Ao usar o Stripe Elements:

- **Escopo PCI reduzido** - Dados do cartão nunca tocam os servidores do OmniCRM
- **Campos hospedados pelo Stripe** - A entrada do cartão ocorre no iframe do Stripe
- **Sem armazenamento de cartões** - Tokens usados em vez de dados brutos do cartão
- **Transmissão segura** - Toda comunicação via HTTPS/TLS

Veja `integrations_stripe` para detalhes de segurança do Stripe.

## Fluxos de Trabalho Comuns

### Fluxo de Trabalho 1: Cliente Adiciona Primeiro Método de Pagamento

**Cenário:** Novo cliente se inscrevendo para o serviço

1. O cliente cria uma conta
2. Seleciona o plano de serviço
3. Solicitado a adicionar método de pagamento durante o checkout
4. Insere detalhes do cartão no modal do Stripe
5. Cartão tokenizado e salvo
6. Automaticamente definido como padrão
7. Serviço provisionado
8. Primeira cobrança processada

### Fluxo de Trabalho 2: Cliente Atualiza Cartão Vencendo

**Cenário:** Cartão de crédito prestes a expirar

1. O cliente recebe notificação por e-mail (60 dias antes do vencimento)
2. Faz login no portal de autoatendimento
3. Navega até **Faturamento** → **Métodos de Pagamento**
4. Revê cartão atual vencendo 12/2025
5. Clica em **"Adicionar Método de Pagamento"**
6. Insere cartão de substituição com vencimento 12/2028
7. Define novo cartão como padrão
8. Exclui cartão antigo
9. E-mail de confirmação enviado

### Fluxo de Trabalho 3: Funcionário Ajuda Cliente Por Telefone

**Cenário:** Cliente liga: "Meu cartão foi recusado"

1. O cliente liga para o suporte
2. Funcionário verifica a identidade (perguntas de segurança)
3. Funcionário verifica métodos de pagamento: Cartão expirado 01/2025
4. Funcionário: "Seu cartão expirou. Você tem um novo cartão?"
5. O cliente fornece novos detalhes do cartão por telefone
6. Funcionário navega até **Clientes** → **[Cliente]** → **Faturamento**
7. Clica em **"Adicionar Método de Pagamento"**
8. Insere detalhes do cartão enquanto o cliente os lê
9. Define novo cartão como padrão
10. Exclui cartão expirado
11. Tenta pagamento falhado novamente
12. Confirma com o cliente: "Pagamento bem-sucedido, serviço restaurado"

### Fluxo de Trabalho 4: Cliente Empresarial com Múltiplos Cartões

**Cenário:** A empresa quer cartões diferentes para diferentes propósitos

1. Cliente empresarial adiciona cartão principal (Visa terminando em 1111)
2. Define como padrão para cobranças mensais de serviços
3. Adiciona cartão de backup (Mastercard terminando em 2222) para recargas
4. Adiciona cartão de compras (Amex terminando em 3333) para compras de equipamentos

5. Ao fazer recarga, seleciona Mastercard manualmente no checkout
6. Visa padrão ainda é usado para cobrança automática mensal

## Fluxo de Trabalho 5: Gerenciando Cartões Vencendo (Admin)

**Cenário:** Gerenciamento proativo de vencimentos

1. O administrador navega até **Faturamento → Cartões Vencendo**
2. Filtros: "Próximos 30 dias"
3. Vê 15 clientes com cartões prestes a vencer
4. Seleciona todos → **"Enviar E-mails de Lembrete"**
5. O Mailjet envia e-mails personalizados para cada cliente
6. Os clientes atualizam cartões via Self-Care
7. O administrador revisa a lista 1 semana depois
8. Liga para os clientes restantes que não atualizaram
9. Auxilia com atualizações de cartões por telefone

## Solução de Problemas

### "Cartão recusado" ao adicionar método de pagamento

- **Causa:** O Stripe rejeitou o cartão (fundos insuficientes, prevenção de fraudes, recusa do emissor)
- **Solução:**
  - Tente um cartão diferente
  - Entre em contato com o emissor do cartão para autorizar a transação
  - Certifique-se de que o cartão suporta compras online
  - Verifique se o endereço de faturamento corresponde ao cartão registrado

### "Erro ao adicionar método de pagamento" (erro genérico)

- **Causa:** Erro da API do Stripe ou problema de rede
- **Solução:**
  - Atualize a página e tente novamente
  - Verifique a conexão com a internet
  - Verifique se a chave publicável do Stripe está correta em .env
  - Verifique o console do navegador para mensagens de erro específicas
  - Tente um navegador diferente (desative bloqueadores de anúncios)

### Não é possível excluir método de pagamento (botão desativado)

- **Causa:** Tentando excluir o cartão padrão ou é o único cartão
- **Solução:**
  - Defina um cartão diferente como padrão primeiro
  - Se for o único cartão, adicione um novo cartão antes de excluir

### Cartão aparece como expirado, mas não na lista de "Cartões Vencendo"

- **Causa:** O cartão expirou recentemente, cache não atualizado
- **Solução:**
  - Atualize a página
  - Verifique os filtros no painel de Cartões Vencendo
  - Cartões expirados podem se mover para uma visualização diferente

### Novo cartão não aparece imediatamente

- **Causa:** A página não foi atualizada após adicionar o cartão

- **Solução:**
  - A tabela de métodos de pagamento deve atualizar automaticamente
  - Se não, atualize manualmente o navegador
  - Verifique se ocorreu um erro durante o processo de adição

### **Modal do Stripe não carrega**

- **Causa:** A biblioteca Stripe.js não está carregando, problema com a chave da API ou extensão do navegador bloqueando
- **Solução:**
  - Verifique o console do navegador para erros
  - Desative bloqueadores de anúncios e proteção de rastreamento
  - Verifique REACT\_APP\_STRIPE\_PUBLISHABLE\_KEY em .env
  - Certifique-se de que o script Stripe.js carrega (verifique a aba de Rede)
  - Tente o modo de navegação anônima/privada

### **Cliente não recebe notificações de vencimento**

- **Causa:** Mailjet não configurado ou template de e-mail ausente
- **Solução:**
  - Verifique as credenciais do Mailjet em crm\_config.yaml
  - Verifique se o template de e-mail existe para vencimento de cartão
  - Confirme se o endereço de e-mail do cliente é válido
  - Verifique os logs do Mailjet para falhas de entrega

## **Melhores Práticas**

### **Para Clientes:**

- Adicione método de pagamento antes da ativação do serviço para evitar atrasos
- Mantenha pelo menos 2 cartões registrados para redundância
- Atualize cartões prestes a vencer com 30+ dias de antecedência
- Exclua cartões antigos/vencidos para evitar confusão
- Use nomes de usuário descritivos ("Visa Pessoal", "Amex do Trabalho")
- Verifique se o método de pagamento padrão está correto para cobrança automática

### **Para Funcionários de Suporte:**

- Verifique a identidade do cliente antes de acessar métodos de pagamento
- Nunca peça aos clientes para enviar detalhes do cartão por e-mail/SMS/chat
- Processar adições de cartões imediatamente durante as chamadas (não adiar)
- Confirme se o novo cartão está definido como padrão após a adição
- Exclua cartões antigos apenas após confirmar que o novo cartão funciona
- Teste o pagamento após atualizar o cartão expirado (processar autorização de £0.01)

### **Para Administradores:**

- Monitore o painel de Cartões Vencendo semanalmente
- Envie e-mails de lembrete 60/30/7 dias antes do vencimento
- Mantenha as chaves de teste/produção do Stripe separadas para dev vs produção
- Certifique-se de que os templates do Mailjet estão configurados para notificações de vencimento
- Revise relatórios de pagamento falhado para identificar cartões expirados
- Treine a equipe sobre procedimentos seguros de manuseio de cartões

### **Melhores Práticas de Segurança:**



- Use apenas chaves publicáveis do Stripe (nunca chaves secretas no frontend)
- Certifique-se de que todas as páginas de pagamento carreguem via HTTPS
- Revise regularmente o painel do Stripe para atividade suspeita
- Habilite regras de prevenção de fraudes do Stripe
- Exija CVC para todas as transações sem cartão presente
- Registre alterações no método de pagamento no log de atividades

## **Documentação Relacionada**

- `integrations_stripe` - Configuração e configuração da integração com Stripe
- `payments_process` - Processando pagamentos com métodos de pagamento armazenados
- `payments_invoices` - Pagamento automático de faturas usando o cartão padrão
- `features_topup_recharge` - Sistema de recarga usando métodos de pagamento
- `basics_payment` - Conceitos gerais de pagamento e faturamento
- `customer_care` - Portal de autoatendimento para clientes gerenciarem seus próprios cartões

Interface de Gerenciamento de Métodos de Pagamento



# Faturas de Clientes

Transactions </payments\_transaction> são agrupadas para formar uma fatura, que é enviada ao cliente para pagamento.

As faturas têm uma data de início e uma data de término, que é o período que a fatura cobre, e uma data de vencimento, que é a data em que a fatura deve ser paga.

Gerar uma Fatura Proforma

As faturas podem ser geradas automaticamente pelo sistema, por exemplo, quando um serviço é cobrado, uma fatura é criada para o custo de varejo, ou podem ser criadas manualmente, por exemplo, se um cliente solicitar uma cópia de uma fatura, ou se um cliente for cobrado por uma taxa única.

As faturas de clientes são totalmente modeladas com Mailjet <integrations\_mailjet> e podem ser personalizadas para incluir o logotipo da empresa, endereço e detalhes de pagamento, e podem ser enviadas ao cliente por e-mail, ou baixadas como um PDF.

Registro de Atividades

## Personalizando Modelos de Faturas

OmniCRM usa modelos HTML com Jinja2 para gerar faturas. Você pode personalizar completamente o design da fatura, branding, cores e layout.

### Localização do Modelo de Fatura

Os modelos de fatura são armazenados em OmniCRM-API/invoice\_templates/

#### Modelos Padrão:

- norfone\_invoice\_template.html - Modelo de fatura de exemplo
- cifi\_invoice\_template.html - Exemplo de modelo alternativo

#### Configuração:

O modelo de fatura ativo é especificado em OmniCRM-API/crm\_config.yaml:

```
invoice:
  template_filename: 'norfone_invoice_template.html'
```

### Variáveis de Modelo Disponíveis

Os modelos de fatura têm acesso às seguintes variáveis Jinja2:

#### Informações da Fatura:

- {{ invoice\_number }} - ID único da fatura (por exemplo, INV-2025-001234)
- {{ date }} - Data de emissão da fatura (formato ISO: 2025-01-10T12:00:00)
- {{ due\_date }} - Data de vencimento do pagamento (por exemplo, 2025-02-10)
- {{ start\_date }} - Data de início do período de cobrança
- {{ end\_date }} - Data de término do período de cobrança

- {{ total\_amount }} - Valor total da fatura antes dos impostos (numérico)
- {{ total\_tax }} - Valor total do imposto calculado a partir de todas as transações (numérico)

### Informações do Cliente:

- {{ client.name }} - Nome completo ou nome da empresa do cliente
- {{ client.address.address\_line\_1 }} - Linha de endereço 1
- {{ client.address.address\_line\_2 }} - Linha de endereço 2
- {{ client.address.city }} - Cidade
- {{ client.address.state }} - Estado/província
- {{ client.address.zip\_code }} - Código postal/ZIP
- {{ client.address.country }} - País

### Itens de Linha de Transação:

Percorra as transações 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 Transação:

- sub\_transaction.transaction\_id - ID da transação
- sub\_transaction.created - Data/hora da transação
- sub\_transaction.title - Título da transação
- sub\_transaction.description - Descrição detalhada
- sub\_transaction.retail\_cost - Valor do item
- sub\_transaction.tax\_percentage - Percentual de imposto aplicado (por exemplo, 10 para 10%)
- sub\_transaction.tax\_amount - Valor do imposto calculado em dólares

### Exibindo Imposto em Modelos:

```
<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>
```

## Criando um Modelo de Fatura Personalizado

### Passo 1: Copiar Modelo Existente

```
cd OmniCRM-API/invoice_templates/
cp norfone_invoice_template.html your_company_invoice_template.html
```

### Passo 2: Personalizar HTML/CSS

Edite `your_company_invoice_template.html` para corresponder ao seu branding:

## Áreas de Personalização Chave:

### 1. Logotipo e Branding da Empresa

```
<!-- Substitua pela URL do seu logotipo -->


<!-- Atualize o nome da empresa -->
<h1>Nome da Sua Empresa</h1>
```

### 2. Esquema de Cores

```
<style>
  /* Cor principal da marca */
  .navbar {
    background: linear-gradient(to bottom right, #your-color-1, #your-
color-2);
  }

  /* Cabeçalhos da tabela */
  .table thead th {
    background-color: #your-brand-color !important;
    color: white !important;
  }

  /* Botões e links */
  .btn-primary {
    background-color: #your-brand-color;
  }
</style>
```

### 3. Rodapé de Informações da Empresa

```
<footer>
  <p>Nome da Sua Empresa</p>
  <p>123 Rua Comercial, Cidade, País</p>
  <p>Telefone: +1-555-123-4567 | Email: billing@yourcompany.com</p>
  <p>ABN/ID Fiscal: 12345678900</p>
</footer>
```

### 4. Instruções de Pagamento

```
<div class="payment-info">
  <h3>Métodos de Pagamento</h3>
  <p><strong>Online:</strong> Pague em https://yourcompany.com/pay</p>
  <p><strong>Transferência Bancária:</strong></p>
  <ul>
    <li>Nome da Conta: Sua Empresa Ltd</li>
    <li>BSB: 123-456</li>
    <li>Número da Conta: 987654321</li>
    <li>Referência: {{ invoice_number }}</li>
  </ul>
</div>
```

### 5. Termos e Condições

```
<div class="terms">
  <h4>Termos de Pagamento</h4>
  <p>Pagamento devido dentro de 30 dias a partir da data da fatura.</p>
  <p>Taxas por atraso: 2% ao mês sobre saldos em atraso.</p>
  <p>Para dúvidas de cobrança: billing@yourcompany.com</p>
</div>
```

### Passo 3: Atualizar Configuração

Edite OmniCRM-API/crm\_config.yaml:

```
invoice:
  template_filename: 'your_company_invoice_template.html'
```

### Passo 4: Reiniciar API

```
cd OmniCRM-API
sudo systemctl restart omnicrm-api
```

### Passo 5: Testar Geração de Faturas

1. Navegue até um cliente com transações
2. Gere uma fatura de teste
3. Baixe o PDF para verificar a formatação
4. Envie a fatura para você mesmo para testar a entrega do e-mail

## Personalização Avançada

### Conteúdo Condicional:

Use condicionais Jinja2 para mostrar/ocultar conteúdo:

```
{% if total_amount > 1000 %}
  <div class="high-value-notice">
    <p><strong>Nota:</strong> Saldo alto - Plano de pagamento disponível
    mediante solicitação.</p>
  </div>
{% endif %}

{% if client.address.country == "Australia" %}
  <p>GST Incluído: ${% "%.2f"|format(total_amount * 0.10) %}</p>
{% endif %}
```

### Suporte a Múltiplos Idiomas:

Crie modelos específicos para cada idioma:

```
invoice_template_en.html
invoice_template_es.html
invoice_template_fr.html
```

Configure com base na preferência de idioma do cliente.

