Benchmarks

Este diretório contém benchmarks de desempenho para o sistema SMS-C usando Benchee.

Benchmarks Disponíveis

1. Benchmark de SMS Bruto (`raw_sms_bench.exs`)

Benchmark do endpoint da API `submit_message_raw` usando PDUs de SMS reais.

Características:

- Usa PDUs de SMS reais (adicione seus PDUs à lista `@sample_pdus` no arquivo)
- Desativa a detecção de duplicatas limpando as impressões digitais antes de cada iteração
- Gera relatórios em console e HTML

Uso:

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

Saída: `benchmarks/output/raw_sms_benchmark.html`

2. Benchmark da API de Mensagens (`message_api_bench.exs`)

Benchmark de várias operações da API de mensagens, incluindo inserção, recuperação e roteamento.

Características:

- Testa insert_message (simples e com roteamento)
- Testa get_messages_for_smsc
- Testa list_message_queues
- Pré-popula o banco de dados com dados de teste para cenários realistas

Uso:

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

Saída: `benchmarks/output/message_api_benchmark.html`

Configuração

Todos os benchmarks usam Benchee com as seguintes configurações padrão:

- Aquecimento: 2 segundos
- Tempo: 10 segundos
- Tempo de memória: 2 segundos
- Estatísticas estendidas habilitadas
- Relatórios HTML gerados automaticamente

Saídas

Relatórios de benchmark em HTML são gerados em `benchmarks/output/` e incluem:

- Métricas de desempenho detalhadas
- Gráficos de comparação
- Estatísticas de uso de memória
- Análise estatística

Referência da API SMS-C

← Voltar ao Índice da Documentação

Referência para a API REST SMS-C, gerada a partir da tabela de rotas ao vivo (`config/config.exs`) e das implementações do controlador. Cada endpoint abaixo do serviço em execução realmente roteia.

A aplicação em si serve uma especificação OpenAPI legível por máquina e uma interface Swagger UI interativa. Estes estão sempre sincronizados com a versão implantada:

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

Índice

- Visão Geral
- Autenticação
- Convenções de Resposta
- Status
- Mensagens
- Indicação de Mensagem Esperando
- Localizações
- Frontends
- Eventos
- Campanhas
- Outras Famílias de Endpoint
- Códigos de Status HTTP
- Formato de Erro
- Melhores Práticas

Visão Geral

URL Base

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

Porta Padrão: 8443 (configurável via `config :api_ex, api: %{port: ...}`)
Protocolo: HTTPS quando `enable_tls` está definido (o padrão). Todas as rotas estão montadas sob o prefixo `/api`.

Tipo de Conteúdo

Os corpos de requisição e resposta são JSON:

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

Não há versionamento da API baseado em URL. A tabela de rotas é o contrato; consulte `/api/schema` para a forma exata e implantada.

Autenticação

A API **não possui autenticação em nível de aplicação**. O pipeline de requisição apenas negocia o tipo de conteúdo e anexa a especificação OpenAPI; não há verificação de chave API, token bearer ou sessão na frente das rotas.

O controle de acesso deve, portanto, ser tratado fora da aplicação:

- **TLS**: quando `enable_tls` é verdadeiro, o listener termina HTTPS usando o `tls_cert_path` / `tls_key_path` configurado. Use em produção.
- **Controles de rede**: restrinja a porta da API a fontes confiáveis com um firewall ou um proxy reverso upstream. Vários endpoints registram o IP do chamador (de `X-Forwarded-For` ou do socket) para auditoria, mas não bloqueiam com base nisso.

Não exponha a porta da API diretamente a redes não confiáveis.

Convenções de Resposta

Os controladores retornam o recurso **diretamente**, não encapsulado em um envelope `data`.

Recurso único:

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

Coleções são retornadas como um array JSON simples:

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

Erros em nível de controlador usam uma string `error` simples:

```
{ "error": "payload_hex é obrigatório. Para texto simples, use
POST /api/messages_plaintext" }
```

Erros em nível de framework (falhas de roteamento, corpos não analisáveis) retornam como:

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

Status

Verificação de saúde para monitoramento e balanceadores de carga.

Requisição:

```
GET /api/status
```

Resposta (200 OK):

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

Exemplo:

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

Mensagens

A fila de mensagens é o recurso central. O conteúdo da mensagem é transportado como **`payload_hex`** (os octetos TP-User-Data) mais um byte **`dcs`** (esquema de codificação de dados) e um **`udh_hex`** opcional (cabeçalho de dados do usuário). Chamadas em texto simples devem usar Enviar Texto Simples, que realiza a codificação, seleção de DCS e divisão em partes multipart para você.

Objeto Mensagem

Campos retornados para uma mensagem (veja `ApiEx.Schemas.MessageSchema`):

Campo	Tipo	Notas
`id`	inteiro	ID da Mensagem
`source_msisdn`	string	Número de origem
`destination_msisdn`	string	Número de destino
`payload_hex`	string	Octetos TP-User-Data como hex (conteúdo canônico)
`dcs`	string	Esquema de codificação de dados como um byte hex (`00`=GSM7, `08`=UCS2, `04`=8bit, `C8`=MWI)
`udh_hex`	string / null	Octetos do cabeçalho de dados do usuário como hex
`tp_pid`	string / null	TP-Protocol-Identifier como um byte hex. Padrão `00` (SME-to-SME). `7F` = (U)SIM Download de Dados, que uma mensagem de download de dados SMS-PP (SIM OTA) deve carregar, junto com um `dcs` de classe-2, antes que o dispositivo móvel o passe para o UICC (3GPP TS 23.048 / TS 31.111 cláusula 7.1.1)
`message_text`	string / null	Texto UTF-8 derivado para exibição apenas (não usado para entrega)
`source_smsc`	string	Identificador do SMSC de origem
`dest_smsc`	string / null	Identificador do SMSC de destino

Campo	Tipo	Notas
`status`	string	Um dos `queued`, `backoff`, `sent`, `delivered`, `expired`, `dropped`, `auto_replied`, `balance_rejected`, `deleted`
`send_time`	datetime	
`deliver_time`	datetime / null	
`expires`	datetime	
`delivery_attempts`	inteiro	
`deadletter`	boolean	
`message_parts`	inteiro / null	
`message_part_number`	inteiro / null	
`smsc_node_name`	string	Nó que possui a mensagem

Listar Mensagens

Requisição:

```
GET /api/messages
```

Cabeçalhos Opcionais:

- `smsc: <frontend_name>` - modo de polling: retorna mensagens destinadas a este SMSC.

- `include-unrouted: true|1` - em modo de polling, também retorna mensagens sem um registro de localização (padrão: false).
- `x-decode: true|1` - inclui uma visão decodificada de cada carga útil da mensagem.

Parâmetros de Consulta (modo de lista, usados apenas quando não há cabeçalho `smsc` presente):

- `status` - filtrar por um único átomo de status (veja a lista de status acima).
- `limit` - registros máximos (padrão 100, máximo 1000).
- `offset` - registros a serem ignorados (padrão 0).

O endpoint é **sempre paginado** no modo de lista: sem `limit` ele retorna no máximo 100 registros, nunca todo o armazenamento.

Resposta (200 OK) - um array JSON de objetos de mensagem.

Exemplos:

```
# Poll para as mensagens de um frontend
curl -H "smsc: uk_gateway" https://host:8443/api/messages

# Poll incluindo mensagens não roteadas
curl -H "smsc: uk_gateway" -H "include-unrouted: true"
https://host:8443/api/messages

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

Obter Mensagem

Requisição:

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

Cabeçalho opcional `x-decode: true` adiciona uma visão de carga útil decodificada.

Resposta (200 OK): um único objeto de mensagem. **404** com `{"error": "Mensagem não encontrada"}` se o id for desconhecido.

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

Enviar Mensagem

Envie uma mensagem pré-codificada. **`payload_hex é obrigatório`**; requisições sem ele são rejeitadas com 422.

Requisição:

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

Corpo:

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

Campos:

- `source_msisdn`, `destination_msisdn` - números.
- `payload_hex` - **obrigatório**. Octetos TP-User-Data como hex.
- `dcs` - byte hex do esquema de codificação de dados (padrões aplicados a jusante).
- `udh_hex` - cabeçalho de dados do usuário hex opcional.
- `tp_pid` - byte hex do TP-Protocol-Identifier opcional (padrão `00`). Defina `7F` com um `dcs` de classe-2 (por exemplo, `F6`) para download de dados SMS-PP / SIM OTA; sem isso, o dispositivo móvel armazena o pacote seguro em

vez de entregá-lo ao UICC e nenhum Comprovante de Recebimento é retornado.

- `source_smsc` - identifica o sistema que está enviando.
- `source_type` - classificação de origem opcional.

Resposta (201 Created): o objeto de mensagem criado. Rejeições retornam 422 com uma string `error` (por exemplo, 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 Simples

Envie uma string UTF-8 e deixe o SMS-C lidar com a codificação (GSM7 vs UCS2), seleção de DCS e divisão em partes multipart.

Requisição:

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

Corpo:

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

O texto da mensagem pode ser fornecido como `message`, `text` ou `body`. `source_msisdn` e `destination_msisdn` são obrigatórios.

Resposta (201 Created):

- Parte única: o objeto de mensagem criado.
- 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 Bruto

Envie um PDU SMS bruto (hex) para máxima compatibilidade com frontends legados e SIP.

Requisição:

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

Corpo:

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

Campos:

- `message_body` - **obrigatório**. PDU SMS codificado em hex (ou TPDU, veja abaixo).
- `source_msisdn` - opcional.
- `source_smsc` / `smsc_node_name` - obrigatório no modo apenas TPDU.
- `rp_originator_address`, `rp_destination_address` - forneça estes para enviar um TPDU simples (a camada RP é sintetizada).

O IP do chamador é obtido de `X-Forwarded-For` (ou do socket) para resolver o frontend de origem.

Respostas:

- 201 Created com `{ "status": "success", "response": { "id": <id> } }` (um campo `rp_ack` hex é incluído quando um é gerado).
- 200 OK com `{ "status": "success", "response": { "status": "delivery_report_received" } }` quando o PDU é um relatório de entrega em vez de uma nova mensagem.
- 422 com uma string `error` para hex ausente/inválido ou falhas de decodificação.

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

O serviço decodifica o PDU (3GPP TS 23.040), extrai números/texto/DCS, detecta relatórios de entrega e aplica roteamento. Relatórios de entrega são registrados, mas não armazenados como novas mensagens.

Atualizar Mensagem (PATCH)

Atualização parcial de uma mensagem armazenada.

Requisição:

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

Envie quaisquer campos de mensagem graváveis (por exemplo, `dest_smsc`, `deliver_after`, `status`). Mudanças de metadados são registradas no log de eventos.

Resposta (200 OK): o objeto de mensagem atualizado. 404 se não encontrado, 422 em uma atualização rejeitada.

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

Incrementar Tentativa de Entrega

Registre uma entrega falhada: incrementa o contador de tentativas, move a mensagem para `backoff`, e agenda a próxima nova tentativa com backoff exponencial.

Existem duas formas equivalentes:

```
PUT  /api/messages/:id                  # o recurso de mensagens,
PUT = incrementar
POST /api/messages_increment_attempt        (id no corpo)
PUT  /api/messages_increment_attempt/:id    (id na URL)
```

`PUT /api/messages/:id` é distinto de `PATCH /api/messages/:id`: PATCH faz uma atualização parcial, PUT incrementa a tentativa de entrega. Passe `keep_route=true` para manter o atual `dest_smsc` na nova tentativa.

Resposta (200 OK): a mensagem atualizada, por exemplo.

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

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

Marcar como Entregue

Marque uma mensagem como entregue (armazenamento local ou federado).

```
POST /api/messages_mark_delivered           (id no corpo)
PUT  /api/messages_mark_delivered/:id        (id na URL)
```

Cabeçalho opcional `smsc` seleciona o SMSC federado quando a mensagem não está retida localmente.

Resposta (200 OK): o objeto de mensagem entregue. 404 se não encontrado.

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

Reavaliar Mensagem

Reavalie as regras de roteamento para uma mensagem armazenada e atualize seu `dest_smsc`.

Requisição:

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

Corpo: `{ "id": 12345 }`

Resposta (200 OK):

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

404 se a mensagem não for encontrada, 422 se nenhuma rota corresponder.

Limpar Campo

Redefina um campo específico da mensagem para nulo.

Requisição:

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

Corpo: `{ "id": 12345, "field": "dest_smsc" }`

`field` deve ser um dos `delivery_time`, `dest_smsc`, ou `deliver_after`; qualquer outro valor retorna 400. A mudança é registrada no log de eventos.

Resposta (200 OK): o objeto de mensagem atualizado.

Deletar Mensagem

Requisição:

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

Resposta (204 No Content). 404 se não encontrado. A exclusão é permanente.

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

Indicação de Mensagem Esperando

Envie uma notificação MWI (Indicação de Mensagem Esperando). O controlador constrói o PDU MWI de acordo com 3GPP TS 23.038 e armazena uma mensagem pronta para entrega com um `payload_hex` e `dcs` pré-computados.

Requisição:

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

Corpo:

```
{
  "source_msisdn": "12345",
  "destination_msisdn": "15555555678",
  "indication_type": "voicemail",
  "indication_active": true,
  "message_body": "Você tem 1 nova mensagem de voz",
  "source_smsc": "IMS_SMSC"
}
```

Campos:

- `indication_type` - `voicemail` (padrão), `fax`, `email`, ou `outro`.
- `indication_active` - `true` para definir o indicador, `false` para limpá-lo (padrão `true`).
- `message_body` - texto opcional acompanhante (codificado em GSM7).
- `source_msisdn`, `destination_msisdn`, `source_smsc`, `dest_smsc`.

Resposta (201 Created): o objeto de mensagem criado. 422 em caso de falha.

Localizações

Registros de localização de assinantes para entrega móvel-terminada. Recurso REST padrão; o segmento de caminho `:id` é o IMSI.

```
GET    /api/locations           # listar (paginado: ?
limit=&offset=, cabeçalho x-total-count)
GET    /api/locations/:id       # obter por IMSI
POST   /api/locations           # criar ou atualizar (chaveado por
IMSI)
PATCH  /api/locations/:id       # atualização parcial
PUT    /api/locations/:id       # atualizar
DELETE /api/locations/:id       # remover
```

Corpo de Criação/Atualização:

```
{
  "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` e `msisdn` são obrigatórios; os campos restantes são opcionais. As respostas são o objeto de localização (200/201); falhas de validação retornam 422. Chamado por sistemas de gerenciamento de mobilidade quando um assinante registra ou desregistra.

Frontends

Rastreie conexões SMSC de frontends. Este recurso expõe **apenas índice, mostrar e criar** (não há sub-rotas `active`, `stats`, `history` ou `register`).

```
GET  /api/frontends           # listar todos os registros
GET  /api/frontends/:id       # obter um registro
POST /api/frontends           # registrar ou atualizar um frontend
```

Corpo 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` é obrigatório. Se `ip_address` for omitido, ele é preenchido a partir do IP de origem da requisição (`X-Forwarded-For` ou socket). Se já existir um registro ativo para o `frontend_name`, ele é atualizado; caso contrário, um novo é criado. Os registros expiram após aproximadamente 90 segundos, portanto, os frontends devem se registrar periodicamente.

Resposta (201 Created): o objeto de registro.

Eventos

Eventos do ciclo de vida por mensagem. Os eventos são mantidos em ETS com um TTL rotativo.

Obter Eventos de Mensagem

Requisição:

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

`:id` é o ID numérico da mensagem.

Resposta (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"
    }
  ]
}
```


Os eventos são retornados na ordem em que ocorreram, classificados por `event_time_us` (microsegundos desde a época Unix, UTC); `seq` quebra empates dentro de um nó. `event_time` é o mesmo instante que ISO 8601, e `event_epoch` é ele em segundos inteiros. Eventos que acontecem antes que a mensagem tenha um id (a requisição HTTP, decodificação de PDU, tradução de número) mantêm o tempo em que aconteceram.

Uma mensagem criada através de `POST /api/messages` ou `POST /api/messages_raw` começa com `http_request_received` (método, caminho, IP do chamador e um resumo do corpo) e recebe `http_response_sent` (status HTTP e corpo, truncado em 1024 caracteres) quando a resposta é enviada.

400 se o id não for um inteiro.

Registrar Evento

Requisição:

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

Corpo:

```
{
  "message_id": 12345,
  "name": "external_delivery_confirmed",
  "description": "Confirmado pelo sistema downstream"
}
```

`message_id`, `name` e `description` são todos obrigat��rios. A fonte do evento é detectada automaticamente a partir do IP do chamador (comparado com registros de frontend ativos), retornando ao nome do nó configurado.

Se `message_id` for uma mensagem federada (não está no armazenamento local, mas servida do cache de federação), o evento é enviado ao controlador proprietário e registrado lá. Se o proprietário não puder ser alcançado, a resposta é 502 e nada é registrado localmente.

Resposta (201 Created):

```
{
  "status": "success",
  "message": "Evento registrado com sucesso",
  "event": {
    "message_id": 12345,
    "name": "external_delivery_confirmed",
    "description": "Confirmado pelo sistema downstream",
    "event_source": "smpp-gateway-1",
    "event_epoch": 1698672010
  }
}
```

Campanhas

Mensagens em massa: faça upload de uma lista de destinatários nomeada e, em seguida, execute uma campanha com limite de taxa contra ela. Consulte o guia Mensagens em Massa / Campanhas para conceitos (filtro de público, taxa de gotejamento, ciclo de vida, estatísticas).

Listas de Destinatários

```
GET    /api/campaign_lists           # todas as listas
GET    /api/campaign_lists/:id       # uma lista, inclui
"recipients"
POST   /api/campaign_lists           # criar uma lista
PATCH  /api/campaign_lists/:id       # adicionar destinatários
DELETE /api/campaign_lists/:id       # excluir a lista e seus
destinatários
```

Cada destinatário pode carregar até quatro variáveis de modelo (`var1`..`var4`). Forneça destinatários como um array JSON, uma string CSV bruta ou ambos. Destinatários JSON podem ser strings MSISDN simples ou objetos com `var1`..`var4`:

```
{
  "name": "Clientes VIP",
  "description": "Assinantes de alto valor",
  "recipients": [
    { "msisdn": "12025550101", "var1": "Alice", "var2": "Gold" },
    "12025550102"
  ]
}
```

Para CSV, uma linha de cabeçalho nomeando uma coluna `msisdn` / `destination_msisdn` / `number` / `phone` seleciona a coluna MSISDN (caso contrário, a primeira coluna é usada); as colunas restantes se tornam `var1`..`var4` na ordem das colunas:

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

`PATCH` adiciona destinatários (não os substitui).

Resposta (201 Created): `{ "id": 1, "name": "Clientes VIP", "recipient_count": 2 }`

Campanhas

```
GET    /api/campaigns        # todas as campanhas (paginadas, mais
novas primeiro)
GET    /api/campaigns/:id    # uma campanha, incluindo
estatísticas ao vivo
POST   /api/campaigns        # criar uma campanha (começa em
rascunho)
DELETE /api/campaigns/:id    # parar uma campanha, mas manter seu
registro
```

Criar Corpo:

```
{
  "name": "Aviso de manutenção",
  "message": "Manutenção planejada hoje à noite 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` - **obrigatório**. `message` suporta variáveis de template por destinatário (`{{ var1 }}`..`{{ var4 }}`, `{{ msisdn }}`).

- `list_id` - lista de destinatários. **Omitir para direcionar todos os assinantes atualmente ativos (registrados).**
- `audience` - `active_only` (padrão) ou `all`. Ignorado quando `list_id` é omitido.
- `drip_tps` - mensagens enviadas para a fila por segundo (padrão da configuração).
- `validity_hours` - validade da mensagem em horas; define o `expires` de cada mensagem.
- `deliver_after` - tempo ISO 8601 opcional para reter mensagens até (envio agendado).

`DELETE` interrompe a entrega, mas mantém o registro para que o histórico e as estatísticas permaneçam visíveis; retorna a campanha cancelada (200) e é purgada automaticamente uma vez que ultrapassa o TTL de retenção.

Resposta (`GET /api/campaigns/:id`):


```
{
  "id": 7,
  "name": "Aviso de manutenção",
  "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 uma Campanha

Requisição:

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

Corpo: `{ "campaign_id": 7, "action": "start" }`

`action` é um dos `start`, `pause`, `resume`, `cancel`. Retorna a campanha atualizada (200), `404` se não encontrado, ou `422` se a ação for inválida para o estado atual.

Resultados por Destinatário

```
GET /api/campaign_targets/:id              # :id é o ID da campanha
GET /api/campaign_targets/:id?state=failed # filtrar por estado do destinatário
```

Suporta paginação `limit`/`offset`; a contagem total (após qualquer filtro de `state`) está no cabeçalho de resposta `x-total-count`. Cada entrada carrega o `state` do destinatário (`pending` / `queued` / `delivered` / `failed` / `skipped_inactive`).

Outras Famílias de Endpoint

As seguintes famílias de rotas fazem parte da mesma API, mas são documentadas em detalhes em seus guias dedicados. Os caminhos estão listados aqui para completude.

Roteamento

```
GET    /api/routes            # listar rotas
GET    /api/routes/:id        # obter uma rota
POST   /api/routes            # criar uma rota
PATCH  /api/routes/:id        # atualizar
PUT    /api/routes/:id        # atualizar
DELETE /api/routes/:id        # deletar
GET    /api/routes/export     # exportar todas as rotas
POST   /api/routes/import     # importação em massa
POST   /api/routes/simulate   # simular uma decisão de rota
```

Veja o guia de Roteamento SMS.

Traduções de Números

```
GET    /api/translations           # listar
GET    /api/translations/:id       # obter
POST   /api/translations           # criar
PATCH  /api/translations/:id       # atualizar
DELETE /api/translations/:id       # deletar
GET    /api/translations/export    # exportar
POST   /api/translations/import    # importação em massa
POST   /api/translations/simulate  # simular uma tradução
```

Veja o guia de Tradução de Números.

Operações

```
GET  /api/operations                    # índice de operações
POST /api/operations/clear_messages     # purgar o armazenamento
de mensagens
POST /api/operations/clear_routes       # purgar rotas
POST /api/operations/clear_translations # purgar traduções
POST /api/operations/clear_frontends    # purgar registros de
frontend
```

Estas são ações de manutenção destrutivas; proteja a porta da API de acordo.

Federação

```
GET  /api/federation/status          # status da federação
GET  /api/federation/identity        # identidade de federação
deste nó
POST /api/federation/health          # relatório de saúde do par
POST /api/federation/registry        # troca de registro
POST /api/federation/notify          # notificação de mensagem
recebida
POST /api/federation/delivery_status # callback de status de
entrega
POST /api/federation/delivery_attempt # callback de tentativa de
entrega falhada
POST /api/federation/event           # callback de evento de
mensagem
GET  /api/federation_messages        # visualização de mensagens
federadas
```

Veja o guia de Federação.

Diameter, Logs e DNS

```
GET  /api/diameter   # status de peer/dip Diameter Sh
GET  /api/logs       # logs estruturados recentes
POST /api/naptr      # teste de resolução NAPTR
```

A pesquisa de assinantes Diameter Sh é descrita no guia de pesquisa de assinantes HSS.

Códigos de Status HTTP

Código	Significado	Uso Típico
200	OK	Leitura ou atualização bem-sucedida
201	Criado	Recurso criado
202	Aceito	Aceito para processamento assíncrono
204	Sem Conteúdo	Exclusão bem-sucedida
400	Solicitação Inválida	Entrada malformada (por exemplo, id não inteiro, campo inválido)
404	Não Encontrado	Recurso desconhecido
422	Entidade Não Processável	Falha de validação
500	Erro Interno do Servidor	Erro inesperado do servidor

Formato de Erro

Erros em nível de controlador retornam uma string `error` plana:

```
{ "error": "Mensagem não encontrada" }
```

Erros em nível de framework (rotas não correspondidas, corpos não analisáveis) retornam o envelope normalizado:

```
{ "status": "error", "response": "Não Encontrado" }
```

Melhores Práticas

Submissão de Mensagens

1. **Pré-codificado vs texto simples**: envie `payload_hex` para `/api/messages` quando você já tiver codificado o TPDU; caso contrário, use `/api/messages_plaintext` e deixe o SMS-C codificar e dividir.
2. **Sempre defina** `source_smsc` para que eventos e roteamento possam atribuir a mensagem.
3. **Use números no estilo E.164** de forma consistente.
4. **Trate 422** inspecionando a string `error` em vez de tentar novamente cegamente.

Integração de Frontend

1. **Re-registre** em `POST /api/frontends` bem dentro da janela de expiração de ~90 segundos.
2. **Poll** com o cabeçalho `smsc` para suas mensagens; use `include-unrouted` apenas se você quiser deliberadamente tráfego não roteado.
3. **Marque como entregue** via `/api/messages_mark_delivered` após o sucesso.
4. **Incrementar em caso de falha** via `PUT /api/messages/:id` (ou `/api/messages_increment_attempt`) para agendar uma nova tentativa com backoff.
5. **Registre eventos** para confirmações a jusante.

Operacional

1. Termine o TLS e restrinja a porta da API na camada de rede; não há autenticação embutida.
2. Reutilize conexões HTTP e defina timeouts de cliente sensatos.
3. Monitore as métricas do Prometheus (veja METRICS.md) para saúde da fila e entrega.

Documentação de Conformidade de Interceptação ANSSI R226

Objetivo do Documento: Este documento fornece especificações técnicas necessárias para a autorização ANSSI R226 sob os Artigos R226-3 e R226-7 do Código Penal Francês para o Centro de Serviço de SMS OmniMessage (SMSc).

Classificação: Documentação de Conformidade Regulatória

Autoridade Alvo: Agence nationale de la sécurité des systèmes d'information (ANSSI)

Regulamentação: R226 - Proteção da Privacidade da Correspondência e Interceptação Legal

---

1. ESPECIFICAÇÕES TÉCNICAS DETALHADAS

1.1 Ficha Técnica Comercial

Nome do Produto: OmniMessage SMSc (Centro de Serviço de SMS)
Tipo de Produto: Centro de Mensagens de Telecomunicações
Função Principal: Roteamento, armazenamento e entrega de mensagens SMS
Protocolos de Rede: REST API (HTTPS), protocolos SMS (SMPP, IMS, SS7/MAP via frontends externos)
Modelo de Implantação: Aplicação de servidor local

Stack Tecnológico: Elixir/Erlang, Phoenix Framework, Mnesia, MySQL/PostgreSQL

Capacidades Principais

Processamento de Mensagens:

- Fila de mensagens SMS centralizada com REST API
- Design independente de protocolo suportando frontends SMPP, IMS, SS7/MAP
- Motor de roteamento dinâmico com roteamento baseado em prefixo
- Lógica de repetição com retrocesso exponencial
- Expiração de mensagens e gerenciamento de fila de cartas mortas
- Geração e arquivamento de Registros de Detalhes de Chamadas (CDR)
- Desempenho: ~1.750 mensagens/segundo de taxa de inserção, capacidade de 150 milhões de mensagens/dia

Armazenamento de Mensagens:

- **Fila de Mensagens Ativas:** Banco de dados em memória Mnesia com persistência em disco opcional
  - Armazenamento primário: RAM para acesso ultra-rápido (latência sub-milissegundo)
  - Backup em disco: modo `disc_copies` grava em disco para recuperação em caso de falha
  - Recuperação automática: Mensagens sobrevivem a reinicializações do sistema
  - Retenção: Configurável (padrão 24 horas), depois limpeza automática
- **Arquivo CDR de Longo Prazo:** Banco de dados MySQL/PostgreSQL (separado da fila de mensagens)
  - CDRs gravados quando mensagens são entregues, expiradas, falhadas ou rejeitadas
  - Banco de dados SQL usado SOMENTE para exportação/arquivamento de CDR, NÃO para operações de mensagens ativas
  - Sem impacto no desempenho do roteamento de mensagens (gravação assíncrona)
- **Benefícios da Arquitetura de Dois Níveis:**

    - Fila ativa: Extremamente rápida (1.750 msg/sec) sem gargalo SQL
    - Arquivo CDR: Retenção de longo prazo (meses/anos) para faturamento e interceptação legal
    - Separação clara: Operações de mensagens nunca tocam SQL
- Suporte a cluster para alta disponibilidade (replicação Mnesia entre nós)

Interfaces de Rede:

- **REST API:** HTTPS (porta 8443) para comunicação com frontends externos
- **Painel de Controle:** HTTPS (porta 8086) para gerenciamento baseado na web
- **Protocolos de Frontend:** SMPP, IMS, SS7/MAP (via aplicações de gateway externas)
- **Banco de Dados:** MySQL/PostgreSQL para armazenamento de CDR

Roteamento e Processamento:

- Roteamento SMS dinâmico com atualizações de configuração em tempo real
- Correspondência baseada em prefixo (números chamadores/chamados)
- Filtragem de SMSC de origem e tipo
- Balanceamento de carga baseado em prioridade e peso
- Tradução e normalização de números
- Suporte a pesquisa DNS ENUM (Mapeamento de Números E.164)
- Capacidades de resposta automática e descarte de mensagens
- Controle de cobrança por rota (integração OCS sobre Diameter Ro)

 Arquitetura e recursos completos documentados em README.md

1.2 Capacidades de Interceptação

OmniMessage implementa as interfaces de interceptação ETSI TS 103 221 (X1, X2, X3). OmniMessage atua como o Elemento de Rede e Ponto de Interceptação. Uma Instalação de Monitoramento de Aplicação da Lei provisiona alvos via X1. OmniMessage entrega SMS interceptados como 3GPP

TS 33.128 xIRI via X2. Para os papéis de interface e a referência completa de parâmetros, veja Interceptação Legal (X1/X2/X3).

1.2.1 Aquisição de Mensagens

Captura de Mensagens SMS:

- O OmniMessage SMSc processa todas as mensagens SMS entre assinantes e redes externas
- Acesso completo aos metadados e conteúdo da mensagem, incluindo:
  - MSISDN de origem (número móvel)
  - MSISDN de destino (número móvel)
  - IMSI de origem (Identidade Internacional do Assinante Móvel)
  - IMSI de destino
  - Corpo da mensagem (conteúdo de texto)
  - Dados PDU (Unidade de Dados de Protocolo) brutos
  - Informações TP-DCS (Esquema de Codificação de Dados)
  - Codificação da mensagem (GSM7, UCS-2, 8-bit, Latin-1)
  - Indicadores de mensagem multipart e dados de reassemblagem
  - Informações do Cabeçalho de Dados do Usuário (UDH)

Aquisição de Metadados de Mensagens:

- Registros de Detalhes de Chamadas (CDR) completos armazenados no banco de dados com:
  - ID da mensagem (identificador único)
  - Número chamador (MSISDN de origem)
  - Número chamado (MSISDN de destino)
  - Timestamp de submissão (quando a mensagem entrou no sistema)
  - Timestamp de entrega (quando a mensagem foi entregue)
  - Timestamp de expiração (quando a mensagem expirou se indeliverável)
  - Status (entregue, expirado, falhado, rejeitado)
  - Contagem de tentativas de entrega
  - Partes da mensagem (para SMS concatenados/multi-part)
  - Identificador SMSC de origem

- Identificador SMSC de destino
- Nó de origem (nome do nó do cluster Erlang)
- Nó de destino (para implantações distribuídas)
- Sinalizador de carta morta (indicador de exaustão de tentativas)

📄 Esquema completo de CDR documentado em CDR_SCHEMA.md

Acesso à Fila de Mensagens:

- Monitoramento em tempo real da fila de mensagens
- Endpoints da REST API para recuperação de mensagens
- Consultas de banco de dados para busca histórica de mensagens
- Capacidades de filtragem por:
  - Número de telefone (origem/destino)
  - Gateway SMSC
  - Faixa de tempo
  - Status da mensagem
  - Tentativas de entrega

📄 Documentação completa da API em API_REFERENCE.md

1.2.2 Capacidades de Processamento de Dados

Arquitetura de Armazenamento de Mensagens (Sistema de Dois Níveis):

O SMSc utiliza uma sofisticada arquitetura de armazenamento em dois níveis que separa o processamento operacional de mensagens do arquivamento de longo prazo:

Nível 1: Fila de Mensagens Ativas (Mnesia)

- **Propósito:** Operações de roteamento e entrega de mensagens em tempo real
- **Tecnologia:** Banco de dados distribuído Erlang Mnesia
- **Modo de Armazenamento:** Em memória com backup disc_copies
  - Armazenamento primário em RAM para máxima velocidade
  - Sincronização automática em disco para recuperação em caso de falha

    - Mensagens persistem entre reinicializações do sistema
- **Desempenho:** Operações de leitura/gravação sub-milissegundo
- **Retenção:** Curto prazo (padrão 24 horas), configurável
- **Limpeza:** Arquivamento automático para o banco de dados CDR, depois exclusão do Mnesia
- **Operações:** Todas as operações da fila de mensagens (inserir, atualizar, status de entrega, roteamento)
- **Recurso Crítico:** O banco de dados SQL NUNCA é consultado durante o roteamento/entrega de mensagens

Nível 2: Arquivo CDR (MySQL/PostgreSQL)

- **Propósito:** Armazenamento de longo prazo para faturamento, análises e interceptação legal
- **Tecnologia:** Banco de dados SQL tradicional (MySQL ou PostgreSQL)
- **Gatilho de Gravação:** CDRs gravados SOMENTE quando mensagens atingem o estado final:
    - Mensagem entregue com sucesso
    - Mensagem expirou (excedeu o período de validade)
    - Mensagem falhou permanentemente
    - Mensagem rejeitada por regras de roteamento
- **Modo de Gravação:** Gravação assíncrona em lote (sem impacto no desempenho do roteamento de mensagens)
- **Retenção:** Longo prazo (meses a anos), configurável conforme requisitos regulatórios
- **Operações:** Consultas históricas, relatórios, conformidade, interceptação legal
- **Acesso:** Consultas SQL, REST API (futuro), exportação CSV/JSON

Principais Benefícios Arquitetônicos:

1. **Desempenho:** Operações de roteamento ativas nunca tocam SQL (sem gargalo de banco de dados)
2. **Escalabilidade:** Mnesia lida com 1.750+ mensagens/segundo sem sobrecarga SQL

3. **Confiabilidade:** O modo disc_copies garante que não haja perda de mensagens em caso de falha
4. **Conformidade:** O banco de dados CDR fornece um registro de auditoria permanente
5. **Separação de Preocupações:** Dados operacionais vs. dados arquivados claramente separados

Ciclo de Vida da Mensagem:

```
1. Mensagem submetida → Armazenada no Mnesia (RAM + backup em
disco)
2. Mensagem roteada ⇒ Consulta Mnesia (ultra-rápida)
3. Mensagem entregue/expirada → CDR gravado no SQL (assíncrono)
4. Após 24h → Mensagem excluída do Mnesia (trabalhador de limpeza)
5. CDR permanece no SQL → Disponível para consultas de
interceptação legal (anos)
```

Retenção e Recuperação de Dados:

- Retenção ou exclusão configurável do corpo da mensagem para privacidade
- Preservação de dados binários (armazenamento PDU bruto em Mnesia e CDR)
- Capacidade de busca de texto completo (se habilitada no banco de dados CDR)
- Campos CDR indexados para consultas rápidas de interceptação legal

Rastreamento de Frontend:

- Rastreamento em tempo real de frontends SMSC externos (SMPP, IMS, gateways MAP)
- Registro de frontend com monitoramento de heartbeat
- Rastreamento de status de saúde (ativo/expirado)
- Histórico de tempo de atividade/inatividade
- Rastreamento de endereço IP e nome do host
- Registro de configuração específico do frontend

1.2.3 Capacidades de Análise

Monitoramento em Tempo Real:

- Painel da interface web mostrando:
  - Fila de mensagens ativas
  - Submissão e entrega de mensagens
  - Decisões de roteamento e seleção de gateway
  - Status do gateway de frontend
  - Utilização de recursos do sistema
- Integração de métricas Prometheus para monitoramento operacional
- Métricas de desempenho (taxa de transferência, latência, taxas de sucesso)

 Guia completo de monitoramento em OPERATIONS_GUIDE.md
 Documentação de métricas em METRICS.md

Análise Histórica:

- Banco de dados CDR consultável por:
  - Faixa de tempo
  - Número da parte chamadora/chamada
  - Status da mensagem
  - Gateway SMSC
  - Tentativas de entrega
  - Conteúdo da mensagem (busca de texto completo se habilitada)
- Capacidades de análise estatística:
  - Volume de mensagens por hora/dia/mês
  - Taxas de sucesso/falha por rota
  - Tempos médios de entrega
  - Análise de mensagens multi-part
  - Padrões de falha de entrega

Rastreamento de Assinantes:

- Histórico de mensagens por número de telefone (MSISDN)
- Rastreamento baseado em IMSI (quando disponível de frontends IMS/MAP)

- Análise de padrões de chamadas
- Correlação de partes de comunicação
- Análise temporal (frequência de mensagens, padrões de tempo)

Análise de Rede:

- Métricas de desempenho de rotas
- Disponibilidade e saúde do gateway
- Visualização do fluxo de mensagens
- Distribuição de nós de cluster (implantações multi-nós)
- Análise de tentativas de entrega
- Análise de padrões de repetição

Inteligência de Números:

- Normalização de números E.164
- Identificação de país/região a partir do prefixo do número
- Regras de tradução e reescrita de números
- Pesquisa DNS ENUM para inteligência de roteamento
- Decisões de roteamento baseadas em prefixo

 Guia de tradução de números em number_translation_guide.md
 Guia de roteamento em sms_routing_guide.md

1.3 Capacidades de Contramedidas

1.3.1 Mecanismos de Proteção à Privacidade

Confidencialidade da Comunicação:

- HTTPS/TLS para comunicações da REST API
- Autenticação baseada em certificado
- Criptografia de conexão de banco de dados (suporte a TLS)
- Exclusão configurável do corpo da mensagem após a entrega

Controle de Acesso:

- Controle de acesso à interface web
- Mecanismos de autenticação da API
- Controles de acesso ao banco de dados
- Autenticação de registro de frontend

Registro de Auditoria:

- Registro completo de eventos do sistema
- Registro de submissão/entrega de mensagens
- Rastreamento de alterações de configuração
- Registro de ações administrativas
- Registro estruturado com níveis configuráveis

1.3.2 Recursos de Proteção de Dados

Privacidade da Mensagem:

- Exclusão configurável do corpo da mensagem após a entrega
- Corpo da mensagem excluído da exibição da interface (opcional)
- Corpo da mensagem excluído das exportações (opcional)
- Campo do corpo da mensagem CDR pode ser definido como NULL para privacidade

Segurança do Banco de Dados:

- Suporte à criptografia de tabela MySQL (ENCRYPTION='Y')
- Suporte à criptografia de dados transparente do PostgreSQL
- Separação de funções de acesso ao banco de dados
- Contas de usuário somente leitura para análises
- Acesso restrito ao conteúdo da mensagem

Fortalecimento do Sistema:

- Portas de rede expostas mínimas
- Gerenciamento de certificados TLS
- Armazenamento seguro de configurações
- Separação de configurações baseadas em ambiente

- Segurança de cluster com protocolo de distribuição Erlang

1.4 Arquitetura de Armazenamento: Design de Dois Níveis Mnesia + SQL

Visão Geral

O OmniMessage SMSc emprega uma arquitetura de armazenamento em dois níveis única, especificamente projetada para separar o processamento operacional de mensagens de alto desempenho do armazenamento de conformidade e arquivamento de longo prazo.

Nível 1: Fila de Mensagens em Memória Mnesia

O que é Mnesia?

- Banco de dados distribuído embutido no runtime Erlang/OTP
- Armazenamento híbrido: Primário em memória com backup automático em disco
- Transações compatíveis com ACID
- Replicação de cluster entre múltiplos nós

Modo de Armazenamento: disc_copies

- **Primário em Memória:** Todas as mensagens ativas armazenadas em RAM
  - Operações de leitura/gravação ultrarrápidas (sub-milissegundo)
  - Sem I/O de disco durante operações normais de roteamento de mensagens
  - Permite taxa de transferência de 1.750+ mensagens/segundo
- **Backup em Disco (Automático):** Mnesia sincroniza RAM com disco
  - Gravações ocorrem assíncronas em segundo plano
  - Cópia em disco atualizada em cada compromisso de transação
  - Recuperação de falhas: O sistema reinicia com todas as mensagens intactas
  - Localização: diretório `Mnesia.*/` nos dados da aplicação

Ciclo de Vida da Mensagem no Mnesia:

1. Mensagem chega via REST API → Inserida na RAM do Mnesia + backup em disco
2. Motor de roteamento consulta Mnesia → Resposta instantânea (acesso à memória)
3. Gateway externo consulta por mensagens → Consulta Mnesia (acesso à memória)
4. Gateway atualiza status de entrega → Atualização Mnesia (memória + disco)
5. Após entrega/expiração → Mensagem marcada para limpeza
6. Trabalhador de limpeza (padrão de 24h) → Mensagem excluída do Mnesia

Recurso Crítico de Desempenho:

- **ZERO consultas ao banco de dados SQL** durante o roteamento/entrega ativa de mensagens
- SQL é completamente ignorado para o processamento operacional de mensagens
- Isso elimina o gargalo tradicional do SMS-C (I/O de banco de dados)

Nível 2: Banco de Dados SQL para Exportação/Arquivamento de CDR

O que é CDR (Registro de Detalhes de Chamadas)?

- Registro de auditoria permanente de metadados e conteúdo da mensagem
- Gravado no banco de dados MySQL ou PostgreSQL
- Usado para faturamento, análises, conformidade e interceptação legal

Quando os CDRs são Gravados: Registros de CDR são criados SOMENTE quando mensagens atingem um estado final:

- □ Mensagem entregue com sucesso
- □ Mensagem expirou (excedeu o período de validade sem entrega)
- □ Mensagem falhou permanentemente (número inválido, erro de roteamento)
- □ Mensagem rejeitada (regras de roteamento, falha de validação)

Como os CDRs são Gravados:

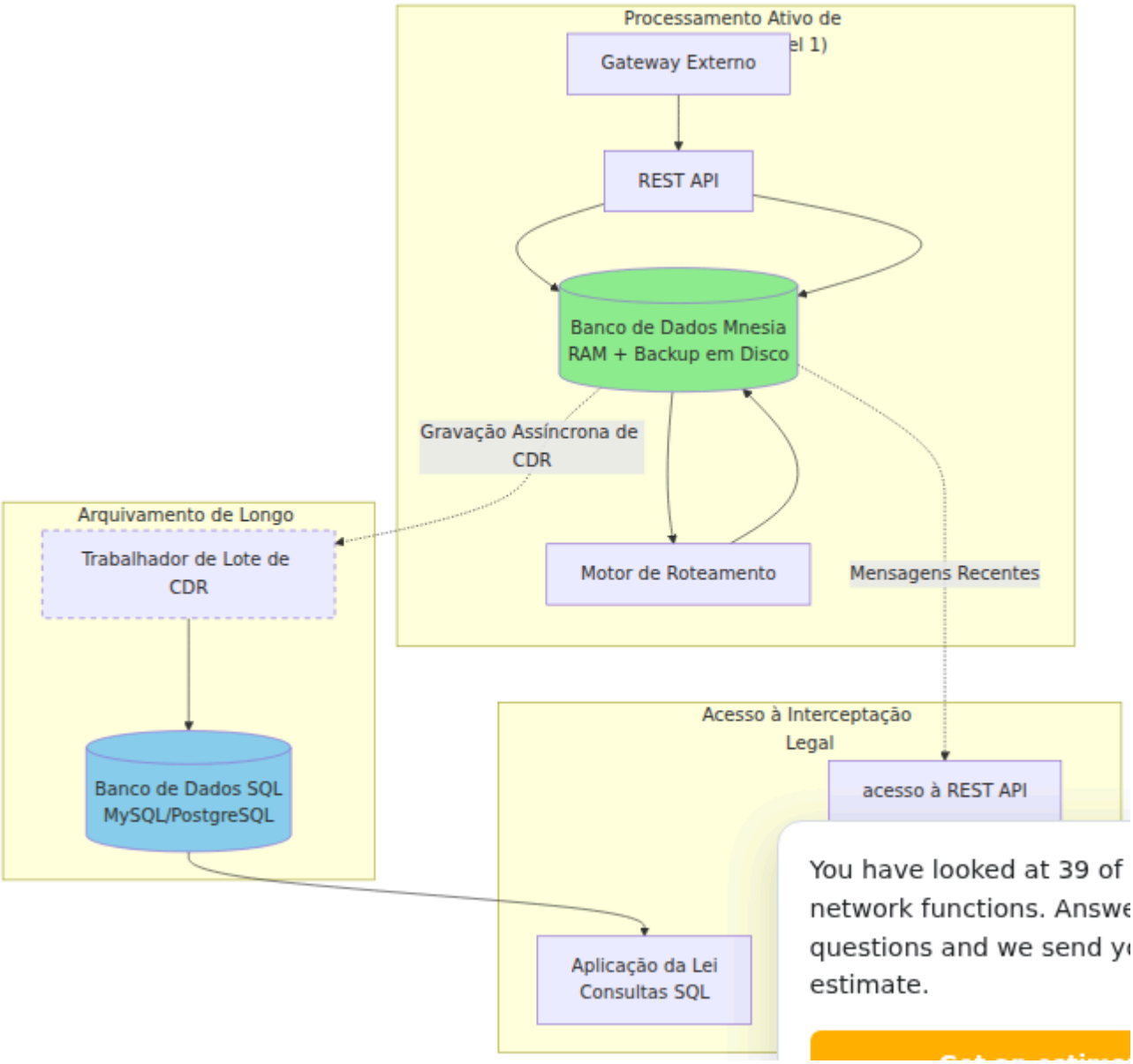
- **Gravação assíncrona em lote:** CDRs gravados em processo de trabalhador em segundo plano
- **Sem bloqueio:** O roteamento de mensagens nunca espera pela gravação SQL
- **Inserções em lote:** Múltiplos CDRs agrupados (padrão 100) e gravados juntos
- **Intervalo de flush:** 100ms padrão (configurável)
- **Tratamento de erros:** Gravações de CDR falhadas registradas, processamento de mensagens continua

```
# Configuração em config/runtime.exs
config :sms_c,
  batch_insert_batch_size: 100,          # Tamanho do lote para
gravações de CDR
  batch_insert_flush_interval_ms: 100    # Intervalo de flush
```

Propósito do Banco de Dados SQL:

- □ NÃO usado para: Operações da fila de mensagens ativas
- □ NÃO usado para: Decisões de roteamento de mensagens
- □ NÃO usado para: Entrega de mensagens em tempo real
- □ SOMENTE usado para: Arquivamento de longo prazo de CDR e consultas históricas
- □ SOMENTE usado para: Consultas de interceptação legal (meses/anos de histórico)
- □ SOMENTE usado para: Relatórios de faturamento e análises

Diagrama de Arquitetura



Legenda:

- Linhas sólidas: Operações síncronas (tempo real)
- Linhas tracejadas: Operações assíncronas (em segundo plano)
- Verde: Nível de alto desempenho (em memória)
- Azul: Nível de arquivamento (SQL persistente)

Implicações da Interceptação Legal

Mensagens Recentes (< 24 horas):

- Acessíveis via Mnesia (consultas da REST API)
- Recuperação ultra-rápida
- Conteúdo completo da mensagem disponível

- Monitoramento em tempo real possível

Mensagens Históricas (> 24 horas):

- Acessíveis via banco de dados SQL (tabela CDR)
- Desempenho padrão de consulta SQL
- Metadados completos da mensagem sempre disponíveis
- Corpo da mensagem disponível (a menos que o modo de privacidade esteja habilitado)

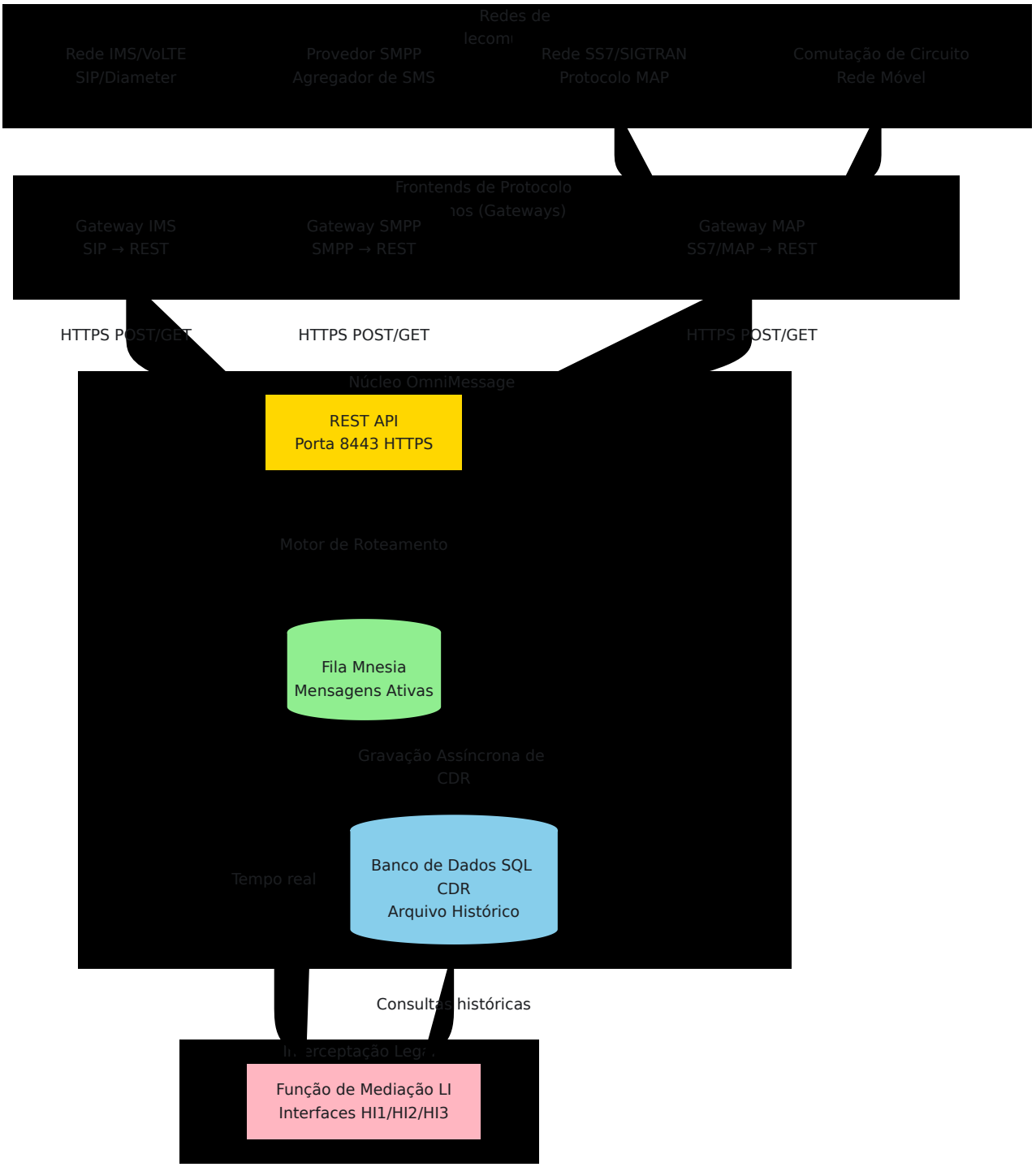
Benefícios de Conformidade:

1. **Sem perda de dados:** O modo disc_copies garante que as mensagens sobrevivam a falhas
2. **Registro de auditoria permanente:** CDRs retidos por anos no banco de dados SQL
3. **Desempenho:** Consultas de interceptação legal não impactam o roteamento de mensagens
4. **Flexibilidade:** Mensagens recentes (Mnesia) + mensagens históricas (SQL) ambas acessíveis

1.5 Arquitetura de Integração de Frontend Multi-Protocolo

O OmniMessage SMSc emprega um design central independente de protocolo que se interfaceia com gateways específicos de protocolo externos (frontends) via uma REST API unificada. Esta arquitetura permite que a interceptação legal capture mensagens independentemente de qual protocolo de telecomunicações foi usado para enviá-las ou recebê-las.

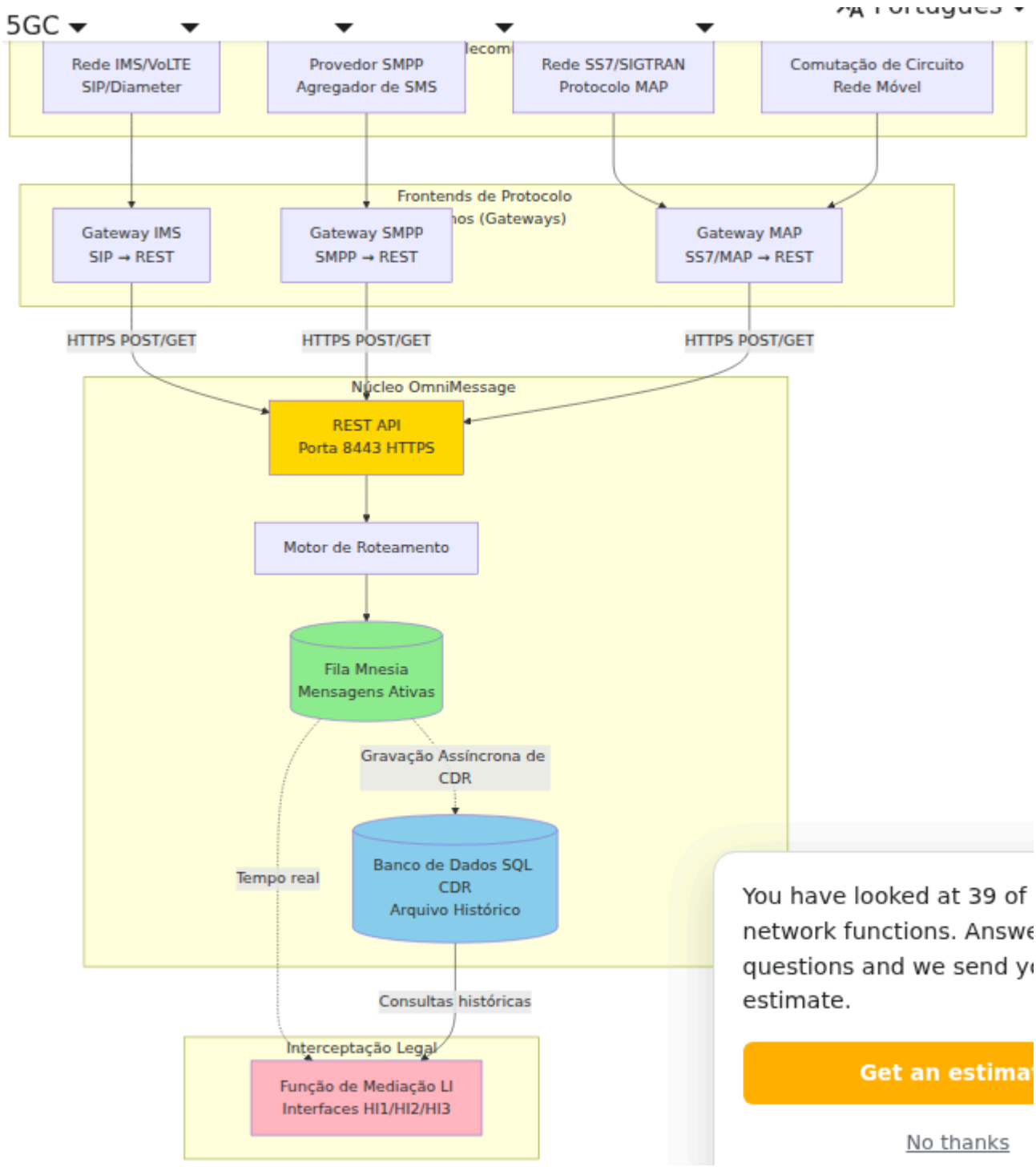
Visão Geral da Arquitetura



Detalhes de Integração de Protocolo de Frontend

1. Integração de Frontend IMS/SIP

Redes IMS usam o protocolo SIP para mensagens SMS sobre IP. O gateway IMS traduz entre SIP e a REST API do SMSc.



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

Dados de Interceptação Específicos do IMS:

- IMSI de origem/destino (a partir do registro IMS)
- Cabeçalhos SIP de Identidade Assertiva
- ID de Chamada SIP para correlação
- Localização da rede IMS (Informações de Rede de Acesso P)
- Perfis de assinantes do HSS IMS

