

Referencias

Este directorio contiene referencias de rendimiento para el sistema SMS-C utilizando Benchee.

Referencias Disponibles

1. Referencia de SMS en Crudo (`raw_sms_bench.exs`)

Mide el rendimiento del endpoint de API `submit_message_raw` utilizando PDUs de SMS reales.

Características:

- Utiliza PDUs de SMS reales (agrega tus PDUs a la lista `@sample_pdus` en el archivo)
- Desactiva la detección de duplicados limpiando las huellas digitales antes de cada iteración
- Genera informes tanto en consola como en HTML

Uso:

```
mix run benchmarks/raw_sms_bench.exs
```

Salida: `benchmarks/output/raw_sms_benchmark.html`

2. Referencia de API de Mensajes (`message_api_bench.exs`)

Mide varias operaciones de la API de mensajes, incluyendo inserción, recuperación y enrutamiento.

Características:

- Prueba insert_message (simple y con enrutamiento)
- Prueba get_messages_for_smsc
- Prueba list_message_queues
- Pre-puebla la base de datos con datos de prueba para escenarios realistas

Uso:

```
mix run benchmarks/message_api_bench.exs
```

Salida: benchmarks/output/message_api_benchmark.html

Configuración

Todas las referencias utilizan Benchee con la siguiente configuración predeterminada:

- Calentamiento: 2 segundos
- Tiempo: 10 segundos
- Tiempo de memoria: 2 segundos
- Estadísticas extendidas habilitadas
- Informes HTML generados automáticamente

Salidas

Los informes de referencia en HTML se generan en benchmarks/output/ e incluyen:

- Métricas de rendimiento detalladas
- Gráficos de comparación
- Estadísticas de uso de memoria
- Análisis estadístico

Documentación de Operaciones SMS-C

[← Volver al README Principal](#)

Bienvenido a la documentación de operaciones de SMS-C. Esta guía completa cubre todos los aspectos de la configuración, operación, monitoreo y solución de problemas del sistema SMS-C.

Visión General de la Documentación

Introducción

- [Referencia de Configuración](#) - Opciones de configuración completas y ejemplos

Operaciones Diarias

- [Guía de Operaciones](#) - Tareas diarias, monitoreo y mantenimiento
- [Guía de Enrutamiento SMS](#) - Gestión y configuración de rutas
- [Búsqueda de Suscriptores HSS](#) - Interfaz Diameter Sh para la detección de suscriptores en la red
- [Federación Geográfica](#) - Distribución multi-sitio basada en HTTP con descubrimiento DNS SRV
- [Referencia de API](#) - Documentación completa de la API con ejemplos

Rendimiento y Monitoreo

- [Ajuste de Rendimiento](#) - Optimización para diferentes cargas de trabajo
- [Guía de Métricas](#) - Métricas de Prometheus y monitoreo

Solución de Problemas

- [Guía de Solución de Problemas](#) - Problemas comunes y soluciones

Cumplimiento y Regulación

- [Cumplimiento de Intercepción ANSSI R226](#) - Especificaciones técnicas de intercepción legal en Francia
 - Integración de frontend multi-protocolo (IMS/SIP, SMPP, SS7/MAP)
 - Interfaces de intercepción legal ETSI X1/X2/X3
 - Arquitectura de almacenamiento de dos niveles Mnesia + SQL
 - Esquema CDR para consultas de intercepción legal
 - Capacidades de cifrado y criptoanálisis

Enlaces Rápidos

Tareas Comunes

- [Enviando un Mensaje](#)
- [Creando una Ruta](#)
- [Verificando el Estado del Mensaje](#)
- [Monitoreando la Salud del Sistema](#)
- [Manejando Fallos de Entrega](#)

Ejemplos de Configuración

- [Almacenamiento y Retención de Mensajes](#)
- [Configuración de Exportación CDR](#)
- [Controles de Privacidad](#)
- [Configuración de Alto Volumen](#)
- [Federación Geográfica](#)
- [Enrutamiento Geográfico](#)
- [Balanceo de Carga](#)
- [Configuración Diameter Sh / HSS](#)

- Configuración ENUM/NAPTR
- Cobro OCS
- Traducción de Números

Monitoreo y Alertas

- Métricas Clave
- Alertas Recomendadas
- Plantillas de Dashboard

Visión General de la Arquitectura del Sistema

El SMS-C es una plataforma de enrutamiento de mensajes distribuida y de alto rendimiento con los siguientes componentes clave:

Componentes Principales

- **Almacenamiento de Mensajes** - Almacenamiento rápido basado en Mnesia con retención configurable y exportación CDR
- **Motor de Enrutamiento** - Reglas de enrutamiento basadas en Mnesia con coincidencia de prefijos y balanceo de carga
- **Traducción de Números** - Normalización de números basada en regex con orden de prioridad
- **Integración de Cobro** - Cobro en línea OCS con políticas basadas en rutas
- **Búsqueda ENUM** - Enrutamiento de números basado en DNS con caché
- **Registro de Eventos** - Seguimiento del ciclo de vida del mensaje
- **Exportación CDR** - Exportación automática a base de datos SQL para facturación/análisis a largo plazo

Interfaces Externas

- **API REST** - Envío y gestión de mensajes (HTTPS)
- **Interfaz Web** - Gestión de rutas, navegador de mensajes, monitoreo

- **Prometheus** - Exposición de métricas para monitoreo
- **OCS** - Integración de cobro/facturación
- **DNS** - Búsquedas ENUM/NAPTR para enrutamiento

Distribución y Alta Disponibilidad

- **Federación Geográfica** - Distribución multi-controlador basada en HTTP con descubrimiento DNS SRV
- **Cola de Reenvío** - Encolado automático de mensajes y reintento cuando los controladores remotos no están disponibles
- **Balanceo de Carga** - Distribución de rutas ponderadas

Documentación Relacionada

- **Puntos de Referencia de Rendimiento** - Pruebas de rendimiento y resultados
- **Referencia del Esquema CDR** - Esquema completo de base de datos CDR con ejemplos SQL

Requisitos del Sistema

Requisitos Mínimos

- **CPU**: 2 núcleos
- **RAM**: 4 GB
- **Disco**: 50 GB (crece con la retención de mensajes)
- **SO**: Linux (recomendado), macOS (desarrollo)
- **Erlang/OTP**: 26.x o posterior
- **Elixir**: 1.15.x o posterior
- **Base de Datos SQL**: MySQL 8.0+, MariaDB 10.5+, o PostgreSQL 13+ (para almacenamiento CDR)

Recomendaciones para Producción

- **CPU:** 8+ núcleos
- **RAM:** 16+ GB
- **Disco:** 500+ GB SSD
- **Red:** 1 Gbps+
- **Base de Datos SQL:** Servidor dedicado con replicación (para almacenamiento CDR)

Puertos de Red

- **80/443** - Interfaz Web (HTTP/HTTPS)
- **8443** - API (HTTPS) y comunicación entre pares de federación
- **9568** - Métricas de Prometheus

Soporte y Recursos

Registros

- **Registros de Aplicación:** `/var/log/sms_c/` (producción) o consola (desarrollo)
- **Registros de Interfaz Web:** Visor de registros en tiempo real en `/logs`
- **Registros de Eventos:** Seguimiento de eventos por mensaje a través de la API

Diagnósticos

- **Verificación de Salud:** `GET /api/status`
- **Métricas:** `GET http://localhost:9568/metrics` (formato Prometheus)
- **Estado del Frontend:** Interfaz Web en `/frontend_status`
- **Cola de Mensajes:** Interfaz Web en `/message_queue`

Cómo Obtener Ayuda

1. Consulta la [Guía de Solución de Problemas](#)
2. Revisa los registros de la aplicación
3. Verifica las métricas de Prometheus en busca de anomalías
4. Usa el simulador de enrutamiento para probar la lógica de enrutamiento
5. Examina los registros de eventos por mensaje

Información de la Versión n

Esta documentación está actualizada a:

- **Última Actualización:** 2025-10-30
- **Versión SMS-C:** Última versión de desarrollo
- **Elixir Soportado:** 1.15.x - 1.17.x
- **Erlang/OTP Soportado:** 26.x - 27.x

Convenciones de Documentación

A lo largo de esta documentación:

- **Ejemplos de configuración** muestran valores típicos; ajusta para tu entorno
- **Ejemplos de API** utilizan el formato de línea de comandos `curl`
- **Direcciones IP y dominios** son solo ejemplos; reemplaza con tus valores reales
- **Nombres de métricas** siguen las convenciones de nomenclatura de Prometheus
- **Todas las marcas de tiempo** están en UTC a menos que se especifique lo contrario

Inicio Rápido

1. **Configuración:** Configura a través de `config/runtime.exs` - consulta [Referencia de Configuración](#)
2. **Rutas Iniciales:** Crea reglas de enrutamiento a través de la Interfaz Web o archivo de configuración - consulta [Guía de Enrutamiento SMS](#)
3. **Enviar Mensaje de Prueba:** Usa la API o la Interfaz Web - consulta [Referencia de API](#)
4. **Monitorear:** Configura la recopilación de Prometheus - consulta [Guía de Métricas](#)

Comentarios sobre la Documentación

Esta documentación se mantiene junto con la base de código de SMS-C. Para correcciones o mejoras, actualiza los archivos markdown en el directorio

`docs/`.

Documentación de Cumplimiento de Intercepción ANSSI R226

Propósito del Documento: Este documento proporciona especificaciones técnicas requeridas para la autorización ANSSI R226 bajo los Artículos R226-3 y R226-7 del Código Penal Francés para el Centro de Servicio de SMS OmniMessage (SMSc).

Clasificación: Documentación de Cumplimiento Regulatorio

Autoridad Objetivo: Agence nationale de la sécurité des systèmes d'information (ANSSI)

Regulación: R226 - Protección de la Privacidad de Correspondencia y la Intercepción Legal

1. ESPECIFICACIONES TÉCNICAS DETALLADAS

1.1 Ficha Técnica Comercial

Nombre del Producto: OmniMessage SMSc (Centro de Servicio de SMS)

Tipo de Producto: Centro de Mensajes de Telecomunicaciones

Función Principal: Enrutamiento, almacenamiento y entrega de mensajes SMS

Protocolos de Red: REST API (HTTPS), protocolos SMS (SMPP, IMS, SS7/MAP a través de frontends externos)

Modelo de Despliegue: Aplicación de servidor local

Tecnología: Elixir/Erlang, Phoenix Framework, Mnesia, MySQL/PostgreSQL

Capacidades Centrales

Procesamiento de Mensajes:

- Cola de mensajes SMS centralizada con REST API
- Diseño independiente de protocolo que soporta frontends SMPP, IMS, SS7/MAP
- Motor de enrutamiento dinámico con enrutamiento basado en prefijos
- Lógica de reintento con retroceso exponencial
- Manejo de expiración de mensajes y cola de cartas muertas
- Generación y archivo de Registros de Detalles de Llamadas (CDR)
- Rendimiento: ~1,750 mensajes/segundo de tasa de inserción, capacidad de 150 millones de mensajes/día

Almacenamiento de Mensajes:

- **Cola de Mensajes Activa:** Base de datos en memoria Mnesia con persistencia en disco opcional
 - Almacenamiento primario: RAM para acceso ultra-rápido (latencia sub-milisegundo)
 - Respaldo en disco: el modo `disc_copies` escribe en disco para recuperación ante fallos
 - Recuperación automática: Los mensajes sobreviven a los reinicios del sistema
 - Retención: Configurable (24 horas por defecto), luego limpieza automática
- **Archivo CDR a Largo Plazo:** Base de datos MySQL/PostgreSQL (separada de la cola de mensajes)
 - CDRs escritos cuando los mensajes son entregados, expiran, fallan o son rechazados
 - Base de datos SQL utilizada SOLAMENTE para exportación/archivo de CDR, NO para operaciones de mensajes activos
 - Sin impacto en el rendimiento del enrutamiento de mensajes (escritura asíncrona)

- **Beneficios de Arquitectura de Dos Niveles:**
 - Cola activa: Rápida como un rayo (1,750 msg/sec) sin cuellos de botella SQL
 - Archivo CDR: Retención a largo plazo (meses/años) para facturación e interceptación legal
 - Separación limpia: Las operaciones de mensajes nunca tocan SQL
- Soporte de clúster para alta disponibilidad (replicación de Mnesia entre nodos)

Interfaces de Red:

- **REST API:** HTTPS (puerto 8443) para comunicación de frontend externo
- **Panel de Control:** HTTPS (puerto 8086) para gestión basada en web
- **Protocolos de Frontend:** SMPP, IMS, SS7/MAP (a través de aplicaciones de puerta de enlace externas)
- **Base de Datos:** MySQL/PostgreSQL para almacenamiento de CDR

Enrutamiento y Procesamiento:

- Enrutamiento SMS dinámico con actualizaciones de configuración en tiempo de ejecución
- Coincidencia basada en prefijos (números de llamada/llamados)
- Filtrado de SMSC de origen y tipo
- Balanceo de carga basado en prioridad y peso
- Traducción y normalización de números
- Soporte de búsqueda DNS ENUM (E.164 Number Mapping)
- Capacidades de respuesta automática y caída de mensajes
- Control de carga por ruta (integración CGRates)

□ **Arquitectura y características completas documentadas en [README.md](#)**

1.2 Capacidades de Interceptación

1.2.1 Adquisición de Mensajes

Captura de Mensajes SMS:

- El OmniMessage SMS Sc procesa todos los mensajes SMS entre suscriptores y redes externas
- Acceso completo a los metadatos y contenido del mensaje, incluyendo:
 - MSISDN de origen (número móvil)
 - MSISDN de destino (número móvil)
 - IMSI de origen (Identidad Internacional de Suscriptor Móvil)
 - IMSI de destino
 - Cuerpo del mensaje (contenido de texto)
 - Datos PDU (Unidad de Datos de Protocolo) en bruto
 - Información TP-DCS (Esquema de Codificación de Datos)
 - Codificación del mensaje (GSM7, UCS-2, 8-bit, Latin-1)
 - Indicadores de mensaje multipartes y datos de reensamblaje
 - Información del Encabezado de Datos del Usuario (UDH)

Adquisición de Metadatos de Mensajes:

- Registros de Detalles de Llamadas (CDR) completos almacenados en la base de datos con:
 - ID de mensaje (identificador único)
 - Número de llamada (MSISDN de origen)
 - Número llamado (MSISDN de destino)
 - Marca de tiempo de envío (cuando el mensaje ingresó al sistema)
 - Marca de tiempo de entrega (cuando el mensaje fue entregado)
 - Marca de tiempo de expiración (cuando el mensaje expiró si no se pudo entregar)
 - Estado (entregado, expirado, fallido, rechazado)
 - Conteo de intentos de entrega
 - Partes del mensaje (para SMS concatenados/multi-partes)
 - Identificador de SMSC de origen
 - Identificador de SMSC de destino
 - Nodo de origen (nombre del nodo del clúster Erlang)
 - Nodo de destino (para despliegues distribuidos)
 - Indicador de carta muerta (indicador de agotamiento de reintentos)

□ **Esquema completo de CDR documentado en [CDR_SCHEMA.md](#)**

Acceso a la Cola de Mensajes:

- Monitoreo en tiempo real de la cola de mensajes
- Puntos finales de REST API para recuperación de mensajes
- Consultas de base de datos para búsqueda histórica de mensajes
- Capacidades de filtrado por:
 - Número de teléfono (origen/destino)
 - Puerta de enlace SMSC
 - Rango de tiempo
 - Estado del mensaje
 - Intentos de entrega

□ **Documentación completa de la API en [API_REFERENCE.md](#)**

1.2.2 Capacidades de Procesamiento de Datos

Arquitectura de Almacenamiento de Mensajes (Sistema de Dos Niveles):

El SMSc utiliza una sofisticada arquitectura de almacenamiento de dos niveles que separa el procesamiento operativo de mensajes de la archivación a largo plazo:

Nivel 1: Cola de Mensajes Activa (Mnesia)

- **Propósito:** Operaciones de enrutamiento y entrega de mensajes en tiempo real
- **Tecnología:** Base de datos distribuida Erlang Mnesia
- **Modo de Almacenamiento:** En memoria con respaldo `disc_copies`
 - Almacenamiento primario en RAM para máxima velocidad
 - Sincronización automática en disco para recuperación ante fallos
 - Los mensajes persisten a través de reinicios del sistema
- **Rendimiento:** Operaciones de lectura/escritura sub-milisegundo
- **Retención:** A corto plazo (24 horas por defecto), configurable
- **Limpieza:** Archivado automático a la base de datos CDR, luego eliminación de Mnesia

- **Operaciones:** Todas las operaciones de cola de mensajes (inserción, actualización, estado de entrega, enrutamiento)
- **Característica Crítica:** La base de datos SQL NUNCA se consulta durante el enrutamiento/entrega de mensajes

Nivel 2: Archivo CDR (MySQL/PostgreSQL)

- **Propósito:** Almacenamiento a largo plazo para facturación, análisis e interceptación legal
- **Tecnología:** Base de datos SQL tradicional (MySQL o PostgreSQL)
- **Disparador de Escritura:** CDRs escritos SOLAMENTE cuando los mensajes alcanzan un estado final:
 - Mensaje entregado con éxito
 - Mensaje expirado (excedió el período de validez)
 - Mensaje fallido permanentemente
 - Mensaje rechazado por reglas de enrutamiento
- **Modo de Escritura:** Escritura por lotes asíncrona (sin impacto en el rendimiento del enrutamiento de mensajes)
- **Retención:** A largo plazo (meses a años), configurable según requisitos regulatorios
- **Operaciones:** Consultas históricas, informes, cumplimiento, interceptación legal
- **Acceso:** Consultas SQL, REST API (futuro), exportación CSV/JSON

Beneficios Clave de la Arquitectura:

1. **Rendimiento:** Las operaciones de enrutamiento activas nunca tocan SQL (sin cuellos de botella de base de datos)
2. **Escalabilidad:** Mnesia maneja más de 1,750 mensajes/segundo sin sobrecarga de SQL
3. **Confiabilidad:** El modo `disc_copies` asegura que no haya pérdida de mensajes en caso de fallos
4. **Cumplimiento:** La base de datos CDR proporciona una auditoría permanente
5. **Separación de Preocupaciones:** Datos operativos vs. datos archivados claramente separados

Ciclo de Vida del Mensaje:

1. Mensaje enviado → Almacenado en Mnesia (RAM + respaldo en disco)
2. Mensaje enrutado → Consulta Mnesia (ultra-rápido)
3. Mensaje entregado/expirado → CDR escrito en SQL (asíncrono)
4. Después de 24h → Mensaje eliminado de Mnesia (trabajador de limpieza)
5. CDR permanece en SQL → Disponible para consultas de interceptación legal (años)

Retención y Recuperación de Datos:

- Retención o eliminación configurable del cuerpo del mensaje por privacidad
- Preservación de datos binarios (almacenamiento de PDU en bruto en Mnesia y CDR)
- Capacidad de búsqueda de texto completo (si se habilita en la base de datos CDR)
- Campos CDR indexados para consultas rápidas de interceptación legal

Seguimiento de Frontend:

- Seguimiento en tiempo real de frontends SMSC externos (puertas de enlace SMPP, IMS, MAP)
- Registro de frontend con monitoreo de latidos
- Seguimiento del estado de salud (activo/expirado)
- Historial de tiempo de actividad/inactividad
- Seguimiento de dirección IP y nombre de host
- Registro de configuración específica de frontend

1.2.3 Capacidades de Análisis

Monitoreo en Tiempo Real:

- Panel de control de interfaz web que muestra:
 - Cola de mensajes activa
 - Envío y entrega de mensajes
 - Decisiones de enrutamiento y selección de puerta de enlace

- Estado de la puerta de enlace de frontend
- Utilización de recursos del sistema
- Integración de métricas Prometheus para monitoreo operativo
- Métricas de rendimiento (rendimiento, latencia, tasas de éxito)

☐ **Guía de monitoreo completa en [OPERATIONS_GUIDE.md](#)**

☐ **Documentación de métricas en [METRICS.md](#)**

Análisis Histórico:

- Base de datos CDR consultable por:
 - Rango de tiempo
 - Número de parte que llama/llamada
 - Estado del mensaje
 - Puerta de enlace SMSC
 - Intentos de entrega
 - Contenido del mensaje (búsqueda de texto completo si se habilita)
- Capacidades de análisis estadístico:
 - Volumen de mensajes por hora/día/mes
 - Tasas de éxito/fallo por ruta
 - Tiempos de entrega promedio
 - Análisis de mensajes multipartes
 - Patrones de entrega fallida

Seguimiento de Suscriptores:

- Historial de mensajes por número de teléfono (MSISDN)
- Seguimiento basado en IMSI (cuando está disponible desde frontends IMS/MAP)
- Análisis de patrones de llamada
- Correlación de partes de comunicación
- Análisis temporal (frecuencia de mensajes, patrones de tiempo)

Análisis de Red:

- Métricas de rendimiento de ruta

- Disponibilidad y salud de la puerta de enlace
- Visualización del flujo de mensajes
- Distribución de nodos de clúster (despliegues multinodo)
- Análisis de intentos de entrega
- Análisis de patrones de reintento

Inteligencia de Números:

- Normalización de números E.164
- Identificación de país/región a partir del prefijo del número
- Reglas de traducción y reescritura de números
- Búsqueda DNS ENUM para inteligencia de enrutamiento
- Decisiones de enrutamiento basadas en prefijos

□ **Guía de traducción de números en [number_translation_guide.md](#)**

□ **Guía de enrutamiento en [sms_routing_guide.md](#)**

1.3 Capacidades de Contramedidas

1.3.1 Mecanismos de Protección de Privacidad

Confidencialidad de la Comunicación:

- HTTPS/TLS para comunicaciones de REST API
- Autenticación basada en certificados
- Cifrado de conexión a la base de datos (soporte TLS)
- Opción de eliminación del cuerpo del mensaje configurable después de la entrega

Control de Acceso:

- Control de acceso a la interfaz web
- Mecanismos de autenticación de API
- Controles de acceso a la base de datos
- Autenticación de registro de frontend

Registro de Auditoría:

- Registro completo de eventos del sistema
- Registro de envío/entrega de mensajes
- Seguimiento de cambios de configuración
- Registro de acciones administrativas
- Registro estructurado con niveles configurables

1.3.2 Características de Protección de Datos

Privacidad del Mensaje:

- Opción de eliminación del cuerpo del mensaje configurable después de la entrega
- Cuerpo del mensaje excluido de la visualización de la interfaz (opcional)
- Cuerpo del mensaje excluido de las exportaciones (opcional)
- Campo del cuerpo del mensaje CDR puede establecerse en NULL por privacidad

Seguridad de la Base de Datos:

- Soporte de cifrado de tabla MySQL (ENCRYPTION='Y')
- Soporte de cifrado de datos transparente en PostgreSQL
- Separación de roles de acceso a la base de datos
- Cuentas de usuario de solo lectura para análisis
- Acceso restringido al contenido del mensaje

Endurecimiento del Sistema:

- Puertos de red expuestos mínimos
- Gestión de certificados TLS
- Almacenamiento de configuración seguro
- Separación de configuración basada en el entorno
- Seguridad de clúster con protocolo de distribución Erlang

1.4 Arquitectura de Almacenamiento: Diseño de Dos Niveles Mnesia + SQL

Visión General

El OmniMessage SMSc emplea una arquitectura de almacenamiento de dos niveles única diseñada específicamente para separar el procesamiento operativo de mensajes de la archivación a largo plazo para cumplimiento.

Nivel 1: Cola de Mensajes en Memoria Mnesia

¿Qué es Mnesia?

- Base de datos distribuida integrada en el tiempo de ejecución Erlang/OTP
- Almacenamiento híbrido: Primario en memoria con respaldo automático en disco
- Transacciones compatibles con ACID
- Replicación de clúster entre múltiples nodos

Modo de Almacenamiento: `disc_copies`

- **Primario en Memoria:** Todos los mensajes activos almacenados en RAM
 - Operaciones de lectura/escritura ultrarrápidas (sub-milisegundo)
 - Sin I/O de disco durante operaciones normales de enrutamiento de mensajes
 - Permite un rendimiento de más de 1,750 mensajes/segundo
- **Respaldo en Disco (Automático):** Mnesia sincroniza RAM con disco
 - Las escrituras ocurren de manera asíncrona en segundo plano
 - La copia en disco se actualiza en cada confirmación de transacción
 - Recuperación ante fallos: El sistema se reinicia con todos los mensajes intactos
 - Ubicación: directorio `Mnesia.*/*` en datos de la aplicación

Ciclo de Vida del Mensaje en Mnesia:

1. Mensaje llega a través de REST API → Insertado en RAM de Mnesia + respaldo en disco

2. Motor de enrutamiento consulta Mnesia → Respuesta instantánea (acceso a memoria)
3. Puerta de enlace externa consulta mensajes → Consulta Mnesia (acceso a memoria)
4. Puerta de enlace actualiza estado de entrega → Actualización de Mnesia (memoria + disco)
5. Después de entrega/expiración → Mensaje marcado para limpieza
6. Trabajador de limpieza (24h por defecto) → Mensaje eliminado de Mnesia

Característica Crítica de Rendimiento:

- **CERO consultas a la base de datos SQL** durante el enrutamiento/entrega activa de mensajes
- SQL se omite completamente para el procesamiento operativo de mensajes
- Esto elimina el cuello de botella tradicional del SMS-C (I/O de base de datos)

Nivel 2: Base de Datos SQL para Exportación/Archivado de CDR

¿Qué es CDR (Registro de Detalles de Llamadas)?

- Registro de auditoría permanente de metadatos y contenido de mensajes
- Escrito en base de datos MySQL o PostgreSQL
- Utilizado para facturación, análisis, cumplimiento e interceptación legal

Cuándo se Escriben los CDR: Los registros CDR se crean SOLAMENTE cuando los mensajes alcanzan un estado final:

- Mensaje entregado con éxito
- Mensaje expirado (excedió el período de validez sin entrega)
- Mensaje fallido permanentemente (número inválido, error de enrutamiento)
- Mensaje rechazado (reglas de enrutamiento, fallo de validación)

Cómo se Escriben los CDR:

- **Escritura por lotes asíncrona:** CDRs escritos en proceso de trabajo en segundo plano

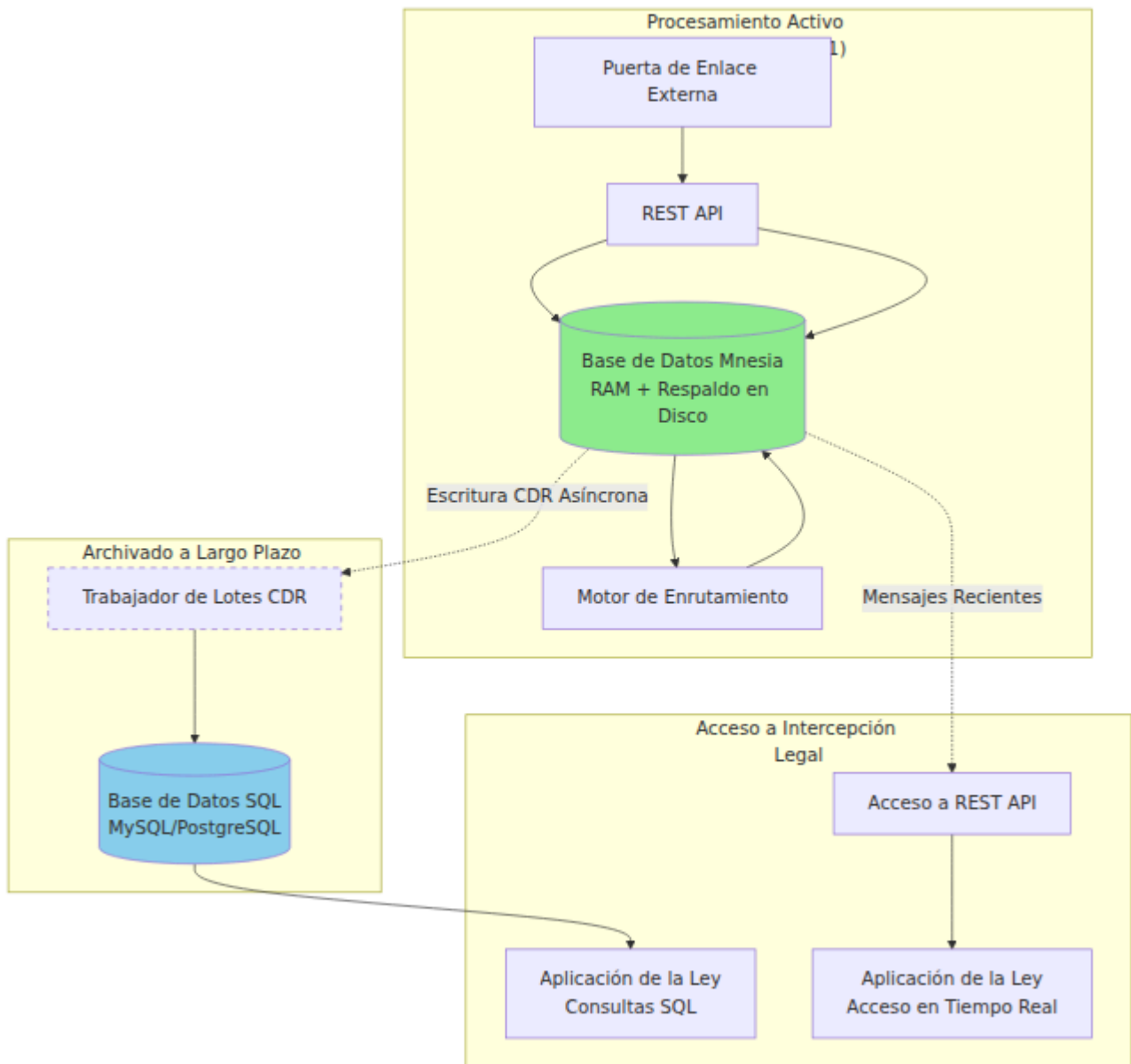
- **Sin bloqueo:** El enrutamiento de mensajes nunca espera por escritura SQL
- **Inserciones por lotes:** Múltiples CDRs agrupados (100 por defecto) y escritos juntos
- **Intervalo de vaciado:** 100ms por defecto (configurable)
- **Manejo de errores:** Escrituras de CDR fallidas registradas, el procesamiento de mensajes continúa

```
# Configuración en config/runtime.exs
config :sms_c,
  batch_insert_batch_size: 100,          # Tamaño del lote para
  escrituras de CDR
  batch_insert_flush_interval_ms: 100   # Intervalo de vaciado
```

Propósito de la Base de Datos SQL:

- NO utilizada para: Operaciones de cola de mensajes activas
- NO utilizada para: Decisiones de enrutamiento de mensajes
- NO utilizada para: Entrega de mensajes en tiempo real
- SOLAMENTE utilizada para: Archivado a largo plazo de CDR y consultas históricas
- SOLAMENTE utilizada para: Consultas de interceptación legal (meses/años de historia)
- SOLAMENTE utilizada para: Informes de facturación y análisis

Diagrama de Arquitectura



Leyenda:

- Líneas sólidas: Operaciones sincrónicas (tiempo real)
- Líneas discontinuas: Operaciones asíncronas (en segundo plano)
- Verde: Nivel de alto rendimiento (en memoria)
- Azul: Nivel de archivo (SQL persistente)

Implicaciones de Interceptación Legal

Mensajes Recientes (< 24 horas):

- Accesibles a través de Mnesia (consultas de REST API)
- Recuperación ultra-rápida
- Contenido completo del mensaje disponible

- Monitoreo en tiempo real posible

Mensajes Históricos (> 24 horas):

- Accesibles a través de la base de datos SQL (tabla CDR)
- Rendimiento de consulta SQL estándar
- Metadatos completos del mensaje siempre disponibles
- Cuerpo del mensaje disponible (a menos que se habilite el modo de privacidad)

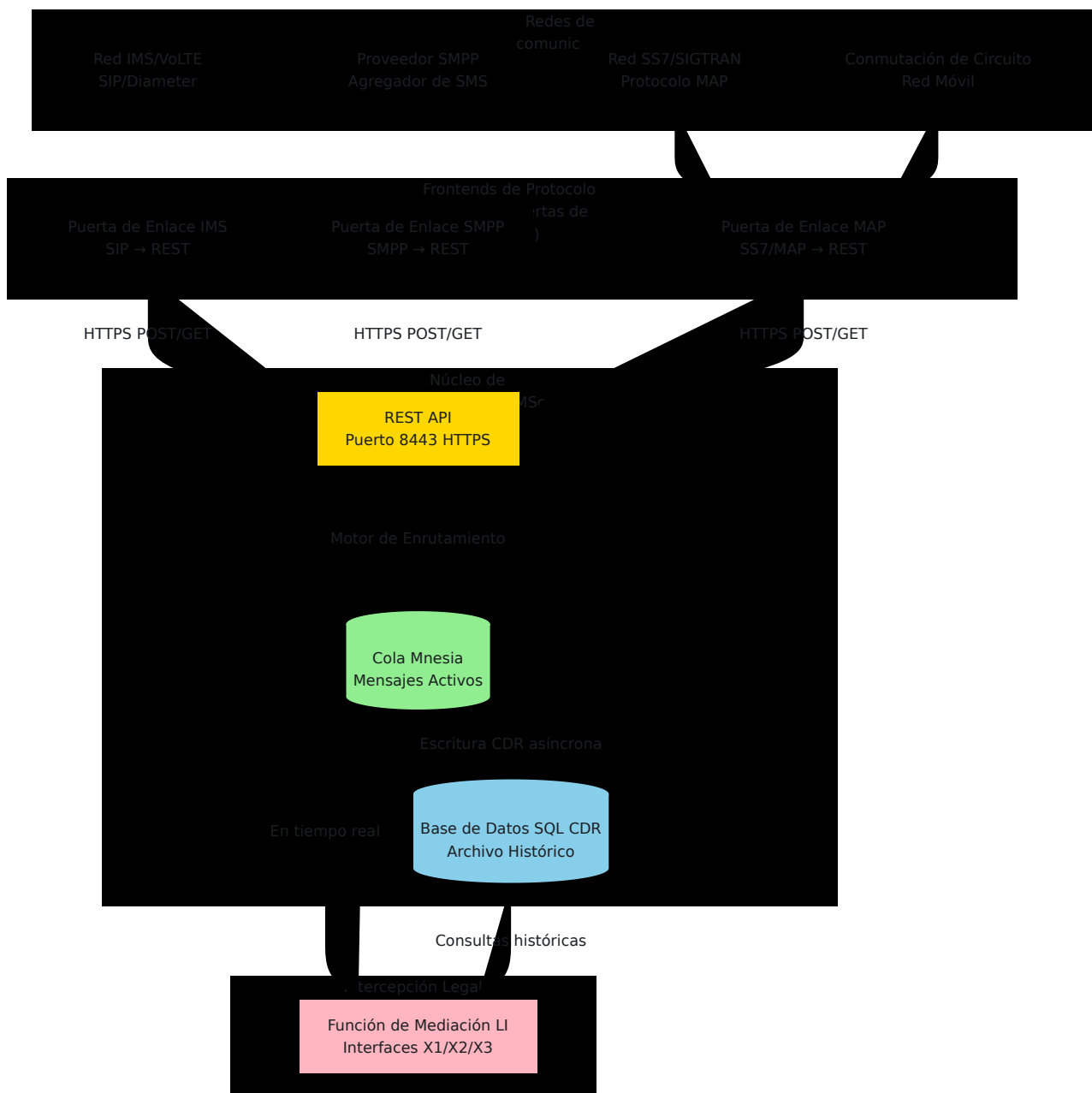
Beneficios de Cumplimiento:

1. **Sin pérdida de datos:** El modo `disc_copies` asegura que los mensajes sobrevivan a fallos
2. **Auditoría permanente:** Los CDR se retienen durante años en la base de datos SQL
3. **Rendimiento:** Las consultas de interceptación legal no impactan el enrutamiento de mensajes
4. **Flexibilidad:** Mensajes recientes (Mnesia) + mensajes históricos (SQL) ambos accesibles

1.5 Arquitectura de Integración de Frontend Multi-Protocolo

El OmniMessage SMSc emplea un diseño central agnóstico al protocolo que se conecta con puertas de enlace específicas de protocolo externas (frontends) a través de una REST API unificada. Esta arquitectura permite que la interceptación legal capture mensajes independientemente de qué protocolo de telecomunicaciones se utilizó para enviarlos o recibirlos.

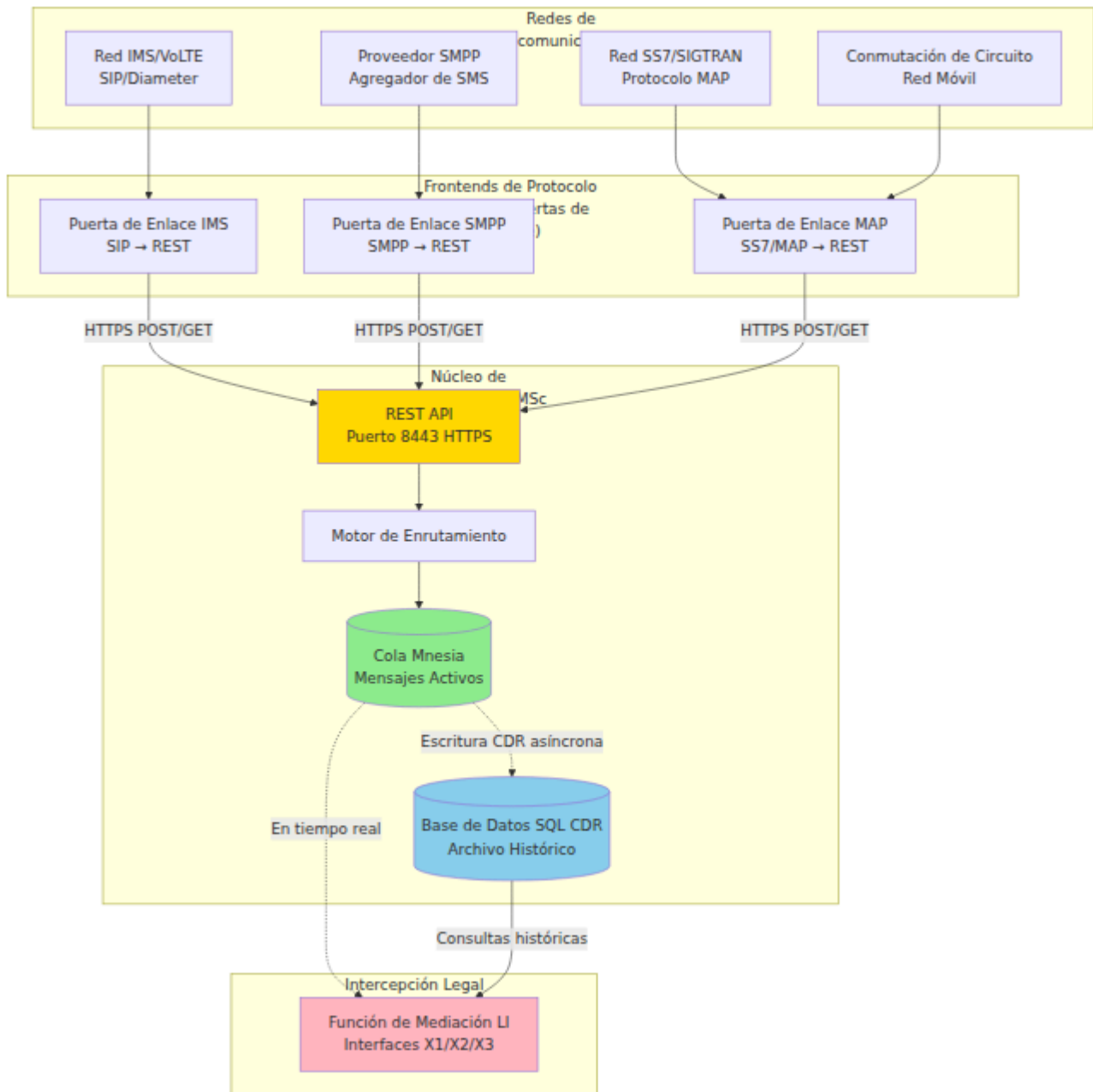
Visión General de la Arquitectura



Detalles de Integración de Frontend por Protocolo

1. Integración de Frontend IMS/SIP

Las redes IMS utilizan el protocolo SIP para mensajería SMS sobre IP. La puerta de enlace IMS traduce entre SIP y la REST API del SMSc.



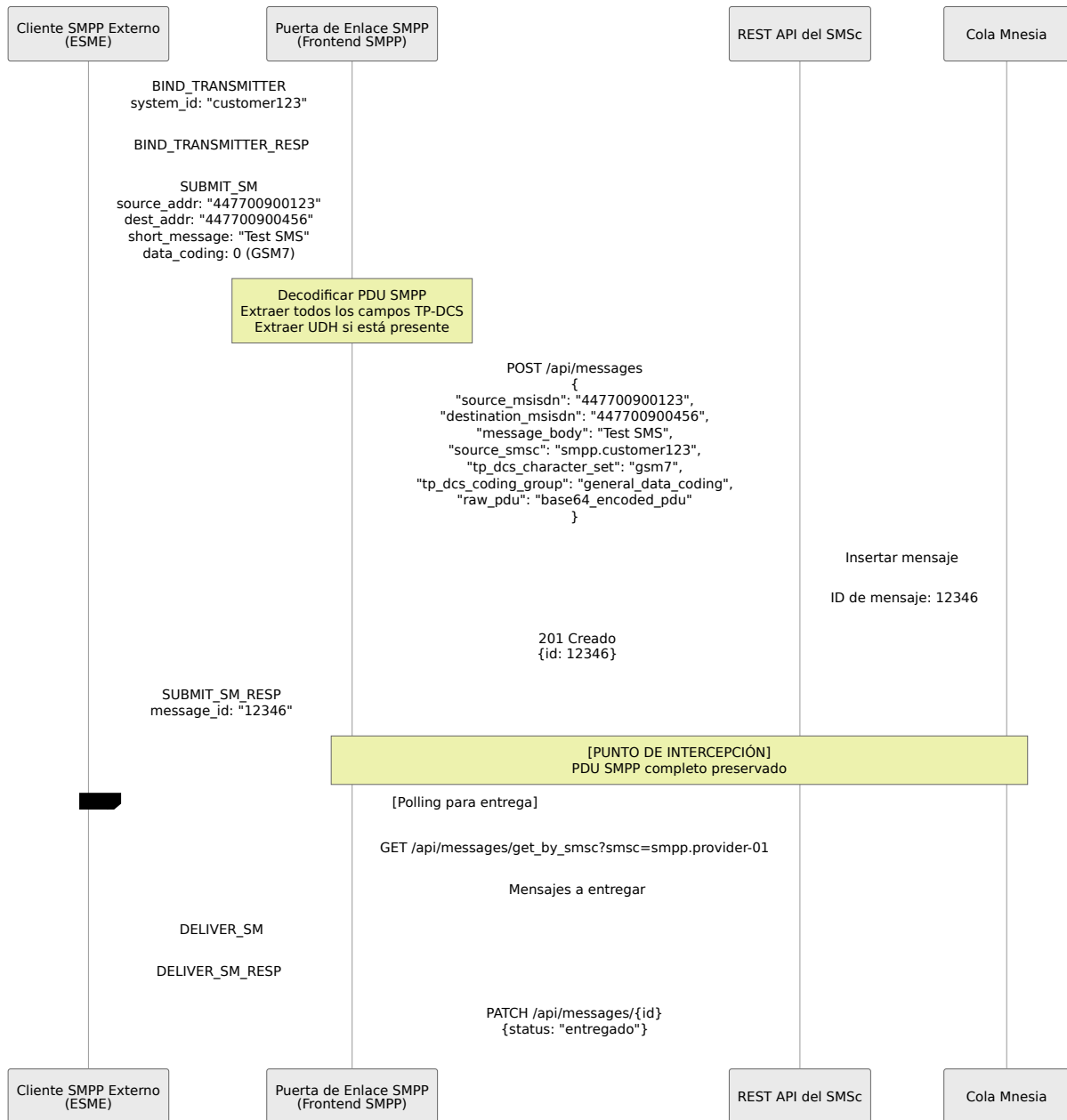
Datos de Interceptación Específicos de IMS:

- IMSI de origen/destino (de registro IMS)
- Encabezados SIP de Identidad Afirmada
- SIP Call-ID para correlación
- Ubicación de la red IMS (P-Access-Network-Info)
- Perfiles de suscriptores desde HSS de IMS

2. Integración de Frontend SMPP

SMPP es el protocolo estándar de la industria para agregadores y proveedores de SMS. La puerta de enlace SMPP traduce mensajes basados en PDU a

llamadas de REST API.

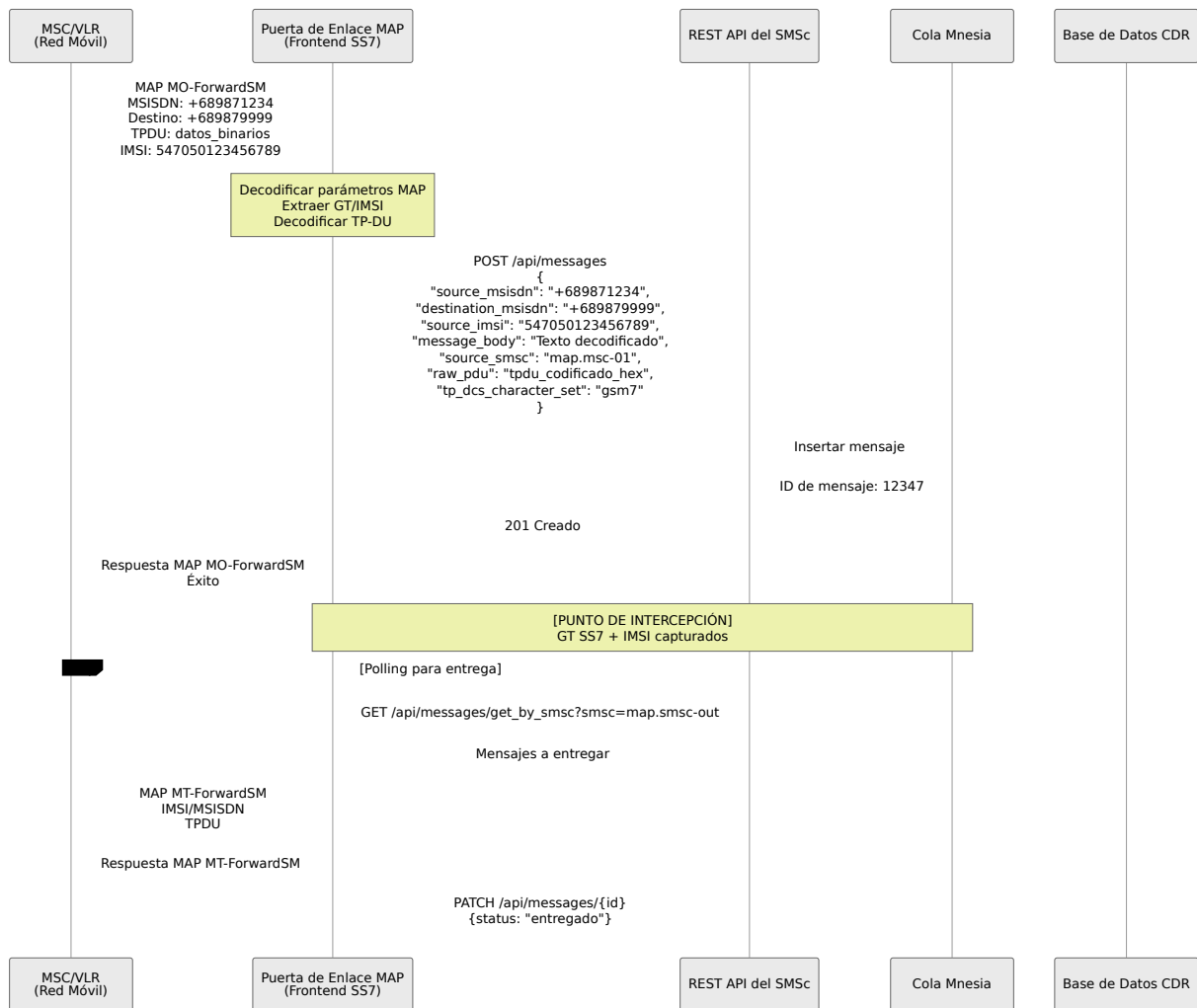


Datos de Intercepción Específicos de SMPP:

- PDU SMPP completo (formato binario preservado)
- Detalles del Esquema de Codificación de Datos (DCS)
- Encabezado de Datos de Usuario (UDH) para mensajes concatenados
- ID del sistema ESME (identificación del cliente)
- Información del plan de numeración TON/NPI
- Banderas de entrega registrada

3. Integración de Frontend SS7/MAP

Las redes de conmutación de circuitos heredadas utilizan el protocolo MAP SS7 para SMS. La puerta de enlace MAP traduce entre señales SS7 y REST API.



Datos de Interceptación Específicos de SS7/MAP:

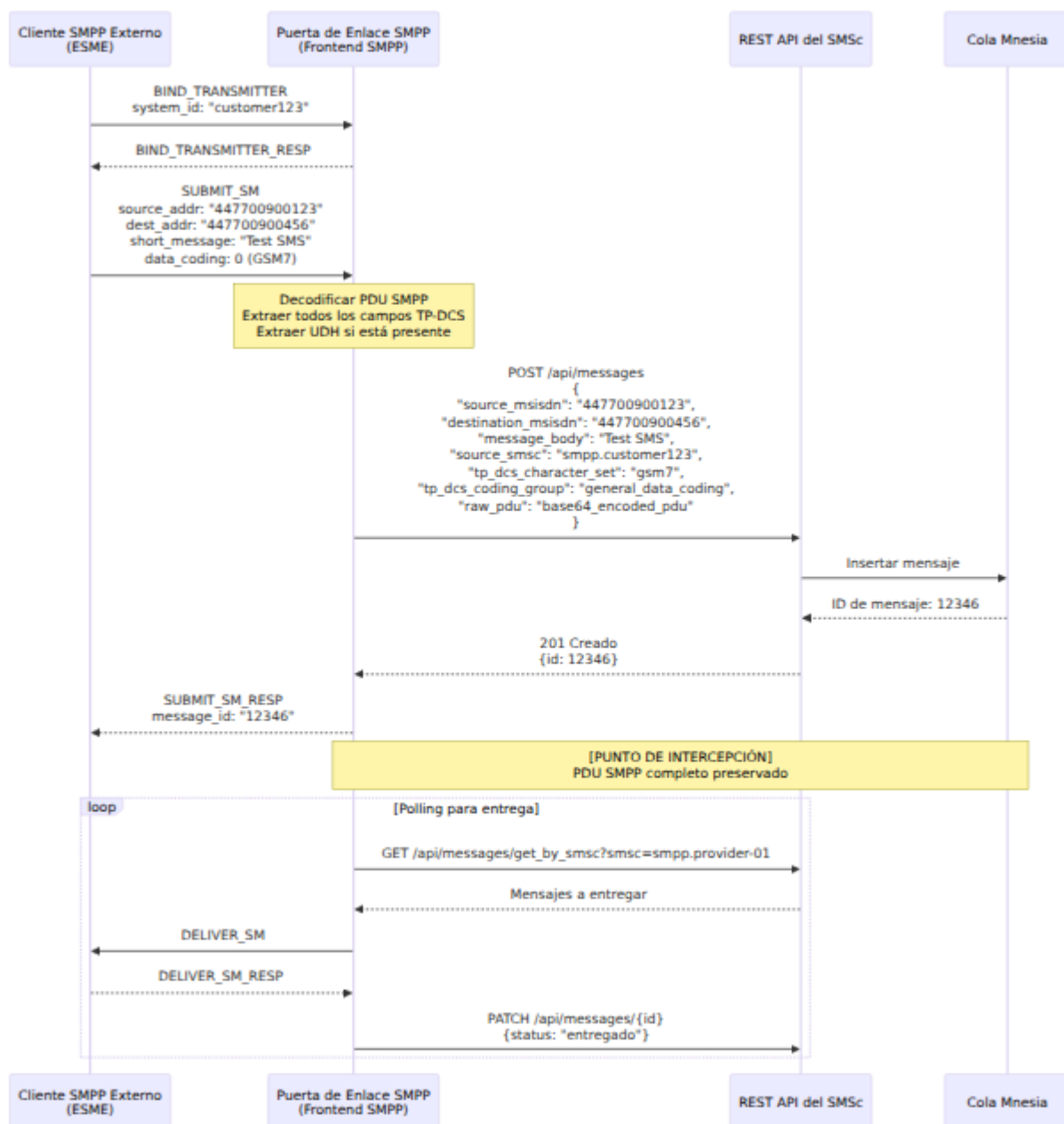
- IMSI de mensajes MAP
- Direcciones de Título Global (GT)
- Dirección MSC/VLR (identificación del elemento de red)
- Direcciones de parte que llama/llamada SCCP
- Códigos de operación MAP
- Formato binario TP-User-Data

Intercepción Unificada a Través de Todos los Protocolos

Beneficio Clave para la Interceptación Legal: Independientemente de qué protocolo se utilizó (IMS/SIP, SMPP o SS7/MAP), todos los mensajes convergen en el núcleo del SMSc con una estructura de datos normalizada, permitiendo:

1. **Monitoreo Agnóstico al Protocolo:** Un solo punto de intercepción captura todos los tipos de mensajes
2. **Formato CDR Unificado:** Todos los protocolos escriben en el mismo esquema CDR
3. **Correlación Inter-Protocolo:** Rastrear mensajes a través de límites de protocolo
4. **Preservación Completa de Metadatos:** Campos específicos de protocolo preservados en CDR

Resumen del Flujo de Datos:



Identificación de Protocolo en CDR:

- El campo `source_smsc` indica el protocolo de frontend (por ejemplo, "ims.gateway-01", "smpp.customer123", "map.msc-01")
- Permite filtrado y análisis por tipo de protocolo
- Consultas de interceptación legal pueden dirigirse a protocolos específicos o a todos los protocolos

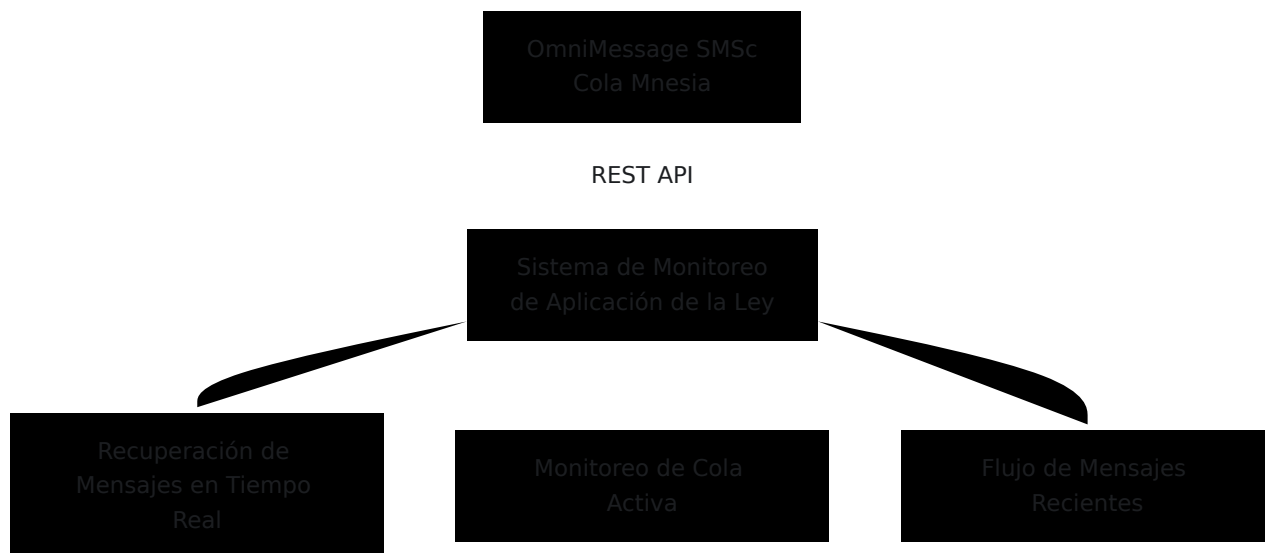
1.6 Arquitectura Técnica para Interceptación Legal

Puntos de Integración de Interceptación Legal

La arquitectura de almacenamiento de dos niveles proporciona múltiples puntos de acceso para la interceptación legal, optimizados tanto para monitoreo en tiempo real (Mnesia) como para análisis histórico (SQL).

1. Acceso a REST API para Mensajes Recientes (Mnesia):

Acceso a mensajes activos en la cola de Mnesia (típicamente las últimas 24 horas):



Puntos finales de API para Interceptación en Tiempo Real:

- `GET /api/messages` - Listar mensajes activos con filtrado
- `GET /api/messages/{id}` - Obtener detalles de un mensaje específico (de Mnesia)

- `GET /api/messages/get_by_smsc?smsc=X` - Obtener mensajes por puerta de enlace
- Todas las consultas impactan a Mnesia (en memoria) para respuesta instantánea

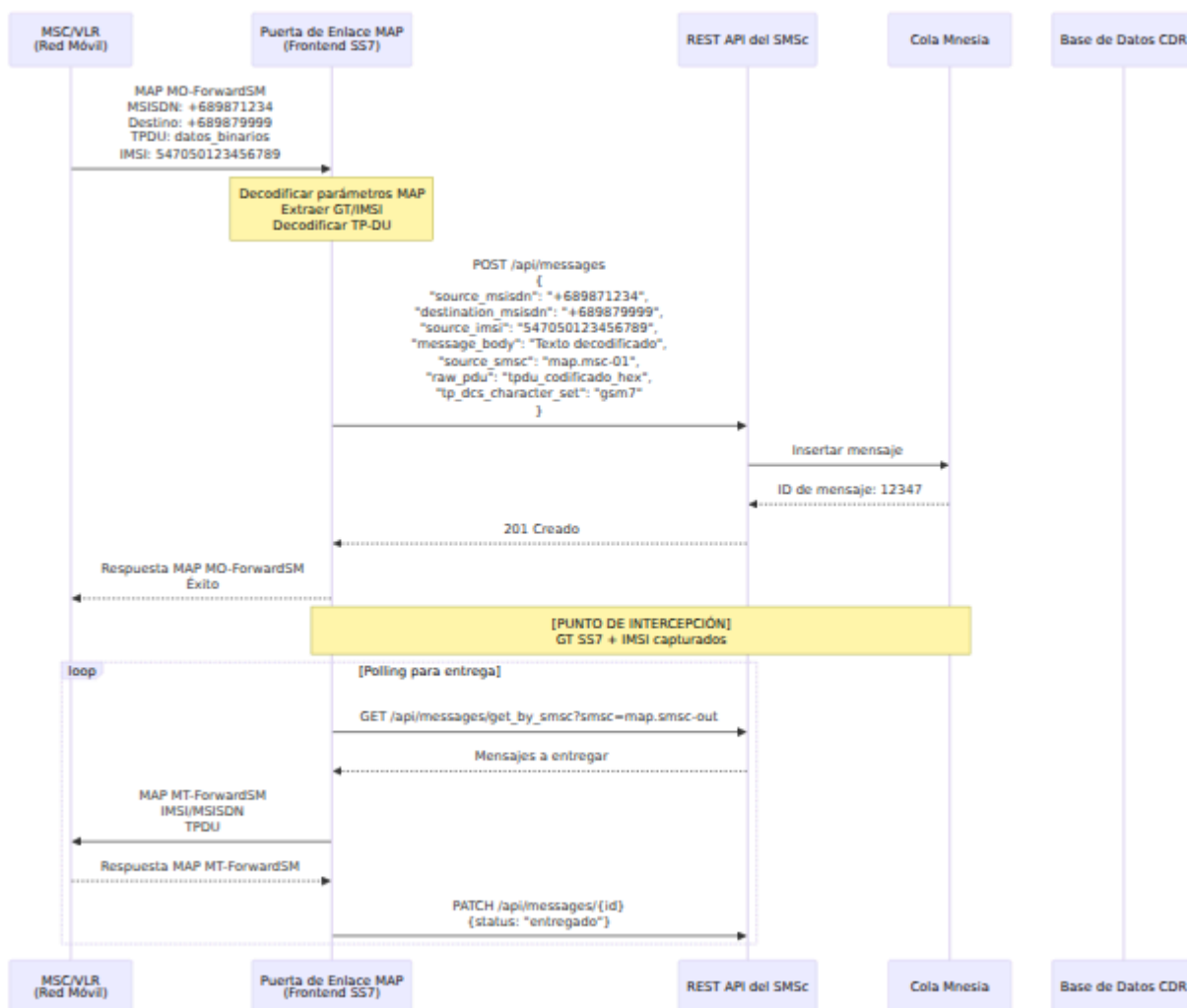
Nota: Estos puntos finales consultan la cola de mensajes activa de Mnesia, proporcionando acceso a mensajes que están siendo procesados actualmente o que fueron entregados recientemente (dentro del período de retención).

Parámetros de Consulta:

- Filtrar por MSISDN de origen/destino
- Filtrar por rango de tiempo
- Filtrar por puerta de enlace SMSC
- Filtrar por estado del mensaje
- Soporte de ordenamiento y paginación

2. Acceso Directo a la Base de Datos CDR para Mensajes Históricos (SQL):

Acceso a mensajes archivados en la base de datos SQL (todos los mensajes entregados/expirados/fallidos):



Acceso Directo SQL:

- Credenciales de base de datos de solo lectura para sistemas autorizados
- Acceso a consultas SQL a la tabla `cdrs` (auditoría permanente)
- **Método de Acceso:** Cliente SQL estándar (mysql, psql, DBeaver, etc.)
- **Fuente de Datos:** Solo mensajes archivados (no cola activa)
- Campos indexados para búsqueda eficiente:
 - `calling_number` (indexado) - Número de teléfono de origen
 - `called_number` (indexado) - Número de teléfono de destino
 - `message_id` (indexado) - Identificador único del mensaje
 - `submission_time` (indexado) - Cuando el mensaje ingresó al sistema
 - `status` (indexado) - Estado final de entrega
 - `dest_smsc` (indexado) - Puerta de enlace utilizada para la entrega

Nota: La base de datos CDR contiene registros permanentes de todos los mensajes procesados. Esta es la fuente de datos principal para consultas

históricas de intercepción legal (meses/años de datos).