### Cálculos Personalizados:

```
<!-- Exibir subtotal e detalhamento do imposto -->
<tr>
  <td colspan="4" class="text-right"><strong>Subtotal:</strong></td>
```

```

        <td>${{ "%.2f"|format(total_amount) }}</td>
</tr>
<tr>
    <td colspan="4" class="text-right"><strong>Imposto:</strong></td>
    <td>${{ "%.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:** A variável `total_tax` é calculada automaticamente somando o `tax_amount` de todas as transações na fatura. O imposto de cada transação é calculado com base em seu campo `tax_percentage`, que é definido como o `tax_percentage` do produto ou 0% se não especificado.

### Código QR para Pagamento:

Gere um código QR para pagamento móvel:

```

<div class="qr-payment">
    
    <p>Escaneie com seu telefone para pagar instantaneamente</p>
</div>

```

## Melhores Práticas de Estilização de PDF

OmniCRM usa **WeasyPrint** para converter HTML em PDF. Siga estas diretrizes:

### CSS Suportado:

- A maioria das propriedades CSS 2.1
- CSS3 limitado (flexbox, algumas transformações)
- Fontes da web via @font-face

### Não Suportado:

- JavaScript
- CSS Grid (use tabelas em vez disso)
- Animações complexas
- Algumas propriedades CSS modernas

### Tamanho da Página e Margens:

```

@page {
    size: A4;
    margin: 1cm;
}

body {
    font-family: Arial, sans-serif;
    font-size: 10pt;
}

```

### Estilização Específica para Impressão:

```

@media print {

```

```

.no-print {
    display: none;
}

.page-break {
    page-break-after: always;
}
}

```

### Layout da Tabela:

```

.table {
    table-layout: fixed;
    width: 100%;
}

.table th, .table td {
    word-wrap: break-word;
    padding: 4px;
}

```

### Incorporação de Fonte:

Para fontes personalizadas, use fontes seguras para a web ou incorpore:

```

@font-face {
    font-family: 'YourFont';
    src: url('https://yourcompany.com/fonts/yourfont.woff2') format('woff2');
}

body {
    font-family: 'YourFont', Arial, sans-serif;
}

```

## Testando Modelos de Faturas

### Lista de Verificação de Teste:

1. **Inspeção Visual:**
  - O logotipo é exibido corretamente
  - As cores correspondem às diretrizes da marca
  - O texto é legível (não muito pequeno)
  - As tabelas estão alinhadas corretamente
  - Todas as seções estão presentes
2. **Precisão dos Dados:**
  - Detalhes do cliente corretos
  - Os valores das transações somam corretamente
  - As datas estão formatadas corretamente
  - Todas as variáveis estão substituindo corretamente
3. **Qualidade do PDF:**
  - O tamanho do arquivo é razoável (<5MB)
  - As imagens estão nítidas e claras
  - Nenhum texto cortado ou transbordando
  - Quebras de página em locais apropriados
4. **Faturas de Múltiplas Páginas:**
  - Cabeçalhos se repetem em cada página
  - Números de página são exibidos
  - Listas longas de transações paginam corretamente

## 5. Entrega de E-mail:

- O PDF é anexado ao e-mail
- O tamanho do arquivo está abaixo do limite do Mailjet (15MB)
- Renderiza no Gmail, Outlook, Apple Mail

## Comando de Teste (Geração Manual):

Você pode testar a geração de faturas via API:

```
curl -X GET "http://localhost:5000/crm/invoice/{invoice_id}/pdf" \
-H "Authorization: Bearer YOUR_TOKEN" \
--output test_invoice.pdf
```

## Problemas Comuns de Modelos

### Variáveis não substituindo:

- **Causa:** Erro de digitação no nome da variável ou dados ausentes
- **Solução:** Verifique a ortografia exatamente (sensível a maiúsculas e minúsculas), verifique se os dados existem no banco de dados

### Estilização do PDF quebrada:

- **Causa:** Propriedade CSS não suportada
- **Solução:** Use propriedades CSS 2.1, teste com CSS compatível com WeasyPrint

### Imagens não aparecem:

- **Causa:** URLs relativas ou recursos externos bloqueados
- **Solução:** Use URLs HTTPS absolutas, certifique-se de que as imagens sejam acessíveis publicamente

### Tabelas transbordando a página:

- **Causa:** Larguras de coluna fixas muito largas
- **Solução:** Use larguras percentuais, `table-layout: fixed`

### Fontes não renderizando:

- **Causa:** Fonte não incorporada ou indisponível
- **Solução:** Use fontes seguras para a web (Arial, Times New Roman, etc.) ou incorpore corretamente fontes personalizadas

### Falha na geração de PDF:

- **Causa:** Erros de sintaxe HTML ou falha do WeasyPrint
- **Solução:** Valide o HTML, verifique os logs do WeasyPrint, simplifique layouts complexos

## Cache de PDF de Faturas

Para melhorar o desempenho e reduzir a geração redundante de PDFs, o OmniCRM inclui um sistema de cache de PDF de faturas. Quando um PDF de fatura é gerado pela primeira vez, ele é armazenado em cache no banco de dados para solicitações subsequentes.

### Como Funciona o Cache de PDF:

1. **Primeira Solicitação** - Quando um PDF de fatura é solicitado (download ou e-mail), o



sistema:

- Gera o PDF a partir do modelo de fatura
  - Codifica o PDF como Base64
  - Calcula um hash SHA256 do conteúdo do PDF
  - Armazena na tabela Invoice\_PDF\_Cache com:
    - Referência do ID da fatura
    - Dados do PDF (codificados em Base64)
    - Nome do arquivo
    - Hash do conteúdo (para verificação de integridade)
    - Timestamp de criação
2. **Solicitações Subsequentes** - Quando a mesma fatura é solicitada novamente:
- O sistema verifica se há PDF em cache pelo invoice\_id
  - Se o cache existir e for válido, retorna o PDF em cache imediatamente
  - Atualiza o timestamp last\_accessed para rastrear o uso do cache
3. **Invalidade do Cache** - PDFs em cache são invalidados quando:
- A fatura é modificada (transações adicionadas/removidas, detalhes alterados)
  - O modelo de fatura é atualizado
  - A limpeza manual do cache é acionada

### Benefícios:

- **Desempenho** - Entrega instantânea do PDF para solicitações repetidas (sem atraso na regeneração)
- **Consistência** - Mesmo PDF para todos os downloads de uma fatura (a menos que a fatura seja modificada)
- **Carga do Servidor** - Reduz o uso da CPU na geração de PDFs
- **Experiência do Usuário** - Um indicador de carregamento aparece durante a geração inicial, solicitações subsequentes são instantâneas

### Gerenciamento de Cache:

O Cache de PDF de Faturas é gerenciado automaticamente pelo sistema. Entradas de cache antigas ou não utilizadas podem ser purgadas periodicamente com base em:

- Idade (por exemplo, remover entradas de cache com mais de 90 dias)
- Padrões de acesso (remover entradas não acessadas em 30 dias)
- Limites de armazenamento (implementar limites de tamanho de cache, se necessário)

### Comportamento da API:

Ao baixar uma fatura via API ou UI:

- Primeira solicitação: Mostra indicador de carregamento enquanto o PDF é gerado, depois armazena em cache
- Solicitações subsequentes: Download imediato do cache
- O acerto/erro de cache é transparente para o usuário

**Importante:** Quando você atualizar seu modelo de fatura, limpe o cache para garantir que novas faturas usem o design atualizado:

```
-- Limpar todos os PDFs de fatura em cache (executar no MySQL)
DELETE FROM Invoice_PDF_Cache;
```

Ou atualize crm\_config.yaml para invalidar automaticamente o cache na alteração do modelo.

## Acessando Faturas

As faturas podem ser visualizadas no nível do sistema ou por cliente:

### Visualização por Cliente:

1. Navegue até **Clientes** → [Selecionar Cliente]
2. Clique na aba **Cobrança**
3. Veja a lista de faturas no terceiro cartão

### Visualização em Todo o Sistema:

1. Navegue até **Cobrança** → **Faturas** (no menu principal)
2. Veja todas as faturas de todos os clientes

## Widgets de Estatísticas de Faturas

Na parte superior da página de faturas, quatro cartões de estatísticas exibem resumos financeiros.

Estatísticas e Lista de Faturas

### Descrições dos Widgets:

- **Total de Faturas** - Soma de todos os custos de varejo das faturas (de todos os tempos) e contagem de faturas enviadas
- **Faturas Não Pagas** - Soma das faturas ainda não pagas e contagem de faturas não pagas
- **Faturas Este Mês** - Soma das faturas criadas neste mês civil com contagem
- **Faturas Mês Passado** - Soma das faturas criadas no mês civil passado com contagem

### Formatação de Valores:

- Valores acima de 1.000: Exibir como sufixo "k" (por exemplo, \$1.5k)
- Valores acima de 1.000.000: Exibir como sufixo "M" (por exemplo, \$2.3M)
- Valores acima de 1.000.000.000: Exibir como sufixo "B" (por exemplo, \$1.1B)

### Indicadores de Tendência:

- Widgets para "Este Mês" e "Mês Passado" mostram mudança percentual
- Seta verde para cima: Aumento em relação ao período anterior
- Seta vermelha para baixo: Diminuição em relação ao período anterior
- Seta cinza para a direita: Sem mudança

## Lista de Faturas

A tabela de faturas exibe todas as faturas com as seguintes colunas:

Lista Global de Faturas

### Descrições das Colunas:

- **ID** - ID único da fatura
- **Título** - Título/descrição da fatura
- **Período** - Período de cobrança (data de início - data de término) ou "N/A" para faturas únicas
- **Data de Vencimento** - Data de vencimento do pagamento

- **Criado** - Data de criação da fatura
- **Valor** - Valor total da fatura (custo de varejo)
- **Status** - Pago, Não Pago ou Devolvido
- **Ações** - Ações disponíveis (varia de acordo com o status)

### Ícones de Ação:

- **↓ (Baixar)** - Baixar PDF da fatura
- **✖ (Excluir)** - Anular fatura (somente se não paga)
- **💳 (Pagar)** - Pagar fatura online (somente se não paga)
- **✉ (E-mail)** - Enviar e-mail da fatura para o cliente
- **💰 (Reembolsar)** - Reembolsar pagamento Stripe (somente para faturas Stripe pagas)

## Gerando uma Fatura

Clique em "+ Gerar Fatura Proforma" para criar uma nova fatura.

Modal de Geração de Fatura com Pré-visualização de Transação

### Descrições dos Campos:

- **Buscar Clientes** - Selecionar cliente (apenas exibido na visualização em todo o sistema, preenchido automaticamente na visualização do cliente)
- **Título** - Título/nome da fatura (opcional, padrão para "Fatura para [Período]")
- **Data de Início** - Início do período de cobrança (padrão para 14 dias atrás)
- **Data de Término** - Término do período de cobrança (padrão para hoje)
- **Data de Vencimento** - Prazo de pagamento (padrão para hoje)
- **Pré-visualização de Transação** - Mostra todas as transações não faturadas no intervalo de datas com a capacidade de incluir/excluir transações específicas

### Seleção de Transação:

- **✓ (Sinal Verde)** - Clique para excluir uma transação da fatura
- **✖ (X Vermelho)** - Clique para incluir uma transação anteriormente excluída
- **Selecionar Todos** - Incluir todas as transações exibidas
- **Limpar Todos** - Excluir todas as transações
- Transações excluídas aparecem esmaecidas com texto riscado
- Totais em tempo real são atualizados à medida que você seleciona/deseleciona transações

### O Que Acontece:

1. O sistema encontra todas as transações não faturadas para o cliente dentro do intervalo de datas
2. Exibe a pré-visualização da transação com a capacidade de incluir/excluir transações individuais
3. Mostra o cálculo em tempo real do subtotal, imposto e total com base nas transações selecionadas
4. Apenas as transações selecionadas (incluídas) são adicionadas à fatura
5. Gera o PDF da fatura e o armazena em cache
6. Marca as transações selecionadas como faturadas (campo `invoice_id` populado)
7. Transações excluídas permanecem não faturadas e disponíveis para futuras faturas
8. A fatura aparece na lista com status "Não Pago"

### Exemplos de Casos de Uso:

**Cobrança Mensal:** Defina a data de início para o primeiro dia do mês, a data de término para o último dia do mês, a pré-visualização mostra todas as transações não faturadas desse período.

Selecione todas ou exclua manualmente as específicas.

**Fatura Específica de Serviço:** Use o mesmo intervalo de datas, depois exclua manualmente transações indesejadas (por exemplo, exclua transações não móveis para criar fatura apenas para móveis).

**Fatura Única:** Defina tanto a data de início quanto a data de término para o mesmo dia, a pré-visualização mostra apenas transações dessa data. Exclua quaisquer cobranças não relevantes para esta fatura específica.

## Visualizando Detalhes da Fatura

Clique em qualquer linha de fatura na tabela para visualizar os detalhes completos da fatura, incluindo todas as transações, totais e ações disponíveis.

Visualização dos Detalhes da Fatura

### Modal de Detalhes da Fatura:

- **Informações da Fatura** - Mostra ID da fatura, título, datas, status de pagamento e status de anulação
- **Lista de Transações** - Exibe todas as transações incluídas na fatura com:
  - Data da transação
  - Título e descrição
  - Custo de varejo
  - Valor do imposto e percentual (formatado como \$10.00 (10%))
  - Transações isentas de imposto mostram "-" na coluna de Imposto
- **Resumo dos Totais** - Cálculo em tempo real mostrando:
  - Contagem de transações
  - Subtotal (soma de todos os custos de varejo)
  - Imposto (soma de todos os valores de imposto)
  - Total da Fatura (subtotal + imposto)
- **Botões de Ação** - Mesmas ações disponíveis como na tabela:
  - **Baixar PDF** - Baixar PDF da fatura (sempre disponível)
  - **Enviar E-mail** - Enviar fatura para o cliente (faturas não anuladas)
  - **Pagar Fatura** - Processar pagamento (faturas não pagas, não anuladas apenas)
  - **Reembolsar** - Reembolsar pagamento Stripe (apenas faturas Stripe pagas)
  - **Excluir** - Anular fatura (faturas não pagas, não anuladas apenas)

## Baixando PDFs de Faturas

Clique no **ícone de download (↓)** na tabela ou no botão "**Baixar PDF**" na modal de detalhes da fatura para baixar uma fatura como PDF.

### Processo de Download:

1. Clique no ícone de download ao lado da fatura
2. Um indicador de carregamento aparece durante a geração (apenas na primeira vez)
3. O navegador solicita salvar o arquivo: Invoice\_01234.pdf
4. O PDF é aberto ou salvo na pasta Downloads

### Comportamento de Cache de PDF:

- **Primeiro Download** - PDF gerado a partir do modelo, armazenado em cache no banco de dados (pode levar 2-3 segundos)
- **Downloads Subsequentes** - Download instantâneo do cache
- **Invalidez do Cache** - Cache limpo se a fatura for modificada ou o modelo atualizado

## Resolvendo Problemas de Download:

- **O indicador de carregamento nunca para** - Verifique o console do navegador, a API pode estar inativa
- **PDF em branco ou corrompido** - Verifique o modelo da fatura para erros de sintaxe
- **Download falha** - Verifique as configurações do bloqueador de pop-ups, tente um navegador diferente

## Pagando Faturas

Clique no **ícone de pagamento** (💎) para pagar uma fatura online.

Modal de Pagamento de Fatura

### Processo de Pagamento:

1. Clique no ícone de pagamento na fatura não paga
2. Modal de pagamento se abre mostrando os detalhes da fatura
3. Selecione o método de pagamento:
  - **Transação Stripe** - Cobrar cartão de crédito salvo (disponível para todos os usuários)
  - **Dinheiro** - Pagamento manual em dinheiro (apenas para funcionários)
  - **Reembolso** - Aplicar reembolso como pagamento (apenas para funcionários)
  - **Transação POS** - Terminal de ponto de venda (apenas para funcionários)
  - **Transferência Bancária** - Transferência bancária manual (apenas para funcionários)
4. Se Stripe for selecionado:
  - Selecione o cartão dos métodos de pagamento salvos
  - O cartão padrão é pré-selecionado
  - Clique para selecionar um cartão diferente
5. Se outro método for selecionado:
  - Insira o número de referência (opcional)
6. Clique em "**Pagar Fatura**" para processar
7. O sistema processa o pagamento:
  - **Stripe** - Cobra o cartão via API Stripe
  - **Outros métodos** - Cria uma transação negativa para o valor do pagamento
8. O status da fatura muda para "Pago"
9. Notificação de sucesso exibida

### Autoatendimento vs Pagamento de Funcionários:

**:doc: Portal de Autoatendimento <self\_care\_portal> (Clientes):**

- Apenas pagamento Stripe disponível
- Deve ter método de pagamento salvo
- Aviso exibido se não houver métodos de pagamento
- Link para adicionar método de pagamento fornecido

### Portal de Funcionários (Administradores):

- Todos os métodos de pagamento disponíveis
- Pode marcar fatura como paga manualmente (dinheiro, POS, transferência bancária)
- Pode inserir números de referência para rastreamento

### Aviso de Método de Pagamento:

Se o cliente não tiver métodos de pagamento salvos, um aviso é exibido solicitando que ele

adicione um método de pagamento antes de poder pagar faturas.

Aviso de Método de Pagamento Ausente

## Enviando Faturas por E-mail

Clique no **ícone de e-mail** (✉) para enviar a fatura ao cliente.

### O Que Acontece:

1. Clique no ícone de e-mail ao lado da fatura
2. O sistema recupera o PDF da fatura do cache (ou gera se não estiver em cache)
3. Envia e-mail via Mailjet <integrations\_mailjet> usando `api_crmCommunicationCustomerInvoice` modelo
4. O e-mail inclui:
  - PDF da fatura como anexo
  - Nome do cliente
  - Número da fatura e data de vencimento
  - Valor total devido
  - Link para pagar a fatura online
  - Link para visualizar/baixar a fatura
5. Notificação de sucesso: "E-mail da fatura enviado com sucesso"

### Destinatários do E-mail:

O e-mail é enviado a todos os contatos do cliente com tipo "cobrança" ou contato principal se não existir contato de cobrança.

### Variáveis do Modelo de E-mail:

- `{{ var:customer_name }}` - Nome completo do cliente
- `{{ var:invoice_number }}` - ID da fatura
- `{{ var:invoice_date }}` - Data de emissão da fatura
- `{{ var:due_date }}` - Data de vencimento do pagamento
- `{{ var:total_amount }}` - Valor total devido
- `{{ var:invoice_url }}` - Link para visualizar/baixar PDF
- `{{ var:pay_url }}` - Link para pagar a fatura online

### Resolvendo Problemas de E-mail:

- **E-mail não enviado** - Verifique as credenciais da API Mailjet em `crm_config.yaml`
- **Cliente não recebendo** - Verifique os endereços de e-mail dos contatos do cliente
- **PDF não anexado** - Verifique se a geração do PDF foi bem-sucedida (tente baixar primeiro)

## Anulando Faturas

Clique no **ícone de exclusão** (✖) para anular uma fatura.

### Requisitos:

- A fatura deve estar **Não Paga**
- Faturas pagas não podem ser anuladas (devem ser reembolsadas em vez disso)

### Como Anular:

1. Localize a fatura não paga na lista

2. Clique no ícone de exclusão (✖)
3. Confirme no modal:

Modal de Confirmação de Anulação de Fatura

#### O Que Acontece:

- A fatura é marcada como `void = true`
- Todas as transações são desvinculadas da fatura (`invoice_id` definido como nulo)
- As transações se tornam "não faturadas" novamente
- As transações podem ser incluídas em uma nova fatura
- A fatura aparece na lista com o prefixo "Anulado:" no título
- Ações da fatura desativadas (sem download, pagamento ou e-mail)
- Pode ser visualizada filtrando por faturas "Anuladas"

#### Notas Importantes:

- Anular não é o mesmo que reembolsar
- **Anular** = "Esta fatura nunca deveria ter existido" (erro de cobrança, duplicado)
- **Reembolsar** = "Reverter uma fatura paga válida" (devolver dinheiro ao cliente)

## Reembolsando Faturas

Clique no **ícone de reembolso** (↶) para reembolsar uma fatura paga.

#### Requisitos:

- A fatura deve estar **Paga**
- A fatura deve ter sido paga via **Stripe**
- A fatura deve ter uma `payment_reference` válida (ID da intenção de pagamento Stripe)
- Disponível apenas para usuários funcionários (não para Autoatendimento)

#### Como Reembolsar:

1. Localize a fatura paga do Stripe
2. Clique no ícone de reembolso (↶)
3. O modal de confirmação de reembolso se abre:

Modal de Confirmação de Reembolso de Fatura

4. Clique em "**Confirmar Reembolso**"
5. O sistema processa o reembolso do Stripe:
  - Chama a API do Stripe para reembolsar o pagamento
  - Cria uma transação de reembolso no Stripe
  - Atualiza a fatura com `refund_reference`
6. O status da fatura muda para "Reembolsada"
7. Notificação de sucesso exibida

#### O Que Acontece Após o Reembolso:

- A fatura permanece no sistema (não anulada)
- O status mostra "Reembolsada"
- As transações permanecem vinculadas à fatura
- O cliente recebe o reembolso no método de pagamento original (3-7 dias úteis)
- O painel do Stripe mostra a transação de reembolso

#### Restrições de Reembolso:

- Não é possível reembolsar faturas pagas em dinheiro, POS ou transferência bancária (reversão manual necessária)
- Não é possível reembolsar parcialmente (apenas valor total da fatura)
- Não é possível reembolsar duas vezes

## Pesquisando e Filtrando Faturas

### Pesquisa

Use a barra de pesquisa para encontrar faturas. Pesquisas em:

- ID da fatura
- Título da fatura
- Nome do cliente (apenas visualização em todo o sistema)

### Filtros

Aplique filtros para restringir a lista de faturas:

#### Filtros Disponíveis:

- **Status de Anulação** - Todos, Anulado, Não Anulado
- **Status de Pagamento** - Todos, Pago, Não Pago

#### Ações de Filtro:

- **Aplicar Filtros** - Aplicar filtros selecionados à lista
- **Redefinir Filtros** - Limpar todos os filtros e mostrar todas as faturas

### Classificação

Clique em qualquer cabeçalho de coluna para classificar:

- **ID** - Classificar por ID da fatura (mais recente/mais antigo)
- **Título** - Classificar alfabeticamente
- **Data de Vencimento** - Classificar por data de vencimento
- **Criado** - Classificar por data de criação
- **Valor** - Classificar por custo de varejo (mais alto/mais baixo)
- **Status** - Classificar por status de pagamento (pago primeiro ou não pago primeiro)

Clique novamente para inverter a direção da classificação (crescente ↔ decrescente).

### Paginação

Navegue por grandes listas de faturas com controles de página mostrando a página atual, total de páginas e seletor de itens por página (10, 25, 50 ou 100 itens).

## Fluxos de Trabalho Comuns de Faturas

### Fluxo de Trabalho 1: Cobrança Mensal com Pré-visualização de Transação

1. O final do mês chega (por exemplo, 31 de janeiro)
2. Navegue até **Cobrança → Faturas**
3. Clique em "+ Gerar Fatura Proforma"
4. Selecione o cliente (ou faça por cliente se estiver cobrando muitos clientes)
5. Defina as datas:



- Data de Início: 2025-01-01
  - Data de Término: 2025-01-31
  - Data de Vencimento: 2025-02-15 (15 dias a partir de agora)
  - Título: "Serviços de Janeiro de 2025" (opcional)
6. A seção **Pré-visualização de Transação** aparece mostrando todas as transações não faturadas de janeiro
  7. Revise a pré-visualização:
    - Todas as transações estão incluídas por padrão
    - Verifique os totais: Subtotal, Imposto e Total da Fatura