```
**2. Integração Frontend SMPP**

O SMPP é o protocolo padrão da indústria para agregadores de SMS e provedores de serviços. O gateway SMPP traduz mensagens SMPP baseadas em PDU para chamadas de API REST.

```mermaid
sequenceDiagram
    participant ESME as Cliente SMPP Externo<br/> (ESME)
    participant GW as Gateway SMPP<br/>(Frontend SMPP)
    participant API as API REST SMSc
    participant Queue as Fila Mnesia

    ESME->>GW: BIND_TRANSMITTER<br/>system_id: "customer123"
    GW-->>ESME: BIND_TRANSMITTER_RESP

    ESME->>GW: SUBMIT_SM<br/>source_addr: "447700900123"
<br/>dest_addr: "447700900456"<br/>short_message: "Teste de SMS"
<br/>data_coding: 0 (GSM7)

    Note over GW: Decodificar SMPP PDU<br/>Extrair todos os campos
TP-DCS<br/>Extrair UDH se presente

    GW->>API: POST /api/messages<br/>{<br/>  "source_msisdn":
"447700900123",<br/>  "destination_msisdn": "447700900456",<br/>
"message_body": "Teste de SMS",<br/>  "source_smsc":
"smpp.customer123",<br/>  "tp_dcs_character_set": "gsm7",<br/>
"tp_dcs_coding_group": "general_data_coding",<br/>  "raw_pdu":
"base64_encoded_pdu"<br/>}

    API->>Queue: Inserir mensagem
    Queue-->>API: ID da Mensagem: 12346
    API-->>GW: 201 Criado<br/>{id: 12346}

    GW-->>ESME: SUBMIT_SM_RESP<br/>message_id: "12346"

    Note over GW,Queue: [PONTO DE INTERCEÇÃO]<br/>PDU SMPP
completo preservado

    loop Polling para entrega
        GW->>API: GET /api/messages/get_by_smsc?
smsc=smpp.provider-01
        API-->>GW: Mensagens a serem entregues
```
```

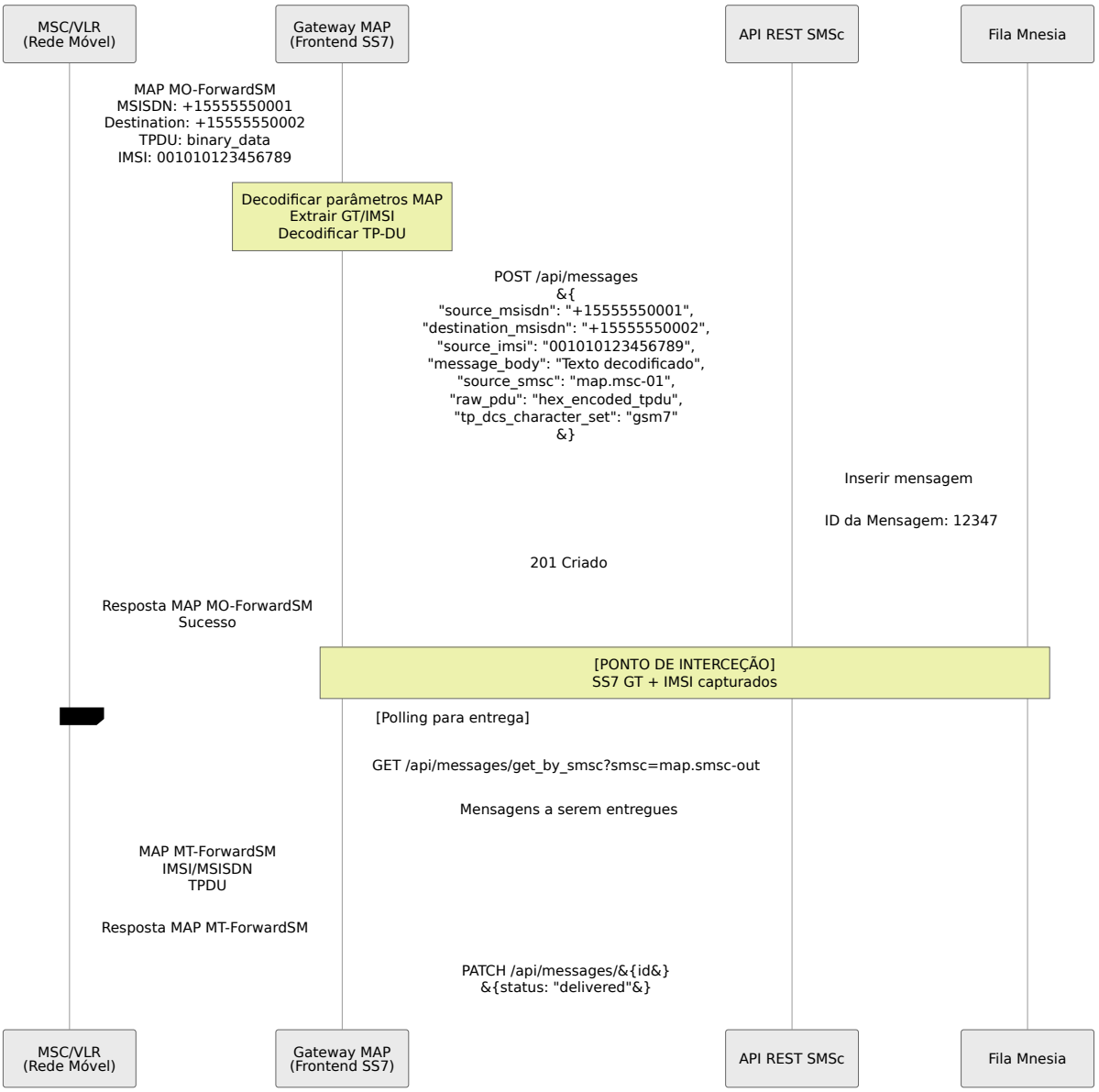
```
        GW->>ESME: DELIVER_SM
        ESME-->>GW: DELIVER_SM_RESP
        GW->>API: PATCH /api/messages/&#123;id&#125;
<br/>&#123;status: "delivered"&#125;
    end
```

Dados de Interceptação Específicos do SMPP:

- PDU SMPP completo (formato binário preservado)
- Detalhes do Esquema de Codificação de Dados (DCS)
- Cabeçalho de Dados do Usuário (UDH) para mensagens concatenadas
- system_id ESME (identificação do cliente)
- Informações do plano de numeração TON/NPI
- Flags de entrega registrada

3. Integração Frontend SS7/MAP

Redes legadas de comutação de circuitos usam o protocolo SS7 MAP para SMS. O gateway MAP traduz entre sinalização SS7 e API REST.



Dados de Interceptação Específicos do SS7/MAP:

- IMSI de mensagens MAP
- Endereços de Título Global (GT)
- Endereço MSC/VLR (identificação do elemento de rede)
- Endereços de chamada/recepção SCCP
- Códigos de operação MAP
- Formato binário TP-User-Data

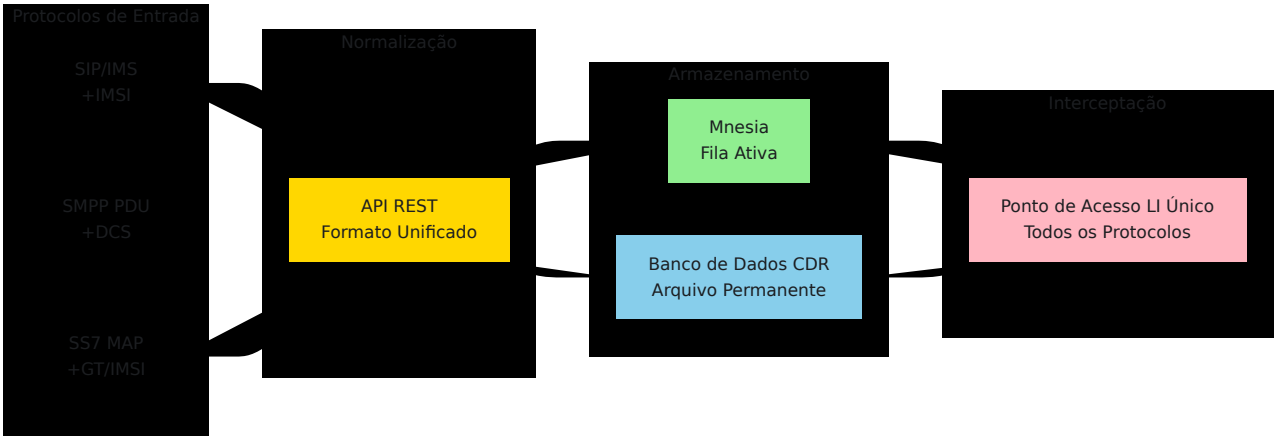
Interceptação Unificada em Todos os Protocolos

Benefício Chave para Interceptação Legal: Independentemente do protocolo utilizado (IMS/SIP, SMPP ou SS7/MAP), todas as mensagens

convergem no núcleo SMSc com uma estrutura de dados normalizada, permitindo:

1. **Monitoramento Agnóstico ao Protocolo:** Um único ponto de interceptação captura todos os tipos de mensagens
2. **Formato CDR Unificado:** Todos os protocolos escrevem no mesmo esquema CDR
3. **Correlação Entre Protocolos:** Rastrear mensagens através de fronteiras de protocolo
4. **Preservação Completa de Metadados:** Campos específicos do protocolo preservados no CDR

Resumo do Fluxo de Dados:



Identificação de Protocolo no CDR:

- O campo `source_smsc` indica o protocolo frontend (por exemplo, "ims.gateway-01", "smpp.customer123", "map.msc-01")
- Permite filtragem e análise por tipo de protocolo
- Consultas de interceptação legal podem direcionar protocolos específicos ou todos os protocolos

1.6 Arquitetura Técnica para Interceptação Legal

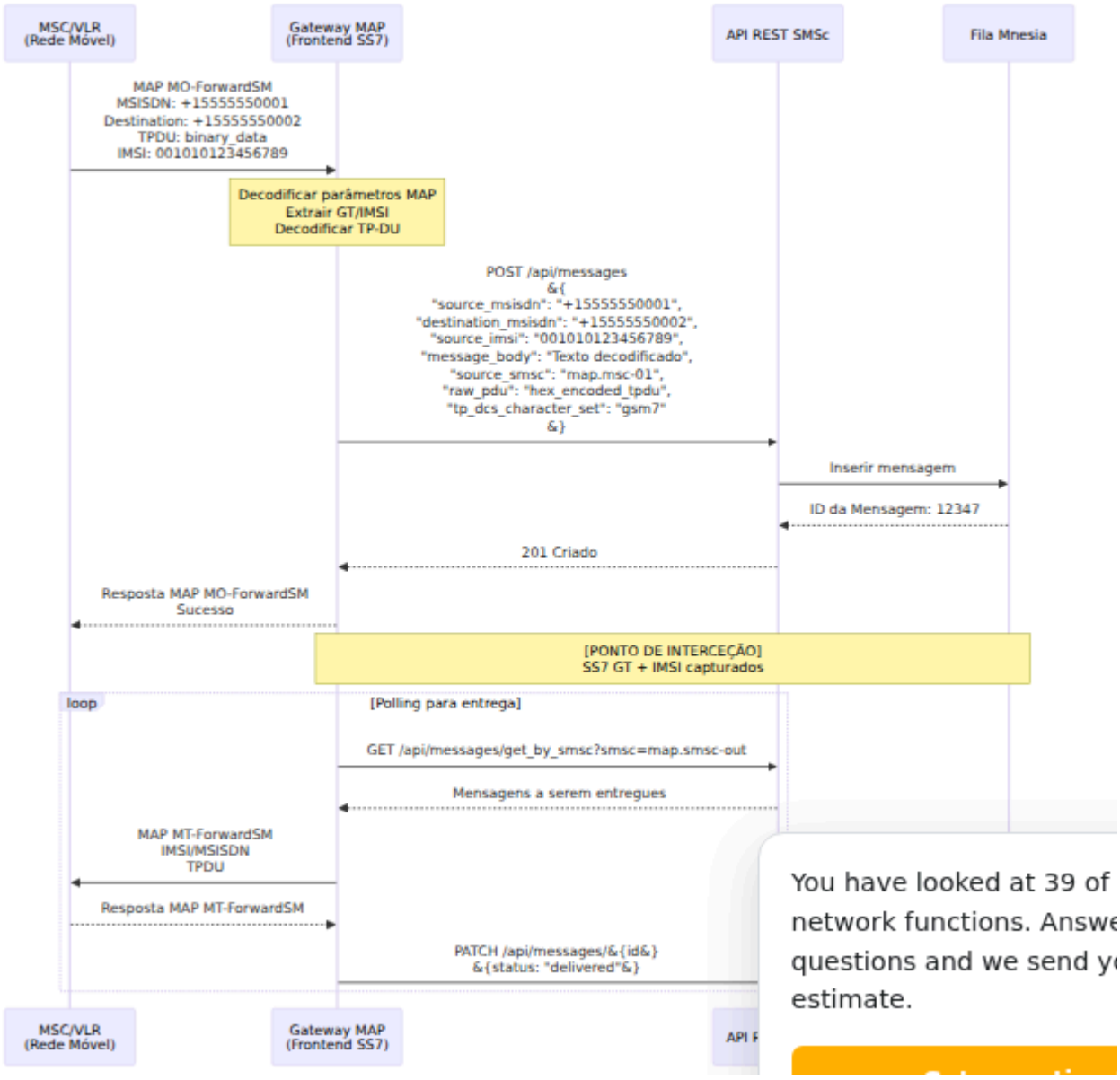
Pontos de Integração de Interceptação Legal

A arquitetura de armazenamento em duas camadas fornece múltiplos pontos de acesso para interceptação legal, otimizados tanto para monitoramento em

tempo real (Mnesia) quanto para análise histórica (SQL).

1. Acesso à API REST para Mensagens Recentes (Mnesia):

Acesso a mensagens ativas na fila Mnesia (tipicamente últimas 24 horas):



Endpoints da API para Interceptação em Tempo Real:

- `GET /api/messages` - Listar mensagens ativas com filtragem
- `GET /api/messages/{id}` - Obter detalhes de mensagem específica (da Mnesia)
- `GET /api/messages/get_by_smsc?smsc=X` - Obter mensagens por gateway
- Todas as consultas atingem a Mnesia (em memória) para resposta instantânea

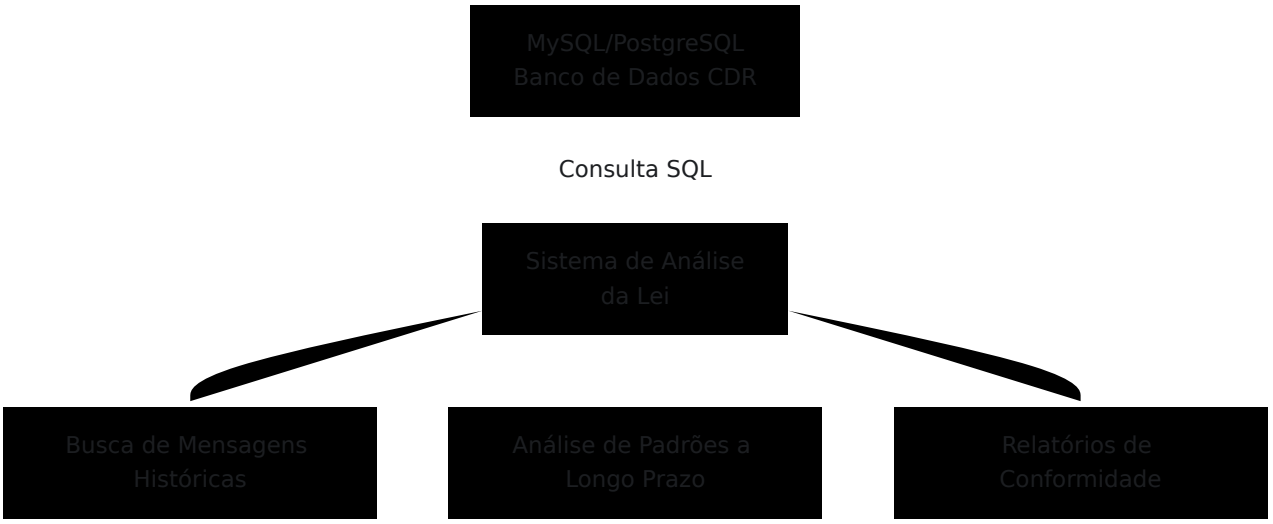
Nota: Esses endpoints consultam a fila de mensagens ativas da Mnesia, fornecendo acesso a mensagens que estão sendo processadas ou que foram entregues recentemente (dentro do período de retenção).

Parâmetros de Consulta:

- Filtrar por MSISDN de origem/destino
- Filtrar por intervalo de tempo
- Filtrar por gateway SMSC
- Filtrar por status da mensagem
- Suporte a ordenação e paginação

2. Acesso Direto ao Banco de Dados CDR para Mensagens Históricas (SQL):

Acesso a mensagens arquivadas no banco de dados SQL (todas as mensagens entregues/expiradas/falhadas):



Acesso SQL Direto:

- Credenciais de banco de dados somente leitura para sistemas autorizados
- Acesso a consultas SQL à tabela `cdrs` (trilha de auditoria permanente)
- **Método de Acesso:** Cliente SQL padrão (mysql, psql, DBeaver, etc.)
- **Fonte de Dados:** Somente mensagens arquivadas (não fila ativa)
- Campos indexados para busca eficiente:
  - `calling_number` (indexado) - Número de telefone de origem
  - `called_number` (indexado) - Número de telefone de destino

- `message_id` (indexado) - Identificador único da mensagem
- `submission_time` (indexado) - Quando a mensagem entrou no sistema
- `status` (indexado) - Status final de entrega
- `dest_smsc` (indexado) - Gateway utilizado para entrega

Nota: O banco de dados CDR contém registros permanentes de todas as mensagens processadas. Esta é a principal fonte de dados para consultas históricas de interceptação legal (meses/anos de dados).

3. Feed de Mensagens em Tempo Real (PubSub):

- Integração Phoenix PubSub para eventos em tempo real
- Notificações de envio de mensagens
- Notificações de entrega de mensagens
- Eventos de mudança de status de mensagens
- Filtragem de eventos configurável por critérios
- Suporte a WebSocket para monitoramento ao vivo

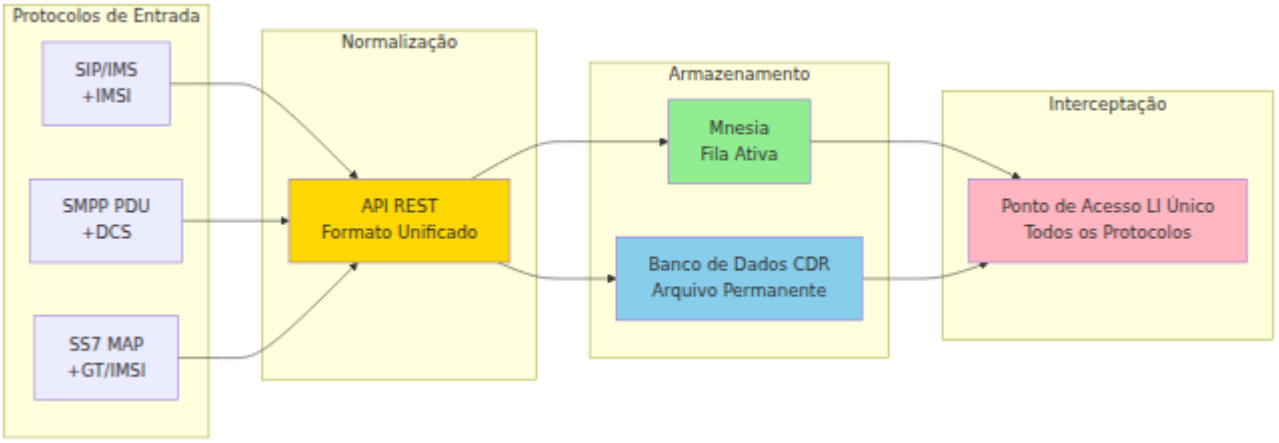
4. Interface de Exportação em Lote:

- Exportação CSV de registros CDR
- Exportação JSON para acesso programático
- Campos de exportação configuráveis
- Exportações baseadas em intervalo de tempo
- Exportações conscientes da privacidade (exclusão opcional do corpo da mensagem)

Interfaces Padrão de Interceptação Legal da ETSI

O OmniMessage SMSc fornece a base para implementar interfaces de interceptação legal compatíveis com a ETSI. O núcleo do SMSc não implementa nativamente as interfaces de transferência HI1/HI2/HI3. Ele fornece todos os pontos de acesso de dados necessários. Você pode integrar esses pontos de acesso com sistemas externos de Função de Mediação de Interceptação Legal (LIMF).

Interfaces LI Padrão da ETSI:



Descrições das Interfaces:

Interface HI1 - Função de Administração:

- **Propósito:** Provisionamento de mandatos e alvos da aplicação da lei para o sistema de interceptação
- **Direção:** LEMF → LIMF (bidirecional)
- **Funções:**
  - Ativar/desativar interceptação para alvos específicos (MSISDNs, IMSIs)
  - Definir duração e período de validade da interceptação
  - Configurar critérios de filtragem (números de telefone, janelas de tempo)
  - Recuperar status de interceptação
- **Integração com SMSc:**
  - LIMF mantém lista de alvos (banco de dados de mandatos)
  - LIMF consulta CDR/API do SMSc para mensagens correspondentes
  - LIMF filtra com base nos critérios provisionados pelo HI1

Interface HI2 - Entrega de IRI (Informações Relacionadas à Interceptação):

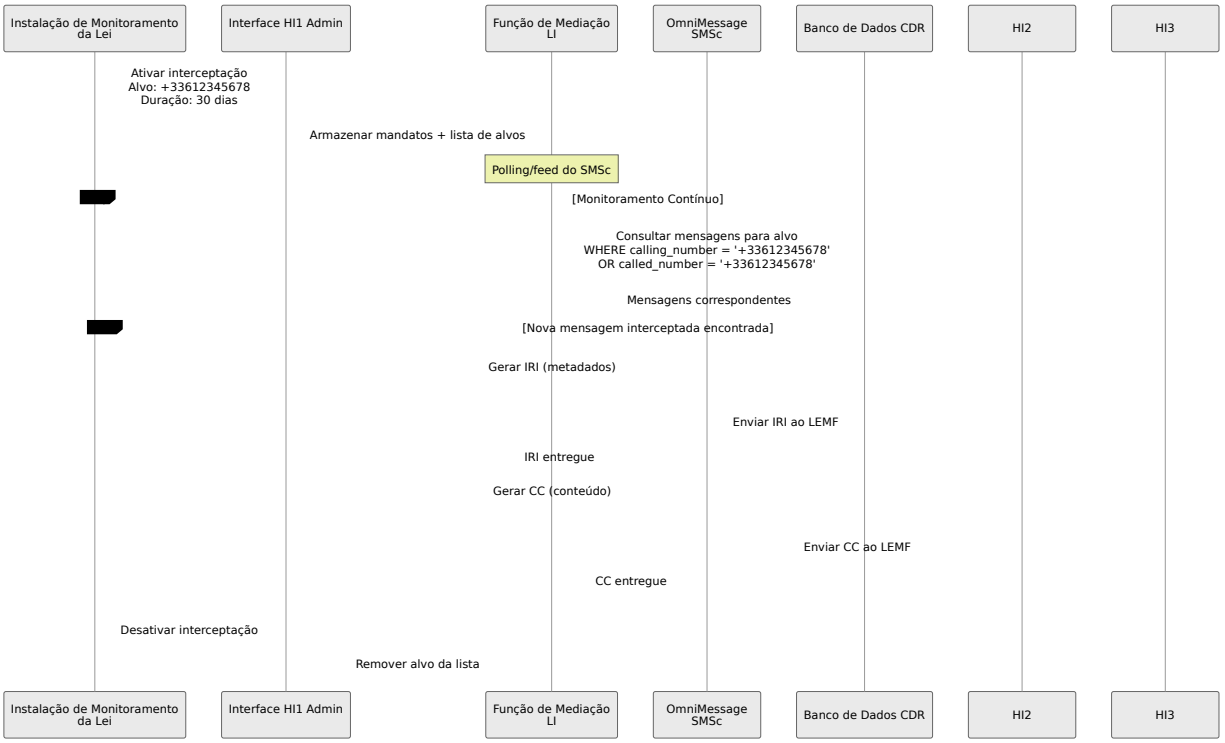
- **Propósito:** Entregar metadados de mensagens para a aplicação da lei
- **Direção:** LIMF → LEMF (unidirecional)
- **Formato de Dados:** XML/ASN.1 compatível com ETSI TS 102 232-x
- **Conteúdo do CDR do SMSc:**
  - ID da Mensagem
  - Número chamador (MSISDN de origem)

    - Número chamado (MSISDN de destino)
    - IMSI (origem e destino, se disponível)
    - Timestamp de submissão
    - Timestamp de entrega
    - Status da mensagem (entregue/falhada/expirada)
    - Tentativas de entrega
    - Informações do gateway SMSC (origem/destino)
    - Localizaç��o da rede (se disponível)
- **Integração com SMSc:**
    - LIMF consulta o banco de dados CDR para números de telefone alvo
    - LIMF transforma registros CDR no formato IRI da ETSI
    - LIMF entrega IRI ao LEMF via HI2

Interface HI3 - Entrega de CC (Conteúdo da Comunicação):

- **Propósito:** Entregar o conteúdo real da mensagem para a aplicação da lei
- **Direção:** LIMF → LEMF (unidirecional)
- **Formato de Dados:** Compatível com ETSI TS 102 232-x
- **Conteúdo do SMSc:**
    - Corpo da mensagem (conteúdo de texto)
    - PDU bruto (dados SMS binários)
    - Informações de codificação de caracteres
    - Segmentos de mensagem multipart
    - Informações TP-DCS
    - Cabeçalho de Dados do Usuário (UDH)
- **Integração com SMSc:**
    - LIMF recupera o conteúdo da mensagem do campo `message_body` do CDR
    - LIMF recupera dados PDU brutos se disponíveis
    - LIMF empacota o conteúdo no formato CC da ETSI
    - LIMF entrega CC ao LEMF via HI3

Arquitetura de Implementação:



Mapeamento de Dados do SMSc para Interfaces LI:

Campo de Dados do SMSc	HI2 (IRI)	HI3 (CC)	Coluna da Tabela CDR
ID da Mensagem	□ ID de Correlação	□ Referência	`message_id`
Número Chamador	□ Parte A	-	`calling_number`
Número Chamado	□ Parte B	-	`called_number`
Tempo de Submissão	□ Timestamp	-	`submission_time`
Tempo de Entrega	□ Conclusão	-	`delivery_time`
Status	□ Resultado	-	`status`
Corpo da Mensagem	-	□ Conteúdo	`message_body`
PDU Bruto	-	□ Binário	(Mnesia/CDR)
SMSC de Origem	□ Elemento de rede	-	`source_smsc`
SMSC de Destino	□ Elemento de rede	-	`dest_smsc`
IMSI	□ ID do Assinante	-	(Via frontends)

Opções de Integração LIMF:

Opção 1: Arquitetura de Polling

- LIMF consulta periodicamente o banco de dados CDR (a cada 1-60 segundos)

- Consultas SQL filtram por números de telefone alvo da lista de mandatos HI1
- Baixa complexidade, fácil de implementar
- Pequeno atraso entre a entrega da mensagem e a entrega da LI

Opção 2: Arquitetura de Feed em Tempo Real

- PubSub do SMSc publica eventos de mensagens
- LIMF se inscreve no fluxo de mensagens em tempo real
- LIMF filtra com base na lista de alvos
- Latência quase zero para interceptação legal
- Requer desenvolvimento de integração personalizada

Opção 3: Arquitetura Híbrida

- Mensagens recentes: Feed em tempo real PubSub (< 24 horas)
- Mensagens históricas: Polling do banco de dados CDR
- Equilíbrio ideal entre latência e confiabilidade

Mecanismos de Gatilho de Interceptação

Interceptação Baseada em Alvo:

- Correspondência de número de telefone (MSISDN)
- Alvo baseado em IMSI (quando disponível)
- Listas de vigilância configuráveis
- Visões de banco de dados para isolamento de alvos
- Filtragem de API por identificadores de alvos

Interceptação Baseada em Evento:

- Todas as mensagens para/de números específicos
- Mensagens via gateways SMSC específicos
- Mensagens com características específicas (multipart, entrega falhada, etc.)
- Roteamento geográfico (via ENUM ou correspondência de prefixo)

Interceptação Baseada em Tempo:

- Filtragem de intervalo de data/hora em consultas CDR
- Aplicação de período de retenção
- Arquivamento automático de mensagens antigas
- Políticas de retenção de dados configuráveis

Exemplos de Consultas SQL para Interceptação Legal:

```sql
-- Obter todas as mensagens para o número alvo
SELECT * FROM cdrs
WHERE calling_number = '+33612345678'
   OR called_number = '+33612345678'
ORDER BY submission_time DESC;

-- Obter mensagens em uma janela de tempo 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;

-- Obter conversa entre duas 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 ENCRIPTAÇÃO E CRIPTOANÁLISE

2.1 Visão Geral das Capacidades Criptográficas

O OmniMessage SMSc implementa mecanismos criptográficos para garantir comunicações e proteger dados sensíveis. Esta seção documenta todas as capacidades criptográficas de acordo com os requisitos da ANSSI.

2.2 Criptografia da Camada de Transporte

2.2.1 Implementação TLS/SSL

Protocolos Suportados:

- TLS 1.2 (RFC 5246)
- TLS 1.3 (RFC 8446) - Recomendado
- SSL 2.0/3.0: NÃO SUPORTADO (vulnerabilidades conhecidas)
- TLS 1.0/1.1: DESATIVADO (não recomendado)

Implementação:

- Biblioteca SSL/TLS Erlang/OTP (validada criptograficamente)
- Servidor web Cowboy com suporte a TLS
- Endpoints HTTPS do Phoenix Framework

Conjuntos de Cifras:

O sistema usa a seleção padrão de conjuntos de cifras seguras do Erlang/OTP, que inclui:

Preferido - TLS 1.3:

- TLS_AES_256_GCM_SHA384
- TLS_AES_128_GCM_SHA256
- TLS_CHACHA20_POLY1305_SHA256

Suportado - TLS 1.2:

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

Recursos de Segurança:

- Perfeita Segurança de Chave Futura (PFS) via troca de chaves ECDHE/DHE
- Grupos de Diffie-Hellman fortes (mínimo de 2048 bits)
- Suporte a Criptografia de Curva Elíptica
- Suporte a Indicação de Nome do Servidor (SNI)

Gerenciamento de Certificados:

- Suporte a certificados X.509
- Tamanhos de chave RSA: mínimo de 2048 bits, recomendado 4096 bits
- Suporte a ECDSA
- Validação da cadeia de certificados
- Certificados autoassinados (apenas para desenvolvimento)
- Integração com CA externa

Localização da Configuração TLS:

```
# config/runtime.exs
config :api_ex,
  api: %&#123;
    enable_tls: true,
    tls_cert_path: "priv/cert/omnitouch.crt",
    tls_key_path: "priv/cert/omnitouch.pem"
  &#125;
```

□ Referência completa de configuração em CONFIGURATION.md

Aplicações:

- HTTPS para API REST (porta 8443)

- HTTPS para painel de controle web (porta 8086)
- Conexões de banco de dados (MySQL/PostgreSQL sobre TLS)

2.3 Criptografia de Dados em Repouso

2.3.1 Criptografia de Banco de Dados

Criptografia MySQL/MariaDB:

- Suporte à criptografia em nível de tabela
- Algoritmo de criptografia AES-256
- Criptografia de dados transparente (TDE)

```
-- Habilitar criptografia para a tabela CDR
ALTER TABLE cdrs ENCRYPTION='Y';
```

Criptografia PostgreSQL:

- Suporte à criptografia de dados transparente
- Criptografia em nível de sistema de arquivos
- Criptografia em nível de coluna (extensão pgcrypto)

2.3.2 Armazenamento em Disco Mnesia

Banco de Dados Mnesia:

- Armazenamento de cópias em disco para persistência de mensagens
- Criptografia em nível de sistema de arquivos recomendada (LUKS, dm-crypt)
- Proteção de memória via isolamento da VM Erlang

2.3.3 Criptografia de Sistema de Arquivos

Armazenamento de Dados Sensíveis:

- Arquivos de configuração: Criptografia de sistema de arquivos recomendada

- Chaves privadas: Permissões de arquivo (0600) + criptografia de sistema de arquivos
- Arquivos de log: Criptografia configurável para logs arquivados
- Exportações CDR: Armazenamento criptografado para exportações sensíveis

Armazenamento de Chaves:

- Certificados e chaves TLS armazenados em `priv/cert/`
- Armazenamentos de chaves baseados em arquivos com permissões restritas
- Procedimentos seguros de rotação de chaves

2.4 Autenticação e Controle de Acesso

2.4.1 Autenticação da API

Segurança da API REST:

- Criptografia de transporte HTTPS/TLS obrigatória
- Autenticação baseada em cabeçalho (cabeçalho SMSc para identificação do frontend)
- Controle de acesso baseado em IP (nível de firewall)
- Autenticação de cliente baseada em certificado (opcional)

Registro de Frontend:

- Identificação única do frontend (nome, tipo, IP, hostname)
- Autenticação baseada em heartbeat
- Gerenciamento de sessão baseado em expiração (timeout de 90 segundos)
- Rastreamento e monitoramento do frontend

2.4.2 Autenticação de Banco de Dados

Controle de Acesso ao Banco de Dados:

- Autenticação por nome de usuário/senha
- Suporte a conexão TLS/SSL

- Restrições de conexão baseadas em IP
- Controle de acesso baseado em funções (RBAC)

Configuração:

```
# config/runtime.exs
config :sms_c, SmsC.Repo,
  username: "omnitouch",
  password: "omnitouch2024",  # Deve usar senhas fortes em
produção
  hostname: "localhost",
  ssl: true  # Habilitar TLS para conexões de banco de dados
```

Recomendações de Controle de Acesso:

```
-- Criar usuário somente leitura para acesso da aplicação da lei
CREATE USER 'li_readonly'@'%' IDENTIFIED BY 'secure_password';
GRANT SELECT ON sms_c.cdrs TO 'li_readonly'@'%';

-- Criar usuário limitado sem acesso ao corpo da mensagem
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 Detalhes dos Algoritmos Criptográficos

2.5.1 Algoritmos de Hashing

Disponíveis no Erlang/OTP:

- SHA-256, SHA-384, SHA-512 (recomendados)
- SHA-1 (descontinuado, compatibilidade legada apenas)
- MD5 (descontinuado, não usado para segurança)
- BLAKE2 (disponível em versões modernas do OTP)

Uso:

- Impressão digital de mensagens (detecção de duplicatas)
- Verificação de integridade de dados
- Integridade de logs de auditoria

2.5.2 Criptografia Simétrica

Algoritmos Disponíveis:
- AES (Padrão de Criptografia Avançada)
  - AES-128-GCM
  - AES-256-GCM
  - AES-128-CBC
  - AES-256-CBC
- ChaCha20-Poly1305

Tamanhos de Chave:
- 128 bits (mínimo)
- 256 bits (recomendado)

Uso:
- Criptografia de sessão TLS
- Criptografia de banco de dados em repouso
- Criptografia opcional do corpo da mensagem

2.5.3 Criptografia Assimétrica

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

Uso:
- Autenticação de certificado TLS
- Assinaturas digitais
- Troca de chaves

2.6 Segurança do Protocolo SMS

2.6.1 Codificação de Mensagens SMS

Suporte à Codificação de Caracteres:
- GSM 7-bit (codificação SMS padrão)
- UCS-2 (Unicode, 16-bit)
- Dados binários de 8 bits
- Latin-1

```
**TP-DCS (Esquema de Codificação de Dados):**
- Indicação da classe da mensagem
- Flags de compressão
- Especificação do grupo de codificação
- Identificação do conjunto de caracteres

**Sem Criptografia SMS Nativa:**
- O protocolo SMS não fornece criptografia de ponta a ponta
- Conteúdo da mensagem acessível no nível do SMSc
- Permite interceptação legal conforme necessário

#### 2.6.2 Considerações de Segurança do Protocolo

**Protocolo SMPP (Frontend Externo):**
- Autenticação de nome de usuário/senha no nível SMPP
- Suporte a TLS disponível (SMPP sobre TLS)
- Autenticação de bind

**Protocolo IMS (Frontend Externo):**
- Mensagens baseadas em SIP
- Mecanismos de autenticação SIP
- Integração com a segurança da rede central IMS

**Protocolo SS7/MAP (Frontend Externo):**
- Segurança da rede SS7
- Autenticação do protocolo MAP
- Segurança da camada SCCP/TCAP

**Nota:** A segurança específica do protocolo é implementada em
gateways de frontend externos, não no núcleo do SMSc.

### 2.7 Capacidades de Criptoanálise e Avaliação de Segurança

#### 2.7.1 Ferramentas de Análise de Protocolo

**Capacidades de Depuração Integradas:**
- Sistema de registro abrangente
- Rastreio do fluxo de mensagens
- Registro de solicitações/respostas da API
- Registro de consultas ao banco de dados
- Rastreamento de erros e exceções

**Integração Externa:**
- Saída de registro padrão (stdout/arquivos)
```

- Suporte à captura PCAP para análise de rede
- Registro de consultas ao banco de dados para investigações forenses
- Exportação de métricas Prometheus

2.7.2 Considerações de Avaliação de Vulnerabilidades

Limitações Conhecidas:
- O protocolo SMS é inerentemente não criptografado (por design, permite interceptação legal)
- Credenciais do banco de dados em arquivos de configuração (deve usar gerenciamento de segredos)
- Suporte a certificados autoassinados (apenas para desenvolvimento/testes)

Recomendações de Fortalecimento de Segurança:
- Use suítes de cifra TLS fortes
- Implemente criptografia de conexão ao banco de dados
- Use gerenciamento de segredos externo (Vault, AWS Secrets Manager)
- Atualizações regulares de segurança para Erlang/OTP e dependências
- Restrições de firewall nas portas da API
- Lista de permissões de IP para acesso ao frontend

Teste de Segurança:
- Escaneamento regular de vulnerabilidades de dependências
- Suporte a testes de penetração
- Validação da configuração TLS
- Auditorias de segurança do banco de dados
- Revisão de controle de acesso

2.8 Infraestrutura de Gerenciamento de Chaves

2.8.1 Geração de Chaves

Geração de Certificado TLS:
```bash
# Gerar chave privada (RSA 4096 bits)
openssl genrsa -out omnitouch.pem 4096

# Gerar solicitação de assinatura de certificado
openssl req -new -key omnitouch.pem -out omnitouch.csr
```

```
# Certificado autoassinado (desenvolvimento)
openssl x509 -req -days 365 -in omnitouch.csr -signkey
omnitouch.pem -out omnitouch.crt

# Produção: Obter certificado de CA confiável
```

Geração de Números Aleatórios:

- Erlang/OTP CSPRNG (Gerador de Números Pseudo-Aleatórios Criptograficamente Seguro)
- Pool de entropia do sistema (/dev/urandom)
- Aleatoriedade forte para chaves de sessão, IDs, tokens

2.8.2 Armazenamento e Proteção de Chaves

Armazenamento de Chave Privada:

- Sistema de arquivos com permissões restritas (0600)
- Armazenado no diretório `priv/cert/`
- Formato PEM (opcionalmente criptografado)
- Procedimentos de backup seguro

Rotação de Chaves:

- Renovação de certificado TLS (anualmente recomendado)
- Rotação de credenciais do banco de dados
- Rotação de tokens da API (se implementado)

2.8.3 Distribuição de Chaves

Distribuição de Certificados:

- Instalação manual em `priv/cert/`
- Referências em arquivos de configuração
- Suporte ao protocolo ACME possível (Let's Encrypt)

Distribuição de Chaves Simétricas:

- Troca de chaves fora de banda para credenciais de banco de dados

- Acordo de chave Diffie-Hellman em TLS
- Sem transmissão de chave em texto claro

2.9 Conformidade e Padrões

Esta seção documenta a conformidade com padrões internacionais de telecomunicações, estruturas regulatórias e especificações de segurança aplicáveis ao processamento de SMS em todos os protocolos suportados.

2.9.1 Conformidade do Protocolo SMS sobre SS7/MAP

Padrões 3GPP e ETSI:

- **3GPP TS 23.040:** Realização técnica do Serviço de Mensagem Curta (SMS) - Especificação do protocolo SMS central
- **3GPP TS 23.038:** Alfabetos e informações específicas de idioma - Codificação de caracteres (GSM7, UCS-2)
- **3GPP TS 29.002:** Especificação da Parte de Aplicação Móvel (MAP) - Sinalização SS7 para SMS
- **3GPP TS 23.003:** Numeração, endereçamento e identificação - Formatos MSISDN, IMSI
- **ETSI TS 100 901:** Especificação do Serviço de Mensagem Curta Ponto-a-Ponto
- **ETSI TS 100 902:** Especificação do Serviço de Mensagem Curta de Transmissão de Célula

Padrões de Sinalização SS7:

- **ITU-T Q.711-Q.716:** Parte de Controle de Conexão de Sinalização (SCCP)
- **ITU-T Q.771-Q.775:** Parte de Aplicação de Capacidades de Transação (TCAP)
- **ITU-T Q.701-Q.710:** Parte de Transferência de Mensagens (MTP) Níveis 1-3
- **ETSI EN 300 356:** Sistema de Sinalização No.7 - Parte do Usuário ISDN (ISUP)

Padrões de Segurança para SS7/MAP:

- **GSMA FS.07:** Segurança de Sinalização SS7 e Diameter - Controles de segurança básicos
- **GSMA FS.11:** Diretrizes de Monitoramento de Segurança SS7
- **3GPP TS 33.117:** Catálogo de requisitos gerais de garantia de segurança
- **ETSI TS 133 210:** Segurança do domínio da rede - Segurança da camada de rede IP

Interceptação Legal para SS7/MAP:

- **ETSI TS 101 671:** Interceptação Legal (LI); Interface de entrega para a interceptação legal do tráfego de telecomunicações
- **ETSI TS 102 232-1:** Interceptação Legal (LI); Especificação de entrega - Parte 1: Interface de entrega para gerenciamento de LI
- **3GPP TS 33.107:** Arquitetura e funções de Interceptação Legal para redes 3G

2.9.2 Conformidade do Protocolo SMS sobre IMS

Padrões IMS 3GPP:

- **3GPP TS 23.228:** Subsistema Multimídia IP (IMS) - Arquitetura de estágio 2
- **3GPP TS 24.229:** Protocolo de Controle de Chamadas Multimídia IP - Procedimentos SIP e SDP
- **3GPP TS 24.341:** Suporte de SMS sobre redes IP - Especificação de estágio 3
- **3GPP TS 23.204:** Suporte ao Serviço de Mensagem Curta (SMS) sobre acesso IP genérico 3GPP - Estágio 2
- **3GPP TS 29.228:** Interfaces Cx e Dx do Subsistema Multimídia (IM)

Padrões de Segurança IMS:

- **3GPP TS 33.203:** Segurança 3G; Segurança de acesso para serviços baseados em IP (IMS AKA)
- **3GPP TS 33.210:** Segurança 3G; Segurança do Domínio da Rede (NDS); Segurança da camada de rede IP (IPsec)

- **3GPP TS 33.310:** Segurança do Domínio da Rede (NDS); Estrutura de Autenticação (AF)
- **ETSI TS 133 203:** Segurança de acesso para serviços baseados em IP

Padrões do Protocolo SIP:

- **RFC 3261:** SIP: Protocolo de Iniciação de Sessão - Especificação central
- **RFC 3428:** Extensão SIP para Mensagens Instantâneas - Método MESSAGE
- **RFC 3325:** Extensões Privadas ao SIP para Identidade Afirmada
- **RFC 5765:** Questões de Segurança e Soluções em Sistemas Peer-to-Peer

Interceptação Legal para IMS:

- **ETSI TS 102 232-5:** Interceptação Legal (LI); Especificação de entrega - Parte 5: LI independente de serviço para serviços do Subsistema Multimídia IP
- **3GPP TS 33.107:** Requisitos e arquitetura de Interceptação Legal
- **3GPP TS 33.108:** Interface de entrega para Interceptação Legal (LI)

2.9.3 Conformidade do Protocolo SMPP

Especificação SMPP:

- **SMPP v3.4:** Especificação do Protocolo de Mensagem Curta Ponto-a-Ponto - Padrão da indústria
- **SMPP v5.0:** Protocolo SMPP estendido com recursos aprimorados

Diretrizes de Segurança SMPP:

- **TLS sobre SMPP:** Segurança da camada de transporte para conexões SMPP (SMPP sobre TLS)
- **Autenticação de Bind SMPP:** Autenticação de ID do sistema e senha
- **Controle de Acesso baseado em IP:** Restrições em nível de rede para conexões SMPP

Padrões de Interoperabilidade:

- **GSM 03.40 (ETSI TS 100 901):** Realização técnica do SMS Ponto-a-Ponto (PP)

- **GSM 03.38 (ETSI TS 100 900):** Alfabetos e informações específicas de idioma
- **GSM 04.11 (ETSI TS 100 942):** Suporte a SMS Ponto-a-Ponto na interface de rádio móvel

Conformidade de Codificação de Mensagens:

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

2.9.4 Conformidade com Padrões Criptográficos

TLS e Segurança de Rede:

- **NIST SP 800-52:** Diretrizes para a Seleção, Configuração e Uso de Implementações TLS
- **NIST SP 800-131A:** Transição do Uso de Algoritmos Criptográficos e Comprimentos de Chaves
- **RFC 7525:** Recomendações para Uso Seguro de TLS e DTLS
- **RFC 8446:** O Protocolo de Segurança da Camada de Transporte (TLS) Versão 1.3

Padrões de Algoritmos Criptográficos:

- **FIPS 197:** Padrão de Criptografia Avançada (AES)
- **FIPS 180-4:** Padrão de Hash Seguro (família SHA-2)
- **NIST SP 800-38D:** Recomendação para Modos de Operação de Cifra de Bloco: Modo GCM
- **RFC 7539:** ChaCha20 e Poly1305 para Protocolos IETF

Gerenciamento de Chaves:

- **NIST SP 800-57:** Recomendação para Gerenciamento de Chaves
- **RFC 5280:** Infraestrutura de Chave Pública X.509 da Internet Certificado e Perfil de CRL

2.10 Resistência à Criptoanálise

2.10.1 Princípios de Design

Defesa Contra Criptoanálise:

- Nenhum algoritmo criptográfico personalizado/proprietário
- Apenas algoritmos padrão da indústria, revisados por pares
- Atualizações regulares de segurança para bibliotecas criptográficas
- Depreciação de algoritmos fracos
- Uso de criptografia autenticada (GCM, Poly1305)

2.10.2 Segurança Operacional

Rotação de Chaves:

- Procedimentos de renovação de certificado TLS
- Rotação de chaves de sessão (por sessão para TLS)
- Políticas de rotação de credenciais de banco de dados

Monitoramento e Detecção:

- Registro de autenticações falhadas
- Monitoramento de expiração de certificados
- Registro de falhas de handshake TLS
- Detecção de anomalias para falhas de criptografia
- Alerta de eventos de segurança

---

3. CONTROLE DE INTERCEPTAÇÃO E AUTORIZAÇÃO

3.1 Controle de Acesso para Interceptação Legal

Autorização Administrativa:

- Acesso de administrador do sistema necessário para configuração
- Controles de acesso em nível de banco de dados para consultas CDR
- Acesso à API restrito por IP/autenticação
- Registro de auditoria de todos os acessos

Integração com Estrutura Legal:

- Rastreamento de mandados de interceptação (integração com sistema externo)
- Listas de autorização de identificadores de alvo (visões de banco de dados)
- Consultas com limite de tempo (cláusulas WHERE SQL)
- Aplicação automática via políticas de acesso

3.2 Retenção de Dados e Privacidade

Políticas de Retenção:

- Retenção de mensagens ativas: Configurável (padrão 24 horas no Mnesia)
- Retenção de CDR: Configurável (típico de 6 meses a 2 anos)
- Arquivamento automático do Mnesia para SQL
- Purgação automática de CDRs antigos (baseado em cron)

Proteções de Privacidade:

- Opção de exclusão do corpo da mensagem após a entrega
- Exclusão do corpo da mensagem da UI/exportações
- Criptografia de banco de dados em repouso
- Registro e monitoramento de acesso
- Princípio de coleta mínima de dados

Configuração:

```
# config/runtime.exs
config :sms_c,
  # Retenção de mensagens Mnesia antes do arquivamento
  message_retention_hours: 24,

  # Excluir corpo da mensagem após entrega por privacidade
  delete_message_body_after_delivery: false,  # Defina como true
para modo de privacidade

  # Controle de gravação de CDR
  cdr_enabled: true,

  # Configurações de arquivamento em lote
  batch_insert_batch_size: 100,
  batch_insert_flush_interval_ms: 100
```

 Veja CONFIGURATION.md para todas as configurações de retenção

3.3 Interfaces de Entrega para Aplicação da Lei

Interfaces Padrão:

1. Acesso à API REST:

- Endpoints HTTPS para recuperação de mensagens
- Troca de dados em formato JSON
- Autenticação e autorização
- Filtragem de consultas por critérios de alvo

2. Acesso Direto ao Banco de Dados:

- Credenciais SQL somente leitura
- Consultas SQL padrão
- Acesso à tabela CDR
- Capacidades de pesquisa indexada

3. Exportação em Lote:

- Formato de exportação CSV

- Formato de exportação JSON
- Exportações baseadas em intervalo de tempo
- Seleção de campo configurável

Formatos de Entrega:

IRI (Informação Relacionada à Interceptação):

- Campos de metadados CDR:
  - ID da mensagem
  - Números chamadores/chamados
  - Carimbos de data/hora (submissão, entrega, expiração)
  - Status
  - Tentativas de entrega
  - Informações de roteamento SMSC
  - Informações do nó (rastreamento de cluster)

CC (Conteúdo da Comunicação):

- Corpo da mensagem (conteúdo de texto)
- Dados PDU brutos
- Informações de codificação
- Montagem de mensagem multipart

Exemplo de Exportação:

```
# Exportação CSV para aplicação da lei
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. SEGURANÇA E INTEGRIDADE DO SISTEMA

4.1 Segurança da Aplicação

Segurança Elixir/Erlang:

- Isolamento e sandboxing da VM Erlang
- Isolamento e supervisão de processos
- Recuperação de falhas e tolerância a falhas
- Sem vulnerabilidades de estouro de buffer (tempo de execução gerenciado)

Gerenciamento de Dependências:

- Bloqueio de versão de dependências (mix.lock)
- Escaneamento de vulnerabilidades de segurança
- Atualizações regulares de dependências
- Pegada mínima de dependências

4.2 Segurança de Rede

Exposição de Rede:

- Portas expostas mínimas:
  - 8443 (API REST HTTPS)
  - 8086 (Painel de Controle HTTPS)
- Configuração de firewall recomendada
- Lista de permissões de IP para acesso ao frontend
- Implantação DMZ para serviços voltados para a internet

Segmentação de Rede:

- Rede de gerenciamento separada
- Rede de banco de dados isolada
- Separação da rede de gateway do frontend
- Rede de comunicação de cluster (distribuição Erlang)

4.3 Monitoramento e Detecção de Intrusões

Capacidades de Registro:

- Registro estruturado da aplicação
- Níveis de registro configuráveis
- Rotação e arquivamento de registros
- Suporte à integração Syslog
- Registro centralizado (compatível com ELK stack)

Monitoramento de Eventos de Segurança:

- Tentativas de autenticação falhadas
- Padrões de mensagens incomuns
- Falhas de conexão ao banco de dados
- Falhas de handshake TLS
- Anomalias de recursos do sistema

Métricas e Alertas:

- Exportação de métricas Prometheus
- Monitoramento de taxa de mensagens
- Rastreamento de taxa de erros
- Utilização de recursos do sistema
- Regras de alerta personalizadas

Documentação completa de monitoramento em OPERATIONS_GUIDE.md e METRICS.md

4.4 Alta Disponibilidade e Recuperação de Desastres

Suporte a Cluster:

- Capacidade de cluster distribuído Erlang
- Replicação do Mnesia entre nós
- Failover automático
- Descoberta e junção de nós

Redundância de Dados:

- Cópias de disco do Mnesia em todos os nós do cluster
- Replicação de banco de dados SQL (nativa do MySQL/PostgreSQL)
- Procedimentos de backup de CDR
- Backup de configuração

Procedimentos de Recuperação:

- Backup e restauração de banco de dados
- Recuperação de tabelas do Mnesia
- Restauração de configuração
- Procedimentos de substituição de nós

---

5. REFERÊNCIAS DE DOCUMENTAÇÃO

5.1 Manuais Técnicos

Documentação disponível no repositório do projeto:

- **README.md** - Visão geral do sistema, arquitetura e recursos
- **CONFIGURATION.md** - Referência completa de configuração
- **API_REFERENCE.md** - Documentação da API REST
- **OPERATIONS_GUIDE.md** - Procedimentos operacionais e monitoramento
- **CDR_SCHEMA.md** - Esquema do banco de dados de Registro de Detalhes de Chamadas
- **sms_routing_guide.md** - Configuração de roteamento SMS
- **number_translation_guide.md** - Normalização de números
- **METRICS.md** - Métricas e monitoramento Prometheus
- **TROUBLESHOOTING.md** - Problemas comuns e soluções

5.2 Certificações de Segurança

- **Relatórios de Testes de Penetração:** [A serem fornecidos mediante solicitação]
- **Relatórios de Auditoria de Segurança:** [A serem fornecidos mediante solicitação]
- **Avaliações de Vulnerabilidades:** [A serem fornecidos mediante solicitação]
- **Validação Criptográfica Erlang/OTP:** Biblioteca criptográfica padrão da indústria

5.3 Documentação de Conformidade

- **Solicitação de Autorização ANSSI R226:** Este documento
- **Conformidade com Interceptação Legal:** Conforme exigido pela regulamentação de telecomunicações francesa

- **Conformidade com Proteção de Dados:** Considerações GDPR para dados de mensagens

---

6. INFORMAÇÕES DE CONTATO

Informações do Fornecedor/Operador:

- Nome da Empresa: Omnitouch Network Services Pty Ltd
- Endereço: PO BOX 296, QUINNS ROCKS WA 6030, AUSTRÁLIA
- Pessoa de Contato: Equipe de Conformidade
- Email: compliance@omnitouch.com.au

Contato de Segurança Técnica:

- Nome: Equipe de Conformidade
- Email: compliance@omnitouch.com.au

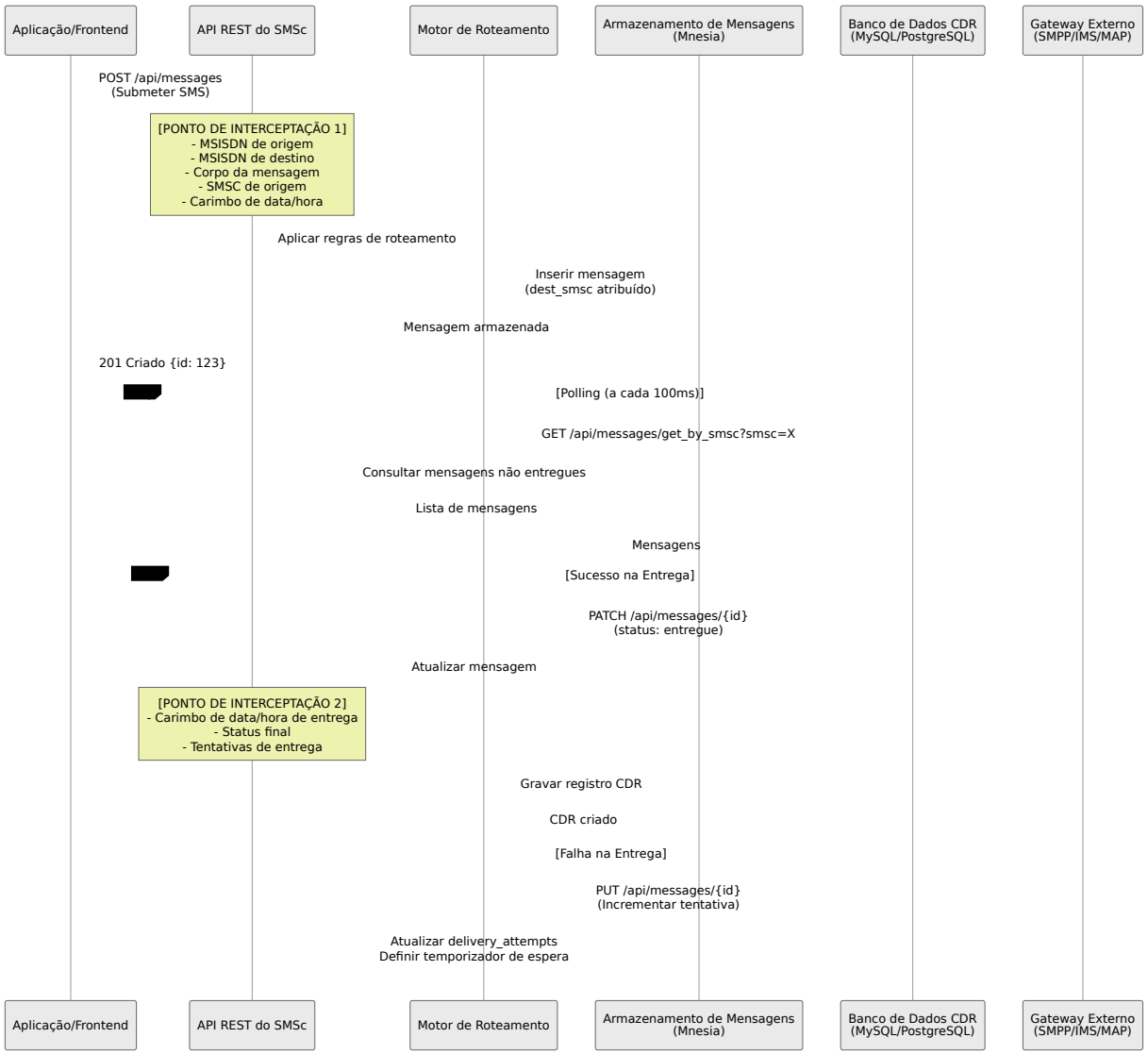
Contato Legal/Conformidade:

- Nome: Equipe de Conformidade
- Email: compliance@omnitouch.com.au

APÊNDICES

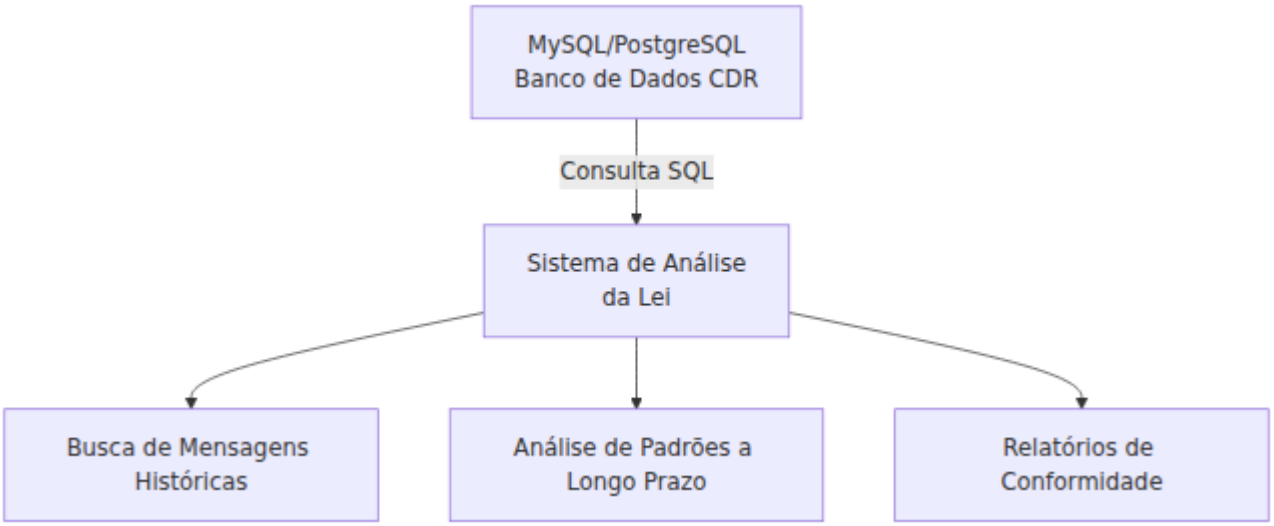
Apêndice A: Fluxo de Mensagens SMS com Pontos de Interceptação

A.1 Fluxo de SMS de Saída (Terminado em Móvel)



Legenda: `[PONTO DE INTERCEPTAÇÃO]` = Pontos onde os dados de interceptação legal são capturados e armazenados

A.2 Fluxo de SMS de Entrada (Originado em Móvel)



Apêndice B: Esquema CDR para Interceptação Legal

O OmniMessage SMSc armazena Registros de Detalhes de Chamadas em um banco de dados SQL (MySQL ou PostgreSQL) para retenção de longo prazo e acesso à interceptação legal.

