



SMSc - Aplicación del Centro de Servicios SMS

Aplicación del Centro SMS construida con Phoenix/Elixir, que proporciona una cola de mensajes centralizada y una API REST para el enrutamiento y la entrega de SMS.

Documentación

Documentación Principal

- [Índice Completo de Documentación](#) - Comienza aquí para toda la documentación
- [Referencia de Configuración](#) - Opciones de configuración completas
- [Referencia de API](#) - Documentación de la API REST
- [Guía de Operaciones](#) - Operaciones diarias y monitoreo
- [Guía de Enrutamiento de SMS](#) - Gestión y configuración de rutas
- [Guía de Traducción de Números](#) - Normalización y reescritura de números
- [Ajuste de Rendimiento](#) - Optimización para diferentes cargas de trabajo
- [Guía de Métricas](#) - Métricas de Prometheus y monitoreo
- [Guía de Solución de Problemas](#) - Problemas comunes y soluciones
- [Esquema CDR](#) - Formato de registro de detalles de llamadas

Documentación de Cumplimiento

- [Cumplimiento de Intercepción ANSSI R226](#) - Especificaciones técnicas de intercepción legal en Francia

Rendimiento

- [Referencias de Rendimiento](#) - Pruebas de rendimiento

Descripción General de la Arquitectura

El núcleo de SMS_C proporciona una cola de mensajes independiente del protocolo y una API REST. Los frontends SMSC externos (SMPP, IMS, SS7/MAP) se conectan como puertas de enlace independientes que se comunican con el núcleo a través de la API REST.

Flujo de Mensajes

Flujo de Mensajes Salientes (MT - Móvil Terminado)

Flujo de Mensajes Entrantes (MO - Móvil Originado)

Características Clave

1. Diseño Independiente del Protocolo

- El núcleo SMS_C maneja la persistencia de mensajes, el enrutamiento y la API
- Los frontends externos (SMPP, IMS, SS7/MAP) manejan la comunicación específica del protocolo
- Todos los frontends se comunican a través de una API REST unificada
- Agregar nuevos protocolos sin cambiar el núcleo
- Cada uno puede escalarse de forma independiente

2. Enrutamiento de Mensajes

- **Motor de enrutamiento dinámico** con configuración en tiempo de ejecución
- Enrutamiento basado en prefijos (números llamados/llamantes)
- Filtrado de SMSC y tipo de fuente (IMS/Circuito Conmutado/SMPP)
- Enrutamiento basado en prioridad con balanceo de carga basado en peso
- Capacidades de enrutamiento de respuesta automática y eliminación de mensajes
- Control de cargos por ruta
- Interfaz web para gestión de rutas
- Actualizaciones de ruta en tiempo real sin interrupción del servicio

❖ Consulta la [Guía de Enrutamiento de SMS](#) para documentación completa

3. Lógica de Reintento con Retroceso Exponencial

- Reintento automático en caso de fallo de entrega
- Retroceso exponencial: 1min, 2min, 4min, 8min, etc.
- Intentos de reintento máximos configurables
- Manejo de expiración de mensajes
- Seguimiento de reintentos por mensaje

Guía de Operaciones

Puntos de Acceso:

- API REST: <https://localhost:8443> (o <http://localhost:8080> sin TLS)
- Panel de Control: <https://localhost:8086>
- Documentación de API (Swagger UI): <https://localhost:8443/api/docs>

Iniciar Frontends Externos: Cada frontend de protocolo es una aplicación independiente. Consulta la documentación de cada frontend para instrucciones de inicio.

Configuración

Toda la configuración se gestiona directamente en `config/runtime.exs`. No se utilizan variables de entorno.

Configuración del Núcleo

Actualmente no se utilizan variables de entorno para la configuración de la aplicación núcleo. Los puertos del servidor y los nombres de host se configuran en `config/runtime.exs`:

- Servidor API: Puerto 8443 (HTTPS), escuchando en 0.0.0.0
- Panel de Control: Puerto 80 (HTTP), escuchando en 0.0.0.0

Configuración de la Base de Datos

Los ajustes de la base de datos se configuran en `config/runtime.exs`:

- Nombre de usuario: `omnitouch`
- Contraseña: `omnitouch2024`
- Nombre del host: `localhost`
- Puerto: `3306`
- Nombre de la base de datos: `smc_new`
- Tamaño del grupo: `1`

Configuración del Clúster

Los ajustes del clúster se configuran en `config/runtime.exs`:

- Nodos del clúster: `""` (vacío - sin agrupamiento por defecto)
- Consulta de clúster DNS: `nil`

Configuración de la Cola de Mensajes

Los ajustes de la cola de mensajes se configuran en `config/runtime.exs`:

- Tiempo de carta muerta: `1440` minutos (24 horas antes de que el mensaje expire)

Integración de Cargos

Los ajustes de cargos se configuran en `config/runtime.exs`:

- URL: `http://localhost:2080/jsonrpc`
- Inquilino: `mnc057.mcc505.3gppnetwork.org`
- Destino: `55512341234`
- Fuente: `00101900000257`
- Asunto: `00101900000257`
- Cuenta: `00101900000257`

Configuración de Enrutamiento de SMS

El sistema de enrutamiento de SMS utiliza rutas dinámicas respaldadas por bases de datos que pueden ser gestionadas a través de la interfaz web o el archivo de configuración. Las rutas se cargan desde `config/runtime.exs` en el primer inicio.

Ejemplo de Configuración:

```
config :sms_c, :sms_routes, [  
  %{  
    called_prefix: "+44",  
    dest_smsc: "InternationalGW",  
    weight: 100,  
    priority: 100,  
    description: "SMS Internacional del Reino Unido",  
    enabled: true  
  },  
  %{  
    called_prefix: "1900",  
    dest_smsc: "PremiumGW",  
    charged: :yes,  
    priority: 50,  
    description: "Números Premium de EE. UU.",  
    enabled: true  
  }  
]
```

Características:

- Coincidencia basada en prefijos (números llamados/llamantes)
- Filtrado de SMSC de origen y tipo
- Enrutamiento basado en prioridad y peso
- Capacidades de respuesta automática y eliminación
- Control de cargos por ruta
- Gestión en tiempo de ejecución a través de la interfaz web en `/routing`

❖ Consulta la [Guía de Enrutamiento de SMS](#) para documentación

completa, ejemplos y referencia de API.

Puntos Finales de la API REST

Operaciones de Cola de Mensajes

Enviar SMS (Crear Mensaje)

POST /api/messages

Content-Type: application/json

```
{
  "source_msisdn": "+1234567890",
  "destination_msisdn": "+0987654321",
  "message_body": "¡Hola, Mundo!",
  "source_smsc": "web-app",
  "dest_smsc": "smpp-provider",          # Opcional - el motor de
enrutamiento asigna si es nulo
  "tp_dcs_character_set": "gsm7",        # Opcional: gsm7, 8bit,
latin1, ucs2
  "tp_dcs_coding_group": "general_data_coding",
  "expires": "2025-10-17T10:30:00Z"     # Opcional - por defecto 24h
desde ahora
}
```

Campos Requeridos:

- destination_msisdn - Número de teléfono de destino
- message_body - Contenido del texto del mensaje
- source_msisdn - Número de teléfono de origen
- source_smsc - Identificador del sistema de origen

Campos Opcionales:

- dest_smsc - Puerta de enlace de destino (el motor de enrutamiento asigna si no se proporciona)
- source_imsi, dest_imsi - Identificadores IMSI
- tp_dcs_character_set - Codificación de caracteres (gsm7, 8bit, latin1, ucs2)
- tp_dcs_coding_group - Grupo de codificación DCS
- tp_dcs_compressed - Indicador de compresión (booleano)
- tp_dcs_has_message_class - Indicador de clase de mensaje (booleano)
- tp_dcs_message_class - Valor de clase de mensaje
- tp_user_data_header - Encabezado de datos de usuario (mapa)
- message_part_number, message_parts - Campos de mensaje multipart
- expires - Marca de tiempo de expiración (por defecto 24 horas)
- deliver_after - Marca de tiempo de entrega retrasada

- deadletter, raw_data_flag, raw_sip_flag - Indicadores booleanos

Respuesta:

```
{
  "status": "success",
  "data": {
    "id": 123,
    "source_msisdn": "+1234567890",
    "destination_msisdn": "+0987654321",
    "dest_smsc": "smpp-provider",
    "message_body": "¡Hola, Mundo!",
    "deliver_time": null,
    "delivery_attempts": 0,
    "expires": "2025-10-17T10:30:00Z",
    "inserted_at": "2025-10-16T10:30:00Z"
  }
}
```

Obtener Mensajes para SMSC

```
GET /api/messages/get_by_smsc?smsc=mi-nombre-smsc
```

Devuelve todos los mensajes no entregados donde:

- destination_smsc es nulo O coincide con el nombre SMSC proporcionado
- El mensaje no ha expirado
- Listo para enviar (deliver_after es nulo o está en el pasado)

Respuesta:

```
{
  "status": "success",
  "data": [
    {
      "id": 123,
      "source_msisdn": "+1234567890",
      "destination_msisdn": "+0987654321",
      "message_body": "Hola",
      "destination_smsc": "mi-nombre-smsc",
      "delivery_attempts": 0
    }
  ]
}
```

Listar Mensajes con Filtrado Opcional de SMSC

```
# Listar todos los mensajes en la cola
```

```
GET /api/messages
```

```
# Listar mensajes para SMSC específico (con filtrado de encabezados)  
GET /api/messages  
SMSc: mi-nombre-smsc
```

Sin Encabezado SMSc: Devuelve todos los mensajes en la cola independientemente del estado de entrega o expiración.

Con Encabezado SMSc: Devuelve mensajes no entregados donde:

- `dest_smsc` coincide con el valor del encabezado O `dest_smsc` es nulo
- `deliver_time` es nulo (no entregado aún)
- `deliver_after` es nulo o antes/igual al tiempo actual (listo para entregar)
- `expires` es después del tiempo actual (no ha expirado)
- Ordenado por tiempo de inserción (el más antiguo primero)

Nota: El enfoque del encabezado SMSc permite que los frontends externos consulten sus mensajes utilizando el mismo patrón de punto final, con el encabezado controlando el comportamiento de filtrado.

Respuesta:

```
[  
  {  
    "id": 123,  
    "source_msisdn": "+1234567890",  
    "destination_msisdn": "+0987654321",  
    "message_body": "¡Hola, Mundo!",  
    "dest_smsc": "mi-nombre-smsc",  
    "deliver_time": null,  
    "delivery_attempts": 0,  
    "expires": "2025-10-17T10:30:00Z",  
    "inserted_at": "2025-10-16T10:30:00Z"  
  }  
]
```

Obtener Mensaje Único

```
GET /api/messages/{id}
```

Actualizar Mensaje

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

```
{  
  "status": "delivered",
```

```
"delivered_at": "2025-10-16T10:30:00Z"
}
```

Eliminar SMS

```
DELETE /api/messages/{id}
```

Manejar Fallo de Entrega (Incrementar Contador de Reintentos)

Cuando un mensaje falla temporalmente en la entrega, incrementa el contador de intentos de entrega y programa un reintento con retroceso exponencial.

Método 1: Usando PUT (Recomendado)

```
# Simple y semántico - PUT indica actualizar el estado de entrega
PUT /api/messages/{id}
```

Método 2: Usando Punto Final Explícito

```
# Punto final alternativo explícito
POST /api/messages/{id}/increment_delivery_attempt
```

Ambos métodos incrementan `delivery_attempts` y establecen un retraso de retroceso exponencial a través de `deliver_after`:

Intento Fórmula de Retroceso Retraso Tiempo Total

1º	2^1 minutos	2 min	2 min
2º	2^2 minutos	4 min	6 min
3º	2^3 minutos	8 min	14 min
4º	2^4 minutos	16 min	30 min
5º	2^5 minutos	32 min	1h 2min
6º	2^6 minutos	64 min	2h 6min

Respuesta:

```
{
  "id": 123,
  "delivery_attempts": 1,
  "deliver_after": "2025-10-20T19:05:00Z",
  "deliver_time": null,
  "expires": "2025-10-21T19:03:00Z",
  ...
}
```

Nota: Los mensajes con `deliver_after` futuro se filtran automáticamente de las solicitudes GET hasta que expire el período de retroceso.

Actualizar Mensaje (Actualización Parcial)

Para actualizar campos específicos del mensaje (comportamiento inalterado):

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

{
  "dest_smsc": "puerta-actualizada",
  "status": "delivered"
}
```

Importante: PUT y PATCH se comportan de manera diferente:

- **PUT** → Incrementa los intentos de entrega con retroceso (no se requiere cuerpo)
- **PATCH** → Realiza actualizaciones parciales de campos (se requiere cuerpo)

Seguimiento de Salud del Frontend

El núcleo SMS_C rastrea la salud y disponibilidad de los frontends externos a través de un sistema de registro. Esto permite monitorear el tiempo de actividad del frontend, detectar fallos y mantener datos históricos de disponibilidad.

Nota: El registro del frontend NO se utiliza para la entrega o enrutamiento de mensajes. Los mensajes se enrutan en función del campo `dest_smsc`. El sistema de registro existe únicamente para el monitoreo de salud y visibilidad operativa.

Cómo Funciona el Registro del Frontend

Cada frontend externo (puerta de enlace SMPP, IMS, SS7/MAP) envía periódicamente un registro de latido al núcleo SMS_C:

1. **Intervalo de Latido:** Los frontends deben registrarse cada 30-60 segundos
2. **Tiempo de Expiración:** Los registros expiran después de 90 segundos sin una actualización
3. **Gestión Automática del Estado:**
 - Nuevos frontends crean un nuevo registro de registro
 - Frontends activos existentes actualizan su registro (extiende la expiración)
 - Frontends expirados que vuelven a estar en línea crean un nuevo período de registro

Puntos Finales de Registro del Frontend

Registrar/Actualizar Frontend (Latido)

POST /api/frontends

Content-Type: application/json

```
{
  "frontend_name": "puerta-sMPP-1",
  "frontend_type": "SMPP",
  "ip_address": "10.0.1.5",
  "hostname": "smpp-gw-01",
  "uptime_seconds": 3600,
  "configuration": "{\"port\": 2775, \"system_id\": \"usuario_SMPP\"}"
}
```

Campos Requeridos:

- frontend_name - Identificador único para la instancia del frontend
- frontend_type - Tipo de frontend (SMPP, IMS, MAP, etc.)

Campos Opcionales:

- ip_address - Dirección IP del frontend (detectada automáticamente desde la fuente de la solicitud si no se proporciona)
- hostname - Nombre de host del servidor frontend
- uptime_seconds - Segundos desde que se inició el frontend
- configuration - Cadena JSON con la configuración específica del frontend

Nota: Si no se proporciona ip_address, el núcleo SMS_C utilizará automáticamente la IP de origen de la solicitud HTTP. Esto funciona tanto con conexiones directas como con solicitudes proxy (a través del encabezado X-Forwarded-For).

Respuesta:

```
{
  "id": 42,
  "frontend_name": "puerta-sMPP-1",
  "frontend_type": "SMPP",
  "ip_address": "10.0.1.5",
  "hostname": "smpp-gw-01",
  "uptime_seconds": 3600,
  "status": "activo",
  "last_seen_at": "2025-10-20T10:30:00Z",
  "expires_at": "2025-10-20T10:31:30Z",
  "inserted_at": "2025-10-20T10:00:00Z"
}
```

```
}
```

Listar Todos los Registros de Frontend

```
GET /api/frontends
```

Devuelve todos los registros de frontend (activos y expirados), ordenados por la actividad más reciente.

Listar Solo Frontends Activos

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

Devuelve solo los frontends actualmente activos (no expirados).

Obtener Estadísticas del Frontend

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

Devuelve estadísticas resumidas:

```
{
  "active": 5,
  "expired": 12,
  "unique_frontends": 8
}
```

Obtener Historial del Frontend

```
GET /api/frontends/history/{frontend_name}
```

Devuelve todos los registros históricos para un frontend específico, útil para analizar patrones de tiempo de actividad/inactividad.

Ejemplo:

```
GET /api/frontends/history/puerta-sMPP-1
```

Obtener Registro Específico

```
GET /api/frontends/{id}
```

Implementación en Frontends Externos

Los frontends externos deben implementar una tarea en segundo plano que envíe latidos:

Ejemplo (pseudocódigo):

```
import time
import requests

def send_heartbeat():
    """Enviar latido cada 30 segundos"""
    while True:
        try:
            data = {
                "frontend_name": "mi-puerta-sMPP",
                "frontend_type": "SMPP",
                "ip_address": get_local_ip(),
                "hostname": get_hostname(),
                "uptime_seconds": get_uptime()
            }

            response = requests.post(
                "https://smc-core:8443/api/frontends",
                json=data,
                timeout=5
            )

            if response.status_code in [200, 201]:
                logger.debug("Latido enviado con éxito")
            else:
                logger.error(f"Fallo en el latido: {response.status_code}")

        except Exception as e:
            logger.error(f"Error en el latido: {e}")

        time.sleep(30) # Enviar cada 30 segundos

# Iniciar latido en hilo en segundo plano
threading.Thread(target=send_heartbeat, daemon=True).start()
```

Monitoreo de Salud del Frontend

Panel de Control - La interfaz web en <https://localhost:8086> muestra:

- Frontends actualmente activos
- Marca de tiempo de la última vez que se vio cada frontend
- Seguimiento del tiempo de actividad
- Disponibilidad histórica

Consultas API:

```
# Obtener todos los frontends activos
curl https://localhost:8443/api/frontends/active

# Comprobar si un frontend específico está activo
curl https://localhost:8443/api/frontends/history/puerta-sMPP-1 | jq
'.[0].status'

# Obtener estadísticas de salud
curl https://localhost:8443/api/frontends/stats
```

Otros Puntos Finales

Estado

```
GET /api/status
```

Ubicaciones

```
GET /api/locations
POST /api/locations
GET /api/locations/{id}
PATCH /api/locations/{id}
DELETE /api/locations/{id}
```

Eventos SS7

```
GET /api/ss7_events
POST /api/ss7_events
GET /api/ss7_events/{id}
PATCH /api/ss7_events/{id}
DELETE /api/ss7_events/{id}
```

Cola de Mensajes MMS

```
GET /api/mms_message_queues
POST /api/mms_message_queues
GET /api/mms_message_queues/{id}
PATCH /api/mms_message_queues/{id}
DELETE /api/mms_message_queues/{id}
```

Rendimiento

El núcleo SMS_C ofrece un rendimiento excepcional utilizando Mnesia para el almacenamiento de mensajes en memoria con archivo automático a SQL para la retención a largo plazo de CDR.

Resultados de Referencia

Medido en Intel i7-8650U @ 1.90GHz (8 núcleos):

Rendimiento de Inserción de Mensajes:

- `insert_message` (con enrutamiento): **1,750 msg/sec** (0.58ms latencia promedio)
- `insert_message` (simple): **1,750 msg/sec** (0.57ms latencia promedio)
- **Capacidad de ~150 millones de mensajes por día**

Rendimiento de Consulta:

- `get_messages_for_smsc`: 800 msg/sec (1.25ms promedio)
- `list_message_queues`: Acceso rápido en memoria
- Uso de memoria: 62 KB por operación de inserción

Arquitectura

Estrategia de Almacenamiento:

- **Mensajes Activos**: Almacenados en Mnesia (en memoria + disco) para acceso ultrarrápido
- **Archivo de Mensajes**: Archivado automáticamente a SQL para almacenamiento a largo plazo de CDR
- **Retención**: Período de retención configurable (predeterminado: 24 horas)
- **Sin cuello de botella SQL**: Todas las operaciones de mensajes activos evitan SQL

Configuración

El almacenamiento y la retención de mensajes se configuran en `config/runtime.exs`:

```
config :sms_c,
  message_retention_hours: 24,           # Archivar mensajes más
  antiguos de 24 horas
  batch_insert_batch_size: 100,         # Tamaño del lote de CDR para
  archivo SQL
  batch_insert_flush_interval_ms: 100   # Intervalo de vaciado de CDR
```

Para orientación detallada sobre ajustes, consulta: [docs/PERFORMANCE_TUNING.md](#)

Monitoreo

Panel de Control - Interfaz web en `https://localhost:8086`

- Ver cola de mensajes
- Enviar mensajes de prueba
- Gestionar enrutamiento de SMS (ver [Guía de Enrutamiento](#))
- Simular decisiones de enrutamiento
- Ver recursos del sistema
- Rastrear estadísticas de trabajadores por lotes

Estadísticas de Trabajadores por Lotes:

```
# Obtener estadísticas actuales del trabajador por lotes
SmsC.Messaging.BatchInsertWorker.stats()
```

Devuelve:

```
%{
  total_enqueued: 10000,
  total_flushed: 9900,
  current_queue_size: 100,
  last_flush_duration_ms: 45
}
```

Registros - Registros de la aplicación escritos en stdout

```
# Ver registros en tiempo real
tail -f log/dev.log
```

Solución de Problemas

Puerto Ya Está en Uso

```
# Encontrar proceso que usa el puerto
lsof -i :4000
```

```
# Matar el proceso
kill -9 <PID>
```

Frontend Externo No Conectando

Síntomas: Mensajes atascados en la cola, registros del frontend muestran errores de conexión

Verifica:

- Verifica que `API_BASE_URL` esté configurado correctamente en el frontend
- Comprueba que el núcleo `SMS_C` esté en ejecución y accesible
- Revisa las reglas de red/firewall
- Verifica la configuración del frontend

Solución:

```
# Probar conectividad API desde el frontend
curl http://localhost:4000/api/status

# Reiniciar frontend
export API_BASE_URL="http://localhost:4000"
# Iniciar aplicación frontend
```

Mensajes No Entregados

Síntomas: Mensajes permanecen no entregados, los intentos de reintento se incrementan

Verifica:

1. Registros del frontend para errores de envío
2. Conectividad de red externa
3. Configuración del frontend (credenciales, direcciones)
4. Compatibilidad del formato del mensaje

Ver mensajes fallidos:

```
# Obtener mensajes con intentos de reintento
curl https://localhost:8443/api/messages | jq '.data[] |
select(.delivery_attempts > 0)'
```

Alta Latencia de Mensajes

Síntomas: Mensajes tardando más de lo esperado, acumulación en la cola

Verifica:

1. Intervalo de sondeo del frontend (puede ser necesario disminuir para un sondeo más frecuente)
2. Rendimiento de la base de datos
3. Latencia de red hacia sistemas externos

Monitorear profundidad de la cola:

```
watch -n 5 'curl -s https://localhost:8443/api/messages | jq ".data |
length"'
```




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 al limpiar 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 el rendimiento de varias operaciones de 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.

Descripción 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
- [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 multiprotocolo (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

- [Enviar un Mensaje](#)
- [Crear una Ruta](#)
- [Verificar el Estado del Mensaje](#)
- [Monitorear la Salud del Sistema](#)
- [Manejar Fallos de Entrega](#)

Ejemplos de Configuración

- [Almacenamiento y Retención de Mensajes](#)
- [Configuración de Exportación CDR](#)
- [Controles de Privacidad](#)
- [Configuración de Alto Volumen](#)
- [Enrutamiento Geográfico](#)
- [Balanceo de Carga](#)
- [Configuración ENUM/NAPTR](#)
- [Carga OCS](#)
- [Traducción de Números](#)

Monitoreo y Alertas

- [Métricas Clave](#)
- [Alertas Recomendadas](#)
- [Plantillas de Tablero](#)

Descripción 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 Carga** - Carga OCS en línea 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 carga/facturación
- **DNS** - Búsquedas ENUM/NAPTR para enrutamiento

Distribución y Alta Disponibilidad

- **Clustering Multi-Nodo** - Procesamiento de mensajes distribuido
- **Replicación Mnesia** - Sincronización de rutas entre nodos
- **Failover Automático** - Manejo de fallos de nodo
- **Balanceo de Carga** - Distribución de rutas ponderadas

Documentación Relacionada

- [Referencias de Benchmarks de Rendimiento](#) - Pruebas de rendimiento y resultados
- [Referencia de 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)
- **4369** - Mapeador de Puertos Erlang (clustering)
- **9100-9200** - Distribución Erlang (clustering)
- **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

Obtener Ayuda

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

Información de la Versión

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 recolección 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/`.



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 futuras versiones utilizarán el versionado de URL:

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

Autenticación

Certificados de Cliente TLS (Recomendado)

Las implementaciones en producción deben utilizar la autenticación con 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 "smc: 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_smc": "api_client",  
    "dest_smc": "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:

- Necesite 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 el destino
- deliver_after - Retrasar la entrega
- message_body - Actualizar el texto del mensaje
- status - Cambiar el 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 codificado en hexadecimal (Unidad de Datos de Protocolo de Transporte)

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 - Acknowledgments de 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 información de ubicación de suscriptores para la entrega de mensajes terminados en 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 basada en 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 - Bandera 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": "smp",
      "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 del 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 del 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 Mensajes

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 descendente"
}'
```

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 MMS adicionales

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 en 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:** Hacer llamadas API concurrentes
4. **Monitorear Métricas:** Vigilar 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 de 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 de 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 de 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 de API: {e}")
            return False

```

```

# Uso
client = SMSClient("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 (asíncronicos)
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")

    # O 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

Funciones 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 Detalles 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 Archivo](#)
- [Integración de Facturación](#)

Descripción General

La tabla cdrs almacena Registros de Detalles 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 emisión de facturas
- 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
id	BIGINT	NO	Clave primaria autoincremental para el registro CDR

Identificación del Mensaje

Campo	Tipo	Nulo	Descripción
message_id	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
calling_number	VARCHAR(255)	NO	MSISDN de origen (número móvil) del remitente del mensaje. Típicamente en formato E.164 (por ejemplo, +15551234567).
called_number	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
source_smsc	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.
dest_smsc	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
origin_node	VARCHAR(255)	SÍ	Nombre del nodo Erlang donde se recibió originalmente el mensaje (por ejemplo, sms@node1.example.com). Útil para la solución de problemas y

Campo	Tipo	Nulo	Descripción
destination_node	VARCHAR(255)	SÍ	análisis de distribución de carga. 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 tiempo 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
status	VARCHAR(50)	NO	Estado final del mensaje. Valores válidos: delivered, expired, failed, rejected
delivery_attempts	INT	NO	Número de intentos de entrega realizados antes del estado final. Predeterminado: 0. Rango: 0-255 típicamente.
message_parts	INT	SÍ	Número de segmentos SMS para mensajes concatenados. 1 para mensajes de una sola parte, 2+ para multi-partes. NULL si se desconoce.
deadletter	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
delivered	Entregado con éxito al destinatario	Sí	Establecido
expired	Excedió el período de validez sin entrega	Depende de la política de facturación	NULL
failed	Fallo permanente de entrega (número inválido, etc.)	Depende de la política de facturación	NULL
rejected	Rechazado por reglas de enrutamiento o validación	No	NULL

Cuerpo del Mensaje

Campo	Tipo	Nulo	Descripción
message_body	TEXT	SÍ	El contenido real del mensaje SMS. Puede ser NULL si delete_message_body_after_delivery está habilitado por motivos 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/archivo
- **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 se actualizó.