    - Verifique se todas as cobranças estão corretas
  8. Clique em "**Gerar Fatura**" (o botão mostra a contagem de transações, por exemplo, "Gerar Fatura (15)")
  9. Fatura criada com todas as transações selecionadas
  10. Clique na linha da fatura para visualizar os detalhes e verificar
  11. Clique no botão "**Enviar E-mail**" na modal de detalhes ou no ícone de e-mail na tabela
  12. O cliente recebe o e-mail da fatura com PDF e link para pagamento

## Fluxo de Trabalho 2: Faturamento Seletivo de Transações

1. O cliente tem vários serviços (Móvel + Internet) e cobranças diversas
2. Deseja faturas separadas para cada serviço
3. **Gerar a primeira fatura (Serviços Móveis):**
  - Clique em "**+ Gerar Fatura Proforma**"
  - Título: "Serviços Móveis - Janeiro de 2025"
  - Início/Fim: 1-31 de janeiro
  - Data de Vencimento: 15 de fevereiro
  - Na pré-visualização de transações, **exclua** todas as transações não móveis:
    - Clique no botão **X** ao lado das transações de Internet
    - Clique no botão **X** ao lado das cobranças diversas
    - Apenas as transações de serviços móveis permanecem selecionadas
  - Verifique se os totais refletem apenas serviços móveis
  - Clique em "**Gerar Fatura**" (mostra a contagem de transações móveis)
4. **Gerar a segunda fatura (Serviços de Internet):**
  - Clique em "**+ Gerar Fatura Proforma**" novamente
  - Título: "Serviços de Internet - Janeiro de 2025"
  - Início/Fim: 1-31 de janeiro (mesmo período)
  - Na pré-visualização de transações:
    - Transações móveis já faturadas (não aparecem)
    - Exclua cobranças diversas usando o botão **X**
    - Apenas as transações de serviços de Internet permanecem
  - Clique em "**Gerar Fatura**"
5. **Gerar a terceira fatura (Cobranças Adicionais):**
  - Clique em "**+ Gerar Fatura Proforma**" novamente
  - Título: "Cobranças Adicionais - Janeiro de 2025"
  - Apenas cobranças diversas não faturadas aparecem na pré-visualização
  - Clique em "**Selecionar Todos**" para incluir todas
  - Clique em "**Gerar Fatura**"
6. Envie todas as três faturas para o cliente

## Fluxo de Trabalho 3: Excluindo Transações Disputadas ou Pendentes

1. O final do período de cobrança chega
2. Navegue até a aba **Cobrança** do cliente
3. Clique em "**+ Gerar Fatura Proforma**"
4. Defina as datas do período de cobrança
5. A pré-visualização de transações mostra 20 transações
6. O cliente disputou uma cobrança e outra está pendente investigação

7. Na pré-visualização de transações:
  - Localize a transação disputada (por exemplo, "Cobrança de excesso de dados")
  - Clique no botão **X** para excluí-la
  - Localize a transação pendente (por exemplo, "Taxa de instalação")
  - Clique no botão **X** para excluí-la
  - A contagem de transações é atualizada: "18 Transações selecionadas"
  - Os totais são recalculados automaticamente
8. Revise os totais atualizados (exclui os valores disputados)
9. Clique em **"Gerar Fatura (18)"**
10. Fatura gerada com apenas transações aprovadas
11. Transações disputadas/pendentes permanecem não faturadas para o próximo ciclo de cobrança

#### **Fluxo de Trabalho 4: Revisão Rápida da Fatura e Ajuste**

1. O funcionário gera a fatura mensal
2. A pré-visualização de transações mostra um total inesperadamente alto
3. Revise cada transação na pré-visualização:
  - Note a cobrança duplicada pelo mesmo serviço
  - Clique em **X** para excluir a duplicata
  - Note a transação de teste que não deveria ser cobrada
  - Clique em **X** para excluir a transação de teste
4. Os totais são atualizados em tempo real
5. Verifique se o novo total corresponde ao valor esperado
6. Clique em **"Gerar Fatura"** com as transações corrigidas
7. Volte e anule/exclua as transações excluídas, se necessário
8. Envie a fatura para o cliente com confiança

#### **Fluxo de Trabalho 5: Fatura de Instalação Única**

1. O técnico de campo completa a instalação
2. O funcionário adiciona a transação de instalação manualmente
3. Navegue até a aba **Cobrança** do cliente
4. Clique em **" + Gerar Fatura Proforma "**
5. Defina as datas:
  - Data de Início: hoje
  - Data de Término: hoje
  - Data de Vencimento: hoje + 7 dias
  - Título: "Serviços de Instalação"
6. A pré-visualização de transações mostra apenas as transações de hoje
7. Verifique se a cobrança de instalação aparece
8. Exclua quaisquer cobranças recorrentes usando o botão **X** (se presentes)
9. Clique em **"Gerar Fatura"**
10. Envie para o cliente imediatamente
11. O cliente paga online via Stripe

#### **Fluxo de Trabalho 6: Revisando a Fatura Antes do Contato com o Cliente**

1. O cliente liga com uma dúvida de cobrança
2. O funcionário navega até a lista de faturas do cliente
3. **Clique na linha da fatura** para abrir o modal de Detalhes da Fatura
4. Revise as informações da fatura:
  - ID da fatura, datas, status
  - Todas as transações incluídas com descrições
  - Detalhamento do imposto por transação
  - Subtotal, Imposto e Totais
5. Responda às perguntas do cliente com detalhes exatos

6. Se o cliente solicitar o PDF, clique no botão "**Baixar PDF**" no modal
7. Se o cliente solicitar o reenvio do e-mail, clique no botão "**Enviar E-mail**"
8. Feche o modal quando terminar

## Fluxo de Trabalho 7: Corrigindo Erro de Cobrança

1. O cliente relata uma cobrança incorreta
2. O funcionário clica na linha da fatura para visualizar os detalhes
3. Revê a lista de transações no modal de Detalhes da Fatura
4. Identifica a transação incorreta
5. A fatura está não paga, então pode ser anulada
6. Clique no botão "**Excluir**" no rodapé do modal
7. Confirme a anulação
8. As transações se tornam não faturadas novamente
9. O funcionário modifica ou remove a transação incorreta da lista de transações
10. Gere uma nova fatura com as transações corrigidas:
  - Use a pré-visualização de transações para excluir a transação corrigida, se necessário
  - Inclua apenas cobranças válidas
11. Envie a fatura corrigida para o cliente

## Fluxo de Trabalho 8: Processando Múltiplos Pagamentos

1. O cliente traz dinheiro para pagar várias faturas
2. Navegue até a aba **Cobrança** do cliente
3. Veja as faturas não pagas
4. Clique na primeira linha da fatura para visualizar os detalhes
5. Verifique o valor e as transações
6. Clique no botão "**Pagar Fatura**" no rodapé do modal
7. Selecione o método de pagamento "**Dinheiro**"
8. Insira a referência: "Dinheiro pago em 2025-01-15"
9. Clique em "**Pagar Fatura**"
10. O modal se fecha, a fatura é marcada como "Paga"
11. Repita para as faturas restantes
12. Todas as faturas agora estão marcadas como "Pagas"

## Fluxo de Trabalho 9: Lidando com Solicitação de Reembolso

1. O cliente solicita reembolso por pagamento em excesso
2. O funcionário verifica se a fatura foi paga via Stripe
3. Navegue até a fatura na lista
4. Clique na linha da fatura para visualizar os detalhes
5. Verifique as informações de pagamento e o valor
6. Clique no botão "**Reembolsar**" no rodapé do modal (apenas aparece para faturas Stripe)
7. Confirme o reembolso
8. O sistema processa o reembolso do Stripe
9. O status da fatura muda para "Reembolsada"
10. O cliente recebe o reembolso em 3-7 dias úteis
11. O funcionário acompanha o cliente para confirmar o recebimento

## Resolvendo Problemas

### Não é possível gerar fatura - Nenhuma transação encontrada

- **Causa:** Nenhuma transação não faturada no intervalo de datas especificado
- **Solução:** Verifique a lista de transações, verifique se as transações existem e não estão já faturadas. Ajuste o intervalo de datas ou remova o filtro.

### Falha na geração do PDF da fatura

- **Causa:** Erro de sintaxe no modelo, falha do WeasyPrint ou dados do cliente ausentes
- **Solução:** Verifique o HTML do modelo da fatura em busca de erros, verifique se os campos de endereço do cliente estão preenchidos, revise os logs da API.

### Pagamento falha com erro do Stripe

- **Causa:** Cartão recusado, fundos insuficientes, cartão expirado ou problema na API do Stripe
- **Solução:** Tente um método de pagamento diferente, verifique se o cartão é válido, verifique o painel do Stripe para a razão da recusa.

### Não é possível anular fatura

- **Causa:** Fatura já paga
- **Solução:** Faturas pagas não podem ser anuladas. Se o reembolso for necessário, use a função de reembolso para faturas Stripe ou crie uma transação de crédito manualmente.

### E-mail da fatura não enviado

- **Causa:** Credenciais da API Mailjet inválidas, cliente sem contato de cobrança ou modelo de e-mail ausente
- **Solução:** Verifique a configuração do Mailjet em `crm_config.yaml`, verifique os contatos do cliente, verifique se o modelo de e-mail da fatura existe.

### Botão de reembolso não aparecendo

- **Causa:** Fatura paga em dinheiro/POS/transferência bancária (não Stripe), ou fatura não paga
- **Solução:** O botão de reembolso aparece apenas para pagamentos Stripe. Para outros métodos de pagamento, crie uma transação de crédito manual.

### Download do PDF mostra design antigo do modelo

- **Causa:** PDF armazenado em cache antes da atualização do modelo
- **Solução:** Limpe o cache do PDF da fatura: `DELETE FROM Invoice_PDF_Cache WHERE invoice_id = X;`

### Cliente não pode pagar fatura (sem métodos de pagamento)

- **Causa:** Nenhum método de pagamento salvo no portal de Autoatendimento
- **Solução:** O cliente deve adicionar um cartão de crédito na página **Métodos de Pagamento** antes de pagar faturas.

### Múltiplas faturas geradas para o mesmo período

- **Causa:** Funcionário gerou a fatura duas vezes, ou os intervalos de datas se sobrepõem
- **Solução:** Anule a fatura duplicada. Ajuste os intervalos de datas para evitar sobreposição. Use a pré-visualização de transações para garantir conjuntos de transações exclusivos.

### Pré-visualização de transações mostra nenhuma transação

- **Causa:** Todas as transações no intervalo de datas já estão faturadas ou não existem transações
- **Solução:** Verifique se o intervalo de datas está correto. Verifique a lista de transações para confirmar se as transações existem. Filtre as faturas para ver qual fatura contém as transações.

## Não é possível excluir transação da geração da fatura

- **Causa:** Transação já faturada ou problema no navegador
- **Solução:** Verifique se a transação aparece na pré-visualização com um sinal verde. Atualize a página e tente novamente. Limpe o cache do navegador se o problema persistir.

## Total da fatura não corresponde ao valor esperado

- **Causa:** Transações inesperadas incluídas, imposto não calculado ou transações excluídas ainda contadas
- **Solução:** Revise a pré-visualização de transações cuidadosamente. Verifique o custo de varejo e o imposto de cada transação. Verifique se as transações excluídas estão esmaecidas. Verifique o distintivo de contagem de transações no botão Gerar Fatura.

## Botão Gerar Fatura está desativado

- **Causa:** Nenhuma transação selecionada ou intervalo de datas inválido
- **Solução:** Certifique-se de que pelo menos uma transação esteja incluída (não excluída). Verifique se a Data de Início é anterior à Data de Término. Verifique se a Data de Vencimento está definida.

## Modal de Detalhes da Fatura não abrindo

- **Causa:** Erro de JavaScript ou página não carregada completamente
- **Solução:** Atualize a página. Verifique o console do navegador em busca de erros. Tente um navegador diferente. Verifique a conexão com a internet.

## Imposto da transação não exibido nos Detalhes da Fatura

- **Causa:** Transação tem imposto de 0% ou tax\_amount é nulo
- **Solução:** Verifique se a transação tem tax\_percentage definido. Verifique se tax\_amount foi calculado quando a transação foi criada. Atualize a transação, se necessário.

## Botões de ação ausentes no modal de Detalhes da Fatura

- **Causa:** Fatura está anulada ou usuário não tem permissões
- **Solução:** Faturas anuladas mostram apenas o botão Baixar PDF. Verifique o status da fatura. Verifique o papel e as permissões do usuário.

## Documentação Relacionada

- integrations\_mailjet - Entrega de e-mail de fatura e modelos
- administration\_configuration - Configuração do modelo de fatura
- payments\_transaction - Criando transações que aparecem nas faturas
- payments\_process - Processando pagamentos de faturas
- basics\_payment - Métodos de pagamento e integração com Stripe



# Processar Pagamentos

A maioria dos pagamentos será processada automaticamente pelo sistema, mas há momentos em que você pode precisar processar um pagamento manualmente.

Para pagar uma fatura, selecione a fatura não paga e clique no botão "Pagar".

Isso abrirá um formulário de pagamento, onde você pode inserir o método de pagamento e clicar em "Enviar" para processar o pagamento.

O cliente receberá automaticamente um recibo pelo pagamento, e a fatura será marcada como paga.

Para transferências bancárias, você pode inserir a referência de pagamento e a data em que o pagamento foi feito (se diferente da data atual).

Pagar uma Fatura Pagar uma Fatura



# Transações do Cliente

Qualquer coisa que custe dinheiro no sistema é registrada como uma transação sob o cliente.

Cada transação tem um valor monetário para o custo de atacado e custo de varejo, e uma descrição do que a transação é.

As transações podem ser geradas automaticamente pelo sistema, por exemplo, quando um serviço é provisionado, uma transação é criada para o custo de configuração, e quando um serviço é faturado, uma transação é criada para o custo de varejo.

As transações também podem ser criadas manualmente, por exemplo, se um cliente receber um crédito, uma transação é criada para o valor do crédito, ou se uma taxa de instalação for cobrada, uma transação é criada para a taxa de instalação.

As transações são agrupadas para formar Faturas <payments\_invoices>, que são enviadas ao cliente para pagamento.

Gerenciamento de Transações

## Acessando Transações

As transações podem ser visualizadas no nível do sistema ou por cliente:

### Visualização por Cliente:

1. Navegue até **Clientes** → **[Selecionar Cliente]**
2. Clique na aba **Faturamento**
3. Veja a lista de transações no primeiro cartão

### Visualização em Todo o Sistema:

1. Navegue até **Faturamento** → **Transações** (do menu principal)
2. Veja todas as transações de todos os clientes

## Widgets de Estatísticas de Transações

Na parte superior da página de transações, quatro cartões de estatísticas exibem resumos financeiros:

Estatísticas e Lista de Transações

### Descrições dos Widgets:

- **Total de Transações** - Soma de todos os custos de varejo das transações (de todos os tempos)
- **Total de Transações Não Faturadas** - Soma das transações ainda não incluídas em uma fatura
- **Total de Transações Este Mês** - Soma das transações criadas neste mês calendário
- **Total de Transações Mês Passado** - Soma das transações criadas no mês calendário passado

### Formatação de Valores:

- Valores acima de 1.000: Exibir como sufixo "k" (ex: \$1.5k)
- Valores acima de 1.000.000: Exibir como sufixo "M" (ex: \$2.3M)
- Valores acima de 1.000.000.000: Exibir como sufixo "B" (ex: \$1.1B)

## Lista de Transações

A tabela de transações exibe todas as transações com as seguintes colunas:

### Descrições das Colunas:

- **ID** - ID único da transação
- **Data** - Data de criação da transação
- **Título** - Nome curto da transação
- **Descrição** - Descrição detalhada do que a transação é
- **Valor** - Custo de varejo (positivo para cobranças, negativo para créditos)
- **Fatura** - ID da fatura se a transação foi faturada (link clicável)