B.1 Campos Chave CDR para Interceptação Legal

Nome do Campo	Tipo	Descrição
`id`	BIGINT	Chave primária auto-incrementada
`message_id`	BIGINT	Identificador único da mensagem
`calling_number`	VARCHAR(255)	MSISDN de origem
`called_number`	VARCHAR(255)	MSISDN de destino
`source_smsc`	VARCHAR(255)	Identificador do gateway de origem
`dest_smsc`	VARCHAR(255)	Identificador do gateway de destino
`origin_node`	VARCHAR(255)	Nó de cluster Erlang (origem)
`destination_node`	VARCHAR(255)	Nó de cluster Erlang (entrega)
`submission_time`	DATETIME	Carimbo de data/hora de submissão da mensagem
`delivery_time`	DATETIME	Carimbo de data/hora de entrega da mensagem

Nome do Campo	Tipo	Descrição
`expiry_time`	DATETIME	Carimbo de data/hora de expiração da mensagem
`status`	VARCHAR(50)	Status da mensagem (entregue/expirado/falhado/rejeitado)
`delivery_attempts`	INT	Número de tentativas de entrega
`message_parts`	INT	Número de segmentos de SMS
`deadletter`	BOOLEAN	Flag de fila de cartas mortas
`message_body`	TEXT	Conteúdo da mensagem SMS
`inserted_at`	DATETIME	Carimbo de data/hora de criação do CDR
`updated_at`	DATETIME	Carimbo de data/hora de atualização do CDR

📄 Documentação completa do esquema com exemplos SQL em CDR_SCHEMA.md

B.2 Exemplos de Consulta CDR para Interceptação Legal

Consultar todas as mensagens para o número alvo:

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

Consultar mensagens dentro da janela de tempo:

```
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 para CSV para a aplicação da lei:

```
.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 Acesso ao Banco de Dados CDR

1. Acesso SQL Direto:

- Credenciais de banco de dados somente leitura
- Consultas SQL padrão
- Conectividade JDBC/ODBC
- Ferramentas de cliente de banco de dados (MySQL Workbench, pgAdmin)

2. Acesso via API REST:

- Aprimoramento futuro: API REST para consultas CDR
- Respostas em formato JSON

- Autenticação e autorização
- Filtragem de parâmetros de consulta

3. Exportação em Lote:

- Exportação CSV via linha de comando mysql/psql
- Scripts de exportação automatizados
- Exportações agendadas via cron

B.4 Retenção e Privacidade CDR

Configuração de Retenção:

```
# config/runtime.exs
config :sms_c,
  # Excluir corpo da mensagem após a entrega (modo de privacidade)
  delete_message_body_after_delivery: false,  # true para
privacidade

  # Ativar/desativar gravação de CDR
  cdr_enabled: true,

  # Configurações de arquivamento de Mnesia para CDR
  message_retention_hours: 24
```

Opções de Privacidade:

- O corpo da mensagem pode ser definido como NULL após a entrega
- Criptografia de tabela de banco de dados (MySQL ENCRYPTION='Y')
- Restrições de acesso em nível de coluna
- Exportações mascaradas para análises

Apêndice C: Referência da API REST para Interceptação

C.1 Endpoints de Recuperação de Mensagens

Obter todas as mensagens:

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

Obter mensagens por SMSC:

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

Obter mensagem específica:

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

Formato de Resposta:

```
{
  "status": "success",
  "data": [
    {
      "id": 12345,
      "message_id": 12345,
      "source_msisdn": "+33612345678",
      "destination_msisdn": "+33687654321",
      "message_body": "Conteúdo da mensagem aqui",
      "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"
    }
  ]
}
```

📖 Documentação completa da API em API_REFERENCE.md

C.2 Filtragem e Pesquisa

Parâmetros de Consulta:

- `source_smsc` - Filtrar por gateway de origem
- `dest_smsc` - Filtrar por gateway de destino
- Filtragem por intervalo de tempo (via consultas ao banco de dados CDR)
- Filtragem por status (via consultas ao banco de dados CDR)

Aprimoramentos Futuros:

- API REST para consultas CDR
- Filtragem avançada por número de telefone
- Filtragem por intervalo de datas
- Pesquisa de texto completo no corpo da mensagem

Apêndice D: Exemplos de Configuração

D.1 Configuração de Certificado TLS

Gerar Certificado TLS:

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

# Gerar solicitação de assinatura 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 autoassinado (desenvolvimento/teste)
openssl x509 -req -days 365 -in priv/cert/omnitouch.csr \
  -signkey priv/cert/omnitouch.pem -out priv/cert/omnitouch.crt

# Produção: Enviar CSR para CA confiável para assinatura
```

Configurar na aplicação:

```
# 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 Configuração de Criptografia de Banco de Dados

Criptografia de Tabela MySQL:

```
-- Ativar criptografia para a tabela CDR
ALTER TABLE cdrs ENCRYPTION='Y';

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

Criptografia de Conexão 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 Configuração de Privacidade

Ativar Exclusão do Corpo da Mensagem:

```
# config/runtime.exs
config :sms_c,
  # Excluir corpo da mensagem após entrega bem-sucedida
  delete_message_body_after_delivery: true,

  # Ocultar corpo da mensagem na interface do usuário
  hide_message_body_in_ui: true,

  # Ocultar corpo da mensagem nas exportações
  hide_message_body_in_export: true
```

Criar Visão de Banco de Dados que Preserve a Privacidade:

```
-- Criar visão sem corpos de mensagens para análises gerais
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 acesso a usuários de análise
GRANT SELECT ON cdrs_metadata TO 'analytics'@'%';
```

Apêndice E: Glossário

Órgãos Reguladores e Normas

- **ANSSI:** Agence nationale de la sécurité des systèmes d'information - Agência Nacional de Cibersegurança da França
- **ETSI:** European Telecommunications Standards Institute
- **3GPP:** 3rd Generation Partnership Project - Normas de telecomunicações móveis
- **IETF:** Internet Engineering Task Force - Órgão de normas da Internet

Termos de Telecomunicações

- **SMSc:** SMS Service Center - Sistema central para roteamento e entrega de mensagens SMS
- **SMPP:** Short Message Peer-to-Peer protocol - Padrão da indústria para troca de SMS
- **ESME:** External Short Message Entity - Aplicação cliente SMPP
- **IMS:** IP Multimedia Subsystem - Arquitetura de rede All-IP para serviços multimídia
- **SIP:** Session Initiation Protocol - Protocolo de sinalização para mensagens IMS
- **P-CSCF:** Proxy Call Session Control Function - Ponto de entrada da rede IMS
- **S-CSCF:** Serving Call Session Control Function - Controle de sessão IMS
- **HSS:** Home Subscriber Server - Banco de dados de assinantes IMS
- **SS7/MAP:** Signaling System 7 / Mobile Application Part - Protocolos de sinalização móvel legados
- **MSC:** Mobile Switching Center - Elemento de rede comutado por circuito
- **VLR:** Visitor Location Register - Banco de dados de localização de assinantes
- **GT:** Global Title - Esquema de endereçamento SS7
- **SCCP:** Signaling Connection Control Part - Camada de rede SS7
- **TCAP:** Transaction Capabilities Application Part - Camada de aplicação SS7
- **MSISDN:** Mobile Station International Subscriber Directory Number - Número de telefone
- **IMSI:** International Mobile Subscriber Identity - Identificador único de assinante
- **E.164:** Plano de numeração internacional para números de telefone
- **ENUM:** E.164 Number Mapping - Mapeamento de número de telefone baseado em DNS para URI
- **PDU:** Protocol Data Unit - Mensagem SMS codificada em binário
- **TP-DCS:** Transfer Protocol Data Coding Scheme - Especificação de codificação SMS
- **TP-DU:** Transfer Protocol Data Unit - Formato PDU específico de SMS
- **UDH:** User Data Header - Cabeçalho para mensagens SMS concatenadas/especiais

- **TON/NPI:** Type of Number / Numbering Plan Indicator - Classificação de formato de número
- **GSM7:** GSM 7-bit default alphabet - Codificação de caracteres SMS padrão
- **UCS-2:** Universal Character Set 2-byte - Codificação Unicode para SMS

Componentes do Sistema

- **Mnesia:** Sistema de banco de dados distribuído Erlang - Armazenamento em memória/disco
- **CDR:** Call Detail Record - Registro de cobrança e análise para mensagens
- **REST API:** Representational State Transfer - Arquitetura de API baseada em HTTP
- **Phoenix:** Framework web Elixir
- **Cowboy:** Servidor HTTP Erlang
- **Ecto:** Wrapper de banco de dados Elixir e linguagem de consulta
- **PubSub:** Padrão de mensagens Publish-Subscribe

Interceptação Legal

- **LI:** Lawful Interception - Monitoramento legal de telecomunicações
- **LIMF:** Lawful Interception Mediation Function - Sistema que interage entre a rede de telecomunicações e a aplicação da lei
- **LEMF:** Law Enforcement Monitoring Facility - Sistema de aplicação da lei que recebe dados interceptados
- **IRI:** Intercept Related Information - Metadados de chamada/mensagem para a aplicação da lei
- **CC:** Content of Communication - Conteúdo real da mensagem
- **HI1 Interface:** ETSI LI administrative handover interface - Interface de provisão de mandados e ativação de alvos
- **HI2 Interface:** ETSI LI handover interface for IRI delivery - Transferência de metadados para a aplicação da lei
- **HI3 Interface:** ETSI LI handover interface for CC delivery - Transferência de conteúdo para a aplicação da lei
- **R226:** Artigos R226-3 e R226-7 do Código Penal Francês que regem equipamentos de interceptação

- **ETSI:** European Telecommunications Standards Institute - Define padrões de LI
- **ETSI TS 102 232:** Especificação técnica para interfaces de transferência de interceptação legal

Processamento de Mensagens

- **MT:** Mobile Terminated - Mensagem de saída para assinante móvel
- **MO:** Mobile Originated - Mensagem de entrada de assinante móvel
- **DLR:** Delivery Receipt - Confirmação de entrega da mensagem
- **Dead Letter:** Mensagem que falhou na entrega após todas as tentativas de reenvio
- **Exponential Backoff:** Aumento do atraso de reenvio (2min, 4min, 8min, etc.)

Segurança e Criptografia

- **TLS:** Transport Layer Security - Protocolo de criptografia
- **PFS:** Perfect Forward Secrecy - Propriedade criptográfica para segurança da chave de sessão
- **AES:** Advanced Encryption Standard
- **RSA:** Rivest-Shamir-Adleman - Criptografia de chave pública
- **ECDSA:** Elliptic Curve Digital Signature Algorithm
- **SHA:** Secure Hash Algorithm
- **X.509:** Padrão de certificado
- **CA:** Certificate Authority
- **CSPRNG:** Gerador de Números Pseudo-Aleatórios Criptograficamente Seguro

Termos de Banco de Dados

- **MySQL:** Banco de dados relacional de código aberto
- **PostgreSQL:** Banco de dados objeto-relacional de código aberto
- **TDE:** Transparent Data Encryption
- **RBAC:** Controle de Acesso Baseado em Funções

Versão do Documento: 1.0
Data: 2025-11-29
Preparado para: Aplicação de Autorização ANSSI R226
Classificação do Documento: Conformidade Regulatória - Confidencial

Mensagens em Massa / Campanhas

← Voltar ao Índice da Documentação

Visão Geral

O recurso de campanhas envia uma única mensagem de texto simples para muitos destinatários enquanto protege o SMSC de sobrecarga. Você faz o upload de uma **lista de destinatários** nomeada uma vez, e então executa uma ou mais **campanhas** contra ela. Cada campanha distribui mensagens na fila normal de mensagens a uma taxa configurável e relata o progresso ao vivo e estatísticas de entrega.

Dois objetos reutilizáveis:

Objeto	O que é
Lista de Campanha	Um conjunto nomeado e reutilizável de MSISDNs de destinatários, cada um com até quatro variáveis opcionais. Faça o upload uma vez (array JSON ou CSV), reutilize em campanhas.
Campanha	Uma execução de envio: uma lista (ou *todos os assinantes ativos*) + um modelo de mensagem, distribuído na fila.

Tudo está exposto através da API HTTP. Todo o estado é armazenado no Mnesia junto com o restante do SMSC.

Segmentação: listas e o padrão de assinante ativo

Os destinatários de uma campanha vêm de uma das duas fontes:

- **Uma lista de destinatários** (`list_id`) - os MSISDNs que você fez o upload.
- **Nenhuma lista** - se você criar uma campanha **sem** um `list_id`, ela tem como alvo **todos os assinantes atualmente ativos (registrados)** - cada MSISDN com um registro não expirado no armazenamento de localização no momento em que a campanha começa. Esta é a maneira mais fácil de alcançar "todos na rede agora".

Variáveis por destinatário e modelagem

Cada destinatário da lista pode carregar até quatro variáveis - **`var1`**, **`var2`**, **`var3`**, **`var4`** - para coisas como primeiro nome, sobrenome ou nome do plano. A mensagem da campanha é um **modelo**: `{{ var1 }}`…`{{ var4 }}` (e `{{ msisdn }}`) são inseridos para cada destinatário quando a mensagem é enviada. O espaço em branco dentro das chaves é opcional, e qualquer marcador de posição sem valor é renderizado como uma string vazia.

> A modelagem é deliberadamente mínima - substituição de variáveis apenas (sem condicionais ou filtros), então não há dependência extra e nenhuma surpresa no que os assinantes recebem.

Fazendo upload de variáveis via CSV - a coluna MSISDN é a nomeada `msisdn`/`destination_msisdn`/`number`/`phone` (ou a primeira coluna se nenhuma for nomeada); cada outra coluna se torna `var1`…`var4` na ordem das colunas:

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

Fazendo upload de variáveis via JSON - objetos de destinatários podem definir as variáveis diretamente:

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

Exemplo de modelo:

```
Oi {{ var1 }}, seu plano {{ var2 }} renova em breve. Responda STOP
para cancelar.
```

Para o destinatário `12025550101` acima (`var1=Alice`, `var2=Gold`) isso envia: `Oi Alice, seu plano Gold renova em breve. Responda STOP para cancelar.`

Filtro de audiência

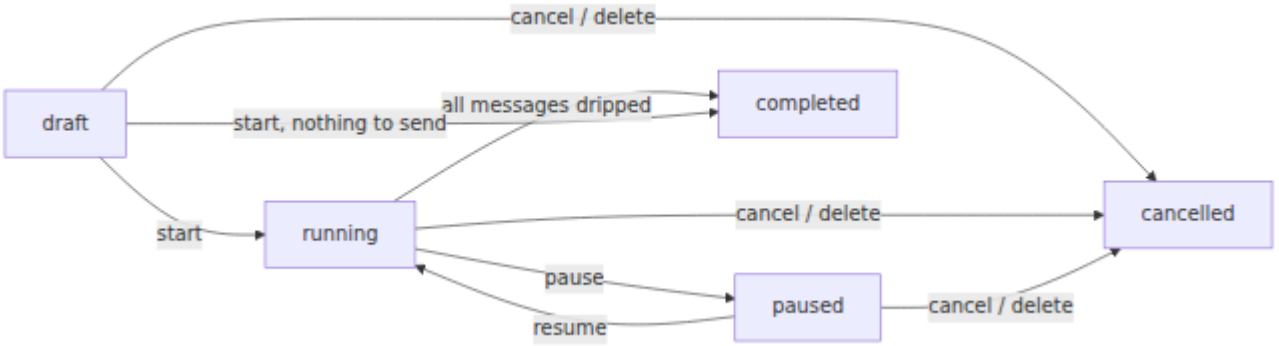
Quando uma campanha usa uma lista, a flag `audience` decide quem realmente recebe uma mensagem:

Audiência	Comportamento
`active_only` (padrão)	Apenas membros da lista que estão **registrados** (uma localização não expirada) no momento em que a campanha começa. Todos os outros são registrados como `skipped_inactive` e nunca recebem.
`all`	Todos os membros da lista, independentemente do registro.

A verificação de registro é um **instantâneo tirado no momento de início** - assinantes que se registram depois não são capturados. Quando uma campanha **não tem lista**, o conjunto de destinatários já é a base de

assinantes ativos, então `audience` é implicitamente `active_only` e qualquer valor fornecido é ignorado.

Ciclo de Vida



1. **draft** - criado, sem alvos ainda.
2. **start** - a campanha tira um instantâneo de seus alvos (de sua lista ou da base de assinantes ativos), então passa para **running**. Se não há nada para enviar, ela vai direto para **completed**.
3. **running** - o trabalhador de distribuição envia mensagens para a fila.
4. **pause / resume** - parar temporariamente / continuar a distribuição.
5. **completed** - cada alvo foi distribuído na fila (a entrega continua assíncronamente, e as estatísticas de entrega continuam atualizando).
6. **cancelled** - parado permanentemente.

Deletando uma campanha

Deletar uma campanha **a para, mas mantém o registro** - ela é cancelada em vez de excluída, então seu histórico e estatísticas finais permanecem visíveis. Campanhas paradas e completadas são removidas automaticamente uma vez que envelhecem além do TTL de retenção (`campaign_retention_hours`, padrão 7 dias).

Taxa de Distribuição (proteção contra sobrecarga)

Cada campanha tem um `drip_tps` (mensagens enviadas para a fila por segundo, padrão da configuração). Um trabalhador em segundo plano acorda em um intervalo fixo e envia no máximo `drip_tps × interval` mensagens por campanha por tick - então uma campanha de um milhão de destinatários nunca inunda a fila. A taxa é **por campanha**; várias campanhas em execução distribuem a sua própria taxa.

Progresso e estatísticas de entrega

Cada objeto de campanha inclui um bloco `stats` ao vivo (poll `GET /api/campaigns/:id`):

Campo	Significado
`total`	Alvos registrados no início
`pending`	Ainda não distribuídos na fila
`queued`	Enviados, aguardando um relatório de entrega
`delivered`	Confirmado como entregue
`failed`	Falha na entrega / enviado para carta morta
`skipped_inactive`	Ignorado porque não registrado no início
`dispatched`	`queued + delivered + failed`
`progress_percent`	Percentual de alvos que não estão mais `pending`
`delivery_percent`	Percentual de mensagens enviadas entregues

Contagens de entrega e falha atualizam automaticamente à medida que os relatórios de entrega fluem de volta pelo caminho normal de entrega do SMSC. Detalhes por destinatário (e filtragem por estado, por exemplo, todos que falharam) estão disponíveis via `GET /api/campaign_targets/:id?state=failed`.

Configuração

```
config :sms_c, :campaigns,
  # Iniciar o trabalhador de distribuição (defina como falso para
pausar TODA entrega de campanhas)
  enabled: true,
  # Intervalo entre ticks de distribuição em milissegundos
  drip_tick_ms: 1000,
  # Mensagens por segundo por campanha quando uma campanha não
define seu próprio drip_tps
  default_drip_tps: 50,
  # Janela de validade da mensagem padrão quando uma campanha não
define validity_hours
  message_validity_hours: 72,
  # Quanto tempo uma campanha parada/completada persiste antes de
ser excluída (horas)
  campaign_retention_hours: 168
```

Validade e entrega agendada

Uma campanha pode definir, no momento da criação:

- **`validity_hours`** - o período de validade da mensagem. Cada mensagem enviada é carimbada com um `expires` para que mensagens não entregues expirem em vez de permanecer na fila. Padrão é `message_validity_hours` da configuração.
- **`deliver_after`** - um horário ISO8601 opcional (deve estar no futuro) para um envio agendado.

Como a entrega agendada interage com a distribuição. A campanha ainda começa a distribuir *imediatamente* a `drip_tps`, mas o relógio de cada

mensagem é deslocado para frente por um deslocamento fixo `delta = deliver_after − started_at`. Assim, uma mensagem distribuída em tempo real *t* é carimbada `send_time`/`deliver_after = t + delta` e `expires = t + delta + validity_hours`. Como as mensagens são inseridas a `drip_tps` em tempo real, seus valores `deliver_after` por mensagem saem **escalonados a `drip_tps` começando em `deliver_after`** - a entrega é limitada por taxa em vez de um rebanho estrondoso no instante agendado. Sem `deliver_after`, `delta = 0` e as mensagens são distribuídas e entregues imediatamente.

Fluxo de trabalho típico

```
# 1. Faça o upload de uma lista de destinatários nomeada com variávei
destinatário (CSV ou 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. Crie uma campanha contra essa lista, modelando as variáveis
curl -k -X POST https://smsc:8443/api/campaigns \
  -H 'Content-Type: application/json' \
  -d '{"name":"Aviso de Renovação","message":"Oi {{ var1 }}, seu {{ v
plano renova em breve.",
       "source_msisdn":"12345","source_smsc":"IMS_SMSC",
       "list_id":1,"audience":"active_only","drip_tps":50}'
# => {"id": 7, "status": "draft", ...}

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

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

Para enviar mensagens **a todos os assinantes ativos**, omita `list_id` na etapa 2 (e pule a etapa 1 completamente).

Veja também

- Referência da API de Campanhas
- Guia de Roteamento de SMS - como mensagens distribuídas são roteadas/entregues
- Referência de Configuração

Referência do Esquema CDR (Registro de Detalhes da Chamada)

← Voltar ao Índice da Documentação

Referência completa para a tabela do banco de dados CDR usada para armazenamento de mensagens a longo prazo, faturamento e análises.

Índice

- Visão Geral
- Esquema da Tabela
- Descrições dos Campos
- Exemplos de SQL
- Índices
- Tipos de Dados por Banco de Dados
- Considerações de Privacidade
- Retenção e Arquivamento
- Integração de Faturamento

Visão Geral

A tabela `cdrs` armazena Registros de Detalhes de Chamadas para todas as mensagens SMS que o sistema processa. O sistema escreve um CDR no momento em que uma mensagem atinge um estado terminal: `entregue`, `expirado`, `falhou` ou `rejeitado`. O sistema registra um `balance_rejected` fora de balanço como `rejeitado`. Uma varredura de tempo de retenção atua como um suporte para qualquer coisa que envelheça fora do Mnesia sem uma gravação ao vivo. Veja Quando os CDRs São Escritos para o caminho completo de gravação e garantias de idempotência.

Os CDRs fornecem armazenamento a longo prazo separado do banco de dados operacional Mnesia, permitindo:

- Faturamento e cobrança
- Análises e relatórios
- Conformidade e auditoria
- Histórico de mensagens além do período de retenção do Mnesia

Esquema da Tabela

MySQL / MariaDB

```
CREATE TABLE cdrs (
  id BIGINT AUTO_INCREMENT PRIMARY KEY,

  -- Identificação da mensagem
  message_id BIGINT NOT NULL,

  -- Números de telefone
  calling_number VARCHAR(255) NOT NULL,
  called_number VARCHAR(255) NOT NULL,

  -- Roteamento SMSC
  source_smsc VARCHAR(255),
  dest_smsc VARCHAR(255),

  -- Informações do nó (para implantações em cluster)
  origin_node VARCHAR(255),
  destination_node VARCHAR(255),

  -- Carimbos de data/hora
  submission_time DATETIME NOT NULL,
  delivery_time DATETIME,
  expiry_time DATETIME,

  -- Status e metadados
  status VARCHAR(50) NOT NULL,
  delivery_attempts INT DEFAULT 0,
  message_parts INT,
  deadletter BOOLEAN DEFAULT FALSE,

  -- Corpo da mensagem opcional (controles de privacidade)
  message_body TEXT,

  -- Carimbos de auditoria
  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)
);
```

> A chave única `(origin_node, message_id)` dá a cada CDR uma identidade globalmente única em todo o cluster. Isso também torna as gravações de CDR idempotentes. A gravação em quase tempo real e o suporte de retenção podem ambos tentar uma inserção. A segunda inserção é um no-op via `ON CONFLICT DO NOTHING`.

PostgreSQL

```
CREATE TABLE cdrs (
  id BIGSERIAL PRIMARY KEY,

  -- Identificação da mensagem
  message_id BIGINT NOT NULL,

  -- Números de telefone
  calling_number VARCHAR(255) NOT NULL,
  called_number VARCHAR(255) NOT NULL,

  -- Roteamento SMSC
  source_smsc VARCHAR(255),
  dest_smsc VARCHAR(255),

  -- Informações do nó (para implantações em cluster)
  origin_node VARCHAR(255),
  destination_node VARCHAR(255),

  -- Carimbos de data/hora
  submission_time TIMESTAMP NOT NULL,
  delivery_time TIMESTAMP,
  expiry_time TIMESTAMP,

  -- Status e metadados
  status VARCHAR(50) NOT NULL,
  delivery_attempts INTEGER DEFAULT 0,
  message_parts INTEGER,
  deadletter BOOLEAN DEFAULT FALSE,

  -- Corpo da mensagem opcional (controles de privacidade)
  message_body TEXT,

  -- Carimbos de auditoria
  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);
```

Descrições dos Campos

Chave Primária

Campo	Tipo	Nulo	Descrição
`id`	BIGINT	NÃO	Chave primária auto-incrementada para o registro CDR

Identificação da Mensagem

Campo	Tipo	Nulo	Descrição
`message_id`	BIGINT	NÃO	Identificador da mensagem da fila de mensagens SMS-C. Gerado por um contador **por nó** (persistido no Mnesia, para que sobreviva a reinicializações), portanto, único **dentro de um único nó**. Para uma chave globalmente única em um cluster de múltiplos nós, use `(origin_node, message_id)` juntos - este par é imposto por um índice único e é a chave de junção/deduplicação recomendada para sistemas de faturamento.

Números de Telefone

Campo	Tipo	Nulo	Descrição
`calling_number`	VARCHAR(255)	NÃO	MSISDN de origem (número de celular) do remetente da mensagem. Normalmente no formato E.164 (por exemplo, +15551234567).
`called_number`	VARCHAR(255)	NÃO	MSISDN de destino (número de celular) do destinatário da mensagem. Normalmente no formato E.164 (por exemplo, +15551234567).

Roteamento SMSC

Campo	Tipo	Nulo	Descrição
`source_smsc`	VARCHAR(255)	SIM	Nome ou identificador do SMSC de origem que enviou a mensagem. NULL se enviado via API ou outra interface que não seja SMSC.
`dest_smsc`	VARCHAR(255)	SIM	Nome ou identificador do SMSC de destino que entregou (ou tentou entregar) a mensagem. NULL se a mensagem nunca foi roteada.

Informações do Nó

Para implantações em cluster, rastreia quais nós manipularam a mensagem:

Campo	Tipo	Nulo	Descrição
`origin_node`	VARCHAR(255)	SIM	Nome do nó Erlang onde a mensagem foi originalmente recebida (por exemplo, "sms@node1.example.com"). Útil para solução de problemas e análise de distribuição de carga.
`destination_node`	VARCHAR(255)	SIM	Nome do nó Erlang de onde a mensagem foi entregue (se diferente da origem). NULL para implantações de nó único ou se a mensagem nunca foi entregue.

Carimbos de Data/Hora

Todos os carimbos de data/hora são armazenados em UTC:

Campo	Tipo	Nulo	Descrição
`submission_time`	DATETIME	NÃO	Quando a mensagem foi primeiro submetida ao SMS-C. Usado como o tempo de início para cálculos de faturamento.
`delivery_time`	DATETIME	SIM	Quando a mensagem foi entregue com sucesso. NULL se a mensagem expirou, falhou ou foi rejeitada.
`expiry_time`	DATETIME	SIM	Quando a mensagem expirou (tornou-se indeliverável). NULL se a mensagem foi entregue ou ainda está pendente.

Cálculo da Duração da Entrega:

```
TIMESTAMPDIFF(SECOND, submission_time, delivery_time) AS
delivery_duration_seconds
```

Status e Metadados

Campo	Tipo	Nulo	Descrição
`status`	VARCHAR(50)	NÃO	Status final da mensagem. Valores válidos: `entregue`, `expirado`, `falhou`, `rejeitado`
`delivery_attempts`	INT	NÃO	Número de tentativas de entrega feitas antes do status final. Padrão: 0. Faixa: 0-255 normalmente.
`message_parts`	INT	SIM	Número de segmentos SMS para mensagens concatenadas. 1 para mensagens de uma parte, 2+ para multi-partes. NULL se desconhecido.
`deadletter`	BOOLEAN	NÃO	Se o sistema moveu a mensagem para a fila de cartas mortas. TRUE indica que o sistema não conseguiu entregar a mensagem e esgotou todas as tentativas. Padrão: FALSE

Valores de Status:

Status	Descrição	Faturável	Tempo de Entrega
`entregue`	Entregue com sucesso ao destinatário	Sim	Definido
`expirado`	Excedeu o período de validade sem entrega	Depende da política de faturamento	NULL
`falhou`	Falha permanente na entrega (número inválido, etc.)	Depende da política de faturamento	NULL
`rejeitado`	Rejeitado por regras de roteamento ou validação	Não	NULL

Corpo da Mensagem

Campo	Tipo	Nulo	Descrição
`message_body`	TEXT	SIM	O conteúdo real da mensagem SMS. Pode ser NULL se `delete_message_body_after_delivery` estiver habilitado para privacidade. O comprimento máximo varia de acordo com o banco de dados (normalmente 65.535 caracteres para o tipo TEXT).

Modos de Privacidade:

- **Retenção total:** Corpo da mensagem armazenado no CDR para conformidade/arquivamento
- **Modo de privacidade:** Corpo da mensagem definido como NULL quando `delete_message_body_after_delivery: true`

- **Modo de conformidade:** Corpo armazenado criptografado ou hash (requer implementação personalizada)

Carimbos de Auditoria

Campo	Tipo	Nulo	Descrição
`inserted_at`	DATETIME	NÃO	Quando o registro CDR foi primeiro inserido no banco de dados. Normalmente o mesmo que ou logo após delivery_time/expiry_time.
`updated_at`	DATETIME	NÃO	Quando o registro CDR foi atualizado pela última vez. O mesmo que inserted_at se nunca atualizado.

Exemplos de SQL

Consultas Básicas

Encontrar todos os CDRs para um número de telefone específico:

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

Contar mensagens por status:

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

Tempo médio de entrega para mensagens entregues:

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

Consultas de Faturamento

Volume diário de mensagens por SMSC de destino:

```sql
SELECT
  DATE(submission_time) AS date,
  dest_smsc,
  COUNT(*) AS message_count,
  SUM(CASE WHEN status = 'entregue' 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;
```

Mensagens faturáveis para um cliente (por prefixo de número de chamada):

```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 = 'entregue'
  AND submission_time >= '2025-10-01'
  AND submission_time < '2025-11-01'
GROUP BY DATE(submission_time);
```

Análise de desempenho de rota:

```sql
SELECT
  dest_smsc,
  COUNT(*) AS total_messages,
  SUM(CASE WHEN status = 'entregue' THEN 1 ELSE 0 END) AS
delivered,
  ROUND(100.0 * SUM(CASE WHEN status = 'entregue' 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 Análise

Mensagens por hora do dia (padrão de tráfego):

```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;
```

Análise de mensagens 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 = 'entregue'
GROUP BY message_parts
ORDER BY message_parts;
```

Análise de mensagens falhadas:

```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 ('falhou', 'expirado')
  AND submission_time >= DATE_SUB(NOW(), INTERVAL 7 DAY)
GROUP BY called_number
HAVING failure_count >= 5
ORDER BY failure_count DESC;
```

Consultas de Conformidade e Auditoria

Encontrar todas as mensagens entre duas partes em um intervalo de tempo:

```
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;
```

Aplicação da política de retenção (deletar CDRs antigos):

```
-- Encontrar registros mais antigos que o período de retenção
(exemplo: 2 anos)
SELECT COUNT(*) FROM cdrs
WHERE submission_time < DATE_SUB(NOW(), INTERVAL 2 YEAR);

-- Deletar registros antigos (usar com cautela!)
DELETE FROM cdrs
WHERE submission_time < DATE_SUB(NOW(), INTERVAL 2 YEAR)
LIMIT 10000;  -- Deleção em lote para evitar bloqueio
```

Análise de Cluster

Distribuição de mensagens entre nós:

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

Índices

Os seguintes índices são criados para otimizar consultas comuns:

Nome do Índice	Colunas	Propósito
`PRIMARY`	`id`	Chave primária, garante registro único
`cdrs_origin_node_message_id_index`	`origin_node, message_id`	**Único.** Identidade CDR globalmente única em todo o cluster; torna gravações idempotentes
`idx_cdrs_message_id`	`message_id`	Lookup CDR pelo ID original da mensagem
`idx_cdrs_calling_number`	`calling_number`	Encontrar mensagens de um remetente específico
`idx_cdrs_called_number`	`called_number`	Encontrar mensagens para um destinatário específico
`idx_cdrs_status`	`status`	Filtrar por status de entrega

Nome do Índice	Colunas	Propósito
`idx_cdrs_submission_time`	`submission_time`	Consultas baseadas em tempo, períodos de faturamento
`idx_cdrs_dest_smsc`	`dest_smsc`	Análise de desempenho de rota

Recomendações de Índice Adicionais

Para implantações de alto volume, considere estes índices adicionais:

Índice composto para consultas de faturamento:

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

Índice composto para análise de rota:

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

Índice composto para buscas de conformidade:

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

Índice de texto completo para buscas no corpo da mensagem (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 Dados por Banco de Dados

Mapeamentos de tipos de campo entre os bancos de dados suportados:

Campo	MySQL/MariaDB	PostgreSQL	Notas
`id`	BIGINT AUTO_INCREMENT	BIGSERIAL	Inteiro de 64 bits, auto-incrementado
`message_id`	BIGINT	BIGINT	Inteiro de 64 bits
Campos de string	VARCHAR(255)	VARCHAR(255)	String de comprimento variável, max 255 chars
`message_body`	TEXT	TEXT	Texto grande, até 65.535 bytes (MySQL), ilimitado (PostgreSQL)
Carimbos de data/hora	DATETIME	TIMESTAMP	Carimbos UTC recomendados
Inteiros	INT	INTEGER	Inteiro com sinal de 32 bits
Booleanos	BOOLEAN (TINYINT(1))	BOOLEAN	MySQL armazena como 0/1

Considerações de Privacidade

A tabela CDR pode conter informações pessoais sensíveis (números de telefone, conteúdo da mensagem). Considere estas medidas de privacidade:

1. Privacidade do Corpo da Mensagem

Opções de configuração em `config/runtime.exs`:

```
config :sms_c,
  # Deletar corpo da mensagem após entrega bem-sucedida
  delete_message_body_after_delivery: true,

  # Ocultar corpo da mensagem na interface do usuário
  hide_message_body_in_ui: true,

  # Ocultar corpo da mensagem em exportações
  hide_message_body_in_export: true
```

2. Mascaramento de Números de Telefone

Para análises que não requerem números completos:

```
-- Mascarar os últimos 4 dígitos dos números de telefone
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. Criptografia de Banco de Dados

Ativar criptografia em repouso para o servidor de banco de dados:

MySQL:

```
-- Ativar criptografia de tabela
ALTER TABLE cdrs ENCRYPTION='Y';
```

PostgreSQL: Use criptografia de dados transparente do PostgreSQL (TDE) ou criptografia em nível de sistema de arquivos.

4. Controles de Acesso

Restringir o acesso à tabela CDR:

```
-- Criar usuário de faturamento somente leitura
CREATE USER 'billing_ro'@'%' IDENTIFIED BY 'secure_password';
GRANT SELECT ON sms_c.cdrs TO 'billing_ro'@'%';

-- Criar usuário de análise limitado (sem acesso ao corpo da
mensagem)
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'@'%';
```

Retenção e Arquivamento

Políticas de Retenção

Defina períodos de retenção com base em requisitos regulatórios e de negócios:

Indústria	Retenção Típica	Base Regulatória
Telecom (EUA)	18-24 meses	FCC, leis estaduais
Telecom (UE)	6 meses - 2 anos	GDPR, ePrivacy
Financeira	5-7 anos	SOX, SEC
Saúde	6 anos	HIPAA

Estratégia de Arquivamento

1. Particionamento por Data (MySQL 8.0+, PostgreSQL 11+)

```
-- Particionamento MySQL por mês
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
);

-- Deletar partição antiga (arquivamento rápido)
ALTER TABLE cdrs DROP PARTITION p202510;
```

2. Arquivar para Armazenamento Frio

```
-- Exportar CDRs antigos para tabela de arquivo
CREATE TABLE cdrs_archive LIKE cdrs;

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

-- Verificar e deletar da tabela principal
DELETE FROM cdrs
WHERE submission_time < DATE_SUB(NOW(), INTERVAL 2 YEAR);
```

3. Script de Limpeza Automatizado

```
#!/bin/bash
# cleanup_old_cdrs.sh - Executar via cron

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

# Arquivar registros antigos
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 do Cron:

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

Integração de Faturamento

Esquema de Tarifas

Crie uma tabela de tarifas separada para faturamento:

```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 exemplo
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', 'Austrália', 0.0100, 0.0100, '2025-01-01'),
('+', 'Padrão internacional', 0.0150, 0.0150, '2025-01-01');
```

Consulta de Faturamento

Junte os CDRs com as tarifas para faturamento:

```
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 = 'entregue'
  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;
```

Exportação para Sistemas de Faturamento

Exportação CSV embutida (recomendada):

O SMS-C pode renderizar CDRs para um arquivo CSV RFC 4180 diretamente, sem passar pelo cliente SQL. Em uma versão, dirija-o do console remoto:

```
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: :entregue)'
```

Opções: `:since`, `:until` (ambos `DateTime`), `:status` (um átomo/string ou uma lista, por exemplo, `[:entregue, :falhou]`), e `:limit`. O valor retornado é `{:ok, {path, row_count}}`. A coluna `message_body` respeita a configuração `hide_message_body_in_export` - defina como `true` para emitir `[REDACTED]` em vez do conteúdo da mensagem. Uma função de exportação acompanhante retorna o mesmo CSV como uma string se você preferir transmiti-lo para outro lugar.

Exportação direta do 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 = 'entregue'
" --batch --silent | sed 's/\t/,/g' > billing_export_202510.csv
```

Veja Também

- Guia de Configuração - Configurar as definições de exportação de CDR
- Guia de Operações - Procedimentos de manutenção de CDR
- Referência da API - Consultar CDRs via API REST

Referência de Configuração do SMS-C

← Voltar ao Índice da Documentação

Referência completa para todas as opções de configuração do SMS-C com exemplos para cenários de implantação comuns.

Índice

- Arquivos de Configuração
- Configuração do Banco de Dados
- Configuração da API
- Configuração da Interface Web
- Configuração de Federação
- Configuração da Fila de Mensagens
- Configuração de Cobrança
- Configuração do Diameter Sh (HSS)
- Configuração ENUM
- Configuração de Tradução de Números
- Intercepção Legal (X1/X2/X3)
- Configuração de Roteamento
- Configuração de Ajuste de Desempenho
- Configuração de Registro
- Cenários Comuns de Configuração

Arquivos de Configuração

O SMS-C utiliza três arquivos principais de configuração:

config/config.exs

Configuração estática carregada em tempo de compilação. Contém:

- Padrões globais da aplicação
- Configuração do logger
- Configurações de desenvolvimento/teste
- Parâmetros de ajuste de desempenho

config/runtime.exs

Configuração em tempo de execução carregada na inicialização. Contém:

- Configurações de conexão com o banco de dados
- Configuração de cluster
- Integração com serviços externos (OCS, ENUM)
- Rotas iniciais e regras de tradução
- Configurações específicas do ambiente

config/prod.exs (opcional)

Substituições específicas para produção.

Melhor Prática: Use variáveis de ambiente em `runtime.exs` para valores sensíveis como senhas e chaves de API.

Configuração de Armazenamento SQL CDR

O SMS-C utiliza **Mnesia** para dados operacionais (fila de mensagens, regras de roteamento, traduções de números) e suporta **bancos de dados SQL** externos para armazenamento de CDR (Registro de Detalhes de Chamadas) a longo prazo, cobrança e análises.

Bancos de Dados SQL Suportados

O sistema suporta os seguintes bancos de dados SQL para exportação de CDR:

Banco de Dados	Versão	Adaptador	Porta Padrão	Melhor Pa
MySQL	8.0+	`Ecto.Adapters.MyXQL`	3306	Uso geral, confiabilida comprovad
MariaDB	10.5+	`Ecto.Adapters.MyXQL`	3306	Compatível com MySQL código aber
PostgreSQL	13+	`Ecto.Adapters.Postgres`	5432	Recursos avançados, suporte a JSON

Nota: Mnesia é utilizado automaticamente para dados operacionais (fila de mensagens, roteamento, traduções) e não requer configuração. O banco de dados SQL é **usado apenas** para exportação de CDR e armazenamento a longo prazo.

Quando os CDRs São Escritos

Os CDRs são escritos no banco de dados SQL em dois momentos:

1. **Quase em tempo real, no estado terminal** - no momento em que uma mensagem atinge um resultado final (`delivered`, `expired`, `failed`, `rejected` ou `balance_rejected`), um CDR é escrito de forma assíncrona. Este é o caminho principal e mantém a cobrança atual sem esperar pela retenção de mensagens.
2. **Retenção como suporte** - quando o trabalhador de retenção remove uma mensagem expirada do Mnesia, ele exporta um CDR primeiro. Ele registra `expired` para qualquer coisa que nunca atingiu um estado terminal. O

trabalhador só exclui uma mensagem uma vez que sua exportação de CDR tenha sido bem-sucedida. Nada é perdido se o banco de dados de CDR estiver brevemente indisponível: o trabalhador tenta a exportação na próxima varredura.

Ambos os caminhos são idempotentes. Cada CDR usa o par único `(origin_node, message_id)` como sua chave. As gravações usam `ON CONFLICT DO NOTHING`. O suporte, portanto, nunca duplica uma mensagem que a gravação quase em tempo real já cobriu.

Um contador por nó gera `message_id` (persistido no Mnesia, para que sobreviva a reinicializações). `message_id` é, portanto, único **dentro de um único nó**. Quando vários nós SMS-C escrevem em um banco de dados CDR compartilhado, `(origin_node, message_id)` juntos formam a chave globalmente única. Veja a Referência do Esquema CDR. Defina `cdr_enabled: false` para desabilitar completamente a gravação de CDR.

Exportação CSV / Arquivo de Texto

Os CDRs podem ser exportados do banco de dados SQL para um arquivo CSV RFC 4180. Em uma versão (onde `mix` não está disponível), execute a partir do console remoto:

```
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])'
```

Opções suportadas: `:since`, `:until` (ambos `DateTime`), `:status` (um átomo/string ou lista) e `:limit`. A coluna `message_body` respeita a configuração `hide_message_body_in_export` (emitida como `[REDACTED]` quando habilitada).

Configuração 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")
```

Configuração 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")
```

Escolhendo um Banco de Dados SQL

MySQL/MariaDB - Recomendado para a maioria das implantações:

- Excelente desempenho para gravações de CDR
- Confiabilidade comprovada em ambientes de telecomunicações
- Amplo suporte de ferramentas para sistemas de cobrança
- Configuração de replicação fácil

PostgreSQL - Considere se você precisar:

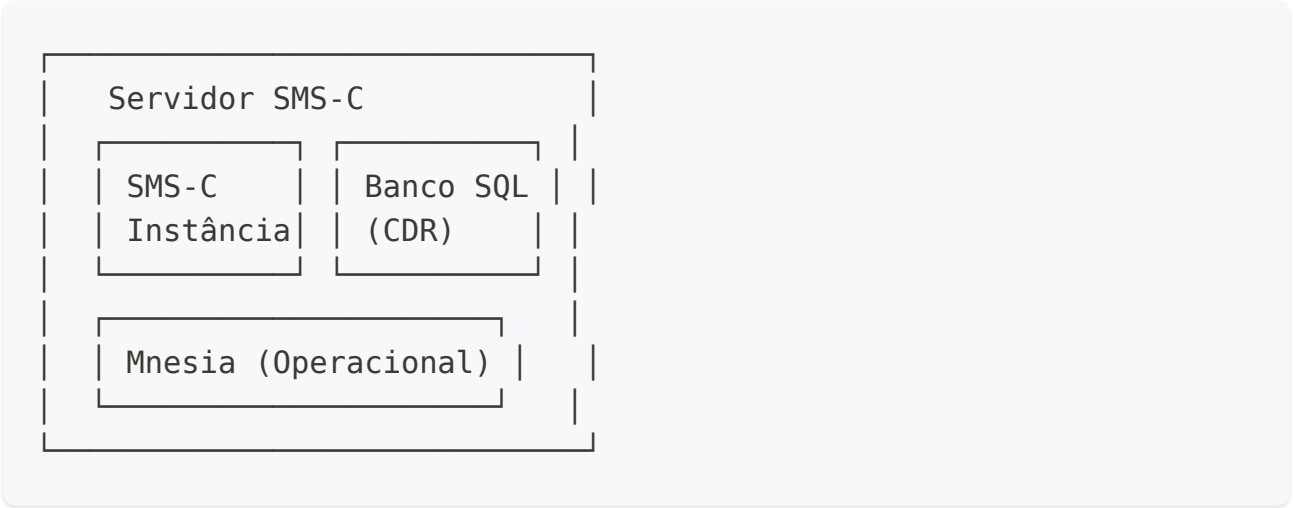
- Recursos avançados de JSON/JSONB para análises

- Consultas complexas sobre dados de CDR
- Infraestrutura existente do PostgreSQL
- PostGIS para análise geográfica

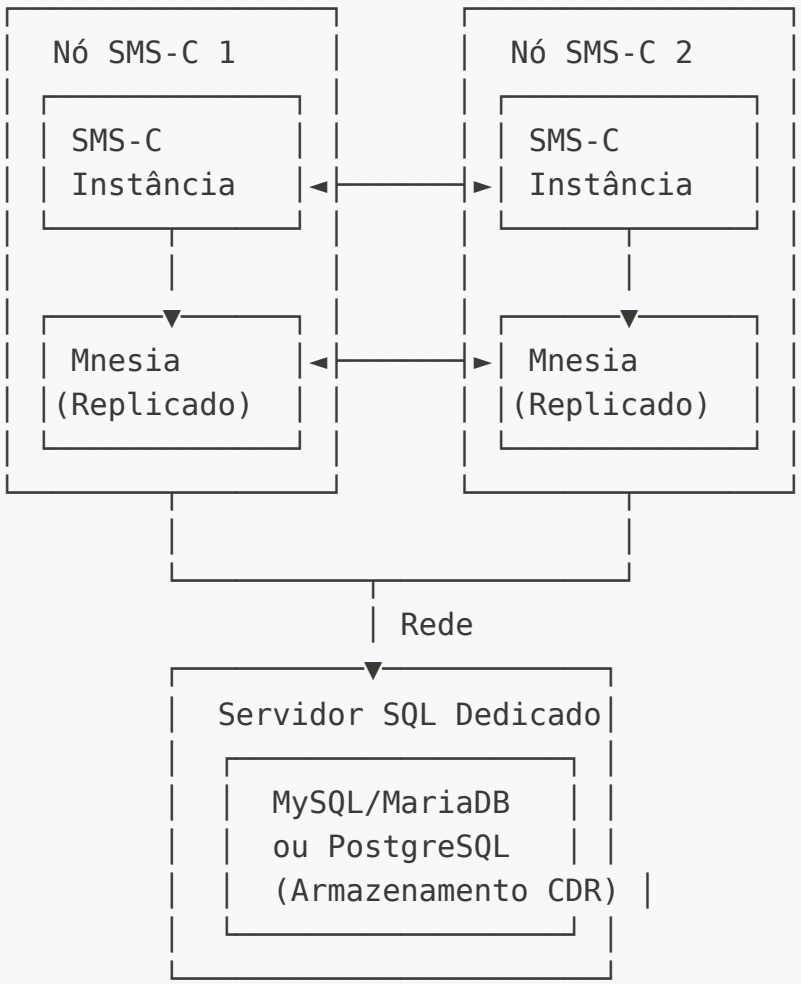
Topologias de Implantação

Importante: O banco de dados SQL CDR pode ser executado em um **servidor separado** de suas instâncias SMS-C. Esta é a abordagem recomendada para implantações em produção.

Implantação em Servidor Único (Desenvolvimento/Teste):



Implantação Distribuída (Produção - Recomendado):



Benefícios de um Servidor SQL Separado:

- **Isolamento de Desempenho**: Gravações de CDR não impactam o processamento de mensagens
- **Escalabilidade**: Escale o banco de dados e o processamento de mensagens de forma independente
- **Confiabilidade**: A manutenção do banco de dados não afeta o tempo de atividade do SMS-C
- **Gerenciamento de Dados**: Armazenamento centralizado de CDR para múltiplas instâncias SMS-C
- **Flexibilidade de Backup**: Agendas de backup e políticas de retenção independentes

Diretrizes de Tamanho do Pool

Carga de Trabalho	Tamanho do Pool	Descrição
Desenvolvimento	5-10	Concurrência mínima
Baixo Volume (< 100 msg/sec)	10-15	Pequenas implantações
Volume Médio (100-1000 msg/sec)	20-30	Produção típica
Alto Volume (> 1000 msg/sec)	40-100	Cenários de alta vazão

Cálculo: `pool_size = (operações DB concorrentes esperadas) * 1.5`

Exemplos de Conexão com o Banco de Dados

Usando Variáveis de Ambiente (Recomendado para Produção):

```
# Definir variáveis de ambiente
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
```

Configuração Direta (Apenas para Desenvolvimento):

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

Monitoramento do Pool de Conexões

Monitore o uso do pool através de métricas do Prometheus:

- `ecto_pools_queue_time` - Tempo esperando por conexão
- `ecto_pools_query_time` - Tempo de execução da consulta
- `ecto_pools_connected_count` - Conexões ativas

Alerta se o tempo de espera exceder 100ms consistentemente - indica necessidade de um pool maior.

Configuração da API

A API REST fornece capacidades de submissão e gerenciamento de mensagens.

Configuração Básica da 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"
  }
```

As configurações da API estão sob o mapa `:api` da aplicação `:api_ex`. A tabela de rotas (`routes:`) é definida em tempo de compilação em

`config/config.exs` e não é editada aqui.

Parâmetro	Tipo	Obrigatório	Padrão	D
`port`	Inteiro	Não	`8443`	P p R
`listen_ip`	String	Não	`0.0.0.0`	E d p U `0` p a i
`enable_tls`	Booleano	Não	`true`	S A H
`tls_cert_path`	String	Se `enable_tls`	`priv/cert/omnitouch.crt`	C p c T
`tls_key_path`	String	Se `enable_tls`	`priv/cert/omnitouch.pem`	C p c p T

Configuração TLS/SSL

Configuração de Produção com 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"
  }
```

Configuração de Desenvolvimento sem TLS:

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

Configuração do Certificado da API

Gere um certificado autoassinado para testes:

```
# Criar diretório de certificados
mkdir -p priv/cert

# Gerar chave privada
openssl genrsa -out priv/cert/server.key 2048

# Gerar solicitação de assinatura 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"

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

# Definir permissões
chmod 600 priv/cert/server.key
chmod 644 priv/cert/server.crt
```

Para produção, use certificados de uma CA confiável (Let's Encrypt, CA comercial, etc.).

Controle de Acesso da API

Lista Branca de IPs (Firewall de Aplicação):

```
# 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
```

Autenticação por Chave da API (Nível de Aplicação):

Configure via plug personalizado no roteador - veja o Guia de Operações para detalhes de implementação.

Configuração da Interface Web

A interface web fornece gerenciamento de rotas, navegação de mensagens e monitoramento.

Configuração Básica da Interface Web

```
# config/runtime.exs
config :control_panel, ControlPanelWeb.Endpoint,
  https: [
    port: 443,
    keyfile: "priv/cert/omnitouch.pem",
    certfile: "priv/cert/omnitouch.crt"
  ]
```

A interface web é um endpoint Phoenix. O TLS é configurado através da lista de palavras-chave `https:` do endpoint (não `tls_cert_path`/`tls_key_path`), e o HTTPS é imposto via `force_ssl` em `config/config.exs`.

Parâmetro	Tipo	Obrigatório	Padrão	Descriç
`port`	Inteiro	Não	`443`	Porta HTTPS para o painel d controle
`keyfile`	String	Sim	`priv/cert/omnitouch.pem`	Caminh para a chave privada TLS.
`certfile`	String	Sim	`priv/cert/omnitouch.crt`	Caminh para o certifica TLS.

Configuração da Interface Web para Produção

```
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"
  ]
```

Configuração de Proxy Reverso (Recomendado)

Use Nginx ou Apache como proxy reverso para segurança e recursos adicionais:

Exemplo de Configuração do 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;

  # Autenticação básica para segurança adicional
  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;
  }

  # Suporte a 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;
  }
}
```

Configuração de Federação

OmniMessage usa federação baseada em HTTP para implantações de múltiplos controladores. Os controladores se descobrem via registros SRV DNS ou listas de pares estáticas, trocam registros de saúde e frontend via HTTPS, e encaminham mensagens para controladores remotos quando necessário.

Veja o **Guia de Federação Geográfica** para a arquitetura completa, exemplos de implantação e resolução de problemas.

Início Rápido

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

Referência Completa de Parâmetros

Parâmetro	Tipo	Padrão	Descrição
`enabled`	Booleano	`false`	Interruptor mestre para federação.
`dns_srv_domain`	String	`""`	Domínio SRV DNS para descoberta de pares.
`dns_poll_interval_ms`	Inteiro	`30000`	Intervalo de re-resolução DNS.
`health_check_interval_ms`	Inteiro	`10000`	Intervalo de verificação de saúde dos pares.
`registry_sync_interval_ms`	Inteiro	`15000`	Intervalo de troca de registro frontend.
`forward_retry_interval_ms`	Inteiro	`5000`	Intervalo de tentativa de reenvio de mensagens enfileiradas.
`forward_max_retries`	Inteiro	`50`	Máximo de tentativas de reenvio antes de marcar como falhado.
`http_timeout_ms`	Inteiro	`5000`	Tempo limite para todas as chamadas HTTP aos pares.

Parâmetro	Tipo	Padrão	Descrição
`api_port`	Inteiro	`8443`	Porta da API para construção de URLs de pares.
`static_peers`	Lista	`[]`	Lista de pares estáticos (alternativa ao DNS SRV).

Requisitos de Rede

A federação requer apenas a porta **8443** (HTTPS) entre os sites, em comparação com as portas 4369 + 9100-9200 para clustering Erlang.

```
# Permitir tráfego de federação de sites 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
```

Configuração da Fila de Mensagens

Controla o comportamento de retenção e expiração de mensagens.

Expiração de Mensagens

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

Valores Comuns:

- **60** - 1 hora (teste/desenvolvimento)
- **1440** - 24 horas (produção típica)
- **4320** - 3 dias (retenção estendida)

- **10080** - 7 dias (retenção máxima)

Mensagens mais antigas que este valor tornam-se indesejáveis e são marcadas para limpeza.

Configuração de Tentativas de Entrega

O comportamento de tentativas utiliza retrocesso exponencial:

```
Delay de Tentativa = 2^(contagem_de_tentativas) minutos
```

Tentativa	Delay
1	2 minutos
2	4 minutos
3	8 minutos
4	16 minutos
5	32 minutos
6	64 minutos
7	128 minutos
8	256 minutos

Tentativas máximas antes da carta morta: Limitadas por `dead_letter_time_minutes`.

Configuração de Limpeza

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

Intervalos de Limpeza:

- **cleanup_interval_minutes**: Com que frequência o trabalhador de limpeza é executado (padrão: 10)
- **fingerprint_ttl_minutes**: Janela de detecção de duplicatas (padrão: 5)
- **event_ttl_days**: Retenção de logs de eventos (padrão: 7)

Configuração de Cobrança

A cobrança online é realizada através da interface **Diameter Ro** (Aplicação de Controle de Crédito, RFC 4006). Para cada mensagem cobrada, o SMS-C envia um Credit-Control-Request (Event-Request, `CC-Request-Type 4`) para o OCS pelo **remetente** da mensagem; o OCS autoriza ou recusa com base no saldo do assinante. Isso reutiliza a pilha Diameter (veja a seção Diameter Sh abaixo) - a aplicação `:ro` deve ser registrada.

Habilitar Cobrança

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

config :sms_c, :charging,
  # O que fazer quando o remetente está sem saldo:
  #   :reject     - marcar a mensagem como :balance_rejected e não
entregá-la
  #   :auto_reply - enviar também SMS out_of_balance_reply_text de
volta ao remetente
  out_of_balance_action: :reject,
  out_of_balance_reply_text: "Sua mensagem não pôde ser enviada
devido a saldo insuficiente.",
  # Modo de falha quando o OCS apresenta erros ou a pilha Diameter
não está configurada
  # (NÃO é uma resposta definitiva de saldo insuficiente):
  #   :allow  - falhar aberto, entregar a mensagem (padrão)
  #   :reject - falhar fechado, tratar como saldo insuficiente
  on_charging_error: :allow,
  # Perfil de classificação selecionando como o OCS classifica a
mensagem. Enviado como o CCR
  # Service-Context-Id.
  rating_profile: "sms@sms-c"
```

A cobrança também requer a pilha Diameter habilitada com a aplicação `:ro` (Controle de Crédito é base IETF, fornecedor 0):

```
config :sms_c,
  diameter_enabled: true

config :diameter_ex,
  diameter: %{
    service_name: :omnimessage,
    host: "smsc01",
    realm: "epc.mnc001.mcc001.3gppnetwork.org",
    # ... veja a seção Diameter Sh para a pilha/pares 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}
        ]
      }
    ]
  }
```

Cobrança por Rota

A política `charged` de cada rota controla a cobrança de forma independente do padrão global:

`charged`	Comportamento
`:yes`	Sempre verificar saldo via OCS
`:no`	Nunca cobrar
`:default`	Usar `default_charging_enabled`

Desabilitar Cobrança

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

Quando desabilitada (o padrão), as rotas `:default` roteiam sem qualquer verificação de saldo OCS. Rotas individuais ainda podem optar por participar com `charged: :yes`.

Padrões e Degradação Elegante

O bloco `config :sms_c, :charging` é opcional - cada chave tem um padrão sensato (`out_of_balance_action: :reject`, um texto de resposta embutido, e `rating_profile: "sms@sms-c"`), e `default_charging_enabled` tem como padrão `false`.

Por padrão, o carregamento **falha aberto** (`on_charging_error: :allow`), correspondendo ao caminho de carregamento HTTP anterior que registrava falhas, mas nunca bloqueava a entrega:

- Se uma rota solicitar carregamento, mas a pilha Diameter **não** estiver habilitada/ configurada, a mensagem é **permitida** (registrada como um aviso).
- Se o OCS estiver inacessível ou retornar um erro de protocolo, a mensagem é **permitida**.

Defina `on_charging_error: :reject` para **falhar fechado** em vez disso - erros do OCS e uma pilha não configurada são então tratados como fora de balanço e a `out_of_balance_action` se aplica.

De qualquer forma, uma **resposta definitiva fora de balanço** do OCS sempre aciona a `out_of_balance_action` configurada. Mensagens rejeitadas são marcadas com status `:balance_rejected`.

Configuração Diameter Sh (HSS)

OmniMessage pode consultar o HSS via a interface Diameter Sh para detectar assinantes on-net que estão temporariamente offline, impedindo que mensagens sejam roteadas para fora do gateway padrão. Consulte o Guia de Consulta de Assinantes HSS para a documentação operacional completa.