3. Flujo de Mensajes en Tiempo Real (PubSub):

- Integración de Phoenix PubSub para eventos en tiempo real
- Notificaciones de envío de mensajes
- Notificaciones de entrega de mensajes
- Eventos de cambio de estado de mensajes
- Filtrado de eventos configurable por criterios
- Soporte de WebSocket para monitoreo en vivo

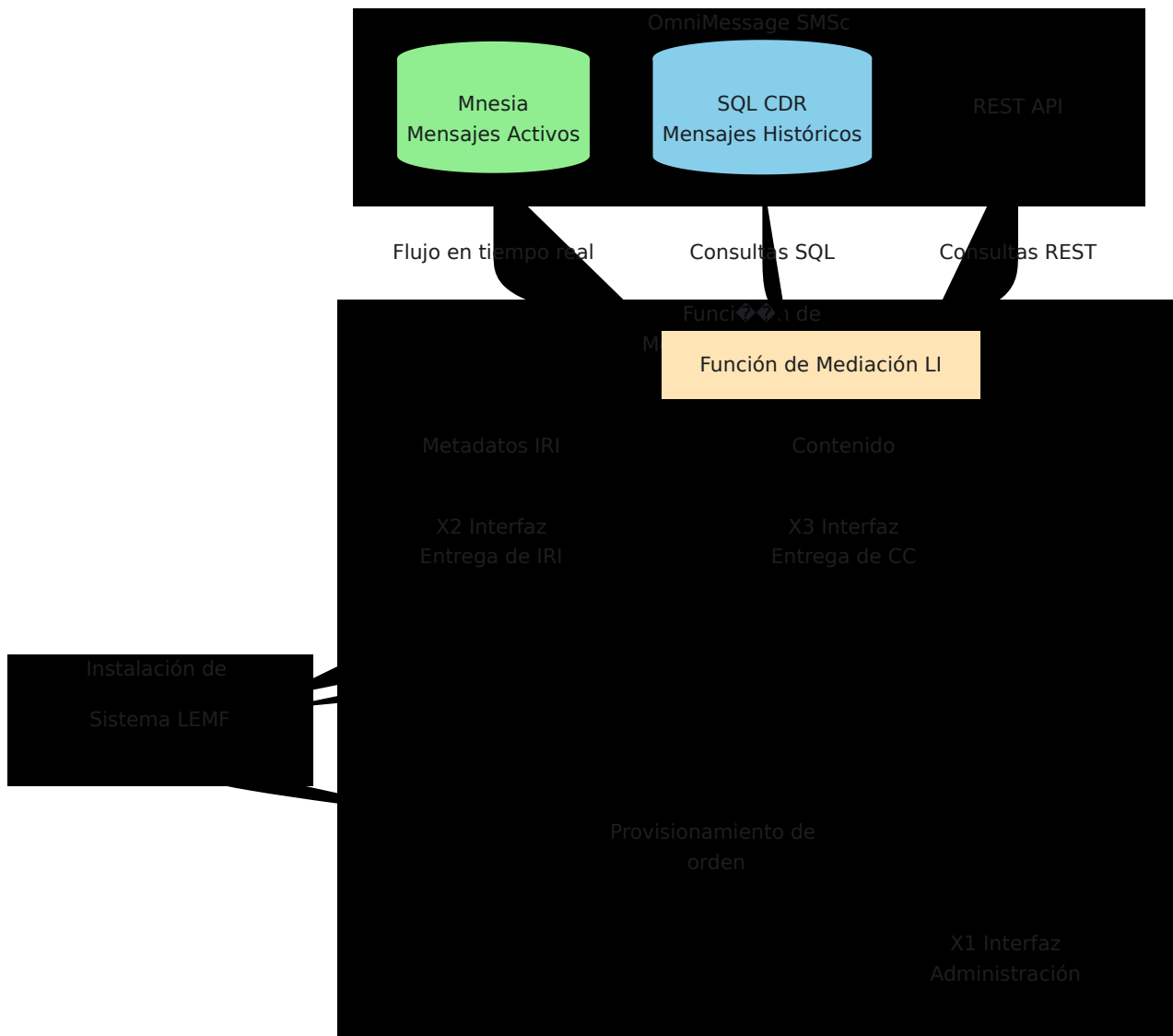
4. Interfaz de Exportación por Lotes:

- Exportación CSV de registros CDR
- Exportación JSON para acceso programático
- Campos de exportación configurables
- Exportaciones basadas en rango de tiempo
- Exportaciones conscientes de la privacidad (opción de exclusión del cuerpo del mensaje)

Interfaces de Estándar de Intercepción Legal de ETSI

El OmniMessage SMSc proporciona la base para implementar interfaces de intercepción legal compatibles con ETSI. Aunque el núcleo del SMSc no implementa nativamente las interfaces X1/X2/X3, proporciona todos los puntos de acceso de datos necesarios que pueden integrarse con sistemas externos de Función de Mediación de Intercepción Legal (LIMF).

Interfaces LI Estándar de ETSI:



Descripciones de Interfaces:

Interfaz X1 - Función Administrativa:

- **Propósito:** Provisionamiento de órdenes y objetivos de interceptación de la aplicación de la ley al sistema de interceptación
- **Dirección:** LEMF → LIMF (bidireccional)
- **Funciones:**
 - Activar/desactivar interceptación para objetivos específicos (MSISDN, IMSIs)
 - Establecer duración y período de validez de la interceptación
 - Configurar criterios de filtrado (números de teléfono, ventanas de tiempo)
 - Recuperar estado de interceptación

- **Integración con SMSc:**

- LIMF mantiene lista de objetivos (base de datos de órdenes)
- LIMF consulta CDR/API del SMSc para mensajes coincidentes
- LIMF filtra según criterios provisionados por X1

Interfaz X2 - Entrega de IRI (Información Relacionada con la Intercepción):

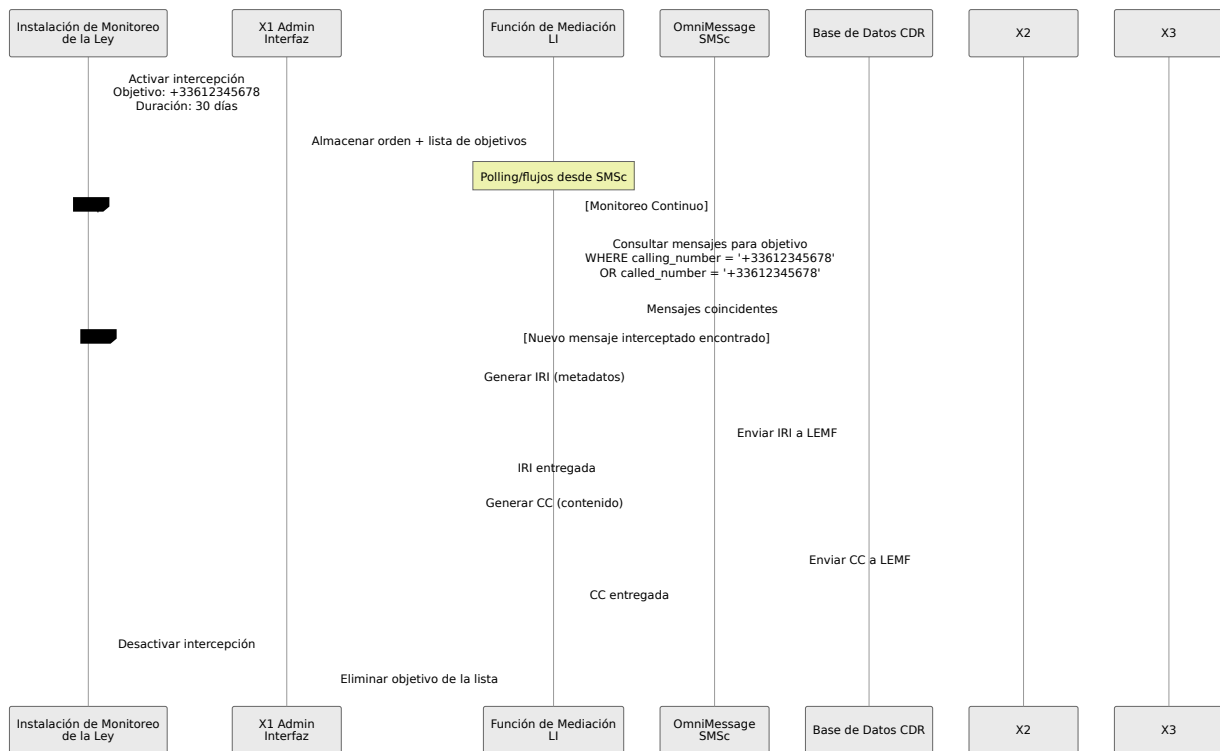
- **Propósito:** Entregar metadatos de mensajes a la aplicación de la ley
- **Dirección:** LIMF → LEMF (unidireccional)
- **Formato de Datos:** XML/ASN.1 conforme a ETSI TS 102 232-x
- **Contenido del SMSc CDR:**
 - ID de mensaje
 - Número de llamada (MSISDN de origen)
 - Número llamado (MSISDN de destino)
 - IMSI (origen y destino, si está disponible)
 - Marca de tiempo de envío
 - Marca de tiempo de entrega
 - Estado del mensaje (entregado/fallido/expirado)
 - Intentos de entrega
 - Información de puerta de enlace SMSC (origen/destino)
 - Ubicación de la red (si está disponible)
- **Integración con SMSc:**
 - LIMF consulta la base de datos CDR para números de teléfono objetivo
 - LIMF transforma registros CDR en formato ETSI IRI
 - LIMF entrega IRI a LEMF a través de X2

Interfaz X3 - Entrega de CC (Contenido de la Comunicación):

- **Propósito:** Entregar el contenido real del mensaje a la aplicación de la ley
- **Dirección:** LIMF → LEMF (unidireccional)
- **Formato de Datos:** Conforme a ETSI TS 102 232-x
- **Contenido del SMSc:**
 - Cuerpo del mensaje (contenido de texto)
 - PDU en bruto (datos binarios de SMS)

- Información de codificación de caracteres
- Segmentos de mensajes multipartes
- Información TP-DCS
- Encabezado de Datos de Usuario (UDH)
- **Integración con SMSc:**
 - LIMF recupera el contenido del mensaje del campo `message_body` de CDR
 - LIMF recupera datos de PDU en bruto si están disponibles
 - LIMF empaqueta el contenido en formato ETSI CC
 - LIMF entrega CC a LEMF a través de X3

Arquitectura de Implementación:



Mapeo de Datos del SMSc a las Interfaces LI:

Campo de Datos del SMSc	X2 (IRI)	X3 (CC)	Columna de Tabla CDR
ID de Mensaje	☐ ID de correlación	☐ Referencia	message_id
Número de Llamada	☐ Parte A	-	calling_number
Número Llamado	☐ Parte B	-	called_number
Marca de Tiempo de Envío	☐ Marca de tiempo	-	submission_time
Marca de Tiempo de Entrega	☐ Finalización	-	delivery_time
Estado	☐ Resultado	-	status
Cuerpo del Mensaje	-	☐ Contenido	message_body
PDU en Bruto	-	☐ Binario	(Mnesia/CDR)
SMSC de Origen	☐ Elemento de red	-	source_smsc
SMSC de Destino	☐ Elemento de red	-	dest_smsc
IMSI	☐ ID de Suscriptor	-	(A través de frontends)

Opciones de Integración LIMF:

Opción 1: Arquitectura de Polling

- LIMF consulta periódicamente la base de datos CDR (cada 1-60 segundos)

- Consultas SQL filtran por números de teléfono objetivo de la lista de órdenes X1
- Baja complejidad, fácil de implementar
- Ligero retraso entre la entrega del mensaje y la entrega de LI

Opción 2: Arquitectura de Flujo en Tiempo Real

- SMSc PubSub publica eventos de mensajes
- LIMF se suscribe al flujo de mensajes en tiempo real
- LIMF filtra según la lista de objetivos
- Latencia casi cero para la intercepción legal
- Requiere desarrollo de integración personalizado

Opción 3: Arquitectura Híbrida

- Mensajes recientes: Flujo en tiempo real PubSub (< 24 horas)
- Mensajes históricos: Polling de base de datos CDR
- Balance óptimo de latencia y confiabilidad

Mecanismos de Activación de Intercepción

Intercepción Basada en Objetivos:

- Coincidencia de números de teléfono (MSISDN)
- Objetivos basados en IMSI (cuando están disponibles)
- Listas de vigilancia configurables
- Vistas de base de datos para aislamiento de objetivos
- Filtrado de API por identificadores de objetivos

Intercepción Basada en Eventos:

- Todos los mensajes de/hacia números específicos
- Mensajes a través de puertas de enlace SMSC específicas
- Mensajes con características específicas (multipartes, entrega fallida, etc.)
- Enrutamiento geográfico (a través de ENUM o coincidencia de prefijos)

Intercepción Basada en Tiempo:

- Filtrado de rango de fecha/hora en consultas CDR
- Aplicación de período de retención
- Archivado automático de mensajes antiguos
- Políticas de retención de datos configurables

Ejemplo de Consultas SQL para Intercepción Legal:

```
-- Obtener todos los mensajes para el número objetivo
SELECT * FROM cdrs
WHERE calling_number = '+33612345678'
      OR called_number = '+33612345678'
ORDER BY submission_time DESC;

-- Obtener mensajes en una ventana de tiempo específica
SELECT * FROM cdrs
WHERE (calling_number = '+33612345678' OR called_number =
'+33612345678')
      AND submission_time BETWEEN '2025-11-01 00:00:00' AND '2025-11-
30 23:59:59'
ORDER BY submission_time;

-- Obtener conversación entre dos partes
SELECT * FROM cdrs
WHERE (calling_number = '+33612345678' AND called_number =
'+33687654321')
      OR (calling_number = '+33687654321' AND called_number =
'+33612345678')
ORDER BY submission_time;
```

2. CAPACIDADES DE CIFRADO Y CRIPTOANÁLISIS

2.1 Visión General de Capacidades Criptográficas

El OmniMessage SMS Sc implementa mecanismos criptográficos para asegurar comunicaciones y proteger datos sensibles. Esta sección documenta todas las capacidades criptográficas de acuerdo con los requisitos de ANSSI.

2.2 Cifrado de Capa de Transporte

2.2.1 Implementación de TLS/SSL

Protocolos Soportados:

- TLS 1.2 (RFC 5246)
- TLS 1.3 (RFC 8446) - Recomendado
- SSL 2.0/3.0: NO SOPORTADO (vulnerabilidades conocidas)
- TLS 1.0/1.1: DEPRECATED (no recomendado)

Implementación:

- Biblioteca SSL/TLS de Erlang/OTP (validada criptográficamente)
- Servidor web Cowboy con soporte TLS
- Puntos finales HTTPS del Phoenix Framework

Conjuntos de Cifrado:

El sistema utiliza la selección de conjuntos de cifrado seguros predeterminados de Erlang/OTP, que incluye:

Preferido - TLS 1.3:

- TLS_AES_256_GCM_SHA384
- TLS_AES_128_GCM_SHA256
- TLS_CHACHA20_POLY1305_SHA256

Soportado - TLS 1.2:

- ECDHE-RSA-AES256-GCM-SHA384
- ECDHE-RSA-AES128-GCM-SHA256
- DHE-RSA-AES256-GCM-SHA384
- DHE-RSA-AES128-GCM-SHA256

Características de Seguridad:

- Secreto Perfecto hacia Adelante (PFS) a través de intercambio de claves ECDHE/DHE
- Grupos de Diffie-Hellman fuertes (mínimo 2048 bits)
- Soporte de Criptografía de Curva Elíptica
- Soporte de Indicación de Nombre de Servidor (SNI)

Gestión de Certificados:

- Soporte de certificados X.509
- Tamaños de clave RSA: mínimo 2048 bits, recomendado 4096 bits
- Soporte de ECDSA
- Validación de cadena de certificados
- Certificados autofirmados (solo para desarrollo)
- Integración con CA externa

Ubicación de Configuración de TLS:

```
# config/runtime.exs
config :api_ex,
  api: %{
    enable_tls: true,
    tls_cert_path: "priv/cert/omnitouch.crt",
    tls_key_path: "priv/cert/omnitouch.pem"
  }
```

☐ Referencia de configuración completa en [CONFIGURATION.md](#)

Aplicaciones:

- HTTPS para REST API (puerto 8443)
- HTTPS para panel de control web (puerto 8086)
- Conexiones a la base de datos (MySQL/PostgreSQL sobre TLS)

2.3 Cifrado de Datos en Reposo

2.3.1 Cifrado de Base de Datos

Cifrado de MySQL/MariaDB:

- Soporte de cifrado a nivel de tabla
- Algoritmo de cifrado AES-256
- Cifrado de datos transparente (TDE)

```
-- Habilitar cifrado para la tabla CDR  
ALTER TABLE cdrs ENCRYPTION='Y';
```

Cifrado de PostgreSQL:

- Soporte de cifrado de datos transparente
- Cifrado a nivel de sistema de archivos
- Cifrado a nivel de columna (extensión pgcrypto)

2.3.2 Almacenamiento en Disco de Mnesia

Base de Datos Mnesia:

- Almacenamiento de copias en disco para persistencia de mensajes
- Cifrado a nivel de sistema de archivos recomendado (LUKS, dm-crypt)
- Protección de memoria a través de aislamiento de VM Erlang

2.3.3 Cifrado de Sistema de Archivos

Almacenamiento de Datos Sensibles:

- Archivos de configuración: Se recomienda cifrado de sistema de archivos
- Claves privadas: Permisos de archivo (0600) + cifrado de sistema de archivos

- Archivos de registro: Cifrado configurable para registros archivados
- Exportaciones CDR: Almacenamiento cifrado para exportaciones sensibles

Almacenamiento de Claves:

- Certificados y claves TLS almacenados en `priv/cert/`
- Almacenes de claves basados en archivos con permisos restringidos
- Procedimientos seguros de rotación de claves

2.4 Autenticación y Control de Acceso

2.4.1 Autenticación de API

Seguridad de REST API:

- Cifrado de transporte HTTPS/TLS obligatorio
- Autenticación basada en encabezados (encabezado SMSc para identificación de frontend)
- Control de acceso basado en IP (nivel de firewall)
- Autenticación de cliente basada en certificados (opcional)

Registro de Frontend:

- Identificación única de frontend (nombre, tipo, IP, nombre de host)
- Autenticación basada en latidos
- Gestión de sesiones basada en expiración (tiempo de espera de 90 segundos)
- Seguimiento y monitoreo de frontend

2.4.2 Autenticación de Base de Datos

Control de Acceso a la Base de Datos:

- Autenticación de nombre de usuario/contraseña
- Soporte de conexión TLS/SSL
- Restricciones de conexión basadas en IP
- Control de acceso basado en roles (RBAC)

Configuración:

```
# config/runtime.exs
config :sms_c, SmsC.Repo,
  username: "omnitouch",
  password: "omnitouch2024", # Debe usar contraseñas fuertes en
  producción
  hostname: "localhost",
  ssl: true # Habilitar TLS para conexiones a la base de datos
```

Recomendaciones de Control de Acceso:

```
-- Crear usuario de solo lectura para acceso de la ley
CREATE USER 'li_readonly'@'%' IDENTIFIED BY 'secure_password';
GRANT SELECT ON sms_c.cdcs TO 'li_readonly'@'%';

-- Crear usuario limitado sin acceso al contenido del mensaje
CREATE USER 'analytics'@'%' IDENTIFIED BY 'secure_password';
GRANT SELECT (id, message_id, calling_number, called_number,
              source_smsc, dest_smsc, submission_time,
              delivery_time,
              status, delivery_attempts)
ON sms_c.cdcs TO 'analytics'@'%';
```

2.5 Detalles de Algoritmos Criptográficos

2.5.1 Algoritmos de Hash

Disponibles en Erlang/OTP:

- SHA-256, SHA-384, SHA-512 (recomendado)
- SHA-1 (deprecado, solo compatibilidad heredada)
- MD5 (deprecado, no utilizado para seguridad)
- BLAKE2 (disponible en versiones modernas de OTP)

Uso:

- Huellas digitales de mensajes (detección de duplicados)

- Verificación de integridad de datos
- Integridad de registros de auditoría

2.5.2 Cifrado Simétrico

Algoritmos Disponibles:

- AES (Estándar de Cifrado Avanzado)
 - AES-128-GCM
 - AES-256-GCM
 - AES-128-CBC
 - AES-256-CBC
- ChaCha20-Poly1305

Tamaños de Clave:

- 128 bits (mínimo)
- 256 bits (recomendado)

Uso:

- Cifrado de sesión TLS
- Cifrado de base de datos en reposo
- Cifrado opcional del cuerpo del mensaje

2.5.3 Cifrado Asimétrico

Algoritmos Soportados:

- RSA (mínimo 2048 bits, recomendado 4096 bits)
- ECDSA (Algoritmo de Firma Digital de Curva Elíptica)
 - Curvas P-256, P-384, P-521
- Ed25519 (EdDSA)

Uso:

- Autenticación de certificados TLS
- Firmas digitales

- Intercambio de claves

2.6 Seguridad del Protocolo SMS

2.6.1 Codificación de Mensajes SMS

Soporte de Codificación de Caracteres:

- GSM 7-bit (codificación estándar de SMS)
- UCS-2 (Unicode, 16-bit)
- Datos binarios de 8 bits
- Latin-1

TP-DCS (Esquema de Codificación de Datos):

- Indicación de clase de mensaje
- Banderas de compresión
- Especificación de grupo de codificación
- Identificación de conjunto de caracteres

Sin Cifrado SMS Nativo:

- El protocolo SMS no proporciona cifrado de extremo a extremo
- El contenido del mensaje es accesible a nivel de SMSc
- Permite la interceptación legal según se requiera

2.6.2 Consideraciones de Seguridad del Protocolo

Protocolo SMPP (Frontend Externo):

- Autenticación de nombre de usuario/contraseña a nivel de SMPP
- Soporte TLS disponible (SMPP sobre TLS)
- Autenticación de enlace

Protocolo IMS (Frontend Externo):

- Mensajería basada en SIP
- Mecanismos de autenticación SIP

- Integración con la seguridad de la red central IMS

Protocolo SS7/MAP (Frontend Externo):

- Seguridad de la red SS7
- Autenticación del protocolo MAP
- Seguridad de la capa SCCP/TCAP

Nota: La seguridad específica del protocolo se implementa en las puertas de enlace de frontend externas, no en el núcleo del SMSc.

2.7 Capacidades de Criptoanálisis y Evaluación de Seguridad

2.7.1 Herramientas de Análisis de Protocolos

Capacidades de Depuración Integradas:

- Sistema de registro completo
- Trazado del flujo de mensajes
- Registro de solicitudes/respuestas de API
- Registro de consultas a la base de datos
- Seguimiento de errores y excepciones

Integración Externa:

- Salida de registro estándar (stdout/archivos)
- Soporte de captura PCAP para análisis de red
- Registro de consultas a la base de datos para forense
- Exportación de métricas de Prometheus

2.7.2 Consideraciones de Evaluación de Vulnerabilidades

Limitaciones Conocidas:

- Protocolo SMS inherentemente no cifrado (por diseño, permite interceptación legal)

- Credenciales de base de datos en archivos de configuración (debería usar gestión de secretos)
- Soporte de certificados autofirmados (solo para desarrollo/pruebas)

Recomendaciones de Endurecimiento de Seguridad:

- Usar conjuntos de cifrado TLS fuertes
- Implementar cifrado de conexión a la base de datos
- Usar gestión de secretos externa (Vault, AWS Secrets Manager)
- Actualizaciones de seguridad regulares para Erlang/OTP y dependencias
- Restricciones de firewall en puertos de API
- Lista blanca de IP para acceso a frontend

Pruebas de Seguridad:

- Escaneo regular de vulnerabilidades de dependencias
- Soporte de pruebas de penetración
- Validación de configuración TLS
- Auditorías de seguridad de base de datos
- Revisión de control de acceso

2.8 Infraestructura de Gestión de Claves

2.8.1 Generación de Claves

Generación de Certificado TLS:

```
# Generar clave privada (RSA 4096 bits)
openssl genrsa -out omnitouch.pem 4096

# Generar solicitud de firma de certificado
openssl req -new -key omnitouch.pem -out omnitouch.csr

# Certificado autofirmado (desarrollo)
openssl x509 -req -days 365 -in omnitouch.csr -signkey
omnitouch.pem -out omnitouch.crt

# Producción: Obtener certificado de CA confiable
```

Generación de Números Aleatorios:

- CSPRNG de Erlang/OTP (Generador de Números Aleatorios Criptográficamente Seguro)
- Pool de entropía del sistema (/dev/urandom)
- Aleatoriedad fuerte para claves de sesión, IDs, tokens

2.8.2 Almacenamiento y Protección de Claves

Almacenamiento de Clave Privada:

- Sistema de archivos con permisos restringidos (0600)
- Almacenado en el directorio `priv/cert/`
- Formato PEM (opcionalmente cifrado)
- Procedimientos seguros de respaldo

Rotación de Claves:

- Procedimientos de renovación de certificados TLS (anualmente recomendado)
- Rotación de credenciales de base de datos
- Rotación de tokens de API (si se implementa)

2.8.3 Distribución de Claves

Distribución de Certificados:

- Instalación manual en `priv/cert/`
- Referencias en archivos de configuración
- Soporte posible para el protocolo ACME (Let's Encrypt)

Distribución de Claves Simétricas:

- Intercambio de claves fuera de banda para credenciales de base de datos
- Acuerdo de claves Diffie-Hellman en TLS
- Ninguna transmisión de claves en texto claro

2.9 Cumplimiento y Normas

Esta sección documenta el cumplimiento con estándares internacionales de telecomunicaciones, marcos regulatorios y especificaciones de seguridad aplicables al procesamiento de SMS a través de todos los protocolos soportados.

2.9.1 Cumplimiento de SMS sobre Protocolo SS7/MAP

Normas de 3GPP y ETSI:

- **3GPP TS 23.040:** Realización técnica del Servicio de Mensajes Cortos (SMS) - Especificación del protocolo SMS central
- **3GPP TS 23.038:** Alfabetos e información específica de idioma - Codificación de caracteres (GSM7, UCS-2)
- **3GPP TS 29.002:** Especificación de Parte de Aplicación Móvil (MAP) - Señalización SS7 para SMS
- **3GPP TS 23.003:** Numeración, direccionamiento e identificación - Formatos MSISDN, IMSI
- **ETSI TS 100 901:** Especificación del Servicio de Mensajes Cortos Punto a Punto
- **ETSI TS 100 902:** Especificación del Servicio de Mensajes Cortos de Difusión Celular

Normas de Señalización SS7:

- **ITU-T Q.711-Q.716:** Parte de Control de Conexión de Señalización (SCCP)

- **ITU-T Q.771-Q.775:** Parte de Aplicación de Capacidades de Transacción (TCAP)
- **ITU-T Q.701-Q.710:** Parte de Transferencia de Mensajes (MTP) Niveles 1-3
- **ETSI EN 300 356:** Sistema de Señalización No.7 - Parte de Usuario ISDN (ISUP)

Normas de Seguridad para SS7/MAP:

- **GSMA FS.07:** Seguridad de Señalización SS7 y Diameter - Controles de seguridad básicos
- **GSMA FS.11:** Directrices de Monitoreo de Seguridad SS7
- **3GPP TS 33.117:** Catálogo de requisitos generales de garantía de seguridad
- **ETSI TS 133 210:** Seguridad del dominio de red - Seguridad de la capa de red IP

Intercepción Legal para SS7/MAP:

- **ETSI TS 101 671:** Intercepción Legal (LI); Interfaz de entrega para la intercepción legal del tráfico de telecomunicaciones
- **ETSI TS 102 232-1:** Intercepción Legal (LI); Especificación de entrega - Parte 1: Interfaz de entrega para gestión de LI
- **3GPP TS 33.107:** Arquitectura y funciones de intercepción legal para redes 3G

2.9.2 Cumplimiento de SMS sobre Protocolo IMS

Normas IMS de 3GPP:

- **3GPP TS 23.228:** Subsistema Multimedia IP (IMS) - Arquitectura de etapa 2
- **3GPP TS 24.229:** Protocolo de Control de Llamadas Multimedia IP - Procedimientos SIP y SDP
- **3GPP TS 24.341:** Soporte de SMS sobre redes IP - Especificación de etapa 3
- **3GPP TS 23.204:** Soporte del Servicio de Mensajes Cortos (SMS) sobre acceso IP genérico 3GPP - Etapa 2

- **3GPP TS 29.228:** Interfaces Cx y Dx del Subsistema Multimedia (IM)

Normas de Seguridad IMS:

- **3GPP TS 33.203:** Seguridad 3G; Seguridad de acceso para servicios basados en IP (IMS AKA)
- **3GPP TS 33.210:** Seguridad 3G; Seguridad del Dominio de Red (NDS); Seguridad de la capa de red IP (IPsec)
- **3GPP TS 33.310:** Seguridad del Dominio de Red (NDS); Marco de Autenticación (AF)
- **ETSI TS 133 203:** Seguridad de acceso para servicios basados en IP

Normas del Protocolo SIP:

- **RFC 3261:** SIP: Protocolo de Inicio de Sesión - Especificación central
- **RFC 3428:** Extensión SIP para Mensajería Instantánea - Método MESSAGE
- **RFC 3325:** Extensiones Privadas a SIP para Identidad Afirmada
- **RFC 5765:** Problemas de Seguridad y Soluciones en Sistemas Peer-to-Peer

Intercepción Legal para IMS:

- **ETSI TS 102 232-5:** Intercepción Legal (LI); Especificación de entrega - Parte 5: LI independiente del servicio para servicios del Subsistema Multimedia IP
- **3GPP TS 33.107:** Requisitos y arquitectura de intercepción legal
- **3GPP TS 33.108:** Interfaz de entrega para Intercepción Legal (LI)

2.9.3 Cumplimiento del Protocolo SMPP

Especificación SMPP:

- **SMPP v3.4:** Especificación del Protocolo de Mensajes Cortos Peer-to-Peer - Estándar de la industria
- **SMPP v5.0:** Protocolo SMPP extendido con características mejoradas

Directrices de Seguridad SMPP:

- **TLS sobre SMPP:** Seguridad de la capa de transporte para conexiones SMPP (SMPP sobre TLS)

- **Autenticación de Enlace SMPP:** Autenticación de ID de sistema y contraseña
- **Control de Acceso Basado en IP:** Restricciones a nivel de red para conexiones SMPP

Normas de Interoperabilidad:

- **GSM 03.40 (ETSI TS 100 901):** Realización técnica del SMS Punto a Punto (PP)
- **GSM 03.38 (ETSI TS 100 900):** Alfabetos e información específica de idioma
- **GSM 04.11 (ETSI TS 100 942):** Soporte de SMS Punto a Punto en la interfaz de radio móvil

Cumplimiento de Codificación de Mensajes:

- **ITU-T T.50:** Alfabeto Internacional No. 5 (IA5)
- **ISO/IEC 8859-1:** Codificación de caracteres Latin-1
- **ISO/IEC 10646:** Conjunto de Caracteres Universal (UCS-2/UTF-16)

2.9.4 Cumplimiento de Normas Criptográficas

Seguridad TLS y de Red:

- **NIST SP 800-52:** Directrices para la Selección, Configuración y Uso de Implementaciones TLS
- **NIST SP 800-131A:** Transición del Uso de Algoritmos Criptográficos y Longitudes de Clave
- **RFC 7525:** Recomendaciones para el Uso Seguro de TLS y DTLS
- **RFC 8446:** El Protocolo de Seguridad de Capa de Transporte (TLS) Versión 1.3

Normas de Algoritmos Criptográficos:

- **FIPS 197:** Estándar de Cifrado Avanzado (AES)
- **FIPS 180-4:** Estándar de Hash Seguro (familia SHA-2)
- **NIST SP 800-38D:** Recomendación para Modos de Operación de Cifrado de Bloques: Modo GCM

- **RFC 7539:** ChaCha20 y Poly1305 para Protocolos IETF

Gestión de Claves:

- **NIST SP 800-57:** Recomendación para la Gestión de Claves
- **RFC 5280:** Perfil de Certificado y CRL de Infraestructura de Clave Pública X.509 de Internet

2.10 Resistencia al Criptoanálisis

2.10.1 Principios de Diseño

Defensa Contra el Criptoanálisis:

- No algoritmos criptográficos personalizados/proprietarios
- Solo algoritmos de revisión por pares estándar de la industria
- Actualizaciones de seguridad regulares para bibliotecas criptográficas
- Deprecación de algoritmos débiles
- Uso de cifrado autenticado (GCM, Poly1305)

2.10.2 Seguridad Operativa

Rotación de Claves:

- Procedimientos de renovación de certificados TLS
- Rotación de claves de sesión (por sesión para TLS)
- Políticas de rotación de credenciales de base de datos

Monitoreo y Detección:

- Registro de intentos de autenticación fallidos
 - Monitoreo de expiración de certificados
 - Registro de fallos de apretón de manos TLS
 - Detección de anomalías para fallos de cifrado
 - Alertas de eventos de seguridad
-

3. CONTROL DE INTERCEPCIÓN Y AUTORIZACIÓN

3.1 Control de Acceso para Intercepción Legal

Autorización Administrativa:

- Se requiere acceso de administrador del sistema para la configuración
- Controles de acceso a nivel de base de datos para consultas CDR
- Acceso a API restringido por IP/autenticación
- Registro de auditoría de todos los accesos

Integración del Marco Legal:

- Seguimiento de órdenes de intercepción (integración de sistema externo)
- Listas de autorización de identificadores de objetivo (vistas de base de datos)
- Consultas limitadas en el tiempo (cláusulas WHERE SQL)
- Aplicación automática a través de políticas de acceso

3.2 Retención de Datos y Privacidad

Políticas de Retención:

- Retención de mensajes activos: Configurable (24 horas por defecto en Mnesia)
- Retención de CDR: Configurable (típico de 6 meses a 2 años)
- Archivado automático de Mnesia a SQL
- Purga automática de CDR antiguos (basada en cron)

Protecciones de Privacidad:

- Opción de eliminación del cuerpo del mensaje después de la entrega
- Exclusión del cuerpo del mensaje de la interfaz/exportaciones
- Cifrado de base de datos en reposo
- Registro y monitoreo de acceso

- Principio de mínima recopilación de datos

Configuración:

```
# config/runtime.exs
config :sms_c,
  # Retención de mensajes de Mnesia antes del archivado
  message_retention_hours: 24,

  # Eliminar el cuerpo del mensaje después de la entrega por
  # privacidad
  delete_message_body_after_delivery: false, # Establecer
  verdadero para modo de privacidad

  # Control de escritura de CDR
  cdr_enabled: true,

  # Configuración de archivado por lotes
  batch_insert_batch_size: 100,
  batch_insert_flush_interval_ms: 100
```

☐ Ver **CONFIGURATION.md** para todas las configuraciones de retención

3.3 Interfaces de Entrega para la Aplicación de la Ley

Interfaces Estándar:

1. Acceso a REST API:

- Puntos finales HTTPS para recuperación de mensajes
- Intercambio de datos en formato JSON
- Autenticación y autorización
- Filtrado de consultas por criterios de objetivo

2. Acceso Directo a la Base de Datos:

- Credenciales de solo lectura SQL
- Consultas SQL estándar

- Acceso a la tabla CDR
- Capacidades de búsqueda indexadas

3. Exportación por Lotes:

- Formato de exportación CSV
- Formato de exportación JSON
- Exportaciones basadas en rango de tiempo
- Selección de campos configurable

**Formatos de Entrega:

Referencia de la API SMS-C

[← Volver al Índice de Documentación](#) | [README Principal](#)

Referencia completa para todos los puntos finales de la API REST SMS-C con ejemplos de solicitud/respuesta.

Tabla de Contenidos

- [Descripción General de la API](#)
- [Autenticación](#)
- [Formatos de Respuesta Comunes](#)
- [Punto Final de Estado](#)
- [API de Cola de Mensajes](#)
- [API de PDU SMS en Crudo](#)
- [API de Gestión de Ubicación](#)
- [API de Registro de Frontend](#)
- [API de Registro de Eventos](#)
- [API de Mensajes MMS](#)
- [API de Eventos SS7](#)
- [Códigos de Error](#)
- [Limitación de Tasa](#)
- [Mejores Prácticas](#)

Descripción General de la API

La API REST SMS-C proporciona acceso programático a funciones de envío, enrutamiento y gestión de mensajes.

URL Base

```
https://api.example.com:8443/api
```

Puerto Predeterminado: 8443 (configurable)

Protocolo: HTTPS (TLS requerido en producción)

Tipo de Contenido

Todas las solicitudes y respuestas utilizan JSON:

```
Content-Type: application/json
```

Versionado de la API

La API actual es la versión 1 (implícita). Las versiones futuras utilizarán el versionado de URL:

```
https://api.example.com:8443/api/v2/...
```

Autenticación

Certificados de Cliente TLS (Recomendado)

Las implementaciones en producción deben utilizar autenticación de certificado de cliente TLS:

```
curl --cert client.crt --key client.key \  
https://api.example.com:8443/api/status
```

Autenticación con Clave API

Autenticación personalizada con clave API a través del encabezado `X-API-Key`:

```
curl -H "X-API-Key: your_api_key_here" \  
https://api.example.com:8443/api/status
```

Lista Blanca de IP

Restringir el acceso a la API a direcciones IP de confianza a nivel de firewall.

Formatos de Respuesta Comunes

Respuesta de Éxito

```
{  
  "data": {  
    ...  
  }  
}
```

Respuesta de Error

```
{  
  "errors": {  
    "detail": "Mensaje de error que describe lo que salió mal"  
  }  
}
```

Respuesta de Lista

```
{  
  "data": [  
    {...},  
    {...}  
  ]  
}
```

Punto Final de Estado

Punto de verificación de salud para monitoreo y balanceadores de carga.

Obtener Estado de la API

Solicitud:

```
GET /api/status
```

Respuesta (200 OK):

```
{  
  "status": "ok",  
  "application": "OmniMessage",  
  "timestamp": "2025-10-30T12:34:56Z"  
}
```

Ejemplo:

```
curl https://api.example.com:8443/api/status
```

Casos de Uso:

- Verificaciones de salud del balanceador de carga
- Monitoreo de conectividad del sistema
- Verificación de disponibilidad del servicio

API de Cola de Mensajes

Puntos finales centrales para el envío y gestión de mensajes.

Listar Mensajes

Recuperar mensajes de la cola.

Solicitud:

```
GET /api/messages
```

Encabezados Opcionales:

- `smsc: frontend_name` - Filtrar por SMSC de destino
- `include-unrouted: true|false|1|0` - Incluir mensajes sin registro de ubicación (predeterminado: false)
 - `false` (predeterminado): Solo devolver mensajes con enrutamiento explícito o registro de ubicación
 - `true`: Incluir mensajes sin registro de ubicación (modo compatible hacia atrás)

Parámetros de Consulta:

- `status` - Filtrar por estado: `pending`, `delivered`, `expired`, `dropped`
- `source_smsc` - Filtrar por SMSC de origen
- `dest_smsc` - Filtrar por SMSC de destino
- `limit` - Limitar resultados (predeterminado: 100, máximo: 1000)
- `offset` - Desplazamiento de paginación

Respuesta (200 OK):

```
{
  "data": [
    {
      "id": 12345,
      "source_msisdn": "+15551234567",
      "destination_msisdn": "+447700900000",
      "message_body": "Hello World",
      "source_smsc": "api_client",
      "dest_smsc": "uk_gateway",
      "status": "pending",
      "send_time": "2025-10-30T12:00:00Z",
      "deliver_time": null,
      "delivery_attempts": 0,
      "inserted_at": "2025-10-30T12:00:00Z"
    }
  ]
}
```

Ejemplos:

Obtener mensajes pendientes para un SMSC específico (solo con enrutamiento explícito o ubicación):

```
curl -H "smsc: uk_gateway" \
  https://api.example.com:8443/api/messages
```

Obtener mensajes pendientes incluyendo mensajes no enrutados (compatible hacia atrás):

```
curl -H "smsc: uk_gateway" \
  -H "include-unrouted: true" \
  https://api.example.com:8443/api/messages
```

Obtener todos los mensajes entregados:

```
curl "https://api.example.com:8443/api/messages?
status=delivered&limit=50"
```

Obtener Mensaje Único

Recuperar detalles de un mensaje específico.

Solicitud:

```
GET /api/messages/:id
```

Respuesta (200 OK):

```
{
  "data": {
    "id": 12345,
    "source_msisdn": "+15551234567",
    "destination_msisdn": "+447700900000",
    "message_body": "Hello World",
    "source_smsc": "api_client",
    "dest_smsc": "uk_gateway",
    "source_imsi": null,
    "dest_imsi": null,
    "message_parts": 1,
    "message_part_number": 1,
    "tp_data_coding_scheme": "00",
    "tp_user_data_header": null,
    "status": "pending",
    "send_time": "2025-10-30T12:00:00Z",
    "deliver_time": null,
    "expires": "2025-10-31T12:00:00Z",
    "deadletter": false,
    "delivery_attempts": 0,
    "charge_failed": false,
    "deliver_after": "2025-10-30T12:00:00Z",
    "raw_data_flag": false,
    "raw_sip_flag": false,
    "raw_pdu": null,
    "inserted_at": "2025-10-30T12:00:00Z",
    "updated_at": "2025-10-30T12:00:00Z"
  }
}
```

Ejemplo:

```
curl https://api.example.com:8443/api/messages/12345
```

Enviar Mensaje (Sincrónico)

Enviar un mensaje y recibir el ID del mensaje de inmediato.

Solicitud:

```
POST /api/messages
Content-Type: application/json
```

Cuerpo:

```
{
  "source_msisdn": "+15551234567",
  "destination_msisdn": "+447700900000",
  "message_body": "Hello World",
  "source_smsc": "api_client"
}
```

Campos Opcionales:

- `dest_smsc` - Sobrescribir la decisión de enrutamiento
- `send_time` - Programar para entrega futura (ISO 8601)
- `message_parts` - Total de partes para mensaje multipart
- `message_part_number` - Número de parte (indexado desde 1)
- `tp_data_coding_scheme` - DCS de SMS (predeterminado: "00")
- `source_imsi` - IMSI del suscriptor de origen
- `dest_imsi` - IMSI del suscriptor de destino

Respuesta (201 Creado):

```
{
  "data": {
    "id": 12345,
    "source_msisdn": "+15551234567",
    "destination_msisdn": "+447700900000",
    "message_body": "Hello World",
    "source_smsc": "api_client",
    "dest_smsc": "uk_gateway",
    "status": "pending",
    "send_time": "2025-10-30T12:00:00Z",
    "inserted_at": "2025-10-30T12:00:00Z"
  }
}
```

Ejemplo:

```
curl -X POST https://api.example.com:8443/api/messages \
-H "Content-Type: application/json" \
-d '{
  "source_msisdn": "+15551234567",
  "destination_msisdn": "+447700900000",
  "message_body": "Hello World",
  "source_smsc": "api_client"
}'
```

Rendimiento: ~70 mensajes/segundo, 14ms de tiempo de respuesta promedio

Usar Cuando:

- Necesita el ID del mensaje de inmediato
- Procesando mensajes/segundo
- Requiere confirmación inmediata

Enviar Mensaje (Asincrónico)

Enviar un mensaje con alto rendimiento (procesamiento por lotes).

Solicitud:

```
POST /api/messages/create_async
Content-Type: application/json
```

Cuerpo: Igual que el punto final sincrónico

Respuesta (202 Aceptado):

```
{
  "data": {
    "status": "accepted",
    "message": "Mensaje en cola para procesamiento"
  }
}
```

Ejemplo:

```
curl -X POST
https://api.example.com:8443/api/messages/create_async \
-H "Content-Type: application/json" \
-d '{
  "source_msisdn": "+15551234567",
  "destination_msisdn": "+447700900000",
  "message_body": "Mensaje de notificación masiva",
  "source_smsc": "bulk_api"
}'
```

Rendimiento: ~4,650 mensajes/segundo, 0.22ms de tiempo de respuesta promedio

Latencia: El mensaje aparece en la base de datos dentro de 100ms (configurable)

Usar Cuando:

- Mensajería masiva de alto volumen (> 100 msg/sec)
- No necesita el ID del mensaje en la respuesta de la API
- El rendimiento es más importante que la confirmación instantánea

Actualizar Mensaje

Actualizar parcialmente los campos del mensaje.

Solicitud:

```
PATCH /api/messages/:id
Content-Type: application/json
```

Cuerpo:

```
{
  "dest_smsc": "alternate_gateway",
  "deliver_after": "2025-10-30T14:00:00Z"
}
```

Campos Actualizables:

- `dest_smsc` - Cambiar destino
- `deliver_after` - Retrasar entrega
- `message_body` - Actualizar texto del mensaje
- `status` - Cambiar estado

Respuesta (200 OK):

```
{
  "data": {
    "id": 12345,
    "dest_smsc": "alternate_gateway",
    "deliver_after": "2025-10-30T14:00:00Z",
    ...
  }
}
```

Ejemplo:

```
curl -X PATCH https://api.example.com:8443/api/messages/12345 \  
-H "Content-Type: application/json" \  
-d '{  
  "dest_smsc": "backup_gateway"  
}'
```

Marcar Mensaje Como Entregado

Marcar un mensaje como entregado con éxito.

Solicitud:

```
POST /api/messages/:id/mark_delivered  
Content-Type: application/json
```

Cuerpo:

```
{  
  "dest_smsc": "uk_gateway"  
}
```

Respuesta (200 OK):

```
{  
  "data": {  
    "id": 12345,  
    "status": "delivered",  
    "deliver_time": "2025-10-30T12:05:30Z",  
    "dest_smsc": "uk_gateway",  
    ...  
  }  
}
```

Ejemplo:

```
curl -X POST
https://api.example.com:8443/api/messages/12345/mark_delivered \
-H "Content-Type: application/json" \
-d '{
  "dest_smsc": "uk_gateway"
}'
```

Caso de Uso: Llamado por sistemas frontend después de la entrega exitosa

Incrementar Intento de Entrega

Incrementar el contador de reintentos y aplicar retroceso exponencial.

Solicitud:

```
PUT /api/messages/:id
```

Respuesta (200 OK):

```
{
  "data": {
    "id": 12345,
    "delivery_attempts": 2,
    "deliver_after": "2025-10-30T12:08:00Z",
    ...
  }
}
```

Cálculo de Retroceso:

```
deliver_after = ahora + 2^(delivery_attempts) minutos
```

Ejemplo:

```
curl -X PUT https://api.example.com:8443/api/messages/12345
```

Caso de Uso: Llamado por el frontend después de un fallo de entrega para programar un reintento

Eliminar Mensaje

Eliminar un mensaje de la cola.

Solicitud:

```
DELETE /api/messages/:id
```

Respuesta (204 Sin Contenido)

Ejemplo:

```
curl -X DELETE https://api.example.com:8443/api/messages/12345
```

Advertencia: Eliminar mensajes los elimina permanentemente. Usar con precaución.

API de PDU SMS en Crudo

Enviar mensajes SMS como PDU en crudo (Unidad de Datos de Protocolo) para máxima compatibilidad con sistemas heredados.

Enviar SMS en Crudo (Sincrónico)

Solicitud:

```
POST /api/messages_raw  
Content-Type: application/json
```

Cuerpo:

```
{
  "pdu": "0001000B916407007009F0000004D4F29C0E",
  "source_smsc": "legacy_system"
}
```

Formato de PDU: PDU de SMS TPDU codificado en hexadecimal

Respuesta (201 Creado):

```
{
  "data": {
    "id": 12346,
    "source_msisdn": "+447700900000",
    "destination_msisdn": "+447700900000",
    "message_body": "Test",
    "source_smsc": "legacy_system",
    "raw_pdu": "0001000B916407007009F0000004D4F29C0E",
    ...
  }
}
```

Ejemplo:

```
curl -X POST https://api.example.com:8443/api/messages_raw \
-H "Content-Type: application/json" \
-d '{
  "pdu": "0001000B916407007009F0000004D4F29C0E",
  "source_smsc": "legacy_system"
}'
```

Enviar SMS en Crudo (Asincrónico)

Solicitud:

```
POST /api/messages_raw/async
Content-Type: application/json
```

Cuerpo: Igual que el sincrónico

Respuesta (202 Aceptado):

```
{
  "data": {
    "status": "accepted",
    "message": "PDU en cola para procesamiento"
  }
}
```

Ejemplo:

```
curl -X POST https://api.example.com:8443/api/messages_raw/async \
-H "Content-Type: application/json" \
-d '{
  "pdu": "0001000B916407007009F0000004D4F29C0E",
  "source_smsc": "legacy_gateway"
}'
```

Manejo de PDU

El sistema automáticamente:

1. Decodifica PDU utilizando estándares de SMS (3GPP TS 23.040)
2. Extrae números de teléfono, texto del mensaje, DCS
3. Detecta informes de entrega (CP-ACK, RP-ACK, etc.)
4. Realiza búsqueda de IMSI a MSISDN si es necesario
5. Aplica reglas de enrutamiento
6. Almacena el PDU original para referencia

Detección de Informe de Entrega:

- CP-ACK, CP-ERROR - Reconocimientos del Protocolo de Conexión
- RP-ACK, RP-ERROR, RP-SMMA - Respuestas del Protocolo de Relé
- Los informes de entrega se registran pero no se almacenan como mensajes

API de Gestión de Ubicación

Gestionar la información de ubicación del suscriptor para la entrega de mensajes móviles.

Listar Ubicaciones

Solicitud:

```
GET /api/locations
```

Respuesta (200 OK):

```
{
  "data": [
    {
      "id": 1,
      "msisdn": "+15551234567",
      "imsi": "001001000000001",
      "location": "msc1.region1.example.com",
      "ran_location": "cell_tower_12345",
      "imei": "123456789012345",
      "ims_capable": true,
      "csfb": false,
      "registered": true,
      "expires": "2025-10-30T13:00:00Z",
      "user_agent": "Samsung Galaxy",
      "inserted_at": "2025-10-30T12:00:00Z",
      "updated_at": "2025-10-30T12:00:00Z"
    }
  ]
}
```

Ejemplo:

```
curl https://api.example.com:8443/api/locations
```

Obtener Ubicación

Solicitud:

```
GET /api/locations/:id
```

Respuesta (200 OK):

```
{
  "data": {
    "id": 1,
    "msisdn": "+15551234567",
    "imsi": "001001000000001",
    ...
  }
}
```

Ejemplo:

```
curl https://api.example.com:8443/api/locations/1
```

Crear/Actualizar Ubicación

Crea una nueva ubicación o actualiza una existente según el IMSI (identificador único).

Solicitud:

```
POST /api/locations
Content-Type: application/json
```

Cuerpo:

```
{
  "msisdn": "+15551234567",
  "imsi": "001001000000001",
  "location": "msc1.region1.example.com",
  "ran_location": "cell_tower_12345",
  "imei": "123456789012345",
  "ims_capable": true,
  "csfb": false,
  "registered": true,
  "expires": "2025-10-30T13:00:00Z",
  "user_agent": "Samsung Galaxy"
}
```

Campos Requeridos:

- `imsi` - Identificador único del suscriptor
- `msisdn` - Número de teléfono

Campos Opcionales:

- `location` - Dirección MSC/VLR
- `ran_location` - ID de torre/célula
- `imei` - Identificador del dispositivo
- `ims_capable` - Capacidad de IMS VoLTE
- `csfb` - Indicador de retroceso de circuito
- `registered` - Actualmente registrado
- `expires` - Expiración del registro
- `user_agent` - Modelo/información del dispositivo

Respuesta (201 Creado o 200 OK):

```
{
  "data": {
    "id": 1,
    "msisdn": "+15551234567",
    ...
  }
}
```

Ejemplo:

```
curl -X POST https://api.example.com:8443/api/locations \
-H "Content-Type: application/json" \
-d '{
  "msisdn": "+15551234567",
  "imsi": "001001000000001",
  "location": "msc1.region1.example.com",
  "ims_capable": true,
  "registered": true
}'
```

Caso de Uso: Llamado por sistemas de gestión de movilidad (HSS, MME, etc.) cuando el suscriptor se registra

Actualizar Ubicación

Solicitud:

```
PATCH /api/locations/:id
Content-Type: application/json
```

Cuerpo: Actualización parcial con cualquier campo de ubicación

Respuesta (200 OK):

```
{
  "data": {
    "id": 1,
    ...
  }
}
```

Ejemplo:

```
curl -X PATCH https://api.example.com:8443/api/locations/1 \  
-H "Content-Type: application/json" \  
-d '{  
  "location": "msc2.region2.example.com",  
  "ran_location": "cell_tower_67890"  
}'
```

Eliminar Ubicación

Solicitud:

```
DELETE /api/locations/:id
```

Respuesta (204 Sin Contenido)

Ejemplo:

```
curl -X DELETE https://api.example.com:8443/api/locations/1
```

Caso de Uso: Llamado cuando el suscriptor se desregistra o expira

API de Registro de Frontend

Rastrear y gestionar conexiones de SMSC de frontend.

Listar Todos los Frontends

Solicitud:

```
GET /api/frontends
```

Respuesta (200 OK):

```
{
  "data": [
    {
      "id": 1,
      "frontend_name": "uk_gateway_1",
      "frontend_type": "smpp",
      "ip_address": "10.0.1.50",
      "hostname": "gateway1.uk.example.com",
      "uptime_seconds": 86400,
      "configuration": {
        "max_throughput": 1000,
        "bind_type": "transceiver"
      },
      "status": "active",
      "expires_at": "2025-10-30T12:02:00Z",
      "last_seen_at": "2025-10-30T12:00:30Z",
      "inserted_at": "2025-10-29T12:00:00Z",
      "updated_at": "2025-10-30T12:00:30Z"
    }
  ]
}
```

Ejemplo:

```
curl https://api.example.com:8443/api/frontends
```

Listar Solo Frontends Activos

Solicitud:

```
GET /api/frontends/active
```

Respuesta (200 OK): Mismo formato, solo frontends activos

Ejemplo:

```
curl https://api.example.com:8443/api/frontends/active
```

Caso de Uso: Obtener lista de destinos disponibles para enrutamiento

Obtener Estadísticas de Frontend

Solicitud:

```
GET /api/frontends/stats
```

Respuesta (200 OK):

```
{
  "data": {
    "active_count": 5,
    "expired_count": 2,
    "unique_frontends": 7,
    "total_registrations": 1523
  }
}
```

Ejemplo:

```
curl https://api.example.com:8443/api/frontends/stats
```

Obtener Historial de Frontend

Solicitud:

```
GET /api/frontends/history/:name
```

Respuesta (200 OK):

```
{
  "data": [
    {
      "id": 1,
      "frontend_name": "uk_gateway_1",
      "status": "active",
      "inserted_at": "2025-10-30T12:00:00Z",
      ...
    },
    {
      "id": 2,
      "frontend_name": "uk_gateway_1",
      "status": "expired",
      "inserted_at": "2025-10-29T12:00:00Z",
      ...
    }
  ]
}
```

Ejemplo:

```
curl
https://api.example.com:8443/api/frontends/history/uk_gateway_1
```

Registrar Frontend

Registrar o actualizar la conexión de frontend.

Solicitud:

```
POST /api/frontends/register
Content-Type: application/json
```

Cuerpo:

```
{
  "frontend_name": "uk_gateway_1",
  "frontend_type": "smpp",
  "ip_address": "10.0.1.50",
  "hostname": "gateway1.uk.example.com",
  "uptime_seconds": 86400,
  "configuration": {
    "max_throughput": 1000,
    "bind_type": "transceiver",
    "system_id": "gateway1"
  }
}
```

Campos Requeridos:

- `frontend_name` - Identificador único para el frontend
- `frontend_type` - Tipo: `smpp`, `sip`, `http`, etc.

Campos Opcionales:

- `ip_address` - IP del frontend
- `hostname` - Nombre de host del frontend
- `uptime_seconds` - Tiempo de actividad desde el inicio
- `configuration` - Objeto de configuración personalizada

Respuesta (201 Creado):

```
{
  "data": {
    "id": 1,
    "frontend_name": "uk_gateway_1",
    "status": "active",
    "expires_at": "2025-10-30T12:01:30Z",
    ...
  }
}
```

Ejemplo:

```
curl -X POST https://api.example.com:8443/api/frontends/register \
-H "Content-Type: application/json" \
-d '{
  "frontend_name": "uk_gateway_1",
  "frontend_type": "smpp",
  "ip_address": "10.0.1.50",
  "hostname": "gateway1.uk.example.com"
}'
```

Tiempo de Espera de Registro: 90 segundos (los frontends deben volver a registrarse cada 60-90 segundos)

Caso de Uso: Llamado periódicamente por sistemas frontend para mantener el estado activo

API de Registro de Eventos

Rastrear eventos del ciclo de vida del mensaje.

Obtener Eventos de Mensaje

Solicitud:

```
GET /api/events/:message_id
```

Respuesta (200 OK):

```
{
  "data": [
    {
      "event_epoch": 1698672000,
      "name": "message_inserted",
      "description": "Mensaje insertado en la cola",
      "event_source": "node1@server.example.com"
    },
    {
      "event_epoch": 1698672001,
      "name": "message_routed",
      "description": "Enrutado a uk_gateway a través de route_id=42",
      "event_source": "node1@server.example.com"
    },
    {
      "event_epoch": 1698672005,
      "name": "message_delivered",
      "description": "Entregado con éxito",
      "event_source": "node2@server.example.com"
    }
  ]
}
```

Ejemplo:

```
curl https://api.example.com:8443/api/events/12345
```

Tipos de Eventos:

- `message_inserted` - Mensaje creado
- `message_routed` - Decisión de enrutamiento tomada
- `message_delivered` - Entrega exitosa
- `message_failed` - Fallo en la entrega
- `message_dropped` - Eliminado por ruta
- `auto_reply_sent` - Respuesta automática activada
- `number_translated` - Transformación de número aplicada
- `routing_failed` - No se encontró ruta

- `charging_failed` - Error de cobro

Registrar Evento

Solicitud:

```
POST /api/events  
Content-Type: application/json
```

Cuerpo:

```
{  
  "message_id": 12345,  
  "name": "custom_event",  
  "description": "Descripción del evento personalizado",  
  "event_source": "external_system"  
}
```

Respuesta (201 Creado):

```
{  
  "data": {  
    "message_id": 12345,  
    "name": "custom_event",  
    "description": "Descripción del evento personalizado",  
    "event_source": "external_system",  
    "event_epoch": 1698672010  
  }  
}
```

Ejemplo:

```
curl -X POST https://api.example.com:8443/api/events \  
-H "Content-Type: application/json" \  
-d '{  
  "message_id": 12345,  
  "name": "external_delivery_confirmed",  
  "description": "Confirmado por el sistema downstream"  
}'
```

Retención de Eventos: 7 días (configurable)

API de Mensajes MMS

Gestionar mensajes del Servicio de Mensajería Multimedia (MMS).

Listar Mensajes MMS

Solicitud:

```
GET /api/mms_messages
```

Respuesta (200 OK): Similar a los mensajes SMS con campos adicionales de MMS

Crear Mensaje MMS

Solicitud:

```
POST /api/mms_messages  
Content-Type: application/json
```

Cuerpo:

```
{
  "source_msisdn": "+15551234567",
  "destination_msisdn": "+447700900000",
  "subject": "Foto",
  "content_type": "image/jpeg",
  "content_location": "https://cdn.example.com/media/12345.jpg",
  "message_size": 524288
}
```

Respuesta (201 Creado): Objeto completo del mensaje MMS

API de Eventos SS7

Rastrear eventos de señalización SS7.

