# Referencias de rendimiento

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

## Referencias disponibles

### 1. Referencia de SMS en bruto (`raw_sms_bench.exs`)

Mide el rendimiento del endpoint 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 dactilares 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 la API de mensajes, incluyendo inserción, recuperación y enrutamiento.

**Características:**

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

**Uso:**

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

**Salida:** `benchmarks/output/message_api_benchmark.html`

# Configuración

Todas las referencias utilizan Benchee con la siguiente configuración predeterminada:

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

# Salidas

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

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

# Referencia de la API SMS-C

**← Volver al Índice de Documentación**

Referencia para la API REST SMS-C, generada a partir de la tabla de rutas en vivo (`config/config.exs`) y las implementaciones de los controladores. Cada punto final a continuación es realmente enrutado por el servicio en ejecución.

La aplicación en sí sirve una especificación OpenAPI legible por máquina y una interfaz Swagger UI interactiva. Estas siempre están sincronizadas con la construcción desplegada:

```
GET /api/schema      # Especificación JSON OpenAPI
GET /api/docs        # Swagger UI
```

## Tabla de Contenidos


- Descripción General
- Autenticación
- Convenciones de Respuesta
- Estado
- Mensajes
- Indicación de Mensaje en Espera
- Ubicaciones
- Frontends
- Eventos
- Campañas
- Otras Familias de Puntos Finales
- Códigos de Estado HTTP
- Formato de Error
- Mejores Prácticas

# Descripción General

## URL Base

```
https://<host>:8443/api
```

**Puerto por Defecto**: 8443 (configurable a través de `config :api_ex, api: %{port: ...}`) **Protocolo**: HTTPS cuando `enable_tls` está configurado (el valor por defecto). Todas las rutas están montadas bajo el prefijo `/api`.

## Tipo de Contenido

Los cuerpos de solicitud y respuesta son JSON:

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

No hay versionado de API basado en URL. La tabla de rutas es el contrato; consulte `/api/schema` para la forma exacta y desplegada.

# Autenticación

La API **no tiene autenticación a nivel de aplicación**. La tubería de solicitud solo negocia el tipo de contenido y adjunta la especificación OpenAPI; no hay clave de API, token de portador o verificación de sesión frente a las rutas.

Por lo tanto, se espera que el control de acceso se maneje fuera de la aplicación:

- **TLS**: cuando `enable_tls` es verdadero, el oyente termina HTTPS utilizando el `tls_cert_path` / `tls_key_path` configurado. Úselo en producción.
- **Controles de red**: restrinja el puerto de la API a fuentes de confianza con un firewall o un proxy inverso ascendente. Varios puntos finales registran la IP del llamador (de `X-Forwarded-For` o el socket) para auditoría, pero no se bloquean en ello.

No exponga el puerto de la API directamente a redes no confiables.

# Convenciones de Respuesta

Los controladores devuelven el recurso **directamente**, no envuelto en un sobre `data`.

Recurso único:

```
{ "id": 12345, "status": "queued" }
```

Las colecciones se devuelven como un arreglo JSON desnudo:

```
[ { "id": 12345 }, { "id": 12346 } ]
```

Los errores a nivel de controlador utilizan una cadena `error` plana:

```
{ "error": "payload_hex is required. For plaintext, use POST
/api/messages_plaintext" }
```

Los errores a nivel de marco (fallos de enrutamiento, cuerpos no analizables) regresan como:

```
{ "status": "error", "response": "Not Found" }
```

# Estado

Verificación de salud para monitoreo y balanceadores de carga.

**Solicitud**:

```
GET /api/status
```

**Respuesta** (200 OK):

```
{
  "status": "ok",
  "application": "OmniMessage",
  "timestamp": "2025-11-01T23:00:00Z"
}
```

**Ejemplo**:

```
curl https://host:8443/api/status
```

# Mensajes

La cola de mensajes es el recurso central. El contenido del mensaje se transporta como **`payload_hex`** (los octetos TP-User-Data) más un byte de **`dcs`** (esquema de codificación de datos) y un opcional **`udh_hex`** (encabezado de datos del usuario). Los llamadores de texto plano deben usar Enviar Texto Plano, que realiza la codificación, selección de DCS y división en múltiples partes por usted.

## Objeto Mensaje

Campos devueltos para un mensaje (ver `ApiEx.Schemas.MessageSchema`):

| Campo | Tipo | Notas |
|---|---|---|
| `id` | entero | ID del mensaje |
| `source_msisdn` | cadena | Número de origen |
| `destination_msisdn` | cadena | Número de destino |
| `payload_hex` | cadena | Octetos TP-User-Data como hex (contenido canónico) |
| `dcs` | cadena | Esquema de codificación de datos como un byte hex (`00`=GSM7, `08`=UCS2, `04`=8bit, `C8`=MWI) |
| `udh_hex` | cadena / nulo | Octetos del encabezado de datos del usuario como hex |
| `tp_pid` | cadena / nulo | TP-Protocol-Identifier como un byte hex. Por defecto es `00` (SME-to-SME). `7F` = (U)SIM Data Download, que un mensaje de descarga de datos SMS-PP (SIM OTA) debe llevar, junto con un `dcs` de clase-2, antes de que el dispositivo lo pase al UICC (3GPP TS 23.048 / TS 31.111 cláusula 7.1.1) |
| `message_text` | cadena / nulo | Texto UTF-8 derivado solo para visualización (no se utiliza para la entrega) |
| `source_smsc` | cadena | Identificador de SMSC de origen |
| `dest_smsc` | cadena / nulo | Identificador de SMSC de destino |

| Campo | Tipo | Notas |
|---|---|---|
| `status` | cadena | Uno de `queued`, `backoff`, `sent`, `delivered`, `expired`, `dropped`, `auto_replied`, `balance_rejected`, `deleted` |
| `send_time` | datetime | |
| `deliver_time` | datetime / nulo | |
| `expires` | datetime | |
| `delivery_attempts` | entero | |
| `deadletter` | booleano | |
| `message_parts` | entero / nulo | |
| `message_part_number` | entero / nulo | |
| `smsc_node_name` | cadena | Nodo que posee el mensaje |

# Listar Mensajes

**Solicitud**:

```
GET /api/messages
```

**Encabezados Opcionales**:

- `smsc: <frontend_name>` - modo de sondeo: devolver mensajes destinados a este SMSC.

- `include-unrouted: true|1` - en modo de sondeo, también devolver mensajes sin una registro de ubicación (por defecto: falso).
- `x-decode: true|1` - incluir una vista decodificada de cada carga útil del mensaje.

**Parámetros de Consulta** (modo de lista, utilizado solo cuando no hay encabezado `smsc` presente):

- `status` - filtrar por un solo átomo de estado (ver la lista de estados arriba).
- `limit` - registros máximos (por defecto 100, máximo 1000).
- `offset` - registros a omitir (por defecto 0).

El punto final está **siempre paginado** en modo de lista: sin `limit` devuelve como máximo 100 registros, nunca toda la tienda.

**Respuesta** (200 OK) - un arreglo JSON de objetos de mensaje.

**Ejemplos**:

```
# Sondear los mensajes de un frontend
curl -H "smsc: uk_gateway" https://host:8443/api/messages

# Sondear incluyendo mensajes no enrutados
curl -H "smsc: uk_gateway" -H "include-unrouted: true"
https://host:8443/api/messages

# Listar mensajes entregados
curl "https://host:8443/api/messages?status=delivered&limit=50"
```

# Obtener Mensaje

**Solicitud**:

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

El encabezado opcional `x-decode: true` agrega una vista de carga útil decodificada.

**Respuesta** (200 OK): un solo objeto de mensaje. **404** con `{"error": "Message not found"}` si el id es desconocido.

```
curl https://host:8443/api/messages/12345
```

## Enviar Mensaje

Enviar un mensaje pre-codificado. **`payload_hex` es requerido**; las solicitudes sin él son rechazadas con 422.

**Solicitud**:

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

**Cuerpo**:

```
{
  "source_msisdn": "15555551234",
  "destination_msisdn": "15555555678",
  "payload_hex": "c8329bfd06dddf723619",
  "dcs": "00",
  "source_smsc": "api_client"
}
```

**Campos**:

- `source_msisdn`, `destination_msisdn` - números.
- `payload_hex` - **requerido**. Octetos TP-User-Data como hex.
- `dcs` - byte hex del esquema de codificación de datos (valores por defecto aplicados a continuación).
- `udh_hex` - encabezado de datos del usuario hex opcional.
- `tp_pid` - byte hex opcional del TP-Protocol-Identifier (valor por defecto `00`). Establecer `7F` con un `dcs` de clase-2 (por ejemplo, `F6`) para descarga de datos SMS-PP / SIM OTA; sin él, el dispositivo almacena el paquete asegurado en lugar de entregarlo al UICC y no se recibe prueba de entrega.

- `source_smsc` - identifica el sistema que envía.
- `source_type` - clasificación de origen opcional.

**Respuesta** (201 Creado): el objeto de mensaje creado. Los rechazos devuelven 422 con una cadena `error` (por ejemplo, falta `payload_hex`).

```
curl -X POST https://host:8443/api/messages \
  -H "Content-Type: application/json" \
  -d '{
    "source_msisdn": "15555551234",
    "destination_msisdn": "15555555678",
    "payload_hex": "c8329bfd06dddf723619",
    "dcs": "00",
    "source_smsc": "api_client"
  }'
```

## Enviar Texto Plano

Enviar una cadena UTF-8 y dejar que el SMS-C maneje la codificación (GSM7 vs UCS2), selección de DCS y división en múltiples partes.

**Solicitud**:

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

**Cuerpo**:

```
{
  "source_msisdn": "15555551234",
  "destination_msisdn": "15555555678",
  "message": "Hello World",
  "source_smsc": "api_client"
}
```

El texto del mensaje puede ser dado como `message`, `text` o `body`. `source_msisdn` y `destination_msisdn` son requeridos.

**Respuesta** (201 Creado):

- Parte única: el objeto de mensaje creado.
- Multipart: `{ "parts": <n>, "messages": [ ... ] }`.

```
curl -X POST https://host:8443/api/messages_plaintext \
  -H "Content-Type: application/json" \
  -d '{
    "source_msisdn": "15555551234",
    "destination_msisdn": "15555555678",
    "message": "Hello World",
    "source_smsc": "api_client"
  }'
```

# Enviar PDU Crudo

Enviar un PDU SMS crudo (hex) para máxima compatibilidad con frontends heredados y SIP.

**Solicitud**:

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

**Cuerpo**:

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

**Campos**:

- `message_body` - **requerido**. PDU SMS codificado en hex (o TPDU, ver abajo).
- `source_msisdn` - opcional.
- `source_smsc` / `smsc_node_name` - requerido en modo solo TPDU.

- `rp_originator_address`, `rp_destination_address` - proporcione estos para enviar un TPDU desnudo (la capa RP es sintetizada).

La IP del llamador se toma de `X-Forwarded-For` (o el socket) para resolver el frontend de origen.

**Respuestas**:

- 201 Creado con `{ "status": "success", "response": { "id": <id> } }` (se incluye un campo hex `rp_ack` cuando se genera uno).
- 200 OK con `{ "status": "success", "response": { "status": "delivery_report_received" } }` cuando el PDU es un informe de entrega en lugar de un nuevo mensaje.
- 422 con una cadena `error` por hex faltante/inválido o fallos de decodificación.

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

El servicio decodifica el PDU (3GPP TS 23.040), extrae números/texto/DCS, detecta informes de entrega y aplica enrutamiento. Los informes de entrega se registran pero no se almacenan como nuevos mensajes.

# Actualizar Mensaje (PATCH)

Actualización parcial de un mensaje almacenado.

**Solicitud**:

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

Envíe cualquier campo de mensaje escribible (por ejemplo, `dest_smsc`, `deliver_after`, `status`). Los cambios de metadatos se registran en el registro de eventos.

**Respuesta** (200 OK): el objeto de mensaje actualizado. 404 si no se encuentra, 422 en un actualización rechazada.

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

## Incrementar Intento de Entrega

Registrar una entrega fallida: incrementa el contador de intentos, mueve el mensaje a `backoff`, y programa el próximo reintento con retroceso exponencial.

Existen dos formas equivalentes:

```
PUT  /api/messages/:id                  # el recurso de mensajes,
PUT = incrementar
POST /api/messages_increment_attempt        (id en el cuerpo)
PUT  /api/messages_increment_attempt/:id    (id en la URL)
```

`PUT /api/messages/:id` es distinto de `PATCH /api/messages/:id`: PATCH realiza una actualización parcial, PUT incrementa el intento de entrega. Pase `keep_route=true` para retener el actual `dest_smsc` en el reintento.

**Respuesta** (200 OK): el mensaje actualizado, por ejemplo.

```
{ "id": 12345, "status": "backoff", "delivery_attempts": 2 }
```

```
curl -X PUT https://host:8443/api/messages/12345
```

## Marcar como Entregado

Marcar un mensaje como entregado (almacenamiento local o federado).

```
POST /api/messages_mark_delivered           (id en el cuerpo)
PUT  /api/messages_mark_delivered/:id        (id en la URL)
```

El encabezado opcional `smsc` selecciona el SMSC federado cuando el mensaje no se mantiene localmente.

**Respuesta** (200 OK): el objeto de mensaje entregado. 404 si no se encuentra.

```
curl -X POST https://host:8443/api/messages_mark_delivered \
  -H "Content-Type: application/json" \
  -d '{ "id": 12345 }'
```

## Reenviar Mensaje

Reevaluar las reglas de enrutamiento para un mensaje almacenado y actualizar su `dest_smsc`.

**Solicitud**:

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

**Cuerpo**: `{ "id": 12345 }`

**Respuesta** (200 OK):

```
{ "message_id": 12345, "new_dest_smsc": "uk_gateway", "status":
"rerouted" }
```

404 si el mensaje no se encuentra, 422 si no coincide ninguna ruta.

## Limpiar Campo

Restablecer un campo específico del mensaje a nulo.

**Solicitud**:

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

**Cuerpo**: `{ "id": 12345, "field": "dest_smsc" }`

`field` debe ser uno de `delivery_time`, `dest_smsc`, o `deliver_after`; cualquier otro valor devuelve 400. El cambio se registra en el registro de eventos.

**Respuesta** (200 OK): el objeto de mensaje actualizado.

## Eliminar Mensaje

**Solicitud**:

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

**Respuesta** (204 Sin Contenido). 404 si no se encuentra. La eliminación es permanente.

```
curl -X DELETE https://host:8443/api/messages/12345
```

# Indicación de Mensaje en Espera

Enviar una notificación MWI (Indicación de Mensaje en Espera). El controlador construye el PDU MWI según 3GPP TS 23.038 y almacena un mensaje listo para entregar con un `payload_hex` y `dcs` precomputados.

**Solicitud**:

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

**Cuerpo**:

```
{
  "source_msisdn": "12345",
  "destination_msisdn": "15555555678",
  "indication_type": "voicemail",
  "indication_active": true,
  "message_body": "You have 1 new voicemail",
  "source_smsc": "IMS_SMSC"
}
```

**Campos**:

- `indication_type` - `voicemail` (por defecto), `fax`, `email`, u `other`.
- `indication_active` - `true` para establecer el indicador, `false` para borrarlo (por defecto `true`).
- `message_body` - texto opcional acompañante (codificado en GSM7).
- `source_msisdn`, `destination_msisdn`, `source_smsc`, `dest_smsc`.

**Respuesta** (201 Creado): el objeto de mensaje creado. 422 en caso de fallo.

# Ubicaciones

Registros de ubicación de suscriptores para entrega móvil-terminada. Recurso REST estándar; el segmento de ruta `:id` es el IMSI.

```
GET    /api/locations           # listar (paginado: ?
limit=&offset=, encabezado x-total-count)
GET    /api/locations/:id       # obtener por IMSI
POST   /api/locations           # crear o actualizar (identificado
por IMSI)
PATCH  /api/locations/:id       # actualización parcial
PUT    /api/locations/:id       # actualizar
DELETE /api/locations/:id       # eliminar
```

**Cuerpo de Crear/Actualizar**:

```
{
  "msisdn": "15555551234",
  "imsi": "001001000000001",
  "location": "msc1.region1.example.com",
  "ran_location": "cell_tower_12345",
  "imei": "123456789012345",
  "ims_capable": true,
  "csfb": false,
  "registered": true,
  "expires": "2025-11-01T23:30:00Z",
  "user_agent": "Samsung Galaxy"
}
```

`imsi` y `msisdn` son requeridos; los campos restantes son opcionales. Las respuestas son el objeto de ubicación (200/201); las fallas de validación devuelven 422. Llamado por sistemas de gestión de movilidad cuando un suscriptor se registra o se da de baja.

# Frontends

Rastrear conexiones SMSC de frontends. Este recurso expone **índice, mostrar y crear solamente** (no hay sub-rutas de `active`, `stats`, `history`, o `register`).

```
GET  /api/frontends           # listar todas las registraciones
GET  /api/frontends/:id       # obtener una registración
POST /api/frontends           # registrar o refrescar un frontend
```

**Cuerpo de Registro** (`POST /api/frontends`):

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

`frontend_name` es requerido. Si se omite `ip_address`, se completa desde la IP de origen de la solicitud (`X-Forwarded-For` o socket). Si ya existe una registración activa para el `frontend_name`, se actualiza; de lo contrario, se crea una nueva. Las registraciones expiran después de aproximadamente 90 segundos, por lo que los frontends deben re-registrarse periódicamente.

**Respuesta** (201 Creado): el objeto de registración.

# Eventos

Eventos del ciclo de vida por mensaje. Los eventos se mantienen en ETS con un TTL rodante.

## Obtener Eventos de Mensaje

**Solicitud**:

```
GET /api/events/:id
```

`:id` es el ID numérico del mensaje.

**Respuesta** (200 OK):


```
{
  "message_id": 12345,
  "events": [
    {
      "event_time": "2023-10-30T13:20:00.104211Z",
      "event_time_us": 1698672000104211,
      "event_epoch": 1698672000,
      "seq": -5764607523034231 00,
      "name": "http_request_received",
      "description": "POST /api/messages_raw from
10.0.1.5\nsource_msisdn: 61400000001\nfrontend: ims-frontend-
1\nmessage_body: 46 hex chars",
      "event_source": "smsc-node-1"
    },
    {
      "event_time": "2023-10-30T13:20:00.107950Z",
      "event_time_us": 1698672000107950,
      "event_epoch": 1698672000,
      "seq": -5764607523034228 00,
      "name": "http_response_sent",
      "description": "201 {\"status\":\"success\",\"response\":
{\"id\":12345,\"rp_ack\":\"033b\"}}",
      "event_source": "smsc-node-1"
    }
  ]
}
```


Los eventos se devuelven en el orden en que ocurrieron, ordenados por `event_time_us` (microsegundos desde la época Unix, UTC); `seq` rompe empates dentro de un nodo. `event_time` es el mismo instante que ISO 8601, y `event_epoch` es en segundos enteros. Los eventos que ocurren antes de que el mensaje tenga un id (la solicitud HTTP, decodificación PDU, traducción de números) mantienen el tiempo en que ocurrieron.

Un mensaje creado a través de `POST /api/messages` o `POST /api/messages_raw` comienza con `http_request_received` (método, ruta, IP del llamador y un resumen del cuerpo) y recibe `http_response_sent` (estado HTTP y cuerpo, truncado a 1024 caracteres) cuando se envía la respuesta.

400 si el id no es un entero.

## Registrar Evento

**Solicitud**:

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

**Cuerpo**:

```
{
  "message_id": 12345,
  "name": "external_delivery_confirmed",
  "description": "Confirmed by downstream system"
}
```

`message_id`, `name` y `description` son todos requeridos. La fuente del evento es detectada automáticamente desde la IP del llamador (coincidida contra las registraciones de frontend activas), retrocediendo al nombre de nodo configurado.

Si `message_id` es un mensaje federado (no en la tienda local, pero servido desde la caché de federación), el evento se envía al controlador propietario y se registra allí. Si no se puede alcanzar al propietario, la respuesta es 502 y nada se registra localmente.

**Respuesta** (201 Creado):

```
{
  "status": "success",
  "message": "Event recorded successfully",
  "event": {
    "message_id": 12345,
    "name": "external_delivery_confirmed",
    "description": "Confirmed by downstream system",
    "event_source": "smpp-gateway-1",
    "event_epoch": 1698672010
  }
}
```

# Campañas

Mensajería masiva: cargue una lista de destinatarios con nombre, luego ejecute una campaña limitada por tasa contra ella. Consulte la guía de Mensajería Masiva / Campañas para conceptos (filtro de audiencia, tasa de goteo, ciclo de vida, estadísticas).

## Listas de Destinatarios

```
GET    /api/campaign_lists           # todas las listas
GET    /api/campaign_lists/:id       # una lista, incluye
"recipients"
POST   /api/campaign_lists           # crear una lista
PATCH  /api/campaign_lists/:id       # agregar destinatarios
DELETE /api/campaign_lists/:id       # eliminar la lista y sus
destinatarios
```

Cada destinatario puede llevar hasta cuatro variables de plantilla (`var1`..`var4`). Proporcione destinatarios como un arreglo JSON, una cadena CSV cruda, o ambos. Los destinatarios JSON pueden ser cadenas MSISDN simples o objetos con `var1`..`var4`:

```
{
  "name": "VIP customers",
  "description": "High value subscribers",
  "recipients": [
    { "msisdn": "12025550101", "var1": "Alice", "var2": "Gold" },
    "12025550102"
  ]
}
```

Para CSV, una fila de encabezado que nombra una columna `msisdn` / `destination_msisdn` / `number` / `phone` selecciona la columna MSISDN (de lo contrario, se utiliza la primera columna); las columnas restantes se convierten en `var1`..`var4` en orden de columna:

```
{ "name": "Imported", "csv":
"msisdn,first,plan\n12025550101,Alice,Gold\n" }
```

`PATCH` agrega destinatarios (no los reemplaza).

**Respuesta** (201 Creado): `{ "id": 1, "name": "VIP customers", "recipient_count": 2 }`

## Campañas

```
GET    /api/campaigns        # todas las campañas (paginado, más
recientes primero)
GET    /api/campaigns/:id    # una campaña, incluyendo
estadísticas en vivo
POST   /api/campaigns        # crear una campaña (comienza en
borrador)
DELETE /api/campaigns/:id    # detener una campaña pero mantener
su registro
```

**Crear Cuerpo**:

```
{
  "name": "Aviso de mantenimiento",
  "message": "Mantenimiento programado esta noche 02:00-03:00.",
  "source_msisdn": "12345",
  "source_smsc": "IMS_SMSC",
  "list_id": 1,
  "audience": "active_only",
  "drip_tps": 50
}
```

**Campos**:

- `name`, `message`, `source_msisdn`, `source_smsc` - **requeridos**. `message` soporta variables de plantilla por destinatario (`{{ var1 }}`..`{{ var4 }}`, `{{ msisdn }}`).
- `list_id` - lista de destinatarios. **Omitir para dirigir a todos los suscriptores actualmente activos (registrados).**
- `audience` - `active_only` (predeterminado) o `all`. Ignorado cuando `list_id` es omitido.
- `drip_tps` - mensajes enviados a la cola por segundo (predeterminado de la configuración).
- `validity_hours` - validez del mensaje en horas; establece el `expires` de cada mensaje.
- `deliver_after` - tiempo ISO 8601 opcional para retener mensajes hasta (envío programado).

`DELETE` detiene la entrega pero mantiene el registro para que la historia y las estadísticas permanezcan visibles; devuelve la campaña cancelada (200) y se purga automáticamente una vez que pasa el TTL de retención.

**Respuesta** (`GET /api/campaigns/:id`):

```
{
  "id": 7,
  "name": "Aviso de mantenimiento",
  "list_id": 1,
  "audience": "active_only",
  "status": "running",
  "drip_tps": 50,
  "total_targets": 1000,
  "stats": {
    "total": 1000, "pending": 600, "queued": 300,
    "delivered": 90, "failed": 10, "skipped_inactive": 0,
    "dispatched": 400, "progress_percent": 40, "delivery_percent":
22
  }
}
```

## Controlar una Campaña

**Solicitud**:

```
POST /api/campaigns/control
Content-Type: application/json
```

**Cuerpo**: `{ "campaign_id": 7, "action": "start" }`

`action` es uno de `start`, `pause`, `resume`, `cancel`. Devuelve la campaña actualizada (200), `404` si no se encuentra, o `422` si la acción es inválida para el estado actual.

## Resultados por Destinatario

```
GET /api/campaign_targets/:id              # :id es el ID de la
campaña
GET /api/campaign_targets/:id?state=failed # filtrar por estado
del destinatario
```

Soporta paginación `limit`/`offset`; el conteo total (después de cualquier filtro `state`) está en el encabezado de respuesta `x-total-count`. Cada entrada lleva el `state` del destinatario (`pending` / `queued` / `delivered` / `failed` / `skipped_inactive`).

# Otras Familias de Endpoints

Las siguientes familias de rutas son parte de la misma API pero están documentadas en profundidad en sus guías dedicadas. Las rutas se enumeran aquí por completitud.

## Enrutamiento

```
GET    /api/routes             # listar rutas
GET    /api/routes/:id         # obtener una ruta
POST   /api/routes             # crear una ruta
PATCH  /api/routes/:id         # actualizar
PUT    /api/routes/:id         # actualizar
DELETE /api/routes/:id         # eliminar
GET    /api/routes/export      # exportar todas las rutas
POST   /api/routes/import      # importación masiva
POST   /api/routes/simulate    # simulación de una decisión de ruta
```

Consulta la guía de Enrutamiento SMS.

## Traducciones de Números

```
GET    /api/translations           # listar
GET    /api/translations/:id       # obtener
POST   /api/translations           # crear
PATCH  /api/translations/:id       # actualizar
DELETE /api/translations/:id       # eliminar
GET    /api/translations/export    # exportar
POST   /api/translations/import    # importación masiva
POST   /api/translations/simulate  # simulación de una traducción
```

Consulta la guía de Traducción de Números.

## Operaciones

```
GET  /api/operations                    # índice de operaciones
POST /api/operations/clear_messages     # purgar el almacén de
mensajes
POST /api/operations/clear_routes       # purgar rutas
POST /api/operations/clear_translations # purgar traducciones
POST /api/operations/clear_frontends    # purgar registros de
frontend
```

Estas son acciones de mantenimiento destructivas; protege el puerto de la API en consecuencia.

## Federación

```
GET  /api/federation/status          # estado de la federación
GET  /api/federation/identity        # identidad de federación de
este nodo
POST /api/federation/health          # informe de salud del par
POST /api/federation/registry        # intercambio de registro
POST /api/federation/notify          # notificación de mensaje
entrante
POST /api/federation/delivery_status # callback de estado de
entrega
POST /api/federation/delivery_attempt # callback de intento de
entrega fallido
POST /api/federation/event           # callback de evento de
mensaje
GET  /api/federation_messages        # vista de mensajes federados
```

Consulta la guía de Federación.

## Diámetro, Registros y DNS

```
GET  /api/diameter   # estado de peer/dip de Diameter Sh
GET  /api/logs       # registros estructurados recientes
POST /api/naptr      # prueba de resolución NAPTR
```

La búsqueda de suscriptores de Diameter Sh se describe en la guía de búsqueda de suscriptores HSS.

# Códigos de Estado HTTP

| Código | Significado | Uso Típico |
|---|---|---|
| 200 | OK | Lectura o actualización exitosa |
| 201 | Creado | Recurso creado |
| 202 | Aceptado | Aceptado para procesamiento asíncrono |
| 204 | Sin Contenido | Eliminación exitosa |
| 400 | Solicitud Incorrecta | Entrada mal formada (por ejemplo, id no entero, campo inválido) |
| 404 | No Encontrado | Recurso desconocido |
| 422 | Entidad No Procesable | Fallo de validación |
| 500 | Error Interno del Servidor | Error inesperado del servidor |

# Formato de Error

Los errores a nivel de controlador devuelven una cadena `error` plana:

```
{ "error": "Mensaje no encontrado" }
```

Los errores a nivel de marco (rutas no coincidentes, cuerpos no analizables) devuelven el sobre normalizado:

```
{ "status": "error", "response": "No Encontrado" }
```

# Mejores Prácticas

## Envío de Mensajes

1. **Pre-codificado vs texto plano**: envía `payload_hex` a `/api/messages` cuando ya hayas codificado el TPDU; de lo contrario, usa `/api/messages_plaintext` y deja que el SMS-C codifique y divida.
2. **Siempre establece `source_smsc`** para que los eventos y el enrutamiento puedan atribuir el mensaje.
3. **Usa números estilo E.164** de manera consistente.
4. **Maneja 422** inspeccionando la cadena `error` en lugar de reintentar ciegamente.

## Integración de Frontend

1. **Re-registra** en `POST /api/frontends` bien dentro de la ventana de caducidad de ~90 segundos.
2. **Consulta** con el encabezado `smsc` para tus mensajes; usa `include-unrouted` solo si deseas deliberadamente tráfico no enrutado.
3. **Marca como entregado** a través de `/api/messages_mark_delivered` después del éxito.
4. **Incrementa en caso de fallo** a través de `PUT /api/messages/:id` (o `/api/messages_increment_attempt`) para programar un reintento con retroceso.
5. **Registra eventos** para confirmaciones posteriores.

## Operacional

1. Termina TLS y restringe el puerto de la API en la capa de red; no hay autenticación incorporada.
2. Reutiliza conexiones HTTP y establece tiempos de espera de cliente razonables.

3. Observa las métricas de Prometheus (consulta METRICS.md) para la salud de la cola y la entrega.

# 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 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 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 SMPP, IMS, frontends 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 Detalle 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 (predeterminado 24 horas), luego limpieza automática
- **Archivo CDR a Largo Plazo:** Base de datos MySQL/PostgreSQL (separada de la cola de mensajes)
  - CDRs escritos cuando los mensajes son entregados, expiran, fallan o son rechazados
  - Base de datos SQL utilizada SOLAMENTE para exportación/archivo de CDR, NO para operaciones de mensajes activos
  - Sin impacto en el rendimiento del enrutamiento de mensajes (escritura asíncrona)

- **Beneficios de Arquitectura de Dos Niveles:**
  - Cola activa: Rápido como un rayo (1,750 msg/sec) sin cuellos de botella SQL
  - Archivo CDR: Retención a largo plazo (meses/años) para facturación e intercepción 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 con frontends externos
- **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 dinámico de SMS con actualizaciones de configuración en tiempo de ejecución
- Coincidencia basada en prefijos (números que llaman/reciben)
- 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 (Mapeo de Números E.164)
- Capacidades de respuesta automática y eliminación de mensajes
- Control de carga por ruta (integración OCS sobre Diameter Ro)

 **Arquitectura y características completas documentadas en README.md**

# 1.2 Capacidades de Intercepción

OmniMessage implementa las interfaces de intercepción ETSI TS 103 221 (X1, X2, X3). OmniMessage actúa como el Elemento de Red y Punto de Intercepción.

Una Instalación de Monitoreo de Aplicaciones de la Ley provisiona objetivos a través de X1. OmniMessage entrega SMS interceptados como 3GPP TS 33.128 xIRI a través de X2. Para los roles de interfaz y la referencia completa de parámetros, consulte Intercepción Legal (X1/X2/X3).

### 1.2.1 Adquisición de Mensajes

**Captura de Mensajes SMS:**

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

**Adquisición de Metadatos de Mensajes:**

- Registros de Detalle de Llamadas (CDR) completos almacenados en la base de datos con:
  - ID del mensaje (identificador único)
  - Número que llama (MSISDN de origen)
  - Número llamado (MSISDN de destino)
  - Marca de tiempo de envío (cuando el mensaje ingresó al sistema)
  - Marca de tiempo de entrega (cuando el mensaje fue entregado)
  - Marca de tiempo de expiración (cuando el mensaje expiró si no se pudo entregar)
  - Estado (entregado, expirado, fallido, rechazado)
  - Conteo de intentos de entrega

    - Partes del mensaje (para SMS concatenados/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 a la base de datos para búsqueda de mensajes históricos
- Capacidades de filtrado por:
    - Número de teléfono (origen/destino)
    - Puerta de enlace SMSC
    - Rango de tiempo
    - Estado del mensaje
    - Intentos de entrega

**📄 Documentación completa de la API en API_REFERENCE.md**

### 1.2.2 Capacidades de Procesamiento de Datos

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

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

**Nivel 1: Cola de Mensajes Activa (Mnesia)**

- **Propósito:** Operaciones de enrutamiento y entrega de mensajes en tiempo real
- **Tecnología:** Base de datos distribuida Erlang Mnesia

- **Modo de Almacenamiento:** En memoria con respaldo disc_copies
  - Almacenamiento primario en RAM para máxima velocidad
  - Sincronización automática en disco para recuperación ante fallos
  - Los mensajes persisten a través de reinicios del sistema
- **Rendimiento:** Operaciones de lectura/escritura sub-milisegundo
- **Retención:** A corto plazo (predeterminado 24 horas), configurable
- **Limpieza:** Archivado automático a la base de datos CDR, luego eliminación de Mnesia
- **Operaciones:** Todas las operaciones de cola de mensajes (inserción, actualización, estado de entrega, enrutamiento)
- **Característica Crítica:** La base de datos SQL NUNCA es consultada durante el enrutamiento/entrega de mensajes

**Nivel 2: Archivo CDR (MySQL/PostgreSQL)**

- **Propósito:** Almacenamiento a largo plazo para facturación, análisis e intercepción legal
- **Tecnología:** Base de datos SQL tradicional (MySQL o PostgreSQL)
- **Disparador de Escritura:** CDRs escritos SOLAMENTE 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 asíncrona por lotes (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, intercepción legal
- **Acceso:** Consultas SQL, REST API (futuro), exportación CSV/JSON

**Beneficios Arquitectónicos Clave:**

1. **Rendimiento:** Las operaciones de enrutamiento activas nunca tocan SQL (sin cuellos de botella de base de datos)

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

**Ciclo de Vida del Mensaje:**

```
1. Mensaje enviado → Almacenado en Mnesia (RAM + respaldo en
disco)
2. Mensaje enrutado → Consulta a 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
intercepción legal (años)
```

**Retención y Recuperación de Datos:**

- Retención o eliminación configurable del cuerpo del mensaje por privacidad
- Preservación de datos binarios (almacenamiento de PDU en bruto 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 intercepción legal

**Seguimiento de Frontend:**

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

- Registro de configuración específica de frontend

### 1.2.3 Capacidades de Análisis

**Monitoreo en Tiempo Real:**

- Panel de control de interfaz web que muestra:
  - Cola de mensajes activa
  - Envío y entrega de mensajes
  - Decisiones de enrutamiento y selección de puerta de enlace
  - Estado de la puerta de enlace del 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 la parte que llama/recibe
  - 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 promedio de entrega
  - Análisis de mensajes multipartes
  - Patrones de entrega fallida

**Seguimiento de Suscriptores:**

- Historial de mensajes por número de teléfono (MSISDN)

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

**Análisis de Red:**

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

**Inteligencia de Números:**

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

**Guía de traducción de números en number_translation_guide.md Guía de enrutamiento en sms_routing_guide.md**

# 1.3 Capacidades de Contramedidas

### 1.3.1 Mecanismos de Protección de 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 del Mensaje:**

- 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 ser establecido 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

**Endurecimiento del Sistema:**

- Puertos de red expuestos mínimos
- Gestión de certificados TLS

- Almacenamiento seguro de configuraciones
- Separación de configuraciones basada en el entorno
- Seguridad de clúster con protocolo de distribución Erlang

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

### Descripción 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 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 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 forma asíncrona en segundo plano
  - La copia en disco se actualiza en cada confirmación de transacción
  - Recuperación ante fallos: El sistema se reinicia con todos los mensajes intactos
  - Ubicación: directorio `Mnesia.*/` en 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. El motor de enrutamiento consulta a Mnesia → Respuesta instantánea (acceso a memoria)
3. La puerta de enlace externa sondea mensajes → Consulta a Mnesia (acceso a memoria)
4. La puerta de enlace actualiza el estado de entrega → Actualización de Mnesia (memoria + disco)
5. Después de la entrega/expiración → Mensaje marcado para limpieza
6. Trabajador de limpieza (predeterminado 24h) → Mensaje eliminado de Mnesia

**Característica Crítica de Rendimiento:**

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

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

**¿Qué es CDR (Registro de Detalle 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 intercepción legal

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

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

**Cómo se Escriben los CDR:**

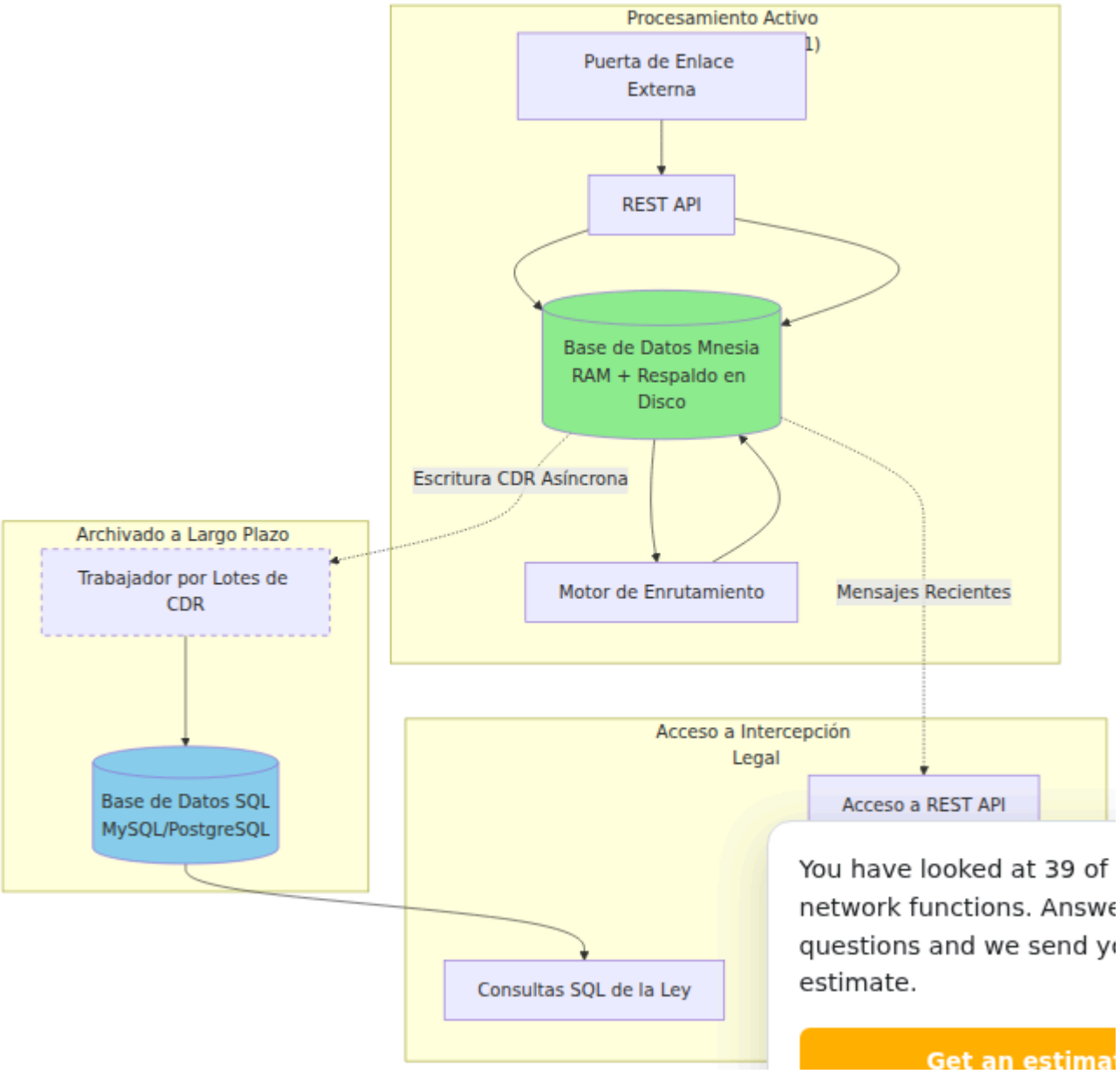
- **Escritura asíncrona por lotes:** CDRs escritos en proceso de trabajador en segundo plano
- **Sin bloqueo:** El enrutamiento de mensajes nunca espera por la escritura SQL
- **Inserciones por lotes:** Múltiples CDRs agrupados (predeterminado 100) y escritos juntos
- **Intervalo de vaciado:** 100ms predeterminado (configurable)
- **Manejo de errores:** Escrituras de CDR fallidas registradas, el procesamiento de mensajes continúa

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

**Propósito de la Base de Datos SQL:**

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

**Diagrama de Arquitectura**




**Leyenda:**

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

**Implicaciones de Intercepción Legal**

**Mensajes Recientes (< 24 horas):**

- Accesibles a través de Mnesia (consultas REST API)
- Recuperación ultra-rápida

- Contenido completo del mensaje disponible
- Monitoreo en tiempo real posible

**Mensajes Históricos (> 24 horas):**

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

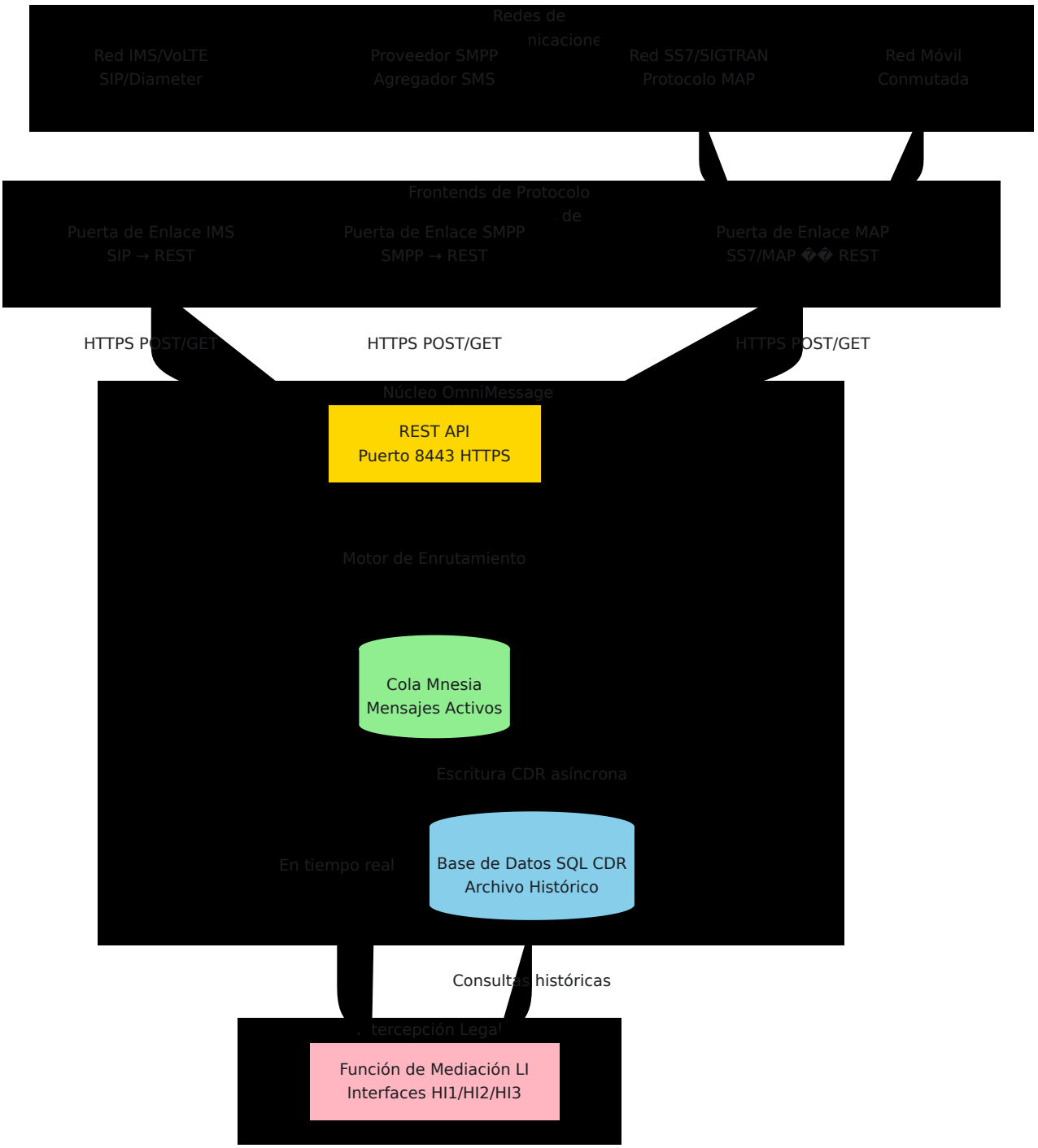
**Beneficios de Cumplimiento:**

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 SMSc emplea un diseño central independiente de protocolos que se conecta con puertas de enlace específicas de protocolo externas (frontends) 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.

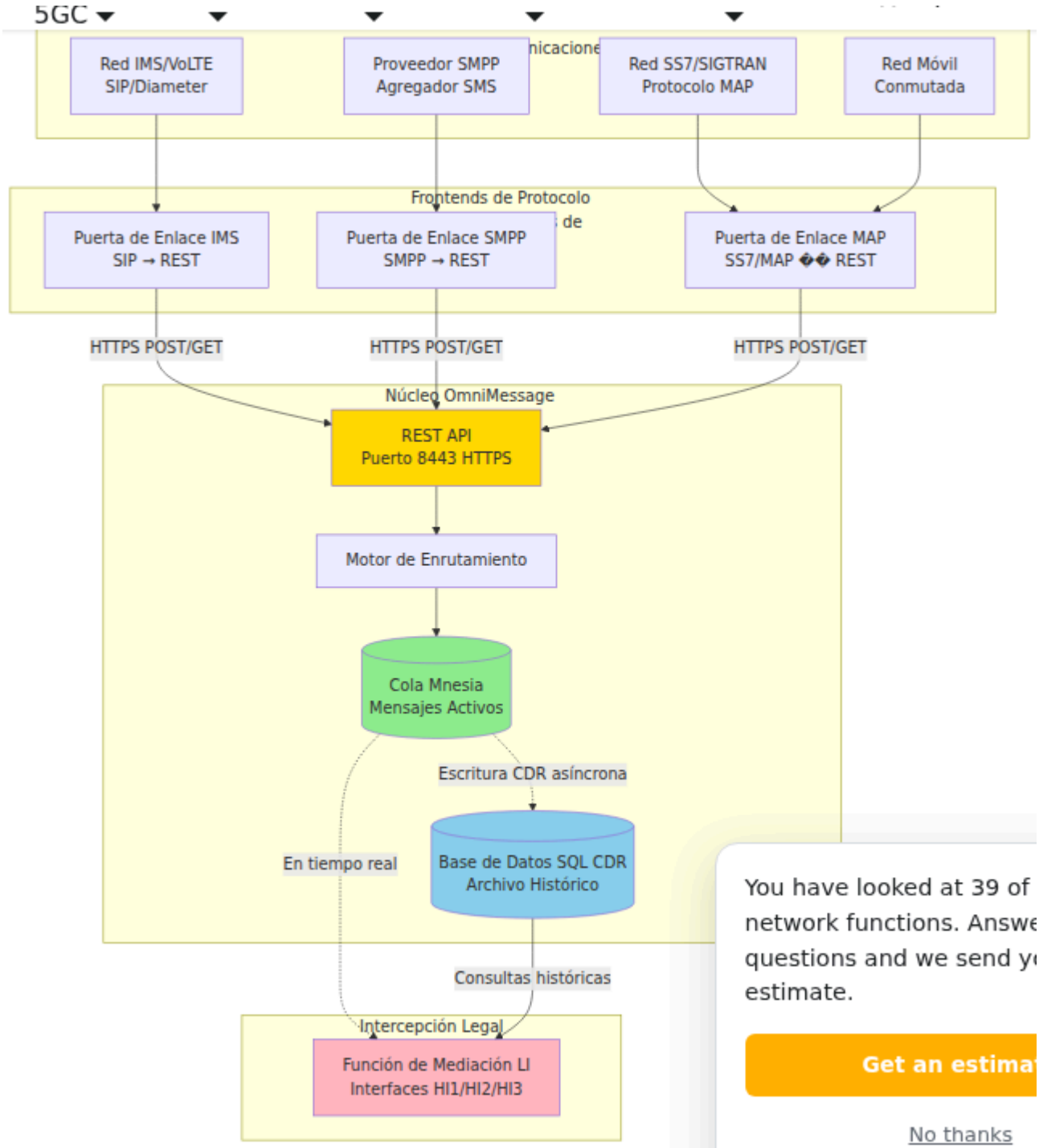
**Descripción General de la Arquitectura**




## Detalles de Integración de Protocolo de Frontend

### 1. Integración de Frontend IMS/SIP

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



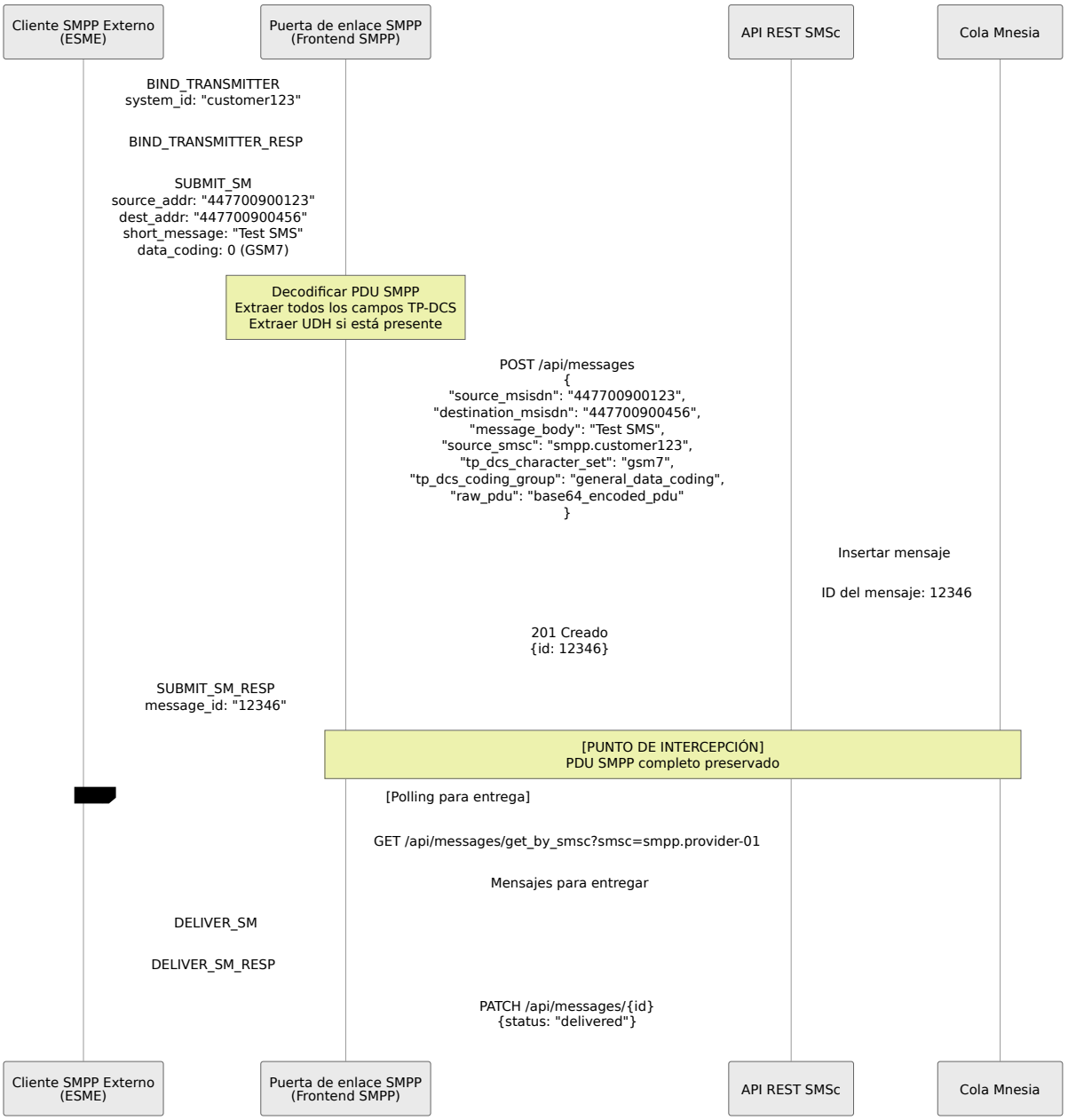

You have looked at 39 of
network functions. Answe
questions and we send y
estimate.
Get an estima
No thanks

**Datos de Intercepción Específicos de IMS:**

- IMSI de origen/destino (de la inscripción IMS)
- Encabezados SIP de Identidad Afirmada P
- ID de Llamada SIP para correlación
- Ubicación de la red IMS (Información de Red de Acceso P)
- Perfiles de suscriptor de IMS HSS

## 2. INTEGRACIÓN DEL FRONTEND SMPP

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

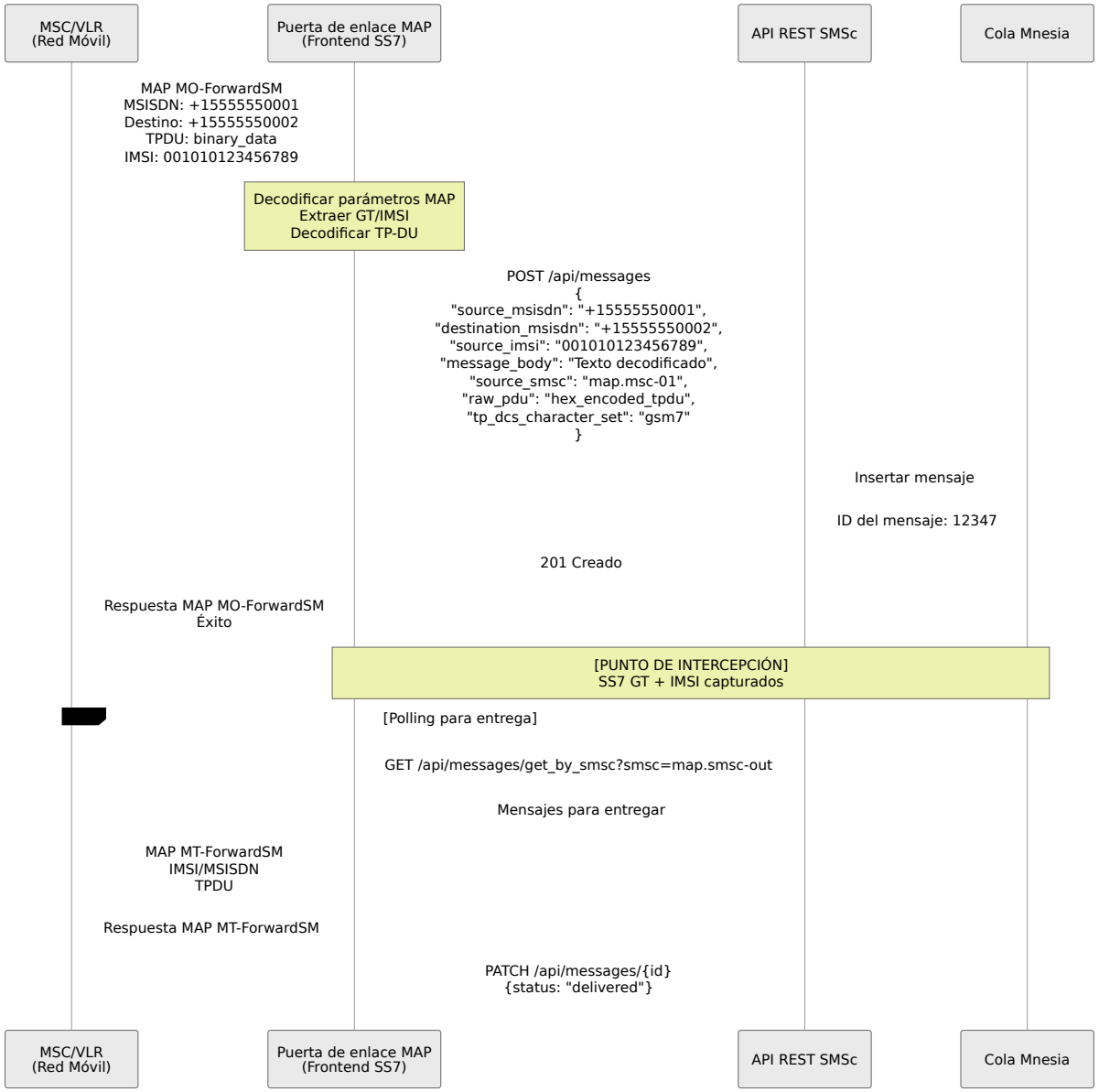



## Datos de Intercepción Específicos de SMPP:

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

## 3. INTEGRACIÓN DEL FRONTEND SS7/MAP

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




### Datos de Intercepción Específicos de SS7/MAP:

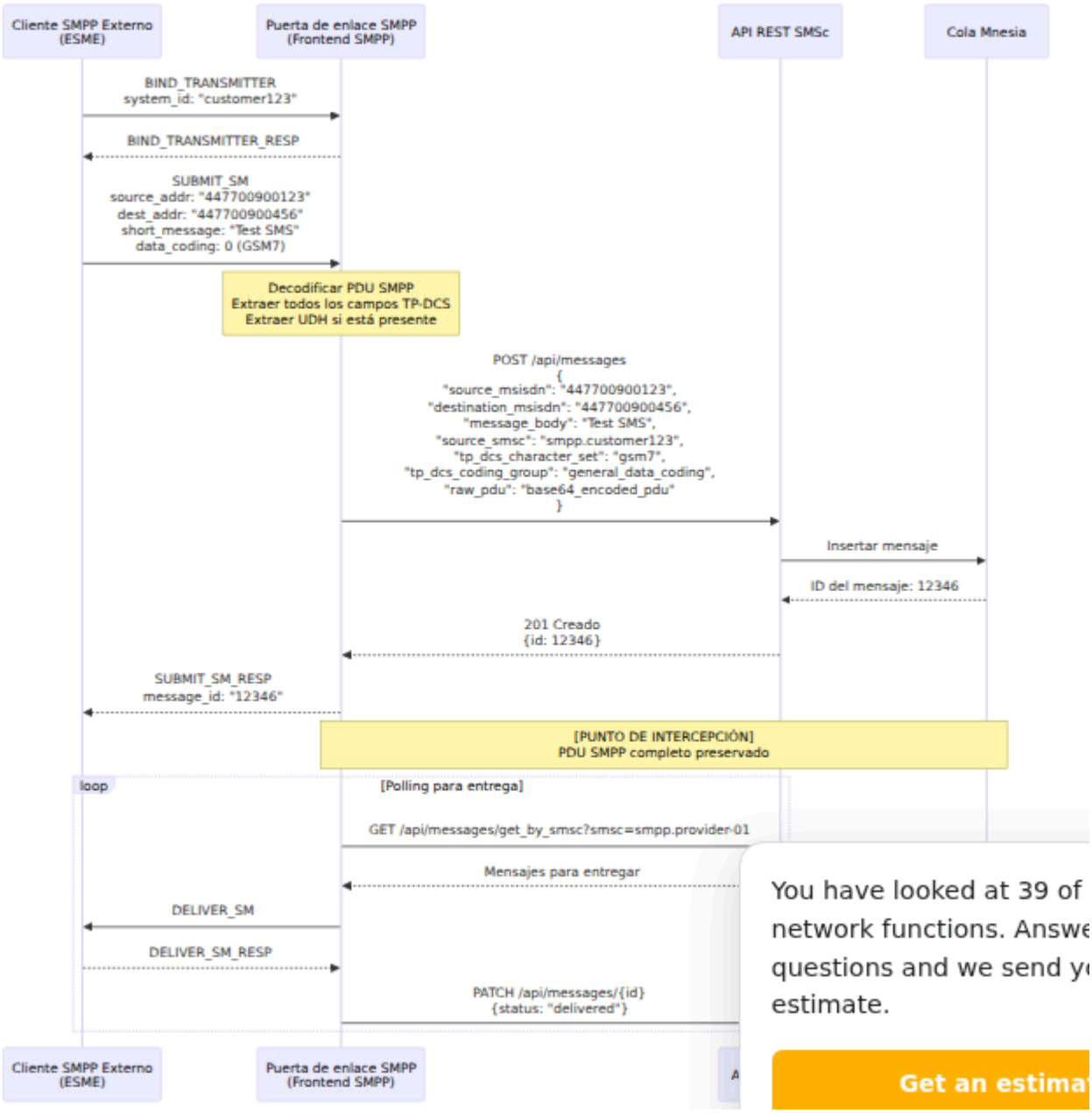
- IMSI de mensajes MAP
- Direcciones de Título Global (GT)
- Dirección MSC/VLR (identificación del elemento de red)
- Direcciones de parte llamada/llamante SCCP
- Códigos de operación MAP
- Formato binario TP-User-Data