```
# Habilitar dip HSS no caminho de roteamento
config :sms_c,
  diameter_enabled: true

# Configuração da pilha 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
      }
    ]
  }
```

Parâmetro	Tipo	Requerido	Padrão	Descri
`diameter_enabled`	Booleano	Não	`false`	Habilitar co de assinant durante a correspondê rotas (para com critério `:on_net`/`:o` Quando `fal` esses critér falham fech
`hss_cache_ttl_seconds`	Inteiro	Não	`7200`	TTL (segund resultados c consulta HS cache
`mock_sh`	Booleano	Não	`false`	Usar respos simuladas p testes (nenh pilha Diame iniciada)
`mock_sh_on_net_numbers`	Lista	Não	`[]`	MSISDNs tra como on-ne modo simul

Para a referência completa de parâmetros da pilha Diameter (host, realm, peers, applications), consulte o Guia de Consulta de Assinantes HSS.

Configuração ENUM

Consultas de números E.164 baseadas em DNS para roteamento inteligente.

Desabilitar ENUM (Padrão)

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

Habilitar ENUM com DNS Padrão

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

Habilitar ENUM com 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 do Google (fallback)
    {"1.1.1.1", 53}     # DNS Cloudflare (fallback)
  ],
  enum_timeout: 3000  # 3 segundos (failover mais rápido)
```

Prioridade do Domínio ENUM

Os domínios são consultados em ordem até uma consulta bem-sucedida:

```
config :sms_c,
  enum_domains: [
    "e164.internal.example.com",  # Tente o interno primeiro
    "e164.carrier.net",            # Depois o carrier
    "e164.arpa"                    # Depois o registro público
  ]
```

Ajuste de Desempenho ENUM

Para Redes de Baixa Latência:

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

Para Links de Alta Latência/Satélite:

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

Exemplo de Configuração de DNS ENUM

Configurar Zona ENUM Privada (formato BIND9):

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

; Número: +1-555-0100 se torna
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!" .
```

Testar Resolução ENUM:

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

# A saída esperada inclui 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!" .
```

Configuração de Tradução de Números

Normalização de números baseada em regex aplicada antes do roteamento.

Desabilitar Tradução de Números

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

Exemplos Básicos de Tradução de Números

Adicionar Código de País a Números Locais:

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

Normalizar Formato Internacional:

```
%{
  calling_prefix: nil,
  called_prefix: nil,
  source_smsc: nil,
  calling_match: "^00(\d+)$",                # Correspondência do
prefixo 00
  calling_replace: "+\1",                    # Substituir por +
  called_match: "^00(\d+)$",
  called_replace: "+\1",
  priority: 10,
  description: "Converter prefixo internacional 00 para +",
  enabled: true
}
```

Remover Caracteres de Formatação:

```
%{
  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 formatação de número de telefone dos
EUA",
  enabled: true
}
```

Tradução Específica do Carrier

Remoção de Código de Rota:

```
%{
  calling_prefix: nil,
  called_prefix: "101",                        # Apenas para
prefixo 101
  source_smsc: "carrier_a",                    # Apenas deste
carrier
  calling_match: nil,                          # Não alterar
chamada
  calling_replace: nil,
  called_match: "^101(\d+)$",                 # Remover código de
rota 101
  called_replace: "\1",
  priority: 5,
  description: "Remover código de rota do carrier do número
chamado",
  enabled: true
}
```

Tradução de Múltiplas Regras

As regras são avaliadas em ordem de prioridade (número menor = maior prioridade):

```
config :sms_c,
  translation_rules: [
    # Prioridade 1: Regras mais específicas primeiro
    %{
      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: "Normalização de número premium",
      enabled: true
    },

    # Prioridade 50: Regras gerais
    %{
      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: "Normalização geral de 10 dígitos",
      enabled: true
    }
  ]
```

Interceptação Legal (X1/X2/X3)

OmniMessage atua como um Ponto de Interceptação de Serviço de Mensagens Curtas (POI). OmniMessage implementa as interfaces de interceptação ETSI TS

103 221. OmniMessage opera como o Elemento de Rede (NE). Uma Instalação de Monitoramento de Aplicações de Justiça (LEMF) provisiona alvos de interceptação via X1. Quando uma mensagem corresponde a um alvo provisionado, OmniMessage codifica a mensagem como um registro xIRI TS 33.128 3GPP. OmniMessage então envia o registro para o LEMF via X2.

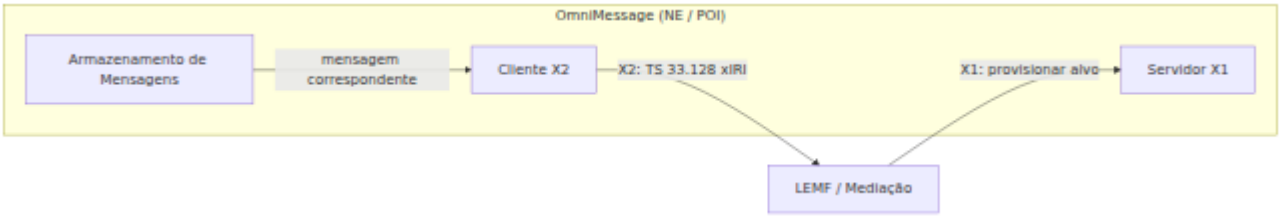
Visão Geral da Interface

Interface	Propósito	Papel do OmniMessage	Transporte	Usado
X1	Provisionar e remover tarefas e destinos de interceptação	Servidor	HTTPS com TLS mútuo	Sim
X2	Entregar Informações Relacionadas à Interceptação (cada SMS interceptado como um TS 33.128 xIRI)	Cliente	TCP, TLS opcional	Sim
X3	Entregar Conteúdo da Comunicação (mídia)	Cliente	TCP	Não. SMS não tem plano de mídia.

Referências de padrões:

Referência	Título
ETSI TS 103 221-1	Interface de interceptação interna X1 (provisionamento)
ETSI TS 103 221-2	Interface de interceptação interna X2/X3 (entrega)
3GPP TS 33.128	Formatos de carga útil de interceptação legal

Arquitetura



Configuração

Configure a interceptação sob `config :sms_c, :li`. Defina os valores em `config/runtime.exs` para cada implantação. A interceptação está desabilitada por padrão. Nenhum processo de interceptação começa, e nenhuma interceptação ocorre, até que você defina `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]
```

Parâmetros de Nível Superior

Parâmetro	Tipo	Requerido	Padrão	Descrição
`enabled`	Booleano	Não	`false`	Interruptor mestre. Quando `false`, nenhum processo de interceptação começa.
`ne_identifier`	String	Não	`"smsc-01"`	Identificador da Função de Rede (atributo 18 TS 103 221-2). OmniMessage envia este valor em cada registro X2. O LEMF o usa para rotular a conexão pelo nome em vez de pelo endereço IP.
`x1`	Lista de palavras-chave	Não	`[]`	Servidor de provisionamento X1. Consulte Parâmetros X1.
`x2`	Lista de palavras-chave	Não	`[]`	Cliente de entrega X2. Consulte Parâmetros X2.
`x3`	Lista de palavras-chave	Não	`[]`	Aceito para simetria com a biblioteca de interceptação. OmniMessage não usa X3. Deixe como `[enabled: false]`.

Parâmetros X1

OmniMessage executa o servidor X1. O LEMF se conecta ao servidor para provisionar alvos. TLS é obrigatório (cláusula 7.2.2.1 TS 103 221). O servidor não inicia em modo plaintext a menos que você defina `allow_insecure: true`.

Parâmetro	Tipo	Requerido	Padrão	Descrição
`enabled`	Booleano	Não	`false`	Iniciar o servidor X1.
`port`	Inteiro	Não	`8443`	Porta TCP para o servidor X1.
`tls`	Lista de palavras-chave	Sim	-	Configurações TLS para o servidor X1. Consulte Parâmetros TLS X1.
`allow_insecure`	Booleano	Não	`false`	Executar o servidor X1 sobre HTTP plaintext. Use esta opção apenas em laboratório. Não use esta opção em produção.

Parâmetros TLS X1

Parâmetro	Tipo	Requerido	Padrão	Descrição
`certfile`	String	Sim	-	Caminho para o certificado do servidor, em formato PEM.
`keyfile`	String	Sim	-	Caminho para a chave privada do servidor, em formato PEM.
`cacertfile`	String	Não	-	Caminho para a autoridade certificadora que verifica os certificados dos clientes. Defina este valor para habilitar TLS mútuo. Implantações de produção requerem TLS mútuo.

Parâmetros X2

OmniMessage executa o cliente X2. O cliente se conecta ao LEMF. Ele envia cada SMS interceptado como um registro xIRI TS 33.128. O registro carrega a mensagem SMS e um Identificador de Alvo Correspondente.

Parâmetro	Tipo	Requerido	Padrão	Descrição
`enabled`	Booleano	Não	`false`	Iniciar o cliente X2.
`address`	String	Sim	-	Endereço IP ou nome do host do endpoint X2 do LEMF.
`port`	Inteiro	Sim	-	Porta TCP do endpoint X2 do LEMF.
`tls`	Booleano ou lista de palavras-chave	Não	`false`	Defina `true` ou uma lista de palavras-chave de opções TLS para entregar X2 via TLS. Defina `false` para TCP plaintext.

Persistência

OmniMessage mantém tarefas e destinos de interceptação provisionados na memória por padrão. Para manter os alvos provisionados após uma reinicialização, habilite o armazenamento de interceptação persistente na configuração de implantação. OmniMessage grava as tarefas e destinos no disco. OmniMessage os recarrega na próxima inicialização.

Conformidade

Para uma visão geral da conformidade de interceptação, consulte Conformidade de Interceptação ANSSI R226.

Configuração de Roteamento

Regras de roteamento iniciais carregadas na primeira inicialização. Consulte o Guia de Roteamento SMS para a documentação completa de roteamento.

Carregar Rotas da Configuração

```
# config/runtime.exs
config :sms_c,
  sms_routes: [
    # Exemplo de roteamento 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: "Roteamento para a América do Norte",
      enabled: true
    },

    # Exemplo de roteamento 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: "Gateway primário do 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: "Gateway de backup do Reino Unido (30%)",
      enabled: true
    },

    # Apenas on-net - restringir um bind SMPP a destinos apenas
on-net
    %{
      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 - terminação apenas on-net",
      enabled: true
    }
  ]
```

Pular Carregamento Inicial de Rotas

```
# Não carregar rotas da configuração (gerenciar apenas via UI Web)
config :sms_c,
  sms_routes: []
```

Rotas definidas na configuração são SOMENTE carregadas se a tabela de roteamento estiver vazia (primeira inicialização).

Configuração de Ajuste de Desempenho

Trabalhador de Inserção em Lote

```
# config/config.exs
config :sms_c,
  batch_insert_batch_size: 100,           # Mensagens por lote
  batch_insert_flush_interval_ms: 100     # Tempo máximo de espera
em ms
```

Perfis de Desempenho:

Perfil	Tamanho do Lote	Intervalo	Taxa de Transferência	Latência
Alto Volume	200	200ms	~5,000 msg/sec	Até 200ms
Equilibrado	100	100ms	~4,500 msg/sec	Até 100ms
Baixa Latência	50	20ms	~3,000 msg/sec	Até 20ms
Tempo Real	10	10ms	~1,500 msg/sec	Até 10ms

Configuração de Registro

Níveis 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 Produção: `:info` ou `:warning` **Recomendado para Desenvolvimento**: `:debug`

Destinos de Saída de Registro

Apenas Console (Desenvolvimento):

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

Registrador de Arquivo (Produção):

```
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]
```

Rotação de Registro

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
        # Sinalizar aplicação para reabrir arquivo de log
        systemctl reload omnimessage
    endscript
}
```

Cenários Comuns de Configuração

Agregador de Alto Volume

Otimize para máxima taxa de transferência (5,000+ mensagens/segundo):

```
# Banco de dados
config :sms_c, SmsC.Repo,
  pool_size: 50

# Trabalhador em lote
config :sms_c,
  batch_insert_batch_size: 200,
  batch_insert_flush_interval_ms: 200

# Retenção de mensagens
config :sms_c,
  dead_letter_time_minutes: 1440  # 24 horas

# Carregamento (desabilitado para desempenho)
config :sms_c,
  default_charging_enabled: false

# Limpeza (intervalos estendidos)
config :sms_c,
  cleanup_interval_minutes: 30
```

Mensagens em Tempo Real para Empresas

Otimize para baixa latência (< 20ms):

```
# Banco de dados
config :sms_c, SmsC.Repo,
  pool_size: 20

# Trabalhador em lote (baixa latência)
config :sms_c,
  batch_insert_batch_size: 20,
  batch_insert_flush_interval_ms: 10

# Retenção de mensagens
config :sms_c,
  dead_letter_time_minutes: 4320  # 3 dias

# Carregamento (habilitado) - verificações de saldo via Diameter
Ro
# (consulte a seção de Configuração de Carregamento para os blocos
:charging + :diameter_ex)
config :sms_c,
  default_charging_enabled: true
```

Provedor de Serviço Multi-Tenant

Configuração separada por inquilino:

```
# Ambiente do Inquilino 1
export DB_NAME=sms_c_tenant1
export NODE_NAME=sms_tenant1@node1.example.com
# A classificação OCS por inquilino é selecionada via a
configuração de :charging rating_profile

# Ambiente do Inquilino 2
export DB_NAME=sms_c_tenant2
export NODE_NAME=sms_tenant2@node1.example.com
```

Redundância Geográfica

Cluster em regiões:

```
# Cluster da Costa Leste dos EUA
config :sms_c,
  cluster_nodes: [
    :"sms@us-east-1a.example.com",
    :"sms@us-east-1b.example.com",
    :"sms@us-west-1a.example.com"  # Cross-region para DR
  ],
  smsc_node_name: "us-east-1a"
```

Validação de Configuração

Teste a configuração antes da implantação:

```
# Verificar a sintaxe da configuração
mix compile

# Validar a conexão com o banco de dados
mix ecto.create
mix ecto.migrate

# Verificar o peer Diameter Ro para o OCS (se o carregamento
estiver habilitado), em um console remoto:
#   :diameter.service_info(:omnimessage, :connections)

# Iniciar a aplicação em modo interativo
iex -S mix phx.server
```

Referência de Variáveis de Ambiente

Variáveis de ambiente comuns usadas na configuração:

Variável	Propósito	Exemplo
`DB_USERNAME`	Nome de usuário do banco de dados	`sms_prod_user`
`DB_PASSWORD`	Senha do banco de dados	`strong_password`
`DB_HOSTNAME`	Host do banco de dados	`db.internal.example.com`
`DB_PORT`	Porta do banco de dados	`3306`
`DB_NAME`	Nome do banco de dados	`sms_c_production`
`DB_POOL_SIZE`	Tamanho do pool de conexões	`30`
`API_PORT`	Porta de escuta da API	`8443`
`API_LISTEN_IP`	IP de escuta da API	`0.0.0.0`
`WEB_PORT`	Porta da interface web	`443`
`NODE_NAME`	Nome do nó Erlang	`sms@node1.example.com`
`ERLANG_COOKIE`	Segredo do cluster	`shared_cookie_value`

Melhores Práticas de Configuração

1. **Use Variáveis de Ambiente** para valores sensíveis (senhas, chaves de API)
2. **Teste Alterações de Configuração** em staging antes da produção
3. **Documente Configurações Personalizadas** nas notas de implantação
4. **Controle de Versão dos Arquivos de Configuração** (excluindo segredos)
5. **Monitore Após Alterações** para regressões de desempenho

6. **Mantenha Backups** de configurações funcionais
7. **Valide Antes de Reiniciar** para evitar falhas na inicialização
8. **Use Nomenclatura Consistente** entre ambientes
9. **Defina Limites de Recursos** apropriados ao hardware
10. **Revise Periodicamente** para remover recursos não utilizados

Solução de Problemas de Configuração

Sintoma	Causa Provável	Solução
O aplicativo não inicia	Erro de sintaxe na configuração	Verifique os logs, valide a sintaxe
Falha na conexão com o banco de dados	Credenciais/host incorretos	Verifique as variáveis de ambiente DB_*
API não acessível	Porta/IP de vinculação incorretos	Verifique API_PORT e listen_ip
Nós do cluster não se conectam	Incompatibilidade de cookie, firewall	Verifique ERLANG_COOKIE, verifique as portas 4369, 9100-9200
Falhas de cobrança	OCS/DRA inacessível	Verifique o peer Diameter Ro (falhas de cobrança abertas)
Consultas ENUM falham	Servidor DNS inacessível	Teste a conectividade DNS, verifique o tempo limite
Desempenho ruim	Configurações de lote incorretas	Revise a seção de Configuração de Ajuste de Desempenho acima
Mensagens não roteadas	Rotas não carregadas	Verifique a configuração sms_routes ou a interface Web

Para ajuda adicional, consulte o Guia de Solução de Problemas.

Configuração de Armazenamento de Mensagens (Mnesia)

Retenção de Mensagens

As mensagens são armazenadas no Mnesia para acesso rápido com limpeza automática configurável.

```
config :sms_c,
  # Quanto tempo manter mensagens no Mnesia (horas).
  # O fallback embutido é 24 se não definido; o runtime.exs
enviado define 168 (7 dias).
  message_retention_hours: 168,

  # Com que frequência verificar mensagens antigas (minutos)
  retention_check_interval_minutes: 60
```

Recomendações:

- **Produção:** 24-72 horas (equilibrar necessidades operacionais vs memória)
- **Desenvolvimento:** 4-8 horas (limpeza mais rápida para testes)
- **Alto volume:** 12-24 horas (conservar memória)

Impacto na Memória:

- Mensagem média: ~1KB
- 10.000 mensagens: ~10MB
- 100.000 mensagens: ~100MB

Exportação de CDR (Registro de Detalhes da Chamada)

Quando as mensagens são entregues ou expiradas, os CDRs podem ser escritos automaticamente no seu banco de dados Ecto para armazenamento a longo prazo e análises de cobrança.

```
config :sms_c,
  # Habilitar/desabilitar escrita de CDR (padrão embutido: false)
  cdr_enabled: true
```

Registros de CDR Incluem:

- ID da mensagem, números chamadores/chamados
- SMSC de origem/destino
- Nó de origem/destino (para clusters)
- Carimbos de data/hora de submissão, entrega, expiração
- Status, tentativas de entrega
- Corpo da mensagem opcional (veja controles de privacidade)

Quando Desabilitar:

- Ambientes de teste onde CDRs não são necessários
- Solução temporária de problemas para reduzir a carga do banco de dados

Controles de Privacidade

Configure a visibilidade e retenção do corpo da mensagem para conformidade com a privacidade.

```
config :sms_c,
  # Excluir corpo da mensagem do Mnesia após entrega bem-sucedida
  delete_message_body_after_delivery: false,

  # Ocultar corpo da mensagem na interface web
  hide_message_body_in_ui: false,

  # Ocultar corpo da mensagem em exportações CSV
  hide_message_body_in_export: false
```

Casos de Uso:

Configuração	Caso de Uso
`delete_message_body_after_delivery: true`	Economizar espaço no Mnesia, conformidade de privacidade
`hide_message_body_in_ui: true`	Impedir que o operador veja o conteúdo da mensagem
`hide_message_body_in_export: true`	Conformidade de exportação de dados, relatórios sanitizados

Exemplos de Configurações:

Privacidade Máxima (Conformidade)

```
config :sms_c,
  delete_message_body_after_delivery: true,
  hide_message_body_in_ui: true,
  hide_message_body_in_export: true,
  cdr_enabled: true  # Manter CDRs sem corpos
```

Desenvolvimento (Visibilidade Total)

```
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 Inicialização

Na inicialização do aplicativo, o status da configuração é registrado:

```
[info] Armazenamento de mensagens: Mnesia (retenção: 24h)
[info] Exportação de CDR: HABILITADA
[info] Exclusão de corpo após entrega: DESABILITADA
[info] Cobrança OCS: HABILITADA (url: http://..., tenant: ...)
```

Isso fornece visibilidade imediata sobre os recursos ativos.

Federação Geográfica

← Voltar ao Índice da Documentação | **Referência de Configuração** | **Guia de Operações**

Visão Geral

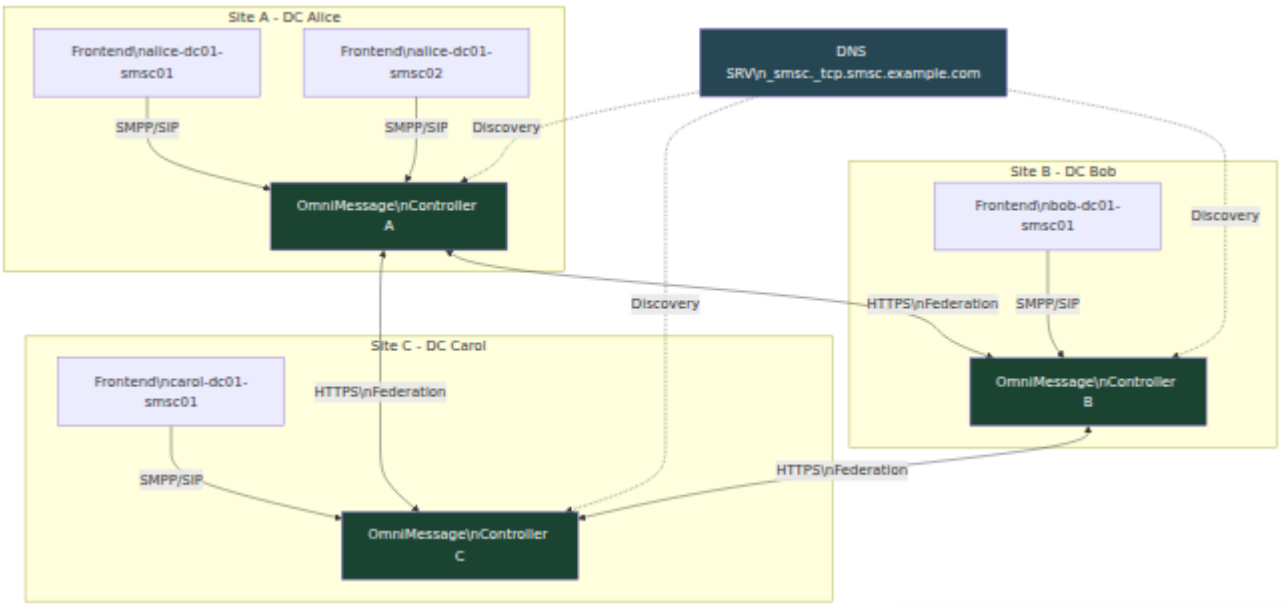
OmniMessage utiliza **federação baseada em HTTP** com um **modelo de pull** para implantações de múltiplos controladores em centros de dados ou regiões. Cada controlador opera de forma independente - gerenciando sua própria fila de mensagens, tabela de roteamento e conjunto de frontends conectados. Os controladores se descobrem através de registros DNS SRV (ou configuração estática), trocando informações de saúde e registro de frontends via HTTPS.

Quando o roteamento determina que uma mensagem pertence a um site remoto, a mensagem **permanece na Mnesia do controlador de origem**. Uma notificação leve é enviada ao controlador de destino, que aciona uma consulta imediata. O FederationPoller do controlador de destino periodicamente (e mediante notificação) consulta os pares de origem em busca de mensagens destinadas aos seus frontends locais, as armazena em cache e as disponibiliza para os frontends locais. Após a entrega, o controlador de destino reporta o status de entrega de volta ao originador.

Características Principais

Aspecto	Detalhe
Rede	Funciona sobre qualquer WAN, incluindo links não confiáveis
Estado compartilhado	Nenhum - cada controlador é independente
Propriedade da mensagem	As mensagens permanecem no controlador de origem até serem entregues
Segurança	HTTPS com TLS
Portas	Uma única porta HTTPS (8443)
Meta de escala	5-20 controladores
Tratamento de partições	Retentativas do Poller - sem split-brain

Arquitetura



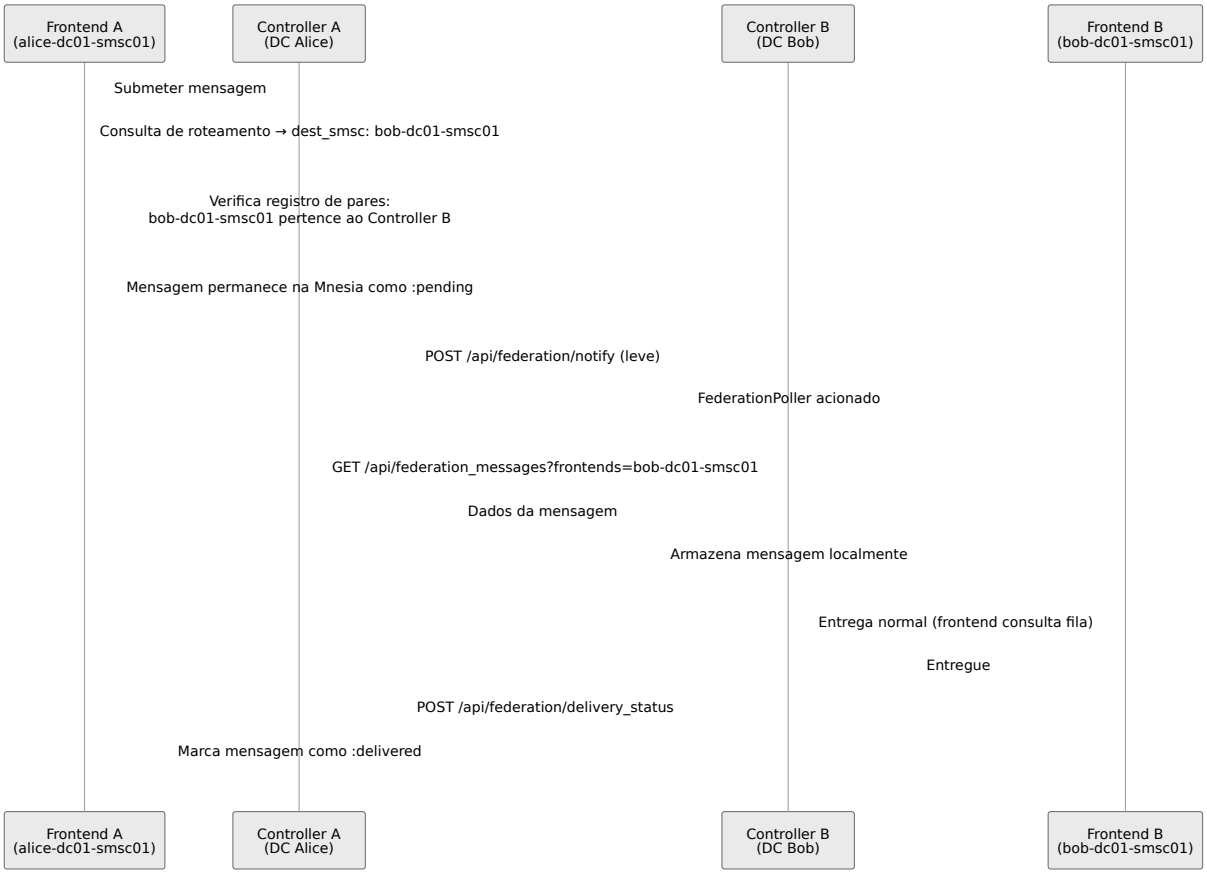
Decisões de Design Chave

- **Fluxo de mensagens (modelo de pull)**: As mensagens permanecem no controlador de origem. O controlador de destino as puxa via polling periódico e notificação sob demanda. Os frontends apenas se comunicam com seu controlador de origem.
- **Tratamento de partições**: Retentativas do Poller automaticamente. As mensagens permanecem seguras no controlador de origem até serem entregues com sucesso.
- **Sincronização de configuração**: Nenhuma. Cada controlador gerencia sua própria tabela de roteamento de forma independente.
- **Malha de pares**: Malha completa. Todos os controladores trocam informações de saúde e registros de frontends com todos os outros controladores.
- **Descoberta**: Registros DNS SRV (recomendado) ou lista de pares estática (para ambientes sem DNS dinâmico).

Fluxo de Mensagens (Modelo de Pull)

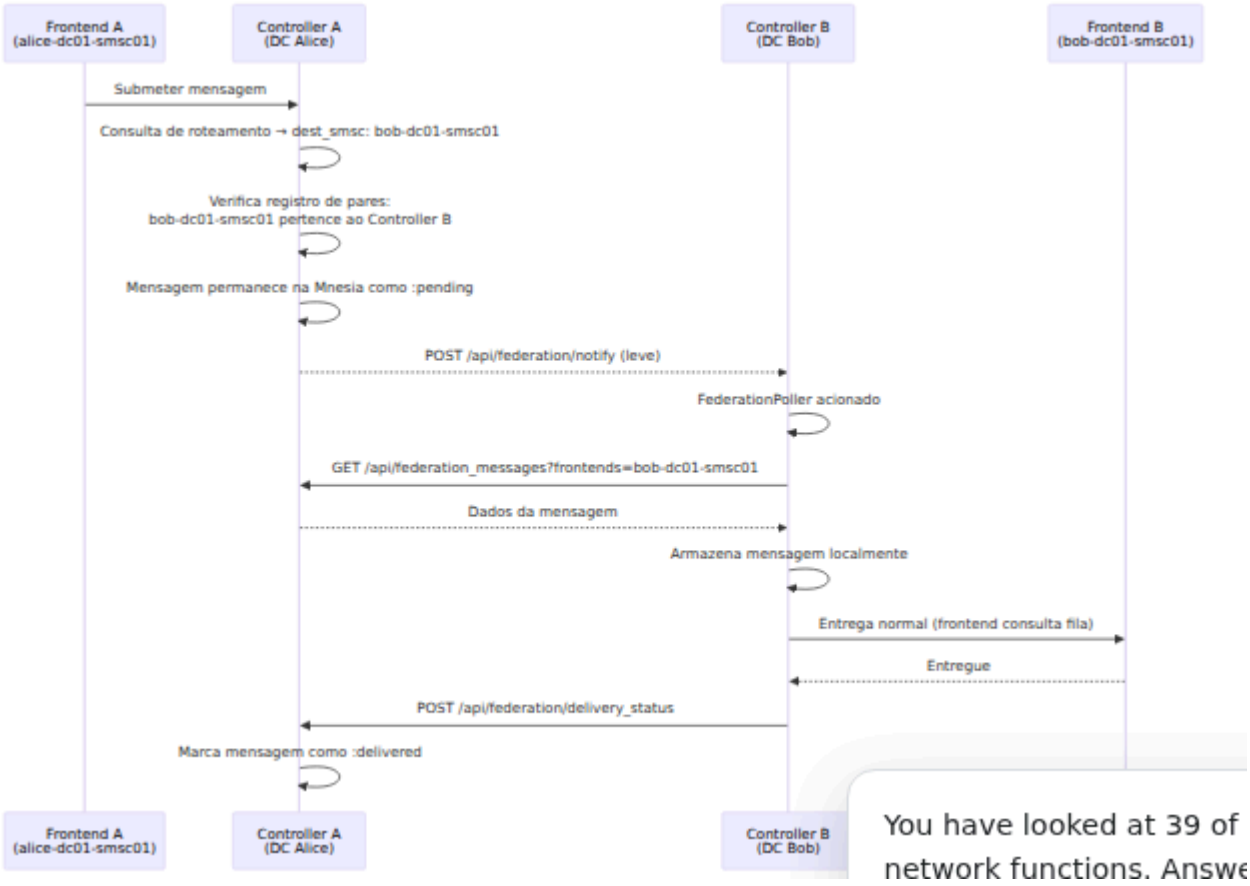
Quando uma mensagem chega a um controlador e o roteamento determina que o frontend de destino está em um controlador remoto:

1. A mensagem permanece na Mnesia do controlador de origem como `:pending`
2. Uma notificação leve é enviada ao controlador de destino (fire-and-forget)
3. O FederationPoller do controlador de destino busca a mensagem via GET /api/federation_messages
4. A mensagem é armazenada em cache localmente e disponibilizada para os frontends locais
5. Após a entrega, o destino reporta o status de volta ao originador



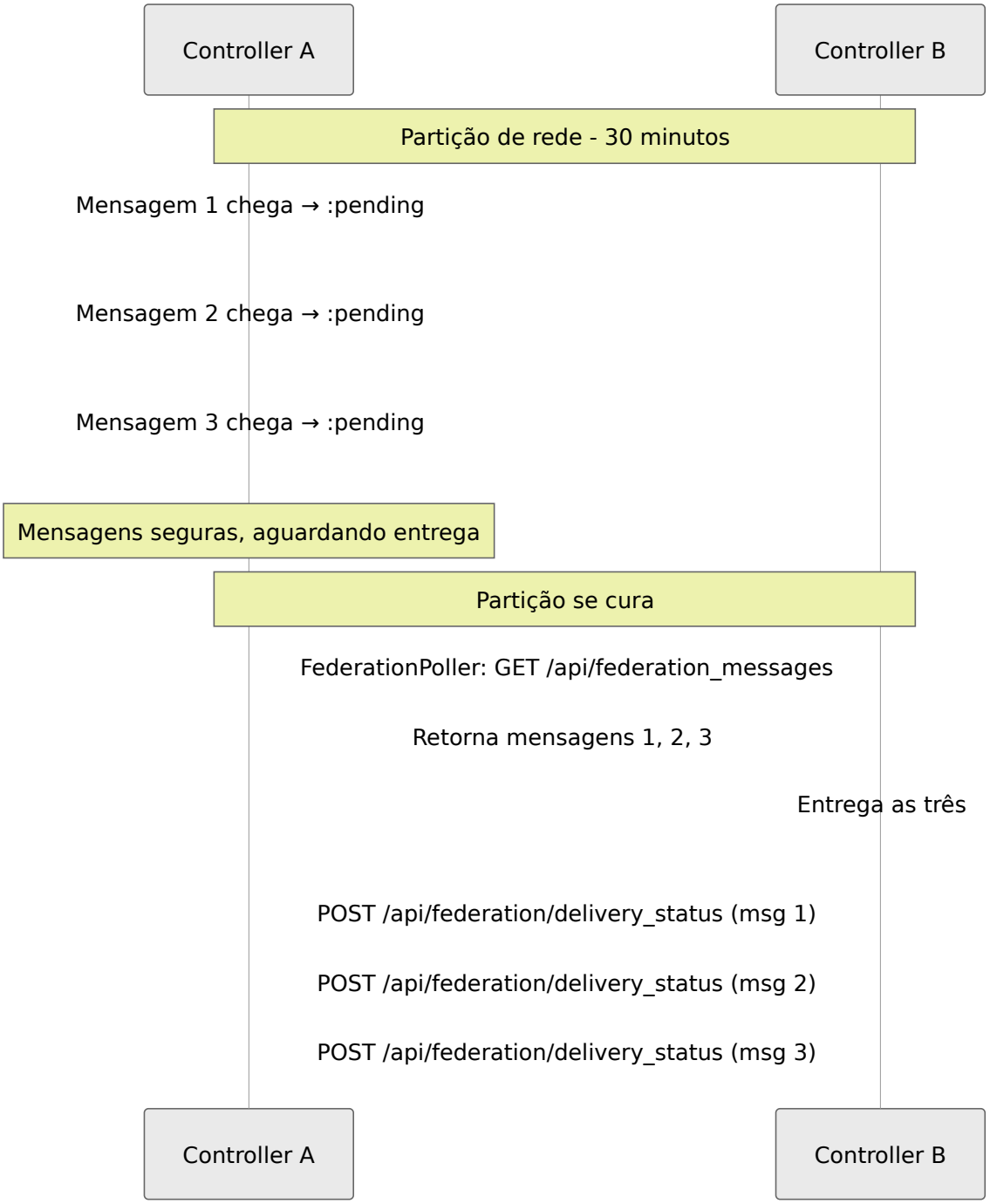
Par Inacessível - Falha na Notificação

Se o controlador de destino estiver fora do ar quando a notificação for enviada, a notificação falha silenciosamente. A mensagem permanece `:pending` no originador. Quando o controlador de destino se recupera, seu FederationPoller retoma o polling periódico e descobre a mensagem durante o próximo ciclo (padrão: a cada 5 segundos). Nenhuma mensagem é perdida.



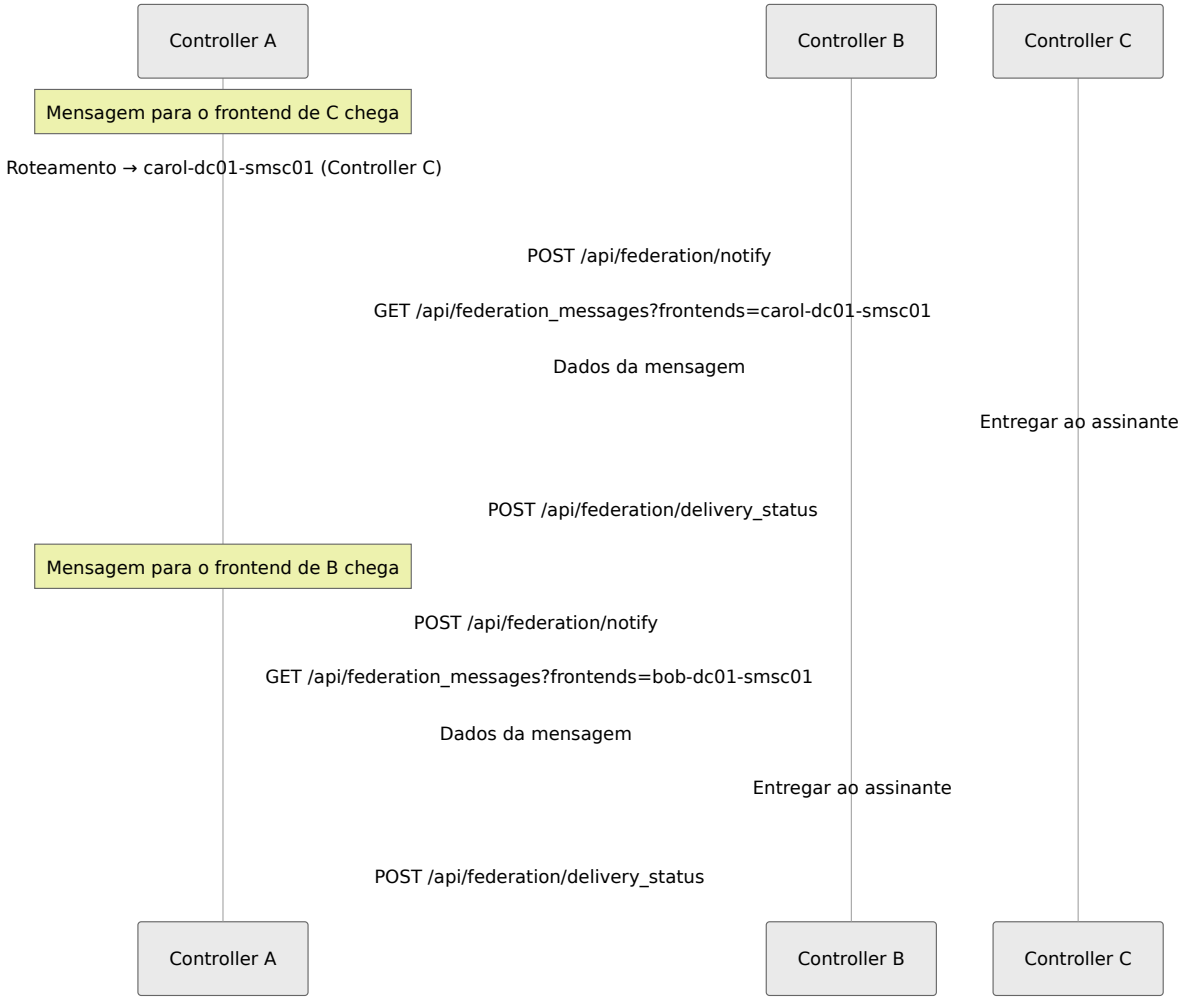
Recuperação de Partição - Múltiplas Mensagens em Espera

Durante uma partiç��o prolongada, mensagens se acumulam no controlador de origem. Quando o link se recupera, o FederationPoller se atualiza automaticamente - todas as mensagens pendentes são retornadas em uma única consulta.



Fluxo de Mensagens em Três Sites

Em uma malha de múltiplos sites, cada controlador apenas consulta mensagens destinadas aos seus próprios frontends.



Saúde dos Pares e Sincronização de Registro

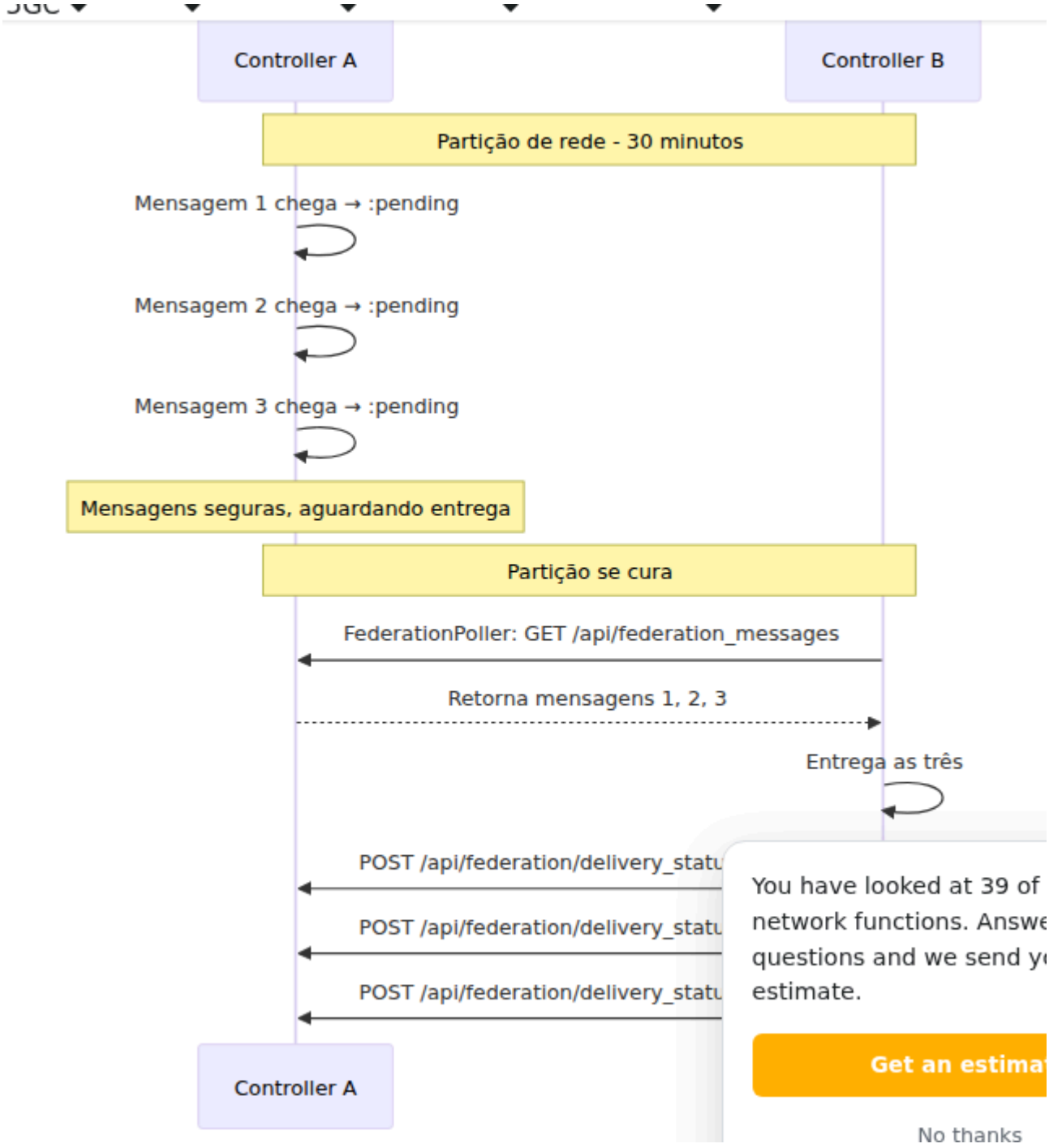
Os controladores de federação mantêm a consciência uns dos outros através de dois ciclos periódicos:

Ciclo de Verificação de Saúde

A cada 10 segundos (configurável), cada controlador chama `POST /api/federation/health` em cada par conhecido. A chamada é bidirecional - o chamador envia seu próprio status e recebe o status do par na resposta. Um par é marcado como **não saudável** após 3 falhas consecutivas.

Ciclo de Sincronização de Registro

A cada 15 segundos (configurável), cada controlador chama `POST /api/federation/registry` em cada par saudável. A chamada troca listas de frontends - o chamador envia seus frontends ativos e recebe os frontends ativos do par. É assim que os controladores aprendem qual par possui qual frontend.



Status dos pares e seu efeito no polling de federação:

Status	Verificações de Saúde	FederationPoller	Sincronização de Registro
Desconhecido	Em progresso	Não consultado	Não tentado
Saudável	Aprovado	Polling ativo	Ativo
Degradado	1-2 falhas	Não consultado	Não tentado
Não saudável	3+ falhas	Não consultado	Não tentado

Configuração

A federação é configurada em `config/runtime.exs` sob a chave `:federation`. Está **desativada por padrão** e deve ser explicitamente habilitada.

Configuração Mínima (Descoberta DNS SRV)

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

Referência Completa de Configuração

```
# config/runtime.exs
config :sms_c, :federation,
  # Interruptor mestre - a federação está completamente inativa
quando falso
  enabled: true,

  # Domínio DNS SRV para descoberta de pares (deixe vazio para
usar static_peers em vez disso)
  dns_srv_domain: "_smsc._tcp.smsc.example.com",

  # Com que frequência re-resolver registros DNS SRV
(milissegundos)
  dns_poll_interval_ms: 30_000,

  # Com que frequência trocar status de saúde com todos os pares
conhecidos (milissegundos)
  health_check_interval_ms: 10_000,

  # Com que frequência trocar registros de frontends com pares
saudáveis (milissegundos)
  registry_sync_interval_ms: 15_000,

  # Com que frequência o FederationPoller consulta pares saudáveis
por mensagens (milissegundos)
  poll_interval_ms: 5_000,

  # Timeout HTTP para todas as chamadas de API peer-to-peer
(milissegundos)
  http_timeout_ms: 5_000,

  # Porta da API usada ao construir URLs de pares a partir de
registros DNS SRV
  api_port: 8443,

  # Lista de pares estática - usada quando dns_srv_domain está
vazio
  static_peers: []
```

Parâmetros da Federação

Parâmetro	Tipo	Requerido	Padrão	Desc
`enabled`	Booleano	Sim	`false`	Interrupt mestre. `false`, c serviços federaçã iniciam, não desc ou conta pares.
`dns_srv_domain`	String	Não	`""`	Domínio SRV para descobe automát pares. Q vazio, `static_` é usado disso.
`dns_poll_interval_ms`	Inteiro	Não	`30000`	Intervalo tentativa re-resolu DNS SRV Valores r baixos detectar pares m rapidam mas aun a carga DNS.

Parâmetro	Tipo	Requerido	Padrão	Desc
`health_check_interval_ms`	Inteiro	Não	`10000`	Intervalo rodadas verificaç saúde. C rodada c cada par conhecid
`registry_sync_interval_ms`	Inteiro	Não	`15000`	Intervalo rodadas sincroniz de regist frontend rodada c cada par saudáve
`poll_interval_ms`	Inteiro	Não	`5000`	Com que frequênc Federati consulta saudáve mensage Notificaç acionam imediatc
`http_timeout_ms`	Inteiro	Não	`5000`	Timeout cada cha HTTPS individua par. Apli verificaç saúde, sincroniz

Parâmetro	Tipo	Requerido	Padrão	Desc
				de regist busca de mensag
`api_port`	Inteiro	Não	`8443`	Porta da usada ao construi de pares partir de registros SRV. Dev correspo porta da configur `config :api_ex`
`static_peers`	Lista	Não	`[]`	Lista de configur de pares ambient DNS SRV entrada mapa co chaves `port`.

Configuração de Par Estático

Para ambientes onde DNS SRV não está disponível, configure os pares explicitamente:

```
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}
  ]
```

Parâmetro	Tipo	Requerido	Padrão	Descrição
`host`	String	Sim	-	Endereço IP ou nome do host do controlador remoto.
`port`	Inteiro	Não	`8443`	Porta da API HTTPS do controlador remoto.

Configuração de Registro DNS SRV

Os registros DNS SRV permitem que os controladores se descubram automaticamente. Cada controlador resolve o domínio configurado e usa os pares de host/porta resultantes como pares.

Configuração da Zona DNS:

```
; Prioridade Peso Porta Destino
_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 os destinos
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
```

Adicionando um novo site: Adicione um novo registro SRV ao DNS. Todos os controladores existentes descobrirão o novo par dentro de um ciclo de `dns_poll_interval_ms` (padrão 30 segundos).

Removendo um site: Remova o registro SRV do DNS. Controladores existentes removerão o par que partiu dentro de um ciclo de polling.

Requisitos de Rede

A federação requer apenas uma única porta HTTPS entre os sites.

Porta	Protocolo	Direção	Propósito
8443	TCP/TLS	Bidirecional	Todo o tráfego de federação (saúde, registro, notificação, mensagens, status de entrega)
53	UDP/TCP	Saída	Resolução DNS SRV (se usando descoberta DNS)

Regras de firewall (por site):

```
# Permitir tráfego de federação dos IPs dos 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
```

Endpoints da API de Federação

O controlador serve todos os endpoints de federação na porta padrão da API (8443) sob `/api/federation/`. Controladores pares chamam esses endpoints, não clientes externos ou frontends.

Endpoint	Método	Propósito
`/api/federation/identity`	GET	Retorna o nome do nó deste controlador e a versão do protocolo
`/api/federation/health`	POST	Troca bidirecional de saúde - o chamador envia status, o chamado responde com o seu próprio
`/api/federation/registry`	POST	Troca bidirecional de registros de frontends
`/api/federation/notify`	POST	Receber notificação leve de que uma mensagem está disponível para um frontend local
`/api/federation/delivery_status`	POST	Receber confirmação de entrega do controlador de destino
`/api/federation/delivery_attempt`	POST	Receber uma tentativa de entrega falhada do controlador de destino; executa o backoff de retentativa e retorna a mensagem atualizada
`/api/federation/event`	POST	Receber um evento de mensagem que um frontend do controlador de destino relatou; registra no log de eventos local

Endpoint	Método	Propósito
`/api/federation_messages`	GET	Retorna mensagens pendentes para frontends especificados (usado pelo FederationPoller)

Todas as requisições e respostas usam JSON. A identificação do par é transportada no cabeçalho `X-Federation-Node`.

Exemplos de Implantação

Exemplo 1: Implantação em Três Sites com DNS SRV

Três centros de dados, cada um com frontends locais para sua base de assinantes. DNS SRV fornece descoberta.

```
# Site 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"
```

```
# Site 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"
```

```
# Site 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"
```

Como funciona: Todos os três controladores resolvem o mesmo domínio SRV e se descobrem automaticamente. Cada controlador registra seus frontends locais. Quando uma mensagem chega ao DC Alice destinada a um assinante atendido pelo DC Bob, a tabela de roteamento seleciona `bob-dc01-smsc01` como destino. A camada de federação detecta que este frontend pertence ao Controller B e envia uma notificação. O FederationPoller do Controller B consulta o Controller A e armazena a mensagem em cache para entrega local.

Exemplo 2: Dois Sites Ativos-Ativos com Pares Estáticos

Dois centros de dados conectados por um link dedicado. Sem DNS SRV disponível.