Listar Eventos SS7

Solicitud:

```
GET /api/ss7_events
```

Respuesta (200 OK):

```
{
  "data": [
    {
      "id": 1,
      "event_type": "MAP_UPDATE_LOCATION",
      "imsi": "001001000000001",
      "msisdn": "+15551234567",
      "timestamp": "2025-10-30T12:00:00Z",
      ...
    }
  ]
}
```

Crear Evento SS7

Solicitud:

```
POST /api/ss7_events  
Content-Type: application/json
```

Cuerpo:

```
{  
  "event_type": "MAP_UPDATE_LOCATION",  
  "imsi": "001001000000001",  
  "msisdn": "+15551234567"  
}
```

Respuesta (201 Creado): Objeto completo del evento

Códigos de Error

Códigos de Estado HTTP

Código	Significado	Descripción
200	OK	Solicitud exitosa
201	Creado	Recurso creado con éxito
202	Aceptado	Solicitud aceptada para procesamiento
204	Sin Contenido	Eliminación exitosa
400	Solicitud Incorrecta	Formato de solicitud inválido
401	No Autorizado	Se requiere autenticación
403	Prohibido	Permisos insuficientes
404	No Encontrado	El recurso no existe
422	Entidad No Procesable	Errores de validación
429	Demasiadas Solicitudes	Límite de tasa excedido
500	Error Interno del Servidor	Error del servidor
503	Servicio No Disponible	Temporalmente no disponible

Formato de Respuesta de Error

```
{
  "errors": {
    "detail": "La validación falló: destination_msisdn es
requerido"
  }
}
```

Mensajes de Error Comunes

Error	Causa	Solución
"destination_msisdn es requerido"	Campo requerido faltante	Incluir destination_msisdn en la solicitud
"Formato de número de teléfono inválido"	Número mal formado	Usar formato E.164: +15551234567
"Mensaje demasiado largo"	Excede el límite de tamaño	Dividir en múltiples partes
"No se encontró ruta"	Fallo en el enrutamiento	Verificar la configuración de enrutamiento
"Fallo en el cobro"	Error de OCS	Verificar la conectividad del sistema de cobro
"Mensaje no encontrado"	ID de mensaje inválido	Verificar que el ID exista
"Frontend no registrado"	SMSC desconocido	Registrar el frontend primero

Limitación de Tasa

Límites Predeterminados

Punto Final	Límite	Ventana
POST /api/messages	100 req/sec	Por IP
POST /api/messages/create_async	1000 req/sec	Por IP
POST /api/messages_raw	100 req/sec	Por IP
GET /api/*	1000 req/sec	Por IP

Encabezados de Límite de Tasa

```
X-RateLimit-Limit: 100  
X-RateLimit-Remaining: 95  
X-RateLimit-Reset: 1698672060
```

Límite de Tasa Excedido

Respuesta (429 Demasiadas Solicitudes):

```
{  
  "errors": {  
    "detail": "Límite de tasa excedido. Reintentar después de 5 segundos."  
  }  
}
```

Mejores Prácticas

Envío de Mensajes

1. **Usar Async para Masivo:** Usar `/create_async` para > 100 msg/sec
2. **Incluir source_smsc:** Siempre identificar su sistema
3. **Validar Números:** Usar formato E.164 (+código de país)
4. **Manejar Errores:** Implementar lógica de reintento para errores 5xx
5. **Verificar Enrutamiento:** Probar rutas antes del envío masivo

Integración de Frontend

1. **Registrar Regularmente:** Volver a registrar cada 60 segundos
2. **Consultar Mensajes:** Consultar con encabezado `smsc` para sus mensajes
3. **Usar include-unrouted con Sabiduría:** Por defecto, solo se devuelven mensajes con enrutamiento explícito o registro de ubicación. Establecer `include-unrouted: true` solo si necesita comportamiento compatible hacia atrás para recibir todos los mensajes no enrutados
4. **Marcar Como Entregado:** Siempre llamar a `mark_delivered` después del éxito
5. **Incrementar en Fallo:** Usar el punto final PUT para lógica de reintento
6. **Monitorear Eventos:** Verificar el registro de eventos para problemas de entrega

Rendimiento

1. **Agrupamiento de Conexiones:** Reutilizar conexiones HTTP
2. **Solicitudes por Lotes:** Agrupar múltiples mensajes por solicitud
3. **Procesamiento Paralelo:** Realizar llamadas API concurrentes
4. **Monitorear Métricas:** Observar Prometheus para cuellos de botella
5. **Establecer Tiempos de Espera:** Usar un tiempo de espera de 30 segundos para llamadas API

Seguridad

1. **Usar TLS:** Siempre usar HTTPS en producción
2. **Validar Certificados:** No omitir la validación de certificados
3. **Rotar Claves API:** Cambiar claves regularmente
4. **Lista Blanca de IP:** Restringir a fuentes conocidas
5. **Registrar Actividad API:** Monitorear patrones sospechosos

Manejo de Errores

1. **Reintentar Errores 5xx:** Los errores del servidor suelen ser temporales
2. **No Reintentar 4xx:** Los errores del cliente necesitan correcciones de código
3. **Retroceso Exponencial:** Esperar más entre reintentos
4. **Interruptor de Circuito:** Detenerse después de fallos repetidos
5. **Alertar sobre Patrones:** Monitorear tasas de error

Ejemplo de Integración (Python)

```
import requests
import time

class SMSCClient:
    def __init__(self, base_url, api_key=None):
        self.base_url = base_url
        self.session = requests.Session()
        if api_key:
            self.session.headers.update({"X-API-Key": api_key})

    def submit_message(self, from_num, to_num, text,
async_mode=False):
        endpoint = "/messages/create_async" if async_mode else
"/messages"
        url = f"{self.base_url}{endpoint}"

        payload = {
            "source_msisdn": from_num,
            "destination_msisdn": to_num,
            "message_body": text,
            "source_smsc": "python_client"
        }

        try:
            response = self.session.post(url, json=payload,
timeout=30)
            response.raise_for_status()
            return response.json()["data"]
        except requests.exceptions.RequestException as e:
            print(f"Error API: {e}")
            return None

    def get_pending_messages(self, smsc_name,
include_unrouted=False):
        url = f"{self.base_url}/messages"
        headers = {"smsc": smsc_name}

        # Incluir mensajes no enrutados si se solicita (modo
compatible hacia atrás)
        if include_unrouted:
            headers["include-unrouted"] = "true"
```

```

        try:
            response = self.session.get(url, headers=headers,
timeout=30)
            response.raise_for_status()
            return response.json()["data"]
        except requests.exceptions.RequestException as e:
            print(f"Error API: {e}")
            return []

    def mark_delivered(self, message_id, smsc_name):
        url = f"
{self.base_url}/messages/{message_id}/mark_delivered"
        payload = {"dest_smsc": smsc_name}

        try:
            response = self.session.post(url, json=payload,
timeout=30)
            response.raise_for_status()
            return True
        except requests.exceptions.RequestException as e:
            print(f"Error API: {e}")
            return False

# Uso
client = SMSCClient("https://api.example.com:8443/api",
api_key="your_key")

# Enviar mensaje único
result = client.submit_message("+15551234567", "+447700900000",
"Hello")
print(f"ID del Mensaje: {result['id']}")

# Enviar mensajes masivos (asincrónicos)
for i in range(1000):
    client.submit_message("+15551234567", f"+44770090{i:04d}",
f"Bulk {i}", async_mode=True)

# Bucle de sondeo de frontend
while True:
    # Obtener mensajes con enrutamiento explícito o registro de
ubicación
    messages = client.get_pending_messages("my_gateway")

```

```
# 0 usar include_unrouted=True para comportamiento compatible
hacia atrás
# messages = client.get_pending_messages("my_gateway",
include_unrouted=True)

for msg in messages:
    # Entregar mensaje a través de su protocolo
    success = deliver_via_smpp(msg)

    if success:
        client.mark_delivered(msg["id"], "my_gateway")
    else:
        # Incrementar para reintento
        requests.put(f"
{client.base_url}/messages/{msg['id']}")

time.sleep(5) # Sondear cada 5 segundos
```

Registro de Cambios de la API

Versión 1 (Actual)

- Lanzamiento inicial
- CRUD de cola de mensajes
- Envío de PDU en crudo
- Gestión de ubicación
- Registro de frontend
- Registro de eventos

Características Planeadas

- Envío de mensajes por lotes (una sola solicitud, múltiples mensajes)
- Plantillas de mensajes
- API de entrega programada
- Webhooks en tiempo real para eventos
- Punto final de API GraphQL

- Autenticación OAuth2

Para preguntas o problemas con la API, consulte la [Guía de Solución de Problemas](#) o contacte con soporte.

Referencia del Esquema CDR (Registro de Detalle de Llamadas)

[← Volver al Índice de Documentación](#) | [README Principal](#)

Referencia completa para la tabla de base de datos CDR utilizada para el almacenamiento a largo plazo de mensajes, facturación y análisis.

Tabla de Contenidos

- [Descripción General](#)
- [Esquema de la Tabla](#)
- [Descripciones de Campos](#)
- [Ejemplos de SQL](#)
- [Índices](#)
- [Tipos de Datos por Base de Datos](#)
- [Consideraciones de Privacidad](#)
- [Retención y Archivado](#)
- [Integración de Facturación](#)

Descripción General

La tabla `cdrs` almacena Registros de Detalle de Llamadas para todos los mensajes SMS procesados por el sistema. Los CDR se escriben cuando:

- Los mensajes se entregan con éxito
- Los mensajes expiran sin entrega

- Los mensajes fallan permanentemente
- Los mensajes son rechazados

Los CDR proporcionan almacenamiento a largo plazo separado de la base de datos operativa Mnesia, permitiendo:

- Facturación y facturación
- Análisis e informes
- Cumplimiento y auditoría
- Historial de mensajes más allá del período de retención de Mnesia

Esquema de la Tabla

MySQL / MariaDB

```
CREATE TABLE cdrs (  
  id BIGINT AUTO_INCREMENT PRIMARY KEY,  
  
  -- Identificación del mensaje  
  message_id BIGINT NOT NULL,  
  
  -- Números de teléfono  
  calling_number VARCHAR(255) NOT NULL,  
  called_number VARCHAR(255) NOT NULL,  
  
  -- Enrutamiento SMSC  
  source_smsc VARCHAR(255),  
  dest_smsc VARCHAR(255),  
  
  -- Información del nodo (para implementaciones en clúster)  
  origin_node VARCHAR(255),  
  destination_node VARCHAR(255),  
  
  -- Tiempos  
  submission_time DATETIME NOT NULL,  
  delivery_time DATETIME,  
  expiry_time DATETIME,  
  
  -- Estado y metadatos  
  status VARCHAR(50) NOT NULL,  
  delivery_attempts INT DEFAULT 0,  
  message_parts INT,  
  deadletter BOOLEAN DEFAULT FALSE,  
  
  -- Cuerpo del mensaje opcional (controles de privacidad)  
  message_body TEXT,  
  
  -- Tiempos de auditoría  
  inserted_at DATETIME NOT NULL,  
  updated_at DATETIME NOT NULL,  
  
  -- Índices  
  INDEX idx_cdrs_message_id (message_id),
```

```
INDEX idx_cdrs_calling_number (calling_number),
INDEX idx_cdrs_called_number (called_number),
INDEX idx_cdrs_status (status),
INDEX idx_cdrs_submission_time (submission_time),
INDEX idx_cdrs_dest_smsc (dest_smsc)
);
```

PostgreSQL

```
CREATE TABLE cdrs (  
  id BIGSERIAL PRIMARY KEY,  
  
  -- Identificación del mensaje  
  message_id BIGINT NOT NULL,  
  
  -- Números de teléfono  
  calling_number VARCHAR(255) NOT NULL,  
  called_number VARCHAR(255) NOT NULL,  
  
  -- Enrutamiento SMSC  
  source_smsc VARCHAR(255),  
  dest_smsc VARCHAR(255),  
  
  -- Información del nodo (para implementaciones en clúster)  
  origin_node VARCHAR(255),  
  destination_node VARCHAR(255),  
  
  -- Tiempos  
  submission_time TIMESTAMP NOT NULL,  
  delivery_time TIMESTAMP,  
  expiry_time TIMESTAMP,  
  
  -- Estado y metadatos  
  status VARCHAR(50) NOT NULL,  
  delivery_attempts INTEGER DEFAULT 0,  
  message_parts INTEGER,  
  deadletter BOOLEAN DEFAULT FALSE,  
  
  -- Cuerpo del mensaje opcional (controles de privacidad)  
  message_body TEXT,  
  
  -- Tiempos de auditoría  
  inserted_at TIMESTAMP NOT NULL,  
  updated_at TIMESTAMP NOT NULL  
);  
  
-- Índices  
CREATE INDEX idx_cdrs_message_id ON cdrs(message_id);  
CREATE INDEX idx_cdrs_calling_number ON cdrs(calling_number);  
CREATE INDEX idx_cdrs_called_number ON cdrs(called_number);
```

```
CREATE INDEX idx_cdrs_status ON cdrs(status);
CREATE INDEX idx_cdrs_submission_time ON cdrs(submission_time);
CREATE INDEX idx_cdrs_dest_smsc ON cdrs(dest_smsc);
```

Descripciones de Campos

Clave Primaria

Campo	Tipo	Nulo	Descripción
<code>id</code>	BIGINT	NO	Clave primaria autoincremental para el registro CDR

Identificación del Mensaje

Campo	Tipo	Nulo	Descripción
<code>message_id</code>	BIGINT	NO	Identificador único del mensaje de la cola de mensajes SMS-C. Hace referencia al ID del mensaje original en Mnesia.

Números de Teléfono

Campo	Tipo	Nulo	Descripción
<code>calling_number</code>	VARCHAR(255)	NO	MSISDN de origen (número móvil) del remitente del mensaje. Típicamente en formato E.164 (por ejemplo, +15551234567).
<code>called_number</code>	VARCHAR(255)	NO	MSISDN de destino (número móvil) del destinatario del mensaje. Típicamente en formato E.164 (por ejemplo, +15551234567).

Enrutamiento SMSC

Campo	Tipo	Nulo	Descripción
<code>source_smsc</code>	VARCHAR(255)	SÍ	Nombre o identificador del SMSC de origen que envió el mensaje. NULL si se envió a través de API u otra interfaz que no sea SMSC.
<code>dest_smsc</code>	VARCHAR(255)	SÍ	Nombre o identificador del SMSC de destino que entregó (o intentó entregar) el mensaje. NULL si el mensaje nunca fue enrutado.

Información del Nodo

Para implementaciones en clúster, rastrea qué nodos manejaron el mensaje:

Campo	Tipo	Nulo	Descripción
<code>origin_node</code>	VARCHAR(255)	Sí	Nombre del nodo Erlang donde se recibió originalmente el mensaje (por ejemplo, " <code>sms@node1.example.com</code> "). Útil para la resolución de problemas y el análisis de distribución de carga.
<code>destination_node</code>	VARCHAR(255)	Sí	Nombre del nodo Erlang desde donde se entregó el mensaje (si es diferente del origen). NULL para implementaciones de un solo nodo o si el mensaje nunca fue entregado.

Tiempos

Todos los tiempos se almacenan en UTC:

Campo	Tipo	Nulo	Descripción
submission_time	DATETIME	NO	Cuando el mensaje fue enviado por primera vez al SMS-C. Se utiliza como hora de inicio para los cálculos de facturación.
delivery_time	DATETIME	SÍ	Cuando el mensaje fue entregado con éxito. NULL si el mensaje expiró, falló o fue rechazado.
expiry_time	DATETIME	SÍ	Cuando el mensaje expiró (se volvió indeliverable). NULL si el mensaje fue entregado o aún está pendiente.

Cálculo de la Duración de Entrega:

```

TIMESTAMPDIFF(SECOND, submission_time, delivery_time) AS
delivery_duration_seconds

```

Estado y Metadatos

Campo	Tipo	Nulo	Descripción
<code>status</code>	VARCHAR(50)	NO	Estado final del mensaje. Valores válidos: <code>delivered</code> , <code>expired</code> , <code>failed</code> , <code>rejected</code>
<code>delivery_attempts</code>	INT	NO	Número de intentos de entrega realizados antes del estado final. Predeterminado: 0. Rango: 0-255 típicamente.
<code>message_parts</code>	INT	SÍ	Número de segmentos SMS para mensajes concatenados. 1 para mensajes de una sola parte, 2+ para múltiples partes. NULL si se desconoce.
<code>deadletter</code>	BOOLEAN	NO	Si el mensaje fue movido a la cola de cartas muertas. TRUE indica que el mensaje no pudo ser entregado y se agotaron todos los reintentos. Predeterminado: FALSE

Valores de Estado:

Estado	Descripción	Facturable	Tiempo de Entrega
<code>delivered</code>	Entregado con éxito al destinatario	Sí	Establecido
<code>expired</code>	Excedió el período de validez sin entrega	Depende de la política de facturación	NULL
<code>failed</code>	Fallo de entrega permanente (número inválido, etc.)	Depende de la política de facturación	NULL
<code>rejected</code>	Rechazado por reglas de enrutamiento o validación	No	NULL

Cuerpo del Mensaje

Campo	Tipo	Nulo	Descripción
<code>message_body</code>	TEXT	Sí	El contenido real del mensaje SMS. Puede ser NULL si <code>delete_message_body_after_delivery</code> está habilitado por razones de privacidad. La longitud máxima varía según la base de datos (típicamente 65,535 caracteres para el tipo TEXT).

Modos de Privacidad:

- **Retención completa:** Cuerpo del mensaje almacenado en CDR para cumplimiento/archivado
- **Modo de privacidad:** Cuerpo del mensaje establecido en NULL cuando `delete_message_body_after_delivery: true`

- **Modo de cumplimiento:** Cuerpo almacenado encriptado o hash (requiere implementación personalizada)

Tiempos de Auditoría

Campo	Tipo	Nulo	Descripción
inserted_at	DATETIME	NO	Cuando el registro CDR fue insertado por primera vez en la base de datos. Típicamente el mismo que o poco después de delivery_time/expiry_time.
updated_at	DATETIME	NO	Cuando el registro CDR fue actualizado por última vez. Igual a inserted_at si nunca fue actualizado.

Ejemplos de SQL

Consultas Básicas

Encontrar todos los CDR para un número de teléfono específico:

```
SELECT * FROM cdrs
WHERE calling_number = '+15551234567'
      OR called_number = '+15551234567'
ORDER BY submission_time DESC
LIMIT 100;
```

Contar mensajes por estado:

```
SELECT status, COUNT(*) as count
FROM cdrs
GROUP BY status;
```

Tiempo promedio de entrega para mensajes entregados:

```
SELECT AVG(TIMESTAMPDIFF(SECOND, submission_time, delivery_time))
AS avg_delivery_seconds
FROM cdrs
WHERE status = 'delivered'
AND delivery_time IS NOT NULL;
```

Consultas de Facturación

Volumen diario de mensajes por SMSC de destino:

```
SELECT
  DATE(submission_time) AS date,
  dest_smsc,
  COUNT(*) AS message_count,
  SUM(CASE WHEN status = 'delivered' THEN 1 ELSE 0 END) AS
delivered_count,
  SUM(message_parts) AS total_segments
FROM cdrs
WHERE submission_time >= DATE_SUB(NOW(), INTERVAL 30 DAY)
GROUP BY DATE(submission_time), dest_smsc
ORDER BY date DESC, message_count DESC;
```

Mensajes facturables para un cliente (por prefijo de número de llamada):

```
SELECT
  DATE(submission_time) AS date,
  COUNT(*) AS message_count,
  SUM(message_parts) AS total_segments,
  SUM(message_parts) * 0.01 AS total_cost
FROM cdrs
WHERE calling_number LIKE '+1555%'
AND status = 'delivered'
AND submission_time >= '2025-10-01'
AND submission_time < '2025-11-01'
GROUP BY DATE(submission_time);
```

Análisis de rendimiento de rutas:

```

SELECT
  dest_smsc,
  COUNT(*) AS total_messages,
  SUM(CASE WHEN status = 'delivered' THEN 1 ELSE 0 END) AS
delivered,
  ROUND(100.0 * SUM(CASE WHEN status = 'delivered' THEN 1 ELSE 0
END) / COUNT(*), 2) AS delivery_rate_pct,
  AVG(delivery_attempts) AS avg_attempts,
  AVG(TIMESTAMPDIFF(SECOND, submission_time, delivery_time)) AS
avg_delivery_seconds
FROM cdrs
WHERE submission_time >= DATE_SUB(NOW(), INTERVAL 7 DAY)
  AND dest_smsc IS NOT NULL
GROUP BY dest_smsc
ORDER BY delivery_rate_pct DESC;

```

Consultas de Análisis

Mensajes por hora del día (patrón de tráfico):

```

SELECT
  HOUR(submission_time) AS hour,
  COUNT(*) AS message_count
FROM cdrs
WHERE submission_time >= DATE_SUB(NOW(), INTERVAL 7 DAY)
GROUP BY HOUR(submission_time)
ORDER BY hour;

```

Análisis de mensajes de múltiples partes:

```

SELECT
  message_parts,
  COUNT(*) AS message_count,
  AVG(TIMESTAMPDIFF(SECOND, submission_time, delivery_time)) AS
avg_delivery_seconds
FROM cdrs
WHERE message_parts IS NOT NULL
  AND status = 'delivered'
GROUP BY message_parts
ORDER BY message_parts;

```

Análisis de mensajes fallidos:

```

SELECT
  called_number,
  COUNT(*) AS failure_count,
  AVG(delivery_attempts) AS avg_attempts,
  MAX(submission_time) AS last_failure
FROM cdrs
WHERE status IN ('failed', 'expired')
  AND submission_time >= DATE_SUB(NOW(), INTERVAL 7 DAY)
GROUP BY called_number
HAVING failure_count >= 5
ORDER BY failure_count DESC;

```

Consultas de Cumplimiento y Auditoría

Encontrar todos los mensajes entre dos partes en un rango de tiempo:

```

SELECT
  submission_time,
  calling_number,
  called_number,
  status,
  message_body,
  delivery_time
FROM cdrs
WHERE (
  (calling_number = '+15551234567' AND called_number =
'+15559876543')
  OR
  (calling_number = '+15559876543' AND called_number =
'+15551234567')
)
AND submission_time >= '2025-10-01'
AND submission_time < '2025-11-01'
ORDER BY submission_time;

```

Aplicación de política de retención (eliminar CDR antiguos):

```

-- Encontrar registros más antiguos que el período de retención
(ejemplo: 2 años)
SELECT COUNT(*) FROM cdrs
WHERE submission_time < DATE_SUB(NOW(), INTERVAL 2 YEAR);

-- Eliminar registros antiguos (usar con precaución!)
DELETE FROM cdrs
WHERE submission_time < DATE_SUB(NOW(), INTERVAL 2 YEAR)
LIMIT 10000; -- Eliminación por lotes para evitar bloqueos

```

Análisis de Clúster

Distribución de mensajes entre nodos:

```

SELECT
  origin_node,
  COUNT(*) AS message_count,
  SUM(CASE WHEN status = 'delivered' THEN 1 ELSE 0 END) AS
delivered_count
FROM cdrs
WHERE submission_time >= DATE_SUB(NOW(), INTERVAL 1 DAY)
GROUP BY origin_node;

```

Índices

Los siguientes índices se crean para optimizar consultas comunes:

Nombre del Índice	Columnas	Propósito
PRIMARY	id	Clave primaria, asegura registro único
idx_cdrs_message_id	message_id	Buscar CDR por ID de mensaje original
idx_cdrs_calling_number	calling_number	Encontrar mensajes de un remitente específico
idx_cdrs_called_number	called_number	Encontrar mensajes a un destinatario específico
idx_cdrs_status	status	Filtrar por estado de entrega
idx_cdrs_submission_time	submission_time	Consultas basadas en tiempo, períodos de facturación
idx_cdrs_dest_smsc	dest_smsc	Análisis de rendimiento de rutas

Recomendaciones de Índices Adicionales

Para implementaciones de alto volumen, considere estos índices adicionales:

Índice compuesto para consultas de facturación:

```
CREATE INDEX idx_cdrs_billing ON cdrs(calling_number,  
submission_time, status);
```

Índice compuesto para análisis de rutas:

```
CREATE INDEX idx_cdrs_route_perf ON cdrs(dest_smsc,  
submission_time, status);
```

Índice compuesto para búsquedas de cumplimiento:

```
CREATE INDEX idx_cdrs_party_time ON cdrs(calling_number,  
called_number, submission_time);
```

Índice de texto completo para búsquedas en el cuerpo del mensaje (MySQL):

```
ALTER TABLE cdrs ADD FULLTEXT INDEX idx_cdrs_message_body_ft  
(message_body);  
  
-- Uso:  
SELECT * FROM cdrs  
WHERE MATCH(message_body) AGAINST('keyword' IN NATURAL LANGUAGE  
MODE);
```

Tipos de Datos por Base de Datos

Mapeo de tipos de campo en las bases de datos soportadas:

Campo	MySQL/MariaDB	PostgreSQL	Notas
<code>id</code>	BIGINT AUTO_INCREMENT	BIGSERIAL	Entero de 64 bits, autoincremental
<code>message_id</code>	BIGINT	BIGINT	Entero de 64 bits
Campos de cadena	VARCHAR(255)	VARCHAR(255)	Cadena de longitud variable, máx 255 caracteres
<code>message_body</code>	TEXT	TEXT	Texto grande, hasta 65,535 bytes (MySQL), ilimitado (PostgreSQL)
Tiempos	DATETIME	TIMESTAMP	Se recomiendan marcas de tiempo UTC
Enteros	INT	INTEGER	Entero con signo de 32 bits
Booleanos	BOOLEAN (TINYINT(1))	BOOLEAN	MySQL lo almacena como 0/1

Consideraciones de Privacidad

La tabla CDR puede contener información personal sensible (números de teléfono, contenido del mensaje). Considere estas medidas de privacidad:

1. Privacidad del Cuerpo del Mensaje

Opciones de configuración en `config/runtime.exs`:

```
config :sms_c,  
  # Eliminar el cuerpo del mensaje después de la entrega exitosa  
  delete_message_body_after_delivery: true,  
  
  # Ocultar el cuerpo del mensaje en la interfaz de usuario  
  hide_message_body_in_ui: true,  
  
  # Ocultar el cuerpo del mensaje en las exportaciones  
  hide_message_body_in_export: true
```

2. Enmascaramiento de Números de Teléfono

Para análisis que no requieren números completos:

```
-- Enmascarar los últimos 4 dígitos de los números de teléfono  
SELECT  
  CONCAT(SUBSTRING(calling_number, 1, LENGTH(calling_number) - 4),  
  'XXXX') AS masked_calling,  
  CONCAT(SUBSTRING(called_number, 1, LENGTH(called_number) - 4),  
  'XXXX') AS masked_called,  
  COUNT(*) AS message_count  
FROM cdrs  
GROUP BY masked_calling, masked_called;
```

3. Encriptación de Base de Datos

Habilitar encriptación en reposo para el servidor de base de datos:

MySQL:

```
-- Habilitar encriptación de tabla  
ALTER TABLE cdrs ENCRYPTION='Y';
```

PostgreSQL: Utilizar encriptación de datos transparente (TDE) de PostgreSQL o encriptación a nivel de sistema de archivos.

4. Controles de Acceso

Restringir el acceso a la tabla CDR:

```
-- Crear usuario de facturación de solo lectura
CREATE USER 'billing_ro'@'%' IDENTIFIED BY 'secure_password';
GRANT SELECT ON sms_c.cdcs TO 'billing_ro'@'%';

-- Crear usuario de análisis limitado (sin acceso al cuerpo del
mensaje)
CREATE USER 'analytics'@'%' IDENTIFIED BY 'secure_password';
GRANT SELECT (id, message_id, calling_number, called_number,
source_smsc,
                dest_smsc, submission_time, delivery_time, status,
                delivery_attempts, message_parts)
ON sms_c.cdcs TO 'analytics'@'%';
```

Retención y Archivado

Políticas de Retención

Definir períodos de retención basados en requisitos regulatorios y comerciales:

Industria	Retención Típica	Base Regulatoria
Telecomunicaciones (EE.UU.)	18-24 meses	FCC, leyes estatales
Telecomunicaciones (UE)	6 meses - 2 años	GDPR, ePrivacy
Financiera	5-7 años	SOX, SEC
Salud	6 años	HIPAA

Estrategia de Archivado

1. Partición por Fecha (MySQL 8.0+, PostgreSQL 11+)

```
-- Particionamiento de MySQL por mes
ALTER TABLE cdrs PARTITION BY RANGE (TO_DAYS(submission_time)) (
  PARTITION p202510 VALUES LESS THAN (TO_DAYS('2025-11-01')),
  PARTITION p202511 VALUES LESS THAN (TO_DAYS('2025-12-01')),
  PARTITION p202512 VALUES LESS THAN (TO_DAYS('2026-01-01')),
  PARTITION p_future VALUES LESS THAN MAXVALUE
);

-- Eliminar partición antigua (archivamiento rápido)
ALTER TABLE cdrs DROP PARTITION p202510;
```

2. Archivar a Almacenamiento Frío

```
-- Exportar CDR antiguos a tabla de archivo
CREATE TABLE cdrs_archive LIKE cdrs;

INSERT INTO cdrs_archive
SELECT * FROM cdrs
WHERE submission_time < DATE_SUB(NOW(), INTERVAL 2 YEAR);

-- Verificar y eliminar de la tabla principal
DELETE FROM cdrs
WHERE submission_time < DATE_SUB(NOW(), INTERVAL 2 YEAR);
```

3. Script de Limpieza Automatizado

```
#!/bin/bash
# cleanup_old cdrs.sh - Ejecutar a través de cron

MYSQL_USER="cleanup_user"
MYSQL_PASS="secure_password"
MYSQL_DB="sms_c"
RETENTION_DAYS=730 # 2 años

# Archivar registros antiguos
mysql -u"$MYSQL_USER" -p"$MYSQL_PASS" "$MYSQL_DB" <<EOF
INSERT INTO cdrs_archive
SELECT * FROM cdrs
WHERE submission_time < DATE_SUB(NOW(), INTERVAL $RETENTION_DAYS
DAY)
LIMIT 100000;

DELETE FROM cdrs
WHERE submission_time < DATE_SUB(NOW(), INTERVAL $RETENTION_DAYS
DAY)
LIMIT 100000;
EOF
```

Entrada de Cron:

```
# Ejecutar diariamente a las 2 AM
0 2 * * * /usr/local/bin/cleanup_old_cdrs.sh >>
/var/log/sms_c/cleanup.log 2>&1
```

Integración de Facturación

Esquema de Tarifa

Crear una tabla de tarifas separada para la facturación:

```

CREATE TABLE billing_rates (
  id INT AUTO_INCREMENT PRIMARY KEY,
  destination_prefix VARCHAR(20) NOT NULL,
  description VARCHAR(255),
  rate_per_message DECIMAL(10, 6) NOT NULL,
  rate_per_segment DECIMAL(10, 6) NOT NULL,
  currency VARCHAR(3) DEFAULT 'USD',
  effective_date DATE NOT NULL,
  expiry_date DATE,
  INDEX idx_prefix (destination_prefix),
  INDEX idx_dates (effective_date, expiry_date)
);

-- Tarifas de ejemplo
INSERT INTO billing_rates (destination_prefix, description,
rate_per_message, rate_per_segment, effective_date) VALUES
('+1', 'Estados Unidos/Canadá', 0.0050, 0.0050, '2025-01-01'),
('+44', 'Reino Unido', 0.0080, 0.0080, '2025-01-01'),
('+61', 'Australia', 0.0100, 0.0100, '2025-01-01'),
('+', 'Predeterminado internacional', 0.0150, 0.0150, '2025-01-01');

```

Consulta de Facturación

Unir CDR con tarifas para la facturación:

```

SELECT
  DATE(c.submission_time) AS date,
  c.dest_smsc AS route,
  LEFT(c.called_number,
    CASE
      WHEN c.called_number LIKE '+1%' THEN 2
      WHEN c.called_number LIKE '+%' THEN
LENGTH(SUBSTRING_INDEX(c.called_number, '', 4))
      ELSE 0
    END
  ) AS destination_prefix,
  COUNT(*) AS message_count,
  SUM(c.message_parts) AS segment_count,
  COALESCE(r.rate_per_segment, 0.015) AS rate,
  SUM(c.message_parts) * COALESCE(r.rate_per_segment, 0.015) AS
total_cost
FROM cdrs c
LEFT JOIN billing_rates r ON c.called_number LIKE
CONCAT(r.destination_prefix, '%')
  AND c.submission_time >= r.effective_date
  AND (r.expiry_date IS NULL OR c.submission_time < r.expiry_date)
WHERE c.status = 'delivered'
  AND c.submission_time >= '2025-10-01'
  AND c.submission_time < '2025-11-01'
GROUP BY date, route, destination_prefix
ORDER BY date DESC, total_cost DESC;

```

Exportar para Sistemas de Facturación

Exportación CSV:

```
mysql -u billing_ro -p -D sms_c -e "  
SELECT  
    id,  
    message_id,  
    calling_number,  
    called_number,  
    dest_smsc,  
    submission_time,  
    delivery_time,  
    status,  
    message_parts  
FROM cdrs  
WHERE submission_time >= '2025-10-01'  
    AND submission_time < '2025-11-01'  
    AND status = 'delivered'  
" --batch --silent | sed 's/\t/,/g' > billing_export_202510.csv
```

Ver También

- [Guía de Configuración](#) - Configurar ajustes de exportación de CDR
- [Guía de Operaciones](#) - Procedimientos de mantenimiento de CDR
- [Referencia de API](#) - Consultar CDR a través de la API REST

Referencia de Configuración de SMS-C

[← Volver al Índice de Documentación](#) | [README Principal](#)

Referencia completa para todas las opciones de configuración de SMS-C con ejemplos para escenarios de implementación comunes.

Tabla de Contenidos

- [Archivos de Configuración](#)
- [Configuración de Base de Datos](#)
- [Configuración de API](#)
- [Configuración de Interfaz Web](#)
- [Configuración de Federación](#)
- [Configuración de Cola de Mensajes](#)
- [Configuración de Carga](#)
- [Configuración de Diameter Sh \(HSS\)](#)
- [Configuración de ENUM](#)
- [Configuración de Traducción de Números](#)
- [Configuración de Enrutamiento](#)
- [Configuración de Ajuste de Rendimiento](#)
- [Configuración de Registro](#)
- [Escenarios Comunes de Configuración](#)

Archivos de Configuración

El SMS-C utiliza tres archivos de configuración principales:

config/config.exs

Configuración estática cargada en tiempo de compilación. Contiene:

- Valores predeterminados a nivel de aplicación
- Configuración del registrador
- Configuraciones de desarrollo/prueba
- Parámetros de ajuste de rendimiento

config/runtime.exs

Configuración en tiempo de ejecución cargada al inicio. Contiene:

- Configuraciones de conexión a la base de datos
- Configuración del clúster
- Integración de servicios externos (OCS, ENUM)
- Rutas iniciales y reglas de traducción
- Configuraciones específicas del entorno

config/prod.exs (opcional)

Sobrescrituras específicas de producción.

Mejor Práctica: Utilizar variables de entorno en `runtime.exs` para valores sensibles como contraseñas y claves de API.

Configuración de Almacenamiento SQL CDR

El SMS-C utiliza **Mnesia** para datos operativos (cola de mensajes, reglas de enrutamiento, traducciones de números) y admite **bases de datos SQL externas** para almacenamiento a largo plazo de CDR (Registro de Detalles de Llamadas), facturación y análisis.

Bases de Datos SQL Soportadas

El sistema admite las siguientes bases de datos SQL para exportación de CDR:

Base de Datos	Versión	Adaptador	Puerto Predeterminado
MySQL	8.0+	Ecto.Adapters.MyXQL	3306
MariaDB	10.5+	Ecto.Adapters.MyXQL	3306
PostgreSQL	13+	Ecto.Adapters.Postgres	5432

Nota: Mnesia se utiliza automáticamente para datos operativos (cola de mensajes, enrutamiento, traducciones) y no requiere configuración. La base de datos SQL se utiliza **solo** para exportación de CDR y almacenamiento a largo plazo.

Configuración de MySQL / MariaDB

```
# config/runtime.exs
config :sms_c, SmsC.Repo,
  adapter: Ecto.Adapters.MyXQL,
  username: System.get_env("DB_USERNAME") || "sms_user",
  password: System.get_env("DB_PASSWORD") || "secure_password",
  hostname: System.get_env("DB_HOSTNAME") || "localhost",
  port: String.to_integer(System.get_env("DB_PORT") || "3306"),
  database: System.get_env("DB_NAME") || "sms_c_prod",
  pool_size: String.to_integer(System.get_env("DB_POOL_SIZE") ||
"20")
```

Configuración de PostgreSQL

```
# config/runtime.exs
config :sms_c, SmsC.Repo,
  adapter: Ecto.Adapters.Postgres,
  username: System.get_env("DB_USERNAME") || "sms_user",
  password: System.get_env("DB_PASSWORD") || "secure_password",
  hostname: System.get_env("DB_HOSTNAME") || "localhost",
  port: String.to_integer(System.get_env("DB_PORT") || "5432"),
  database: System.get_env("DB_NAME") || "sms_c_prod",
  pool_size: String.to_integer(System.get_env("DB_POOL_SIZE") ||
"20")
```

Elegir una Base de Datos SQL

MySQL/MariaDB - Recomendado para la mayoría de las implementaciones:

- Excelente rendimiento para escrituras de CDR
- Fiabilidad comprobada en entornos de telecomunicaciones
- Amplio soporte de herramientas para sistemas de facturación
- Configuración de replicación sencilla

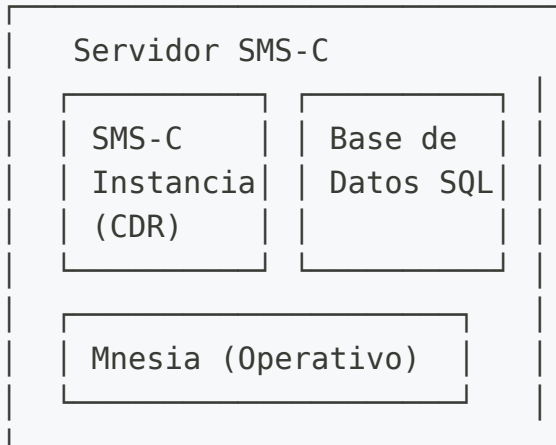
PostgreSQL - Considerar si necesitas:

- Características avanzadas de JSON/JSONB para análisis
- Consultas complejas sobre datos de CDR
- Infraestructura existente de PostgreSQL
- PostGIS para análisis geográfico

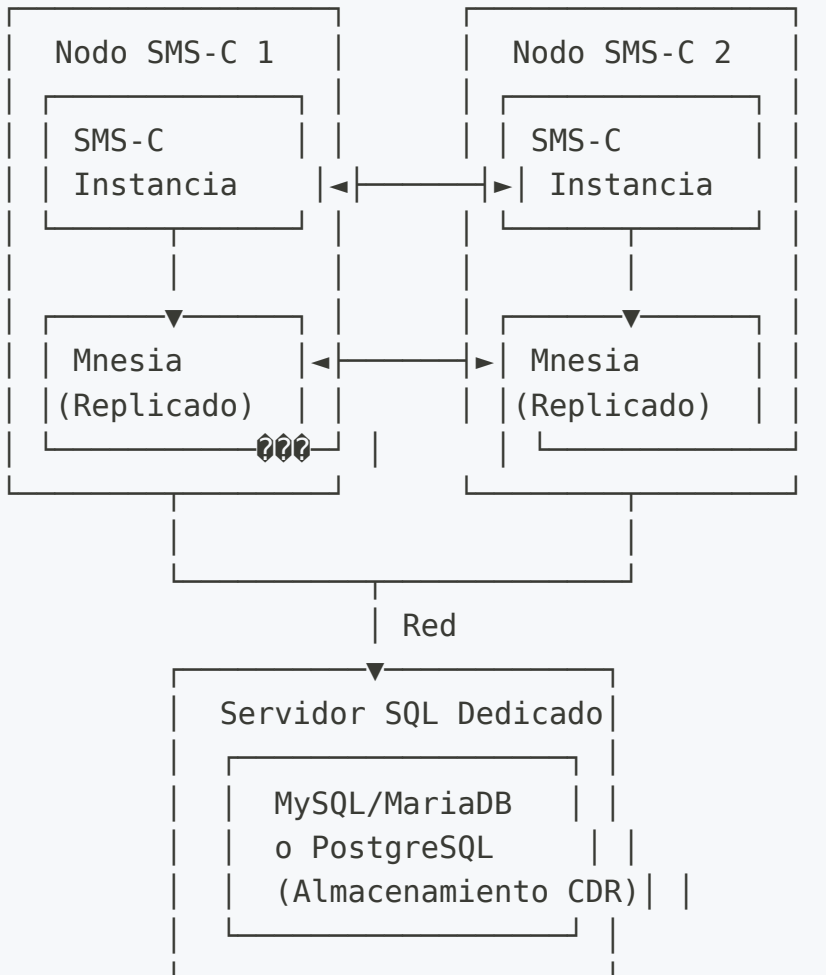
Topologías de Implementación

Importante: La base de datos SQL CDR puede ejecutarse en un **servidor separado** de tu(s) instancia(s) de SMS-C. Este es el enfoque recomendado para implementaciones en producción.

Implementación en un Solo Servidor (Desarrollo/Pruebas):



Implementación Distribuida (Producción - Recomendado):



Beneficios de un Servidor SQL Separado:

- **Aislamiento de Rendimiento:** Las escrituras de CDR no afectan el procesamiento de mensajes
- **Escalabilidad:** Escalar independientemente la base de datos y el procesamiento de mensajes
- **Fiabilidad:** El mantenimiento de la base de datos no afecta el tiempo de actividad de SMS-C
- **Gestión de Datos:** Almacenamiento centralizado de CDR para múltiples instancias de SMS-C
- **Flexibilidad de Respaldo:** Programas de respaldo y políticas de retención independientes

Directrices de Tamaño de Pool

Carga de Trabajo	Tamaño del Pool	Descripción
Desarrollo	5-10	Concurrencia mínima
Bajo Volumen (< 100 msg/sec)	10-15	Implementaciones pequeñas
Volumen Medio (100-1000 msg/sec)	20-30	Producción típica
Alto Volumen (> 1000 msg/sec)	40-100	Escenarios de alto rendimiento

Cálculo: `pool_size = (operaciones concurrentes de DB esperadas) * 1.5`

Ejemplos de Conexión a la Base de Datos

Usando Variables de Entorno (Recomendado para Producción):

```
# Establecer variables de entorno
export DB_USERNAME=sms_prod_user
export DB_PASSWORD=strong_password_here
export DB_HOSTNAME=db-primary.internal.example.com
export DB_PORT=3306
export DB_NAME=sms_c_production
export DB_POOL_SIZE=30
```

Configuración Directa (Solo Desarrollo):

```
config :sms_c, SmsC.Repo,
  username: "dev_user",
  password: "dev_password",
  hostname: "localhost",
  database: "sms_c_dev",
  pool_size: 5
```

Monitoreo del Pool de Conexiones

Monitorear el uso del pool a través de métricas de Prometheus:

- `ecto_pools_queue_time` - Tiempo de espera por conexión
- `ecto_pools_query_time` - Tiempo de ejecución de consultas
- `ecto_pools_connected_count` - Conexiones activas

Alertar si el tiempo de espera supera los 100 ms de manera consistente - indica la necesidad de un pool más grande.

Configuración de API

La API REST proporciona capacidades de envío y gestión de mensajes.

Configuración Básica de API

```
# config/runtime.exs
config :api_ex,
  port: String.to_integer(System.get_env("API_PORT") || "8443"),
  listen_ip: System.get_env("API_LISTEN_IP") || "0.0.0.0",
  enable_tls: System.get_env("API_ENABLE_TLS") != "false"
```

Configuración de TLS/SSL

Configuración de Producción con TLS (Recomendado):

```
config :api_ex,
  port: 8443,
  listen_ip: "0.0.0.0",
  enable_tls: true,
  tls_cert_path: "/etc/sms_c/certs/server.crt",
  tls_key_path: "/etc/sms_c/certs/server.key"
```

Configuración de Desarrollo sin TLS:

```
config :api_ex,
  port: 8080,
  listen_ip: "127.0.0.1",
  enable_tls: false
```

Configuración del Certificado de API

Generar un certificado autofirmado para pruebas:

```
# Crear directorio de certificados
mkdir -p priv/cert

# Generar clave privada
openssl genrsa -out priv/cert/server.key 2048

# Generar solicitud de firma de certificado
openssl req -new -key priv/cert/server.key -out
priv/cert/server.csr \
  -subj "/C=US/ST=State/L=City/O=Organization/CN=sms-
api.example.com"

# Generar certificado autofirmado (válido por 365 días)
openssl x509 -req -days 365 -in priv/cert/server.csr \
  -signkey priv/cert/server.key -out priv/cert/server.crt

# Establecer permisos
chmod 600 priv/cert/server.key
chmod 644 priv/cert/server.crt
```

Para producción, utilizar certificados de una CA confiable (Let's Encrypt, CA comercial, etc.).

Control de Acceso a la API

Lista Blanca de IP (Cortafuegos de Aplicación):

```
# Usando iptables (Linux)
iptables -A INPUT -p tcp --dport 8443 -s 10.0.0.0/8 -j ACCEPT
iptables -A INPUT -p tcp --dport 8443 -j DROP

# Usando firewalld (Red Hat/CentOS)
firewall-cmd --permanent --add-rich-rule='rule family="ipv4"
source address="10.0.0.0/8" port protocol="tcp" port="8443"
accept'
firewall-cmd --reload
```

Autenticación con Clave de API (Nivel de Aplicación):

Configurar a través de un plug personalizado en el enrutador - ver Guía de Operaciones para detalles de implementación.

Configuración de Interfaz Web

La interfaz web proporciona gestión de rutas, navegación de mensajes y monitoreo.

Configuración Básica de Interfaz Web

```
# config/runtime.exs
config :control_panel,
  port: String.to_integer(System.get_env("WEB_PORT") || "80"),
  hostname: System.get_env("WEB_HOSTNAME") || "localhost",
  enable_tls: System.get_env("WEB_ENABLE_TLS") == "true"
```

Configuración de Interfaz Web de Producción

```
config :control_panel,
  port: 443,
  hostname: "sms-admin.example.com",
  enable_tls: true,
  tls_cert_path: "/etc/sms_c/certs/web.crt",
  tls_key_path: "/etc/sms_c/certs/web.key"
```

Configuración de Proxy Inverso (Recomendado)

Usar Nginx o Apache como proxy inverso para mayor seguridad y características:

Ejemplo de Configuración de Nginx:

```

upstream sms_web {
    server 127.0.0.1:4000;
    keepalive 32;
}

server {
    listen 80;
    server_name sms-admin.example.com;
    return 301 https://$server_name$request_uri;
}

server {
    listen 443 ssl http2;
    server_name sms-admin.example.com;

    ssl_certificate /etc/letsencrypt/live/sms-
admin.example.com/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/sms-
admin.example.com/privkey.pem;
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers HIGH:!aNULL:!MD5;

    # Autenticación básica para mayor seguridad
    auth_basic "SMS-C Admin";
    auth_basic_user_file /etc/nginx/.htpasswd;

    location / {
        proxy_pass http://sms_web;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }

    # Soporte para WebSocket para LiveView
    location /live {
        proxy_pass http://sms_web;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
    }
}

```

```
    proxy_read_timeout 86400;
  }
}
```

Configuración de Federación

OmniMessage utiliza federación basada en HTTP para implementaciones de múltiples controladores. Los controladores se descubren entre sí a través de registros SRV DNS o listas de pares estáticas, intercambian registros de salud y frontend a través de HTTPS, y reenvían mensajes a controladores remotos cuando es necesario.

Consulta la [Guía de Federación Geográfica](#) para la arquitectura completa, ejemplos de implementación y solución de problemas.

Inicio Rápido

```
# config/runtime.exs
config :sms_c, :federation,
  enabled: true,
  dns_srv_domain: "_smc._tcp.smc.example.com"
```

Referencia Completa de Parámetros

Parámetro	Tipo	Predeterminado	Descripción
<code>enabled</code>	Booleano	<code>false</code>	Interruptor maestro para la federación
<code>dns_srv_domain</code>	Cadena	<code>""</code>	Dominio DNS SRV para descubrimiento de pares.
<code>dns_poll_interval_ms</code>	Entero	<code>30000</code>	Intervalo de resolución DNS.
<code>health_check_interval_ms</code>	Entero	<code>10000</code>	Intervalo de verificación de salud de pares
<code>registry_sync_interval_ms</code>	Entero	<code>15000</code>	Intervalo de intercambio registros frontend.
<code>forward_retry_interval_ms</code>	Entero	<code>5000</code>	Intervalo de reintento de mensajes en cola.
<code>forward_max_retries</code>	Entero	<code>50</code>	Máximo de intentos de reenvío antes de marcar como fallido.

Parámetro	Tipo	Predeterminado	Descripción
<code>http_timeout_ms</code>	Entero	<code>5000</code>	Tiempo de espera para todas las llamadas HTTP de pares.
<code>api_port</code>	Entero	<code>8443</code>	Puerto de API para construir URLs de pares.
<code>static_peers</code>	Lista	<code>[]</code>	Lista de pares estáticos (alternativa a DNS SRV).

Requisitos de Red

La federación requiere solo el puerto **8443** (HTTPS) entre sitios, en comparación con los puertos 4369 + 9100-9200 para el agrupamiento de Erlang.

```
# Permitir tráfico de federación desde sitios pares
iptables -A INPUT -p tcp -s 10.0.1.0/24 --dport 8443 -j ACCEPT
iptables -A INPUT -p tcp -s 10.0.2.0/24 --dport 8443 -j ACCEPT
```

Configuración de Cola de Mensajes

Controla el comportamiento de retención y expiración de mensajes.

Expiración de Mensajes

```
# config/runtime.exs
config :sms_c,
  dead_letter_time_minutes: 1440 # 24 horas
```

Valores Comunes:

- **60** - 1 hora (pruebas/desarrollo)
- **1440** - 24 horas (producción típica)
- **4320** - 3 días (retención extendida)
- **10080** - 7 días (retención máxima)

Los mensajes más antiguos que este valor se vuelven no entregables y se marcan para limpieza.

Configuración de Reintentos de Entrega

El comportamiento de reintentos utiliza retroceso exponencial:

```
Retraso de Reintento = 2^(número_de_intentos) minutos
```

Intento	Retraso
1	2 minutos
2	4 minutos
3	8 minutos
4	16 minutos
5	32 minutos
6	64 minutos
7	128 minutos
8	256 minutos

Máximo de intentos antes de la carta muerta: Limitado por `dead_letter_time_minutes`.

Configuración de Limpieza

```
# config/config.exs
config :sms_c,
  cleanup_interval_minutes: 10,
  fingerprint_ttl_minutes: 5,
  event_ttl_days: 7
```

Intervalos de Limpieza:

- **cleanup_interval_minutes:** Con qué frecuencia se ejecuta el trabajador de limpieza (predeterminado: 10)
- **fingerprint_ttl_minutes:** Ventana de detección de duplicados (predeterminado: 5)
- **event_ttl_days:** Retención de registros de eventos (predeterminado: 7)

Configuración de Carga

Integración con OCS para carga y facturación en línea.

Habilitar Carga

```
# config/runtime.exs
config :sms_c,
  default_charging_enabled: true,
  ocs_url: "http://ocs.internal.example.com:2080/jsonrpc",
  ocs_tenant: "sms.example.com",
  ocs_destination: "default",
  ocs_source: "sms_platform",
  ocs_subject: "sms_user",
  ocs_account: "default_account"
```

Deshabilitar Carga

```
# config/runtime.exs
config :sms_c,
  default_charging_enabled: false
```

Cuando está deshabilitado, todos los mensajes se procesan sin verificaciones de carga.

Configuración de Carga por Inquilino

```
config :sms_c,
  ocs_url: System.get_env("OCS_URL") ||
  "http://localhost:2080/jsonrpc",
  ocs_tenant: System.get_env("OCS_TENANT") ||
  "tenant1.example.com",
  ocs_account: System.get_env("OCS_ACCOUNT") || "default"
```

Variables de Entorno por Inquilino:

```
# Inquilino 1
export OCS_TENANT=tenant1.example.com
export OCS_ACCOUNT=tenant1_account

# Inquilino 2
export OCS_TENANT=tenant2.example.com
export OCS_ACCOUNT=tenant2_account
```

Comportamiento de Fallo de Carga

Configurar qué sucede cuando falla la carga:

```
config :sms_c,
  charging_failure_action: :allow # o :deny
```

- **:allow** - Procesar el mensaje incluso si falla la carga (registrar error)
- **:deny** - Rechazar el mensaje si falla la carga

Ejemplo de Conexión a OCS

Probar Conectividad OCS:

```
# Probar API OCS
curl -X POST http://ocs.internal.example.com:2080/jsonrpc \
  -H "Content-Type: application/json" \
  -d '{
    "method": "SessionSv1.AuthorizeEvent",
    "params": [{
      "Tenant": "sms.example.com",
      "Account": "test_account",
      "Destination": "1234567890",
      "Usage": 100
    }],
    "id": 1
  }'
```

Respuesta esperada:

```
{
  "id": 1,
  "result": {
    "Attributes": {},
    "MaxUsage": 100,
    ...
  }
}
```

Configuración de Diameter Sh (HSS)

OmniMessage puede consultar el HSS a través de la interfaz Diameter Sh para detectar suscriptores en red que están temporalmente fuera de línea, evitando que los mensajes se enruten a través de la puerta de enlace predeterminada. Consulta la [Guía de Búsqueda de Suscriptores HSS](#) para la documentación operativa completa.

```
# Habilitar dip HSS en la ruta de enrutamiento
config :sms_c,
  diameter_enabled: true

# Configuración de la pila Diameter
config :diameter_ex,
  diameter: %{
    service_name: :omnimessage,
    listen_ip: "0.0.0.0",
    listen_port: 3868,
    decode_format: :map,
    host: "smsc01",
    realm: "epc.mnc005.mcc547.3gppnetwork.org",
    product_name: "OmniMessage",
    request_timeout: 5000,
    control_module: SmsC.Diameter.Control,
    processor_module: DiameterEx.Processor,
    vendor_id: 10415,
    supported_vendor_ids: [10415],
    applications: [
      %{
        application_name: :sh,
        application_dictionary: :diameter_gen_3gpp_sh,
        vendor_specific_application_ids: [
          %{vendor_id: 10415, auth_application_id: 16_777_217,
acct_application_id: nil}
        ]
      }
    ],
    peers: [
      %{
        host: "dra01.epc.mnc005.mcc547.3gppnetwork.org",
        ip: "10.0.0.1",
        port: 3868,
        realm: "epc.mnc005.mcc547.3gppnetwork.org",
        tls: false,
        transport: :diameter_tcp,
        initiate_connection: true
      }
    ]
  }
}
```

Parámetro	Tipo	Requerido	Predeterminado	D
<code>diameter_enabled</code>	Booleano	No	<code>false</code>	H d la d si H r e
<code>mock_sh</code>	Booleano	No	<code>false</code>	U re H si p (s D ir
<code>mock_sh_on_net_numbers</code>	Lista	No	<code>[]</code>	M tr c e si

Para la referencia completa de parámetros de la pila Diameter (host, realm, peers, applications), consulta la [Guía de Búsqueda de Suscriptores HSS](#).

Configuración de ENUM

Búsquedas de números E.164 basadas en DNS para enrutamiento inteligente.

Deshabilitar ENUM (Predeterminado)

```
# config/runtime.exs
config :sms_c,
  enum_enabled: false
```

Habilitar ENUM con DNS Predeterminado

```
config :sms_c,
  enum_enabled: true,
  enum_domains: ["e164.arpa", "e164.org"],
  enum_dns_servers: [], # Usar DNS predeterminado del sistema
  enum_timeout: 5000 # 5 segundos
```

Habilitar ENUM con Servidores DNS Personalizados

```
config :sms_c,
  enum_enabled: true,
  enum_domains: ["e164.internal.example.com", "e164.arpa"],
  enum_dns_servers: [
    {"10.0.1.53", 53}, # Servidor DNS interno
    {"8.8.8.8", 53}, # DNS Público de Google (fallback)
    {"1.1.1.1", 53} # DNS de Cloudflare (fallback)
  ],
  enum_timeout: 3000 # 3 segundos (failover más rápido)
```

Prioridad de Dominio ENUM

Los dominios se consultan en orden hasta que se logra una búsqueda exitosa:

```
config :sms_c,  
  enum_domains: [  
    "e164.internal.example.com", # Intentar interno primero  
    "e164.carrier.net",         # Luego carrier  
    "e164.arpa"                 # Luego registro público  
  ]
```

Ajuste de Rendimiento ENUM

Para Redes de Baja Latencia:

```
enum_timeout: 2000 # 2 segundos
```

Para Enlaces de Alta Latencia/Satélite:

```
enum_timeout: 10000 # 10 segundos
```

Ejemplo de Configuración de DNS ENUM

Configurar Zona ENUM Privada (formato BIND9):

```
; Archivo de zona para e164.internal.example.com  
$ORIGIN e164.internal.example.com.  
$TTL 300  
  
; Número: +1-555-0100 se convierte en  
0.0.1.0.5.5.5.1.e164.internal.example.com  
0.0.1.0.5.5.5.1.e164.internal.example.com. IN NAPTR 100 10 "u"  
"E2U+sip" "!^.*$!sip:15550100@voip-gateway.example.com!" .  
0.0.1.0.5.5.5.1.e164.internal.example.com. IN NAPTR 100 20 "u"  
"E2U+pstn" "!^.*$!pstn:gateway-a.example.com!" .  
  
; Número: +1-555-0200  
0.0.2.0.5.5.5.1.e164.internal.example.com. IN NAPTR 100 10 "u"  
"E2U+sip" "!^.*$!sip:15550200@voip-gateway.example.com!" .
```

Probar Resolución ENUM:

```
# Consultar dominio ENUM
dig @10.0.1.53 NAPTR 0.0.1.0.5.5.5.1.e164.internal.example.com

# La salida esperada incluye registros NAPTR:
# 0.0.1.0.5.5.5.1.e164.internal.example.com. 300 IN NAPTR 100 10
"u" "E2U+sip" "!^.*$!sip:15550100@voip-gateway.example.com!" .
```

Configuración de Traducción de Números

Normalización de números basada en expresiones regulares aplicada antes del enrutamiento.

Deshabilitar Traducción de Números

```
# config/runtime.exs
config :sms_c,
  translation_rules: []
```

Ejemplos Básicos de Traducción de Números

Agregar Código de País a Números Locales:

```

config :sms_c,
  translation_rules: [
    %{
      calling_prefix: nil,
      called_prefix: "",
      source_smsc: nil,
      calling_match: "^(\\d{10})$",           # Coincidir números
de 10 dígitos
      calling_replace: "+1\\1",             # Prepend +1
      called_match: "^(\\d{10})$",
      called_replace: "+1\\1",
      priority: 100,
      description: "Agregar +1 a números norteamericanos de 10
dígitos",
      enabled: true
    }
  ]

```

Normalizar Formato Internacional:

```

%{
  calling_prefix: nil,
  called_prefix: nil,
  source_smsc: nil,
  calling_match: "^00(\\d+)$",           # Coincidir prefijo
00
  calling_replace: "+\\1",               # Reemplazar con +
  called_match: "^00(\\d+)$",
  called_replace: "+\\1",
  priority: 10,
  description: "Convertir prefijo internacional 00 a +",
  enabled: true
}

```

Eliminar Caracteres de Formato:

```
%{
  calling_prefix: nil,
  called_prefix: nil,
  source_smsc: nil,
  calling_match: "^\\+?1?[\\s\\-\\.\\(\\)]*(\\d{3})[\\s\\-\\.\\(\\)]*(\\d{3})
[\\s\\-\\.\\(\\)]*(\\d{4})$",
  calling_replace: "+1\\1\\2\\3",
  called_match: "^\\+?1?[\\s\\-\\.\\(\\)]*(\\d{3})[\\s\\-\\.\\(\\)]*(\\d{3})
[\\s\\-\\.\\(\\)]*(\\d{4})$",
  called_replace: "+1\\1\\2\\3",
  priority: 50,
  description: "Normalizar formato de número telefónico de EE.
UU.",
  enabled: true
}
```

Traducción Específica de Carrier

Eliminación de Código de Ruta:

```
%{
  calling_prefix: nil,
  called_prefix: "101", # Solo para prefijo
101
  source_smsc: "carrier_a", # Solo de este
carrier
  calling_match: nil, # No cambiar
llamada
  calling_replace: nil,
  called_match: "^101(\\d+)$", # Eliminar código de
ruta 101
  called_replace: "\\1",
  priority: 5,
  description: "Eliminar código de ruta de carrier del número
llamado",
  enabled: true
}
```

Traducción de Múltiples Reglas

Las reglas se evalúan en orden de prioridad (número más bajo = mayor prioridad):

```
config :sms_c,  
  translation_rules: [  
    # Prioridad 1: Reglas más específicas primero  
    %{  
      calling_prefix: "1555",  
      called_prefix: nil,  
      source_smsc: nil,  
      calling_match: "^(1555\d{7})$",  
      calling_replace: "+\1",  
      called_match: nil,  
      called_replace: nil,  
      priority: 1,  
      description: "Normalización de número premium",  
      enabled: true  
    },  
  
    # Prioridad 50: Reglas generales  
    %{  
      calling_prefix: nil,  
      called_prefix: nil,  
      source_smsc: nil,  
      calling_match: "^(\\d{10})$",  
      calling_replace: "+\1",  
      called_match: "^(\\d{10})$",  
      called_replace: "+\1",  
      priority: 50,  
      description: "Normalización general de 10 dígitos",  
      enabled: true  
    }  
  ]  
]
```

Configuración de Enrutamiento

Reglas de enrutamiento iniciales cargadas en el primer inicio. Consulta la [Guía de Enrutamiento SMS](#) para la documentación completa de enrutamiento.

Cargar Rutas desde la Configuración

```
# config/runtime.exs
config :sms_c,
  sms_routes: [
    # Ejemplo de enrutamiento geográfico
    %{
      calling_regex: nil,
      called_regex: ~r/^\+1/,
      source_smsc: nil,
      dest_smsc: "north_america_gateway",
      source_type: nil,
      enum_domain: nil,
      auto_reply: false,
      auto_reply_message: nil,
      drop: false,
      charged: :default,
      on_net_only: false,
      weight: 100,
      priority: 50,
      description: "Enrutamiento de América del Norte",
      enabled: true
    },

    # Ejemplo de enrutamiento balanceado
    %{
      calling_regex: nil,
      called_regex: ~r/^\+44/,
      source_smsc: nil,
      dest_smsc: "uk_gateway_1",
      source_type: nil,
      enum_domain: nil,
      auto_reply: false,
      auto_reply_message: nil,
      drop: false,
      charged: :default,
      on_net_only: false,
      weight: 70,
      priority: 50,
      description: "Puerta de enlace primaria del Reino Unido
(70%)",
      enabled: true
    },
  ],
```

```
%{
  calling_regex: nil,
  called_regex: ~r/^\+44/,
  source_smsc: nil,
  dest_smsc: "uk_gateway_2",
  source_type: nil,
  enum_domain: nil,
  auto_reply: false,
  auto_reply_message: nil,
  drop: false,
  charged: :default,
  on_net_only: false,
  weight: 30,
  priority: 50,
  description: "Puerta de enlace de respaldo del Reino Unido
(30%)",
  enabled: true
},

# Solo en red – restringir un enlace SMPP a destinos solo en
red
%{
  calling_regex: nil,
  called_regex: nil,
  source_smsc: "carrier_smpp_bind",
  dest_smsc: "local_msc",
  source_type: :smpp,
  enum_domain: nil,
  auto_reply: false,
  auto_reply_message: nil,
  drop: false,
  charged: :default,
  on_net_only: true,
  weight: 100,
  priority: 50,
  description: "Carrier X – terminación solo en red",
  enabled: true
}
]
```

Omitir Carga Inicial de Rutas

```
# No cargar rutas desde la configuración (gestionar solo a través
de la Interfaz Web)
config :sms_c,
  sms_routes: []
```

Las rutas definidas en la configuración solo se cargan si la tabla de enrutamiento está vacía (primer inicio).

Configuración de Ajuste de Rendimiento

Consulta la [Guía de Ajuste de Rendimiento](#) para estrategias de optimización detalladas.