**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 de SMSc con una estructura de datos normalizada, lo que permite:

1. **Monitoreo Agnóstico al Protocolo:** Un único punto de intercepción captura todos los tipos de mensajes
2. **Formato CDR Unificado:** Todos los protocolos escriben en el mismo esquema CDR
3. **Correlación Inter-Protocolo:** Rastrear mensajes a través de fronteras de protocolo
4. **Preservación Completa de Metadatos:** Campos específicos del 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 filtrar y analizar por tipo de protocolo
- Las consultas de intercepción legal pueden dirigirse a protocolos específicos o a todos los protocolos

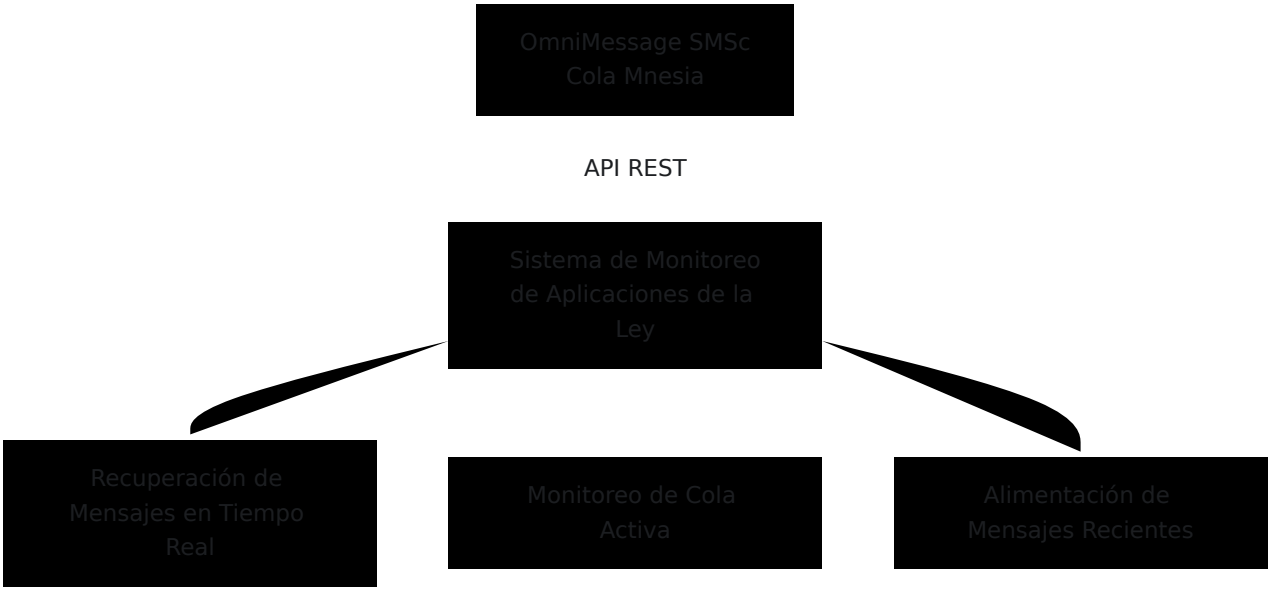
# 1.6 Arquitectura Técnica para la Intercepción Legal

### 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 la API REST para Mensajes Recientes (Mnesia):**

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




**Puntos de Acceso de la 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 una respuesta instantánea

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

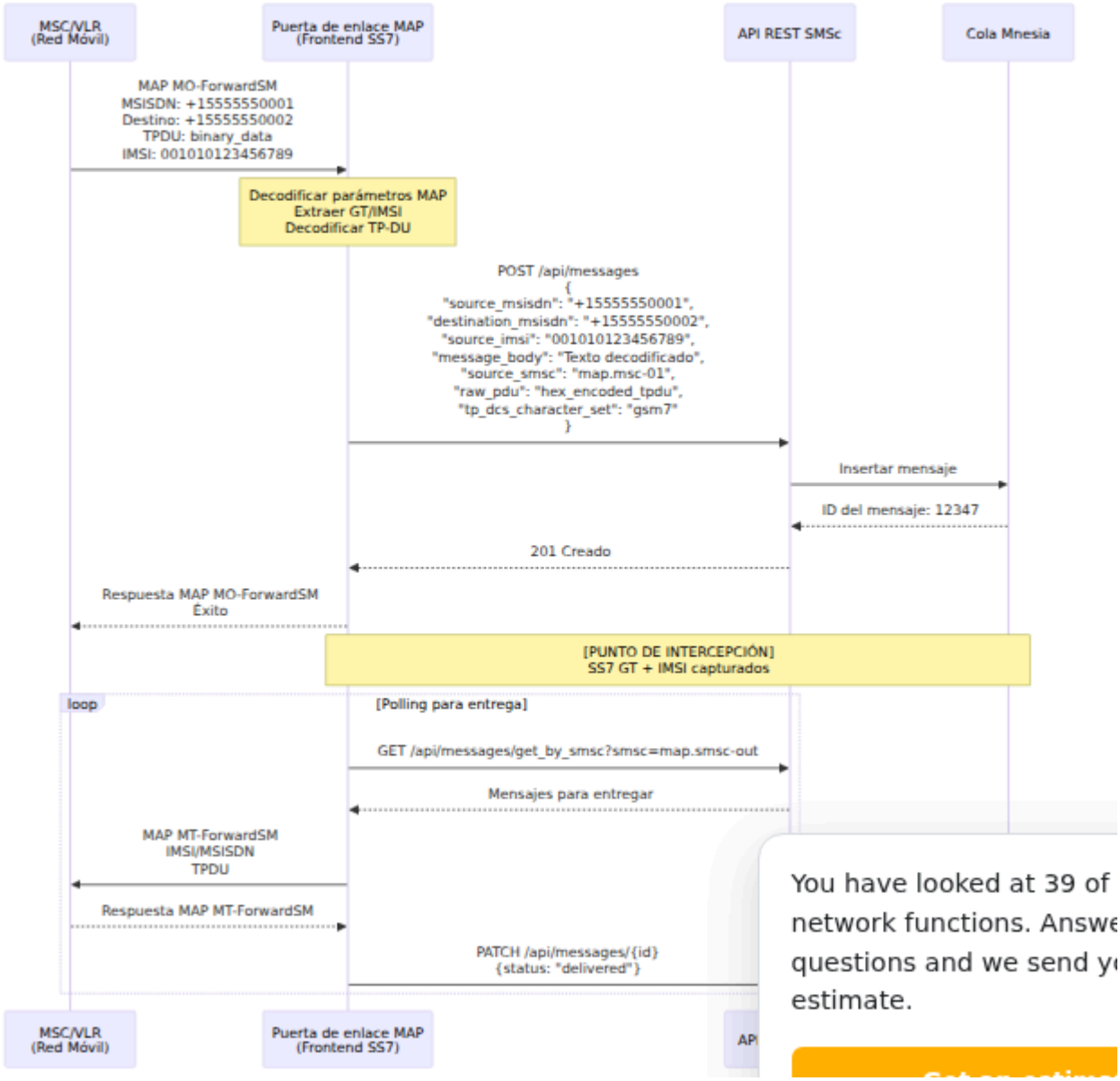
**Parámetros de Consulta:**

- Filtrar por MSISDN de origen/destino

- Filtrar por rango de tiempo
- Filtrar por puerta de enlace SMSC
- Filtrar por estado del mensaje
- Soporte para ordenación 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 SQL Directo:**

- Credenciales de base de datos de solo lectura para sistemas autorizados

- Acceso a consultas SQL a la tabla `cdrs` (rastro de 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) - Cuándo se ingresó el mensaje en el 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 de intercepción legal históricas (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

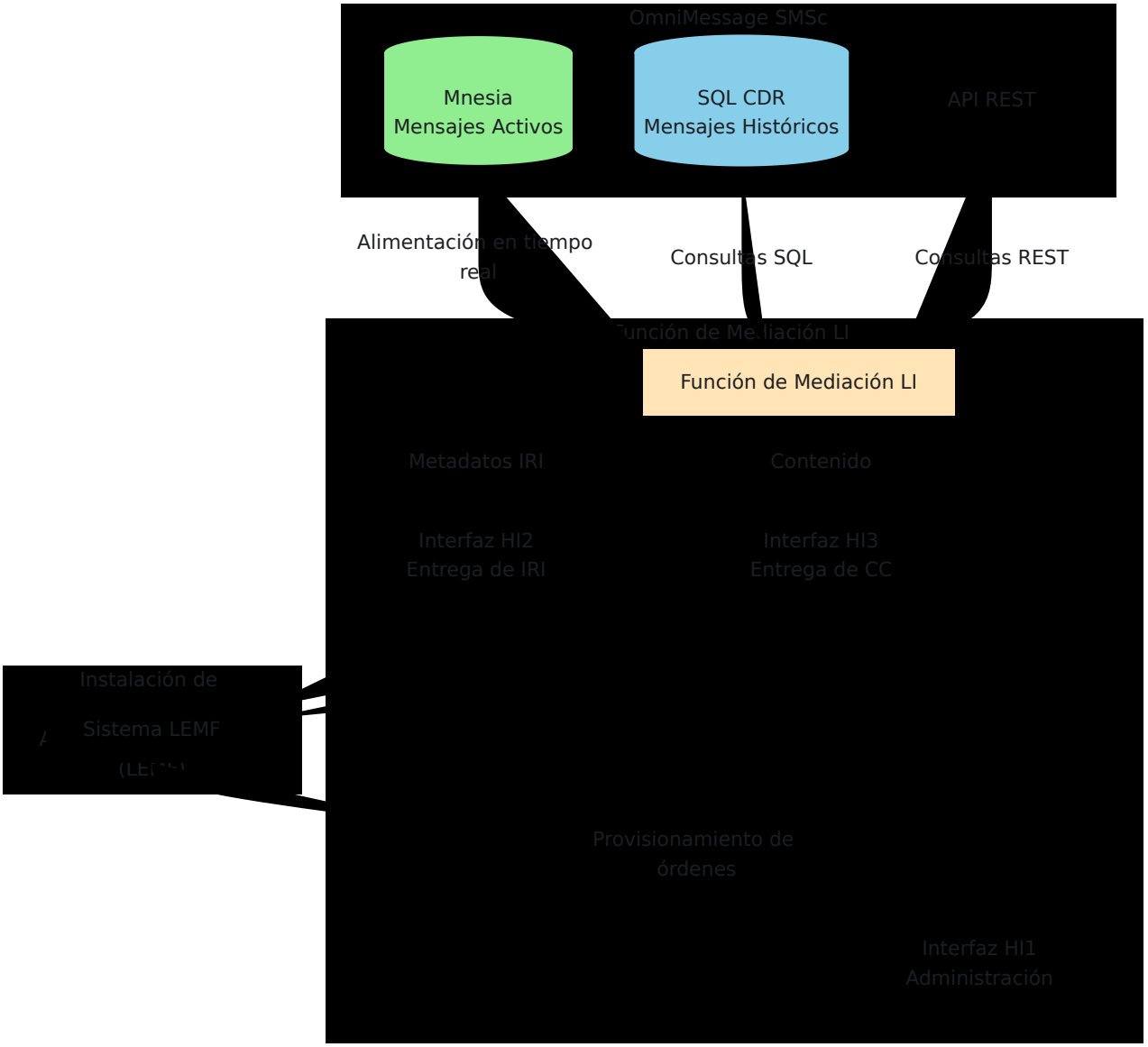
**4. Interfaz de Exportación por Lotes:**

- 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 (exclusión opcional del cuerpo del mensaje)

**Interfaces Estándar de Intercepción Legal de ETSI**

El OmniMessage SMSc proporciona la base para implementar interfaces de intercepción legal compatibles con ETSI. El núcleo de SMSc no implementa nativamente las interfaces de transferencia HI1/HI2/HI3. Proporciona todos los puntos de acceso de datos necesarios. Puede integrar estos puntos de acceso con sistemas externos de Función de Mediación de Intercepción Legal (LIMF).

**Interfaces LI Estándar de ETSI:**




**Descripciones de Interfaces:**

**Interfaz HI1 - Función de Administración:**

- **Propósito:** Provisionamiento de órdenes y objetivos de la aplicación de la ley al sistema de intercepción
- **Dirección:** LEMF → LIMF (bidireccional)

- **Funciones:**
  - Activar/desactivar la intercepción para objetivos específicos (MSISDN, IMSIs)
  - Establecer duración de la intercepción y período de validez
  - Configurar criterios de filtrado (números de teléfono, ventanas de tiempo)
  - Recuperar estado de intercepción
- **Integración con SMSc:**
  - LIMF mantiene una lista de objetivos (base de datos de órdenes)
  - LIMF consulta CDR/API de SMSc para mensajes coincidentes
  - LIMF filtra según los criterios provisionados por HI1

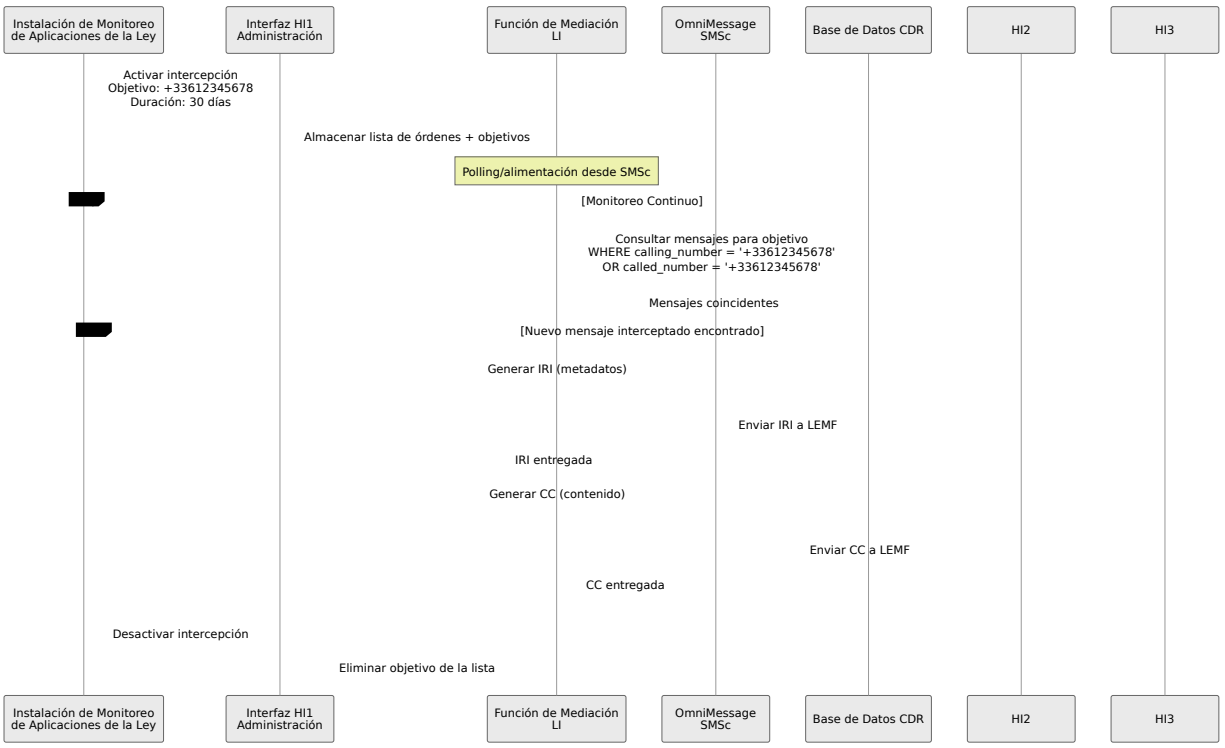
**Interfaz HI2 - Entrega de IRI (Información Relacionada con la Intercepción):**

- **Propósito:** Entregar metadatos de mensajes a la aplicación de la ley
- **Dirección:** LIMF → LEMF (unidireccional)
- **Formato de Datos:** XML/ASN.1 conforme a ETSI TS 102 232-x
- **Contenido de CDR de SMSc:**
  - 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 la 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 ETSI IRI
  - LIMF entrega IRI a LEMF a través de HI2

**Interfaz HI3 - Entrega de CC (Contenido de la Comunicación):**

- **Propósito:** Entregar el contenido real del mensaje a la aplicación de la ley
- **Dirección:** LIMF → LEMF (unidireccional)
- **Formato de Datos:** Conforme a ETSI TS 102 232-x
- **Contenido de SMSc:**
  - Cuerpo del mensaje (contenido de texto)
  - PDU crudo (datos binarios de SMS)
  - Información de codificación de caracteres
  - Segmentos de mensajes multipartes
  - Información TP-DCS
  - Encabezado de Datos del Usuario (UDH)
- **Integración con SMSc:**
  - LIMF recupera el contenido del mensaje del campo `message_body` de CDR
  - LIMF recupera datos de PDU crudo si están disponibles
  - LIMF empaqueta el contenido en formato ETSI CC
  - LIMF entrega CC a LEMF a través de HI3

### Arquitectura de Implementación:




### Mapeo de Datos de SMSc a Interfaces LI:

| Campo de Datos de SMSc | HI2 (IRI) | HI3 (CC) | Columna de Tabla CDR |
|---|---|---|---|
| ID del Mensaje | □ ID de Correlación | □ Referencia | `message_id` |
| Número de Llamada | □ Parte A | - | `calling_number` |
| Número Llamado | □ Parte B | - | `called_number` |
| Hora de Envío | □ Marca de Tiempo | - | `submission_time` |
| Hora de Entrega | □ Finalización | - | `delivery_time` |
| Estado | □ Resultado | - | `status` |
| Cuerpo del Mensaje | - | □ Contenido | `message_body` |
| PDU Crudo | - | □ 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 filtradas por números de teléfono objetivo de la lista de órdenes HI1
- Baja complejidad, fácil de implementar
- Ligera demora 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 personalizado

**Opción 3: Arquitectura Híbrida**

- Mensajes recientes: Alimentación en tiempo real 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 número 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

**Ejemplos de Consultas SQL para Intercepción Legal:**

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

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

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

---

# 2. CAPACIDADES DE ENCRIPTACIÓN Y CRIPTOANÁLISIS

## 2.1 Descripción General de Capacidades Criptográficas

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

## 2.2 Encriptación en la 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 Framework Phoenix

**Conjuntos de Cifrado:**

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

**Preferido - TLS 1.3:**

- TLS_AES_256_GCM_SHA384
- TLS_AES_128_GCM_SHA256
- TLS_CHACHA20_POLY1305_SHA256

**Soportado - TLS 1.2:**

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

**Características de Seguridad:**

- Perfect Forward Secrecy (PFS) a través de intercambio de claves ECDHE/DHE
- Grupos de Diffie-Hellman fuertes (mínimo de 2048 bits)
- Soporte para Criptografía de Curva Elíptica
- Soporte para Server Name Indication (SNI)

**Gestión de Certificados:**

- Soporte para certificados X.509
- Tamaños de clave RSA: mínimo de 2048 bits, recomendado 4096 bits
- Soporte para 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 completa de configuración en CONFIGURATION.md**

**Aplicaciones:**

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

# 2.3 Encriptación de Datos en Reposo

### 2.3.1 Encriptación de Base de Datos

**Encriptación de MySQL/MariaDB:**

- Soporte para encriptación a nivel de tabla
- Algoritmo de encriptación AES-256
- Encriptación de datos transparente (TDE)

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

**Encriptación de PostgreSQL:**

- Soporte para encriptación de datos transparente
- Encriptación a nivel de sistema de archivos
- Encriptación 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
- Se recomienda encriptación a nivel de sistema de archivos (LUKS, dm-crypt)
- Protección de memoria a través de aislamiento de la VM de Erlang

### 2.3.3 Encriptación del Sistema de Archivos

**Almacenamiento de Datos Sensibles:**

- Archivos de configuración: Se recomienda encriptación del sistema de archivos

- Claves privadas: Permisos de archivo (0600) + encriptación del sistema de archivos
- Archivos de registro: Encriptación configurable para registros archivados
- Exportaciones CDR: Almacenamiento encriptado 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 la API REST:**

- Encriptación de transporte HTTPS/TLS obligatoria
- Autenticación basada en encabezados (encabezado SMSc para identificación del frontend)
- Control de acceso basado en IP (nivel de firewall)
- Autenticación de cliente basada en certificado (opcional)

**Registro de Frontend:**

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

### 2.4.2 Autenticación de Base de Datos

**Control de Acceso a la Base de Datos:**

- Autenticación de nombre de usuario/contraseña
- Soporte para conexión TLS/SSL
- Restricciones de conexión basadas en IP

- Control de acceso basado en roles (RBAC)

**Configuración:**

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

**Recomendaciones de Control de Acceso:**

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

-- Crear usuario 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)
ON sms_c.cdrs 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 bits (codificación SMS estándar)
- 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 de conjunto de caracteres

**Sin Cifrado SMS Nativo:**

- El protocolo SMS no proporciona cifrado de extremo a extremo
- Contenido del mensaje accesible a nivel de SMSc
- 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 SMPP
- Soporte TLS disponible (SMPP sobre TLS)
- Autenticación de enlace

**Protocolo IMS (Frontend Externo):**

- Mensajería basada en SIP

- Mecanismos de autenticación SIP
- Integración con la seguridad de la red central IMS

**Protocolo SS7/MAP (Frontend Externo):**

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

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

# 2.7 Capacidades de Criptoanálisis y Evaluación de Seguridad

### 2.7.1 Herramientas de Análisis de Protocolo

**Capacidades de Depuración Integradas:**

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

**Integración Externa:**

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

### 2.7.2 Consideraciones de Evaluación de Vulnerabilidades

**Limitaciones Conocidas:**

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

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

**Recomendaciones de Fortalecimiento de Seguridad:**

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

**Pruebas de Seguridad:**

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

# 2.8 Infraestructura de Gestión de Claves

### 2.8.1 Generación de Claves

**Generación de Certificados 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 de confianza
```

**Generación de Números Aleatorios:**

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

### 2.8.2 Almacenamiento y Protección de Claves

**Almacenamiento de Clave Privada:**

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

**Rotación de Claves:**

- Renovación de certificados TLS (anualmente recomendado)
- Rotación de credenciales de base de datos
- Rotación de tokens 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
- Sin transmisión de claves en texto claro

# 2.9 Cumplimiento y Normas

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

### 2.9.1 Cumplimiento del Protocolo SMS sobre SS7/MAP

**Normas 3GPP y ETSI:**

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

**Normas de Señalización SS7:**

- **ITU-T Q.711-Q.716:** Parte de Control de Conexión de Señalización (SCCP)
- **ITU-T Q.771-Q.775:** Parte de Aplicación de Capacidades de Transacción (TCAP)

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

**Normas de Seguridad para SS7/MAP:**

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

**Interceptación Legal para SS7/MAP:**

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

### 2.9.2 Cumplimiento del Protocolo SMS sobre IMS

**Normas IMS de 3GPP:**

- **3GPP TS 23.228:** Subsistema Multimedia IP (IMS) - Arquitectura de etapa 2
- **3GPP TS 24.229:** Protocolo de Control de Llamadas Multimedia IP - Procedimientos SIP y SDP
- **3GPP TS 24.341:** Soporte de SMS sobre redes IP - Especificación de etapa 3
- **3GPP TS 23.204:** Soporte del Servicio de Mensajes Cortos (SMS) sobre acceso IP genérico 3GPP - Etapa 2
- **3GPP TS 29.228:** Interfaces Cx y Dx del Subsistema Multimedia (IM)

**Normas de Seguridad IMS:**

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

**Normas del Protocolo SIP:**

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

**Interceptación Legal para IMS:**

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

### 2.9.3 Cumplimiento del Protocolo SMPP

**Especificación SMPP:**

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

**Directrices de Seguridad SMPP:**

- **TLS sobre SMPP:** Seguridad de la capa de transporte para conexiones SMPP (SMPP sobre TLS)
- **Autenticación de Enlace SMPP:** Autenticación de ID de sistema y contraseña

- **Control de Acceso basado en IP:** Restricciones a nivel de red para conexiones SMPP

**Normas de Interoperabilidad:**

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

**Cumplimiento de Codificación de Mensajes:**

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

### 2.9.4 Cumplimiento de Normas Criptográficas

**TLS y Seguridad de Red:**

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

**Normas de Algoritmos Criptográficos:**

- **FIPS 197:** Estándar de Cifrado Avanzado (AES)
- **FIPS 180-4:** Estándar de Hash Seguro (familia SHA-2)
- **NIST SP 800-38D:** Recomendación para Modos de Operación de Cifrado de Bloques: Modo GCM
- **RFC 7539:** ChaCha20 y Poly1305 para Protocolos IETF

**Gestión de Claves:**

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

# 2.10 Resistencia al Criptoanálisis

### 2.10.1 Principios de Diseño

**Defensa Contra el Criptoanálisis:**

- Sin algoritmos criptográficos personalizados/proprietarios
- Solo algoritmos estándar de la industria revisados por pares
- Actualizaciones de seguridad regulares para bibliotecas criptográficas
- Deprecación de algoritmos débiles
- Uso de cifrado autenticado (GCM, Poly1305)

### 2.10.2 Seguridad Operativa

**Rotación de Claves:**

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

**Monitoreo y Detección:**

- Registro de autenticaciones fallidas
- Monitoreo de expiración de certificados
- Registro de fallos en el apretón de manos TLS
- Detección de anomalías para fallos de cifrado
- Alertas de eventos de seguridad

---

# 3. CONTROL DE INTERCEPTACIÓN Y AUTORIZACIÓN

## 3.1 Control de Acceso para Interceptación Legal

**Autorización Administrativa:**

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

**Integración del Marco Legal:**

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

## 3.2 Retención de Datos y Privacidad

**Políticas de Retención:**

- Retención de mensajes activos: Configurable (24 horas por defecto en Mnesia)
- Retención de CDR: Configurable (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 de usuario/exportaciones

- Cifrado de base de datos en reposo
- Registro y monitoreo de accesos
- Principio de mínima recopilación de datos

**Configuración:**

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

  # Eliminar el cuerpo del mensaje después de la entrega por
privacidad
  delete_message_body_after_delivery: false,  # Establecer 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 todos los ajustes de retención**

## 3.3 Interfaces de Entrega para la Aplicación de la Ley

**Interfaces Estándar:**

**1. Acceso a la API REST:**

- 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 SQL de solo lectura
- Consultas SQL estándar
- Acceso a la tabla CDR
- Capacidades de búsqueda indexada

**3. Exportación por Lotes:**

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

**Formatos de Entrega:**

**IRI (Información Relacionada con la Interceptación):**

- 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 del 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 la aplicación de la ley
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 la VM de Erlang
- Aislamiento y supervisión de procesos
- Recuperación ante fallos y tolerancia a fallos
- Sin vulnerabilidades de desbordamiento de búfer (tiempo de ejecución gestionado)

**Gestión de Dependencias:**

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

- Huella mínima de dependencias

## 4.2 Seguridad de la Red

**Exposición de la Red:**

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

**Segmentación de la Red:**

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

## 4.3 Monitoreo y Detección de Intrusiones

**Capacidades de Registro:**

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

**Monitoreo de Eventos de Seguridad:**

- Intentos de autenticación fallidos
- Patrones de mensajes inusuales
- Fallos de conexión a la base de datos
- Fallos en el apretón de manos TLS

- Anomalías en los recursos del sistema

**Métricas y Alertas:**

- Exportación de métricas de Prometheus
- Monitoreo del 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 de Erlang
- Replicación de Mnesia entre nodos
- Conmutación por error automática
- 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 la API REST
- **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 y monitoreo de Prometheus
- **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 Interceptación 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 de Seguridad Técnica:**

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

**Contacto Legal/Cumplimiento:**

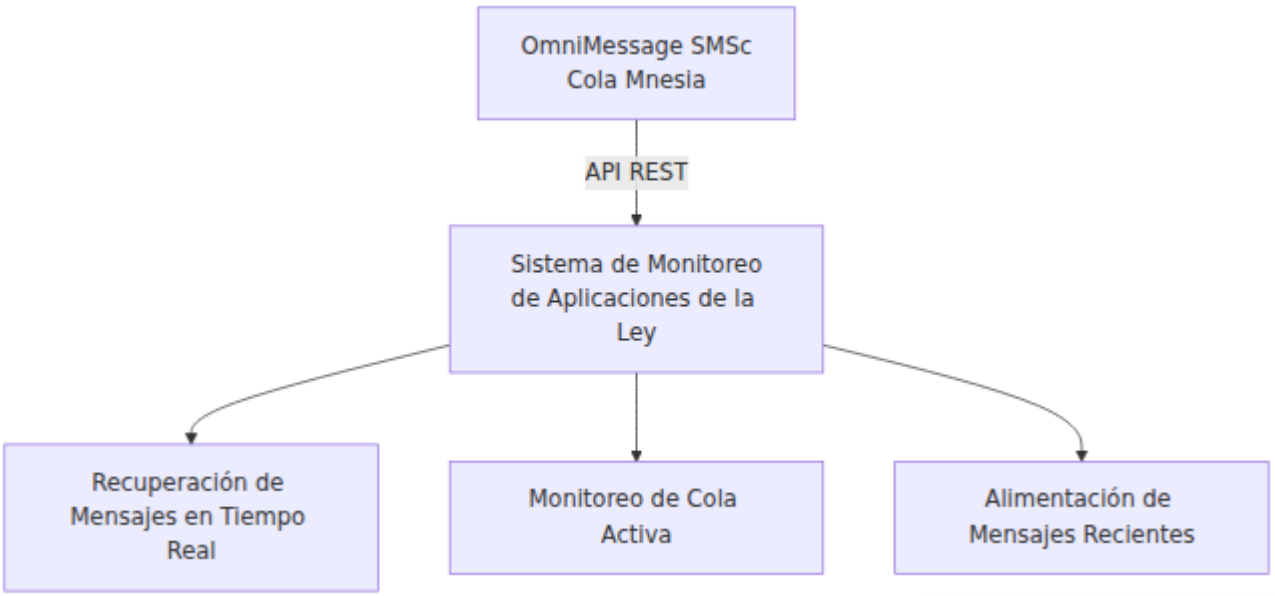
- Nombre: Equipo de Cumplimiento
- Correo Electrónico: compliance@omnitouch.com.au

---

# APÉNDICES

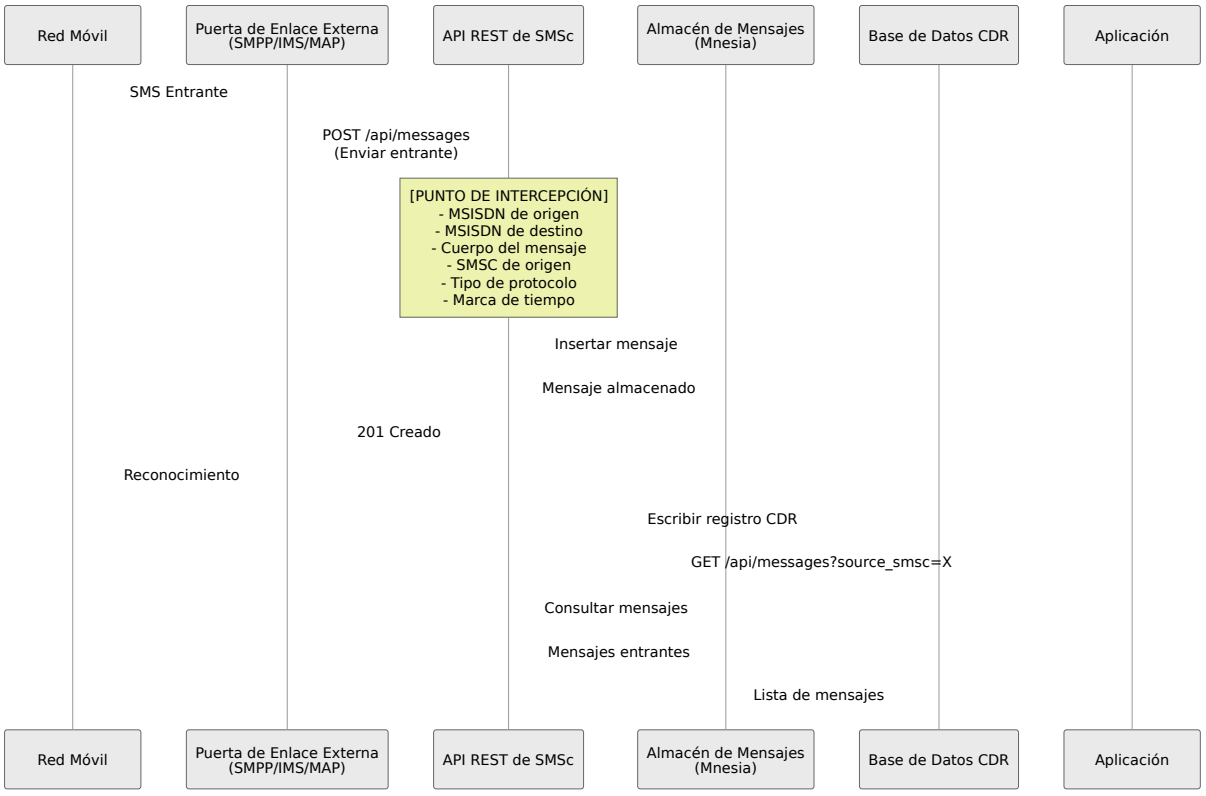
## Apéndice A: Flujo de Mensajes SMS con Puntos de Interceptación

### A.1 Flujo de SMS Saliente (Terminado en Móvil)




**Leyenda:** `[PUNTO DE INTERCEPCIÓN]` = Puntos donde se captura y almacena la información de interceptación legal

## A.2 Flujo de SMS Entrante (Origen Móvil)



# Apéndice B: Esquema CDR para Interceptación Legal

El SMSc OmniMessage almacena Registros de Detalles de Llamadas en una base de datos SQL (MySQL o PostgreSQL) para retención a largo plazo y acceso a la interceptación legal.

### B.1 Campos Clave de CDR para Interceptación Legal

| Nombre del Campo | Tipo | Descripción |
|---|---|---|
| `id` | BIGINT | Clave primaria autoincremental |
| `message_id` | BIGINT | Identificador único del mensaje |
| `calling_number` | VARCHAR(255) | MSISDN de origen |
| `called_number` | VARCHAR(255) | MSISDN de destino |
| `source_smsc` | VARCHAR(255) | Identificador de puerta de enlace de origen |
| `dest_smsc` | VARCHAR(255) | Identificador de puerta de enlace de destino |
| `origin_node` | VARCHAR(255) | Nodo de clúster Erlang (origen) |
| `destination_node` | VARCHAR(255) | Nodo de clúster Erlang (entrega) |
| `submission_time` | DATETIME | Marca de tiempo de envío del mensaj |

| Nombre del Campo | Tipo | Descripción |
|---|---|---|
| `delivery_time` | DATETIME | Marca de tiempo de entrega del mensaje |
| `expiry_time` | DATETIME | Marca de tiempo de expiración del mensaje |
| `status` | VARCHAR(50) | Estado del mensaje (entregado/expirado/fallido/rechazad |
| `delivery_attempts` | INT | Número de intentos de entrega |
| `message_parts` | INT | Número de segmentos de SMS |
| `deadletter` | BOOLEAN | Indicador de cola de cartas muertas |
| `message_body` | TEXT | Contenido del mensaje SMS |
| `inserted_at` | DATETIME | Marca de tiempo de creación del CDR |
| `updated_at` | DATETIME | Marca de tiempo de actualización del CDR |

**Documentación completa del esquema con ejemplos SQL en CDR_SCHEMA.md**

### B.2 Ejemplos de Consultas CDR para Intercepción Legal

**Consultar todos los mensajes para el número objetivo:**

```
SELECT * FROM cdrs
WHERE calling_number = '+33612345678'
   OR called_number = '+33612345678'
ORDER BY submission_time DESC;
```

**Consultar mensajes dentro de la ventana de tiempo:**

```
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;
```

**Exportar a CSV para la aplicación de la ley:**

```
.mode csv
.output /tmp/interception_report.csv
SELECT message_id, calling_number, called_number, message_body,
       submission_time, delivery_time, status
FROM cdrs
WHERE calling_number = '+33612345678'
ORDER BY submission_time DESC;
```

### B.3 Métodos de Acceso a la Base de Datos CDR

**1. Acceso SQL Directo:**

- Credenciales de base de datos de solo lectura
- Consultas SQL estándar
- Conectividad JDBC/ODBC
- Herramientas de cliente de base de datos (MySQL Workbench, pgAdmin)

**2. Acceso a través de API REST:**

- Mejora futura: API REST para consultas CDR
- Respuestas en formato JSON
- Autenticación y autorización
- Filtrado de parámetros de consulta

**3. Exportación por Lotes:**

- Exportación CSV a través de la línea de comandos mysql/psql
- Scripts de exportación automatizados
- Exportaciones programadas a través de cron

### B.4 Retención y Privacidad de CDR

**Configuración de Retención:**

```
# config/runtime.exs
config :sms_c,
  # Eliminar el cuerpo del mensaje después de la entrega (modo de
privacidad)
  delete_message_body_after_delivery: false,  # true para
privacidad

  # Habilitar/deshabilitar escritura de CDR
  cdr_enabled: true,

  # Configuraciones de archivo de CDR a Mnesia
  message_retention_hours: 24
```

**Opciones de Privacidad:**

- El cuerpo del mensaje puede establecerse en NULL después de la entrega
- Cifrado de tabla de base de datos (MySQL ENCRYPTION='Y')
- Restricciones de acceso a nivel de columna
- Exportaciones enmascaradas para análisis

# Apéndice C: Referencia de API REST para Intercepción

## C.1 Puntos de Acceso para Recuperación de Mensajes

**Obtener todos los mensajes:**

```
GET /api/messages
Authorization: Bearer <token>
```

**Obtener mensajes por SMSC:**

```
GET /api/messages/get_by_smsc?smsc=gateway-name
```

**Obtener mensaje específico:**

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

**Formato de Respuesta:**

```
{
  "status": "success",
  "data": [
    {
      "id": 12345,
      "message_id": 12345,
      "source_msisdn": "+33612345678",
      "destination_msisdn": "+33687654321",
      "message_body": "Contenido del mensaje aquí",
      "source_smsc": "ims.gateway",
      "dest_smsc": "smpp.provider",
      "status": "delivered",
      "delivery_attempts": 1,
      "inserted_at": "2025-11-29T10:30:00Z",
      "deliver_time": "2025-11-29T10:30:05Z",
      "expires": "2025-11-30T10:30:00Z"
    }
  ]
}
```


**Documentación completa de la API en API_REFERENCE.md**

### C.2 Filtrado y Búsqueda

**Parámetros de Consulta:**

- `source_smsc` - Filtrar por puerta de enlace de origen
- `dest_smsc` - Filtrar por puerta de enlace de destino
- Filtrado por rango de tiempo (a través de consultas de base de datos CDR)
- Filtrado por estado (a través de consultas de base de datos CDR)

**Mejoras Futuras:**

- API REST para consultas CDR
- Filtrado avanzado por número de teléfono
- Filtrado por rango de fechas
- Búsqueda de texto completo en el cuerpo del mensaje

# Apéndice D: Ejemplos de Configuración

### D.1 Configuración del Certificado TLS

**Generar Certificado TLS:**

```
# Generar clave privada RSA de 4096 bits
openssl genrsa -out priv/cert/omnitouch.pem 4096

# Generar solicitud de firma de certificado
openssl req -new -key priv/cert/omnitouch.pem -out
priv/cert/omnitouch.csr \
  -subj "/C=FR/ST=IDF/L=Paris/O=Omnitouch/CN=smsc.example.com"

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

# Producción: Enviar CSR a CA de confianza para firma
```

**Configurar en la aplicación:**

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

### D.2 Configuración de Cifrado de Base de Datos

**Cifrado de Tabla MySQL:**

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

-- Verificar estado de cifrado
SELECT TABLE_NAME, CREATE_OPTIONS
FROM information_schema.TABLES
WHERE TABLE_SCHEMA = 'sms_c' AND TABLE_NAME = 'cdrs';
```

**Cifrado de Conexión PostgreSQL:**

```
# config/runtime.exs
config :sms_c, SmsC.Repo,
  username: "omnitouch",
  password: "secure_password",
  hostname: "localhost",
  database: "sms_c",
  ssl: true,
  ssl_opts: [
    verify: :verify_peer,
    cacertfile: "/path/to/ca.crt",
    certfile: "/path/to/client.crt",
    keyfile: "/path/to/client.key"
  ]
```

### D.3 Configuración de Privacidad

**Habilitar Eliminación del Cuerpo del Mensaje:**

```
# 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
```

**Crear Vista de Base de Datos que Preserva la Privacidad:**

```
-- Crear vista sin cuerpos de mensaje para análisis general
CREATE VIEW cdrs_metadata AS
SELECT
  id, message_id, calling_number, called_number,
  source_smsc, dest_smsc, origin_node, destination_node,
  submission_time, delivery_time, expiry_time,
  status, delivery_attempts, message_parts, deadletter,
  inserted_at, updated_at
FROM cdrs;

-- Conceder acceso a usuarios de análisis
GRANT SELECT ON cdrs_metadata TO 'analytics'@'%';
```

# Apéndice E: Glosario

### Organismos Regulatorios y Estándares

- **ANSSI:** Agence nationale de la sécurité des systèmes d'information - Agencia Nacional de Ciberseguridad de Francia
- **ETSI:** European Telecommunications Standards Institute
- **3GPP:** 3rd Generation Partnership Project - Estándares de telecomunicaciones móviles
- **IETF:** Internet Engineering Task Force - Organismo de estándares de Internet

### Términos de Telecomunicaciones

- **SMSc:** SMS Service Center - Sistema central para el enrutamiento y entrega de mensajes SMS
- **SMPP:** Short Message Peer-to-Peer protocol - Estándar de la industria para el intercambio de SMS
- **ESME:** External Short Message Entity - Aplicación cliente SMPP
- **IMS:** IP Multimedia Subsystem - Arquitectura de red All-IP para servicios multimedia
- **SIP:** Session Initiation Protocol - Protocolo de señalización para mensajería IMS

- **P-CSCF:** Proxy Call Session Control Function - Punto de entrada de la red IMS
- **S-CSCF:** Serving Call Session Control Function - Control de sesión IMS
- **HSS:** Home Subscriber Server - Base de datos de suscriptores IMS
- **SS7/MAP:** Signaling System 7 / Mobile Application Part - Protocolos de señalización móviles heredados
- **MSC:** Mobile Switching Center - Elemento de red conmutada por circuitos
- **VLR:** Visitor Location Register - Base de datos de ubicación de suscriptores
- **GT:** Global Title - Esquema de direccionamiento SS7
- **SCCP:** Signaling Connection Control Part - Capa de red SS7
- **TCAP:** Transaction Capabilities Application Part - Capa de aplicación SS7
- **MSISDN:** Mobile Station International Subscriber Directory Number - Número de teléfono
- **IMSI:** International Mobile Subscriber Identity - Identificador único de suscriptor
- **E.164:** Plan de numeración internacional para números de teléfono
- **ENUM:** E.164 Number Mapping - Mapeo de número de teléfono basado en DNS a URI
- **PDU:** Protocol Data Unit - Mensaje SMS codificado en binario
- **TP-DCS:** Transfer Protocol Data Coding Scheme - Especificación de codificación de SMS
- **TP-DU:** Transfer Protocol Data Unit - Formato PDU específico de SMS
- **UDH:** User Data Header - Encabezado para mensajes SMS concatenados/especiales
- **TON/NPI:** Type of Number / Numbering Plan Indicator - Clasificación de formato de número
- **GSM7:** GSM 7-bit default alphabet - Codificación de caracteres SMS estándar
- **UCS-2:** Universal Character Set 2-byte - Codificación Unicode para SMS

**Componentes del Sistema**

- **Mnesia:** Sistema de base de datos distribuido de Erlang - Almacenamiento en memoria/disco
- **CDR:** Call Detail Record - Registro de facturación y análisis para mensajes

- **REST API:** Representational State Transfer - Arquitectura de API basada en HTTP
- **Phoenix:** Framework web de Elixir
- **Cowboy:** Servidor HTTP de Erlang
- **Ecto:** Envoltura de base de datos y lenguaje de consulta de Elixir
- **PubSub:** Patrón de mensajería Publicar-Suscribir

### Intercepción Legal

- **LI:** Lawful Interception - Monitoreo legal de telecomunicaciones
- **LIMF:** Lawful Interception Mediation Function - Sistema que intermedia entre la red de telecomunicaciones y la aplicación de la ley
- **LEMF:** Law Enforcement Monitoring Facility - Sistema de aplicación de la ley que recibe datos interceptados
- **IRI:** Intercept Related Information - Metadatos de llamada/mensaje para la aplicación de la ley
- **CC:** Content of Communication - Contenido real del mensaje
- **HI1 Interface:** ETSI LI administrative handover interface - Provisión de órdenes y activación de objetivos
- **HI2 Interface:** ETSI LI handover interface for IRI delivery - Transferencia de metadatos a la aplicación de la ley
- **HI3 Interface:** ETSI LI handover interface for CC delivery - Transferencia de contenido a la aplicación de la ley
- **R226:** Artículos R226-3 y R226-7 del Código Penal Francés que rigen el equipo de intercepción
- **ETSI:** European Telecommunications Standards Institute - Define estándares de LI
- **ETSI TS 102 232:** Especificación técnica para interfaces de transferencia de intercepción legal

### Procesamiento de Mensajes

- **MT:** Mobile Terminated - Mensaje saliente a suscriptor móvil
- **MO:** Mobile Originated - Mensaje entrante de suscriptor móvil
- **DLR:** Delivery Receipt - Confirmación de entrega del mensaje

- **Dead Letter:** Mensaje que falló en la entrega después de todos los intentos de reintento
- **Exponential Backoff:** Retraso de reintento creciente (2min, 4min, 8min, etc.)

### Seguridad y Cifrado

- **TLS:** Transport Layer Security - Protocolo de cifrado
- **PFS:** Perfect Forward Secrecy - Propiedad criptográfica para la seguridad de la clave de sesión
- **AES:** Advanced Encryption Standard
- **RSA:** Rivest-Shamir-Adleman - Criptografía de clave pública
- **ECDSA:** Elliptic Curve Digital Signature Algorithm
- **SHA:** Secure Hash Algorithm
- **X.509:** Estándar de certificados
- **CA:** Certificate Authority
- **CSPRNG:** Generador de Números Pseudoaleatorios Criptográficamente Seguros

### Términos de Base de Datos

- **MySQL:** Base de datos relacional de código abierto
- **PostgreSQL:** Base de datos objeto-relacional de código abierto
- **TDE:** Transparent Data Encryption
- **RBAC:** Role-Based Access Control

---

**Versión del Documento:** 1.0
**Fecha:** 2025-11-29
**Preparado para:** Solicitud de Autorización ANSSI R226
**Clasificación del Documento:** Cumplimiento Regulatorio - Confidencial

# Mensajería Masiva / Campañas

← Volver al Índice de Documentación

## Descripción General

La función de campañas envía un solo mensaje de texto en formato plano a muchos destinatarios mientras protege el SMSC de sobrecargas. Subes una **lista de destinatarios** nombrada una vez, luego ejecutas una o más **campañas** contra ella. Cada campaña gotea mensajes en la cola de mensajes normal a una tasa configurable y reporta el progreso en vivo y las estadísticas de entrega.

Dos objetos reutilizables:

| Objeto | Qué es |
|---|---|
| **Lista de campañas** | Un conjunto nombrado y reutilizable de MSISDNs de destinatarios, cada uno con hasta cuatro variables opcionales. Sube una vez (arreglo JSON o CSV), reutiliza en campañas. |
| **Campaña** | Una ejecución de envío: una lista (o *todos los suscriptores activos*) + una plantilla de mensaje, goteada en la cola. |

Todo está expuesto a través de la API HTTP. Todo el estado se almacena en Mnesia junto con el resto del SMSC.

# Segmentación: listas y el predeterminado de suscriptores activos

Los destinatarios de una campaña provienen de una de dos fuentes:

- **Una lista de destinatarios** (`list_id`) - los MSISDNs que subiste.
- **Sin lista en absoluto** - si creas una campaña **sin** un `list_id`, se dirige a **todos los suscriptores actualmente activos (registrados)** - cada MSISDN con una inscripción no expirada en el almacén de ubicaciones en el momento en que comienza la campaña. Esta es la forma más fácil de llegar a "todos en la red en este momento".

# Variables por destinatario y plantillas

Cada destinatario de la lista puede llevar hasta cuatro variables - **`var1`**, **`var2`**, **`var3`**, **`var4`** - para cosas como nombre, apellido o nombre del plan. El mensaje de la campaña es una **plantilla**: `{{ var1 }}`…`{{ var4 }}` (y `{{ msisdn }}`) se insertan para cada destinatario cuando se envía el mensaje. El espacio en blanco dentro de las llaves es opcional, y cualquier marcador de posición sin valor se renderiza como una cadena vacía.

> La plantilla es deliberadamente mínima - solo sustitución de variables (sin condicionales ni filtros), por lo que no hay dependencia adicional y ninguna sorpresa en lo que reciben los suscriptores.

**Subiendo variables a través de CSV** - la columna MSISDN es la que se llama `msisdn`/`destination_msisdn`/`number`/`phone` (o la primera columna si ninguna está nombrada); cada otra columna se convierte en `var1`…`var4` en orden de columna:

```
msisdn,first,last,plan
12025550101,Alice,Smith,Gold
12025550102,Bob,Jones,Silver
```

**Subiendo variables a través de JSON** - los objetos de destinatarios pueden establecer las vars directamente:

```
{ "name": "VIPs", "recipients": [
  { "msisdn": "12025550101", "var1": "Alice", "var2": "Gold" }
] }
```

**Ejemplo de plantilla**:

```
Hola {{ var1 }}, tu plan {{ var2 }} se renueva pronto. Responde
STOP para darte de baja.
```

Para el destinatario `12025550101` anterior (`var1=Alice`, `var2=Gold`) esto envía: `Hola Alice, tu plan Gold se renueva pronto. Responde STOP para darte de baja.`

# Filtro de audiencia

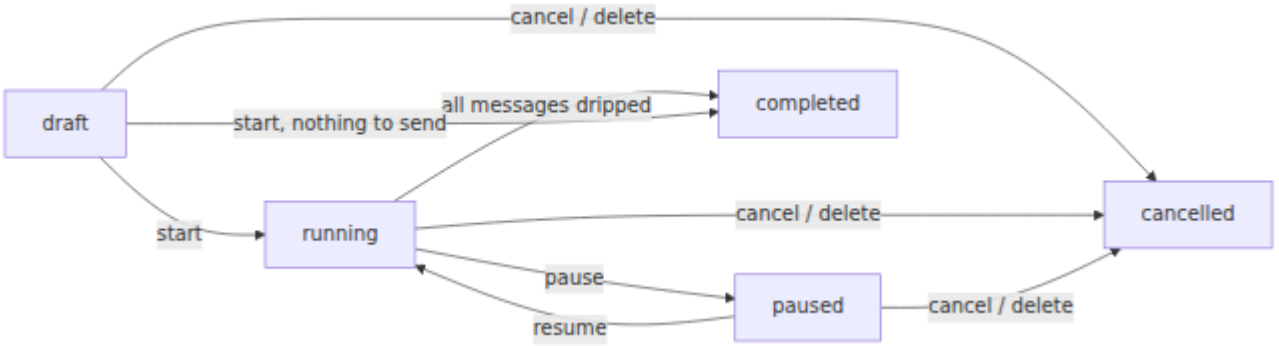
Cuando una campaña utiliza una lista, la bandera `audience` decide quién recibe realmente un mensaje:

| Audiencia | Comportamiento |
|---|---|
| `active_only` (predeterminado) | Solo miembros de la lista que están **registrados** (una ubicación no expirada) en el momento en que comienza la campaña. Todos los demás se registran como `skipped_inactive` y nunca se envían. |
| `all` | Cada miembro de la lista, independientemente de la inscripción. |

La verificación de registro es una **instantánea tomada al inicio** - los suscriptores que se registran más tarde no son recogidos. Cuando una campaña **no tiene lista**, el conjunto de destinatarios ya es la base de

suscriptores activos, por lo que `audience` es implícitamente `active_only` y cualquier valor proporcionado es ignorado.

# Ciclo de Vida




1. **draft** - creado, sin objetivos aún.
2. **start** - la campaña toma instantáneas de sus objetivos (de su lista o de la base de suscriptores activos), luego se mueve a **running**. Si no hay nada que enviar, pasa directamente a **completed**.
3. **running** - el trabajador de goteo envía mensajes a la cola.
4. **pause / resume** - detener temporalmente / continuar goteando.
5. **completed** - cada objetivo ha sido goteado en la cola (la entrega luego continúa de forma asíncrona, y las estadísticas de entrega siguen actualizándose).
6. **cancelled** - detenido permanentemente.

## Eliminando una campaña

Eliminar una campaña **la detiene pero mantiene el registro** - se cancela en lugar de purgarse, por lo que su historia y estadísticas finales permanecen visibles. Las campañas detenidas y completadas se eliminan automáticamente una vez que superan el tiempo de retención TTL (`campaign_retention_hours`, predeterminado 7 días).

# Tasa de Goteo (protección contra sobrecarga)

Cada campaña tiene un `drip_tps` (mensajes enviados a la cola por segundo, predeterminado de la configuración). Un trabajador en segundo plano se activa en un intervalo fijo y envía como máximo `drip_tps × interval` mensajes por campaña por tick - así que una campaña de un millón de destinatarios nunca inunda la cola. La tasa es **por campaña**; múltiples campañas en ejecución gotean a su propia tasa.

# Progreso y estadísticas de entrega

Cada objeto de campaña incluye un bloque `stats` en vivo (consulta `GET /api/campaigns/:id`):

| Campo | Significado |
|---|---|
| `total` | Objetivos tomados al inicio |
| `pending` | Aún no goteados en la cola |
| `queued` | Enviados, esperando un informe de entrega |
| `delivered` | Confirmado entregado |
| `failed` | Entrega fallida / enviada a la cola de mensajes muertos |
| `skipped_inactive` | Omitido porque no estaba registrado al inicio |
| `dispatched` | `queued + delivered + failed` |
| `progress_percent` | Porcentaje de objetivos que ya no están `pending` |
| `delivery_percent` | Porcentaje de mensajes enviados entregados |

Los conteos de entrega y fallas se actualizan automáticamente a medida que los informes de entrega fluyen de regreso a través de la ruta de entrega normal del SMSC. Detalles por destinatario (y filtrado por estado, por ejemplo, todos los que fallaron) están disponibles a través de `GET /api/campaign_targets/:id?state=failed`.

# Configuración

```
config :sms_c, :campaigns,
  # Iniciar el trabajador de goteo (establecer en falso para
pausar TODA la entrega de campañas)
  enabled: true,
  # Intervalo entre ticks de goteo en milisegundos
  drip_tick_ms: 1000,
  # Mensajes por campaña por segundo cuando una campaña no
establece su propio drip_tps
  default_drip_tps: 50,
  # Ventana de validez del mensaje predeterminada cuando una
campaña no establece validity_hours
  message_validity_hours: 72,
  # Cuánto tiempo persiste una campaña detenida/completada antes
de ser purgada (horas)
  campaign_retention_hours: 168
```

# Validez y entrega programada

Una campaña puede establecer, en el momento de la creación:

- **`validity_hours`** - el período de validez del mensaje. Cada mensaje enviado se marca con un `expires` para que los mensajes no entregados caduquen en lugar de permanecer en la cola. Por defecto, se establece en `message_validity_hours` de la configuración.
- **`deliver_after`** - un tiempo ISO8601 opcional (debe estar en el futuro) para un envío programado.

**Cómo la entrega programada interactúa con el goteo.** La campaña aún comienza a gotear *inmediatamente* a `drip_tps`, pero el reloj de cada mensaje

se adelanta por un desplazamiento fijo `delta = deliver_after − started_at`. Así que un mensaje goteado en tiempo real *t* se marca `send_time`/`deliver_after = t + delta` y `expires = t + delta + validity_hours`. Debido a que los mensajes se insertan a `drip_tps` en tiempo real, sus valores `deliver_after` por mensaje salen **escalonados a `drip_tps` comenzando en `deliver_after`** - la entrega está limitada por la tasa en lugar de ser un aluvión en el instante programado. Sin `deliver_after`, `delta = 0` y los mensajes se gotean y entregan inmediatamente.

# Flujo de trabajo típico

```
# 1. Subir una lista de destinatarios nombrada con variables por dest
(CSV o JSON)
curl -k -X POST https://smsc:8443/api/campaign_lists \
  -H 'Content-Type: application/json' \
  -d '{"name":"Clientes VIP",

"csv":"msisdn,first,plan\n12025550101,Alice,Gold\n12025550102,Bob,Sil
# => {"id": 1, "recipient_count": 2, ...}

# 2. Crear una campaña contra esa lista, plantillando en las variable
curl -k -X POST https://smsc:8443/api/campaigns \
  -H 'Content-Type: application/json' \
  -d '{"name":"Aviso de renovación","message":"Hola {{ var1 }}, tu pl
var2 }} se renueva pronto.",
       "source_msisdn":"12345","source_smsc":"IMS_SMSC",
       "list_id":1,"audience":"active_only","drip_tps":50}'
# => {"id": 7, "status": "draft", ...}

# 3. Iniciarlo
curl -k -X POST https://smsc:8443/api/campaigns/control \
  -H 'Content-Type: application/json' \
  -d '{"campaign_id":7,"action":"start"}'

# 4. Consultar progreso / estadísticas de entrega
curl -k https://smsc:8443/api/campaigns/7
# => {... "stats":
{"total":..,"delivered":..,"failed":..,"progress_percent":..}}
```

Para enviar mensajes a **todos los suscriptores activos**, omite `list_id` en el paso 2 (y salta completamente el paso 1).

# Véase también

- Referencia de la API de Campañas
- Guía de Enrutamiento de SMS - cómo se enrutan/entregan los mensajes goteados
- Referencia de Configuración

# Referencia del Esquema CDR (Registro de Detalles de Llamadas)

**← Volver al Índice de Documentación**

Referencia completa para la tabla de base de datos CDR utilizada para el almacenamiento de mensajes a largo plazo, 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 que el sistema procesa. El sistema escribe un CDR en el momento en que un mensaje alcanza un estado terminal: `delivered`, `expired`, `failed` o `rejected`. El sistema registra un `balance_rejected` fuera de balance como `rejected`. Un barrido de tiempo de retención actúa como un respaldo

para cualquier cosa que envejezca fuera de Mnesia sin una escritura activa. Consulte Cuándo se Escriben los CDRs para la ruta de escritura completa y las garantías de idempotencia.