```
# DC Primário (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 Secundário (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}
  ]
```

Como funciona: Cada controlador é configurado com o endereço IP do outro. As verificações de saúde e as sincronizações de registro ocorrem através do link dedicado. As mensagens permanecem no controlador que as recebeu. O FederationPoller de cada controlador consulta o par por mensagens destinadas aos frontends locais.

Exemplo 3: Polling Agressivo para Links de Baixa Latência

Para sites conectados por links confiáveis e de baixa latência onde uma federação rápida é 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: Implantações em campus ou áreas metropolitanas onde os sites estão conectados por links de < 10 ms RTT e uma rápida recuperação é mais importante do que minimizar o tráfego de controle.

Métricas

A federação expõe os seguintes eventos de telemetria que podem ser observados via Prometheus na porta 9568.

Métrica: `sms_c_federation_message_received_count` **Tipo**: Contador **Descrição**: Número de notificações de federação recebidas de pares **Rótulos**:

- `origin_node` - Par que enviou a notificação

Métrica: `sms_c_federation_health_check_count` **Tipo**: Contador **Descrição**: Resultados de verificação de saúde por par **Rótulos**:

- `peer` - Identificador do par
- `result` - `ok` ou `failed`

Consultas de exemplo:

```
# Taxa de notificações por minuto
rate(sms_c_federation_message_received_count[5m]) * 60

# Falhas de verificação de saúde por par
sum by (peer)
(rate(sms_c_federation_health_check_count{result="failed"}[5m]))
```

Indicadores Chave a Monitorar

O que Observar	Onde	Limite de Alerta
Mensagens federadas em cache	`GET /api/federation/status` ou UI Web	Contagem crescente (par não entregando)
Status de saúde dos pares	`GET /api/federation/identity` em cada par	Qualquer par não saudável > 5 minutos
Latência de verificação de saúde	Logs da aplicação	> 2 segundos (degradação do link)

Solução de Problemas

Par Não Descoberto

Sintomas: Logs do controlador mostram que nenhum par foi descoberto; `cluster_status` retorna vazio.

Possíveis causas:

- Domínio DNS SRV está incorreto ou não resolvível
- Servidor DNS está inacessível a partir do host do controlador
- Registros SRV ainda não se propagaram
- A federação não está habilitada (`enabled: false`)

Resolução:

1. Verifique a resolução DNS SRV a partir do host do controlador: `dig SRV _smsc._tcp.smsc.example.com`
2. Verifique se `enabled: true` está configurado na configuração de federação

3. Verifique os logs da aplicação para mensagens de "Descoberta DNS de Federação" na inicialização
4. Se estiver usando pares estáticos, verifique se a lista `static_peers` contém entradas corretas de host/porta

Marcação de Saúde do Par

Sintomas: Mensagens para um site específico não estão sendo consultadas. Os logs mostram mensagens de "Verificação de saúde falhou".

Causas possíveis:

- O controlador remoto está fora do ar
- Firewall bloqueando a porta 8443 entre os sites
- Problemas com o certificado TLS
- Partição de rede entre os sites

Resolução:

1. Verifique se o controlador remoto está em execução: `curl -k https://<peer-host>:8443/api/federation/identity`
2. Verifique se as regras do firewall permitem tráfego na porta 8443 a partir do IP do controlador local
3. Revise os logs da aplicação em ambos os lados em busca de erros de TLS ou de conexão
4. Verifique a conectividade da rede entre os sites

Mensagens Não Estão Sendo Entregues ao Site Remoto

Sintomas: Mensagens permanecem `:pending` no controlador de origem. A contagem de mensagens federadas em cache é 0 no destino.

Causas possíveis:

- O controlador de destino está com problemas (o FederationPoller apenas consulta pares saudáveis)

- O FederationPoller não está em execução
- O registro do frontend não está sincronizado (o destino não sabe quais frontends são locais)

Resolução:

1. Verifique o status do par no controlador de destino
2. Verifique se o FederationPoller está na árvore de supervisão (verifique os logs da aplicação para "Federation poller started")
3. Verifique se os registros dos frontends estão sincronizados (`GET /api/federation/status` - verifique as listas de frontends)
4. Teste manualmente o endpoint federation_messages: `curl -k 'https://<origin>:8443/api/federation_messages?frontends=<frontend_name>&include_unrouted=false'`

Consulta de Assinante HSS (Diameter Sh)

← Voltar para o Índice de Documentação | **Guia de Roteamento SMS**

Visão Geral

OmniMessage pode consultar o Servidor de Assinantes (HSS) via a interface Diameter Sh para determinar se um MSISDN pertence à rede (on-net) ou não (off-net). Esta resposta é usada como um **critério de correspondência de rota**: as rotas podem exigir que o número de origem ou de destino esteja on-net ou off-net antes de serem aplicadas.

A consulta é **preguiçosa** - ocorre *durante a avaliação da tabela de rotas*, apenas quando uma rota em avaliação declara um requisito de `originating_network` ou `terminating_network` de `:on_net` ou `:off_net`. Rotas que deixam ambos como `:any` (o padrão) nunca acionam uma consulta. Não há uma consulta global ao HSS realizada antes do roteamento.

OmniMessage envia um **User-Data-Request (UDR)** para o número. Se o HSS confirmar que o assinante existe (código de resultado 2001), o número é **on-net**; se o HSS retornar *Usuário Desconhecido* (código de resultado 5001), o número é **off-net**. Veja Guia de Roteamento SMS → Correspondência On-Net / Off-Net para como esses critérios se combinam com o restante de uma rota.

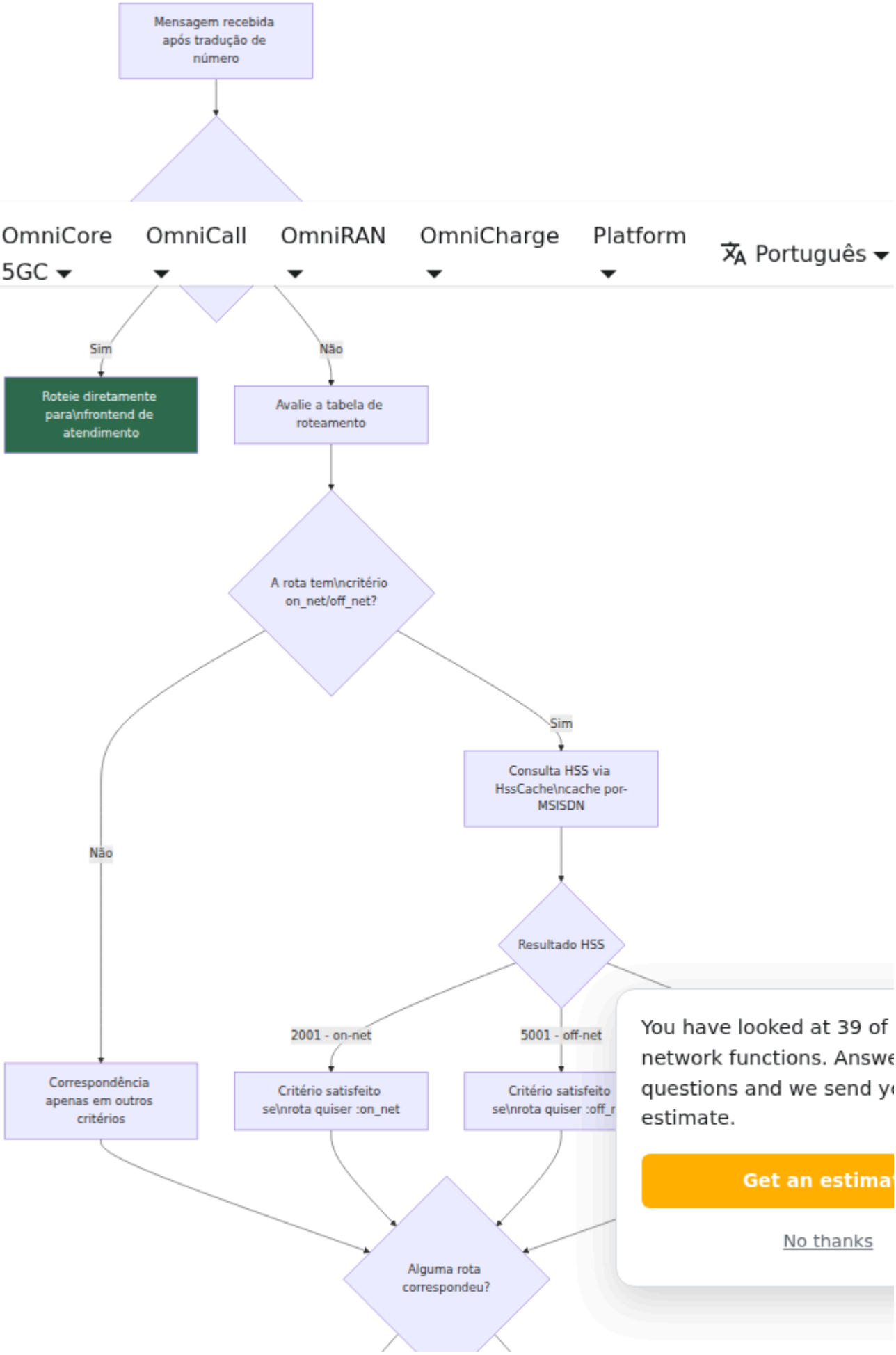
> **On-net vs off-net - HSS ou ENUM?** O HSS (Sh) responde "este número é um assinante em *nosso* HSS?" com um resultado explícito on/off-net. ENUM (DNS NAPTR) responde "a qual domínio de operadora este número resolve?" e você deriva on/off-net correspondendo ao seu próprio domínio. Use Sh quando seu HSS for autoritativo; use ENUM para portabilidade/roteamento entre operadoras. Veja Guia de Roteamento SMS → Consulta ENUM.

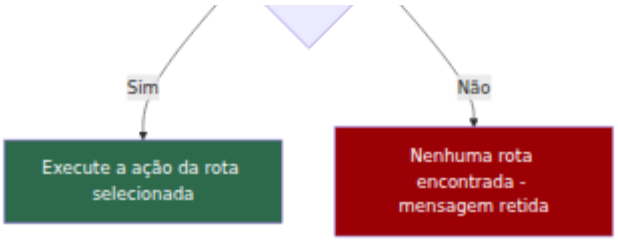
Especificações Relevantes

Especificação	Título
3GPP TS 29.329	Interface Sh baseada em Diameter
3GPP TS 29.328	Interface IMS Sh - fluxos de sinalização e conteúdos de mensagens
RFC 6733	Protocolo Base Diameter

Onde se Encaixa a Consulta HSS

OmniMessage avalia uma mensagem nesta ordem:


Mensagem recebida após tradução de número
OmniCore 5GC
OmniCall
OmniRAN
OmniCharge
Platform
Português
Sim
Não
Roteie diretamente para\nfrontend de atendimento
Avalie a tabela de roteamento
A rota tem\ncritério on_net/off_net?
Sim
Não
Consulta HSS via HssCache\ncache por-MSISDN
Resultado HSS
2001 - on-net
5001 - off-net
Correspondência apenas em outros critérios
Critério satisfeito se\nrota quiser :on_net
Critério satisfeito se\nrota quiser :off_r
Alguma rota correspondeu?
You have looked at 39 of
network functions. Answe
questions and we send y
estimate.
Get an estima
No thanks



Roteamento Baseado em Localização (verificado primeiro)

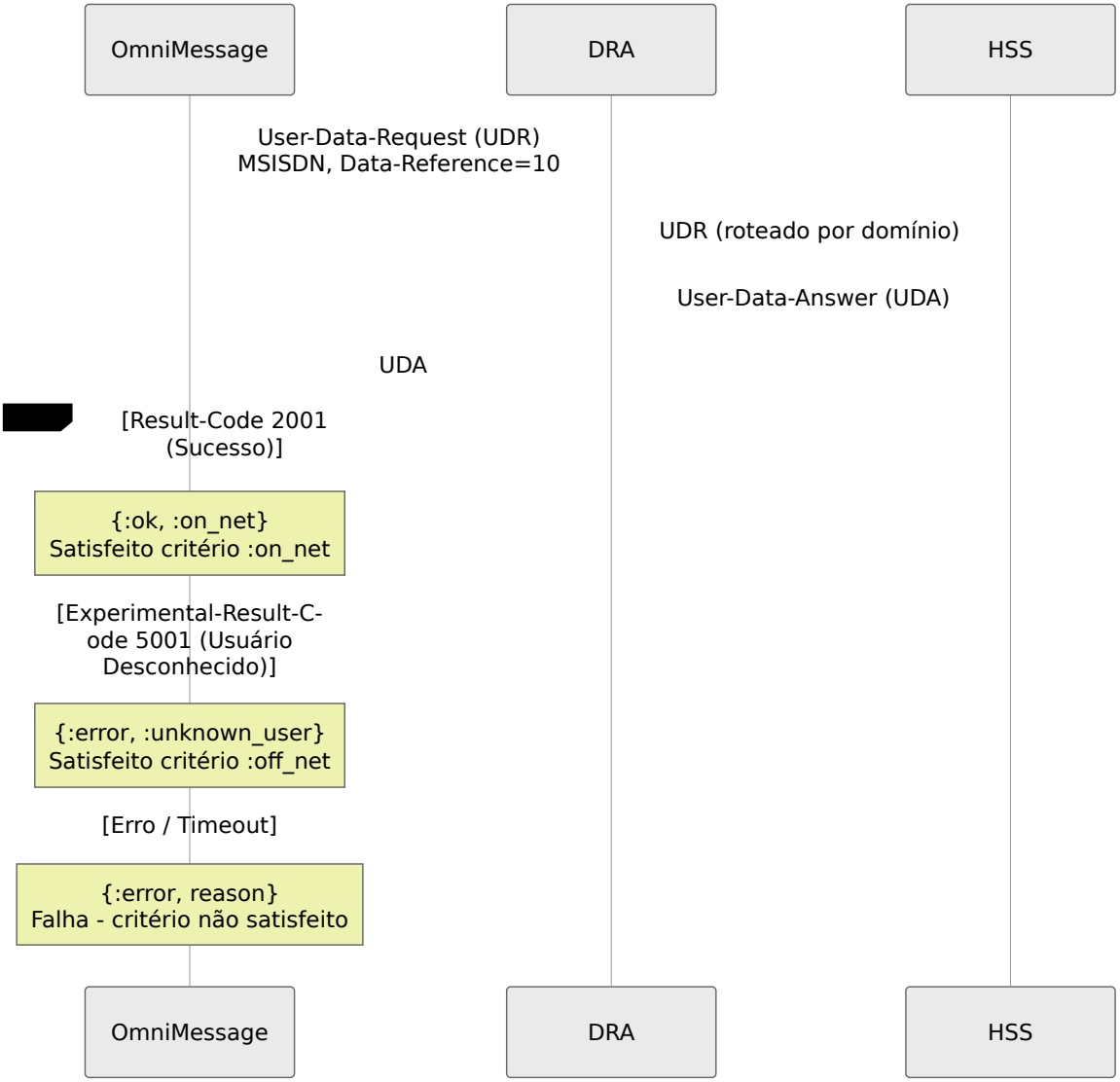
Se o MSISDN de destino for encontrado na tabela de registros/localizações, a mensagem é roteada diretamente para o frontend que atende aquele assinante. Nenhuma consulta ao HSS ou avaliação da tabela de roteamento ocorre. (Veja Expiração de Localização.)

Tabela de Roteamento + Consulta HSS Preguiçosa

Se não houver registro ativo, o roteador avalia a tabela de roteamento. Para cada rota que declara `originating_network`/`terminating_network` de `:on_net` ou `:off_net`, o roteador realiza uma consulta HSS (via cache) como parte da correspondência daquela rota. Uma rota só corresponde se satisfizer seu critério de rede juntamente com todos os seus outros critérios. Se o Diameter estiver desativado ou o HSS não fornecer uma resposta definitiva, o critério falha (fail-closed) e a rota não corresponde.

Reter uma mensagem para **armazenar e encaminhar** é um conceito separado: a *ação* da rota `:saf` a aciona, não a consulta HSS. Veja Entrega Armazenar e Encaminhar.

Fluxo de Mensagens Diameter Sh



Parâmetros UDR

AVP	Valor	Descrição
Session-Id	Auto-gerado	Identificador único da sessão
Auth-Session-State	1 (NO_STATE_MAINTAINED)	Nenhum estado de sessão necessário para consultas
Data-Reference	10	Solicitar todos os dados do usuário
Origin-Host	Host configurado + domínio	Identidade Diameter do OmniMessage
Origin-Realm	Domínio configurado	Domínio Diameter do OmniMessage
Destination-Realm	Domínio configurado	Domínio HSS (roteado via DRA)
User-Identity	MSISDN codificado em TBCD	O número do assinante
Server-Name	"OmniMessage"	Identidade do servidor de aplicação
Vendor-Specific-Application-Id	Auth: 16777217, Vendor: 10415	Identificador da aplicação Sh por 3GPP TS 29.329

Códigos de Resultado UDA

A verificação da existência do assinante mapeia o UDA para um dos três resultados:

Código	Tipo	Resultado	Significado para correspondência de rota
2001	Result-Code	`{:ok, :on_net}`	On-net - satisfaz `:on_net`, falha `:off_net`
5001	Experimental-Result-Code	`{:error, :unknown_user}`	Off-net - satisfaz `:off_net`, falha `:on_net`
Outro / timeout / sem pares	-	`{:error, reason}`	Falha - não satisfaz nenhum; a rota não corresponde

Configuração

Habilitando a Consulta HSS

Dois blocos de configuração em `config/runtime.exs` controlam a consulta HSS:

1. A flag `diameter_enabled` sob `:sms_c` - controla se as consultas HSS são realizadas durante a correspondência de rotas. Quando `false`, qualquer rota com um critério `:on_net`/`:off_net` falha em corresponder (fail-closed).
2. A configuração `:diameter_ex` - configura a pilha Diameter em si (identidade, pares, aplicações)

Configuração Mínima

```
# Habilitar consultas HSS durante a correspondência de rotas
config :sms_c,
  diameter_enabled: true

# Configuração da pilha 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
      }
    ]
  }
```

Parâmetros OmniMessage

Parâmetro	Tipo	Obrigatório	Padrão	Descr
`diameter_enabled`	Booleano	Não	`false`	Habilitar c HSS duran correspon de rotas. C `false`, rot critérios `:on_net`/`:` falham.
`hss_cache_ttl_seconds`	Inteiro	Não	`7200`	TTL (segur para result consulta H cache. Def em `runtim` (não expos esquema Y operador).
`mock_sh`	Booleano	Não	`false`	Usar respo HSS simul vez de cor Diameter r Apenas pa testes.
`mock_sh_on_net_numbers`	Lista	Não	`[]`	Lista de M tratados c net quand `mock_sh` é

Parâmetros da Pilha Diameter

Parâmetro	Tipo	Obrigatório
`service_name`	Átomo	Sim
`listen_ip`	String	Não
`listen_port`	Inteiro	Não
`decode_format`	Átomo	Sim
`host`	String	Sim
`realm`	String	Sim
`product_name`	String	Não
`request_timeout`	Inteiro	Não
`peer_selection_algorithm`	Átomo	Não

Parâmetro	Tipo	Obrigatório
`allow_undefined_peers_to_connect`	Booleano	Não
`log_unauthorized_peer_connection_attempts`	Booleano	Não
`control_module`	Módulo	Sim
`processor_module`	Módulo	Sim
`auth_application_ids`	Lista	Não
`acct_application_ids`	Lista	Não
`vendor_id`	Inteiro	Sim
`supported_vendor_ids`	Lista	Sim

Parâmetros da Aplicação

A lista `applications` configura quais aplicações Diameter estão habilitadas. Para consulta HSS, apenas a aplicação Sh é necessária.

Parâmetro	Tipo	Obrigatório	Padrão	
`application_name`	Átomo	Sim	-	Ide apl par
`application_dictionary`	Átomo	Sim	-	Mó Dia ser `:d`
`vendor_specific_application_ids`	Lista	Sim	-	Ide apl for

Identificador de Aplicação Específico do Fornecedor (Sh):

Parâmetro	Tipo	Valor	Descrição
`vendor_id`	Inteiro	`10415`	ID do fornecedor 3GPP
`auth_application_id`	Inteiro	`16_777_217`	ID da aplicação Sh por 3GPP TS 29.329
`acct_application_id`	-	`nil`	Não utilizado para Sh

Parâmetros de Par

A lista `peers` define os pares Diameter (tipicamente DRAs ou o HSS diretamente) aos quais o OmniMessage se conecta.

Parâmetro	Tipo	Obrigatório	Padrão	De
`host`	String	Sim	-	Identi Host-D do par Deve corres exatar identi config par.
`ip`	String	Sim	-	Ender par pa conex
`port`	Inteiro	Não	3868	Porta do par
`realm`	String	Sim	-	Domír Diame
`tls`	Booleano	Não	`false`	Habilit para e conex
`transport`	Átomo	Não	`:diameter_tcp`	Protoc transp `:diam` ou `:diam`
`initiate_connection`	Booleano	Não	`true`	Se `tru` OmniN inicia TCP cc Se `fa`

Parâmetro	Tipo	Obrigatório	Padrão	De
				OmniM espera conect

Múltiplos Pares

Configure múltiplos pares para redundância. O OmniMessage seleciona entre pares conectados usando o `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 ambientes de teste sem um HSS ativo, habilite o modo simulado:

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

No modo simulado, qualquer MSISDN na lista `mock_sh_on_net_numbers` resolve para `{:ok, :on_net}` (equivalente ao código de resultado 2001). Todos os outros MSISDNs resolvem para `{:error, :unknown_user}` (equivalente ao código de resultado 5001). Nenhuma pilha Diameter é iniciada e nenhuma conexão de rede é feita.

Cache de Resultados HSS

As consultas HSS são armazenadas em cache por MSISDN pelo `SmsC.Diameter.HssCache` para evitar viagens redundantes de Diameter:

- **TTL**: `hss_cache_ttl_seconds` (padrão 7200 = 2 horas)
- **Compartilhado**: múltiplas rotas que precisam do status do mesmo MSISDN em uma avaliação compartilham uma única consulta
- **Atualização forçada**: o cache é ignorado (uma nova consulta é feita) quando uma mensagem é explicitamente re-roteada
- **Predicados**: um resultado é tratado como on-net apenas para `{:ok, :on_net}` e como off-net apenas para `{:error, :unknown_user}`; todo outro resultado (incluindo `nil` quando o Diameter está desativado) é tratado como nenhum - o que produz correspondência fail-closed.

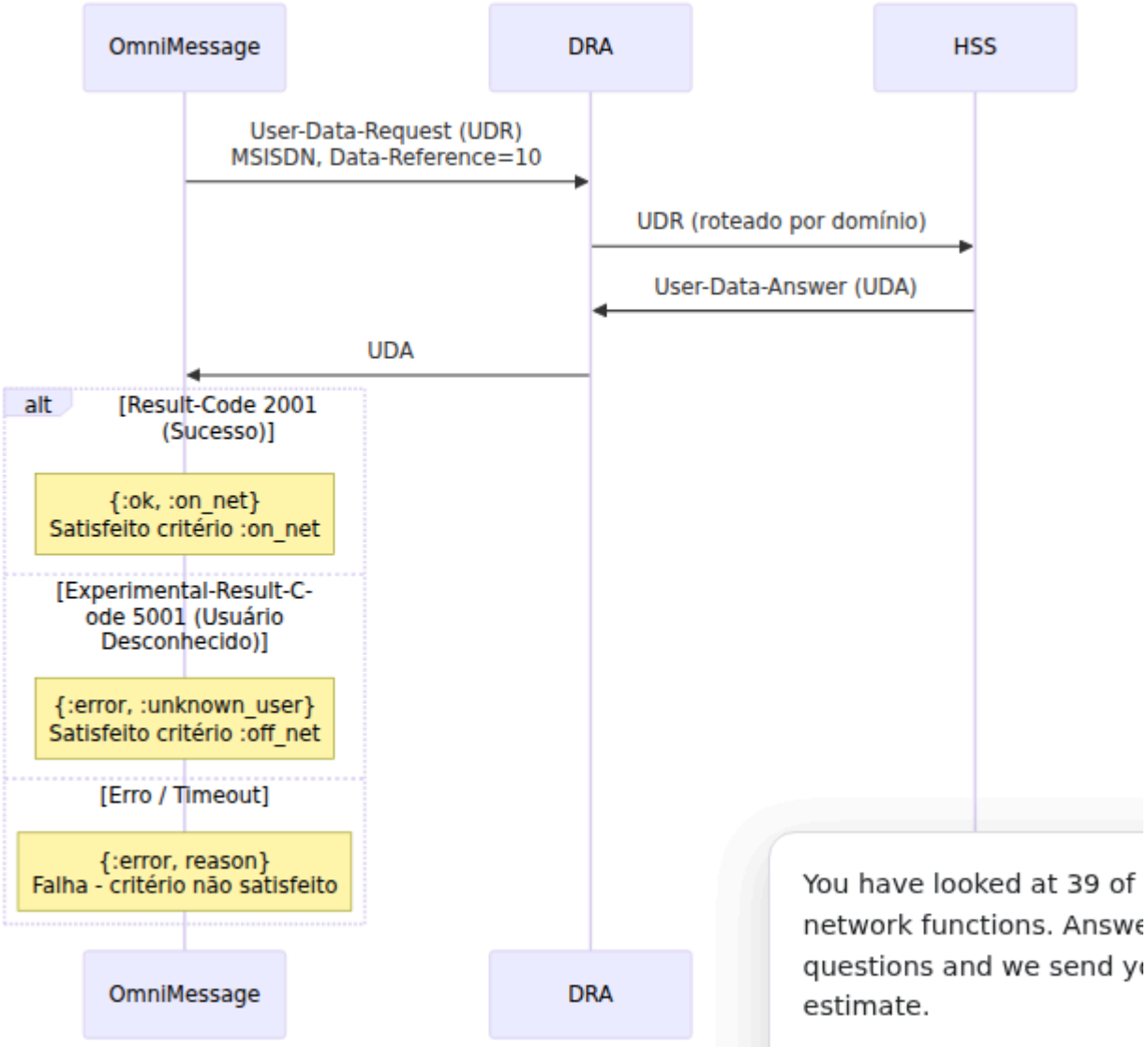
Entrega Armazenar e Encaminhar

A **ação** da rota `:saf` impulsiona o armazenamento e encaminhamento, não a consulta HSS. Uma rota com `action: :saf` retém a mensagem para entrega posterior. Para reter *apenas* assinantes on-net, combine `:saf` com

`terminating_network: :on_net` e forneça uma rota de fallback off-net (veja Guia de Roteamento SMS → `saf`).

Quando uma mensagem é retida para SAF, ela permanece na fila de mensagens Mnesia com `dest_smsc` não definido. Quando o assinante se registra e uma atualização de localização chega, o loop de entrega pega mensagens pendentes para aquele MSISDN e as roteia para o frontend de atendimento.

Mensagens na fila estão sujeitas à política de retenção padrão. Se o assinante não se registrar antes que a mensagem expire, a mensagem é movida para a fila de cartas mortas de acordo com o `dead_letter_time_minutes` configurado.



Notas Operacionais

Registro de Inicialização

OmniMessage registra o status Diameter Sh na inicialização:

- **Habilitado**: `Diameter Sh (HSS): HABILITADO (host: smsc01, realm: epc.mnc001.mcc001.3gppnetwork.org, pares: 2)`
- **Desabilitado**: `Diameter Sh (HSS): DESABILITADO`

Comportamento Fail-Closed

Se uma consulta HSS não retornar uma resposta definitiva - timeout, sem pares conectados, resposta não tratada ou um código de resultado inesperado - `subscriber_exists?/1` retorna `{:error, reason}`. Para uma rota que requer `:on_net` ou `:off_net`, esse critério **não** é satisfeito, portanto a rota **não corresponde** e a avaliação continua com a próxima rota.

Este é um design **fail-closed**: sem uma resposta definitiva, o OmniMessage não tratará um número como on-net ou off-net, portanto uma rota apenas on-net não será acionada e uma rota apenas off-net também não será acionada. Forneça uma rota de fallback catch-all (`terminating_network: :any`) se você quiser que as mensagens sejam roteadas durante interrupções do HSS.

Quando **Diameter está explicitamente desativado** (`diameter_enabled: false`), a consulta em cache retorna `nil` e a mesma correspondência fail-closed se aplica a qualquer rota com um critério `:on_net`/`:off_net`. Rotas que deixam ambos os campos de rede em `:any` não são afetadas.

Observabilidade

As consultas HSS são observáveis através de:

- **Registros de rota por mensagem** - cada evento `route_evaluated` da rota inclui a verificação de rede, por exemplo, `term_network: ✓ (terminating requires on_net, +15551234 is on-net)` ou `✗ (... is`

`off-net)`. Visível no log de atividade de mensagens do painel de controle e na API de Eventos.

- **Telemetria Diameter Sh** - `SmsC.Telemetry` emite métricas de requisições, respostas (rotuladas por resultado), duração e erros (veja METRICS.md).
- **Registros da aplicação** - linhas `HSS dip: <msisdn> is ON-NET (2001)` / `is NOT in HSS (5001)` do `SmsC.Diameter.Sh`.

Expiração de Localização

As verificações de roteamento baseadas em localização verificam o timestamp `expires` nos registros de assinantes. Se o registro de um assinante tiver expirado, por exemplo, porque ele entrou no modo avião, o roteador trata a localização como inativa. A avaliação então prossegue para a tabela de roteamento, incluindo qualquer consulta HSS preguiçosa que uma rota correspondente exija.

Isso impede que mensagens sejam enviadas para um frontend que não pode mais alcançar o assinante, evitando falhas de entrega desnecessárias e ciclos de tentativas.

Aplicação de On-Net / Off-Net em Nível de Rota

As rotas aplicam requisitos on-net/off-net através de dois campos - **não** flags booleanas e **não** uma ação de rota:

Campo	Valores	Verificações	Comportamento
`originating_network`	`:any` | `:on_net` | `:off_net`	Número chamador	A rota corresponde apenas se o status HSS do MSISDN de origem corresponder (fail-closed)
`terminating_network`	`:any` | `:on_net` | `:off_net`	Número chamado	A rota corresponde apenas se o status HSS do MSISDN de destino corresponder (fail-closed)

O padrão para ambos é `:any` (sem verificação HSS). Um valor de `:on_net` requer código de resultado 2001; `:off_net` requer 5001; qualquer resposta indefinida falha no critério.

Exemplo de Configuração

```
config :sms_c, :sms_routes, [
  # Destinos on-net apenas - entregar para gateway 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: "Gateway local - destinos on-net apenas",
    originating_network: :any,
    terminating_network: :on_net,
    enabled: true
  },

  # Fallback off-net - catch-all para tudo que a rota on-net não
pegou
  %{
    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: "Gateway externo - fallback",
    originating_network: :any,
    terminating_network: :any,
    enabled: true
  }
]
```

Como funciona: para uma mensagem para um assinante não registrado:

1. **Verificação de localização** - nenhum registro ativo encontrado

2. **Avaliação de rota** - as rotas são verificadas em ordem de prioridade
3. A rota P10 "local-gateway" tem `terminating_network: :on_net`, então uma consulta HSS é realizada no número chamado. Se o HSS retornar 2001 (on-net), a rota corresponde e a mensagem é entregue ao `local-gateway`. Se retornar 5001 (off-net) ou falhar (fail-closed), a rota não corresponde.
4. A rota P255 "external-gateway" tem `terminating_network: :any`, então não precisa de resposta HSS e corresponde como fallback.

Caso de uso: manter o tráfego on-net na infraestrutura local enquanto o tráfego off-net (ou incerto do HSS) atinge um transportador externo. Para *manter* assinantes on-net para armazenamento e encaminhamento em vez de entregar a um gateway, defina a `action` da rota P10 como `:saf`.

Solução de Problemas

Assinantes On-Net Roteados para o Gateway de Fallback

Sintomas: Mensagens para assinantes on-net seguem a rota off-net/fallback em vez da rota on-net.

Possíveis causas:

- `diameter_enabled` está `false` - o critério `:on_net` falha em fechar, então a rota on-net nunca corresponde
- Nenhum par Diameter está conectado (verifique os logs da aplicação para o status de conexão do par)
- O HSS está retornando erros ou expirando (verifique a telemetria de erro Sh e as linhas de log "HSS dip")
- A tradução de número está produzindo um formato de número diferente do que o HSS espera

Resolução:

1. Verifique `diameter_enabled: true` em `runtime.exs`

2. Verifique os logs para o status de conexão do par Diameter na inicialização (`Diameter Sh (HSS): ENABLED (... peers: N)`)
3. Procure por linhas de log `HSS dip:` e métricas de erro Sh
4. Confirme que o formato do número após a tradução corresponde ao que o HSS espera (E.164 sem `+`)

Mensagens Retidas Mas Nunca Entregues

Sintomas: Mensagens estão corretamente retidas (`:saf`) para assinantes on-net, mas nunca entregues, mesmo após o assinante se registrar.

Possíveis causas:

- Atualizações de localização não estão chegando ao OmniMessage
- O MSISDN no registro de localização não corresponde ao `destination_msisdn` da mensagem em fila
- As mensagens expiraram e foram movidas para a fila de cartas mortas

Resolução:

1. Verifique se as atualizações de localização/registro estão sendo recebidas (verifique a página de Registros no Painel de Controle)
2. Confirme que o formato do MSISDN é consistente entre a fila de mensagens e a tabela de registro
3. Verifique `dead_letter_time_minutes` - aumente se os assinantes puderem estar offline por períodos prolongados

Par Diameter Não Conectando

Sintomas: Os logs não mostram conexões de pares Diameter na inicialização.

Possíveis causas:

- IP ou porta do par incorretos na configuração
- Firewall bloqueando a porta 3868
- Desconexão de identidade do host do par (o `host` configurado deve corresponder exatamente à identidade Diameter do par)

- Desconexão de domínio entre OmniMessage e o par

Resolução:

1. Verifique se o endereço IP e a porta do par estão corretos
2. Confirme que as regras do firewall permitem tráfego TCP na porta configurada
3. Verifique se o valor `host` do par corresponde exatamente ao que o par anuncia em sua troca de capacidade
4. Certifique-se de que o `realm` corresponda entre as configurações do OmniMessage e do par

Documentação de Métricas Prometheus do SMS-C

← Voltar para o Índice da Documentação

Visão Geral

Este documento descreve todas as métricas Prometheus expostas pelo sistema SMS-C. Essas métricas são projetadas para que a equipe de operações monitore a saúde do sistema, o desempenho e resolva problemas.

Acessando Métricas

O endpoint de métricas do Prometheus está disponível em:

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

Este endpoint expõe métricas no formato de texto do Prometheus que um servidor Prometheus pode coletar. O sistema atualiza as métricas em tempo real à medida que processa mensagens.

Convenção de Nomenclatura de Métricas

Todas as métricas seguem o padrão: `sms_c.<categoria>.<nome_da_métrica>.<tipo>`

Categorias:

- `license` - Métricas de status da licença
- `message` - Métricas de processamento de mensagens
- `routing` - Métricas de decisão de roteamento
- `enum` - Métricas de consulta ENUM/NAPTR
- `delivery` - Métricas de entrega de mensagens
- `queue` - Métricas de gerenciamento de fila
- `charging` - Métricas de cobrança
- `mnesia` - Métricas de banco de dados
- `frontend` - Métricas de conexão do frontend
- `location` - Métricas de localização/registro
- `phoenix.endpoint` - Métricas de requisições da API HTTP
- `vm` - Métricas do sistema Erlang VM

Métricas de Licença

sms_c_license_status

Tipo: Gauge

Descrição: Status atual da licença do sistema OmniMessage SMS-C.

Valores:

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

Rótulos: Nenhum

Nome do Produto: `omnimessage`

Caso de Uso: Monitorar a validade da licença para garantir que o sistema esteja operando com uma licença válida. Quando inválida, as mensagens ainda são recebidas, mas roteadas para o destino "NOLICENCE" em vez do roteamento normal.

Comportamento Quando a Licença é Inválida:

- Mensagens de entrada são **aceitas** e armazenadas
- Destino da mensagem (`dest_smsc`) é **automaticamente definido como "NOLICENCE"**
- Roteamento normal é **bypassed**
- UI e monitoramento permanecem **acessíveis**
- Banco de dados e todos os serviços permanecem **operacionais**

Alertas:

```
- alert: SMS_C_License_Invalid
  expr: sms_c_license_status == 0
  for: 1m
  labels:
    severity: critical
  annotations:
    summary: "Licença SMS-C inválida ou expirada"
    description: "O status da licença é inválido - mensagens sendo
roteadas para NOLICENCE"
```

Exemplo de Consultas Prometheus:

```
# Verificar se a licença é válida
sms_c_license_status == 1

# Alertar sobre licença inválida
sms_c_license_status == 0

# Contar mensagens roteadas para NOLICENCE (indica problema de
licença)
sms_c_routing_route_matched_count{dest_smsc="NOLICENCE"}
```

Métricas de Processamento de Mensagens

sms_c_message_received_count

Tipo: Counter

Descrição: Número total de mensagens recebidas pelo SMS-C de todas as fontes.

Rótulos:

- `source_smsc`: Nome do SMSC de origem que enviou a mensagem
- `source_type`: Tipo de conexão de origem (ims, circuit_switched, smpp)
- `message_type`: Tipo de mensagem (sms, mms)

Caso de Uso: Monitorar o volume de mensagens recebidas por fonte e tipo. Usar para detectar padrões de tráfego, identificar períodos de pico e detectar anomalias no fluxo de mensagens.

Alertas: Definir alertas para quedas repentinas (potenciais problemas de conectividade da fonte) ou picos (potenciais ataques/spam).

---

sms_c_message_validated_count

Tipo: Counter

Descrição: Número total de validações de mensagens realizadas.

Rótulos:

- `valid`: Se a validação passou (true ou false)

Caso de Uso: Acompanhar taxas de sucesso/falha de validação. Altas taxas de falha podem indicar mensagens malformadas ou problemas de integração.

Alertas: Alertar quando a taxa de falha de validação exceder o limite (por exemplo, > 5% de falhas).

---

sms_c_message_processing_stop_duration

Tipo: Histogram

Descrição: Tempo necessário para processar uma mensagem desde o recebimento até a conclusão (inclui validação, roteamento e enfileiramento).

Unidade: Milissegundos

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

Rótulos:

- `success`: Se o processamento foi bem-sucedido (true ou false)

Caso de Uso: Monitorar o desempenho do processamento de mensagens de ponta a ponta. Identificar lentidões no pipeline de processamento.

Alertas: Alertar quando a latência p95 ou p99 exceder os limites do SLA.

Métricas de Roteamento

sms_c_routing_route_matched_count

Tipo: Counter

Descrição: Número total de vezes que uma rota específica foi correspondida e selecionada para o roteamento de mensagens.

Rótulos:

- `route_id`: Identificador único da rota correspondida
- `dest_smsc`: SMSC de destino selecionada pela rota

- `priority`: Valor de prioridade da rota correspondida

Caso de Uso: Entender quais rotas estão sendo usadas com mais frequência. Identificar rotas subutilizadas ou sobrecarregadas. Útil para planejamento de capacidade e otimização de rotas.

Alertas: Alertar se rotas de alta prioridade raramente forem correspondidas (pode indicar configuração de roteamento incorreta).

---

sms_c_routing_failed_count

Tipo: Counter

Descrição: Número total de falhas de roteamento em que nenhuma rota adequada pôde ser encontrada.

Rótulos:

- `reason`: Motivo da falha (no_route_found, validation_failed, etc.)

Caso de Uso: Acompanhar falhas de roteamento para identificar lacunas de configuração ou padrões de tráfego inesperados.

Alertas: Alertar sobre quaisquer falhas de roteamento, pois indicam que as mensagens não podem ser entregues.

---

sms_c_routing_action_count

Tipo: Counter

Descrição: Número total de ações especiais de roteamento realizadas.

Rótulos:

- `action`: Tipo de ação (drop, auto_reply, forward)
- `route_id`: Rota que acionou a ação

Caso de Uso: Monitorar regras de queda (anti-spam), uso de auto-resposta e padrões de encaminhamento.

Alertas: Alertar sobre picos inesperados em ações de queda (pode indicar ataque de spam).

---

sms_c_routing_stop_duration

Tipo: Histogram

Descrição: Tempo necessário para avaliar todas as rotas e selecionar a melhor correspondência.

Unidade: Milissegundos

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

Rótulos:

- `dest_smsc`: SMSC de destino selecionada

Caso de Uso: Monitorar o desempenho do mecanismo de roteamento. Roteamento lento indica muitas rotas ou lógica de correspondência complexa.

Alertas: Alertar quando o roteamento levar consistentemente mais tempo do que o esperado (por exemplo, p95 > 50ms).

Métricas de Consulta ENUM/NAPTR

sms_c_enum_cache_hit_count

Tipo: Counter

Descrição: Número total de consultas ENUM atendidas a partir do cache (não exigiu consulta DNS).

Rótulos:

- `domain`: Domínio ENUM consultado

Caso de Uso: Monitorar a eficácia do cache. Altas taxas de acerto de cache reduzem a carga do DNS e melhoram o desempenho.

Alertas: Alertar se a taxa de acerto do cache cair abaixo do limite (pode indicar problemas de cache ou tráfego incomum).

---

sms_c_enum_cache_miss_count

Tipo: Counter

Descrição: Número total de consultas ENUM que exigiram uma consulta DNS (não estão no cache).

Rótulos:

- `domain`: Domínio ENUM consultado

Caso de Uso: Acompanhar falhas de cache para entender a eficácia do cache. Usar com a contagem de acertos para calcular a taxa de acerto.

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

---

sms_c_enum_cache_size_size

Tipo: Gauge

Descrição: Número atual de entradas no cache ENUM.

Caso de Uso: Monitorar o tamanho do cache para garantir que não cresça de forma descontrolada. Ajudar a ajustar as configurações de TTL do cache.

Alertas: Alertar se o tamanho do cache exceder os limites esperados (pode indicar vazamento de memória).

---

sms_c_enum_lookup_stop_duration

Tipo: Histogram

Descrição: Tempo necessário para completar uma consulta ENUM (incluindo consulta DNS se não estiver em cache).

Unidade: Milissegundos

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

Rótulos:

- `domain`: Domínio ENUM consultado
- `success`: Se a consulta foi bem-sucedida (true ou false)
- `cache_hit`: Se o resultado foi atendido a partir do cache (true ou false)

Caso de Uso: Monitorar o desempenho da consulta ENUM. Identificar servidores DNS lentos ou problemas de rede.

Alertas: Alertar quando o tempo de consulta p95 exceder o limite de tempo limite.

---

sms_c_enum_naptr_records_record_count

Tipo: Histogram

Descrição: Número de registros NAPTR retornados por uma consulta ENUM bem-sucedida.

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

Rótulos:

- `domain`: Domínio ENUM consultado

Caso de Uso: Entender a distribuição de registros ENUM. A maioria das consultas deve retornar de 1 a 3 registros.

Alertas: Alertar se frequentemente retornar 0 registros (problema de configuração DNS).

Métricas de Entrega

sms_c_delivery_queued_count

Tipo: Counter

Descrição: Número total de mensagens enfileiradas para entrega a um SMSC de destino.

Rótulos:

- `dest_smsc`: Nome do SMSC de destino

Caso de Uso: Monitorar o fluxo de mensagens para cada destino. Útil para planejamento de capacidade.

Alertas: Comparar com contagens de sucesso/falha de entrega para detectar acúmulo.

---

sms_c_delivery_attempted_count

Tipo: Counter

Descrição: Número total de tentativas de entrega feitas (inclui tentativas de reenvio).

Rótulos:

- `dest_smsc`: Nome do SMSC de destino

Caso de Uso: Acompanhar o volume de tentativas de entrega. Alta contagem de tentativas em relação à contagem enfileirada indica comportamento de reenvio.

---

sms_c_delivery_succeeded_count

Tipo: Counter

Descrição: Número total de mensagens entregues com sucesso ao SMSC de destino.

Rótulos:

- `dest_smsc`: Nome do SMSC de destino

Caso de Uso: Acompanhar entregas bem-sucedidas por destino. Métrica de sucesso primária.

Alertas: Alertar se a taxa de sucesso cair abaixo do limite do SLA.

Cálculo: `success_rate = succeeded / queued`

---

sms_c_delivery_failed_count

Tipo: Counter

Descrição: Número total de mensagens que falharam na entrega após todas as tentativas de reenvio.

Rótulos:

- `dest_smsc`: Nome do SMSC de destino
- `reason`: Motivo da falha

Caso de Uso: Acompanhar falhas de entrega para identificar destinos problemáticos ou padrões de falha.

Alertas: Alertar sobre taxas de falha elevadas ou motivos de falha específicos.

---

sms_c_delivery_dead_letter_count

Tipo: Counter

Descrição: Número total de mensagens movidas para a fila de cartas mortas (indisponíveis).

Rótulos:

- `reason`: Motivo da carta morta (por exemplo, `max_retries_exceeded`, `expired`)

Caso de Uso: Monitorar mensagens não entregáveis que requerem intervenção manual.

Alertas: Alertar sobre quaisquer eventos de carta morta, pois representam falha completa de entrega.

---

sms_c_delivery_succeeded_attempt_count

Tipo: Histogram

Descrição: Número de tentativas de entrega necessárias antes da entrega bem-sucedida.

Buckets: 1, 2, 3, 5, 10

Rótulos:

- `dest_smsc`: Nome do SMSC de destino

Caso de Uso: Entender o comportamento de reenvio. A maioria das entregas deve ser bem-sucedida na primeira tentativa.

Alertas: Alertar se a contagem média de tentativas exceder 2 (indica problemas de confiabilidade do destino).

---

sms_c_delivery_failed_attempt_count

Tipo: Histogram

Descrição: Número de tentativas de entrega feitas antes da falha final.

Buckets: 1, 2, 3, 5, 10

Rótulos:

- `dest_smsc`: Nome do SMSC de destino

Caso de Uso: Entender quantas tentativas de reenvio ocorrem antes de desistir.

---

sms_c_delivery_time_delta_delta_ms

Tipo: Histogram

Descrição: Tempo desde a submissão da mensagem até a confirmação de entrega. Esta métrica captura a latência completa de ponta a ponta desde que uma mensagem entra no SMS-C até que o sistema confirma a entrega. Ela fornece rótulos detalhados para análise por origem, destino e comportamento de reenvio.

Unidade: Milissegundos

Buckets: 50, 100, 250, 500, 1000, 2000, 5000, 10000, 20000, 60000, 600000 ms (50ms a 10 minutos)

Rótulos:

- `source_smsc`: SMSC de origem que submeteu a mensagem
- `dest_smsc`: SMSC de destino que entregou a mensagem
- `delivery_attempts`: Número de tentativas de entrega necessárias (0 = sucesso na primeira tentativa)

Caso de Uso: Analisar o desempenho de entrega de ponta a ponta por par origem-destino. Identificar caminhos lentos, combinações problemáticas de

SMSC e correlacionar latência com comportamento de reenvio. Essencial para monitoramento de SLA e planejamento de capacidade.

Alertas:

- Alertar quando p95 exceder o limite do SLA para combinações específicas de origem/destino
- Alertar quando mensagens consistentemente exigirem várias tentativas

Exemplo de Consultas Prometheus:

```
# Tempo de entrega p95 por SMSC de origem e destino
histogram_quantile(0.95,
  sum by (source_smsc, dest_smsc, le) (
    rate(sms_c_delivery_time_delta_delta_ms_bucket[5m])
  )
)

# Tempo de entrega p99 geral
histogram_quantile(0.99,
  sum by (le) (
    rate(sms_c_delivery_time_delta_delta_ms_bucket[5m])
  )
)

# Tempo médio de entrega por SMSC de origem
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]))

# Tempo de entrega para mensagens que exigem reenvios vs sucesso na
primeira tentativa
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 origem-destino mais 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])
    )
  )
)

# Porcentagem de mensagens entregues dentro de 2 segundos
```

```
sum(rate(sms_c_delivery_time_delta_delta_ms_bucket{le="2000"}[5m])) /
sum(rate(sms_c_delivery_time_delta_delta_ms_count[5m])) * 100

# Mensagens entregues dentro de 2 segundos por destino
sum by (dest_smsc)
(rate(sms_c_delivery_time_delta_delta_ms_bucket{le="2000"}[5m])) /
sum by (dest_smsc) (rate(sms_c_delivery_time_delta_delta_ms_count[5m]
* 100
```

---

sms_c_delivery_time_by_smsc_delta_ms

Tipo: Histogram

Descrição: Tempo desde a submissão da mensagem até a confirmação de entrega, rotulado apenas por SMSC de origem e destino. Esta é uma variante de menor cardinalidade de `sms_c_delivery_time_delta_delta_ms` - omite o rótulo `delivery_attempts` para que o tempo médio de entrega por par origem/destino possa ser lido diretamente sem agregar por contagens de tentativas. Emitido do mesmo evento `[:sms_c, :delivery, :time_delta]`.

Unidade: Milissegundos

Buckets: 50, 100, 250, 500, 1000, 2000, 5000, 10000, 20000, 60000, 600000 ms (50ms a 10 minutos)

Rótulos:

- `source_smsc`: SMSC de origem que submeteu a mensagem
- `dest_smsc`: SMSC de destino que entregou a mensagem

Caso de Uso: Dashboard do tempo médio de entrega de ponta a ponta por par origem/destino. Preferir esta métrica em vez de `sms_c_delivery_time_delta_delta_ms` para painéis de média/SLA onde a divisão por contagem de tentativas não é necessária; usar a variante `_delta` quando precisar segmentar pelo comportamento de reenvio.

Exemplo de Consultas Prometheus:

```
# Tempo médio de entrega (ms) por SMSC de origem/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]))

# Tempo médio de entrega geral
sum(rate(sms_c_delivery_time_by_smsc_delta_ms_sum[5m])) /
sum(rate(sms_c_delivery_time_by_smsc_delta_ms_count[5m]))

# Tempo de entrega p95 por SMSC de origem/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 Fila

sms_c_queue_size_size

Tipo: Gauge

Descrição: Número total atual de mensagens na fila (todos os estados combinados).

Rótulos:

- `queue_type`: Tipo de fila (message_queue, dead_letter)

Caso de Uso: Monitorar a profundidade da fila para detectar acúmulos ou problemas de processamento.

Alertas: Alertar quando o tamanho da fila exceder os limites de capacidade.

---

sms_c_queue_size_pending

Tipo: Gauge

Descrição: Número atual de mensagens pendentes de entrega (ainda não tentadas).

Rótulos:

- `queue_type`: Tipo de fila

Caso de Uso: Monitorar a contagem de mensagens pendentes. Altas contagens pendentes indicam atrasos no processamento.

Alertas: Alertar quando a contagem pendente exceder o limite por um período prolongado.

---

sms_c_queue_size_failed

Tipo: Gauge

Descrição: Número atual de mensagens em estado de falha (aguardando reenvio).

Rótulos:

- `queue_type`: Tipo de fila

Caso de Uso: Monitorar a acumulação de mensagens falhadas. Indica problemas de entrega.

Alertas: Alertar sobre contagens elevadas de falhas, pois isso impacta as taxas de entrega.

---

sms_c_queue_size_delivered

Tipo: Gauge

Descrição: Número atual de mensagens entregues aguardando limpeza/remoção da fila.

Rótulos:

- `queue_type`: Tipo de fila

Caso de Uso: Monitorar o atraso na limpeza. Altas contagens indicam que o processo de limpeza está atrasado.

Alertas: Alertar se as mensagens entregues se acumularem significativamente.

---

sms_c_queue_oldest_message_age_seconds

Tipo: Gauge

Descrição: Idade (em segundos) da mensagem mais antiga atualmente em estado pendente.

Rótulos:

- `queue_type`: Tipo de fila

Caso de Uso: Detectar envelhecimento de mensagens e paralisações no processamento. Crítico para monitoramento de SLA.

Alertas: Alertar quando a idade da mensagem mais antiga exceder o limite do SLA (por exemplo, > 300 segundos).

Métricas de Cobrança

A cobrança é realizada através da interface **Diameter Ro** (Controle de Crédito). O rótulo `account` carrega o MSISDN do assinante cobrado (o remetente da mensagem). Um evento `charging_failed` é emitido tanto para uma recusa definitiva fora de balanço (`reason: :no_credit`) quanto para um erro de OCS (transporte/protocolo). Apenas uma recusa `:no_credit` rejeita

uma mensagem (status `:balance_rejected`); erros de OCS falham abertos e a mensagem é entregue.

sms_c_charging_requested_count

Tipo: Counter

Descrição: Número total de requisições de cobrança (Diameter Ro CCR Event-Requests) enviadas ao OCS.

Rótulos:

- `account`: MSISDN do assinante cobrado (remetente da mensagem)

Caso de Uso: Acompanhar o volume de cobrança por conta. Útil para reconciliação de faturamento.

---

sms_c_charging_succeeded_count

Tipo: Counter

Descrição: Número total de operações de cobrança bem-sucedidas.

Rótulos:

- `account`: Identificador da conta cobrada

Caso de Uso: Monitorar a taxa de sucesso de cobrança por conta.

Cálculo: `success_rate = succeeded / requested`

---

sms_c_charging_failed_count

Tipo: Counter

Descrição: Número total de operações de cobrança que falharam.

Rótulos:

- `account`: Identificador da conta
- `reason`: Motivo da falha

Caso de Uso: Identificar falhas de cobrança que podem impactar a receita ou exigir intervenção na conta.

Alertas: Alertar sobre taxas de falha de cobrança elevadas.

---

sms_c_charging_succeeded_duration

Tipo: Histogram

Descrição: Tempo necessário para completar uma solicitação de cobrança bem-sucedida.

Unidade: Milissegundos

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

Rótulos:

- `account`: Identificador da conta

Caso de Uso: Monitorar o desempenho do sistema de faturamento. Cobranças lentas podem atrasar a entrega de mensagens.

Alertas: Alertar quando o tempo de cobrança p95 exceder o limite.

Métricas de Saúde do Sistema

sms_c_mnesia_table_size_record_count

Tipo: Gauge

Descrição: Número atual de registros em cada tabela do banco de dados Mnesia. Consultado a cada 10 segundos.

Rótulos:

- `table`: Nome da tabela

Tabelas Monitoradas:

- `sms_route` - Regras de roteamento SMS
- `message_store` - Fila de mensagens (pendentes, entregues, mensagens falhadas)
- `location_store` - Dados de localização/registro de assinantes
- `frontend_store` - Registros de frontend/SMSC
- `translation_rule` - Número de regras de tradução
- `cell_tower_store` - Dados de localização de torres de celular
- `message_events` - Logs de eventos de mensagens

Caso de Uso: Monitorar o crescimento do banco de dados. Detectar acúmulo de dados inesperado. Planejamento de capacidade.

Alertas: Alertar sobre taxas de crescimento inesperadas da tabela ou quando se aproximar dos limites de capacidade.

Exemplo de Consultas Prometheus:

```
# Todos os tamanhos de tabela
sms_c_mnesia_table_size_record_count

# Tamanho da fila de mensagens
sms_c_mnesia_table_size_record_count{table="message_store"}

# Registros de assinantes ativos
sms_c_mnesia_table_size_record_count{table="location_store"}

# Contagem de regras de roteamento
sms_c_mnesia_table_size_record_count{table="sms_route"}
```

---

sms_c_frontend_status_count

Tipo: Gauge

Descrição: Número de frontends em cada status de conexão.

Rótulos:

- `frontend_name`: Identificador do frontend
- `status`: Status da conexão (connected, disconnected)

Caso de Uso: Monitorar a conectividade do frontend. Detectar falhas de conexão.

Alertas: Alertar quando frontends esperados se desconectarem.

---

sms_c_location_registered_count

Tipo: Counter

Descrição: Número total de registros de localização/assinante recebidos pelo sistema.

Rótulos:

- `location`: Nome do frontend/SMSC onde o assinante está registrado
- `ims_capable`: Se o assinante suporta IMS (true/false)

Caso de Uso: Monitorar a atividade de registro de assinantes. Acompanhar assinantes IMS vs não-IMS. Detectar tempestades ou falhas de registro.

Alertas: Definir alertas para:

- Quedas na taxa de registro (pode indicar problemas de rede)
- Picos incomuns em registros
- Alta proporção de registros não-IMS (fluxo de dispositivos legados)

Exemplo de Consulta:

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

# Proporção de registro IMS vs não-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

Descrição: Número atual de registros de assinantes ativos, agrupados por localização (frontend/SMSC) e capacidade IMS. Esta métrica é consultada a cada 10 segundos e reflete o estado atual do armazenamento de localização.

Rótulos:

- `location`: Nome do frontend/SMSC onde os assinantes estão registrados
- `ims_capable`: Se os registros são capazes de IMS (true/false)

Caso de Uso: Monitorar a distribuição atual de assinantes entre frontends. Acompanhar a adoção do IMS. Identificar desequilíbrios de carga entre frontends. Planejamento de capacidade para cada instância de SMSC.

Alertas: Definir alertas para:

- Quedas repentinas em registros para uma localização (pode indicar problemas no frontend)
- Distribuição desequilibrada entre frontends
- Total de registros se aproximando dos limites de capacidade

Exemplo de Consultas Prometheus:

```
# Total de registros ativos
sum(sms_c_location_active_registrations_count)

# Registros ativos por localização
sum by (location) (sms_c_location_active_registrations_count)

# Registros ativos capazes de IMS por localização
sum by (location)
(sms_c_location_active_registrations_count{ims_capable="true"})

# Taxa de adoção de IMS por localização
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 não-IMS
sum by (ims_capable) (sms_c_location_active_registrations_count)

# Localizações com mais registros
topk(10, sum by (location)
(sms_c_location_active_registrations_count))

# Distribuição de registros como porcentagem do total
sum by (location) (sms_c_location_active_registrations_count) /
sum(sms_c_location_active_registrations_count) * 100
```

Métricas de Requisições da API HTTP

phoenix_endpoint_stop_duration

Tipo: Distribuição (Histogram)

Descrição: Duração do processamento de requisições HTTP em milissegundos, desde o início da requisição até a conclusão da resposta.

Rótulos:

- `method`: Método HTTP (por exemplo, `GET`, `POST`)
- `path`: caminho da requisição com segmentos de alta cardinalidade (UUIDs, IDs numéricos) colapsados para `/:id` (por exemplo, `/api/messages`, `/api/messages/:id`)

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

Caso de Uso: Monitorar o desempenho da API. Identificar endpoints lentos. Acompanhar SLAs de tempo de resposta.

Alertas: Definir alertas para:

- Latência P95 > 500ms para endpoints críticos
- Latência P99 > 1s para qualquer endpoint
- Tendências de latência crescentes

Exemplo de Consulta:

```
# Tempo de resposta p95 por endpoint
histogram_quantile(0.95,
  rate(phoenix_endpoint_stop_duration_bucket[5m]))

# Requisições mais lentas que 1 segundo
sum(rate(phoenix_endpoint_stop_duration_bucket{le="1000"}[5m]))
```

---

phoenix_endpoint_stop_count

Tipo: Counter

Descrição: Número total de requisições HTTP concluídas, categorizadas por método, caminho e código de status HTTP.

Rótulos:

- `method`: Método HTTP (por exemplo, `GET`, `POST`)
- `path`: caminho da requisição (segmentos de alta cardinalidade colapsados para `/:id`)

- `status`: código de status HTTP (200, 201, 400, 404, 500, etc.)

Caso de Uso: Monitorar o volume de requisições da API e taxas de sucesso. Acompanhar taxas de erro por endpoint.

Alertas: Definir alertas para:

- Taxa de erro > 5% para qualquer endpoint
- Erros 5xx em endpoints críticos
- Quedas repentinas no volume de requisições

Exemplo de Consulta:

```
# Taxa de requisições por caminho
sum by (path) (rate(phoenix_endpoint_stop_count[5m]))

# Taxa de erro por caminho
sum by (path) (rate(phoenix_endpoint_stop_count{status=~"5.."}
[5m])) /
sum by (path) (rate(phoenix_endpoint_stop_count[5m]))

# Taxa de sucesso
sum(rate(phoenix_endpoint_stop_count{status=~"2.."}[5m])) /
sum(rate(phoenix_endpoint_stop_count[5m]))
```

---

phoenix_router_dispatch_exception_count

Tipo: Counter

Descrição: Número total de exceções/erros levantados durante o processamento de requisições HTTP.

Rótulos:

- `route`: Rota do endpoint da API onde a exceção ocorreu
- `kind`: Tipo de exceção (error, exit, throw)

Caso de Uso: Acompanhar erros de aplicação. Identificar endpoints problemáticos. Monitorar a estabilidade do sistema.

Alertas: Definir alertas para qualquer valor diferente de zero em endpoints críticos.

Exemplo de Consulta:

```
# Taxa de exceções por endpoint
rate(phoenix_router_dispatch_exception_count[5m])

# Total de exceções na última hora
increase(phoenix_router_dispatch_exception_count[1h])
```

Métricas Diameter Sh

O SMS-C consulta o HSS pela interface **Diameter Sh** (3GPP TS 29.329) para classificar assinantes como on-net ou off-net durante a avaliação de rotas. Essas métricas são emitidas apenas quando o Diameter está habilitado.

sms_c_diameter_peer_status

Tipo: Gauge

Descrição: Se o par Diameter está conectado (`1`) ou não (`0`).

Rótulos:

- `origin_host`: Origin-Host Diameter configurado do par
- `ip`: endereço IP do par

Exemplo de Consulta:

```
# Pares atualmente fora do ar
sms_c_diameter_peer_status == 0
```

---

sms_c_diameter_sh_request_count

Tipo: Counter

Descrição: Número total de requisições de consulta Diameter Sh (UDR) enviadas.

---

sms_c_diameter_sh_response_count

Tipo: Counter

Descrição: Número total de respostas de consulta Diameter Sh (UDA), por resultado.