- **Status** - Marca de verificação se faturada, traço se ainda não faturada

### Ações Por Linha:

Cada linha tem um menu de ações ( : ) com opções:

- **Ver Detalhes** - Abre o modal de detalhes da transação
- **Baixar PDF da Fatura** - Baixar PDF (somente se faturada)
- **Anular Transação** - Marcar transação como anulada (somente se não faturada)

## Tipos de Transações

As transações se dividem em duas categorias principais:

### Transações de Débito (Cobranças)

Valores positivos que aumentam o saldo devedor do cliente:

- **Taxas de Configuração de Serviço** - Cobranças únicas quando o serviço é provisionado
- **Taxas Mensais de Serviço** - Cobranças recorrentes por serviços
- **Taxas de Instalação** - Cobranças por visitas de técnicos de campo
- **Cobranças de Equipamento** - Cobranças por modems, roteadores, cartões SIM
- **Taxas de Pagamento Atrasado** - Penalidades por faturas em atraso
- **Cobranças Manuais** - Cobranças personalizadas adicionadas pela equipe

### Transações de Crédito (Pagamentos/Reembolsos)

Valores negativos que diminuem o saldo devedor do cliente:

- **Pagamentos em Dinheiro** - Cliente pagou em dinheiro
- **Pagamentos com Cartão** - Cliente pagou com cartão de crédito/débito
- **Pagamentos por Transferência Bancária** - Cliente pagou via transferência bancária
- **Créditos de Conta** - Créditos de boa vontade, compensação
- **Reembolsos** - Dinheiro devolvido ao cliente
- **Descontos** - Descontos promocionais ou de fidelidade

## Adicionando uma Transação Manualmente

Clique em "+ Adicionar Transação" para abrir o modal de adicionar transação.



## Transação de Débito (Cobrança):

Adicionar Modal de Transação de Débito

## Transação de Crédito (Pagamento/Reembolso):

Adicionar Modal de Transação de Crédito

### Descrições dos Campos:

- **Tipo de Transação** - Selecione Débito (cobrança) ou Crédito (pagamento/reembolso)
- **Tipo de Crédito** - Se Crédito selecionado, escolha o método de pagamento (Dinheiro, Cartão, Transferência Bancária)
- **Título** - Nome curto para a transação (obrigatório)
- **Descrição** - Explicação detalhada (opcional)
- **Custo de Varejo** - Valor que o cliente paga (obrigatório, número positivo)
- **Custo de Atacado** - Seu custo (opcional, para rastreamento de margem)
- **Percentual de Imposto** - Taxa de imposto aplicada a esta transação (opcional, padrão para imposto do produto ou 0%)
- **Serviço** - Vincular transação a um serviço específico (opcional)
- **Site** - Vincular transação a um site específico (opcional)
- **Data da Transação** - Data da transação (padrão para hoje)

### Validação:

- Título e custo de varejo são obrigatórios
- O custo de varejo deve ser um número positivo
- Se o tipo de Crédito for selecionado, um tipo de crédito deve ser escolhido

### O que Acontece:

1. Transação criada no banco de dados
2. Aparece na lista de transações do cliente
3. Incluída na contagem de "Transações Não Faturadas"
4. Disponível para inclusão na próxima geração de fatura
5. Entrada de log de atividade criada

## Pesquisando e Filtrando Transações

### Pesquisa

Use a barra de pesquisa para encontrar transações. Pesquisa em:

- ID da Transação
- Título
- Descrição
- ID da Fatura

### Filtros

Aplique filtros para restringir a lista de transações:

#### Filtros Disponíveis:

- **Status de Anulação** - Todos, Anulados, Não Anulados
- **Status da Fatura** - Todos, Faturados, Não Faturados

### Ações de Filtro:

- **Aplicar Filtros** - Aplicar filtros selecionados à lista
- **Redefinir Filtros** - Limpar todos os filtros e mostrar todas as transações

### Classificação

Clique em qualquer cabeçalho de coluna para classificar:

- **ID** - Classificar por ID da transação (mais recente/mais antiga)
- **Data** - Classificar por data da transação
- **Título** - Classificar alfabeticamente
- **Valor** - Classificar por custo de varejo (mais alto/mais baixo)
- **Fatura** - Classificar por ID da fatura

Clique novamente para inverter a direção da classificação (crescente ↔ decrescente).

## Anulando Transações

Transações adicionadas por erro podem ser **anuladas** (marcadas como excluídas).

### Requisitos:

- A transação NÃO deve estar faturada
- Uma vez faturadas, as transações não podem ser anuladas (devem ser reembolsadas em vez disso)

### Como Anular:

1. Localize a transação na lista
2. Clique no menu de ações ( : )
3. Selecione "**Anular Transação**"
4. Confirme no modal

Modal de Confirmação de Anulação de Transação

### O que Acontece:

- Transação marcada como `void = true`
- Não aparece mais na lista padrão de transações
- Excluída da geração de fatura
- Pode ser visualizada filtrando por transações "Anuladas"
- Deduza do total de "Transações Não Faturadas"

**Nota:** Anular não é o mesmo que reembolsar. Anular significa "esta transação nunca deveria ter existido." Reembolsar significa "reverter uma transação válida."

## Imposto sobre Transações

As transações podem incluir imposto, que é calculado automaticamente com base na configuração de imposto do produto ou especificado manualmente por transação.

Exibição de Imposto da Transação

### Comportamento do Imposto:

- **Transações de Débito (Cobranças)** - O imposto é aplicado às cobranças com base em:

- **Percentual de Imposto do Produto** - Se a transação estiver vinculada a um produto, o percentual de imposto do produto é aplicado automaticamente
- **Substituição Manual** - A equipe pode substituir o percentual de imposto ao criar uma transação
- **Valor do Imposto** - Calculado como:  $\text{retail\_cost} \times (\text{tax\_percentage} / 100)$
- **Formato de Exibição** - Exibido como: \$10.00 (10%) nas listas de transações
- **Transações de Crédito (Pagamentos/Reembolsos)** - Nenhum imposto é aplicado a créditos
  - O campo percentual de imposto é oculto para transações de crédito
  - O imposto é automaticamente definido como 0% para todos os pagamentos e reembolsos
  - Créditos reduzem o saldo devedor do cliente sem implicações fiscais

### Exemplo de Cálculo de Imposto:

- Produto: Plano Móvel com 10% de imposto, \$50.00 custo de varejo
- Cálculo Automático de Imposto:  $\$50.00 \times 0.10 = \$5.00$
- Exibição: \$5.00 (10%)

### Imposto Zero (NIL/Isento):

- Produtos podem ser isentos de imposto definindo o percentual de imposto como 0
- O imposto padrão é 0% se não especificado
- Transações isentas de imposto mostram "-" na coluna de Imposto

Transação com Imposto Zero

## Visualização de Detalhes da Transação

Clique em uma transação para ver todos os detalhes:

## Transações Faturadas vs Não Faturadas

### Transações Não Faturadas:

- Ainda não incluídas em nenhuma fatura
- Disponíveis para a próxima geração de fatura
- Podem ser anuladas
- Contam para o total de "Transações Não Faturadas"
- O status mostra traço (-)

### Transações Faturadas:

- Incluídas em uma fatura
- Não podem ser anuladas (devem ser reembolsadas se necessário)
- ID da fatura clicável (link para detalhes da fatura)
- O status mostra marca de verificação (✓)
- Não podem ser modificadas

### Geração de Fatura:

Quando você gera uma fatura para um cliente:

1. O sistema encontra todas as transações não faturadas para aquele cliente
2. Opcionalmente, filtra por intervalo de datas
3. Transações incluídas na nova fatura
4. Campo `invoice_id` da transação populado

5. Transação agora marcada como "faturada"

Veja `payments_invoices` para detalhes sobre geração de faturas.

## Fluxos de Trabalho Comuns

### Fluxo de Trabalho 1: Crédito Manual por Interrupção de Serviço

1. Cliente liga: "O serviço ficou fora do ar por 2 dias"
2. A equipe decide creditar £10
3. Navegue até a aba **Faturamento** do cliente
4. Clique em **" + Adicionar Transação "**
5. Selecione o tipo de transação **Crédito**
6. Selecione o tipo de crédito **Pagamento em Dinheiro**
7. Insira o título: "Crédito por Interrupção de Serviço"
8. Insira a descrição: "Compensação por interrupção de 2 dias 8-9 Jan"
9. Insira o custo de varejo: 10.00
10. Selecione o serviço afetado no dropdown
11. Clique em **"Adicionar Transação"**
12. A transação aparece com o valor -£10.00
13. Será incluída na próxima fatura como crédito

### Fluxo de Trabalho 2: Taxa de Instalação Manual

1. Técnico de campo instala o serviço
2. A equipe precisa cobrar £75 de taxa de instalação
3. Navegue até a aba **Faturamento** do cliente
4. Clique em **" + Adicionar Transação "**
5. Selecione o tipo de transação **Débito**
6. Insira o título: "Taxa de Instalação"
7. Insira a descrição: "Visita do técnico de campo para instalação de fibra"
8. Insira o custo de varejo: 75.00
9. Insira o custo de atacado: 45.00 (opcional, para rastreamento de margem)
10. Selecione o serviço instalado
11. Selecione o site onde foi instalado
12. Clique em **"Adicionar Transação"**
13. A transação aparece na lista de não faturadas
14. Será incluída na próxima fatura

### Fluxo de Trabalho 3: Anulando Transação Duplicada

1. A equipe nota uma transação duplicada
2. Verifique se a transação NÃO foi faturada
3. Clique no menu de ações ( : ) na transação duplicada
4. Selecione **"Anular Transação"**
5. Confirme no modal
6. Transação removida da lista
7. Total não faturado diminui de acordo

### Fluxo de Trabalho 4: Encontrando Transações para Fatura

1. Precisa gerar fatura mensal
2. Clique no **Filtro de Fatura: "Não Faturadas"**
3. Clique em **Aplicar Filtros**
4. Veja todas as transações não faturadas
5. Anote o valor total dos widgets

6. Navegue para gerar fatura
7. Selecione intervalo de datas (ex: 1-31 Jan)
8. Transações no intervalo incluídas na fatura

## Solução de Problemas

### Não é possível anular a transação

- **Causa:** Transação já faturada
- **Solução:** A transação faz parte do histórico de faturas. Se um reembolso for necessário, crie uma transação de Crédito em vez disso.

### Transações duplicadas aparecendo

- **Causa:** Serviço cobrado várias vezes ou erro de provisionamento
- **Solução:** Anule a(s) transação(ões) duplicadas se não faturadas. Se faturadas, emita crédito.

### Transação não aparecendo na lista

- **Causa:** Filtros aplicados ou transação anulada
- **Solução:** Clique em "Redefinir Filtros" para mostrar todas as transações. Para ver transações anuladas, filtre por "Anular: Anulado".

### Total não faturado não corresponde ao esperado

- **Causa:** Algumas transações já foram faturadas ou transações anuladas foram excluídas
- **Solução:** Aplique o filtro "Fatura: Não Faturada" para ver apenas as não faturadas. Verifique transações anuladas separadamente.

### Não é possível adicionar transação (campo do cliente desativado)

- **Causa:** Visualizando a página de transações específicas do cliente
- **Solução:** O cliente está pré-selecionado. Se você precisar adicionar uma transação para um cliente diferente, vá para a página de Transações em todo o sistema.

## Documentação Relacionada

- payments\_invoices - Geração e gerenciamento de faturas
- payments\_process - Processamento de pagamentos contra faturas
- basics\_payment - Visão geral dos métodos de pagamento
- csa\_activity\_log - Visualizando o histórico de transações no log de atividade



# Controle de Acesso Baseado em Regras

## Funções, Permissões e Usuários no OmniCRM

O OmniCRM utiliza **controle de acesso baseado em funções (RBAC)**: pessoas (Usuários) são atribuídas a uma ou mais Funções, e cada Função é um conjunto de Permissões. As Permissões são a menor unidade de acesso (por exemplo, `view_customer`, `create_inventory`). O acesso efetivo de um usuário é a **união** das permissões de todas as funções atribuídas.

## Propósito

O RBAC permite:

1. **Proteção de Dados** — Os usuários veem e fazem apenas o que estão autorizados.
2. **Adequação Operacional** — As funções refletem as funções de trabalho (Admin, Suporte, Finanças, Administrador de Clientes).
3. **Administração Simples** — Conceda acesso atribuindo funções; evite a microgestão por usuário.
4. **Isolamento de Inquilinos** — As permissões “ver próprio ...” limitam a visibilidade aos dados do cliente/inquilino de um usuário.

Permissões RBAC

## Como Usuários, Funções e Permissões se Inter-relacionam

- **Usuários** — Pessoas reais que fazem login no OmniCRM.
- **Permissões** — Capacidades atômicas (por exemplo, `view_customer`, `delete_product`).
- **Funções** — Conjuntos nomeados de permissões (por exemplo, *Admin*, *Suporte*, *Finanças*).
- **Atribuição** — Os usuários recebem uma ou mais funções; as permissões se agregam.

Permissões RBAC

A autenticação prova *quem você é* (JWT, chave de API ou IP na lista branca). A autorização (funções/permissões) decide *o que você pode fazer*.

# Gerenciando Usuários

O sistema de gerenciamento de usuários do OmniCRM permite que administradores criem e gerenciem usuários da equipe (administradores, agentes de atendimento ao cliente), visualizem e modifiquem funções de usuários, redefinam senhas, gerenciem autenticação de dois fatores e controlem o acesso dos usuários.

## Tipos de Usuários

**Usuários Clientes** - Criados via autoinscrição ou por administradores. Atribuídos automaticamente à função "Cliente". Esses usuários acessam o portal de autoatendimento para gerenciar seus serviços, visualizar uso, pagar faturas, etc.

**Usuários da Equipe** - Criados por administradores com permissões apropriadas. Podem ser atribuídos a funções como Admin, Suporte, Finanças, etc. Esses usuários acessam a interface do CRM para gerenciar clientes, provisionar serviços, lidar com faturamento, etc.

**Usuários Administrativos** - Usuários com a permissão admin. Têm acesso total ao sistema, incluindo gerenciamento de usuários, gerenciamento de funções e todos os endpoints protegidos.

O usuário administrativo inicial é criado pela equipe do Omnitouch quando o sistema é implantado.

## Adicionando Novos Usuários (Administradores e CSAs)

Os administradores podem criar novos usuários da equipe através da interface da Web ou da API.

### Via Interface da Web

1. **Navegue até Usuários & Funções** - Acesse a interface de gerenciamento de usuários no menu de administração
2. **Clique em "Adicionar Usuário"** - Abre o formulário de criação de usuário

Adicionando um Usuário

3. **Preencha os Detalhes do Usuário:**
  - **Nome de Usuário** - Nome de usuário único para login (obrigatório)
  - **Email** - Endereço de email do usuário (obrigatório, deve ser único)
  - **Senha** - Senha temporária (obrigatória, o usuário deve mudar no primeiro login)
  - **Primeiro Nome** - Primeiro nome do usuário (obrigatório)
  - **Nome do Meio** - Nome do meio do usuário (opcional)

- **Último Nome** - Último nome do usuário (obrigatório)
  - **Número de Telefone** - Número de telefone de contato (opcional)
  - **Função** - Selecione uma ou mais funções para atribuir (obrigatório)
  - **Contato do Cliente** - Opcionalmente, vincule o usuário a um registro de contato do cliente (para usuários clientes)
4. **Clique em "Criar Usuário"** - O usuário é criado e pode fazer login imediatamente com as credenciais fornecidas
  5. **Usuário recebe notificação** - Opcionalmente, envie um email de boas-vindas com instruções de login

### Melhores Práticas:

- Use uma senha temporária como TempP@ssw0rd! e exija que o usuário a mude no primeiro login
- Atribua funções apropriadas com base na função de trabalho (veja os Designs de Função Típicos abaixo)
- Ative a 2FA para todos os funcionários administrativos e de suporte
- Vincule usuários clientes ao seu registro de contato do cliente para um escopo de dados adequado

### Via API

Crie um usuário programaticamente:

**Endpoint:** POST /auth/users

**Permissão Necessária:** admin

#### Corpo da Solicitação:

```
{
  "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"
}
```

#### Resposta:

```
{
  "id": 123,
  "username": "john.smith",
  "email": "john.smith@company.com",
  "first_name": "John",
  "last_name": "Smith",
}
```



```
"roles": ["Support"],  
"created": "2025-01-04T10:30:00Z"  
}
```

### Atribuindo Múltiplas Funções:

Os usuários podem ter várias funções. As permissões são aditivas (união de todas as permissões de função atribuídas).

Para atribuir várias funções, inclua-as na solicitação:

```
{  
  "username": "jane.doe",  
  "email": "jane.doe@company.com",  
  "password": "TempP@ssw0rd!",  
  "first_name": "Jane",  
  "last_name": "Doe",  
  "role": "Support, Finance"  
}
```

Ou use o endpoint de atribuição de função após a criação do usuário:

```
POST /auth/roles/{role_id}/users/{user_id}
```

## Visualizando e Pesquisando Usuários

### Listar Todos os Usuários (Admin):

GET /auth/users

Retorna uma lista paginada de todos os usuários com suas funções e informações básicas.

### Pesquisar Usuários:

GET /auth/users/  
search?search={query}&filters={"role":["Support"]}&page=1&per\_page=50

Filtrar por:

- Nome da função
- Domínio de email
- Status ativo/excluído
- Status de 2FA habilitado
- Data do último login

### Obter Usuário Específico:

GET /auth/users/{user\_id}

Retorna detalhes completos do usuário, incluindo:

- Informações pessoais
- Funções atribuídas e permissões efetivas
- Status de 2FA
- Último login e informações da sessão
- Contato do cliente vinculado (se aplicável)

## Criando e Gerenciando Funções

As funções são coleções de permissões que podem ser atribuídas a usuários. Em vez de atribuir permissões individualmente a cada usuário, você cria funções que agrupam permissões relacionadas e atribui essas funções aos usuários.

### Criando uma Nova Função

#### Via Interface da Web:

1. Navegue até **Usuários & Funções** → aba **Funções**
2. Clique em **"Criar Função"**
3. Insira os detalhes da função:
  - **Nome** - Nome curto e descritivo (por exemplo, "Tier2\_Support")
  - **Descrição** - Explique o propósito e as responsabilidades da função
4. Clique em **"Criar"**
5. A função é criada sem permissões; adicione permissões na próxima etapa

#### Via API:

**Endpoint:** POST /auth/roles

**Permissão Necessária:** admin

#### Solicitação:

```
{
  "name": "Tier2_Support",
  "description": "Equipe de suporte de nível 2 com acesso de
provisionamento elevado"
}
```

#### Resposta:

```
{
  "id": 45,
  "name": "Tier2_Support",
  "description": "Equipe de suporte de nível 2 com acesso de
provisionamento elevado",
  "permissions": [],
}
```

```
"users": []  
}
```

## Adicionando Permissões a uma Função

Uma vez que uma função é criada, atribua permissões para definir o que os usuários com essa função podem fazer.

### Via Interface da Web:

1. Navegue até **Usuários & Funções** → aba **Funções**
2. Clique no nome da função para visualizar os detalhes
3. Na seção **Permissões**, clique em **"Adicionar Permissão"**
4. Selecione uma ou mais permissões da lista
5. Clique em **"Adicionar"** - As permissões são imediatamente atribuídas

Gerenciando Permissões de Função Atribuindo Permissões à Função

### Via API:

**Endpoint:** POST /auth/roles/{role\_id}/permissions

### Solicitação:

```
{  
  "permission_id": 123  
}
```

Ou adicione várias permissões:

```
{  
  "permission_ids": [123, 124, 125]  
}
```

## Exemplo: Criando uma Função "Especialista em Provisionamento"

Essa função pode visualizar clientes, gerenciar serviços e provisionar:

1. Crie a função:

```
POST /auth/roles  
{  
  "name": "Provisioning_Specialist",  
  "description": "Pode provisionar serviços e gerenciar serviços  
de clientes"  
}
```

2. Adicione permissões:

```
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
  ]
}
```

## Removendo Permissões de uma Função

### Via Interface da Web:

1. Navegue até os detalhes da função
2. Na lista de **Permissões**, clique no botão **"X"** ou **"Remover"** ao lado da permissão
3. Confirme a remoção

### Via API:

**Endpoint:** DELETE /auth/roles/{role\_id}/permissions/{permission\_id}

### Exemplo:

```
DELETE /auth/roles/45/permissions/31
```

Isso remove a permissão create\_provision da função.

## Editando Detalhes da Função

Atualize o nome ou a descrição da função:

### Via Interface da Web:

1. Navegue até **Usuários & Funções** → aba **Funções**
2. Clique na função para editar
3. Modifique o nome ou a descrição da função
4. Clique em **"Salvar"**

Editando Detalhes da Função

### Via API:

**Endpoint:** PUT /auth/roles/{role\_id}

```
{
  "name": "Senior_Support",
  "description": "Equipe de suporte sênior com acesso total ao cliente"
}
```

## Excluindo uma Função

**Aviso:** Excluir uma função a remove de todos os usuários atribuídos. Certifique-se de que os usuários tenham funções alternativas ou perderão o acesso.

### Via API:

DELETE /auth/roles/{role\_id}

**Melhor Prática:** Em vez de excluir, considere arquivar ou renomear funções que não são mais necessárias.

## Atribuindo Funções a Usuários

### Durante a Criação do Usuário:

Inclua a função na solicitação de criação do usuário (veja "Adicionando Novos Usuários" acima).

### Para Usuários Existentes:

#### Via Interface da Web:

1. Navegue até **Usuários & Funções** → aba **Usuários**
2. Clique no usuário para editar
3. Na seção **Funções**, selecione/deselecione funções
4. Clique em "**Salvar**"

Editando Funções do Usuário

#### Via API:

Atualize as funções do usuário:

**Endpoint:** PUT /auth/users/{user\_id}

```
{
  "role": "Support,Finance"
}
```

Ou atribua uma única função a um usuário via endpoint de função:

**Endpoint:** POST /auth/roles/{role\_id}/users/{user\_id}

## Visualizando Atribuições de Funções

### Todos os usuários em uma função:

GET /auth/roles/{role\_id}/users

Retorna uma lista de todos os usuários atribuídos a essa função.

### Todas as funções de um usuário:

GET /auth/users/{user\_id}

A resposta inclui um array roles com todas as funções atribuídas.

## Gerenciando Senhas de Usuários

O OmniCRM fornece vários métodos para gerenciamento de senhas, dependendo do contexto.

### Redefinição de Senha pelo Usuário

Usuários que esqueceram sua senha podem redefini-la por conta própria através da página de login:

1. **Clique em "Esqueceu a Senha"** na página de login
2. **Insira o endereço de email** - O sistema envia um email de redefinição de senha
3. **Verifique o email** - O email contém um link de redefinição seguro com token (válido por 1 hora)
4. **Clique no link** - Abre o formulário de redefinição de senha
5. **Insira a nova senha** - Deve atender aos requisitos de complexidade da senha:
  - Mínimo de 8 caracteres
  - Pelo menos uma letra maiúscula
  - Pelo menos uma letra minúscula
  - Pelo menos um número
  - Pelo menos um caractere especial
6. **Enviar** - A senha é imediatamente atualizada; o usuário pode fazer login com a nova senha

### Fluxo da API:

1. **Solicitar redefinição:**

**Endpoint:** POST /auth/forgot\_password

```
{  
  "email": "user@example.com"
```

```
}
```

O sistema gera um token de redefinição e envia um email.

## 2. Redefinir com token:

**Endpoint:** POST /auth/reset\_password

```
{
  "token": "abc123...",
  "new_password": "NewSecureP@ssw0rd!"
}
```

## Redefinição de Senha pelo Administrador

Os administradores podem redefinir a senha de um usuário sem exigir verificação de email. Isso define uma senha temporária que o usuário deve mudar no próximo login.

### Via Interface da Web:

1. Navegue até **Usuários & Funções** → **Usuários**
2. Encontre o usuário e clique no botão **"Redefinir Senha"**
3. Insira uma senha temporária
4. Clique em **"Redefinir"**
5. Notifique o usuário sobre sua senha temporária (via canal seguro)
6. O usuário deve mudar a senha no próximo login

### Via API:

**Endpoint:** POST /auth/users/{user\_id}/admin\_reset\_password

**Permissão Necessária:** admin

### Solicitação:

```
{
  "new_password": "TempP@ssw0rd!",
  "force_change": true
}
```

### Parâmetros:

- new\_password - Senha temporária a ser definida
- force\_change (opcional) - Se verdadeiro, o usuário deve mudar a senha no próximo login

## Usuário Muda Sua Própria Senha

Usuários autenticados podem mudar sua própria senha a partir de seu perfil:

**Endpoint:** POST /auth/change\_password

**Solicitação:**

```
{
  "current_password": "OldP@ssw0rd!",
  "new_password": "NewSecureP@ssw0rd!"
}
```

O sistema valida a senha atual antes de permitir a mudança.

## Segurança da Senha

- As senhas são hashadas usando bcrypt (segurança do werkzeug)
- Nunca armazenadas em texto simples
- Tokens de redefinição expiram após 1 hora
- Tentativas de login falhadas podem acionar o bloqueio da conta (configurável)
- O rastreamento do histórico de senhas impede a reutilização (configurável)
- Requisitos de complexidade aplicados

## Gerenciando Autenticação de Dois Fatores (2FA)

O OmniCRM suporta autenticação de dois fatores baseada em TOTP para maior segurança. Os administradores podem habilitar, desabilitar e redefinir a 2FA para usuários.

Popup de aviso de 2FA do OmniCRM mostrado para usuários administrativos

## Habilitando 2FA para um Usuário

**Via Interface da Web:**

1. Navegue até **Usuários & Funções** → **Usuários**
2. Clique no usuário para visualizar os detalhes
3. Na seção **Segurança**, clique em **"Habilitar 2FA"**
4. O sistema gera:
  - Segredo TOTP (código QR exibido)
  - 10 códigos de backup (uso único)
5. O usuário escaneia o código QR com o aplicativo autenticador (Google Authenticator, Authy, etc.)
6. O usuário insere o código de verificação do aplicativo para confirmar a configuração
7. O usuário salva os códigos de backup em local seguro



8. A 2FA está agora habilitada; necessária para todos os futuros logins

Página de inscrição de 2FA do OmniCRM

#### Via API:

1. **Gerar segredo TOTP:**

**Endpoint:** POST /2fa/enable/user/{user\_id}

**Resposta:**

```
{
  "otp_secret": "JBSWY3DPEHPK3PXP",
  "qr_code_url": "otpauth://totp/OmniCRM:user@example.com?secret=JBSWY3DPEHPK3PXP&issuer=OmniCRM",
  "backup_codes": [
    "12345678",
    "23456789",
    "34567890",
    ...
  ]
}
```

2. **Verificar configuração:**

**Endpoint:** POST /2fa/verify-setup/user/{user\_id}

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

Retorna {"verified": true} em caso de sucesso.

### Fluxo de Login 2FA

Uma vez que a 2FA está habilitada, o processo de login muda:

1. O usuário insere nome de usuário e senha
2. O sistema valida as credenciais
3. Se válidas, solicita o código 2FA
4. O usuário insere o código do aplicativo autenticador OU código de backup
5. O sistema verifica o código
6. Em caso de sucesso, o usuário é logado

Login 2FA do OmniCRM

#### Códigos de Backup:

- 10 códigos gerados durante a configuração da 2FA
- Uso único apenas (consumidos após o uso)
- Usados se o aplicativo autenticador não estiver disponível
- Podem ser regenerados pelo usuário ou administrador

## Verificando o Código 2FA

**Endpoint:** POST /2fa/verify/user/{user\_id}

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

Aceita ambos:

- **Código TOTP** (6 dígitos do aplicativo autenticador)
- **Código de backup** (8 dígitos da lista de códigos de backup)

## Regenerando Códigos de Backup

Se um usuário esgotar os códigos de backup ou os perder, gere novos:

### Via Interface da Web:

1. Navegue até os detalhes do usuário
2. Clique em **"Regenerar Códigos de Backup"**
3. Exiba ou envie novos códigos para o usuário
4. Códigos antigos são invalidados

### Via API:

**Endpoint:** POST /2fa/backup-codes/regenerate/user/{user\_id}

### Resposta:

```
{  
  "backup_codes": [  
    "98765432",  
    "87654321",  
    "76543210",  
    ...  
  ]  
}
```

## Redefinição 2FA pelo Administrador

Se um usuário perder o acesso ao seu aplicativo autenticador e todos os códigos de backup, um administrador pode desabilitar e reabilitar a 2FA.

## Via Interface da Web:

1. Navegue até **Usuários & Funções → Usuários**
2. Clique no usuário
3. Clique no botão **"Redefinir 2FA"**
4. Confirme a redefinição
5. A 2FA é desabilitada; o usuário pode fazer login apenas com a senha
6. Oriente o usuário a configurar a 2FA novamente com um novo segredo

Redefinição de 2FA do OmniCRM

## Via API:

**Endpoint:** POST /2fa/admin/disable/user/{user\_id}

**Permissão Necessária:** admin

Isso desabilita completamente a 2FA para o usuário:

- Segredo TOTP limpo
- Códigos de backup limpos
- Flag `is_2fa_enabled` definida como falsa

O usuário pode então reabilitar a 2FA para obter um novo segredo e códigos de backup.

## Redefinição de 2FA pelo Usuário (Novo Dispositivo)

Se um usuário obtiver um novo dispositivo, mas ainda tiver acesso aos códigos de backup:

**Endpoint:** POST /2fa/reset-for-new-device/user/{user\_id}

```
{  
  "backup_code": "12345678"  
}
```

O sistema valida o código de backup, então gera um novo segredo TOTP e códigos de backup. O usuário pode configurar o aplicativo autenticador em seu novo dispositivo.

## Melhores Práticas de 2FA

- **Exigir 2FA para todos os funcionários administrativos e de suporte**
- **Armazenar códigos de backup com segurança** (gerenciador de senhas ou nota segura)
- **Regenerar códigos de backup** após usar vários
- **Usar aplicativos autenticadores respeitáveis** (Google Authenticator, Authy, Microsoft Authenticator)

- **Documentar procedimentos de redefinição de 2FA** para a equipe de suporte
- **Auditar o uso de 2FA** - monitorar quais usuários têm 2FA habilitada

## Atualizando Informações do Usuário

Os administradores podem atualizar os detalhes do usuário a qualquer momento.

### Via Interface da Web:

1. Navegue até **Usuários & Funções → Usuários**
2. Clique no usuário para editar
3. Modifique quaisquer campos editáveis:
  - Primeiro nome, nome do meio, último nome
  - Endereço de email (exige verificação)
  - Número de telefone
  - Funções
  - Vinculação de contato do cliente
4. Clique em **"Salvar"**

### Via 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"
}
```

### Mudanças de Email:

Quando o email é alterado, o novo email é marcado como pendente até ser verificado:

- O campo `pending_email` armazena o novo email
- Email de verificação enviado para o novo endereço
- O usuário clica no link para verificar
- O campo `email` é atualizado para o novo valor
- A flag `email_verified` é definida como verdadeira

## Excluindo Usuários

O OmniCRM utiliza **exclusões suaves** para usuários - eles são marcados como excluídos, mas não removidos do banco de dados. Isso preserva trilhas de auditoria e dados históricos.