Los CDRs 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
  UNIQUE KEY cdrs_origin_node_message_id_index (origin_node,
```

```
message_id),
  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)
);
```

> La clave única `(origin_node, message_id)` le da a cada CDR una identidad única a nivel global en el clúster. También hace que las escrituras de CDR sean idempotentes. La escritura casi en tiempo real y el respaldo de retención pueden intentar ambas un inserto. El segundo inserto es un no-op a través de `ON CONFLICT DO NOTHING`.

## PostgreSQL

```sql
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 UNIQUE INDEX cdrs_origin_node_message_id_index ON
cdrs(origin_node, message_id);
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 del mensaje de la cola de mensajes SMS-C. Generado por un contador **por nodo** (persistido en Mnesia, por lo que sobrevive a reinicios), por lo tanto, solo es único **dentro de un solo nodo**. Para una clave única a nivel global en un clúster de múltiples nodos, use `(origin_node, message_id)` juntos - este par se hace cumplir mediante un índice único y es la clave de unión/deduplicación recomendada para sistemas de facturación. |

## 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 análisis de distribución de carga. |
| `destination_node` | VARCHAR(255) | SÍ | Nombre del nodo Erlang desde donde se entregó el mensaje (si es diferente del origen). NULL para implementaciones de un solo nodo o si el mensaje nunca fue entregado. |

## Tiempos

Todos los tiempos se almacenan en UTC:

| Campo | Tipo | Nulo | Descripción |
| --- | --- | --- | --- |
| `submission_time` | DATETIME | NO | Cuando el mensaje fue enviado por primera vez al SMS-C. Se utiliza como el 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 sistema movió el mensaje a la cola de cartas muertas. TRUE indica que el sistema no pudo entregar el mensaje y agotó 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 fue actualizado. |

# Ejemplos de SQL

## Consultas Básicas

**Encontrar todos los CDRs para un número de teléfono específico:**

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

**Contar mensajes por estado:**

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

**Tiempo promedio de entrega para mensajes entregados:**

```sql
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:**

```sql
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):**

```sql
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:**

```sql
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):**

```sql
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:**

```sql
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:**

```sql
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 la política de retención (eliminar CDRs antiguos):**

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

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

# Análisis de Clúster

**Distribución de mensajes entre nodos:**

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

# Índices

Los siguientes índices se crean para optimizar consultas comunes:

| Nombre del Índice | Columnas | Propósito |
| --- | --- | --- |
| `PRIMARY` | `id` | Clave primaria, asegura registro único |
| `cdrs_origin_node_message_id_index` | `origin_node, message_id` | **Único.** Identidad CDR única a nivel global en el clúster; hace que las escrituras sean idempotentes |
| `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 |

| Nombre del Índice | Columnas | Propósito |
| --- | --- | --- |
| `idx_cdrs_submission_time` | `submission_time` | Consultas basadas en tiempo, períodos de facturación |
| `idx_cdrs_dest_smsc` | `dest_smsc` | Análisis de rendimiento de rutas |

## Recomendaciones de Índices Adicionales

Para implementaciones de alto volumen, considere estos índices adicionales:

**Índice compuesto para consultas de facturación:**

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

**Índice compuesto para análisis de rutas:**

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

**Índice compuesto para búsquedas de cumplimiento:**

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

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

```sql
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 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 almacena como 0/1 |

# Consideraciones de Privacidad

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

## 1. Privacidad del Cuerpo del Mensaje

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

```
config :sms_c,
  # Eliminar el cuerpo del mensaje después de la entrega exitosa
  delete_message_body_after_delivery: true,

  # Ocultar el cuerpo del mensaje en la interfaz de usuario
  hide_message_body_in_ui: true,

  # Ocultar el cuerpo del mensaje en las exportaciones
  hide_message_body_in_export: true
```

## 2. Enmascaramiento de Números de Teléfono

Para análisis que no requieren números completos:

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

## 3. 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 (TDE) de PostgreSQL o cifrado a nivel de sistema de archivos.

## 4. Controles de Acceso

Restringir el acceso a la tabla CDR:

```
-- Crear usuario de facturación de solo lectura
CREATE USER 'billing_ro'@'%' IDENTIFIED BY 'secure_password';
GRANT SELECT ON sms_c.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 |
|---|---|---|
| Telecom (EE. UU.) | 18-24 meses | FCC, leyes estatales |
| Telecom (UE) | 6 meses - 2 años | GDPR, ePrivacy |
| Financiera | 5-7 años | SOX, SEC |
| Salud | 6 años | HIPAA |

## Estrategia de Archivo

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

```sql
-- 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**

```sql
-- Exportar CDRs 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 Tarifas

Crear una tabla de tarifas separada para la facturación:

```sql
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 CDRs 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;
```

## Exportación para Sistemas de Facturación

**Exportación CSV incorporada (recomendada):**

El SMS-C puede renderizar CDRs a un archivo CSV RFC 4180 directamente, sin pasar por el cliente SQL. En una versión, impulsarlo desde la consola remota:

```
bin/omnimessage rpc
'SmsC.Messaging.CDR.export_csv_to_file("/tmp/cdrs-202607.csv",
since: ~U[2026-07-01 00:00:00Z], until: ~U[2026-08-01 00:00:00Z],
status: :delivered)'
```

Opciones: `:since`, `:until` (ambos `DateTime`), `:status` (un átomo/cadena o una lista, por ejemplo, `[:delivered, :failed]`), y `:limit`. El valor devuelto es `{:ok, {path, row_count}}`. La columna `message_body` respeta la configuración `hide_message_body_in_export` - configúrelo en `true` para emitir `[REDACTED]` en lugar del contenido del mensaje. Una función de exportación complementaria devuelve el mismo CSV como una cadena si prefiere transmitirlo a otro lugar.

**Exportación directa del cliente SQL (alternativa):**

```
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
```

# Véase También

- Guía de Configuración - Configurar opciones de exportación de CDR
- Guía de Operaciones - Procedimientos de mantenimiento de CDR
- Referencia de API - Consultar CDRs a través de la API REST

# Referencia de Configuración de SMS-C

**← Volver al Índice de Documentación**

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

## Tabla de Contenidos


- Archivos de Configuración
- Configuración de Base de Datos
- Configuración de API
- Configuración de Interfaz Web
- Configuración de Federación
- Configuración de Cola de Mensajes
- Configuración de Carga
- Configuración de Diameter Sh (HSS)
- Configuración de ENUM
- Configuración de Traducción de Números
- Intercepción Legal (X1/X2/X3)
- Configuración de Enrutamiento
- Configuración de Ajuste de Rendimiento
- Configuración de Registro
- Escenarios Comunes de Configuración


## Archivos de Configuración

El SMS-C utiliza tres archivos de configuración principales:

## config/config.exs

Configuración estática cargada en tiempo de compilación. Contiene:

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

## config/runtime.exs

Configuración en tiempo de ejecución cargada al inicio. Contiene:

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

## config/prod.exs (opcional)

Sobrescrituras específicas de producción.

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

# Configuración de Almacenamiento SQL CDR

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

# Bases de Datos SQL Soportadas

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

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

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

# Cuándo se Escriben los CDR

Los CDR se escriben en la base de datos SQL en dos momentos:

1. **Casi en tiempo real, en estado terminal** - en el momento en que un mensaje alcanza un resultado final (`delivered`, `expired`, `failed`, `rejected`, o `balance_rejected`) se escribe un CDR de forma asíncrona. Este es el camino principal y mantiene la facturación actual sin esperar la retención de mensajes.
2. **Respaldo de tiempo de retención** - cuando el trabajador de retención elimina un mensaje expirado de Mnesia, primero exporta un CDR. Registra

`expired` para cualquier cosa que nunca alcanzó un estado terminal. El trabajador solo elimina un mensaje una vez que su exportación de CDR tiene éxito. No se pierde nada si la base de datos de CDR está brevemente no disponible: el trabajador reintenta la exportación en la siguiente barrida.

Ambos caminos son idempotentes. Cada CDR utiliza el par único `(origin_node, message_id)` como su clave. Las escrituras utilizan `ON CONFLICT DO NOTHING`. Por lo tanto, el respaldo nunca duplica un mensaje que la escritura casi en tiempo real ya cubrió.

Un contador por nodo genera `message_id` (persistido en Mnesia, por lo que sobrevive a reinicios). `message_id` es, por lo tanto, único **dentro de un solo nodo**. Cuando varios nodos SMS-C escriben en una base de datos CDR compartida, `(origin_node, message_id)` juntos son la clave globalmente única. Consulta la Referencia del Esquema CDR. Establece `cdr_enabled: false` para deshabilitar completamente la escritura de CDR.

## Exportación CSV / Archivo de Texto

Los CDR se pueden exportar de la base de datos SQL a un archivo CSV RFC 4180. En una versión (donde `mix` no está disponible) se puede ejecutar desde la consola remota:

```
bin/omnimessage rpc
'SmsC.Messaging.CDR.export_csv_to_file("/tmp/cdrs-202607.csv",
since: ~U[2026-07-01 00:00:00Z], until: ~U[2026-08-01 00:00:00Z])'
```

Opciones soportadas: `:since`, `:until` (ambos `DateTime`), `:status` (un átomo/cadena o lista), y `:limit`. La columna `message_body` respeta la configuración `hide_message_body_in_export` (emitida como `[REDACTED]` cuando está habilitada).

## Configuración de MySQL / MariaDB

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

## Configuración de PostgreSQL

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

## Elegir una Base de Datos SQL

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

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

**PostgreSQL** - Considera si necesitas:

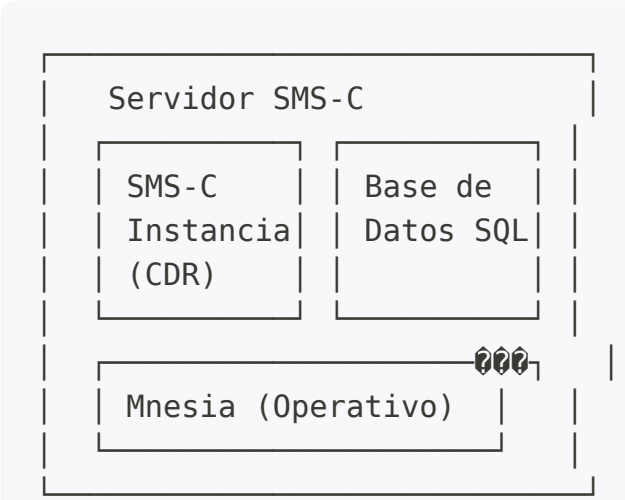
- Características avanzadas de JSON/JSONB para análisis

- Consultas complejas sobre datos de CDR
- Infraestructura PostgreSQL existente
- PostGIS para análisis geográfico

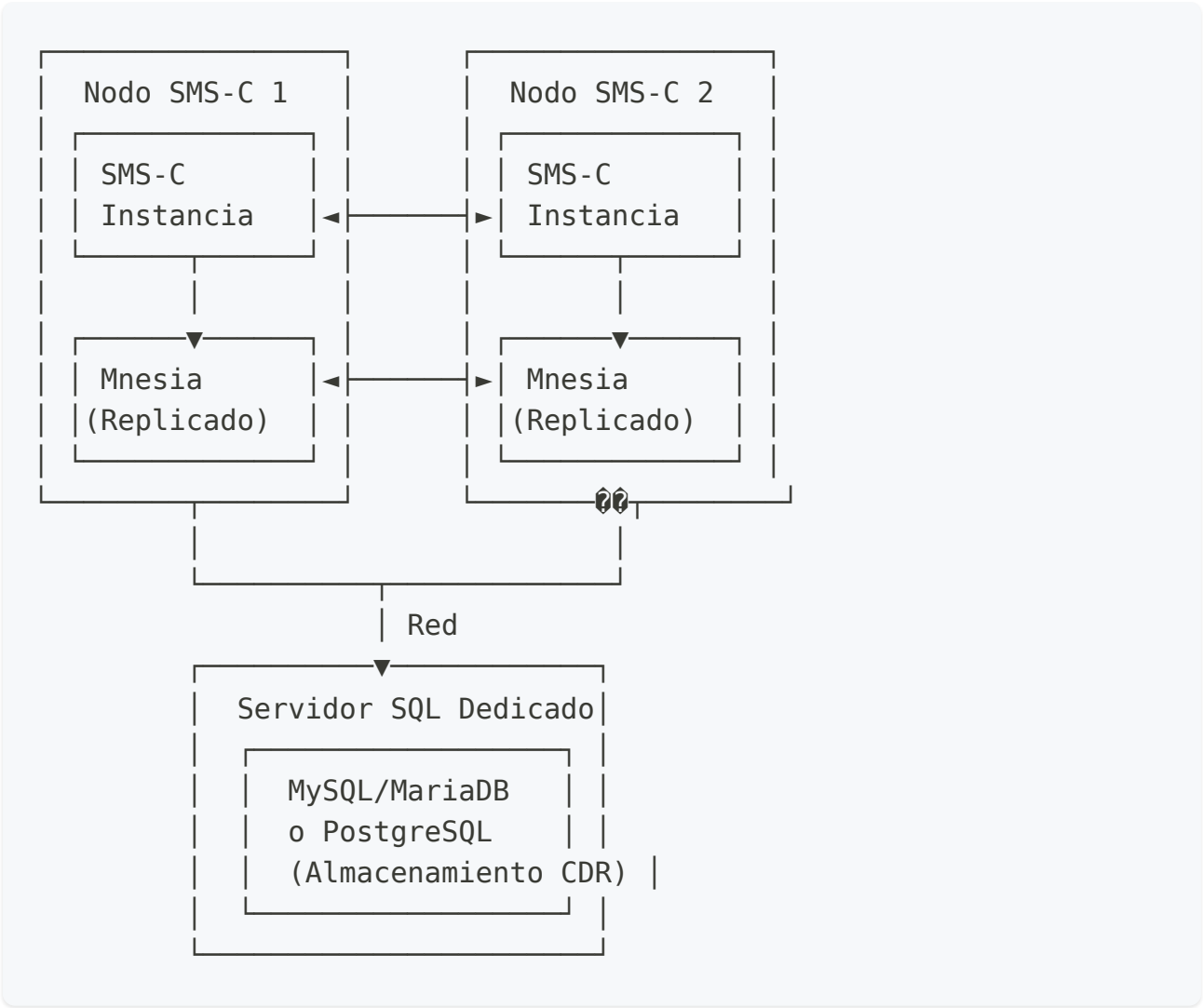
## Topologías de Implementación

**Importante**: La base de datos SQL CDR puede ejecutarse en un **servidor separado** de tus instancias SMS-C. Este es el enfoque recomendado para implementaciones en producción.

**Implementación en un Solo Servidor** (Desarrollo/Pruebas):




**Implementación Distribuida** (Producción - Recomendado):




**Beneficios de un Servidor SQL Separado**:

- **Aislamiento de Rendimiento**: Las escrituras de CDR no afectan el procesamiento de mensajes
- **Escalabilidad**: Escalar independientemente la base de datos y el procesamiento de mensajes
- **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**: `pool_size = (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

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

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

Alerta si el tiempo de espera supera consistentemente 100ms - 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,
  api: %{
    port: 8443,
    listen_ip: "0.0.0.0",
    enable_tls: true,
    tls_cert_path: "priv/cert/omnitouch.crt",
    tls_key_path: "priv/cert/omnitouch.pem"
  }
```

La configuración de la API vive bajo el mapa `:api` de la aplicación `:api_ex`. La tabla de rutas (`routes:`) se define en tiempo de compilación en

`config/config.exs` y no se edita aquí.

| Parámetro | Tipo | Requerido | Predeterminado | De |
|---|---|---|---|---|
| `port` | Entero | No | `8443` | Pu pa RE |
| `listen_ip` | Cadena | No | `0.0.0.0` | Di en la `0.` pa las int |
| `enable_tls` | Booleano | No | `true` | Se a t HT |
| `tls_cert_path` | Cadena | Si `enable_tls` | `priv/cert/omnitouch.crt` | Ru ce TL |
| `tls_key_path` | Cadena | Si `enable_tls` | `priv/cert/omnitouch.pem` | Ru cla pri |

## Configuración TLS/SSL

**Configuración de Producción con TLS** (Recomendado):

```
config :api_ex,
  api: %{
    port: 8443,
    listen_ip: "0.0.0.0",
    enable_tls: true,
    tls_cert_path: "priv/cert/omnitouch.crt",
    tls_key_path: "priv/cert/omnitouch.pem"
  }
```

**Configuración de Desarrollo sin TLS**:

```
config :api_ex,
  api: %{
    port: 8080,
    listen_ip: "127.0.0.1",
    enable_tls: false
  }
```

# Configuración del Certificado de API

Genera 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, utiliza certificados de una CA confiable (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 con Clave de API** (Nivel de Aplicación):

Configura a través de un plug personalizado en el enrutador - consulta 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, ControlPanelWeb.Endpoint,
  https: [
    port: 443,
    keyfile: "priv/cert/omnitouch.pem",
    certfile: "priv/cert/omnitouch.crt"
  ]
```

La interfaz web es un endpoint de Phoenix. TLS se configura a través de la lista de palabras clave `https:` del endpoint (no `tls_cert_path`/`tls_key_path`), y HTTPS se aplica obligatoriamente a través de `force_ssl` en `config/config.exs`.

| Parámetro | Tipo | Requerido | Predeterminado | Descrip |
|---|---|---|---|---|
| `port` | Entero | No | `443` | Puerto HTTPS p el panel control. |
| `keyfile` | Cadena | Sí | `priv/cert/omnitouch.pem` | Ruta a la clave privada |
| `certfile` | Cadena | Sí | `priv/cert/omnitouch.crt` | Ruta al certifica TLS. |

## Configuración de Producción de Interfaz Web

```
config :control_panel, ControlPanelWeb.Endpoint,
  url: [host: "sms-admin.example.com", path: "/"],
  https: [
    port: 443,
    keyfile: "priv/cert/web.pem",
    certfile: "priv/cert/web.crt"
  ]
```

## Configuración de Proxy Inverso (Recomendado)

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

**Ejemplo de Configuración de Nginx**:

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

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

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

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

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

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

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

```
    proxy_read_timeout 86400;
  }
}
```

# Configuración de Federación

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

Consulta la **Guía de Federación Geográfica** para la arquitectura completa, ejemplos de implementación y solución de problemas.

## Inicio Rápido

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

## Referencia Completa de Parámetros

| Parámetro | Tipo | Predeterminado | Descripció |
|---|---|---|---|
| `enabled` | Booleano | `false` | Interruptor maestro para la federación |
| `dns_srv_domain` | Cadena | `""` | Dominio SRV DNS para descubrimien de pares. |
| `dns_poll_interval_ms` | Entero | `30000` | Intervalo de resolución DNS. |
| `health_check_interval_ms` | Entero | `10000` | Intervalo de verificación d salud de par |
| `registry_sync_interval_ms` | Entero | `15000` | Intervalo de intercambio registro de frontend. |
| `forward_retry_interval_ms` | Entero | `5000` | Intervalo de reintento de mensajes en cola. |
| `forward_max_retries` | Entero | `50` | Máximos intentos de reenvío ante de marcar como fallido. |

| Parámetro | Tipo | Predeterminado | Descripció |
|---|---|---|---|
| `http_timeout_ms` | Entero | `5000` | Tiempo de espera para todas las llamadas HT a pares. |
| `api_port` | Entero | `8443` | Puerto de AP para constru URLs de pare |
| `static_peers` | Lista | `[]` | Lista de pare estática (alternativa a DNS SRV). |

## Requisitos de Red

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

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

# Configuración de Cola de Mensajes

Controla el comportamiento de retención y expiración de mensajes.

## Expiración de Mensajes

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

**Valores Comunes**:

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

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

## Configuración de Reintento de Entrega

El comportamiento de reintento utiliza retroceso exponencial:

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

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

Máximos intentos antes de carta muerta: Limitado por `dead_letter_time_minutes`.

## Configuración de Limpieza

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

**Intervalos de Limpieza**:

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

# Configuración de Carga

La carga en línea se realiza a través de la interfaz **Diameter Ro** (Aplicación de Control de Crédito, RFC 4006). Para cada mensaje cargado, el SMS-C envía un Credit-Control-Request (Event-Request, `CC-Request-Type 4`) al OCS para el **remitente** del mensaje; el OCS autoriza o rechaza según el saldo del suscriptor. Esto reutiliza la pila Diameter (consulta la sección Diameter Sh a continuación) - la aplicación `:ro` debe estar registrada.

## Habilitar Carga

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

config :sms_c, :charging,
  # Qué hacer cuando el remitente está fuera de saldo:
  #   :reject     - marcar el mensaje como :balance_rejected y no
entregarlo
  #   :auto_reply - además, SMS out_of_balance_reply_text de
vuelta al remitente
  out_of_balance_action: :reject,
  out_of_balance_reply_text: "Su mensaje no pudo ser enviado
debido a saldo insuficiente.",
  # Modo de falla cuando el OCS da error o la pila Diameter no
está configurada
  # (NO es una respuesta definitiva fuera de saldo):
  #   :allow  - falla abierta, entrega el mensaje (predeterminado)
  #   :reject - falla cerrada, trata como fuera de saldo
  on_charging_error: :allow,
  # Perfil de calificación que selecciona cómo el OCS califica el
mensaje. Enviado como el CCR
  # Service-Context-Id.
  rating_profile: "sms@sms-c"
```

La carga también requiere que la pila Diameter esté habilitada con la aplicación `:ro` (Control de Crédito es base IETF, proveedor 0):

```
config :sms_c,
  diameter_enabled: true

config :diameter_ex,
  diameter: %{
    service_name: :omnimessage,
    host: "smsc01",
    realm: "epc.mnc001.mcc001.3gppnetwork.org",
    # ... consulta la sección Diameter Sh para la pila/compañeros
completos ...
    applications: [
      %{
        application_name: :ro,
        application_dictionary: :diameter_gen_3gpp_ro,
        vendor_specific_application_ids: [
          %{vendor_id: 0, auth_application_id: 4,
acct_application_id: nil}
        ]
      }
    ]
  }
```

## Carga por Ruta

La política `charged` de cada ruta controla la carga de forma independiente del predeterminado global:

| `charged` | Comportamiento |
|---|---|
| `:yes` | Siempre verificar saldo a través del OCS |
| `:no` | Nunca cargar |
| `:default` | Usar `default_charging_enabled` |

## Deshabilitar Carga

```
config :sms_c,
  default_charging_enabled: false
```

Cuando está deshabilitado (el predeterminado), las rutas `:default` se enrutan sin ninguna verificación de saldo OCS. Las rutas individuales aún pueden optar por participar con `charged: :yes`.

## Valores predeterminados y degradación elegante

El bloque `config :sms_c, :charging` es opcional; cada clave tiene un valor predeterminado razonable (`out_of_balance_action: :reject`, un texto de respuesta incorporado, y `rating_profile: "sms@sms-c"`), y `default_charging_enabled` tiene como valor predeterminado `false`.

Por defecto, la carga **falla abierta** (`on_charging_error: :allow`), coincidiendo con la ruta de carga HTTP anterior que registraba fallos pero nunca bloqueaba la entrega:

- Si una ruta solicita carga pero la pila de Diameter **no** está habilitada/configurada, el mensaje es **permitido** (registrado como una advertencia).
- Si el OCS no es accesible o devuelve un error de protocolo, el mensaje es **permitido**.

Establezca `on_charging_error: :reject` para **fallar cerrado** en su lugar; los errores de OCS y una pila no configurada se tratan como fuera de balance y se aplica la `out_of_balance_action`.

De cualquier manera, una **respuesta definitiva fuera de balance** del OCS siempre activa la `out_of_balance_action` configurada. Los mensajes rechazados se marcan con el estado `:balance_rejected`.

# Configuración de Diameter Sh (HSS)

OmniMessage puede consultar el HSS a través de la interfaz Diameter Sh para detectar suscriptores en la red que están temporalmente fuera de línea, evitando que los mensajes se enruten a través de la puerta de enlace predeterminada. Consulte la Guía de búsqueda de suscriptores HSS para obtener la documentación operativa completa.

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

# Configuración de la pila de Diameter
config :diameter_ex,
  diameter: %{
    service_name: :omnimessage,
    listen_ip: "0.0.0.0",
    listen_port: 3868,
    decode_format: :map,
    host: "smsc01",
    realm: "epc.mnc001.mcc001.3gppnetwork.org",
    product_name: "OmniMessage",
    request_timeout: 5000,
    control_module: SmsC.Diameter.Control,
    processor_module: DiameterEx.Processor,
    vendor_id: 10415,
    supported_vendor_ids: [10415],
    applications: [
      %{
        application_name: :sh,
        application_dictionary: :diameter_gen_3gpp_sh,
        vendor_specific_application_ids: [
          %{vendor_id: 10415, auth_application_id: 16_777_217,
acct_application_id: nil}
        ]
      }
    ],
    peers: [
      %{
        host: "dra01.epc.mnc001.mcc001.3gppnetwork.org",
        ip: "10.0.0.1",
        port: 3868,
        realm: "epc.mnc001.mcc001.3gppnetwork.org",
        tls: false,
        transport: :diameter_tcp,
        initiate_connection: true
      }
    ]
  }
```

| Parámetro | Tipo | Requerido | Predeterminado | |
|---|---|---|---|---|
| `diameter_enabled` | Booleano | No | `false` | H d d c ru c : C e c |
| `hss_cache_ttl_seconds` | Entero | No | `7200` | T lc b c |
| `mock_sh` | Booleano | No | `false` | U H p la |
| `mock_sh_on_net_numbers` | Lista | No | `[]` | M c m |

Para la referencia completa de parámetros de la pila de Diameter (host, realm, peers, applications), consulte la Guía de búsqueda de suscriptores HSS.

# Configuración de ENUM

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

## Deshabilitar ENUM (Predeterminado)

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

## Habilitar ENUM con DNS Predeterminado

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

## Habilitar ENUM con Servidores DNS Personalizados

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

## Prioridad de Dominio ENUM

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

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

# Ajuste de Rendimiento ENUM

**Para Redes de Baja Latencia**:

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

**Para Enlaces de Alta Latencia/Satélite**:

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

# Ejemplo de Configuración de DNS ENUM

**Configurar Zona ENUM Privada** (formato BIND9):

```
; Archivo de zona para e164.internal.example.com
$ORIGIN e164.internal.example.com.
$TTL 300

; Número: +1-555-0100 se convierte en
0.0.1.0.5.5.5.1.e164.internal.example.com
0.0.1.0.5.5.5.1.e164.internal.example.com. IN NAPTR 100 10 "u"
"E2U+sip" "!^.*$!sip:15550100@voip-gateway.example.com!" .
0.0.1.0.5.5.5.1.e164.internal.example.com. IN NAPTR 100 20 "u"
"E2U+pstn" "!^.*$!pstn:gateway-a.example.com!" .

; Número: +1-555-0200
0.0.2.0.5.5.5.1.e164.internal.example.com. IN NAPTR 100 10 "u"
"E2U+sip" "!^.*$!sip:15550200@voip-gateway.example.com!" .
```

**Probar Resolución ENUM**:

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

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

# Configuración de Traducción de Números

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

## Deshabilitar Traducción de Números

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

## Ejemplos Básicos de Traducción de Números

**Agregar Código de País a Números Locales**:

```
config :sms_c,
  translation_rules: [
    %{
      calling_prefix: nil,
      called_prefix: "",
      source_smsc: nil,
      calling_match: "^(\d{10})$",          # Coincidir números
de 10 dígitos
      calling_replace: "+1\1",               # Anteponer +1
      called_match: "^(\d{10})$",
      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",                        # Solo para prefijo
101
  source_smsc: "carrier_a",                    # Solo de este
carrier
  calling_match: nil,                          # No cambiar
llamada
  calling_replace: nil,
  called_match: "^101(\d+)$",                 # Eliminar código de
ruta 101
  called_replace: "\1",
  priority: 5,
  description: "Eliminar código de ruta de carrier del número
llamado",
  enabled: true
}
```

## Traducción de Múltiples Reglas

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

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

    # Prioridad 50: Reglas generales
    %{
      calling_prefix: nil,
      called_prefix: nil,
      source_smsc: nil,
      calling_match: "^(\d{10})$",
      calling_replace: "+1\1",
      called_match: "^(\d{10})$",
      called_replace: "+1\1",
      priority: 50,
      description: "Normalización general de 10 dígitos",
      enabled: true
    }
  ]
```

# Intercepción Legal (X1/X2/X3)

OmniMessage actúa como un Punto de Intercepción de Servicio de Mensajes Cortos (POI). OmniMessage implementa las interfaces de intercepción ETSI TS

103 221. OmniMessage se ejecuta como el Elemento de Red (NE). Una Instalación de Monitoreo de Cumplimiento de la Ley (LEMF) provisiona objetivos de intercepción a través de X1. Cuando un mensaje coincide con un objetivo provisionado, OmniMessage codifica el mensaje como un registro xIRI TS 33.128 de 3GPP. OmniMessage luego envía el registro al LEMF a través de X2.

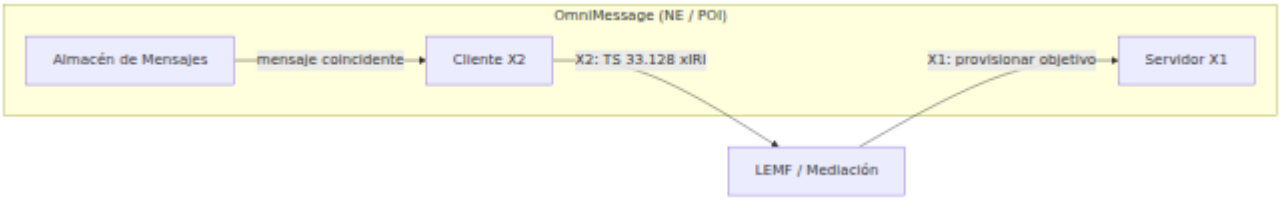
## Resumen de la Interfaz

| Interfaz | Propósito | Rol de OmniMessage | Transporte | Usado |
|---|---|---|---|---|
| **X1** | Provisionar y eliminar tareas y destinos de intercepción | Servidor | HTTPS con TLS mutuo | Sí |
| **X2** | Entregar Información Relacionada con la Intercepción (cada SMS interceptado como un TS 33.128 xIRI) | Cliente | TCP, TLS opcional | Sí |
| **X3** | Entregar Contenido de Comunicación (media) | Cliente | TCP | No. SMS no tiene plano de media. |

Referencias de estándares:

| Referencia | Título |
| --- | --- |
| ETSI TS 103 221-1 | Interfaz de intercepción interna X1 (provisionamiento) |
| ETSI TS 103 221-2 | Interfaz de intercepción interna X2/X3 (entrega) |
| 3GPP TS 33.128 | Formatos de carga útil de intercepción legal |

## Arquitectura




## Configuración

Configure la intercepción bajo `config :sms_c, :li`. Establezca los valores en `config/runtime.exs` para cada implementación. La intercepción está deshabilitada por defecto. No se inicia ningún proceso de intercepción y no ocurre ninguna intercepción hasta que establezca `enabled: true`.

```
config :sms_c, :li,
  enabled: true,
  ne_identifier: "smsc-poi-01",
  x1: [
    enabled: true,
    port: 8443,
    tls: [
      certfile: "/etc/omnimessage/li/x1-cert.pem",
      keyfile: "/etc/omnimessage/li/x1-key.pem",
      cacertfile: "/etc/omnimessage/li/lemf-ca.pem"
    ]
  ],
  x2: [enabled: true, address: "192.0.2.10", port: 12345, tls:
false]
```

## Parámetros de Nivel Superior

| Parámetro | Tipo | Requerido | Predeterminado | Descripc |
|---|---|---|---|---|
| `enabled` | Booleano | No | `false` | Interruptor maestro. Cu es `false`, n inicia ningú proceso de intercepció |
| `ne_identifier` | Cadena | No | `"smsc-01"` | Identificado Función de (atributo 18 103 221-2). OmniMessa envía este v en cada reg X2. El LEMF para etique conexión pc nombre en de por direc IP. |
| `x1` | Lista de palabras clave | No | `[]` | Servidor de aprovisiona X1. Consult Parámetros |
| `x2` | Lista de palabras clave | No | `[]` | Cliente de entrega X2. Consulte Parámetros |
| `x3` | Lista de palabras clave | No | `[]` | Aceptado p simetría cor biblioteca d intercepció |

| Parámetro | Tipo | Requerido | Predeterminado | Descripc |
|---|---|---|---|---|
| | | | | OmniMessa utiliza X3. D como `[enab false]`. |

### Parámetros de X1

OmniMessage ejecuta el servidor X1. El LEMF se conecta al servidor para aprovisionar objetivos. TLS es obligatorio (cláusula 7.2.2.1 de TS 103 221-1). El servidor no se inicia en modo de texto plano a menos que establezca `allow_insecure: true`.

| Parámetro | Tipo | Requerido | Predeterminado | Descripc |
|---|---|---|---|---|
| `enabled` | Booleano | No | `false` | Iniciar el servidor X |
| `port` | Entero | No | `8443` | Puerto TCP el servidor |
| `tls` | Lista de palabras clave | Sí | - | Configurac TLS para e servidor X Consulte Parámetro de X1. |
| `allow_insecure` | Booleano | No | `false` | Ejecutar el servidor X sobre HTTI texto plan Use esta o solo en un laboratorio use esta o en produc |

**Parámetros TLS de X1**

| Parámetro | Tipo | Requerido | Predeterminado | Descripción |
|---|---|---|---|---|
| `certfile` | Cadena | Sí | - | Ruta al certificad del servidor, en formato PEM. |
| `keyfile` | Cadena | Sí | - | Ruta a la clave privada del servidor, en formato PEM. |
| `cacertfile` | Cadena | No | - | Ruta a la autoridad de certificación que verifica los certificados de cliente. Establezca este valor para habilitar TLS mutuo. Las implementacione de producción requieren TLS mutuo. |

**Parámetros de X2**

OmniMessage ejecuta el cliente X2. El cliente se conecta al LEMF. Envía cada SMS interceptado como un registro xIRI TS 33.128. El registro lleva el mensaje SMS y un Identificador de Objetivo Coincidente.

| Parámetro | Tipo | Requerido | Predeterminado | Descripción |
|---|---|---|---|---|
| `enabled` | Booleano | No | `false` | Iniciar el cliente X2. |
| `address` | Cadena | Sí | - | Dirección IP o nombre de host del punto final X2 del LEMF. |
| `port` | Entero | Sí | - | Puerto TCP del punto final X2 del LEMF. |
| `tls` | Booleano o lista de palabras clave | No | `false` | Establezca `true` o una lista de palabras clave de opciones TLS para entregar X2 a través de TLS. Establezca `false` para TCP en texto plano. |

## Persistencia

OmniMessage mantiene las tareas y destinos de intercepción provisionados en memoria por defecto. Para mantener los objetivos provisionados a través de un reinicio, habilite el almacenamiento persistente de intercepción en la

configuración de implementación. OmniMessage escribe las tareas y destinos en disco. OmniMessage los recarga en el siguiente inicio.

### Cumplimiento

Para obtener una visión general del cumplimiento de la intercepción, consulte Cumplimiento de Intercepción ANSSI R226.

## Configuración de Enrutamiento

Reglas de enrutamiento iniciales cargadas en el primer inicio. Consulte la Guía de Enrutamiento SMS para obtener la documentación completa de enrutamiento.

## Cargar Rutas desde la Configuración

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

    # Ejemplo de enrutamiento balanceado
    %{
      calling_regex: nil,
      called_regex: ~r/^\+44/,
      source_smsc: nil,
      dest_smsc: "uk_gateway_1",
      source_type: nil,
      enum_domain: nil,
      enum_result_domain: nil,
      action: :route,
      auto_reply_message: nil,
      charged: :default,
      originating_network: :any,
      terminating_network: :any,
      weight: 70,
      priority: 50,
      description: "Puerta de enlace principal del Reino Unido
(70%)",
```

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

    # Solo en la red - restringir un enlace SMPP a destinos solo
en la red
    %{
      calling_regex: nil,
      called_regex: nil,
      source_smsc: "carrier_smpp_bind",
      dest_smsc: "local_msc",
      source_type: :smpp,
      enum_domain: nil,
      enum_result_domain: nil,
      action: :route,
      auto_reply_message: nil,
      charged: :default,
      originating_network: :any,
      terminating_network: :on_net,
      weight: 100,
      priority: 50,
      description: "Carrier X - terminación solo en la red",
      enabled: true
    }
  ]
```

## Omitir Carga Inicial de Rutas

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

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

# Configuración de Ajuste de Rendimiento

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

# Configuración de Registro

## Niveles de Registro

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

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

## Destinos de Salida de Registro

**Solo Consola** (Desarrollo):

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

**Registrador de Archivos** (Producción):

```
config :logger,
  backends: [:console, {LoggerFileBackend, :file_log}]

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

## Rotación de Registros

**Usando logrotate** (Linux):

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

# Escenarios Comunes de Configuración

## Agregador de Alto Volumen

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

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

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

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

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

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

## Mensajería en Tiempo Real para Empresas

Optimizar para baja latencia (< 20ms):

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

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

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

# Carga (habilitada) - verificaciones de balance a través de
Diameter Ro
# (consulte la sección de Configuración de Carga para los bloques
:charging + :diameter_ex)
config :sms_c,
  default_charging_enabled: true
```

## Proveedor de Servicios Multitenencia

Configuración separada por inquilino:

```
# Entorno del inquilino 1
export DB_NAME=sms_c_tenant1
export NODE_NAME=sms_tenant1@node1.example.com
# La calificación por inquilino se selecciona a través de la
configuración de perfil de calificación :charging

# Entorno del inquilino 2
export DB_NAME=sms_c_tenant2
export NODE_NAME=sms_tenant2@node1.example.com
```

## Redundancia Geográfica

Clúster a través de regiones:

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

# Validación de Configuración

Pruebe 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

# Verificar el peer de Diameter Ro al OCS (si la carga está
habilitada), en una consola remota:
#   :diameter.service_info(:omnimessage, :connections)

# 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` |
| `DB_POOL_SIZE` | Tamaño del grupo 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` |

# Mejores Prácticas de Configuración

1. **Usar Variables de Entorno** para valores sensibles (contraseñas, claves 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 detectar 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 |
| La conexión a la base de datos falla | 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, firewall | Verificar ERLANG_COOKIE, verificar puertos 4369, 9100-9200 |
| Fallos de carga | OCS/DRA inalcanzable | Verificar el par Diameter Ro (fallas de carga abiertas) |
| Fallos en búsquedas ENUM | Servidor DNS inalcanzable | Probar conectividad DNS, verificar tiempo de espera |
| Bajo rendimiento | Configuraciones de lote incorrectas | Revisar la sección de Configuración de Ajuste de Rendimiento anterior |
| Mensajes no enrutados | Rutas no cargadas | Verificar la configuración sms_routes o la 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).
  # La opción predeterminada incorporada es 24 si no se establece;
el runtime.exs enviado establece 168 (7 días).
  message_retention_hours: 168,

  # Con qué frecuencia verificar mensajes antiguos (minutos)
  retention_check_interval_minutes: 60
```

**Recomendaciones:**

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

**Impacto en la Memoria:**

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

## Exportación de CDR (Registro de Detalles de Llamadas)

Cuando los mensajes 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 la escritura de CDR (predeterminado
incorporado: false)
  cdr_enabled: true
```

**Los Registros de CDR Incluyen:**

- ID del 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

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

```
config :sms_c,
  # Eliminar el cuerpo del mensaje de Mnesia después de la entrega
exitosa
  delete_message_body_after_delivery: false,

  # Ocultar el cuerpo del mensaje en la interfaz web
  hide_message_body_in_ui: false,

  # Ocultar el cuerpo del mensaje en exportaciones CSV
  hide_message_body_in_export: false
```

**Casos de Uso:**

| Configuración | Caso de Uso |
| --- | --- |
| `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 |

**Ejemplos de Configuraciones:**

*Máxima Privacidad (Cumplimiento)*

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

*Desarrollo (Visibilidad Completa)*

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

# Registro de Inicio

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

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

Esto proporciona visibilidad inmediata sobre las características activas.

# Federación Geográfica

**← Volver al Índice de Documentación** | **Referencia de Configuración** | **Guía de Operaciones**

## Descripción General

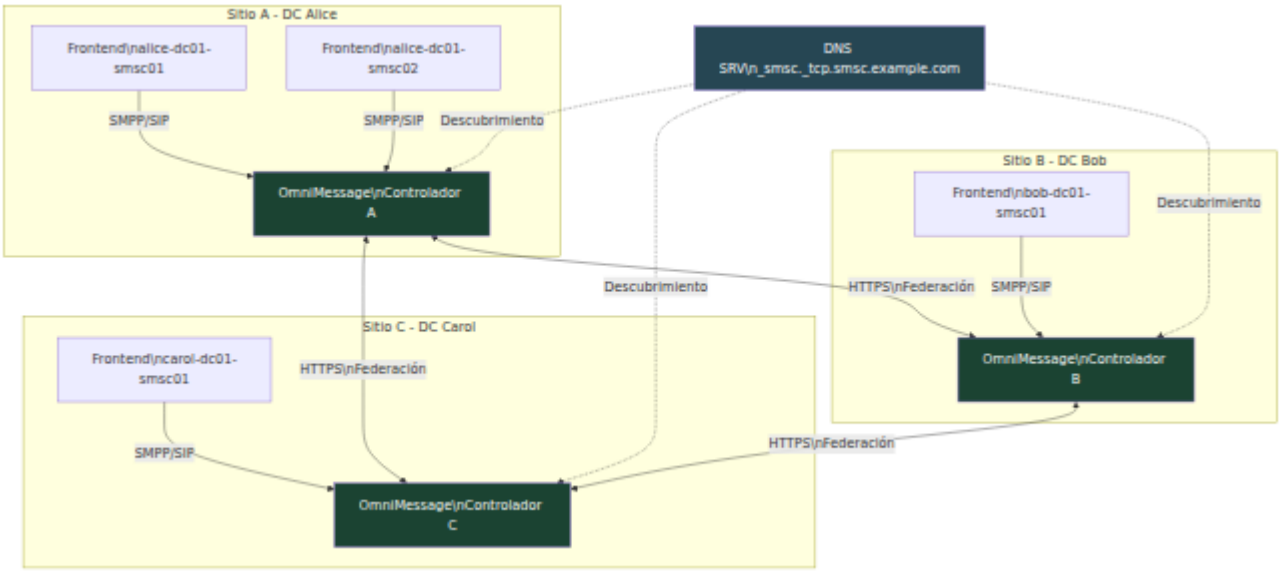
OmniMessage utiliza **federación basada en HTTP** con un **modelo de extracción** para implementaciones de múltiples controladores a través de centros de datos o regiones. Cada controlador opera de manera independiente, gestionando su propia cola de mensajes, tabla de enrutamiento y conjunto de frontends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

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

## Características Clave

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

# Arquitectura



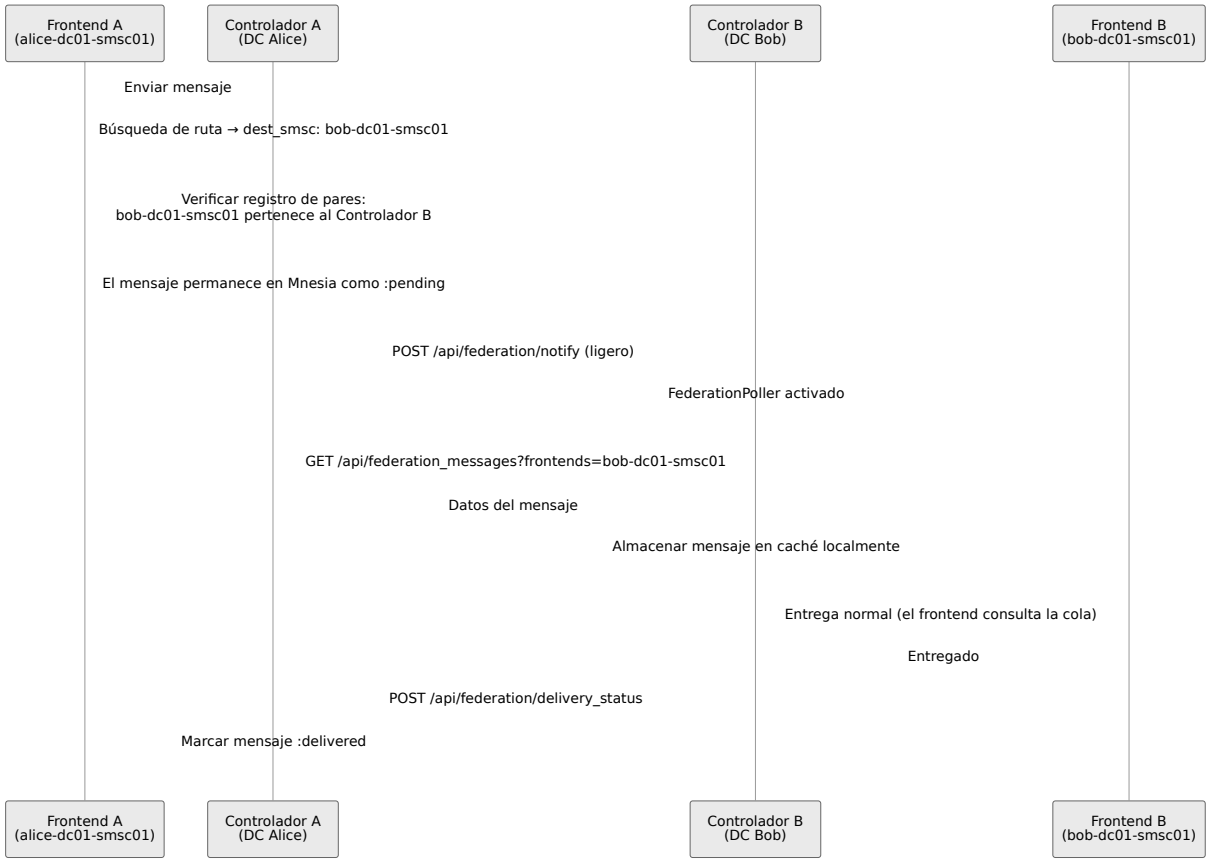

## Decisiones de Diseño Clave

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

# Flujo de Mensajes (Modelo de Extracción)

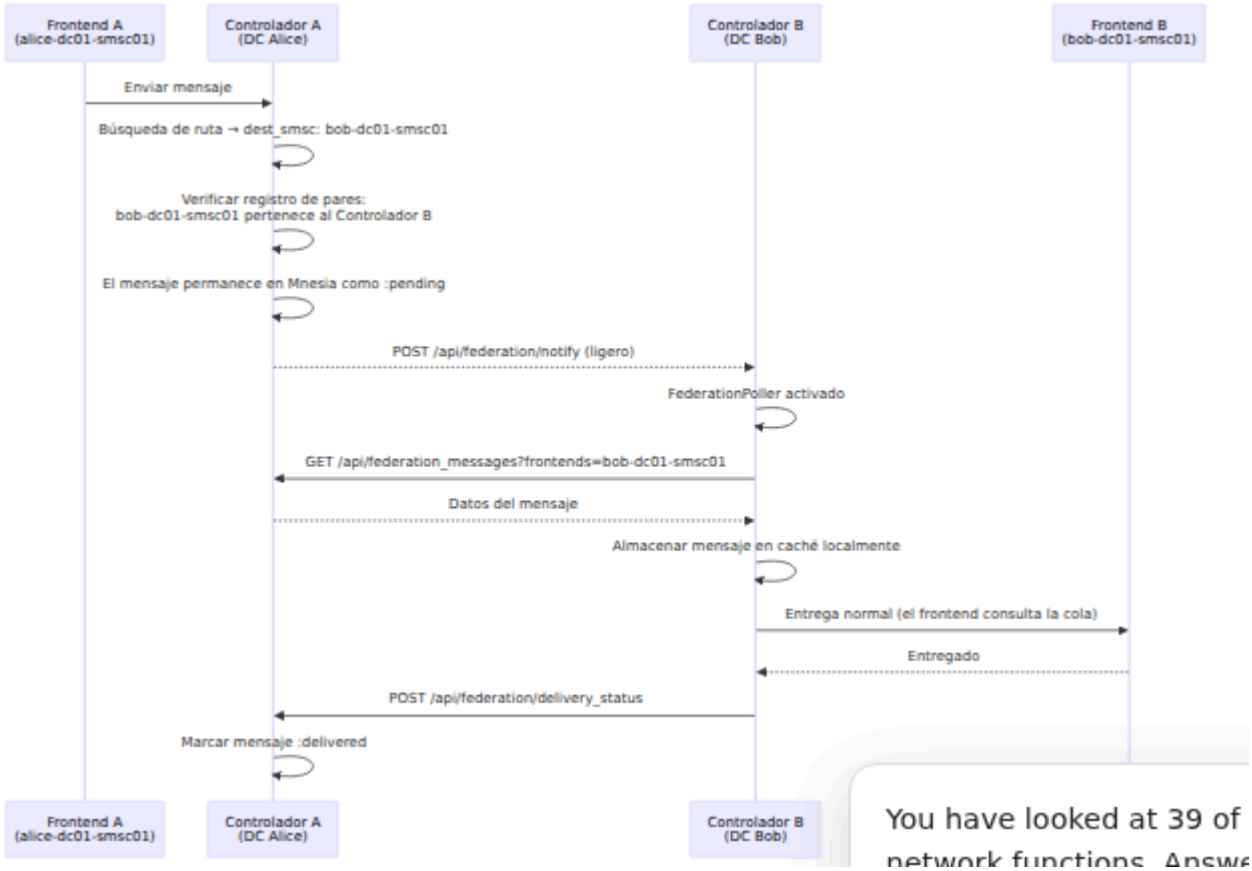
Cuando un mensaje llega a un controlador y el enrutamiento determina que el frontend de destino está en un controlador remoto:

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



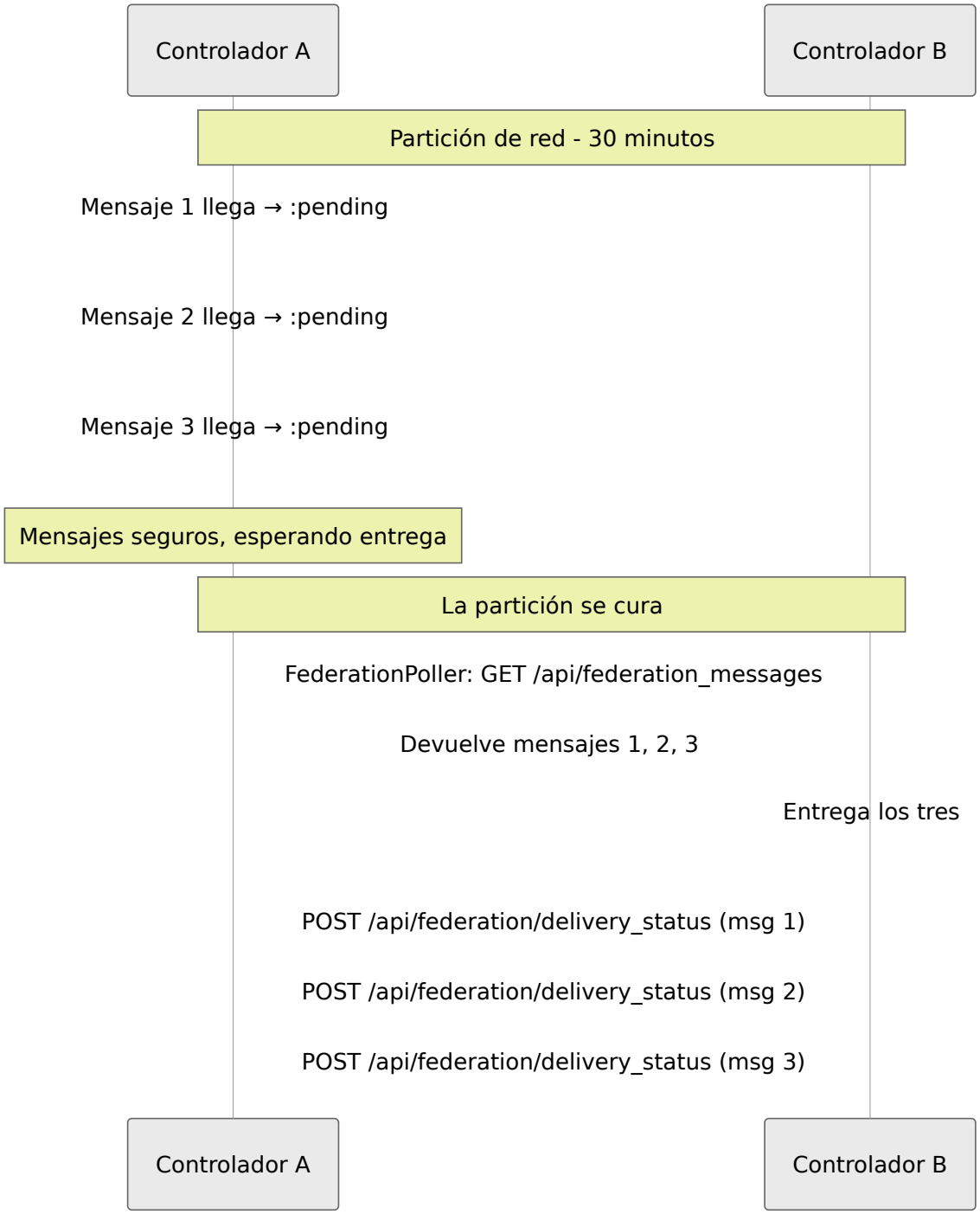
# Par Inalcanzable - Fallo de Notificación

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



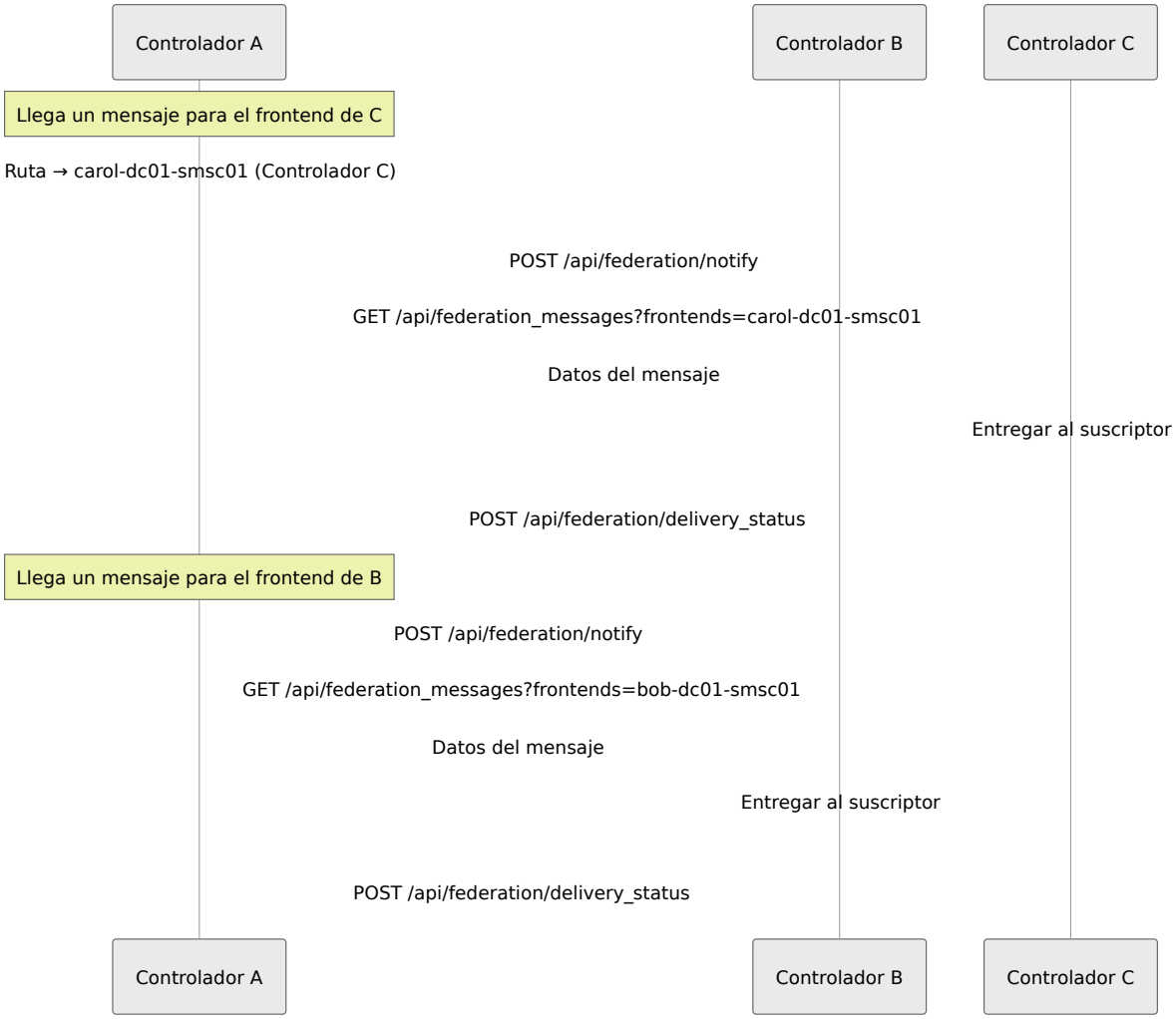

# Recuperación de Partición - Múltiples Mensajes en Cola

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




# Flujo de Mensajes en Tres Sitios

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




# Salud de Pares y Sincronización de Registro

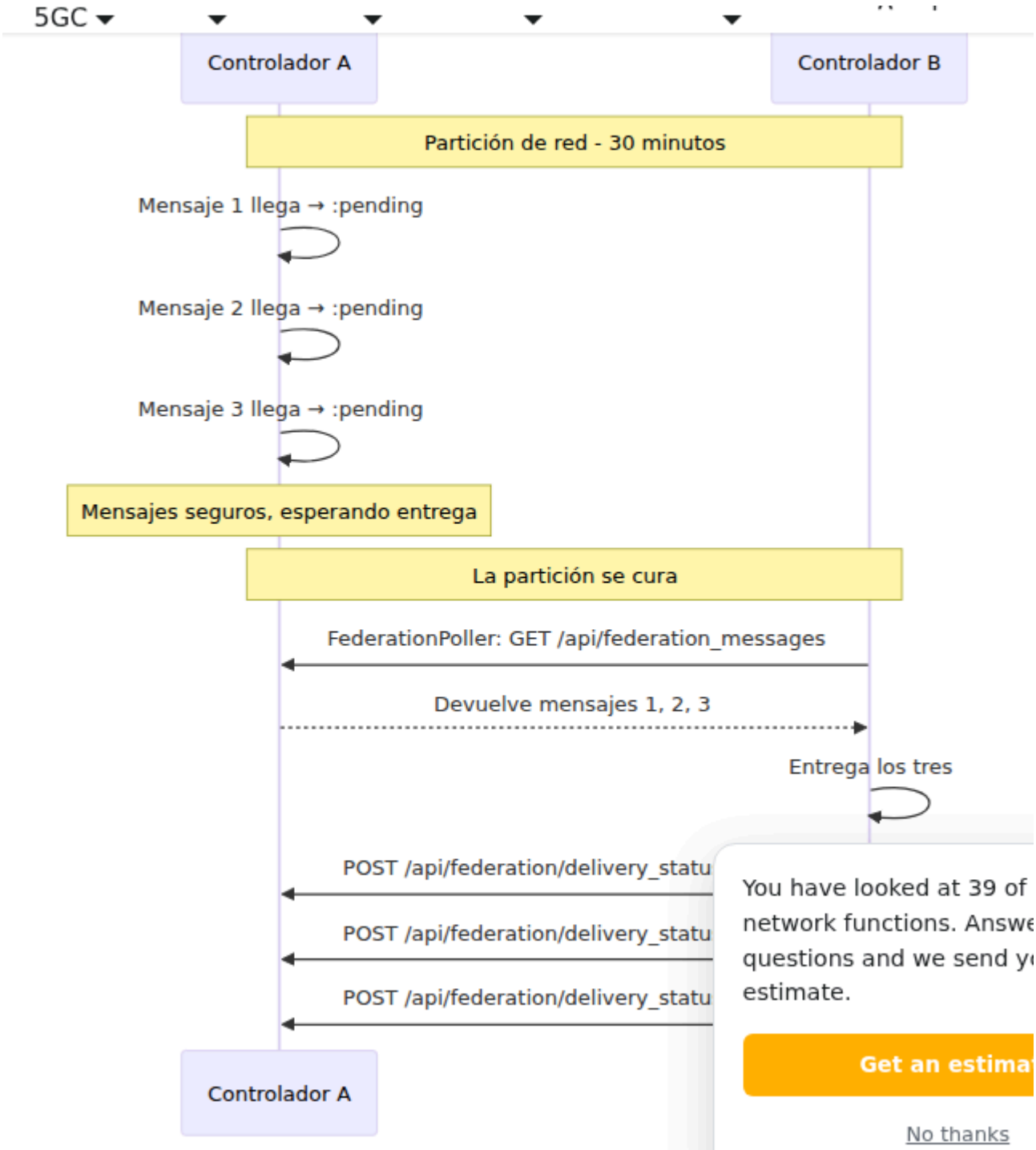
Los controladores de federación mantienen la conciencia entre sí a través de dos ciclos periódicos:

## Ciclo de Verificación de Salud

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

# Ciclo de Sincronización de Registro

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




**Estados de pares y su efecto en la consulta de federación:**

| Estado | Verificaciones de Salud | FederationPoller | Sincronización de Registro |
|---|---|---|---|
| **Desconocido** | En progreso | No consultado | No intentado |
| **Saludable** | Aprobadas | Consulta activa | Activo |
| **Degradado** | 1-2 fallos | No consultado | No intentado |
| **No Saludable** | 3+ fallos | No consultado | No intentado |

# Configuración

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

## Configuración Mínima (Descubrimiento DNS SRV)

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

## Referencia Completa de Configuración

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

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

  # Con qué frecuencia volver a resolver los registros DNS SRV
(milisegundos)
  dns_poll_interval_ms: 30_000,

  # Con qué frecuencia intercambiar el estado de salud con todos
los pares conocidos (milisegundos)
  health_check_interval_ms: 10_000,

  # Con qué frecuencia intercambiar registros de frontends con
pares saludables (milisegundos)
  registry_sync_interval_ms: 15_000,

  # Con qué frecuencia el FederationPoller consulta a los pares
saludables en busca de mensajes (milisegundos)
  poll_interval_ms: 5_000,

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

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

  # Lista de pares estática - utilizada cuando dns_srv_domain está
vacío
  static_peers: []
```

## Parámetros de Federación

| Parámetro | Tipo | Requerido | Predeterminado |
|---|---|---|---|
| `enabled` | Booleano | Sí | `false` |
| `dns_srv_domain` | Cadena | No | `""` |
| `dns_poll_interval_ms` | Entero | No | `30000` |

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

| Parámetro | Tipo | Requerido | Predeterminado |
| --- | --- | --- | --- |
| | | | |
| `api_port` | Entero | No | `8443` |
| `static_peers` | Lista | No | `[]` |

# Configuración de Pares Estáticos

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

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

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

# Configuración de Registros DNS SRV

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

**Configuración de Zona DNS:**

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

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

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

**Eliminar un sitio**: Elimine el registro SRV de DNS. Los controladores existentes eliminarán el par que se fue dentro de un ciclo de consulta.

# Requisitos de Red

La federación requiere solo un único puerto HTTPS entre sitios.

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

**Reglas de firewall** (por sitio):

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

# Puntos de API de Federación

El controlador sirve todos los puntos finales de federación en el puerto API estándar (8443) bajo `/api/federation/`. Los controladores pares llaman a estos puntos finales, no los clientes externos o frontends.

| Punto final | Método | Propósito |
| --- | --- | --- |
| `/api/federation/identity` | GET | Devuelve el nombre del nodo de este controlador y la versión del protocolo |
| `/api/federation/health` | POST | Intercambio de salud bidireccional - el llamador envía estado, el llamado responde con el propio |
| `/api/federation/registry` | POST | Intercambio bidireccional de registros de frontends |
| `/api/federation/notify` | POST | Recibir notificación ligera de que un mensaje está disponible para un frontend local |
| `/api/federation/delivery_status` | POST | Recibir confirmación de entrega del controlador de destino |
| `/api/federation/delivery_attempt` | POST | Recibir un intento de entrega fallido del controlador de destino; ejecuta la espera de reintento y devuelve el mensaje actualizado |
| `/api/federation/event` | POST | Recibir un evento de mensaje que un frontend del controlador de destino informó; lo |

| Punto final | Método | Propósito |
|---|---|---|
| | | registra en el registro de eventos local |
| `/api/federation_messages` | GET | Devuelve mensajes pendientes para los frontends especificados (utilizado por FederationPoller) |

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

# Ejemplos de Implementación

## Ejemplo 1: Implementación de Tres Sitios con DNS SRV

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

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

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

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

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

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

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

**Cómo funciona:** Los tres controladores resuelven el mismo dominio SRV y se descubren automáticamente. Cada controlador registra sus frontends locales. Cuando un mensaje llega a DC Alice destinado a un suscriptor atendido por DC Bob, la tabla de enrutamiento selecciona `bob-dc01-smsc01` como destino. La capa de federación detecta que este frontend pertenece al Controlador B y envía una notificación. El FederationPoller del Controlador B consulta al Controlador A y almacena el mensaje en caché para la entrega local.

## Ejemplo 2: Implementación Activa-Activa de Dos Sitios con Pares Estáticos

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

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

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

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

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

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

## Ejemplo 3: Consulta Agresiva para Enlaces de Baja Latencia

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

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

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

# Métricas

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

**Métrica**: `sms_c_federation_message_received_count` **Tipo**: Contador **Descripción**: Número de notificaciones de federación recibidas de pares **Etiquetas**:

- `origin_node` - Par que envió la notificación

**Métrica**: `sms_c_federation_health_check_count` **Tipo**: Contador **Descripción**: Resultados de la verificación de salud por par **Etiquetas**:

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

**Consultas de ejemplo**:

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

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

## Indicadores Clave a Monitorear

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

# Solución de Problemas

## Par No Descubierto

**Síntomas**: Los registros del controlador no muestran pares descubiertos; `cluster_status` devuelve vacío.

**Causas posibles**:

- El dominio DNS SRV es incorrecto o no resolvible
- El servidor DNS es inalcanzable desde el host del controlador
- Los registros SRV aún no se han propagado
- La federación no está habilitada (`enabled: false`)

**Resolución**:

1. Verifique la resolución DNS SRV desde el host del controlador: `dig SRV _smsc._tcp.smsc.example.com`
2. Verifique que `enabled: true` esté configurado en la configuración de federación

3. Verifique los registros de la aplicación para mensajes de "Descubrimiento DNS de Federaciones" al inicio
4. Si utiliza pares estáticos, verifique que la lista `static_peers` contenga entradas correctas de host/puerto

## Marcado de Pares No Saludable

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

**Causas posibles**:

- El controlador remoto está caído
- El firewall bloquea el puerto 8443 entre sitios
- Problemas con el certificado TLS
- Partición de red entre sitios

**Resolución**:

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

## Mensajes No Entregados al Sitio Remoto

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

**Causas posibles**:

- El controlador de destino no está saludable (FederationPoller solo consulta pares saludables)
- FederationPoller no está en funcionamiento

- El registro de frontend no está sincronizado (el destino no sabe qué frontends son locales)

**Resolución**:

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

# Búsqueda de Suscriptores HSS (Diameter Sh)

**← Volver al Índice de Documentación** | **Guía de Enrutamiento SMS**

## Descripción General

OmniMessage puede consultar el Servidor de Suscriptores en el Hogar (HSS) a través de la interfaz Diameter Sh para determinar si un MSISDN pertenece a la red (en la red) o no (fuera de la red). Esta respuesta se utiliza como un **criterio de coincidencia de ruta**: las rutas pueden requerir que el número de origen o de destino esté en la red o fuera de la red antes de que se apliquen.

La consulta es **perezosa** - ocurre *durante la evaluación de la tabla de rutas*, solo cuando una ruta que se está evaluando declara un requisito de `originating_network` o `terminating_network` de `:on_net` o `:off_net`. Las rutas que dejan ambos en `:any` (el valor predeterminado) nunca activan una consulta. No se realiza una búsqueda global en el HSS antes del enrutamiento.

OmniMessage envía una **Solicitud de Datos de Usuario (UDR)** para el número. Si el HSS confirma que el suscriptor existe (código de resultado 2001) el número está **en la red**; si el HSS devuelve *Usuario Desconocido* (código de resultado 5001) el número está **fuera de la red**. Consulte Guía de Enrutamiento SMS → Coincidencia en la Red / Fuera de la Red para ver cómo estos criterios se combinan con el resto de una ruta.

> **En la red vs fuera de la red - ¿HSS o ENUM?** El HSS (Sh) responde "¿es este número un suscriptor en *nuestro* HSS?" con un resultado explícito de en/fuera de la red. ENUM (DNS NAPTR) responde "¿a qué dominio de operador se resuelve este número?" y se deriva en/fuera de la red al hacer coincidir su propio dominio. Use Sh cuando su HSS sea autoritativo; use

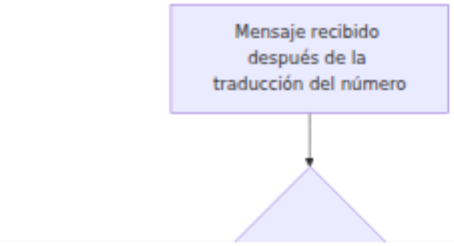
> ENUM para portabilidad/enrutamiento entre operadores. Consulte Guía de Enrutamiento SMS → Búsqueda ENUM.

## Especificaciones Relevantes

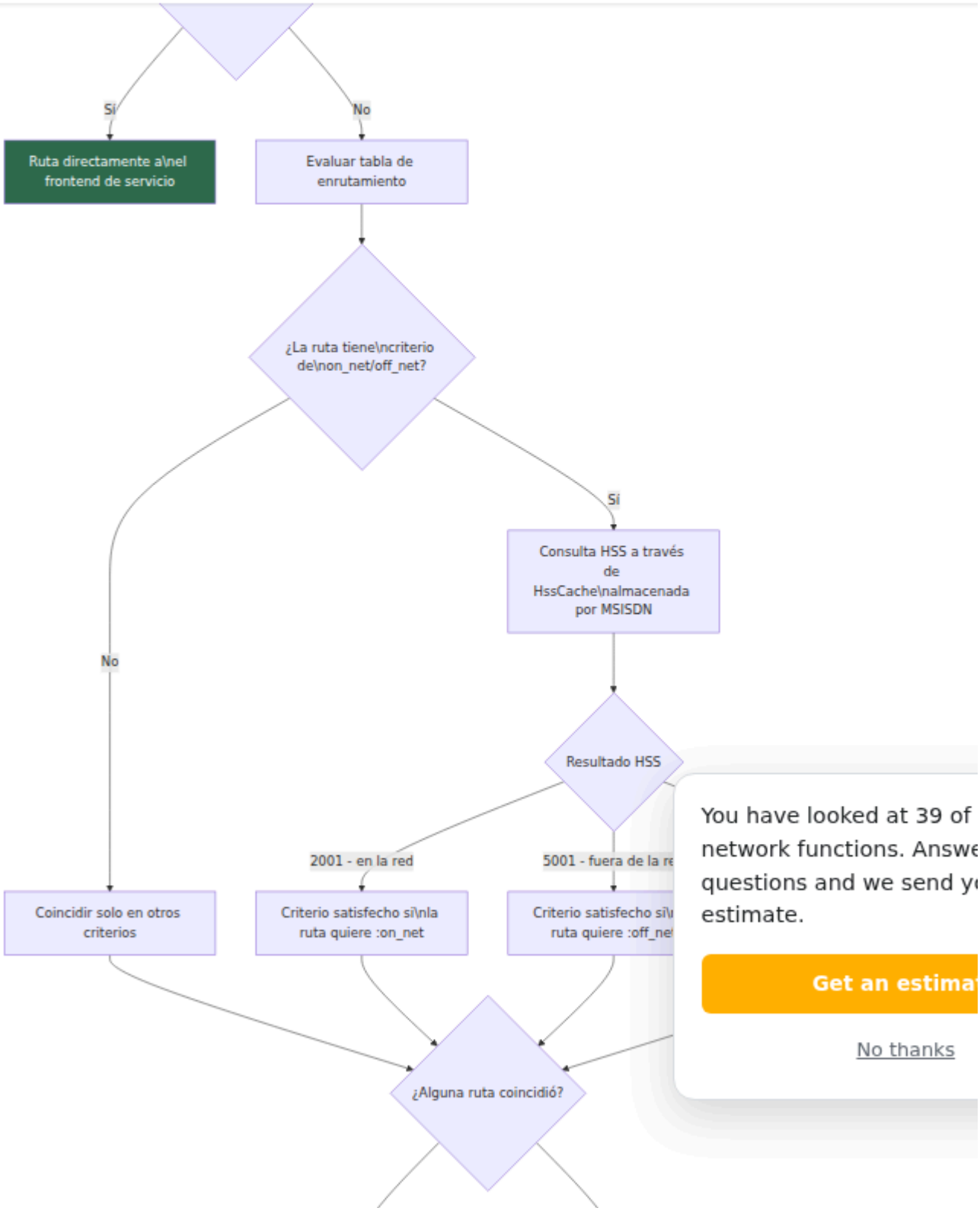
| Especificación | Título |
|---|---|
| 3GPP TS 29.329 | Interfaz Sh basada en Diameter |
| 3GPP TS 29.328 | Interfaz IMS Sh - flujos de señalización y contenidos de mensajes |
| RFC 6733 | Protocolo Base Diameter |

# Dónde Encaja la Consulta HSS

OmniMessage evalúa un mensaje en este orden:






You have looked at 39 of network functions. Answe questions and we send y estimate.

Get an estima

No thanks




## Enrutamiento Basado en Ubicación (verificado primero)

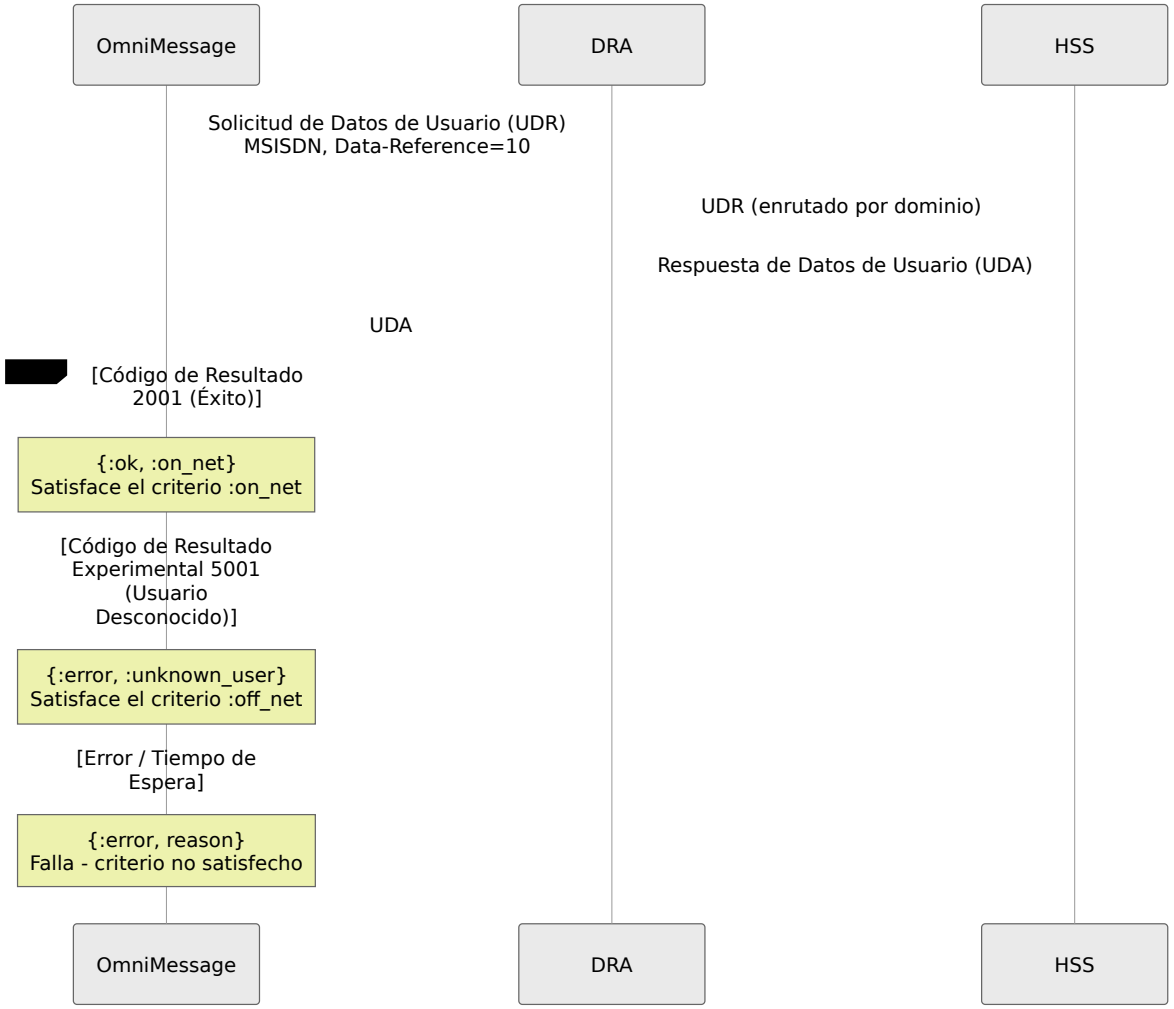
Si el MSISDN de destino se encuentra en la tabla de registros/ubicaciones, el mensaje se enruta directamente al frontend que atiende a ese suscriptor. No se realiza búsqueda en el HSS ni evaluación de la tabla de enrutamiento. (Consulte Expiración de Ubicación.)

## Tabla de Enrutamiento + Consulta HSS Perezosa

Si no hay un registro activo, el enrutador evalúa la tabla de enrutamiento. Para cada ruta que declara `originating_network`/`terminating_network` de `:on_net` o `:off_net`, el enrutador realiza una consulta HSS (a través de la caché) como parte de la coincidencia de esa ruta. Una ruta solo coincide si satisface su criterio de red junto con todos sus otros criterios. Si Diameter está deshabilitado o el HSS no da una respuesta definitiva, el criterio falla (fail-closed) y la ruta no coincide.

Retener un mensaje para **almacenamiento y reenvío** es un concepto separado: la *acción* de la ruta `:saf` lo activa, no la consulta HSS. Consulte Entrega de Almacenamiento y Reenvío.

# Flujo de Mensajes Diameter Sh



## Parámetros UDR

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

## Códigos de Resultado UDA

La verificación de existencia de suscriptores mapea la UDA a uno de tres resultados:

| Código | Tipo | Resultado | Significado para la coincidencia de ruta |
|---|---|---|---|
| 2001 | Código de Resultado | `{:ok, :on_net}` | En la red - satisface `:on_net`, falla `:off_net` |
| 5001 | Código de Resultado Experimental | `{:error, :unknown_user}` | Fuera de la red - satisface `:off_net`, falla `:on_net` |
| Otro / tiempo de espera / sin pares | - | `{:error, reason}` | Falla - no satisface ninguno; la ruta no coincide |

# Configuración

## Habilitar Búsqueda HSS

Dos bloques de configuración en `config/runtime.exs` controlan la búsqueda HSS:

1. La bandera `diameter_enabled` bajo `:sms_c` - controla si se realizan consultas HSS durante la coincidencia de rutas. Cuando es `false`, cualquier ruta con un criterio `:on_net`/`:off_net` falla en coincidir (fail-closed).
2. La configuración `:diameter_ex` - configura la pila Diameter en sí (identidad, pares, aplicaciones)

## Configuración Mínima

```
# Habilitar consultas HSS durante la coincidencia de rutas
config :sms_c,
  diameter_enabled: true

# Configuración de la pila Diameter
config :diameter_ex,
  diameter: %{
    service_name: :omnimessage,
    listen_ip: "0.0.0.0",
    listen_port: 3868,
    decode_format: :map,
    host: "smsc01",
    realm: "epc.mnc001.mcc001.3gppnetwork.org",
    product_name: "OmniMessage",
    request_timeout: 5000,
    peer_selection_algorithm: :random,
    allow_undefined_peers_to_connect: true,
    log_unauthorized_peer_connection_attempts: true,
    control_module: SmsC.Diameter.Control,
    processor_module: DiameterEx.Processor,
    auth_application_ids: [],
    acct_application_ids: [],
    vendor_id: 10415,
    supported_vendor_ids: [10415],
    applications: [
      %{
        application_name: :sh,
        application_dictionary: :diameter_gen_3gpp_sh,
        vendor_specific_application_ids: [
          %{
            vendor_id: 10415,
            auth_application_id: 16_777_217,
            acct_application_id: nil
          }
        ]
      }
    ],
    peers: [
      %{
        port: 3868,
        host: "dra01.epc.mnc001.mcc001.3gppnetwork.org",
```

```
        ip: "10.0.0.1",
        realm: "epc.mnc001.mcc001.3gppnetwork.org",
        tls: false,
        transport: :diameter_tcp,
        initiate_connection: true
      }
    ]
  }
```

## Parámetros de OmniMessage

| Parámetro | Tipo | Requerido | Predeterminado | |
|---|---|---|---|---|
| `diameter_enabled` | Booleano | No | `false` | H H c r `f` c `:` fa |
| `hss_cache_ttl_seconds` | Entero | No | `7200` | T p d e E `r` e e o |
| `mock_sh` | Booleano | No | `false` | U H lu D S |
| `mock_sh_on_net_numbers` | Lista | No | `[]` | L tr la `m` |

## Parámetros de la Pila Diameter

| Parámetro | Tipo | Requerido | P |
|---|---|---|---|
| `service_name` | Átomo | Sí | - |
| `listen_ip` | Cadena | No | ' |
| `listen_port` | Entero | No | 3 |
| `decode_format` | Átomo | Sí | - |
| `host` | Cadena | Sí | - |
| `realm` | Cadena | Sí | - |
| `product_name` | Cadena | No | - |
| `request_timeout` | Entero | No | 5 |
| `peer_selection_algorithm` | Átomo | No | : |

| Parámetro | Tipo | Requerido | P |
|---|---|---|---|
| `allow_undefined_peers_to_connect` | Booleano | No | t |
| `log_unauthorized_peer_connection_attempts` | Booleano | No | t |
| `control_module` | Módulo | Sí | - |
| `processor_module` | Módulo | Sí | - |
| `auth_application_ids` | Lista | No | [ |
| `acct_application_ids` | Lista | No | [ |
| `vendor_id` | Entero | Sí | - |
| `supported_vendor_ids` | Lista | Sí | - |

## Parámetros de Aplicación

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

| Parámetro | Tipo | Requerido | Predetermina |
|---|---|---|---|
| `application_name` | Átomo | Sí | - |
| `application_dictionary` | Átomo | Sí | - |
| `vendor_specific_application_ids` | Lista | Sí | - |

**ID de Aplicación Específica del Proveedor (Sh):**

| Parámetro | Tipo | Valor | Descripción |
|---|---|---|---|
| `vendor_id` | Entero | `10415` | ID de proveedor 3GPP |
| `auth_application_id` | Entero | `16_777_217` | ID de aplicación Sh según 3GPP TS 29.329 |
| `acct_application_id` | - | `nil` | No se utiliza para Sh |

## Parámetros de Pares

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

| Parámetro | Tipo | Requerido | Predeterminado | De |
|---|---|---|---|---|
| `host` | Cadena | Sí | - | Ident Host del p Debe exac con l confi par. |
| `ip` | Cadena | Sí | - | Direc par p cone |
| `port` | Entero | No | 3868 | Puer del p |
| `realm` | Cadena | Sí | - | Dom Diam par. |
| `tls` | Booleano | No | `false` | Habi para cone |
| `transport` | Átomo | No | `:diameter_tcp` | Proto trans `:dia` o `:dia` |
| `initiate_connection` | Booleano | No | `true` | Si es Omn inicia cone con e |

| Parámetro | Tipo | Requerido | Predeterminado | De |
|---|---|---|---|---|
| | | | | `fals` Omn espe par s |

## Múltiples Pares

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

```
peers: [
  %{
    port: 3868,
    host: "dra01.epc.mnc001.mcc001.3gppnetwork.org",
    ip: "10.0.0.1",
    realm: "epc.mnc001.mcc001.3gppnetwork.org",
    tls: false,
    transport: :diameter_tcp,
    initiate_connection: true
  },
  %{
    port: 3868,
    host: "dra02.epc.mnc001.mcc001.3gppnetwork.org",
    ip: "10.0.0.2",
    realm: "epc.mnc001.mcc001.3gppnetwork.org",
    tls: false,
    transport: :diameter_tcp,
    initiate_connection: true
  }
]
```

# Modo Simulado

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

```
config :sms_c,
  diameter_enabled: true,
  mock_sh: true,
  mock_sh_on_net_numbers: [
    "15555550001",
    "15555550002"
  ]
```

En modo simulado, cualquier MSISDN en la lista `mock_sh_on_net_numbers` se resuelve a `{:ok, :on_net}` (equivalente al código de resultado 2001). Todos los demás MSISDN se resuelven a `{:error, :unknown_user}` (equivalente al código de resultado 5001). No se inicia ninguna pila Diameter y no se realizan conexiones de red.

# Caché de Resultados HSS

Las búsquedas HSS se almacenan en caché por MSISDN mediante `SmsC.Diameter.HssCache` para evitar viajes redundantes de Diameter:

- **TTL**: `hss_cache_ttl_seconds` (predeterminado 7200 = 2 horas)
- **Compartido**: múltiples rutas que necesitan el mismo estado de MSISDN en una evaluación comparten una única consulta
- **Actualización forzada**: la caché se omite (se realiza una nueva consulta) cuando un mensaje se reenvía explícitamente
- **Predicados**: un resultado se trata como en la red solo para `{:ok, :on_net}` y como fuera de la red solo para `{:error, :unknown_user}`; cualquier otro resultado (incluido `nil` cuando Diameter está deshabilitado) se trata como ninguno - lo que produce coincidencias fail-closed.

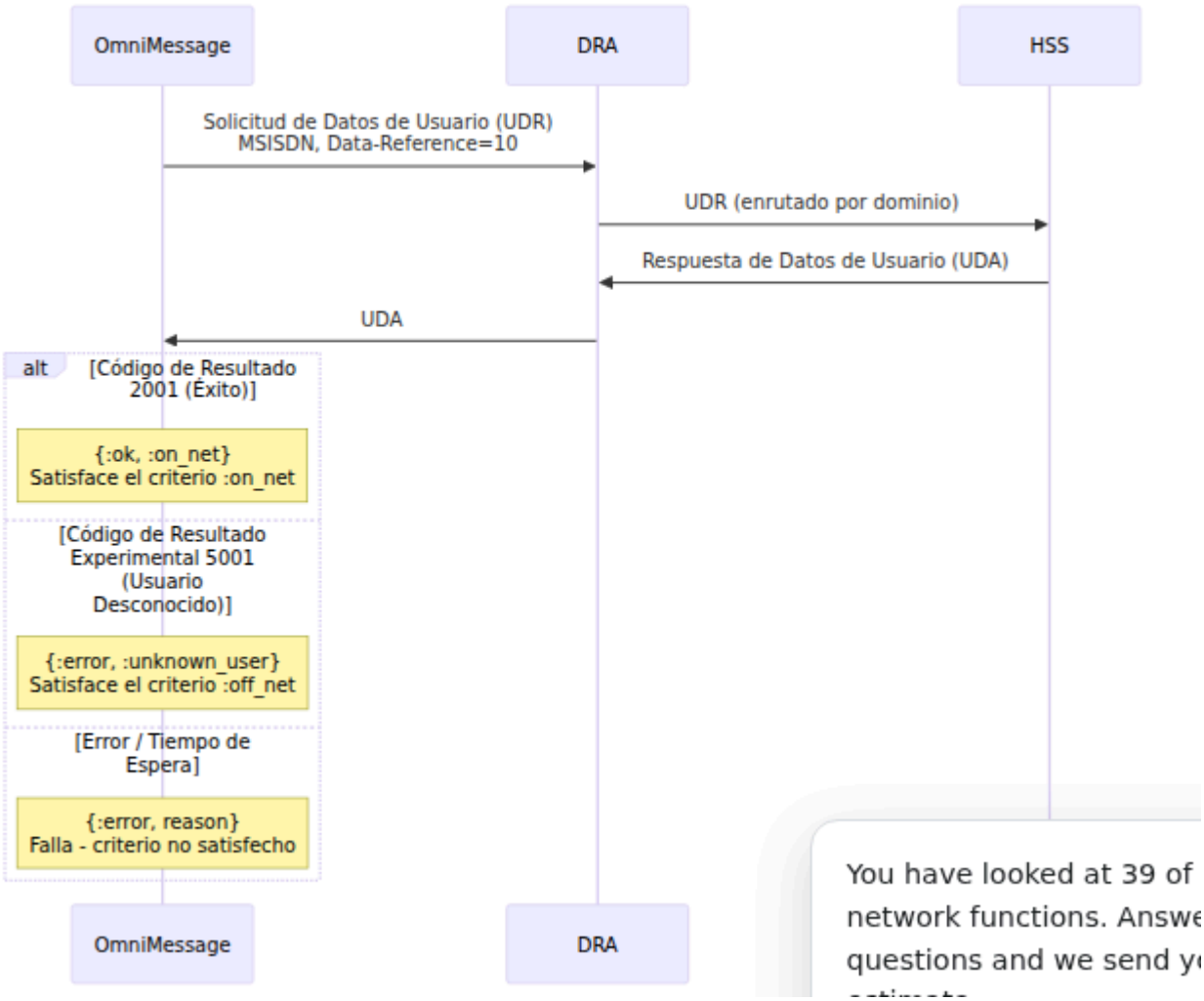
# Entrega de Almacenamiento y Reenvío

La **acción** de ruta `:saf` impulsa el almacenamiento y reenvío, no la consulta HSS. Una ruta con `action: :saf` retiene el mensaje para su entrega posterior. Para retener *solo* a suscriptores en la red, combine `:saf` con

`terminating_network: :on_net` y proporcione una ruta de respaldo fuera de la red (consulte Guía de Enrutamiento SMS → `saf`).

Cuando un mensaje se retiene para SAF, permanece en la cola de mensajes de Mnesia con `dest_smsc` sin establecer. Cuando el suscriptor se registra y llega una actualización de ubicación, el bucle de entrega recoge los mensajes pendientes para ese MSISDN y los enruta al frontend de servicio.

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



# Notas Operativas

## Registro de Inicio

OmniMessage registra el estado de Diameter Sh al inicio:

- **Habilitado**: `Diameter Sh (HSS): HABILITADO (host: smsc01, realm: epc.mnc001.mcc001.3gppnetwork.org, peers: 2)`
- **Deshabilitado**: `Diameter Sh (HSS): DESHABILITADO`

## Comportamiento Fail-Closed

Si una búsqueda HSS no devuelve una respuesta definitiva - tiempo de espera, sin pares conectados, respuesta no manejada o un código de resultado inesperado - `subscriber_exists?/1` devuelve `{:error, reason}`. Para una ruta que requiere `:on_net` o `:off_net`, ese criterio **no** se satisface, por lo que la ruta **no coincide** y la evaluación continúa con la siguiente ruta.

Este es un diseño **fail-closed**: sin una respuesta definitiva, OmniMessage no tratará un número como en la red o fuera de la red, por lo que una ruta solo en la red no se activará y una ruta de puerta de enlace solo fuera de la red tampoco se activará. Proporcione una ruta de respaldo general (`terminating_network: :any`) si desea que los mensajes se enruten durante las interrupciones del HSS.

Cuando **Diameter está explícitamente deshabilitado** (`diameter_enabled: false`), la búsqueda en caché devuelve `nil` y la misma coincidencia fail-closed se aplica a cualquier ruta con un criterio `:on_net`/`:off_net`. Las rutas que dejan ambos campos de red en `:any` no se ven afectadas.

## Observabilidad

Las consultas HSS son observables a través de:

- **Registros de ruta por mensaje** - cada evento `route_evaluated` de la ruta incluye la verificación de red, por ejemplo, `term_network: ✓ (el destino requiere on_net, +15551234 está en la red)` o `✗ (... está`

`fuera de la red)`. Visible en el registro de actividad de mensajes del panel de control y en la API de Eventos.

- **Telemetría Diameter Sh** - `SmsC.Telemetry` emite métricas de solicitud, respuesta (etiquetadas por resultado), duración y errores (consulte METRICS.md).
- **Registros de aplicación** - líneas `HSS dip: <msisdn> está EN LA RED (2001)` / `NO está en HSS (5001)` de `SmsC.Diameter.Sh`.

## Expiración de Ubicación

El enrutamiento basado en ubicación verifica la marca de tiempo `expires` en los registros de suscriptores. Si el registro de un suscriptor ha expirado, por ejemplo, porque entró en modo avión, el enrutador trata la ubicación como inactiva. La evaluación luego procede a la tabla de enrutamiento, incluida cualquier consulta HSS perezosa que requiera una ruta coincidente.

Esto evita que los mensajes se envíen a un frontend que ya no puede alcanzar al suscriptor, evitando fallos de entrega innecesarios y ciclos de reintento.

# Aplicación de Requisitos En la Red / Fuera de la Red a Nivel de Ruta

Las rutas aplican requisitos en la red/fuera de la red a través de dos campos - **no** banderas booleanas y **no** una acción de ruta:

| Campo | Valores | Verificaciones | Comportamiento |
|---|---|---|---|
| `originating_network` | `:any` \| `:on_net` \| `:off_net` | Número de llamada | La ruta coincide solo si el estado HSS del MSISDN de origen coincide (fail-closed) |
| `terminating_network` | `:any` \| `:on_net` \| `:off_net` | Número llamado | La ruta coincide solo si el estado HSS del MSISDN de destino coincide (fail-closed) |

El valor predeterminado para ambos es `:any` (sin verificación HSS). Un valor de `:on_net` requiere el código de resultado 2001; `:off_net` requiere 5001; cualquier respuesta indefinida falla el criterio.

## Ejemplo de Configuración

```
config :sms_c, :sms_routes, [
  # Solo destinos en la red - entregar a puerta de enlace local
  %{
    calling_regex: nil,
    called_regex: ~r/^1555/,
    source_smsc: nil,
    dest_smsc: "local-gateway",
    source_type: nil,
    action: :route,
    charged: :default,
    weight: 100,
    priority: 10,
    description: "Puerta de enlace local - solo destinos en la
red",
    originating_network: :any,
    terminating_network: :on_net,
    enabled: true
  },

  # Respaldo fuera de la red - captura todo lo que la ruta en la
red no tomó
  %{
    calling_regex: nil,
    called_regex: nil,
    source_smsc: nil,
    dest_smsc: "external-gateway",
    source_type: :ims,
    action: :route,
    charged: :default,
    weight: 100,
    priority: 255,
    description: "Puerta de enlace externa - respaldo",
    originating_network: :any,
    terminating_network: :any,
    enabled: true
  }
]
```

**Cómo funciona:** para un mensaje a un suscriptor no registrado:

1. **Verificación de ubicación** - no se encontró registro activo
2. **Evaluación de ruta** - las rutas se verifican en orden de prioridad
3. La ruta P10 "local-gateway" tiene `terminating_network: :on_net`, por lo que se realiza una consulta HSS sobre el número llamado. Si el HSS devuelve 2001 (en la red), la ruta coincide y el mensaje se entrega a `local-gateway`. Si devuelve 5001 (fuera de la red) o falla (fail-closed), la ruta no coincide.
4. La ruta P255 "external-gateway" tiene `terminating_network: :any`, por lo que no necesita respuesta HSS y coincide como respaldo.

**Caso de uso:** mantener el tráfico en la red en la infraestructura local mientras el tráfico fuera de la red (o incierto de HSS) llega a un transportista externo. Para *retener* a los suscriptores en la red para almacenamiento y reenvío en lugar de entregar a un gateway, establezca la `acción` de la ruta P10 en `:saf`.

# Solución de problemas

## Suscriptores en la Red Ruteados al Gateway de Respaldo

**Síntomas**: Los mensajes a los suscriptores en la red toman la ruta fuera de la red/respaldo en lugar de la ruta en la red.

**Causas posibles**:

- `diameter_enabled` es `false` - el criterio `:on_net` falla en cerrado, por lo que la ruta en la red nunca coincide
- No hay pares de Diameter conectados (verifique los registros de la aplicación para el estado de conexión de pares)
- HSS está devolviendo errores o se está agotando el tiempo (verifique la telemetría de errores de Sh y las líneas de registro "HSS dip")
- La traducción de números está produciendo un formato de número diferente al que espera el HSS

**Resolución**:

1. Verifique `diameter_enabled: true` en `runtime.exs`
2. Verifique los registros para el estado de conexión de pares de Diameter al inicio (`Diameter Sh (HSS): ENABLED (... peers: N)`)
3. Busque líneas de registro `HSS dip:` y métricas de error de Sh
4. Confirme que el formato de número después de la traducción coincide con lo que espera el HSS (E.164 sin `+`)

## Mensajes Retenidos Pero Nunca Entregados

**Síntomas**: Los mensajes se retienen correctamente (`:saf`) para los suscriptores en la red pero nunca se entregan, incluso después de que el suscriptor se registra.

**Causas posibles**:

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

**Resolución**:

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

## Par de Diameter No Conectando

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

**Causas posibles**:

- IP o puerto de par incorrecto en la configuración

- Firewall bloqueando el puerto 3868
- Desajuste de identidad del host del par (el `host` configurado debe coincidir exactamente con la identidad de Diameter del par)
- Desajuste de dominio entre OmniMessage y el par

**Resolución**:

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

# Documentación de Métricas Prometheus de SMS-C

**← Volver al Índice de Documentación**

## 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 un servidor de Prometheus puede raspar. El sistema actualiza las métricas en tiempo real a medida que procesa mensajes.

## Convención de Nombres de Métricas

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

Categorías:

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

# Métricas de Licencia

## sms_c_license_status

**Tipo**: Gauge

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

**Valores**:

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

**Etiquetas**: Ninguna

**Nombre del Producto**: `omnimessage`

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

**Comportamiento Cuando la Licencia es Inválida**:

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

**Alertas**:

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

**Ejemplo de Consultas Prometheus**:

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

# Alertar sobre licencia inválida
sms_c_license_status == 0

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

---

# Métricas de Procesamiento de Mensajes

## sms_c_message_received_count

**Tipo**: Counter

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

**Etiquetas**:

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

**Caso de Uso**: Monitorear el volumen de mensajes entrantes por fuente y tipo. 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 origen) o picos (posible ataque/spam).

---

## sms_c_message_validated_count

**Tipo**: Counter

**Descripción**: Número total de validaciones de mensajes realizadas.

**Etiquetas**:

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

**Caso de Uso**: Rastrear tasas de éxito/fallo de validaciones. Altas tasas de fallo pueden indicar mensajes mal formados 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**: Histogram

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

**Unidad**: Milisegundos

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

**Etiquetas**:

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

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

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

---

# Métricas de Enrutamiento

## sms_c_routing_route_matched_count

**Tipo**: Counter

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

**Etiquetas**:

- `route_id`: Identificador único de la ruta coincidente

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

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

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

---

## sms_c_routing_failed_count

**Tipo**: Counter

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

**Etiquetas**:

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

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

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

---

## sms_c_routing_action_count

**Tipo**: Counter

**Descripción**: Número total de acciones especiales de enrutamiento tomadas.

**Etiquetas**:

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

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

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

---

## sms_c_routing_stop_duration

**Tipo**: Histogram

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

**Unidad**: Milisegundos

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

**Etiquetas**:

- `dest_smsc`: SMSC de destino seleccionado

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

**Alertas**: 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**: Counter

**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 de la caché cae por debajo del umbral (puede indicar problemas de caché o tráfico inusual).

---

## sms_c_enum_cache_miss_count

**Tipo**: Counter

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

**Etiquetas**:

- `domain`: Dominio ENUM consultado

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

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

---

## sms_c_enum_cache_size_size

**Tipo**: Gauge

**Descripción**: Número actual de entradas en la caché ENUM.

**Caso de Uso**: Monitorear el tamaño de la caché para asegurar que no crezca 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**: Histogram

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

**Unidad**: Milisegundos

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

**Etiquetas**:

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

**Caso de Uso**: Monitorear el rendimiento de la búsqueda 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**: Histogram

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

**Buckets**: 0, 1, 2, 3, 5, 10

**Etiquetas**:

- `domain`: Dominio ENUM consultado

**Caso de Uso**: Entender la distribución de registros ENUM. La mayoría de las búsquedas deberían devolver de 1 a 3 registros.

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

---

# Métricas de Entrega

## sms_c_delivery_queued_count

**Tipo**: Counter

**Descripción**: Número total de mensajes en cola para entrega a un SMSC de destino.

**Etiquetas**:

- `dest_smsc`: Nombre del SMSC de destino

**Caso de Uso**: Monitorear el flujo de mensajes a cada destino. Útil para la planificación de capacidad.

**Alertas**: Comparar con los conteos de éxito/fallo de entrega para detectar acumulación.

---

## sms_c_delivery_attempted_count

**Tipo**: Counter

**Descripción**: Número total de intentos de entrega realizados (incluye reintentos).

**Etiquetas**:

- `dest_smsc`: Nombre del SMSC de destino

**Caso de Uso**: Rastrear el volumen de intentos de entrega. Un alto conteo de intentos en relación con el conteo en cola indica comportamiento de reintento.

---

## sms_c_delivery_succeeded_count

**Tipo**: Counter

**Descripción**: Número total de mensajes entregados con éxito al SMSC de destino.

**Etiquetas**:

- `dest_smsc`: Nombre del SMSC de destino

**Caso de Uso**: Rastrear entregas exitosas por destino. Métrica principal de éxito.

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

**Cálculo**: `success_rate = succeeded / queued`

---

## sms_c_delivery_failed_count

**Tipo**: Counter

**Descripción**: Número total de mensajes que fallaron en la entrega después de todos los intentos de reintento.

**Etiquetas**:

- `dest_smsc`: Nombre del SMSC de destino
- `reason`: Razón del fallo

**Caso de Uso**: Rastrear fallos de entrega para identificar destinos problemáticos o patrones de fallo.

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

---

## sms_c_delivery_dead_letter_count

**Tipo**: Counter

**Descripción**: Número total de mensajes movidos a la cola de cartas muertas (indeseables).

**Etiquetas**:

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

**Caso de Uso**: Monitorear mensajes indeseables que requieren intervención manual.

**Alertas**: Alertar sobre cualquier evento de carta muerta ya que representan un fallo total de entrega.

---

## sms_c_delivery_succeeded_attempt_count

**Tipo**: Histogram

**Descripción**: Número de intentos de entrega requeridos antes de una entrega exitosa.

**Buckets**: 1, 2, 3, 5, 10

**Etiquetas**:

- `dest_smsc`: Nombre del SMSC de destino

**Caso de Uso**: Entender el comportamiento de reintentos. La mayoría de las entregas deberían tener éxito en el primer intento.

**Alertas**: Alertar si el conteo promedio de intentos excede 2 (indica problemas de confiabilidad del destino).

---

## sms_c_delivery_failed_attempt_count

**Tipo**: Histogram

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

**Buckets**: 1, 2, 3, 5, 10

**Etiquetas**:

- `dest_smsc`: Nombre del SMSC de destino

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

---

## sms_c_delivery_time_delta_delta_ms

**Tipo**: Histogram

**Descripción**: Tiempo desde la presentación del mensaje hasta la confirmación de entrega. Esta métrica captura la latencia completa de extremo a extremo desde que un mensaje entra en el SMS-C hasta que el sistema confirma la entrega. Proporciona etiquetas detalladas para análisis por fuente, destino y comportamiento de reintento.

**Unidad**: Milisegundos

**Buckets**: 50, 100, 250, 500, 1000, 2000, 5000, 10000, 20000, 60000, 600000 ms (50ms a 10 minutos)

**Etiquetas**:

- `source_smsc`: SMSC de origen que presentó el mensaje
- `dest_smsc`: SMSC de destino que entregó el mensaje
- `delivery_attempts`: Número de intentos de entrega requeridos (0 = éxito en el primer intento)

**Caso de Uso**: Analizar el rendimiento de entrega de extremo a extremo por par fuente-destino. Identificar caminos lentos, combinaciones problemáticas de SMSC y correlacionar la latencia con el comportamiento de reintento. Esencial para el monitoreo de SLA y la planificación de capacidad.

**Alertas**:

- Alertar cuando p95 exceda el umbral de SLA para combinaciones específicas de fuente/destino

- Alertar cuando los mensajes consistentemente requieran múltiples intentos

**Ejemplo de Consultas Prometheus**:

```
# Tiempo de entrega p95 por SMSC de origen y destino
histogram_quantile(0.95,
  sum by (source_smsc, dest_smsc, le) (
    rate(sms_c_delivery_time_delta_delta_ms_bucket[5m])
  )
)

# Tiempo de entrega p99 en general
histogram_quantile(0.99,
  sum by (le) (
    rate(sms_c_delivery_time_delta_delta_ms_bucket[5m])
  )
)

# Tiempo promedio de entrega por SMSC de origen
sum by (source_smsc) (rate(sms_c_delivery_time_delta_delta_ms_sum[5m]
/
sum by (source_smsc)
(rate(sms_c_delivery_time_delta_delta_ms_count[5m]))

# Tiempo de entrega para mensajes que requieren reintentos vs éxito e
el primer intento
histogram_quantile(0.95,
  sum by (le)
(rate(sms_c_delivery_time_delta_delta_ms_bucket{delivery_attempts="0"
[5m]))
)
histogram_quantile(0.95,
  sum by (le)
(rate(sms_c_delivery_time_delta_delta_ms_bucket{delivery_attempts!="0
[5m]))
)

# Pares de origen-destino más lentos (por p95)
topk(10,
  histogram_quantile(0.95,
    sum by (source_smsc, dest_smsc, le) (
      rate(sms_c_delivery_time_delta_delta_ms_bucket[1h])
    )
  )
)

# Porcentaje de mensajes entregados en 2 segundos
```

```
sum(rate(sms_c_delivery_time_delta_delta_ms_bucket{le="2000"}[5m])) /
sum(rate(sms_c_delivery_time_delta_delta_ms_count[5m])) * 100

# Mensajes entregados en 2 segundos por destino
sum by (dest_smsc)
(rate(sms_c_delivery_time_delta_delta_ms_bucket{le="2000"}[5m])) /
sum by (dest_smsc) (rate(sms_c_delivery_time_delta_delta_ms_count[5m])
* 100
```

---

## sms_c_delivery_time_by_smsc_delta_ms

**Tipo**: Histogram

**Descripción**: Tiempo desde la presentación del mensaje hasta la confirmación de entrega, etiquetado solo por SMSC de origen y destino. Esta es una variante de menor cardinalidad de `sms_c_delivery_time_delta_delta_ms` - omite la etiqueta `delivery_attempts` para que el tiempo promedio de entrega por par fuente/destino se pueda leer directamente sin agregar a través de los conteos de intentos. Emitido desde el mismo evento `[:sms_c, :delivery, :time_delta]`.

**Unidad**: Milisegundos

**Buckets**: 50, 100, 250, 500, 1000, 2000, 5000, 10000, 20000, 60000, 600000 ms (50ms a 10 minutos)

**Etiquetas**:

- `source_smsc`: SMSC de origen que presentó el mensaje
- `dest_smsc`: SMSC de destino que entregó el mensaje

**Caso de Uso**: Mostrar el tiempo promedio de entrega de extremo a extremo por par fuente/destino. Preferir esta métrica sobre `sms_c_delivery_time_delta_delta_ms` para paneles promedio/SLA donde no se necesita el desglose por conteo de intentos; usar la variante `_delta` cuando necesite segmentar por comportamiento de reintento.

**Ejemplo de Consultas Prometheus**:

```
# Tiempo promedio de entrega (ms) por SMSC de origen/destino
sum by (source_smsc, dest_smsc)
(rate(sms_c_delivery_time_by_smsc_delta_ms_sum[5m])) /
sum by (source_smsc, dest_smsc)
(rate(sms_c_delivery_time_by_smsc_delta_ms_count[5m]))

# Tiempo promedio de entrega general
sum(rate(sms_c_delivery_time_by_smsc_delta_ms_sum[5m])) /
sum(rate(sms_c_delivery_time_by_smsc_delta_ms_count[5m]))

# Tiempo de entrega p95 por SMSC de origen/destino
histogram_quantile(0.95,
  sum by (source_smsc, dest_smsc, le) (
    rate(sms_c_delivery_time_by_smsc_delta_ms_bucket[5m])
  )
)
```

---

# 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 fallido (esperando 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 impacta las tasas de entrega.

---

## sms_c_queue_size_delivered

**Tipo**: Gauge

**Descripción**: Número actual de mensajes entregados esperando limpieza/eliminación de la cola.

**Etiquetas**:

- `queue_type`: Tipo de cola

**Caso de Uso**: Monitorear el retraso de 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 mensajes y paradas 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

El cobro se realiza a través de la interfaz **Diameter Ro** (Control de Crédito). La etiqueta `account` lleva el MSISDN del suscriptor cobrado (el remitente del mensaje). Un evento `charging_failed` se emite tanto para un rechazo definitivo por falta de saldo (`reason: :no_credit`) como para un error de OCS (transporte/protocolo). Solo un rechazo `:no_credit` rechaza un mensaje

(estado `:balance_rejected`); los errores de OCS fallan abiertamente y el mensaje se entrega.

## sms_c_charging_requested_count

**Tipo**: Counter

**Descripción**: Número total de solicitudes de cobro (Diameter Ro CCR Event-Requests) enviadas al OCS.

**Etiquetas**:

- `account`: MSISDN del suscriptor cobrado (remitente del mensaje)

**Caso de Uso**: Rastrear el volumen de cobro por cuenta. Útil para la conciliación de facturación.

---

## sms_c_charging_succeeded_count

**Tipo**: Counter

**Descripción**: Número total de operaciones de cobro exitosas.

**Etiquetas**:

- `account`: Identificador de cuenta cobrada

**Caso de Uso**: Monitorear la tasa de éxito de cobro por cuenta.

**Cálculo**: `success_rate = succeeded / requested`

---

## sms_c_charging_failed_count

**Tipo**: Counter

**Descripción**: Número total de operaciones de cobro fallidas.

**Etiquetas**:

- `account`: Identificador de cuenta
- `reason`: Razón del fallo

**Caso de Uso**: Identificar fallos de cobro que pueden impactar los ingresos o requerir intervención de la cuenta.

**Alertas**: Alertar sobre tasas de fallo de cobro elevadas.

---

## sms_c_charging_succeeded_duration

**Tipo**: Histogram

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

**Unidad**: Milisegundos

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

**Etiquetas**:

- `account`: Identificador de cuenta

**Caso de Uso**: Monitorear el rendimiento del sistema de facturación. Un cobro lento puede retrasar la entrega del mensaje.

**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. Se consulta cada 10 segundos.

**Etiquetas**:

- `table`: Nombre de la tabla

**Tablas Rastreadas**:

- `sms_route` - Reglas de enrutamiento de SMS
- `message_store` - Cola de mensajes (mensajes pendientes, entregados, fallidos)
- `location_store` - Datos de ubicación/registro de suscriptores
- `frontend_store` - Registros de frontend/SMSC
- `translation_rule` - Reglas de traducción de números
- `cell_tower_store` - Datos de ubicación de torres celulares
- `message_events` - Registros de eventos de mensajes

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

**Alertas**: Alertar sobre tasas de crecimiento de tablas inesperadas o cuando se acerquen a los límites de capacidad.

**Ejemplo de Consultas Prometheus**:

```
# Todos los tamaños de tabla
sms_c_mnesia_table_size_record_count

# Tamaño de la cola de mensajes
sms_c_mnesia_table_size_record_count{table="message_store"}

# Registros de suscriptores activos
sms_c_mnesia_table_size_record_count{table="location_store"}

# Conteo de reglas de enrutamiento
sms_c_mnesia_table_size_record_count{table="sms_route"}
```

---

## sms_c_frontend_status_count

**Tipo**: Gauge

**Descripción**: Número de frontends en cada estado de conexión.

**Etiquetas**:

- `frontend_name`: Identificador del frontend
- `status`: Estado de conexión (conectado, desconectado)

**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**: Counter

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

**Etiquetas**:

- `location`: Nombre del frontend/SMSC donde está registrado el suscriptor
- `ims_capable`: Si el suscriptor soporta IMS (true/false)

**Caso de Uso**: Monitorear la actividad de registro de suscriptores. Rastrear suscriptores IMS vs no IMS. Detectar tormentas de registro o fallos.

**Alertas**: Establecer alertas para:

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

**Ejemplo de Consulta**:

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

# Ratio de registro IMS vs no IMS
sum(rate(sms_c_location_registered_count{ims_capable="true"}[5m]))
/
sum(rate(sms_c_location_registered_count[5m]))
```

---

## sms_c_location_active_registrations_count

**Tipo**: Gauge

**Descripción**: Número actual de registros de suscriptores activos, agrupados por ubicación (frontend/SMSC) y capacidad IMS. Esta métrica se consulta cada 10 segundos y refleja el estado actual de la tienda de ubicaciones.

**Etiquetas**:

- `location`: Nombre del frontend/SMSC donde están registrados los suscriptores
- `ims_capable`: Si los registros son capaces de IMS (true/false)

**Caso de Uso**: Monitorear la distribución actual de suscriptores a través de frontends. Rastrear la adopción de IMS. Identificar desequilibrios de carga entre frontends. Planificación de capacidad para cada instancia de SMSC.

**Alertas**: Establecer alertas para:

- Caídas repentinas en registros para una ubicación (puede indicar problemas de frontend)
- Distribución desequilibrada entre frontends
- Registros totales acercándose a los límites de capacidad

**Ejemplo de Consultas Prometheus**:

```
# Total de registros activos
sum(sms_c_location_active_registrations_count)

# Registros activos por ubicación
sum by (location) (sms_c_location_active_registrations_count)

# Registros IMS-capaces activos por ubicación
sum by (location)
(sms_c_location_active_registrations_count{ims_capable="true"})

# Tasa de adopción de IMS por ubicación
sum by (location)
(sms_c_location_active_registrations_count{ims_capable="true"}) /
sum by (location) (sms_c_location_active_registrations_count) *
100

# Total de registros IMS vs no IMS
sum by (ims_capable) (sms_c_location_active_registrations_count)

# Ubicaciones con más registros
topk(10, sum by (location)
(sms_c_location_active_registrations_count))

# Distribución de registros como porcentaje del total
sum by (location) (sms_c_location_active_registrations_count) /
sum(sms_c_location_active_registrations_count) * 100
```

---

# Métricas de Solicitudes de API HTTP

## phoenix_endpoint_stop_duration

**Tipo**: Distribución (Histogram)

**Descripción**: Duración del procesamiento de solicitudes HTTP en milisegundos, desde el inicio de la solicitud hasta la finalización de la respuesta.

**Etiquetas**:

- `method`: Método HTTP (por ejemplo, `GET`, `POST`)
- `path`: ruta de solicitud con segmentos de alta cardinalidad (UUIDs, IDs numéricos) colapsados a `/:id` (por ejemplo, `/api/messages`, `/api/messages/:id`)

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

**Caso de Uso**: Monitorear el rendimiento de la API. Identificar endpoints lentos. Rastrear SLAs de tiempo de respuesta.

**Alertas**: Establecer alertas para:

- Latencia P95 > 500ms para endpoints críticos
- Latencia P99 > 1s para cualquier endpoint
- Tendencias de latencia en aumento

**Ejemplo de Consulta**:

```
# Tiempo de respuesta p95 por endpoint
histogram_quantile(0.95,
  rate(phoenix_endpoint_stop_duration_bucket[5m]))

# Solicitudes más lentas de 1 segundo
sum(rate(phoenix_endpoint_stop_duration_bucket{le="1000"}[5m]))
```

---

# phoenix_endpoint_stop_count

**Tipo**: Counter

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

**Etiquetas**:

- `method`: Método HTTP (por ejemplo, `GET`, `POST`)
- `path`: ruta de solicitud (segmentos de alta cardinalidad colapsados a `/:id`)

- `status`: código de estado HTTP (200, 201, 400, 404, 500, etc.)

**Caso de Uso**: Monitorear el volumen de solicitudes de API y tasas de éxito. Rastrear tasas de error por endpoint.

**Alertas**: Establecer alertas para:

- Tasa de error > 5% para cualquier endpoint
- Errores 5xx en endpoints críticos
- Caídas repentinas en el volumen de solicitudes

**Ejemplo de Consulta**:

```
# Tasa de solicitudes por ruta
sum by (path) (rate(phoenix_endpoint_stop_count[5m]))

# Tasa de error por ruta
sum by (path) (rate(phoenix_endpoint_stop_count{status=~"5.."}
[5m])) /
sum by (path) (rate(phoenix_endpoint_stop_count[5m]))

# Tasa de éxito
sum(rate(phoenix_endpoint_stop_count{status=~"2.."}[5m])) /
sum(rate(phoenix_endpoint_stop_count[5m]))
```

---

# phoenix_router_dispatch_exception_count

**Tipo**: Counter

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

**Etiquetas**:

- `route`: Ruta del endpoint de API donde ocurrió la excepción
- `kind`: Tipo de excepción (error, exit, throw)

**Caso de Uso**: Rastrear errores de aplicación. Identificar endpoints problemáticos. Monitorear la estabilidad del sistema.

**Alertas**: Establecer alertas para cualquier valor distinto de cero en endpoints críticos.

**Ejemplo de Consulta**:

```
# Tasa de excepciones por endpoint
rate(phoenix_router_dispatch_exception_count[5m])

# Total de excepciones en la última hora
increase(phoenix_router_dispatch_exception_count[1h])
```

---

# Métricas de Sh de Diameter

El SMS-C consulta el HSS a través de la interfaz **Diameter Sh** (3GPP TS 29.329) para clasificar a los suscriptores como en red o fuera de red durante la evaluación de rutas. Estas métricas se emiten solo cuando Diameter está habilitado.

## sms_c_diameter_peer_status

**Tipo**: Gauge

**Descripción**: Si el par Diameter está conectado (`1`) o no (`0`).

**Etiquetas**:

- `origin_host`: Origin-Host Diameter configurado del par
- `ip`: dirección IP del par

**Ejemplo de Consulta**:

```
# Pares actualmente caídos
sms_c_diameter_peer_status == 0
```

---

## sms_c_diameter_sh_request_count

**Tipo**: Counter

**Descripción**: Número total de solicitudes de búsqueda Diameter Sh (UDR) enviadas.

---

## sms_c_diameter_sh_response_count

**Tipo**: Counter

**Descripción**: Número total de respuestas de búsqueda Diameter Sh (UDA), por resultado.

**Etiquetas**:

- `result`: resultado de la búsqueda (por ejemplo, `on_net`, `unknown_user`, `error`)
- `result_code`: Resultado Experimental de Diameter / Código de Resultado (por ejemplo, `2001`, `5001`)

**Ejemplo de Consulta**:

```
# Ratio de errores de búsqueda Sh
sum(rate(sms_c_diameter_sh_response_count{result="error"}[5m]))
  / sum(rate(sms_c_diameter_sh_response_count[5m]))
```

---

## sms_c_diameter_sh_response_duration_ms

**Tipo**: Distribución (Histogram)

**Descripción**: Duración de búsqueda Diameter Sh en milisegundos.

**Etiquetas**:

- `result`: resultado de la búsqueda

**Buckets**: 5, 10, 25, 50, 100, 250, 500, 1000, 2500 ms

**Ejemplo de Consulta**:

```
# Latencia de búsqueda Sh p95
histogram_quantile(0.95,
rate(sms_c_diameter_sh_response_duration_ms_bucket[5m]))
```

---

## sms_c_diameter_sh_error_count

**Tipo**: Counter

**Descripción**: Número total de errores de búsqueda Diameter Sh (fallos de transporte/tiempo de espera/decodificación, distintos de un UDA negativo).