Rótulos:

- `result`: resultado da consulta (por exemplo, `on_net`, `unknown_user`, `error`)
- `result_code`: Resultado Experimental do Diameter / Código de Resultado (por exemplo, `2001`, `5001`)

Exemplo de Consulta:

```
# Taxa de erro de consulta 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: Distribuição (Histogram)

Descrição: Duração da consulta Diameter Sh em milissegundos.

Rótulos:

- `result`: resultado da consulta

Buckets: 5, 10, 25, 50, 100, 250, 500, 1000, 2500 ms

Exemplo de Consulta:

```
# Latência de consulta Sh p95
histogram_quantile(0.95,
rate(sms_c_diameter_sh_response_duration_ms_bucket[5m]))
```

---

sms_c_diameter_sh_error_count

Tipo: Counter

Descrição: Número total de erros de consulta Diameter Sh (falhas de transporte/timeout/decode, distintas de um UDA negativo).

Rótulos:

- `reason`: motivo do erro

Métricas de Federação

A federação encaminha mensagens entre controladores SMS-C por meio de um modelo de par baseado em pull. Veja Federação.

sms_c_federation_forward_attempted_count

Tipo: Counter

Descrição: Número total de mensagens encaminhadas para controladores remotos.

Rótulos:

- `dest_controller`: identidade do controlador de destino

---

sms_c_federation_forward_completed_count

Tipo: Counter

Descrição: Número total de mensagens entregues com sucesso a controladores remotos.

Rótulos:

- `dest_controller`: identidade do controlador de destino

---

sms_c_federation_forward_queued_count

Tipo: Counter

Descrição: Número total de mensagens enfileiradas porque um par estava inacessível.

Rótulos:

- `dest_controller`: identidade do controlador de destino

---

sms_c_federation_forward_failed_count

Tipo: Counter

Descrição: Número total de mensagens que excederam o número máximo de tentativas.

Rótulos:

- `dest_controller`: identidade do controlador de destino

---

sms_c_federation_message_received_count

Tipo: Counter

Descrição: Número total de mensagens recebidas de controladores remotos.

Rótulos:

- `origin_node`: nó do controlador de origem

---

sms_c_federation_health_check_count

Tipo: Counter

Descrição: Número total de tentativas de verificação de saúde de pares, por par e resultado.

Rótulos:

- `peer`: identidade do par
- `result`: resultado da verificação de saúde (por exemplo, `healthy`, `unhealthy`)

---

sms_c_federation_peer_status_count

Tipo: Gauge

Descrição: Contagem atual de pares de federação por status.

Rótulos:

- `status`: status do par (por exemplo, `healthy`, `degraded`, `unhealthy`)

---

sms_c_federation_queue_depth_count

Tipo: Gauge

Descrição: Profundidade atual da fila de encaminhamento da federação.

Exemplo de Consulta:

```
# Alertar se a fila de encaminhamento estiver acumulando
sms_c_federation_queue_depth_count > 1000
```

Métricas do Erlang VM

vm_memory_total

Tipo: Gauge

Descrição: Memória total alocada pela Erlang VM em bytes.

Caso de Uso: Monitorar o uso geral de memória. Detectar vazamentos de memória. Planejar capacidade.

Alertas: Alertar quando o uso de memória > 80% da memória disponível do sistema.

---

vm_memory_processes

Tipo: Gauge

Descrição: Memória usada pelos processos Erlang em bytes.

Caso de Uso: Acompanhar o consumo de memória dos processos. Fonte mais comum de crescimento de memória.

Alertas: Alertar sobre taxa de crescimento alta sustentada.

---

vm_total_run_queue_lengths_total

Tipo: Gauge

Descrição: Número total de processos aguardando para serem agendados em todos os escalonadores de CPU.

Caso de Uso: Medir a carga do sistema. Valores altos indicam saturação da CPU.

Alertas: Alertar quando consistentemente > 10 * número de núcleos de CPU.

---

vm_system_counts_process_count

Tipo: Gauge

Descrição: Número atual de processos em execução na VM.

Caso de Uso: Monitorar padrões de criação de processos. Detectar vazamentos de processos.

Alertas: Alertar quando se aproximar do limite de processos (padrão 262.144).

Coleta e Polling de Métricas

O sistema coleta automaticamente as seguintes métricas a cada 10 segundos:

- Tamanhos e idades das filas
- Tamanhos das tabelas Mnesia
- Estatísticas do cache ENUM

Todas as outras métricas são acionadas por eventos e emitidas quando a ação correspondente ocorre.

Padrões Comuns de Monitoramento

Taxa de Sucesso de Entrega por Destino

Acompanhe a taxa de sucesso da entrega de mensagens para cada SMSC de destino:

Fórmula: `(sms_c_delivery_succeeded_count) / (sms_c_delivery_queued_count)`

Interpretação: Deve ser > 95% para destinos saudáveis. Taxas mais baixas indicam problemas de entrega.

Latência de Mensagem de Ponta a Ponta

Monitore o tempo total desde o recebimento da mensagem até a entrega:

Métricas:

- `sms_c_message_processing_stop_duration` (processamento)
- `sms_c_delivery_time_delta_delta_ms` (entrega)

Interpretação: A soma representa a latência total voltada para o usuário.

Eficácia do Cache ENUM

Meça quão bem o cache ENUM está funcionando:

Fórmula: `(sms_c_enum_cache_hit_count) / (sms_c_enum_cache_hit_count + sms_c_enum_cache_miss_count)`

Interpretação: Deve ser > 80% após o aquecimento. Taxas mais baixas podem indicar TTL curto ou alta variação de tráfego.

Utilização de Rotas

Identifique quais rotas lidam com mais tráfego:

Métrica: `sms_c_routing_route_matched_count` agrupada por `route_id`

Interpretação: Use para identificar rotas quentes para otimização e planejamento de capacidade.

Tendência de Acúmulo de Fila

Monitore se a fila de mensagens está crescendo (acúmulo) ou diminuindo (recuperando):

Métricas:

- `sms_c_queue_size_pending` (atual pendente)
- `sms_c_queue_oldest_message_age_seconds` (idade em tendência)

Interpretação: Contagem pendente crescente + idade crescente = acúmulo se formando.

Taxa de Reenvio

Entenda com que frequência os reenvios de entrega são necessários:

Métrica: percentis do histograma `sms_c_delivery_succeeded_attempt_count`

Interpretação: Se p95 > 1, a maioria das mensagens requer reenvios. Indica problemas de confiabilidade do destino.

Alertas Recomendados

Alerta	Condição	Sever
Alta Taxa de Falha de Roteamento	`routing_failed_count` aumento	Crítico
Acúmulo de Fila	`queue_size_pending` > limite	Aviso
Mensagens Antigas na Fila	`queue_oldest_message_age_seconds` > 300	Crítico
Pico de Falhas de Entrega	`delivery_failed_count` pico	Alto
Eventos de Carta Morta	`delivery_dead_letter_count` > 0	Alto
Timeouts de Consulta ENUM	`enum_lookup_stop_duration` p95 > 5000ms	Aviso
Baixa Taxa de Acerto de Cache	Taxa de acerto do cache ENUM < 0.7	Aviso
Frontend Desconectado	`frontend_status_count{status="disconnected"}` > 0	Alto
Falhas de Cobrança	`charging_failed_count` > limite	Alto

Alerta	Condição	Sever
Processamento de Mensagens Lento	`message_processing_stop_duration` p95 > 1000ms	Aviso

Recomendações de Dashboard

Dashboard de Operações

Propósito: Monitoramento em tempo real da saúde do sistema

Painéis:

1. Através de mensagens (recebidas/processadas/entregues por minuto)
2. Tamanhos de fila (pendente, falhada, entregue)
3. Taxa de sucesso de entrega por destino
4. Latência de processamento e entrega p95
5. Status de frontends ativos
6. Alertas atuais

Dashboard de Desempenho

Propósito: Análise de desempenho do sistema

Painéis:

1. Histograma da duração do processamento de mensagens
2. Histograma da duração do roteamento
3. Histograma da duração da consulta ENUM
4. Histograma da duração da cobrança
5. Distribuição de tentativas de entrega
6. Taxas de acerto do cache

Dashboard de Negócios

Propósito: Análise de tráfego e uso

Painéis:

1. Mensagens por SMSC de origem
2. Mensagens por SMSC de destino
3. Mapa de calor de utilização de rotas
4. Contagens de ações de auto-resposta e queda
5. Estatísticas de uso de ENUM
6. Volume de cobrança por conta

Retenção de Métricas

Configurações de retenção recomendadas para Prometheus:

- **Métricas brutas**: 15 dias
- **Agregados de 5 minutos**: 90 dias
- **Agregados de 1 hora**: 2 anos

Isso fornece um histórico recente detalhado enquanto mantém tendências de longo prazo para planejamento de capacidade.

Resolução de Problemas com Métricas

Cenário: Mensagens Não Estão Sendo Entregues

Passos de Investigação:

1. Verifique `sms_c_message_received_count` - As mensagens estão sendo recebidas?
2. Verifique `sms_c_routing_failed_count` - Elas estão sendo roteadas?
3. Verifique `sms_c_delivery_queued_count` - Elas estão sendo enfileiradas?
4. Verifique `sms_c_delivery_failed_count` - As tentativas de entrega estão falhando?
5. Verifique os rótulos `dest_smsc` para identificar o destino problemático

---

Cenário: Processamento de Mensagens Lento

Passos de Investigação:

1. Verifique o histograma `sms_c_message_processing_stop_duration` - Tempo geral de processamento
2. Verifique `sms_c_routing_stop_duration` - O roteamento está lento?
3. Verifique `sms_c_enum_lookup_stop_duration` - As consultas ENUM estão lentas?
4. Verifique `sms_c_charging_succeeded_duration` - A cobrança está lenta?
5. Identifique o gargalo e investigue o componente específico

---

Cenário: Crescimento da Fila de Mensagens

Passos de Investigação:

1. Verifique a tendência de `sms_c_queue_size_pending` - Está crescendo?
2. Verifique `sms_c_delivery_attempted_count` - Estão ocorrendo tentativas de entrega?
3. Verifique `sms_c_delivery_failed_count` - Estão falhando?
4. Verifique `sms_c_delivery_time_delta_delta_ms` - A entrega está demorando muito?
5. Verifique os rótulos `dest_smsc` para identificar destinos lentos

---

Exemplos de Consultas Prometheus

Através de Mensagens

Mensagens Recebidas Por Segundo (média de 5 minutos):

```
rate(sms_c_message_received_count[5m])
```

Mensagens Recebidas Por Minuto (média de 1 hora):

```
rate(sms_c_message_received_count[1h]) * 60
```

Total de Mensagens Hoje:

```
increase(sms_c_message_received_count[24h])
```

Mensagens por Tipo de Fonte:

```
sum by (source_type) (rate(sms_c_message_received_count[5m]))
```

Mensagens por SMSC de Origem:

```
sum by (source_smsc) (rate(sms_c_message_received_count[5m]))
```

Desempenho de Entrega

Taxa de Sucesso de Entrega (Porcentagem):

```
(rate(sms_c_delivery_succeeded_count[5m]) /
rate(sms_c_delivery_queued_count[5m])) * 100
```

Taxa de Falha de Entrega (Porcentagem):

```
(rate(sms_c_delivery_failed_count[5m]) /
rate(sms_c_delivery_queued_count[5m])) * 100
```

Tentativas Médias de Entrega (p95):

```
histogram_quantile(0.95,
sms_c_delivery_succeeded_attempt_count_bucket)
```

Sucesso de Entrega por Destino:

```
sum by (dest_smsc) (rate(sms_c_delivery_succeeded_count[5m]))
```

Motivos de Falha de Entrega:

```
sum by (reason) (rate(sms_c_delivery_failed_count[5m]))
```

Tempo para Entrega (p95):

```
histogram_quantile(0.95,
sms_c_delivery_time_delta_delta_ms_bucket)
```

Tempo para Entrega (p99):

```
histogram_quantile(0.99,
sms_c_delivery_time_delta_delta_ms_bucket)
```

Métricas de Fila

Mensagens Pendentes Atuais:

```
sms_c_queue_size_pending
```

Mensagens Falhadas Aguardando Reenvio:

```
sms_c_queue_size_failed
```

Idade da Mensagem Mais Antiga (Minutos):

```
sms_c_queue_oldest_message_age_seconds / 60
```

Taxa de Crescimento da Fila (Mensagens/Hora):

```
rate(sms_c_queue_size_size[1h]) * 3600
```

Mensagens Entrando na Fila:

```
rate(sms_c_delivery_queued_count[5m])
```

Mensagens Saindo da Fila:

```
rate(sms_c_delivery_succeeded_count[5m]) +
rate(sms_c_delivery_failed_count[5m])
```

Acúmulo na Fila (Entrando - Saindo):

```
rate(sms_c_delivery_queued_count[5m]) -
(rate(sms_c_delivery_succeeded_count[5m]) +
rate(sms_c_delivery_failed_count[5m]))
```

Desempenho de Roteamento

Taxa de Sucesso de Roteamento:

```
(1 - (rate(sms_c_routing_failed_count[5m]) /
(rate(sms_c_routing_route_matched_count[5m]) +
rate(sms_c_routing_failed_count[5m])))) * 100
```

Rotas Mais Usadas:

```
topk(10, sum by (route_id, dest_smsc)
(rate(sms_c_routing_route_matched_count[1h])))
```

Falhas de Roteamento por Minuto:

```
rate(sms_c_routing_failed_count[5m]) * 60
```

Ações de Queda por Hora:

```
increase(sms_c_routing_action_count{action="drop"}[1h])
```

Ações de Auto-Resposta por Hora:

```
increase(sms_c_routing_action_count{action="auto_reply"}[1h])
```

Desempenho ENUM

Taxa de Acerto do Cache 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]))
```

Porcentagem de Acerto do Cache 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
```

Latência de Consulta ENUM (p95):

```
histogram_quantile(0.95, sms_c_enum_lookup_stop_duration_bucket)
```

Consultas ENUM por Segundo (Cacheadas vs Não Cacheadas):

```
# Cacheadas (rápido)
rate(sms_c_enum_cache_hit_count[5m])

# Não Cacheadas (exige consulta DNS)
rate(sms_c_enum_cache_miss_count[5m])
```

Média de Registros NAPTR Retornados:

```
rate(sms_c_enum_naptr_records_record_count_sum[5m]) /
rate(sms_c_enum_naptr_records_record_count_count[5m])
```

Tamanho do Cache ENUM:

```
sms_c_enum_cache_size_size
```

Desempenho de Processamento

Latência de Processamento de Mensagens (p95):

```
histogram_quantile(0.95,
sms_c_message_processing_stop_duration_bucket)
```

Taxa de Falhas de Processamento:

```
rate(sms_c_message_processing_stop_duration_count{success="false"}
[5m])
```

Taxa de Falha de Validação:

```
rate(sms_c_message_validated_count{valid="false"}[5m]) /
rate(sms_c_message_validated_count[5m])
```

Métricas de Cobrança

Taxa de Sucesso de Cobrança:

```
rate(sms_c_charging_succeeded_count[5m]) /
rate(sms_c_charging_requested_count[5m])
```

Falhas de Cobrança por Minuto:

```
rate(sms_c_charging_failed_count[5m]) * 60
```

Latência de Cobrança (p95):

```
histogram_quantile(0.95, sms_c_charging_succeeded_duration_bucket)
```

Volume de Cobrança por Conta:

```
sum by (account) (rate(sms_c_charging_requested_count[1h]))
```

Saúde do Frontend

Frontends Ativos:

```
sum(sms_c_frontend_status_count{status="connected"})
```

Frontends Desconectados:

```
sum(sms_c_frontend_status_count{status="disconnected"})
```

Frontends por Nome:

```
sum by (frontend_name)
(sms_c_frontend_status_count{status="connected"})
```

Saúde do Sistema

Tamanhos das Tabelas Mnesia:

```
sms_c_mnesia_table_size_record_count
```

Contagem de Rotas:

```
sms_c_mnesia_table_size_record_count{table="sms_route"}
```

Contagem de Regras de Tradução:

```
sms_c_mnesia_table_size_record_count{table="translation_rule"}
```

Exemplos de Dashboard do Grafana

Dashboard 1: Operações em Tempo Real

Propósito: Monitorar a atividade e saúde atual do sistema.

Painéis:

1. **Através de Mensagens (Gráfico)**
    - Consulta: `rate(sms_c_message_received_count[5m])`
    - Consulta: `rate(sms_c_delivery_succeeded_count[5m])`
    - Unidade: mensagens/segundo
    - Legenda: {{source_type}}
2. **Taxa de Sucesso de Entrega (Gauge)**
    - Consulta: `(rate(sms_c_delivery_succeeded_count[5m]) / rate(sms_c_delivery_queued_count[5m])) * 100`
    - Unidade: porcentagem (0-100)
    - Limites:
        - Vermelho: < 90
        - Amarelo: 90-95
        - Verde: > 95
3. **Profundidade da Fila (Gráfico)**
    - Consulta: `sms_c_queue_size_pending`
    - Consulta: `sms_c_queue_size_failed`
    - Unidade: mensagens
    - Legenda: {{queue_type}}
4. **Idade da Mensagem Mais Antiga (Stat)**
    - Consulta: `sms_c_queue_oldest_message_age_seconds / 60`
    - Unidade: minutos
    - Limites:
        - Verde: < 5
        - Amarelo: 5-10
        - Vermelho: > 10
5. **Frontends Ativos (Stat)**
    - Consulta: `sum(sms_c_frontend_status_count{status="connected"})`
    - Unidade: contagem
    - Cor: Azul

6. **Falhas de Roteamento (Gráfico)**
    - Consulta: `rate(sms_c_routing_failed_count[5m]) * 60`
    - Unidade: falhas/minuto
    - Limite de alerta: > 0

Dashboard 2: Análise de Desempenho

Propósito: Analisar o desempenho do sistema e identificar gargalos.

Painéis:

1. **Latência de Ponta a Ponta (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)
    - Unidade: milissegundos
    - Legenda: Percentil
2. **Latências de Componentes (Bar Gauge)**
    - Roteamento: `histogram_quantile(0.95, sms_c_routing_stop_duration_bucket)`
    - ENUM: `histogram_quantile(0.95, sms_c_enum_lookup_stop_duration_bucket)`
    - Cobrança: `histogram_quantile(0.95, sms_c_charging_succeeded_duration_bucket)`
    - Entrega: `histogram_quantile(0.95, sms_c_delivery_time_delta_delta_ms_bucket)`
    - Unidade: milissegundos
    - Barras horizontais
3. **Distribuição de Tentativas de Entrega (Heatmap)**

- Consulta: `sms_c_delivery_succeeded_attempt_count_bucket`
- Mostra quantas tentativas são normalmente necessárias
- Escala de cores: Azul (1 tentativa) a Vermelho (muitas tentativas)

4. **Desempenho do Cache ENUM (Gráfico)**

   - Taxa de Acerto: `rate(sms_c_enum_cache_hit_count[5m]) / (rate(sms_c_enum_cache_hit_count[5m]) + rate(sms_c_enum_cache_miss_count[5m]))`
   - Tamanho do Cache: `sms_c_enum_cache_size_size`
   - Eixo Y duplo (taxa vs tamanho)

5. **Taxa de Sucesso de Processamento (Gauge)**

   - Consulta: `(rate(sms_c_message_processing_stop_duration_count{success="true"}[5m]) / rate(sms_c_message_processing_stop_duration_count[5m])) * 100`
   - Unidade: porcentagem
   - Limites:
     - Vermelho: < 95
     - Amarelo: 95-99
     - Verde: > 99

Dashboard 3: Análise de Tráfego

Propósito: Analisar padrões de tráfego de mensagens e distribuição de roteamento.

Painéis:

1. **Mensagens por Tipo de Fonte (Gráfico de Pizza)**

   - Consulta: `sum by (source_type) (increase(sms_c_message_received_count[1h]))`
   - Mostra a distribuição: IMS vs CS vs SMPP

2. **Mensagens por SMSC de Origem (Gráfico de Barras)**

- Consulta: `sum by (source_smsc) (rate(sms_c_message_received_count[1h]))`
- Top 10 fontes
- Barras horizontais

3. **Utilização de Rotas (Tabela)**

- Colunas:
  - ID da Rota
  - SMSC de Destino
  - Mensagens (1h): `sum by (route_id, dest_smsc) (increase(sms_c_routing_route_matched_count[1h]))`
  - Prioridade
  - Taxa de Sucesso
- Ordenado por contagem de mensagens

4. **Entrega por Destino (Gráfico)**

- Consulta: `sum by (dest_smsc) (rate(sms_c_delivery_succeeded_count[5m]))`
- Unidade: mensagens/segundo
- Gráfico de área empilhada
- Legenda: {{dest_smsc}}

5. **Ações de Queda/Auto-Resposta (Stat)**

- Caídas: `increase(sms_c_routing_action_count{action="drop"}[1h])`
- Auto-Resposta: `increase(sms_c_routing_action_count{action="auto_reply"}[1h])`
- Estatísticas lado a lado

6. **Padrão de Tráfego Horário (Gráfico)**

- Consulta: `rate(sms_c_message_received_count[1h]) * 3600`
- Intervalo de tempo: Últimos 7 dias
- Mostra padrões diários

Dashboard 4: Capacidade e Recursos

Propósito: Monitorar o uso de recursos e limites de capacidade.

Painéis:

1. **Capacidade da Fila (Gráfico)**
   - Atual: `sms_c_queue_size_size`
   - Linha de capacidade: Valor fixo baseado nos limites do sistema
   - Mostra a tendência de utilização
2. **Crescimento das Tabelas do Banco de Dados (Gráfico)**
   - Mensagens: `sms_c_mnesia_table_size_record_count{table="sms_route"}`
   - Traduções: `sms_c_mnesia_table_size_record_count{table="translation_rule"}`
   - Tendência ao longo dos últimos 30 dias
3. **Tendência de Acúmulo de Mensagens (Gráfico)**
   - Consulta: `rate(sms_c_delivery_queued_count[5m]) - (rate(sms_c_delivery_succeeded_count[5m]) + rate(sms_c_delivery_failed_count[5m]))`
   - Positivo = acúmulo crescente
   - Negativo = recuperando
4. **Pico de Tráfego (Stat)**
   - Consulta: `max_over_time(rate(sms_c_message_received_count[5m])[24h:])`
   - Mostra a maior taxa de 5m na última 24h
   - Unidade: mensagens/segundo
5. **Utilização de Capacidade (Gauge)**
   - Consulta: `(rate(sms_c_message_received_count[5m]) / MAX_CAPACITY) * 100`

- Substitua MAX_CAPACITY pelo limite do seu sistema
- Unidade: porcentagem
- Limites:
  - Verde: < 70
  - Amarelo: 70-85
  - Vermelho: > 85

Dashboard 5: Conformidade com SLA

Propósito: Acompanhar métricas de SLA e conformidade.

Painéis:

1. **Conformidade com SLA (Gauge)**
   - Sucesso de Entrega: `(rate(sms_c_delivery_succeeded_count[1h]) / rate(sms_c_delivery_queued_count[1h])) * 100`
   - Linha de alvo em 99%
   - Limites:
     - Vermelho: < 95
     - Amarelo: 95-99
     - Verde: >= 99
2. **Mensagens Entregues Dentro do SLA (Stat)**
   - Consulta: `count(sms_c_delivery_time_delta_delta_ms_bucket{le="5000"}) / count(sms_c_delivery_time_delta_delta_ms_bucket)`
   - Mostra porcentagem entregue dentro de 5 segundos
   - Unidade: porcentagem
3. **Violações de SLA (Counter)**
   - Mensagens que excedem 5 minutos: `increase(sms_c_queue_oldest_message_age_seconds{} > 300)[24h:]`
   - Deve ser 0
4. **Uptime (Stat)**

- Consulta: `up{job="sms-c"}`
- Binário: 1 = ativo, 0 = inativo
- Mostra o status atual

5. **Tendência da Taxa de Sucesso Diária (Gráfico)**
   - Consulta: `avg_over_time((rate(sms_c_delivery_succeeded_count[1h]) / rate(sms_c_delivery_queued_count[1h]))[24h:1h])`
   - Intervalo de tempo: Últimos 30 dias
   - Linha de SLA em 99%

Exemplos de Regras de Alerta

Alertas Críticos

Falhas de Roteamento:

```
alert: RoutingFailuresDetected
expr: increase(sms_c_routing_failed_count[5m]) > 0
for: 2m
labels:
  severity: critical
annotations:
  summary: "{{ $value }} falhas de roteamento nos últimos 5 minutos"
  description: "Mensagens não podem ser roteadas. Verifique a configuração de roteamento."
```

Acúmulo de Fila:

```
alert: MessageQueueBacklog
expr: sms_c_queue_size_pending > 10000
for: 5m
labels:
  severity: critical
annotations:
  summary: "A fila de mensagens tem {{ $value }} mensagens
pendentes"
  description: "A fila está acumulando. Verifique o desempenho de
entrega."
```

Mensagens Antigas na Fila:

```
alert: OldMessagesInQueue
expr: sms_c_queue_oldest_message_age_seconds > 300
for: 2m
labels:
  severity: critical
annotations:
  summary: "A mensagem mais antiga tem {{ $value }} segundos"
  description: "Mensagens não estão sendo entregues. Verifique os
frontends."
```

Todos os Frontends Desconectados:

```
alert: NoActiveFrontends
expr: sum(sms_c_frontend_status_count{status="connected"}) == 0
for: 1m
labels:
  severity: critical
annotations:
  summary: "Nenhum frontend conectado"
  description: "Nenhum caminho de entrega disponível. Verifique a
conectividade do frontend."
```

Crescimento da Fila de Cartas Mortas:

```
alert: DeadLetterMessagesIncreasing
expr: rate(sms_c_delivery_dead_letter_count[10m]) > 0
for: 5m
labels:
  severity: critical
annotations:
  summary: "{{ $value }} mensagens movidas para a fila de cartas
mortas"
  description: "Mensagens estão se tornando não entregáveis.
Investigue as falhas."
```

Alertas de Aviso

Baixa Taxa de Sucesso 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: "A taxa de sucesso de entrega é {{ $value |
humanizePercentage }}"
  description: "Taxa de sucesso abaixo de 95%. Investigue as
falhas de entrega."
```

Alta Taxa de Reenvio:

```
alert: HighDeliveryRetryRate
expr: histogram_quantile(0.95,
sms_c_delivery_succeeded_attempt_count_bucket) > 2
for: 15m
labels:
  severity: warning
annotations:
  summary: "Tentativas de entrega no percentil 95: {{ $value }}"
  description: "Mensagens exigindo múltiplas tentativas. Verifique
a confiabilidade do destino."
```

Processamento de Mensagens Lento:

```
alert: SlowMessageProcessing
expr: histogram_quantile(0.95,
sms_c_message_processing_stop_duration_bucket) > 1000
for: 10m
labels:
  severity: warning
annotations:
  summary: "Tempo de processamento no percentil 95: {{ $value
}}ms"
  description: "O processamento de mensagens está lento. Verifique
os recursos do sistema."
```

Consultas ENUM Falhando:

```
alert: HighEnumFailureRate
expr: rate(sms_c_enum_lookup_stop_duration_count{success="false"}
[10m]) > 0.1
for: 10m
labels:
  severity: warning
annotations:
  summary: "Taxa de falha de consulta ENUM: {{ $value }}"
  description: "Consultas DNS falhando. Verifique os servidores
DNS."
```

Baixa Taxa de Acerto do Cache 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: "Taxa de acerto do cache ENUM: {{ $value |
humanizePercentage }}"
  description: "Baixa eficiência do cache. Pode indicar tráfego de
números únicos."
```

Falhas de Cobrança:

```
alert: ChargingFailuresDetected
expr: rate(sms_c_charging_failed_count[10m]) > 0.05
for: 10m
labels:
  severity: warning
annotations:
  summary: "Taxa de falha de cobrança: {{ $value }}"
  description: "Erros no sistema de cobrança. Verifique a
conectividade com o OCS."
```

Notas Adicionais

- Todas as métricas de duração usam precisão de nanosegundos internamente, mas são convertidas para milissegundos para relatórios
- Métricas de contagem são cumulativas e devem ser usadas com as funções `rate()` ou `increase()` em consultas Prometheus
- Métricas de gauge representam valores instantâneos no momento da coleta
- Métricas de histograma fornecem cálculos percentuais (p50, p95, p99) e podem ser usadas para criar mapas de calor
- Todas as métricas incluem rótulos padrão adicionados pelo Prometheus (instância, trabalho, etc.)

- Ao criar dashboards, use intervalos de tempo apropriados: 5m para tempo real, 1h para tendências, 24h+ para planejamento de capacidade
- Configure regras de gravação no Prometheus para consultas complexas frequentemente usadas para melhorar o desempenho do dashboard
- Use a modelagem de variáveis no Grafana para dashboards dinâmicos (selecionar dest_smsc, source_smsc, etc.)

Centro MMS (MMSc)

← Voltar para o Índice da Documentação | **Referência de Configuração** | **Referência da API**

Índice

- Visão Geral
- Arquitetura
- Processamento de Submissão de Mensagens
- Fluxos de Entrega
- Notificação MMS
- Determinação On/Off-Net
- Configuração
- Armazenamento
- Comportamento em Tempo de Execução
- Observabilidade
- Solução de Problemas

Visão Geral

OmniMessage inclui um **Centro MMS (MMSc)** nativo que recebe, armazena e entrega mensagens multimídia. Ele implementa as duas interfaces de interconexão MMS definidas pelo 3GPP TS 23.140: **MM1** para comunicação direta com o celular (HTTP/WAP) e **MM4** para retransmissão entre operadoras (SMTP).

O MMSc está integrado ao núcleo do OmniMessage. Ele não funciona como um serviço separado e não tem dependência de banco de dados externo ou servidor web. Quando um assinante envia ou recebe um MMS, o MMSc:

1. Armazena o corpo bruto do MMS no sistema de arquivos local e seus metadados no Mnesia.

2. Constrói uma **notificação MMS** e a submete como um SMS comum terminado em móvel, seja na fila de mensagens deste nó ou para o SMS-C definido em `notify_api_url`. O SMS-C então traduz, roteia e entrega como qualquer outro SMS.
3. Serve o MMS armazenado de volta para o celular que o recupera através do MM1 quando busca a URL de notificação.

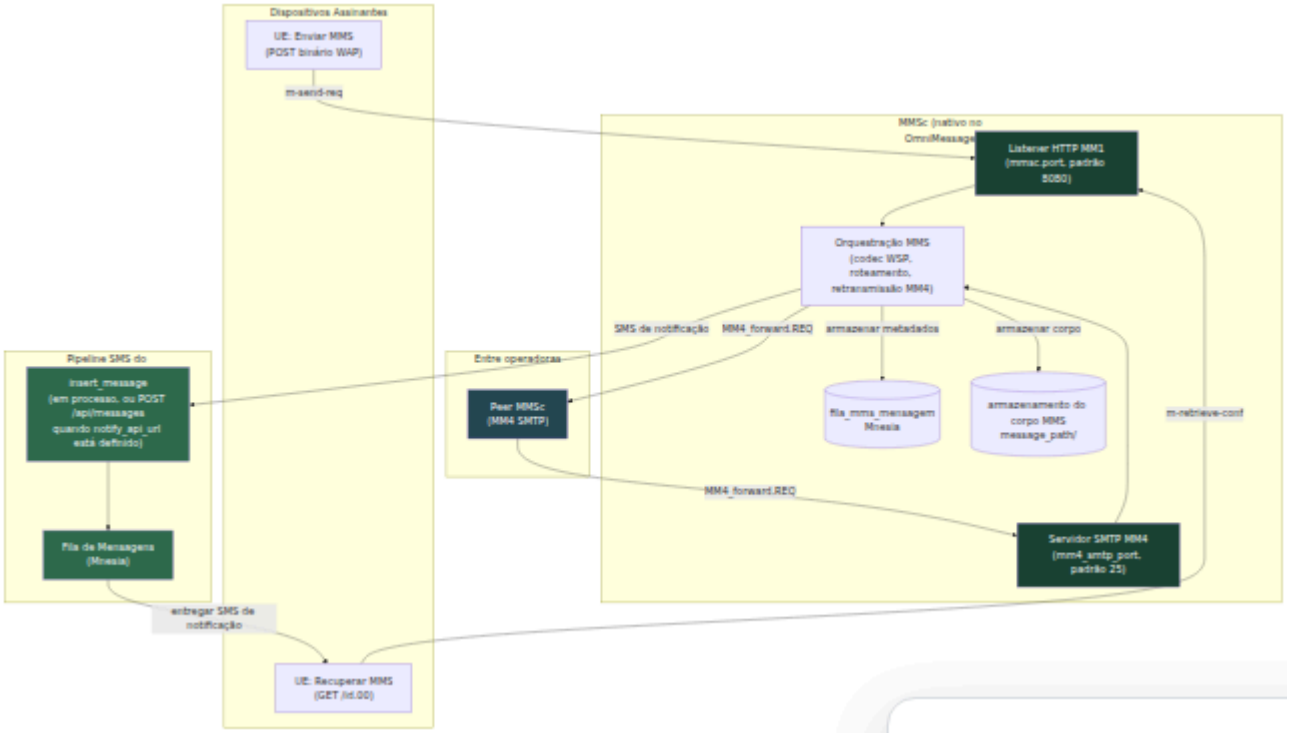
Como a notificação é injetada como um SMS normal, o MMS reutiliza toda a infraestrutura existente de roteamento, tradução e entrega de SMS sem chamadas HTTP autodirigidas.

Especificações Relevantes

Especificação	Título
3GPP TS 23.140	Serviço de Mensagens Multimídia (MMS): descrição funcional
3GPP TS 26.140	MMS: formatos de mídia e codecs
Protocolo de Encapsulamento OMA MMS	Codificação binária WAP MM1

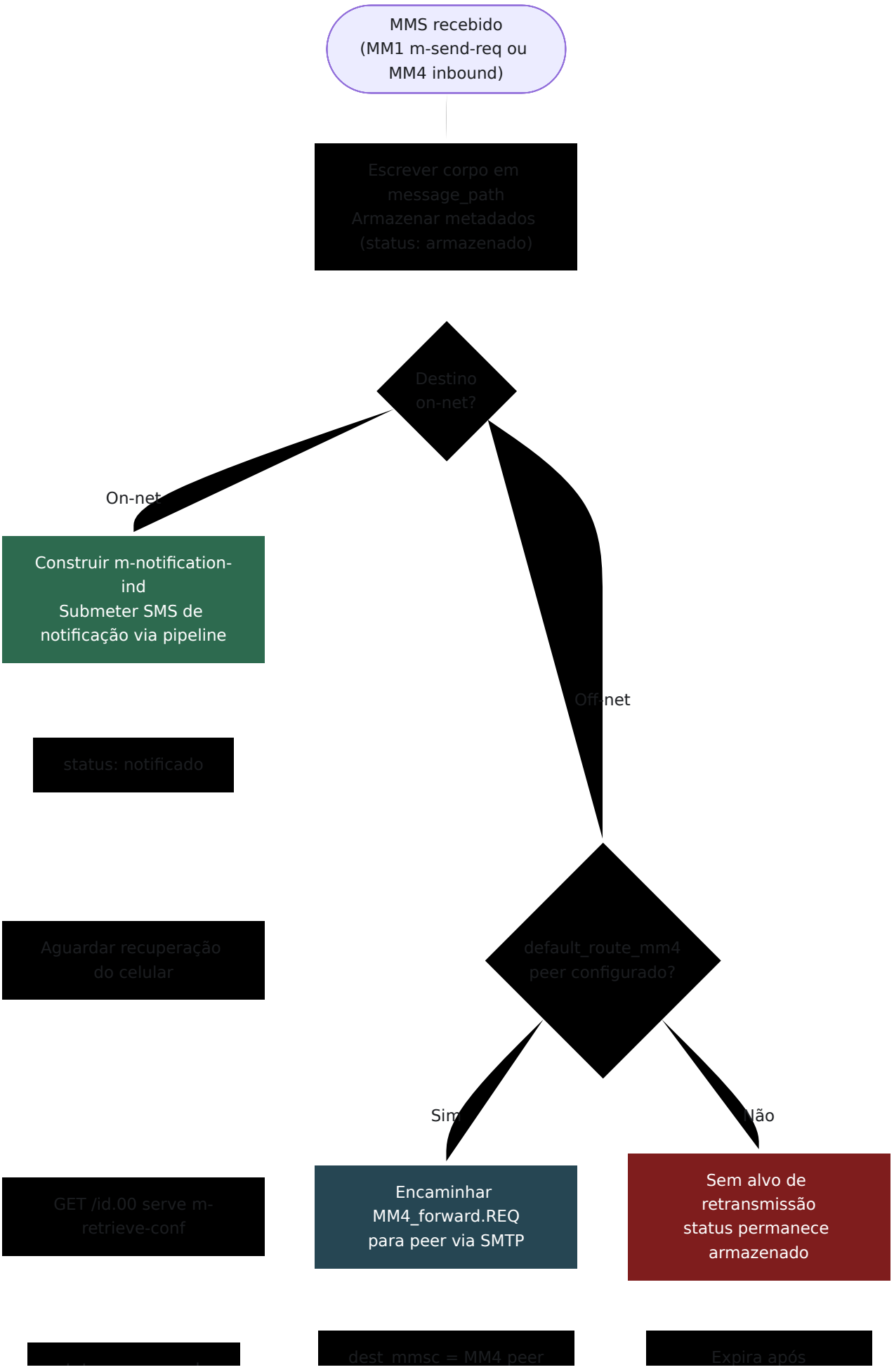
Arquitetura

As interfaces MM1 (HTTP do celular) e MM4 (SMTP entre operadoras) escutam em suas próprias portas dedicadas, separadas da API do OmniMessage. Os metadados do MMS residem no Mnesia ao lado do armazenamento de mensagens SMS. O MMSc mantém os corpos brutos do MMS como arquivos no sistema de arquivos local. O registro de metadados armazena apenas o caminho do arquivo.



Processamento de Submissão de Mensagens

Cada MMS submetido (de um celular via MM1, ou de um par via MM4) segue o mesmo caminho de processamento: armazenar o corpo, armazenar metadados, classificar o destino e, em seguida, notificar localmente ou retransmitir off-net.

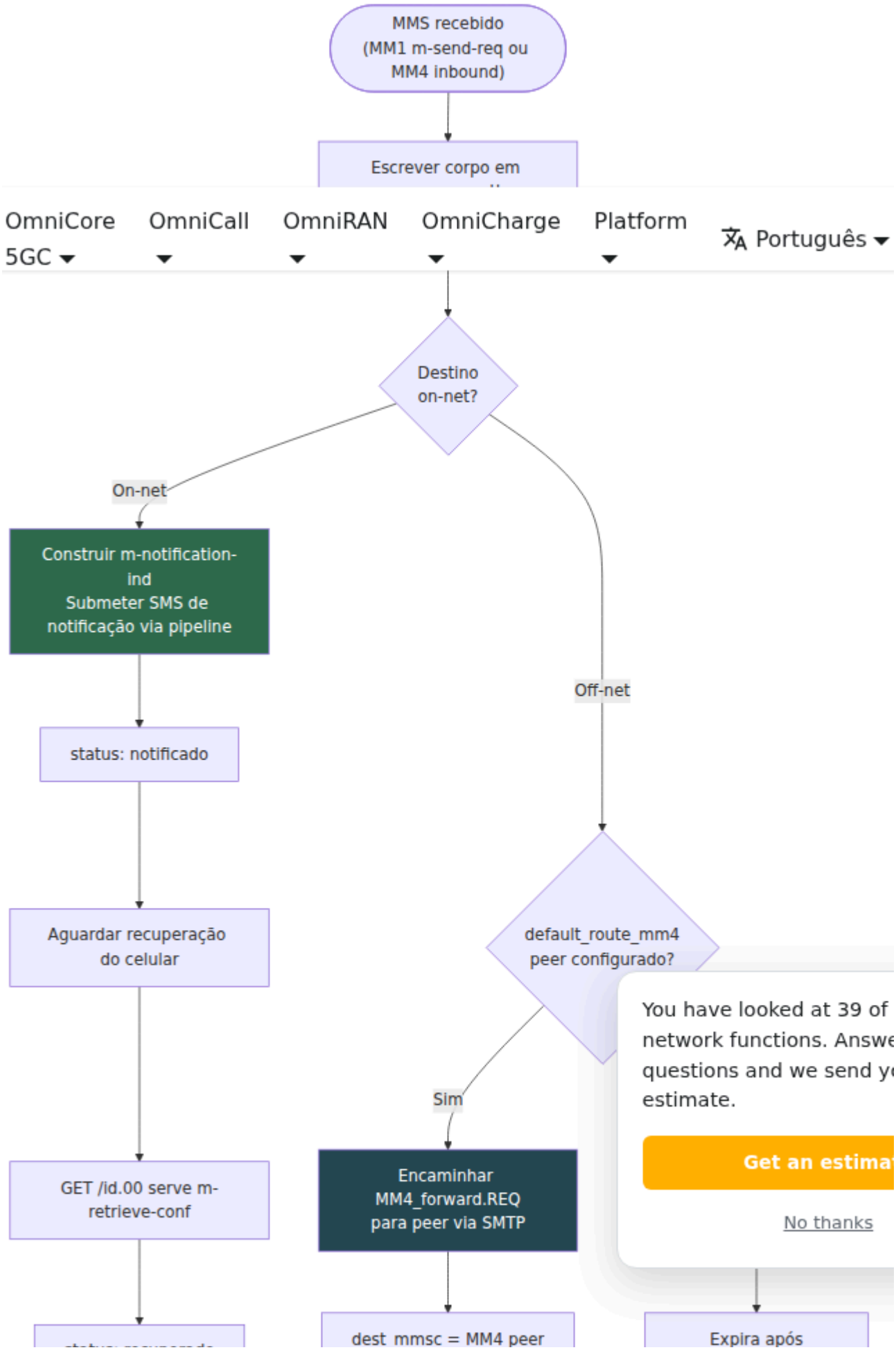

MMS recebido
(MM1 m-send-req ou
MM4 inbound)
Escrever corpo em
message_path
Armazenar metadados
(status: armazenado)
Destino
on-net?
On-net
Construir m-notification-
ind
Submeter SMS de
notificação via pipeline
status: notificado
Off-net
Aguardar recuperação
do celular
default_route_mm4
peer configurado?
Sim
Não
GET /id.00 serve m-
retrieve-conf
Encaminhar
MM4_forward.REQ
para peer via SMTP
Sem alvo de
retransmissão
status permanece
armazenado
dest_mmsc = MM4 peer
Expira após

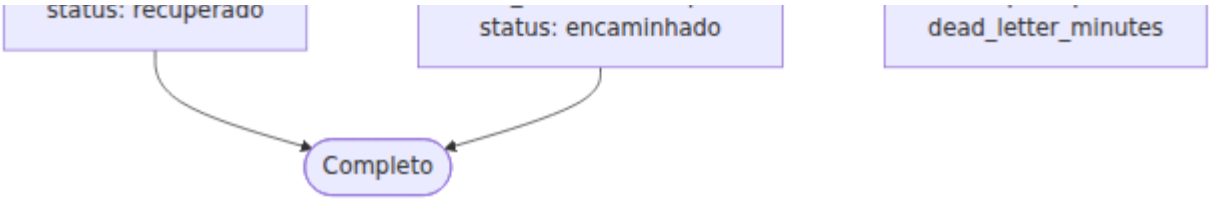


Fluxos de Entrega

Entrega On-Net (MM1)

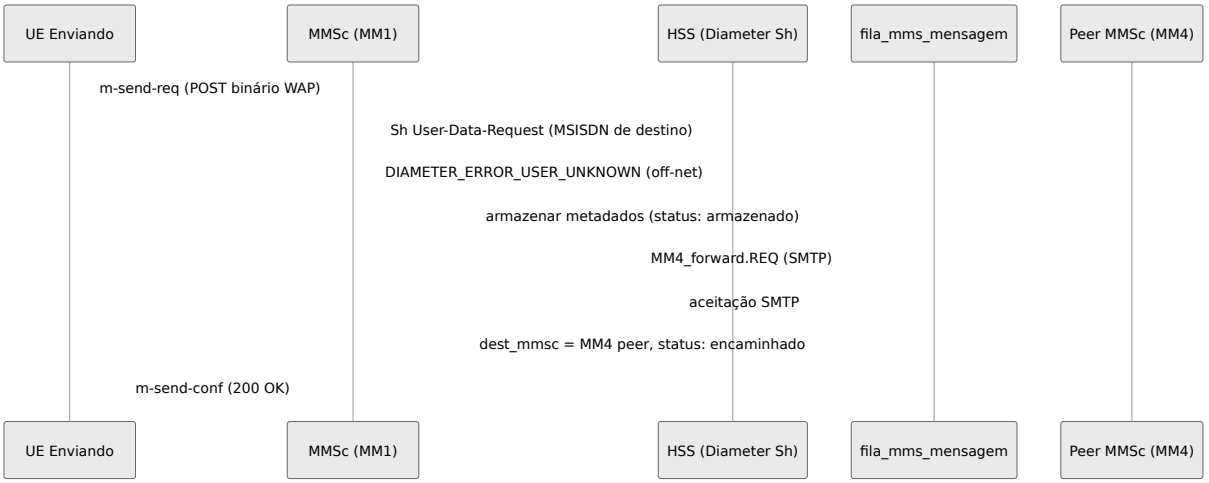
Um assinante envia um MMS para um destinatário atendido pela mesma rede.


MMS recebido (MM1 m-send-req ou MM4 inbound)
Escrever corpo em
OmniCore 5GC
OmniCall
OmniRAN
OmniCharge
Platform
Português
Destino on-net?
On-net
Construir m-notification-ind
Submeter SMS de notificação via pipeline
status: notificado
Aguardar recuperação do celular
GET /id.00 serve m-retrieve-conf
Off-net
default_route_mm4 peer configurado?
Sim
Encaminhar MM4_forward.REQ para peer via SMTP
dest_mmsc = MM4 peer
Expira após
You have looked at 39 of
network functions. Answe
questions and we send y
estimate.
Get an estima
No thanks



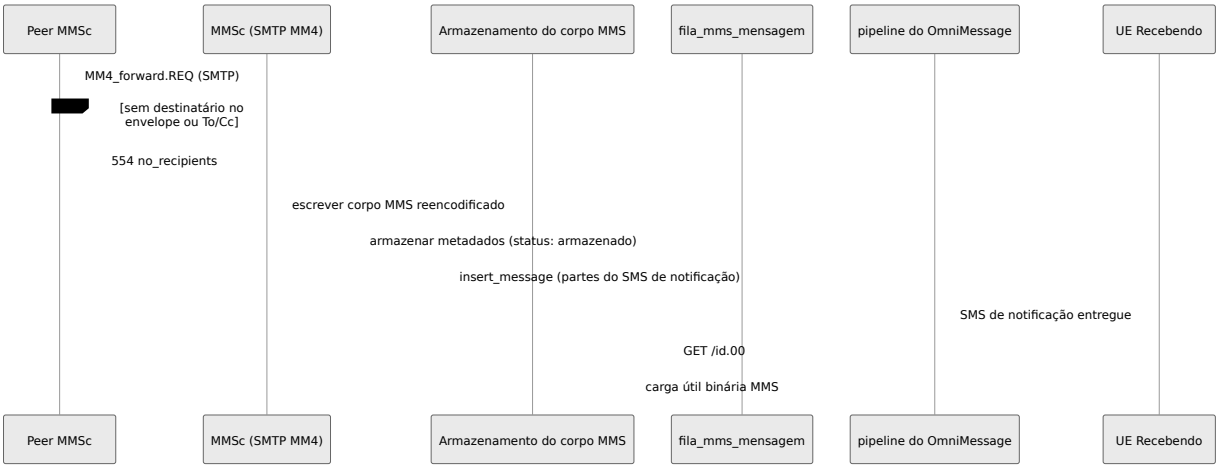
Entrega Off-Net (MM4)

Um assinante envia um MMS para um destinatário em outra operadora.



MM4 Inbound (de uma Operadora Parceira)

Um assinante em outra rede envia um MMS para um assinante on-net.



- **Entrega** vai para os destinatários do envelope SMTP (`RCPT TO`); os pares endereçam um envelope por assinante on-net. Apenas quando o envelope não contém destinatário, os cabeçalhos `To`/`Cc` decidem.
- **Membros do grupo**: `To` + `Cc` juntos formam a lista de membros armazenada em `recipient_list` e listada no `m-retrieve-conf`, para que o

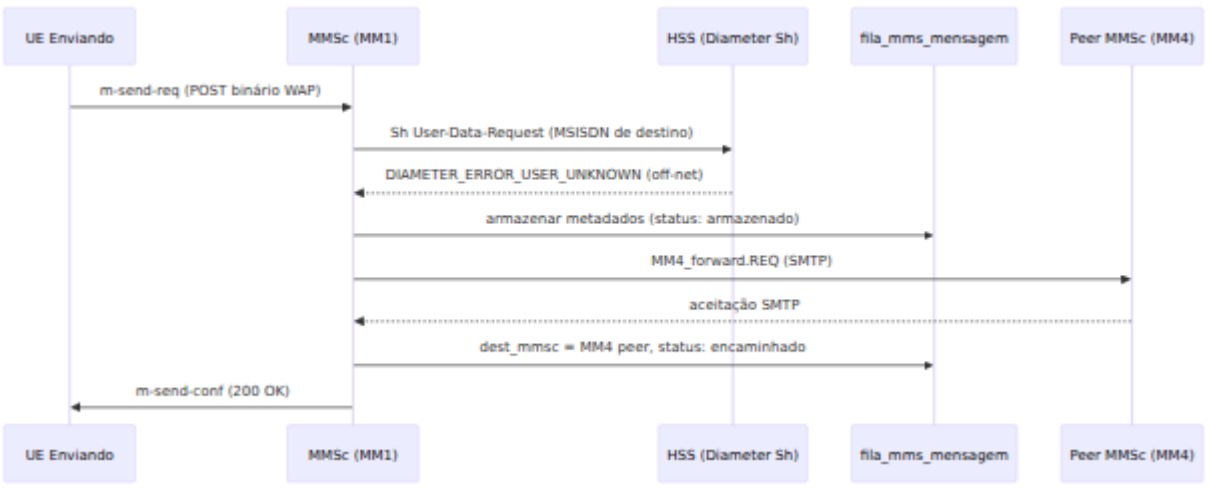
celular mantenha um MMS em grupo em seu thread de grupo.

- **Mídia**: cada parte não-SMIL da árvore MIME (incluindo multiparts aninhados) é reencodificada no corpo WSP armazenado. Partes de texto são convertidas de seu charset declarado para UTF-8.
- Uma mensagem sem destinatário no envelope ou em `To`/`Cc` é recusada com `554` em vez de ser aceita e descartada.

Notificação MMS

A notificação MMS é um SMS binário que transporta um **PDU `m-notification-ind` codificado em WSP**. Ele usa codificação de dados de 8 bits (DCS `0x44`), é dividido em segmentos SMS concatenados e é submetido à fila de mensagens como partes comuns de SMS MT. Cada notificação usa sua própria referência de concatenação, de modo que um celular recebendo várias notificações em rápida sucessão reassemble cada uma separadamente. Portanto, todas as regras de tradução, roteamento e entrega de SMS se aplicam a ela inalteradas.

- Por padrão, as partes são inseridas na fila deste nó (em processo).
- Quando `notify_api_url` está definido, cada parte é submetida a esse controlador SMS-C com `POST /api/messages` e enfileirada lá. Isso permite que o MMSc funcione em seu próprio nó (por exemplo, ao lado de um gateway SMPP) enquanto as notificações são entregues pelo SMS-C principal. Os IDs de mensagem retornados estão vinculados ao MMS como `notify_message_ids`. Uma submissão rejeitada ou falhada é registrada, e o MMS permanece armazenado sem essa parte de notificação.



O local de recuperação codificado na notificação é:

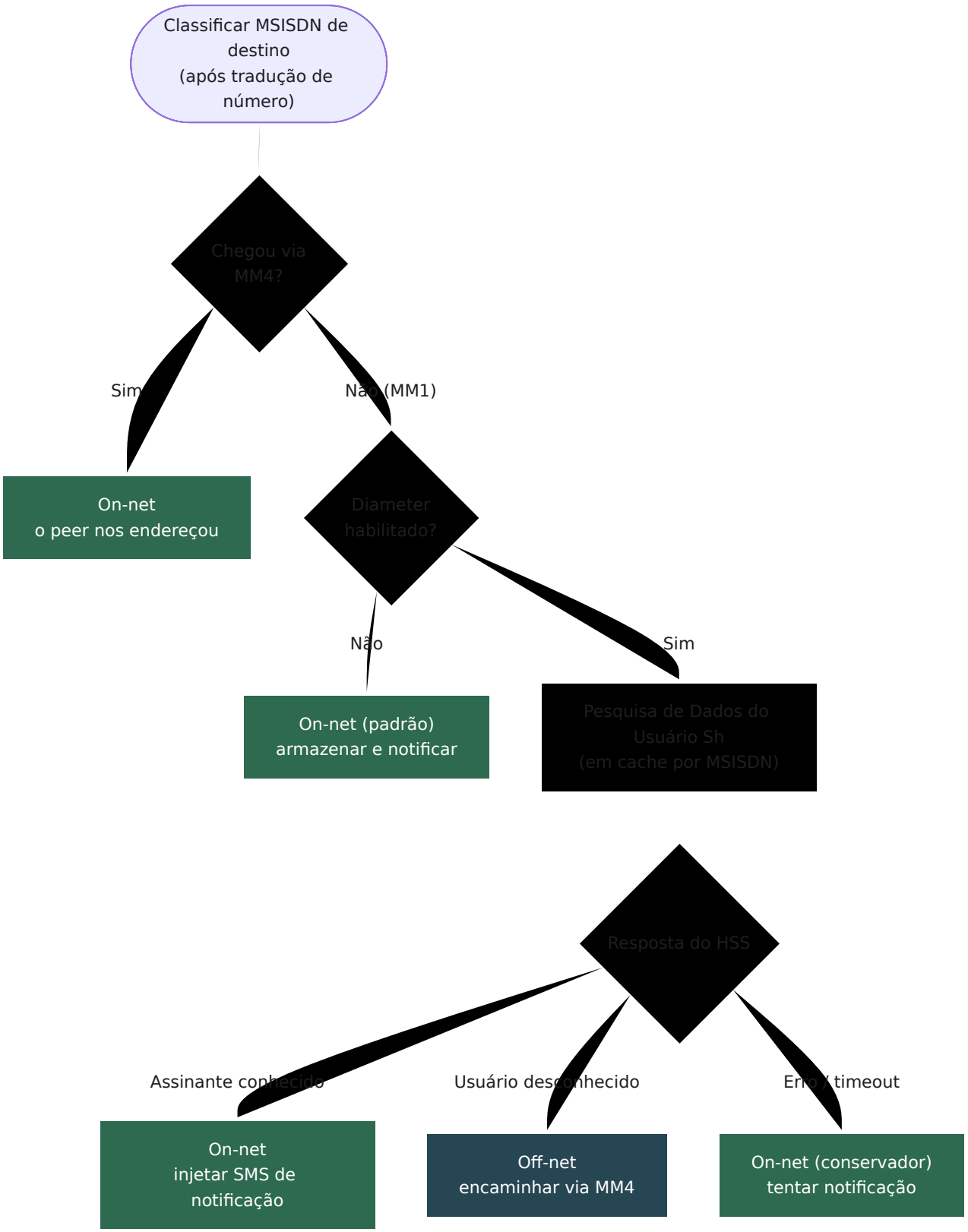
```
http://<mms_domain>/<unique_id>.00
```

- `mms_domain` é o `mmsc.mms_domain` configurado.
- `<unique_id>.00` é a chave de recuperação. O manipulador MM1 remove o sufixo `.00` para resolver a mensagem armazenada.

O celular receptor abre esta URL via MM1 para buscar o MMS como um `m-retrieve-conf`.

Determinação On/Off-Net

Antes que o MMSc decida se deve notificar localmente ou retransmitir via MM4, ele pergunta ao HSS via **Diameter Sh** se o destino é um assinante desta rede. Ele usa a mesma pesquisa em cache do Sh que o SMS-C usa para correspondência de rota `terminating_network`, de modo que MMS e SMS concordem sobre quem está on-net.


Classificar MSISDN de destino
(após tradução de número)
Chegou via MM4?
Sim
Não (MM1)
On-net
o peer nos endereçou
Diameter habilitado?
Não
Sim
On-net (padrão)
armazenar e notificar
Pesquisa de Dados do Usuário Sh
(em cache por MSISDN)
Resposta do HSS
Assinante conhecido
Usuário desconhecido
Erro / timeout
On-net
injetar SMS de notificação
Off-net
encaminhar via MM4
On-net (conservador)
tentar notificação

Condição	Classificação	Ação
MMS recebido via MM4	On-net	Armazenar localmente e injetar o SMS de notificação; nunca retransmitir de volta via MM4
HSS conhece o assinante	On-net	Armazenar localmente e injetar o SMS de notificação
HSS responde `DIAMETER_ERROR_USER_UNKNOWN`	Off-net	Armazenar localmente e encaminhar para o peer via MM4
Erro de pesquisa Sh ou timeout	On-net (conservador)	Armazenar localmente e tentar a notificação; um aviso é registrado
Diameter não habilitado	On-net (padrão)	Armazenar localmente e injetar o SMS de notificação; um aviso é registrado

A retransmissão MM4 só é tentada para destinos que o HSS relata positivamente como desconhecidos. O emparelhamento Diameter Sh é o próprio do SMS-C (`diameter_enabled` e os pares `:diameter_ex`); o MMSc não possui configurações de classificação separadas.

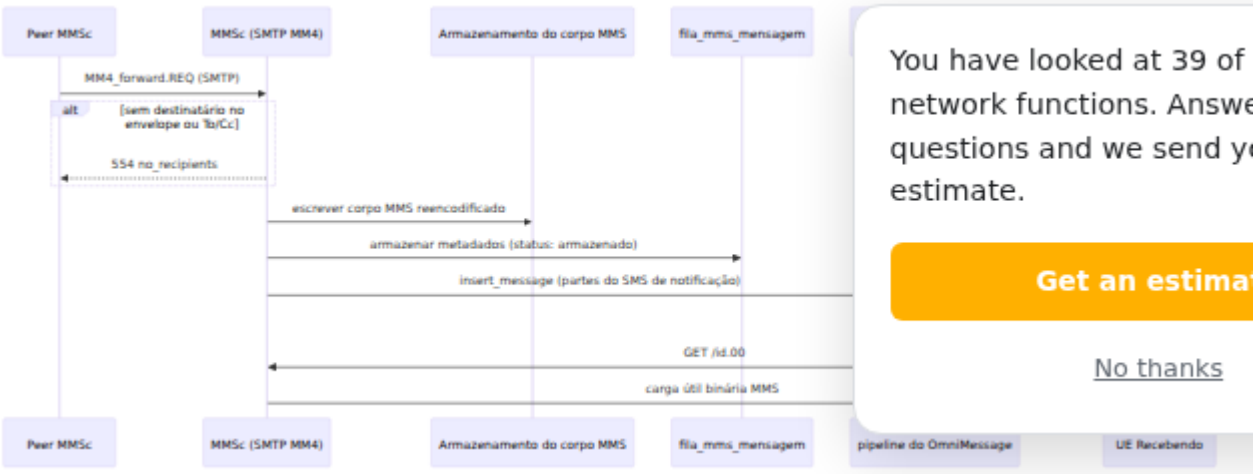
Configuração

Você configura o MMSc em `runtime.exs` sob a chave `:mmsc` da aplicação `:sms_c`. Ele está **desativado por padrão**. Se o bloco `:mmsc` estiver ausente, o

recurso está completamente dormente: o MMSc não vincula portas e não inicia ouvintes. Implantações existentes, portanto, não são afetadas até que você o habilite explicitamente.

Estrutura de Configuração

Cada chave de configuração controla uma parte específica do MMSc. O diagrama mostra quais chaves controlam quais processos.



Exemplo de Configuração

```
# 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 autônomo: submeter SMS de notificação à API do
controlador SMS-C
  # (não definido = inserir na própria fila de mensagens deste nó)
  notify_api_url: nil,

  # Resolução de assinante (IP do cliente MM1 para MSISDN)
  subscriber_ip_api_urls: ["https://10.0.0.10:8443",
"https://10.0.0.11:8443"],