## Excluindo um Usuário

### Via Interface da Web:

1. Navegue até **Usuários & Funções** → **Usuários**
2. Encontre o usuário a ser excluído
3. Clique no botão "**Excluir**"
4. Confirme a exclusão
5. O usuário é imediatamente desconectado e não pode fazer login novamente

### Via API:

**Endpoint:** DELETE /auth/users/{user\_id}

**Permissão Necessária:** admin

### O que Acontece:

- A flag deleted é definida como True
- O timestamp deleted\_at é registrado
- O usuário não pode fazer login
- Todas as sessões ativas são invalidadas
- O usuário ainda aparece nos logs de auditoria e registros históricos
- Dados vinculados (contatos de clientes, atividades) preservados

## Visualizando Usuários Excluídos

### Filtrar por usuários excluídos:

GET /auth/users/search?filters={"deleted":[true]}

## Restaurando um Usuário Excluído

Se um usuário foi excluído por engano, os administradores podem restaurá-lo:

**Endpoint:** PUT /auth/users/{user\_id}

```
{  
  "deleted": false  
}
```

Isso limpa a flag deleted e permite que o usuário faça login novamente.

**Nota:** A senha do usuário permanece inalterada, para que ele possa usar sua senha anterior.

## Excluindo Permanentemente um Usuário

**Aviso:** Isso é irreversível e remove todos os dados do usuário do banco de dados.

Não exposto via UI. Apenas disponível via acesso direto ao banco de dados por razões de conformidade (por exemplo, solicitações de exclusão de dados GDPR).

## Melhores Práticas para Exclusão de Usuários

- **Excluir suavemente por padrão** - Preserva trilhas de auditoria
- **Documentar razões para exclusão** - Adicionar nota no log de atividades antes de excluir
- **Transferir propriedade** - Reatribuir tickets abertos do usuário, tarefas antes de excluir
- **Revisar acesso** - Garantir que nenhum processo crítico dependa do usuário
- **Arquivar dados** - Exportar o histórico de trabalho do usuário, se necessário
- **Notificar equipes relevantes** - Informar gerentes/colegas sobre a exclusão

## Catálogo de Permissões

As permissões geralmente seguem padrões CRUD:

- view\_\* — ler/navegar
- create\_\* — criar/adicionar
- update\_\* — editar/modificar
- delete\_\* — excluir/remover

Algumas entidades também incluem variantes “**ver próprio ...**” que restringem a visibilidade aos dados do cliente/inquilino atual.

Permissões RBAC

## Global / Administrativa

- admin — Acesso administrativo total (gerenciar usuários, funções e permissões; acessar todos os endpoints protegidos).
- can\_impersonate — Atuar temporariamente como outro usuário (auditado; para suporte/resolução de problemas).

## Clientes e Registros Relacionados

- **Cliente**
  - view\_customer, create\_customer, update\_customer, delete\_customer
  - **Escopo próprio:** *ver próprio cliente*
- **Site do Cliente**
  - view\_customer\_site, create\_customer\_site, update\_customer\_site, delete\_customer\_site
  - **Escopo próprio:** *ver próprio site do cliente*
- **Contato do Cliente**
  - view\_customer\_contact, create\_customer\_contact, update\_customer\_contact, delete\_customer\_contact

- **Escopo próprio:** *ver próprio contato do cliente*
- **Atributo do Cliente** (veja Atributos do Cliente <administration\_attributes>)
  - view\_customer\_attribute, create\_customer\_attribute, update\_customer\_attribute, delete\_customer\_attribute
  - **Escopo próprio:** *ver próprio atributo do cliente*
- **Tag do Cliente** (veja Tags do Cliente <administration\_tags>)
  - view\_customer\_tag, create\_customer\_tag, update\_customer\_tag, delete\_customer\_tag
  - **Escopo próprio:** *ver própria tag do cliente*
- **Serviço do Cliente**
  - view\_customer\_service, create\_customer\_service, update\_customer\_service, delete\_customer\_service
  - **Escopo próprio:** *ver próprio serviço do cliente*
- **Atividade do Cliente**
  - view\_customer\_activity, create\_customer\_activity, update\_customer\_activity, delete\_customer\_activity
  - **Escopo próprio:** *ver própria atividade do cliente*

## Faturamento

- **Cartão Stripe**
  - view\_customer\_stripe\_card, create\_customer\_stripe\_card, update\_customer\_stripe\_card, delete\_customer\_stripe\_card
  - **Escopo próprio:** *ver próprio cartão stripe do cliente*
- **Transações**
  - view\_customer\_transaction, create\_customer\_transaction, update\_customer\_transaction, delete\_customer\_transaction
  - **Escopo próprio:** *ver própria transação do cliente*
- **Faturas**
  - view\_customer\_invoice, create\_customer\_invoice, update\_customer\_invoice, delete\_customer\_invoice
  - **Escopo próprio:** *ver própria fatura do cliente*

## Comunicações

- view\_communication, create\_communication, update\_communication, delete\_communication
- **Escopo próprio:** *ver própria comunicação*

## Inventário e Modelos

- **Inventário**
  - view\_inventory, create\_inventory, update\_inventory, delete\_inventory
  - **Escopo próprio:** *ver próprio inventário*
- **Modelo de Inventário**

- view\_inventory\_template, create\_inventory\_template, update\_inventory\_template, delete\_inventory\_template
- **Escopo próprio:** *ver próprio modelo de inventário*

## Produtos

- view\_product, create\_product, update\_product, delete\_product

## Transmissão de Célula (CBC)

- view\_cbc\_message, create\_cbc\_message, update\_cbc\_message, delete\_cbc\_message

## Provisionamento

- **Provisionar**
  - view\_provision, create\_provision, update\_provision, delete\_provision
  - **Escopo próprio:** *ver próprio provisionamento*
- **Evento de Provisionamento**
  - view\_provision\_event, create\_provision\_event, update\_provision\_event, delete\_provision\_event

## Acesso “Ver Próprio”

As permissões “ver próprio ...” limitam as leituras (e opcionalmente edições, onde implementadas) aos dados associados ao **próprio cliente/inquilino** do usuário. Por exemplo, uma função *Administrador de Clientes* pode gerenciar os contatos, sites, faturas e serviços de *seu* inquilino, mas não pode ver outros inquilinos.

## Designs de Função Típicos

Função	Permissões Típicas	Notas
Administrador do Sistema	admin, opcionalmente can_impersonate; além de CRUD amplo conforme necessário	Controle total sobre usuários/ funções/ permissões
Suporte	view_customer, view_customer_service, view_communication, view_provision; atualizações opcionais	Adicione can_impersonate se permitido
Finanças	view_customer_invoice, view_customer_transaction, view_product; opcional create_customer_invoice	Pesado em leitura; escrita limitada
Administrador de Clientes (inquilino)	"ver próprio ..." em clientes, sites, contatos, serviços, inventário, faturas, transações, comunicações, provisionamento	Gerenciamento com escopo de inquilino
Auditor	Ampla view_* apenas	Sem criar/atualizar/



Função	Permissões Típicas	Notas
Apenas de Leitura		excluir

Exemplos de Funções e Permissões Incluídas (resumo)

## Gerenciando Funções e Permissões via API

Todos os endpoints requerem permissão admin.

### Listar permissões

**Endpoint:** GET /auth/permissions

### Criar uma permissão (raro)

**Endpoint:** POST /auth/permissions

#### Corpo da Solicitação:

```
{
  "name": "view_example",
  "description": "Acesso somente leitura a objetos de exemplo"
}
```

### Listar funções

**Endpoint:** GET /auth/roles

### Criar uma função

**Endpoint:** POST /auth/roles

#### Corpo da Solicitação:

```
{
  "name": "Support",
  "description": "Equipe de suporte de nível 1"
}
```

### Adicionar uma permissão a uma função

**Endpoint:** POST /auth/roles/{role\_id}/permissions

#### Corpo da Solicitação:

```
{
```

```
"permission_id": 123
}
```

### Remover uma permissão de uma função

**Endpoint:** DELETE /auth/roles/{role\_id}/permissions/{permission\_id}

## Atribuindo Funções a Usuários

### Criar um usuário com função

**Endpoint:** POST /auth/users

#### Corpo da Solicitação:

```
{
  "username": "sara",
  "email": "sara@example.com",
  "password": "TempP@ssw0rd!",
  "first_name": "Sara",
  "last_name": "Ng",
  "phone_number": "+61...",
  "role": "Support"
}
```

### Atualizar a função de um usuário

**Endpoint:** PUT /auth/users/{user\_id}

#### Corpo da Solicitação:

```
{
  "role": "Finance"
}
```

### Listar usuários (somente Admin)

**Endpoint:** GET /auth/users

### Impersonação (Controlada)

- **Requerido:** can\_impersonate ou admin

### Iniciar impersonação

**Endpoint:** POST /auth/impersonate

## Corpo da Solicitação:

```
{ "user_id": 42 }
```

## Parar impersonação

**Endpoint:** POST /auth/stop\_impersonation

## Melhores Práticas

- **Privilégio mínimo primeiro.** Comece com funções mínimas; adicione permissões conforme necessário.
- **Preferir “ver próprio ...”.** Use permissões com escopo de inquilino para funções voltadas para o cliente.
- **Manter funções estáveis.** Atualize as permissões das funções quando as funcionalidades mudarem—não edite cada usuário.
- **Auditar regularmente.** Revise quem tem admin ou can\_impersonate.

## FAQ

**Um usuário pode ter várias funções?** Sim. As permissões são aditivas.

**Preciso de permissões personalizadas?** Geralmente não. O catálogo embutido cobre a maioria das necessidades.

**Como as regras “ver próprio ...” sabem o que é meu?** Elas avaliam a ligação entre seu usuário/contato e seu cliente (inquilino).



# Portal de Autoatendimento

O Portal de Autoatendimento é uma interface voltada para o cliente que permite aos usuários finais gerenciar suas próprias contas, visualizar o uso, pagar faturas e modificar serviços sem a necessidade de assistência da equipe de atendimento ao cliente.

## Métodos de Acesso:

- Login direto via credenciais do cliente
- Impersonação de Funcionário <customer\_care> para solução de problemas (de Clientes → Contatos → "Login como Usuário")

Veja também: Atendimento ao Cliente <customer\_care> para detalhes sobre impersonação, Fluxos de Autenticação <authentication\_flows> para o processo de login.

## Propósito

O Portal de Autoatendimento fornece aos clientes:

1. **Gerenciamento de Conta** - Visualizar e atualizar informações pessoais, contatos e endereços
2. **Visão Geral dos Serviços** - Ver todos os serviços ativos, uso e datas de expiração
3. **Rastreamento de Uso** - Monitorar dados, voz, SMS e saldos monetários
4. **Acesso à Faturamento** - Visualizar e pagar faturas, gerenciar métodos de pagamento
5. **Modificações de Serviço** - Adicionar recargas, comprar complementos, modificar serviços
6. **Disponibilidade 24/7** - Acessar informações da conta a qualquer momento sem ligar para o suporte

## Visão Geral do Portal

Quando os clientes fazem login no Portal de Autoatendimento, eles veem um painel personalizado com:

### Seções de Navegação:

- **Painel** - Visão rápida dos serviços e atividades recentes
- **Conta** - Detalhes pessoais, contatos, sites
- **Serviços** - Lista de todos os serviços com status e detalhes

- **Uso** - Informações sobre consumo de saldo e expiração
- **Faturamento** - Faturas, transações, métodos de pagamento
- **Recarga** - Comprar crédito de dados, voz, SMS ou crédito monetário

## Detalhes da Conta

A seção de Conta exibe informações do cliente e permite atualizações limitadas de autoatendimento.

Portal de Autoatendimento - Detalhes da Conta

### Campos Editáveis:

Os clientes podem atualizar:

- Endereço de e-mail (requer verificação)
- Número de telefone
- Senha
- Preferências de notificação

### Informações Somente para Visualização:

- ID do cliente
- Data de criação da conta
- Tipo de cliente (Individual/Empresa)
- Sites (endereço)
- Contatos vinculados

### Atualizando Informações da Conta:

1. Navegue até **Conta → Detalhes**
2. Clique em **"Editar"** ao lado do campo a ser atualizado
3. Insira novas informações
4. Clique em **"Salvar Alterações"**
5. Para alterações de e-mail, verifique via link enviado para o novo endereço

### Recursos de Segurança:

- Mudanças de senha requerem a senha atual
- Mudanças de e-mail requerem verificação
- Atividade registrada para trilha de auditoria
- Configurações de 2FA (se habilitadas)

## Visão Geral dos Serviços

A seção de Serviços mostra todos os serviços ativos e inativos para o cliente.

### Exibição do Cartão de Serviço:

Cada serviço exibe:

- **Nome do Serviço** - Identificador legível por humanos (por exemplo, "Móvel - +44 7700 900123")
- **Produto** - Nome do plano ou produto
- **Status** - Ativo, Suspenso, Expirado, Cancelado
- **Data de Criação** - Data de ativação do serviço
- **Data de Expiração** - Quando o serviço expira (se aplicável)
- **Custo Mensal** - Cobrança recorrente
- **Renovação Automática** - Indicador Habilitado/Desabilitado

### Ações do Serviço:

- **Ver Uso** - Ver consumo de saldo (Dados, Voz, SMS, Monetário)
- **Recarga** - Adicionar crédito ou dados
- **Complementos** - Comprar recursos adicionais
- **Modificar** - Alterar parâmetros do serviço (se permitido)
- **Ver Detalhes** - Ver configuração completa do serviço

### Indicadores de Status do Serviço:

- **Ativo** - Serviço está operacional
- **Expirando em Breve** - Renova ou expira em <7 dias
- **Suspenso** - Serviço temporariamente desativado (problema de pagamento, suspensão manual)
- **Expirado** - Serviço não está mais ativo

## Rastreamento de Uso

Os clientes podem monitorar seu uso em todos os tipos de saldo em tempo real.

### Uso de Dados

Visualize o consumo de dados com detalhamentos por bucket e expiração.

### Exibição de Uso de Dados:

- **Saldo Total** - Todos os buckets de dados combinados
- **Usado Neste Período** - Consumo desde a última renovação
- **Barra de Progresso** - Representação visual do consumo
- **Informações de Expiração** - Quando cada bucket expira
- **Quebra de Bucket** - Múltiplos buckets de dados com ordem de prioridade

## **Prioridade de Bucket:**

Quando existem múltiplos buckets de dados (por exemplo, plano base + recargas), eles consomem em ordem de peso:

- **Peso 10** - Consumido primeiro (tipicamente dados promocionais/bônus)
- **Peso 20** - Consumido segundo (tipicamente dados de recarga)
- **Peso 30** - Consumido por último (tipicamente dados do plano base)

## **Uso de Voz**

Acompanhe os minutos de chamada restantes em todos os buckets de voz.