**Etiquetas**:

- `reason`: razón del error

---

# Métricas de Federación

La federación reenvía mensajes entre controladores SMS-C a través de un modelo de pares basado en extracción. Ver Federación.

## sms_c_federation_forward_attempted_count

**Tipo**: Counter

**Descripción**: Número total de mensajes reenviados a controladores remotos.

**Etiquetas**:

- `dest_controller`: identidad del controlador de destino

---

## sms_c_federation_forward_completed_count

**Tipo**: Counter

**Descripción**: Número total de mensajes entregados con éxito a controladores remotos.

**Etiquetas**:

- `dest_controller`: identidad del controlador de destino

---

## sms_c_federation_forward_queued_count

**Tipo**: Counter

**Descripción**: Número total de mensajes en cola porque un par no estaba disponible.

**Etiquetas**:

- `dest_controller`: identidad del controlador de destino

---

## sms_c_federation_forward_failed_count

**Tipo**: Counter

**Descripción**: Número total de mensajes que excedieron el número máximo de reintentos.

**Etiquetas**:

- `dest_controller`: identidad del controlador de destino

---

## sms_c_federation_message_received_count

**Tipo**: Counter

**Descripción**: Número total de mensajes recibidos de controladores remotos.

**Etiquetas**:

- `origin_node`: nodo del controlador de origen

---

## sms_c_federation_health_check_count

**Tipo**: Counter

**Descripción**: Número total de intentos de verificación de salud de pares, por par y resultado.

**Etiquetas**:

- `peer`: identidad del par
- `result`: resultado de la verificación de salud (por ejemplo, `healthy`, `unhealthy`)

---

## sms_c_federation_peer_status_count

**Tipo**: Gauge

**Descripción**: Número actual de pares de federación por estado.

**Etiquetas**:

- `status`: estado del par (por ejemplo, `healthy`, `degraded`, `unhealthy`)

---

## sms_c_federation_queue_depth_count

**Tipo**: Gauge

**Descripción**: Profundidad actual de la cola de reenvío de la federación.

**Ejemplo de Consulta**:

```
# Alertar si la cola de reenvío se está acumulando
sms_c_federation_queue_depth_count > 1000
```

---

# 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 alta sostenida.

---

## 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 (por defecto 262,144).

---

# Recolección y Sondeo de Métricas

El sistema recolecta 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**: `(sms_c_delivery_succeeded_count) / (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_time_delta_delta_ms` (entrega)

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

---

## Efectividad de la Caché ENUM

Medir qué tan bien está funcionando la caché ENUM:

**Fórmula**: `(sms_c_enum_cache_hit_count) / (sms_c_enum_cache_hit_count + 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 la mayor parte del tráfico:

**Métrica**: `sms_c_routing_route_matched_count` agrupada por `route_id`

**Interpretación**: Utilizar 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` (edad en tendencia)

**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 confiabilidad del destino.

---

# Alertas Recomendadas

| Alerta | Condición | Sever |
|---|---|---|
| Alta Tasa de Fallos de Enrutamiento | Aumento de `routing_failed_count` | Crítica |
| Acumulación de Cola | `queue_size_pending` > umbral | Advert |
| Mensajes Antiguos en Cola | `queue_oldest_message_age_seconds` > 300 | Crítica |
| Pico de Fallos de Entrega | Pico en `delivery_failed_count` | Alta |
| Eventos de Carta Muerta | `delivery_dead_letter_count` > 0 | Alta |
| Tiempos de Espera de Búsqueda ENUM | p95 de `enum_lookup_stop_duration` > 5000ms | Advert |
| Baja Tasa de Aciertos en Caché | Tasa de aciertos de caché ENUM < 0.7 | Advert |
| Frontend Desconectado | `frontend_status_count{status="disconnected"}` > 0 | Alta |
| Fallos de Cobro | `charging_failed_count` > umbral | Alta |

| Alerta | Condición | Sever |
| --- | --- | --- |
| Procesamiento Lento de Mensajes | p95 de `message_processing_stop_duration` > 1000ms | Advert |

---

# Recomendaciones de Dashboard

## Dashboard de Operaciones

**Propósito**: Monitoreo en tiempo real de la salud del sistema

**Paneles**:

1. Throughput de mensajes (recibidos/procesados/entregados por minuto)
2. Tamaños de cola (pendientes, fallidos, entregados)
3. Tasa de éxito de entrega por destino
4. Latencia de procesamiento y entrega p95
5. Estado de frontends activos
6. Alertas actuales

---

## Dashboard de Rendimiento

**Propósito**: Análisis del rendimiento del sistema

**Paneles**:

1. Histograma de duración del procesamiento de mensajes
2. Histograma de duración de enrutamiento
3. Histograma de duración de búsqueda ENUM
4. Histograma de duración de cobro
5. Distribución de intentos de entrega
6. Tasas de aciertos en caché

## Dashboard de Negocios

**Propósito**: Análisis de tráfico y uso

**Paneles**:

1. Mensajes por SMSC de origen
2. Mensajes por SMSC de destino
3. Mapa de calor de utilización de rutas
4. Conteos de acciones de respuesta automática y eliminación
5. Estadísticas de uso de ENUM
6. Volumen de cobro por cuenta

---

# Retención de Métricas

Configuraciones de retención recomendadas para Prometheus:

- **Métricas en bruto**: 15 días
- **Agregados de 5 minutos**: 90 días
- **Agregados de 1 hora**: 2 años

Esto proporciona un historial reciente detallado mientras se mantienen tendencias a largo plazo para la planificación de capacidad.

---

# Solución de Problemas con Métricas

## Escenario: Mensajes No Entregados

**Pasos de Investigación**:

1. Verificar `sms_c_message_received_count` - ¿Se están recibiendo mensajes?

2. Verificar `sms_c_routing_failed_count` - ¿Se están enrutando?
3. Verificar `sms_c_delivery_queued_count` - ¿Se están encolando?
4. Verificar `sms_c_delivery_failed_count` - ¿Están fallando los intentos de entrega?
5. Verificar etiquetas `dest_smsc` para identificar destinos problemáticos

---

# Escenario: Procesamiento Lento de Mensajes

**Pasos de Investigación**:

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

---

# Ejemplos de Consultas Prometheus

## Throughput 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 Fallos de Entrega (Porcentaje)**:

```
(rate(sms_c_delivery_failed_count[5m]) /
rate(sms_c_delivery_queued_count[5m])) * 100
```

**Intentos Promedio de Entrega (p95)**:

```
histogram_quantile(0.95,
sms_c_delivery_succeeded_attempt_count_bucket)
```

**Éxito de Entrega por Destino**:

```
sum by (dest_smsc) (rate(sms_c_delivery_succeeded_count[5m]))
```

**Razones de Fallo de Entrega**:

```
sum by (reason) (rate(sms_c_delivery_failed_count[5m]))
```

**Tiempo de Entrega (p95)**:

```
histogram_quantile(0.95,
sms_c_delivery_time_delta_delta_ms_bucket)
```

**Tiempo de Entrega (p99)**:

```
histogram_quantile(0.99,
sms_c_delivery_time_delta_delta_ms_bucket)
```

# Métricas de Cola

**Mensajes Pendientes Actuales**:

```
sms_c_queue_size_pending
```

**Mensajes Fallidos Esperando Reintento**:

```
sms_c_queue_size_failed
```

**Edad del Mensaje Más Antiguo (Minutos)**:

```
sms_c_queue_oldest_message_age_seconds / 60
```

**Tasa de Crecimiento de 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])))
```

**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 de 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 de Caché ENUM**:

```
(rate(sms_c_enum_cache_hit_count[5m]) /
(rate(sms_c_enum_cache_hit_count[5m]) +
rate(sms_c_enum_cache_miss_count[5m])) ) * 100
```

**Latencia de Búsqueda ENUM (p95)**:

```
histogram_quantile(0.95, sms_c_enum_lookup_stop_duration_bucket)
```

**Búsquedas ENUM por Segundo (Cacheadas vs No Cacheadas)**:

```
# Cacheadas (rápido)
rate(sms_c_enum_cache_hit_count[5m])

# No cacheadas (requiere consulta DNS)
rate(sms_c_enum_cache_miss_count[5m])
```

**Promedio de Registros NAPTR Devueltos**:

```
rate(sms_c_enum_naptr_records_record_count_sum[5m]) /
rate(sms_c_enum_naptr_records_record_count_count[5m])
```

**Tamaño de la Caché ENUM**:

```
sms_c_enum_cache_size_size
```

# Rendimiento de Procesamiento

**Latencia de Procesamiento de Mensajes (p95)**:

```
histogram_quantile(0.95,
sms_c_message_processing_stop_duration_bucket)
```

**Latencia de Procesamiento de Mensajes (p99)**:

```
histogram_quantile(0.99,
sms_c_message_processing_stop_duration_bucket)
```

**Fallos de Procesamiento**:

```
rate(sms_c_message_processing_stop_duration_count{success="false"}
[5m])
```

**Tasa de Fallos de Validación**:

```
rate(sms_c_message_validated_count{valid="false"}[5m]) /
rate(sms_c_message_validated_count[5m])
```

# Métricas de Cobro

**Tasa de Éxito de Cobro**:

```
rate(sms_c_charging_succeeded_count[5m]) /
rate(sms_c_charging_requested_count[5m])
```

**Fallos de Cobro por Minuto**:

```
rate(sms_c_charging_failed_count[5m]) * 60
```

**Latencia de Cobro (p95)**:

```
histogram_quantile(0.95, sms_c_charging_succeeded_duration_bucket)
```

**Volumen de Cobro por Cuenta**:

```
sum by (account) (rate(sms_c_charging_requested_count[1h]))
```

# Salud del Frontend

**Frontends Activos**:

```
sum(sms_c_frontend_status_count{status="connected"})
```

**Frontends Desconectados**:

```
sum(sms_c_frontend_status_count{status="disconnected"})
```

**Frontends por Nombre**:

```
sum by (frontend_name)
(sms_c_frontend_status_count{status="connected"})
```

## Salud del Sistema

**Tamaños de Tablas Mnesia**:

```
sms_c_mnesia_table_size_record_count
```

**Conteo de Rutas**:

```
sms_c_mnesia_table_size_record_count{table="sms_route"}
```

**Conteo de Reglas de Traducción**:

```
sms_c_mnesia_table_size_record_count{table="translation_rule"}
```

# Ejemplos de Dashboard de Grafana

## Dashboard 1: Operaciones en Tiempo Real

**Propósito**: Monitorear la actividad y salud actual del sistema.

**Paneles**:

1. **Throughput de Mensajes (Gráfico)**

- Consulta: `rate(sms_c_message_received_count[5m])`
- Consulta: `rate(sms_c_delivery_succeeded_count[5m])`
- Unidad: mensajes/segundo
- Leyenda: {{source_type}}

2. **Tasa de Éxito de Entrega (Gauge)**

- Consulta: `(rate(sms_c_delivery_succeeded_count[5m]) / rate(sms_c_delivery_queued_count[5m])) * 100`
- Unidad: porcentaje (0-100)
- Umbrales:
  - Rojo: < 90
  - Amarillo: 90-95
  - Verde: > 95

3. **Profundidad de Cola (Gráfico)**

- Consulta: `sms_c_queue_size_pending`
- Consulta: `sms_c_queue_size_failed`
- Unidad: mensajes
- Leyenda: {{queue_type}}

4. **Edad del Mensaje Más Antiguo (Estadística)**

- Consulta: `sms_c_queue_oldest_message_age_seconds / 60`
- Unidad: minutos
- Umbrales:
  - Verde: < 5
  - Amarillo: 5-10
  - Rojo: > 10

5. **Frontends Activos (Estadística)**

- Consulta: `sum(sms_c_frontend_status_count{status="connected"})`
- Unidad: conteo
- Color: Azul

6. **Fallos de Enrutamiento (Gráfico)**

- Consulta: `rate(sms_c_routing_failed_count[5m]) * 60`
- Unidad: fallos/minuto
- Umbral de alerta: > 0

# Dashboard 2: Análisis de Rendimiento

**Propósito**: Analizar el rendimiento del sistema e identificar cuellos de botella.

**Paneles**:

1. **Latencia de Extremo a Extremo (Gráfico)**
   - Consulta: `histogram_quantile(0.50, sms_c_message_processing_stop_duration_bucket)` (p50)
   - Consulta: `histogram_quantile(0.95, sms_c_message_processing_stop_duration_bucket)` (p95)
   - Consulta: `histogram_quantile(0.99, sms_c_message_processing_stop_duration_bucket)` (p99)
   - Unidad: milisegundos
   - Leyenda: Percentil
2. **Latencias de Componentes (Barra Gauge)**
   - Enrutamiento: `histogram_quantile(0.95, sms_c_routing_stop_duration_bucket)`
   - ENUM: `histogram_quantile(0.95, sms_c_enum_lookup_stop_duration_bucket)`
   - Cobro: `histogram_quantile(0.95, sms_c_charging_succeeded_duration_bucket)`
   - Entrega: `histogram_quantile(0.95, sms_c_delivery_time_delta_delta_ms_bucket)`
   - Unidad: milisegundos
   - Barras horizontales
3. **Distribución de Intentos de Entrega (Mapa de Calor)**
   - Consulta: `sms_c_delivery_succeeded_attempt_count_bucket`

- Muestra cuántos intentos se necesitan típicamente
- Escala de color: Azul (1 intento) a Rojo (muchos intentos)

4. **Rendimiento de Caché ENUM (Gráfico)**

   - Tasa de Aciertos: `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`
   - Eje Y dual (tasa vs tamaño)

5. **Tasa de Éxito de Procesamiento (Gauge)**

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

# Dashboard 3: Análisis de Tráfico

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

**Paneles**:

1. **Mensajes por Tipo de Origen (Gráfico Circular)**

   - Consulta: `sum by (source_type) (increase(sms_c_message_received_count[1h]))`
   - Muestra distribución: IMS vs CS vs SMPP

2. **Mensajes por SMSC de Origen (Gráfico de Barras)**

- Consulta: `sum by (source_smsc) (rate(sms_c_message_received_count[1h]))`
- Top 10 fuentes
- Barras horizontales

3. **Utilización de Rutas (Tabla)**
   - Columnas:
     - Route ID
     - SMSC de Destino
     - Mensajes (1h): `sum by (route_id, dest_smsc) (increase(sms_c_routing_route_matched_count[1h]))`
     - Prioridad
     - Tasa de Éxito
   - Ordenado por conteo de mensajes

4. **Entrega por Destino (Gráfico)**
   - Consulta: `sum by (dest_smsc) (rate(sms_c_delivery_succeeded_count[5m]))`
   - Unidad: mensajes/segundo
   - Gráfico de área apilada
   - Leyenda: {{dest_smsc}}

5. **Acciones de Eliminación/Respuesta Automática (Estadística)**
   - Eliminados: `increase(sms_c_routing_action_count{action="drop"}[1h])`
   - Respuesta Automática: `increase(sms_c_routing_action_count{action="auto_reply"}[1h])`
   - Estadísticas lado a lado

6. **Patrón de Tráfico Horario (Gráfico)**
   - Consulta: `rate(sms_c_message_received_count[1h]) * 3600`
   - Rango de tiempo: Últimos 7 días
   - Muestra patrones diarios

# Dashboard 4: Capacidad y Recursos

**Propósito**: Monitorear el uso de recursos y límites de capacidad.

**Paneles**:

1. **Capacidad de Cola (Gráfico)**
   - Actual: `sms_c_queue_size_size`
   - Línea de capacidad: Valor fijo basado en límites del sistema
   - Muestra tendencia de utilización
2. **Crecimiento de Tablas de Base de Datos (Gráfico)**
   - Mensajes: `sms_c_mnesia_table_size_record_count{table="sms_route"}`
   - Traducciones: `sms_c_mnesia_table_size_record_count{table="translation_rule"}`
   - Tendencia durante los últimos 30 días
3. **Tendencia de Acumulación de Mensajes (Gráfico)**
   - Consulta: `rate(sms_c_delivery_queued_count[5m]) - (rate(sms_c_delivery_succeeded_count[5m]) + rate(sms_c_delivery_failed_count[5m]))`
   - Positivo = acumulación creciente
   - Negativo = poniéndose al día
4. **Tráfico Máximo (Estadística)**
   - Consulta: `max_over_time(rate(sms_c_message_received_count[5m])[24h:])`
   - Muestra la tasa más alta de 5 minutos en las últimas 24 horas
   - Unidad: mensajes/segundo
5. **Utilización de Capacidad (Gauge)**
   - Consulta: `(rate(sms_c_message_received_count[5m]) / MAX_CAPACITY) * 100`

- Reemplace MAX_CAPACITY con su límite del sistema
- Unidad: porcentaje
- Umbrales:
  - Verde: < 70
  - Amarillo: 70-85
  - Rojo: > 85

# Dashboard 5: Cumplimiento de SLA

**Propósito**: Rastrear métricas de SLA y cumplimiento.

**Paneles**:

1. **Cumplimiento de SLA (Gauge)**
   - Éxito de Entrega: `(rate(sms_c_delivery_succeeded_count[1h]) / rate(sms_c_delivery_queued_count[1h])) * 100`
   - Línea objetivo en 99%
   - Umbrales:
     - Rojo: < 95
     - Amarillo: 95-99
     - Verde: >= 99
2. **Mensajes Entregados Dentro del SLA (Estadística)**
   - Consulta: `count(sms_c_delivery_time_delta_delta_ms_bucket{le="5000"}) / count(sms_c_delivery_time_delta_delta_ms_bucket)`
   - Muestra porcentaje entregado dentro de 5 segundos
   - Unidad: porcentaje
3. **Violaciones de SLA (Counter)**
   - Mensajes que exceden 5 minutos: `increase(sms_c_queue_oldest_message_age_seconds{} > 300)[24h:]`
   - Debería ser 0
4. **Tiempo de Actividad (Estadística)**

- Consulta: `up{job="sms-c"}`
- Binario: 1 = activo, 0 = inactivo
- Muestra estado actual

5. **Tendencia de Tasa de Éxito Diaria (Gráfico)**

   - Consulta: `avg_over_time((rate(sms_c_delivery_succeeded_count[1h]) / rate(sms_c_delivery_queued_count[1h]))[24h:1h])`
   - Rango de tiempo: Últimos 30 días
   - Línea de SLA en 99%

# Ejemplos de Reglas de Alerta

## Alertas Críticas

**Fallos de Enrutamiento**:

```
alert: RoutingFailuresDetected
expr: increase(sms_c_routing_failed_count[5m]) > 0
for: 2m
labels:
  severity: critical
annotations:
  summary: "{{ $value }} fallos de enrutamiento en los últimos 5 minutos"
  description: "Los mensajes no pueden ser enrutados. Verifique la configuración de enrutamiento."
```

**Acumulación de Cola**:

```
alert: MessageQueueBacklog
expr: sms_c_queue_size_pending > 10000
for: 5m
labels:
  severity: critical
annotations:
  summary: "La cola de mensajes tiene {{ $value }} mensajes
pendientes"
  description: "La cola se está acumulando. Verifique el
rendimiento de entrega."
```

**Mensajes Antiguos en Cola**:

```
alert: OldMessagesInQueue
expr: sms_c_queue_oldest_message_age_seconds > 300
for: 2m
labels:
  severity: critical
annotations:
  summary: "El mensaje más antiguo tiene {{ $value }} segundos"
  description: "Los mensajes no se están entregando. Verifique los
frontends."
```

**Todos los Frontends Desconectados**:

```
alert: NoActiveFrontends
expr: sum(sms_c_frontend_status_count{status="connected"}) == 0
for: 1m
labels:
  severity: critical
annotations:
  summary: "No hay frontends conectados"
  description: "No hay ruta de entrega disponible. Verifique la
conectividad del frontend."
```

**Crecimiento de la Cola de Cartas Muertas**:

```
alert: DeadLetterMessagesIncreasing
expr: rate(sms_c_delivery_dead_letter_count[10m]) > 0
for: 5m
labels:
  severity: critical
annotations:
  summary: "{{ $value }} mensajes movidos a la cola de cartas
muertas"
  description: "Los mensajes se están volviendo indeseables.
Investigue los fallos."
```

## Alertas de Advertencia

**Baja Tasa de Éxito de Entrega**:

```
alert: LowDeliverySuccessRate
expr: (rate(sms_c_delivery_succeeded_count[10m]) /
rate(sms_c_delivery_queued_count[10m])) < 0.95
for: 10m
labels:
  severity: warning
annotations:
  summary: "La tasa de éxito de entrega es {{ $value |
humanizePercentage }}"
  description: "La tasa de éxito está por debajo del 95%.
Investigue 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 confiabilidad del destino."
```

**Procesamiento Lento de Mensajes**:

```
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 de caché ENUM: {{ $value |
humanizePercentage }}"
  description: "Baja eficiencia de la 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 de OCS."
```

# Notas Adicionales

- Todas las métricas de duración utilizan precisión de nanosegundos internamente pero se convierten a milisegundos para el informe.
- Las métricas de contador son acumulativas y deben ser utilizadas con las funciones `rate()` o `increase()` en las consultas de Prometheus.
- Las métricas de gauge representan valores instantáneos en el momento de la recolección.
- Las métricas de histograma proporcionan cálculos de percentiles (p50, p95, p99) y pueden ser utilizadas para crear mapas de calor.
- Todas las métricas incluyen etiquetas predeterminadas añadidas por Prometheus (instancia, trabajo, etc.).

- Al crear dashboards, utilizar rangos de tiempo apropiados: 5m para tiempo real, 1h para tendencias, 24h+ para planificación de capacidad.
- Configurar reglas de grabación en Prometheus para consultas complejas de uso frecuente para mejorar el rendimiento del dashboard.
- Utilizar plantillas de variables en Grafana para dashboards dinámicos (seleccionar dest_smsc, source_smsc, etc.).

# Centro MMS (MMSc)

**← Volver al Índice de Documentación** | **Referencia de Configuración** | **Referencia de API**

## Tabla de Contenidos


- Descripción General
- Arquitectura
- Procesamiento de Envío de Mensajes
- Flujos de Entrega
- Notificación MMS
- Determinación On/Off-Net
- Configuración
- Almacenamiento
- Comportamiento en Tiempo de Ejecución
- Observabilidad
- Solución de Problemas


## Descripción General

OmniMessage incluye un **Centro MMS (MMSc)** nativo que recibe, almacena y entrega mensajes multimedia. Implementa las dos interfaces de interconexión MMS definidas por 3GPP TS 23.140: **MM1** para comunicación directa con el teléfono (HTTP/WAP) y **MM4** para retransmisión entre operadores (SMTP).

El MMSc está integrado en el núcleo de OmniMessage. No se ejecuta como un servicio separado y no tiene dependencia de bases de datos externas o servidores web. Cuando un suscriptor envía o recibe un MMS, el MMSc:

1. Almacena el cuerpo MMS sin procesar en el sistema de archivos local y su metadato en Mnesia.

2. Construye una **notificación MMS** y la envía como un SMS terminado en móvil ordinario, ya sea en la cola de mensajes de este nodo o al SMS-C configurado en `notify_api_url`. El SMS-C luego traduce, enruta y entrega como cualquier otro SMS.
3. Sirve el MMS almacenado de vuelta al teléfono que lo recupera a través de MM1 cuando obtiene la URL de notificación.

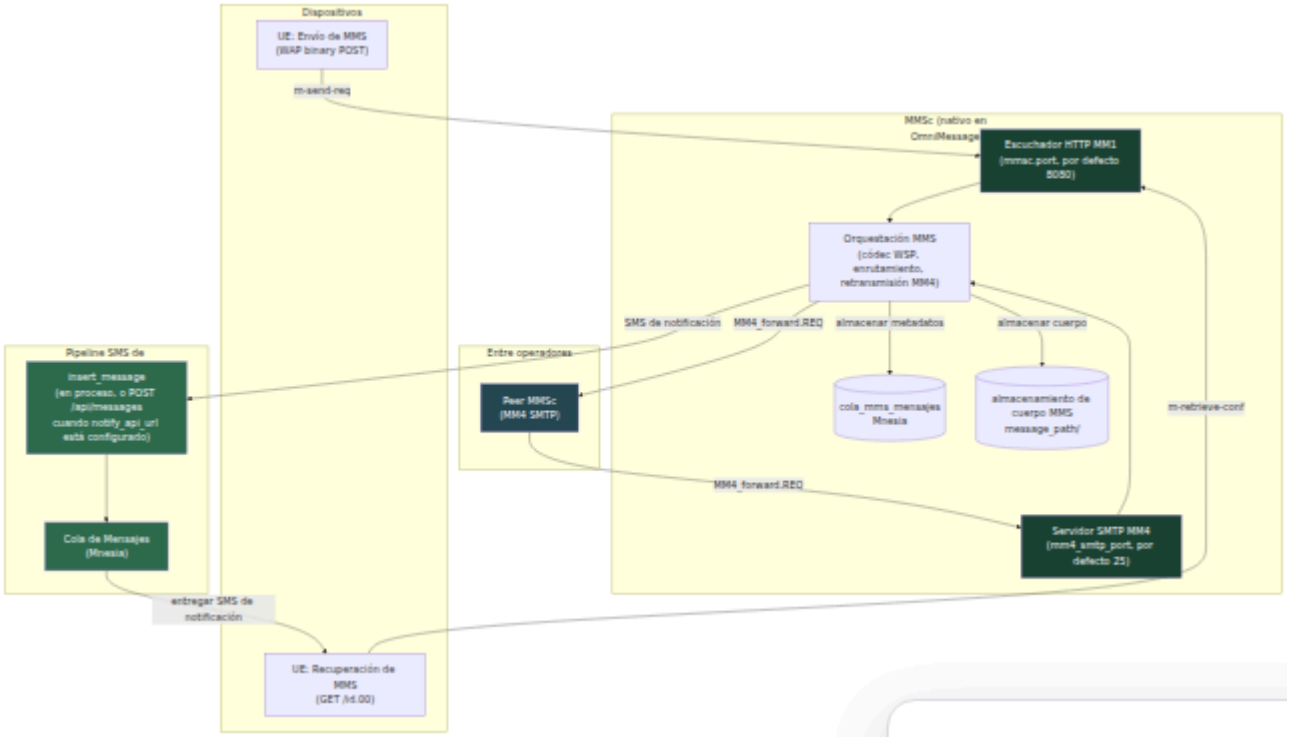
Debido a que la notificación se inyecta como un SMS normal, MMS reutiliza toda la maquinaria existente de enrutamiento, traducción y entrega de SMS sin llamadas HTTP autodirigidas.

## Especificaciones Relevantes

| Especificación | Título |
|---|---|
| 3GPP TS 23.140 | Servicio de Mensajería Multimedia (MMS): descripción funcional |
| 3GPP TS 26.140 | MMS: formatos de medios y códecs |
| Protocolo de Encapsulación OMA MMS | Codificación binaria WAP MM1 |

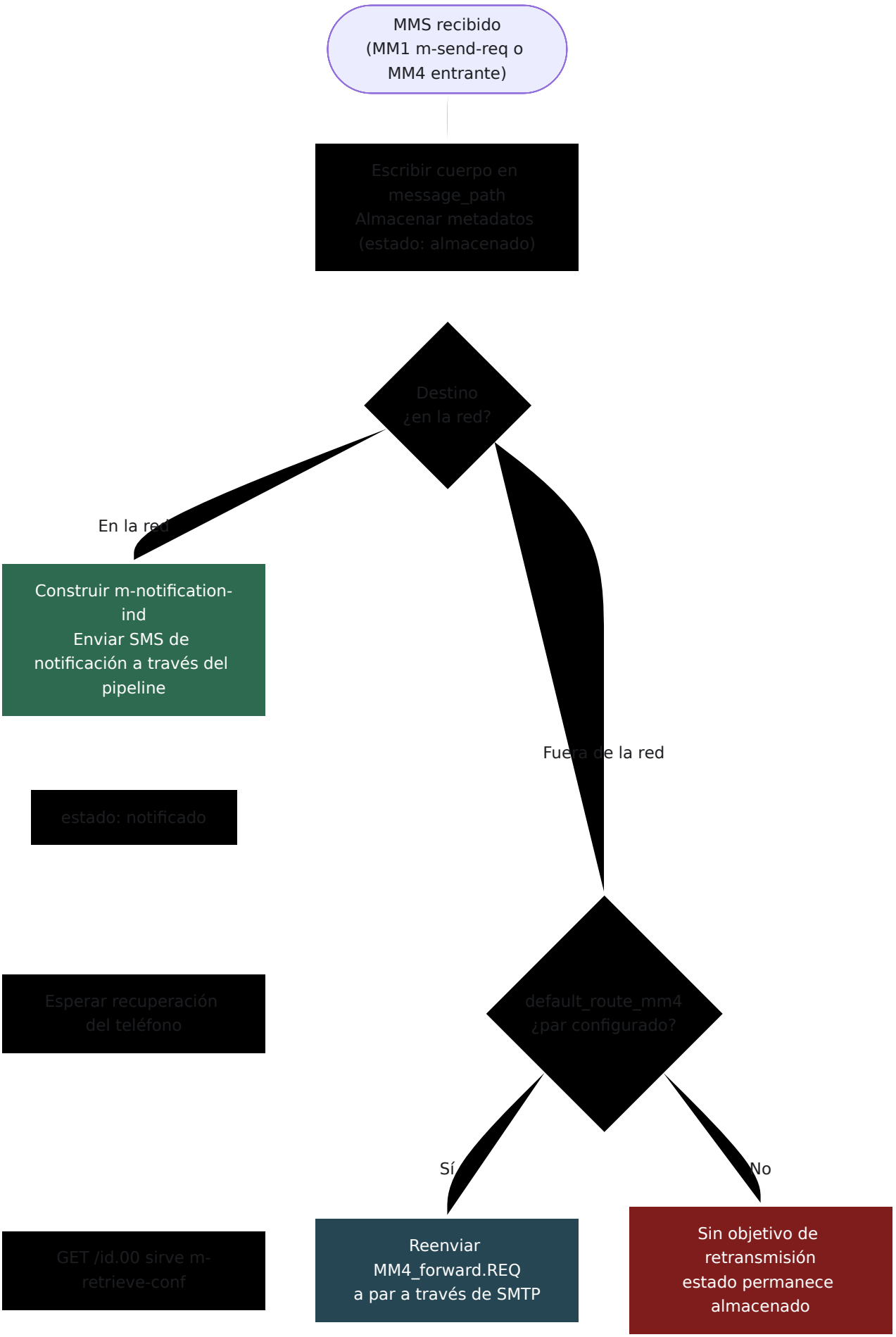
# Arquitectura

Las interfaces MM1 (HTTP del teléfono) y MM4 (SMTP entre operadores) escuchan en su propio puerto dedicado, separado de la API de OmniMessage. Los metadatos MMS residen en Mnesia junto con el almacenamiento de mensajes SMS. El MMSc mantiene los cuerpos MMS sin procesar como archivos en el sistema de archivos local. El registro de metadatos solo almacena la ruta del archivo.




# Procesamiento de Envío de Mensajes

Cada MMS enviado (desde un teléfono a través de MM1, o desde un par a través de MM4) sigue el mismo camino de procesamiento: almacenar el cuerpo, almacenar metadatos, clasificar el destino y luego notificar localmente o retransmitir fuera de la red.

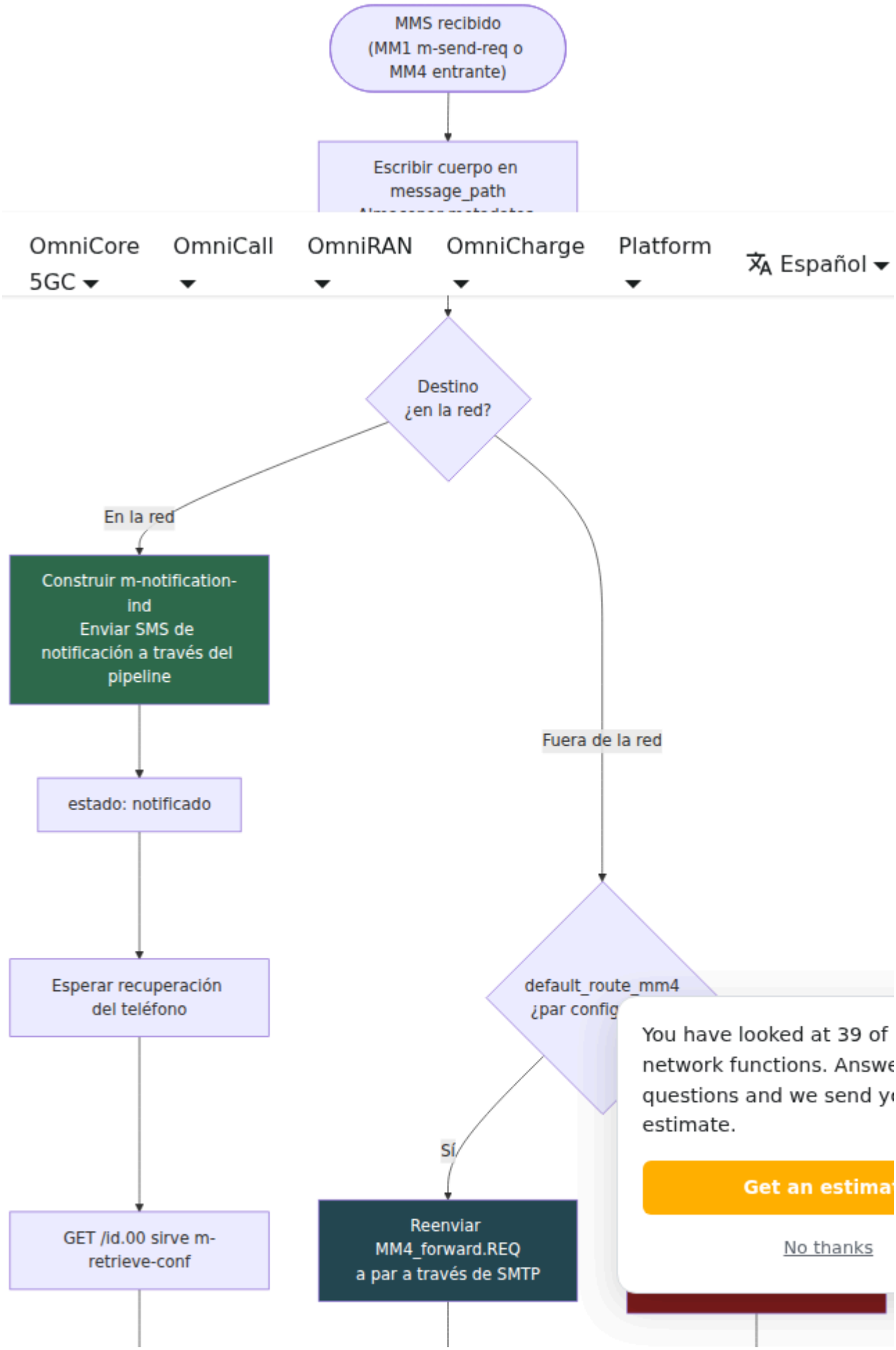

MMS recibido
(MM1 m-send-req o
MM4 entrante)
Escribir cuerpo en
message_path
Almacenar metadatos
(estado: almacenado)
Destino
¿en la red?
En la red
Construir m-notification-
ind
Enviar SMS de
notificación a través del
pipeline
estado: notificado
Fuera de la red
Esperar recuperación
del teléfono
default_route_mm4
¿par configurado?
Sí
No
GET /id.00 sirve m-
retrieve-conf
Reenviar
MM4_forward.REQ
a par a través de SMTP
Sin objetivo de
retransmisión
estado permanece
almacenado

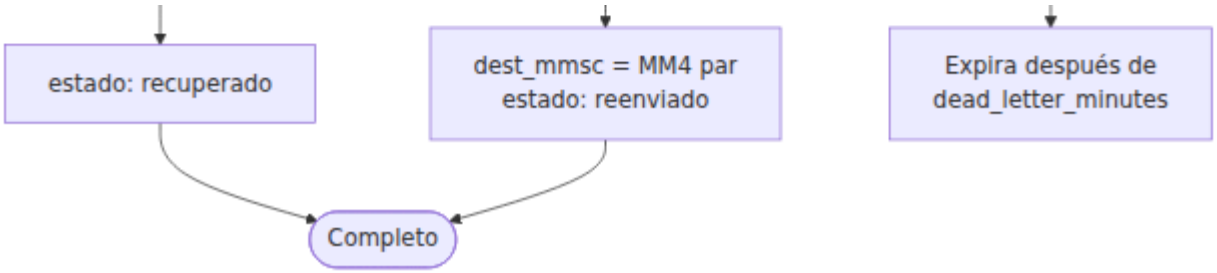



# Flujos de Entrega

## Entrega En la Red (MM1)

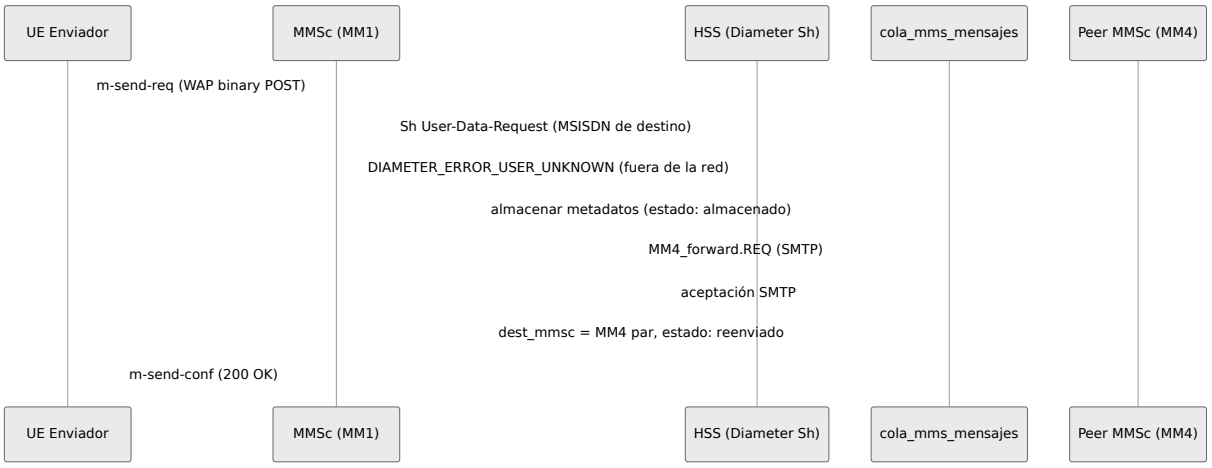
Un suscriptor envía un MMS a un destinatario atendido por la misma red.


MMS recibido
(MM1 m-send-req o
MM4 entrante)
Escribir cuerpo en
message_path
OmniCore
5GC
OmniCall
OmniRAN
OmniCharge
Platform
Español
Destino
¿en la red?
En la red
Construir m-notification-
ind
Enviar SMS de
notificación a través del
pipeline
estado: notificado
Esperar recuperación
del teléfono
GET /id.00 sirve m-
retrieve-conf
Fuera de la red
default_route_mm4
¿par config
Sí
Reenviar
MM4_forward.REQ
a par a través de SMTP
You have looked at 39 of
network functions. Answe
questions and we send y
estimate.
Get an estima
No thanks



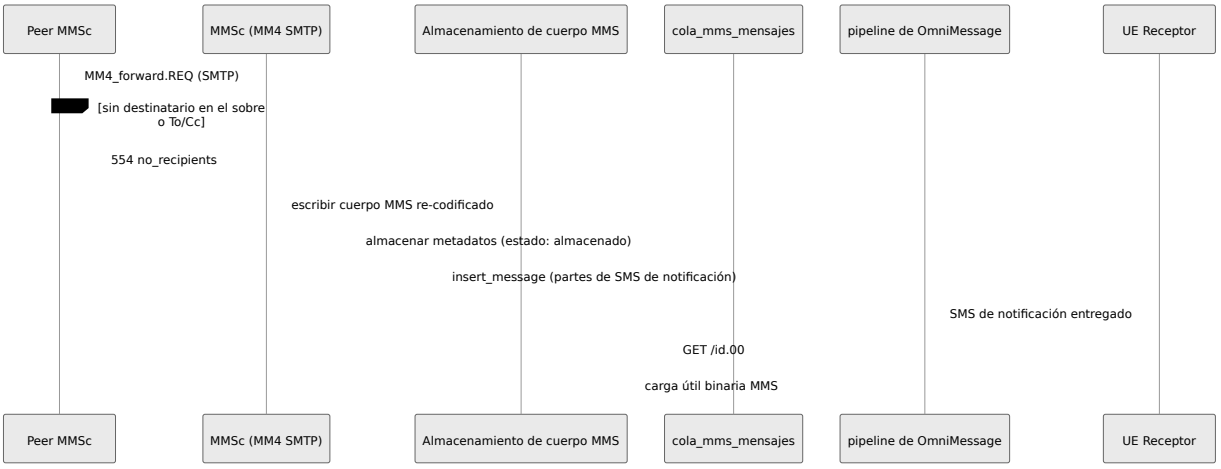

# Entrega Fuera de la Red (MM4)

Un suscriptor envía un MMS a un destinatario en otro operador.




# MM4 Entrante (de un Operador Par)

Un suscriptor en otra red envía un MMS a un suscriptor en la red.



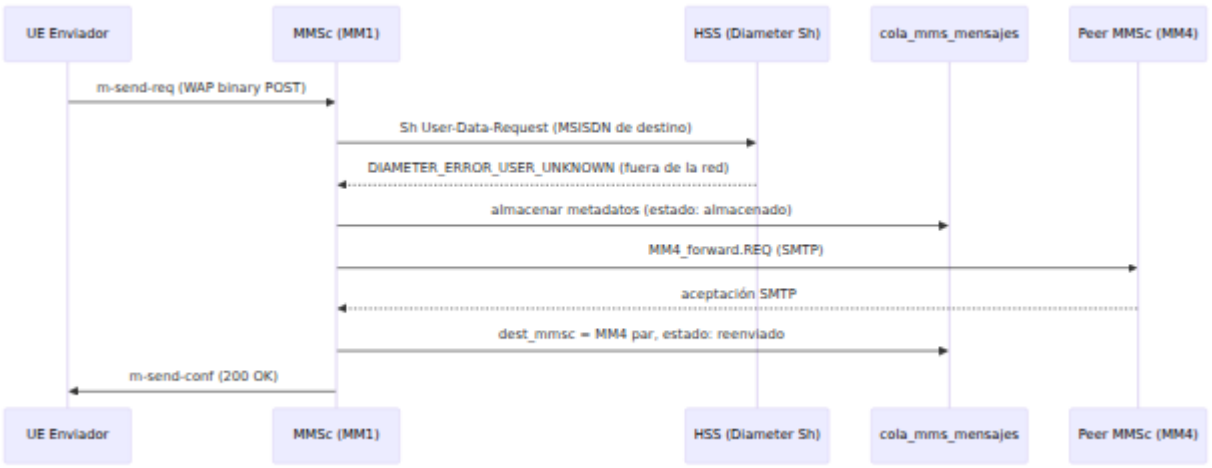

- **Entrega** va a los destinatarios del sobre SMTP (`RCPT TO`); los pares dirigen un sobre por cada suscriptor en la red. Solo cuando el sobre no lleva destinatario, los encabezados `To`/`Cc` deciden.

- **Miembros del grupo**: `To` + `Cc` juntos forman la lista de miembros almacenada en `recipient_list` y listada en el `m-retrieve-conf`, para que el teléfono mantenga un MMS grupal en su hilo grupal.
- **Medios**: cada parte no-SMIL del árbol MIME (incluyendo multiparts anidados) se re-codifica en el cuerpo WSP almacenado. Las partes de texto se convierten de su conjunto de caracteres declarado a UTF-8.
- Un mensaje sin destinatario en el sobre o en `To`/`Cc` es rechazado con `554` en lugar de ser aceptado y descartado.

# Notificación MMS

La notificación MMS es un SMS binario que lleva un **PDU `m-notification-ind` codificado en WSP**. Utiliza codificación de datos de 8 bits (DCS `0x44`), se divide en segmentos SMS concatenados y se envía a la cola de mensajes como partes ordinarias de SMS MT. Cada notificación utiliza su propia referencia de concatenación, por lo que un teléfono que recibe varias notificaciones en rápida sucesión vuelve a ensamblar cada una por separado. Por lo tanto, cada regla de traducción, enrutamiento y entrega de SMS se aplica a ella sin cambios.

- Por defecto, las partes se insertan en la cola de este nodo (en proceso).
- Cuando `notify_api_url` está configurado, cada parte se envía a ese controlador SMS-C con `POST /api/messages` y se coloca en la cola allí. Esto permite que el MMSc funcione en su propio nodo (por ejemplo, junto a una puerta de enlace SMPP) mientras las notificaciones son entregadas por el SMS-C principal. Los IDs de mensaje devueltos están vinculados al MMS como `notify_message_ids`. Una presentación rechazada o fallida se registra, y el MMS permanece almacenado sin esa parte de notificación.




La ubicación de recuperación codificada en la notificación es:

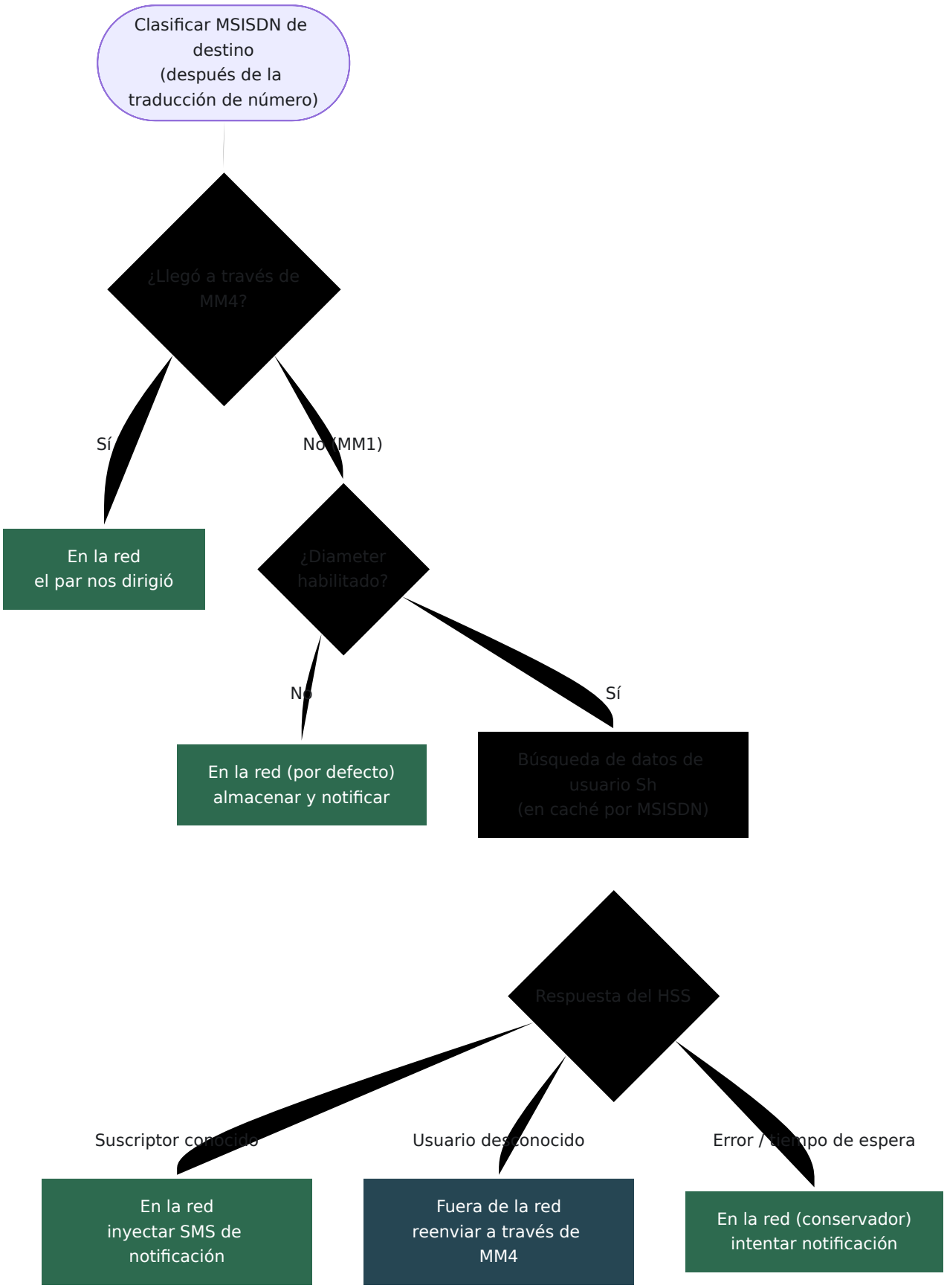
```
http://<mms_domain>/<unique_id>.00
```

- `mms_domain` es el `mmsc.mms_domain` configurado.
- `<unique_id>.00` es la clave de recuperación. El controlador MM1 elimina el sufijo `.00` para resolver el mensaje almacenado.

El teléfono receptor abre esta URL a través de MM1 para obtener el MMS como un `m-retrieve-conf`.

# Determinación On/Off-Net

Antes de que el MMSc decida si notificar localmente o retransmitir a través de MM4, pregunta al HSS a través de **Diameter Sh** si el destino es un suscriptor de esta red. Utiliza la misma búsqueda en caché de Sh que el SMS-C utiliza para la coincidencia de ruta de `terminating_network`, por lo que MMS y SMS están de acuerdo sobre quién está en la red.


Clasificar MSISDN de destino (después de la traducción de número)
¿Llegó a través de MM4?
Sí
No (MM1)
En la red el par nos dirigió
¿Diameter habilitado?
No
Sí
En la red (por defecto) almacenar y notificar
Búsqueda de datos de usuario Sh (en caché por MSISDN)
Respuesta del HSS
Suscriptor conocido
Usuario desconocido
Error / tiempo de espera
En la red inyectar SMS de notificación
Fuera de la red reenviar a través de MM4
En la red (conservador) intentar notificación

| Condición | Clasificación | Acción |
|---|---|---|
| MMS recibido a través de MM4 | En la red | Almacenar localmente e inyectar el SMS de notificación; nunca reenviado de vuelta a través de MM4 |
| HSS conoce al suscriptor | En la red | Almacenar localmente e inyectar el SMS de notificación |
| HSS responde `DIAMETER_ERROR_USER_UNKNOWN` | Fuera de la red | Almacenar localmente y reenviar al par a través de MM4 |
| Error de búsqueda de Sh o tiempo de espera | En la red (conservador) | Almacenar localmente e intentar la notificación; se registra una advertencia |
| Diameter no habilitado | En la red (por defecto) | Almacenar localmente e inyectar el SMS de notificación; se registra una advertencia |

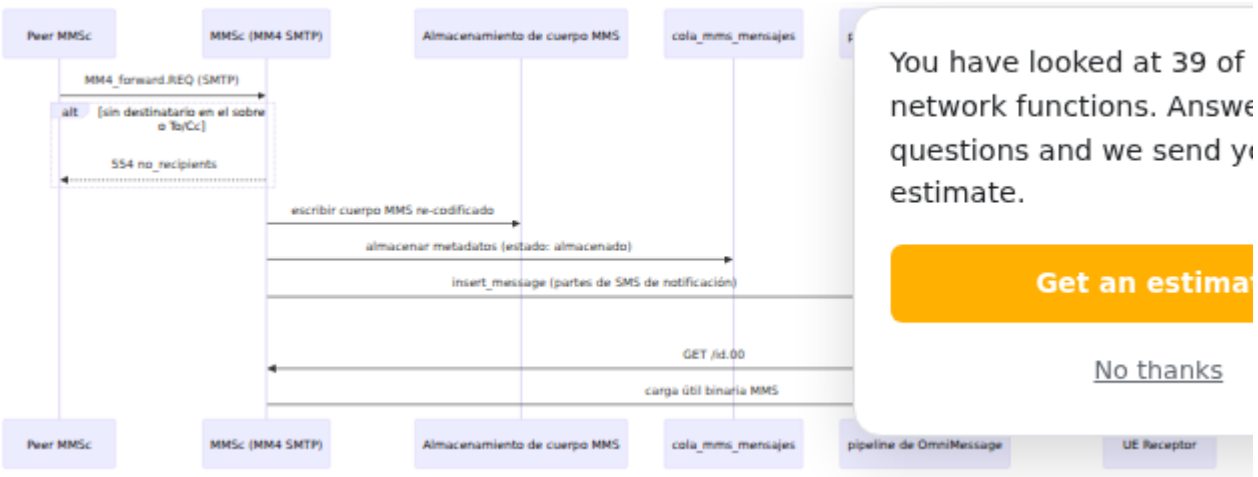
La retransmisión MM4 solo se intenta para destinos que el HSS informa positivamente como desconocidos. El emparejamiento Diameter Sh es el propio del SMS-C (`diameter_enabled` y los pares `:diameter_ex`); el MMSc no tiene configuraciones de clasificación separadas.

# Configuración

Se configura el MMSc en `runtime.exs` bajo la clave `:mmsc` de la aplicación `:sms_c`. Está **deshabilitado por defecto**. Si el bloque `:mmsc` está ausente, la función está completamente inactiva: el MMSc no vincula puertos ni inicia escuchadores. Por lo tanto, las implementaciones existentes no se ven afectadas hasta que lo habilite explícitamente.

## Estructura de Configuración

Cada clave de configuración impulsa una parte específica del MMSc. El diagrama muestra qué claves controlan qué proceso.



## Ejemplo de Configuración

```
# config/runtime.exs
config :sms_c, :mmsc,
  # Núcleo
  enabled: true,
  port: 8080,
  host: "0.0.0.0",
  mms_domain: "mms.example.com",
  message_path: "/var/lib/omnimessage/stored_mms",
  notify_source_msisdn: "6675",
  dead_letter_minutes: 1440,

  # MMSc independiente: enviar SMS de notificación a la API del
controlador SMS-C
  # (no configurado = insertar en la cola de mensajes de este
nodo)
  notify_api_url: nil,

  # Resolución de suscriptor (IP del cliente MM1 a MSISDN)
  subscriber_ip_api_urls: ["https://10.0.0.10:8443",
"https://10.0.0.11:8443"],

  # MM4 entrante/saliente (entre operadores)
  mm4_enabled: false,
  mm4_smtp_host: "0.0.0.0",
  mm4_smtp_port: 25,
  mm4_domain: "mms.example.com",
  default_route_mm4: nil
```

## Parámetros `:mmsc`

| Parámetro | Tipo | Requerido | Por Defect |
|---|---|---|---|
| `enabled` | Booleano | No | `false` |
| `port` | Entero | No | `8080` |
| `host` | Cadena | No | `0.0.0.0` |
| `mms_domain` | Cadena | Sí (si está habilitado) | - |
| `message_path` | Cadena | No | `/var/lib/omnimessage` |
| `notify_source_msisdn` | Cadena | No | `6675` |
| `dead_letter_minutes` | Entero | No | `1440` |
| `notify_api_url` | Cadena | No | `nil` |

| Parámetro | Tipo | Requerido | Por Defect |
| --- | --- | --- | --- |
| | | | |
| `subscriber_ip_api_urls` | Lista de cadenas | Sí (si está habilitado) | `[]` |
| `mm4_enabled` | Booleano | No | `false` |
| `mm4_smtp_host` | Cadena | No | `0.0.0.0` |
| `mm4_smtp_port` | Entero | No | `25` |
| `mm4_domain` | Cadena | No | se retrocede a `mms_dor` |

| Parámetro | Tipo | Requerido | Por Defect |
|---|---|---|---|
| | | | |
| `default_route_mm4` | Cadena | No | `nil` |

## Pares MM4

Los objetivos de retransmisión MM4 salientes se configuran por separado, porque son registros estructurados de host/puerto/dominio en lugar de configuraciones escalares:

```
# config/runtime.exs
config :sms_c, :mmsc_mm4_peers, %{
  "carrier-a" => [
    %{host: "mm4.carrier-a.example.net", port: 25, mms_domain:
"mms.carrier-a.example.net"}
  ]
}
```

`mmsc.default_route_mm4` selecciona qué grupo de pares se utiliza. Cada grupo es una lista; cuando contiene múltiples entradas, se elige una por intento de retransmisión para que la carga se distribuya entre los hosts de entrada de un par.

**Parámetros de pares MM4:**

| Parámetro | Tipo | Requerido | Por Defecto | Descripción |
|---|---|---|---|---|
| `host` | Cadena | Sí | - | Nombre de host o IP SMTP par. El MMSc se conecta directamente a él, sin bús MX. |
| `port` | Entero | No | `25` | Puerto SMTP MM4 del par. |
| `mms_domain` | Cadena | No | `""` | Dominio MMS del par, utili construir la dirección RCPT `<msisdn>/TYPE=PLMN@<mms` y registrado como `dest_m` |

## Nodo MMSc Independiente

El MMSc puede funcionar como su propio nodo OmniMessage, separado del SMS-C. Por ejemplo, puede ejecutarse en un host que ya termina las interfaces MM1 y MM4. El nodo maneja solo MMS, y el SMS-C entrega las notificaciones.

| Configuración | Valor en un MMSc independiente |
| --- | --- |
| `mmsc.notify_api_url` | La URL base de la API del SMS-C. Los SMS de notificación se envían allí y son entregados por el SMS-C. |
| `mmsc.host` / `mmsc.mm4_smtp_host` | Las direcciones de interfaz orientadas al teléfono y al par, para que MM1 y MM4 escuchen solo donde se necesita. |
| `cdr_enabled` | `false`. El nodo mantiene el estado MMS en Mnesia y los cuerpos en disco, y no necesita una base de datos SQL. |
| `enum_enabled` | `false`, sin rutas SMS. El nodo no enruta SMS. |
| Diameter (`diameter_enabled` + `:diameter_ex` peers) | Habilitado con un par Sh, directamente al HSS o a través del DRA, para la decisión on/off-net. El Origin-Host del nodo debe ser aceptado por ese par. |
| API / panel de control | En una dirección de gestión y un puerto no utilizados por otros servicios en el host. |

Un nodo independiente alcanza tres servicios a través de la red:

- la API del SMS-C, para notificaciones;
- la API del HSS (`subscriber_ip_api_urls`), para resolución de remitentes;
- el HSS o DRA a través de Diameter, para on/off-net.

Las métricas del nodo muestran fallos de cada uno: `notifications` con `result="error"`, solicitudes MM1 con estado `511`, y decisiones de enrutamiento `sh_error`.

# Almacenamiento

Los **metadatos** MMS se almacenan en la tabla Mnesia `:mms_message_queue`, creada automáticamente al inicio junto con el almacenamiento de mensajes SMS. No hay una base de datos separada que provisionar. El **cuerpo** MMS sin procesar se almacena como un archivo `.mms` bajo `message_path`; solo su ruta se mantiene en el registro de metadatos.

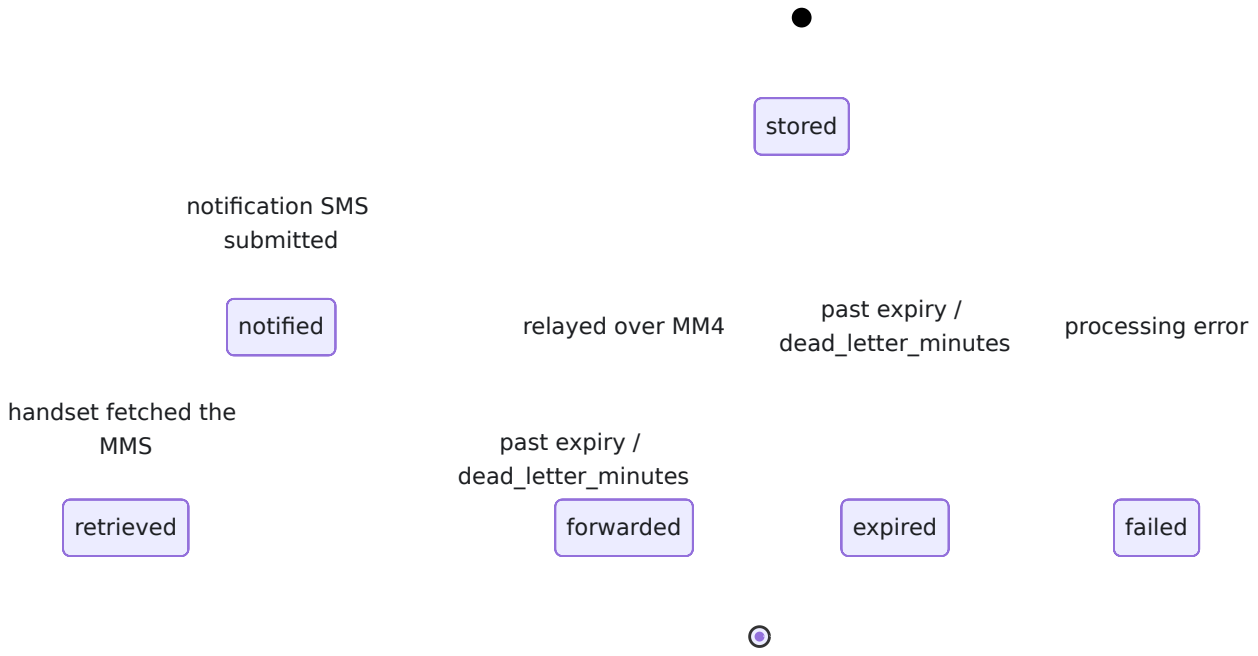
Hay un registro de metadatos por destinatario. Todos los destinatarios de un MMS comparten el cuerpo almacenado, y cada uno tiene su propio `unique_id`, por lo que la recuperación y el reconocimiento de un miembro del grupo actualizan el registro de ese miembro.

## Campos de `mms_message_queue`

| Campo | Descripción |
|---|---|
| `mms_id` | Clave primaria entera de un contador de Mnesia. |
| `unique_id` | Identificador de este destinatario, utilizado en su ruta de URL de recuperación (`<unique_id>.00`). |
| `source_msisdn` | MSISDN del suscriptor que envía. |
| `destination_msisdn` | MSISDN del destinatario principal. |
| `recipient_list` | Lista completa de To/Cc (CSV) para MMS grupales y la recuperación-conf. |
| `subject` | Línea de asunto del MMS. |
| `file_path` | Ruta al cuerpo `.mms` almacenado bajo `message_path`. |
| `source_mmsc` | Identidad del MMSc o par MM4 que originó el mensaje. |
| `dest_mmsc` | Establecido en `MM4_<peer>` después del reenvío saliente, o el destino de entrega después de la entrega local. |
| `on_net` | Si el MMSc clasificó el destino como en la red. |
| `notify_message_ids` | IDs de mensajes SMS creados para la notificación (enlaces al almacén de mensajes SMS). |
| `send_time` | Cuándo recibió este MMSc el MMS. |
| `notify_time` | Cuándo el sistema envió el SMS de notificación. |

| Campo | Descripción |
| --- | --- |
| `retrieve_time` | Cuándo el suscriptor recuperó el MMS. |
| `expires` | Tiempo de caducidad después del cual el MMS no se entrega. |
| `status` | Estado del ciclo de vida (ver más abajo). |
| `deadletter` | Establecido cuando el mensaje es marcado como carta muerta. |
| `inserted_at` / `updated_at` | Tiempos de marca de tiempo del ciclo de vida del registro. |

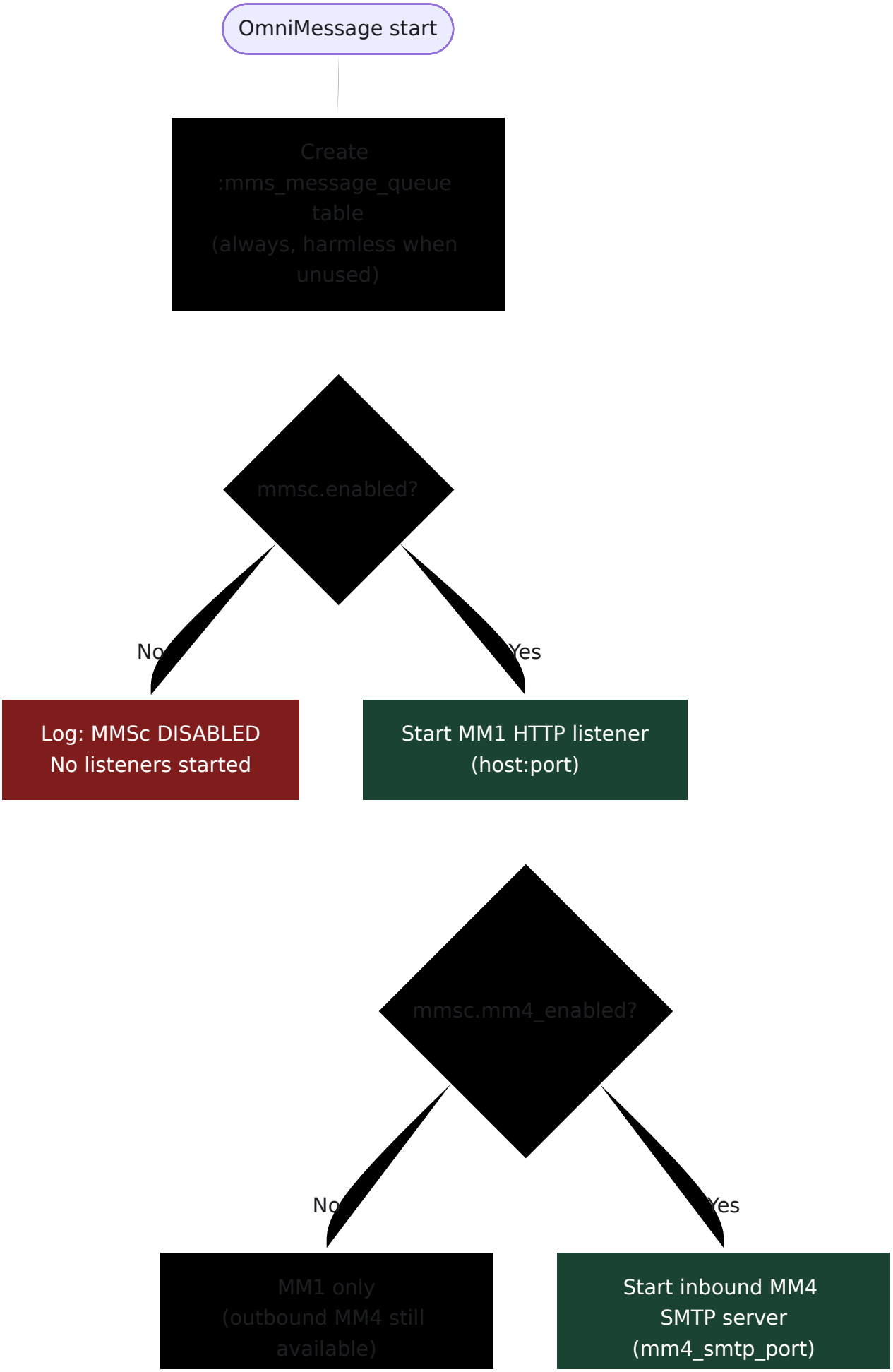
## Ciclo de Vida del Mensaje



| Estado | Significado |
| --- | --- |
| `stored` | Cuerpo escrito y metadatos registrados; aún no se ha tomado ninguna acción de entrega. |
| `notified` | SMS de notificación enviado a la cola para un destinatario en la red. |
| `retrieved` | El destinatario recuperó el MMS a través de MM1. |
| `forwarded` | Reenviado a un operador par a través de MM4. |
| `expired` | Pasó su caducidad o `dead_letter_minutes` sin entrega. |
| `failed` | Error de procesamiento o entrega. |

# Comportamiento en Tiempo de Ejecución

El MMSc se ejecuta como procesos supervisados dentro de la versión de OmniMessage. Nada se despliega o inicia por separado. El diagrama muestra qué puertas de configuración escucha cada oyente al inicio.


OmniMessage start
Create :mms_message_queue table (always, harmless when unused)
mmsc.enabled?
No
Yes
Log: MMSc DISABLED
No listeners started
Start MM1 HTTP listener (host:port)
mmsc.mm4_enabled?
No
Yes
MM1 only (outbound MM4 still available)
Start inbound MM4 SMTP server (mm4_smtp_port)

- **El oyente HTTP MM1** se inicia solo cuando `mmsc.enabled` es `true`. Se vincula a `mmsc.host:mmsc.port` y atiende las presentaciones de los dispositivos móviles (`m-send-req`), respuestas de notificación (`m-notifyresp-ind`), reconocimientos (`m-acknowledge-ind`) y recuperaciones (`GET /<unique_id>.00`).
- **El servidor SMTP MM4** se inicia solo cuando tanto `mmsc.enabled` como `mmsc.mm4_enabled` son `true`. Se vincula a `mmsc.mm4_smtp_port` y acepta mensajes `MM4_forward.REQ` entrantes de operadores pares.
- **El reenvío MM4 saliente** se realiza en el proceso cuando un destino se clasifica como fuera de la red; se conecta al par configurado a través de SMTP. No se requiere un agente de transferencia de correo externo.
- **La creación de la tabla** siempre se ejecuta al inicio. Cuando el MMSc está deshabilitado, esto simplemente crea una tabla vacía `:mms_message_queue`, lo cual es inofensivo.

**Vinculación de puertos:** el puerto MM4 predeterminado (25) es privilegiado. El servicio OmniMessage debe ejecutarse con la capacidad de vincular puertos bajos (`CAP_NET_BIND_SERVICE`) o usar un `mm4_smtp_port` más alto detrás de un reenvío de puerto. El oyente MM1 predeterminado utiliza el puerto no privilegiado 8080.

**TLS:** el oyente MM1 sirve HTTP sin cifrar. Termina TLS en un proxy inverso frente a `mmsc.port` si se requiere cifrado orientado al dispositivo móvil.

# Observabilidad

## API de Cola MMS

La API de OmniMessage expone el almacén de MMS, una entrada por destinatario. Se enumera bajo **MMS** en la documentación de la API (`/api/docs`).

| Solicitud | Propósito |
| --- | --- |
| `GET /api/mms_message_queues` | Listar MMS, el más nuevo primero. Filtros: `status`, `destination_msisdn`, `limit` (predeterminado 100). |
| `GET /api/mms_message_queues/<mms_id>` | Un MMS con sus partes decodificadas e historial de eventos (recibido, fuera de la red, notificación enviada, MM4 reenviado o fallido, recuperado, reconocido). |
| `PATCH /api/mms_message_queues/<mms_id>` | Actualizar `status`, `dest_mmsc` o `deadletter`. |
| `DELETE /api/mms_message_queues/<mms_id>` | Eliminar el registro de un destinatario. El cuerpo almacenado se elimina una vez que ningún otro registro de destinatario lo utiliza. |
| `GET /api/mms_message_queues/<mms_id>/parts` | Las partes multimedia del MMS, numeradas desde 1, con tipo MIME, nombre de archivo y tamaño. |
| `GET /api/mms_message_queues/<mms_id>/parts/<n>` | Descargar la parte `n` con su tipo MIME y nombre de archivo. Una descarga |

| Solicitud | Propósito |
| --- | --- |
| | de API no es una recuperación de dispositivo móvil: no cambia ningún estado y no agrega ningún evento. |

## Línea de Tiempo de Eventos

Cada registro de destinatario lleva una línea de tiempo de eventos, devuelta por `GET /api/mms_message_queues/<mms_id>`:

| Evento | Detalle |
| --- | --- |
| `sender_lookup` | Uno por nodo OmniHSS consultado para la IP del dispositivo móvil que envía: la URL de la solicitud, la respuesta y el MSISDN encontrado. |
| `received` | MM1: IP del cliente y encabezados HTTP (User-Agent, X-Forwarded-For, Host, tamaño), los encabezados decodificados de `m-send-req` y las partes. MM4: dirección del par SMTP, nombre HELO, `MAIL FROM` / `RCPT TO`, cada encabezado de correo electrónico y las partes. Ambos incluyen el destinatario y su clave de recuperación. |
| `routing` | En la red o fuera de la red, y por qué: el código de resultado Sh y HSS que responde, un fallo de Sh, Diameter deshabilitado, o llegada a través de MM4. |
| `notify_part_submitted` / `notify_part_failed` | Uno por parte de SMS de notificación: número de parte y UDH, el ID del mensaje que devolvió el SMS-C (o el error), y la presentación (estado HTTP y URL, o cola local). |
| `notify_submitted` | Todos los IDs de mensajes de notificación para el destinatario. |
| `mm4_forwarded` / `mm4_failed` | Dirección del par, nombre EHLO, `MAIL FROM` / `RCPT TO`, tamaño, duración, los encabezados enviados y la respuesta del par o el error. |
| `retrieved` / `retrieve_refused` | Cada intento de descarga del dispositivo móvil, servido o rechazado (por ejemplo, 410 cuando ya fue recuperado), con la IP del |

| Evento | Detalle |
| --- | --- |
| | cliente y los encabezados HTTP y si el MSISDN detrás de esa IP es el destinatario. Una discrepancia se registra solo; no cambia lo que se sirve. |
| `notifyresp` / `acknowledged` | El `m-notifyresp-ind` / `m-acknowledge-ind` del dispositivo móvil, con sus detalles y encabezados del cliente. |

Los eventos se mantienen en memoria durante el período de retención del mensaje y no se preservan a través de un reinicio.

## Métricas

El MMSc exporta estas métricas en el punto final de Prometheus del nodo (puerto 9568):

| Métrica | Tipo | Etiquetas | Des |
| --- | --- | --- | --- |
| `sms_c_mmsc_mm1_requests_count` | Contador | `type`, `status` | Solicitu tipo de ( `m_send` `m_noti` `m_ackn` `retrie` HTTP. U en el es significa búsque remiter fallando |
| `sms_c_mmsc_notifications_count` | Contador | `result`, `transport` | Partes d notifica enviada ( `transp` `local` `api`). |
| `sms_c_mmsc_mm4_forward_count` | Contador | `peer`, `result` | Reenvío pares M |
| `sms_c_mmsc_mm4_forward_duration_ms` | Histograma | `peer` | Tiempo entrega al par. |
| `sms_c_mmsc_mm4_inbound_count` | Contador | `result` | Mensaj entrant `accept` `no_rec` `mm4_pa` |
| `sms_c_mmsc_routing_decisions_count` | Contador | `decision` | Clasific destina |

| Métrica | Tipo | Etiquetas | Des |
|---|---|---|---|
| | | | `on_net` `sh_err` `diamet` Un aum `sh_err` que los destina de la re ser noti lugar d |
| `sms_c_mmsc_store_messages` | Medidor | `status` | MMS al por est |

Los SMS de notificación también aparecen en las métricas de mensajes, cola y CDR del SMS-C como cualquier otro SMS. Ver Referencia de Métricas.

# Solución de Problemas

## MMSc No Inicia

**Síntomas:** No hay oyente MM1 en el puerto configurado; los registros de inicio muestran `MMSc: DISABLED`.

**Causas posibles:**

- El bloque `:mmsc` está ausente de `runtime.exs`.
- `mmsc.enabled` es `false`.

**Resolución:**

1. Agregar el bloque de configuración `:mmsc` y establecer `enabled: true`.
2. Reiniciar el servicio y confirmar que el registro de inicio informa `MMSc: ENABLED` con el puerto y dominio esperados.

# El Dispositivo Móvil No Puede Recuperar el MMS

**Síntomas:** El SMS de notificación se entrega, pero el dispositivo móvil no puede recuperar el contenido.

**Causas posibles:**

- `mms_domain` no es resoluble o no es enrutable desde el APN MMS del dispositivo móvil.
- Un proxy inverso o firewall está bloqueando `mmsc.port`.
- El archivo `.mms` almacenado falta bajo `message_path`.

**Resolución:**

1. Confirmar que el host de la URL de notificación (`mms_domain`) se resuelve y es accesible desde el APN MMS.
2. Verificar el acceso entrante a `mmsc.host:mmsc.port` (y cualquier proxy que termine TLS).
3. Verificar el `status` del mensaje y que el archivo `.mms` exista en `file_path`.

# MMS Fuera de la Red No Se Reenvía

**Síntomas:** MMS a otro operador está almacenado pero nunca se reenvía; `status` permanece `stored`.

**Causas posibles:**

- `default_route_mm4` no está configurado, por lo que no se selecciona ningún grupo par.
- La entrada par en `:mmsc_mm4_peers` tiene el host/puerto incorrecto.
- El destino fue clasificado como en la red: el HSS conoce el número, la búsqueda Sh falló, o Diameter no está habilitado (cada uno de estos predetermina a en la red; los últimos dos registran una advertencia).

**Resolución:**

1. Establecer `default_route_mm4` a una clave presente en `:mmsc_mm4_peers`.

2. Verificar el `host`/`port` del par y que el SMTP saliente hacia él esté permitido.
3. Verificar `diameter_enabled` y que el par Sh (HSS o DRA) esté conectado, para que los números fuera de la red regresen como usuario desconocido.

## MM4 Entrante Rechazado o No Recibido

**Síntomas:** Un operador par no puede entregar a través de MM4.

**Causas posibles:**

- `mm4_enabled` es `false`, por lo que el servidor SMTP no está en funcionamiento.
- El mensaje no nombra a ningún destinatario en el sobre SMTP o en los encabezados `To`/`Cc`, por lo que es rechazado con `554 no_recipients`.
- El servicio no tiene permiso para vincular el puerto privilegiado 25.
- Un firewall está bloqueando el SMTP entrante.

**Resolución:**

1. Establecer `mm4_enabled: true` y reiniciar.
2. Conceder `CAP_NET_BIND_SERVICE` o establecer un `mm4_smtp_port` no privilegiado detrás de un reenvío de puerto.
3. Permitir tráfico entrante a `mm4_smtp_port` desde las direcciones de salida del par.

## Falla en la Resolución de la Dirección del Remitente

**Síntomas:** Las presentaciones desde dispositivos móviles que no son localhost son rechazadas con un error de búsqueda de suscriptor.

**Causas posibles:**

- `subscriber_ip_api_urls` no está configurado, por lo que solo localhost se resuelve a un MSISDN de prueba.

- Ningún nodo HSS configurado tiene un enlace IP para el dispositivo móvil (la sesión de datos del UE aún no es conocida por el HSS).
- Los puntos finales de la API de OmniHSS no son accesibles o están devolviendo respuestas no-200.

**Resolución:**

1. Configurar `subscriber_ip_api_urls` con cada nodo OmniHSS para la resolución de remitentes en producción.
2. Verificar que los puntos finales sean accesibles y devuelvan el mapeo esperado de suscriptor/MSISDN.

# Guía de Traducción de Números SMS-C

← Volver al Índice de Documentación

## Descripción General

El sistema de Traducción de Números SMS-C proporciona una transformación flexible basada en regex de números de teléfono antes de la ruta. Las reglas de traducción pueden normalizar números, agregar prefijos internacionales, formatear números para puertas de enlace específicas y encadenar múltiples transformaciones. Las reglas se almacenan en Mnesia para persistencia y se pueden modificar en tiempo de ejecución sin interrupción del servicio.

## Características Clave

- **Coincidencia basada en prefijos**: Coincidir números por prefijo antes de aplicar transformaciones
- **Transformación basada en regex**: Potente coincidencia y reemplazo de patrones con grupos de captura
- **Filtrado de SMSC de origen**: Aplicar diferentes traducciones según el origen del mensaje
- **Evaluación basada en 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 completa CRUD para la gestión de reglas
- **Herramienta de simulación**: Probar la lógica de traducción con evaluación paso a paso
- **Copia de seguridad/Restauración**: Exportar e importar configuraciones de traducción
- **Integración previa a la ruta**: Traducciones aplicadas antes de la ruta para formatos de número consistentes

# Arquitectura

## Modelo de Datos

Cada regla de traducción contiene los siguientes campos:

| Campo | Tipo | Descripción | Requerido |
|---|---|---|---|
| `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. También actúa como el **gate** para la anulación de TON/NPI de origen | 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 |
| `source_ton` | cadena/nil | Forzar el **tipo** de número de la dirección de origen. Vacío/ausente = dejar inferido | No |

| Campo | Tipo | Descripción | Requerido |
|---|---|---|---|
| `source_npi` | cadena/nil | Forzar el **indicador** de plan de numeración de la dirección de origen. Vacío/ausente = dejar inferido | 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 (el número más bajo primero). Solo se evalúan las reglas habilitadas.

Vea Forzar Tipo de Número (TON) y Indicador de Plan de Numeración (NPI) para la documentación completa de los campos `source_ton` y `source_npi`.

## Algoritmo de Traducción

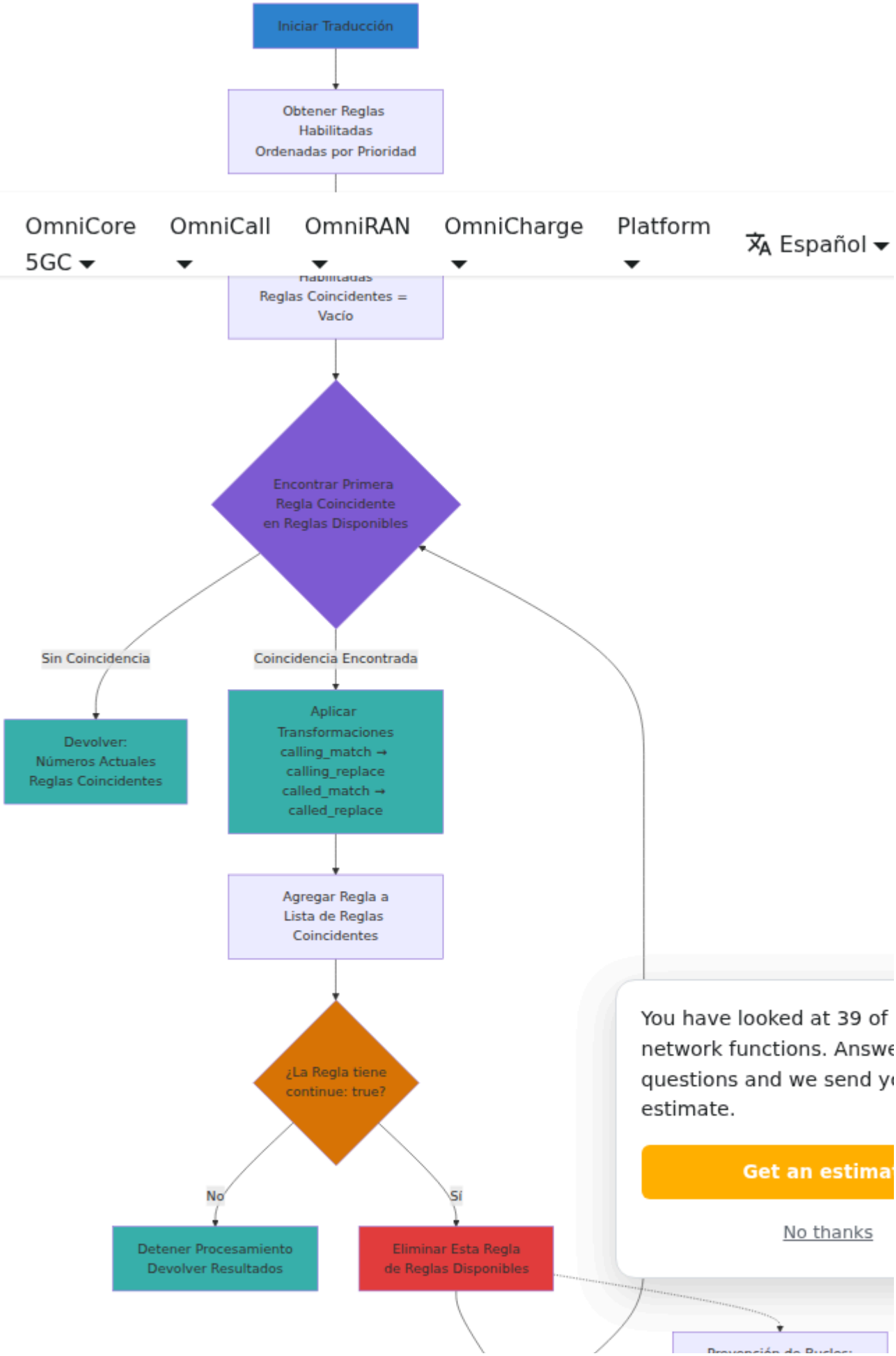
Al traducir números, el sistema:

1. **Recupera reglas habilitadas** ordenadas por prioridad (la más baja 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)

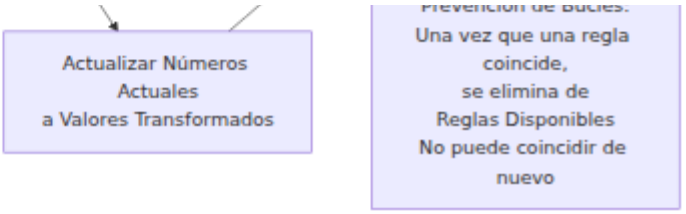
3. **Aplica la primera regla coincidente**:
    - Transformar el número de llamada usando `calling_match` y `calling_replace`
    - Transformar el número llamado usando `called_match` y `called_replace`
4. **Verifica la bandera de continuar**:
    - Si `continue: false` → Detener procesamiento, devolver 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:



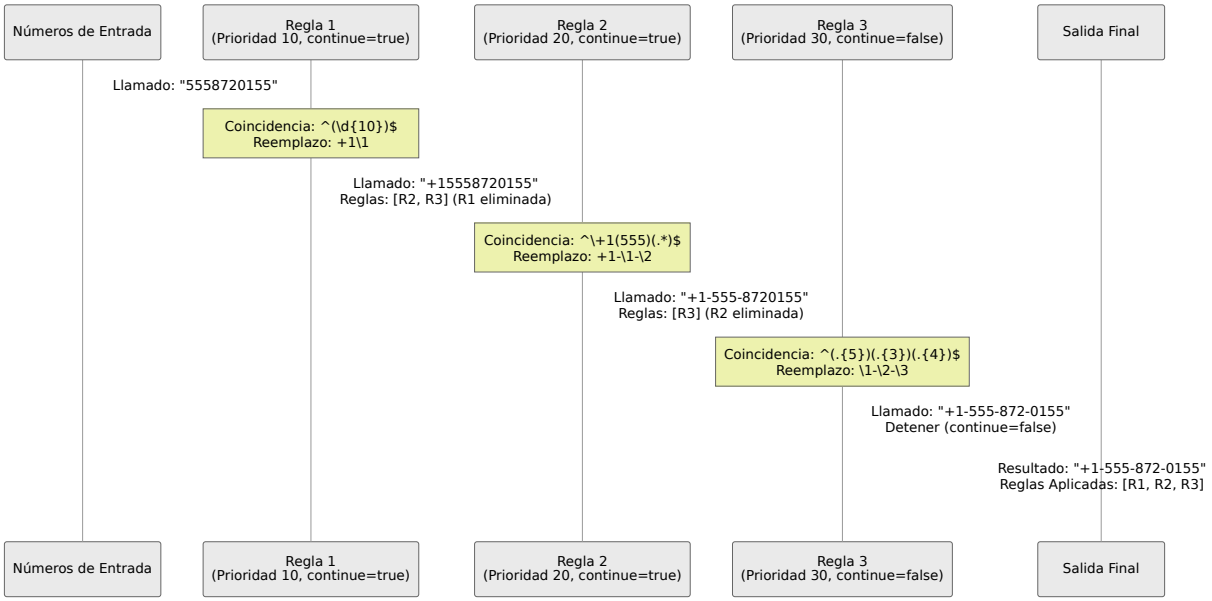

You have looked at 39 of
network functions. Answe
questions and we send y
estimate.
Get an estima
No thanks




## 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




# Forzar Tipo de Número (TON) y Indicador de Plan de Numeración (NPI)

Además de la reescritura de números basada en regex, una regla de traducción puede **forzar** el Tipo de Número (TON) y el Indicador de Plan de Numeración

(NPI) de la dirección de origen (de llamada). Esto es útil cuando la cadena de número del remitente por sí sola no transmite la semántica correcta de la dirección. Por ejemplo, un código corto debe ser señalado como `unknown`/`unknown` en lugar de ser tratado como un número nacional o internacional.

Estas anulaciones son independientes de la reescritura de números: una regla puede forzar TON/NPI **sin** cambiar el número en absoluto.

## Los Campos de Anulación

| Campo | Aplica a | Efecto |
|---|---|---|
| `source_ton` | Dirección de origen (de llamada) | Fuerza el Tipo de Número en la dirección de origen SMS-DELIVER construida (el TPDU) |
| `source_npi` | Dirección de origen (de llamada) | Fuerza el Indicador de Plan de Numeración en la dirección de origen SMS-DELIVER construida (el TPDU) |

**Vacío o ausente = sin anulación.** Cuando un campo se deja vacío (o la regla está ausente por completo), el valor de la dirección se **infiere** en su lugar - vea Inferencia de Recaída a continuación.

> **Nota:** `source_ton`/`source_npi` se integran directamente en la dirección de origen codificada en el **TPDU** (el SMS-DELIVER que el frontend IMS/SS7 transmite). El valor vive en el TPDU codificado - no se almacena nada extra en el registro del mensaje. No hay anulación de TON/NPI de destino: el TPDU SMS-DELIVER no lleva dirección de destino, y los frontends derivan la dirección de destino por sí mismos.

## Valores Permitidos

**Tipo de Número (TON):**

| Valor | Significado |
| --- | --- |
| `unknown` | Tipo de número desconocido |
| `international` | Número internacional (E.164 completo con código de país) |
| `national` | Número nacional (significativo) |
| `network_specific` | Número específico de la red |
| `subscriber` | Número de suscriptor |
| `alphanumeric` | Dirección alfanumérica (por ejemplo, un ID de remitente de nombre de marca) |
| `abbreviated` | Número abreviado |

**Indicador de Plan de Numeración (NPI):**

| Valor | Significado |
|---|---|
| `unknown` | Plan de numeración desconocido |
| `isdn` | Plan de numeración ISDN/telefónico (E.164) |
| `data` | Plan de numeración de datos (X.121) |
| `telex` | Plan de numeración telex |
| `land_mobile` | Plan de numeración de móviles terrestres (E.212) |
| `national` | Plan de numeración nacional |
| `private` | Plan de numeración privado |
| `ermes` | Plan de numeración ERMES |

Cualquier valor fuera de estos conjuntos es rechazado cuando se crea o actualiza la regla.

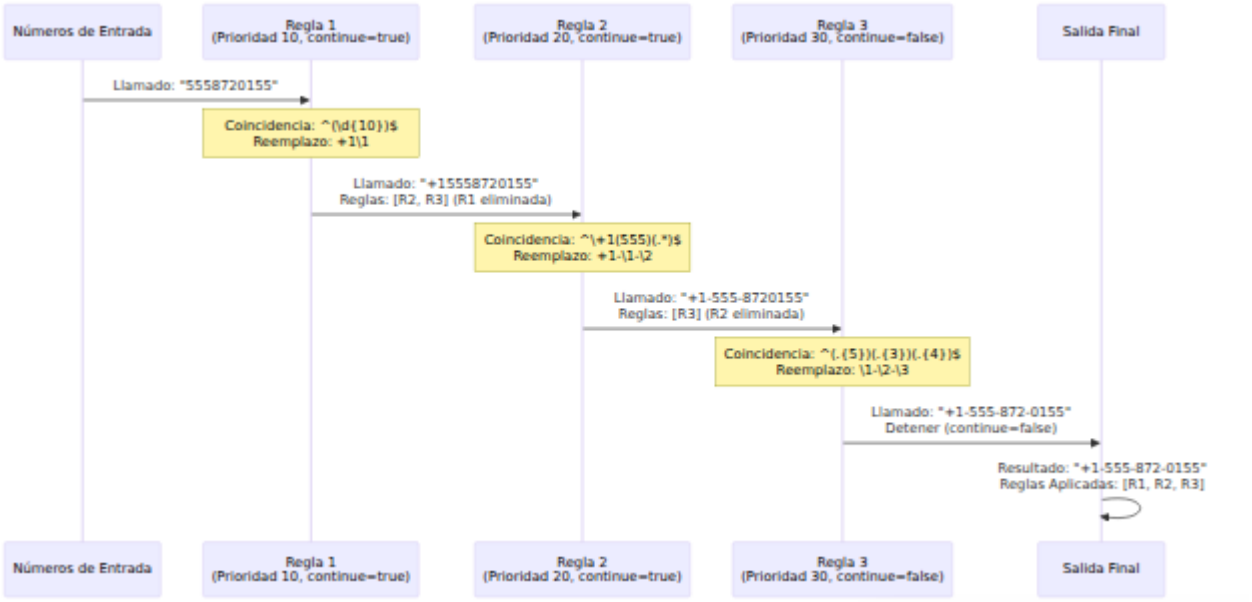
## Gating de Coincidencia: Alcance de una Anulación a un Patrón de Remitente

La anulación está **gated** por el regex de **Coincidencia de Número de Llamada (`calling_match`)** de la regla, que permite que una sola regla apunte a un subconjunto de remitentes por patrón:

- La anulación de origen (`source_ton`/`source_npi`) se aplica solo cuando el regex `calling_match` de la regla coincide con el número de llamada.
- Una **coincidencia en blanco es un comodín** - la anulación se aplica siempre que la regla sea seleccionada de otra manera (por prefijo y/o SMSC de origen).

Una regla puede llevar un `calling_match` con **sin** `calling_replace` correspondiente. En esa configuración, la coincidencia actúa puramente como un **gate** para la anulación y no **reescribe** el número. Esta es la forma

recomendada de aplicar una anulación TON/NPI solo a aquellos números que coinciden con un patrón.




Cuando `continue: true` encadena múltiples reglas, las anulaciones TON/NPI se acumulan con una semántica de **última escritura gana**: una regla posterior que establece un valor no vacío sobrescribe una anterior; un valor vacío nunca borra una anulación previamente establecida.

## Inferencia de Recaída

Cuando **ninguna** regla fuerza una anulación, los valores de la dirección de origen se **infiere** de la cadena de número en el momento de construcción del TPDU:

| Entrada | TON Inferido | NPI Inferido |
|---|---|---|
| Numérico, ≥ 5 dígitos | `international` | `isdn` |
| Numérico, < 5 dígitos | `national` | `isdn` |
| No numérico (remitente alfanumérico, por ejemplo, `Acme`) | `alphanumeric` | `unknown` |

Una anulación forzada siempre tiene prioridad sobre el valor inferido. Forzar solo el TON (dejar NPI en blanco) anula el TON mientras deja el NPI inferido, y viceversa.

## Establecimiento de Anulaciones TON/NPI

Estos campos se pueden establecer a través de tres interfaces:

- **Interfaz Web** - el editor de reglas de Traducción de Números expone menús desplegables para TON de Origen y NPI de Origen. Dejar un menú desplegable sin establecer significa "sin anulación".
- **API REST** - los endpoints `/api/translations` aceptan `source_ton` y `source_npi` como valores de cadena (vea Ejemplos a continuación).
- **Configuración** - la lista `config :sms_c, :translation_rules` acepta los mismos campos como valores de cadena, cargados en el primer inicio.

## Ejemplo Trabajado: Remitentes de Código Corto

Un remitente que es un código corto de 3 a 6 dígitos debe ser señalado como TON/NPI `unknown`/`unknown` **sin** reescribir el número. La regla usa `calling_match` puramente como un gate (sin `calling_replace`):

**Configuración (`config/runtime.exs`):**

```
config :sms_c, :translation_rules, [
  %{
    calling_match: "^\d{3,6}$",
    calling_replace: nil,
    source_ton: "unknown",
    source_npi: "unknown",
    priority: 5,
    continue: true,
    description: "Códigos cortos (3-6 dígitos) => unknown/unknown
TON/NPI",
    enabled: true
  }
]
```

**Cómo funciona:** La regla se evalúa para cada mensaje (sin `calling_prefix`, `called_prefix`, o `source_smsc` que la restrinja). El regex `calling_match` `^\d{3,6}$` gatea la anulación de origen, por lo que `source_ton`/`source_npi` se fuerzan a `unknown`/`unknown` **solo** cuando el número de llamada es una cadena de 3 a 6 dígitos. Debido a que `calling_replace` es `nil`, el número en sí se deja sin cambios. `continue: true` permite que se ejecuten reglas posteriores también.

**Caso de uso:** Tráfico originado por aplicaciones desde códigos cortos que de otro modo se inferirían incorrectamente como números `national` (4 dígitos) o `international` (5-6 dígitos).

**Solicitud equivalente de API REST** - `POST /api/translations`:

```
{
  "calling_match": "^\d{3,6}$",
  "calling_replace": null,
  "source_ton": "unknown",
  "source_npi": "unknown",
  "priority": 5,
  "continue": true,
  "description": "Códigos cortos de operador (3-6 dígitos) =>
unknown/unknown TON/NPI",
  "enabled": true
}
```

La API devuelve la regla creada (incluido su `rule_id` asignado) con los valores TON/NPI serializados como cadenas.

## Ejemplo Trabajado: Forzar Nacional para un Rango Conocido

Forzar la dirección de origen de un rango de números locales conocidos a `national`/`isdn`, sin reescribir el número:

**Configuración (`config/runtime.exs`):**

```
%{
  calling_prefix: "07",
  calling_match: "^07\d{8}$",
  calling_replace: nil,
  source_ton: "national",
  source_npi: "isdn",
  priority: 20,
  continue: false,
  description: "Números locales 07xxxxxxxx => national/isdn",
  enabled: true
}
```

**Cómo funciona:** `calling_prefix: "07"` reduce la regla a números que comienzan con `07`, y el regex `calling_match` gatea la anulación a exactamente un `07` seguido de 8 dígitos. Cuando coincide, la dirección de origen se señala como `national`/`isdn` en lugar de la inferida `international`/`isdn` (que se aplicaría porque el número tiene ≥ 5 dígitos). El número no se reescribe.

**Caso de uso:** Una puerta de enlace o par que requiere que los números locales formateados nacionalmente se marquen como TON `national` incluso si se almacenan en su forma de marcado en bruto.

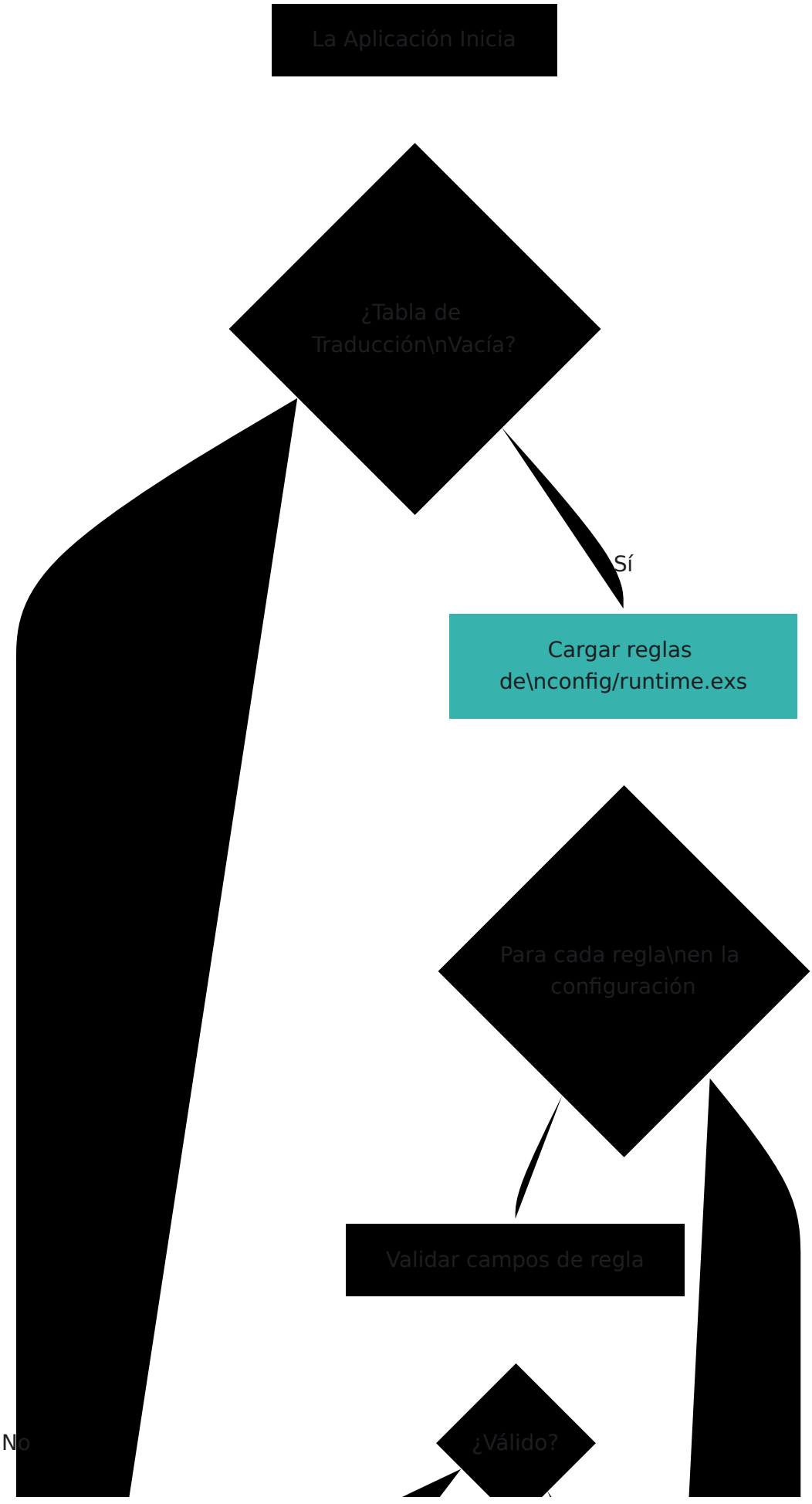
# Configuración

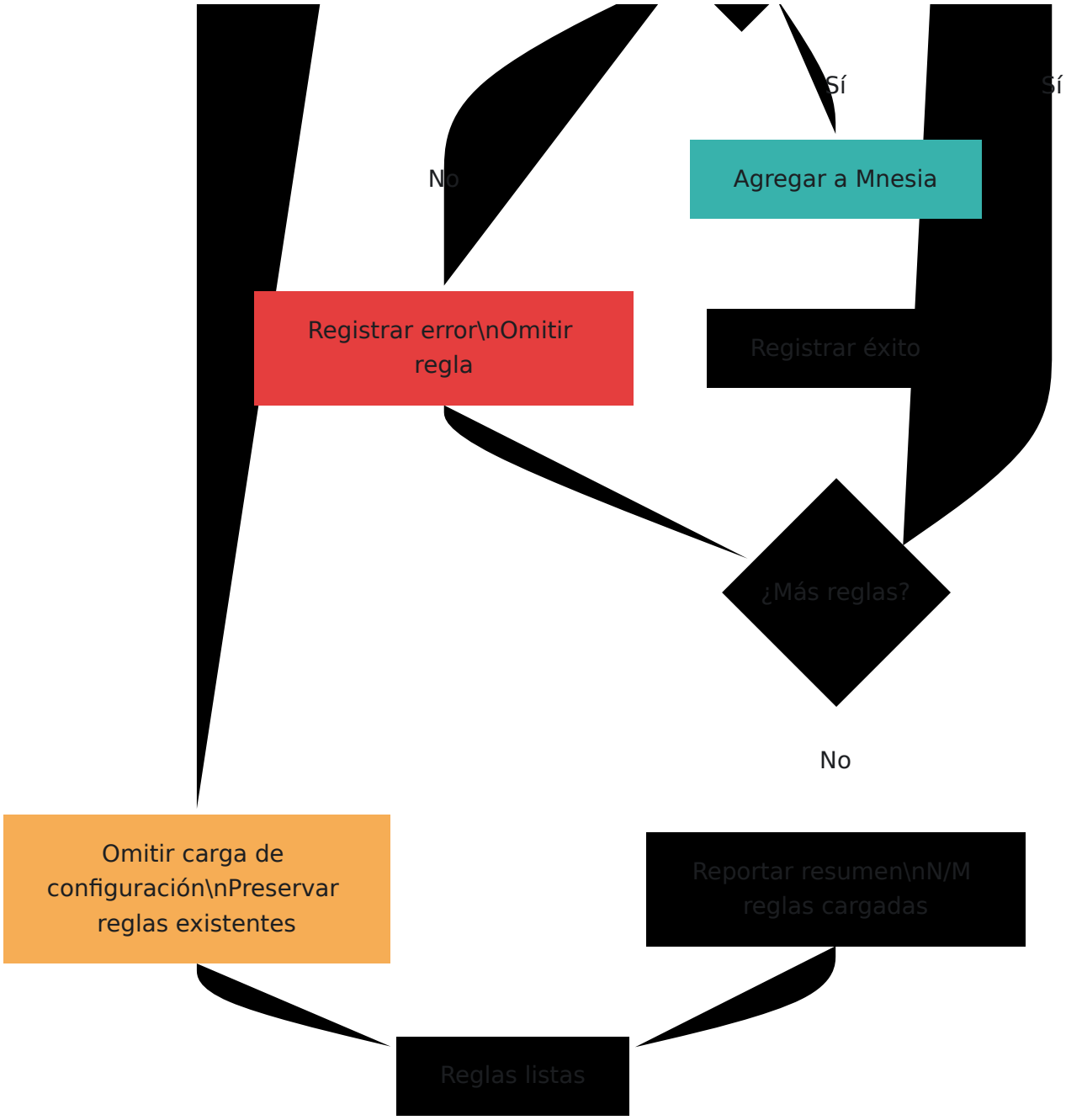
## Carga de Reglas desde el Archivo de Configuración

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 añadidas a través de la interfaz web durante el tiempo de ejecución y evita duplicados en reinicios.

### Flujo de Carga de Configuración


La Aplicación Inicia
¿Tabla de Traducción\nVacía?
Sí
Cargar reglas de\nconfig/runtime.exs
Para cada regla\nen la configuración
Validar campos de regla
¿Válido?
No



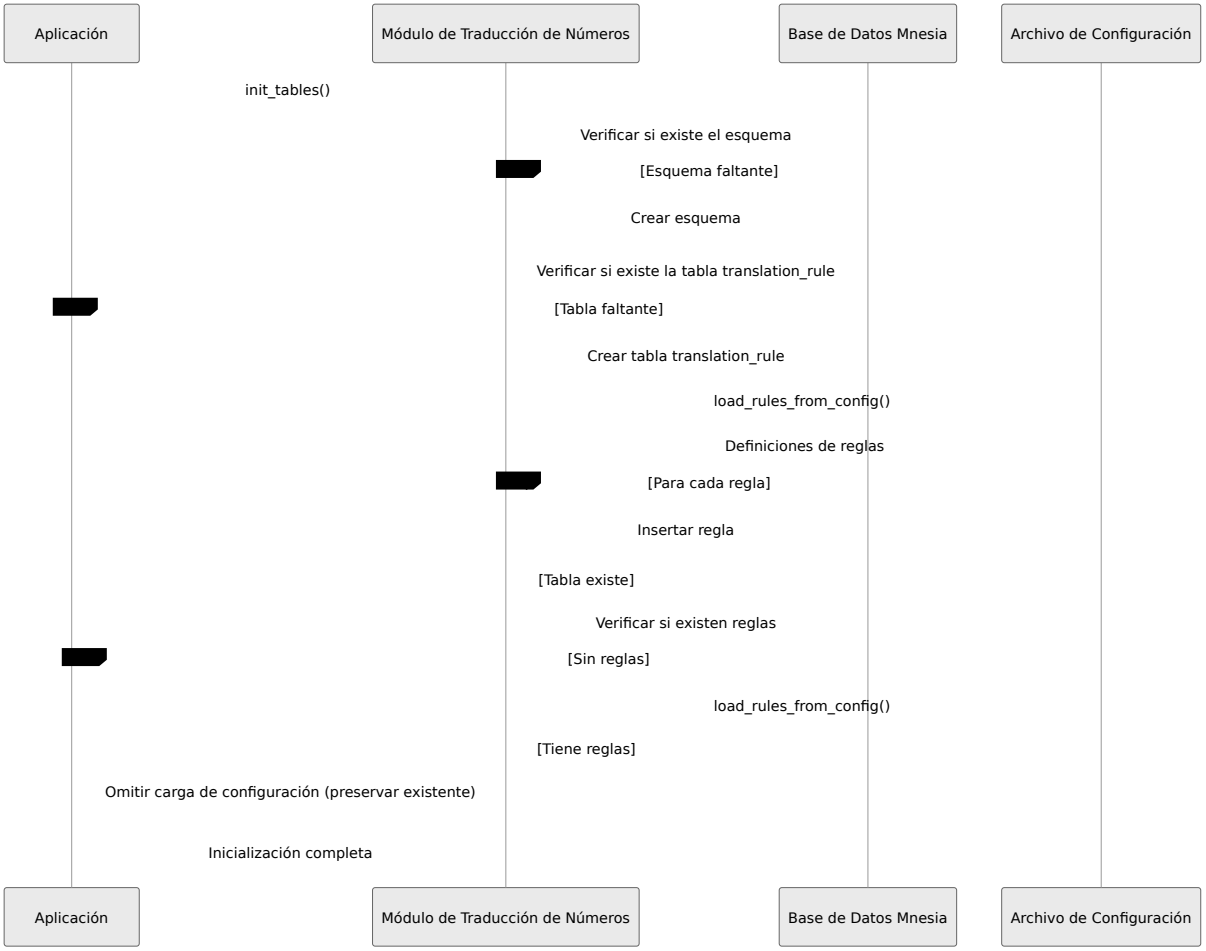

**Ejemplo de Configuración**

```
# config/runtime.exs
config :sms_c, :translation_rules, [
  # Agregar +1 a números de 10 dígitos de EE. UU.
  %{
    calling_prefix: nil,
    called_prefix: nil,
    source_smsc: "us_domestic_smsc",
    calling_match: "^(\d{10})$",
    calling_replace: "+1\1",
    called_match: "^(\d{10})$",
    called_replace: "+1\1",
    priority: 10,
    description: "Agregar +1 a números de 10 dígitos de SMSC
nacional",
    enabled: true,
    continue: false
  },

  # Eliminar ceros iniciales del formato internacional
  %{
    calling_prefix: "00",
    called_prefix: nil,
    source_smsc: nil,
    calling_match: "^00(.+)$",
    calling_replace: "+\1",
    called_match: nil,
    called_replace: nil,
    priority: 5,
    description: "Convertir prefijo internacional 00 a +",
    enabled: true,
    continue: true  # Continuar aplicando más formateo
  },

  # Forzar remitentes de códigos cortos a unknown/unknown TON/NPI
(solo gate, sin reescritura)
  %{
    calling_match: "^\d{3,6}$",
    calling_replace: nil,
    source_ton: "unknown",
    source_npi: "unknown",
    priority: 6,
    description: "Códigos cortos (3-6 dígitos) => unknown/unknown
TON/NPI",
```

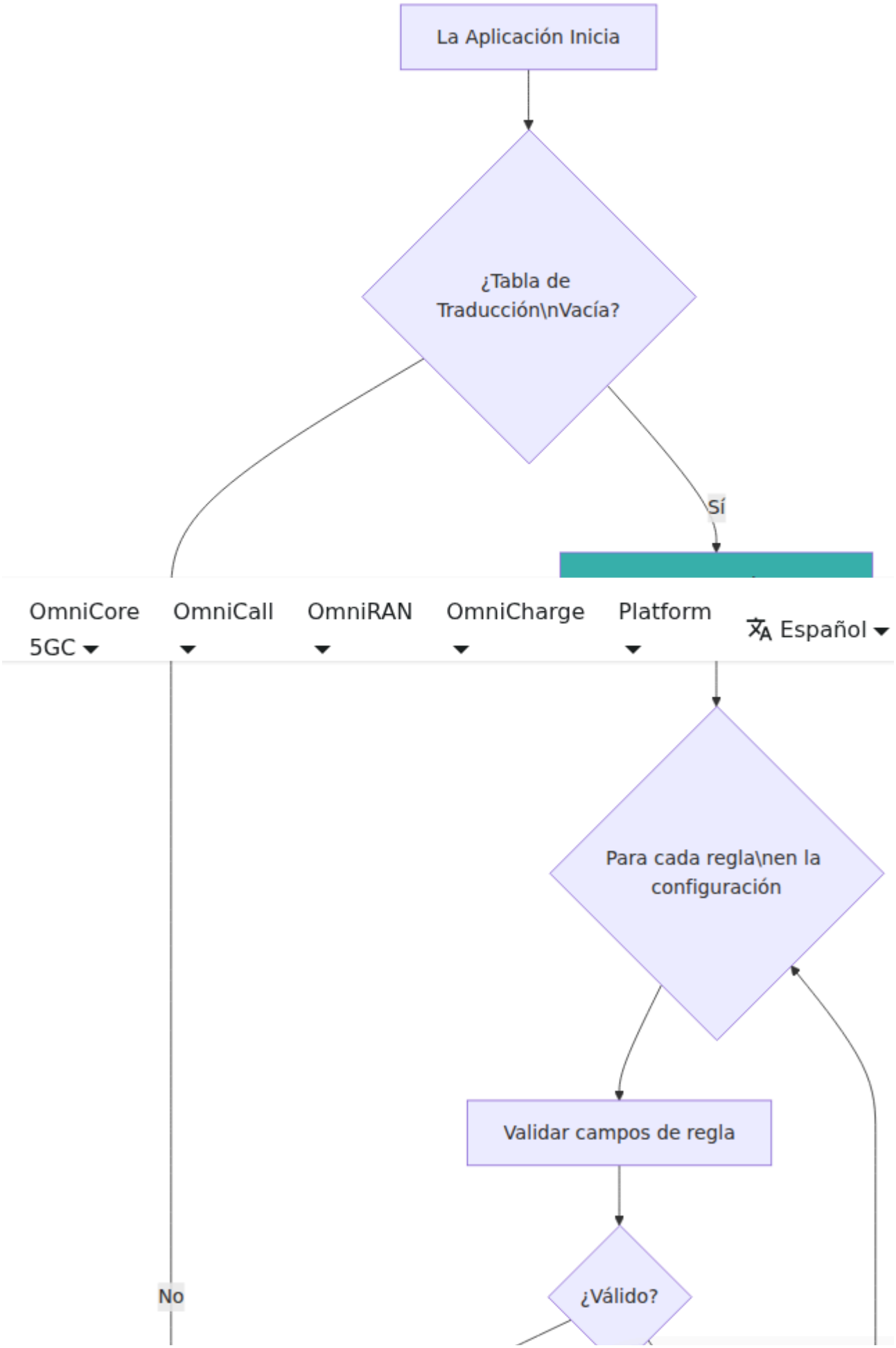
```
    enabled: true,
    continue: true
  },