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 multi-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 a través de 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 Índice 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
id	BIGINT AUTO_INCREMENT	BIGSERIAL	Entero de 64 bits, autoincremental
message_id	BIGINT	BIGINT	Entero de 64 bits
Campos de cadena	VARCHAR(255)	VARCHAR(255)	Cadena de longitud variable, máx. 255 caracteres
message_body	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 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. Cifrado de Base de Datos

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

MySQL:

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

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

4. Controles de Acceso

Restringir el acceso a la tabla CDR:

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

Retención y Archivo

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 Archivo

1. Particionar 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 (archivo rápido)
ALTER TABLE cdrs DROP PARTITION p202510;
```

2. Archivar en 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 Clúster](#)
- [Configuración de Cola de Mensajes](#)
- [Configuración de Carga](#)
- [Configuración de ENUM](#)
- [Configuración de Traducción de Números](#)
- [Configuración de Enrutamiento](#)
- [Configuración de Optimización 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/pruebas
- Parámetros de optimización 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 de CDR SQL

El SMS-C utiliza **Mnesia** para datos operacionales (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 Detalle 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	Mejor Para
MySQL	8.0+	<code>Ecto.Adapters.MyXQL</code>	3306	Propósito general, confiabilidad probada
MariaDB	10.5+	<code>Ecto.Adapters.MyXQL</code>	3306	Compatible con MySQL, código abierto
PostgreSQL	13+	<code>Ecto.Adapters.Postgres</code>	5432	Características avanzadas, soporte JSON

Nota: Mnesia se utiliza automáticamente para datos operacionales (cola de mensajes, enrutamiento, traducciones) y no requiere configuración. La base de datos SQL se utiliza **solo** para la 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")
```

Elección de una Base de Datos SQL

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

- Excelente rendimiento para escrituras de CDR
- Confiabilidad probada en entornos de telecomunicaciones
- Amplio soporte de herramientas para sistemas de facturación
- Configuración de replicación fácil

PostgreSQL - Considerar si necesita:

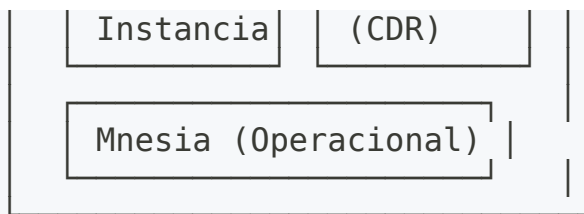
- Características avanzadas de JSON/JSONB para análisis
- Consultas complejas sobre datos de CDR
- Infraestructura existente de PostgreSQL
- PostGIS para análisis geográfico

Topologías de Implementación

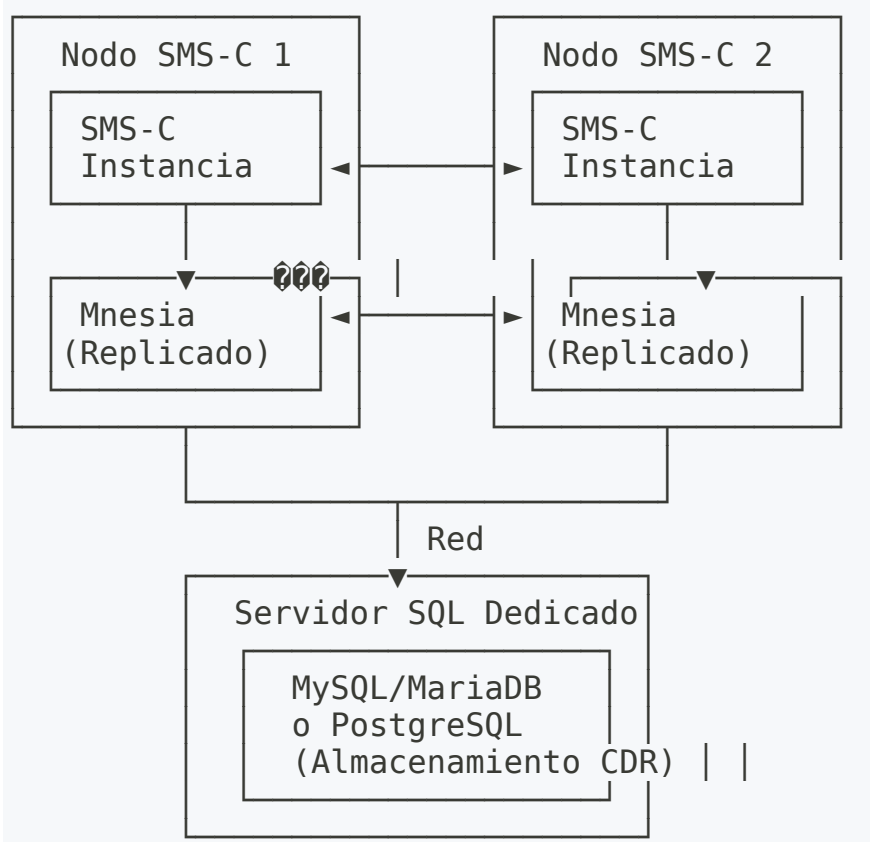
Importante: La base de datos SQL de CDR puede ejecutarse en un **servidor separado** de su(s) instancia(s) de SMS-C. Este es el enfoque recomendado para implementaciones en producción.

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 impactan el procesamiento de mensajes
- **Escalabilidad:** Escalar independientemente la base de datos y el procesamiento de mensajes
- **Confiabilidad:** 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: $\text{pool_size} = (\text{operaciones concurrentes esperadas en DB}) * 1.5$

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 para la conexión
- `ecto_pools_query_time` - Tiempo de ejecución de la consulta
- `ecto_pools_connected_count` - Conexiones activas

Alerta si el tiempo de espera supera los 100 ms de manera constante - 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, use certificados de una CA de confianza (Let's Encrypt, CA comercial, etc.).

Control de Acceso a la API

Lista Blanca de IPs (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 por Clave de API (Nivel de Aplicación):

Configurar a través de un plug personalizado en el enrutador - consulte la 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 Producción de Interfaz Web

```
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)

Utilice 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 Clúster

El SMS-C admite clústeres de múltiples nodos para alta disponibilidad y distribución de carga.

Configuración de Nodo Único

```

# config/runtime.exs
config :sms_c,
  cluster_nodes: [], # Lista vacía = modo de nodo único
  smsc_node_name: "node1"

```

Clúster Estático de Múltiples Nodos

```

# Nodo 1: config/runtime.exs
config :sms_c,
  cluster_nodes: [
    "sms@node1.internal.example.com",
    "sms@node2.internal.example.com",
    "sms@node3.internal.example.com"
  ],
  smsc_node_name: "node1"

# Nodo 2: config/runtime.exs
config :sms_c,
  cluster_nodes: [
    "sms@node1.internal.example.com",
    "sms@node2.internal.example.com",
    "sms@node3.internal.example.com"
  ],
  smsc_node_name: "node2"

```

Auto-Descubrimiento Basado en DNS

```

config :sms_c,
  dns_cluster_query: "sms-cluster.internal.example.com",
  smsc_node_name: System.get_env("NODE_NAME") || "node1"

```

Configuración de DNS para Auto-Descubrimiento:

```
# Configurar registros SRV o A para nodos de clúster
# Registro SRV (preferido):
_sms._tcp.sms-cluster.internal.example.com. IN SRV 0 0 0
node1.internal.example.com.
_sms._tcp.sms-cluster.internal.example.com. IN SRV 0 0 0
node2.internal.example.com.
_sms._tcp.sms-cluster.internal.example.com. IN SRV 0 0 0
node3.internal.example.com.

# Registros A (alternativa):
sms-cluster.internal.example.com. IN A 10.0.1.10
sms-cluster.internal.example.com. IN A 10.0.1.11
sms-cluster.internal.example.com. IN A 10.0.1.12
```

Configuración de Distribución de Erlang

Iniciar Nodos con Nombres Apropriados:

```
# Nodo 1
export NODE_NAME=sms@node1.internal.example.com
export ERLANG_COOKIE=shared_secret_cookie_here
elixir --name $NODE_NAME --cookie $ERLANG_COOKIE -S mix phx.server

# Nodo 2
export NODE_NAME=sms@node2.internal.example.com
export ERLANG_COOKIE=shared_secret_cookie_here
elixir --name $NODE_NAME --cookie $ERLANG_COOKIE -S mix phx.server
```

Importante: Todos los nodos en un clúster DEBEN usar la misma cookie de Erlang por razones de seguridad.

Requisitos de Red del Clúster

Abra estos puertos entre nodos del clúster:

Rango de Puertos	Protocolo	Propósito
4369	TCP	Demonio de Mapeo de Puertos de Erlang (EPMD)
9100-9200	TCP	Distribución de Erlang

Ejemplo de Configuración de Cortafuegos:

```
# Permitir tráfico de clúster desde la red interna
iptables -A INPUT -p tcp -s 10.0.0.0/8 --dport 4369 -j ACCEPT
iptables -A INPUT -p tcp -s 10.0.0.0/8 --dport 9100:9200 -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^(intento_count) 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 del registro 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 se desactiva, 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 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 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 obtiene 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
  ]
```

Optimización de Rendimiento de 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 regex 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 de teléfono de EE. UU.",
    enabled: true
}

```

Traducción Específica de Carrier

Eliminación de Código de Ruta:

```
%{
  calling_prefix: nil,
  called_prefix: "101",
101
  source_smsc: "carrier_a",
  calling_match: nil,
  calling_replace: nil,
  called_match: "^101(\d+)$",
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})$",
      calling_replace: "+1\1",
      called_match: "^(\\d{10})$",
      called_replace: "+1\1",
      priority: 50,
      description: "Normalización general de 10 dígitos",
```

```
    enabled: true
  }
]
```

Configuración de Enrutamiento

Las reglas de enrutamiento iniciales se cargan en el primer inicio. Consulte la [Guía de Enrutamiento SMS](#) para la documentación completa sobre enrutamiento.

Cargar Rutas desde la Configuración

```
# config/runtime.exs
config :sms_c,
  sms_routes: [
    # Ejemplo de enrutamiento geográfico
    %{
      calling_prefix: nil,
      called_prefix: "+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,
      weight: 100,
      priority: 50,
      description: "Enrutamiento de América del Norte",
      enabled: true
    },

    # Ejemplo de enrutamiento balanceado
    %{
      calling_prefix: nil,
      called_prefix: "+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,
      weight: 70,
      priority: 50,
      description: "Puerta de enlace primaria del Reino Unido (70%)",
      enabled: true
    }
  ]
```

```

    },
    %{
      calling_prefix: nil,
      called_prefix: "+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,
      weight: 30,
      priority: 50,
      description: "Puerta de enlace de respaldo del Reino Unido
(30%)",
      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 Optimización de Rendimiento

Consulte la [Guía de Optimización 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

Perfil	Tamaño del Lote	Intervalo	Rendimiento	Latencia
Balanceado	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
        # Señalizar a la aplicación para que reabra 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 (desactivada 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 (verbo)
config :logger, :console,
  level: :debug

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

# Carga (desactivada)
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

Clúster entre 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 para 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 la 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

Variable	Propósito	Ejemplo
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. **Utilizar 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 timeout
Rendimiento deficiente	Configuración de lotes incorrecta	Revisar Guía de Optimización de Rendimiento
Mensajes no enrutados	Rutas no cargadas	Verificar configuración de sms_routes o Interfaz Web

Para ayuda adicional, consulte 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 Memoria:

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

Exportación de CDR (Registro de Detalle de Llamadas)

Cuando los mensajes son entregados o expiran, los CDR pueden ser escritos automáticamente en su 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 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 el cumplimiento de 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
delete_message_body_after_delivery: true	Ahorrar espacio en Mnesia, cumplimiento de privacidad
hide_message_body_in_ui: true	Prevenir que el operador vea el contenido del mensaje
hide_message_body_in_export: true	Cumplimiento de exportación de datos, informes sanitizados

Configuraciones de Ejemplo:

Privacidad Máxima (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 registran el estado de la configuración:

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

Esto proporciona visibilidad inmediata sobre las características activas.



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 raspadas 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:

- `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 Procesamiento de Mensajes

`sms_c_message_received_count`

Tipo: Contador

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. Utilizar 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: Contador

Descripción: Número total de validaciones de mensajes realizadas.

Etiquetas:

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

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

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

sms_c_message_processing_stop_duration

Tipo: Histograma

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

Unidad: Milisegundos

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

Etiquetas:

- `success`: Si el procesamiento fue exitoso (true o false)

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

Alertas: Alertar cuando la latencia p95 o p99 exceda los umbrales de SLA.

Métricas de Enrutamiento

sms_c_routing_route_matched_count

Tipo: Contador

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

Alertas: Alertar si las rutas de alta prioridad rara vez son coincidentes (puede indicar una mala configuración de enrutamiento).

sms_c_routing_failed_count

Tipo: Contador

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

sms_c_routing_action_count

Tipo: Contador

Descripción: Número total de acciones de enrutamiento especiales realizadas.

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: Alertar sobre picos inesperados en acciones de eliminación (puede indicar un ataque de spam).

sms_c_routing_stop_duration

Tipo: Histograma

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

Unidad: Milisegundos

Cubos: 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: Alertar 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: Contador

Descripción: Número total de búsquedas ENUM atendidas 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: Alertar 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: Contador

Descripción: Número total de búsquedas ENUM que requirieron una consulta DNS (no están 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: $\text{cache_hit_rate} = \text{hits} / (\text{hits} + \text{misses})$

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

sms_c_enum_lookup_stop_duration

Tipo: Histograma

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

Unidad: Milisegundos

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

Etiquetas:

- domain: Dominio ENUM consultado
- success: Si la búsqueda fue exitosa (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: Alertar cuando el tiempo de búsqueda p95 exceda el umbral de tiempo de espera.

sms_c_enum_naptr_records_record_count

Tipo: Histograma

Descripción: Número de registros NAPTR devueltos por una búsqueda ENUM exitosa.

Cubos: 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: Alertar si frecuentemente se devuelven 0 registros (problema de configuración de DNS).

Métricas de Entrega

sms_c_delivery_queued_count

Tipo: Contador

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 hacia 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: Contador

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: Contador

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: Alertar si la tasa de éxito cae por debajo del umbral de SLA.

Cálculo: $\text{success_rate} = \text{succeeded} / \text{queued}$

`sms_c_delivery_failed_count`

Tipo: Contador

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: Alertar sobre tasas de fallo elevadas o razones de fallo específicas.

sms_c_delivery_dead_letter_count

Tipo: Contador

Descripción: Número total de mensajes movidos a la cola de cartas muertas (no entregables).

Etiquetas:

- reason: Razón para la carta muerta (por ejemplo, max_retries_exceeded, expired)

Caso de Uso: Monitorear mensajes no entregables que requieren intervención manual.

Alertas: Alertar sobre cualquier evento de carta muerta ya que representan un fallo completo en la entrega.

sms_c_delivery_succeeded_duration

Tipo: Histograma

Descripción: Tiempo de extremo a extremo desde que el mensaje se encola hasta la entrega exitosa.

Unidad: Milisegundos

Cubos: 100, 500, 1000, 5000, 10000, 30000, 60000 ms

Etiquetas:

- dest_smsc: Nombre del SMSC de destino

Caso de Uso: Monitorear la latencia de entrega. Identificar destinos lentos o problemas de red.

Alertas: Alertar cuando el tiempo de entrega p95 exceda los umbrales de SLA.

sms_c_delivery_succeeded_attempt_count

Tipo: Histograma

Descripción: Número de intentos de entrega requeridos antes de la entrega exitosa.

Cubos: 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: Alertar si el conteo promedio de intentos excede 2 (indica problemas de fiabilidad del destino).

sms_c_delivery_failed_attempt_count

Tipo: Histograma

Descripción: Número de intentos de entrega realizados antes del fallo final.

Cubos: 1, 2, 3, 5, 10

Etiquetas:

- dest_smsc: Nombre del SMSC de destino

Caso de Uso: Entender cuántos reintentos ocurren antes de rendirse.

Métricas de Cola

sms_c_queue_size_size

Tipo: Gauge

Descripción: Número total actual de mensajes en la 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: Alertar cuando el tamaño de la cola exceda 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: Alertar cuando el conteo pendiente exceda el umbral durante un período prolongado.

sms_c_queue_size_failed

Tipo: Gauge

Descripción: Número actual de mensajes en estado de fallo (a la espera de reintento).

Etiquetas:

- queue_type: Tipo de cola

Caso de Uso: Monitorear la acumulación de mensajes fallidos. Indica problemas de entrega.

Alertas: Alertar sobre conteos fallidos elevados ya que impactan las tasas de entrega.

sms_c_queue_size_delivered

Tipo: Gauge

Descripción: Número actual de mensajes entregados a la espera de 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: Alertar 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 los mensajes y los retrasos en el procesamiento. Crítico para el monitoreo de SLA.

Alertas: Alertar cuando la edad del mensaje más antiguo exceda el umbral de SLA (por ejemplo, > 300 segundos).

Métricas de Cobro

sms_c_charging_requested_count

Tipo: Contador

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 facturación.

sms_c_charging_succeeded_count

Tipo: Contador

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: Contador

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 en la cuenta.

Alertas: Alertar sobre tasas de fallo de cobro elevadas.

sms_c_charging_succeeded_duration

Tipo: Histograma

Descripción: Tiempo tomado para completar una solicitud de cobro exitosa.

Unidad: Milisegundos

Cubos: 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: Alertar cuando el tiempo de cobro p95 exceda 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.

Etiquetas:

- table: Nombre de la tabla (por ejemplo, sms_route)

Caso de Uso: Monitorear el crecimiento de la base de datos. Detectar acumulación de datos inesperada.

Alertas: Alertar sobre tasas de crecimiento inesperadas de la tabla.

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: Alertar cuando los frontends esperados se desconecten.

sms_c_location_registered_count

Tipo: Contador

Descripción: Número total de registros de ubicación/suscriptor recibidos por el sistema.

Etiquetas:

- location: Nombre del frontend/SMSC donde el suscriptor está registrado
- 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 o fallos de registro.

Alertas: Establecer alertas para:

- Caídas en la tasa de registro (puede indicar problemas de red)
- Picos inusuales en los registros
- Alta proporción de registros no IMS (afluencia de dispositivos heredados)

Consulta de Ejemplo:

```
# Tasa de registro por minuto
rate(sms_c_location_registered_count[1m])

# Proporción 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]))
```

Métricas de Solicitudes de API HTTP

phoenix_endpoint_stop_duration

Tipo: Distribución (Histograma)

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 API (por ejemplo, /api/messages, /api/frontends)

Cubos: 10ms, 50ms, 100ms, 250ms, 500ms, 1s, 2.5s, 5s

Caso de Uso: Monitorear el rendimiento de la API. Identificar endpoints lentos. Rastrear los SLA 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

Consulta de Ejemplo:

```
# 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: Contador

Descripción: Número total de solicitudes HTTP completadas, categorizadas por ruta y código de estado HTTP.

Etiquetas:

- route: Ruta del endpoint de API
- status: Código de estado HTTP (200, 201, 400, 404, 500, etc.)

Caso de Uso: Monitorear el volumen de solicitudes de API y las 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

Consulta de Ejemplo:

```
# 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: Contador

Descripción: Número total de excepciones/errores generados durante el procesamiento de solicitudes HTTP.

Etiquetas:

- route: Ruta del endpoint de 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.

Consulta de Ejemplo:

```
# 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: Alertar 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: Alertar sobre una 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: Alertar cuando consistentemente > 10 * 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: Alertar 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 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:

- `sms_c_message_processing_stop_duration` (procesamiento)
- `sms_c_delivery_succeeded_duration` (entrega)

Interpretación: La suma representa la latencia total que enfrenta el usuario.

Efectividad de la Caché ENUM

Medir cuán 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 variación 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` (pendiente actual)
- `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 p95 > 1, la mayoría de los mensajes requieren reintentos. Indica problemas de fiabilidad del destino.

Alertas Recomendadas

Alerta	Condición	Severidad	Descripción
Alta Tasa de Fallos de Enrutamiento	routing_failed_count aumento	Crítica	Los mensajes no pueden ser enrutados
Acumulación de Cola Mensajes Antiguos en Cola	queue_size_pending > umbral	Advertencia	Mensajes acumulándose
Pico de Fallos de Entrega	queue_oldest_message_age_seconds > 300	Crítica	Violación de SLA
Eventos de Carta Muerta	delivery_failed_count pico	Alta	Problemas de destino
Timeouts de Búsqueda ENUM	delivery_dead_letter_count > 0	Alta	Mensajes no entregables
Baja Tasa de Aciertos en Caché	enum_lookup_stop_duration p95 > 5000ms	Advertencia	Problemas de DNS
Frontend Desconectado	enum_lookup_stop_duration p95 > 5000ms	Advertencia	Caché ineficaz
Fallos de Cobro	Tasa de aciertos en caché ENUM < 0.7	Alta	Pérdida de conectividad
Procesamiento de Mensajes Lento	frontend_status_count{status="disconnected"} > 0	Alta	Problemas de facturación
	charging_failed_count > umbral	Advertencia	Degradación del rendimiento
	message_processing_stop_duration p95 > 1000ms		

Recomendaciones de Tablero

Tablero 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 colas (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

Tablero 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
-

Tablero 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 bruto:** 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 el destino problemático
-

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 el cuello de botella e investigar el componente específico
-

Escenario: Crecimiento de la Cola de Mensajes

Pasos de Investigación:

1. Verificar la 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_succeeded_duration - ¿Está tardando demasiado la entrega?
 5. Verificar etiquetas dest_smsc para identificar destinos lentos
-

Ejemplos de Consultas de 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 Origen:

```
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_succeeded_duration_bucket)
```

Tiempo de Entrega (p99):

```
histogram_quantile(0.99, sms_c_delivery_succeeded_duration_bucket)
```

Métricas de Cola

Mensajes Pendientes Actuales:

```
sms_c_queue_size_pending
```

Mensajes Fallidos a la Espera de Reintento:

```
sms_c_queue_size_failed
```

Edad del Mensaje Más Antiguo (Minutos):

```
sms_c_queue_oldest_message_age_seconds / 60
```

Tasa de Crecimiento de la 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])))
```

Latencia de Enrutamiento (p50, p95, p99):

```
histogram_quantile(0.50, sms_c_routing_stop_duration_bucket)
histogram_quantile(0.95, sms_c_routing_stop_duration_bucket)
histogram_quantile(0.99, sms_c_routing_stop_duration_bucket)
```

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 (En Caché vs No En Caché):

```
# En caché (rápido)
rate(sms_c_enum_cache_hit_count[5m])
```

```
# No en caché (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 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 Tableros de Grafana

Tablero 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 la 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

Tablero 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_succeeded_duration_bucket)`
- Unidad: milisegundos
- Barras horizontales

3. Distribución de Intentos de Entrega (Heatmap)

- Consulta: `sms_c_delivery_succeeded_attempt_count_bucket`
- Muestra cuántos intentos son típicamente necesarios
- Escala de color: Azul (1 intento) a Rojo (muchos intentos)

4. Rendimiento de la Caché ENUM (Gráfico)

- Tasa de Aciertos: $\text{rate}(\text{sms_c_enum_cache_hit_count}[5\text{m}]) / (\text{rate}(\text{sms_c_enum_cache_hit_count}[5\text{m}]) + \text{rate}(\text{sms_c_enum_cache_miss_count}[5\text{m}]))$
- Tamaño de la Caché: `sms_c_enum_cache_size_size`
- Eje Y dual (tasa vs tamaño)

5. Tasa de Éxito de Procesamiento (Gauge)

- Consulta: $(\text{rate}(\text{sms_c_message_processing_stop_duration_count}\{\text{success}=\text{"true"}\}[5\text{m}]) / \text{rate}(\text{sms_c_message_processing_stop_duration_count}[5\text{m}])) * 100$
- Unidad: porcentaje
- Umbrales:
 - Rojo: < 95
 - Amarillo: 95-99
 - Verde: > 99

Tablero 3: Análisis de Tráfico

Propósito: Analizar patrones de tráfico de mensajes y distribución de enrutamiento.

Paneles:

1. Mensajes por Tipo de Origen (Gráfico de Pastel)

- Consulta: `sum by (source_type) (increase(sms_c_message_received_count[1h]))`
- Muestra la 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:
 - ID de Ruta
 - 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

Tablero 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 la 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: $\text{rate}(\text{sms_c_delivery_queued_count}[5\text{m}]) - (\text{rate}(\text{sms_c_delivery_succeeded_count}[5\text{m}]) + \text{rate}(\text{sms_c_delivery_failed_count}[5\text{m}]))$
- Positivo = acumulación creciente
- Negativo = poniéndose al día

4. Tráfico Máximo (Stat)

- Consulta: $\text{max_over_time}(\text{rate}(\text{sms_c_message_received_count}[5\text{m}])[24\text{h}])$
- Muestra la tasa más alta de 5 minutos en las últimas 24 horas
- Unidad: mensajes/segundo

5. Utilización de Capacidad (Gauge)

- Consulta: $(\text{rate}(\text{sms_c_message_received_count}[5\text{m}]) / \text{MAX_CAPACITY}) * 100$
- Reemplazar MAX_CAPACITY con el límite de su sistema
- Unidad: porcentaje
- Umbrales:
 - Verde: < 70
 - Amarillo: 70-85
 - Rojo: > 85

Tablero 5: Cumplimiento de SLA

Propósito: Rastrear métricas de SLA y cumplimiento.

Paneles:

1. Cumplimiento de SLA (Gauge)

- Éxito de Entrega: $(\text{rate}(\text{sms_c_delivery_succeeded_count}[1\text{h}]) / \text{rate}(\text{sms_c_delivery_queued_count}[1\text{h}])) * 100$
- Línea objetivo en 99%
- Umbrales:
 - Rojo: < 95
 - Amarillo: 95-99
 - Verde: >= 99

2. Mensajes Entregados Dentro del SLA (Stat)

- Consulta: $\text{count}(\text{sms_c_delivery_succeeded_duration_bucket}\{\text{le}="5000"\}) / \text{count}(\text{sms_c_delivery_succeeded_duration_bucket})$
- Muestra el porcentaje entregado dentro de 5 segundos
- Unidad: porcentaje

3. Violaciones de SLA (Contador)

- Mensajes que exceden 5 minutos:

- `increase(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 el 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

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 no entregables. 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 los fallos 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 informar.
- Las métricas de contador son acumulativas y deben usarse 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 percentiles (p50, p95, p99) y pueden usarse para crear mapas de calor.
- Todas las métricas incluyen etiquetas predeterminadas añadidas por Prometheus (instancia, trabajo, etc.).
- Al crear tableros, 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 tablero.
- Use plantillas de variables en Grafana para tableros dinámicos (seleccionar `dest_smsc`, `source_smsc`, etc.).



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. Verificar 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 (debe ser < 1%)
- Acumulación de cola (debe ser < 1000 pendientes)
- Tasa de éxito de entrega (debe ser > 95%)
- Estado de conexión del frontend (todos los frontends esperados activos)

3. Verificar la Cola de Mensajes

Acceda a la interfaz web: https://sms-admin.example.com/message_queue

Revise:

- Total de mensajes pendientes (debe ser bajo)
- Edad del mensaje más antiguo (debe 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. Verificar 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 de clúster

Monitoreo del Volumen de Mensajes

Verificar los Contadores 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/fines de semana: Menor volumen
- Tasa de entrega: Debe ser $> 95\%$

Condiciones de Alerta:

- Caída repentina en mensajes ($> 50\%$ de disminución)

- Aumento repentino en mensajes (> 200% de aumento)
- Tasa de entrega cae 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 Tablero Operativo:

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 (Respuesta Inmediata Requerida):

```
# 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 atrasado
- 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 (Investigación Necesaria):

```
# Tasa de éxito de entrega en caída
- 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 excedió el límite de reintentos

Verificar 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 falla

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 consulte:

```
# 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 funcione el enrutamiento basado en ubicación:

- 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 registrarse nuevamente

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
```

Verificar 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)
 - **Prioridad:** Prioridad de la ruta (1-255, menor = mayor prioridad)
 - **Peso:** Peso de balanceo de carga (1-100)

- **Descripción:** Descripción legible por humanos
 - **Habilitado:** Marque 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 Balanceada por 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: "Primaria del Reino Unido (70%)"

Ruta 2:

- Prefijo Llamado: +44
- SMSC de Destino: uk_backup
- Prioridad: 50
- Peso: 30
- Descripción: "Respaldo del Reino Unido (30%)"

Probar Rutas

Simulador de Enrutamiento:

1. Navegue a /simulator
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 de 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:** Desmarque "Habilitado" (eliminación temporal)
- **Ajustar Peso:** Cambiar la distribución de balanceo de carga
- **Cambiar Prioridad:** Reordenar la evaluación de la 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 las rutas existentes
 - **Reemplazar:** Eliminar todas 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 en busca de 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 (debe 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 en busca de 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 del 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 anteriores

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 10 dígitos de EE. UU.",
  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 de 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 de carrier 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 hayan transformado correctamente

Deshabilitar Regla de Traducción

Establezca `enabled: false` en la regla:

```
%{
  ...
  enabled: false
}
```

Reinicie la aplicación.

Mantenimiento del Sistema

Mantenimiento de la Base de Datos

Verificar 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:** Consulta `information_schema.tables` para el tamaño de la base de datos
- **PostgreSQL:** Utilice la función `pg_database_size()` o el comando `\l+` en `psql`

Limpiar Registros de CDR Antiguos:

Los registros de CDR deben ser archivados y eliminados periódicamente según su política de retención:

- Configure el archivo de archivo 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 un período de bajo tráfico para mantener un rendimiento óptimo.

Mantenimiento de la Base de Datos Mnesia

Verificar Tamaño de la Tabla Mnesia:

```
# En la consola de 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 de la copia de seguridad de Mnesia
:mnesia.backup("/var/backups/sms_c/mnesia_backup.bup")
```

Restaurar Mnesia:

```
# A través de la importación de la Interfaz Web
# O restaurar copia de seguridad:
: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
shardscripts
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 en busca de errores
- Verifique que el procesamiento de mensajes se reanude

Copia de Seguridad y Recuperación

Qué Respalidar

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 copia de seguridad 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 la 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
# - Comprima la salida para ahorrar espacio
# - Configure el 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):

```
# Diariamente 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 dirija 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 copia de seguridad
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 en 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 están entregando 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 con una sola ruta
- Alertas de advertencia no críticas

Lista de Verificación de Incidentes

1. Evaluar Severidad:

- Verificar alertas de Prometheus
- Revisar métricas del panel

- Verificar estado de la cola de mensajes
- Verificar conexiones de frontend

2. Reunir Información:

- ¿Se realizaron cambios de configuración recientes?
- ¿Se realizaron implementaciones recientes?
- Estado de dependencias externas (OCS, DNS)?
- ¿Mensajes de error en los registros?