Trabajador de Inserción por Lotes

```
# config/config.exs
config :sms_c,
  batch_insert_batch_size: 100,           # Mensajes por lote
  batch_insert_flush_interval_ms: 100    # Tiempo máximo de
espera en ms
```

Perfiles de Rendimiento:

Perfil	Tamaño del Lote	Intervalo	Rendimiento	Latencia
Alto Volumen	200	200ms	~5,000 msg/sec	Hasta 200ms
Equilibrado	100	100ms	~4,500 msg/sec	Hasta 100ms
Baja Latencia	50	20ms	~3,000 msg/sec	Hasta 20ms
Tiempo Real	10	10ms	~1,500 msg/sec	Hasta 10ms

Configuración de Registro

Niveles de Registro

```
# config/config.exs
config :logger, :console,
  level: :info, # :debug, :info, :warning, :error
  format: "$time $metadata[$level] $message\n",
  metadata: [:request_id, :message_id, :route_id]
```

Recomendado para Producción: `:info` o `:warning` **Recomendado para Desarrollo:** `:debug`

Destinos de Salida de Registro

Solo Consola (Desarrollo):

```
config :logger,
  backends: [:console]
```

Registrador de Archivos (Producción):

```
config :logger,  
  backends: [:console, {LoggerFileBackend, :file_log}]  
  
config :logger, :file_log,  
  path: "/var/log/sms_c/application.log",  
  level: :info,  
  format: "$time $metadata[$level] $message\n",  
  metadata: [:request_id, :message_id]
```

Rotación de Registros

Usando logrotate (Linux):

```
# /etc/logrotate.d/sms_c  
/var/log/sms_c/*.log {  
    daily  
    rotate 30  
    compress  
    delaycompress  
    notifempty  
    create 0644 sms_user sms_group  
    sharedscripts  
    postrotate  
        # Señalar a la aplicación que vuelva a abrir el archivo de  
registro  
        systemctl reload sms_c  
    endscript  
}
```

Escenarios Comunes de Configuración

Agregador de Alto Volumen

Optimizar para el máximo rendimiento (5,000+ mensajes/segundo):

```
# Base de Datos
config :sms_c, SmsC.Repo,
  pool_size: 50

# Trabajador por lotes
config :sms_c,
  batch_insert_batch_size: 200,
  batch_insert_flush_interval_ms: 200

# Retención de Mensajes
config :sms_c,
  dead_letter_time_minutes: 1440 # 24 horas

# Carga (deshabilitada por rendimiento)
config :sms_c,
  default_charging_enabled: false

# Limpieza (intervalos extendidos)
config :sms_c,
  cleanup_interval_minutes: 30
```

Mensajería en Tiempo Real para Empresas

Optimizar para baja latencia (< 20ms):

```
# Base de Datos
config :sms_c, SmsC.Repo,
  pool_size: 20

# Trabajador por lotes (baja latencia)
config :sms_c,
  batch_insert_batch_size: 20,
  batch_insert_flush_interval_ms: 10

# Retención de Mensajes
config :sms_c,
  dead_letter_time_minutes: 4320 # 3 días

# Carga (habilitada)
config :sms_c,
  default_charging_enabled: true,
  ocs_url: "http://ocs.local:2080/jsonrpc"
```

Desarrollo/Pruebas

Optimizar para depuración y visibilidad:

```
# Base de Datos
config :sms_c, SmsC.Repo,
  pool_size: 5

# Trabajador por lotes (inmediato)
config :sms_c,
  batch_insert_batch_size: 1,
  batch_insert_flush_interval_ms: 10

# Registro (verbose)
config :logger, :console,
  level: :debug

# Retención de Mensajes (corta)
config :sms_c,
  dead_letter_time_minutes: 60 # 1 hora

# Carga (deshabilitada)
config :sms_c,
  default_charging_enabled: false
```

Proveedor de Servicios Multi-Inquilino

Configuración separada por inquilino:

```
# Entorno del Inquilino 1
export DB_NAME=sms_c_tenant1
export OCS_TENANT=tenant1.example.com
export OCS_ACCOUNT=tenant1_account
export NODE_NAME=sms_tenant1@node1.example.com

# Entorno del Inquilino 2
export DB_NAME=sms_c_tenant2
export OCS_TENANT=tenant2.example.com
export OCS_ACCOUNT=tenant2_account
export NODE_NAME=sms_tenant2@node1.example.com
```

Redundancia Geográfica

Agrupar a través de regiones:

```
# Clúster de EE. UU. Este
config :sms_c,
  cluster_nodes: [
    : "sms@us-east-1a.example.com",
    : "sms@us-east-1b.example.com",
    : "sms@us-west-1a.example.com" # Cross-region for DR
  ],
  smsc_node_name: "us-east-1a"
```

Validación de Configuración

Probar la configuración antes de la implementación:

```
# Verificar la sintaxis de configuración
mix compile

# Validar conexión a la base de datos
mix ecto.create
mix ecto.migrate

# Probar conectividad OCS (si está habilitada)
curl -X POST http://localhost:2080/jsonrpc -H "Content-Type:
application/json" \
  -d '{"method":"SessionSv1.Ping","params":[],"id":1}'

# Iniciar aplicación en modo interactivo
iex -S mix phx.server
```

Referencia de Variables de Entorno

Variables de entorno comunes utilizadas en la configuración:

Variable	Propósito	Ejemplo
DB_USERNAME	Nombre de usuario de la base de datos	sms_prod_user
DB_PASSWORD	Contraseña de la base de datos	strong_password
DB_HOSTNAME	Host de la base de datos	db.internal.example.com
DB_PORT	Puerto de la base de datos	3306
DB_NAME	Nombre de la base de datos	sms_c_production
DB_POOL_SIZE	Tamaño del pool de conexiones	30
API_PORT	Puerto de escucha de la API	8443
API_LISTEN_IP	IP de escucha de la API	0.0.0.0
WEB_PORT	Puerto de la Interfaz Web	443
NODE_NAME	Nombre del nodo Erlang	sms@node1.example.com
ERLANG_COOKIE	Secreto del clúster	shared_cookie_value
OCS_URL	URL de la API OCS	http://ocs.local:2080/jsonrpc
OCS_TENANT	Inquilino OCS	sms.example.com

Mejores Prácticas de Configuración

1. **Usar Variables de Entorno** para valores sensibles (contraseñas, claves de API)
2. **Probar Cambios de Configuración** en staging antes de producción
3. **Documentar Configuraciones Personalizadas** en notas de implementación
4. **Control de Versiones de Archivos de Configuración** (excluyendo secretos)
5. **Monitorear Después de Cambios** para regresiones de rendimiento
6. **Mantener Copias de Seguridad** de configuraciones funcionales
7. **Validar Antes de Reiniciar** para evitar fallos de inicio
8. **Usar Nombres Consistentes** en todos los entornos
9. **Establecer Límites de Recursos** apropiados para el hardware
10. **Revisar Periódicamente** para eliminar características no utilizadas

Solución de Problemas de Configuración

Síntoma	Causa Probable	Solución
La aplicación no inicia	Error de sintaxis en la configuración	Verificar registros, validar sintaxis
Falla la conexión a la base de datos	Credenciales/host incorrectos	Verificar variables de entorno DB_*
API no accesible	Puerto/IP de enlace incorrecto	Verificar API_PORT y listen_ip
Los nodos del clúster no se conectan	Desajuste de cookie, cortafuegos	Verificar ERLANG_COOKIE, revisar puertos 4369, 9100-9200
Fallos de carga	OCS inalcanzable	Probar conectividad a ocs_url
Fallos en búsquedas ENUM	Servidor DNS inalcanzable	Probar conectividad DNS, verificar tiempo de espera
Rendimiento deficiente	Configuración incorrecta de lotes	Revisar Guía de Ajuste de Rendimiento
Mensajes no enrutados	Rutas no cargadas	Verificar configuración de sms_routes o Interfaz Web

Para ayuda adicional, consulta la [Guía de Solución de Problemas](#).

Configuración de Almacenamiento de Mensajes (Mnesia)

Retención de Mensajes

Los mensajes se almacenan en Mnesia para un acceso rápido con limpieza automática configurable.

```
config :sms_c,  
  # Cuánto tiempo mantener los mensajes en Mnesia (horas)  
  message_retention_hours: 24,  
  
  # Con qué frecuencia verificar mensajes antiguos (minutos)  
  retention_check_interval_minutes: 60
```

Recomendaciones:

- **Producción:** 24-72 horas (equilibrar necesidades operativas vs memoria)
- **Desarrollo:** 4-8 horas (limpieza más rápida para pruebas)
- **Alto volumen:** 12-24 horas (conservar memoria)

Impacto en la Memoria:

- Mensaje promedio: ~1KB
- 10,000 mensajes: ~10MB
- 100,000 mensajes: ~100MB

Exportación de CDR (Registro de Detalles de Llamadas)

Cuando los mensajes se entregan o expiran, los CDR pueden escribirse automáticamente en tu base de datos Ecto para almacenamiento a largo plazo y análisis de facturación.

```
config :sms_c,  
  # Habilitar/deshabilitar escritura de CDR  
  cdr_enabled: true
```

Los Registros de CDR Incluyen:

- ID de mensaje, números de llamada/llamados
- SMSC de origen/destino
- Nodo de origen/destino (para clústeres)
- Tiempos de envío, entrega, expiración
- Estado, intentos de entrega
- Cuerpo del mensaje opcional (ver controles de privacidad)

Cuándo Deshabilitar:

- Entornos de prueba donde no se necesitan CDR
- Solución de problemas temporal para reducir la carga de la base de datos

Controles de Privacidad

Configurar la visibilidad y retención del cuerpo del mensaje para cumplir con la privacidad.

```
config :sms_c,  
  # Eliminar el cuerpo del mensaje de Mnesia después de la entrega  
  exitosa  
  delete_message_body_after_delivery: false,  
  
  # Ocultar el cuerpo del mensaje en la interfaz web  
  hide_message_body_in_ui: false,  
  
  # Ocultar el cuerpo del mensaje en exportaciones CSV  
  hide_message_body_in_export: false
```

Casos de Uso:

Configuración	Caso de Uso
<code>delete_message_body_after_delivery: true</code>	Ahorrar espacio en Mnesia, cumplimiento de privacidad
<code>hide_message_body_in_ui: true</code>	Prevenir que el operador vea el contenido del mensaje
<code>hide_message_body_in_export: true</code>	Cumplimiento de exportación de datos, informes sanitizados

Ejemplos de Configuraciones:

Máxima Privacidad (Cumplimiento)

```
config :sms_c,
  delete_message_body_after_delivery: true,
  hide_message_body_in_ui: true,
  hide_message_body_in_export: true,
  cdr_enabled: true # Mantener CDRs sin cuerpos
```

Desarrollo (Visibilidad Completa)

```
config :sms_c,
  delete_message_body_after_delivery: false,
  hide_message_body_in_ui: false,
  hide_message_body_in_export: false,
  cdr_enabled: true
```

Registro de Inicio

Al iniciar la aplicación, se registra el estado de la configuración:

```
[info] Almacenamiento de mensajes: Mnesia (retención: 24h)
[info] Exportación de CDR: HABILITADA
[info] Eliminación después de la entrega: DESHABILITADA
[info] Carga OCS: HABILITADA (url: http://..., inquilino: ...)
```

Esto proporciona visibilidad inmediata sobre las características activas.

Federación Geográfica

[← Volver al Índice de Documentación](#) | [Referencia de Configuración](#) | [Guía de Operaciones](#)

Descripción General

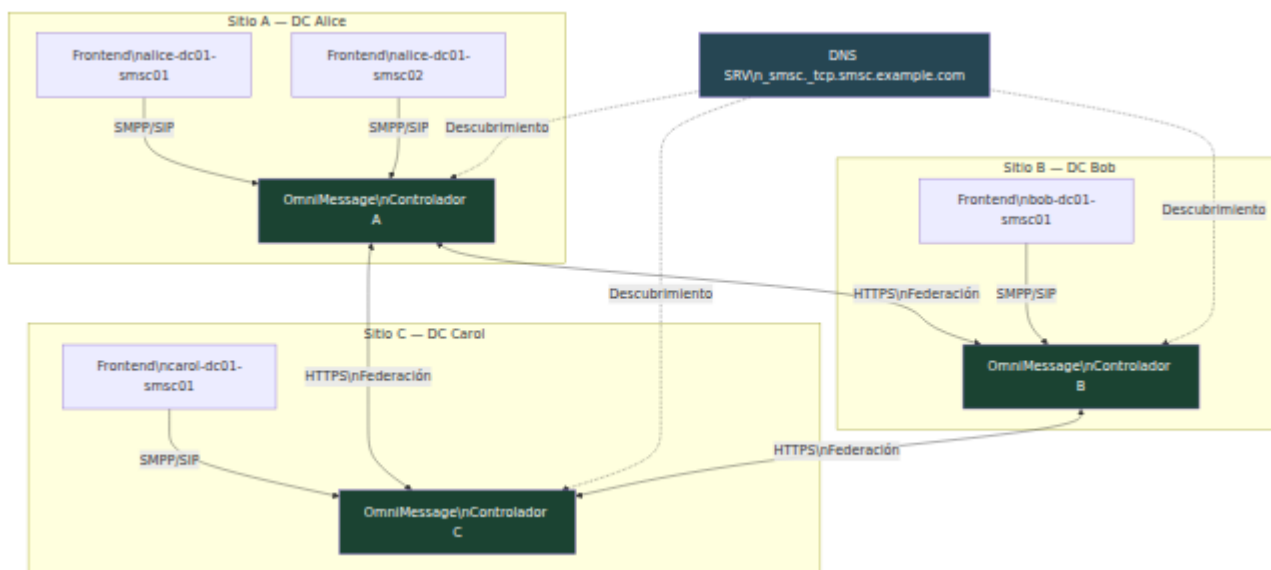
OmniMessage utiliza **federación basada en HTTP** con un **modelo de extracción** para implementaciones de múltiples controladores en centros de datos o regiones. Cada controlador opera de manera independiente, gestionando su propia cola de mensajes, tabla de enrutamiento y conjunto de frontends conectados. Los controladores se descubren entre sí a través de registros DNS SRV (o configuración estática), intercambiando información de salud y registro de frontends a través de HTTPS.

Cuando el enrutamiento determina que un mensaje pertenece a un sitio remoto, el mensaje **permanece en la Mnesia del controlador de origen**. Se envía una notificación ligera al controlador de destino, lo que desencadena una consulta inmediata. El FederationPoller del controlador de destino consulta periódicamente (y al recibir notificación) a los pares de origen en busca de mensajes destinados a sus frontends locales, los almacena en caché y los pone a disposición de los frontends locales. Después de la entrega, el controlador de destino informa el estado de entrega de vuelta al origen.

Características Clave

Aspecto	Detalle
Red	Funciona sobre cualquier WAN, incluyendo enlaces poco fiables
Estado compartido	Ninguno — cada controlador es independiente
Propiedad del mensaje	Los mensajes permanecen en el controlador de origen hasta ser entregados
Seguridad	HTTPS con TLS
Puertos	Un solo puerto HTTPS (8443)
Objetivo de escalado	5-20 controladores
Manejo de particiones	Reintentos del Poller — sin división de cerebro

Arquitectura



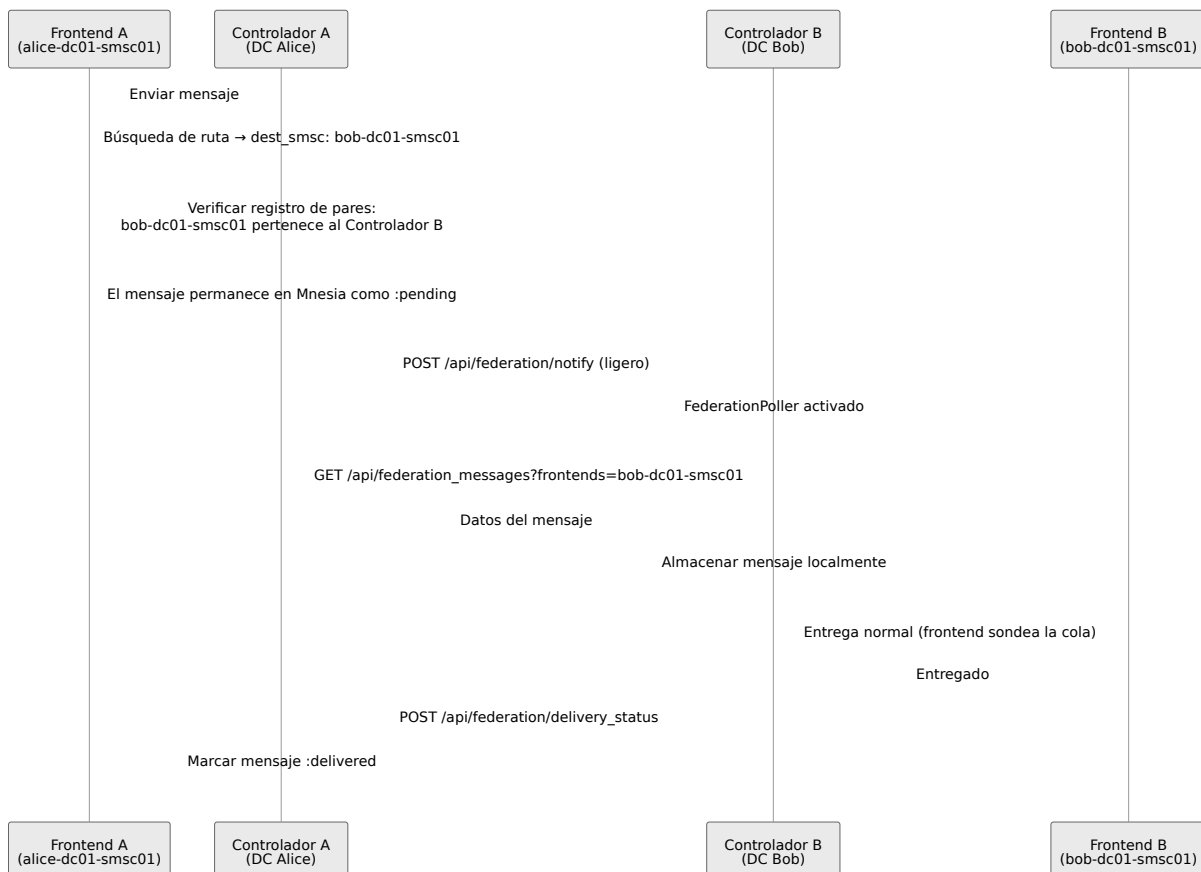
Decisiones de Diseño Clave

- **Flujo de mensajes (modelo de extracción):** Los mensajes permanecen en el controlador de origen. El controlador de destino los extrae mediante sondeos periódicos y notificaciones bajo demanda. Los frontends solo se comunican con su controlador local.
- **Manejo de particiones:** Los reintentos del Poller se realizan automáticamente. Los mensajes permanecen seguros en el controlador de origen hasta ser entregados con éxito.
- **Sincronización de configuración:** Ninguna. Cada controlador gestiona su propia tabla de enrutamiento de manera independiente.
- **Malla de pares:** Malla completa. Todos los controladores intercambian información de salud y registros de frontends con todos los demás controladores.
- **Descubrimiento:** Registros DNS SRV (recomendado) o lista de pares estática (para entornos sin DNS dinámico).

Flujo de Mensajes (Modelo de Extracción)

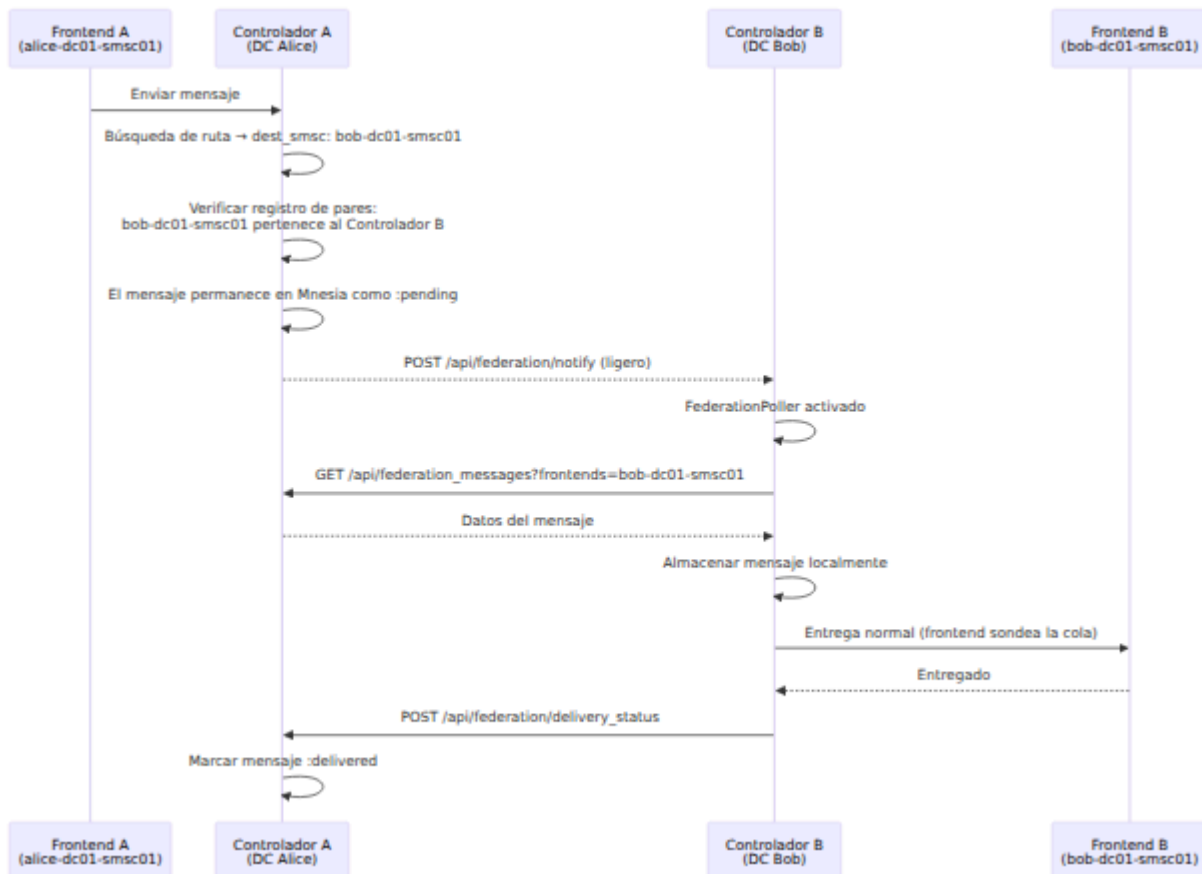
Cuando un mensaje llega a un controlador y el enrutamiento determina que el frontend de destino está en un controlador remoto:

1. El mensaje permanece en la Mnesia del controlador de origen como `:pending`
2. Se envía una notificación ligera al controlador de destino (fire-and-forget)
3. El FederationPoller del controlador de destino obtiene el mensaje a través de `GET /api/federation_messages`
4. El mensaje se almacena en caché localmente y se pone a disposición de los frontends locales
5. Después de la entrega, el destino informa el estado de vuelta al origen



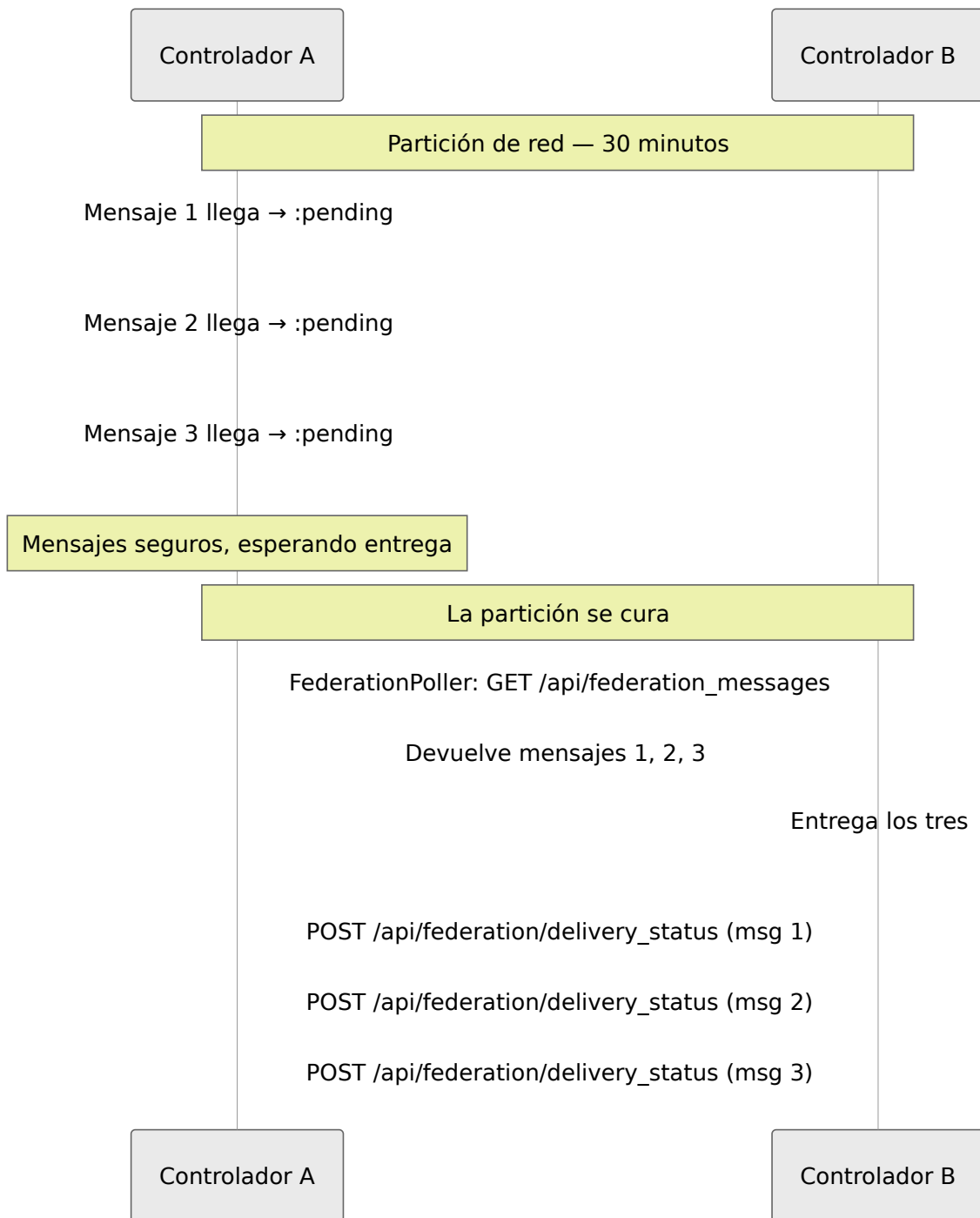
Par Inalcanzable – Fallo de Notificación

Si el controlador de destino está inactivo cuando se envía la notificación, la notificación falla silenciosamente. El mensaje permanece `:pending` en el origen. Cuando el controlador de destino se recupera, su `FederationPoller` reanuda el sondeo periódico y descubre el mensaje durante el siguiente ciclo (predeterminado: cada 5 segundos). No se pierden mensajes.



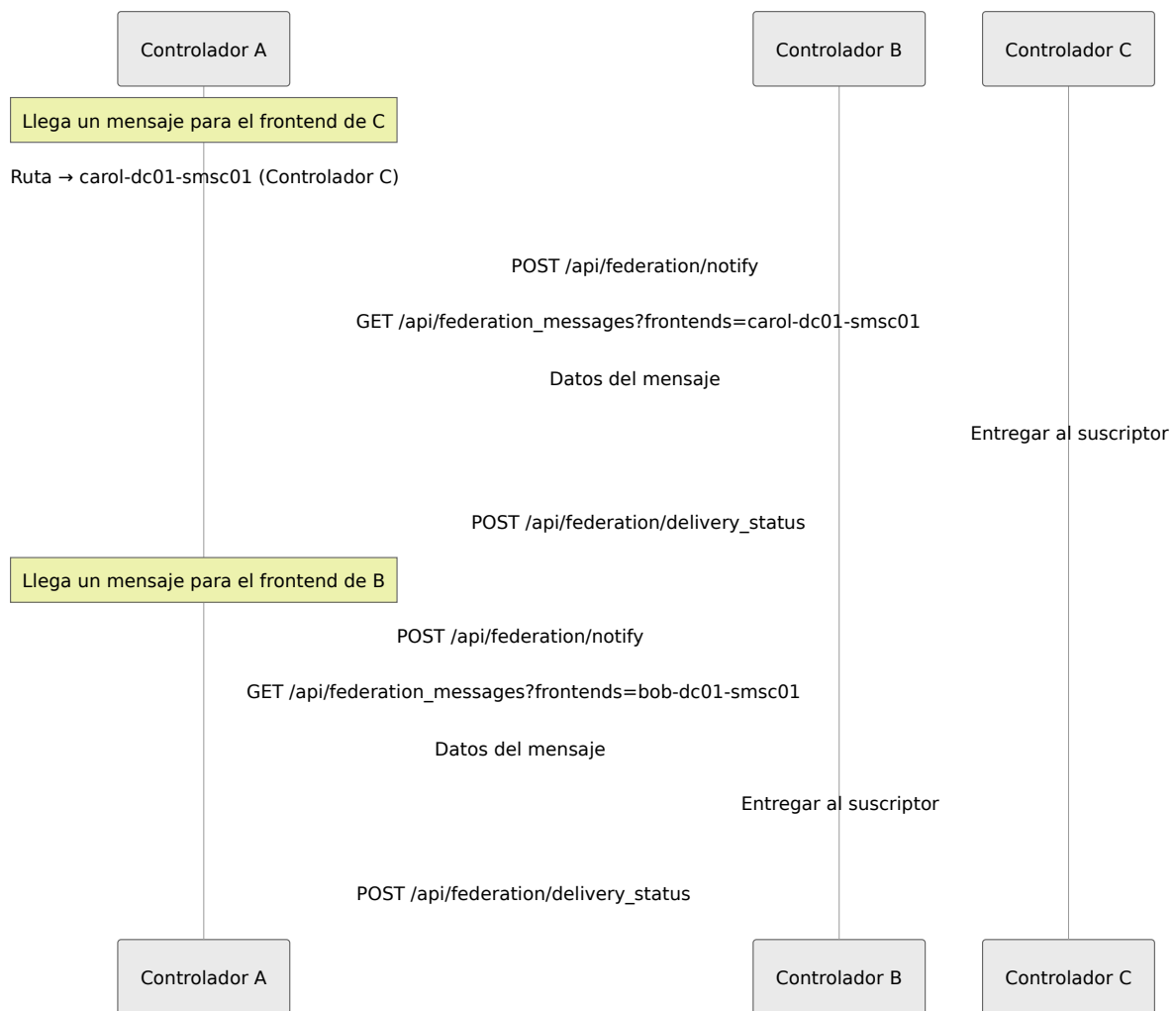
Recuperación de Partición – Múltiples Mensajes en Cola

Durante una partición prolongada, los mensajes se acumulan en el controlador de origen. Cuando se recupera el enlace, el `FederationPoller` se pone al día automáticamente: todos los mensajes pendientes se devuelven en una sola consulta.



Flujo de Mensajes en Tres Sitios

En una malla de múltiples sitios, cada controlador solo sondea mensajes destinados a sus propios frontends.



Salud de Pares y Sincronización de Registros

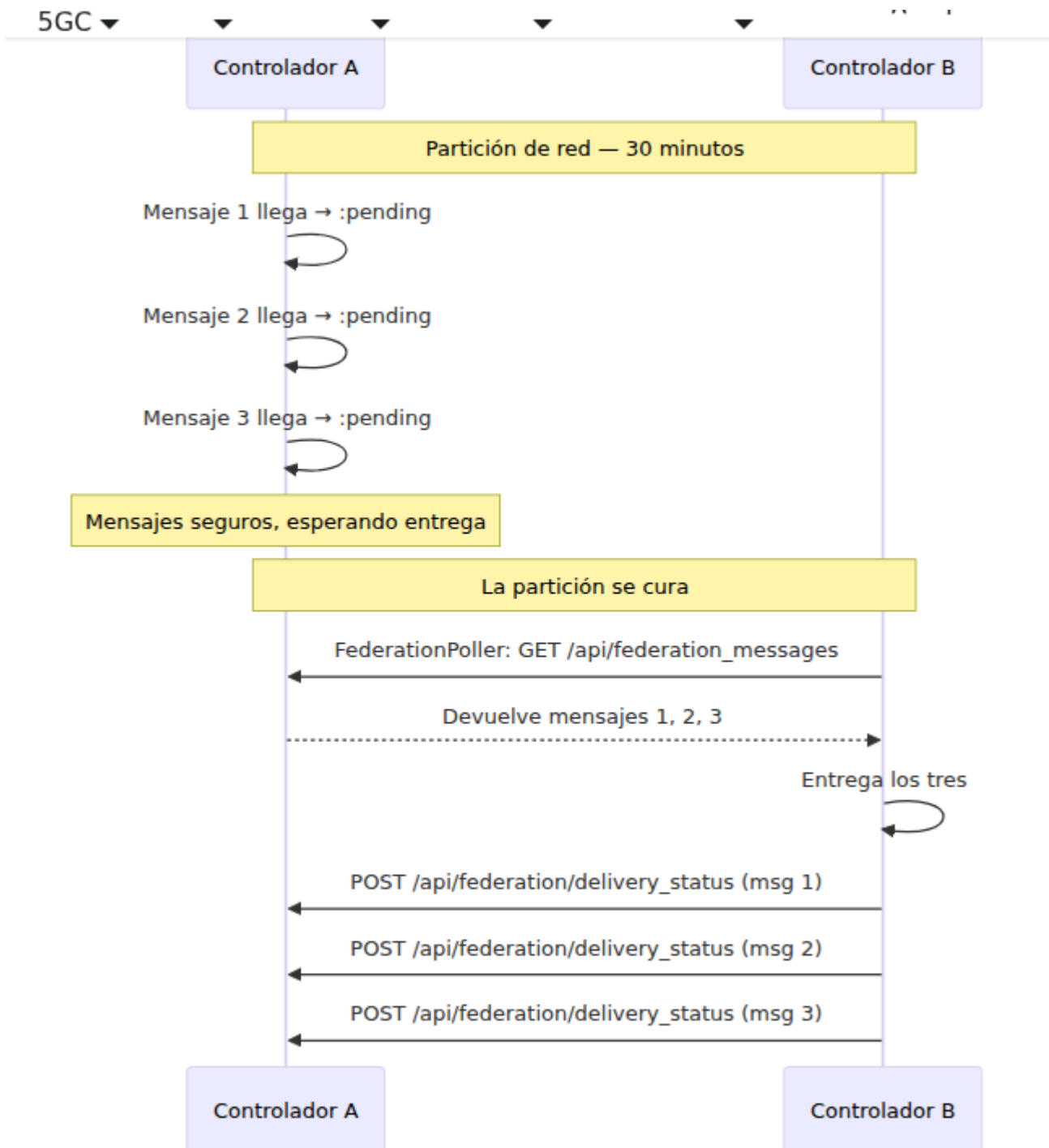
Los controladores de federación mantienen la conciencia entre sí a través de dos ciclos periódicos:

Ciclo de Verificación de Salud

Cada 10 segundos (configurable), cada controlador llama a `POST /api/federation/health` en cada par conocido. La llamada es bidireccional: el llamador envía su propio estado y recibe el estado del par en la respuesta. Un par se marca como **no saludable** después de 3 fallos consecutivos.

Ciclo de Sincronización de Registros

Cada 15 segundos (configurable), cada controlador llama a `POST /api/federation/registry` en cada par saludable. La llamada intercambia listas de frontends: el llamador envía a sus frontends activos y recibe los frontends activos del par. Así es como los controladores aprenden qué par posee qué frontend.



Estados de pares y su efecto en el sondeo de federación:

Estado	Verificaciones de Salud	FederationPoller	Sincronización de Registros
Desconocido	En progreso	No sondeado	No intentado
Saludable	Aprobadas	Sondeo activo	Activo
Degradado	1-2 fallos	No sondeado	No intentado
No Saludable	3+ fallos	No sondeado	No intentado

Configuración

La federación se configura en `config/runtime.exs` bajo la clave `:federation`. Está **desactivada por defecto** y debe habilitarse explícitamente.

Configuración Mínima (Descubrimiento DNS SRV)

```
# config/runtime.exs
config :sms_c, :federation,
  enabled: true,
  dns_srv_domain: "_smc._tcp.smc.example.com"
```

Referencia Completa de Configuración

```
# config/runtime.exs
config :sms_c, :federation,
  # Interruptor maestro – la federación está completamente
  inactiva cuando es falso
  enabled: true,

  # Dominio DNS SRV para descubrimiento de pares (dejar vacío para
  usar static_peers en su lugar)
  dns_srv_domain: "_smc._tcp.smc.example.com",

  # Cada cuánto tiempo volver a resolver registros DNS SRV
  (milisegundos)
  dns_poll_interval_ms: 30_000,

  # Cada cuánto tiempo intercambiar estado de salud con todos los
  pares conocidos (milisegundos)
  health_check_interval_ms: 10_000,

  # Cada cuánto tiempo intercambiar registros de frontends con
  pares saludables (milisegundos)
  registry_sync_interval_ms: 15_000,

  # Cada cuánto tiempo el FederationPoller consulta a pares
  saludables en busca de mensajes (milisegundos)
  poll_interval_ms: 5_000,

  # Tiempo de espera HTTP para todas las llamadas API entre pares
  (milisegundos)
  http_timeout_ms: 5_000,

  # Puerto API utilizado al construir URLs de pares a partir de
  registros DNS SRV
  api_port: 8443,

  # Lista de pares estáticos – utilizada cuando dns_srv_domain
  está vacío
  static_peers: []
```

Parámetros de Federación

Parámetro	Tipo	Requerido	Predeterminado
<code>enabled</code>	Booleano	Sí	<code>false</code>
<code>dns_srv_domain</code>	Cadena	No	<code>""</code>
<code>dns_poll_interval_ms</code>	Entero	No	<code>30000</code>

Parámetro	Tipo	Requerido	Predeterminado
health_check_interval_ms	Entero	No	10000
registry_sync_interval_ms	Entero	No	15000
poll_interval_ms	Entero	No	5000
http_timeout_ms	Entero	No	5000

Parámetro	Tipo	Requerido	Predeterminado
<code>api_port</code>	Entero	No	<code>8443</code>
<code>static_peers</code>	Lista	No	<code>[]</code>

Configuración de Pares Estáticos

Para entornos donde DNS SRV no está disponible, configure los pares explícitamente:

```

config :sms_c, :federation,
  enabled: true,
  dns_srv_domain: "",
  static_peers: [
    %{host: "10.0.1.2", port: 8443},
    %{host: "10.0.2.2", port: 8443},
    %{host: "10.0.3.2", port: 8443}
  ]

```

Parámetro	Tipo	Requerido	Predeterminado	Descripción
host	Cadena	Sí	-	Dirección IP o nombre de host del controlador remoto.
port	Entero	No	8443	Puerto API HTTPS del controlador remoto.

Configuración de Registros DNS SRV

Los registros DNS SRV permiten a los controladores descubrirse automáticamente entre sí. Cada controlador resuelve el dominio configurado y utiliza los pares de host/puerto resultantes como pares.

Configuración de Zona DNS:

```

; Prioridad Peso Puerto Objetivo
_smsc._tcp.smsc.example.com. 86400 IN SRV 10 100 8443 smsc-
alice.smsc.example.com.
_smsc._tcp.smsc.example.com. 86400 IN SRV 10 100 8443 smsc-
bob.smsc.example.com.
_smsc._tcp.smsc.example.com. 86400 IN SRV 10 100 8443 smsc-
carol.smsc.example.com.

; Registros A para los objetivos
smsc-alice.smsc.example.com.      86400 IN A 10.0.1.2
smsc-bob.smsc.example.com.       86400 IN A 10.0.2.2
smsc-carol.smsc.example.com.     86400 IN A 10.0.3.2

```

Agregar un nuevo sitio: Agregue un nuevo registro SRV a DNS. Todos los controladores existentes descubrirán el nuevo par dentro de un ciclo de `dns_poll_interval_ms` (predeterminado 30 segundos).

Eliminar un sitio: Elimine el registro SRV de DNS. Los controladores existentes eliminarán el par salido dentro de un ciclo de sondeo.

Requisitos de Red

La federación requiere solo un único puerto HTTPS entre sitios.

Puerto	Protocolo	Dirección	Propósito
8443	TCP/TLS	Bidireccional	Todo el tráfico de federación (salud, registro, notificación, mensajes, estado de entrega)
53	UDP/TCP	Saliente	Resolución DNS SRV (si se utiliza descubrimiento DNS)

Reglas de firewall (por sitio):

```
# Permitir tráfico de federación desde las IPs de los controladores pares
iptables -A INPUT -p tcp -s 10.0.1.0/24 --dport 8443 -j ACCEPT
iptables -A INPUT -p tcp -s 10.0.2.0/24 --dport 8443 -j ACCEPT
iptables -A INPUT -p tcp -s 10.0.3.0/24 --dport 8443 -j ACCEPT
```

Puntos de Acceso de la API de Federación

Todos los puntos finales de federación se sirven en el puerto API estándar (8443) bajo `/api/federation/`. Estos puntos finales son llamados por controladores pares, no por clientes externos o frontends.

Punto final	Método	Propósito
<code>/api/federation/identity</code>	GET	Devuelve el nombre del nodo de este controlador y la versión del protocolo
<code>/api/federation/health</code>	POST	Intercambio de salud bidireccional — el llamador envía estado, el llamado responde con el propio
<code>/api/federation/registry</code>	POST	Intercambio bidireccional de registros de frontends
<code>/api/federation/notify</code>	POST	Recibir notificación ligera de que un mensaje está disponible para un frontend local
<code>/api/federation/delivery_status</code>	POST	Recibir confirmación de entrega del controlador de destino
<code>/api/federation_messages</code>	GET	Devuelve mensajes pendientes para los frontends especificados (utilizado por FederationPoller)

Todas las solicitudes y respuestas utilizan JSON. La identificación del par se transporta en el encabezado `X-Federation-Node`.

Ejemplos de Implementación

Ejemplo 1: Implementación de Tres Sitios con DNS SRV

Tres centros de datos, cada uno con frontends locales para su base de suscriptores. DNS SRV proporciona descubrimiento.

```
# Sitio A – DC Alice (config/runtime.exs)
config :sms_c,
  smsc_node_name: "alice-dc01-smsc01"

config :sms_c, :federation,
  enabled: true,
  dns_srv_domain: "_smsc._tcp.smsc.example.com"
```

```
# Sitio B – DC Bob (config/runtime.exs)
config :sms_c,
  smsc_node_name: "bob-dc01-smsc01"

config :sms_c, :federation,
  enabled: true,
  dns_srv_domain: "_smsc._tcp.smsc.example.com"
```

```
# Sitio C – DC Carol (config/runtime.exs)
config :sms_c,
  smsc_node_name: "carol-dc01-smsc01"

config :sms_c, :federation,
  enabled: true,
  dns_srv_domain: "_smsc._tcp.smsc.example.com"
```

Cómo funciona: Los tres controladores resuelven el mismo dominio SRV y se descubren automáticamente. Cada controlador registra sus frontends locales. Cuando un mensaje llega a DC Alice destinado a un suscriptor atendido por DC Bob, la tabla de enrutamiento selecciona `bob-dc01-smsc01` como destino. La capa de federación detecta que este frontend pertenece al Controlador B y

envía una notificación. El FederationPoller del Controlador B consulta al Controlador A y almacena el mensaje en caché para la entrega local.

Ejemplo 2: Dos Sitios Activo-Activo con Pares Estáticos

Dos centros de datos conectados por un enlace dedicado. No hay DNS SRV disponible.

```
# DC Primario (config/runtime.exs)
config :sms_c,
  smsc_node_name: "primary-smsc01"

config :sms_c, :federation,
  enabled: true,
  dns_srv_domain: "",
  static_peers: [
    %{host: "10.200.1.5", port: 8443}
  ]
```

```
# DC Secundario (config/runtime.exs)
config :sms_c,
  smsc_node_name: "secondary-smsc01"

config :sms_c, :federation,
  enabled: true,
  dns_srv_domain: "",
  static_peers: [
    %{host: "10.100.1.5", port: 8443}
  ]
```

Cómo funciona: Cada controlador está configurado con la dirección IP del otro. Las verificaciones de salud y las sincronizaciones de registros se realizan a través del enlace dedicado. Los mensajes permanecen en el controlador que los recibió. El FederationPoller de cada controlador consulta al par en busca de mensajes destinados a los frontends locales.

Ejemplo 3: Sondeo Agresivo para Enlaces de Baja Latencia

Para sitios conectados por enlaces fiables y de baja latencia donde la rápida federación es importante:

```
config :sms_c, :federation,  
  enabled: true,  
  dns_srv_domain: "_smc.tcp.cluster.internal",  
  health_check_interval_ms: 5_000,  
  registry_sync_interval_ms: 5_000,  
  poll_interval_ms: 1_000,  
  http_timeout_ms: 2_000
```

Caso de uso: Implementaciones en campus o áreas metropolitanas donde los sitios están conectados por enlaces de < 10 ms RTT y la rápida conmutación por error es más importante que minimizar el tráfico de control.

Métricas

La federación expone los siguientes eventos de telemetría que pueden ser observados a través de Prometheus en el puerto 9568.

Métrica: `sms_c_federation_message_received_count` **Tipo:** Contador

Descripción: Número de notificaciones de federación recibidas de pares

Etiquetas:

- `origin_node` - Par que envió la notificación

Métrica: `sms_c_federation_health_check_count` **Tipo:** Contador

Descripción: Resultados de verificaciones de salud por par **Etiquetas:**

- `peer` - Identificador del par
- `result` - `ok` o `failed`

Consultas de ejemplo:

```
# Tasa de notificación por minuto
rate(sms_c_federation_message_received_count[5m]) * 60

# Fallos en verificaciones de salud por par
sum by (peer)
(rate(sms_c_federation_health_check_count{result="failed"}[5m]))
```

Indicadores Clave a Monitorear

Qué Observar	Dónde	Umbral de Alerta
Mensajes federados en caché	GET /api/federation/status o Interfaz Web	Conteo creciente (par no entregando)
Estado de salud de pares	GET /api/federation/identity en cada par	Cualquier par no saludable > 5 minutos
Latencia de verificación de salud	Registros de aplicación	> 2 segundos (degradación del enlace)

Solución de Problemas

Par No Descubierta

Síntomas: Los registros del controlador muestran que no se han descubierta pares; `cluster_status` devuelve vacío.

Causas posibles:

- El dominio DNS SRV es incorrecto o no resoluble
- El servidor DNS es inaccesible desde el host del controlador
- Los registros SRV aún no se han propagado

- La federación no está habilitada (`enabled: false`)

Resolución:

1. Verifique la resolución DNS SRV desde el host del controlador: `dig SRV _smc._tcp.smc.example.com`
2. Verifique que `enabled: true` esté configurado en la configuración de federación
3. Revise los registros de la aplicación para mensajes de "Descubrimiento DNS de Federaciones" al inicio
4. Si se utilizan pares estáticos, verifique que la lista `static_peers` contenga entradas correctas de host/puerto

Par Marcado como No Saludable

Síntomas: Los mensajes a un sitio específico no están siendo sondeados. Los registros muestran mensajes de "Verificación de salud fallida".

Causas posibles:

- El controlador remoto está inactivo
- Firewall bloqueando el puerto 8443 entre sitios
- Problemas con el certificado TLS
- Partición de red entre sitios

Resolución:

1. Verifique que el controlador remoto esté en funcionamiento: `curl -k https://<peer-host>:8443/api/federation/identity`
2. Verifique que las reglas del firewall permitan tráfico en el puerto 8443 desde la IP del controlador local
3. Revise los registros de la aplicación en ambos lados en busca de errores de TLS o conexión
4. Verifique la conectividad de red entre sitios

Mensajes No Entregados al Sitio Remoto

Síntomas: Los mensajes permanecen `:pending` en el controlador de origen. El conteo de mensajes federados en caché es 0 en el destino.

Causas posibles:

- El controlador de destino está no saludable (el FederationPoller solo consulta pares saludables)
- El FederationPoller no está en funcionamiento
- El registro de frontends no está sincronizado (el destino no sabe qué frontends son locales)

Resolución:

1. Verifique el estado del par en el controlador de destino
2. Verifique que el FederationPoller esté en el árbol de supervisión (revise los registros de la aplicación para "Federation poller started")
3. Verifique que los registros de frontends estén sincronizados (`GET /api/federation/status` — verifique las listas de frontends)
4. Pruebe manualmente el punto final `federation_messages`: `curl -k 'https://<origin>:8443/api/federation_messages?frontends=<frontend_name>&include_unrouted=false'`

Búsqueda de Suscriptores HSS (Diameter Sh)

[← Volver al Índice de Documentación](#) | [Guía de Enrutamiento de SMS](#)

Resumen

OmniMessage puede consultar el Servidor de Suscriptores en el Hogar (HSS) a través de la interfaz Diameter Sh para determinar si un suscriptor llamado pertenece a la red antes de tomar una decisión de enrutamiento. Esto evita que los mensajes destinados a suscriptores en la red sean enrutados incorrectamente a través de una puerta de enlace predeterminada cuando el suscriptor está temporalmente fuera de línea o no registrado.

Sin la búsqueda de HSS habilitada, un mensaje a un suscriptor en la red que no está actualmente registrado caerá en la tabla de enrutamiento estándar. Si existe una ruta de captura o predeterminada, el mensaje se enviará a una puerta de enlace externa, lo cual es incorrecto para un suscriptor que pertenece a la red.

Con la búsqueda de HSS habilitada, OmniMessage envía un **User-Data-Request (UDR)** al HSS para el número llamado. Si el HSS confirma que el suscriptor existe (código de resultado 2001), el mensaje se mantiene en la cola de mensajes para entrega de almacenamiento y reenvío cuando el suscriptor se registre nuevamente. Si el HSS devuelve "usuario desconocido" (código de resultado 5001), el suscriptor está fuera de la red y el enrutamiento estándar procede normalmente.

Especificaciones Relevantes

Especificación	Título
3GPP TS 29.329	Interfaz Sh basada en Diameter
3GPP TS 29.328	Interfaz IMS Sh — flujos de señalización y contenidos de mensajes
RFC 6733	Protocolo Base Diameter

Cadena de Prioridad de Enrutamiento

La búsqueda de HSS se integra en la decisión de enrutamiento de OmniMessage como **Prioridad 2**, entre la verificación de ubicación/registro y la tabla de enrutamiento basada en regex.

Mensaje recibido después de la traducción del número

OmniCore 5GC

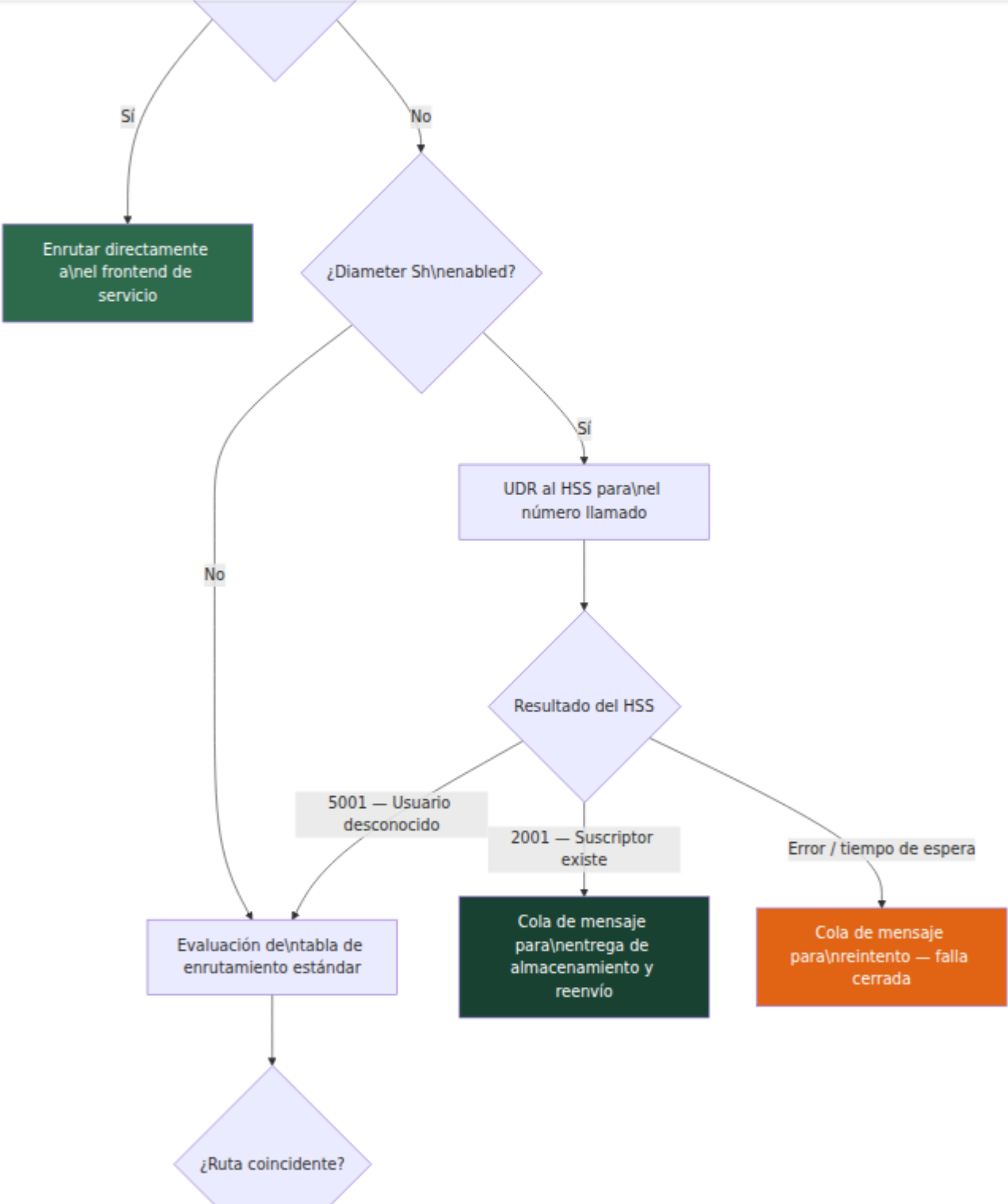
OmniCall

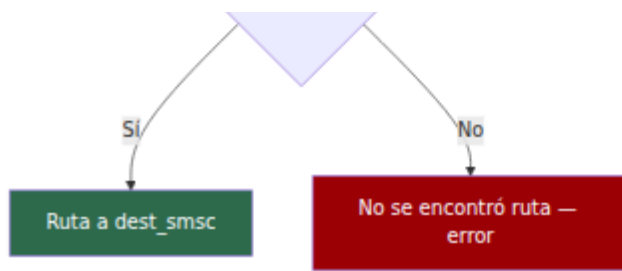
OmniRAN

OmniCharge

Platform

🇪🇸 Español





Prioridad 1 — Enrutamiento Basado en Ubicación

Si el MSISDN de destino se encuentra en la tabla de registros/ubicaciones, el mensaje se enruta directamente al frontend que atiende a ese suscriptor. No se realiza búsqueda de HSS ni evaluación de la tabla de enrutamiento.

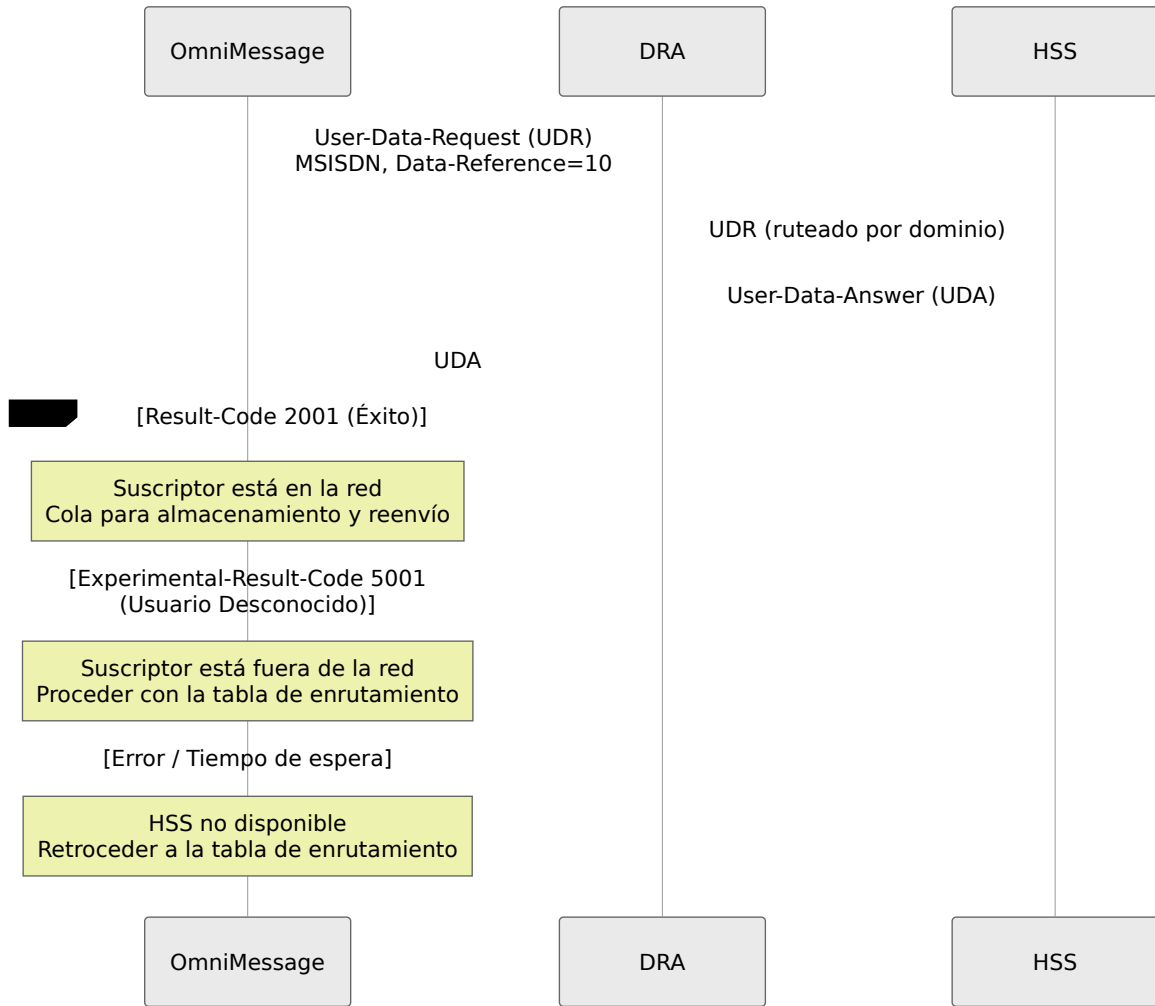
Prioridad 2 — Búsqueda de Suscriptores HSS

Si no se encuentra un registro y Diameter Sh está habilitado, OmniMessage consulta al HSS para determinar si el suscriptor está en la red. Un suscriptor en la red que simplemente está fuera de línea tendrá su mensaje en cola; se entregará cuando el suscriptor se registre y llegue una actualización de ubicación.

Prioridad 3 — Tabla de Enrutamiento

Si el HSS confirma que el suscriptor es desconocido (fuera de la red), o si Diameter no está habilitado, se evalúa la tabla de enrutamiento estándar basada en regex. Consulte la [Guía de Enrutamiento de SMS](#) para más detalles.

Flujo de Mensajes Diameter Sh



Parámetros UDR

AVP	Valor	Descripción
Session-Id	Generado automáticamente	Identificador de sesión único
Auth-Session-State	1 (NO_STATE_MAINTAINED)	No se requiere estado de sesión para consultas
Data-Reference	10	Solicitar todos los datos del usuario
Origin-Host	Host configurado + dominio	Identidad Diameter de OmniMessage
Origin-Realm	Dominio configurado	Dominio Diameter de OmniMessage
Destination-Realm	Dominio configurado	Dominio HSS (ruteado a través de DRA)
User-Identity	MSISDN codificado en TBCD	El número del suscriptor llamado
Server-Name	"OmniMessage"	Identidad del servidor de aplicaciones
Vendor-Specific-Application-Id	Auth: 16777217, Vendor: 10415	Identificador de aplicación Sh según 3GPP TS 29.329

Códigos de Resultado UDA

Código	Tipo	Significado	Acción de OmniMessage
2001	Result-Code	Éxito — suscriptor encontrado	Cola de mensaje para almacenamiento y reenvío
5001	Experimental-Result-Code	Usuario desconocido — no en HSS	Proceder con la tabla de enrutamiento
Otro	Experimental-Result-Code	Error inesperado	Registrar advertencia, cola para reintento (falla cerrada)

Configuración

Habilitando la Búsqueda de HSS

La búsqueda de HSS se controla mediante dos bloques de configuración en `config/runtime.exs`:

1. La bandera `diameter_enabled` bajo `:sms_c` — controla si se realiza la consulta HSS en la ruta de enrutamiento
2. La configuración `:diameter_ex` — configura la pila Diameter en sí (identidad, pares, aplicaciones)

Configuración Mínima

```
# Habilitar consulta HSS en la ruta de enrutamiento
config :sms_c,
  diameter_enabled: true

# Configuración de la pila Diameter
config :diameter_ex,
  diameter: %{
    service_name: :omnimessage,
    listen_ip: "0.0.0.0",
    listen_port: 3868,
    decode_format: :map,
    host: "smsc01",
    realm: "epc.mnc005.mcc547.3gppnetwork.org",
    product_name: "OmniMessage",
    request_timeout: 5000,
    peer_selection_algorithm: :random,
    allow_undefined_peers_to_connect: true,
    log_unauthorized_peer_connection_attempts: true,
    control_module: SmsC.Diameter.Control,
    processor_module: DiameterEx.Processor,
    auth_application_ids: [],
    acct_application_ids: [],
    vendor_id: 10415,
    supported_vendor_ids: [10415],
    applications: [
      %{
        application_name: :sh,
        application_dictionary: :diameter_gen_3gpp_sh,
        vendor_specific_application_ids: [
          %{
            vendor_id: 10415,
            auth_application_id: 16_777_217,
            acct_application_id: nil
          }
        ]
      }
    ],
    peers: [
      %{
        port: 3868,
        host: "dra01.epc.mnc005.mcc547.3gppnetwork.org",
```

```
    ip: "10.0.0.1",
    realm: "epc.mnc005.mcc547.3gppnetwork.org",
    tls: false,
    transport: :diameter_tcp,
    initiate_connection: true
  }
]
}
```

Parámetros de OmniMessage

Parámetro	Tipo	Requerido	Predeterminado	Descripción
<code>diameter_enabled</code>	Booleano	No	<code>false</code>	Indica si se debe habilitar el protocolo Diameter.
<code>mock_sh</code>	Booleano	No	<code>false</code>	Indica si se debe utilizar un servidor de prueba (Mock) para el protocolo SIP.
<code>mock_sh_on_net_numbers</code>	Lista	No	<code>[]</code>	Lista de números de red para los cuales se debe utilizar el servidor de prueba.