  # Formatear números del Reino Unido para una 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
  }
]
```

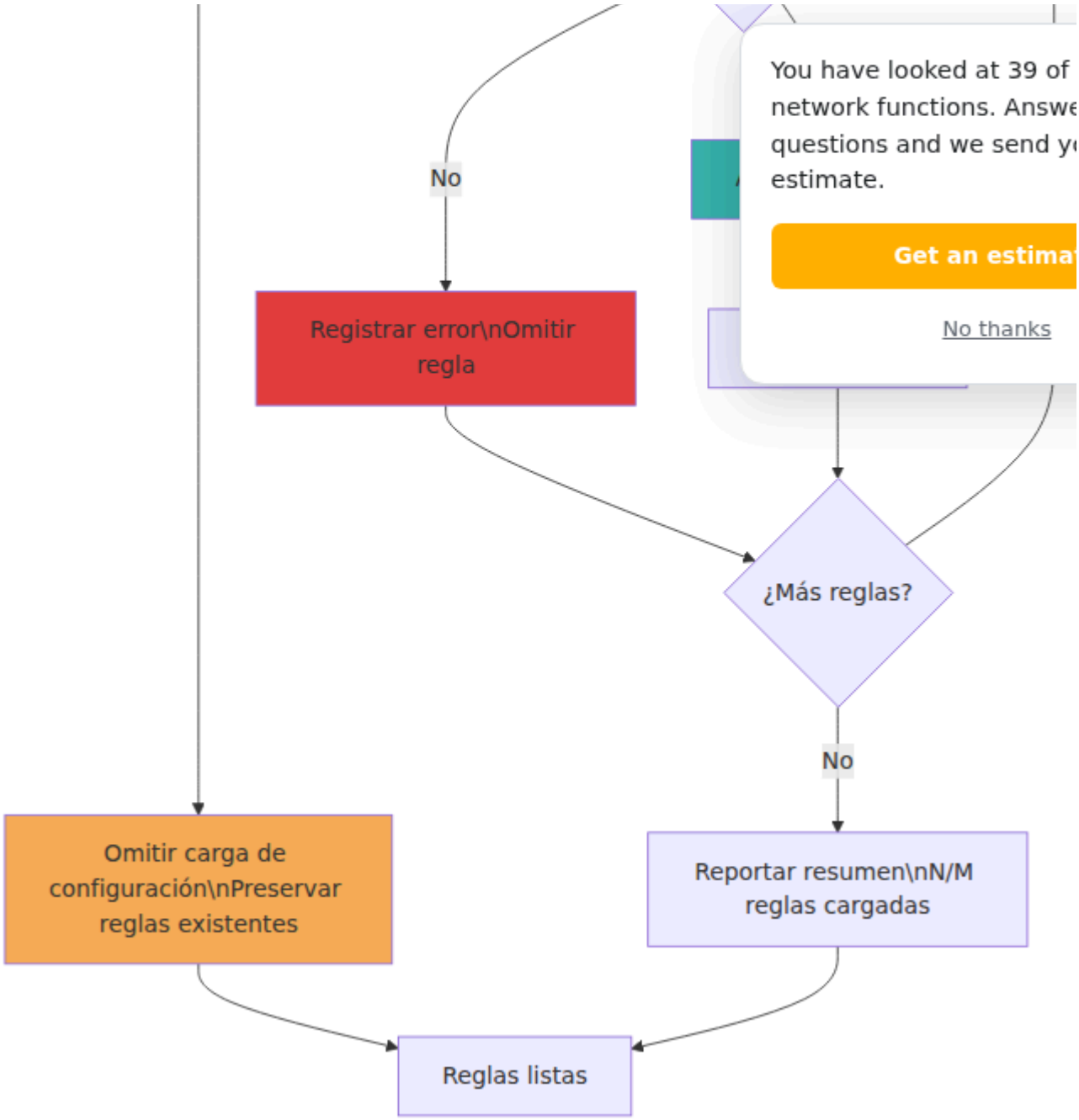
# Comenzando

## Flujo de Inicialización



# Flujo de Traducción de Mensajes

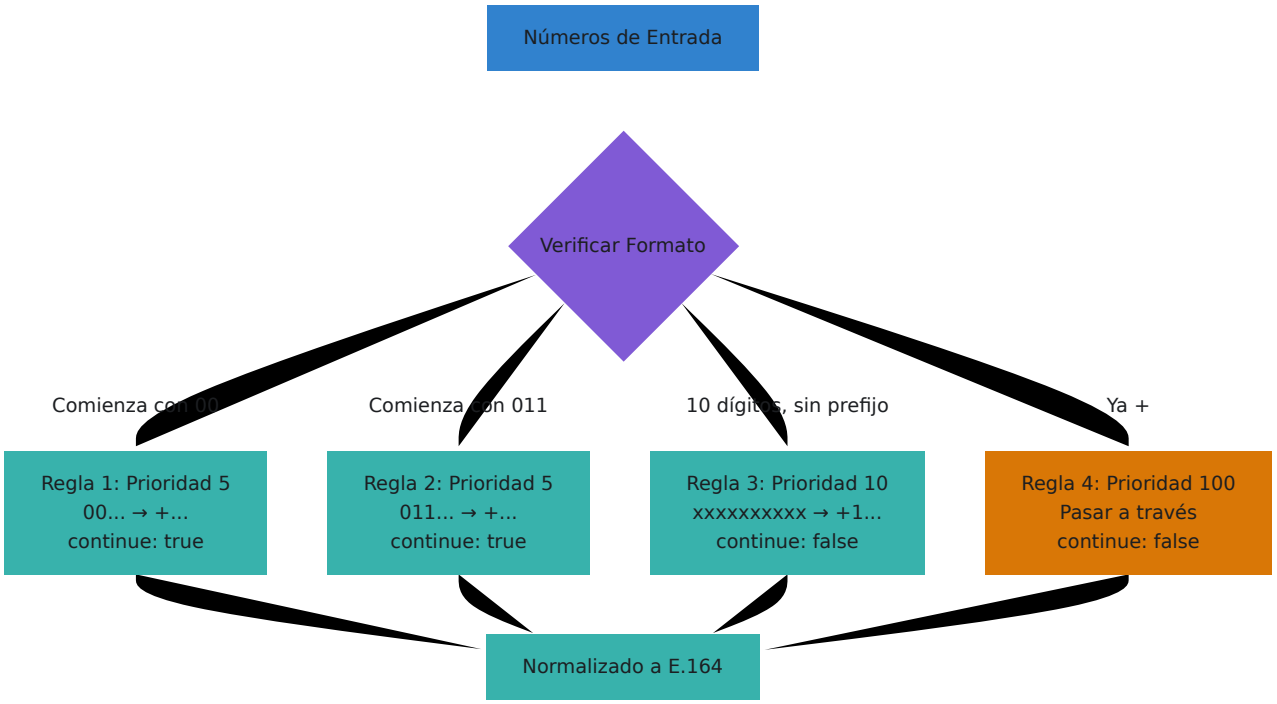

La Aplicación Inicia
¿Tabla de Traducción\nVacía?
Sí
OmniCore 5GC
OmniCall
OmniRAN
OmniCharge
Platform
Español
Para cada regla\nen la configuración
Validar campos de regla
¿Válido?
No




# Casos de Uso Comunes

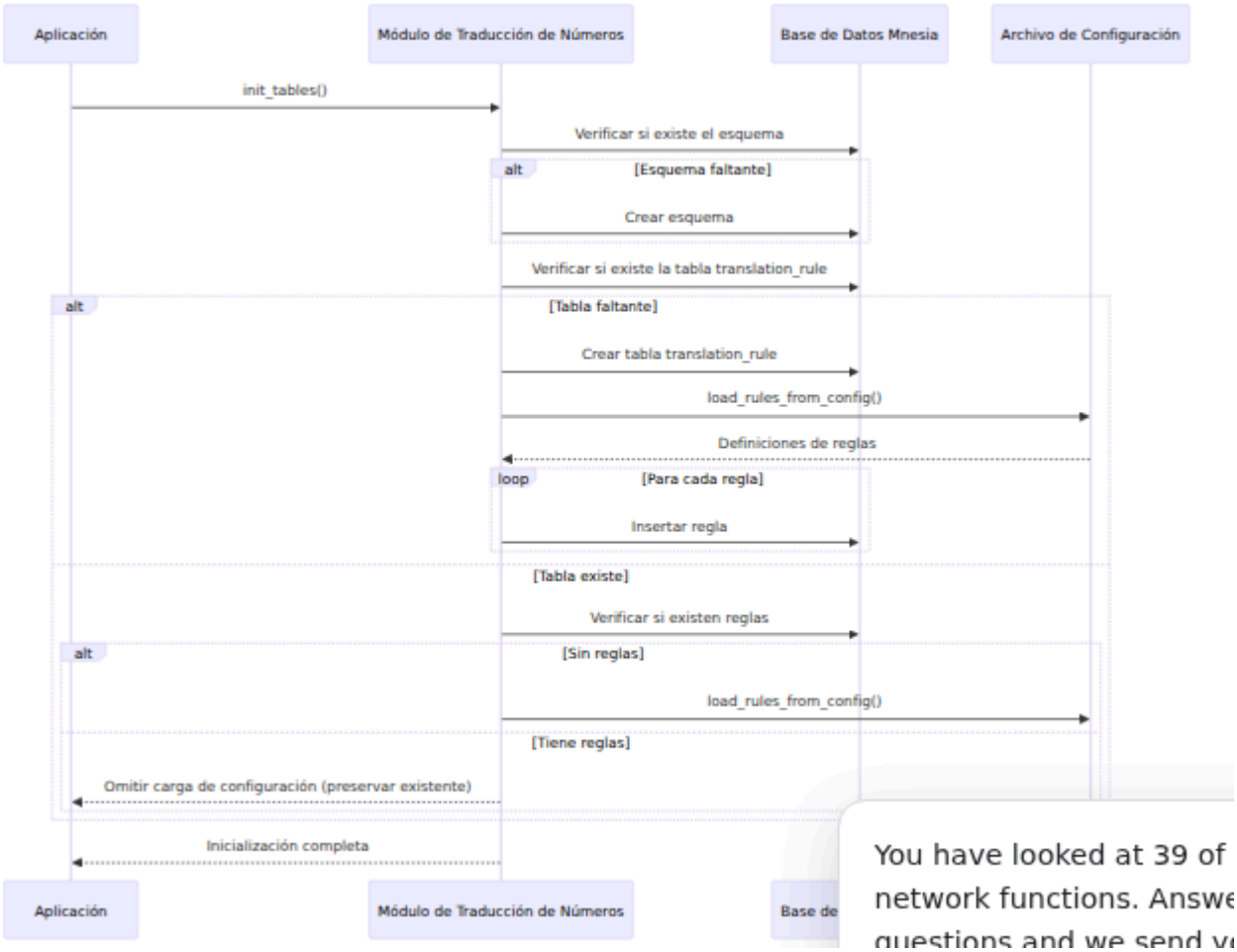
## Normalización de Números Internacionales

Normalizar varios formatos internacionales a E.164:



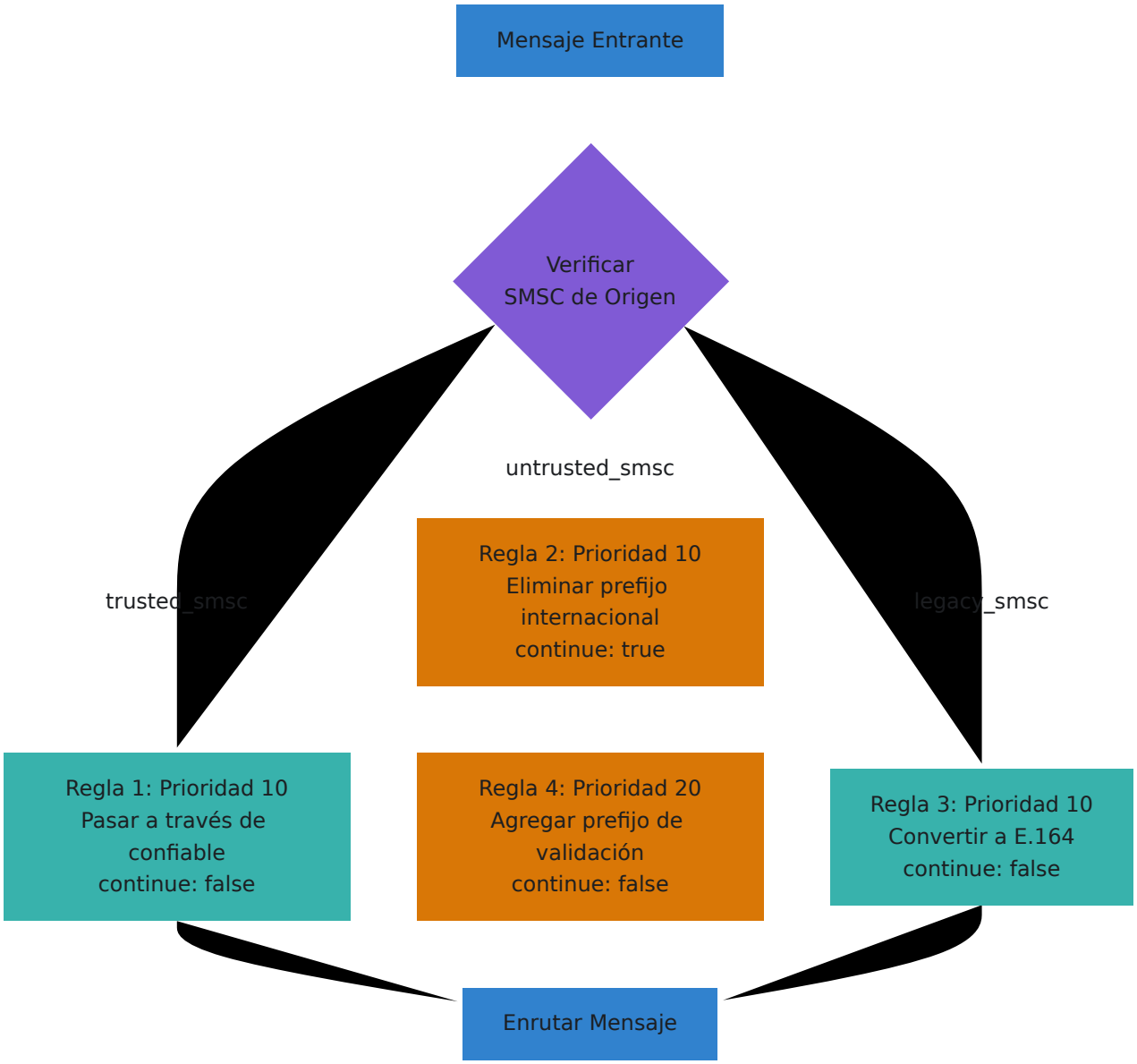

# Formateo Específico de Puerta de Enlace

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



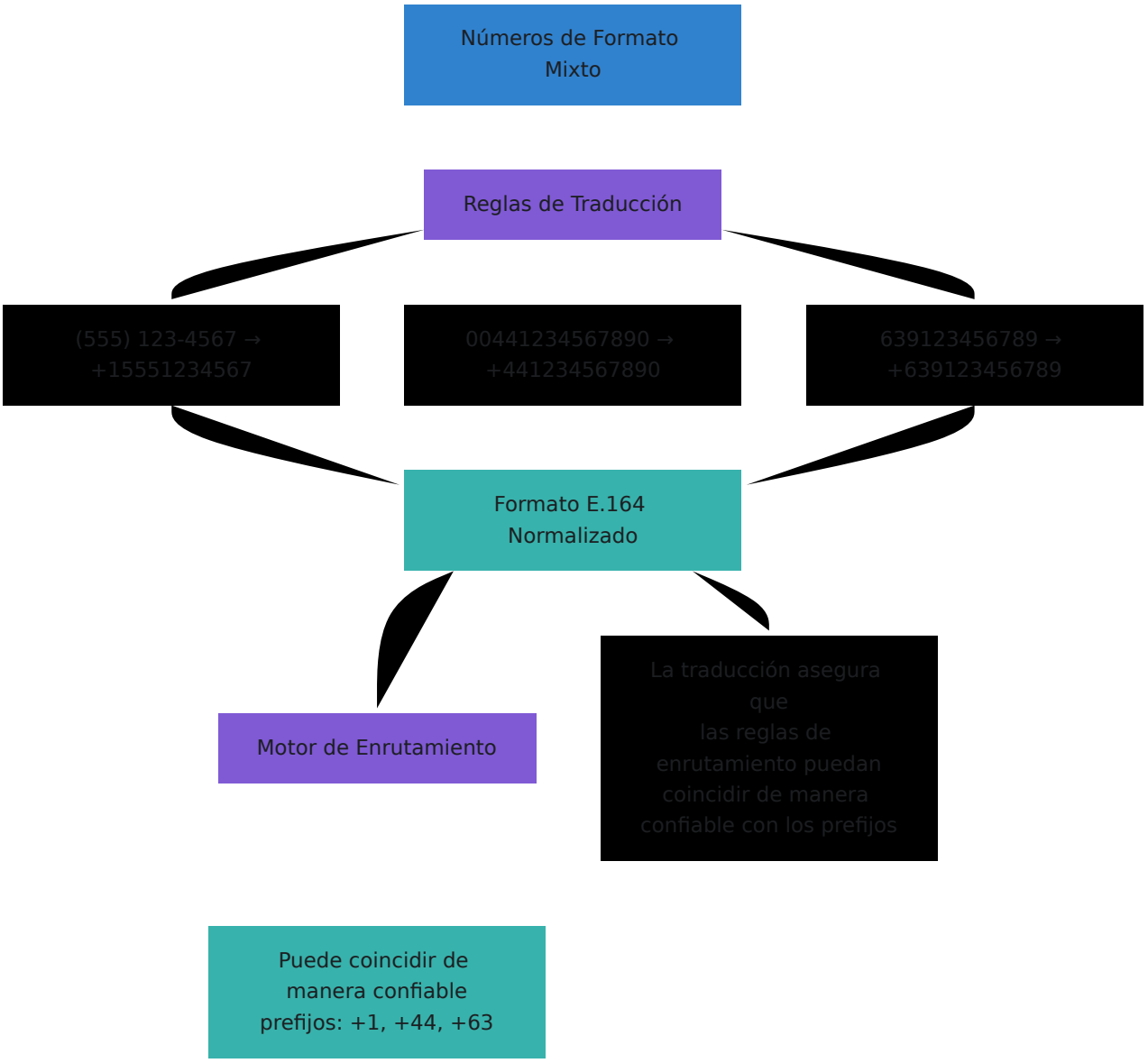

# Traducciones Específicas de SMSC

Aplicar diferentes traducciones según la fuente del mensaje:



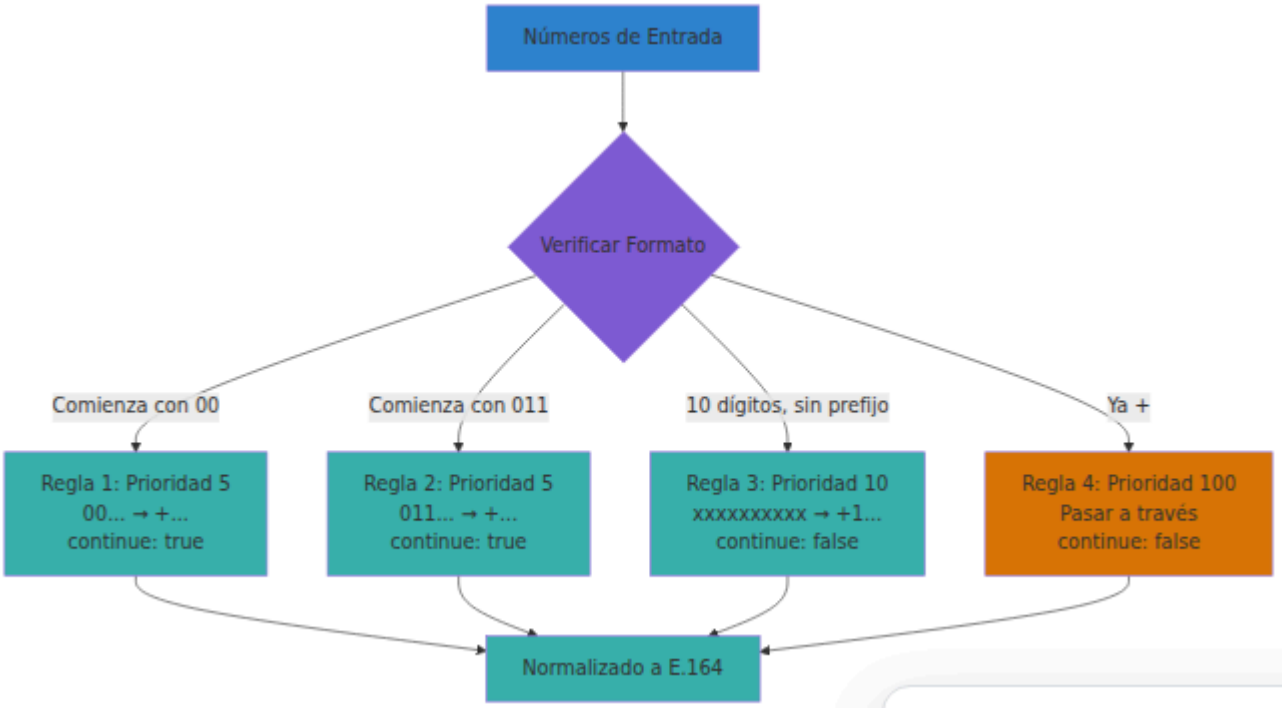

# Preparación de Enrutamiento Basada en Prefijos

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




# Manejo de Portabilidad de Números

Manejar números portados que requieren cambios de prefijo:




# Interfaz Web

## UI de Gestión de Reglas de Traducción

Acceda a la interfaz de gestión de reglas en `/number_translation` (a través del menú de navegación):

**Características:**

- Ver todas las reglas en una tabla ordenable por prioridad
- Agregar nuevas reglas con validación de formularios
- Editar reglas existentes
- Habilitar/deshabilitar reglas sin eliminar
- Eliminar reglas con confirmación
- Indicador visual para reglas con `continue: true`
- Importar/Exportar reglas como JSON

**Agregar una Regla:**

1. Complete los criterios de coincidencia (opcional):
    - Prefijo de llamada (por ejemplo, "+1", "44")

    - Prefijo llamado (por ejemplo, "+639", "1555")
    - SMSC de origen (dejar vacío para cualquier)
2. Defina transformaciones (opcional):
    - Coincidencia y reemplazo de regex para el número de llamada
    - Coincidencia y reemplazo de regex para el número llamado
3. Establezca la anulación de la dirección de origen (opcional):
    - Menús desplegables de **TON de Origen / NPI de Origen** - forzar el Tipo de Número / Indicador de Plan de Numeración de la dirección de origen (integrado en el TPDU). Gateado por el regex de Coincidencia de Número de Llamada.
    - Deje un menú desplegable sin establecer para dejar el valor inferido. Vea Forzar Tipo de Número (TON) y Indicador de Plan de Numeración (NPI).
4. Establezca la prioridad (1-255, menor = mayor prioridad)
5. Establezca el estado:
    - **Habilitado**: La regla está activa
    - **Continuar Procesando**: Continuar evaluando más reglas después de esta
6. Agregue una descripción
7. Haga clic en "Agregar Regla" o "Actualizar Regla"

**Alternar Continuar Procesando:**

- **Detener** (predeterminado): Detener el procesamiento después de que esta regla coincida
- **Continuar**: Aplicar esta regla y continuar evaluando las reglas restantes
- Las reglas con continuar habilitado muestran una insignia azul "↓ Continuar" en la tabla

**Editar una Regla:**

1. Haga clic en "Editar" junto a la regla
2. Modifique los campos según sea necesario
3. Haga clic en "Actualizar Regla"

**Indicadores de Tabla de Reglas:**

- La insignia **Habilitado/Deshabilitado** muestra el estado de la regla
- La insignia **↓ Continuar** muestra reglas que continuarán procesando
- La insignia **Prioridad** muestra el orden de evaluación
- Los patrones regex se muestran en fuente monoespaciada para mayor claridad

# Simulador de Traducción

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

**Características:**

- Probar la lógica de traducción con números reales
- **Transformación Paso a Paso** mostrando cada regla aplicada
- Ver valores antes/después para cada transformación
- Ver qué reglas coincidieron y por qué
- Cargar escenarios de ejemplo para pruebas rápidas
- Ver historial de pruebas (últimas 10 pruebas)

**Usando el Simulador:**

1. Ingrese parámetros de prueba:
    - Número de llamada (de)
    - Número llamado (a)
    - SMSC de origen (opcional)
2. Haga clic en "Probar Traducción"
3. Ver resultados completos:
    - **Resultado de Traducción**: Números finales después de todas las transformaciones
    - **Reglas Aplicadas**: Conteo y lista de todas las reglas que coincidieron
    - **Transformaciones Paso a Paso**: Vista detallada de cada regla:
        - Número de paso e información de la regla
        - Descripción de la regla
        - Antes → Después para ambos números de llamada y llamados
        - Indicador "↓ Continuar" para reglas que continuaron procesando

- Transformaciones resaltadas en verde
- Valores sin cambios marcados como "pasados a través"

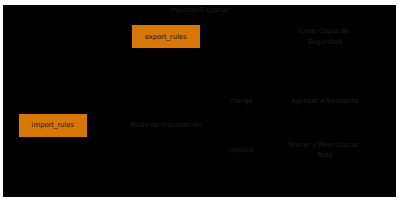
4. Cargar ejemplos preconfigurados usando los botones de ejemplo
5. Revisar el historial de pruebas para comparar diferentes escenarios

**Ejemplo de Salida:**

```
Resultado de Traducción
═══════════════════════════════════════════════
Número de Llamada:  5551234567 → +1-555-123-4567
Número Llamado:   5558720155 → +1-555-872-0155
✓ 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: 5558720155 → +15558720155       │
└───────────────────────────────────────────────┘

┌─ Paso 2 ──────────────────────────────────────┐
│ Regla #2 (Prioridad 20)          ↓ Continuar │
│ Formatear código de área con guiones             │
│                                              │
│ Llamado: +15558720155 → +1-555-8720155   │
└──────────────────────────────────────────────┘

┌─ Paso 3 ──────────────────────────────────────┐
│ Regla #3 (Prioridad 30)                       │
│ Formato final para puerta de enlace              │
│                                              │
│ Llamado: +1-555-8720155 → +1-555-872-0155│
└──────────────────────────────────────────────►
```

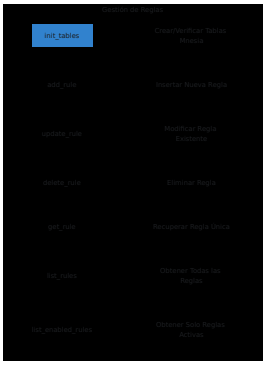
# Referencia de API

## Resumen de Operaciones Centrales










## 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, overrides, [rules_applied]}` - Siempre exitoso
- `overrides` es un mapa `%{source_ton:, source_npi:}` (cada `nil` cuando ninguna regla coincidente lo estableció; última escritura gana cuando múltiples reglas establecen un valor)
- Devuelve números originales y anulaciones vacías si no coinciden reglas
- Devuelve la lista de todas las reglas que se aplicaron (en orden)

```
# Ejemplo de uso
{:ok, new_calling, new_called, _overrides, rules} =
  NumberTranslation.translate_numbers(
    calling_number: "5551234567",
    called_number: "5558720155",
    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 puertas de enlace
   - **51-100**: Transformaciones opcionales
   - **101+**: Reglas de captura total o 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 esperadas

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 prefijo para reducir evaluaciones de regex
   - Eliminar o deshabilitar reglas no utilizadas

2. **Optimizar patrones regex**:

   - Usar coincidencias de prefijo primero (más rápido que regex)
   - Mantener patrones regex simples
   - Evitar patrones que consumen mucho tiempo de retroceso

3. **Limitar encadenamiento de reglas**:

   - Cadenas largas (5+ reglas) pueden afectar el rendimiento
   - Considerar combinar múltiples pasos en una regla si es posible
   - Monitorear la latencia de traducción con métricas de Telemetría

# Operaciones

1. **Probar antes de implementar**:
    - Usar el simulador con ejemplos del mundo real
    - Probar casos límite (números vacíos, caracteres especiales)
    - Verificar el comportamiento de la bandera continuar
2. **Hacer copias de seguridad regularmente**:
    - Exportar reglas antes de realizar cambios importantes
    - Controlar versiones de sus exportaciones
    - Probar importaciones en no producción primero
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 iniciales: `^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: `\(` o `\)`

# Resolución de Problemas

## Regla No Coincidente

**Síntoma**: Se espera que la regla coincida, los números pasan sin cambios

**Causas posibles:**

- El prefijo no coincide (verificar coincidencia exacta del prefijo)
- 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 prefijo (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 incorrectas de grupos de captura (\1, \2, etc.)

**Soluciones:**

1. Usar el simulador para ver las transformaciones paso a paso
2. Verificar que el patrón regex capture los grupos correctos
3. Verificar la sintaxis del patrón de reemplazo
4. Probar el regex en un probador de regex en línea
5. Revisar la prioridad de la regla y las banderas de continuar

# Bucle Infinito / Degradación del Rendimiento

**Síntoma**: La traducción 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. Reportar 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 las banderas 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

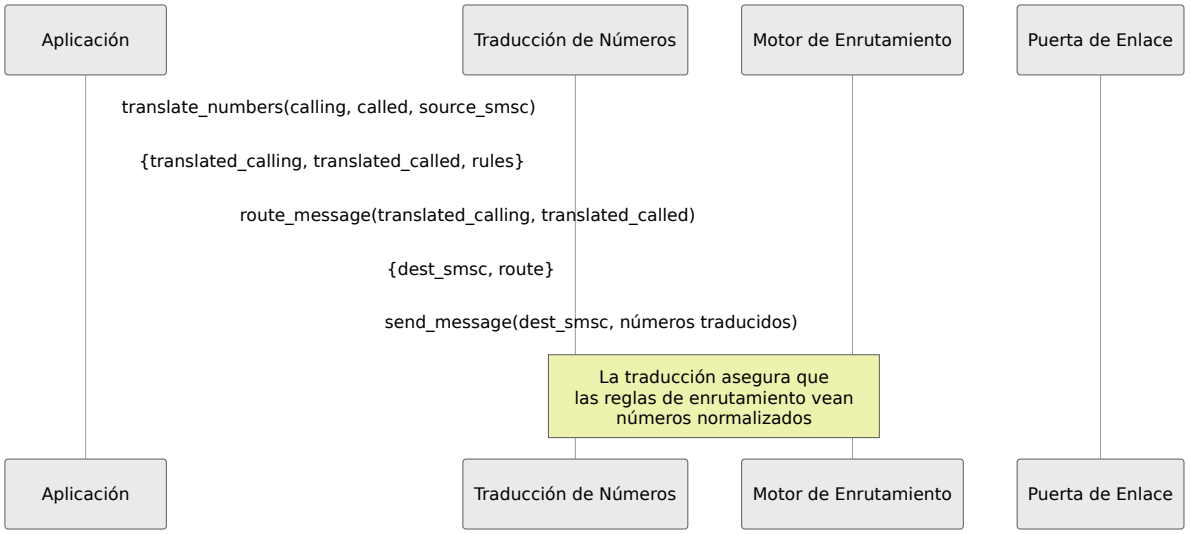
**Soluciones:**

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 la evaluación
- `translation_none`: No coincidieron reglas

Habilitar el registro pasando `message_id` a `translate_numbers/1`.

# Métricas de Telemetría

Monitorear el rendimiento de la traducción con Telemetría:

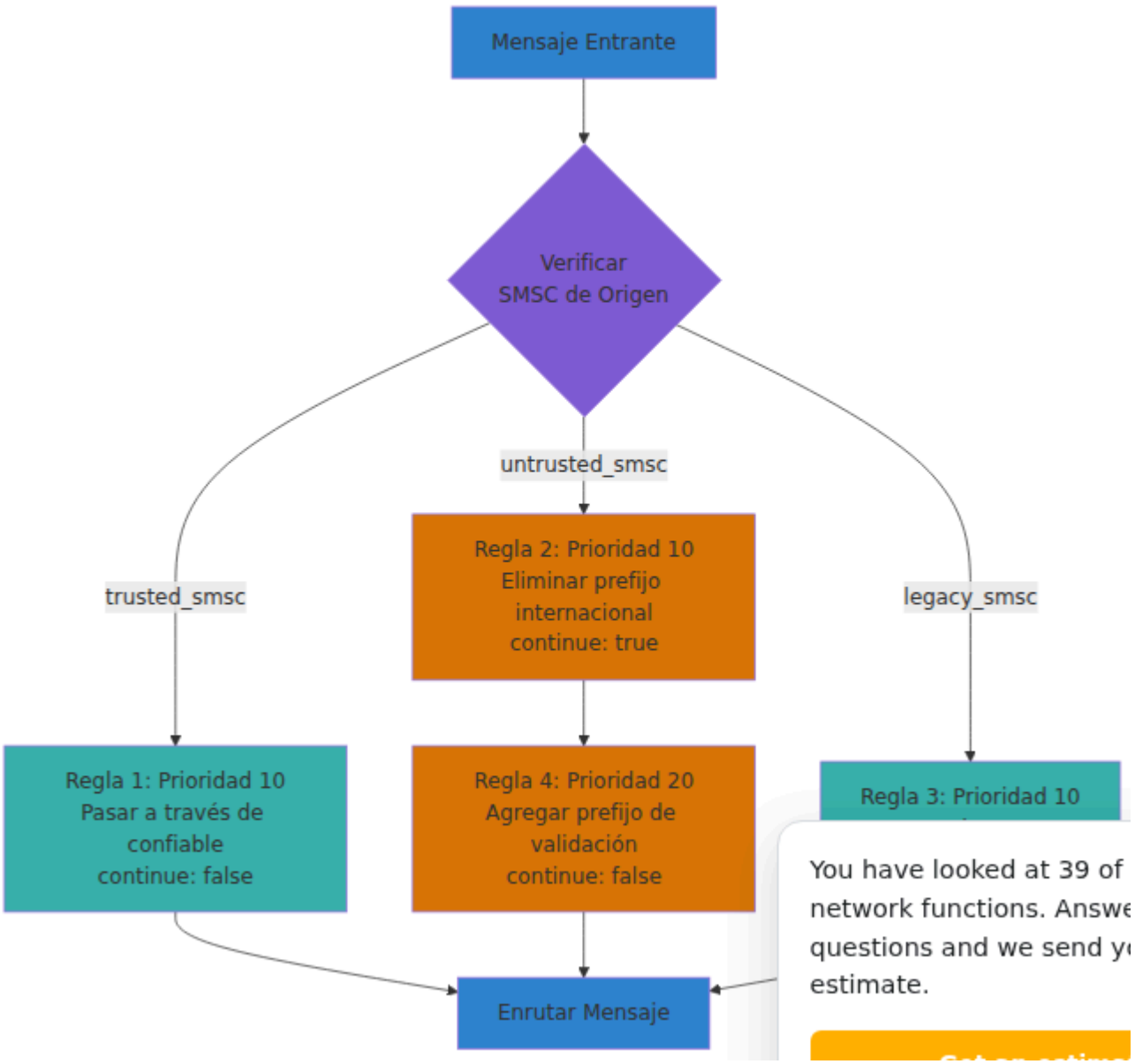
```
:telemetry.attach(
  "number-translation-handler",
  [:sms_c, :number_translation, :translate, :stop],
  fn _event_name, measurements, metadata, _config ->
    # measurements: %{duration: microseconds}
    # metadata: %{rules_applied: count, ...}
  end,
  nil
)
```

**Métricas clave a monitorear:**

- Duración de la traducción (p50, p95, p99)
- Reglas aplicadas por mensaje
- Reglas coincidentes vs no coincidentes
- Uso de la bandera continuar

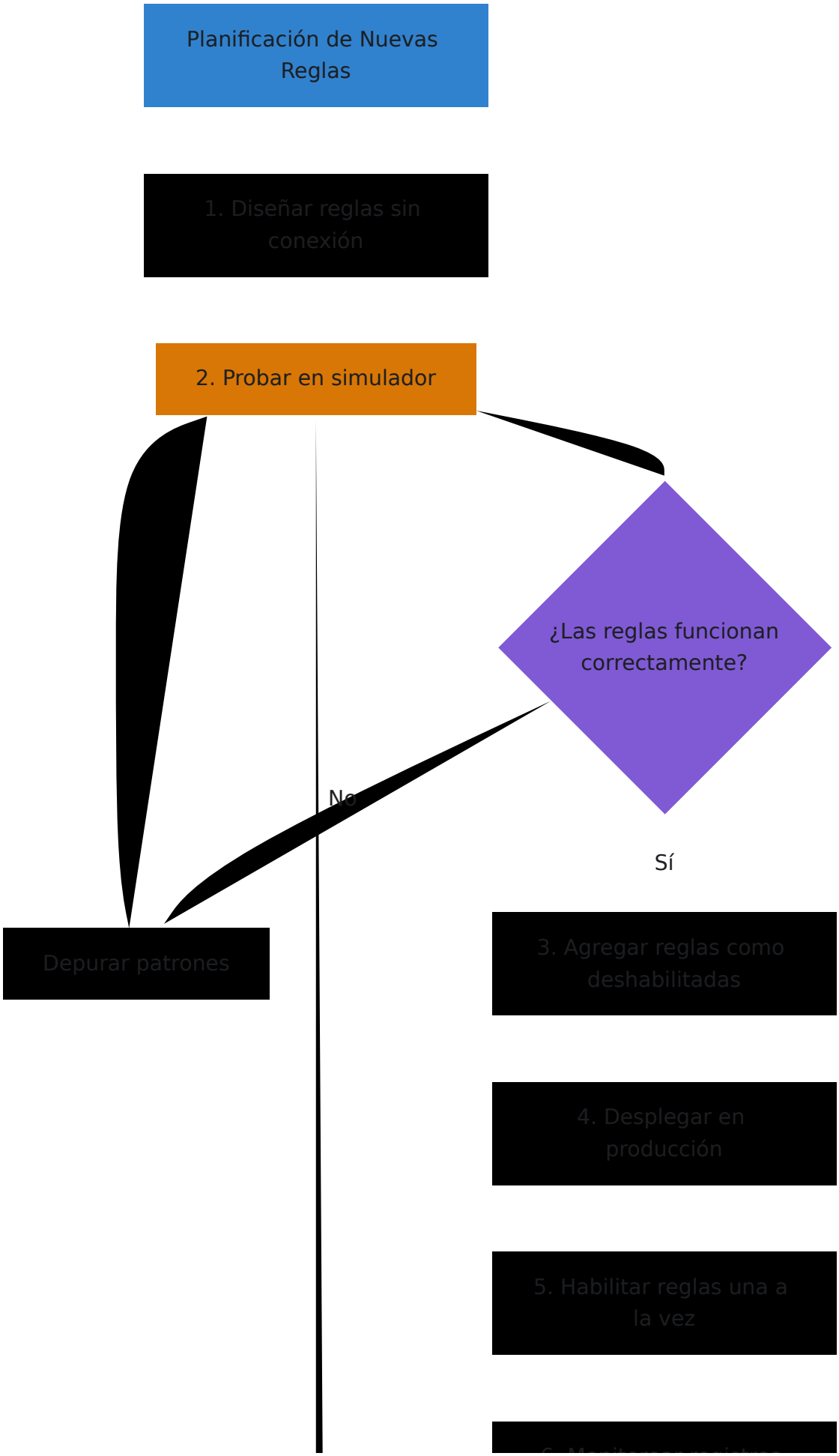
# Clustering

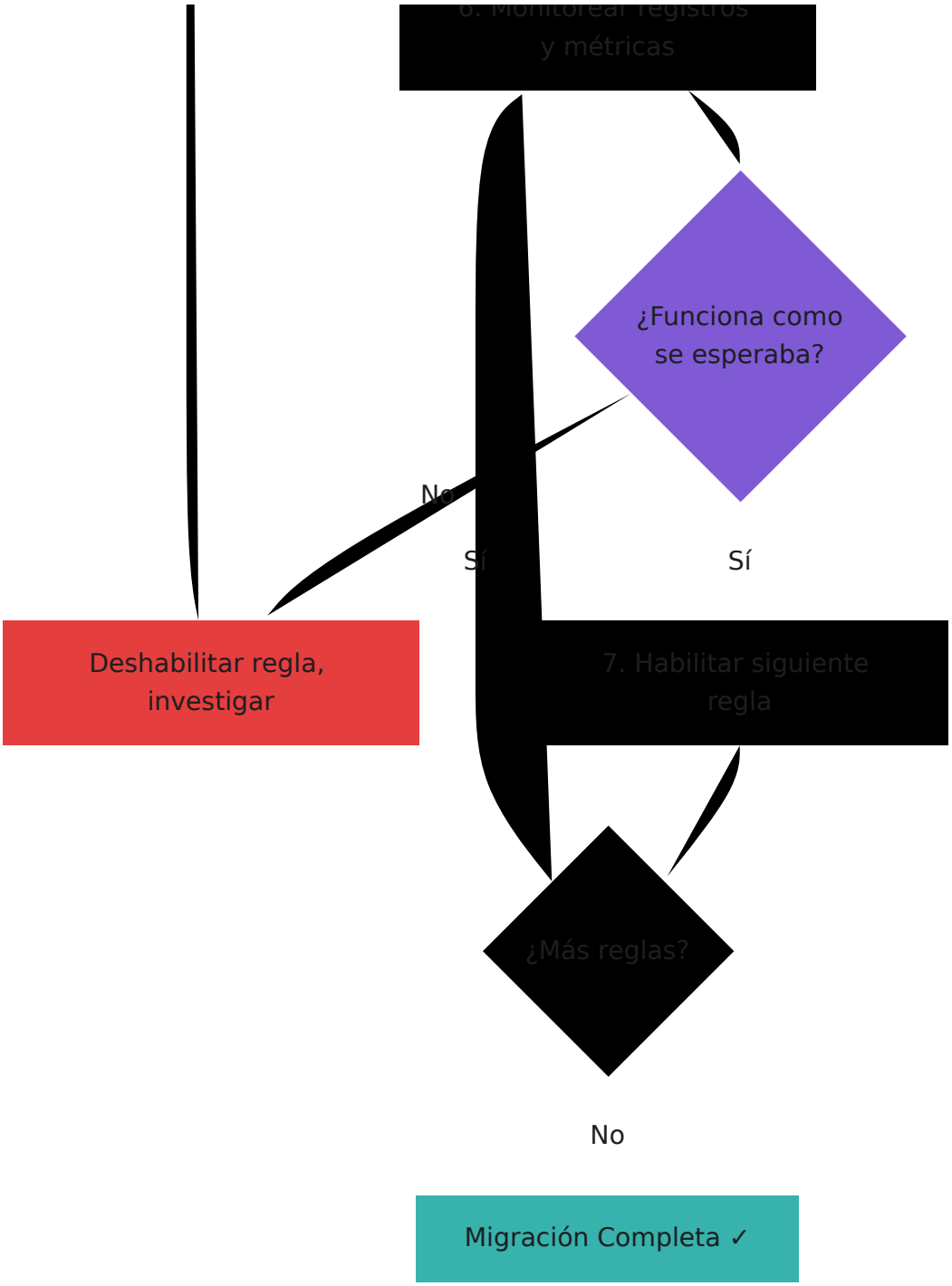
Las tablas de Mnesia se distribuyen automáticamente entre nodos en clúster. Las reglas de traducción se replican para alta disponibilidad.




## Estrategias de Migración

Al implementar nuevas reglas de traducción:


Planificación de Nuevas Reglas
1. Diseñar reglas sin conexión
2. Probar en simulador
¿Las reglas funcionan correctamente?
No
Sí
Depurar patrones
3. Agregar reglas como deshabilitadas
4. Desplegar en producción
5. Habilitar reglas una a la vez




# 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 prefijo 00 a +, luego formatear para puerta de enlace

```
# Regla 1: Convertir 00 a + (continúa a la siguiente regla)
%{
  calling_match: "^00(.+)$",
  calling_replace: "+\1",
  called_match: "^00(.+)$",
  called_replace: "+\1",
  priority: 5,
  description: "Convertir prefijo internacional 00 a +",
  enabled: true,
  continue: true  # Continuar para formatear
}

# Regla 2: Formatear para puerta de enlace (detiene el
procesamiento)
%{
  calling_match: "^\+(\d+)$",
  calling_replace: "00\1",
  called_match: "^\+(\d+)$",
  called_replace: "00\1",
  priority: 10,
  description: "Formatear números + como 00 para puerta de
enlace",
  enabled: true,
  continue: false  # Detener después de esto
}

# Caso de prueba:
# Paso 1: "00441234567890" → "+441234567890" (Regla 1, continuar)
# Paso 2: "+441234567890" → "00441234567890" (Regla 2, detener)
# Resultado: "00441234567890"
# Reglas aplicadas: [Regla 1, Regla 2]
```

## Ejemplo 3: Manejo Específico de SMSC

**Requisito**: Aplicar diferentes reglas según el SMSC de origen

```
# Regla 1: SMSC de confianza - pasar a través (prioridad 5)
%{
  source_smsc: "trusted_gateway",
  calling_match: nil,  # Sin transformación
  calling_replace: nil,
  called_match: nil,
  called_replace: nil,
  priority: 5,
  description: "Pasar números de puerta de enlace confiable",
  enabled: true,
  continue: false
}

# Regla 2: SMSC no confiable - normalizar (prioridad 10)
%{
  source_smsc: "untrusted_gateway",
  calling_match: "^(.*)$",
  calling_replace: "+VALIDATE\1",
  called_match: "^(.*)$",
  called_replace: "+VALIDATE\1",
  priority: 10,
  description: "Agregar prefijo de validación para fuente no
confiable",
  enabled: true,
  continue: false
}

# Regla 3: Captura total para otros SMSCs (prioridad 100)
%{
  source_smsc: nil,  # Comodín
  calling_match: "^(\d{10})$",
  calling_replace: "+1\1",
  called_match: "^(\d{10})$",
  called_replace: "+1\1",
  priority: 100,
  description: "Predeterminado: 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 iniciales (continuar)
%{
  calling_match: "^0+(.+)$",
  calling_replace: "\1",
  called_match: "^0+(.+)$",
  called_replace: "\1",
  priority: 5,
  description: "Eliminar ceros iniciales",
  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:

- Utilice el simulador para trabajar a través de ejemplos trabajados y depurar la lógica de traducción
- Revisar registros de eventos para decisiones de traducción
- Monitorear el contenido de la tabla Mnesia: `:mnesia.table_info(:translation_rule, :size)`
- Monitorear métricas de Telemetría para problemas de rendimiento

# Guía de Operaciones de SMS-C

**← Volver al Índice de Documentación**

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 control de Prometheus y verifique:

- Rendimiento de mensajes (últimas 24 horas)
- Tasa de fallos de enrutamiento (debe ser < 1%)
- Respaldo 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 revise los archivos de registro

Busque:

- Mensajes de nivel de error
- Fallos de enrutamiento
- Fallos de carga
- Problemas de conexión a la base de datos
- Problemas con nodos del clúster

# Monitoreo del Volumen de Mensajes

**Verificar Cuentas de Mensajes por Hora**:

Utilice la consulta de Prometheus:

```
# Mensajes recibidos por hora
increase(sms_c_message_received_count[1h])

# Mensajes entregados por hora
increase(sms_c_delivery_succeeded_count[1h])

# Calcular tasa de entrega
rate(sms_c_delivery_succeeded_count[1h]) /
rate(sms_c_message_received_count[1h])
```

**Patrones Esperados**:

- Horas laborales: Mayor volumen
- Noches/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 Vigilar

### 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 el sistema no pudo enrutear
- **Alerta**: Cualquier fallo (> 0)
- **Consulta**: `increase(sms_c_routing_failed_count[5m])`

**Ruta Coincidente** (`sms_c_routing_route_matched_count`):

- **Qué**: Qué rutas se están utilizando
- **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 la 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** (Se Requiere Respuesta Inmediata):

```
# No se encontró ruta - los mensajes no pueden ser entregados
- alert: RoutingFailures
  expr: increase(sms_c_routing_failed_count[5m]) > 0
  severity: critical
  description: "{{ $value }} mensajes fallaron en el enrutamiento
en los últimos 5 minutos"

# Cola acumulándose - el procesamiento se está quedando atrás
- alert: QueueBacklog
  expr: sms_c_queue_size_pending > 10000
  severity: critical
  description: "La cola tiene {{ $value }} mensajes pendientes"

# Mensajes envejeciendo - entrega atascada
- alert: OldMessagesInQueue
  expr: sms_c_queue_oldest_message_age_seconds > 300
  severity: critical
  description: "El mensaje más antiguo tiene {{ $value }}
segundos"

# Frontend desconectado - sin ruta de entrega
- alert: FrontendDisconnected
  expr: sms_c_frontend_status_count{status="disconnected"} > 0
  severity: critical
  description: "{{ $value }} frontends desconectados"
```

**Alertas de Advertencia** (Se Necesita Investigación):

```
# Tasa de éxito de entrega en descenso
- 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: "95 percentil de intentos de entrega: {{ $value }}"

# Búsquedas de ENUM lentas o fallidas
- 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 todos los detalles 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 (retraso de 2 min)
   ↓
5. delivery_attempt_3 - Tercer intento fallido (retraso de 4 min)
   ↓
6. message_dead_letter - Se superó el límite de reintentos
```

# Verificar Estado de Entrega

**Esperando Entrega**:

- Estado: "queued" (listo para enviar ahora) o "backoff" (esperando un reintento)
- deliver_after: nil/pasado para `queued`, una marca de tiempo futura para `backoff`
- delivery_attempts: 0 para un mensaje `queued` nuevo, mayor para `backoff`

**Mensajes Entregados**:

- Estado: "delivered" (entrega confirmada) o "sent" (enviado, sin DLR aún)
- deliver_time: Marca de tiempo de entrega
- dest_smsc: Frontend que entregó

**Mensajes No Entregados / Fallidos**:

- Estado: "backoff" con altos delivery_attempts - aún reintentando (limitado a un intervalo de 30 minutos, hasta que se alcance el `expires` del mensaje)
- Estado: "expired", "dropped", "balance_rejected" o "auto_replied" - terminal
- Verifique el registro de eventos para razones de fallo

# Enrutamiento de Mensajes Basado en Ubicación

El SMS-C admite la recuperación de mensajes basada en la ubicación, lo que permite a los frontends recibir automáticamente mensajes destinados a suscriptores registrados en su ubicación.

**Cómo Funciona**:

Cuando un frontend consulta mensajes pendientes utilizando `get_messages_for_smsc(smsc_name)`, el sistema devuelve mensajes de dos maneras:

1. **Enrutamiento Explícito** - Mensajes donde `dest_smsc` coincide explícitamente con el nombre del frontend
2. **Enrutamiento Basado en Ubicación** - Mensajes donde:
   - `dest_smsc` es `null` (no enrutado explícitamente)
   - `destination_msisdn` tiene un registro de ubicación activo
   - El campo `location` de la ubicación coincide con el nombre del frontend
   - La ubicación no ha expirado

**Escenario de Ejemplo**:

Un suscriptor con MSISDN `+447700900123` se registra en el frontend `uk_gateway`:

```
# Suscriptor se registra (crea registro de ubicación)
POST /api/locations
{
  "msisdn": "+447700900123",
  "imsi": "234150123456789",
  "location": "uk_gateway",
  "expires": "2025-11-01T12:00:00Z"
}
```

Cuando llega un mensaje para este suscriptor sin enrutamiento explícito:

```
# Mensaje enviado sin dest_smsc
POST /api/messages
{
  "source_msisdn": "+15551234567",
  "destination_msisdn": "+447700900123",
  "message_body": "Hello",
  "source_smsc": "api"
  # Nota: dest_smsc es null
}
```

El frontend `uk_gateway` recibirá automáticamente este mensaje cuando realice una consulta:

```
# Frontend consulta mensajes
GET /api/messages/queue?smsc=uk_gateway

# Devuelve el mensaje aunque dest_smsc sea null
# porque el suscriptor de destino está registrado en uk_gateway
```

**Requisitos de Ubicación**:

Para que el enrutamiento basado en ubicación funcione:

- La tabla `locations` debe tener una entrada para el `destination_msisdn`
- El campo `location` debe coincidir con el nombre del SMSC que consulta
- La marca de tiempo `expires` debe estar en el futuro

**Monitoreo del Enrutamiento Basado en Ubicación**:

Verifique los registros de ubicación:

```
# A través de la API
GET /api/locations/{msisdn}

# Verifique si la ubicación ha expirado
# el campo expires debe ser > hora 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/drop)
   - **Solo En Red**: Restringir esta ruta solo a destinos en red (requiere Diameter/HSS)
   - **Prioridad**: Prioridad de la ruta (1-255, menor = mayor prioridad)
   - **Peso**: Peso de balanceo de carga (1-100)

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

Ruta 2:

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

**Ejemplo: Ruta Solo En Red**:

Restringir a un operador externo o aplicación interna para que solo envíe a suscriptores en red. Los mensajes a destinos fuera de la red son descartados.

- SMSC de Origen: `carrier_smpp_bind`

- SMSC de Destino: `local_msc`
- Solo En Red: marcado
- Prioridad: 50
- Peso: 100
- Descripción: "Carrier X - solo en red"

Esto requiere que Diameter/HSS esté habilitado. Si el HSS no está disponible, los mensajes se descartan (fail-closed).

# 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 eligió el sistema
   - **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" (remoción temporal)
- **Ajustar Peso**: Cambiar la distribución de balanceo de carga
- **Cambiar Prioridad**: Reordenar la evaluación de rutas
- **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 todo e importar

**Casos de Uso**:

- Copia de seguridad antes de cambios importantes
- Copiar rutas entre entornos
- Recuperación de desastres
- Versionado de configuración

# Gestión de Frontend

## Monitorear Conexiones de Frontend

**Interfaz Web**: Navegue a `/frontend_status`

**Verifique**:

- Todos los frontends esperados están "activos"
- Los tiempos de última vista son recientes (< 90 segundos)
- No hay frontends expirados inesperados

**A través de la API**:

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

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

## Investigar Desconexiones

**Frontend Expirado**:

1. Verifique los registros del frontend para errores
2. Verifique la conectividad de red al SMS-C
3. Confirme que el frontend esté en ejecución
4. Verifique la lógica de registro del frontend (debe registrarse nuevamente cada 60s)

**Registro No Mostrando**:

1. Verifique que el frontend esté llamando a POST `/api/frontends/register`
2. Revise los registros de la API para errores de registro
3. Verifique el formato del payload JSON
4. Pruebe el registro manualmente con curl

**Ejemplo de Registro Manual**:

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

# Ver Historial de Frontend

**Interfaz Web**:

1. Navegue a `/frontend_status`
2. Encuentre el frontend en la lista
3. Haga clic en "Historial"
4. Revise 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 EE. UU. de 10 dígitos",
  enabled: true
}
```

**Normalizar Formato Internacional**:

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

**Eliminación de Código Específico del Transportista**:

```
%{
  calling_prefix: nil,
  called_prefix: "101",
  source_smsc: "carrier_a",
  calling_match: nil,
  calling_replace: nil,
  called_match: "^101(\d+)$",
  called_replace: "\1",
  priority: 5,
  description: "Eliminar código de transportista del transportista
A",
  enabled: true
}
```

# Probar Reglas de Traducción

Después de los cambios de configuración:

1. Reinicie la aplicación para cargar nuevas reglas
2. Envíe un mensaje de prueba con origen/destino que debería coincidir
3. Verifique el registro de eventos para el evento `number_translated`
4. Verifique que los números se transformaron correctamente

## Deshabilitar Regla de Traducción

Establezca `enabled: false` en la regla:

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

Reinicie la aplicación.

# Mantenimiento del Sistema

## Mantenimiento de Base de Datos

**Verificar Tamaño de Base de Datos**:

Utilice sus herramientas de gestión de bases de datos para monitorear el tamaño de almacenamiento de CDR:

- **MySQL/MariaDB**: Consulte `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 archivarse y purgarse 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 en un almacén de datos o almacenamiento en frío
- Elimine registros archivados de la base de datos operativa en lotes para evitar contención de bloqueos

**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 baja actividad
- **PostgreSQL**: Ejecute `VACUUM ANALYZE` regularmente (o habilite autovacuum)

**Ejecutar Semanalmente** durante un período de baja actividad para mantener un rendimiento óptimo.

# Mantenimiento de Base de Datos Mnesia

**Verificar Tamaño de Tabla Mnesia**:

```
# En la consola IEx
:mnesia.table_info(:sms_route, :size)
:mnesia.table_info(:translation_rule, :size)
```

**Respaldar Tablas Mnesia**:

```
# Exportar rutas (Interfaz Web)
# Navegue a /sms_routing
# Haga clic en "Exportar Rutas"

# O a través de respaldo de Mnesia
:mnesia.backup("/var/backups/sms_c/mnesia_backup.bup")
```

**Restaurar Mnesia**:

```
# A través de la importación de la Interfaz Web
# O restaurar respaldo:
:mnesia.restore("/var/backups/sms_c/mnesia_backup.bup", [])
```

# Rotación de Registros

Configure logrotate para los registros de la aplicación:

```
# /etc/logrotate.d/sms_c
/var/log/sms_c/*.log {
    daily
    rotate 30
    compress
    delaycompress
    notifempty
    create 0644 sms_user sms_group
    sharedscripts
    postrotate
        systemctl reload omnimessage || true
    endscript
}
```

# Reiniciar Aplicación

**Reinicio Graceful** (cero tiempo de inactividad en clúster):

```
# Reinicie un nodo a la vez
systemctl restart omnimessage

# Espere a que el nodo se una al clúster
# Repita para cada nodo
```

**Reinicio de Emergencia** (todos los nodos):

```
systemctl restart omnimessage
```

**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 Respaldar

1. **Archivos de Configuración**:
    - `config/runtime.exs`
    - `config/config.exs`
    - `config/prod.exs` (si existe)
2. **Tablas de Enrutamiento** (Mnesia):
    - Exportar a través de la Interfaz Web
    - O comando de respaldo de Mnesia
3. **Base de Datos SQL CDR**:
    - Copia de seguridad completa diaria
    - Copias de seguridad de registros de transacciones (continuas)
4. **Certificados TLS**:
    - `priv/cert/*.crt`
    - `priv/cert/*.key`

## Procedimientos de Copia de Seguridad

**Copia de Seguridad de Configuración Diaria**:

```
#!/bin/bash
# /opt/omnimessage/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/omnimessage/config $BACKUP_DIR/

# Copia de seguridad de certificados
cp -r /opt/omnimessage/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/omnimessage/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: Utilizar mysqldump con --single-transaction para
consistencia
# PostgreSQL: Utilizar pg_dump -F c para formato personalizado

# Estructura de ejemplo (adapte a su base de datos):
# - Utilice la herramienta de copia de seguridad apropiada
(mysqldump, pg_dump)
# - Habilite copias de seguridad seguras para consistencia
# - Comprimir salida para ahorrar espacio
# - Configurar período de retención (por ejemplo, 30 días)

# Eliminar copias de seguridad antiguas
find $BACKUP_DIR -name "sms_c_*.gz" -mtime +30 -delete

echo "Copia de seguridad de base de datos completada:
sms_c_${DATE}"
```

**Copia de Seguridad de Tablas de Enrutamiento**:

```
#!/bin/bash
# /opt/omnimessage/scripts/backup_routes.sh

BACKUP_DIR="/var/backups/sms_c/routes"
DATE=$(date +%Y%m%d)

mkdir -p $BACKUP_DIR

# Exportar a través de la API
curl https://api.example.com:8443/api/routes/export \
  > $BACKUP_DIR/routes_${DATE}.json

echo "Copia de seguridad de rutas completada: routes_${DATE}.json"
```

**Programar Copias de Seguridad** (crontab):

```
# Diario a las 2 AM
0 2 * * * /opt/omnimessage/scripts/backup_config.sh
0 2 * * * /opt/omnimessage/scripts/backup_database.sh
0 2 * * * /opt/omnimessage/scripts/backup_routes.sh
```

# Procedimientos de Recuperación

**Restaurar Configuración**:

```
# Detener aplicación
systemctl stop omnimessage

# Restaurar archivos de configuración
cp -r /var/backups/sms_c/20251030/config/*
/opt/omnimessage/config/

# Restaurar certificados
cp -r /var/backups/sms_c/20251030/cert/*
/opt/omnimessage/priv/cert/

# Iniciar aplicación
systemctl start omnimessage
```

**Restaurar Base de Datos SQL CDR**:

Utilice herramientas de restauración apropiadas para su base de datos:

- **MySQL/MariaDB**: Descomprima y canalice al cliente mysql
- **PostgreSQL**: Utilice pg_restore con volúmenes de volcado de formato personalizado

**Importante**: Detenga la aplicación SMS-C antes de restaurar la base de datos para evitar conflictos de datos.

**Restaurar Tablas de Enrutamiento**:

1. Navegue a la Interfaz Web `/sms_routing`

2. Haga clic en "Importar Rutas"
3. Seleccione el archivo JSON de respaldo
4. Elija el modo "Reemplazar"
5. Confirme la importación

# Planificación de Capacidad

## Monitorear Tendencias de Crecimiento

**Tendencia de Volumen de Mensajes**:

Consulta de Prometheus (promedio de 30 días):