3. Acciones Inmediatas:

- Detener cambios en curso
- Revertir implementaciones recientes si se sospecha que son la causa
- Habilitar registros detallados si es necesario
- Notificar a las partes interesadas

4. Investigación:

- Revisar registros de la aplicación
- Verificar uso de recursos del sistema
- Examinar rendimiento de la base de datos
- Probar dependencias externas

5. Resolución:

- Aplicar la solución
- Probar en el simulador
- Implementar en producción
- Monitorear para mejoras

6. Post-Incidente:

- Documentar la causa raíz
- Actualizar monitoreo/alertas
- Implementar medidas preventivas
- Actualizar manuales de operación

Incidentes Comunes

Alta Acumulación de Cola:

1. Verificar tasa de éxito de entrega
2. Verificar que los frontends estén conectados y consultando
3. Verificar rendimiento de la base de datos
4. Revisar Prometheus en busca de cuellos de botella
5. Considerar aumentar el tamaño/intervalo de lotes

Fallos de Enrutamiento:

1. Revisar configuración de enrutamiento
2. Probar en simulador de enrutamiento
3. Verificar rutas faltantes
4. Verificar que exista una ruta de captura

5. Revisar registros de eventos para razones de falla

Desconexiones de Frontend:

1. Verificar estado del sistema del frontend
2. Verificar conectividad de red
3. Revisar registros del frontend
4. Probar registro manual de API
5. Verificar reglas de firewall

Procesamiento de Mensajes Lento:

1. Verificar rendimiento de consultas de base de datos
2. Revisar configuración de trabajadores por lotes
3. Verificar recursos adecuados (CPU/Memoria)
4. Verificar retrasos en búsquedas de ENUM
5. Revisar rendimiento del sistema de carga

Para procedimientos de solución de problemas detallados, consulte la [Guía de Solución de Problemas](#).



Guía de Optimización de 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.

Resumen 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 Mnesia](#)
- [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:

Almacenamiento Activo de Mensajes (Mnesia)

- **Propósito:** Inserción, enrutamiento y entrega de mensajes ultra-rápidos
- **Almacenamiento:** En memoria con persistencia en disco (disc_copies)
- **Rendimiento:** 1,750 msg/sec de rendimiento de inserción, 0.58ms de latencia
- **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
 - Archivado 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
 - Archivado 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

Archivado 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 archivado)

Equilibrado (Recomendado)

```
batch_insert_batch_size: 100
```

```
batch_insert_flush_interval_ms: 100
```

- Buen equilibrio para la mayoría de las implementaciones
- CDRs 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 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 completo de tabla)  
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 de CDR pesados
```

Directrices:

- Implementación estándar: pool_size: 10
- Informes de CDR pesados: pool_size: 20-30
- Solo archivado: pool_size: 5

Pruebas de Rendimiento

Ejecución de Pruebas

El proyecto incluye pruebas basadas en Benchee para pruebas de rendimiento:

```
# Prueba de rendimiento de API SMS en crudo (compara sincrónico vs  
asíncrono)
```

```
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:

Nombre	desviación	mediana	99º %	ips	promedio
submit_message_raw_async (batch)	±41.72%	0.184 ms	0.55 ms	4.65 K	0.22 ms
submit_message_raw (sync)	±33.42%	12.57 ms	33.71 ms	0.0696 K	14.36 ms

Métricas clave:

- **ips:** Iteraciones por segundo (más alto es mejor)
- **promedio:** Tiempo de ejecución promedio (más bajo es mejor)
- **mediana:** Valor medio, más representativo que el promedio para distribuciones sesgadas
- **99º %:** Latencia del percentil 99 (importante para el cumplimiento de 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
- ♡ Post-procesamiento asíncrono (enrutamiento, carga, eventos)
- ♡ Almacenamiento en memoria de Mnesia vs I/O de disco de MySQL
- ♡ Reducción del 50% en memoria

Monitoreo

Estadísticas de 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)
- Aumento de la duración de flush (degradación del rendimiento de la base de datos)
- Errores de flush > 0 (problemas de conectividad con 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 de disco, optimice tablas
4. Usando API asíncrona cuando necesita sincrónica: Cambie a punto final 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 zonas horarias** - la carga máxima varía según la región

Configuraciones de Ejemplo

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 en Tiempo Real para Empresas

```
# 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](#)



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 en el 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 el encabezado smsc correctamente?

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 configuración de enrutamiento
- Verificar prioridades de ruta
- Probar en simulador de enrutamiento
- Verificar que el nombre del frontend coincida con `dest_smsc` en los mensajes

Mensajes Programados para el Futuro:

- Verificar marca de tiempo `deliver_after`
- Restablecer 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` en aumento
- 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 descendente 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"}'

# Restablecer 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 verdaderamente 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 en aumento
- 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 deshabilitó accidentalmente las rutas

Ruta Incorrecta Seleccionada

Síntomas:

- Mensajes enrutados a 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 puntajes de prioridad y 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 (prioridad más baja)

Ajustar Pesos:

Cambiar distribución de balanceo de carga:

Primario (70%):

Peso: 70

Respaldo (30%):

Peso: 30

Agregar Ruta Más Específica:

Sobrescribir 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 se seleccione la ruta
- 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 mayor 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 de API lentas
- 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 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 de OCS
- Aumentar tiempo de espera de 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 de 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 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:

- `sms_c_queue_size_pending` en aumento
- Edad del mensaje más antiguo en aumento

- El procesamiento no puede seguir el 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
- Solucionar problemas descendentes
- 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: "no se puede conectar a la base de datos"
- API devuelve 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
# 0
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=usuario_correcto  
export DB_PASSWORD=contraseña_correcta  
  
# 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 del valor actual
```

Consultas Lentas

Síntomas:

- Tiempo de consulta de base de datos alto
- Respuestas de API lentas
- Cola de 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 en el Frontend

Frontend No Aparece como Activo

Síntomas:

- El estado del frontend muestra "expirado"
- El frontend no está en la lista activa
- Los 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 '{
  "frontend_name": "uk_gateway",
  "frontend_type": "smpp",
  "ip_address": "10.0.1.50"
}';
```

Si tiene éxito, el problema está en el código/configuración del frontend.

Registro Expirando:

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_smsc()
    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 registros en el historial
- Conexión inestable

Pasos de Diagnóstico:

1. Verificar Salud del Frontend:

- ¿Es estable el proceso del frontend?
- ¿Hay bloqueos 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:

- Solucionar problemas de la aplicación del frontend
- Aumentar recursos del frontend
- Revisar registros del frontend para errores

Problemas de Red:

- Verificar conectividad intermitente
- Revisar registros de firewall
- Verificar chequeos de salud del balanceador de carga

Intervalo de Registro Incorrecto:

Intervalo correcto:

```
REGISTRATION_INTERVAL = 60 # segundos
```

Problemas de Carga/Facturación

Fallos de Carga

Síntomas:

- sms_c_charging_failed_count en aumento
- Registro de eventos muestra "charging_failed"
- Mensajes marcados como charge_failed: true

Pasos de Diagnóstico:

1. Verificar Conectividad OCS:

```
# Probar API OCS
curl -X POST http://ocs.example.com:2080/jsonrpc \
  -H "Content-Type: application/json" \
  -d '{"method": "SessionSv1.Ping",
    "params": [],
    "id": 1
  }';
```

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://host-correcto:2080/jsonrpc",
  ocs_tenant: "tenant_correcto"
```

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 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 Corto:

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 el DNS personalizado alcanzable?  
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: [  
    {"8.8.8.8", 53}, # DNS Público de Google  
    {"1.1.1.1", 53}, # DNS de Cloudflare
```

```
]
```

Dominio ENUM Incorrecto:

Actualizar dominio:

```
config :sms_c,  
  enum_domains: ["el64.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:

- Baja tasa de aciertos de caché (< 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:

- ¿Son los números repetidos?
- ¿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 memoria del sistema 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=valor_secreto_mismo_aquí

# 0 actualizar ~/.erlang.cookie
echo "valor_secreto_mismo_aquí" > ~/.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
- Diferentes datos 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 de 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 espera de conexión
- 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 Devuelve 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
- Solucionar problema de conexión

Sin Memoria:

- Reiniciar aplicación
- Aumentar memoria del sistema
- Verificar fugas de memoria

Error de Aplicación:

- Verificar error específico en registros
- Solucionar problema de configuración
- Reiniciar aplicación

Problemas de Interfaz Web

No Se Puede Acceder a la Interfaz Web

Síntomas:

- Tiempo de espera de conexión
- 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 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 de Proxy:

Si se usa un proxy inverso, asegurar soporte para WebSocket:

```
location /live &#123;  
    proxy_http_version 1.1;  
    proxy_set_header Upgrade $http_upgrade;  
    proxy_set_header Connection "upgrade";  
&#125;
```

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`



Documentación de Cumplimiento de Intercepción ANSSI R226

Propósito del Documento: Este documento proporciona las 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)

Reglamento: R226 - Protección de la Privacidad de la 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 Principales

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 opcional en disco
 - 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 SOLO 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: Extremadamente rápida (1,750 msg/sec) sin cuellos de botella SQL
 - Archivo CDR: Retención a largo plazo (meses/años) para facturación e interceptación legal
 - Separación clara: 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 Intercepción

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 de Usuario (UDH)

Adquisición de Metadatos de Mensajes:

- Registros de Detalles de Llamadas (CDR) completos almacenados en la base de datos con:
 - ID del 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 se entregó el mensaje)
 - 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/multiparte)
 - 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 SMSd 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:** 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 la cola de mensajes (inserción, actualización, estado de entrega, enrutamiento)
- **Característica Crítica:** La base de datos SQL NUNCA es consultada 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 SOLO cuando los mensajes alcanzan el 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 en la base de datos)
2. **Escalabilidad:** Mnesia maneja más de 1,750 mensajes/segundo sin sobrecarga de SQL
3. **Fiabilidad:** El modo `disc_copies` asegura que no haya pérdida de mensajes en caso de fallo
4. **Cumplimiento:** La base de datos CDR proporciona una auditoría permanente
5. **Separación de Preocupaciones:** Datos operativos vs. datos de archivo 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 configurable del cuerpo del mensaje o eliminación por privacidad
- Preservación de datos binarios (almacenamiento de PDU en bruto tanto en Mnesia como en CDR)
- Capacidad de búsqueda de texto completo (si está habilitada en la base de datos CDR)
- Campos de 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 específico 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 de 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 está habilitada)
- 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 de 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 la 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)
- Eliminación configurable del cuerpo del mensaje 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 de Mensajes:

- Eliminación configurable del cuerpo del mensaje 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 tablas 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

Fortalecimiento 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 y el cumplimiento a largo plazo.

Nivel 1: Cola de Mensajes en Memoria Mnesia

¿Qué es Mnesia?

- Base de datos distribuida integrada en el tiempo de ejecución de Erlang/OTP
- Almacenamiento híbrido: Primario en memoria con respaldo automático en disco
- Transacciones compatibles con ACID
- Replicación de clúster a través de 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 las 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
 - Copia en disco actualizada en cada confirmación de transacción
 - Recuperación ante fallos: El sistema se reinicia con todos los mensajes intactos

- Ubicación: directorio Mnesia.* / en los datos de la aplicación

Ciclo de Vida del Mensaje en Mnesia:

1. Mensaje llega a través de REST API → Insertado en Mnesia RAM + 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 de 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 de CDR se crean SOLO 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 la escritura en 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
-  SOLO utilizada para: Archivado a largo plazo de CDR y consultas históricas
-  SOLO utilizada para: Consultas de interceptación legal (meses/años de historia)
-  SOLO 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 ultrarrápida
- 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 los fallos
2. **Auditoría permanente:** CDRs retenidos durante años en la base de datos SQL
3. **Rendimiento:** Consultas de intercepción 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 SMS Sc emplea un diseño central independiente de protocolo que se conecta con puertas de enlace (frontends) específicas de protocolo externas a través de una REST API unificada. Esta arquitectura permite que la intercepción legal capture mensajes independientemente de qué protocolo de telecomunicaciones se utilizó para enviarlos o recibirlos.

Visión General de la Arquitectura

Detalles de Integración de Protocolo de Frontend

1. Integración de Frontend IMS/SIP

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

Datos de Intercepción Específicos de IMS:

- IMSI de origen/destino (de registro IMS)
- Encabezados SIP de Identidad Afirmada P
- SIP Call-ID para correlación
- Ubicación de red IMS (P-Access-Network-Info)
- Perfiles de suscriptores de HSS 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 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
- Identificación 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 SS7 MAP para SMS. La puerta de enlace MAP traduce entre señalización SS7 y REST API.

Datos de Intercepción Específicos de SS7/MAP:

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

Intercepción Unificada a Través de Todos los Protocolos

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

1. **Monitoreo Independiente de Protocolo:** Un único punto de intercepción captura todos los tipos de mensajes
2. **Formato CDR Unificado:** Todos los protocolos escriben en el mismo esquema de CDR
3. **Correlación Inter-Protocolo:** Seguimiento de 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 del frontend (por ejemplo, "ims.gateway-01", "smpp.customer123", "map.msc-01")
- Permite filtrado y análisis por tipo de protocolo
- Consultas de intercepción legal pueden dirigirse a protocolos específicos o a todos los protocolos

1.6 Arquitectura Técnica para Intercepción Legal

Puntos de Integración de Intercepción Legal

La arquitectura de almacenamiento de dos niveles proporciona múltiples puntos de acceso para la intercepción 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 Intercepción 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 acceden 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 a 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. Alimentación 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 Estándar de Intercepción Legal de ETSI

El OmniMessage SMS*Sc* proporciona la base para implementar interfaces de intercepción legal compatibles con ETSI. Si bien el núcleo del SMS*Sc* 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 intercepción de las fuerzas del orden al sistema de intercepción
- **Dirección:** LEMF → LIMF (bidireccional)
- **Funciones:**
 - Activar/desactivar la intercepción para objetivos específicos (MSISDN, IMSIs)
 - Establecer duración de intercepción y período de validez
 - Configurar criterios de filtrado (números de teléfono, ventanas de tiempo)
 - Recuperar estado de intercepción
- **Integración con SMS*Sc*:**
 - LIMF mantiene la lista de objetivos (base de datos de órdenes)
 - LIMF consulta CDR/API del SMS*Sc* 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 las fuerzas del orden
- **Dirección:** LIMF → LEMF (unidireccional)
- **Formato de Datos:** XML/ASN.1 conforme a ETSI TS 102 232-x
- **Contenido del SMSc CDR:**
 - ID del 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 los registros CDR en formato IRI de ETSI
 - 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 las fuerzas del orden
- **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 CC de ETSI
 - LIMF entrega CC a LEMF a través de X3

Arquitectura de Implementación:

Mapeo de Datos del SMSc a Interfaces LI:

Campo de Datos del SMSc	X2 (IRI)	X3 (CC)	Columna de Tabla CDR
ID del Mensaje	◈ ID de correlación	◈ Referencia	message_id
Número de Llamada	◈ Parte A	-	calling_number

Campo de Datos del SMSc	X2 (IRI)	X3 (CC)	Columna de Tabla CDR
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 Alimentación en Tiempo Real

- SMSc PubSub publica eventos de mensajes
- LIMF se suscribe a la corriente de mensajes en tiempo real
- LIMF filtra según la lista de objetivos
- Latencia casi nula para la intercepción legal
- Requiere desarrollo de integración personalizada

Opción 3: Arquitectura Híbrida

- Mensajes recientes: Alimentación en tiempo real de PubSub (< 24 horas)
- Mensajes históricos: Polling de base de datos CDR
- Equilibrio óptimo de latencia y fiabilidad

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 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 SMSG 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 por defecto 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:

- Perfect Forward Secrecy (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 de 2048 bits, recomendado 4096 bits
- Soporte de ECDSA
- Validación de cadena de certificados
- Certificados autofirmados (solo para desarrollo)
- Integración de 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 de 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: Cifrado de sistema de archivos recomendado
- Claves privadas: Permisos de archivo (0600) + cifrado de sistema de archivos
- Archivos de registro: Cifrado configurable para registros archivados
- Exportaciones de 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 (timeout 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 de base de datos
```

Recomendaciones de Control de Acceso:

```
-- Crear usuario de solo lectura para acceso de fuerzas del orden
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 Hashing

Disponibles en Erlang/OTP:

- SHA-256, SHA-384, SHA-512 (recomendados)
- 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 de 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 bits)
- 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 del 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 SMS
- Permite la interceptación legal según sea necesario

2.6.2 Consideraciones de Seguridad del Protocolo

Protocolo SMPP (Frontend Externo):

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

Protocolo IMS (Frontend Externo):

- Mensajería basada en SIP
- Mecanismos de autenticación SIP
- Integración con seguridad de red central IMS

Protocolo SS7/MAP (Frontend Externo):

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

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

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 forenses
- Exportación de métricas de Prometheus

2.7.2 Consideraciones de Evaluación de Vulnerabilidades

Limitaciones Conocidas:

- Protocolo SMS inherentemente no cifrado (por diseño, permite la interceptación legal)
- Credenciales de base de datos en archivos de configuración (debería utilizar gestión de secretos)
- Soporte de certificados autofirmados (solo para desarrollo/pruebas)

Recomendaciones de Fortalecimiento de Seguridad:

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

Pruebas de Seguridad:

- Escaneo regular de vulnerabilidades de dependencias
- Soporte para pruebas de penetración
- Validación de configuración TLS
- Auditorías de seguridad de la 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 Pseudo-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
- No transmisión de claves en texto claro

2.9 Cumplimiento y Normativas

Cumplimiento de Normas Criptográficas:

- NIST SP 800-52: Directrices TLS
- NIST SP 800-131A: Transiciones de algoritmos criptográficos
- RFC 7525: Recomendaciones TLS

- ETSI TS 133 310: Seguridad de red (para integración IMS)

Regulaciones Francesas de Criptografía:

- No criptografía restringida a la exportación (todos los algoritmos estándar)
- Declaración de medios criptográficos (si corresponde)
- Certificación de productos criptográficos de ANSSI (si es necesario)

2.10 Resistencia al Criptoanálisis

2.10.1 Principios de Diseño

Defensa Contra Criptoanálisis:

- No algoritmos criptográficos personalizados/proprietarios
- Solo algoritmos estándar de la industria, revisados por pares
- Actualizaciones regulares de seguridad 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 handshake TLS
- Detección de anomalías para fallos de cifrado
- Alertas de eventos de seguridad

3. CONTROL Y AUTORIZACIÓN DE INTERCEPCIÓN

3.1 Control de Acceso para Intercepción Legal

Autorización Administrativa:

- Se requiere acceso de administrador del sistema para 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 interceptación (integración de sistema externo)
- Listas de autorización de identificadores de objetivos (vistas de base de datos)
- Consultas con límite de 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 (típicamente 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 cuerpo del mensaje después de la entrega por privacidad
  delete_message_body_after_delivery: false, # Establecer en true
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 Transferencia para Fuerzas del Orden

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 SQL de solo lectura
- 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:

IRI (Información Relacionada con la Intercepción):

- Campos de metadatos CDR:
 - ID del mensaje
 - Números de llamada/llamados
 - Tiempos de marca (envío, entrega, expiración)
 - Estado
 - Intentos de entrega
 - Información de enrutamiento SMSC
 - Información de nodo (seguimiento de clúster)

CC (Contenido de la Comunicación):

- Cuerpo del mensaje (contenido de texto)
- Datos PDU en bruto
- Información de codificación
- Ensamblaje de mensajes multipartes

Ejemplo de Exportación:

```
# Exportación CSV para fuerzas del orden
mysql -u li_readonly -p -D sms_c -e "
```

```
SELECT
  message_id,
  calling_number,
  called_number,
  message_body,
  submission_time,
  delivery_time,
  status
FROM cdrs
WHERE (calling_number = '+33612345678' OR called_number =
'+33612345678')
  AND submission_time BETWEEN '2025-11-01' AND '2025-11-30'
ORDER BY submission_time
" --batch --silent | sed 's/\t/,/g' > interception_report.csv
```

4. SEGURIDAD E INTEGRIDAD DEL SISTEMA

4.1 Seguridad de la Aplicación

Seguridad de Elixir/Erlang:

- Aislamiento y sandboxing de VM Erlang
- Aislamiento de procesos y supervisión
- Recuperación ante fallos y tolerancia a fallos
- Sin vulnerabilidades de desbordamiento de búfer (runtime gestionado)

Gestión de Dependencias:

- Bloqueo de versiones de dependencias (mix.lock)
- Escaneo de vulnerabilidades de seguridad
- Actualizaciones regulares de dependencias
- Huella de dependencia mínima

4.2 Seguridad de la Red

Exposición de la Red:

- Puertos expuestos mínimos:
 - 8443 (API REST HTTPS)
 - 8086 (Panel de Control HTTPS)
 - 3306/5432 (Base de datos - debe estar en firewall)
- Configuración de firewall recomendada
- Lista blanca de IP para acceso de frontend
- Despliegue en DMZ para servicios expuestos a Internet

Segmentación de Red:

- Red de gestión separada
- Red de base de datos aislada
- Separación de red de puerta de enlace de frontend
- Red de comunicación de clúster (distribución Erlang)

4.3 Monitoreo y Detección de Intrusiones

Capacidades de Registro:

- Registro de aplicación estructurado
- Niveles de registro configurables
- Rotación y archivado de registros
- Soporte de integración de Syslog
- Registro centralizado (compatible con ELK stack)

Monitoreo de Eventos de Seguridad:

- Intentos de autenticación fallidos
- Patrones inusuales de mensajes
- Fallos de conexión a la base de datos
- Fallos de handshake TLS
- Anomalías en recursos del sistema

Métricas y Alertas:

- Exportación de métricas de Prometheus
- Monitoreo de rendimiento de mensajes
- Seguimiento de tasas de error
- Utilización de recursos del sistema
- Reglas de alerta personalizadas

❖ Documentación completa de monitoreo en [OPERATIONS_GUIDE.md](#) y [METRICS.md](#)

4.4 Alta Disponibilidad y Recuperación ante Desastres

Soporte de Clúster:

- Capacidad de clúster distribuido Erlang
- Replicación de Mnesia entre nodos
- Failover automático
- Descubrimiento y unión de nodos

Redundancia de Datos:

- Copias de disco de Mnesia en todos los nodos del clúster
- Replicación de base de datos SQL (nativa de MySQL/PostgreSQL)
- Procedimientos de respaldo de CDR

- Respaldo de configuración

Procedimientos de Recuperación:

- Respaldo y restauración de base de datos
 - Recuperación de tablas de Mnesia
 - Restauración de configuración
 - Procedimientos de reemplazo de nodos
-

5. REFERENCIAS DE DOCUMENTACIÓN

5.1 Manuales Técnicos

Documentación disponible en el repositorio del proyecto:

- [README.md](#) - Visión general del sistema, arquitectura y características
- [CONFIGURATION.md](#) - Referencia completa de configuración
- [API_REFERENCE.md](#) - Documentación de REST API
- [OPERATIONS_GUIDE.md](#) - Procedimientos operativos y monitoreo
- [CDR_SCHEMA.md](#) - Esquema de base de datos de Registros de Detalles de Llamadas
- [sms_routing_guide.md](#) - Configuración de enrutamiento de SMS
- [number_translation_guide.md](#) - Normalización de números
- [METRICS.md](#) - Métricas de Prometheus y monitoreo
- [PERFORMANCE_TUNING.md](#) - Optimización del rendimiento
- [TROUBLESHOOTING.md](#) - Problemas comunes y soluciones

5.2 Certificaciones de Seguridad

- **Informes de Pruebas de Penetración:** [Se proporcionará a solicitud]
- **Informes de Auditoría de Seguridad:** [Se proporcionará a solicitud]
- **Evaluaciones de Vulnerabilidades:** [Se proporcionará a solicitud]
- **Validación Criptográfica de Erlang/OTP:** Biblioteca criptográfica estándar de la industria

5.3 Documentación de Cumplimiento

- **Solicitud de Autorización ANSSI R226:** Este documento
 - **Cumplimiento de Intercepción Legal:** Según lo requerido por las regulaciones de telecomunicaciones francesas
 - **Cumplimiento de Protección de Datos:** Consideraciones de GDPR para datos de mensajes
-

6. INFORMACIÓN DE CONTACTO

Información del Proveedor/Operador:

- Nombre de la Empresa: Omnitouch Network Services Pty Ltd
- Dirección: PO BOX 296, QUINNS ROCKS WA 6030, AUSTRALIA
- Persona de Contacto: Equipo de Cumplimiento
- Correo Electrónico: compliance@omnitouch.com.au

Contacto Técnico de Seguridad:

- Nombre: Equipo de Cumplimiento
- Correo Electrónico: compliance@omnitouch.com.au

Contacto Legal/Cumplimiento:

- Nombre: Equipo de Cumplimiento
 - Correo Electrónico: compliance@omnitouch.com.au
-

APÉNDICES

Apéndice A: Flujo de Mensajes SMS con Puntos de Intercepción

A.1 Flujo de SMS Saliente



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 de patrones y reemplazo con grupos de captura
- **Filtrado de SMSC de origen:** Aplicar diferentes traducciones según el origen del mensaje
- **Evaluación basada en prioridades:** Controlar el orden de las reglas con prioridades configurables (1-255)
- **Encadenamiento de reglas:** Continuar el procesamiento a través de múltiples reglas con prevención de bucles
- **Transformaciones de llamada/llamada 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 CRUD completa 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
rule_id	entero	Identificador único autoincremental	Sí (auto)
calling_prefix	cadena/ nil	Coincidencia de prefijo para el número de llamada (nil = comodín)	No
called_prefix	cadena/ nil	Coincidencia de prefijo para el número llamado (nil = comodín)	No
source_smsc	cadena/ nil	Nombre del SMSC de origen (nil = comodín)	No
calling_match	cadena/ nil	Patrón regex para coincidir con el número de llamada	No
calling_replace	cadena/ nil	Patrón de reemplazo para el número de llamada	No
called_match	cadena/ nil	Patrón regex para coincidir con el número llamado	No
called_replace	cadena/ nil	Patrón de reemplazo para el número llamado	No
priority	entero	Prioridad de la regla (1-255, menor = mayor prioridad)	Sí
description	cadena	Descripción legible por humanos	No
enabled	booleano	Habilitar/deshabilitar regla	Sí
continue	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:

1. **Recupera reglas habilitadas** ordenadas por prioridad (más bajas primero)
2. **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)
3. **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

4. **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**

5. **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:

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

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

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 formato
},

# 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:

Formateo Específico de Puerta de Enlace

Encadenar reglas para formatear números según los requisitos específicos de la puerta de enlace:

Traducciones Específicas de SMSC

Aplicar diferentes traducciones según la fuente del mensaje:

Preparación para el Enrutamiento Basada en Prefijos

Normalizar números antes de enrutar para asegurar coincidencias de prefijos consistentes:

Manejo de Portabilidad de Números

Manejar números portados que requieren cambios de prefijo:

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 cualquiera)
2. Defina transformaciones (opcional):
 - Coincidencia y reemplazo de regex para el número de llamada
 - Coincidencia y reemplazo de regex para el número llamado
3. Establecer prioridad (1-255, menor = mayor prioridad)
4. Establecer estado:
 - **Habilitado**: La regla está activa
 - **Continuar Procesando**: Continuar evaluando más reglas después de esta
5. Agregar descripción
6. Hacer 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. Hacer clic en "Editar" junto a la regla
2. Modificar campos según sea necesario
3. Hacer clic en "Actualizar Regla"

Indicadores de la 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. Hacer 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	

Paso 2

Regla #2 (Prioridad 20)	↓ Continuar
Formatear código de área con guiones	

Llamado: +19078720155 → +1-907-8720155

000

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 hay coincidencias de 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 coincidencias de prefijos para reducir evaluaciones de regex
- Eliminar o deshabilitar reglas no utilizadas

2. Optimizar patrones regex:

- Usar coincidencias 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 de continuar

2. Hacer copias de seguridad regularmente:

- Exportar reglas antes de realizar cambios importantes
- Control de versiones de sus exportaciones
- Probar importaciones primero en no producción

3. Monitorear traducciones:

- Habilitar el 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 10 dígitos de EE. UU.: `^(\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: La regla esperada no coincide, los números pasan sin cambios

Causas posibles:

- El prefijo no coincide (verificar coincidencia exacta del 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 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 prioridad de regla y banderas de continuar

Bucle Infinito / Degradación del Rendimiento

Síntoma: La traducción tarda mucho 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 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 exacta de reglas
2. Revisar 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 se 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 de continuar

Clustering

Las tablas de Mnesia se distribuyen automáticamente entre nodos en clúster. Las reglas de traducción se replican para alta disponibilidad.

Estrategias de Migración

Al implementar nuevas reglas de traducción:

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 desnudos",
  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 el prefijo 00 a +, luego formatear para la 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 métricas de Telemetría para problemas de rendimiento



Guía de Enrutamiento SMS-C

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

Descripción General

El sistema de enrutamiento SMS-C proporciona un enrutamiento flexible y de alto rendimiento de mensajes SMS basado en múltiples criterios, incluidos prefijos de números, identificadores de SMSC, tipos de conexión y más. Las rutas se almacenan en Mnesia para persistencia y pueden modificarse en tiempo de ejecución sin interrupción del servicio.