Parámetro	Tipo	Requerido	Predeterminado	D
				m t

Parámetros de la Pila Diameter

Parámetro	Tipo	Requerido	F
<code>service_name</code>	Átomo	Sí	-
<code>listen_ip</code>	Cadena	No	'
<code>listen_port</code>	Entero	No	3
<code>decode_format</code>	Átomo	Sí	-
<code>host</code>	Cadena	Sí	-
<code>realm</code>	Cadena	Sí	-
<code>product_name</code>	Cadena	No	-
<code>request_timeout</code>	Entero	No	5
<code>peer_selection_algorithm</code>	Átomo	No	:

Parámetro	Tipo	Requerido	F
<code>allow_undefined_peers_to_connect</code>	Booleano	No	t
<code>log_unauthorized_peer_connection_attempts</code>	Booleano	No	t
<code>control_module</code>	Módulo	Sí	-
<code>processor_module</code>	Módulo	Sí	-
<code>auth_application_ids</code>	Lista	No	[
<code>acct_application_ids</code>	Lista	No	[
<code>vendor_id</code>	Entero	Sí	-
<code>supported_vendor_ids</code>	Lista	Sí	-

Parámetros de Aplicación

La lista `applications` configura qué aplicaciones Diameter están habilitadas. Para la búsqueda de HSS, solo se requiere la aplicación Sh.

Parámetro	Tipo	Requerido	Predetermina
<code>application_name</code>	Átomo	Sí	-
<code>application_dictionary</code>	Átomo	Sí	-
<code>vendor_specific_application_ids</code>	Lista	Sí	-

ID de Aplicación Específica del Proveedor (Sh):

Parámetro	Tipo	Valor	Descripción
<code>vendor_id</code>	Entero	<code>10415</code>	ID de proveedor 3GPP
<code>auth_application_id</code>	Entero	<code>16_777_217</code>	ID de aplicación Sh según 3GPP TS 29.329
<code>acct_application_id</code>	-	<code>nil</code>	No utilizado para Sh

Parámetros de Pares

La lista `peers` define los pares Diameter (típicamente DRAs o el HSS directamente) a los que se conecta OmniMessage.

Parámetro	Tipo	Requerido	Predeterminado	De
<code>host</code>	Cadena	Sí	-	Identificadora de red (FQDN) que coincide exactamente con la configuración.
<code>ip</code>	Cadena	Sí	-	Dirección IP para la conexión.
<code>port</code>	Entero	No	3868	Puerto de destino.
<code>realm</code>	Cadena	Sí	-	Nombre de dominio para la conexión.
<code>tls</code>	Booleano	No	<code>false</code>	Habilitado para la conexión.
<code>transport</code>	Átomo	No	<code>:diameter_tcp</code>	Protocolo de transporte: <code>:diameter_tcp</code> o <code>:diameter_udp</code> .
<code>initiate_connection</code>	Booleano	No	<code>true</code>	Si se debe iniciar la conexión por defecto.

Parámetro	Tipo	Requerido	Predeterminado	De
				OmniMessage para s

Múltiples Pares

Configure múltiples pares para redundancia. OmniMessage selecciona entre los pares conectados utilizando el `peer_selection_algorithm`:

```
peers: [
  %{
    port: 3868,
    host: "dra01.epc.mnc005.mcc547.3gppnetwork.org",
    ip: "10.0.0.1",
    realm: "epc.mnc005.mcc547.3gppnetwork.org",
    tls: false,
    transport: :diameter_tcp,
    initiate_connection: true
  },
  %{
    port: 3868,
    host: "dra02.epc.mnc005.mcc547.3gppnetwork.org",
    ip: "10.0.0.2",
    realm: "epc.mnc005.mcc547.3gppnetwork.org",
    tls: false,
    transport: :diameter_tcp,
    initiate_connection: true
  }
]
```

Modo Simulado

Para entornos de prueba sin un HSS en vivo, habilite el modo simulado:

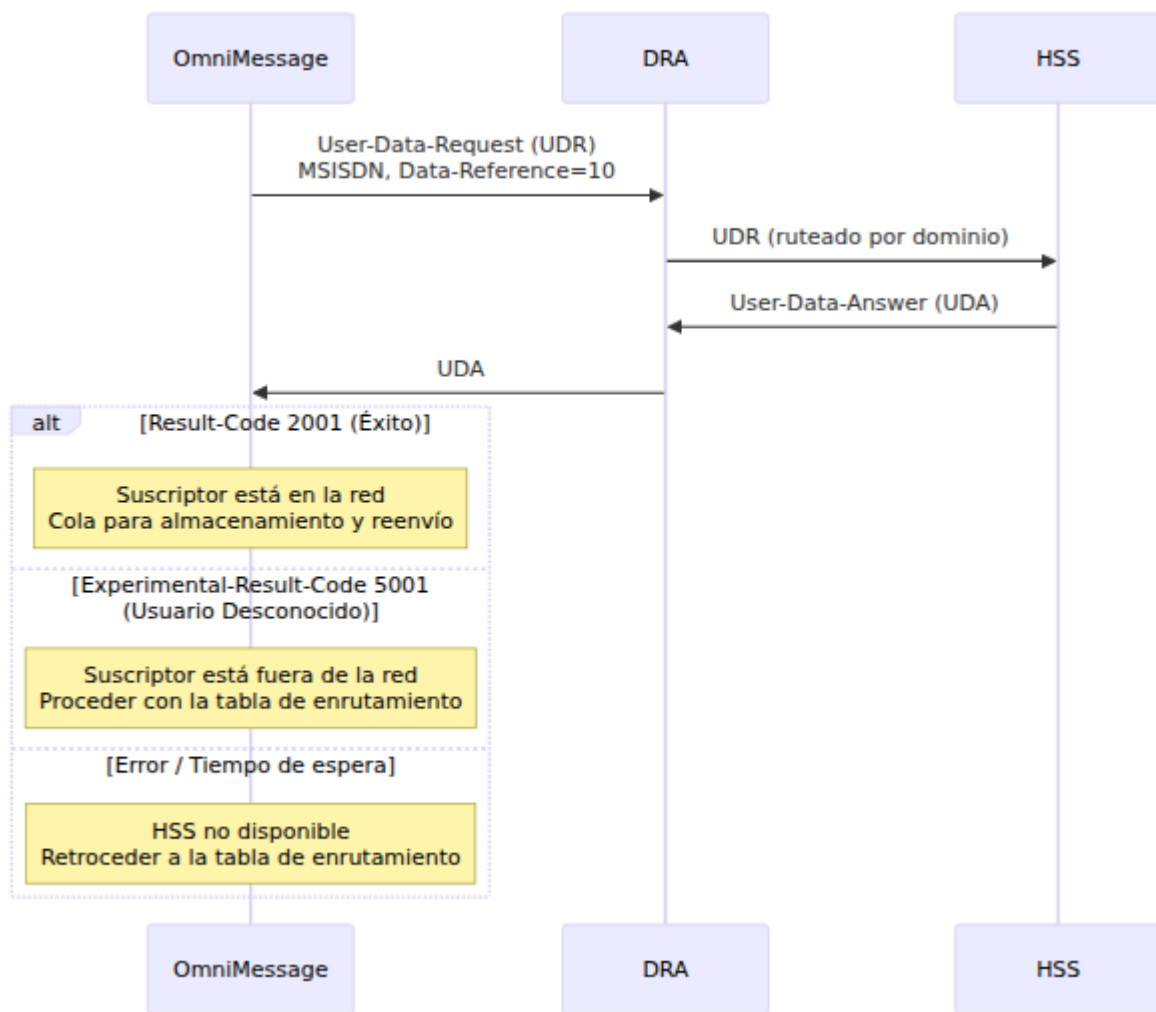
```
config :sms_c,  
  diameter_enabled: true,  
  mock_sh: true,  
  mock_sh_on_net_numbers: [  
    "68987654321",  
    "68912345678"  
  ]
```

En modo simulado, cualquier MSISDN en la lista `mock_sh_on_net_numbers` se trata como un suscriptor en la red (equivalente al código de resultado HSS 2001). Todos los demás MSISDN se tratan como desconocidos (equivalente al código de resultado 5001). No se inicia ninguna pila Diameter y no se realizan conexiones de red.

Entrega de Almacenamiento y Reenvío

Cuando un mensaje está en cola para un suscriptor en la red que no está registrado, el mensaje permanece en la cola de mensajes de Mnesia con `dest_smsc` configurado como `nil`. Cuando el suscriptor se registra y llega una actualización de ubicación, el bucle de entrega recoge los mensajes pendientes para ese MSISDN y los enruta al frontend de servicio.

Los mensajes en la cola están sujetos a la política de retención estándar. Si el suscriptor no se registra antes de que el mensaje expire, el mensaje se mueve a la cola de cartas muertas según el `dead_letter_time_minutes` configurado.



Notas Operativas

Registro de Inicio

OmniMessage registra el estado de Diameter Sh al iniciar:

- **Habilitado:** Diameter Sh (HSS): HABILITADO (host: smsc01, realm: epc.mnc005.mcc547.3gppnetwork.org, peers: 2)
- **Deshabilitado:** Diameter Sh (HSS): DESHABILITADO

HSS Inalcanzable (Falla Cerrada)

Si la búsqueda de HSS falla por cualquier motivo — tiempo de espera, sin pares conectados, respuesta no manejada — OmniMessage registra una advertencia y **cola el mensaje para reintento**. El mensaje permanece con `dest_smsc`

configurado como `nil` y se reevaluará en el siguiente ciclo de sondeo, momento en el cual se intentará nuevamente la consulta HSS.

Este es un diseño de **falla cerrada**: sin una respuesta definitiva del HSS, OmniMessage no enruta el mensaje a una puerta de enlace fuera de la red. Esto previene que los suscriptores en la red tengan sus mensajes enviados incorrectamente a transportistas externos durante las interrupciones del HSS.

Cuando **Diameter está explícitamente deshabilitado** (`diameter_enabled: false`), los mensajes se enrutan a través de la tabla de enrutamiento estándar sin ninguna verificación de HSS. Esta es una elección intencional del operador: el comportamiento de falla cerrada solo se aplica cuando Diameter está habilitado pero la búsqueda falla.

Entradas de Registro de Eventos

Los siguientes tipos de eventos se registran en el registro de eventos de mensajes cuando la búsqueda de HSS está activa:

Tipo de Evento	Descripción
<code>hss_dip_on_net</code>	Suscriptor encontrado en HSS (código de resultado 2001). Mensaje en cola para almacenamiento y reenvío.
<code>hss_dip_off_net</code>	Suscriptor no encontrado en HSS (código de resultado 5001). Procediendo con el enrutamiento estándar.
<code>hss_dip_error</code>	La búsqueda de HSS falló (tiempo de espera, sin conexión, etc.). Mensaje en cola para reintento.

Todos los eventos incluyen metadatos estructurados: `result_code`, `hss` (nombre de host del par), `duration_ms`, `session_id`, `destination_msisdn`, y `message_id`.

Los eventos de enrutamiento estándar (`sms_routing_started`, `sms_routing_selected`, etc.) continúan registrándose para los mensajes que proceden a la tabla de enrutamiento.

Aplicación de Requisitos en la Red a Nivel de Ruta

Además de la consulta HSS de Prioridad 2 (que se aplica globalmente antes del enrutamiento), las rutas individuales pueden aplicar requisitos en la red utilizando tres banderas. Estas se verifican **después** de que una ruta coincide y **antes** de que el mensaje se asigne a `dest_smsc` de esa ruta.

Banderas de Ruta

Bandera	Descripción	Verificación HSS	Comportamiento
<code>on_net_only</code>	Alias obsoleto para <code>terminating_on_net_only</code>	Verifica el número llamado	Descarta el mensaje
<code>originating_on_net_only</code>	Requiere que el remitente (parte que llama) esté en la red	Verifica el número de llamada	Descarta el mensaje
<code>terminating_on_net_only</code>	Requiere que el destino (parte llamada) esté en la red	Verifica el número llamado	Descarta el mensaje

Las tres banderas **falla cerrada**: si el HSS es inalcanzable o Diameter está deshabilitado, el mensaje se descarta en lugar de entregarse a un destino no verificado.

Ejemplo de Configuración

```
config :sms_c, :sms_routes, [  
  # Puerta de enlace SMPP en la red – solo acepta mensajes para  
  destinos confirmados en la red  
  %{  
    calling_regex: nil,  
    called_regex: ~r/^1907/,  
    source_smsc: nil,  
    dest_smsc: "local-gateway",  
    source_type: nil,  
    auto_reply: false,  
    drop: false,  
    charged: :default,  
    weight: 100,  
    priority: 10,  
    description: "Puerta de enlace local – solo destinos en la  
red",  
    on_net_only: false,  
    originating_on_net_only: false,  
    terminating_on_net_only: true,  
    enabled: true  
  },  
  
  # Recaída fuera de la red – captura todo para suscriptores  
  confirmados fuera de la red por HSS  
  %{  
    calling_regex: nil,  
    called_regex: nil,  
    source_smsc: nil,  
    dest_smsc: "external-gateway",  
    source_type: :ims,  
    auto_reply: false,  
    drop: false,  
    charged: :default,  
    weight: 100,  
    priority: 255,  
    description: "Puerta de enlace externa – recaída fuera de la  
red",  
    on_net_only: false,  
    originating_on_net_only: false,  
    terminating_on_net_only: false,  
    enabled: true  
  }  
]
```

```
}  
]
```

Cómo funciona: Cuando llega un mensaje para un suscriptor que no está actualmente registrado:

1. **Verificación de ubicación** — no se encontró registro activo
2. **Consulta HSS** — suscriptor confirmado en la red → mensaje en cola para almacenamiento y reenvío (sin enrutamiento)
3. **Consulta HSS** — suscriptor confirmado fuera de la red → se evalúa la tabla de rutas
4. La ruta "local-gateway" coincide pero tiene `terminating_on_net_only: true` — HSS ya confirmó fuera de la red, por lo que esta ruta se omite (mensaje descartado en esta ruta)
5. La ruta "external-gateway" coincide sin requisito en la red → mensaje enrutado a la puerta de enlace externa

Caso de uso: Prevenir que el tráfico fuera de la red sea enviado a la infraestructura local que solo puede atender a suscriptores en la red, mientras se asegura que el tráfico fuera de la red llegue a un transportista externo.

Interacción con la Consulta HSS de Prioridad 2

La consulta HSS de Prioridad 2 y las banderas de nivel de ruta en la red sirven para diferentes propósitos:

Mecanismo	Cuándo	Propósito	Comportamiento en Caso de
Consulta HSS de Prioridad 2	Antes del enrutamiento	Cola a los suscriptores en la red para almacenamiento y reenvío	Falla cerrada para reintento
<code>terminating_on_net_only</code>	Después de la coincidencia de ruta	Aplicar requisito en la red en rutas específicas	Falla cerrada (descartar mensaje)
<code>originating_on_net_only</code>	Después de la coincidencia de ruta	Aplicar requisito de origen en la red	Falla cerrada (descartar mensaje)

Expiración de Ubicación

El enrutamiento basado en ubicación (Prioridad 1) verifica la marca de tiempo `expires` en los registros de suscriptores. Si el registro de un suscriptor ha expirado — por ejemplo, entró en modo avión — la ubicación se trata como inactiva y el enrutamiento procede a la Prioridad 2 (consulta HSS) y la Prioridad 3 (tabla de enrutamiento).

Esto previene que los mensajes sean enviados a un frontend que ya no puede alcanzar al suscriptor, evitando fallos de entrega innecesarios y ciclos de reintento.

Solución de Problemas

Mensajes que Aún se Enrutan a la Puerta de Enlace Predeterminada para Suscriptores en la

Red

Síntomas: Mensajes a suscriptores en la red que están fuera de línea se envían a la puerta de enlace predeterminada en lugar de ser enrutados.

Causas posibles:

- `diameter_enabled` está configurado como `false`
- No hay pares Diameter conectados (verifique los registros de la aplicación para el estado de conexión de pares)
- HSS está devolviendo errores o tiempo de espera (verifique los registros para advertencias de "consulta HSS fallida")
- La traducción de números está produciendo un formato de número diferente al que espera el HSS

Resolución:

1. Verifique `diameter_enabled: true` en `runtime.exs`
2. Verifique los registros para el estado de conexión de pares Diameter al iniciar
3. Busque advertencias de "consulta HSS fallida" en el registro de la aplicación
4. Confirme que el formato del número llamado después de la traducción coincida con lo que espera el HSS (E.164 sin +)

Mensajes en Cola Pero Nunca Entregados

Síntomas: Los mensajes se encolan correctamente para suscriptores en la red pero nunca se entregan, incluso después de que el suscriptor se registra.

Causas posibles:

- Las actualizaciones de ubicación no están llegando a OmniMessage
- El MSISDN en el registro de ubicación no coincide con el `destination_msisdn` del mensaje en cola
- Los mensajes han expirado y se han movido a la cola de cartas muertas

Resolución:

1. Verifique que se estén recibiendo actualizaciones de ubicación/registro (verifique la página de Registros en el Panel de Control)
2. Confirme que el formato de MSISDN sea consistente entre la cola de mensajes y la tabla de registros
3. Verifique `dead_letter_time_minutes` — aumente si los suscriptores pueden estar fuera de línea por períodos prolongados

Par Diameter No Conectándose

Síntomas: Los registros no muestran conexiones de pares Diameter al iniciar.

Causas posibles:

- IP o puerto de par incorrectos en la configuración
- Cortafuegos bloqueando el puerto 3868
- Desajuste en la identidad del host del par (el `host` configurado debe coincidir exactamente con la identidad Diameter del par)
- Desajuste de dominio entre OmniMessage y el par

Resolución:

1. Verifique que la dirección IP y el puerto del par sean correctos
2. Confirme que las reglas del cortafuegos permitan tráfico TCP en el puerto configurado
3. Verifique que el valor de `host` del par coincida exactamente con lo que el par anuncia en su intercambio de capacidades
4. Asegúrese de que el `realm` coincida entre las configuraciones de OmniMessage y del par

Documentación de Métricas Prometheus de SMS-C

[← Volver al Índice de Documentación](#) | [README Principal](#)

Descripción General

Este documento describe todas las métricas de Prometheus expuestas por el sistema SMS-C. Estas métricas están diseñadas para que el personal de operaciones monitoree la salud del sistema, el rendimiento y solucione problemas.

Acceso a Métricas

El endpoint de métricas de Prometheus está disponible en:

```
http://localhost:9568/metrics
```

Este endpoint expone métricas en formato de texto de Prometheus que pueden ser recopiladas por un servidor de Prometheus. Las métricas se actualizan en tiempo real a medida que el sistema procesa mensajes.

Convención de Nombres de Métricas

Todas las métricas siguen el patrón: `sms_c.<categoría>.<nombre_métrica>.<tipo>`

Categorías:

- `license` - Métricas de estado de licencia
- `message` - Métricas de procesamiento de mensajes
- `routing` - Métricas de decisiones de enrutamiento
- `enum` - Métricas de búsqueda ENUM/NAPTR
- `delivery` - Métricas de entrega de mensajes
- `queue` - Métricas de gestión de colas
- `charging` - Métricas de facturación/cobro
- `mnesia` - Métricas de base de datos
- `frontend` - Métricas de conexión de frontend
- `location` - Métricas de ubicación/registro
- `phoenix.endpoint` - Métricas de solicitudes de API HTTP
- `vm` - Métricas del sistema Erlang VM

Métricas de Licencia

`sms_c_license_status`

Tipo: Gauge

Descripción: Estado actual de la licencia del sistema SMS-C de OmniMessage.

Valores:

- `1` - Licencia válida
- `0` - Licencia inválida/expirada

Etiquetas: Ninguna

Nombre del Producto: `omnimessage`

Caso de Uso: Monitorear la validez de la licencia para asegurar que el sistema esté operando con una licencia válida. Cuando es inválida, los mensajes aún se reciben pero se enrutan al destino "NOLICENCE" en lugar de la ruta normal.

Comportamiento Cuando la Licencia es Inválida:

- Los mensajes entrantes son **aceptados** y almacenados
- El destino del mensaje (`dest_smsc`) se **configura automáticamente como "NOLICENCE"**
- El enrutamiento normal es **omitido**
- La UI y el monitoreo permanecen **accesibles**
- La base de datos y todos los servicios permanecen **operativos**

Alertas:

```
- alert: SMS_C_License_Invalid
  expr: sms_c_license_status == 0
  for: 1m
  labels:
    severity: critical
  annotations:
    summary: "Licencia SMS-C inválida o expirada"
    description: "El estado de la licencia es inválido - los mensajes se están enrutando a NOLICENCE"
```

Ejemplos de Consultas Prometheus:

```
# Verificar si la licencia es válida
sms_c_license_status == 1

# Alerta sobre licencia inválida
sms_c_license_status == 0

# Contar mensajes enrutados a NOLICENCE (indica problema de licencia)
sms_c_routing_route_matched_count{dest_smsc="NOLICENCE"}
```

Métricas de Procesamiento de Mensajes

sms_c_message_received_count

Tipo: Counter

Descripción: Número total de mensajes recibidos por el SMS-C de todas las fuentes.

Etiquetas:

- `source_smsc`: Nombre del SMSC de origen que envió el mensaje
- `source_type`: Tipo de conexión de origen (ims, circuit_switched, smpp)
- `message_type`: Tipo de mensaje (sms, mms)

Caso de Uso: Monitorear el volumen de mensajes entrantes por fuente y tipo. Usar para detectar patrones de tráfico, identificar períodos de alta actividad y detectar anomalías en el flujo de mensajes.

Alertas: Establecer alertas para caídas repentinas (posibles problemas de conectividad de la fuente) o picos (posible ataque/spam).

sms_c_message_validated_count

Tipo: Counter

Descripción: Número total de validaciones de mensajes realizadas.

Etiquetas:

- `valid`: Si la validación pasó (true o false)

Caso de Uso: Rastrear tasas de éxito/fallo de validación. Altas tasas de fallo pueden indicar mensajes mal formados o problemas de integración.

Alertas: Alerta cuando la tasa de fallos de validación excede el umbral (por ejemplo, > 5% de fallos).

sms_c_message_processing_stop_duration

Tipo: Histogram

Descripción: Tiempo tomado para procesar un mensaje desde la recepción hasta la finalización (incluye validación, enrutamiento y encolado).

Unidad: Milisegundos

Buckets: 10, 50, 100, 250, 500, 1000, 2500, 5000 ms

Etiquetas:

- `success`: Si el procesamiento tuvo éxito (true o false)

Caso de Uso: Monitorear el rendimiento del procesamiento de mensajes de extremo a extremo. Identificar desaceleraciones en la tubería de procesamiento.

Alertas: Alerta cuando la latencia p95 o p99 excede los umbrales SLA.

Métricas de Enrutamiento

sms_c_routing_route_matched_count

Tipo: Counter

Descripción: Número total de veces que se coincidió y seleccionó una ruta específica para el enrutamiento de mensajes.

Etiquetas:

- `route_id`: Identificador único de la ruta coincidente

- `dest_smsc`: SMSC de destino seleccionado por la ruta
- `priority`: Valor de prioridad de la ruta coincidente

Caso de Uso: Entender qué rutas se utilizan con más frecuencia. Identificar rutas infrautilizadas o sobrecargadas. Útil para la planificación de capacidad y optimización de rutas.

Alertas: Alerta si las rutas de alta prioridad rara vez se coinciden (puede indicar mala configuración de enrutamiento).

`sms_c_routing_failed_count`

Tipo: Counter

Descripción: Número total de fallos de enrutamiento donde no se pudo encontrar una ruta adecuada.

Etiquetas:

- `reason`: Razón del fallo (no_route_found, validation_failed, etc.)

Caso de Uso: Rastrear fallos de enrutamiento para identificar brechas de configuración o patrones de tráfico inesperados.

Alertas: Alerta sobre cualquier fallo de enrutamiento ya que indican que los mensajes no pueden ser entregados.

`sms_c_routing_action_count`

Tipo: Counter

Descripción: Número total de acciones de enrutamiento especiales tomadas.

Etiquetas:

- `action`: Tipo de acción (drop, auto_reply, forward)
- `route_id`: Ruta que activó la acción

Caso de Uso: Monitorear reglas de eliminación (anti-spam), uso de respuestas automáticas y patrones de reenvío.

Alertas: Alerta sobre picos inesperados en acciones de eliminación (puede indicar ataque de spam).

sms_c_routing_stop_duration

Tipo: Histogram

Descripción: Tiempo tomado para evaluar todas las rutas y seleccionar la mejor coincidencia.

Unidad: Milisegundos

Buckets: 1, 5, 10, 25, 50, 100, 250, 500 ms

Etiquetas:

- `dest_smsc`: SMSC de destino seleccionado

Caso de Uso: Monitorear el rendimiento del motor de enrutamiento. Un enrutamiento lento indica demasiadas rutas o lógica de coincidencia compleja.

Alertas: Alerta cuando el enrutamiento toma consistentemente más tiempo del esperado (por ejemplo, p95 > 50ms).

Métricas de Búsqueda ENUM/NAPTR

sms_c_enum_cache_hit_count

Tipo: Counter

Descripción: Número total de búsquedas ENUM servidas desde la caché (no requirió consulta DNS).

Etiquetas:

- `domain`: Dominio ENUM consultado

Caso de Uso: Monitorear la efectividad de la caché. Altas tasas de aciertos en la caché reducen la carga de DNS y mejoran el rendimiento.

Alertas: Alerta si la tasa de aciertos en la caché cae por debajo del umbral (puede indicar problemas de caché o tráfico inusual).

sms_c_enum_cache_miss_count

Tipo: Counter

Descripción: Número total de búsquedas ENUM que requirieron una consulta DNS (no en caché).

Etiquetas:

- `domain`: Dominio ENUM consultado

Caso de Uso: Rastrear fallos de caché para entender la efectividad de la caché. Usar con el conteo de aciertos para calcular la tasa de aciertos.

Cálculo: `cache_hit_rate = hits / (hits + misses)`

sms_c_enum_cache_size_size

Tipo: Gauge

Descripción: Número actual de entradas en la caché ENUM.

Caso de Uso: Monitorear el tamaño de la caché para asegurar que no esté creciendo sin límites. Ayuda a ajustar la configuración de TTL de la caché.

Alertas: Alerta si el tamaño de la caché excede los límites esperados (puede indicar fuga de memoria).

sms_c_enum_lookup_stop_duration

Tipo: Histogram

Descripción: Tiempo tomado para completar una búsqueda ENUM (incluyendo consulta DNS si no está en caché).

Unidad: Milisegundos

Buckets: 10, 50, 100, 250, 500, 1000, 2500, 5000 ms

Etiquetas:

- `domain`: Dominio ENUM consultado
- `success`: Si la búsqueda tuvo éxito (true o false)
- `cache_hit`: Si el resultado fue servido desde la caché (true o false)

Caso de Uso: Monitorear el rendimiento de las búsquedas ENUM. Identificar servidores DNS lentos o problemas de red.

Alertas: Alerta cuando el tiempo de búsqueda p95 excede el umbral de tiempo de espera.

sms_c_enum_naptr_records_record_count

Tipo: Histogram

Descripción: Número de registros NAPTR devueltos por una búsqueda ENUM exitosa.

Buckets: 0, 1, 2, 3, 5, 10

Etiquetas:

- `domain`: Dominio ENUM consultado

Caso de Uso: Entender la distribución de registros ENUM. La mayoría de las búsquedas deberían devolver de 1 a 3 registros.

Alertas: Alerta si frecuentemente se devuelven 0 registros (problema de configuración DNS).

Métricas de Entrega

sms_c_delivery_queued_count

Tipo: Counter

Descripción: Número total de mensajes en cola para entrega a un SMSC de destino.

Etiquetas:

- `dest_smsc`: Nombre del SMSC de destino

Caso de Uso: Monitorear el flujo de mensajes a cada destino. Útil para la planificación de capacidad.

Alertas: Comparar con los conteos de éxito/fallo de entrega para detectar acumulación.

sms_c_delivery_attempted_count

Tipo: Counter

Descripción: Número total de intentos de entrega realizados (incluye reintentos).

Etiquetas:

- `dest_smsc`: Nombre del SMSC de destino

Caso de Uso: Rastrear el volumen de intentos de entrega. Un alto conteo de intentos en relación con el conteo en cola indica comportamiento de reintento.

sms_c_delivery_succeeded_count

Tipo: Counter

Descripción: Número total de mensajes entregados con éxito al SMSC de destino.

Etiquetas:

- `dest_smsc`: Nombre del SMSC de destino

Caso de Uso: Rastrear entregas exitosas por destino. Métrica principal de éxito.

Alertas: Alerta si la tasa de éxito cae por debajo del umbral SLA.

Cálculo: `success_rate = succeeded / queued`

sms_c_delivery_failed_count

Tipo: Counter

Descripción: Número total de mensajes que fallaron en la entrega después de todos los intentos de reintento.

Etiquetas:

- `dest_smsc`: Nombre del SMSC de destino
- `reason`: Razón del fallo

Caso de Uso: Rastrear fallos de entrega para identificar destinos problemáticos o patrones de fallo.

Alertas: Alerta sobre tasas de fallo elevadas o razones de fallo específicas.

sms_c_delivery_dead_letter_count

Tipo: Counter

Descripción: Número total de mensajes movidos a la cola de cartas muertas (indeseables).

Etiquetas:

- `reason`: Razón para la carta muerta (por ejemplo, `max_retries_exceeded`, `expired`)

Caso de Uso: Monitorear mensajes indeseables que requieren intervención manual.

Alertas: Alerta sobre cualquier evento de carta muerta ya que representan un fallo completo de entrega.

sms_c_delivery_succeeded_attempt_count

Tipo: Histogram

Descripción: Número de intentos de entrega requeridos antes de una entrega exitosa.

Buckets: 1, 2, 3, 5, 10

Etiquetas:

- `dest_smsc`: Nombre del SMSC de destino

Caso de Uso: Entender el comportamiento de reintentos. La mayoría de las entregas deberían tener éxito en el primer intento.

Alertas: Alerta si el conteo promedio de intentos excede 2 (indica problemas de fiabilidad del destino).

sms_c_delivery_failed_attempt_count

Tipo: Histogram

Descripción: Número de intentos de entrega realizados antes del fallo final.

Buckets: 1, 2, 3, 5, 10

Etiquetas:

- `dest_smsc`: Nombre del SMSC de destino

Caso de Uso: Entender cuántos reintentos ocurren antes de rendirse.

sms_c_delivery_time_delta_delta_ms

Tipo: Histogram

Descripción: Tiempo desde la presentación del mensaje hasta la confirmación de entrega. Esta métrica captura la latencia completa de extremo a extremo desde que un mensaje entra en el SMS-C hasta que se confirma la entrega, con etiquetas detalladas para análisis por fuente, destino y comportamiento de reintento.

Unidad: Milisegundos

Buckets: 50, 100, 250, 500, 1000, 2000, 5000, 10000, 20000, 60000, 600000 ms (50ms a 10 minutos)

Etiquetas:

- `source_smsc`: SMSC de origen que presentó el mensaje
- `dest_smsc`: SMSC de destino que entregó el mensaje
- `delivery_attempts`: Número de intentos de entrega requeridos (0 = éxito en el primer intento)

Caso de Uso: Analizar el rendimiento de entrega de extremo a extremo por par fuente-destino. Identificar rutas lentas, combinaciones problemáticas de SMSC y correlacionar la latencia con el comportamiento de reintento. Esencial para el monitoreo de SLA y la planificación de capacidad.

Alertas:

- Alerta cuando p95 excede el umbral SLA para combinaciones específicas de fuente/destino

- Alerta cuando los mensajes requieren consistentemente múltiples intentos

Ejemplos de Consultas Prometheus:

```

# Tiempo de entrega p95 por SMSC de origen y destino
histogram_quantile(0.95,
  sum by (source_smsc, dest_smsc, le) (
    rate(sms_c_delivery_time_delta_delta_ms_bucket[5m])
  )
)

# Tiempo de entrega p99 en general
histogram_quantile(0.99,
  sum by (le) (
    rate(sms_c_delivery_time_delta_delta_ms_bucket[5m])
  )
)

# Tiempo promedio de entrega por SMSC de origen
sum by (source_smsc) (rate(sms_c_delivery_time_delta_delta_ms_sum[5m]
/
sum by (source_smsc)
(rate(sms_c_delivery_time_delta_delta_ms_count[5m])))

# Tiempo de entrega para mensajes que requieren reintentos vs éxito e
el primer intento
histogram_quantile(0.95,
  sum by (le)
(rate(sms_c_delivery_time_delta_delta_ms_bucket{delivery_attempts="0"
[5m]))
)
histogram_quantile(0.95,
  sum by (le)
(rate(sms_c_delivery_time_delta_delta_ms_bucket{delivery_attempts!="0"
[5m]))
)

# Pares de origen-destino más lentos (por p95)
topk(10,
  histogram_quantile(0.95,
    sum by (source_smsc, dest_smsc, le) (
      rate(sms_c_delivery_time_delta_delta_ms_bucket[1h])
    )
  )
)

# Porcentaje de mensajes entregados dentro de 2 segundos

```

```
sum(rate(sms_c_delivery_time_delta_delta_ms_bucket{le="2000"}[5m])) /
sum(rate(sms_c_delivery_time_delta_delta_ms_count[5m])) * 100

# Mensajes entregados dentro de 2 segundos por destino
sum by (dest_smsc)
(rate(sms_c_delivery_time_delta_delta_ms_bucket{le="2000"}[5m])) /
sum by (dest_smsc) (rate(sms_c_delivery_time_delta_delta_ms_count[5m]
* 100
```

Métricas de Cola

sms_c_queue_size_size

Tipo: Gauge

Descripción: Número total actual de mensajes en cola (todos los estados combinados).

Etiquetas:

- `queue_type`: Tipo de cola (message_queue, dead_letter)

Caso de Uso: Monitorear la profundidad de la cola para detectar acumulaciones o problemas de procesamiento.

Alertas: Alerta cuando el tamaño de la cola excede los umbrales de capacidad.

sms_c_queue_size_pending

Tipo: Gauge

Descripción: Número actual de mensajes pendientes de entrega (aún no intentados).

Etiquetas:

- `queue_type`: Tipo de cola

Caso de Uso: Monitorear el conteo de mensajes pendientes. Altos conteos pendientes indican retrasos en el procesamiento.

Alertas: Alerta cuando el conteo pendiente excede el umbral durante un período prolongado.

sms_c_queue_size_failed

Tipo: Gauge

Descripción: Número actual de mensajes en estado de fallo (esperando reintento).

Etiquetas:

- `queue_type`: Tipo de cola

Caso de Uso: Monitorear la acumulación de mensajes fallidos. Indica problemas de entrega.

Alertas: Alerta sobre conteos fallidos elevados ya que impactan las tasas de entrega.

sms_c_queue_size_delivered

Tipo: Gauge

Descripción: Número actual de mensajes entregados que esperan limpieza/eliminación de la cola.

Etiquetas:

- `queue_type`: Tipo de cola

Caso de Uso: Monitorear el retraso en la limpieza. Altos conteos indican que el proceso de limpieza se está quedando atrás.

Alertas: Alerta si los mensajes entregados se acumulan significativamente.

sms_c_queue_oldest_message_age_seconds

Tipo: Gauge

Descripción: Edad (en segundos) del mensaje más antiguo actualmente en estado pendiente.

Etiquetas:

- `queue_type`: Tipo de cola

Caso de Uso: Detectar el envejecimiento de mensajes y paradas en el procesamiento. Crítico para el monitoreo de SLA.

Alertas: Alerta cuando la edad del mensaje más antiguo excede el umbral SLA (por ejemplo, > 300 segundos).

Métricas de Cobro

sms_c_charging_requested_count

Tipo: Counter

Descripción: Número total de solicitudes de cobro/facturación realizadas al OCS o sistema de facturación.

Etiquetas:

- `account`: Identificador de cuenta que se está cobrando

Caso de Uso: Rastrear el volumen de cobro por cuenta. Útil para la conciliación de facturas.

sms_c_charging_succeeded_count

Tipo: Counter

Descripción: Número total de operaciones de cobro exitosas.

Etiquetas:

- `account`: Identificador de cuenta cobrada

Caso de Uso: Monitorear la tasa de éxito de cobro por cuenta.

Cálculo: $\text{success_rate} = \text{succeeded} / \text{requested}$

sms_c_charging_failed_count

Tipo: Counter

Descripción: Número total de operaciones de cobro fallidas.

Etiquetas:

- `account`: Identificador de cuenta
- `reason`: Razón del fallo

Caso de Uso: Identificar fallos de cobro que pueden impactar los ingresos o requerir intervención de la cuenta.

Alertas: Alerta sobre tasas de fallo de cobro elevadas.

sms_c_charging_succeeded_duration

Tipo: Histogram

Descripción: Tiempo tomado para completar una solicitud de cobro exitosa.

Unidad: Milisegundos

Buckets: 10, 50, 100, 250, 500, 1000, 2500, 5000 ms

Etiquetas:

- `account`: Identificador de cuenta

Caso de Uso: Monitorear el rendimiento del sistema de facturación. Un cobro lento puede retrasar la entrega de mensajes.

Alertas: Alerta cuando el tiempo de cobro p95 excede el umbral.

Métricas de Salud del Sistema

`sms_c_mnesia_table_size_record_count`

Tipo: Gauge

Descripción: Número actual de registros en cada tabla de base de datos Mnesia. Se consulta cada 10 segundos.

Etiquetas:

- `table`: Nombre de la tabla

Tablas Rastreadas:

- `sms_route` - Reglas de enrutamiento SMS
- `message_store` - Cola de mensajes (mensajes pendientes, entregados, fallidos)
- `location_store` - Datos de ubicación/registro de suscriptores
- `frontend_store` - Registros de frontend/SMSC
- `translation_rule` - Número de reglas de traducción
- `cell_tower_store` - Datos de ubicación de torres celulares
- `message_events` - Registros de eventos de mensajes

Caso de Uso: Monitorear el crecimiento de la base de datos. Detectar acumulación de datos inesperada. Planificación de capacidad.

Alertas: Alerta sobre tasas de crecimiento inesperadas de tablas o cuando se acercan a los límites de capacidad.

Ejemplos de Consultas Prometheus:

```
# Todos los tamaños de tabla
sms_c_mnesia_table_size_record_count

# Tamaño de la cola de mensajes
sms_c_mnesia_table_size_record_count{table="message_store"}

# Registros de suscriptores activos
sms_c_mnesia_table_size_record_count{table="location_store"}

# Conteo de reglas de enrutamiento
sms_c_mnesia_table_size_record_count{table="sms_route"}
```

sms_c_frontend_status_count

Tipo: Gauge

Descripción: Número de frontends en cada estado de conexión.

Etiquetas:

- `frontend_name`: Identificador del frontend
- `status`: Estado de conexión (connected, disconnected)

Caso de Uso: Monitorear la conectividad del frontend. Detectar fallos de conexión.

Alertas: Alerta cuando los frontends esperados se desconectan.

sms_c_location_registered_count

Tipo: Counter

Descripción: Número total de registros de ubicación/suscriptor recibidos por el sistema.

Etiquetas:

- `location`: Nombre del frontend/SMSC donde está registrado el suscriptor
- `ims_capable`: Si el suscriptor soporta IMS (true/false)

Caso de Uso: Monitorear la actividad de registro de suscriptores. Rastrear suscriptores IMS vs no IMS. Detectar tormentas de registro o fallos.

Alertas: Establecer alertas para:

- Caídas en la tasa de registro (puede indicar problemas de red)
- Picos inusuales en registros
- Alto ratio de registros no IMS (afluencia de dispositivos antiguos)

Ejemplo de Consulta:

```
# Tasa de registro por minuto
rate(sms_c_location_registered_count[1m])

# Ratio de registro IMS vs no IMS
sum(rate(sms_c_location_registered_count{ims_capable="true"}[5m]))
/
sum(rate(sms_c_location_registered_count[5m]))
```

sms_c_location_active_registrations_count

Tipo: Gauge

Descripción: Número actual de registros de suscriptores activos, agrupados por ubicación (frontend/SMSC) y capacidad IMS. Esta métrica se consulta cada 10 segundos y refleja el estado actual de la tienda de ubicaciones.

Etiquetas:

- `location`: Nombre del frontend/SMSC donde están registrados los suscriptores
- `ims_capable`: Si los registros son capaces de IMS (true/false)

Caso de Uso: Monitorear la distribución actual de suscriptores entre frontends. Rastrear la adopción de IMS. Identificar desequilibrios de carga entre frontends. Planificación de capacidad para cada instancia de SMSC.

Alertas: Establecer alertas para:

- Caídas repentinas en registros para una ubicación (puede indicar problemas en el frontend)
- Distribución desequilibrada entre frontends
- Registros totales acercándose a límites de capacidad

Ejemplos de Consultas Prometheus:

```
# Total de registros activos
sum(sms_c_location_active_registrations_count)

# Registros activos por ubicación
sum by (location) (sms_c_location_active_registrations_count)

# Registros activos capaces de IMS por ubicación
sum by (location)
(sms_c_location_active_registrations_count{ims_capable="true"})

# Tasa de adopción de IMS por ubicación
sum by (location)
(sms_c_location_active_registrations_count{ims_capable="true"}) /
sum by (location) (sms_c_location_active_registrations_count) *
100

# Total de registros IMS vs no IMS
sum by (ims_capable) (sms_c_location_active_registrations_count)

# Ubicaciones con más registros
topk(10, sum by (location)
(sms_c_location_active_registrations_count))

# Distribución de registros como porcentaje del total
sum by (location) (sms_c_location_active_registrations_count) /
sum(sms_c_location_active_registrations_count) * 100
```

Métricas de Solicitudes de API HTTP

phoenix_endpoint_stop_duration

Tipo: Distribución (Histogram)

Descripción: Duración del procesamiento de solicitudes HTTP en milisegundos, desde el inicio de la solicitud hasta la finalización de la respuesta.

Etiquetas:

- `route`: Ruta del endpoint de la API (por ejemplo, `/api/messages`, `/api/frontends`)

Buckets: 10ms, 50ms, 100ms, 250ms, 500ms, 1s, 2.5s, 5s

Caso de Uso: Monitorear el rendimiento de la API. Identificar endpoints lentos. Rastrear SLAs de tiempo de respuesta.

Alertas: Establecer alertas para:

- Latencia P95 > 500ms para endpoints críticos
- Latencia P99 > 1s para cualquier endpoint
- Tendencias de latencia en aumento

Ejemplo de Consulta:

```
# Tiempo de respuesta p95 por endpoint
histogram_quantile(0.95,
  rate(phoenix_endpoint_stop_duration_bucket[5m]))

# Solicitudes más lentas de 1 segundo
sum(rate(phoenix_endpoint_stop_duration_bucket{le="1000"}[5m]))
```

phoenix_endpoint_stop_count

Tipo: Counter

Descripción: Número total de solicitudes HTTP completadas, categorizadas por ruta y código de estado HTTP.

Etiquetas:

- `route`: Ruta del endpoint de la API
- `status`: Código de estado HTTP (200, 201, 400, 404, 500, etc.)

Caso de Uso: Monitorear el volumen de solicitudes de API y tasas de éxito. Rastrear tasas de error por endpoint.

Alertas: Establecer alertas para:

- Tasa de error > 5% para cualquier endpoint
- Errores 5xx en endpoints críticos
- Caídas repentinas en el volumen de solicitudes

Ejemplo de Consulta:

```
# Tasa de solicitudes por endpoint
sum by (route) (rate(phoenix_endpoint_stop_count[5m]))

# Tasa de error por endpoint
sum by (route) (rate(phoenix_endpoint_stop_count{status=~"5.."}
[5m])) /
sum by (route) (rate(phoenix_endpoint_stop_count[5m]))

# Tasa de éxito
sum(rate(phoenix_endpoint_stop_count{status=~"2.."}[5m])) /
sum(rate(phoenix_endpoint_stop_count[5m]))
```

phoenix_router_dispatch_exception_count

Tipo: Counter

Descripción: Número total de excepciones/errores generados durante el procesamiento de solicitudes HTTP.

Etiquetas:

- `route`: Ruta del endpoint de la API donde ocurrió la excepción
- `kind`: Tipo de excepción (error, exit, throw)

Caso de Uso: Rastrear errores de aplicación. Identificar endpoints problemáticos. Monitorear la estabilidad del sistema.

Alertas: Establecer alertas para cualquier valor distinto de cero en endpoints críticos.

Ejemplo de Consulta:

```
# Tasa de excepciones por endpoint
rate(phoenix_router_dispatch_exception_count[5m])

# Total de excepciones en la última hora
increase(phoenix_router_dispatch_exception_count[1h])
```

Métricas de Erlang VM

vm_memory_total

Tipo: Gauge

Descripción: Memoria total asignada por la Erlang VM en bytes.

Caso de Uso: Monitorear el uso general de memoria. Detectar fugas de memoria. Planificar capacidad.

Alertas: Alerta cuando el uso de memoria > 80% de la memoria del sistema disponible.

vm_memory_processes

Tipo: Gauge

Descripción: Memoria utilizada por procesos Erlang en bytes.

Caso de Uso: Rastrear el consumo de memoria de procesos. Fuente más común de crecimiento de memoria.

Alertas: Alerta sobre tasa de crecimiento sostenida alta.

vm_total_run_queue_lengths_total

Tipo: Gauge

Descripción: Número total de procesos esperando ser programados en todos los programadores de CPU.

Caso de Uso: Medir la carga del sistema. Valores altos indican saturación de CPU.

Alertas: Alerta cuando consistentemente $> 10 * \text{número de núcleos de CPU}$.

vm_system_counts_process_count

Tipo: Gauge

Descripción: Número actual de procesos en ejecución en la VM.

Caso de Uso: Monitorear patrones de creación de procesos. Detectar fugas de procesos.

Alertas: Alerta cuando se acerque al límite de procesos (predeterminado 262,144).

Recolección y Polling de Métricas

El sistema recopila automáticamente las siguientes métricas cada 10 segundos:

- Tamaños y edades de colas
- Tamaños de tablas de Mnesia
- Estadísticas de caché ENUM

Todas las demás métricas son impulsadas por eventos y se emiten cuando ocurre la acción correspondiente.

Patrones Comunes de Monitoreo

Tasa de Éxito de Entrega por Destino

Rastrear la tasa de éxito de la entrega de mensajes para cada SMSC de destino:

Fórmula: $(\text{sms_c_delivery_succeeded_count}) / (\text{sms_c_delivery_queued_count})$

Interpretación: Debería ser > 95% para destinos saludables. Tasas más bajas indican problemas de entrega.

Latencia de Mensaje de Extremo a Extremo

Monitorear el tiempo total desde la recepción del mensaje hasta la entrega:

Métricas:

- $\text{sms_c_message_processing_stop_duration}$ (procesamiento)
- $\text{sms_c_delivery_time_delta_delta_ms}$ (entrega)

Interpretación: La suma representa la latencia total desde la perspectiva del usuario.

Efectividad de la Caché ENUM

Medir qué tan bien está funcionando la caché ENUM:

Fórmula: $(\text{sms_c_enum_cache_hit_count}) / (\text{sms_c_enum_cache_hit_count} + \text{sms_c_enum_cache_miss_count})$

Interpretación: Debería ser > 80% después del calentamiento. Tasas más bajas pueden indicar TTL corto o alta variabilidad de tráfico.

Utilización de Rutas

Identificar qué rutas manejan más tráfico:

Métrica: `sms_c_routing_route_matched_count` agrupada por `route_id`

Interpretación: Usar para identificar rutas calientes para optimización y planificación de capacidad.

Tendencia de Acumulación de Cola

Monitorear si la cola de mensajes está creciendo (acumulación) o disminuyendo (poniéndose al día):

Métricas:

- `sms_c_queue_size_pending` (actual pendiente)
- `sms_c_queue_oldest_message_age_seconds` (tendencia de edad)

Interpretación: Conteo pendiente creciente + edad en aumento = acumulación formándose.

Tasa de Reintentos

Entender con qué frecuencia se requieren reintentos de entrega:

Métrica: percentiles del histograma

`sms_c_delivery_succeeded_attempt_count`

Interpretación: Si $p_{95} > 1$, la mayoría de los mensajes requieren reintentos. Indica problemas de fiabilidad del destino.

Alertas Recomendadas

Alerta	Condición	Severidad
Alta Tasa de Fallos de Enrutamiento	<code>routing_failed_count</code> aumento	Crítica
Acumulación de Cola	<code>queue_size_pending</code> > umbral	Advertencia
Mensajes Antiguos en Cola	<code>queue_oldest_message_age_seconds</code> > 300	Crítica
Pico de Fallos de Entrega	<code>delivery_failed_count</code> pico	Alta
Eventos de Carta Muerta	<code>delivery_dead_letter_count</code> > 0	Alta
Timeouts de Búsqueda ENUM	<code>enum_lookup_stop_duration</code> p95 > 5000ms	Advertencia
Baja Tasa de Aciertos en Caché	Tasa de aciertos de caché ENUM < 0.7	Advertencia
Frontend Desconectado	<code>frontend_status_count{status="disconnected"}</code> > 0	Alta
Fallos de Cobro	<code>charging_failed_count</code> > umbral	Alta

Alerta	Condición	Severidad
Procesamiento de Mensajes Lento	<code>message_processing_stop_duration</code> p95 > 1000ms	Advertencia

Recomendaciones de Dashboard

Dashboard de Operaciones

Propósito: Monitoreo en tiempo real de la salud del sistema

Paneles:

1. Rendimiento de mensajes (recibidos/procesados/entregados por minuto)
 2. Tamaños de cola (pendientes, fallidos, entregados)
 3. Tasa de éxito de entrega por destino
 4. Latencia de procesamiento y entrega p95
 5. Estado de frontends activos
 6. Alertas actuales
-

Dashboard de Rendimiento

Propósito: Análisis del rendimiento del sistema

Paneles:

1. Histograma de duración del procesamiento de mensajes
2. Histograma de duración de enrutamiento
3. Histograma de duración de búsqueda ENUM
4. Histograma de duración de cobro
5. Distribución de intentos de entrega
6. Tasas de aciertos en caché

Dashboard de Negocios

Propósito: Análisis de tráfico y uso

Paneles:

1. Mensajes por SMSC de origen
 2. Mensajes por SMSC de destino
 3. Mapa de calor de utilización de rutas
 4. Conteos de acciones de respuesta automática y eliminación
 5. Estadísticas de uso de ENUM
 6. Volumen de cobro por cuenta
-

Retención de Métricas

Configuraciones de retención recomendadas para Prometheus:

- **Métricas en crudo:** 15 días
- **Agregados de 5 minutos:** 90 días
- **Agregados de 1 hora:** 2 años

Esto proporciona un historial reciente detallado mientras mantiene tendencias a largo plazo para la planificación de capacidad.

Solución de Problemas con Métricas

Escenario: Mensajes No Entregados

Pasos de Investigación:

1. Verificar `sms_c_message_received_count` - ¿Se están recibiendo mensajes?

2. Verificar `sms_c_routing_failed_count` - ¿Se están enrutando?
 3. Verificar `sms_c_delivery_queued_count` - ¿Se están encolando?
 4. Verificar `sms_c_delivery_failed_count` - ¿Están fallando los intentos de entrega?
 5. Verificar etiquetas `dest_smsc` para identificar destinos problemáticos
-

Escenario: Procesamiento de Mensajes Lento

Pasos de Investigación:

1. Verificar histograma `sms_c_message_processing_stop_duration` - Tiempo de procesamiento general
 2. Verificar `sms_c_routing_stop_duration` - ¿Es lento el enrutamiento?
 3. Verificar `sms_c_enum_lookup_stop_duration` - ¿Son lentas las búsquedas ENUM?
 4. Verificar `sms_c_charging_succeeded_duration` - ¿Es lento el cobro?
 5. Identificar cuellos de botella e investigar el componente específico
-

Escenario: Crecimiento de la Cola de Mensajes

Pasos de Investigación:

1. Verificar tendencia de `sms_c_queue_size_pending` - ¿Está creciendo?
 2. Verificar `sms_c_delivery_attempted_count` - ¿Se están realizando intentos de entrega?
 3. Verificar `sms_c_delivery_failed_count` - ¿Están fallando?
 4. Verificar `sms_c_delivery_time_delta_delta_ms` - ¿Está tardando demasiado la entrega?
 5. Verificar etiquetas `dest_smsc` para identificar destinos lentos
-

Ejemplos de Consultas Prometheus

Rendimiento de Mensajes

Mensajes Recibidos Por Segundo (promedio de 5 minutos):

```
rate(sms_c_message_received_count[5m])
```

Mensajes Recibidos Por Minuto (promedio de 1 hora):

```
rate(sms_c_message_received_count[1h]) * 60
```

Total de Mensajes Hoy:

```
increase(sms_c_message_received_count[24h])
```

Mensajes por Tipo de Fuente:

```
sum by (source_type) (rate(sms_c_message_received_count[5m]))
```

Mensajes por SMSC de Origen:

```
sum by (source_smsc) (rate(sms_c_message_received_count[5m]))
```

Rendimiento de Entrega

Tasa de Éxito de Entrega (Porcentaje):

```
(rate(sms_c_delivery_succeeded_count[5m]) /  
rate(sms_c_delivery_queued_count[5m])) * 100
```

Tasa de Fallo de Entrega (Porcentaje):

```
(rate(sms_c_delivery_failed_count[5m]) /  
rate(sms_c_delivery_queued_count[5m])) * 100
```

Intentos Promedio de Entrega (p95):

```
histogram_quantile(0.95,  
sms_c_delivery_succeeded_attempt_count_bucket)
```

Éxito de Entrega por Destino:

```
sum by (dest_smsc) (rate(sms_c_delivery_succeeded_count[5m]))
```

Razones de Fallo de Entrega:

```
sum by (reason) (rate(sms_c_delivery_failed_count[5m]))
```

Tiempo de Entrega (p95):

```
histogram_quantile(0.95,  
sms_c_delivery_time_delta_delta_ms_bucket)
```

Tiempo de Entrega (p99):

```
histogram_quantile(0.99,  
sms_c_delivery_time_delta_delta_ms_bucket)
```

Métricas de Cola

Mensajes Pendientes Actuales:

```
sms_c_queue_size_pending
```

Mensajes Fallidos Esperando Reintento:

```
sms_c_queue_size_failed
```

Edad del Mensaje Más Antiguo (Minutos):

```
sms_c_queue_oldest_message_age_seconds / 60
```

Tasa de Crecimiento de Cola (Mensajes/Hora):

```
rate(sms_c_queue_size_size[1h]) * 3600
```

Mensajes Entrando en la Cola:

```
rate(sms_c_delivery_queued_count[5m])
```

Mensajes Saliendo de la Cola:

```
rate(sms_c_delivery_succeeded_count[5m]) +  
rate(sms_c_delivery_failed_count[5m])
```

Acumulación de Cola (Entrando - Saliendo):

```
rate(sms_c_delivery_queued_count[5m]) -  
(rate(sms_c_delivery_succeeded_count[5m]) +  
rate(sms_c_delivery_failed_count[5m]))
```

Rendimiento de Enrutamiento

Tasa de Éxito de Enrutamiento:

```
(1 - (rate(sms_c_routing_failed_count[5m]) /  
(rate(sms_c_routing_route_matched_count[5m]) +  
rate(sms_c_routing_failed_count[5m])))) * 100
```

Rutas Más Utilizadas:

```
topk(10, sum by (route_id, dest_smsc)
(rate(sms_c_routing_route_matched_count[1h])))
```

Fallos de Enrutamiento por Minuto:

```
rate(sms_c_routing_failed_count[5m]) * 60
```

Acciones de Eliminación por Hora:

```
increase(sms_c_routing_action_count{action="drop"}[1h])
```

Acciones de Respuesta Automática por Hora:

```
increase(sms_c_routing_action_count{action="auto_reply"}[1h])
```

Rendimiento ENUM

Tasa de Aciertos en Caché ENUM:

```
rate(sms_c_enum_cache_hit_count[5m]) /
(rate(sms_c_enum_cache_hit_count[5m]) +
rate(sms_c_enum_cache_miss_count[5m]))
```

Porcentaje de Aciertos en Caché ENUM:

```
(rate(sms_c_enum_cache_hit_count[5m]) /
(rate(sms_c_enum_cache_hit_count[5m]) +
rate(sms_c_enum_cache_miss_count[5m]))) * 100
```

Latencia de Búsqueda ENUM (p95):

```
histogram_quantile(0.95, sms_c_enum_lookup_stop_duration_bucket)
```

Búsquedas ENUM por Segundo (Cacheadas vs No Cacheadas):

```
# Cacheadas (rápido)
rate(sms_c_enum_cache_hit_count[5m])

# No Cacheadas (requiere consulta DNS)
rate(sms_c_enum_cache_miss_count[5m])
```

Promedio de Registros NAPTR Devueltos:

```
rate(sms_c_enum_naptr_records_record_count_sum[5m]) /
rate(sms_c_enum_naptr_records_record_count_count[5m])
```

Tamaño de la Caché ENUM:

```
sms_c_enum_cache_size_size
```

Rendimiento de Procesamiento

Latencia de Procesamiento de Mensajes (p95):

```
histogram_quantile(0.95,
sms_c_message_processing_stop_duration_bucket)
```

Latencia de Procesamiento de Mensajes (p99):

```
histogram_quantile(0.99,
sms_c_message_processing_stop_duration_bucket)
```

Fallos de Procesamiento:

```
rate(sms_c_message_processing_stop_duration_count{success="false"}[5m])
```

Tasa de Fallos de Validación:

```
rate(sms_c_message_validated_count{valid="false"}[5m]) /  
rate(sms_c_message_validated_count[5m])
```

Métricas de Cobro

Tasa de Éxito de Cobro:

```
rate(sms_c_charging_succeeded_count[5m]) /  
rate(sms_c_charging_requested_count[5m])
```

Fallos de Cobro por Minuto:

```
rate(sms_c_charging_failed_count[5m]) * 60
```

Latencia de Cobro (p95):

```
histogram_quantile(0.95, sms_c_charging_succeeded_duration_bucket)
```

Volumen de Cobro por Cuenta:

```
sum by (account) (rate(sms_c_charging_requested_count[1h]))
```

Salud del Frontend

Frontends Activos:

```
sum(sms_c_frontend_status_count{status="connected"})
```

Frontends Desconectados:

```
sum(sms_c_frontend_status_count{status="disconnected"})
```

Frontends por Nombre:

```
sum by (frontend_name)  
(sms_c_frontend_status_count{status="connected"})
```

Salud del Sistema

Tamaños de Tablas de Mnesia:

```
sms_c_mnesia_table_size_record_count
```

Conteo de Rutas:

```
sms_c_mnesia_table_size_record_count{table="sms_route"}
```

Conteo de Reglas de Traducción:

```
sms_c_mnesia_table_size_record_count{table="translation_rule"}
```

Ejemplos de Dashboard de Grafana

Dashboard 1: Operaciones en Tiempo Real

Propósito: Monitorear la actividad y salud del sistema actual.

Paneles:

1. Rendimiento de Mensajes (Gráfico)

- Consulta: `rate(sms_c_message_received_count[5m])`
- Consulta: `rate(sms_c_delivery_succeeded_count[5m])`
- Unidad: mensajes/segundo
- Leyenda: `{{source_type}}`

2. Tasa de Éxito de Entrega (Gauge)

- Consulta: `(rate(sms_c_delivery_succeeded_count[5m]) / rate(sms_c_delivery_queued_count[5m])) * 100`
- Unidad: porcentaje (0-100)
- Umbrales:
 - Rojo: < 90
 - Amarillo: 90-95
 - Verde: > 95

3. Profundidad de Cola (Gráfico)

- Consulta: `sms_c_queue_size_pending`
- Consulta: `sms_c_queue_size_failed`
- Unidad: mensajes
- Leyenda: `{{queue_type}}`

4. Edad del Mensaje Más Antiguo (Stat)

- Consulta: `sms_c_queue_oldest_message_age_seconds / 60`
- Unidad: minutos
- Umbrales:
 - Verde: < 5
 - Amarillo: 5-10
 - Rojo: > 10

5. Frontends Activos (Stat)

- Consulta: `sum(sms_c_frontend_status_count{status="connected"})`
- Unidad: conteo
- Color: Azul

6. Fallos de Enrutamiento (Gráfico)

- Consulta: `rate(sms_c_routing_failed_count[5m]) * 60`
- Unidad: fallos/minuto
- Umbral de alerta: > 0

Dashboard 2: Análisis de Rendimiento

Propósito: Analizar el rendimiento del sistema e identificar cuellos de botella.

Paneles:

1. Latencia de Extremo a Extremo (Gráfico)

- Consulta: `histogram_quantile(0.50, sms_c_message_processing_stop_duration_bucket)` (p50)
- Consulta: `histogram_quantile(0.95, sms_c_message_processing_stop_duration_bucket)` (p95)
- Consulta: `histogram_quantile(0.99, sms_c_message_processing_stop_duration_bucket)` (p99)
- Unidad: milisegundos
- Leyenda: Percentil

2. Latencias de Componentes (Bar Gauge)

- Enrutamiento: `histogram_quantile(0.95, sms_c_routing_stop_duration_bucket)`
- ENUM: `histogram_quantile(0.95, sms_c_enum_lookup_stop_duration_bucket)`
- Cobro: `histogram_quantile(0.95, sms_c_charging_succeeded_duration_bucket)`
- Entrega: `histogram_quantile(0.95, sms_c_delivery_time_delta_delta_ms_bucket)`
- Unidad: milisegundos
- Barras horizontales

3. Distribución de Intentos de Entrega (Heatmap)

- Consulta: `sms_c_delivery_succeeded_attempt_count_bucket`

- Muestra cuántos intentos se requieren típicamente
- Escala de color: Azul (1 intento) a Rojo (muchos intentos)

4. Rendimiento de Caché ENUM (Gráfico)

- Tasa de Aciertos: $\frac{\text{rate}(\text{sms_c_enum_cache_hit_count}[5\text{m}])}{(\text{rate}(\text{sms_c_enum_cache_hit_count}[5\text{m}]) + \text{rate}(\text{sms_c_enum_cache_miss_count}[5\text{m}]))}$
- Tamaño de Caché: `sms_c_enum_cache_size_size`
- Eje Y dual (tasa vs tamaño)

5. Tasa de Éxito de Procesamiento (Gauge)

- Consulta: $\frac{(\text{rate}(\text{sms_c_message_processing_stop_duration_count}\{\text{success}=\text{"true"}\}[5\text{m}]))}{\text{rate}(\text{sms_c_message_processing_stop_duration_count}[5\text{m}]))} * 100$
- Unidad: porcentaje
- Umbrales:
 - Rojo: < 95
 - Amarillo: 95-99
 - Verde: > 99

Dashboard 3: Análisis de Tráfico

Propósito: Analizar patrones de tráfico de mensajes y distribución de enrutamiento.