  # MM4 inbound/outbound (entre operadoras)
  mm4_enabled: false,
  mm4_smtp_host: "0.0.0.0",
  mm4_smtp_port: 25,
  mm4_domain: "mms.example.com",
  default_route_mm4: nil
```

Parâmetros `:mmsc`

Parâmetro	Tipo	Requerido	Padrão
`enabled`	Booleano	Não	`false`
`port`	Inteiro	Não	`8080`
`host`	String	Não	`0.0.0.0`
`mms_domain`	String	Sim (se habilitado)	-
`message_path`	String	Não	`/var/lib/omnimessage`
`notify_source_msisdn`	String	Não	`6675`
`dead_letter_minutes`	Inteiro	Não	`1440`
`notify_api_url`	String	Não	`nil`

Parâmetro	Tipo	Requerido	Padrão
`subscriber_ip_api_urls`	Lista de strings	Sim (se habilitado)	`[]`
`mm4_enabled`	Booleano	Não	`false`
`mm4_smtp_host`	String	Não	`0.0.0.0`
`mm4_smtp_port`	Inteiro	Não	`25`
`mm4_domain`	String	Não	recai em `mms_domain`

Parâmetro	Tipo	Requerido	Padrão
`default_route_mm4`	String	Não	`nil`

Pares MM4

Os alvos de retransmissão MM4 outbound são configurados separadamente, porque são registros estruturados de host/porta/domínio em vez de configurações 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` seleciona qual grupo de pares é usado. Cada grupo é uma lista; quando contém várias entradas, uma é escolhida por tentativa de retransmissão para que a carga seja distribuída entre os hosts de entrada de um peer.

Parâmetros do peer MM4:

Parâmetro	Tipo	Requerido	Padrão	Descrição
`host`	String	Sim	-	Nome do host ou IP SMTP MI peer. O MMSc se conecta diretamente a ele, sem uma pesquisa MX.
`port`	Inteiro	Não	`25`	Porta SMTP MM4 do peer.
`mms_domain`	String	Não	`""`	Domínio MMS do peer, usad construir o endereço RCPT (`<msisdn>/TYPE=PLMN@<mms_d` e registrado como `dest_mms`

Nó MMSc Autônomo

O MMSc pode funcionar como seu próprio nó OmniMessage, separado do SMS-C. Por exemplo, pode rodar em um host que já termina as interfaces MM1 e MM4. O nó então lida apenas com MMS, e o SMS-C entrega as notificações.

Configuração	Valor em um MMSc autônomo
`mmsc.notify_api_url`	A URL base da API do SMS-C. SMS de notificação são submetidos lá e entregues pelo SMS-C.
`mmsc.host` / `mmsc.mm4_smtp_host`	Os endereços das interfaces voltadas para o celular e para o peer, para que MM1 e MM4 escutem apenas onde necessário.
`cdr_enabled`	`false`. O nó mantém o estado do MMS no Mnesia e os corpos no disco, e não precisa de banco de dados SQL.
`enum_enabled`	`false`, sem rotas SMS. O nó não roteia SMS.
Diameter (`diameter_enabled` + `:diameter_ex` peers)	Habilitado com um peer Sh, diretamente para o HSS ou através do DRA, para a decisão on/off-net. O Origin-Host do nó deve ser aceito por esse peer.
API / painel de controle	Em um endereço de gerenciamento e uma porta não utilizada por outros serviços no host.

Um nó autônomo alcança três serviços pela rede:

- a API do SMS-C, para notificações;
- a API do HSS (`subscriber_ip_api_urls`), para resolução de remetente;
- o HSS ou DRA via Diameter, para on/off-net.

As métricas do nó mostram falhas de cada um: `notifications` com `result="error"`, solicitações MM1 com status `511`, e decisões de roteamento `sh_error`.

Armazenamento

Os **metadados** do MMS são armazenados na tabela Mnesia `:mms_message_queue`, criada automaticamente na inicialização ao lado do armazenamento de mensagens SMS. Não há banco de dados separado a ser provisionado. O **corpo** bruto do MMS é armazenado como um arquivo `.mms` sob `message_path`; apenas seu caminho é mantido no registro de metadados.

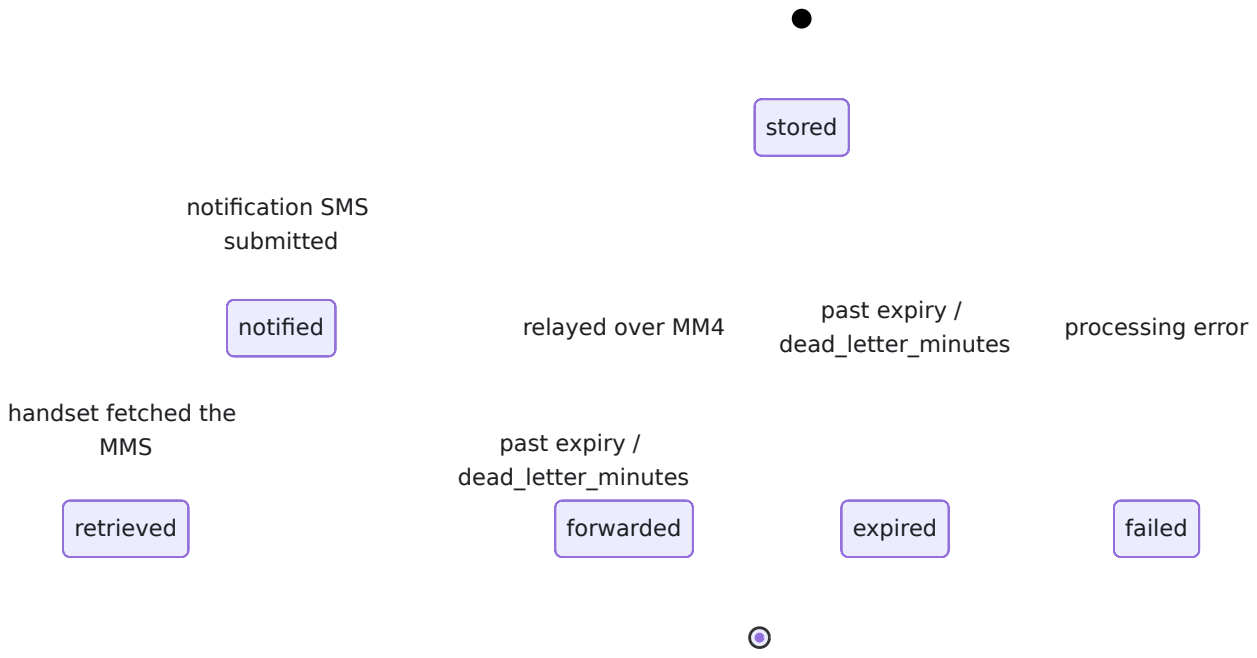
Há um registro de metadados por destinatário. Todos os destinatários de um MMS compartilham o corpo armazenado, e cada um tem seu próprio `unique_id`, de modo que a recuperação e o reconhecimento de um membro do grupo atualizam o registro desse membro.

Campos da `mms_message_queue`

Campo	Descrição
`mms_id`	Chave primária inteira de um contador Mnesia.
`unique_id`	Identificador deste destinatário, usado em seu caminho de URL de recuperação (`<unique_id>.00`).
`source_msisdn`	MSISDN do assinante remetente.
`destination_msisdn`	MSISDN do destinatário principal.
`recipient_list`	Lista completa de To/Cc (CSV) para MMS em grupo e a recuperação-conf.
`subject`	Linha de assunto do MMS.
`file_path`	Caminho para o corpo `.mms` armazenado sob `message_path`.
`source_mmsc`	Identidade do MMSc ou par MM4 que originou a mensagem.
`dest_mmsc`	Definido como `MM4_<peer>` após o revezamento de saída, ou o alvo de entrega após a entrega local.
`on_net`	Se o MMSc classificou o destino como na rede.
`notify_message_ids`	IDs de mensagens SMS criados para a notificação (links para o armazenamento de mensagens SMS).
`send_time`	Quando este MMSc recebeu o MMS.

Campo	Descrição
`notify_time`	Quando o sistema enviou o SMS de notificação.
`retrieve_time`	Quando o assinante recuperou o MMS.
`expires`	Tempo de expiração após o qual o MMS não é entregue.
`status`	Estado do ciclo de vida (veja abaixo).
`deadletter`	Definido quando a mensagem é marcada como carta morta.
`inserted_at` / `updated_at`	Carimbos de data/hora do ciclo de vida do registro.

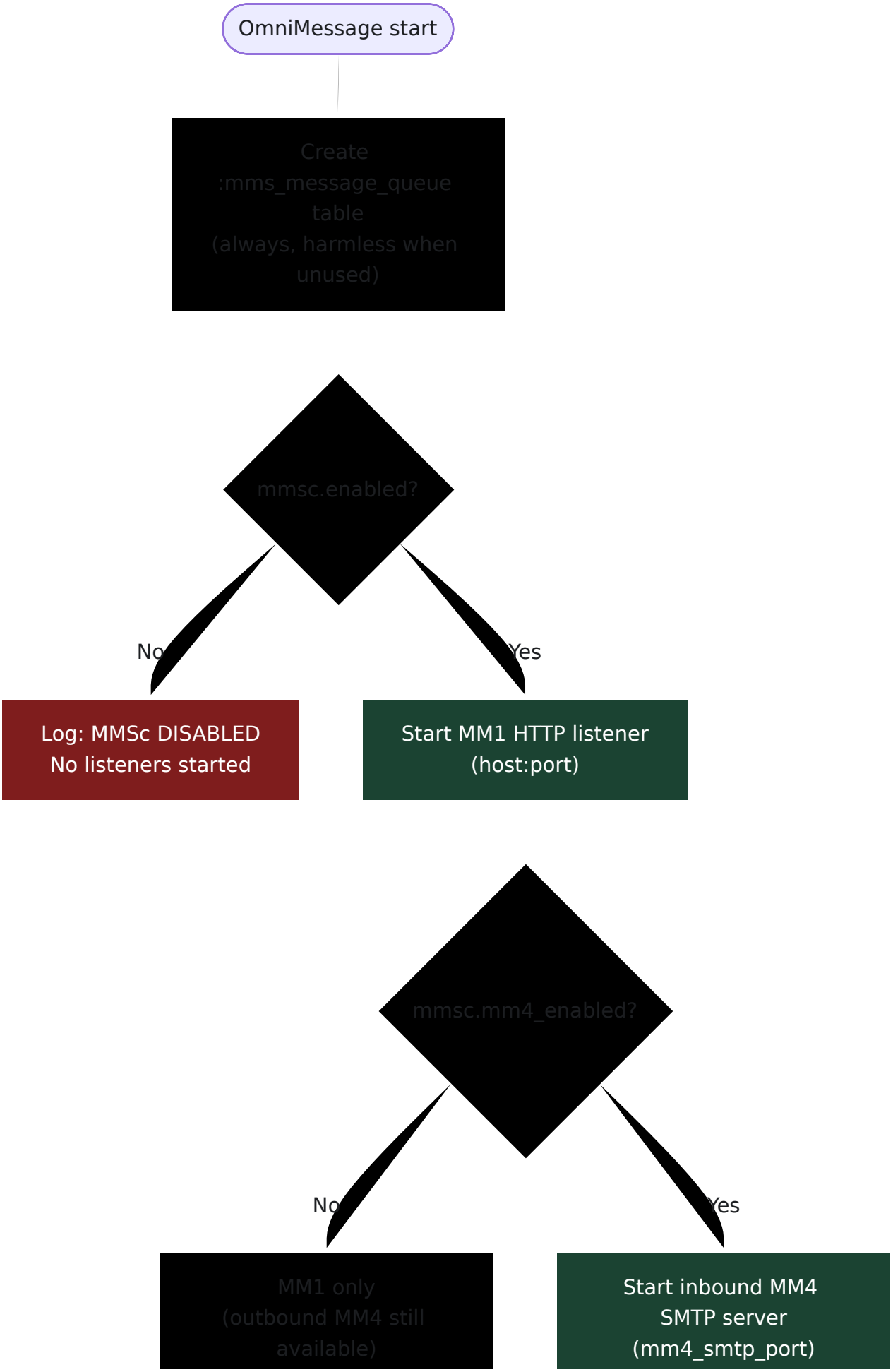
Ciclo de Vida da Mensagem



Status	Significado
`stored`	Corpo escrito e metadados registrados; nenhuma ação de entrega foi realizada ainda.
`notified`	SMS de notificação enviado para o pipeline para um destinatário na rede.
`retrieved`	Destinatário buscou o MMS via MM1.
`forwarded`	Revezado para um transportador par via MM4.
`expired`	Passou sua expiração ou `dead_letter_minutes` sem entrega.
`failed`	Erro de processamento ou entrega.

Comportamento em Tempo de Execução

O MMSc é executado como processos supervisionados dentro da versão do OmniMessage. Nada é implantado ou iniciado separadamente. O diagrama mostra quais portas de configuração cada ouvinte usa na inicialização.


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)

- **Ouvinte HTTP MM1** inicia apenas quando `mmsc.enabled` é `true`. Ele vincula `mmsc.host:mmsc.port` e serve envios de dispositivos móveis (`m-send-req`), respostas de notificação (`m-notifyresp-ind`), reconhecimentos (`m-acknowledge-ind`) e recuperações (`GET /<unique_id>.00`).
- **Servidor SMTP MM4** inicia apenas quando tanto `mmsc.enabled` quanto `mmsc.mm4_enabled` são `true`. Ele vincula `mmsc.mm4_smtp_port` e aceita mensagens `MM4_forward.REQ` de transportadoras pares.
- **Revezamento MM4 de saída** é realizado no processo quando um destino é classificado como fora da rede; ele se conecta ao par configurado via SMTP. Nenhum agente de transferência de e-mail externo é necessário.
- **A criação da tabela** sempre é executada na inicialização. Quando o MMSc está desativado, isso apenas cria uma tabela `:mms_message_queue` vazia, o que é inofensivo.

Vinculação de porta: a porta MM4 padrão (25) é privilegiada. O serviço OmniMessage deve ser executado com a capacidade de vincular portas baixas (`CAP_NET_BIND_SERVICE`) ou usar uma `mm4_smtp_port` mais alta atrás de um redirecionamento de porta. O ouvinte MM1 tem como padrão a porta não privilegiada 8080.

TLS: o ouvinte MM1 serve HTTP simples. Termine o TLS em um proxy reverso na frente de `mmsc.port` se a criptografia voltada para o dispositivo móvel for necessária.

Observabilidade

API da Fila MMS

A API OmniMessage expõe o armazenamento MMS, uma entrada por destinatário. Está listada sob **MMS** na documentação da API (`/api/docs`).

Requisição	Propósito
`GET /api/mms_message_queues`	Lista MMS, os mais novos primeiro. Filtros: `status`, `destination_msisdn`, `limit` (padrão 100).
`GET /api/mms_message_queues/<mms_id>`	Um MMS com suas partes decodificadas e histórico de eventos (recebido, fora da rede, notificação enviada, MM4 revezado ou falhou, recuperado, reconhecido).
`PATCH /api/mms_message_queues/<mms_id>`	Atualizar `status`, `dest_mmsc` ou `deadletter`.
`DELETE /api/mms_message_queues/<mms_id>`	Remover o registro de um destinatário. O corpo armazenado é removido uma vez que nenhum outro registro de destinatário o utilize.
`GET /api/mms_message_queues/<mms_id>/parts`	As partes de mídia do MMS, numeradas de 1, com tipo MIME, nome do arquivo e tamanho.
`GET /api/mms_message_queues/<mms_id>/parts/<n>`	Baixar a parte `n` com seu tipo MIME e nome do arquivo. Um download da API não é uma

Requisição	Propósito
	recuperação de dispositivo móvel: não altera o status e não adiciona evento.

Linha do Tempo de Eventos

Cada registro de destinatário carrega uma linha do tempo de eventos, retornada por `GET /api/mms_message_queues/<mms_id>`:

Evento	Detalhe
`sender_lookup`	Um por nó OmniHSS consultado para o IP do dispositivo móvel que enviou: a URL da requisição, a resposta e o MSISDN encontrado.
`received`	MM1: IP do cliente e cabeçalhos HTTP (User-Agent, X-Forwarded-For, Host, tamanho), os cabeçalhos decodificados `m-send-req` e as partes. MM4: endereço do par SMTP, nome HELO, `MAIL FROM` / `RCPT TO`, cada cabeçalho de e-mail e as partes. Ambos incluem o destinatário e sua chave de recuperação.
`routing`	Na rede ou fora da rede, e por quê: o código de resultado Sh e HSS respondente, uma falha Sh, Diameter desativado, ou chegada via MM4.
`notify_part_submitted` / `notify_part_failed`	Um por parte de SMS de notificação: número da parte e UDH, o ID da mensagem que o SMS-C retornou (ou o erro), e a submissão (status HTTP e URL, ou fila local).
`notify_submitted`	Todos os IDs de mensagens de notificação para o destinatário.
`mm4_forwarded` / `mm4_failed`	Endereço do par, nome EHLO, `MAIL FROM` / `RCPT TO`, tamanho, duração, os cabeçalhos enviados e a resposta do par ou o erro.
`retrieved` / `retrieve_refused`	Cada tentativa de download do dispositivo móvel, atendida ou recusada (por exemplo, 410 quando já recuperado), com o IP do cliente e cabeçalhos HTTP e se o MSISDN por

Evento	Detalhe
	trás desse IP é o destinatário. Um desvio é registrado apenas; não altera o que é servido.
`notifyresp` / `acknowledged`	O `m-notifyresp-ind` / `m-acknowledge-ind` do dispositivo móvel, com seus detalhes e cabeçalhos do cliente.

Os eventos são mantidos na memória durante o período de retenção da mensagem e não são preservados após uma reinicialização.

Métricas

O MMSc exporta essas métricas no endpoint Prometheus do nó (porta 9568):

Métrica	Tipo	Rótulos	De
`sms_c_mmsc_mm1_requests_count`	Contador	`type`, `status`	Solicita por tipo (`m_send` `m_noti` `m_ackn` `retrie` HTTP. U no statu significa consult remete falhand
`sms_c_mmsc_notifications_count`	Contador	`result`, `transport`	Partes d notifica (`transp` `local` c controla
`sms_c_mmsc_mm4_forward_count`	Contador	`peer`, `result`	Reveza para pa
`sms_c_mmsc_mm4_forward_duration_ms`	Histograma	`peer`	Tempo entrega revezar
`sms_c_mmsc_mm4_inbound_count`	Contador	`result`	Mensag recebid `accept` `no_rec` `mm4_pa`
`sms_c_mmsc_routing_decisions_count`	Contador	`decision`	Classifi destina `on_net`

Métrica	Tipo	Rótulos	De
			`sh_err` `diamet` Um aur `sh_err` que des fora da ser noti vez de
`sms_c_mmsc_store_messages`	Gauge	`status`	MMS ar por stat

SMS de notificação também aparecem nas próprias métricas de mensagens do SMS-C, fila e CDRs como qualquer outro SMS. Veja Referência de Métricas.

Solução de Problemas

MMSc Não Inicia

Sintomas: Nenhum ouvinte MM1 na porta configurada; logs de inicialização mostram `MMSc: DISABLED`.

Causas possíveis:

- O bloco `:mmsc` está ausente de `runtime.exs`.
- `mmsc.enabled` é `false`.

Resolução:

1. Adicione o bloco de configuração `:mmsc` e defina `enabled: true`.
2. Reinicie o serviço e confirme que o log de inicialização relata `MMSc: ENABLED` com a porta e domínio esperados.

Dispositivo Móvel Não Consegue Recuperar o MMS

Sintomas: O SMS de notificação é entregue, mas o dispositivo móvel falha ao buscar o conteúdo.

Causas possíveis:

- `mms_domain` não é resolvível ou não é roteável a partir do APN MMS do dispositivo móvel.
- Um proxy reverso ou firewall está bloqueando `mmsc.port`.
- O arquivo `.mms` armazenado está ausente sob `message_path`.

Resolução:

1. Confirme que o host da URL de notificação (`mms_domain`) é resolvível e acessível a partir do APN MMS.
2. Verifique o acesso de entrada a `mmsc.host:mmsc.port` (e qualquer proxy que termine TLS).
3. Verifique o `status` da mensagem e se o arquivo `.mms` existe em `file_path`.

MMS Fora da Rede Não É Revezado

Sintomas: MMS para outra operadora está armazenado, mas nunca é revezado; `status` permanece `stored`.

Causas possíveis:

- `default_route_mm4` não está definido, então nenhum grupo par é selecionado.
- A entrada do par em `:mmsc_mm4_peers` tem o host/porta errados.
- O destino foi classificado como na rede: o HSS conhece o número, a consulta Sh falhou, ou o Diameter não está habilitado (cada um desses padrões é na rede; os dois últimos registram um aviso).

Resolução:

1. Defina `default_route_mm4` para uma chave presente em `:mmsc_mm4_peers`.
2. Verifique o `host`/`port` do par e se o SMTP de saída para ele é permitido.
3. Verifique `diameter_enabled` e se o par Sh (HSS ou DRA) está conectado, para que números fora da rede retornem como usuário-desconhecido.

MM4 Recebido ou Não Recebido

Sintomas: Uma operadora par não consegue entregar via MM4.

Causas possíveis:

- `mm4_enabled` é `false`, então o servidor SMTP não está em execução.
- A mensagem não nomeia nenhum destinatário no envelope SMTP ou nos cabeçalhos `To`/`Cc`, então é recusada com `554 no_recipients`.
- O serviço não tem permissão para vincular a porta privilegiada 25.
- Um firewall está bloqueando SMTP de entrada.

Resolução:

1. Defina `mm4_enabled: true` e reinicie.
2. Conceda `CAP_NET_BIND_SERVICE` ou defina uma `mm4_smtp_port` não privilegiada atrás de um redirecionamento de porta.
3. Permita tráfego de entrada para `mm4_smtp_port` a partir dos endereços de saída do par.

Falha na Resolução do Endereço do Remetente

Sintomas: Submissões de dispositivos móveis não-locais são rejeitadas com um erro de consulta de assinante.

Causas possíveis:

- `subscriber_ip_api_urls` não está definido, então apenas localhost é resolvível para um MSISDN de teste.
- Nenhum nó HSS configurado possui um vínculo de IP para o dispositivo móvel (a sessão de dados do UE ainda não é conhecida pelo HSS).

- Os endpoints da API OmniHSS são inacessíveis ou retornando respostas não-200.

Resolução:

1. Configure `subscriber_ip_api_urls` com cada nó OmniHSS para resolução de remetente em produção.
2. Verifique se os endpoints são acessíveis e retornam o mapeamento esperado de assinante/MSISDN.

Guia de Tradução de Números SMS-C

← Voltar ao Índice da Documentação

Visão Geral

O sistema de Tradução de Números SMS-C fornece uma transformação flexível e baseada em regex de números de telefone antes do roteamento. As regras de tradução podem normalizar números, adicionar prefixos internacionais, formatar números para gateways específicos e encadear múltiplas transformações. As regras são armazenadas no Mnesia para persistência e podem ser modificadas em tempo de execução sem interrupção do serviço.

Principais Recursos

- **Correspondência baseada em prefixo**: Correspondência de números por prefixo antes de aplicar transformações
- **Transformação baseada em regex**: Poderosa correspondência de padrões e substituição com grupos de captura
- **Filtragem de SMSC de origem**: Aplicar diferentes traduções com base na origem da mensagem
- **Avaliação baseada em prioridade**: Controlar a ordem das regras com prioridades configuráveis (1-255)
- **Encadeamento de regras**: Continuar o processamento através de múltiplas regras com prevenção de loops
- **Transformações de chamada/chamada separadas**: Transformação independente para números de origem e destino
- **Carregamento de arquivo de configuração**: Carregar regras iniciais a partir de `runtime.exs` na primeira inicialização
- **Configuração em tempo de execução**: Adicionar, modificar ou desativar regras sem reiniciar

- **Interface Web**: Interface completa de CRUD para gerenciamento de regras
- **Ferramenta de simulação**: Testar a lógica de tradução com avaliação passo a passo
- **Backup/Restauração**: Exportar e importar configurações de tradução
- **Integração pré-roteamento**: Traduções aplicadas antes do roteamento para formatos de número consistentes

Arquitetura

Modelo de Dados

Cada regra de tradução contém os seguintes campos:

Campo	Tipo	Descrição	Obrigatório
`rule_id`	inteiro	Identificador único auto-incrementado	Sim (auto)
`calling_prefix`	string/nil	Correspondência de prefixo para número de chamada (nil = curinga)	Não
`called_prefix`	string/nil	Correspondência de prefixo para número chamado (nil = curinga)	Não
`source_smsc`	string/nil	Nome do SMSC de origem (nil = curinga)	Não
`calling_match`	string/nil	Padrão regex para corresponder ao número de chamada. Também atua como o **gate** para a sobreposição de TON/NPI de origem	Não
`calling_replace`	string/nil	Padrão de substituição para número de chamada	Não
`called_match`	string/nil	Padrão regex para corresponder ao número chamado	Não
`called_replace`	string/nil	Padrão de substituição para número chamado	Não
`source_ton`	string/nil	Forçar o **tipo** de número do endereço de origem. Em branco/ausente = deixar inferido	Não

Campo	Tipo	Descrição	Obrigatório
`source_npi`	string/nil	Forçar o **indicador** do plano de numeração do endereço de origem. Em branco/ausente = deixar inferido	Não
`priority`	inteiro	Prioridade da regra (1-255, menor = maior prioridade)	Sim
`description`	string	Descrição legível por humanos	Não
`enabled`	booleano	Habilitar/desabilitar regra	Sim
`continue`	booleano	Continuar avaliando regras após a correspondência (padrão: falso)	Não

Nota: As regras são avaliadas em ordem de prioridade (número mais baixo primeiro). Apenas regras habilitadas são avaliadas.

Veja Forçando o Tipo de Número (TON) e o Indicador do Plano de Numeração (NPI) para a documentação completa dos campos `source_ton` e `source_npi`.

Algoritmo de Tradução

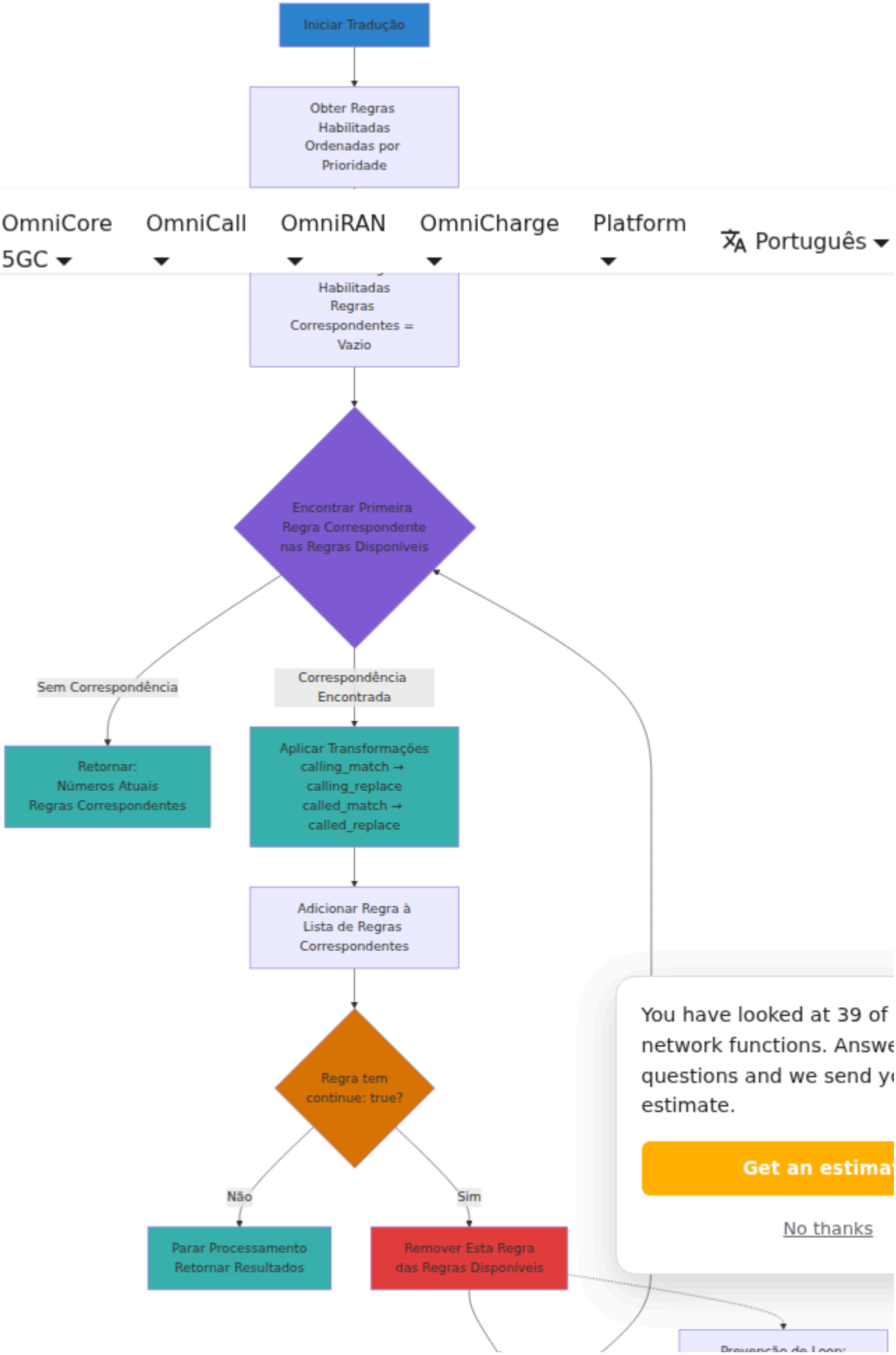
Ao traduzir números, o sistema:

1. **Recupera regras habilitadas** ordenadas por prioridade (mais baixa primeiro)
2. **Avalia regras sequencialmente** com base nos parâmetros da mensagem:
   - Correspondência de `calling_prefix` (se especificado)
   - Correspondência de `called_prefix` (se especificado)
   - Correspondência de `source_smsc` (se especificado)

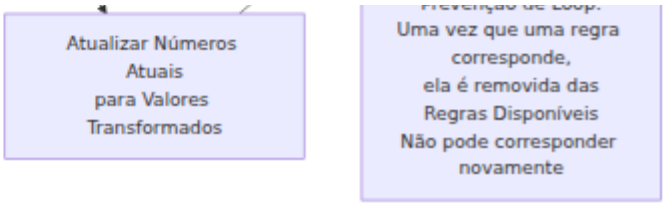
3. **Aplica a primeira regra correspondente**:
    - Transformar o número de chamada usando `calling_match` e `calling_replace`
    - Transformar o número chamado usando `called_match` e `called_replace`
4. **Verifica a flag de continuação**:
    - Se `continue: false` → Parar o processamento, retornar resultado
    - Se `continue: true` → Remover a regra correspondente das regras disponíveis, continuar com o passo 2 usando **números transformados**
5. **Retorna números finais** e lista de todas as regras aplicadas

Encadeamento de Regras com Prevenção de Loop

A flag `continue` permite um poderoso encadeamento de regras enquanto previne loops infinitos:



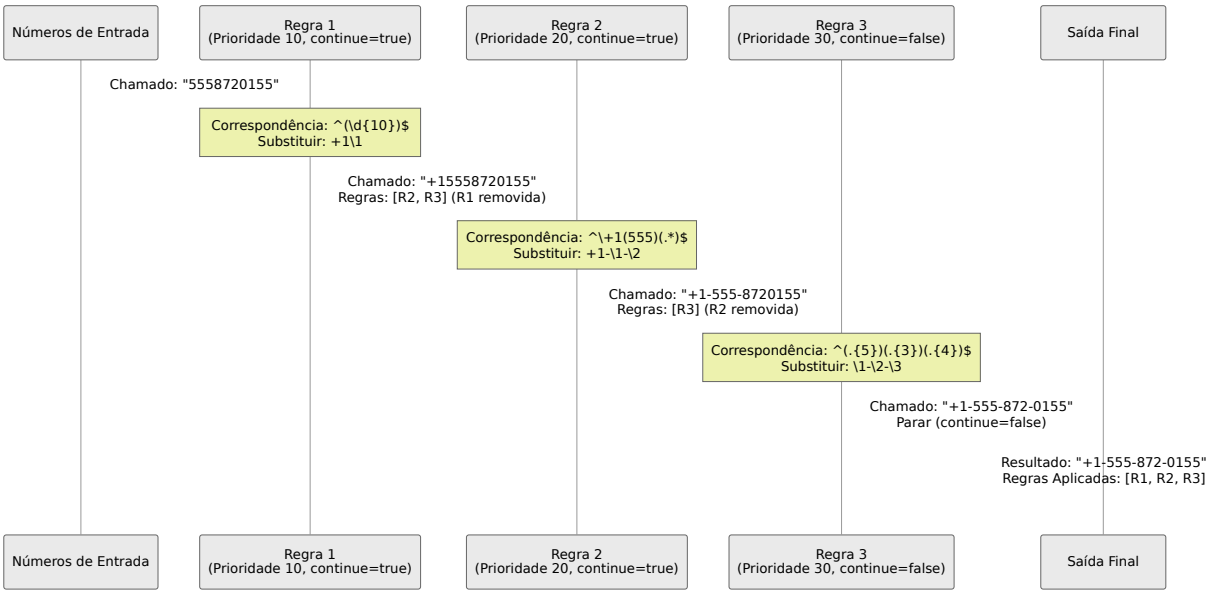
You have looked at 39 of
network functions. Answe
questions and we send y
estimate.
Get an estima
No thanks



Curiosidades

- `nil` ou valores vazios atuam como curingas que correspondem a qualquer valor
- Uma regra sem critérios de correspondência é uma regra catch-all
- Uma regra sem padrões de transformação (correspondência/substituição nil) passa números inalterados

Exemplo: Cenário de Encadeamento de Regras



Forçando Tipo de Número (TON) e Indicador do Plano de Numeração (NPI)

Além da reescrita de números baseada em regex, uma regra de tradução pode **forçar** o Tipo de Número (TON) e o Indicador do Plano de Numeração (NPI) do endereço de origem (originante). Isso é útil quando a string do número do remetente sozinha não transmite a semântica correta do endereço. Por

exemplo, um código curto deve ser sinalizado como `unknown`/`unknown` em vez de tratado como um número nacional ou internacional.

Essas sobreposições são independentes da reescrita de números: uma regra pode forçar TON/NPI **sem** mudar o número.

Os Campos de Sobrescrição

Campo	Aplica-se a	Efeito
`source_ton`	Endereço de origem (originante)	Força o Tipo de Número no endereço SMS-DELIVER originante construído (o TPDU)
`source_npi`	Endereço de origem (originante)	Força o Indicador do Plano de Numeração no endereço SMS-DELIVER originante construído (o TPDU)

Em branco ou ausente = sem sobreposição. Quando um campo é deixado em branco (ou a regra está ausente completamente), o valor do endereço é **inferido** em vez disso - veja Fallback de Inferência abaixo.

> **Nota:** `source_ton`/`source_npi` são incorporados diretamente no endereço originante codificado no **TPDU** (o SMS-DELIVER que o frontend IMS/SS7 transmite). O valor reside no TPDU codificado - nada extra é armazenado no registro da mensagem. Não há sobreposição de TON/NPI de destino: o TPDU SMS-DELIVER não carrega endereço de destino, e frontends derivam endereçamento de destino por conta própria.

Valores Permitidos

Tipo de Número (TON):

Valor	Significado
`unknown`	Tipo de número desconhecido
`international`	Número internacional (E.164 completo com código do país)
`national`	Número nacional (significativo)
`network_specific`	Número específico da rede
`subscriber`	Número de assinante
`alphanumeric`	Endereço alfanumérico (ex: um ID de remetente de nome de marca)
`abbreviated`	Número abreviado

Indicador do Plano de Numeração (NPI):

Valor	Significado
`unknown`	Plano de numeração desconhecido
`isdn`	Plano de numeração ISDN/telefônico (E.164)
`data`	Plano de numeração de dados (X.121)
`telex`	Plano de numeração Telex
`land_mobile`	Plano de numeração móvel terrestre (E.212)
`national`	Plano de numeração nacional
`private`	Plano de numeração privado
`ermes`	Plano de numeração ERMES

Qualquer valor fora desses conjuntos é rejeitado quando a regra é criada ou atualizada.

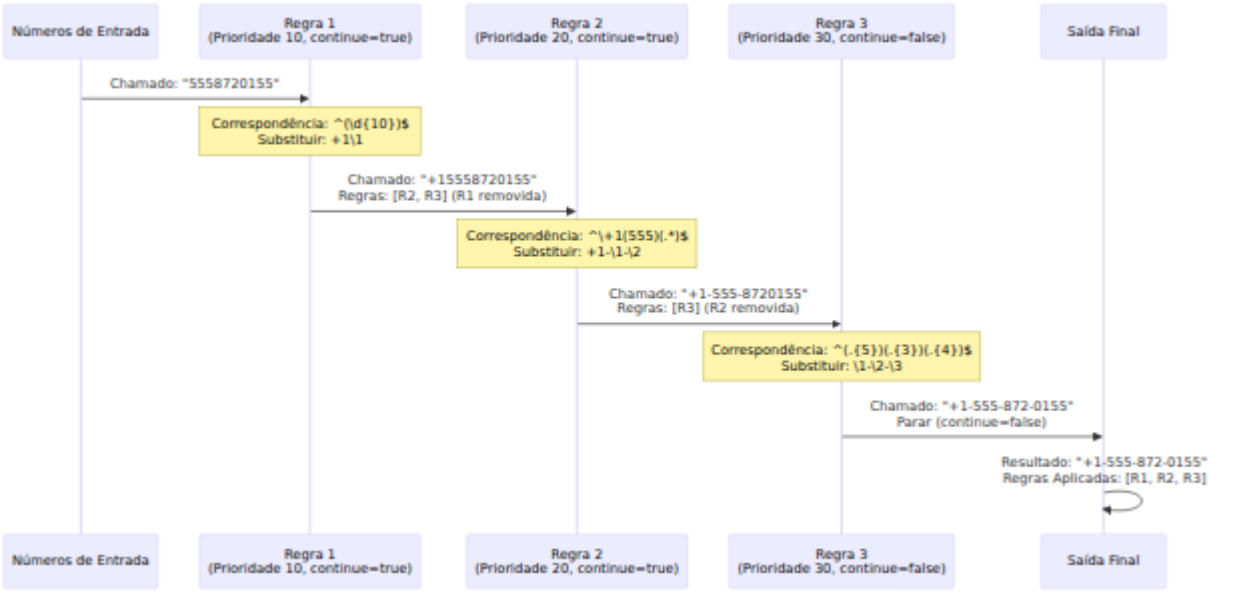
Correspondência de Gating: Escopo de uma Sobrescrição para um Padrão de Remetente

A sobreposição é **gated** pelo regex de **Correspondência de Número de Chamada (`calling_match`)** da regra, que permite que uma única regra tenha como alvo um subconjunto de remetentes por padrão:

- A sobreposição de origem (`source_ton`/`source_npi`) é aplicada apenas quando o regex `calling_match` da regra corresponde ao número de chamada.
- Uma **correspondência em branco é um curinga** - a sobreposição se aplica sempre que a regra for selecionada (por prefixo e/ou SMSC de origem).

Uma regra pode ter um `calling_match` com **nenhum** correspondente `calling_replace`. Nessa configuração, a correspondência atua puramente

como um **gate** para a sobreposição e não **reescreve** o número. Esta é a maneira recomendada de aplicar uma sobreposição de TON/NPI apenas para aqueles números que correspondem a um padrão.



Quando `continue: true` encadeia várias regras, as sobreposições de TON/NPI se acumulam com semântica de **última gravação ganha**: uma regra posterior que define um valor não em branco sobrescreve uma anterior; um valor em branco nunca limpa uma sobreposição previamente definida.

Fallback de Inferência

Quando **nenhuma** regra força uma sobreposição, os valores do endereço de origem são **inferidos** a partir da string do número no momento da construção do TPDU:

Entrada	TON Inferido	NPI Inferido
Numérico, ≥ 5 dígitos	`international`	`isdn`
Numérico, < 5 dígitos	`national`	`isdn`
Não numérico (remetente alfanumérico, ex: `Acme`)	`alphanumeric`	`unknown`

Uma sobreposição forçada sempre tem precedência sobre o valor inferido. Forçar apenas o TON (deixando o NPI em branco) sobrescreve o TON enquanto deixa o NPI inferido, e vice-versa.

Configurando Sobrescrições de TON/NPI

Esses campos podem ser configurados através de três interfaces:

- **Interface Web** - o editor de regras de Tradução de Números expõe dropdowns de TON de Origem e NPI de Origem. Deixar um dropdown não definido significa "sem sobreposição".
- **API REST** - os endpoints `/api/translations` aceitam `source_ton` e `source_npi` como valores de string (veja Exemplos abaixo).
- **Configuração** - a lista `config :sms_c, :translation_rules` aceita os mesmos campos como valores de string, carregados na primeira inicialização.

Exemplo de Trabalho: Remetentes de Código Curto

Um remetente que é um código curto de 3 a 6 dígitos deve ser sinalizado como TON/NPI `unknown`/`unknown` **sem** reescrever o número. A regra usa `calling_match` puramente como um gate (sem `calling_replace`):

Configuração (`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 curtos (3-6 dígitos) => unknown/unknown
TON/NPI",
    enabled: true
  }
]
```

Como funciona: A regra é avaliada para cada mensagem (nenhum `calling_prefix`, `called_prefix`, ou `source_smsc` a restringe). O regex `calling_match` `^\d{3,6}$` controla a sobreposição de origem, então `source_ton`/`source_npi` são forçados a `unknown`/`unknown` **apenas** quando o número de chamada é uma string de 3 a 6 dígitos. Como `calling_replace` é `nil`, o número em si permanece inalterado. `continue: true` permite que regras posteriores sejam executadas também.

Caso de uso: Tráfego originado por aplicativos de códigos curtos que, de outra forma, seriam mal inferidos como números `national` (4 dígitos) ou `international` (5-6 dígitos).

Requisição equivalente da 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 curtos de operador (3-6 dígitos) =>
unknown/unknown TON/NPI",
  "enabled": true
}
```

A API retorna a regra criada (incluindo seu `rule_id` atribuído) com os valores TON/NPI serializados como strings.

Exemplo de Trabalho: Forçar Nacional para um Intervalo Conhecido

Forçar o endereço de origem de um intervalo de números locais conhecidos para `national`/`isdn`, sem reescrever o número:

Configuração (`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 locais 07xxxxxxxx => national/isdn",
  enabled: true
}
```

Como funciona: `calling_prefix: "07"` restringe a regra a números que começam com `07`, e o regex `calling_match` controla a sobreposição para exatamente um `07` seguido de 8 dígitos. Quando correspondido, o endereço de origem é sinalizado como `national`/`isdn` em vez do inferido `international`/`isdn` (que se aplicaria porque o número tem ≥ 5 dígitos). O número não é reescrito.

Caso de uso: Um gateway ou par que exige que números locais formatados nacionalmente sejam sinalizados como TON `national`, mesmo que sejam armazenados em sua forma de discagem bruta.

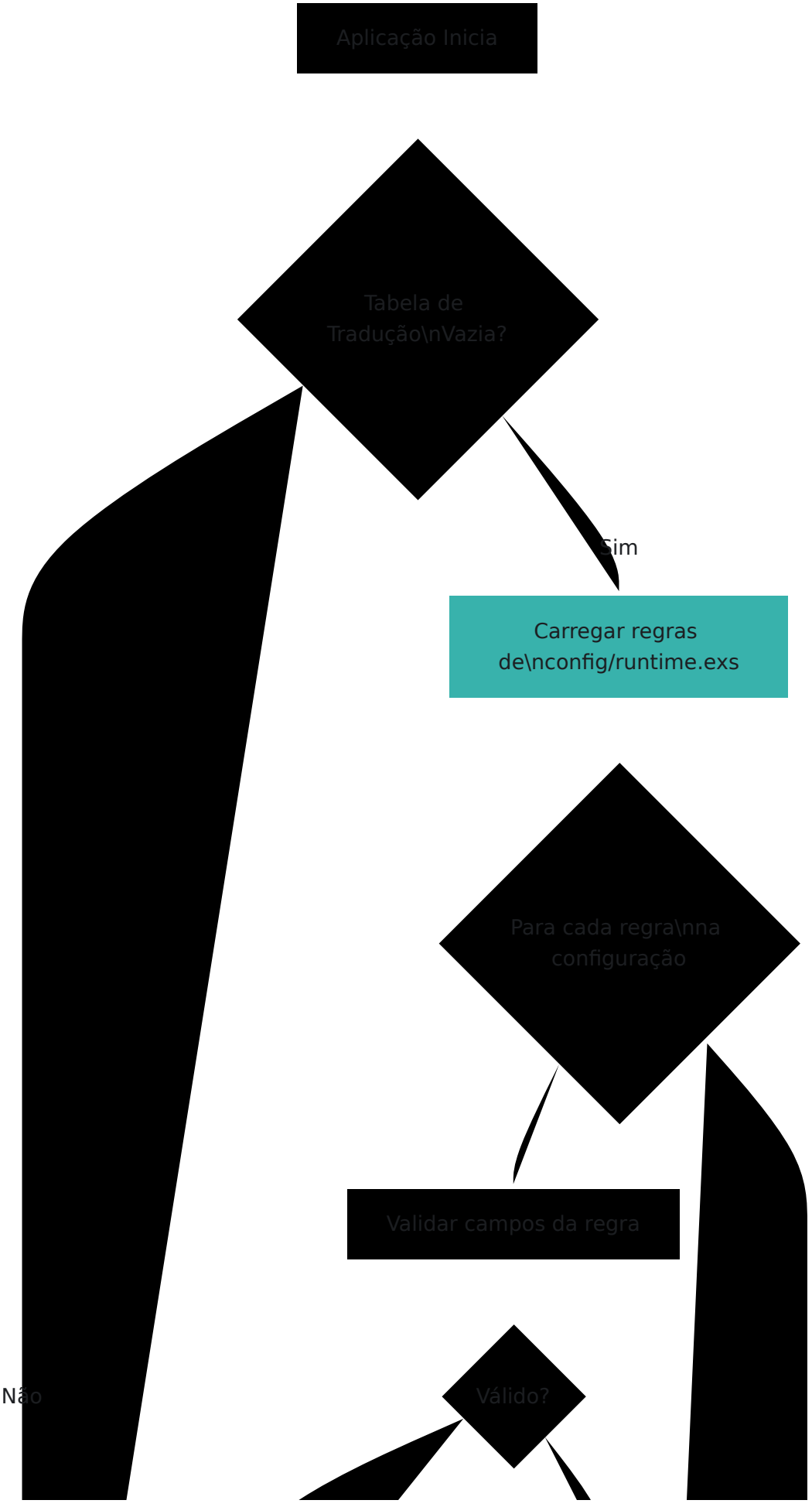
Configuração

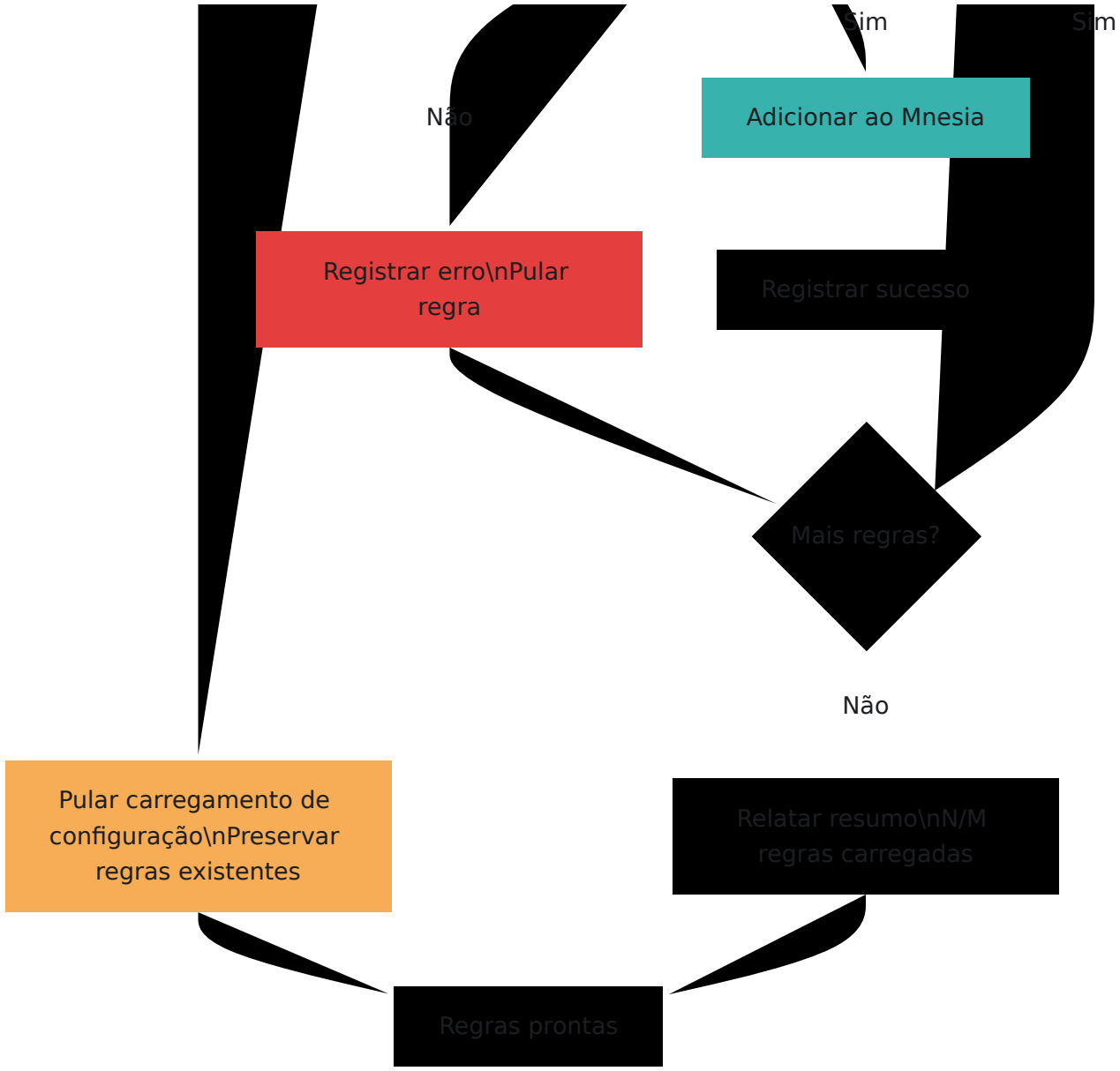
Carregando Regras a partir do Arquivo de Configuração

As regras de tradução podem ser definidas em `config/runtime.exs` e serão carregadas automaticamente na primeira inicialização.

Importante: As regras da configuração são carregadas apenas quando a tabela de tradução está **vazia** (primeira inicialização). Isso preserva regras adicionadas via a Interface Web durante o tempo de execução e previne duplicatas em reinicializações.

Fluxo de Carregamento de Configuração


Aplicação Inicia
Tabela de Tradução\nVazia?
Sim
Carregar regras de\nconfig/runtime.exs
Para cada regra\nna configuração
Validar campos da regra
Válido?
Não



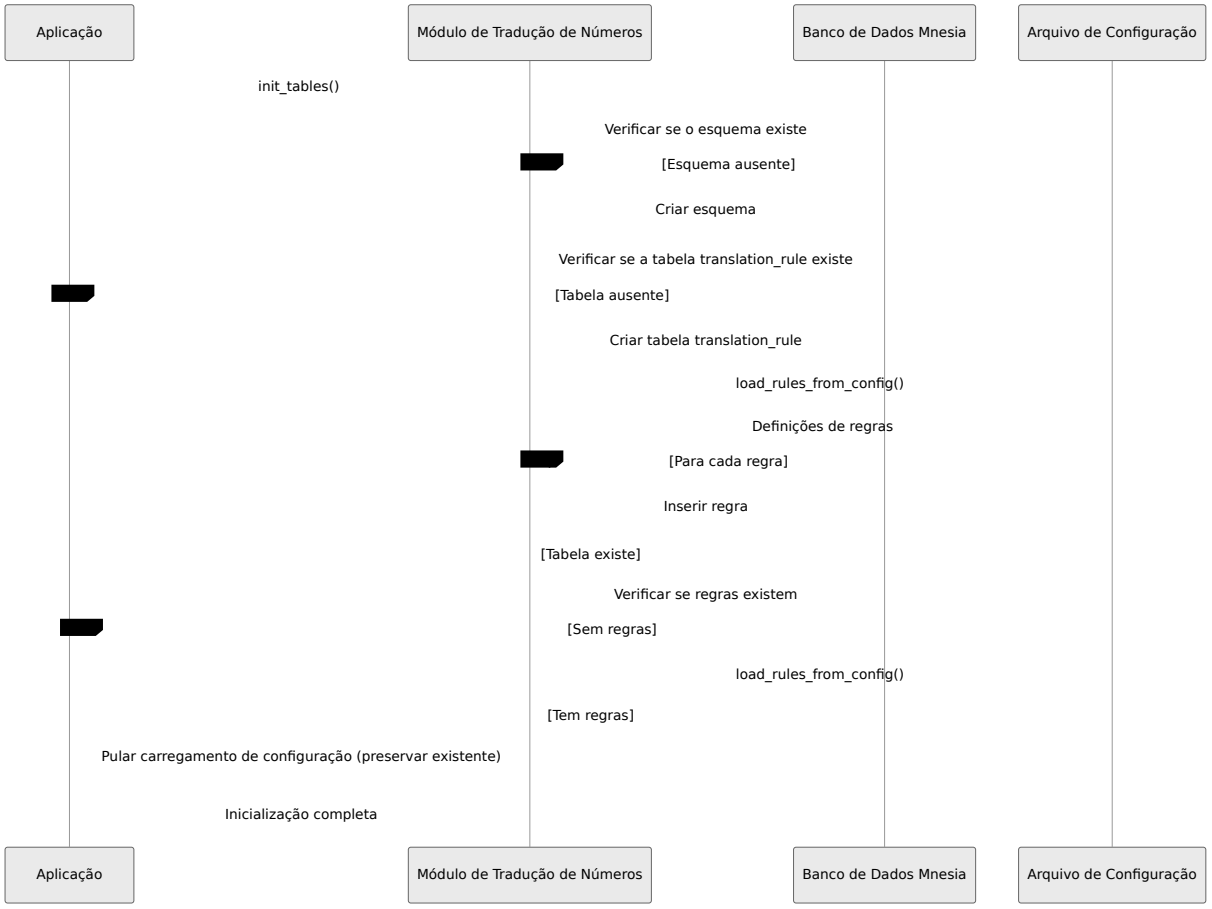
Exemplo de Configuração

```
# config/runtime.exs
config :sms_c, :translation_rules, [
  # Adicionar +1 a números de 10 dígitos dos EUA
  %{
    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: "Adicionar +1 a números de 10 dígitos de SMSC
doméstico",
    enabled: true,
    continue: false
  },

  # Remover zeros à frente do 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: "Converter prefixo internacional 00 para +",
    enabled: true,
    continue: true  # Continuar a aplicar mais formatação
  },

  # Forçar remetentes de códigos curtos para unknown/unknown
TON/NPI (apenas gate, sem reescrita)
  %{
    calling_match: "^\d{3,6}$",
    calling_replace: nil,
    source_ton: "unknown",
    source_npi: "unknown",
    priority: 6,
    description: "Códigos curtos (3-6 dígitos) => unknown/unknown
TON/NPI",
```

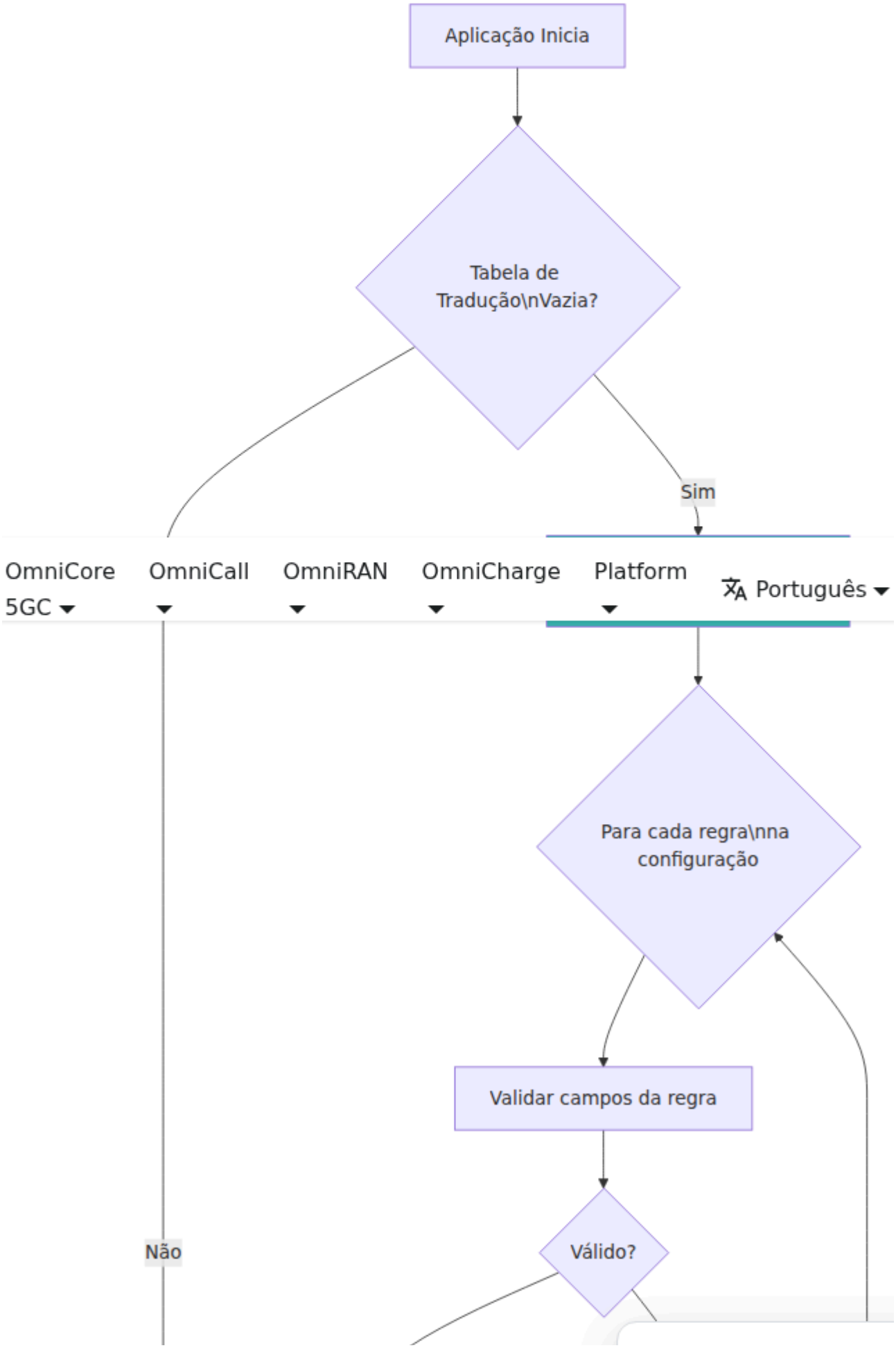
```
    enabled: true,
    continue: true
  },

  # Formatar números do Reino Unido para gateway específico
  %{
    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: "Formatar números do Reino Unido para gateway
legado",
    enabled: true,
    continue: false
  }
]
```