Características Clave

- **Enrutamiento basado en prefijos:** Ruta basada en los prefijos de números llamados/llamantes con lógica de coincidencia más larga gana
- **Enrutamiento basado en SMSC:** Ruta basada en SMSC de origen o destino
- **Enrutamiento basado en tipo:** Ruta basada en el tipo de conexión de origen (IMS, Circuito Conmutado, SMPP)
- **Enrutamiento basado en prioridad:** Controla el orden de selección de rutas con prioridades configurables
- **Balanceo de carga basado en peso:** Distribuye el tráfico entre múltiples rutas utilizando pesos
- **Enrutamiento de respuesta automática:** Envía automáticamente respuestas de vuelta a los originadores de mensajes
- **Enrutamiento de descarte:** Descarta mensajes que coinciden con criterios específicos (filtrado de spam, etc.)
- **Control de carga:** Configura el comportamiento de carga por ruta (Sí/No/Predeterminado)
- **Carga de archivo de configuración:** Carga rutas iniciales desde `runtime.exs` en el primer inicio
- **Configuración en tiempo de ejecución:** Agrega, modifica o desactiva rutas sin reiniciar
- **Interfaz Web:** Interfaz completa CRUD para la gestión de rutas con menú desplegable en el frontend
- **Herramienta de simulación:** Prueba la lógica de enrutamiento antes de la implementación
- **Copia de seguridad/Restauración:** Exporta e importa configuraciones de enrutamiento
- **Soporte ENUM:** Búsqueda de números basada en DNS (para implementación futura)

Arquitectura

Modelo de Datos

Cada ruta contiene los siguientes campos:

Campo	Tipo	Descripción	Requerido
route_id	entero	Identificador único autoincremental	Sí (auto)
calling_prefix	cadena/ nil	Coincidencia de prefijo para el número de llamada (nil = comodín)	No
called_prefix	cadena/ nil	Coincidencia de prefijo para el número llamado (nil = comodín)	No
source_smsc	cadena/ nil	Nombre del SMSC de origen (nil = comodín)	No
dest_smsc	cadena/ nil	Nombre del SMSC de destino (requerido a menos que auto_reply o drop sea verdadero)	Condicional
source_type	átomo/ nil	Tipo de origen: :ims, :circuit_switched, :smpp, o nil	No
enum_domain	cadena/ nil	Dominio DNS ENUM para búsqueda	No
auto_reply	booleano	Si es verdadero, envía respuesta de vuelta al originador	No (predeterminado: falso)
auto_reply_message	cadena/ nil	Texto del mensaje para respuesta automática (requerido si auto_reply es verdadero)	Condicional
drop	booleano	Si es verdadero, descarta el mensaje (filtrado de spam)	No (predeterminado: falso)
charged	átomo	Comportamiento de carga: :yes, :no, o :default	No (predeterminado: :default)
weight	entero	Peso de balanceo de carga (1-100, predeterminado 100)	Sí
priority	entero	Prioridad de ruta (1-255, menor = mayor prioridad)	Sí
description	cadena	Descripción legible por humanos	No
enabled	booleano	Habilitar/deshabilitar ruta	Sí

Nota: Una ruta debe ser uno de tres tipos:

1. **Enrutamiento normal:** auto_reply=false, drop=false, requiere dest_smsc

2. **Respuesta automática:** auto_reply=true, requiere auto_reply_message
3. **Descarte:** drop=true, descarta el mensaje

Algoritmo de Enrutamiento

Al enrutar un mensaje, el sistema sigue este orden de prioridad:

PRIORIDAD 1: Enrutamiento Basado en Ubicación (Más Alto)

1. **Verificar registro de suscriptor:** Si el MSISDN de destino está registrado en la tabla de ubicaciones
2. **Ruta directamente al frontend que sirve:** Salta todas las reglas de enrutamiento y envía directamente al frontend que atiende a ese suscriptor
3. **Esto sucede DESPUÉS de la traducción de números** para asegurar consistencia con los registros de ubicación

PRIORIDAD 2: Reglas de Enrutamiento Estándar (si no se encuentra registro de ubicación)

1. **Filtra rutas habilitadas** que coinciden con TODOS los criterios especificados
2. **Ordena por especificidad** (rutas más específicas primero):
 - Prefijo llamado más largo = mayor especificidad (×100 puntos)
 - Prefijo de llamada más largo = especificidad media (×50 puntos)
 - SMSC de origen especificado = +25 puntos
 - Dominio de resultado ENUM especificado = +15 puntos
 - Tipo de origen especificado = +10 puntos
 - Dominio ENUM especificado = +5 puntos
3. **Agrupar por prioridad** (número menor = mayor prioridad)
4. **Selecciona del grupo de mayor prioridad** utilizando selección aleatoria ponderada
5. **Ejecuta acción de ruta:**
 - **Ruta normal:** Devuelve SMSC de destino para la entrega del mensaje
 - **Ruta de respuesta automática:** Envía respuesta de vuelta al originador de forma asíncrona
 - **Ruta de descarte:** Descarta el mensaje y registra el evento

Comodines

- nil o valores vacíos actúan como comodines que coinciden con cualquier valor
- Una ruta sin criterios especificados es una ruta de captura.

Configuración

Carga de Rutas desde Archivo de Configuración

Las rutas pueden definirse en `config/runtime.exs` y se cargarán automáticamente en el primer inicio. Esto es útil para definir reglas de enrutamiento base que deben estar presentes cuando el sistema se inicia por primera vez.

Importante: Las rutas de configuración solo se cargan cuando la tabla de enrutamiento está **vacía** (primer inicio). Esto preserva las rutas agregadas a través de la interfaz web durante el tiempo de ejecución y previene duplicados en reinicios.

Flujo de Carga de Configuración

Ejemplo de Estructura de Configuración de Ruta

Consulte `config/runtime.exs` y `config/sms_routes.example.exs` para ejemplos completos que incluyen:

- Enrutamiento geográfico
- Rutas de respuesta automática
- Rutas de descarte (filtrado de spam)
- Rutas balanceadas por carga
- Enrutamiento de números premium con carga

Comenzando

Flujo de Inicialización

Resumen de Tipos de Ruta

Flujo de Enrutamiento de Mensajes

Casos de Uso Comunes

Enrutamiento Basado en Ubicación (Prioridad Más Alta)

Ruta mensajes directamente al frontend que atiende a un suscriptor registrado, eludiendo todas las reglas de enrutamiento:

Cómo funciona:

1. El mensaje llega con el número de destino

2. Los números se traducen (si está configurado)
3. El sistema verifica si el MSISDN de destino traducido está en la tabla de ubicaciones
4. Si está registrado, el mensaje se enruta directamente al frontend que atiende a ese suscriptor
5. Las reglas de enrutamiento estándar se **saltan completamente**
6. Si no está registrado, se aplican las reglas de enrutamiento normales

Beneficios:

- **Entrega garantizada** al frontend correcto para suscriptores registrados
- **Enrutamiento más rápido** - no se necesita evaluación de tabla de rutas
- **Enrutamiento preciso** - la ubicación del suscriptor es la fuente de verdad
- **Anula todas las reglas de enrutamiento** - asegura la disponibilidad del suscriptor

Casos de uso:

- Suscriptores IMS/VoLTE registrados en núcleos IMS específicos
- Suscriptores móviles conectados a MSCs específicos
- Suscriptores SIP registrados en servidores de aplicaciones específicos

Enrutamiento Geográfico

Ruta mensajes a SMSCs regionales según el país de destino:

Balanceo de Carga

Distribuir tráfico entre múltiples SMSCs con pesos:

Enrutamiento de Números Premium

Ruta números premium a un manejo especial con prioridad:

Enrutamiento Específico de Protocolo

Ruta basada en el tipo de conexión de origen:

Migración de Red

Durante la migración, enrutar prefijos específicos a la nueva infraestructura:

Enrutamiento Complejo Multi-Criterio

Combinar múltiples criterios para un control detallado:

Interfaz Web

UI de Gestión de Rutas

Acceda a la interfaz de gestión de rutas en `/sms_routing` (configúrelo en su enrutador):

Características:

- Ver todas las rutas en una tabla ordenable
- Agregar nuevas rutas con validación de formulario
- Editar rutas existentes
- Habilitar/deshabilitar rutas sin eliminar
- Eliminar rutas con confirmación
- Actualizaciones en tiempo real (refresco cada 5 segundos)

Agregar una Ruta:

1. Haga clic en "Agregar Nueva Ruta"
2. Complete los campos del formulario (solo se requiere SMSC de destino)
3. Establezca el peso (1-100, predeterminado 100) y la prioridad (1-255, predeterminado 100)
4. Marque "Habilitado" para activar de inmediato
5. Haga clic en "Guardar Ruta"

Editar una Ruta:

1. Haga clic en "Editar" junto a la ruta
2. Modifique los campos según sea necesario
3. Haga clic en "Guardar Ruta"

Deshabilitar una Ruta:

- Haga clic en "Deshabilitar" para desactivar temporalmente sin eliminar
- Haga clic en "Habilitar" para reactivar

Simulador de Enrutamiento

Acceda al simulador en `/simulator` (a través del menú de navegación):

Características:

- Pruebe la lógica de enrutamiento con varios parámetros
- **Evaluación detallada campo por campo** mostrando por qué cada ruta coincidió o no coincidió
- Vea todas las rutas evaluadas en orden de prioridad
- Indicadores visuales para rutas coincidentes/seleccionadas
- Cargue escenarios de ejemplo para pruebas rápidas

- Vea el 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)
 - Tipo de origen (Cualquiera/IMS/Circuito Conmutado/SMPP)
2. Haga clic en "Simular Enrutamiento"
3. Vea resultados completos:
 - **Resultado de Enrutamiento:** Ruta seleccionada y destino (o "No se encontró ruta")
 - **Evaluación de Ruta:** Todas las rutas con análisis campo por campo:
 - ✓ Marca de verificación verde = Campo coincidente
 - ✗ X roja = Campo no coincidente
 - Razón para cada coincidencia/no coincidencia de campo
 - **Indicadores visuales:**
 - Borde verde + insignia "SELECCIONADO" = Ruta realmente utilizada
 - Borde púrpura + insignia "COINCIDIDO" = Rutas que coincidieron pero no fueron seleccionadas
 - Borde gris = Rutas que no coincidieron
4. Cargue ejemplos preconfigurados utilizando los botones de ejemplo
5. Revise el historial de pruebas para comparar diferentes escenarios

Ejemplo de Salida de Evaluación: Para cada ruta, verá por qué coincidió o no:

- **Prefijo de llamada:** "Coincide con el prefijo '1234'" o "No comienza con '44'"
- **Prefijo llamado:** "Comodín (coincide con cualquier)" o "No comienza con '639'"
- **SMSC de origen:** "Coincide con 'smc1'" o "Se esperaba 'untrusted_smc', se obtuvo 'none'"
- **Tipo de origen:** "Comodín (coincide con cualquier)" o "Se esperaba 'smpp', se obtuvo 'IMS'"

Referencia de API

Resumen de Operaciones Principales

Operaciones de Gestión de Rutas

Parámetros de Enrutamiento de Mensajes

route_message 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
- `source_type` (opcional): Tipo de conexión (:ims, :circuit_switched, :smpp)
- `message_id` (opcional): Para registro de eventos

Devuelve:

- `{:ok, dest_smsc, route}` - Ruta encontrada y seleccionada
- `{:error, :no_route_found}` - Sin ruta coincidente

Operaciones de Importación/Exportación

Mejores Prácticas

Diseño de Rutas

1. **Usar prioridades sabiamente:** Reserve prioridades bajas (1-10) para rutas críticas
2. **Mantenerlo simple:** Comience con rutas amplias y agregue específicas según sea necesario
3. **Documentar rutas:** Siempre agregue descripciones a las rutas
4. **Usar captura todo:** Siempre tenga una ruta predeterminada con baja prioridad

Rendimiento

1. **Minimizar el conteo de rutas:** Combine rutas similares siempre que sea posible
2. **Usar prefijos más largos:** Prefijos más específicos reducen el tiempo de evaluación
3. **Deshabilitar rutas no utilizadas:** No elimine rutas que pueda necesitar más adelante; desactívelas

Operaciones

1. **Probar antes de implementar:** Use el simulador para verificar la lógica de enrutamiento
2. **Hacer copias de seguridad regularmente:** Exportar rutas antes de realizar cambios importantes
3. **Monitorear enrutamiento:** Verifique los registros de eventos para decisiones de enrutamiento
4. **Implementación gradual:** Use pesos para cambiar gradualmente el tráfico a nuevas rutas

Pruebas

1. **Escribir pruebas de integración:** Pruebe sus escenarios de enrutamiento específicos
2. **Prueba de carga:** Verifique el rendimiento de enrutamiento bajo carga
3. **Pruebas de conmutación por error:** Asegúrese de que las rutas de respaldo funcionen cuando las primarias fallen

Solución de Problemas

No se Encontró Ruta

Síntoma: { :error, :no_route_found } devuelto

Causas posibles:

- No hay rutas configuradas
- Todas las rutas coincidentes están deshabilitadas
- Los criterios de ruta no coinciden con los parámetros del mensaje
- El prefijo no coincide (verifique errores tipográficos)

Soluciones:

1. Verifique que existan rutas: `SmsRouting.list_enabled_routes()`
2. Use el simulador para probar el enrutamiento con parámetros de mensaje reales
3. Agregue una ruta de captura para depuración: `add_route(%{dest_smsc: "debug_smsc", priority: 255})`
4. Verifique los registros de eventos para detalles de evaluación de enrutamiento

Ruta Incorrecta Seleccionada

Síntoma: Mensaje enrutado a un destino inesperado

Causas posibles:

- Configuración incorrecta de prioridad
- Ruta comodín tiene mayor prioridad
- Cálculo de especificidad favorece una ruta diferente
- Múltiples rutas con los mismos criterios utilizando pesos

Soluciones:

1. Use el simulador para ver todas las rutas coincidentes
2. Verifique los valores de prioridad (menor = mayor prioridad)
3. Verifique las puntuaciones de especificidad en el simulador

4. Revise la distribución de pesos para rutas balanceadas por carga

Problemas de Rendimiento

Síntoma: El enrutamiento es lento

Causas posibles:

- Demasiadas rutas en la base de datos
- Patrones de ruta complejos
- Tabla Mnesia no indexada correctamente

Soluciones:

1. Consolidar rutas similares
2. Eliminar rutas deshabilitadas que ya no se necesitan
3. Asegúrese de que se creen índices de Mnesia (automático en `init_tables`)
4. Considere almacenar en caché decisiones de enrutamiento utilizadas con frecuencia

Temas Avanzados

Integración ENUM/NAPTR

ENUM (Mapeo de Números E.164) proporciona búsqueda de números basada en DNS utilizando registros NAPTR. El SMS-C incluye soporte completo para ENUM con almacenamiento en caché, servidores DNS configurables y coincidencia de rutas basada en resultados de búsqueda ENUM.