Paneles:

1. Mensajes por Tipo de Fuente (Gráfico de Pastel)

- Consulta: $\text{sum by (source_type)} (\text{increase}(\text{sms_c_message_received_count}[1\text{h}]))$
- Muestra distribución: IMS vs CS vs SMPP

2. Mensajes por SMSC de Origen (Gráfico de Barras)

- Consulta: `sum by (source_smsc) (rate(sms_c_message_received_count[1h]))`
- Top 10 fuentes
- Barras horizontales

3. Utilización de Rutas (Tabla)

- Columnas:
 - Route ID
 - SMSC de Destino
 - Mensajes (1h): `sum by (route_id, dest_smsc) (increase(sms_c_routing_route_matched_count[1h]))`
 - Prioridad
 - Tasa de Éxito
- Ordenado por conteo de mensajes

4. Entrega por Destino (Gráfico)

- Consulta: `sum by (dest_smsc) (rate(sms_c_delivery_succeeded_count[5m]))`
- Unidad: mensajes/segundo
- Gráfico de área apilada
- Leyenda: `{{dest_smsc}}`

5. Acciones de Eliminación/Respuesta Automática (Stat)

- Eliminados: `increase(sms_c_routing_action_count{action="drop"}[1h])`
- Respuesta Automática: `increase(sms_c_routing_action_count{action="auto_reply"}[1h])`
- Estadísticas lado a lado

6. Patrón de Tráfico Horario (Gráfico)

- Consulta: `rate(sms_c_message_received_count[1h]) * 3600`
- Rango de tiempo: Últimos 7 días
- Muestra patrones diarios

Dashboard 4: Capacidad y Recursos

Propósito: Monitorear el uso de recursos y límites de capacidad.

Paneles:

1. Capacidad de Cola (Gráfico)

- Actual: `sms_c_queue_size_size`
- Línea de capacidad: Valor fijo basado en límites del sistema
- Muestra tendencia de utilización

2. Crecimiento de Tablas de Base de Datos (Gráfico)

- Mensajes:
`sms_c_mnesia_table_size_record_count{table="sms_route"}`
- Traducciones:
`sms_c_mnesia_table_size_record_count{table="translation_rule"}`
- Tendencia durante los últimos 30 días

3. Tendencia de Acumulación de Mensajes (Gráfico)

- Consulta: `rate(sms_c_delivery_queued_count[5m]) - (rate(sms_c_delivery_succeeded_count[5m]) + rate(sms_c_delivery_failed_count[5m]))`
- Positivo = acumulación creciente
- Negativo = poniéndose al día

4. Tráfico Máximo (Stat)

- Consulta: `max_over_time(rate(sms_c_message_received_count[5m])[24h:])`
- Muestra la tasa más alta de 5m en las últimas 24h
- Unidad: mensajes/segundo

5. Utilización de Capacidad (Gauge)

- Consulta: `(rate(sms_c_message_received_count[5m]) / MAX_CAPACITY) * 100`

- Reemplazar MAX_CAPACITY con el límite de su sistema
- Unidad: porcentaje
- Umbrales:
 - Verde: < 70
 - Amarillo: 70-85
 - Rojo: > 85

Dashboard 5: Cumplimiento de SLA

Propósito: Rastrear métricas de SLA y cumplimiento.

Paneles:

1. Cumplimiento de SLA (Gauge)

- Éxito de Entrega: $\frac{\text{rate}(\text{sms_c_delivery_succeeded_count}[1h])}{\text{rate}(\text{sms_c_delivery_queued_count}[1h])} * 100$
- Línea objetivo en 99%
- Umbrales:
 - Rojo: < 95
 - Amarillo: 95-99
 - Verde: >= 99

2. Mensajes Entregados Dentro de SLA (Stat)

- Consulta: $\frac{\text{count}(\text{sms_c_delivery_time_delta_delta_ms_bucket}\{\text{le}="5000"\})}{\text{count}(\text{sms_c_delivery_time_delta_delta_ms_bucket})}$
- Muestra porcentaje entregado dentro de 5 segundos
- Unidad: porcentaje

3. Violaciones de SLA (Counter)

- Mensajes que exceden 5 minutos: $\text{increase}(\text{sms_c_queue_oldest_message_age_seconds}\{\} > 300)[24h:]$
- Debería ser 0

4. Tiempo de Actividad (Stat)

- Consulta: `up{job="sms-c"}`
- Binario: 1 = activo, 0 = inactivo
- Muestra estado actual

5. Tendencia de Tasa de Éxito Diaria (Gráfico)

- Consulta:
`avg_over_time((rate(sms_c_delivery_succeeded_count[1h]) / rate(sms_c_delivery_queued_count[1h]))[24h:1h])`
- Rango de tiempo: Últimos 30 días
- Línea de SLA en 99%

Ejemplos de Reglas de Alerta

Alertas Críticas

Fallos de Enrutamiento:

```
alert: RoutingFailuresDetected
expr: increase(sms_c_routing_failed_count[5m]) > 0
for: 2m
labels:
  severity: critical
annotations:
  summary: "{{ $value }}" fallos de enrutamiento en los últimos 5 minutos"
  description: "Los mensajes no pueden ser enrutados. Verifique la configuración de enrutamiento."
```

Acumulación de Cola:

```
alert: MessageQueueBacklog
expr: sms_c_queue_size_pending > 10000
for: 5m
labels:
  severity: critical
annotations:
  summary: "La cola de mensajes tiene {{ $value }} mensajes pendientes"
  description: "La cola se está acumulando. Verifique el rendimiento de entrega."
```

Mensajes Antiguos en Cola:

```
alert: OldMessagesInQueue
expr: sms_c_queue_oldest_message_age_seconds > 300
for: 2m
labels:
  severity: critical
annotations:
  summary: "El mensaje más antiguo tiene {{ $value }} segundos"
  description: "Los mensajes no se están entregando. Verifique los frontends."
```

Todos los Frontends Desconectados:

```
alert: NoActiveFrontends
expr: sum(sms_c_frontend_status_count{status="connected"}) == 0
for: 1m
labels:
  severity: critical
annotations:
  summary: "No hay frontends conectados"
  description: "No hay ruta de entrega disponible. Verifique la conectividad del frontend."
```

Crecimiento de la Cola de Cartas Muertas:

```
alert: DeadLetterMessagesIncreasing
expr: rate(sms_c_delivery_dead_letter_count[10m]) > 0
for: 5m
labels:
  severity: critical
annotations:
  summary: "{{ $value }}" mensajes movidos a la cola de cartas
muertas"
  description: "Los mensajes se están volviendo indeseables.
Investigue los fallos."
```

Alertas de Advertencia

Baja Tasa de Éxito de Entrega:

```
alert: LowDeliverySuccessRate
expr: (rate(sms_c_delivery_succeeded_count[10m]) /
rate(sms_c_delivery_queued_count[10m])) < 0.95
for: 10m
labels:
  severity: warning
annotations:
  summary: "La tasa de éxito de entrega es {{ $value |
humanizePercentage }}"
  description: "La tasa de éxito está por debajo del 95%.
Investigue las fallas de entrega."
```

Alta Tasa de Reintentos:

```
alert: HighDeliveryRetryRate
expr: histogram_quantile(0.95,
sms_c_delivery_succeeded_attempt_count_bucket) > 2
for: 15m
labels:
  severity: warning
annotations:
  summary: "Intentos de entrega en el percentil 95: {{ $value }}"
  description: "Los mensajes requieren múltiples intentos.
Verifique la fiabilidad del destino."
```

Procesamiento de Mensajes Lento:

```
alert: SlowMessageProcessing
expr: histogram_quantile(0.95,
sms_c_message_processing_stop_duration_bucket) > 1000
for: 10m
labels:
  severity: warning
annotations:
  summary: "Tiempo de procesamiento en el percentil 95: {{ $value
}}ms"
  description: "El procesamiento de mensajes es lento. Verifique
los recursos del sistema."
```

Búsquedas ENUM Fallando:

```
alert: HighEnumFailureRate
expr: rate(sms_c_enum_lookup_stop_duration_count{success="false"}
[10m]) > 0.1
for: 10m
labels:
  severity: warning
annotations:
  summary: "Tasa de fallos de búsqueda ENUM: {{ $value }}"
  description: "Las búsquedas DNS están fallando. Verifique los
servidores DNS."
```

Baja Tasa de Aciertos en Caché ENUM:

```
alert: LowEnumCacheHitRate
expr: rate(sms_c_enum_cache_hit_count[10m]) /
(rate(sms_c_enum_cache_hit_count[10m]) +
rate(sms_c_enum_cache_miss_count[10m])) < 0.70
for: 30m
labels:
  severity: warning
annotations:
  summary: "Tasa de aciertos en caché ENUM: {{ $value |
humanizePercentage }}"
  description: "Baja eficiencia de caché. Puede indicar tráfico de
números únicos."
```

Fallos de Cobro:

```
alert: ChargingFailuresDetected
expr: rate(sms_c_charging_failed_count[10m]) > 0.05
for: 10m
labels:
  severity: warning
annotations:
  summary: "Tasa de fallos de cobro: {{ $value }}"
  description: "Errores en el sistema de cobro. Verifique la
conectividad con OCS."
```

Notas Adicionales

- Todas las métricas de duración utilizan precisión de nanosegundos internamente pero se convierten a milisegundos para su reporte.
- Las métricas de contador son acumulativas y deben ser utilizadas con las funciones `rate()` o `increase()` en las consultas de Prometheus.
- Las métricas de gauge representan valores instantáneos en el momento de la recolección.
- Las métricas de histograma proporcionan cálculos de percentiles (p50, p95, p99) y pueden ser utilizadas para crear mapas de calor.
- Todas las métricas incluyen etiquetas predeterminadas añadidas por Prometheus (instancia, trabajo, etc.).

- Al crear dashboards, use rangos de tiempo apropiados: 5m para tiempo real, 1h para tendencias, 24h+ para planificación de capacidad.
- Configure reglas de grabación en Prometheus para consultas complejas de uso frecuente para mejorar el rendimiento del dashboard.
- Utilice plantillas de variables en Grafana para dashboards dinámicos (seleccionar `dest_smsc`, `source_smsc`, etc.).

Guía de Traducción de Números SMS-C

[← Volver al Índice de Documentación](#) | [README Principal](#)

Descripción General

El sistema de Traducción de Números SMS-C proporciona una transformación flexible basada en regex de números de teléfono antes de la ruta. Las reglas de traducción pueden normalizar números, agregar prefijos internacionales, formatear números para puertas de enlace específicas y encadenar múltiples transformaciones. Las reglas se almacenan en Mnesia para persistencia y se pueden modificar en tiempo de ejecución sin interrupción del servicio.

Características Clave

- **Coincidencia basada en prefijos:** Coincidir números por prefijo antes de aplicar transformaciones
- **Transformación basada en regex:** Potente coincidencia y reemplazo de patrones con grupos de captura
- **Filtrado de SMSC de origen:** Aplicar diferentes traducciones según el origen del mensaje
- **Evaluación basada en prioridad:** Controlar el orden de las reglas con prioridades configurables (1-255)
- **Encadenamiento de reglas:** Continuar procesando a través de múltiples reglas con prevención de bucles
- **Transformaciones de llamada/recibida separadas:** Transformación independiente para números de origen y destino
- **Carga de archivo de configuración:** Cargar reglas iniciales desde `runtime.exs` en el primer inicio
- **Configuración en tiempo de ejecución:** Agregar, modificar o deshabilitar reglas sin reiniciar

- **Interfaz Web:** Interfaz completa CRUD para la gestión de reglas
- **Herramienta de simulación:** Probar la lógica de traducción con evaluación paso a paso
- **Copia de seguridad/Restauración:** Exportar e importar configuraciones de traducción
- **Integración previa a la ruta:** Traducciones aplicadas antes de la ruta para formatos de número consistentes

Arquitectura

Modelo de Datos

Cada regla de traducción contiene los siguientes campos:

Campo	Tipo	Descripción	Requerido
<code>rule_id</code>	entero	Identificador único autoincremental	Sí (auto)
<code>calling_prefix</code>	cadena/nil	Coincidencia de prefijo para el número de llamada (nil = comodín)	No
<code>called_prefix</code>	cadena/nil	Coincidencia de prefijo para el número llamado (nil = comodín)	No
<code>source_smsc</code>	cadena/nil	Nombre del SMSC de origen (nil = comodín)	No
<code>calling_match</code>	cadena/nil	Patrón regex para coincidir con el número de llamada	No
<code>calling_replace</code>	cadena/nil	Patrón de reemplazo para el número de llamada	No
<code>called_match</code>	cadena/nil	Patrón regex para coincidir con el número llamado	No
<code>called_replace</code>	cadena/nil	Patrón de reemplazo para el número llamado	No
<code>priority</code>	entero	Prioridad de la regla (1-255, menor = mayor prioridad)	Sí
<code>description</code>	cadena	Descripción legible por humanos	No
<code>enabled</code>	booleano	Habilitar/deshabilitar regla	Sí

Campo	Tipo	Descripción	Requerido
<code>continue</code>	booleano	Continuar evaluando reglas después de la coincidencia (predeterminado: falso)	No

Nota: Las reglas se evalúan en orden de prioridad (número más bajo primero). Solo se evalúan las reglas habilitadas.

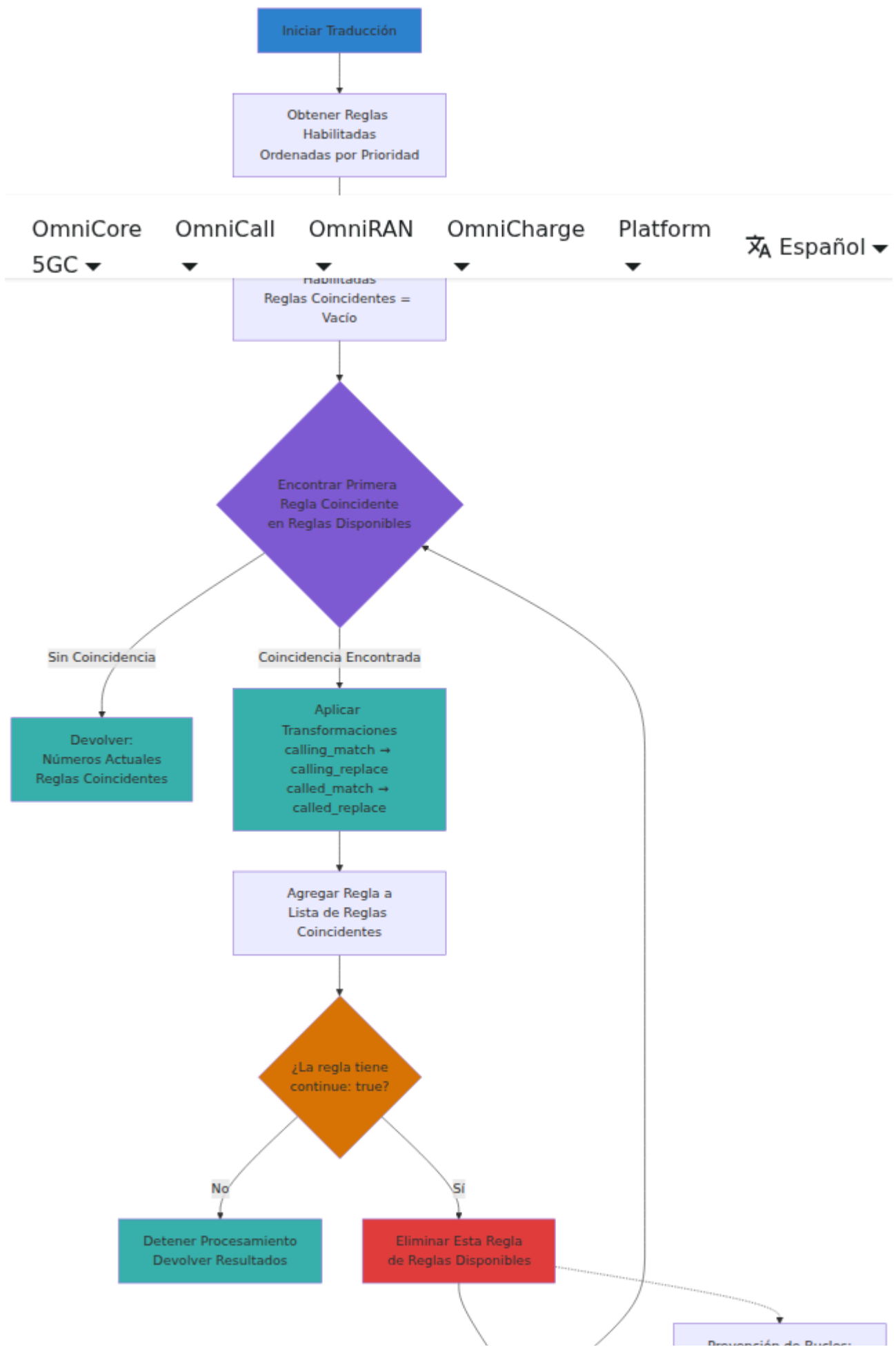
Algoritmo de Traducción

Al traducir números, el sistema:

- Recupera reglas habilitadas** ordenadas por prioridad (primero las más bajas)
- Evalúa reglas secuencialmente** contra los parámetros del mensaje:
 - Coincidir `calling_prefix` (si se especifica)
 - Coincidir `called_prefix` (si se especifica)
 - Coincidir `source_smsc` (si se especifica)
- Aplica la primera regla coincidente:**
 - Transformar el número de llamada usando `calling_match` y `calling_replace`
 - Transformar el número llamado usando `called_match` y `called_replace`
- Verifica la bandera de continuar:**
 - Si `continue: false` → Detener procesamiento, devolver resultado
 - Si `continue: true` → Eliminar la regla coincidente de las reglas disponibles, continuar con el paso 2 usando **números transformados**
- Devuelve los números finales** y la lista de todas las reglas aplicadas

Encadenamiento de Reglas con Prevención de Bucles

La bandera `continue` permite un poderoso encadenamiento de reglas mientras previene bucles infinitos:



Actualizar Números
Actuales
a Valores Transformados

Prevenición de bucles:
Una vez que una regla
coincide,
se elimina de
Reglas Disponibles
No puede coincidir de
nuevo

Comodines

- `nil` o valores vacíos actúan como comodines que coinciden con cualquier valor
- Una regla sin criterios de coincidencia es una regla de captura total
- Una regla sin patrones de transformación (`nil match/replace`) pasa números sin cambios

Ejemplo: Escenario de Encadenamiento de Reglas

```
Parse error on line 20: [...] style R1 fill:#38B2AC style R -----^  
Expecting 'SOLID_OPEN_ARROW', 'DOTTED_OPEN_ARROW', 'SOLID_ARROW',  
'BIDIRECTIONAL_SOLID_ARROW', 'DOTTED_ARROW',  
'BIDIRECTIONAL_DOTTED_ARROW', 'SOLID_CROSS', 'DOTTED_CROSS',  
'SOLID_POINT', 'DOTTED_POINT', got 'TXT'
```

Intente de nuevo

Configuración

Carga de Reglas desde el Archivo de Configuración

Las reglas de traducción se pueden definir en `config/runtime.exs` y se cargarán automáticamente en el primer inicio.

Importante: Las reglas de configuración solo se cargan cuando la tabla de traducción está **vacía** (primer inicio). Esto preserva las reglas agregadas a través de la interfaz web durante el tiempo de ejecución y previene duplicados en reinicios.

Flujo de Carga de Configuración

La Aplicación Inicia

¿Tabla de Traducción
Vacía?

Sí

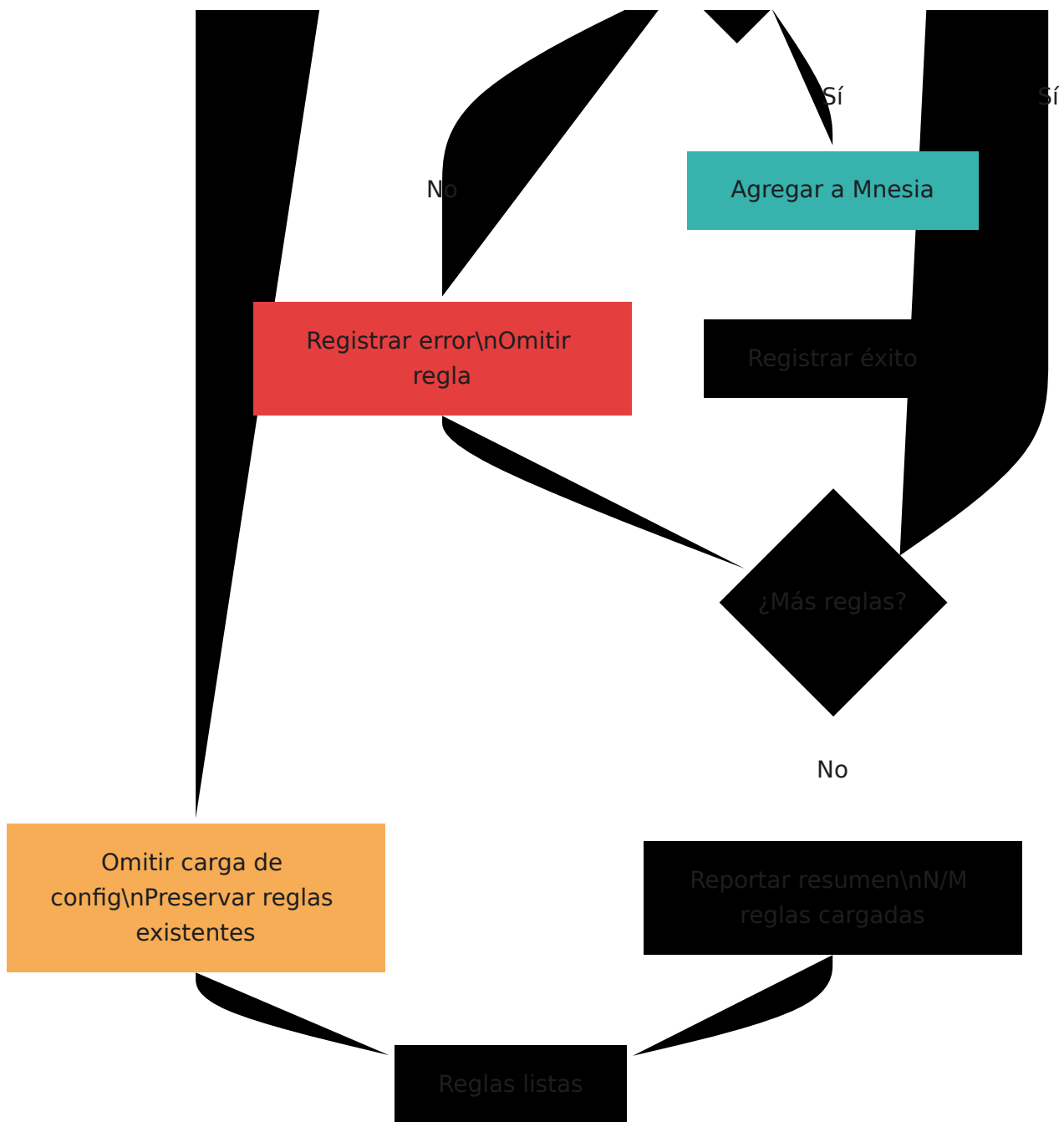
Cargar reglas
de\nconfig/runtime.exs

Para cada regla en la
config

Validar campos de regla

¿Válido?

No



Ejemplo de Configuración


```
# config/runtime.exs
config :sms_c, :translation_rules, [
  # Agregar +1 a números de 10 dígitos de EE. UU.
  %{
    calling_prefix: nil,
    called_prefix: nil,
    source_smsc: "us_domestic_smsc",
    calling_match: "^(\\d{10})$",
    calling_replace: "+1\\1",
    called_match: "^(\\d{10})$",
    called_replace: "+1\\1",
    priority: 10,
    description: "Agregar +1 a números de 10 dígitos de EE. UU.
desde SMSC doméstico",
    enabled: true,
    continue: false
  },

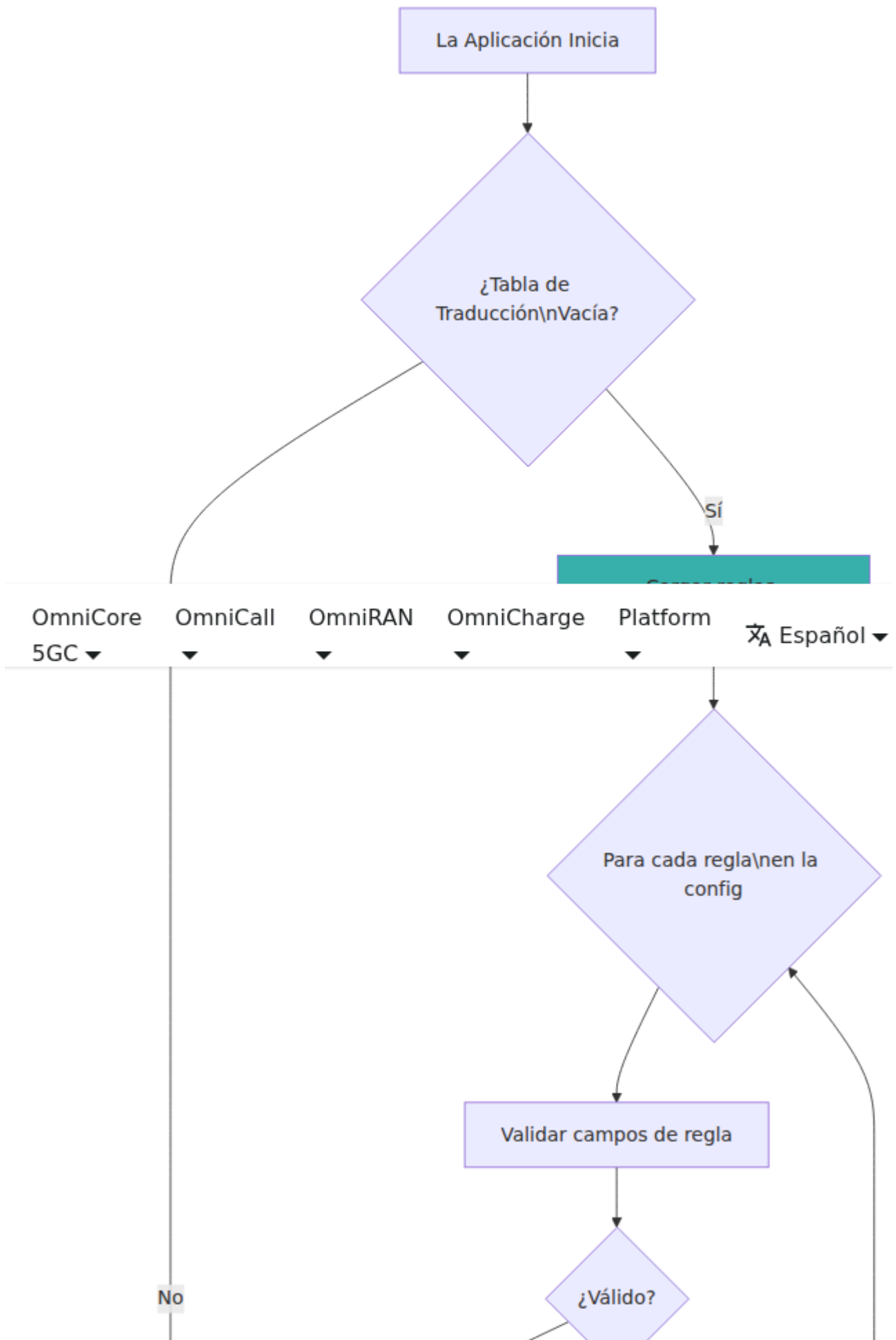
  # Eliminar ceros a la izquierda del formato internacional
  %{
    calling_prefix: "00",
    called_prefix: nil,
    source_smsc: nil,
    calling_match: "^00(.+)$",
    calling_replace: "+\\1",
    called_match: nil,
    called_replace: nil,
    priority: 5,
    description: "Convertir prefijo internacional 00 a +",
    enabled: true,
    continue: true # Continuar aplicando más formateo
  },

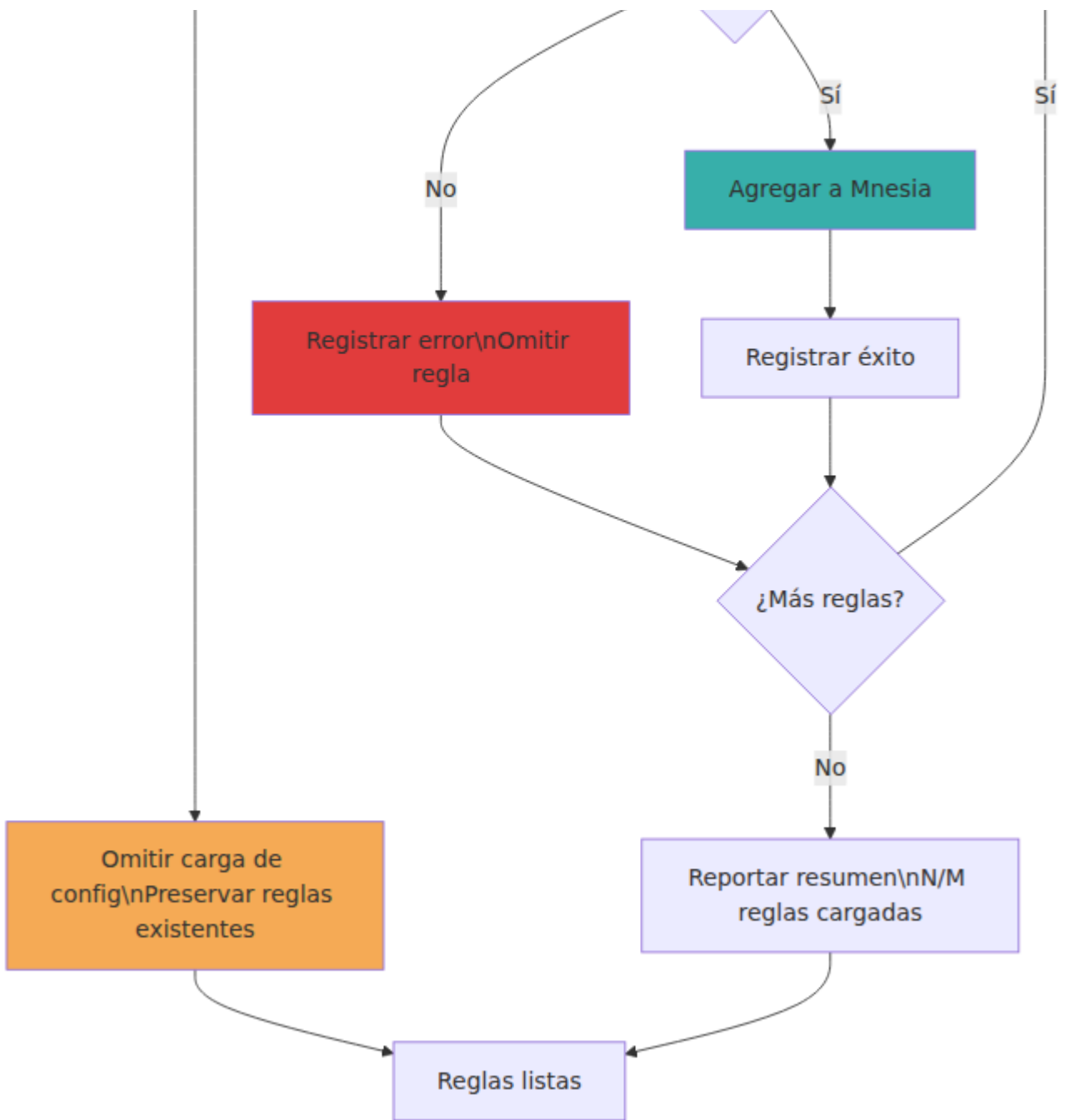
  # Formatear números del Reino Unido para puerta de enlace
específica
  %{
    calling_prefix: "+44",
    called_prefix: "+44",
    source_smsc: nil,
    calling_match: "^\\+44(.*)$",
    calling_replace: "0044\\1",
    called_match: "^\\+44(.*)$",
    called_replace: "0044\\1",
```

```
    priority: 20,  
    description: "Formatear números del Reino Unido para puerta de  
enlace heredada",  
    enabled: true,  
    continue: false  
  }  
]
```

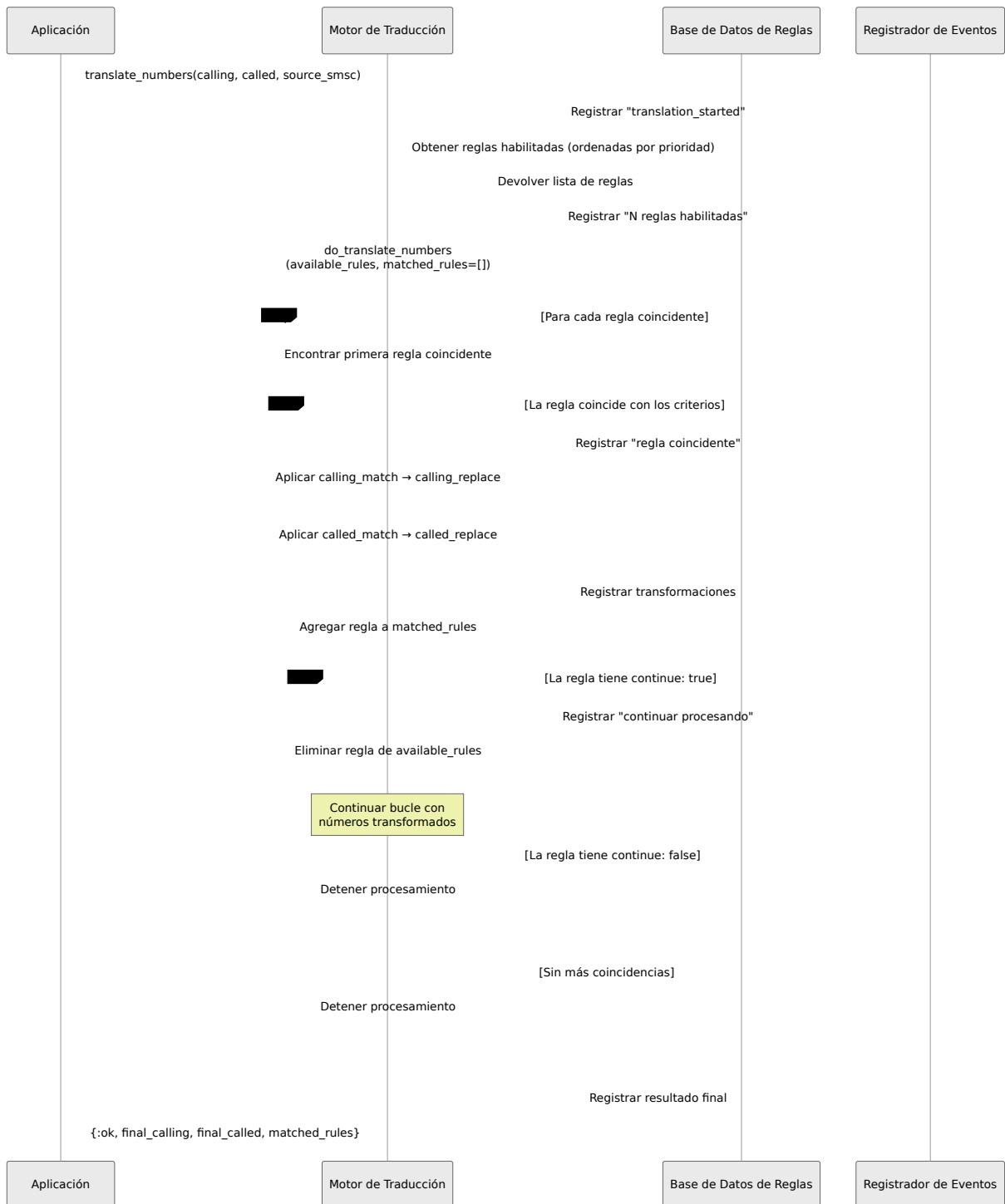
Comenzando

Flujo de Inicialización





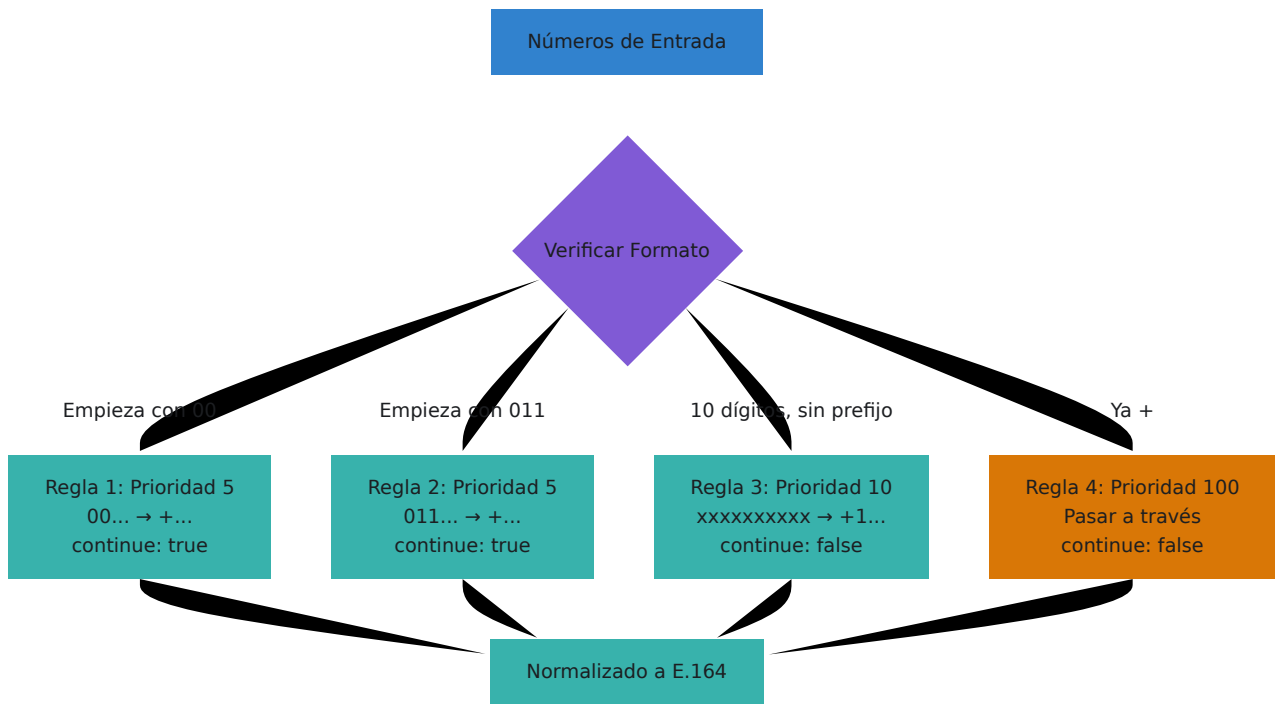
Flujo de Traducción de Mensajes



Casos de Uso Comunes

Normalización de Números Internacionales

Normalizar varios formatos internacionales a E.164:



Formato Específico de Puerta de Enlace

Encadenar reglas para formatear números según los requisitos específicos de la puerta de enlace:

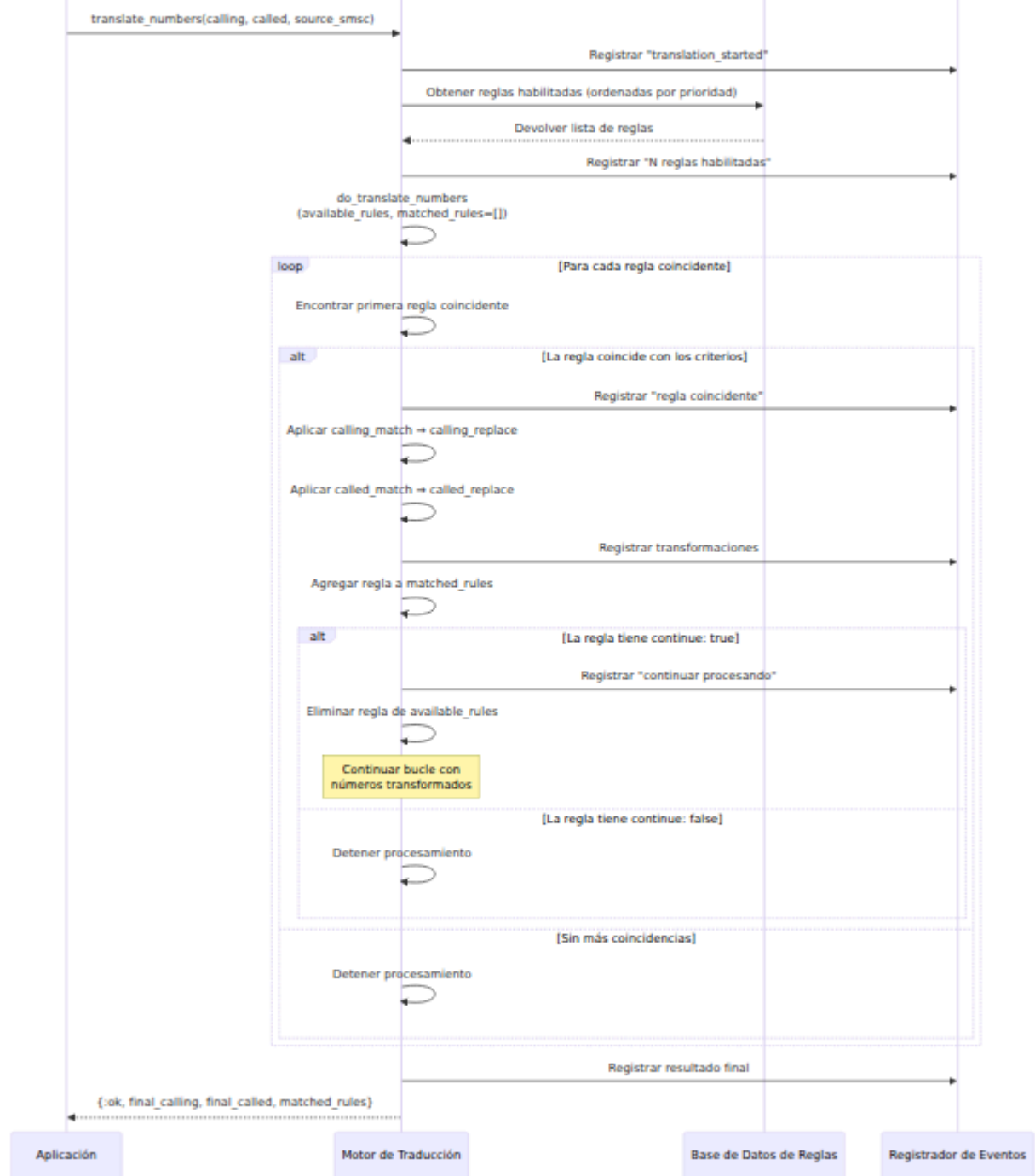
```

Parse error on line 2: ... TD I[Entrada: "5551234567"] --> S1[ -----
^ Expecting 'SQE', 'DOUBLECIRCLEEND', 'PE', '-)', 'STADIUMEND',
'SUBROUTINEEND', 'PIPE', 'CYLINDEREND', 'DIAMOND_STOP', 'TAGEND',
'TRAPEND', 'INVTRAPEND', 'UNICODE_TEXT', 'TEXT', 'TAGSTART', got 'STR'
  
```

Intente de nuevo

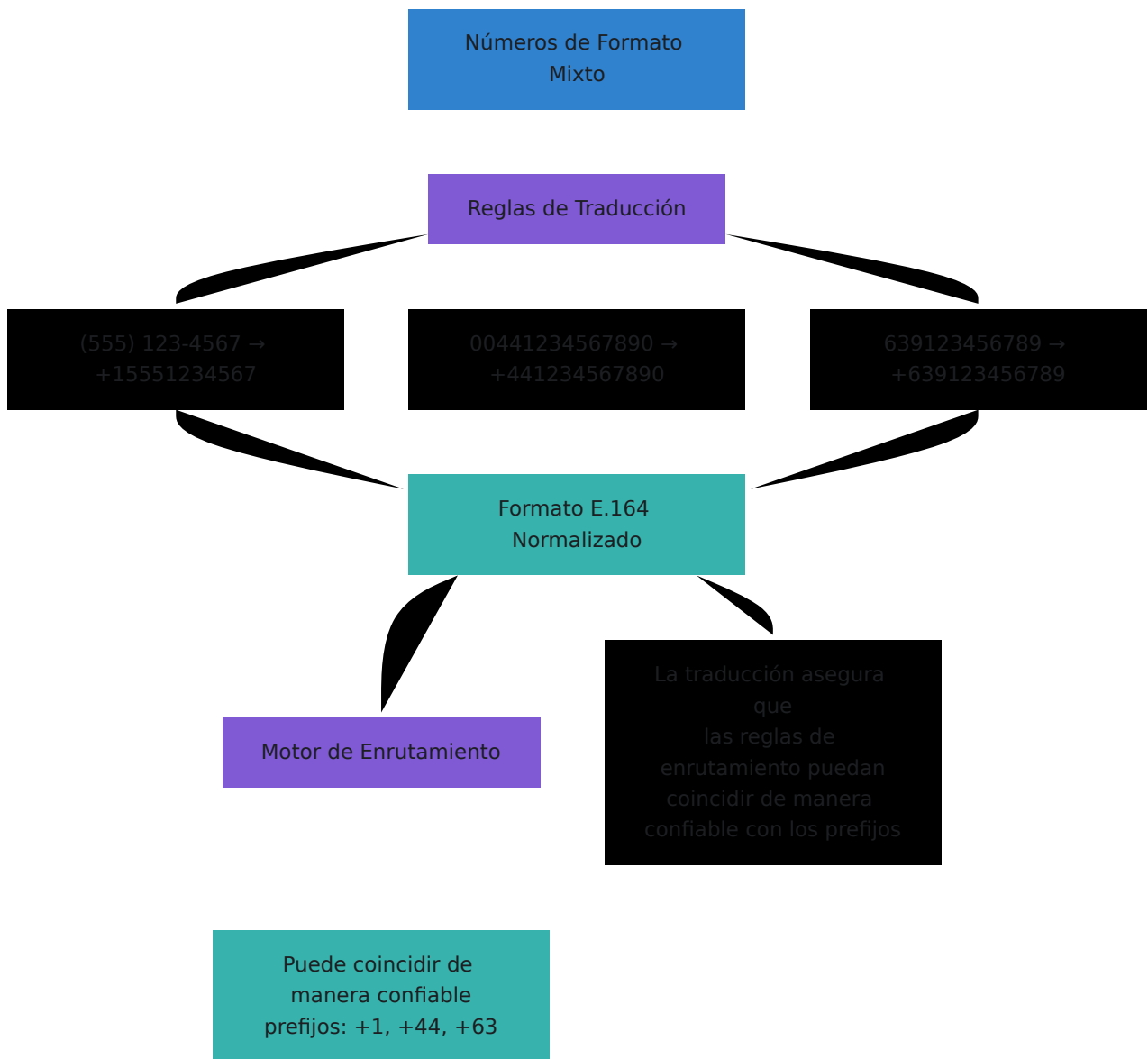
Traducciones Específicas de SMSC

Aplicar diferentes traducciones según la fuente del mensaje:



Preparación de Enrutamiento Basada en Prefijos

Normalizar números antes de enrutar para asegurar una coincidencia de prefijos consistente:



Manejo de Portabilidad de Números

Manejar números portados que requieren cambios de prefijo:

Parse error on line 18: ... style Input fill:#3182CE style R -----^
 Expecting 'SOLID_OPEN_ARROW', 'DOTTED_OPEN_ARROW', 'SOLID_ARROW',
 'BIDIRECTIONAL_SOLID_ARROW', 'DOTTED_ARROW',
 'BIDIRECTIONAL_DOTTED_ARROW', 'SOLID_CROSS', 'DOTTED_CROSS',
 'SOLID_POINT', 'DOTTED_POINT', got 'TXT'

Intente de nuevo

Interfaz Web

Interfaz de Gestión de Reglas de Traducción

Acceda a la interfaz de gestión de reglas en `/number_translation` (a través del menú de navegación):

Características:

- Ver todas las reglas en una tabla ordenable por prioridad
- Agregar nuevas reglas con validación de formularios
- Editar reglas existentes
- Habilitar/deshabilitar reglas sin eliminar
- Eliminar reglas con confirmación
- Indicador visual para reglas con `continue: true`
- Importar/Exportar reglas como JSON

Agregar una Regla:

1. Complete los criterios de coincidencia (opcional):
 - Prefijo de llamada (por ejemplo, "+1", "44")
 - Prefijo llamado (por ejemplo, "+639", "1555")
 - SMSC de origen (dejar vacío para cualquier)
2. Defina las transformaciones (opcional):
 - Coincidencia y reemplazo de regex para el número de llamada
 - Coincidencia y reemplazo de regex para el número llamado
3. Establezca la prioridad (1-255, menor = mayor prioridad)
4. Establezca el estado:
 - **Habilitado:** La regla está activa
 - **Continuar Procesando:** Continuar evaluando más reglas después de esta
5. Agregue una descripción
6. Haga clic en "Agregar Regla" o "Actualizar Regla"

Alternar Continuar Procesando:

- **Detener** (predeterminado): Detener el procesamiento después de que esta regla coincida
- **Continuar**: Aplicar esta regla y continuar evaluando las reglas restantes
- Las reglas con continuación habilitada muestran una insignia azul "↓ Continuar" en la tabla

Editar una Regla:

1. Haga clic en "Editar" junto a la regla
2. Modifique los campos según sea necesario
3. Haga clic en "Actualizar Regla"

Indicadores de Tabla de Reglas:

- La insignia **Habilitado/Deshabilitado** muestra el estado de la regla
- La insignia ↓ **Continuar** muestra las reglas que continuarán procesando
- La insignia **Prioridad** muestra el orden de evaluación
- Los patrones regex se muestran en fuente monoespaciada para mayor claridad

Simulador de Traducción

Acceda al simulador en /translation_simulator (a través del menú de navegación):

Características:

- Probar la lógica de traducción con números reales
- **Transformación Paso a Paso** mostrando cada regla aplicada
- Ver valores antes/después para cada transformación
- Ver qué reglas coincidieron y por qué
- Cargar escenarios de ejemplo para pruebas rápidas
- Ver historial de pruebas (últimas 10 pruebas)

Usando el Simulador:

1. Ingrese parámetros de prueba:

- Número de llamada (de)
 - Número llamado (a)
 - SMSC de origen (opcional)
2. Haga clic en "Probar Traducción"
 3. Ver resultados completos:
 - **Resultado de Traducción:** Números finales después de todas las transformaciones
 - **Reglas Aplicadas:** Conteo y lista de todas las reglas que coincidieron
 - **Transformaciones Paso a Paso:** Vista detallada de cada regla:
 - Número de paso e información de la regla
 - Descripción de la regla
 - Antes → Después para ambos números de llamada y llamados
 - Indicador "↓ Continuar" para reglas que continuaron procesando
 - Transformaciones resaltadas en verde
 - Valores sin cambios marcados como "pasados a través"
 4. Cargar ejemplos preconfigurados usando los botones de ejemplo
 5. Revisar el historial de pruebas para comparar diferentes escenarios

Ejemplo de Salida:

Resultado de Traducción

Número de Llamada: 5551234567 → +1-555-123-4567

Número Llamado: 9078720155 → +1-907-872-0155

✓ Traducido por 3 regla(s)

Transformaciones Paso a Paso

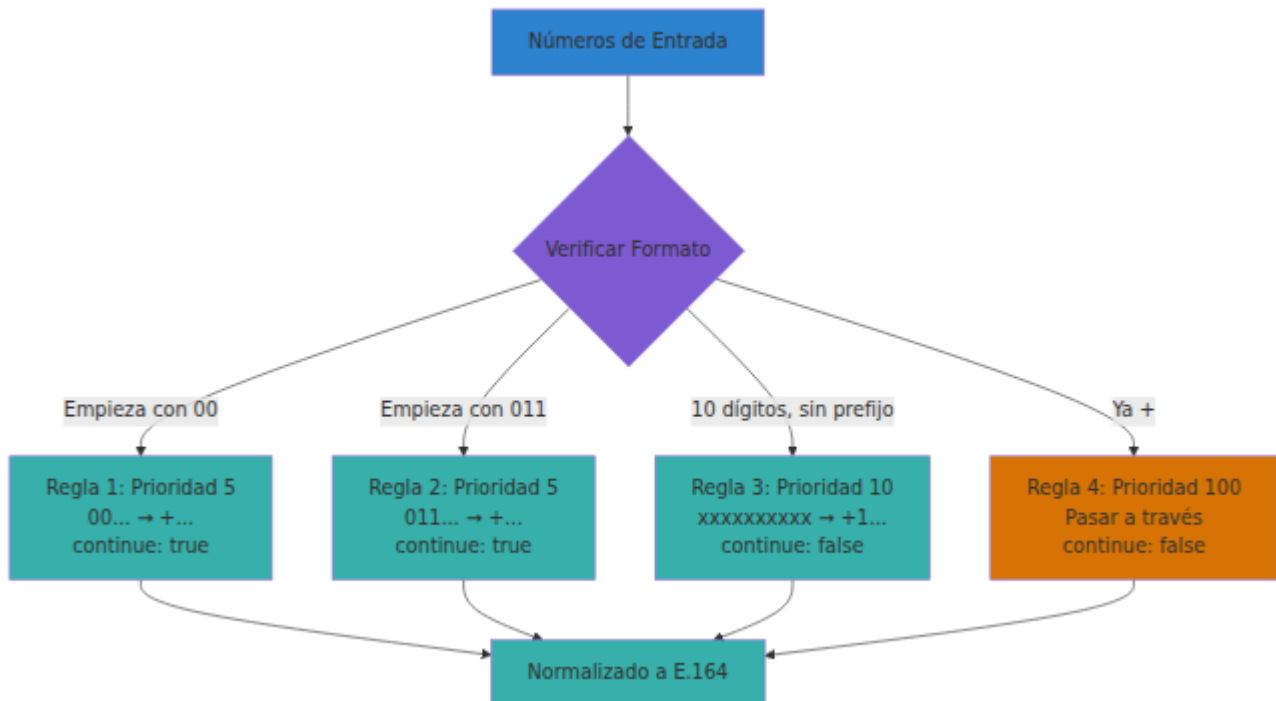
```
┌ Paso 1 ───────────────────────────────────────────────────┐
│ Regla #1 (Prioridad 10)          ↓ Continuar │
│ Agregar código de país a números de 10 dígitos      │
│ ─────────────────────────────────────────────────── │
│ Llamado: 9078720155 → +19078720155           │
│                               00                │
└──────────────────────────────────────────────────┘
```

```
┌ Paso 2 ───────────────────────────────────────────────────┐
│ Regla #2 (Prioridad 20)          ↓ Continuar │
│ Formatear código de área con guiones                │
│ ─────────────────────────────────────────────────── │
│ Llamado: +19078720155 → +1-907-8720155         │
└──────────────────────────────────────────────────┘
```

```
┌ Paso 3 ───────────────────────────────────────────────────┐
│ Regla #3 (Prioridad 30)                        │
│ Formato final para puerta de enlace                │
│ ─────────────────────────────────────────────────── │
│ Llamado: +1-907-8720155 → +1-907-872-0155 │
└──────────────────────────────────────────────────┘
```

Referencia de API

Descripción General de Operaciones Básicas



Parámetros de Traducción

translate_numbers acepta los siguientes parámetros:

- `calling_number` (opcional): Número de teléfono de origen
- `called_number` (opcional): Número de teléfono de destino
- `source_smsc` (opcional): Identificador de SMSC de origen
- `message_id` (opcional): Para registro de eventos

Devuelve:

- `{:ok, translated_calling, translated_called, [rules_applied]}` - Siempre exitoso
- Devuelve números originales si no coinciden reglas
- Devuelve lista de todas las reglas que se aplicaron (en orden)

```
# Ejemplo de uso
{:ok, new_calling, new_called, rules} =
  NumberTranslation.translate_numbers(
    calling_number: "5551234567",
    called_number: "9078720155",
    source_smsc: "domestic_gateway",
    message_id: "msg_123"
  )

# Verificar si ocurrió alguna traducción
if rules != [] do
  Logger.info("Se aplicaron #{length(rules)} reglas de
traducción")
  Enum.each(rules, fn rule ->
    Logger.info(" - Regla ##{rule.rule_id}: #{rule.description}")
  end)
end
```

Operaciones de Gestión de Reglas

```
# Agregar una nueva regla
{:ok, rule} = NumberTranslation.add_rule(%{
  calling_prefix: nil,
  called_prefix: nil,
  source_smsc: "gateway1",
  calling_match: "^(\\d{10})$",
  calling_replace: "+1\\1",
  called_match: "^(\\d{10})$",
  called_replace: "+1\\1",
  priority: 10,
  description: "Agregar +1 a números de 10 dígitos",
  enabled: true,
  continue: false
})

# Actualizar una regla
{:ok, updated_rule} = NumberTranslation.update_rule(rule_id, %{
  enabled: false,
  description: "Deshabilitado para pruebas"
})

# Eliminar una regla
:ok = NumberTranslation.delete_rule(rule_id)

# Obtener una regla específica
rule = NumberTranslation.get_rule(rule_id)

# Listar todas las reglas
all_rules = NumberTranslation.list_rules()

# Listar solo reglas habilitadas (ordenadas por prioridad)
enabled_rules = NumberTranslation.list_enabled_rules()
```

Operaciones de Importación/Exportación

```
# Exportar todas las reglas
backup = NumberTranslation.export_rules()
# Devuelve: %{
#   version: "1.0",
#   exported_at: ~U[2024-01-15 10:30:00Z],
#   count: 5,
#   rules: [...]
# }

# Guardar en archivo JSON
json = Jason.encode!(backup, pretty: true)
File.write!("translation_rules_backup.json", json)

# Importar reglas (fusionar con existentes)
{:ok, %{imported: 3, failed: 0}} =
  NumberTranslation.import_rules(backup, mode: :merge)

# Importar reglas (reemplazar todas las existentes)
{:ok, %{imported: 5, failed: 0}} =
  NumberTranslation.import_rules(backup, mode: :replace)
```

Mejores Prácticas

Diseño de Reglas

1. Mantener prioridades organizadas:

- **1-10**: Reglas de normalización críticas (agregar códigos de país, corregir formatos)
- **11-50**: Formateo específico de puerta de enlace
- **51-100**: Transformaciones opcionales
- **101+**: Reglas de captura total o de depuración

2. Usar continuar estratégicamente:

- Habilitar `continue: true` para reglas de normalización que preparan números para un procesamiento adicional
- Deshabilitar `continue: false` para reglas de formateo final
- Evitar cadenas largas (máximo 3-4 reglas) para mantener el rendimiento

3. Documentar sus reglas:

- Siempre agregar descripciones claras
- Incluir ejemplos en la descripción (por ejemplo, "5551234567 → +15551234567")
- Documentar el propósito y la entrada/salida esperada

4. Probar patrones regex:

- Probar patrones con el simulador antes de implementar
- Usar grupos de captura (`\1`, `\2`) para transformaciones flexibles
- Escapar caracteres especiales de regex (puntos, paréntesis, etc.)

Rendimiento

1. Minimizar el conteo de reglas:

- Combinar reglas similares donde sea posible
- Usar coincidencia de prefijos para reducir evaluaciones de regex
- Eliminar o deshabilitar reglas no utilizadas

2. Optimizar patrones regex:

- Usar coincidencia de prefijos primero (más rápido que regex)
- Mantener patrones regex simples
- Evitar patrones que consumen mucho tiempo de retroceso

3. Limitar el encadenamiento de reglas:

- Cadenas largas (5+ reglas) pueden afectar el rendimiento
- Considerar combinar múltiples pasos en una regla si es posible
- Monitorear la latencia de traducción con métricas de Telemetría

Operaciones

1. Probar antes de implementar:

- Usar el simulador con ejemplos del mundo real
- Probar casos límite (números vacíos, caracteres especiales)
- Verificar el comportamiento de la bandera continuar

2. Hacer copias de seguridad regularmente:

- Exportar reglas antes de hacer cambios importantes
- Controlar versiones de sus exportaciones
- Probar importaciones en no producción primero

3. Monitorear traducciones:

- Habilitar registro de message_id para depuración
- Verificar registros de eventos para decisiones de traducción
- Monitorear qué reglas se están aplicando

4. Despliegue gradual:

- Agregar nuevas reglas como deshabilitadas primero
- Probar con el simulador
- Habilitar y monitorear
- Ajustar según sea necesario

Consejos de Regex

1. Patrones comunes:

- Número de EE. UU. de 10 dígitos: `^\d{10}$`
- Formato internacional: `^\+(\d+)$`
- Eliminar ceros a la izquierda: `^0+(.)$`
- Agregar guiones: `^\d{3}\d{3}\d{4}$` → `\1-\2-\3`

2. Grupos de captura:

- Usar paréntesis para capturar: `^(\\d{3})(\\d{7})$`
- Referencia en reemplazo: `+1\\1\\2`
- Múltiples capturas: `^\\+(\\d{1,3})(\\d+)$` → `00\\1\\2`

3. Escapar caracteres especiales:

- Punto literal: `\\.`
- Más literal: `\\+`
- Paréntesis literales: `\\(` o `\\)`

Solución de Problemas

Regla No Coincidente

Síntoma: Se espera que la regla no coincida, los números pasan sin cambios

Causas posibles:

- El prefijo no coincide (verificar coincidencia exacta de prefijo)
- El SMSC de origen no coincide
- El patrón regex no coincide con el formato de entrada
- La regla está deshabilitada
- Una regla de mayor prioridad coincidió primero (con `continue: false`)

Soluciones:

1. Usar el simulador para ver qué reglas se evalúan
2. Verificar el estado de la regla (habilitada/deshabilitada)
3. Verificar coincidencia de prefijos (sensible a mayúsculas)
4. Probar el patrón regex por separado
5. Verificar el orden de prioridad

Transformación Incorrecta Aplicada

Síntoma: Número transformado pero el resultado es incorrecto

Causas posibles:

- El patrón regex coincide pero el patrón de reemplazo es incorrecto
- Múltiples reglas aplicándose en un orden inesperado
- Referencias de grupos de captura incorrectas (\1, \2, etc.)