```
avg_over_time(sms_c_message_received_count[30d])
```

**Tasa de Crecimiento de 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% de promedio
- **Alto**: > 70% sostenido
- **Crítico**: > 90%

**Uso de Memoria**:

- **Normal**: < 70% de disponible
- **Alto**: > 80%
- **Crítico**: > 90%

**Uso de Disco**:

- **Normal**: < 60% lleno
- **Alto**: > 75%
- **Crítico**: > 85%

**Profundidad de Cola**:

- **Normal**: < 1000 pendientes
- **Alto**: > 5000 pendientes
- **Crítico**: > 10,000 pendientes

# Recomendaciones de Escalado

**Cuándo Escalar Verticalmente** (Actualizar Recursos):

- CPU consistentemente > 70%
- Memoria consistentemente > 80%
- Cuello de botella de nodo único

**Cuándo Escalar Horizontalmente** (Agregar Nodos):

- CPU > 50% en todos los nodos
- Volumen de mensajes > 5,000 msg/sec
- Distribución geográfica necesaria
- Alta disponibilidad requerida

**Escalado de Base de Datos**:

- Réplicas de lectura para consultas de informes
- Optimización de agrupamiento de conexiones
- Optimización de índices
- Particionar tablas grandes por fecha

# Respuesta a Incidentes

## Niveles de Severidad

**Crítico** (Respuesta Inmediata):

- No se entregan mensajes
- Todos los frontends desconectados
- Base de datos no disponible
- API completamente caída

**Alto** (Respuesta dentro de 1 hora):

- Tasa de éxito de entrega < 80%
- Múltiples frontends desconectados
- Fallos de enrutamiento > 10%
- Respaldo de cola en crecimiento

**Medio** (Respuesta dentro de 4 horas):

- Un solo frontend desconectado
- Tasa de éxito de entrega 80-95%
- Procesamiento de mensajes lento
- Fallos en búsquedas de ENUM

**Bajo** (Respuesta dentro de 24 horas):

- Degradación menor del rendimiento
- Problema en una sola ruta
- Alertas de advertencia no críticas

## Lista de Verificación de Incidentes

**1. Evaluar Severidad**:

- Verifique alertas de Prometheus
- Revise métricas del panel de control

- Verifique el estado de la cola de mensajes
- Verifique conexiones de frontend

**2. Reunir Información**:

- ¿Cambios recientes en la configuración?
- ¿Despliegues recientes?
- Estado de dependencias externas (OCS, DNS)?
- Mensajes de error en los registros?

**3. Acciones Inmediatas**:

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

**4. Investigación**:

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

**5. Resolución**:

- Aplicar solución
- Probar en simulador
- Desplegar en producción
- Monitorear para mejoras

**6. Post-Incidente**:

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

# Incidentes Comunes

**Alto Respaldo de Cola**:

1. Verifique la tasa de éxito de entrega
2. Verifique que los frontends estén conectados y consultando
3. Verifique el rendimiento de la base de datos
4. Revise Prometheus en busca de cuellos de botella
5. Considere aumentar el tamaño/intervalo de lote

**Fallos de Enrutamiento**:

1. Revise la configuración de enrutamiento
2. Pruebe en el simulador de enrutamiento
3. Verifique si faltan rutas
4. Verifique que exista una ruta catch-all
5. Revise los registros de eventos para razones de fallo

**Desconexiones de Frontend**:

1. Verifique el estado del sistema del frontend
2. Verifique la conectividad de red
3. Revise los registros del frontend
4. Pruebe el registro manual de la API
5. Verifique las reglas de firewall

**Procesamiento de Mensajes Lento**:

1. Verifique el rendimiento de las consultas de la base de datos
2. Revise la configuración del trabajador por lotes
3. Verifique recursos adecuados (CPU/Memoria)
4. Verifique retrasos en búsquedas de ENUM
5. Revise el rendimiento del sistema de carga

Para procedimientos detallados de solución de problemas, consulte la Guía de Solución de Problemas.

# Enrutamiento de SMS

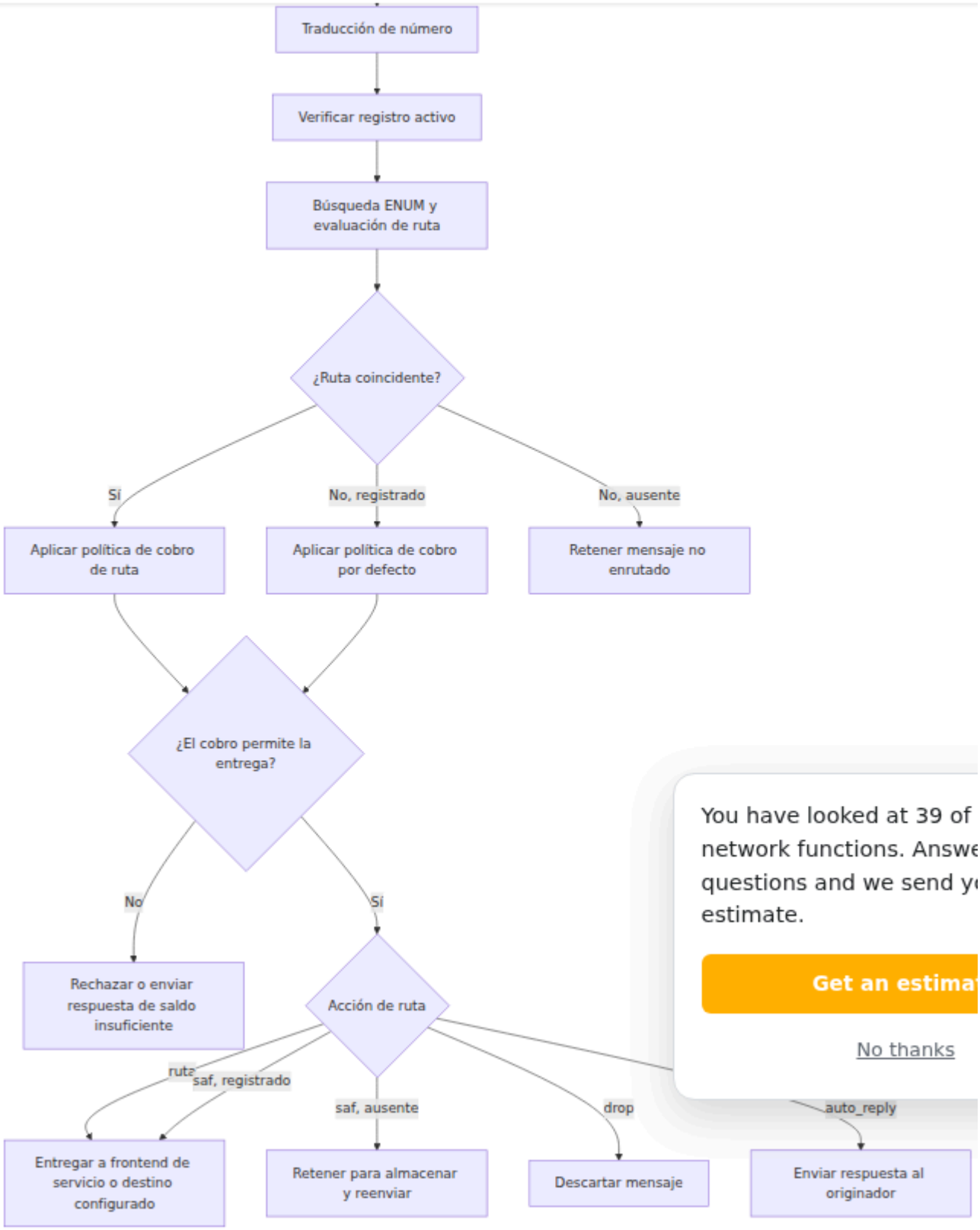
← Volver al Índice de Documentación

## Descripción General

El motor de enrutamiento SMS-C determina a dónde va cada mensaje después de ingresar al sistema. Mnesia almacena las rutas. Puedes modificarlas en tiempo de ejecución a través de la API o del panel de control. Las rutas admiten coincidencias de expresiones regulares, orden de prioridad, balanceo de carga ponderado, coincidencia de resultados ENUM (DNS NAPTR) y coincidencia basada en HSS de red interna/externa.

## Pipeline de Procesamiento de Mensajes

Cada mensaje sigue este pipeline desde la presentación hasta el enrutamiento:




You have looked at 39 of
network functions. Answe
questions and we send y
estimate.
Get an estima
No thanks

El estado de red interna/externa es un criterio de coincidencia (`originating_network` / `terminating_network`). Consulta Coincidencia de Red Interna / Externa.

# Política y Orden de Entrega

1. Verificar si el suscriptor de destino tiene un registro activo.
2. Evaluar las rutas habilitadas y seleccionar la ruta coincidente. Registrar los resultados de coincidencia en el historial del mensaje.
3. Aplicar la política de cobro de la ruta seleccionada antes de la entrega. Si ninguna ruta coincide con un suscriptor registrado, aplicar la política de cobro por defecto.
4. Ejecutar la acción de la ruta. Para `route` y `saf`, un registro activo selecciona el frontend de servicio. Un mensaje `saf` espera si el suscriptor está ausente. Las acciones `drop` y `auto_reply` también se aplican a suscriptores registrados.

# Cobro y Eventos de Mensajes

Un registro activo no elude el cobro. La política `charged` tiene estos valores:

| Valor | Comportamiento |
|---|---|
| `yes` | Solicitar autorización OCS para el remitente. |
| `no` | Omitir la solicitud OCS. |
| `default` | Usar la configuración `default_charging_enabled`. |

Los nuevos mensajes sin un SMSC de destino tienen el estado `routing` hasta que finaliza la evaluación de la ruta y el cobro. Los frontends no pueden recuperarlos durante este paso. Las actualizaciones de registro no pueden liberarlos anticipadamente. Las respuestas automáticas y las presentaciones con un SMSC de destino explícito mantienen su comportamiento de entrega existente.

El historial de mensajes registra estos eventos de cobro:

| Evento | Significado |
| --- | --- |
| `charging_requested` | El SMSC solicita autorización OCS. |
| `charging_succeeded` | El OCS autoriza el cobro. |
| `charging_failed` | El OCS rechaza la solicitud o devuelve un error. La descripción indica si la entrega continúa. |
| `charging_skipped` | La política seleccionada desactiva el cobro. |

El crédito insuficiente rechaza el mensaje o envía la respuesta configurada. Un error de OCS sigue la configuración `on_charging_error`. Con `allow`, la entrega puede continuar después de un error. El evento registra esto como un fallo con entrega permitida, en lugar de un cobro exitoso.

# Modelo de Datos de Ruta

| Campo | Tipo | Descripción |
|---|---|---|
| `route_id` | Entero | Identificador único autogenerado |
| `calling_regex` | Cadena | Expresión regular para el número de origen (`nil` = comodín) |
| `called_regex` | Cadena | Expresión regular para el número de destino (`nil` = comodín) |
| `source_smsc` | Cadena | Nombre del SMSC de origen (`nil` = comodín) |
| `dest_smsc` | Cadena | SMSC de destino. Requerido para `route`; opcional para `saf`; no utilizado por `drop`/`auto_reply` |
| `source_type` | Cadena | Filtrar por tipo de origen: `ims`, `circuit_switched`, `smpp`, o `nil` |
| `enum_domain` | Cadena | Etiqueta de dominio ENUM para la ruta (`nil` = ninguno). Almacenado en la ruta y añade especificidad, pero la consulta DNS real utiliza la configuración global `enum_domains` - consulta Búsqueda ENUM |
| `enum_result_domain` | Cadena | Criterio de coincidencia: la ruta coincide solo si un resultado ENUM NAPTR se resuelve en este dominio (`nil` = comodín) |
| `action` | Cadena | Acción de ruta: `route`, `drop`, `auto_reply`, `saf` |

| Campo | Tipo | Descripción |
|---|---|---|
| `auto_reply_message` | Cadena | Texto de respuesta (requerido cuando la acción es `auto_reply`) |
| `charged` | Cadena | Cobro: `yes`, `no`, `default` |
| `weight` | Entero | Peso de balanceo de carga (1–100) |
| `priority` | Entero | Prioridad (1–255, menor = mayor prioridad) |
| `description` | Cadena | Etiqueta legible por humanos |
| `enabled` | Booleano | Interruptor activo/inactivo |
| `originating_network` | Átomo | Criterio de coincidencia en el estado HSS del MSISDN de origen: `:any` \| `:on_net` \| `:off_net` (por defecto `:any`) |
| `terminating_network` | Átomo | Criterio de coincidencia en el estado HSS del MSISDN de destino: `:any` \| `:on_net` \| `:off_net` (por defecto `:any`) |

# Acciones de Ruta

El campo `action` tiene exactamente cuatro valores válidos: `:route`, `:drop`, `:auto_reply`, `:saf`.

## `route` (Por Defecto)

Entrega el mensaje al `dest_smsc`. Esta es la acción de enrutamiento estándar.

```
{
  "action": "route",
  "called_regex": "^61",
  "dest_smsc": "australia-gw",
  "priority": 100,
  "description": "Rutar números australianos al gateway AU"
}
```

## `drop`

Descarta el mensaje. No se requiere `dest_smsc`.

```
{
  "action": "drop",
  "called_regex": "^900",
  "priority": 5,
  "description": "Bloquear números de tarifa premium"
}
```

## `auto_reply`

Envía `auto_reply_message` de vuelta al originador. El mensaje original se marca como respondido automáticamente.

```
{
  "action": "auto_reply",
  "called_regex": "^911$",
  "auto_reply_message": "Emergencia: SMS a 911 no es soportado.
Por favor, marque 911.",
  "priority": 1,
  "description": "Respuesta automática 911"
}
```

## `saf`

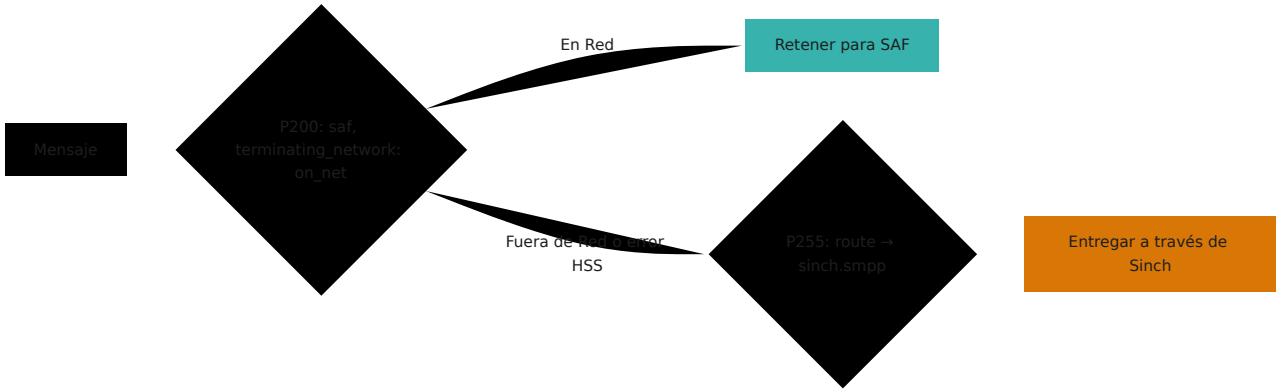
Almacenar y reenviar. El mensaje se retiene y se entrega más tarde, cuando el suscriptor de destino se registre. `dest_smsc` es opcional (si se establece, se

utiliza cuando el mensaje retenido se entrega finalmente).

`saf` en sí mismo es **incondicional** - no realiza una consulta HSS. Para retener *solo* a suscriptores en red para almacenar y reenviar, combina `saf` con un criterio de coincidencia `terminating_network: :on_net` (consulta a continuación), y proporciona una ruta de respaldo fuera de red.

```
{
  "action": "saf",
  "terminating_network": "on_net",
  "priority": 200,
  "description": "Retener suscriptores en red para almacenar y
reenviar"
}
```

**Configuración Típica: Retención en Red + Respaldo Fuera de Red**




La ruta P200 solo *coincide* cuando se confirma que el destino está en red, por lo que los suscriptores en red se retienen para SAF. Los números fuera de red (y errores HSS - fallo cerrado) fallan esa ruta y pasan a la ruta de respaldo P255. Esto asegura que los suscriptores en red nunca se filtren al gateway fuera de red.

# Búsqueda ENUM

ENUM permite que las decisiones de enrutamiento dependan de una búsqueda DNS E.164 NAPTR del número de destino. Esto es típico para la portabilidad de números o el enrutamiento entre operadores, donde DNS (no tu HSS) es la autoridad sobre a qué operador pertenece un número.

## Cómo funciona

1. **Habilitado globalmente.** ENUM solo se ejecuta cuando `enum_enabled: true` y `enum_domains` no está vacío. Es una única búsqueda realizada **una vez por mensaje, antes de que se evalúe la tabla de rutas** - no por ruta.
2. **Consulta.** El número llamado se convierte al formato ENUM inverso (`+15551234` → `4.3.2.1.5.5.5.1.e164.arpa`) y se emite una consulta NAPTR. Cada dominio configurado en `enum_domains` se prueba **en orden**, deteniéndose en el primero que devuelve registros.
3. **Extracción de dominio de resultado.** Cada registro NAPTR devuelto se analiza y reduce a un `result_domain` - las últimas dos etiquetas del campo `replacement`, o del host en una reescritura `sip:user@host` (`sip:+15551234@ims.example.com` → `example.com`).
4. **Coincidencia.** Una ruta con `enum_result_domain` configurado coincide solo si **cualquiera** de los registros NAPTR devueltos tiene un `result_domain` que iguala exactamente esa cadena. `enum_result_domain: nil` es un comodín (siempre coincide, incluso cuando no se realizó ninguna búsqueda ENUM).
5. **Caché.** Los resultados se almacenan en ETS durante 15 minutos por `(número, dominio)`. Los fallos también se almacenan, para evitar golpear DNS en repeticiones fallidas.

## Derivando en red/fuera de red de ENUM

ENUM no devuelve un estado booleano de en red/fuera de red - devuelve un dominio. Expresas "en red" apuntando el `enum_result_domain` de una ruta a tu propio dominio IMS/operador, y dando a todo lo demás una ruta de respaldo. Contrasta esto con la coincidencia HSS (a continuación), que devuelve una respuesta explícita de en red/fuera de red.

## Lo que ENUM *no* hace

- El resultado NAPTR bruto (`enum_result`) se utiliza **solo** como clave de coincidencia de ruta. **No** se utiliza para reescribir la URI de destino o elegir un siguiente salto - a dónde va un mensaje coincidente es completamente

`route.dest_smsc`. El pipeline de entrega en vivo descarta el resultado ENUM bruto después del enrutamiento.

- `enum_domain` en una ruta **no** se utiliza para limitar la consulta DNS (la consulta siempre utiliza la lista global `enum_domains`) y **no** es un criterio de coincidencia. Solo contribuye a puntuación de especificidad.

### Configuración

El comportamiento DNS de ENUM (`enum_enabled`, `enum_domains`, `enum_dns_servers`, `enum_timeout`) se configura globalmente - consulta CONFIGURATION.md → Configuración ENUM.

## Coincidencia de Red Interna / Externa

Una **consulta Diameter Sh al HSS** responde si un destino (o originador) está en red. Las rutas presentan esto como dos criterios de coincidencia. Estos **no** son banderas booleanas y **no** son una acción de ruta. Son criterios de valor enum que el enrutador evalúa junto con las verificaciones de regex/SMSC/tipo. Una ruta solo coincide si satisface sus criterios de red.

### `originating_network` / `terminating_network`

Cada uno toma uno de tres valores:

| Valor | Significado |
| --- | --- |
| `:any` | Sin verificación HSS; coincide tanto en red como fuera de red (por defecto) |
| `:on_net` | La ruta coincide solo si el MSISDN está confirmado **en red** (HSS devuelve éxito, código de resultado 2001) |
| `:off_net` | La ruta coincide solo si el MSISDN está confirmado **fuera de red** (HSS devuelve *Usuario Desconocido*, código de resultado 5001) |

- `originating_network` verifica el número **llamante**; `terminating_network` verifica el número **llamado**.
- **Casos de uso**: restringir una ruta para entregar solo a suscriptores en red (`terminating_network: :on_net`); restringir un enlace SMPP o un operador externo para aceptar solo originadores en red (`originating_network: :on_net`); enviar un respaldo solo fuera de red a través de un gateway externo (`terminating_network: :off_net`).

## Comportamiento de fallo cerrado

Cualquier valor diferente de `:any` requiere una respuesta definitiva del HSS. Si la respuesta no es definitiva, la ruta **no coincide** (y el motor intenta la siguiente ruta):

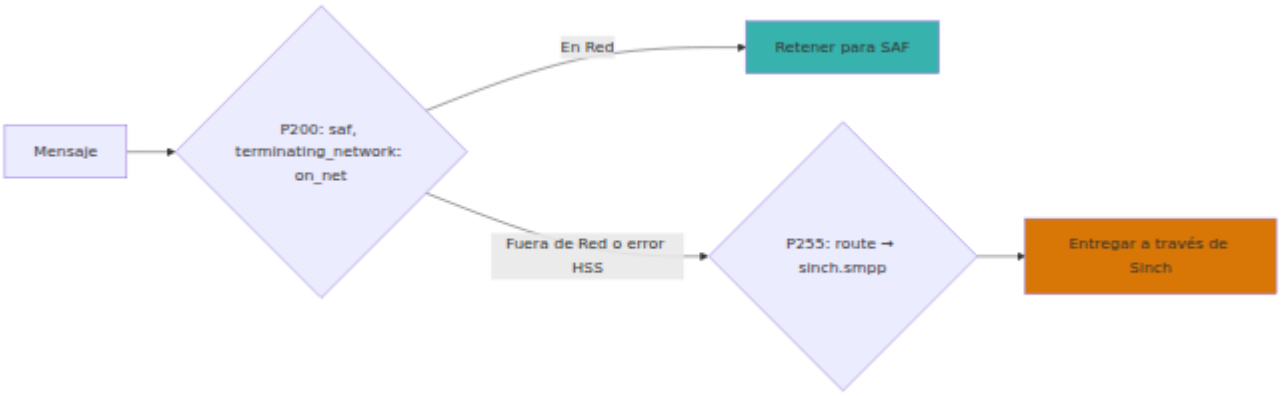
- Diameter deshabilitado → la ruta no coincide
- HSS inalcanzable / tiempo de espera / error de transporte → la ruta no coincide
- HSS devuelve un error diferente de *Usuario Desconocido* → la ruta no coincide

Ten en cuenta que incluso `:off_net` falla cerrado: coincide solo en un *Usuario Desconocido* explícito (5001), no en un fallo de transporte. Esto evita filtrar tráfico a un gateway fuera de red cuando el HSS simplemente está caído.

# Caché de Resultados HSS

Las búsquedas HSS son costosas (un viaje de ida y vuelta Diameter). El sistema almacena en caché los resultados por MSISDN para evitar consultas redundantes.

- **TTL de caché**: 2 horas (configurable a través de `hss_cache_ttl_seconds`)
- **Consulta perezosa**: el HSS se consulta solo cuando una ruta que se está evaluando tiene `originating_network` o `terminating_network` configurados como `:on_net` o `:off_net`. Las rutas que dejan ambos en `:any` nunca desencadenan una consulta.
- **Compartido entre rutas**: si múltiples rutas necesitan el mismo estado de MSISDN, solo ocurre una consulta HSS
- **Reenrutamiento forzado**: la caché se omite (se realiza una nueva consulta) cuando un mensaje se reenvía explícitamente




# Registro de Evaluación de Rutas

Cada ruta evaluada para un mensaje se registra como un evento `route_evaluated` en ese mensaje. El registro incluye:

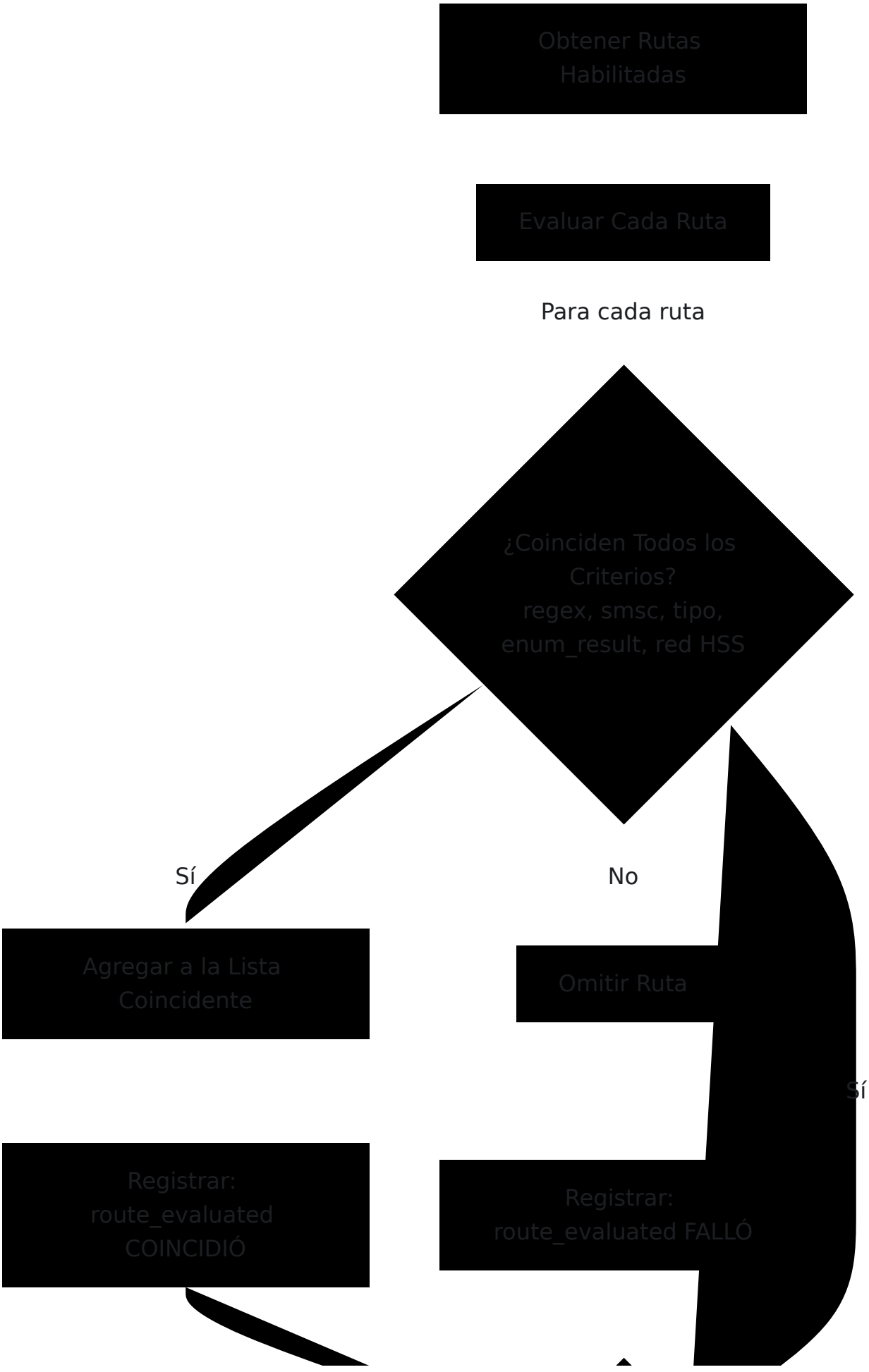
- ID de ruta, acción y prioridad
- Estado de aprobación/fallo para cada criterio de coincidencia (incluyendo `enum_result` y las verificaciones HSS de `orig_network`/`term_network`)
- Detalle de lo que se comparó (patrón regex vs número, nombre de SMSC vs origen, red requerida vs estado HSS real, etc.)

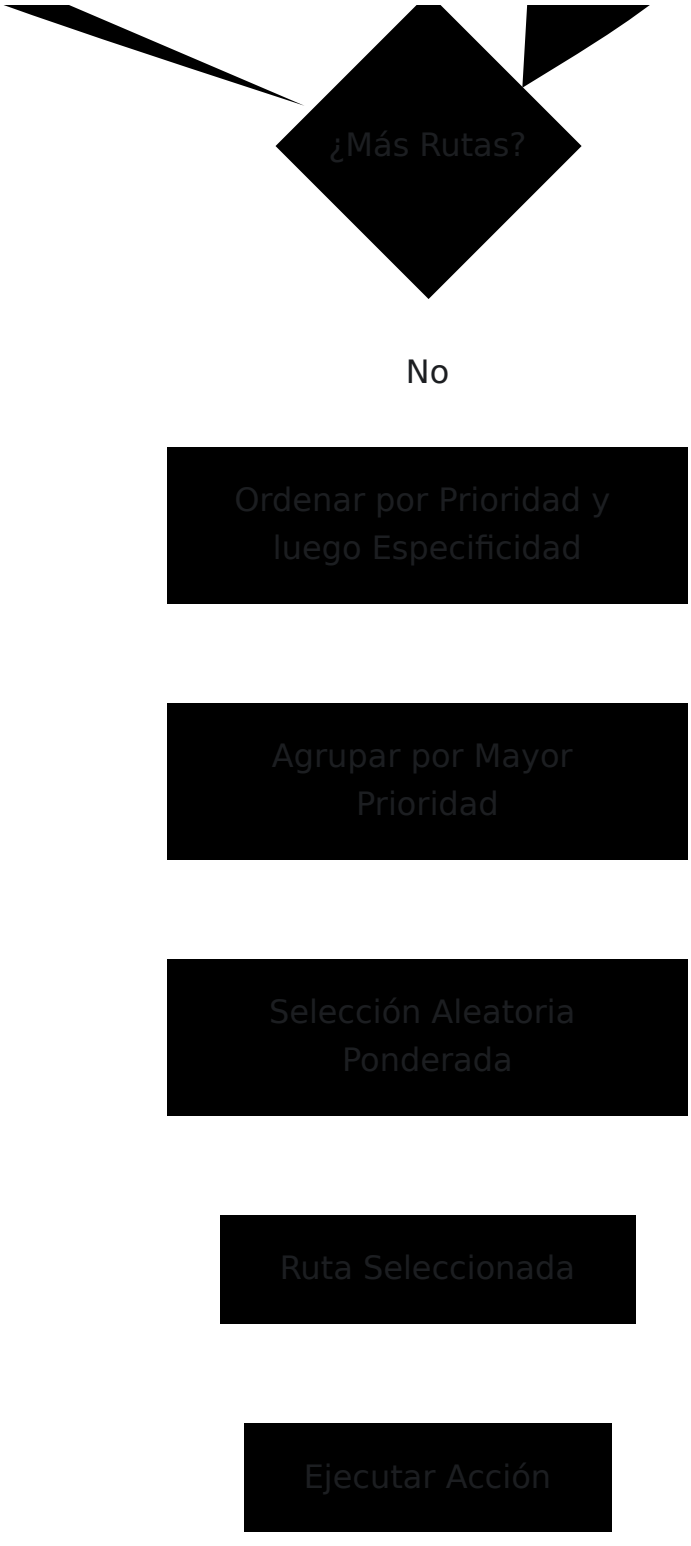
**Ejemplo de entradas de registro de eventos:**

```
Ruta #4 [saf] COINCIDIÓ: P200 "Retener en red" |
  calling_regex: ✓ (comodín), called_regex: ✓ (comodín),
  source_smsc: ✓ (comodín), source_type: ✓ (comodín),
  enum_result: ✓ (comodín),
  term_network: ✓ (la terminación requiere en_red, +15551234 está
en red)

Ruta #3 [route] FALLÓ: P255 "Respaldo Sinch" → sinch.smpp |
  calling_regex: ✓ (comodín), called_regex: ✓ (comodín),
  source_smsc: ✓ (comodín), source_type: ✗ (ims vs
circuit_switched),
  enum_result: ✓ (comodín), term_network: ✓ (cualquiera)
```

Estos eventos son visibles en el registro de actividad de mensajes del panel de control y a través de la API de Eventos.

# Algoritmo de Evaluación de Rutas


Obtener Rutas Habilitadas
Evaluar Cada Ruta
Para cada ruta
¿Coinciden Todos los Criterios? regex, smsc, tipo, enum_result, red HSS
Sí
No
Agregar a la Lista Coincidente
Omitir Ruta
Sí
Registrar: route_evaluated COINCIDIÓ
Registrar: route_evaluated FALLÓ




La verificación HSS de en red/fuera de red es parte de **la coincidencia de criterios** (una ruta que falla simplemente no coincide) - no hay un paso HSS separado posterior a la selección.

## Puntuación de Especificidad

Cuando múltiples rutas coinciden, la especificidad determina cuál se prefiere dentro del mismo nivel de prioridad:

| Criterio | Puntos |
|---|---|
| Longitud del patrón `called_regex` | longitud × 100 |
| Longitud del patrón `calling_regex` | longitud × 50 |
| `source_smsc` especificado | +25 |
| `enum_result_domain` especificado | +15 |
| `source_type` especificado | +10 |
| `enum_domain` especificado | +5 |

Patrones más largos y específicos siempre ganan sobre comodines.

# Configuración

## Cargando Rutas desde la Configuración

Las rutas definidas en `config/runtime.exs` bajo `:sms_routes` se cargan en el primer inicio cuando la tabla de enrutamiento está vacía. Reinicios posteriores preservan las rutas añadidas a través de la API.

```
config :sms_c, :sms_routes, [
  %{
    called_regex: ~r/^15555/,
    dest_smsc: "example-gw",
    action: :route,
    priority: 100,
    description: "Enrutamiento nacional de ejemplo"
  },
  %{
    action: :saf,
    terminating_network: :on_net,
    priority: 200,
    description: "Retener suscriptores en red para almacenar y
reenviar"
  },
  %{
    action: :auto_reply,
    called_regex: ~r/^911$/,
    auto_reply_message: "Emergencia: Por favor, marque 911.",
    priority: 1,
    description: "Respuesta automática 911"
  }
]
```

## Parámetros de Configuración de Ruta

| Parámetro | Tipo | Requerido | Por Defecto | Des |
|---|---|---|---|---|
| `action` | Átomo/Cadena | No | `:route` | Acción c `:route` `:auto_r` |
| `calling_regex` | Regex | No | `nil` | Patrón c regular número `nil` coi todos |
| `called_regex` | Regex | No | `nil` | Patrón c regular número `nil` coi todos |
| `source_smsc` | Cadena | No | `nil` | Filtrar p de SMS `nil` coi todos |
| `dest_smsc` | Cadena | Condicional | `nil` | Requeri `route`. para `sa` utilizad `drop`/`a` |
| `source_type` | Átomo | No | `nil` | Filtrar p origen: `:circu` `:smpp` |

| Parámetro | Tipo | Requerido | Por Defecto | Des |
|---|---|---|---|---|
| `enum_domain` | Cadena | No | `nil` | Etiqueta ENUM p (añade especifi limita la |
| `enum_result_domain` | Cadena | No | `nil` | Coincidi resultad NAPTR s en este `nil` coi todos |
| `auto_reply_message` | Cadena | Condicional | `nil` | Requeri acción e `:auto_r` |
| `charged` | Átomo | No | `:default` | Política `:yes`, `:` `:defaul` |
| `weight` | Entero | No | `100` | Peso de carga (1 |
| `priority` | Entero | No | `100` | Priorida evaluac menor = |
| `description` | Cadena | No | `""` | Descrip por hum |
| `enabled` | Booleano | No | `true` | Si la rut |

| Parámetro | Tipo | Requerido | Por Defecto | Des |
|---|---|---|---|---|
| `originating_network` | Átomo | No | `:any` | Requisit HSS del origen: `:on_net` |
| `terminating_network` | Átomo | No | `:any` | Requisit HSS del destino: `:on_net` |

## Configuración de Caché HSS

| Parámetro | Tipo | Por Defecto | Descripción |
|---|---|---|---|
| `hss_cache_ttl_seconds` | Entero | `7200` | Tiempo de vida para los resultados de búsqueda HSS en caché (segundos). Establecido en `runtime.exs` |

# API

## Crear Ruta

```
POST /api/routes
```

```
{
  "action": "saf",
  "terminating_network": "on_net",
  "priority": 200,
  "description": "Retener suscriptores en red para almacenar y
reenviar"
}
```

## Listar Rutas

```
GET /api/routes
```

## Actualizar Ruta

```
PUT /api/routes/:id
```

## Eliminar Ruta

```
DELETE /api/routes/:id
```

## Importar/Exportar

```
POST /api/routes/import
GET /api/routes/export
```

La importación admite modos `merge` (agregar a existente) y `replace` (borrar y reemplazar todo).

# Solución de Problemas

## No se Encontró Ruta

**Síntomas**: El mensaje permanece pendiente con `dest_smsc: null`, el registro de eventos muestra `sms_routing_failed`

**Causas posibles**:

- No hay rutas habilitadas que coincidan con los criterios del mensaje
- Los patrones de expresiones regulares de la ruta no coinciden con el formato del número (por ejemplo, falta el código de país)
- Cada ruta candidata tiene un requisito de `terminating_network`/`originating_network` que el HSS no satisfizo (fuera de red, o un error HSS bajo fallo cerrado), y no existe una ruta de respaldo `:any`

**Resolución**:

1. Verifica el registro de eventos del mensaje - cada ruta evaluada se registra con aprobación/fallo por criterio (incluyendo la verificación de red HSS)
2. Usa el simulador de enrutamiento en el panel de control para probar con los mismos números
3. Asegúrate de que exista una ruta de respaldo general (`originating_network: :any`, `terminating_network: :any`) en un número de prioridad alto (por ejemplo, 255)

## Suscriptor en Red Enrutado Fuera de Red

**Síntomas**: Los mensajes a suscriptores en red se entregan a través de un gateway externo en lugar de ser retenidos/entregados localmente

**Causas posibles**:

- No hay ruta con `terminating_network: :on_net` configurada para capturar destinos en red

- La ruta en red tiene un *número* de prioridad más alto (menor precedencia) que la ruta de respaldo fuera de red
- Fallo en la búsqueda HSS (verifica `hss_dip_error` / telemetría Sh) - bajo fallo cerrado, la ruta en red no coincide y el tráfico pasa a la ruta de respaldo
- Diameter no habilitado en la configuración

**Resolución**:

1. Agrega una ruta con `terminating_network: :on_net` en un número de prioridad más bajo que la ruta de respaldo fuera de red
2. Verifica que Diameter esté habilitado y que los pares HSS estén conectados a través del panel de control
3. Verifica el estado de la caché HSS y los resultados recientes de consultas en el registro de eventos

# Caché HSS Obsoleta

**Síntomas**: Suscriptor portado o provisionado pero aún enrutando basado en el antiguo estado de en red/fuera de red

**Resolución**:

1. Reenruta el mensaje a través de la API - el reenrutamiento forzado omite la caché HSS
2. Espera a que expire el TTL de la caché (por defecto 2 horas)
3. Reduce `hss_cache_ttl_seconds` si se necesitan actualizaciones más rápidas

# Guía de Solución de Problemas de SMS-C

**← Volver al Índice de Documentación**

Guía completa para diagnosticar y resolver problemas comunes de SMS-C.

## Tabla de Contenidos


- Herramientas de Diagnóstico
- Problemas de Entrega de Mensajes
- Problemas de Enrutamiento
- Problemas de Rendimiento
- Problemas de Base de Datos
- Problemas de Conexión del Frontend
- Problemas de Carga/Facturación
- Problemas de Búsqueda ENUM
- Problemas de Clúster
- Problemas de API
- Problemas de Interfaz Web
- Problemas de Recursos del Sistema

# Herramientas de Diagnóstico

## Verificación Rápida de Salud

```
# 1. Verificar estado de la API
curl https://api.example.com:8443/api/status

# 2. Verificar el endpoint de métricas de Prometheus
curl https://api.example.com:9568/metrics | grep sms_c

# 3. Verificar los registros de la aplicación
tail -f /var/log/sms_c/application.log

# 4. Verificar el estado del proceso
systemctl status omnimessage

# 5. Verificar la conectividad de 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 "en cola"/"retroceso"
- Alto conteo de mensajes no entregados
- 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: "en cola" o "retroceso"
- 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 los registros del sistema del frontend:

- ¿Está el frontend consultando `/api/messages`?

- ¿Está el frontend enviando correctamente el encabezado `smsc`?

**Soluciones**:

**Sin Frontends Conectados**:

```
# Verificar estado del sistema del frontend
systemctl status frontend_service

# Verificar que el frontend pueda alcanzar la API
curl -k https://api.example.com:8443/api/status

# Registrar manualmente el frontend
curl -X POST https://api.example.com:8443/api/frontends/register \
  -H "Content-Type: application/json" \
  -d '{
    "frontend_name": "test_gateway",
    "frontend_type": "smpp",
    "ip_address": "10.0.1.50"
  }'
```

**Mensajes Enrutados al SMSC Incorrecto**:

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

**Mensajes Programados para el Futuro**:

- Verificar la marca de tiempo de `deliver_after`
- Reiniciar si es necesario:

```
curl -X PATCH https://api.example.com:8443/api/messages/12345 \
  -H "Content-Type: application/json" \
  -d '{"deliver_after": "2025-10-30T12:00:00Z"}'
```

# Mensajes Fallando con Reintentos

**Síntomas**:

- Contador de `delivery_attempts` aumentando
- Mensajes con alto conteo de intentos (> 3)
- Retrasos de retroceso exponencial

**Pasos de Diagnóstico**:

1. **Verificar Registro de Eventos**:

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

Buscar:

- Eventos de fallo de entrega
- Descripciones de errores
- Marcas de tiempo de reintentos

2. **Verificar Registros del Frontend**:

- ¿Por qué está fallando el frontend al 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**:

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

# Reiniciar contador de reintentos
curl -X PATCH https://api.example.com:8443/api/messages/12345 \
  -H "Content-Type: application/json" \
  -d '{"delivery_attempts": 0, "deliver_after": "2025-10-
30T12:00:00Z"}'
```

**Número de Destino Inválido**:

- Verificar formato del número
- Revisar reglas de traducción de números
- Eliminar mensaje si es realmente inválido

# Mensajes de Carta Muerta

**Síntomas**:

- `deadletter: true` en el mensaje
- Mensajes pasados el tiempo de expiración
- Estado aún "en cola"/"retroceso" (las entregas fallidas no se convierten en cartas muertas; dejan de ser atendidas una vez pasados `expires`)

**Pasos de Diagnóstico**:

1. **Encontrar Mensajes de Carta Muerta**:

Acceder a la Interfaz Web: `/message_queue`

- Filtrar por estado expirado
- Verificar marcas de tiempo de expiración

2. **Verificar Por Qué Expiró**:

- Revisar registro de eventos
- Verificar historial de intentos de entrega

- Verificar que el enrutamiento fue exitoso

**Soluciones**:

**Extender Expiración**:

```
# Agregar 24 horas a la expiración
curl -X PATCH https://api.example.com:8443/api/messages/12345 \
  -H "Content-Type: application/json" \
  -d '{"expires": "2025-10-31T12:00:00Z", "deadletter": false}'
```

# Problemas de Enrutamiento

## No Se Encontró Ruta

**Síntomas**:

- Error: `no_route_found`
- Métrica `sms_c_routing_failed_count` aumentando
- Registro de eventos muestra "routing_failed"

**Pasos de Diagnóstico**:

1. **Verificar Que Existan Rutas**:

Acceder a la Interfaz Web: `/sms_routing`

- Verificar que las rutas estén configuradas
- Comprobar que al menos una ruta esté habilitada

2. **Probar Enrutamiento**:

Acceder a la Interfaz Web: `/simulator`

- Ingresar parámetros de mensaje (número de llamada, número llamado, SMSC de origen)
- Revisar resultados de evaluación
- Verificar por qué las rutas no coincidieron

3. **Verificar Criterios de Ruta**:

- ¿Se requieren coincidencias de prefijo?
- ¿El filtro de SMSC de origen es demasiado restrictivo?
- ¿Todas las rutas deshabilitadas?

**Soluciones**:

**Sin Rutas Configuradas**:

Agregar ruta de captura:

```
Prefijo de Llamada: (vacío)
Prefijo Llamado: (vacío)
SMSC de Origen: (vacío)
SMSC de Destino: default_gateway
Prioridad: 255
Peso: 100
Habilitado: ✓
Descripción: Ruta predeterminada de captura
```

**Rutas Demasiado Específicas**:

Agregar ruta más amplia:

```
Prefijo Llamado: +
SMSC de Destino: international_gateway
Prioridad: 200
Peso: 100
Habilitado: ✓
Descripción: Captura internacional
```

**Todas las Rutas Deshabilitadas**:

- Habilitar rutas apropiadas a través de la Interfaz Web
- Verificar que la configuración no haya deshabilitado accidentalmente rutas

# Ruta Incorrecta Seleccionada

**Síntomas**:

- Mensajes enrutados a un destino inesperado
- Puerta de enlace incorrecta recibiendo tráfico
- Balanceo de carga no distribuyendo como se esperaba

**Pasos de Diagnóstico**:

1. **Usar Simulador de Enrutamiento**:

Acceder a la Interfaz Web: `/simulator`

- Probar con parámetros de mensaje reales
- Revisar sección "Todas las Coincidencias"
- Verificar puntuaciones 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 la distribución del balanceo de carga:

```
Primaria (70%):
  Peso: 70

Respaldo (30%):
  Peso: 30
```

**Agregar Ruta Más Específica**:

Anular ruta general para un 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 faltante de evento de respuesta automática

**Pasos de Diagnóstico**:

1. **Verificar Configuración de Ruta**:

- `action: :auto_reply`
- `auto_reply_message` contiene texto
- Ruta está habilitada
- Ruta coincide con los criterios del 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 (debe ser más alta que las rutas normales)
- Verificar estado habilitado

**Mensaje No Establecido**:

Editar ruta, agregar mensaje:

```
Respuesta Automática: ✓
Mensaje de Respuesta Automática: "Gracias por su mensaje.
Responderemos pronto."
```

**Prioridad Incorrecta**:

Las rutas de respuesta automática deben tener alta prioridad (número bajo):

```
Ruta de Respuesta Automática:
  Prioridad: 10

Ruta Normal:
  Prioridad: 50
```

# Problemas de Rendimiento

## Alta Latencia en el Procesamiento de Mensajes

**Síntomas**:

- `sms_c_message_processing_stop_duration` p95 > 1000ms
- Respuestas 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 pool 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 pool 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 pool de conexiones
ecto_pools_queue_time
```

**Soluciones**:

**Cuello de Botella de Base de Datos**:

Aumentar tamaño del pool:

```
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 Cola de Respaldo

**Síntomas**:

- `sms_c_queue_size_pending` aumentando
- Edad del mensaje más antiguo aumentando

- El procesamiento no puede seguir el ritmo de la tasa de entrada

**Pasos de Diagnóstico**:

1. **Verificar Tasa de Entrada vs 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 (debe ser de 5-10 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: "imposible conectar a la base de datos"
- API devolviendo errores 500
- La aplicación no se inicia

**Pasos de Diagnóstico**:

1. **Verificar Estado de la Base de Datos SQL CDR**:

```
# MySQL/MariaDB
systemctl status mysql

# PostgreSQL
systemctl status postgresql

# Probar conectividad (MySQL/MariaDB)
mysql -u sms_user -p -h db.example.com -e "SELECT 1"

# Probar conectividad (PostgreSQL)
psql -U sms_user -h db.example.com -d sms_c_prod -c "SELECT 1"
```

2. **Verificar Red**:

```
# Hacer ping al host de la base de datos
ping db.example.com

# Verificar conectividad de puerto (MySQL/MariaDB: 3306,
PostgreSQL: 5432)
telnet db.example.com 3306
# o
telnet db.example.com 5432
```

3. **Verificar Credenciales**:

```
# Verificar variables de entorno
echo $DB_USERNAME
echo $DB_HOSTNAME
echo $DB_PORT

# Intentar conexión manual con las mismas credenciales
(MySQL/MariaDB)
mysql -u $DB_USERNAME -p$DB_PASSWORD -h $DB_HOSTNAME

# Para PostgreSQL:
# psql -U $DB_USERNAME -h $DB_HOSTNAME -d sms_c_prod
```

**Soluciones**:

**Base de Datos Caída**:

```
# Iniciar base de datos (MySQL/MariaDB)
systemctl start mysql

# Iniciar base de datos (PostgreSQL)
systemctl start postgresql
```

**Credenciales Incorrectas**:

Actualizar configuración:

```
export DB_USERNAME=correct_user
export DB_PASSWORD=correct_password

# Reiniciar aplicación
systemctl restart omnimessage
```

**Problema de Red**:

- Verificar reglas de firewall
- Verificar grupos de seguridad (nube)
- Comprobar conectividad VPN/red

**Pool de Conexiones Exhausto**:

Aumentar tamaño del pool:

```
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 pool 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**:

```sql
-- 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**:

```sql
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**:

```sql
-- MySQL/MariaDB
OPTIMIZE TABLE message_queues;
OPTIMIZE TABLE frontend_registrations;

-- PostgreSQL
-- VACUUM ANALYZE message_queues;
-- VACUUM ANALYZE frontend_registrations;
```

**Demasiados Datos**:

Limpiar registros antiguos:

```sql
-- 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 &amp;#123;&amp;#125; + | sort -rh
| head -20

# Encontrar archivos más grandes (PostgreSQL)
find /var/lib/postgresql -type f -exec du -h &amp;#123;&amp;#125; + | sort
-rh | head -20

# Verificar archivos de registro
du -sh /var/log/sms_c/*
```

**Soluciones**:

**Limpiar Datos Antiguos**:

```sql
-- Eliminar mensajes antiguos
DELETE FROM message_queues
WHERE inserted_at < DATE_SUB(NOW(), INTERVAL 90 DAY)
LIMIT 100000;
```

**Rotar Registros**:

```
# Forzar logrotate
logrotate -f /etc/logrotate.d/sms_c

# Limpiar archivos de registro antiguos
find /var/log/sms_c -name "*.log.*" -mtime +30 -delete
```

**Expandir Disco**:

- Redimensionar volumen (nube)
- Agregar nuevo disco y extender volumen
- Mover datos a disco más grande

# Problemas de Conexión del Frontend

## Frontend No Aparece Como Activo

**Síntomas**:

- Estado del frontend muestra "expirado"
- Frontend no en la lista activa
- Mensajes no se entregan al frontend

**Pasos de Diagnóstico**:

1. **Verificar Registro**:

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

2. **Verificar Registros del Frontend**:

- ¿Está el frontend llamando a `/api/frontends/register`?
- ¿Hay errores en la API?
- Frecuencia de registro (debe 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 '&amp;#123;
    "frontend_name": "uk_gateway",
    "frontend_type": "smpp",
    "ip_address": "10.0.1.50"
  &amp;#125;'
```

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

**Registro Expirando**:

Los frontends expiran después de 90 segundos. Asegúrese 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 alternando 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 comprobaciones de salud del balanceador de carga

**Intervalo de Registro Incorrecto**:

Corregir intervalo:

```
REGISTRATION_INTERVAL = 60  # segundos
```

Charging se realiza a través de **Diameter Ro** (Control de Crédito). Tenga en cuenta que el cobro **falla en abierto**: un error de OCS o una pila de Diameter no configurada registra un fallo pero aún entrega el mensaje. Solo una respuesta definitiva fuera de balance rechaza un mensaje (estado `:balance_rejected`).

**Síntomas**:

- `sms_c_charging_failed_count` en aumento
- El registro de eventos muestra "charging_failed" o "balance_rejected"
- Mensajes marcados con estado `:balance_rejected` (fuera de balance), o advertencias en el registro sobre la pila de Diameter no habilitada

**Pasos de Diagnóstico**:

1. **Verifique el estado del par Diameter Ro** (CER/CEA establecido al DRA/OCS):

```
# En una consola remota
:diameter.service_info(:omnimessage, :connections)
```

2. **Confirme que la pila está habilitada y que la aplicación `:ro` está registrada**:

```
grep -E "diameter_enabled|application_name: :ro"
config/runtime.exs
```

Una línea de registro `Charging requested but the Diameter Ro stack is not enabled/configured` significa que se está omitiendo el cobro (mensajes permitidos) - habilite la pila.

3. **Verifique que el perfil de calificación** coincida con el contexto de calificación configurado en el OCS:

```
grep rating_profile config/runtime.exs
```

**Soluciones**:

**OCS / DRA Inalcanzable**:

- Verifique la conectividad IP y que el DRA tenga este nodo provisionado como un par Diameter (Origin-Host/Realm).
- Confirme que el CCR `Service-Context-Id` (`rating_profile`) coincida con el perfil de calificación SMS del OCS.

**Manejo de fuera de balance** (`config :sms_c, :charging`):

```
config :sms_c, :charging,
  out_of_balance_action: :reject,    # o :auto_reply
  out_of_balance_reply_text: "...",
  rating_profile: "sms@sms-c"
```

**Deshabilitar Cobro Temporalmente**:

```
config :sms_c,
  default_charging_enabled: false
```

Reinicie la aplicación.

**Problemas de Cuenta**:

- Verifique que la cuenta exista en el OCS
- Verifique que la cuenta tenga saldo
- Verifique que los planes de calificación estén configurados

# Cobro Demasiado Lento

**Síntomas**:

- `sms_c_charging_succeeded_duration` p95 > 500ms
- Procesamiento de mensajes lento cuando el cobro está habilitado
- Rápido cuando el cobro está deshabilitado

**Pasos de Diagnóstico**:

1. **Verifique la Latencia de Cobro**:

```
histogram_quantile(0.95, sms_c_charging_succeeded_duration_bucket)
```

2. **Verifique el rendimiento del OCS / DRA** (ida y vuelta de Diameter Ro). Alta `sms_c_charging_succeeded_duration` generalmente apunta a la latencia de calificación del OCS o a un par DRA lento/fluctuante.

3. **Verifique la Latencia de Red hacia el DRA**:

```
# Hacer ping al host DRA / OCS
ping -c 10 dra01.example.com
```

**Soluciones**:

**OCS Lento**:

- Optimizar la configuración del OCS
- Agregar recursos al OCS
- Usar un motor de calificación más rápido

**Latencia de Red**:

- Desplegar OCS más cerca del SMS-C
- Usar ruta de red directa
- Evitar VPN/túneles si es posible

**Tiempo de Espera Demasiado Bajo**:

Aumentar el 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
- El registro de eventos muestra errores ENUM
- Rutas con `enum_result_domain` no coincidiendo

**Pasos de Diagnóstico**:

1. **Verifique la Configuración ENUM**:

```
grep -A 10 "enum_" config/runtime.exs
```

2. **Pruebe la 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. **Verifique el Servidor DNS**:

```
# ¿Es accesible el DNS personalizado?
ping 10.0.1.53

# Probar puerto
nc -zv 10.0.1.53 53
```

**Soluciones**:

**Servidor DNS Inalcanzable**:

Usar DNS alternativo:

```
config :sms_c,
  enum_dns_servers: [
    &#123;"8.8.8.8", 53&#125;,  # Google Public DNS
    &#123;"1.1.1.1", 53&#125;   # Cloudflare DNS
  ]
```

**Dominio ENUM Incorrecto**:

Actualizar dominio:

```
config :sms_c,
  enum_domains: ["e164.arpa"]  # Usar dominio estándar
```

**Tiempo de Espera Demasiado Corto**:

Aumentar el 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 en caché (< 70%)
- Tamaño de caché creciendo sin límites
- Uso de memoria alto

**Pasos de Diagnóstico**:

1. **Verifique las 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. **Verifique el Patrón de Tráfico**:

- ¿Se repiten los números?
- ¿TTL de caché apropiado?

**Soluciones**:

**Baja Tasa de Aciertos** (Esperado):

- Tráfico a números únicos (normal)
- Monitorear pero no alarmar si < 70%

**Caché Creciendo**:

Borrar caché a través de la página de prueba NAPTR o reiniciar la aplicación.

**Alto Uso de Memoria**:

- Esperado con caché grande
- Monitorear la memoria del sistema en general
- Considerar ajuste de TTL

# Problemas de Clúster

## Nodo No Puede Unirse al Clúster

**Síntomas**:

- Nodo único en ejecución
- Consultas de clúster devolviendo solo resultados locales
- Errores de distribución de Erlang

**Pasos de Diagnóstico**:

1. **Verifique los Nombres de Nodo**:

```
# En la consola IEx
Node.self()
# Esperado: :sms@node1.example.com

Node.list()
# Esperado: Lista de otros nodos
```

2. **Verifique la Cookie de Erlang**:

```
# Verifique el archivo de cookie
cat ~/.erlang.cookie

# Verifique que sea la misma en todos los nodos
```

3. **Verifique la Red**:

```
# ¿Pueden los nodos alcanzarse entre sí?
ping node2.example.com

# Verifique puertos
nc -zv node2.example.com 4369
nc -zv node2.example.com 9100-9200
```

**Soluciones**:

**Desajuste de Cookie**:

Establezca la misma cookie en todos los nodos:

```
export ERLANG_COOKIE=same_secret_value_here

# O actualice ~/.erlang.cookie
echo "same_secret_value_here" > ~/.erlang.cookie
chmod 400 ~/.erlang.cookie
```

**Firewall Bloqueando**:

Abra los 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**:

Use direcciones IP en lugar de nombres de host:

```
config :sms_c,
  cluster_nodes: [
    :"sms@10.0.1.10",
    :"sms@10.0.1.11"
  ]
```

# Clúster con División de Cerebro

**Síntomas**:

- Nodos en ejecución pero desconectados
- Datos diferentes en diferentes nodos
- Inconsistencias en Mnesia

**Pasos de Diagnóstico**:

1. **Verifique la Conectividad de Nodos**:

```
# En cada nodo (IEx)
Node.list()
```

2. **Verifique Mnesia**:

```
:mnesia.system_info(:running_db_nodes)
```

**Soluciones**:

**Reconectar Nodos**:

```
# Detener todos los nodos
systemctl stop omnimessage

# Iniciar un nodo primero
systemctl start omnimessage  # En node1

# Espere a que se inicie completamente, luego inicie otros
systemctl start omnimessage  # En node2
systemctl start omnimessage  # En node3
```

**Inconsistencia de Mnesia**:

- Exportar rutas desde el nodo correcto
- Detener todos los nodos
- Eliminar el directorio de Mnesia
- Iniciar nodos
- Importar rutas

# Problemas de API

## API No Responde

**Síntomas**:

- Tiempo de conexión agotado

- Conexión rechazada
- Sin respuesta

**Pasos de Diagnóstico**:

1. **Verifique el Proceso de API**:

```
# ¿Está la aplicación en ejecución?
systemctl status omnimessage

# Verifique los puertos en escucha
netstat -tlnp | grep 8443
```

2. **Verifique el Firewall**:

```
# Verifique iptables
iptables -L -n | grep 8443

# Pruebe la conectividad local
curl -k https://localhost:8443/api/status
```

3. **Verifique la Configuración TLS**:

```
# Verifique que el certificado exista
ls -l priv/cert/server.crt priv/cert/server.key

# Verifique la validez del certificado
openssl x509 -in priv/cert/server.crt -noout -dates
```

**Soluciones**:

**Aplicación No Ejecutándose**:

```
systemctl start omnimessage
```

**Firewall Bloqueando**:

```
# Permitir puerto de API
iptables -A INPUT -p tcp --dport 8443 -j ACCEPT
```

**Problemas de Certificado**:

Genere un nuevo certificado (consulte la Guía de Configuración).

**Puerto Incorrecto**:

Verifique la configuración:

```
grep "port:" config/runtime.exs
```

# API Retornando Errores 500

**Síntomas**:

- Error Interno del Servidor
- Código de estado 500
- Error en los registros

**Pasos de Diagnóstico**:

1. **Verifique los Registros de la Aplicación**:

```
tail -100 /var/log/sms_c/application.log | grep "\[error\]"
```

2. **Verifique la Base de Datos**:

```
mysql -u sms_user -p -e "SELECT 1"
```

3. **Verifique Recursos**:

```
# Memoria
free -h

# CPU
top -b -n 1

# Disco
df -h
```

**Soluciones**:

**Base de Datos No Disponible**:

- Inicie la base de datos
- Corrija la configuración de conexión

**Sin Memoria**:

- Reinicie la aplicación
- Aumente la memoria del sistema
- Verifique si hay fugas de memoria

**Error de Aplicación**:

- Verifique el error específico en los registros
- Corrija el problema de configuración
- Reinicie la aplicación

# Problemas de Interfaz Web

## No se Puede Acceder a la Interfaz Web

**Síntomas**:

- Tiempo de conexión agotado
- 404 No Encontrado
- La página no se carga

**Pasos de Diagnóstico**:

1. **Verifique el Estado de la Aplicación**:

```
systemctl status omnimessage
```

2. **Verifique el Puerto**:

```
netstat -tlnp | grep 80
```

3. **Verifique la URL**:

- ¿Nombre de host correcto?
- ¿Puerto correcto?
- ¿HTTP vs HTTPS?

**Soluciones**:

**Puerto Incorrecto**:

Verifique la configuración:

```
grep "control_panel" config/runtime.exs
```

Acceda en el puerto correcto (predeterminado: 80 o 4000).

**Aplicación No Ejecutándose**:

```
systemctl start omnimessage
```

**Firewall**:

```
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

# LiveView No Se Actualiza

**Síntomas**:

- La página se carga pero no se actualiza
- Los datos están desactualizados
- Errores de WebSocket en la consola del navegador

**Pasos de Diagnóstico**:

1. **Verifique la Consola del Navegador**:

- Abra las Herramientas de Desarrollo (F12)
- Busque errores de WebSocket
- Verifique la pestaña de red para solicitudes fallidas

2. **Verifique la Configuración del Proxy**:

Si usa un proxy inverso, asegúrese de que soporte WebSocket:

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

**Soluciones**:

**WebSocket Bloqueado**:

- Configure el proxy para WebSocket
- Verifique el firewall
- Verifique las extensiones del navegador

**Actualizar Página**:

- Actualización forzada (Ctrl+F5)
- Borrar 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. **Verifique el Proceso**:

```
top -b -n 1 | grep beam.smp
```

2. **Verifique 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**:

**Alto Tráfico**:

- Escalar horizontalmente (agregar nodos)
- Escalar verticalmente (agregar CPU)

**Enrutamiento Ineficiente**:

- Reducir el número de rutas
- Optimizar los criterios de ruta

**Demasiadas Búsquedas ENUM**:

- Verifique la tasa de aciertos en caché
- Considere 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. **Verifique la Memoria**:

```
free -h

ps aux | grep beam.smp
```

2. **Verifique los Tamaños de Caché**:

```
sms_c_enum_cache_size_size
```

**Soluciones**:

**Caché ENUM Demasiado Grande**:

- Borrar caché
- Reducir TTL
- Deshabilitar ENUM si no es necesario

**Cola de Lotes Creciendo**:

```
# Verifique las estadísticas de trabajadores (IEx)
SmsC.Messaging.BatchInsertWorker.stats()
```

Si la cola es grande, vacíela manualmente o reinicie.

**Agregar Memoria**:

- Escalar verticalmente
- Agregar swap (temporal)

**Fuga de Memoria**:

- Reinicie la aplicación
- Informe el 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`