¿Qué es ENUM?

ENUM mapea números de teléfono E.164 a nombres DNS utilizando una transformación simple:

- **Número de Teléfono:** +1-212-555-1234
- **Consulta ENUM:** 4.3.2.1.5.5.5.2.1.2.1.e164.arpa
- **Tipo de Registro DNS:** NAPTR (Puntero de Autoridad de Nombres)
- **Resultado:** URI SIP, información de enrutamiento u otros datos de servicio

Configuración

La funcionalidad ENUM se configura en `config/runtime.exs`:

Habilitar Búsquedas ENUM:

Establezca `enum_enabled: true` para habilitar las búsquedas ENUM antes del enrutamiento. Cuando está habilitado, el sistema realizará búsquedas DNS ENUM

para mensajes entrantes y utilizará los resultados en las decisiones de enrutamiento.

Dominios ENUM:

Enumere los dominios ENUM a consultar en orden de prioridad. El sistema intentará cada dominio hasta que ocurra una búsqueda exitosa.

Dominios ENUM comunes:

- e164.arpa - Dominio ENUM oficial de IETF
- e164.org - Registro ENUM alternativo
- Dominios ENUM privados personalizados

Servidores DNS:

Configure servidores DNS específicos para consultas ENUM. Formato: {ip_address, port}

Deje vacío o establezca en [] para utilizar los servidores DNS predeterminados del sistema.

Ejemplo de configuración DNS personalizada:

- DNS Público de Google: {"8.8.8.8", 53}, {"8.8.4.4", 53}
- DNS de Cloudflare: {"1.1.1.1", 53}, {"1.0.0.1", 53}
- DNS ENUM personalizado: {"10.0.0.53", 53}

Tiempo de Espera:

Establezca el tiempo de espera de consulta DNS en milisegundos (predeterminado: 5000 ms). Aumente para redes lentas, disminuya para una conmutación por error más rápida.

Cómo Funcionan las Búsquedas ENUM

Almacenamiento en Caché ENUM

El sistema almacena en caché los resultados de búsqueda ENUM durante 15 minutos para mejorar el rendimiento y reducir la carga de DNS.

Beneficios de la Caché:

- Reduce la carga de consultas DNS
- Mejora la latencia de enrutamiento
- Protege contra fallos del servidor DNS (los resultados en caché permanecen disponibles)

Estadísticas de Caché:

- Vea el tamaño y estado de la caché en la página de Prueba NAPTR
- Monitoree tasas de aciertos/fallos de caché a través de métricas de Prometheus
- Borre la caché manualmente si es necesario (cambios de configuración, pruebas, etc.)

Comportamiento de la Caché:

- Tanto las búsquedas exitosas como las fallidas se almacenan en caché
- Búsquedas fallidas almacenadas en caché para evitar consultas repetidas para números inválidos
- La caché expira automáticamente después de 15 minutos
- La caché sobrevive a los reinicios de la aplicación (almacenada en ETS)

Usando ENUM en Rutas

Las rutas pueden coincidir con los resultados de búsqueda ENUM utilizando el campo `enum_result_domain`:

Escenario de Ejemplo:

La búsqueda ENUM para +1-555-0100 devuelve el registro NAPTR:

- Servicio: E2U+sip
- Reemplazo: sip:customer@voip-carrier.com
- **Dominio de Resultado:** voip-carrier.com

Configuración de Ruta:

Cree una ruta con `enum_result_domain`: "voip-carrier.com" para coincidir con mensajes donde la búsqueda ENUM devolvió este dominio.

Lógica de Coincidencia:

- Si la ruta tiene `enum_result_domain`: nil - coincide con todos los mensajes (comodín)
- Si la ruta tiene `enum_result_domain`: "specific.com" - solo coincide si ENUM devolvió ese dominio
- Las rutas con dominios ENUM coincidentes reciben puntuaciones de especificidad más altas

Cálculo de Prioridad:

Las rutas con dominios de resultado ENUM reciben +15 puntos de especificidad, priorizándolas sobre rutas genéricas.

Pruebas de Búsquedas ENUM

Acceda a la página de Prueba NAPTR en /naptr_test (a través del menú de navegación).

Características:

- Realice búsquedas ENUM en vivo contra servidores DNS configurados
- Vea información detallada del registro NAPTR
- Vea dominios de resultado extraídos de registros NAPTR
- Monitoree estadísticas de caché
- Borre la caché para pruebas

Flujo de Trabajo de Prueba:

1. Ingrese un número de teléfono (con o sin prefijo +)
2. Especifique el dominio ENUM (predeterminado: e164.arpa)
3. Haga clic en "Realizar Búsqueda"
4. Revise los resultados:
 - Registros NAPTR encontrados
 - Valores de orden y preferencia
 - Tipos de servicio (E2U+sip, E2U+tel, etc.)
 - Expresiones regulares
 - Valores de reemplazo
 - **Dominios de resultado extraídos** (utilizados para coincidencia de rutas)

Visualización de la Configuración Actual:

- Servidores DNS utilizados (o "Predeterminado del Sistema")
- Configuración de tiempo de espera
- Tamaño y estado de la caché
- Botón para borrar caché

Comprendiendo los Resultados:

Cada registro NAPTR contiene:

- **Orden:** Prioridad para el procesamiento (menor primero)
- **Preferencia:** Dentro del mismo orden (menor primero)
- **Banderas:** Instrucciones de procesamiento (u=terminal, s=continuar)
- **Servicio:** Tipo de servicio (E2U+sip, E2U+tel, etc.)
- **Regexp:** Expresión de sustitución
- **Reemplazo:** Dominio o dirección alternativa
- **Dominio de Resultado:** Dominio extraído para coincidencia de rutas

Casos de Uso Comunes de ENUM

1. Peering VoIP

Utilice ENUM para identificar números alojados en redes SIP/VoIP y enrutar directamente a puertas de enlace VoIP:

- ENUM devuelve URI SIP: sip:number@voip-carrier.com
- Dominio de resultado: voip-carrier.com
- Ruta con enum_result_domain: "voip-carrier.com" seleccionada
- Tráfico enviado a la puerta de enlace de peering VoIP directa

2. Identificación de Proveedor

Identifique el proveedor que atiende un número y enrute en consecuencia:

- ENUM devuelve información del proveedor
- Dominio de resultado: carrier-a.com
- Ruta a la interconexión del proveedor A
- Optimizar costos y calidad de enrutamiento

3. Portabilidad de Números

Maneje números portados que se trasladaron entre proveedores:

- La búsqueda ENUM devuelve el proveedor actual
- Ruta automáticamente al destino correcto
- No se necesitan actualizaciones manuales de la tabla de enrutamiento

4. Enrutamiento de Menor Costo

Combine ENUM con múltiples rutas:

- ENUM identifica la red de destino
- Múltiples rutas para el mismo dominio con diferentes costos
- Use prioridad y pesos para preferir rutas de menor costo

5. Servicios de Emergencia

Enrute números de emergencia (911, 112, etc.) a los servicios de emergencia adecuados:

- La búsqueda ENUM identifica la puerta de enlace de emergencia local
- Ruta de alta prioridad asegura enrutamiento inmediato
- Sin demora por evaluación de ruta normal

Estrategia de Enrutamiento ENUM

Configuración Recomendada:

1. Rutas ENUM de Alta Prioridad (Prioridad 1-10)

- Rutas que coinciden con dominios de resultado ENUM específicos
- Utilizadas para peering directo, enrutamiento VoIP
- Mayor especificidad, seleccionadas primero

2. Rutas de Prefijo de Prioridad Media (Prioridad 50-100)

- Enrutamiento basado en prefijos estándar
- Utilizadas cuando la búsqueda ENUM falla o no devuelve registros
- Respaldo confiable

3. Captura Todo de Baja Prioridad (Prioridad 200+)

- Ruta predeterminada para todo lo demás
- Asegura que ningún mensaje quede sin enrutamiento

Ejemplo de Jerarquía de Rutas:

- Prioridad 1: enum_result_domain: "sip.carrier.com" → Puerta de enlace VoIP directa
- Prioridad 10: enum_result_domain: "tel.carrier.com" → Puerta de enlace PSTN del proveedor
- Prioridad 50: called_prefix: "+1" → Puerta de enlace predeterminada de América del Norte
- Prioridad 100: called_prefix: "+" → Puerta de enlace predeterminada internacional
- Prioridad 200: Sin criterios → Respaldo definitivo

Consideraciones de Rendimiento

Latencia de Consulta DNS:

Las búsquedas ENUM añaden tiempo de consulta DNS al enrutamiento:

- **En caché:** < 1 ms (rápido)
- **Sin caché:** 10-100 ms (depende del servidor DNS)

Recomendaciones:

- Utilice servidores DNS geográficamente cercanos
- Configure un tiempo de espera apropiado (5000 ms predeterminado)
- Monitoree tasas de aciertos de caché (objetivo > 80%)
- Considere calentar la caché para números conocidos

Escalabilidad:

El sistema de caché maneja escenarios de alto volumen:

- La caché se comparte entre todos los procesos
- Tabla ETS de lectura concurrente para rendimiento
- Limpieza automática de caché a través de TTL
- Escala a millones de entradas en caché

Manejo de Fallos:

Los fallos de ENUM caen de manera elegante en el enrutamiento regular:

- Tiempo de espera DNS → Pasar a la siguiente ruta
- Sin registros NAPTR → Usar rutas basadas en prefijos
- Formato NAPTR inválido → Registrar error, continuar enrutando
- Servidor DNS no disponible → Usar resultados en caché o respaldo

Monitoreo de Operaciones ENUM

Utilice métricas de Prometheus para monitorear el rendimiento ENUM:

- `sms_c_enum_lookup_stop_duration` - Latencia de búsqueda
- `sms_c_enum_cache_hit_count` - Aciertos de caché
- `sms_c_enum_cache_miss_count` - Fallos de caché
- `sms_c_enum_cache_size_size` - Tamaño actual de la caché
- `sms_c_enum_naptr_records_record_count` - Registros NAPTR por búsqueda

Métricas Clave a Monitorear:

- **Tasa de aciertos de caché:** Debe ser > 70% después del calentamiento
- **Duración de búsqueda p95:** Debe ser < 1000 ms
- **Búsquedas fallidas:** Monitorear problemas de DNS

Consulte `docs/METRICS.md` para la documentación completa de métricas.

Solución de Problemas ENUM

Problema: No se Encontraron Registros NAPTR

- Verifique la configuración del dominio ENUM
- Pruebe la conectividad del servidor DNS
- Verifique si el número está realmente en el registro ENUM
- Intente con un dominio ENUM alternativo (por ejemplo, e164.org)
- Utilice la página de Prueba NAPTR para diagnosticar

Problema: Búsquedas ENUM Lentas

- Verifique la latencia del servidor DNS
- Verifique la conectividad de red
- Aumente el tiempo de espera si es necesario
- Considere utilizar servidores DNS más cercanos
- Verifique la tasa de aciertos de caché

Problema: Ruta Incorrecta Seleccionada Después de ENUM

- Verifique el campo `enum_result_domain` en las rutas
- Use el Simulador de Rutas para probar la lógica de enrutamiento
- Verifique que la extracción del dominio de resultado sea correcta
- Revise el formato del registro NAPTR en la página de Prueba

Problema: Búsquedas ENUM Deshabilitadas

- Verifique `enum_enabled: true` en `config/runtime.exs`
- Verifique que la lista de `enum_domains` no esté vacía
- Reinicie la aplicación después de cambios en la configuración
- Verifique los registros de la aplicación para la inicialización de ENUM

Consideraciones de Seguridad

Envenenamiento de Caché DNS:

- Utilice solo servidores DNS de confianza
- Considere DNSSEC si está disponible
- Valide formatos de registros NAPTR
- Monitoree dominios de resultado inesperados

Agotamiento de Recursos:

- Límites de caché evitan el agotamiento de memoria
- Tiempo de espera evita quedarse atascado en DNS lentos
- Búsquedas fallidas almacenadas en caché para evitar tormentas de reintentos

Divulgación de Información:

- Las búsquedas ENUM revelan intenciones de enrutamiento a los servidores DNS
- Utilice servidores DNS privados para enrutamiento sensible
- Considere VPN/DNS encriptado para privacidad

Registro de Eventos

Las decisiones de enrutamiento se registran a través del EventLogger:

- `sms_routing_started`: La evaluación de enrutamiento comienza

- `sms_routing_candidates`: Número de rutas habilitadas encontradas
- `sms_routing_matches`: Número de rutas coincidentes
- `sms_routing_selected`: Detalles de la ruta seleccionada
- `sms_routing_failed`: No se encontró ruta

Habilite el registro pasando `message_id` a `route_message/1`.

Clustering

Las tablas Mnesia se distribuyen automáticamente entre nodos en clúster. Las rutas se replican para alta disponibilidad.

Ejemplos

Consulte la suite de pruebas en `test/sms_c/messaging/sms_routing_test.exs` para ejemplos completos de:

- Coincidencia de prefijos
- Enrutamiento basado en prioridades
- Balanceo de carga basado en peso
- Enrutamiento multi-criterio
- Casos límite

Migración desde el Antiguo Enrutamiento

Si se migra desde el antiguo enrutamiento basado en configuración, siga este proceso:

Detalle de Pasos de Migración

1. Inicializar Tablas

- Crea tablas de enrutamiento Mnesia
- Prepara el sistema para el nuevo enrutamiento

2. Analizar Rutas Antiguas

- **Patrones Regex** → Rutas basadas en prefijos
- **Respuestas enlatadas** → Rutas de respuesta automática
- **Lógica personalizada** → Rutas multi-criterio

3. Probar a Fondo

- Utilice el simulador de enrutamiento
- Verifique todos los escenarios
- Verifique casos límite

4. **Actualizar Código**

- Reemplace las llamadas de enrutamiento antiguas
- Use la API `route_message/1`
- Actualice el manejo de errores

5. **Desplegar y Monitorear**

- Despliegue el nuevo sistema de enrutamiento
- Monitoree problemas
- Mantenga la configuración antigua como respaldo inicialmente

6. **Limpiar**

- Elimine la configuración de enrutamiento antigua
- Elimine el código de migración
- Actualice la documentación

Soporte

Para problemas o preguntas:

- Consulte la suite de pruebas para ejemplos
- Use el simulador para depurar la lógica de enrutamiento
- Revise los registros de eventos para decisiones de enrutamiento
- Verifique el contenido de la tabla Mnesia:
`:mnesia.table_info(:sms_route, :size)`