Soluciones:

1. Usar el simulador para ver las transformaciones paso a paso
2. Verificar que el patrón regex capture los grupos correctos
3. Comprobar la sintaxis del patrón de reemplazo
4. Probar regex en un probador de regex en línea
5. Revisar la prioridad de la regla y las banderas de continuar

Bucle Infinito / Degradación del Rendimiento

Síntoma: La traducción toma mucho tiempo o parece colgarse

Nota: Esto no debería suceder debido a la prevención de bucles, pero si ocurre:

Causas posibles:

- Error en la lógica de prevención de bucles
- Evaluación de regex extremadamente larga
- Cadena de reglas muy larga

Soluciones:

1. Verificar los registros de la aplicación en busca de errores
2. Revisar reglas con continue: true
3. Simplificar patrones regex
4. Reducir el número de reglas encadenadas
5. Informar un error si la prevención de bucles falló

Encadenamiento de Reglas Inesperado

Síntoma: Se aplican más reglas de las esperadas

Causas posibles:

- Las reglas tienen continue: true cuando no deberían
- El orden de prioridad permite múltiples coincidencias
- El número transformado coincide con reglas adicionales

Soluciones:

1. Usar el simulador para ver la cadena de reglas exacta
2. Revisar las banderas de continuar en todas las reglas
3. Ajustar prioridades para controlar el orden
4. Establecer continue: false en la regla final

Traducción No Aplicada Antes del Enrutamiento

Síntoma: El enrutador ve números no traducidos

Causas posibles:

- La traducción no está integrada en el flujo del mensaje
- La traducción ocurre después del enrutamiento
- El código de la aplicación omite la traducción

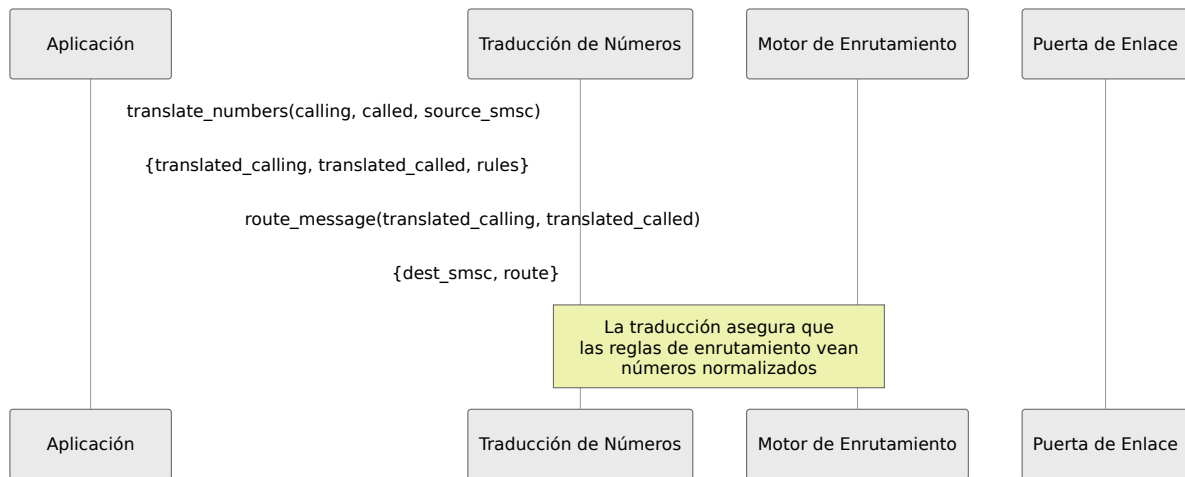
Soluciones:

1. Verificar la integración de la aplicación: la traducción debe llamarse antes del enrutamiento
2. Verificar la tubería de procesamiento del mensaje
3. Revisar registros de eventos para eventos de traducción
4. Asegurarse de que translate_numbers se llame en el orden correcto

Temas Avanzados

Integración con Enrutamiento

La traducción ocurre **antes** del enrutamiento para asegurar formatos de número consistentes:



Registro de Eventos

Las decisiones de traducción se registran a través del EventLogger:

- `translation_started`: La traducción comienza
- `translation_candidates`: Número de reglas habilitadas
- `translation_matched`: Regla coincidente y aplicada
- `translation_calling`: Número de llamada transformado
- `translation_called`: Número llamado transformado
- `translation_continue`: La regla tiene `continue=true`, continuando evaluación
- `translation_none`: No coincidieron reglas

Habilitar el registro pasando `message_id` a `translate_numbers/1`.

Métricas de Telemetría

Monitorear el rendimiento de la traducción con Telemetría:

```
:telemetry.attach(  
  "number-translation-handler",  
  [:sms_c, :number_translation, :translate, :stop],  
  fn _event_name, measurements, metadata, _config ->  
    # measurements: %{duration: microseconds}  
    # metadata: %{rules_applied: count, ...}  
  end,  
  nil  
)
```

Métricas clave a monitorear:

- Duración de la traducción (p50, p95, p99)
- Reglas aplicadas por mensaje
- Reglas coincidentes vs no coincidentes
- Uso de la bandera continuar

Clustering

Las tablas de Mnesia se distribuyen automáticamente entre nodos en clúster. Las reglas de traducción se replican para alta disponibilidad.

Parse error on line 25: ... style New fill:#3182CE style P -----^
Expecting 'SOLID_OPEN_ARROW', 'DOTTED_OPEN_ARROW', 'SOLID_ARROW',
'BIDIRECTIONAL_SOLID_ARROW', 'DOTTED_ARROW',
'BIDIRECTIONAL_DOTTED_ARROW', 'SOLID_CROSS', 'DOTTED_CROSS',
'SOLID_POINT', 'DOTTED_POINT', got 'TXT'

Intente de nuevo

Estrategias de Migración

Al implementar nuevas reglas de traducción:

Planificación de Nuevas Reglas

1. Diseñar reglas fuera de línea

2. Probar en simulador

¿Las reglas funcionan correctamente?

Sí

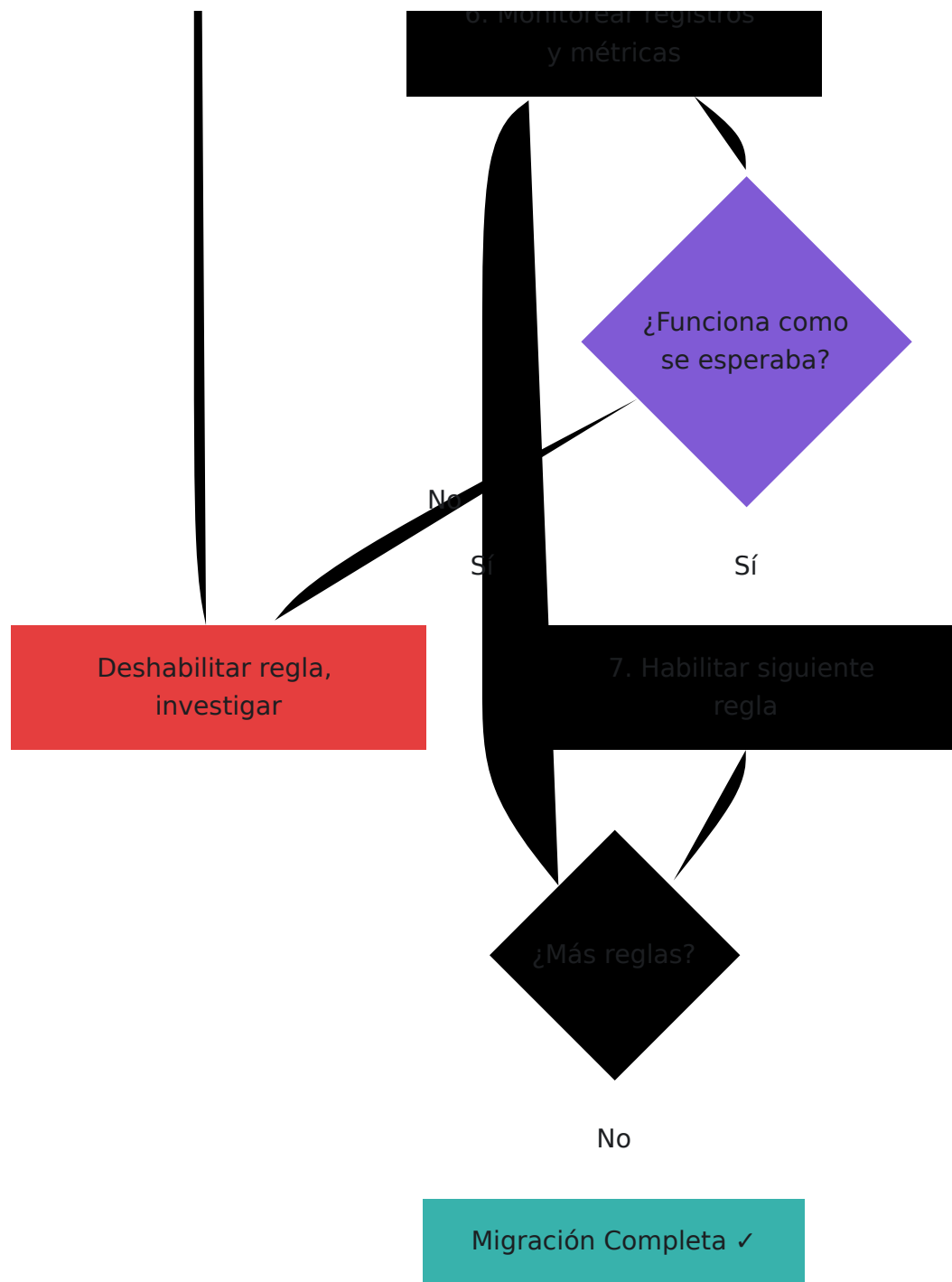
Depurar patrones

3. Agregar reglas como deshabilitadas

4. Desplegar en producción

5. Habilitar reglas una a la vez

No



Ejemplos

Ejemplo 1: Normalización de Números de EE. UU.

Requisito: Convertir varios formatos de números de EE. UU. a E.164 (+1XXXXXXXXXX)


```
# Regla 1: Números de 10 dígitos (prioridad más alta)
%{
  calling_match: "^(\\d{10})$",
  calling_replace: "+1\\1",
  called_match: "^(\\d{10})$",
  called_replace: "+1\\1",
  priority: 5,
  description: "Agregar +1 a números de 10 dígitos sin prefijo",
  enabled: true,
  continue: false
}

# Regla 2: 1 + 10 dígitos (prioridad media)
%{
  calling_match: "^1(\\d{10})$",
  calling_replace: "+1\\1",
  called_match: "^1(\\d{10})$",
  called_replace: "+1\\1",
  priority: 10,
  description: "Convertir 1XXXXXXXXXX a +1XXXXXXXXXX",
  enabled: true,
  continue: false
}

# Casos de prueba:
# "5551234567" → "+15551234567" (Regla 1)
# "15551234567" → "+15551234567" (Regla 2)
# "+15551234567" → "+15551234567" (Sin coincidencia, pasar a través)
```

Ejemplo 2: Conversión de Prefijo Internacional con Encadenamiento

Requisito: Convertir prefijo 00 a +, luego formatear para puerta de enlace

```

# Regla 1: Convertir 00 a + (continúa a la siguiente regla)
%{
  calling_match: "^00(.+)$",
  calling_replace: "+\1",
  called_match: "^00(.+)$",
  called_replace: "+\1",
  priority: 5,
  description: "Convertir prefijo internacional 00 a +",
  enabled: true,
  continue: true # Continuar para formatear
}

# Regla 2: Formatear para puerta de enlace (detiene el
procesamiento)
%{
  calling_match: "^+(\d+)$",
  calling_replace: "00\1",
  called_match: "^+(\d+)$",
  called_replace: "00\1",
  priority: 10,
  description: "Formatear números + como 00 para puerta de
enlace",
  enabled: true,
  continue: false # Detener después de esto
}

# Caso de prueba:
# Paso 1: "00441234567890" → "+441234567890" (Regla 1, continuar)
# Paso 2: "+441234567890" → "00441234567890" (Regla 2, detener)
# Resultado: "00441234567890"
# Reglas aplicadas: [Regla 1, Regla 2]

```

Ejemplo 3: Manejo Específico de SMSC

Requisito: Aplicar diferentes reglas según el SMSC de origen

```
# Regla 1: SMSC de confianza - pasar a través (prioridad 5)
%{
  source_smsc: "trusted_gateway",
  calling_match: nil, # Sin transformación
  calling_replace: nil,
  called_match: nil,
  called_replace: nil,
  priority: 5,
  description: "Pasar números de puerta de enlace confiable",
  enabled: true,
  continue: false
}

# Regla 2: SMSC no confiable - normalizar (prioridad 10)
%{
  source_smsc: "untrusted_gateway",
  calling_match: "^(.*)$",
  calling_replace: "+VALIDATE\1",
  called_match: "^(.*)$",
  called_replace: "+VALIDATE\1",
  priority: 10,
  description: "Agregar prefijo de validación para fuente no
confiable",
  enabled: true,
  continue: false
}

# Regla 3: Captura total para otros SMSCs (prioridad 100)
%{
  source_smsc: nil, # comodín
  calling_match: "^(\\d{10})$",
  calling_replace: "+1\1",
  called_match: "^(\\d{10})$",
  called_replace: "+1\1",
  priority: 100,
  description: "Por defecto: Agregar +1 a números de 10 dígitos",
  enabled: true,
  continue: false
}
```

Ejemplo 4: Cadena de Formateo de Múltiples Pasos

Requisito: Normalizar → Agregar código de país → Formatear con guiones

```
# Regla 1: Eliminar ceros a la izquierda (continuar)
%{
  calling_match: "^0+(.)$",
  calling_replace: "\1",
  called_match: "^0+(.)$",
  called_replace: "\1",
  priority: 5,
  description: "Eliminar ceros a la izquierda",
  enabled: true,
  continue: true
}

# Regla 2: Agregar código de país si falta (continuar)
%{
  calling_match: "^(\\d{10})$",
  calling_replace: "+1\\1",
  called_match: "^(\\d{10})$",
  called_replace: "+1\\1",
  priority: 10,
  description: "Agregar +1 a números de 10 dígitos",
  enabled: true,
  continue: true
}

# Regla 3: Formatear con guiones (detener)
%{
  calling_match: "^\\+1(\\d{3})(\\d{3})(\\d{4})$",
  calling_replace: "+1-\\1-\\2-\\3",
  called_match: "^\\+1(\\d{3})(\\d{3})(\\d{4})$",
  called_replace: "+1-\\1-\\2-\\3",
  priority: 15,
  description: "Formatear como +1-XXX-XXX-XXXX",
  enabled: true,
  continue: false
}

# Caso de prueba:
# Entrada: "005551234567"
# Paso 1: "005551234567" → "5551234567" (Regla 1, continuar)
# Paso 2: "5551234567" → "+15551234567" (Regla 2, continuar)
# Paso 3: "+15551234567" → "+1-555-123-4567" (Regla 3, detener)
```

```
# Resultado: "+1-555-123-4567"
```

```
# Reglas aplicadas: [Regla 1, Regla 2, Regla 3]
```

Soporte

Para problemas o preguntas:

- Verifique la suite de pruebas en `test/sms_c/messaging/number_translation_test.exs` para ejemplos
- Use el simulador para depurar la lógica de traducción
- Revise los registros de eventos para decisiones de traducción
- Verifique el contenido de la tabla Mnesia:
`:mnesia.table_info(:translation_rule, :size)`
- Monitoree las métricas de Telemetría para problemas de rendimiento

Guía de Operaciones de SMS-C

[← Volver al Índice de Documentación](#) | [README Principal](#)

Procedimientos operativos diarios, monitoreo y tareas de mantenimiento para los equipos de operaciones de SMS-C.

Tabla de Contenidos

- [Operaciones Diarias](#)
- [Monitoreo](#)
- [Seguimiento de Mensajes](#)
- [Gestión de Rutas](#)
- [Gestión de Frontend](#)
- [Gestión de Traducción de Números](#)
- [Mantenimiento del Sistema](#)
- [Copia de Seguridad y Recuperación](#)
- [Planificación de Capacidad](#)
- [Respuesta a Incidentes](#)

Operaciones Diarias

Verificación de Salud Matutina

Realice estas verificaciones al inicio de cada día:

1. Comprobar el Estado del Sistema

```
# Verificación de salud de la API
curl https://api.example.com:8443/api/status

# Respuesta esperada:
# {"status":"ok","application":"OmniMessage","timestamp":"2025-10-30T08:00:00Z"}
```

2. Revisar Métricas de Prometheus

Acceda al panel de Prometheus y verifique:

- Rendimiento de mensajes (últimas 24 horas)
- Tasa de fallos de enrutamiento (debería ser < 1%)
- Acumulación de cola (debería ser < 1000 pendientes)
- Tasa de éxito de entrega (debería ser > 95%)
- Estado de conexión del frontend (todos los frontends esperados activos)

3. Comprobar la Cola de Mensajes

Acceda a la interfaz web: https://sms-admin.example.com/message_queue

Revise:

- Total de mensajes pendientes (debería ser bajo)
- Edad del mensaje más antiguo (debería ser < 5 minutos)
- Mensajes con altos intentos de entrega (investigar si > 3)
- Mensajes de carta muerta (investigar cualquier presente)

4. Revisar el Estado del Frontend

Acceda a la interfaz web: https://sms-admin.example.com/frontend_status

Verifique:

- Todos los frontends esperados están activos
- No hay desconexiones no expiradas
- No hay errores de frontend en las últimas 24 horas

5. Comprobar los Registros de la Aplicación

Acceda a la interfaz web: `https://sms-admin.example.com/logs` o verifique los archivos de registro

Busque:

- Mensajes de nivel de error
- Fallos de enrutamiento
- Fallos de carga
- Problemas de conexión a la base de datos
- Problemas con nodos del clúster

Monitoreo del Volumen de Mensajes

Comprobar Cuentas de Mensajes por Hora:

Utilice la consulta de Prometheus:

```
# Mensajes recibidos por hora
increase(sms_c_message_received_count[1h])

# Mensajes entregados por hora
increase(sms_c_delivery_succeeded_count[1h])

# Calcular tasa de entrega
rate(sms_c_delivery_succeeded_count[1h]) /
rate(sms_c_message_received_count[1h])
```

Patrones Esperados:

- Horas laborales: Mayor volumen
- Noches/fin de semana: Menor volumen
- Tasa de entrega: Debería ser > 95%

Condiciones de Alerta:

- Caída repentina en mensajes (> 50% de disminución)

- Aumento repentino en mensajes (> 200% de aumento)
- Caída de la tasa de entrega por debajo del 90%

Monitoreo

Métricas Clave a Observar

Métricas de Procesamiento de Mensajes

Conteo de Mensajes Recibidos (`sms_c_message_received_count`):

- **Qué:** Total de mensajes que ingresan al sistema
- **Alerta:** Caída o aumento repentino
- **Consulta:** `rate(sms_c_message_received_count[5m])`

Duración del Procesamiento de Mensajes

(`sms_c_message_processing_stop_duration`):

- **Qué:** Tiempo de procesamiento de extremo a extremo
- **Alerta:** p95 > 1000ms
- **Consulta:** `histogram_quantile(0.95, sms_c_message_processing_stop_duration)`

Métricas de Enrutamiento

Fallos de Enrutamiento (`sms_c_routing_failed_count`):

- **Qué:** Mensajes que no pudieron ser enrutados
- **Alerta:** Cualquier fallo (> 0)
- **Consulta:** `increase(sms_c_routing_failed_count[5m])`

Ruta Coincidente (`sms_c_routing_route_matched_count`):

- **Qué:** Qué rutas están siendo utilizadas
- **Alerta:** Rutas de alta prioridad no coinciden
- **Consulta:** `sms_c_routing_route_matched_count`

Métricas de Entrega

Tasa de Éxito de Entrega:

- **Qué:** Porcentaje de entregas exitosas
- **Alerta:** Tasa < 95%
- **Consulta:** `rate(sms_c_delivery_succeeded_count[5m]) / rate(sms_c_delivery_queued_count[5m])`

Intentos de Entrega (`sms_c_delivery_succeeded_attempt_count`):

- **Qué:** Reintentos necesarios para la entrega
- **Alerta:** p95 > 2 (demasiados reintentos)
- **Consulta:** `histogram_quantile(0.95, sms_c_delivery_succeeded_attempt_count)`

Métricas de Cola

Tamaño de Cola (`sms_c_queue_size_size`):

- **Qué:** Total de mensajes en cola
- **Alerta:** Tamaño > 10,000
- **Consulta:** `sms_c_queue_size_size`

Edad del Mensaje Más Antiguo

(`sms_c_queue_oldest_message_age_seconds`):

- **Qué:** Edad del mensaje pendiente más antiguo
- **Alerta:** Edad > 300 segundos
- **Consulta:** `sms_c_queue_oldest_message_age_seconds`

Configuración del Panel de Control

Paneles del Dashboard Operacional:

1. Rendimiento de Mensajes (Gráfico)

- Mensajes recibidos (tasa de 5 minutos)
- Mensajes entregados (tasa de 5 minutos)

- Rango de tiempo: Últimas 24 horas

2. **Estado de la Cola** (Estadísticas Únicas)

- Mensajes pendientes actuales
- Edad del mensaje más antiguo
- Conteo de mensajes fallidos

3. **Rendimiento de Entrega** (Gráfico)

- Tasa de éxito a lo largo del tiempo
- Tasa de fallos a lo largo del tiempo
- Rango de tiempo: Últimas 24 horas

4. **Estado de Enrutamiento** (Tabla)

- ID de Ruta
- Conteo de coincidencias (última hora)
- SMSC de destino
- Prioridad

5. **Estado del Frontend** (Tabla)

- Nombre del frontend
- Estado (activo/expirado)
- Última vez visto
- Conteo de mensajes (última hora)

6. **Salud del Sistema** (Estadísticas Únicas)

- Tiempo de respuesta de la API (p95)
- Tiempo de consulta de la base de datos (p95)
- Tiempo de búsqueda de ENUM (p95)

Configuración de Alertas

Alertas Críticas (Se Requiere Respuesta Inmediata):

```
# No se encontró ruta - los mensajes no pueden ser entregados
- alert: RoutingFailures
  expr: increase(sms_c_routing_failed_count[5m]) > 0
  severity: critical
  description: "{{ $value }}" mensajes fallaron en el enrutamiento
en los últimos 5 minutos"

# Cola acumulándose - procesamiento quedándose atrás
- alert: QueueBacklog
  expr: sms_c_queue_size_pending > 10000
  severity: critical
  description: "La cola tiene {{ $value }}" mensajes pendientes"

# Mensajes envejeciendo - entrega atascada
- alert: OldMessagesInQueue
  expr: sms_c_queue_oldest_message_age_seconds > 300
  severity: critical
  description: "El mensaje más antiguo tiene {{ $value }}" segundos
de antigüedad"

# Frontend desconectado - sin ruta de entrega
- alert: FrontendDisconnected
  expr: sms_c_frontend_status_count{status="disconnected"} > 0
  severity: critical
  description: "{{ $value }}" frontends desconectados"
```

Alertas de Advertencia (Se Necesita Investigación):

```

# Tasa de éxito de entrega cayendo
- alert: LowDeliveryRate
  expr: rate(sms_c_delivery_succeeded_count[10m]) /
rate(sms_c_delivery_queued_count[10m]) < 0.90
  severity: warning
  description: "La tasa de éxito de entrega es {{ $value }}"

# Demasiados reintentos de entrega
- alert: HighRetryRate
  expr: histogram_quantile(0.95,
sms_c_delivery_succeeded_attempt_count) > 2
  severity: warning
  description: "Intentos de entrega en el percentil 95: {{ $value
}}"

# Búsquedas de ENUM lentas o fallando
- alert: SlowEnumLookups
  expr: histogram_quantile(0.95, sms_c_enum_lookup_stop_duration)
> 5000
  severity: warning
  description: "Búsquedas de ENUM tardando > 5 segundos"

# Baja tasa de aciertos de caché de ENUM
- alert: LowEnumCacheHitRate
  expr: rate(sms_c_enum_cache_hit_count[10m]) /
(rate(sms_c_enum_cache_hit_count[10m]) +
rate(sms_c_enum_cache_miss_count[10m])) < 0.70
  severity: warning
  description: "Tasa de aciertos de caché de ENUM: {{ $value }}"

```

Seguimiento de Mensajes

Encontrar Mensaje Específico

Por ID de Mensaje:

1. **Interfaz Web:** Navegue a `/message_queue`
2. Ingrese el ID del mensaje en el cuadro de búsqueda
3. Vea los detalles completos y el historial de eventos

A través de la API:

```
curl https://api.example.com:8443/api/messages/12345
```

Por Número de Teléfono:

1. **Interfaz Web:** Navegue a `/message_queue`
2. Ingrese el número de teléfono en el cuadro de búsqueda
3. Vea todos los mensajes para ese número

Rastrear el Ciclo de Vida del Mensaje

Ver Historial de Eventos:

1. **Interfaz Web:** Haga clic en el mensaje en la cola, vea la sección "Eventos"
2. **API:** `GET /api/events/12345`

Secuencia Común de Eventos:

1. `message_inserted` - Mensaje creado
↓
2. `number_translated` - Números normalizados (si está configurado)
📞
3. `message_routed` - Decisión de enrutamiento tomada
↓
4. `charging_attempted` - Verificación de carga (si está habilitada)
↓
5. `message_delivered` - Entregado con éxito

Secuencia de Entrega Fallida:

1. message_inserted
↓
2. message_routed
↓
3. delivery_attempt_1 - Primer intento fallido
↓
4. delivery_attempt_2 - Segundo intento fallido (2min de retraso)
↓
5. delivery_attempt_3 - Tercer intento fallido (4min de retraso)
↓
6. message_dead_letter - Se superó el límite de reintentos

Comprobar Estado de Entrega

Mensajes Pendientes:

- Estado: "pendiente"
- deliver_after: Marca de tiempo futura
- delivery_attempts: 0 o número bajo

Mensajes Entregados:

- Estado: "entregado"
- deliver_time: Marca de tiempo de entrega
- dest_smsc: Frontend que entregó

Mensajes Fallidos:

- Estado: "pendiente" con altos delivery_attempts
- deadletter: true (si expirado)
- Verifique el registro de eventos para razones de fallo

Enrutamiento de Mensajes Basado en Ubicación

El SMS-C admite la recuperación de mensajes basada en la ubicación, lo que permite a los frontends recibir automáticamente mensajes destinados a suscriptores registrados en su ubicación.

Cómo Funciona:

Cuando un frontend consulta mensajes pendientes utilizando `get_messages_for_smsc(smsc_name)`, el sistema devuelve mensajes de dos maneras:

1. **Enrutamiento Explícito** - Mensajes donde `dest_smsc` coincide explícitamente con el nombre del frontend
2. **Enrutamiento Basado en Ubicación** - Mensajes donde:
 - `dest_smsc` es `null` (no enrutado explícitamente)
 - `destination_msisdn` tiene un registro de ubicación activo
 - El campo `location` de la ubicación coincide con el nombre del frontend
 - La ubicación no ha expirado

Escenario de Ejemplo:

Un suscriptor con MSISDN `+447700900123` se registra en el frontend `uk_gateway`:

```
# Suscriptor se registra (crea registro de ubicación)
POST /api/locations
{
  "msisdn": "+447700900123",
  "imsi": "234150123456789",
  "location": "uk_gateway",
  "expires": "2025-11-01T12:00:00Z"
}
```

Cuando llega un mensaje para este suscriptor sin enrutamiento explícito:

```
# Mensaje enviado sin dest_smsc
POST /api/messages
{
  "source_msisdn": "+15551234567",
  "destination_msisdn": "+447700900123",
  "message_body": "Hello",
  "source_smsc": "api"
  # Nota: dest_smsc es null
}
```

El frontend `uk_gateway` recibirá automáticamente este mensaje cuando realice una consulta:

```
# Frontend consulta mensajes
GET /api/messages/queue?smsc=uk_gateway

# Devuelve el mensaje aunque dest_smsc sea null
# porque el suscriptor de destino está registrado en uk_gateway
```

Requisitos de Ubicación:

Para que el enrutamiento basado en ubicación funcione:

- La tabla `locations` debe tener una entrada para el `destination_msisdn`
- El campo `location` debe coincidir con el nombre del SMSC que consulta
- La marca de tiempo `expires` debe estar en el futuro

Monitoreo del Enrutamiento Basado en Ubicación:

Verifique los registros de ubicación:

```
# A través de la API
GET /api/locations/{msisdn}

# Verifique si la ubicación ha expirado
# el campo expires debe ser > tiempo actual
```

Problemas Comunes:

- **Mensaje no entregado:** Verifique si la ubicación ha expirado
- **Frontend incorrecto:** Verifique que el campo `location` coincida con el nombre del frontend esperado
- **Ubicación no encontrada:** El suscriptor puede necesitar volver a registrarse

Intervenciones Manuales

Reintentar Mensaje Fallido:

```
# Restablecer delivery_attempts y deliver_after
curl -X PATCH https://api.example.com:8443/api/messages/12345 \
  -H "Content-Type: application/json" \
  -d '{
    "delivery_attempts": 0,
    "deliver_after": "2025-10-30T12:00:00Z"
  }'
```

Cambiar Destino:

```
# Enrutar a un SMSC diferente
curl -X PATCH https://api.example.com:8443/api/messages/12345 \
  -H "Content-Type: application/json" \
  -d '{
    "dest_smsc": "backup_gateway"
  }'
```

Eliminar Mensaje Atascado:

```
curl -X DELETE https://api.example.com:8443/api/messages/12345
```

Gestión de Rutas

Ver Rutas Actuales

Interfaz Web: Navegue a `/sms_routing`

A través de la API:

```
# Listar todas las rutas
curl https://api.example.com:8443/api/routes
```

Comprobar Uso de Rutas:

Consulta de Prometheus:

```
# Mensajes enrutados por cada ruta (última hora)
increase(sms_c_routing_route_matched_count[1h])
```

Agregar Nueva Ruta

Interfaz Web:

1. Navegue a `/sms_routing`
2. Haga clic en "Agregar Nueva Ruta"
3. Complete los campos:
 - **Prefijo de Llamada:** Prefijo del número de origen (opcional)
 - **Prefijo Llamado:** Prefijo del número de destino (requerido para enrutamiento geográfico)
 - **SMSC de Origen:** Filtro del sistema de origen (opcional)
 - **SMSC de Destino:** Puerta de enlace de destino (requerido a menos que sea respuesta automática/caída)
 - **Solo En Red:** Restringir esta ruta solo a destinos en red (requiere Diameter/HSS)
 - **Prioridad:** Prioridad de la ruta (1-255, menor = mayor prioridad)
 - **Peso:** Peso de balanceo de carga (1-100)

- **Descripción:** Descripción legible por humanos
- **Habilitado:** Marcar para activar inmediatamente

4. Haga clic en "Guardar Ruta"

Ejemplo: Ruta Geográfica:

- Prefijo Llamado: +44
- SMSC de Destino: uk_gateway
- Prioridad: 50
- Peso: 100
- Descripción: "Enrutamiento del Reino Unido"

Ejemplo: Ruta de Balanceo de Carga:

Cree dos rutas con los mismos criterios pero diferentes pesos:

Ruta 1:

- Prefijo Llamado: +44
- SMSC de Destino: uk_primary
- Prioridad: 50
- Peso: 70
- Descripción: "Reino Unido primario (70%)"

Ruta 2:

- Prefijo Llamado: +44
- SMSC de Destino: uk_backup
- Prioridad: 50
- Peso: 30
- Descripción: "Reino Unido de respaldo (30%)"

Ejemplo: Ruta Solo En Red:

Restringir a un transportista externo o aplicación interna para que solo envíe a suscriptores en red. Los mensajes a destinos fuera de la red se eliminan.

- SMSC de Origen: carrier_smp_bind

- SMSC de Destino: `local_msc`
- Solo En Red: marcado
- Prioridad: 50
- Peso: 100
- Descripción: "Transportista X — solo en red"

Esto requiere que Diameter/HSS esté habilitado. Si el HSS no está disponible, los mensajes se eliminan (fallo cerrado).

Probar Rutas

Simulador de Enrutamiento:

1. Navegue a `/simulador`
2. Ingrese parámetros de prueba:
 - Número de Llamada: `+15551234567`
 - Número Llamado: `+447700900000`
 - SMSC de Origen: (opcional)
 - Tipo de Origen: (opcional)
3. Haga clic en "Simular Enrutamiento"
4. Revise los resultados:
 - **Ruta Seleccionada:** Qué ruta fue elegida
 - **Todas las Coincidencias:** Qué rutas coincidieron con los criterios
 - **Evaluación:** Por qué cada ruta coincidió o no coincidió

Probar Antes de la Producción:

- Pruebe todas las nuevas rutas en el simulador
- Verifique que se seleccione la ruta correcta
- Verifique el orden de prioridad
- Valide la distribución del peso

Modificar Ruta Existente

Interfaz Web:

1. Navegue a `/sms_routing`
2. Encuentre la ruta en la lista
3. Haga clic en "Editar"
4. Modifique los campos
5. Haga clic en "Guardar Ruta"

Modificaciones Comunes:

- **Deshabilitar Ruta:** Desmarcar "Habilitado" (eliminación temporal)
- **Ajustar Peso:** Cambiar distribución de balanceo de carga
- **Cambiar Prioridad:** Reordenar evaluación de ruta
- **Actualizar Destino:** Cambiar a un SMSC diferente

Eliminar Ruta

Interfaz Web:

1. Navegue a `/sms_routing`
2. Encuentre la ruta en la lista
3. Haga clic en "Eliminar"
4. Confirme la eliminación

Advertencia: Eliminar rutas es permanente. Considere deshabilitar en su lugar.

Exportar/Importar Rutas

Exportar Rutas (Copia de Seguridad):

1. Navegue a `/sms_routing`
2. Haga clic en "Exportar Rutas"
3. Guarde el archivo JSON

Importar Rutas:

1. Navegue a `/sms_routing`
2. Haga clic en "Importar Rutas"

3. Seleccione el archivo JSON
4. Elija el modo de importación:
 - **Fusionar**: Agregar a rutas existentes
 - **Reemplazar**: Eliminar todo e importar

Casos de Uso:

- Copia de seguridad antes de cambios importantes
- Copiar rutas entre entornos
- Recuperación ante desastres
- Versionado de configuración

Gestión de Frontend

Monitorear Conexiones de Frontend

Interfaz Web: Navegue a `/frontend_status`

Verifique:

- Todos los frontends esperados están "activos"
- Los tiempos de última vista son recientes (< 90 segundos)
- No hay frontends expirados inesperados

A través de la API:

```
# Obtener frontends activos
curl https://api.example.com:8443/api/frontends/active

# Obtener estadísticas
curl https://api.example.com:8443/api/frontends/stats
```

Investigar Desconexiones

Frontend Expirado:

1. Verifique los registros del frontend para errores
2. Verifique la conectividad de red al SMS-C
3. Confirme que el frontend esté en ejecución
4. Verifique la lógica de registro del frontend (debería volver a registrarse cada 60s)

Registro No Mostrando:

1. Verifique que el frontend esté llamando a POST `/api/frontends/register`
2. Verifique los registros de la API para errores de registro
3. Verifique el formato del payload JSON
4. Pruebe el registro manualmente con curl

Ejemplo de Registro Manual:

```
curl -X POST https://api.example.com:8443/api/frontends/register \
-H "Content-Type: application/json" \
-d '{
  "frontend_name": "test_gateway",
  "frontend_type": "smpp",
  "ip_address": "10.0.1.50",
  "hostname": "gateway.example.com"
}'
```

Ver Historial de Frontend

Interfaz Web:

1. Navegue a `/frontend_status`
2. Encuentre el frontend en la lista
3. Haga clic en "Historial"
4. Revise los registros de registros pasados

A través de la API:

```
curl https://api.example.com:8443/api/frontends/history/uk_gateway
```

Casos de Uso:

- Investigar la fiabilidad de la conexión
- Rastrear patrones de tiempo de actividad del frontend
- Identificar cambios de configuración

Gestión de Traducción de Números

Las reglas de traducción de números se gestionan a través de `config/runtime.exs`. Los cambios requieren reiniciar la aplicación.

Ver Reglas de Traducción Activas

Verifique el archivo de configuración:

```
cat config/runtime.exs | grep -A 20 "translation_rules:"
```

Tareas Comunes de Traducción

Agregar Código de País a Números Locales:

Edite `config/runtime.exs`:

```
%{
  calling_prefix: nil,
  called_prefix: nil,
  source_smsc: nil,
  calling_match: "^(\\d{10})$",
  calling_replace: "+1\\1",
  called_match: "^(\\d{10})$",
  called_replace: "+1\\1",
  priority: 100,
  description: "Agregar +1 a números de EE. UU. de 10 dígitos",
  enabled: true
}
```

Normalizar Formato Internacional:

```
%{
  calling_prefix: nil,
  called_prefix: nil,
  source_smsc: nil,
  calling_match: "^00(\d+)$",
  calling_replace: "+\1",
  called_match: "^00(\d+)$",
  called_replace: "+\1",
  priority: 10,
  description: "Convertir prefijo 00 a +",
  enabled: true
}
```

Eliminación de Código Específico del Transportista:

```
%{
  calling_prefix: nil,
  called_prefix: "101",
  source_smsc: "carrier_a",
  calling_match: nil,
  calling_replace: nil,
  called_match: "^101(\d+)$",
  called_replace: "\1",
  priority: 5,
  description: "Eliminar código de transportista del transportista A",
  enabled: true
}
```

Probar Reglas de Traducción

Después de los cambios de configuración:

1. Reinicie la aplicación para cargar nuevas reglas
2. Envíe un mensaje de prueba con origen/destino que debería coincidir
3. Verifique el registro de eventos para el evento `number_translated`
4. Verifique que los números se transformaron correctamente

Deshabilitar Regla de Traducción

Establezca `enabled: false` en la regla:

```
%{  
  ...  
  enabled: false  
}
```

Reinicie la aplicación.

Mantenimiento del Sistema

Mantenimiento de Base de Datos

Comprobar Tamaño de la Base de Datos:

Utilice sus herramientas de gestión de bases de datos para monitorear el tamaño de almacenamiento de CDR:

- **MySQL/MariaDB:** Consultar `information_schema.tables` para el tamaño de la base de datos
- **PostgreSQL:** Utilizar la función `pg_database_size()` o el comando `\l+` en `psql`

Limpiar Registros de CDR Antiguos:

Los registros de CDR deben archivarse y purgarse periódicamente según su política de retención:

- Configure el archivo de registro automático según los requisitos comerciales (típicamente 30-90 días en la base de datos operativa)
- Archive registros más antiguos a un almacén de datos o almacenamiento en frío
- Elimine registros archivados de la base de datos operativa en lotes para evitar contención de bloqueo

Optimizar Tablas:

Periódicamente optimice las tablas de la base de datos para mantener el rendimiento:

- **MySQL/MariaDB:** Ejecute el comando `OPTIMIZE TABLE` durante períodos de bajo tráfico
- **PostgreSQL:** Ejecute `VACUUM ANALYZE` regularmente (o habilite autovacuum)

Ejecutar Semanalmente durante períodos de bajo tráfico para mantener un rendimiento óptimo.

Mantenimiento de Base de Datos Mnesia

Comprobar Tamaño de la Tabla Mnesia:

```
# En la consola IEx
:mnesia.table_info(:sms_route, :size)
:mnesia.table_info(:translation_rule, :size)
```

Respaldar Tablas Mnesia:

```
# Exportar rutas (Interfaz Web)
# Navegue a /sms_routing
# Haga clic en "Exportar Rutas"

# O a través del respaldo de Mnesia
:mnesia.backup("/var/backups/sms_c/mnesia_backup.bup")
```

Restaurar Mnesia:

```
# A través de la importación de la Interfaz Web
# O restaurar respaldo:
:mnesia.restore("/var/backups/sms_c/mnesia_backup.bup", [])
```

Rotación de Registros

Configure logrotate para los registros de la aplicación:

```
# /etc/logrotate.d/sms_c
/var/log/sms_c/*.log {
    daily
    rotate 30
    compress
    delaycompress
    notifempty
    create 0644 sms_user sms_group
    sharedscripts
    postrotate
        systemctl reload sms_c || true
    endscript
}
```

Reiniciar Aplicación

Reinicio Graceful (cero tiempo de inactividad en clúster):

```
# Reinicie un nodo a la vez
systemctl restart sms_c

# Espere a que el nodo se una al clúster
# Repita para cada nodo
```

Reinicio de Emergencia (todos los nodos):

```
systemctl restart sms_c
```

Después del Reinicio:

- Verifique que todos los frontends se reconecten
- Verifique Prometheus para la continuidad de métricas
- Monitoree los registros para errores

- Verifique que el procesamiento de mensajes se reanude

Copia de Seguridad y Recuperación

Qué Respaldar

1. Archivos de Configuración:

- `config/runtime.exs`
- `config/config.exs`
- `config/prod.exs` (si existe)

2. Tablas de Enrutamiento (Mnesia):

- Exportar a través de la Interfaz Web
- O comando de respaldo de Mnesia

3. Base de Datos SQL CDR:

- Copia de seguridad completa diaria
- Copias de seguridad de registros de transacciones (continuas)

4. Certificados TLS:

- `priv/cert/*.crt`
- `priv/cert/*.key`

Procedimientos de Copia de Seguridad

Copia de Seguridad de Configuración Diaria:

```
#!/bin/bash
# /opt/sms_c/scripts/backup_config.sh

BACKUP_DIR="/var/backups/sms_c/$(date +%Y%m%d)"
mkdir -p $BACKUP_DIR

# Copia de seguridad de configuración
cp -r /opt/sms_c/config $BACKUP_DIR/

# Copia de seguridad de certificados
cp -r /opt/sms_c/priv/cert $BACKUP_DIR/

# Establecer permisos
chmod 600 $BACKUP_DIR/cert/*

echo "Copia de seguridad de configuración completada: $BACKUP_DIR"
```

Copia de Seguridad de Base de Datos:

```
#!/bin/bash
# /opt/sms_c/scripts/backup_database.sh

BACKUP_DIR="/var/backups/sms_c/database"
DATE=$(date +%Y%m%d_%H%M%S)

mkdir -p $BACKUP_DIR

# Copia de seguridad de la base de datos SQL CDR
# MySQL/MariaDB: Use mysqldump con --single-transaction para
consistencia
# PostgreSQL: Use pg_dump -F c para formato personalizado

# Estructura de ejemplo (adapte a su base de datos):
# - Use la herramienta de copia de seguridad apropiada (mysqldump,
pg_dump)
# - Habilite copias de seguridad seguras para consistencia
# - Comprimir salida para ahorrar espacio
# - Configurar período de retención (por ejemplo, 30 días)

# Eliminar copias de seguridad antiguas
find $BACKUP_DIR -name "sms_c_*.gz" -mtime +30 -delete

echo "Copia de seguridad de base de datos completada:
sms_c_{$DATE}"
```

Copia de Seguridad de Tablas de Enrutamiento:

```
#!/bin/bash
# /opt/sms_c/scripts/backup_routes.sh

BACKUP_DIR="/var/backups/sms_c/routes"
DATE=$(date +%Y%m%d)

mkdir -p $BACKUP_DIR

# Exportar a través de la API
curl https://api.example.com:8443/api/routes/export \
  > $BACKUP_DIR/routes_{$DATE}.json

echo "Copia de seguridad de rutas completada: routes_{$DATE}.json"
```

Programar Copias de Seguridad (crontab):

```
# Diario a las 2 AM
0 2 * * * /opt/sms_c/scripts/backup_config.sh
0 2 * * * /opt/sms_c/scripts/backup_database.sh
0 2 * * * /opt/sms_c/scripts/backup_routes.sh
```

Procedimientos de Recuperación

Restaurar Configuración:

```
# Detener aplicación
systemctl stop sms_c

# Restaurar archivos de configuración
cp -r /var/backups/sms_c/20251030/config/* /opt/sms_c/config/

# Restaurar certificados
cp -r /var/backups/sms_c/20251030/cert/* /opt/sms_c/priv/cert/

# Iniciar aplicación
systemctl start sms_c
```

Restaurar Base de Datos SQL CDR:

Utilice las herramientas de restauración apropiadas para su base de datos:

- **MySQL/MariaDB:** Descomprima y pase al cliente mysql
- **PostgreSQL:** Use pg_restore con copias de seguridad de formato personalizado

Importante: Detenga la aplicación SMS-C antes de restaurar la base de datos para evitar conflictos de datos.

Restaurar Tablas de Enrutamiento:

1. Navegue a la Interfaz Web `/sms_routing`
2. Haga clic en "Importar Rutas"

3. Seleccione el archivo JSON de respaldo
4. Elija el modo "Reemplazar"
5. Confirme la importación

Planificación de Capacidad

Monitorear Tendencias de Crecimiento

Tendencia de Volumen de Mensajes:

Consulta de Prometheus (promedio de 30 días):

```
avg_over_time(sms_c_message_received_count[30d])
```

Tasa de Crecimiento de la Base de Datos:

```
-- Crecimiento de datos mensual
SELECT
  DATE_FORMAT(inserted_at, '%Y-%m') AS month,
  COUNT(*) AS message_count,
  ROUND(SUM(LENGTH(message_body)) / 1024 / 1024, 2) AS data_mb
FROM message_queues
GROUP BY month
ORDER BY month DESC
LIMIT 12;
```

Indicadores de Capacidad

Uso de CPU:

- **Normal:** < 50% promedio
- **Alto:** > 70% sostenido
- **Crítico:** > 90%

Uso de Memoria:

- **Normal:** < 70% de disponible
- **Alto:** > 80%
- **Crítico:** > 90%

Uso de Disco:

- **Normal:** < 60% lleno
- **Alto:** > 75%
- **Crítico:** > 85%

Profundidad de Cola:

- **Normal:** < 1000 pendientes
- **Alto:** > 5000 pendientes
- **Crítico:** > 10,000 pendientes

Recomendaciones de Escalado

Cuándo Escalar Verticalmente (Actualizar Recursos):

- CPU consistentemente > 70%
- Memoria consistentemente > 80%
- Cuello de botella de un solo nodo

Cuándo Escalar Horizontalmente (Agregar Nodos):

- CPU > 50% en todos los nodos
- Volumen de mensajes > 5,000 msg/sec
- Distribución geográfica necesaria
- Alta disponibilidad requerida

Escalado de Base de Datos:

- Réplicas de lectura para consultas de informes
- Optimización de agrupamiento de conexiones
- Optimización de índices
- Particionar tablas grandes por fecha

Respuesta a Incidentes

Niveles de Severidad

Crítico (Respuesta Inmediata):

- No se entregan mensajes
- Todos los frontends desconectados
- Base de datos no disponible
- API completamente caída

Alto (Respuesta en 1 hora):

- Tasa de éxito de entrega < 80%
- Múltiples frontends desconectados
- Fallos de enrutamiento > 10%
- Acumulación de cola creciente

Medio (Respuesta en 4 horas):

- Un solo frontend desconectado
- Tasa de éxito de entrega 80-95%
- Procesamiento de mensajes lento
- Búsquedas de ENUM fallando

Bajo (Respuesta en 24 horas):

- Degradación menor del rendimiento
- Problema de ruta única
- Alertas de advertencia no críticas

Lista de Verificación de Incidentes

1. Evaluar Severidad:

- Verifique las alertas de Prometheus
- Revise las métricas del panel de control

- Verifique el estado de la cola de mensajes
- Verifique las conexiones de frontend

2. Reunir Información:

- ¿Cambios recientes en la configuración?
- ¿Despliegues recientes?
- Estado de dependencias externas (OCS, DNS)?
- Mensajes de error en los registros?

3. Acciones Inmediatas:

- Detener cambios en curso
- Revertir despliegues recientes si se sospecha que son la causa
- Habilitar registro detallado si es necesario
- Notificar a las partes interesadas

4. Investigación:

- Revisar registros de la aplicación
- Verificar uso de recursos del sistema
- Examinar el rendimiento de la base de datos
- Probar dependencias externas

5. Resolución:

- Aplicar solución
- Probar en simulador
- Desplegar en producción
- Monitorear para mejoras

6. Post-Incidente:

- Documentar causa raíz
- Actualizar monitoreo/alertas
- Implementar medidas preventivas
- Actualizar manuales de procedimientos

Incidentes Comunes

Alta Acumulación de Cola:

1. Verifique la tasa de éxito de entrega
2. Verifique que los frontends estén conectados y consultando
3. Verifique el rendimiento de la base de datos
4. Revise Prometheus en busca de cuellos de botella
5. Considere aumentar el tamaño/intervalo de lotes

Fallos de Enrutamiento:

1. Revise la configuración de enrutamiento
2. Pruebe en el simulador de enrutamiento
3. Verifique si faltan rutas
4. Verifique que exista una ruta de captura
5. Revise los registros de eventos para razones de fallo

Desconexiones de Frontend:

1. Verifique el estado del sistema del frontend
2. Verifique la conectividad de red
3. Revise los registros del frontend
4. Pruebe el registro manual de la API
5. Verifique las reglas del firewall

Procesamiento de Mensajes Lento:

1. Verifique el rendimiento de las consultas de la base de datos
2. Revise la configuración del trabajador por lotes
3. Verifique recursos adecuados (CPU/Memoria)
4. Verifique retrasos en las búsquedas de ENUM
5. Revise el rendimiento del sistema de carga

Para procedimientos de solución de problemas detallados, consulte la [Guía de Solución de Problemas](#).

Guía de Optimización del Rendimiento

[← Volver al Índice de Documentación](#) | [README Principal](#)

Esta guía explica cómo optimizar el rendimiento de SMS-C para diferentes escenarios de carga de trabajo.

Visión General del Rendimiento

SMS-C ofrece un rendimiento de **1,750 mensajes/segundo** utilizando Mnesia para el almacenamiento de mensajes en memoria con archivo automático de base de datos SQL para la retención de CDR.

Métricas Clave de Rendimiento

Medido en Intel i7-8650U @ 1.90GHz (8 núcleos):

Operación	Rendimiento	Latencia (promedio)	Mejora
Inserción de Mensaje (con enrutamiento)	1,750 msg/sec	0.58ms	21x más rápido que SQL
Inserción de Mensaje (simple)	1,750 msg/sec	0.57ms	21x más rápido que SQL
Obtener Mensajes para SMSC	800 msg/sec	1.25ms	Consulta en memoria
Memoria por Inserción	62 KB	-	Reducción del 50%

Capacidad: ~150 millones de mensajes por día en un solo nodo

Tabla de Contenidos

- [Arquitectura de Almacenamiento de Mensajes](#)
- [Optimización de Memoria](#)
- [Configuración de Archivo de CDR](#)
- [Optimización de Consultas](#)
- [Pruebas de Rendimiento](#)

Arquitectura de Almacenamiento de Mensajes

SMS-C utiliza una arquitectura de almacenamiento dual para un rendimiento óptimo:

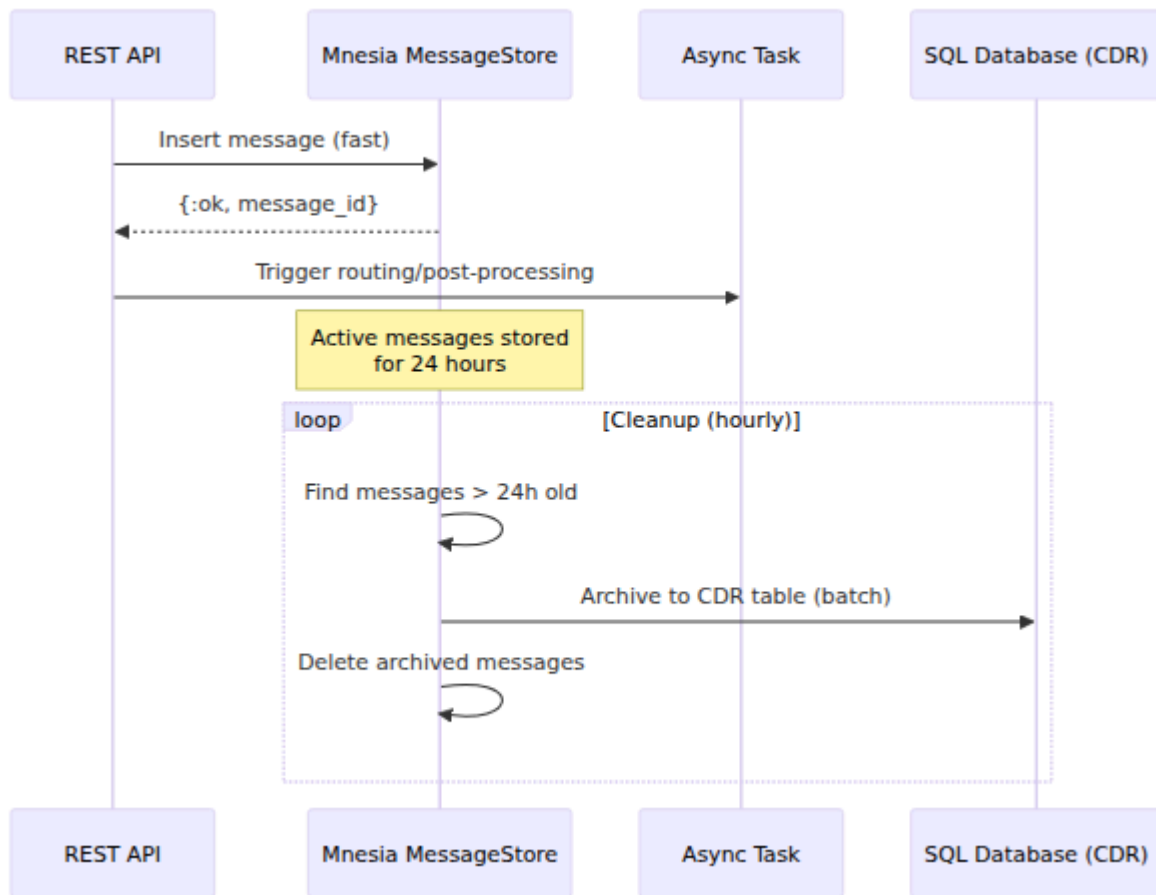
Almacén de Mensajes Activo (Mnesia)

- **Propósito:** Inserción, enrutamiento y entrega de mensajes ultra-rápidos
- **Almacenamiento:** En memoria con persistencia en disco (disc_copies)
- **Rendimiento:** Rendimiento de inserción de 1,750 msg/sec, latencia de 0.58ms
- **Retención:** Configurable (predeterminado: 24 horas)
- **Clustering:** Soporta Mnesia distribuido para escalado horizontal

Archivo de CDR (Base de Datos SQL)

- **Propósito:** Historial de mensajes a largo plazo e informes
- **Almacenamiento:** Base de datos SQL (MySQL/MariaDB o PostgreSQL) para archivo duradero
- **Rendimiento:** Escrituras por lotes para minimizar la carga de la base de datos
- **Retención:** Permanente (o según la política de retención de datos)
- **Consultas:** Análisis, informes, cumplimiento

Flujo de Datos



Optimización de Mnesia

Configuración de Retención de Mensajes

```
# config/runtime.exs
config :sms_c,
  message_retention_hours: 24 # Predeterminado: 24 horas
```

Directrices de Ajuste:

- **Alto volumen (>1M msg/día):** 12-24 horas de retención
 - Minimiza el tamaño de la tabla de Mnesia
 - Consultas más rápidas
 - Archivo más frecuente a MySQL

- **Volumen medio (100K-1M msg/día):** 24-48 horas de retención
 - Buen equilibrio para la mayoría de las implementaciones
 - Búfer adecuado para la lógica de reintento
- **Bajo volumen (<100K msg/día):** 48-168 horas de retención
 - Historial de mensajes más largo en almacenamiento rápido
 - Archivo menos frecuente

Índices de Tabla de Mnesia

MessageStore crea automáticamente índices en:

- `status` - Para filtrar mensajes pendientes/entregados
- `dest_smsc` - Para consultas específicas de SMSC
- `expires` - Para manejo de expiración
- `destination_msisdn` - Para consultas de suscriptores
- `source_msisdn` - Para consultas de suscriptores

Persistencia en Disco de Mnesia

Los mensajes se almacenan como `disc_copies` proporcionando:

- Rendimiento en memoria
- Persistencia automática en disco
- Recuperación ante fallos
- Sin pérdida de datos al reiniciar

Configuración de Archivo de CDR

El `BatchInsertWorker` maneja el archivo de CDR a MySQL utilizando escrituras por lotes:

```
# config/runtime.exs
config :sms_c,
  batch_insert_batch_size: 100,          # Tamaño del lote de CDR
  batch_insert_flush_interval_ms: 100  # Intervalo de auto-
flush
```

Directrices de Ajuste de CDR

Archivo de Alto Volumen

```
batch_insert_batch_size: 200
batch_insert_flush_interval_ms: 200
```

- Lotes más grandes reducen la carga de MySQL
- Mayor latencia para escrituras de CDR (aceptable para archivo)

Equilibrado (Recomendado)

```
batch_insert_batch_size: 100
batch_insert_flush_interval_ms: 100
```

- Buen equilibrio para la mayoría de las implementaciones
- CDR escritos dentro de 100ms

Requisitos de CDR en Tiempo Real

```
batch_insert_batch_size: 20
batch_insert_flush_interval_ms: 20
```

- Escrituras de CDR más rápidas para cumplimiento
- Más operaciones de escritura en MySQL

Optimización de Consultas

Uso Efectivo de los Índices de Mnesia

Las consultas que utilizan campos indexados son las más rápidas:

```
# Consultas rápidas (usar índices)
MessageStore.list(status: :pending)
MessageStore.list(dest_smsc: "gateway-1")
Messaging.get_messages_for_smsc("gateway-1")

# Consultas más lentas (escaneo de tabla completa)
MessageStore.list(limit: :infinity) # Devuelve todos los mensajes
```

Pool de Conexiones MySQL

Para consultas y archivo de CDR, configure el pool de conexiones MySQL:

```
# config/runtime.exs
config :sms_c, SmsC.Repo,
  pool_size: 10 # Aumentar para informes pesados de CDR
```

Directrices:

- Implementación estándar: `pool_size: 10`
- Informes pesados de CDR: `pool_size: 20-30`
- Solo archivo: `pool_size: 5`

Pruebas de Rendimiento

Ejecución de Pruebas

El proyecto incluye pruebas de rendimiento basadas en Benchee:

```
# Prueba de rendimiento de API SMS en crudo (compara sincrónico vs asíncrónico)
```

```
mix run benchmarks/raw_sms_bench.exs
```

```
# Prueba de rendimiento de API de mensajes general
```

```
mix run benchmarks/message_api_bench.exs
```

Interpretación de Resultados

Ejemplo de salida:

Name	ips	average
deviation	median	99th %
submit_message_raw_async (batch)	4.65 K	0.22 ms
±41.72%	0.184 ms	0.55 ms
submit_message_raw (sync)	0.0696 K	14.36 ms
±33.42%	12.57 ms	33.71 ms

Métricas clave:

- **ips**: Iteraciones por segundo (más alto es mejor)
- **average**: Tiempo de ejecución promedio (más bajo es mejor)
- **median**: Valor medio, más representativo que el promedio para distribuciones sesgadas
- **99th %**: Latencia del percentil 99 (importante para el cumplimiento del SLA)

Línea Base de Rendimiento

Rendimiento esperado en hardware moderno (Intel i7-8650U, 8 núcleos):

Métrica	insert_message (Mnesia)	Anterior (MySQL)
Rendimiento (con enrutamiento)	1,750 msg/sec	83 msg/sec
Rendimiento (simple)	1,750 msg/sec	89 msg/sec
Tiempo de Respuesta (promedio)	0.58ms	16ms
Tiempo de Respuesta (p99)	<5ms	30ms
Memoria por operación	62 KB	121 KB
Ganancia de Rendimiento	21x más rápido	-

Mejoras Clave:

- Eliminadas llamadas duplicadas de traducción de números
- Procesamiento asíncrono (enrutamiento, carga, eventos)
- Almacenamiento en memoria de Mnesia vs I/O en disco de MySQL
- Reducción del 50% en memoria

Monitoreo

Estadísticas en Tiempo de Ejecución

Verifique las estadísticas del trabajador por lotes:

```
SmsC.Messaging.BatchInsertWorker.stats()
```

Devuelve:

```
%{
  total_enqueued: 10000,
  total_flushed: 9900,
  total_batches: 99,
  current_queue_size: 100,
  flush_errors: 0,
  last_flush_at: ~U[2025-10-22 12:34:56Z],
  last_flush_count: 100,
  last_flush_duration_ms: 45
}
```

Métricas Clave a Monitorear

1. **Tamaño de Cola:** `current_queue_size` - Debe estar por debajo de `batch_size` la mayor parte del tiempo
2. **Duración de Flush:** `last_flush_duration_ms` - Debe ser $< 100\text{ms}$ para `batch_size=100`
3. **Errores de Flush:** `flush_errors` - Debe ser 0 o muy bajo
4. **Rendimiento:** `total_flushed / uptime` - Debe coincidir con la carga esperada

Alertas

Configure alertas de monitoreo para:

- Tamaño de cola consistentemente en máximo (indica presión de retroceso)
- Duración de flush en aumento (degradación del rendimiento de la base de datos)
- Errores de flush > 0 (problemas de conectividad de la base de datos)
- Rendimiento por debajo de lo esperado (degradación del rendimiento)

Solución de Problemas

Síntoma: Bajo Rendimiento

Causas posibles:

1. Pool de conexiones de base de datos agotado: Aumente `pool_size`
2. Base de datos lenta: Verifique el rendimiento de las consultas, agregue índices
3. Latencia de red: Optimice la ruta de red a la base de datos
4. Tamaño de lote demasiado pequeño: Aumente `batch_insert_batch_size`

Síntoma: Alta Latencia

Causas posibles:

1. Intervalo de flush demasiado alto: Reduzca `batch_insert_flush_interval_ms`
2. Tamaño de lote demasiado alto: Reduzca `batch_insert_batch_size`
3. Escrituras lentas en la base de datos: Verifique I/O en disco, optimice tablas
4. Uso de API asíncrona cuando necesita sincrónica: Cambie a endpoint sincrónico

Síntoma: Problemas de Memoria

Causas posibles:

1. Cola acumulándose: Mensajes acumulándose más rápido que el flush
2. Tamaño de lote demasiado grande: Reduzca `batch_insert_batch_size`
3. Fallos de flush: Verifique `flush_errors` en estadísticas
4. Necesidad de reiniciar el trabajador: `Supervisor.terminate_child/2` y reiniciar

Mejores Prácticas

1. **Comience con los valores predeterminados** (100/100ms) y ajuste según el comportamiento observado
2. **Monitoree en producción** durante al menos 1 semana antes de optimizar
3. **Pruebe cambios de configuración** en staging con carga similar a producción
4. **Utilice pruebas de rendimiento** para validar cambios de configuración
5. **Documente sus decisiones de ajuste** para referencia futura
6. **Configure alertas** antes de optimizar para detectar regresiones
7. **Considere las zonas horarias** - la carga máxima varía según la región

Ejemplos de Configuraciones

Configuración: Agregador de Alto Volumen

```
# config/prod.exs
config :sms_c,
  batch_insert_batch_size: 200,
  batch_insert_flush_interval_ms: 200

config :sms_c, SmsC.Repo,
  pool_size: 50
```

Configuración: Mensajería Empresarial en Tiempo Real

```
# config/prod.exs
config :sms_c,
  batch_insert_batch_size: 20,
  batch_insert_flush_interval_ms: 10

config :sms_c, SmsC.Repo,
  pool_size: 20
```

Configuración: Desarrollo/Pruebas

```
# config/dev.exs
config :sms_c,
  batch_insert_batch_size: 10,
  batch_insert_flush_interval_ms: 50

config :sms_c, SmsC.Repo,
  pool_size: 5
```

Lectura Adicional

- [Guía de Rendimiento de Ecto](#)
- [Documentación de Benchee](#)
- [Phoenix Bajo Presión](#)

Enrutamiento de SMS

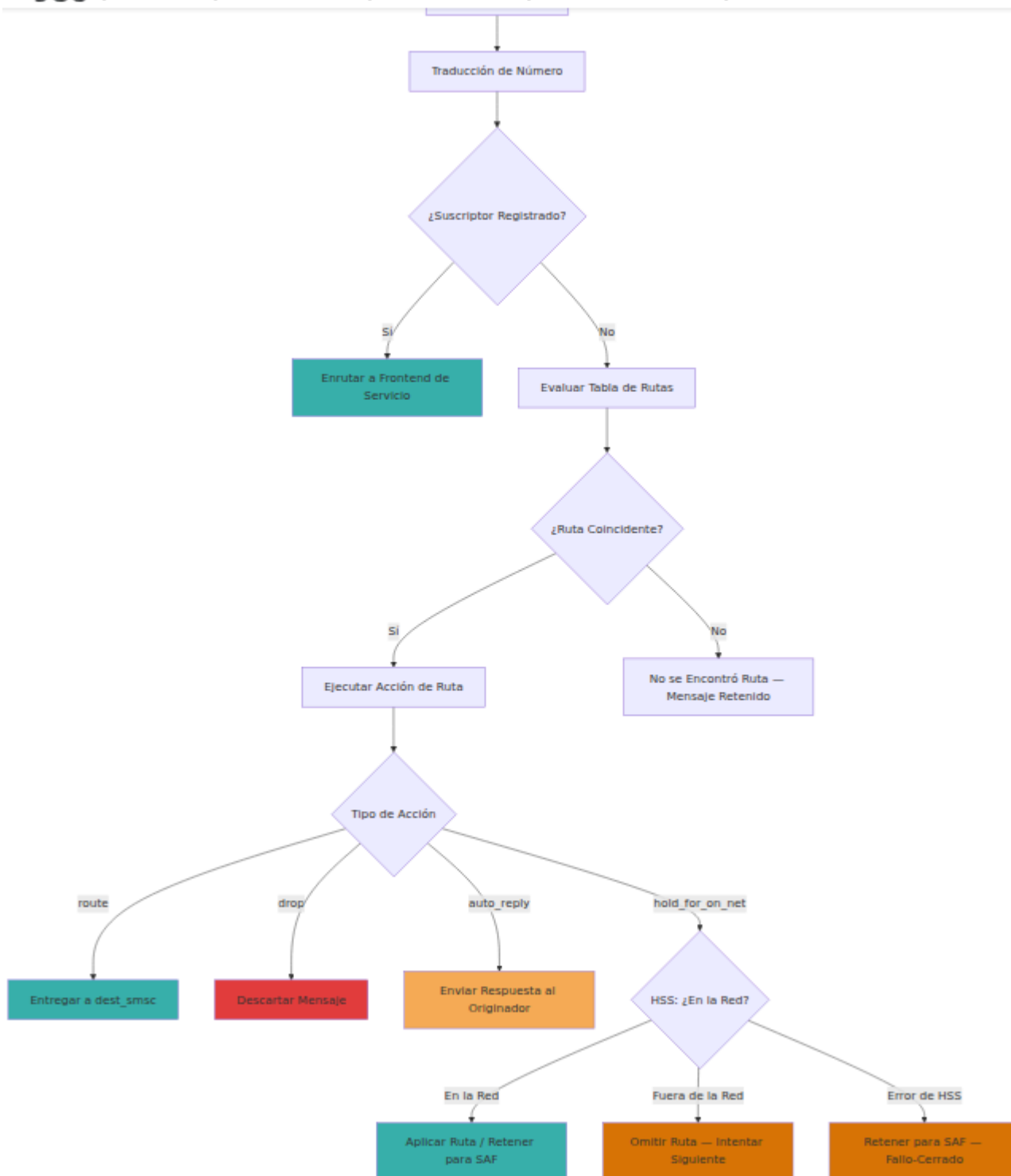
[← Volver al Índice de Documentación](#)

Descripción General

El motor de enrutamiento SMS-C determina a dónde va cada mensaje después de ingresar al sistema. Las rutas se almacenan en Mnesia, se pueden modificar en tiempo de ejecución a través de API o panel de control, y admiten coincidencias de regex, orden de prioridad, balanceo de carga ponderado y decisiones basadas en HSS sobre la red interna/externa.