Portal de Autoatendimento - Uso de Voz

### **Exibição de Uso de Voz:**

- **Minutos Restantes** - Saldo total de voz
- **Minutos Usados** - Consumo neste período
- **Histórico de Chamadas** - Chamadas recentes (se habilitado)
- **Datas de Expiração** - Quando os buckets de voz expiram
- **Minutos Internacionais** - Rastreamento separado (se aplicável)

### **Quebra de Uso:**

- Chamadas On-Net - Chamadas dentro da mesma rede
- Chamadas Off-Net - Chamadas para outras redes
- Chamadas Internacionais - Chamadas para fora do país
- Números Premium - Números de tarifa especial

## **Uso de SMS**

Monitore as permissões e o consumo de mensagens SMS.

### **Exibição de Uso de SMS:**

- **Mensagens Restantes** - Saldo de SMS
- **Usado Neste Mês** - Mensagens enviadas
- **MMS Incluído** - Se MMS conta para o saldo
- **SMS Internacionais** - Rastreamento separado (se aplicável)

## **Saldo Monetário**

Visualize o saldo de crédito pré-pago para serviços pay-as-you-go.

### **Exibição Monetária:**

- **Saldo Atual** - Crédito disponível

- **Última Recarga** - Valor e data da recarga mais recente
- **Data de Expiração** - Quando o saldo expira (se aplicável)
- **Recarga Automática** - Status Habilitado/Desabilitado

## Gerenciamento de Faturamento

Os clientes podem visualizar faturas, transações e gerenciar métodos de pagamento.

### Faturas

Visualize e pague faturas pendentes diretamente do portal.

Portal de Autoatendimento - Faturas

#### Exibição da Lista de Faturas:

- **Número da Fatura** - Identificador único
- **Data** - Data de criação da fatura
- **Data de Vencimento** - Prazo de pagamento
- **Valor** - Valor total da fatura
- **Status** - Pago, Não Pago, Atrasado
- **Ações** - Baixar PDF, Pagar Online

#### Pagando uma Fatura:

1. Navegue até **Faturamento** → **Faturas**
2. Encontre a fatura não paga na lista
3. Clique no botão "**Pagar Agora**"
4. Selecione o método de pagamento (cartão salvo ou novo cartão)
5. Confirme o pagamento
6. Receba um e-mail de confirmação

#### Baixando Faturas:

1. Clique no ícone "**Baixar**" ao lado da fatura
2. O PDF é baixado com todos os detalhes da fatura
3. Salvo para registros fiscais e documentação

### Transações

Visualize o histórico completo de transações, incluindo cobranças, créditos e pagamentos.

#### Exibição de Transação:

- **Data** - Data de criação da transação
- **Descrição** - Para o que é a cobrança/crédito



- **Valor** - Cobrança (positivo) ou crédito (negativo)
- **Fatura** - Qual fatura inclui esta transação
- **Status** - Faturado ou Não Faturado

## Métodos de Pagamento

Gerencie cartões de crédito salvos para faturamento automático e pagamentos online.

### Gerenciamento de Métodos de Pagamento:

- **Adicionar Cartão** - Adicionar cartão de crédito/débito novo de forma segura via Stripe
- **Definir Padrão** - Escolher método de pagamento principal
- **Remover Cartão** - Excluir cartões expirados ou não utilizados
- **Detalhes do Cartão** - Visualizar os últimos 4 dígitos, expiração, marca do cartão

Veja Métodos de Pagamento <payment\_methods> para documentação detalhada sobre gerenciamento de pagamentos.

## Recarga / Recarregamento

Compre crédito adicional de dados, voz, SMS ou crédito monetário instantaneamente.

### Processo de Recarga:

1. Navegue até **Serviços** → Selecione o serviço → "**Recarga**"
2. Escolha o produto de recarga no catálogo
3. Selecione o valor (opções predefinidas ou personalizadas)
4. Revise as informações de custo e expiração
5. Selecione o método de pagamento
6. Confirme a compra
7. Saldo atualizado imediatamente

### Recargas Disponíveis:

- **Recargas de Dados** - Opções de 1GB, 5GB, 10GB, 20GB, 50GB
- **Recargas de Voz** - Minutos de chamada adicionais
- **Pacotes de SMS** - Pacotes de mensagens
- **Crédito Monetário** - Saldo pré-pago (£5, £10, £20, £50, £100)

Veja Recarga & Recarregamento <features\_topup\_recharge> para fluxos de trabalho detalhados de recarga.

# Complementos de Serviço

Compre recursos e melhorias adicionais para serviços existentes.

## Complementos Disponíveis:

- **Roaming Internacional** - Habilitar serviço no exterior
- **Endereço IP Estático** - IP fixo para internet residencial
- **Conteúdo Premium** - Canais IPTV, serviços de streaming
- **Atualizações de Hardware** - Aluguel de modem, set-top boxes
- **Aumentos de Velocidade** - Aumentos temporários de largura de banda

## Comprando Complementos:

1. Navegue até **Serviços** → Selecione o serviço → **"Complementos"**
2. Navegue pelos complementos disponíveis para este tipo de serviço
3. Clique em **"Adicionar ao Serviço"** no complemento desejado
4. Revise o custo (uma vez + recorrente)
5. Confirme a compra
6. Complemento provisionado automaticamente

Veja Modificando Serviços - Complementos <csa\_modify> para detalhes sobre gerenciamento de complementos.

# Notificações & Alertas

Os clientes recebem notificações automatizadas para eventos importantes:

## Notificações por E-mail:

- Fatura gerada e pronta para pagamento
- Confirmação de pagamento recebido
- Avisos de expiração de serviço (7 dias, 3 dias, 1 dia)
- Alertas de saldo baixo (dados, voz, monetário)
- Ativação/desativação de serviço
- Solicitações de redefinição de senha
- Alertas de segurança (login de novo dispositivo)

## Alertas no Portal:

- Faturas não pagas
- Serviços expirando
- Avisos de dados baixos (10% restantes)
- Método de pagamento expirando
- Ações necessárias (verificar e-mail, atualizar método de pagamento)

## Preferências de Notificação:

Os clientes podem configurar:

- Frequência de notificações por e-mail
- Alertas SMS (se habilitados)
- Limiares de alerta (por exemplo, notificar quando <20% de dados restantes)
- Categorias de notificação (faturamento, uso, serviço)

## Limitações de Autoatendimento

Algumas operações requerem assistência da equipe:

### Requer Atendimento ao Cliente:

- Alterar tipo de cliente (Individual ↔ Empresa)
- Transferir serviços entre clientes
- Cancelar serviços (pode ser autoatendimento se habilitado)
- Disputar faturas
- Solicitar reembolsos
- Alterar contato principal
- Problemas complexos de provisionamento

### Restrições de Segurança:

- Não pode visualizar ou modificar contas de outros usuários
- Limitado a seus próprios dados de cliente (isolamento de inquilinos)
- Não pode acessar funções administrativas
- Não pode anular faturas ou transações
- Não pode modificar a configuração do serviço (apenas complementos/recargas)

## Acesso de Impersonação de Funcionários

A equipe de suporte pode acessar o Portal de Autoatendimento como um cliente para solução de problemas.

### Processo de Impersonação:

1. Navegue até **Clientes** → Selecione o cliente
2. Vá para a aba **Contatos**
3. Encontre o contato vinculado à conta do usuário
4. Clique no botão "**Login como Usuário**"
5. Uma nova aba se abre com a visão do Portal de Autoatendimento do cliente
6. Todas as ações são registradas e atribuídas ao usuário impersonado
7. A equipe vê exatamente o que o cliente vê

Login como Usuário - Impersonação

## **Casos de Uso:**

- **Solução de Problemas** - Reproduzir problemas relatados pelo cliente
- **Verificação** - Confirmar que as configurações de serviço aparecem corretamente
- **Treinamento** - Demonstrar recursos do portal
- **Suporte** - Guiar o cliente pelo portal enquanto visualiza sua tela

## **Segurança & Auditoria:**

- Requer permissão `can_impersonate` ou `admin`
- Todas as ações registradas na trilha de auditoria
- O cliente vê a impersonação no log de atividade
- Tempo limite da sessão após inatividade
- Não pode alterar a senha do cliente enquanto impersona

Veja Atendimento ao Cliente - Impersonação de Usuário <customer\_care> para documentação completa sobre impersonação.

## **Responsividade Móvel**

O Portal de Autoatendimento é totalmente responsivo e otimizado para dispositivos móveis.

### **Recursos Móveis:**

- Navegação otimizada para toque
- Layouts simplificados para telas pequenas
- Gestos de deslizar para navegação em carrossel
- Formulários e entradas amigáveis para dispositivos móveis
- Leitura de código QR para provisionamento de eSIM
- Chamada telefônica com um toque
- Integração GPS para preenchimento automático de endereços

### **Aplicativo Web Progressivo (PWA):**

- Instalar como aplicativo na tela inicial
- Visualização offline de dados recentes
- Notificações push (se habilitadas)
- Carregamento rápido com trabalhadores de serviço

## **Redefinição de Senha & Recuperação de Conta**

Os clientes podem redefinir senhas esquecidas sem ligar para o suporte.

### **Redefinição de Senha de Autoatendimento:**

1. Clique em "**Esqueceu a Senha**" na página de login
2. Insira o endereço de e-mail
3. Receba um e-mail de redefinição de senha (válido por 1 hora)
4. Clique no link no e-mail
5. Insira uma nova senha (deve atender aos requisitos de complexidade)
6. Envie e faça login com a nova senha

### **Requisitos de Senha:**

- Mínimo de 8 caracteres
- Pelo menos uma letra maiúscula
- Pelo menos uma letra minúscula
- Pelo menos um número
- Pelo menos um caractere especial (!@#\$%^&\*)

### **Bloqueio de Conta:**

Após 5 tentativas de login falhadas:

- Conta bloqueada por 30 minutos
- E-mail de redefinição de senha enviado automaticamente
- Notificação de segurança enviada para o e-mail registrado

## **Melhores Práticas para Clientes**

### **Recomendações de Segurança:**

1. Habilite 2FA para segurança aprimorada
2. Use uma senha única e forte
3. Mantenha o endereço de e-mail atualizado para notificações
4. Configure o método de pagamento padrão para renovações automáticas
5. Monitore o uso regularmente para evitar cobranças excessivas
6. Salve os códigos de backup em local seguro (se 2FA habilitado)
7. Saia após usar computadores compartilhados/públicos

### **Gerenciamento de Uso:**

1. Habilite alertas de saldo baixo (10-20% restantes)
2. Recargue antes que os saldos expirem para evitar interrupção do serviço
3. Revise faturas mensais para cobranças inesperadas
4. Atualize métodos de pagamento antes que os cartões expirem
5. Monitore o uso de dados ao longo do mês para evitar estrangulamento

### **Escalonamento de Suporte:**

Se o autoatendimento não resolver o problema:

1. Verifique a base de conhecimento / artigos de ajuda (se disponíveis)

2. Revise o log de atividade para alterações recentes
3. Entre em contato com o atendimento ao cliente por telefone, e-mail ou chat
4. Forneça o ID do cliente e detalhes do serviço para uma resolução mais rápida

## Acesso à API

Clientes com requisitos técnicos podem usar a API diretamente.

### Geração de Chave da API:

Disponível para clientes empresariais ou mediante solicitação:

1. Navegue até **Conta → Acesso à API**
2. Clique em **"Gerar Chave da API"**
3. Defina permissões (somente leitura ou leitura e gravação)
4. Defina a data de expiração
5. Salve a chave da API de forma segura (exibida apenas uma vez)

### Casos de Uso da API:

- Monitoramento automatizado de uso
- Integração com sistemas de faturamento internos
- Recargas programáticas
- Provisionamento de serviços via scripts
- Exportações de dados para análise

Veja Documentação da API <concepts\_api> para detalhes e exemplos de endpoints.

## Perguntas Frequentes

### Q: Por que não consigo ver todos os meus serviços?

R: Certifique-se de que está logado com a conta correta. Se você tiver várias contas de cliente, cada uma possui serviços separados. Entre em contato com o suporte para mesclar contas, se necessário.

### Q: Meu pagamento falhou, mas fui cobrado. O que devo fazer?

R: Verifique seu extrato bancário para cobranças pendentes. Se a cobrança aparecer, mas a fatura ainda mostrar não paga, entre em contato com o suporte com o número de referência da transação.

### Q: Como faço para cancelar um serviço?

R: Navegue até os detalhes do serviço e clique em "Cancelar Serviço" (se o

autoatendimento estiver habilitado). Caso contrário, entre em contato com o atendimento ao cliente para processar o cancelamento.

**Q: Posso transferir um serviço para outra pessoa?**

R: Não, transferências de serviço exigem assistência do atendimento ao cliente por razões de segurança e conformidade.

**Q: Por que meu saldo de dados está diminuindo mais rápido do que o esperado?**

R: Verifique atualizações de aplicativos em segundo plano, qualidade de streaming de vídeo e backups automáticos na nuvem. Revise a quebra de uso no portal para consumo detalhado.

**Q: Perdi meu dispositivo 2FA. Como posso recuperar o acesso?**

R: Use os códigos de backup para fazer login, depois desative e reative o 2FA. Se não tiver códigos de backup, entre em contato com o suporte para redefinição do 2FA (requer verificação de identidade).

**Q: Posso pagar uma fatura sem fazer login?**

R: O pagamento de fatura como convidado pode estar disponível via link direto da fatura. Caso contrário, o login é necessário por razões de segurança.

**Q: Como faço para baixar todas as minhas faturas de uma vez?**

R: Atualmente, é necessário baixar cada fatura individualmente. Para downloads em massa, entre em contato com o suporte ou use a API, se disponível.

## **Documentação Relacionada**

- Atendimento ao Cliente - Impersonação de Usuário <customer\_care> - Acesso de solução de problemas da equipe
- Métodos de Pagamento <payment\_methods> - Gerenciando cartões e pagamentos
- Recarga & Recarregamento <features\_topup\_recharge> - Comprando crédito adicional
- Uso de Serviço <csa\_service\_usage> - Compreendendo o rastreamento de saldo
- Visão Geral de Faturamento <billing\_overview> - Conceitos de faturamento e faturas
- Fluxos de Autenticação <authentication\_flows> - Login e segurança
- 2FA <2fa> - Configuração de autenticação de dois fatores
- Modificações de Serviço <csa\_modify> - Adicionando recursos e complementos