Pipeline de Procesamiento de Mensajes

Cada mensaje sigue este pipeline desde la presentación hasta el enrutamiento:



Orden de Prioridad

1. **Enrutamiento basado en ubicación** — si el suscriptor de destino tiene un registro activo, enrutar directamente al frontend de servicio. Se omiten todas las reglas de ruta.

2. **Evaluación de tabla de rutas** — todas las rutas habilitadas se evalúan en función del mensaje. Cada ruta se registra con detalles de éxito/fallo por criterio de coincidencia.
3. **Ejecución de acción de ruta** — la acción de la ruta seleccionada determina qué sucede con el mensaje.

Modelo de Datos de Ruta

Campo	Tipo	Descripción
<code>route_id</code>	Entero	Identificador único generado automáticamente
<code>calling_regex</code>	Cadena	Regex para número de origen (<code>nil</code> = comodín)
<code>called_regex</code>	Cadena	Regex para número de destino (<code>nil</code> = comodín)
<code>source_smsc</code>	Cadena	Nombre del SMSC de origen (<code>nil</code> = comodín)
<code>dest_smsc</code>	Cadena	SMSC de destino. Requerido para la acción <code>route</code> , opcional para <code>hold_for_on_net</code>
<code>source_type</code>	Cadena	Filtrar por tipo de origen: <code>ims</code> , <code>circuit_switched</code> , <code>smpp</code> , o <code>nil</code>
<code>action</code>	Cadena	Acción de ruta: <code>route</code> , <code>drop</code> , <code>auto_reply</code> , <code>hold_for_on_net</code>
<code>auto_reply_message</code>	Cadena	Texto de respuesta (requerido cuando la acción es <code>auto_reply</code>)
<code>charged</code>	Cadena	Carga: <code>yes</code> , <code>no</code> , <code>default</code>
<code>weight</code>	Entero	Peso de balanceo de carga (1-100)
<code>priority</code>	Entero	Prioridad (1-255, menor = mayor prioridad)

Campo	Tipo	Descripción
<code>description</code>	Cadena	Etiqueta legible por humanos
<code>enabled</code>	Booleano	Alternar activo/inactivo
<code>originating_on_net_only</code>	Booleano	Descartar si el MSISDN de origen está fuera de la red según HSS
<code>terminating_on_net_only</code>	Booleano	Descartar si el MSISDN de destino está fuera de la red según HSS

Acciones de Ruta

`route` (Predeterminado)

Entrega el mensaje a `dest_smsc`. Esta es la acción de enrutamiento estándar.

```
{
  "action": "route",
  "called_regex": "^61",
  "dest_smsc": "australia-gw",
  "priority": 100,
  "description": "Enrutar números australianos a la puerta de
enlace AU"
}
```

`drop`

Descarta el mensaje. No se requiere `dest_smsc`.

```
{
  "action": "drop",
  "called_regex": "^900",
  "priority": 5,
  "description": "Bloquear números de tarifa premium"
}
```

auto_reply

Envía `auto_reply_message` de vuelta al originador. El mensaje original se marca como respondido automáticamente.

```
{
  "action": "auto_reply",
  "called_regex": "^911$",
  "auto_reply_message": "Emergencia: SMS a 911 no es compatible.
Por favor, marque 911.",
  "priority": 1,
  "description": "Respuesta automática 911"
}
```

hold_for_on_net

Verifica el MSISDN de destino contra el HSS a través de Diameter Sh. El comportamiento depende del resultado:

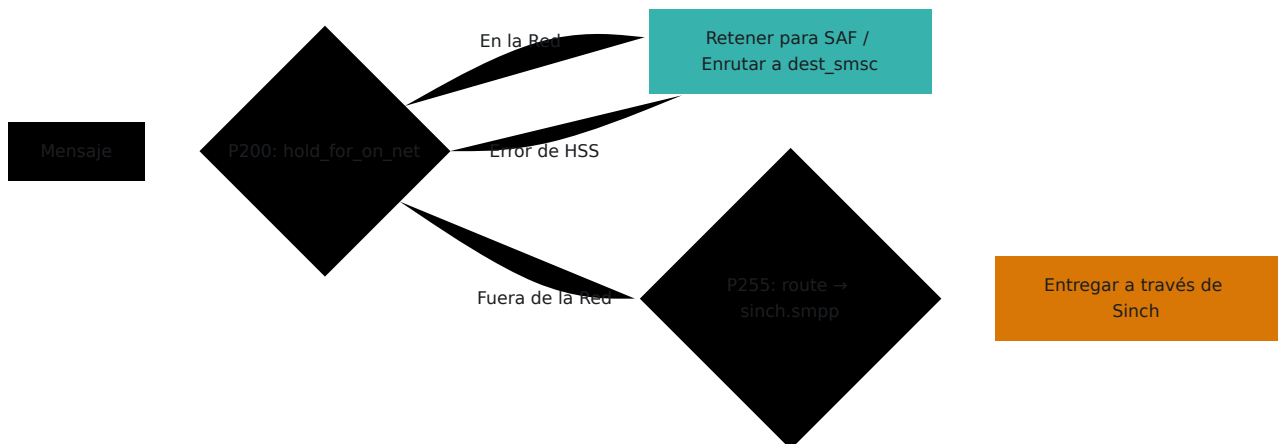
- **En la red:** Se aplica la ruta. Si `dest_smsc` está configurado, entregar allí. Si `dest_smsc` es `null`, retener el mensaje para entrega de almacenamiento y reenvío (SAF) cuando el suscriptor se registre.
- **Fuera de la red:** La ruta no se aplica. El motor de enrutamiento omite esta ruta y continúa evaluando la siguiente ruta coincidente.
- **Error de HSS o Diameter deshabilitado:** Fallo-cerrado. El mensaje se retiene para SAF.

```
{
  "action": "hold_for_on_net",
  "priority": 200,
  "description": "Retener suscriptores en la red para entrega local"
}
```

Con destino explícito:

```
{
  "action": "hold_for_on_net",
  "dest_smsc": "ims-frontend-01",
  "priority": 200,
  "description": "Enrutar en la red a frontend IMS"
}
```

Configuración Típica: Retención en la Red + Recaída Fuera de la Red



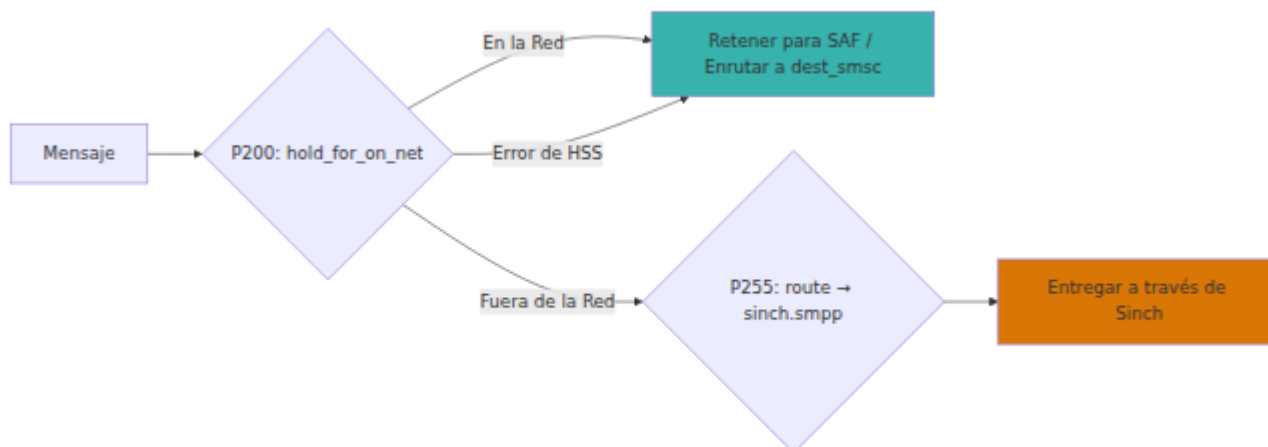
Esto asegura que los suscriptores en la red nunca se filtren a la puerta de enlace fuera de la red. Los números fuera de la red caen limpiamente.

Caché de Resultados de HSS

Las búsquedas de HSS son costosas (ida y vuelta de Diameter al HSS). El sistema almacena en caché los resultados por MSISDN para evitar búsquedas redundantes.

- **TTL de caché:** 2 horas (configurable a través de `hss_cache_ttl_seconds`)

- **Búsqueda perezosa:** HSS solo se consulta cuando una ruta realmente necesita el estado en la red (`hold_for_on_net`, `originating_on_net_only`, `terminating_on_net_only`)
- **Compartido entre rutas:** Si múltiples rutas necesitan el estado del mismo MSISDN, solo ocurre una búsqueda de HSS
- **Reenrutamiento forzado:** Omite la caché (búsqueda fresca) cuando un mensaje se reenvía explícitamente



Banderas de Aplicación en la Red

Estas son banderas booleanas en cualquier ruta que determinan si la ruta se aplica según el estado de HSS. Funcionan independientemente de la acción de la ruta.

`originating_on_net_only`

Si `true`, el MSISDN de origen se verifica contra el HSS. Si está fuera de la red, el mensaje se descarta. Utiliza la caché de HSS.

Caso de uso: restringir un enlace SMPP o un transportista externo para aceptar tráfico solo de originadores en la red.

`terminating_on_net_only`

Si `true`, el MSISDN de destino se verifica contra el HSS. Si está fuera de la red, el mensaje se descarta. Utiliza la caché de HSS.

Caso de uso: restringir una ruta para entregar solo a suscriptores en la red.

Comportamiento de Fallo-Cerrado

Todas las verificaciones en la red fallan-cerrado:

- HSS devuelve error → mensaje descartado
- Diameter deshabilitado → mensaje descartado
- HSS inalcanzable → mensaje descartado

Registro de Evaluación de Rutas

Cada ruta evaluada para un mensaje se registra como un evento

`route_evaluated` en ese mensaje. El registro incluye:

- ID de ruta, acción y prioridad
- Estado de éxito/fallo para cada criterio de coincidencia
- Detalle de lo que se comparó (patrón regex vs número, nombre de SMSC vs origen, etc.)

Ejemplo de entradas de registro de eventos:

```
Ruta #4 [hold_for_on_net] COINCIDIDA: P200 "Retener en la red" |
  calling_regex: ✓ (comodín), called_regex: ✓ (comodín),
  source_smsc: ✓ (comodín), source_type: ✓ (comodín),
  enum_result: ✓ (comodín)
```

```
Ruta #3 [route] FALLIDA: P255 "Recaída de Sinch" → sinch.smpp |
  calling_regex: ✓ (comodín), called_regex: ✓ (comodín),
  source_smsc: ✓ (comodín), source_type: x (ims vs
circuit_switched),
  enum_result: ✓ (comodín)
```

Estos eventos son visibles en el registro de actividad de mensajes del panel de control y a través de la API de Eventos.

Algoritmo de Evaluación de Rutas

Obtener Rutas
Habilitadas

Evaluar Cada Ruta

Para cada ruta

¿Coinciden Todos los
Criterios?

Sí

Agregar a la Lista
Coincidente

No

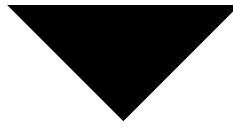
Omitir Ruta

Registro:
route_evaluated
COINCIDIDA

Registro:
route_evaluated FALLIDA

Sí

¿Más Rutas?



No

SELECT

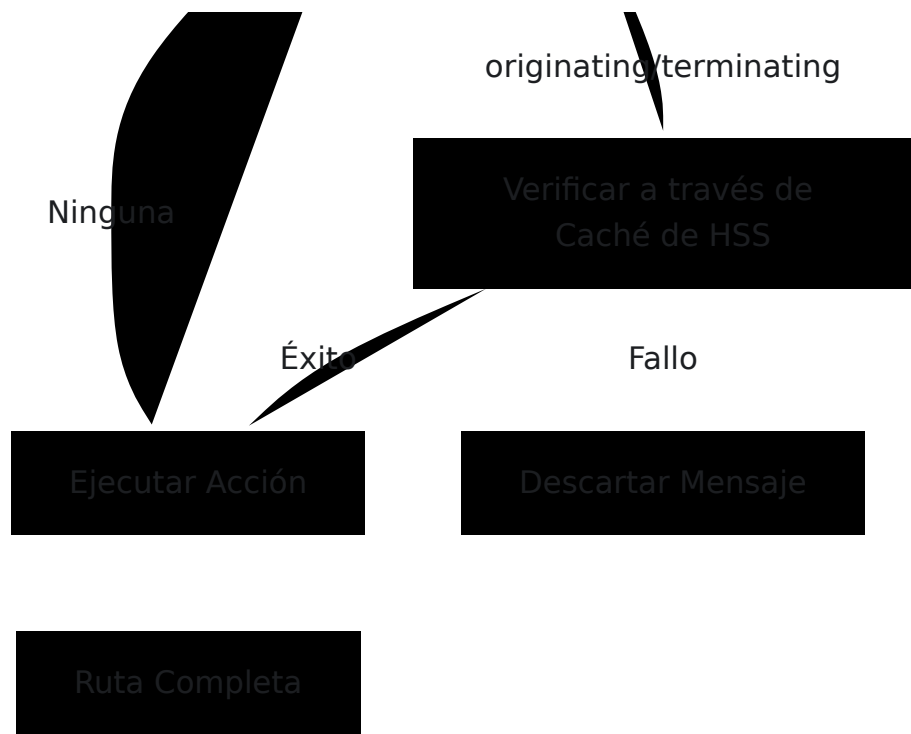
Ordenar por Prioridad y
luego Especificidad

Agrupar por Mayor
Prioridad

Selección Aleatoria
Ponderada

Ruta Seleccionada

¿Verificaciones en la
Red?



Puntuación de Especificidad

Cuando múltiples rutas coinciden, la especificidad determina cuál es preferida dentro del mismo nivel de prioridad:

Criterio	Puntos
Longitud del patrón <code>called_regex</code>	longitud × 100
Longitud del patrón <code>calling_regex</code>	longitud × 50
<code>source_smsc</code> especificado	+25
<code>enum_result_domain</code> especificado	+15
<code>source_type</code> especificado	+10
<code>enum_domain</code> especificado	+5

Patrones más largos y específicos siempre ganan sobre comodines.

Configuración

Cargando Rutas desde la Configuración

Las rutas definidas en `config/runtime.exs` bajo `:sms_routes` se cargan en el primer inicio cuando la tabla de enrutamiento está vacía. Reinicios posteriores preservan las rutas añadidas a través de la API.

```
config :sms_c, :sms_routes, [  
  %{  
    called_regex: ~r/^68987/,  
    dest_smsc: "pacific-gw",  
    priority: 100,  
    description: "Enrutamiento de Islas del Pacífico"  
  },  
  %{  
    action: :hold_for_on_net,  
    priority: 200,  
    description: "Retener en la red para SAF"  
  },  
  %{  
    action: :auto_reply,  
    called_regex: ~r/^911$/,  
    auto_reply_message: "Emergencia: Por favor, marque 911.",  
    priority: 1,  
    description: "Respuesta automática 911"  
  }  
]
```

Parámetros de Configuración de Ruta

Parámetro	Tipo	Requerido	Predeterminación
<code>action</code>	Átomo/Cadena	No	<code>:route</code>
<code>calling_regex</code>	Regex	No	<code>nil</code>
<code>called_regex</code>	Regex	No	<code>nil</code>
<code>source_smsc</code>	Cadena	No	<code>nil</code>
<code>dest_smsc</code>	Cadena	Condicional	<code>nil</code>
<code>source_type</code>	Átomo	No	<code>nil</code>

Parámetro	Tipo	Requerido	Predeterminación
auto_reply_message	Cadena	Condicional	nil
charged	Átomo	No	:default
weight	Entero	No	100
priority	Entero	No	100
description	Cadena	No	""
enabled	Booleano	No	true
originating_on_net_only	Booleano	No	false
terminating_on_net_only	Booleano	No	false

Configuración de Caché de HSS

Parámetro	Tipo	Predeterminado	Descripción
<code>hss_cache_ttl_seconds</code>	Entero	<code>7200</code>	Tiempo de vida para los resultados de búsqueda de HSS en caché (segundos)

API

Crear Ruta

POST /api/routes

```
{
  "action": "hold_for_on_net",
  "priority": 200,
  "description": "Retener suscriptores en la red para entrega local"
}
```

Listar Rutas

GET /api/routes

Actualizar Ruta

PUT /api/routes/:id

Eliminar Ruta

```
DELETE /api/routes/:id
```

Importar/Exportar

```
POST /api/routes/import  
GET /api/routes/export
```

La importación admite modos `merge` (agregar a los existentes) y `replace` (borrar y reemplazar todo).

Solución de Problemas

No se Encontró Ruta

Síntomas: El mensaje permanece pendiente con `dest_smsc: null`, el registro de eventos muestra `sms_routing_failed`

Causas posibles:

- No hay rutas habilitadas que coincidan con los criterios del mensaje
- Los patrones regex de ruta no coinciden con el formato del número (por ejemplo, falta el código de país)
- Todas las rutas coincidentes tienen `hold_for_on_net` y el destino está fuera de la red, pero no existe una ruta de respaldo

Resolución:

1. Verifique el registro de eventos del mensaje: cada ruta evaluada se registra con éxito/fallo por criterio
2. Utilice el simulador de enrutamiento en el panel de control para probar con los mismos números
3. Asegúrese de que exista una ruta de respaldo que cubra todo en un número de prioridad alto (por ejemplo, 255)

Suscriptor en la Red Enrutado Fuera de la Red

Síntomas: Los mensajes a suscriptores en la red se entregan a través de una puerta de enlace externa en lugar de ser retenidos para entrega local

Causas posibles:

- No se configuró ninguna ruta `hold_for_on_net`
- La ruta `hold_for_on_net` tiene un número de prioridad más alto que la ruta de respaldo fuera de la red
- Fallo en la búsqueda de HSS (verifique eventos `hss_dip_error`)
- Diameter no habilitado en la configuración

Resolución:

1. Agregue una ruta `hold_for_on_net` con un número de prioridad más bajo que la ruta de respaldo fuera de la red
2. Verifique que Diameter esté habilitado y que los pares de HSS estén conectados a través del panel de control
3. Verifique el estado de la caché de HSS y los resultados recientes de búsqueda en el registro de eventos

Caché de HSS Obsoleta

Síntomas: Suscriptor portado o provisionado pero aún enrutando según el antiguo estado en la red/fuera de la red

Resolución:

1. Reenrute el mensaje a través de la API: el reenrutamiento forzado omite la caché de HSS
2. Espere a que expire el TTL de la caché (predeterminado 2 horas)
3. Reduzca `hss_cache_ttl_seconds` si se necesitan actualizaciones más rápidas

Guía de Solución de Problemas de SMS-C

[← Volver al Índice de Documentación](#) | [README Principal](#)

Guía completa para diagnosticar y resolver problemas comunes de SMS-C.

Tabla de Contenidos

- [Herramientas de Diagnóstico](#)
- [Problemas de Entrega de Mensajes](#)
- [Problemas de Enrutamiento](#)
- [Problemas de Rendimiento](#)
- [Problemas de Base de Datos](#)
- [Problemas de Conexión del Frontend](#)
- [Problemas de Carga/Facturación](#)
- [Problemas de Búsqueda ENUM](#)
- [Problemas de Clúster](#)
- [Problemas de API](#)
- [Problemas de Interfaz Web](#)
- [Problemas de Recursos del Sistema](#)

Herramientas de Diagnóstico

Verificación Rápida de Salud

```
# 1. Verificar estado de la API
curl https://api.example.com:8443/api/status

# 2. Verificar endpoint de métricas de Prometheus
curl https://api.example.com:9568/metrics | grep sms_c

# 3. Verificar registros de la aplicación
tail -f /var/log/sms_c/application.log

# 4. Verificar estado del proceso
systemctl status sms_c

# 5. Verificar conectividad a la base de datos SQL CDR
(MySQL/MariaDB)
mysql -u sms_user -p -h db.example.com -e "SELECT 1"

# Para PostgreSQL:
# psql -U sms_user -h db.example.com -d sms_c_prod -c "SELECT 1"
```

Análisis de Registros

Ver Errores Recientes:

```
# Últimas 100 entradas de registro de nivel de error
tail -1000 /var/log/sms_c/application.log | grep "\[error\]"

# Buscar patrones de error específicos
grep "routing_failed" /var/log/sms_c/application.log

# Encontrar errores de base de datos SQL
grep -i "database\|sql\|ecto" /var/log/sms_c/application.log |
grep error
```

Monitorear Registros en Tiempo Real:

```
# Seguir registros con filtro
tail -f /var/log/sms_c/application.log | grep -E "
(error|warning|critical)"
```

Consultas de Métricas

Verificar Tasa de Procesamiento de Mensajes:

```
# Mensajes por segundo
rate(sms_c_message_received_count[5m])

# Tasa de éxito de entrega
rate(sms_c_delivery_succeeded_count[5m]) /
rate(sms_c_delivery_queued_count[5m])
```

Verificar Estado de la Cola:

```
# Profundidad actual de la cola
sms_c_queue_size_pending

# Edad del mensaje más antiguo (segundos)
sms_c_queue_oldest_message_age_seconds
```

Verificar Rendimiento del Sistema:

```
# Latencia de procesamiento de mensajes (p95)
histogram_quantile(0.95,
sms_c_message_processing_stop_duration_bucket)

# Latencia de enrutamiento (p95)
histogram_quantile(0.95, sms_c_routing_stop_duration_bucket)
```

Problemas de Entrega de Mensajes

Mensajes No Entregados

Síntomas:

- Mensajes atascados en estado "pendiente"
- Alto conteo de mensajes pendientes
- Sin notificaciones de entrega

Pasos de Diagnóstico:

1. Verificar Conexiones del Frontend:

```
curl https://api.example.com:8443/api/frontends/active
```

Esperado: Lista de frontends activos

Problema: Lista vacía o falta de frontends esperados

2. Verificar Cola de Mensajes:

Acceder a la Interfaz Web: `/message_queue`

- Filtrar por estado: "pendiente"
- Verificar valor de `dest_smsc`
- Verificar que `deliver_after` no esté en el futuro

3. Verificar Enrutamiento:

Acceder a la Interfaz Web: `/simulator`

- Probar con parámetros de mensaje reales
- Verificar que la ruta coincida y el destino sea correcto

4. Verificar Polling del Frontend:

Revisar registros del sistema del frontend:

- ¿Está el frontend consultando `/api/messages?`

- ¿Está el frontend enviando correctamente el encabezado `smcsc`?

Soluciones:

Sin Frontends Conectados:

```
# Verificar estado del sistema del frontend
systemctl status frontend_service

# Verificar que el frontend pueda alcanzar la API
curl -k https://api.example.com:8443/api/status

# Registrar manualmente el frontend
curl -X POST https://api.example.com:8443/api/frontends/register \
  -H "Content-Type: application/json" \
  -d '{
    "frontend_name": "test_gateway",
    "frontend_type": "smpp",
    "ip_address": "10.0.1.50"
  }'
```

Mensajes Enrutados al SMSC Incorrecto:

- Revisar la configuración de enrutamiento
- Verificar prioridades de ruta
- Probar en el simulador de enrutamiento
- Verificar que el nombre del frontend coincida con `dest_smcsc` en los mensajes

Mensajes Programados para el Futuro:

- Verificar la marca de tiempo `deliver_after`
- Reiniciar si es necesario:

```
curl -X PATCH https://api.example.com:8443/api/messages/12345 \
  -H "Content-Type: application/json" \
  -d '{"deliver_after": "2025-10-30T12:00:00Z"}'
```

Mensajes Fallando con Reintentos

Síntomas:

- Contador de `delivery_attempts` aumentando
- Mensajes con alto conteo de intentos (> 3)
- Retrasos de retroceso exponencial

Pasos de Diagnóstico:

1. Verificar Registro de Eventos:

```
curl https://api.example.com:8443/api/events/12345
```

Buscar:

- Eventos de fallo de entrega
- Descripciones de errores
- Marcas de tiempo de reintento

2. Verificar Registros del Frontend:

- ¿Por qué el frontend no puede entregar?
- ¿Errores de red?
- ¿Errores de protocolo?
- ¿Sistema de downstream no disponible?

Soluciones:

Problemas de Red Temporales:

- Esperar reintento (automático)
- Monitorear para entrega exitosa

Fallos Persistentes:

```
# Enrutamiento a puerta de enlace alternativa
curl -X PATCH https://api.example.com:8443/api/messages/12345 \
  -H "Content-Type: application/json" \
  -d '{"dest_smsc": "backup_gateway"}'

# Reiniciar contador de reintentos
curl -X PATCH https://api.example.com:8443/api/messages/12345 \
  -H "Content-Type: application/json" \
  -d '{"delivery_attempts": 0, "deliver_after": "2025-10-30T12:00:00Z"}'
```

Número de Destino Inválido:

- Verificar formato del número
- Revisar reglas de traducción de números
- Eliminar mensaje si es realmente inválido

Mensajes de Carta Muerta

Síntomas:

- `deadletter: true` en el mensaje
- Mensajes pasados el tiempo de expiración
- Estado aún "pendiente"

Pasos de Diagnóstico:

1. Encontrar Mensajes de Carta Muerta:

Acceder a la Interfaz Web: `/message_queue`

- Filtrar por estado expirado
- Verificar marcas de tiempo de expiración

2. Verificar Por Qué Expiró:

- Revisar registro de eventos
- Verificar historial de intentos de entrega
- Verificar que el enrutamiento fue exitoso

Soluciones:

Extender Expiración:

```
# Agregar 24 horas a la expiración
curl -X PATCH https://api.example.com:8443/api/messages/12345 \
  -H "Content-Type: application/json" \
  -d '{"expires": "2025-10-31T12:00:00Z", "deadletter": false}'
```

Problemas de Enrutamiento

No Se Encontró Ruta

Síntomas:

- Error: `no_route_found`
- Métrica `sms_c_routing_failed_count` aumentando
- Registro de eventos muestra "routing_failed"

Pasos de Diagnóstico:

1. Verificar Que Existan Rutas:

Acceder a la Interfaz Web: `/sms_routing`

- Verificar que las rutas estén configuradas
- Comprobar que al menos una ruta esté habilitada

2. Probar Enrutamiento:

Acceder a la Interfaz Web: `/simulator`

- Ingresar parámetros de mensaje (número de llamada, número llamado, SMSC de origen)
- Revisar resultados de evaluación
- Verificar por qué las rutas no coincidieron

3. Verificar Criterios de Ruta:

- ¿Se requieren coincidencias de prefijo?
- ¿Filtro de SMSC de origen demasiado restrictivo?
- ¿Todas las rutas deshabilitadas?

Soluciones:

Sin Rutas Configuradas:

Agregar ruta de captura:

```
Prefijo de Llamada: (vacío)
Prefijo Llamado: (vacío)
SMSC de Origen: (vacío)
SMSC de Destino: default_gateway
Prioridad: 255
Peso: 100
Habilitado: ✓
Descripción: Ruta predeterminada de captura
```

Rutas Demasiado Específicas:

Agregar ruta más amplia:

```
Prefijo Llamado: +
SMSC de Destino: international_gateway
Prioridad: 200
Peso: 100
Habilitado: ✓
Descripción: Captura internacional
```

Todas las Rutas Deshabilitadas:

- Habilitar rutas apropiadas a través de la Interfaz Web
- Verificar que la configuración no haya deshabilitado accidentalmente rutas

Ruta Incorrecta Seleccionada

Síntomas:

- Mensajes enrutados a un destino inesperado
- Puerta de enlace incorrecta recibiendo tráfico
- Balanceo de carga no distribuyendo como se esperaba

Pasos de Diagnóstico:

1. Usar Simulador de Enrutamiento:

Acceder a la Interfaz Web: `/simulator`

- Probar con parámetros de mensaje reales
- Revisar sección "Todas las Coincidencias"
- Verificar prioridades y puntuaciones de especificidad

2. Verificar Prioridades de Ruta:

- Número más bajo = mayor prioridad
- Rutas evaluadas en orden de prioridad
- Dentro de la misma prioridad, se aplican pesos

3. Verificar Especificidad de Ruta:

Puntuación de especificidad:

- Prefijo llamado más largo: +100 puntos por carácter
- Prefijo de llamada más largo: +50 puntos por carácter
- SMSC de origen especificado: +25 puntos
- Tipo de origen especificado: +10 puntos
- Dominio ENUM especificado: +15 puntos

Soluciones:

Ajustar Prioridades:

Hacer que la ruta específica tenga mayor prioridad:

Ruta Premium:

Prefijo Llamado: +1555

Prioridad: 10 (alta prioridad)

Ruta General:

Prefijo Llamado: +1

Prioridad: 50 (baja prioridad)

Ajustar Pesos:

Cambiar distribución de balanceo de carga:

Primario (70%):

Peso: 70

Respaldo (30%):

Peso: 30

Agregar Ruta Más Específica:

Anular ruta general para caso específico:

Ruta Específica:

Prefijo Llamado: +15551234

SMSC de Destino: dedicated_gateway

Prioridad: 1

Ruta General:

Prefijo Llamado: +1

SMSC de Destino: general_gateway

Prioridad: 50

Respuesta Automática No Funciona

Síntomas:

- Ruta de respuesta automática configurada pero no se activa
- No se envían mensajes de respuesta

- Registro de eventos sin evento de respuesta automática

Pasos de Diagnóstico:

1. Verificar Configuración de Ruta:

- `auto_reply: true`
- `auto_reply_message` contiene texto
- Ruta está habilitada
- Ruta coincide con criterios de mensaje

2. Probar en Simulador:

- Verificar que la ruta sea seleccionada
- Comprobar indicación de "auto_reply"

3. Verificar Registro de Eventos:

```
curl https://api.example.com:8443/api/events/12345 | grep auto_reply
```

Soluciones:

Ruta No Coincidente:

- Ampliar criterios (eliminar filtros)
- Verificar prioridad (debería ser más alta que las rutas normales)
- Verificar estado habilitado

Mensaje No Establecido:

Editar ruta, agregar mensaje:

```
Respuesta Automática: ✓  
Mensaje de Respuesta Automática: "Gracias por su mensaje.  
Responderemos pronto."
```

Prioridad Incorrecta:

Las rutas de respuesta automática deben tener alta prioridad (número bajo):

Ruta de Respuesta Automática:

Prioridad: 10

Ruta Normal:

Prioridad: 50

Problemas de Rendimiento

Alta Latencia en el Procesamiento de Mensajes

Síntomas:

- `sms_c_message_processing_stop_duration` p95 > 1000ms
- Respuestas lentas de la API
- Cola acumulándose

Pasos de Diagnóstico:

1. Verificar Latencias de Componentes:

```
# Latencia de enrutamiento
histogram_quantile(0.95, sms_c_routing_stop_duration_bucket)

# Latencia de búsqueda ENUM
histogram_quantile(0.95, sms_c_enum_lookup_stop_duration_bucket)

# Latencia de carga
histogram_quantile(0.95, sms_c_charging_succeeded_duration_bucket)

# Latencia de entrega
histogram_quantile(0.95, sms_c_delivery_succeeded_duration_bucket)
```

2. Verificar Recursos del Sistema:

```
# Uso de CPU
top -b -n 1 | grep sms_c

# Uso de memoria
ps aux | grep beam.smp
```

Soluciones:

Enrutamiento Lento (Muchas rutas):

- Reducir el número de rutas habilitadas
- Combinar rutas similares
- Optimizar criterios de ruta

Búsquedas ENUM Lentas:

- Verificar latencia del servidor DNS
- Aumentar tiempo de espera
- Usar servidores DNS más rápidos/cercanos
- Deshabilitar ENUM si no es necesario

Carga Lenta:

- Verificar rendimiento del OCS
- Aumentar tiempo de espera del OCS
- Deshabilitar carga si no es necesario
- Usar carga asíncrona

Base de Datos Lenta:

- Aumentar tamaño del grupo de conexiones
- Agregar índices
- Optimizar consultas
- Mejorar recursos de la base de datos

Cambios de Configuración:

```
# config/config.exs
# Aumentar tamaño del lote para rendimiento
config :sms_c,
  batch_insert_batch_size: 200,
  batch_insert_flush_interval_ms: 200

# Aumentar grupo de base de datos
config :sms_c, SmsC.Repo,
  pool_size: 50
```

Bajo Rendimiento de Mensajes

Síntomas:

- Procesamiento < 100 msg/sec
- Usando API asíncrona pero aún lenta
- Altos tiempos de respuesta de la API

Pasos de Diagnóstico:

1. Verificar Trabajador de Lotes:

```
# En consola de producción (iex)
SmsC.Messaging.BatchInsertWorker.stats()
```

Buscar:

- `current_queue_size` cerca del máximo
- `flush_errors` > 0
- `last_flush_duration_ms` muy alto

2. Verificar Cuellos de Botella:

```
# Tiempo de consulta de base de datos  
ecto_pools_query_time
```

```
# Tiempo de cola del grupo de conexiones  
ecto_pools_queue_time
```

Soluciones:

Cuello de Botella de Base de Datos:

Aumentar tamaño del grupo:

```
config :sms_c, SmsC.Repo,  
  pool_size: 50 # Aumentar de 20
```

Configuración de Lotes:

Ajustar para rendimiento:

```
config :sms_c,  
  batch_insert_batch_size: 200, # Lotes más grandes  
  batch_insert_flush_interval_ms: 200 # Intervalo más largo
```

Usar Endpoint Asíncrono:

```
# Alto rendimiento: usar /create_async  
curl -X POST  
https://api.example.com:8443/api/messages/create_async  
  
# NO: /api/messages (síncrono)
```

Crecimiento de Respaldo en la Cola

Síntomas:

- `ecto_pools_queue_size_pending` aumentando
- Edad del mensaje más antiguo aumentando

- Procesamiento no puede mantenerse al ritmo de la tasa de entrada

Pasos de Diagnóstico:

1. Verificar Tasa de Entrada vs Tasa de Entrega:

```
# Tasa de entrada
rate(sms_c_message_received_count[5m])

# Tasa de entrega
rate(sms_c_delivery_succeeded_count[5m])
```

2. Verificar Capacidad del Frontend:

- ¿Están los frontends consultando con suficiente frecuencia?
- ¿Están los frontends procesando mensajes lo suficientemente rápido?
- ¿Hay errores en el frontend?

3. Verificar Tasa de Éxito de Entrega:

```
rate(sms_c_delivery_succeeded_count[5m]) /
rate(sms_c_delivery_attempted_count[5m])
```

Soluciones:

Frontends No Consultando:

- Verificar conectividad del frontend
- Verificar intervalo de consulta (debería ser de 5 a 10 segundos)
- Reiniciar servicios del frontend

Frontends Demasiado Lentos:

- Agregar más instancias de frontend
- Optimizar procesamiento del frontend
- Aumentar concurrencia del frontend

Alta Tasa de Reintentos:

- Investigar fallos de entrega
- Corregir problemas de downstream
- Enrutar a puertas de enlace alternativas

Pico Temporal:

- Esperar a que la cola se drene
- Monitorear hasta que vuelva a la normalidad
- Considerar mejoras de capacidad si es recurrente

Problemas de Base de Datos

Fallos de Conexión

Síntomas:

- Error: "unable to connect to database"
- API devolviendo errores 500
- La aplicación no se inicia

Pasos de Diagnóstico:

1. Verificar Estado de la Base de Datos SQL CDR:

```
# MySQL/MariaDB
systemctl status mysql

# PostgreSQL
systemctl status postgresql

# Probar conectividad (MySQL/MariaDB)
mysql -u sms_user -p -h db.example.com -e "SELECT 1"

# Probar conectividad (PostgreSQL)
psql -U sms_user -h db.example.com -d sms_c_prod -c "SELECT 1"
```

2. Verificar Red:

```
# Hacer ping al host de la base de datos
ping db.example.com

# Verificar conectividad de puerto (MySQL/MariaDB: 3306,
PostgreSQL: 5432)
telnet db.example.com 3306
# o
telnet db.example.com 5432
```

3. Verificar Credenciales:

```
# Verificar variables de entorno
echo $DB_USERNAME
echo $DB_HOSTNAME
echo $DB_PORT

# Intentar conexión manual con las mismas credenciales
(MySQL/MariaDB)
mysql -u $DB_USERNAME -p$DB_PASSWORD -h $DB_HOSTNAME

# Para PostgreSQL:
# psql -U $DB_USERNAME -h $DB_HOSTNAME -d sms_c_prod
```

Soluciones:

Base de Datos Caída:

```
# Iniciar base de datos (MySQL/MariaDB)
systemctl start mysql

# Iniciar base de datos (PostgreSQL)
systemctl start postgresql
```

Credenciales Incorrectas:

Actualizar configuración:

```
export DB_USERNAME=correct_user
export DB_PASSWORD=correct_password

# Reiniciar aplicación
systemctl restart sms_c
```

Problema de Red:

- Verificar reglas de firewall
- Verificar grupos de seguridad (nube)
- Verificar conectividad VPN/red

Grupo de Conexiones Agotado:

Aumentar tamaño del grupo:

```
config :sms_c, SmsC.Repo,
  pool_size: 50 # Aumentar desde el valor actual
```

Consultas Lentas

Síntomas:

- Tiempo de consulta de base de datos alto
- Respuestas lentas de la API
- Cola del grupo de conexiones acumulándose

Pasos de Diagnóstico:

1. Verificar Registro de Consultas Lentas:

```
-- MySQL/MariaDB: Habilitar registro de consultas lentas
SET GLOBAL slow_query_log = 'ON';
SET GLOBAL long_query_time = 1; -- Registrar consultas > 1
segundo

-- Ver consultas lentas (MySQL/MariaDB)
SELECT * FROM mysql.slow_log ORDER BY query_time DESC LIMIT 10;

-- PostgreSQL: Habilitar registro de consultas lentas en
postgresql.conf
-- log_min_duration_statement = 1000 # milisegundos
-- Luego verificar registros de PostgreSQL
```

2. Verificar Índices Faltantes:

```
-- Verificar índices de tabla
SHOW INDEX FROM message_queues;

-- Índices esperados:
-- - source_smsc
-- - dest_smsc
-- - send_time
-- - inserted_at
```

3. Verificar Estadísticas de Tabla:

```

-- Tamaños de tabla (MySQL/MariaDB)
SELECT
  table_name,
  table_rows,
  ROUND(data_length / 1024 / 1024, 2) AS data_mb,
  ROUND(index_length / 1024 / 1024, 2) AS index_mb
FROM information_schema.tables
WHERE table_schema = 'sms_c_prod';

-- Tamaños de tabla (PostgreSQL)
-- SELECT schemaname, tablename,
--
pg_size_pretty(pg_total_relation_size(schemaname||'.'||tablename))
AS size
-- FROM pg_tables WHERE schemaname = 'public';

```

Soluciones:

Índices Faltantes:

```

CREATE INDEX idx_message_queues_source_smsc ON
message_queues(source_smsc);
CREATE INDEX idx_message_queues_dest_smsc ON
message_queues(dest_smsc);
CREATE INDEX idx_message_queues_send_time ON
message_queues(send_time);
CREATE INDEX idx_message_queues_status ON message_queues(status);

```

Fragmentación de Tabla:

```

-- MySQL/MariaDB
OPTIMIZE TABLE message_queues;
OPTIMIZE TABLE frontend_registrations;

-- PostgreSQL
-- VACUUM ANALYZE message_queues;
-- VACUUM ANALYZE frontend_registrations;

```

Demasiados Datos:

Limpiar registros antiguos:

```
-- Eliminar mensajes entregados de más de 30 días
DELETE FROM message_queues
WHERE status = 'delivered'
AND deliver_time < DATE_SUB(NOW(), INTERVAL 30 DAY)
LIMIT 10000;
```

Espacio en Disco Lleno

Síntomas:

- Error: "Disco lleno"
- No se puede escribir en la base de datos
- La aplicación se bloquea

Pasos de Diagnóstico:

1. Verificar Uso de Disco:

```
df -h

# Verificar directorio de base de datos SQL (MySQL/MariaDB)
du -sh /var/lib/mysql

# Verificar directorio de base de datos SQL (PostgreSQL)
du -sh /var/lib/postgresql
```

2. Encontrar Archivos Grandes:

```
# Encontrar archivos más grandes (MySQL/MariaDB)
find /var/lib/mysql -type f -exec du -h {} + | sort -rh
| head -20

# Encontrar archivos más grandes (PostgreSQL)
find /var/lib/postgresql -type f -exec du -h {} + | sort
-rh | head -20

# Verificar archivos de registro
du -sh /var/log/sms_c/*
```

Soluciones:

Limpiar Datos Antiguos:

```
-- Eliminar mensajes antiguos
DELETE FROM message_queues
WHERE inserted_at < DATE_SUB(NOW(), INTERVAL 90 DAY)
LIMIT 100000;
```

Rotar Registros:

```
# Forzar logrotate
logrotate -f /etc/logrotate.d/sms_c

# Limpiar archivos de registro antiguos
find /var/log/sms_c -name "*.log.*" -mtime +30 -delete
```

Expandir Disco:

- Redimensionar volumen (nube)
- Agregar nuevo disco y extender volumen
- Mover datos a disco más grande

Problemas de Conexión del Frontend

Frontend No Aparece Como Activo

Síntomas:

- Estado del frontend muestra "expirado"
- Frontend no en la lista activa
- Mensajes no se entregan al frontend

Pasos de Diagnóstico:

1. Verificar Registro:

```
curl https://api.example.com:8443/api/frontends/active | grep frontend_name
```

2. Verificar Registros del Frontend:

- ¿Está el frontend llamando a `/api/frontends/register?`
- ¿Hay errores en la API?
- Frecuencia de registro (debería ser cada 60s)

3. Verificar Registros de API:

```
grep "frontend.*register" /var/log/sms_c/application.log | tail -20
```

Soluciones:

Frontend No Registrándose:

Probar registro manual:

```
curl -X POST https://api.example.com:8443/api/frontends/register \
-H "Content-Type: application/json" \
-d '#123;
  "frontend_name": "uk_gateway",
  "frontend_type": "smpp",
  "ip_address": "10.0.1.50"
&#125;'
```

Si tiene éxito, el problema está en el código/configuración del frontend.

Tiempo de Registro Agotado:

Los frontends expiran después de 90 segundos. Asegurarse de registrar cada 60 segundos:

```
# El frontend debe llamar a registrar cada 60 segundos
while True:
    register_with_smpp()
    time.sleep(60)
```

Problemas de Red:

- Verificar firewall entre frontend y API
- Verificar resolución DNS
- Probar con curl desde el servidor del frontend

Frontend Conectándose/Desconectándose Repetidamente

Síntomas:

- Estado del frontend cambiando entre activo/expirado
- Alto conteo de registro en el historial
- Conexión inestable

Pasos de Diagnóstico:

1. Verificar Salud del Frontend:

- ¿Es estable el proceso del frontend?
- ¿Hay caídas o reinicios?
- ¿Problemas de recursos (CPU/memoria)?

2. Verificar Estabilidad de la Red:

```
# Verificar pérdida de paquetes
ping -c 100 api.example.com

# Verificar reinicios de conexión
netstat -s | grep -i reset
```

3. Verificar Tiempo de Registro:

- ¿Demasiado frecuente? (cada pocos segundos)
- ¿Demasiado infrecuente? (> 90 segundos)

Soluciones:

Frontend Inestable:

- Corregir problemas de la aplicación del frontend
- Aumentar recursos del frontend
- Revisar registros del frontend en busca de errores

Problemas de Red:

- Verificar conectividad intermitente
- Revisar registros del firewall
- Verificar comprobaciones de salud del balanceador de carga

Intervalo de Registro Incorrecto:

Corregir intervalo:

```
REGISTRATION_INTERVAL = 60 # segundos
```

Problemas de Carga/Facturación

Fallos de Carga

Síntomas:

- `sms_c_charging_failed_count` aumentando
- Registro de eventos muestra "charging_failed"
- Mensajes marcados como `charge_failed: true`

Pasos de Diagnóstico:

1. Verificar Conectividad con OCS:

```
# Probar API de OCS
curl -X POST http://ocs.example.com:2080/jsonrpc \
  -H "Content-Type: application/json" \
  -d '&#123;
    "method": "SessionSv1.Ping",
    "params": [],
    "id": 1
  &#125;'
```

Esperado: `{"result": "Pong"}`

2. Verificar Registros de OCS:

```
tail -f /var/log/ocs/ocs.log
```

3. Verificar Configuración:

```
# Verificar URL de OCS
grep ocs_url config/runtime.exs
```

Soluciones:

OCS No Disponible:

```
# Verificar estado de OCS
systemctl status ocs

# Iniciar si es necesario
systemctl start ocs
```

Error de Configuración:

Actualizar configuración:

```
config :sms_c,
  ocs_url: "http://correct-host:2080/jsonrpc",
  ocs_tenant: "correct_tenant"
```

Deshabilitar Carga Temporalmente:

```
config :sms_c,
  default_charging_enabled: false
```

Reiniciar aplicación.

Problemas de Cuenta:

- Verificar que la cuenta exista en OCS
- Verificar que la cuenta tenga saldo
- Revisar que los planes de tarifas estén configurados

Carga Demasiado Lenta

Síntomas:

- `sms_c_charging_succeeded_duration` p95 > 500ms
- Procesamiento de mensajes lento cuando la carga está habilitada
- Rápido cuando la carga está deshabilitada

Pasos de Diagnóstico:

1. Verificar Latencia de Carga:

```
histogram_quantile(0.95, sms_c_charging_succeeded_duration_bucket)
```

2. Verificar Rendimiento de OCS:

```
# Tiempo de respuesta de OCS
curl -w "%#123;time_total%#125;\n" -X POST
http://ocs.example.com:2080/jsonrpc \
  -H "Content-Type: application/json" \
  -d '%#123;"method":"SessionSv1.Ping","params":[],"id":1%#125;'
```

3. Verificar Latencia de Red:

```
# Hacer ping al host de OCS
ping -c 10 ocs.example.com
```

Soluciones:

OCS Lento:

- Optimizar configuración de OCS
- Agregar recursos a OCS
- Usar motor de tarifas más rápido

Latencia de Red:

- Desplegar OCS más cerca de SMS-C
- Usar ruta de red directa
- Evitar VPN/túneles si es posible

Tiempo de Espera Demasiado Bajo:

Aumentar tiempo de espera:

```
config :sms_c,
  ocs_timeout: 5000 # 5 segundos
```

Problemas de Búsqueda ENUM

Fallos en Búsquedas ENUM

Síntomas:

- `sms_c_enum_lookup_stop_duration` mostrando fallos
- Registro de eventos muestra errores ENUM
- Rutas con `enum_result_domain` no coincidiendo

Pasos de Diagnóstico:

1. Verificar Configuración ENUM:

```
grep -A 10 "enum_" config/runtime.exs
```

2. Probar Conectividad DNS:

```
# Probar servidor DNS
dig @8.8.8.8 e164.arpa

# Probar consulta ENUM
# Para +15551234567:
dig @8.8.8.8 NAPTR 7.6.5.4.3.2.1.5.5.5.1.e164.arpa
```

3. Verificar Servidor DNS:

```
# ¿Es accesible el DNS personalizado?
ping 10.0.1.53

# Probar puerto
nc -zv 10.0.1.53 53
```

Soluciones:

Servidor DNS Inalcanzable:

Usar DNS alternativo:

```
config :sms_c,  
  enum_dns_servers: [  
    &#123;"8.8.8.8", 53&#125;, # DNS Público de Google  
    &#123;"1.1.1.1", 53&#125; # DNS de Cloudflare  
  ]
```

Dominio ENUM Incorrecto:

Actualizar dominio:

```
config :sms_c,  
  enum_domains: ["e164.arpa"] # Usar dominio estándar
```

Tiempo de Espera Demasiado Corto:

Aumentar tiempo de espera:

```
config :sms_c,  
  enum_timeout: 10000 # 10 segundos
```

Deshabilitar ENUM (si no es necesario):

```
config :sms_c,  
  enum_enabled: false
```

Problemas de Caché ENUM

Síntomas:

- Tasa de aciertos de caché baja (< 70%)
- Tamaño de caché creciendo sin límites
- Uso de memoria alto

Pasos de Diagnóstico:

1. Verificar Estadísticas de Caché:

```
# Tasa de aciertos de caché
rate(sms_c_enum_cache_hit_count[5m]) /
(rate(sms_c_enum_cache_hit_count[5m]) +
rate(sms_c_enum_cache_miss_count[5m]))

# Tamaño de caché
sms_c_enum_cache_size_size
```

2. Verificar Patrón de Tráfico:

- ¿Se están repitiendo números?
- ¿TTL de caché apropiado?

Soluciones:

Baja Tasa de Aciertos (Esperado):

- Tráfico a números únicos (normal)
- Monitorear pero no alarmar si < 70%

Caché Creciendo:

Limpiar caché a través de la página de prueba NAPTR o reiniciar aplicación.

Uso de Memoria Alto:

- Esperado con caché grande
- Monitorear uso de memoria del sistema en general
- Considerar ajuste de TTL

Problemas de Clúster

Nodo No Puede Unirse al Clúster

Síntomas:

- Nodo único en ejecución
- Consultas de clúster devolviendo solo resultados locales
- Errores de distribución de Erlang

Pasos de Diagnóstico:

1. Verificar Nombres de Nodo:

```
# En consola IEx
Node.self()
# Esperado: :sms@node1.example.com

Node.list()
# Esperado: Lista de otros nodos
```

2. Verificar Cookie de Erlang:

```
# Verificar archivo de cookie
cat ~/.erlang.cookie

# Verificar que sea el mismo en todos los nodos
```

3. Verificar Red:

```
# ¿Pueden los nodos alcanzarse entre sí?
ping node2.example.com

# Verificar puertos
nc -zv node2.example.com 4369
nc -zv node2.example.com 9100-9200
```

Soluciones:

Desajuste de Cookie:

Establecer la misma cookie en todos los nodos:

```
export ERLANG_COOKIE=same_secret_value_here

# 0 actualizar ~/.erlang.cookie
echo "same_secret_value_here" > ~/.erlang.cookie
chmod 400 ~/.erlang.cookie
```

Firewall Bloqueando:

Abrir puertos requeridos:

```
# EPMD
iptables -A INPUT -p tcp --dport 4369 -j ACCEPT

# Distribución de Erlang
iptables -A INPUT -p tcp --dport 9100:9200 -j ACCEPT
```

Problemas de DNS:

Usar direcciones IP en lugar de nombres de host:

```
config :sms_c,
  cluster_nodes: [
    : "sms@10.0.1.10",
    : "sms@10.0.1.11"
  ]
```

División de Clúster

Síntomas:

- Nodos en ejecución pero desconectados
- Datos diferentes en diferentes nodos
- Inconsistencias en Mnesia

Pasos de Diagnóstico:

1. Verificar Conectividad de Nodos:

```
# En cada nodo (IEx)  
Node.list()
```

2. Verificar Mnesia:

```
:mnesia.system_info(:running_db_nodes)
```

Soluciones:

Reconectar Nodos:

```
# Detener todos los nodos  
systemctl stop sms_c  
  
# Iniciar un nodo primero  
systemctl start sms_c # En nodo1  
  
# Esperar a que se inicie completamente, luego iniciar otros  
systemctl start sms_c # En nodo2  
systemctl start sms_c # En nodo3
```

Inconsistencia en Mnesia:

- Exportar rutas desde el nodo correcto
- Detener todos los nodos
- Eliminar directorio de Mnesia
- Iniciar nodos
- Importar rutas

Problemas de API

API No Responde

Síntomas:

- Tiempo de conexión agotado

- Conexión rechazada
- Sin respuesta

Pasos de Diagnóstico:

1. Verificar Proceso de API:

```
# ¿Está la aplicación en ejecución?  
systemctl status sms_c  
  
# Verificar puertos en escucha  
netstat -tlnp | grep 8443
```

2. Verificar Firewall:

```
# Verificar iptables  
iptables -L -n | grep 8443  
  
# Probar conectividad local  
curl -k https://localhost:8443/api/status
```

3. Verificar Configuración TLS:

```
# Verificar que el certificado exista  
ls -l priv/cert/server.crt priv/cert/server.key  
  
# Verificar validez del certificado  
openssl x509 -in priv/cert/server.crt -noout -dates
```

Soluciones:

Aplicación No Ejecutándose:

```
systemctl start sms_c
```

Firewall Bloqueando:

```
# Permitir puerto de API
iptables -A INPUT -p tcp --dport 8443 -j ACCEPT
```

Problemas de Certificado:

Generar nuevo certificado (ver Guía de Configuración).

Puerto Incorrecto:

Verificar configuración:

```
grep "port:" config/runtime.exs
```

API Devolviendo Errores 500

Síntomas:

- Error Interno del Servidor
- Código de estado 500
- Error en registros

Pasos de Diagnóstico:

1. Verificar Registros de Aplicación:

```
tail -100 /var/log/sms_c/application.log | grep "\[error\]"
```

2. Verificar Base de Datos:

```
mysql -u sms_user -p -e "SELECT 1"
```

3. Verificar Recursos:

```
# Memoria
free -h

# CPU
top -b -n 1

# Disco
df -h
```

Soluciones:

Base de Datos No Disponible:

- Iniciar base de datos
- Corregir problema de conexión

Sin Memoria:

- Reiniciar aplicación
- Aumentar memoria del sistema
- Verificar fugas de memoria

Error de Aplicación:

- Verificar error específico en registros
- Corregir problema de configuración
- Reiniciar aplicación

Problemas de Interfaz Web

No Se Puede Acceder a la Interfaz Web

Síntomas:

- Tiempo de conexión agotado
- 404 No Encontrado
- La página no carga

Pasos de Diagnóstico:

1. Verificar Estado de la Aplicación:

```
systemctl status sms_c
```

2. Verificar Puerto:

```
netstat -tlnp | grep 80
```

3. Verificar URL:

- ¿Nombre de host correcto?
- ¿Puerto correcto?
- ¿HTTP vs HTTPS?

Soluciones:

Puerto Incorrecto:

Verificar configuración:

```
grep "control_panel" config/runtime.exs
```

Acceder en el puerto correcto (predeterminado: 80 o 4000).

Aplicación No Ejecutándose:

```
systemctl start sms_c
```

Firewall:

```
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

LiveView No Actualiza

Síntomas:

- La página se carga pero no se actualiza
- Los datos están obsoletos
- Errores de WebSocket en la consola del navegador

Pasos de Diagnóstico:

1. Verificar Consola del Navegador:

- Abrir Herramientas de Desarrollo (F12)
- Buscar errores de WebSocket
- Verificar pestaña de red para solicitudes fallidas

2. Verificar Configuración del Proxy:

Si se usa un proxy inverso, asegurarse de que soporte WebSocket:

```
location /live {  
    proxy_http_version 1.1;  
    proxy_set_header Upgrade $http_upgrade;  
    proxy_set_header Connection "upgrade";  
};
```

Soluciones:

WebSocket Bloqueado:

- Configurar proxy para WebSocket
- Verificar firewall
- Verificar extensiones del navegador

Actualizar Página:

- Actualización forzada (Ctrl+F5)
- Limpiar caché del navegador

Problemas de Recursos del Sistema

Alto Uso de CPU

Síntomas:

- CPU consistentemente > 80%
- Sistema lento
- Aplicación no responde

Pasos de Diagnóstico:

1. Verificar Proceso:

```
top -b -n 1 | grep beam.smp
```

2. Verificar Métricas:

```
# Tasa de procesamiento de mensajes  
rate(sms_c_message_received_count[5m])  
  
# Operaciones de enrutamiento  
rate(sms_c_routing_route_matched_count[5m])
```

Soluciones:

Tráfico Alto:

- Escalar horizontalmente (agregar nodos)
- Escalar verticalmente (agregar CPU)

Enrutamiento Ineficiente:

- Reducir número de rutas
- Optimizar criterios de ruta

Demasiadas Búsquedas ENUM:

- Verificar tasa de aciertos de caché
- Considerar deshabilitar si no es necesario

Alto Uso de Memoria

Síntomas:

- Uso de memoria > 90%
- La aplicación se bloquea
- Errores de falta de memoria

Pasos de Diagnóstico:

1. Verificar Memoria:

```
free -h  
  
ps aux | grep beam.smp
```

2. Verificar Tamaños de Caché:

```
sms_c_enum_cache_size_size
```

Soluciones:

Caché ENUM Demasiado Grande:

- Limpiar caché
- Reducir TTL
- Deshabilitar ENUM si no es necesario

Cola de Lotes Creciendo:

```
# Verificar estadísticas del trabajador (IEx)  
SmsC.Messaging.BatchInsertWorker.stats()
```

Si la cola es grande, vaciar manualmente o reiniciar.

Agregar Memoria:

- Escalar verticalmente
- Agregar swap (temporal)

Fuga de Memoria:

- Reiniciar aplicación
- Reportar problema para investigación

Para asistencia adicional, consulte:

- [Guía de Operaciones](#) - Procedimientos diarios
- [Guía de Configuración](#) - Opciones de configuración
- [Guía de Métricas](#) - Configuración de monitoreo
- Registros de la aplicación - `/var/log/sms_c/application.log`