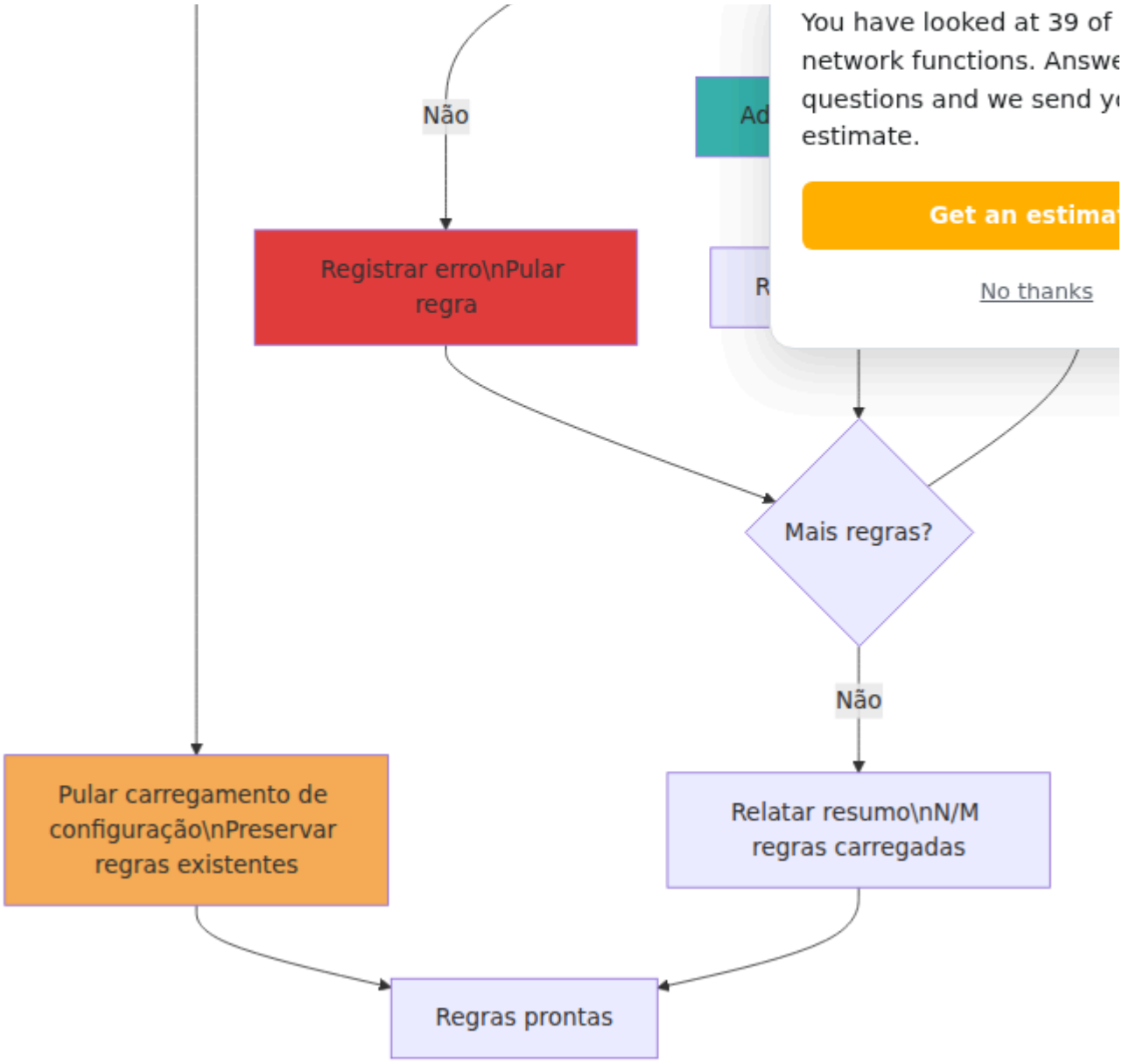
Começando

Fluxo de Inicialização



Fluxo de Tradução de Mensagens

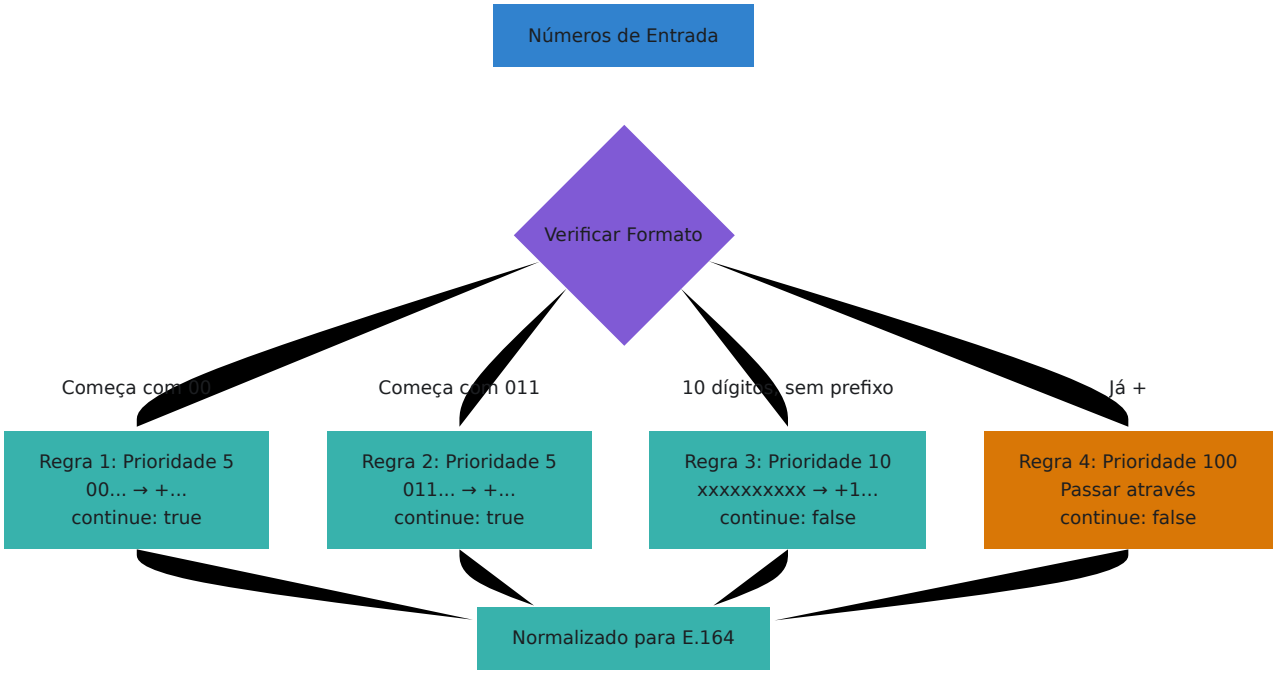

Aplicação Inicia
Tabela de Tradução\nVazia?
Sim
OmniCore 5GC
OmniCall
OmniRAN
OmniCharge
Platform
Português
Para cada regra\nna configuração
Validar campos da regra
Válido?
Não



Casos de Uso Comuns

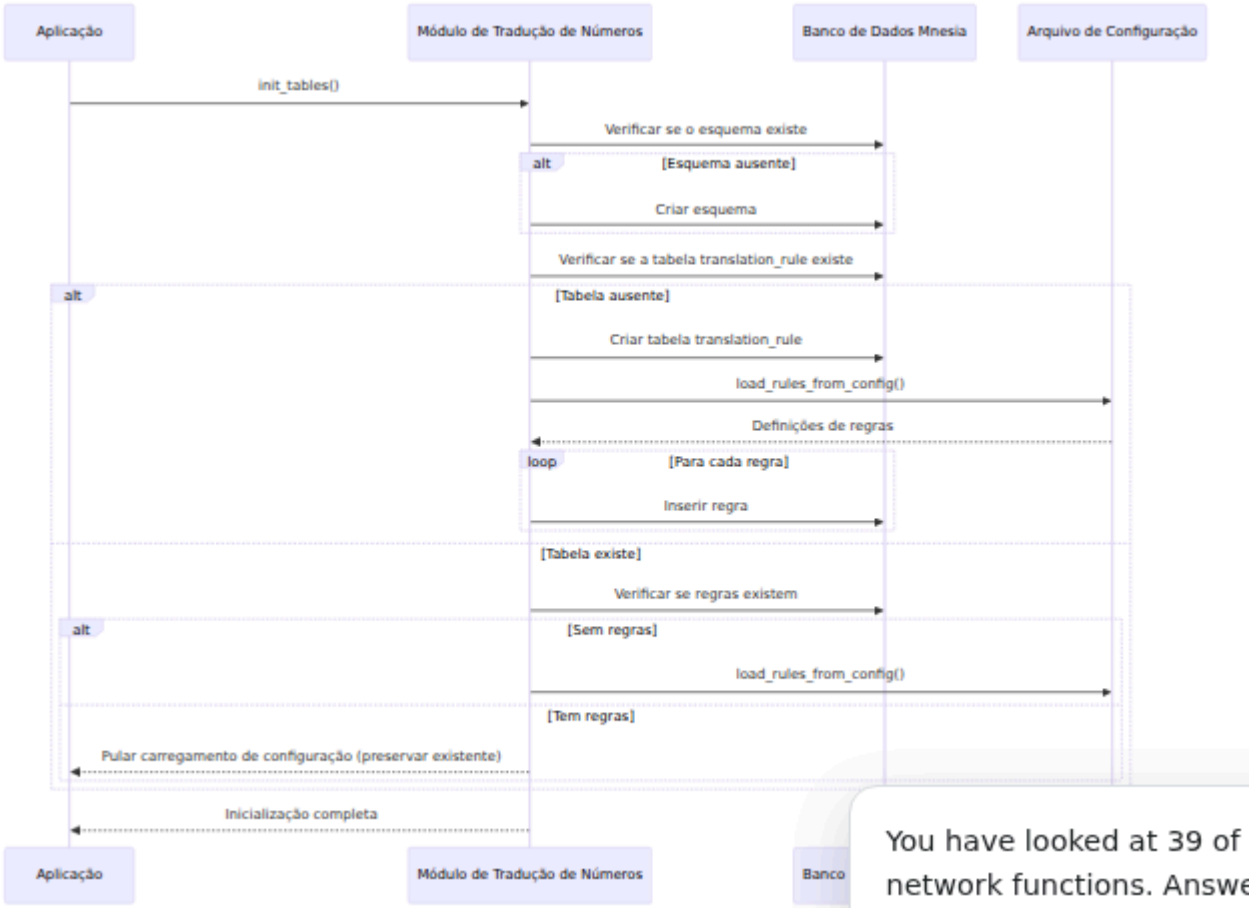
Normalização de Números Internacionais

Normalize vários formatos internacionais para E.164:



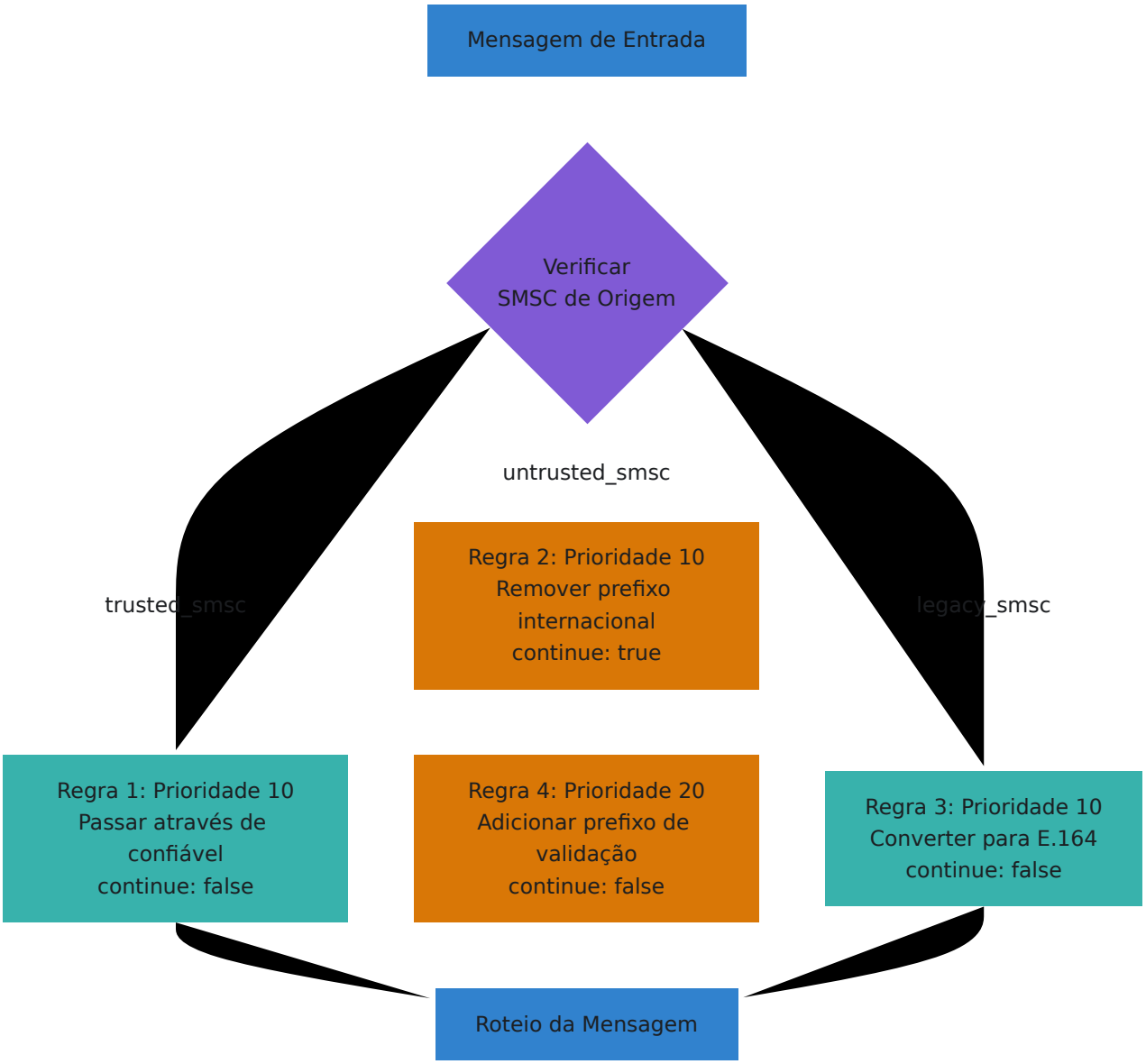
Formatação Específica de Gateway

Encadeie regras para formatar números para requisitos específicos do gateway:



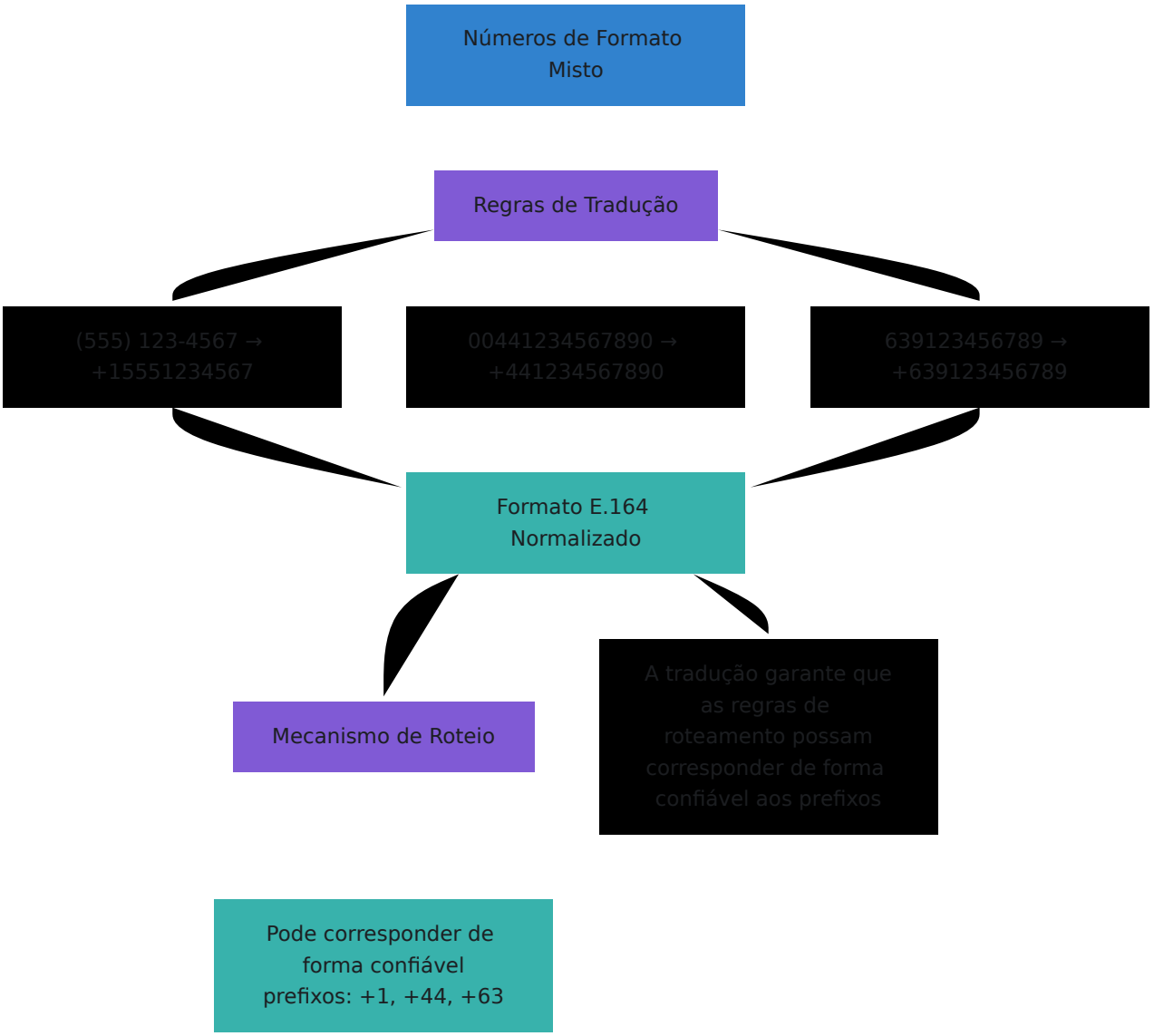
Traduções Específicas de SMSC

Aplique diferentes traduções com base na origem da mensagem:



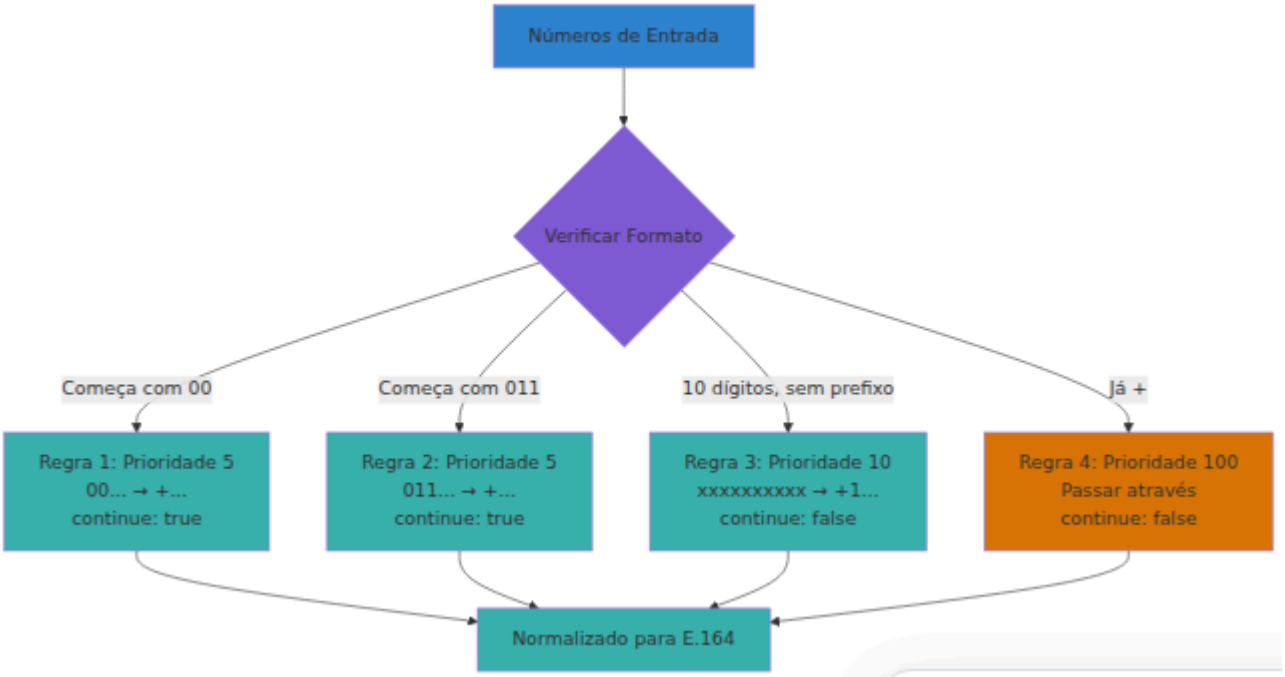
Preparação de Roteamento Baseada em Prefixo

Normalize números antes do roteamento para garantir correspondência de prefixo consistente:



Manipulação de Portabilidade de Números

Lide com números portados que exigem mudanças de prefixo:



Interface Web

UI de Gerenciamento de Regras de Tradução

Acesse a interface de gerenciamento de regras em `/number_translation` (via menu de navegação):

Recursos:

- Visualizar todas as regras em uma tabela ordenável por prioridade
- Adicionar novas regras com validação de formulário
- Editar regras existentes
- Habilitar/desabilitar regras sem excluir
- Excluir regras com confirmação
- Indicador visual para regras com `continue: true`
- Importar/Exportar regras como JSON

Adicionando uma Regra:

1. Preencha os critérios de correspondência (opcional):
    - Prefixo de chamada (ex: "+1", "44")

    - Prefixo chamado (ex: "+639", "1555")
    - SMSC de origem (deixe vazio para qualquer)
2. Defina as transformações (opcional):
    - Correspondência de regex e substituição para número de chamada
    - Correspondência de regex e substituição para número chamado
3. Defina a sobreposição do endereço de origem (opcional):
    - **Source TON / Source NPI** dropdowns - forçar o Tipo de Número do endereço originante / Indicador do Plano de Numeração (incorporado no TPDU). Controlado pelo regex de Correspondência de Número de Chamada.
    - Deixe um dropdown não definido para deixar o valor inferido. Veja Forçando Tipo de Número (TON) e Indicador do Plano de Numeração (NPI).
4. Defina a prioridade (1-255, menor = maior prioridade)
5. Defina o status:
    - **Habilitado**: Regra está ativa
    - **Continuar Processando**: Continuar avaliando mais regras após esta
6. Adicione uma descrição
7. Clique em "Adicionar Regra" ou "Atualizar Regra"

Alternância de Continuar Processando:

- **Parar** (padrão): Parar o processamento após esta regra corresponder
- **Continuar**: Aplicar esta regra e continuar avaliando as regras restantes
- Regras com continue habilitado mostram um distintivo azul "↓ Continuar" na tabela

Editando uma Regra:

1. Clique em "Editar" ao lado da regra
2. Modifique os campos conforme necessário
3. Clique em "Atualizar Regra"

Indicadores da Tabela de Regras:

- O distintivo **Habilitado/Desabilitado** mostra o status da regra
- O distintivo **↓ Continuar** mostra regras que continuarão o processamento

- O distintivo **Prioridade** mostra a ordem de avaliação
- Padrões regex exibidos em fonte monoespaçada para clareza

Simulador de Tradução

Acesse o simulador em `/translation_simulator` (via menu de navegação):

Recursos:

- Testar a lógica de tradução com números reais
- **Transformação Passo a Passo** mostrando cada regra aplicada
- Ver valores antes/depois para cada transformação
- Visualizar quais regras corresponderam e por quê
- Carregar cenários de exemplo para testes rápidos
- Visualizar histórico de testes (últimos 10 testes)

Usando o Simulador:

1. Insira os parâmetros de teste:
    - Número de chamada (de)
    - Número chamado (para)
    - SMSC de origem (opcional)
2. Clique em "Testar Tradução"
3. Visualize resultados abrangentes:
    - **Resultado da Tradução**: Números finais após todas as transformações
    - **Regras Aplicadas**: Contagem e lista de todas as regras que corresponderam
    - **Transformações Passo a Passo**: Visualização detalhada de cada regra:
        - Número do passo e informações da regra
        - Descrição da regra
        - Antes → Depois para números de chamada e chamados
        - Indicador "↓ Continuar" para regras que continuaram o processamento
        - Transformações destacadas em verde

- Valores inalterados marcados como "passaram através"

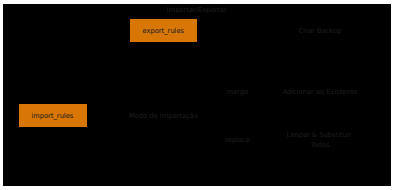
4. Carregue exemplos pré-configurados usando os botões de exemplo
5. Revise o histórico de testes para comparar diferentes cenários

Exemplo de Saída:

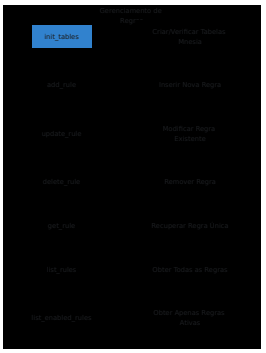
```
Resultado da Tradução
═══════════════════════════════════════════════
Número de Chamada:  5551234567 → +1-555-123-4567
Número Chamado:   5558720155 → +1-555-872-0155
✓ Traduzido por 3 regra(s)

Transformações Passo a Passo
═══════════════════════════════════════════════
┌─ Passo 1 ──────────────────────────────────┐
│ Regra #1 (Prioridade 10)         ↓ Continuar │
│ Adicionar código do país a números de 10 dígitos     │
│                                            │
│ Chamado: 5558720155 → +15558720155       │
└────────────────────────────────────────────┘

┌─ Passo 2 ──────────────────────────────────┐
│ Regra #2 (Prioridade 20)         ↓ Continuar │
│ Formatar código de área com traços             │
│                                           │
│ Chamado: +15558720155 → +1-555-8720155   │
└────────────────────────────────────────────┘

┌─ Passo 3 ──────────────────────────────────┐
│ Regra #3 (Prioridade 30)                     │
│ Formatação final para gateway              │
│                                            │
│ Chamado: +1-555-8720155 → +1-555-872-0155│
└────────────────────────────────────────────┘
```

Referência da API

Visão Geral das Operações Principais







Parâmetros de Tradução

translate_numbers aceita os seguintes parâmetros:

- `calling_number` (opcional): Número de telefone originante
- `called_number` (opcional): Número de telefone de destino
- `source_smsc` (opcional): Identificador do SMSC de origem
- `message_id` (opcional): Para registro de eventos

Retorna:

- `{:ok, translated_calling, translated_called, overrides, [rules_applied]}` - Sempre bem-sucedido
- `overrides` é um mapa `%{source_ton:, source_npi:}` (cada `nil` quando nenhuma regra correspondente definiu; última gravação ganha quando múltiplas regras definem um valor)
- Retorna números originais e sobreposições vazias se nenhuma regra corresponder
- Retorna lista de todas as regras que foram aplicadas (em ordem)

```
# Exemplo 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 se alguma tradução ocorreu
if rules != [] do
  Logger.info("Aplicadas #{length(rules)} regras de tradução")
  Enum.each(rules, fn rule ->
    Logger.info("  - Regra ##{rule.rule_id}: #{rule.description}")
  end)
end
```

Operações de Gerenciamento de Regras

```
# Adicionar uma nova regra
{: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: "Adicionar +1 a números de 10 dígitos",
  enabled: true,
  continue: false
})

# Atualizar uma regra
{:ok, updated_rule} = NumberTranslation.update_rule(rule_id, %{
  enabled: false,
  description: "Desabilitada para teste"
})

# Excluir uma regra
:ok = NumberTranslation.delete_rule(rule_id)

# Obter uma regra específica
rule = NumberTranslation.get_rule(rule_id)

# Listar todas as regras
all_rules = NumberTranslation.list_rules()

# Listar apenas regras habilitadas (ordenadas por prioridade)
enabled_rules = NumberTranslation.list_enabled_rules()
```

Operações de Importação/Exportação

```
# Exportar todas as regras
backup = NumberTranslation.export_rules()
# Retorna: %{
#   version: "1.0",
#   exported_at: ~U[2024-01-15 10:30:00Z],
#   count: 5,
#   rules: [...]
# }

# Salvar em arquivo JSON
json = Jason.encode!(backup, pretty: true)
File.write!("translation_rules_backup.json", json)

# Importar regras (mesclar com existentes)
{:ok, %{imported: 3, failed: 0}} =
  NumberTranslation.import_rules(backup, mode: :merge)

# Importar regras (substituir todas as existentes)
{:ok, %{imported: 5, failed: 0}} =
  NumberTranslation.import_rules(backup, mode: :replace)
```

Melhores Práticas

Design de Regras

1. **Mantenha prioridades organizadas**:
    - **1-10**: Regras de normalização críticas (adicionar códigos de país, corrigir formatos)
    - **11-50**: Formatação específica de gateway
    - **51-100**: Transformações opcionais
    - **101+**: Regras catch-all ou de depuração
2. **Use continue estrategicamente**:

- Habilite `continue: true` para regras de normalização que preparam números para processamento adicional
- Desative `continue: false` para regras de formatação final
- Evite longas cadeias (máximo de 3-4 regras) para manter o desempenho

3. **Documente suas regras**:
   - Sempre adicione descrições claras
   - Inclua exemplos na descrição (ex: "5551234567 → +15551234567")
   - Documente o propósito e a entrada/saída esperada
4. **Teste padrões regex**:
   - Teste padrões com o simulador antes de implantar
   - Use grupos de captura (\1, \2) para transformações flexíveis
   - Escape caracteres especiais de regex (pontos, parênteses, etc.)

Desempenho

1. **Minimize a contagem de regras**:
   - Combine regras semelhantes sempre que possível
   - Use correspondência de prefixo para reduzir avaliações de regex
   - Remova ou desative regras não utilizadas
2. **Otimize padrões regex**:
   - Use correspondência de prefixo primeiro (mais rápido que regex)
   - Mantenha padrões regex simples
   - Evite padrões pesados em retrocesso
3. **Limite o encadeamento de regras**:
   - Longas cadeias (5+ regras) podem impactar o desempenho
   - Considere combinar múltiplos passos em uma regra, se possível
   - Monitore a latência de tradução com métricas de Telemetria

Operações

1. **Teste antes de implantar**:
    - Use o simulador com exemplos do mundo real
    - Teste casos extremos (números vazios, caracteres especiais)
    - Verifique o comportamento da flag continue
2. **Faça backup regularmente**:
    - Exporte regras antes de fazer alterações significativas
    - Controle de versão suas exportações
    - Teste importações em não-produção primeiro
3. **Monitore traduções**:
    - Habilite o registro de message_id para depuração
    - Verifique logs de eventos para decisões de tradução
    - Monitore quais regras estão sendo aplicadas
4. **Implantação gradual**:
    - Adicione novas regras como desabilitadas primeiro
    - Teste com o simulador
    - Habilite e monitore
    - Ajuste conforme necessário

Dicas de Regex

1. **Padrões comuns**:
    - Número de 10 dígitos dos EUA: `^(\d{10})$`
    - Formato internacional: `^\+(\d+)$`
    - Remover zeros à frente: `^0+(.+)$`
    - Adicionar traços: `^(\d{3})(\d{3})(\d{4})$` → `\1-\2-\3`
2. **Grupos de captura**:

- Use parênteses para capturar: `^(\d{3})(\d{7})$`
- Referência na substituição: `+1\1\2`
- Múltiplas capturas: `^\+(\d{1,3})(\d+)$` → `00\1\2`

3. **Escape caracteres especiais**:
   - Ponto literal: `\.`
   - Mais literal: `\+`
   - Parênteses literais: `\(` ou `\)`

Resolução de Problemas

Regra Não Correspondendo

Sintoma: Regra esperada não corresponde, números passam inalterados

Causas possíveis:

- Prefixo não corresponde (verifique a correspondência exata do prefixo)
- SMSC de origem não corresponde
- Padrão regex não corresponde ao formato de entrada
- Regra está desabilitada
- Regra de prioridade mais alta correspondeu primeiro (com continue: false)

Soluções:

1. Use o simulador para ver quais regras são avaliadas
2. Verifique o status da regra (habilitada/desabilitada)
3. Verifique a correspondência de prefixo (sensível a maiúsculas)
4. Teste o padrão regex separadamente
5. Verifique a ordem de prioridade

Transformação Errada Aplicada

Sintoma: Número transformado, mas resultado está incorreto

Causas possíveis:

- Padrão regex corresponde, mas padrão de substituição está errado
- Múltiplas regras aplicando em ordem inesperada
- Referências de grupos de captura incorretas (\1, \2, etc.)

Soluções:

1. Use o simulador para ver as transformações passo a passo
2. Verifique se o padrão regex captura os grupos corretos
3. Verifique a sintaxe do padrão de substituição
4. Teste regex em um testador de regex online
5. Revise a prioridade da regra e as flags de continue

Loop Infinito / Degradação de Desempenho

Sintoma: Tradução leva muito tempo ou parece travar

Nota: Isso não deve acontecer devido à prevenção de loops, mas se acontecer:

Causas possíveis:

- Bug na lógica de prevenção de loops
- Avaliação de regex extremamente longa
- Cadeia de regras muito longa

Soluções:

1. Verifique os logs da aplicação em busca de erros
2. Revise regras com continue: true
3. Simplifique padrões regex
4. Reduza o número de regras encadeadas
5. Relate um bug se a prevenção de loops falhar

Encadeamento de Regras Inesperado

Sintoma: Mais regras aplicadas do que o esperado

Causas possíveis:

- Regras têm continue: true quando não deveriam
- A ordenação de prioridade permite múltiplas correspondências
- Número transformado corresponde a regras adicionais

Soluções:

1. Use o simulador para ver a cadeia exata de regras
2. Revise as flags de continue em todas as regras
3. Ajuste prioridades para controlar a ordem
4. Defina continue: false na regra final

Tradução Não Aplicada Antes do Roteamento

Sintoma: Roteador vê números não traduzidos

Causas possíveis:

- Tradução não integrada no fluxo da mensagem
- Tradução acontecendo após o roteamento
- Código da aplicação ignorando a tradução

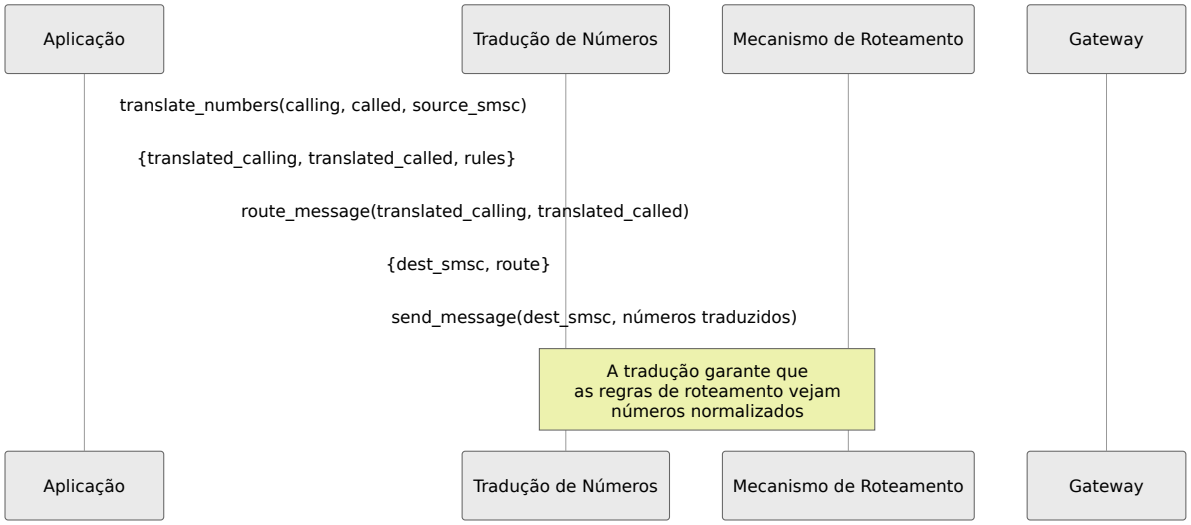
Soluções:

1. Verifique a integração da aplicação: a tradução deve ser chamada antes do roteamento
2. Verifique o pipeline de processamento de mensagens
3. Revise logs de eventos para eventos de tradução
4. Certifique-se de que translate_numbers é chamado na ordem correta

Tópicos Avançados

Integração com Roteamento

A tradução acontece **antes** do roteamento para garantir formatos de número consistentes:



Registro de Eventos

Decisões de tradução são registradas via o EventLogger:

- `translation_started`: A tradução começa
- `translation_candidates`: Número de regras habilitadas
- `translation_matched`: Regra correspondente e aplicada
- `translation_calling`: Número de chamada transformado
- `translation_called`: Número chamado transformado
- `translation_continue`: Regra tem continue=true, continuando avaliação
- `translation_none`: Nenhuma regra correspondente

Habilite o registro passando `message_id` para `translate_numbers/1`.

Métricas de Telemetria

Monitore o desempenho da tradução com Telemetria:

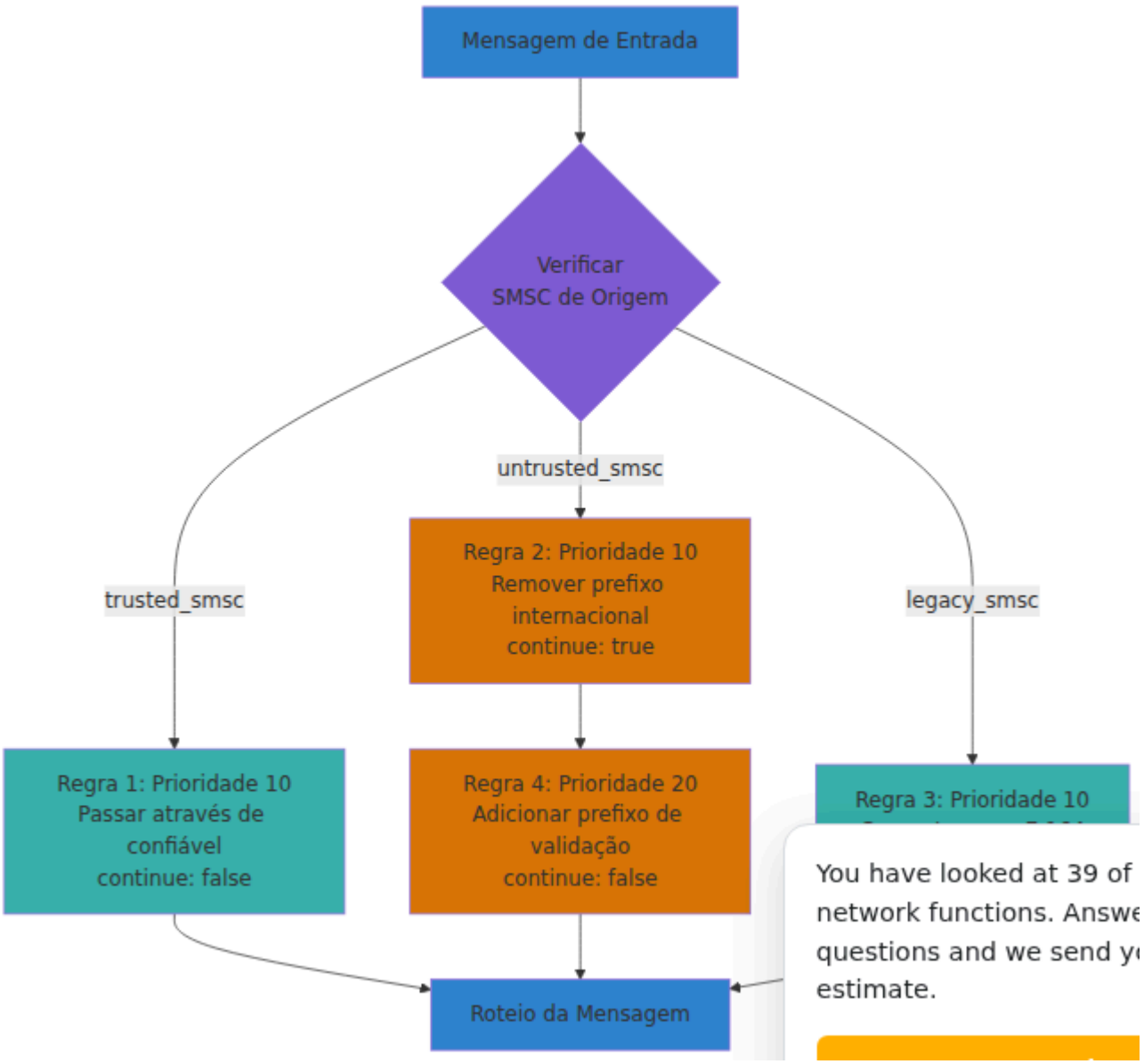
```
:telemetry.attach(
  "number-translation-handler",
  [:sms_c, :number_translation, :translate, :stop],
  fn _event_name, measurements, metadata, _config ->
    # measurements: %{duration: microsegundos}
    # metadata: %{rules_applied: contagem, ...}
  end,
  nil
)
```

Métricas-chave para monitorar:

- Duração da tradução (p50, p95, p99)
- Regras aplicadas por mensagem
- Regras correspondidas vs não correspondidas
- Uso da flag continue

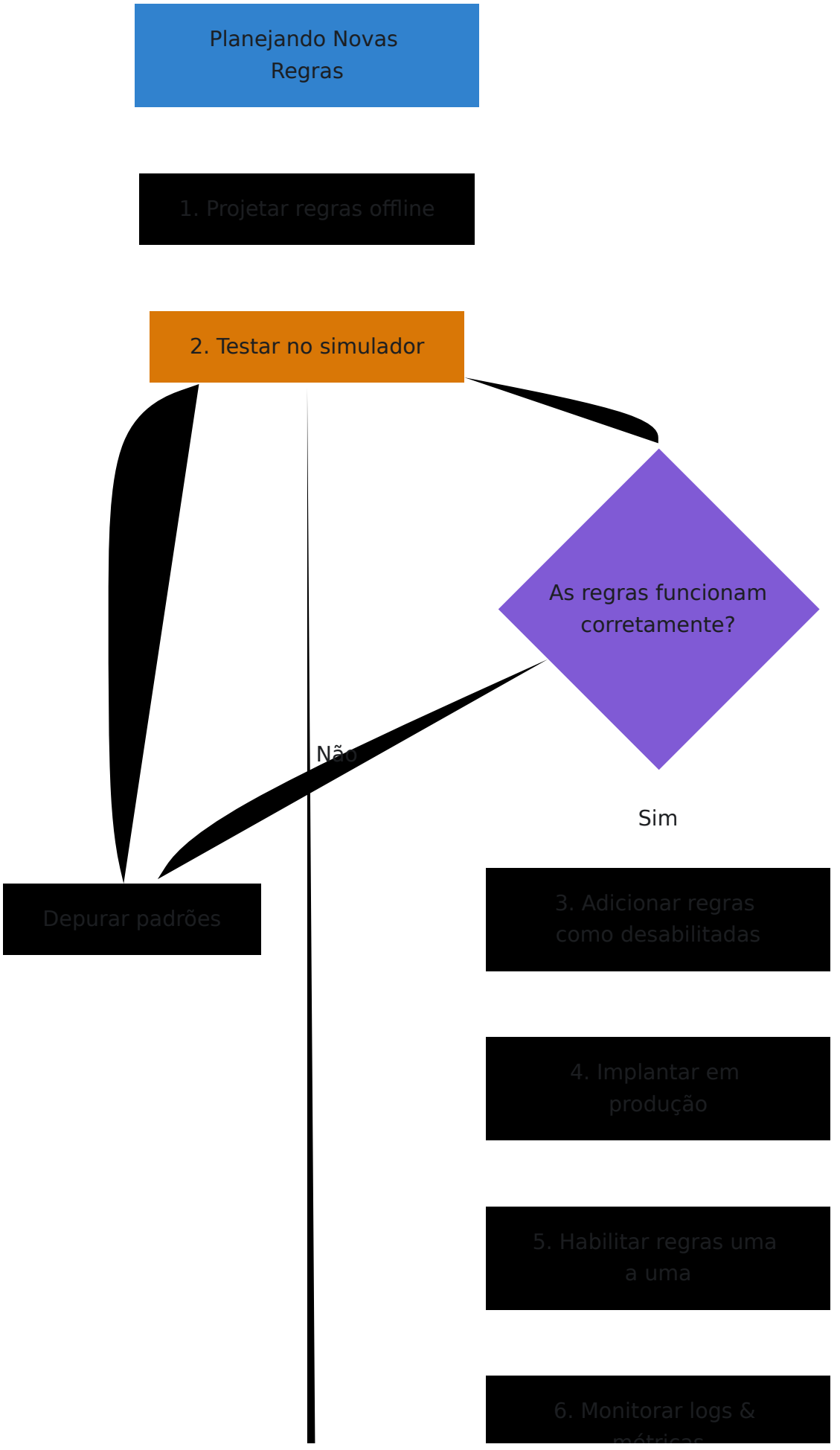
Clustering

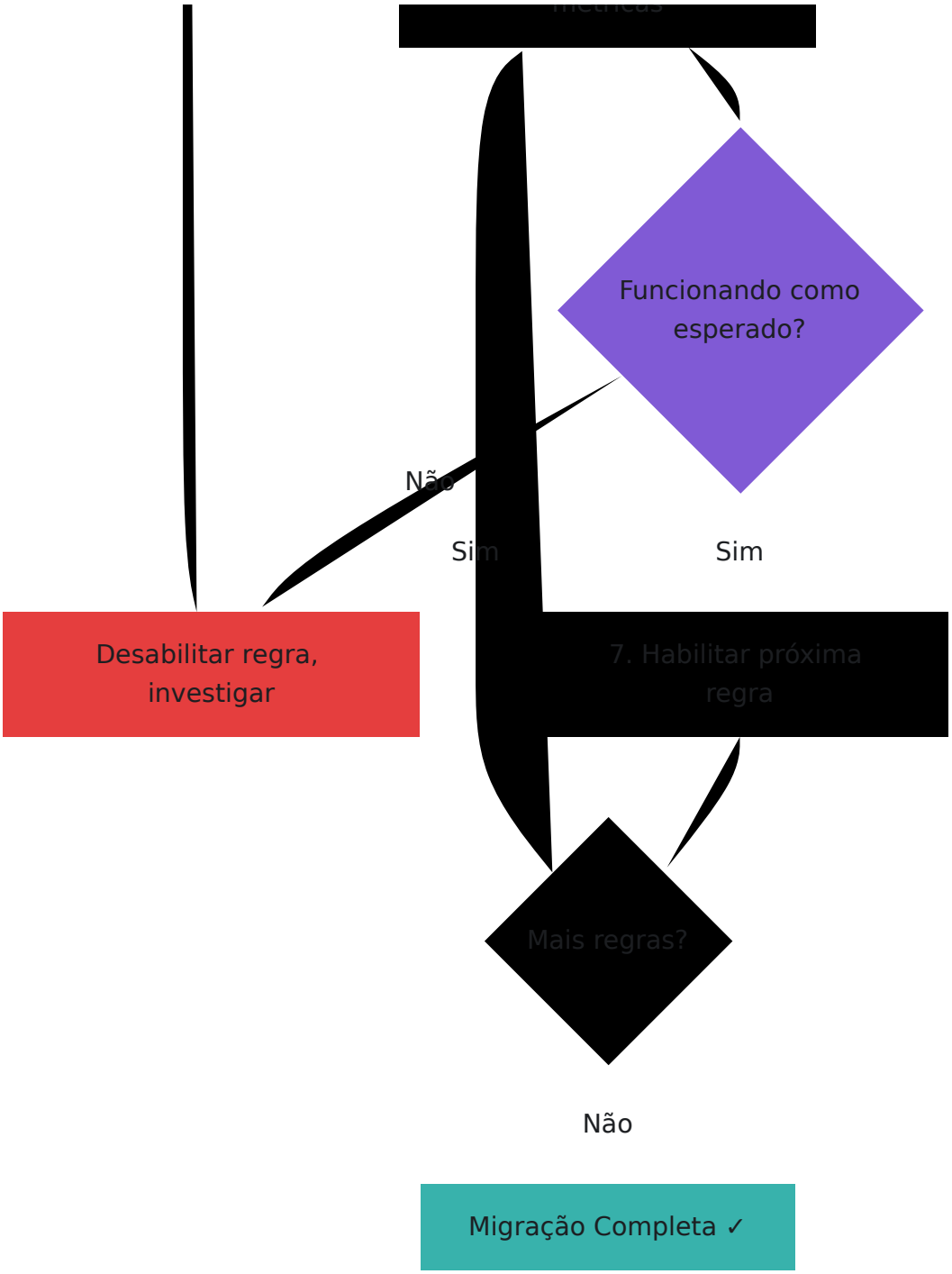
As tabelas Mnesia são automaticamente distribuídas entre nós em cluster. As regras de tradução são replicadas para alta disponibilidade.



Estratégias de Migração

Ao implantar novas regras de tradução:


Planejando Novas Regras
1. Projetar regras offline
2. Testar no simulador
As regras funcionam corretamente?
Não
Sim
Depurar padrões
3. Adicionar regras como desabilitadas
4. Implantar em produção
5. Habilitar regras uma a uma
6. Monitorar logs &



Exemplos

Exemplo 1: Normalização de Números dos EUA

Requisito: Converter vários formatos de números dos EUA para E.164 (+1XXXXXXXXXX)

```
# Regra 1: Números de 10 dígitos (prioridade mais alta)
%{
  calling_match: "^(\d{10})$",
  calling_replace: "+1\1",
  called_match: "^(\d{10})$",
  called_replace: "+1\1",
  priority: 5,
  description: "Adicionar +1 a números de 10 dígitos nu",
  enabled: true,
  continue: false
}

# Regra 2: 1 + 10 dígitos (prioridade média)
%{
  calling_match: "^1(\d{10})$",
  calling_replace: "+1\1",
  called_match: "^1(\d{10})$",
  called_replace: "+1\1",
  priority: 10,
  description: "Converter 1XXXXXXXXXX para +1XXXXXXXXXX",
  enabled: true,
  continue: false
}

# Casos de teste:
# "5551234567" → "+15551234567" (Regra 1)
# "15551234567" → "+15551234567" (Regra 2)
# "+15551234567" → "+15551234567" (Sem correspondência, passar
através)
```

Exemplo 2: Conversão de Prefixo Internacional com Encadeamento

Requisito: Converter prefixo 00 para +, depois formatar para gateway

```
# Regra 1: Converter 00 para + (continua para a próxima regra)
%{
  calling_match: "^00(.+)$",
  calling_replace: "+\1",
  called_match: "^00(.+)$",
  called_replace: "+\1",
  priority: 5,
  description: "Converter prefixo internacional 00 para +",
  enabled: true,
  continue: true  # Continuar a formatar
}

# Regra 2: Formatar para gateway (para)
%{
  calling_match: "^\+(\d+)$",
  calling_replace: "00\1",
  called_match: "^\+(\d+)$",
  called_replace: "00\1",
  priority: 10,
  description: "Formatar números + como 00 para gateway",
  enabled: true,
  continue: false  # Parar após isso
}

# Caso de teste:
# Passo 1: "00441234567890" → "+441234567890" (Regra 1, continue)
# Passo 2: "+441234567890" → "00441234567890" (Regra 2, parar)
# Resultado: "00441234567890"
# Regras aplicadas: [Regra 1, Regra 2]
```

Exemplo 3: Manipulação Específica de SMSC

Requisito: Aplicar regras diferentes com base no SMSC de origem

```
# Regra 1: SMSC Confiável - passar através (prioridade 5)
%{
  source_smsc: "trusted_gateway",
  calling_match: nil,  # Sem transformação
  calling_replace: nil,
  called_match: nil,
  called_replace: nil,
  priority: 5,
  description: "Passar números de gateway confiável",
  enabled: true,
  continue: false
}

# Regra 2: SMSC Não Confiável - normalizar (prioridade 10)
%{
  source_smsc: "untrusted_gateway",
  calling_match: "^(.*)$",
  calling_replace: "+VALIDATE\1",
  called_match: "^(.*)$",
  called_replace: "+VALIDATE\1",
  priority: 10,
  description: "Adicionar prefixo de validação para origem não
confiável",
  enabled: true,
  continue: false
}

# Regra 3: Catch-all para outros SMSCs (prioridade 100)
%{
  source_smsc: nil,  # Curiga
  calling_match: "^(\d{10})$",
  calling_replace: "+1\1",
  called_match: "^(\d{10})$",
  called_replace: "+1\1",
  priority: 100,
  description: "Padrão: Adicionar +1 a números de 10 dígitos",
  enabled: true,
  continue: false
}
```

Exemplo 4: Cadeia de Formatação em Múltiplos Passos

Requisito: Normalizar → Adicionar código do país → Formatar com traços

```
# Regra 1: Remover zeros à frente (continuar)
%{
  calling_match: "^0+(.+)$",
  calling_replace: "\1",
  called_match: "^0+(.+)$",
  called_replace: "\1",
  priority: 5,
  description: "Remover zeros à frente",
  enabled: true,
  continue: true
}

# Regra 2: Adicionar código do país se ausente (continuar)
%{
  calling_match: "^(\d{10})$",
  calling_replace: "+1\1",
  called_match: "^(\d{10})$",
  called_replace: "+1\1",
  priority: 10,
  description: "Adicionar +1 a números de 10 dígitos",
  enabled: true,
  continue: true
}

# Regra 3: Formatar com traços (parar)
%{
  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: "Formatar como +1-XXX-XXX-XXXX",
  enabled: true,
  continue: false
}

# Caso de teste:
# Entrada: "005551234567"
# Passo 1: "005551234567" → "5551234567" (Regra 1, continue)
# Passo 2: "5551234567" → "+15551234567" (Regra 2, continue)
# Passo 3: "+15551234567" → "+1-555-123-4567" (Regra 3, parar)
```

```
# Resultado: "+1-555-123-4567"
# Regras aplicadas: [Regra 1, Regra 2, Regra 3]
```

Suporte

Para problemas ou perguntas:

- Use o simulador para trabalhar através de exemplos e depurar a lógica de tradução
- Revise logs de eventos para decisões de tradução
- Verifique o conteúdo da tabela Mnesia: `:mnesia.table_info(:translation_rule, :size)`
- Monitore métricas de Telemetria para problemas de desempenho

Guia de Operações SMS-C

← Voltar para o Índice da Documentação

Procedimentos operacionais diários, monitoramento e tarefas de manutenção para as equipes de operações SMS-C.

Índice

- Operações Diárias
- Monitoramento
- Rastreamento de Mensagens
- Gerenciamento de Rotas
- Gerenciamento de Frontend
- Gerenciamento de Tradução de Números
- Manutenção do Sistema
- Backup e Recuperação
- Planejamento de Capacidade
- Resposta a Incidentes

Operações Diárias

Verificação de Saúde Matinal

Realize essas verificações no início de cada dia:

1. Verificar Status do Sistema

```
# Verificação de saúde da API
curl https://api.example.com:8443/api/status

# Resposta esperada:
# {"status":"ok","application":"OmniMessage","timestamp":"2025-10-30T08:00:00Z"}
```

2. Revisar Métricas do Prometheus

Acesse o painel do Prometheus e verifique:

- Taxa de mensagens processadas (últimas 24 horas)
- Taxa de falhas de roteamento (deve ser < 1%)
- Pendências na fila (deve ser < 1000 pendentes)
- Taxa de sucesso na entrega (deve ser > 95%)
- Status de conexão do frontend (todos os frontends esperados ativos)

3. Verificar Fila de Mensagens

Acesse a interface da Web: `https://sms-admin.example.com/message_queue`

Revise:

- Total de mensagens pendentes (deve ser baixo)
- Idade da mensagem mais antiga (deve ser < 5 minutos)
- Mensagens com altas tentativas de entrega (investigar se > 3)
- Mensagens de carta morta (investigar qualquer presente)

4. Revisar Status do Frontend

Acesse a interface da Web: `https://sms-admin.example.com/frontend_status`

Verifique:

- Todos os frontends esperados estão ativos
- Nenhuma desconexão não expirada
- Nenhum erro de frontend nas últimas 24 horas

5. Verificar Logs da Aplicação

Acesse a interface da Web: `https://sms-admin.example.com/logs` ou verifique os arquivos de log

Procure por:

- Mensagens de nível de erro
- Falhas de roteamento
- Falhas de cobrança
- Problemas de conexão com o banco de dados
- Problemas com nós do cluster

Monitoramento do Volume de Mensagens

Verificar Contagens de Mensagens Horárias:

Use a consulta do Prometheus:

```
# Mensagens recebidas por hora
increase(sms_c_message_received_count[1h])

# Mensagens entregues por hora
increase(sms_c_delivery_succeeded_count[1h])

# Calcular taxa de entrega
rate(sms_c_delivery_succeeded_count[1h]) /
rate(sms_c_message_received_count[1h])
```

Padrões Esperados:

- Horário comercial: Volume mais alto
- Noites/fins de semana: Volume mais baixo
- Taxa de entrega: Deve ser > 95%

Condições de Alerta:

- Queda repentina nas mensagens (> 50% de diminuição)

- Aumento repentino nas mensagens (> 200% de aumento)
- Taxa de entrega abaixo de 90%

Monitoramento

Métricas Chave a Observar

Métricas de Processamento de Mensagens

Contagem de Mensagens Recebidas (`sms_c_message_received_count`):

- **O que**: Total de mensagens entrando no sistema
- **Alerta**: Queda ou pico repentino
- **Consulta**: `rate(sms_c_message_received_count[5m])`

Duração do Processamento de Mensagens (`sms_c_message_processing_stop_duration`):

- **O que**: Tempo de processamento de ponta a ponta
- **Alerta**: p95 > 1000ms
- **Consulta**: `histogram_quantile(0.95, sms_c_message_processing_stop_duration)`

Métricas de Roteamento

Falhas de Roteamento (`sms_c_routing_failed_count`):

- **O que**: Mensagens que o sistema não conseguiu rotear
- **Alerta**: Qualquer falha (> 0)
- **Consulta**: `increase(sms_c_routing_failed_count[5m])`

Rota Correspondida (`sms_c_routing_route_matched_count`):

- **O que**: Quais rotas estão sendo usadas
- **Alerta**: Rotas de alta prioridade não correspondendo
- **Consulta**: `sms_c_routing_route_matched_count`

Métricas de Entrega

Taxa de Sucesso na Entrega:

- **O que**: Percentual de entregas bem-sucedidas
- **Alerta**: Taxa < 95%
- **Consulta**: `rate(sms_c_delivery_succeeded_count[5m]) / rate(sms_c_delivery_queued_count[5m])`

Tentativas de Entrega (`sms_c_delivery_succeeded_attempt_count`):

- **O que**: Retentativas necessárias para entrega
- **Alerta**: p95 > 2 (muitas retentativas)
- **Consulta**: `histogram_quantile(0.95, sms_c_delivery_succeeded_attempt_count)`

Métricas de Fila

Tamanho da Fila (`sms_c_queue_size_size`):

- **O que**: Total de mensagens na fila
- **Alerta**: Tamanho > 10.000
- **Consulta**: `sms_c_queue_size_size`

Idade da Mensagem Mais Antiga (`sms_c_queue_oldest_message_age_seconds`):

- **O que**: Idade da mensagem pendente mais antiga
- **Alerta**: Idade > 300 segundos
- **Consulta**: `sms_c_queue_oldest_message_age_seconds`

Configuração do Painel

Painéis do Painel Operacional:

1. **Taxa de Mensagens** (Gráfico)
    - Mensagens recebidas (taxa de 5 minutos)
    - Mensagens entregues (taxa de 5 minutos)

- Intervalo de tempo: Últimas 24 horas

2. **Status da Fila** (Estatísticas Únicas)
   - Mensagens pendentes atuais
   - Idade da mensagem mais antiga
   - Contagem de mensagens falhadas
3. **Desempenho de Entrega** (Gráfico)
   - Taxa de sucesso ao longo do tempo
   - Taxa de falha ao longo do tempo
   - Intervalo de tempo: Últimas 24 horas
4. **Status de Roteamento** (Tabela)
   - ID da Rota
   - Contagem de correspondências (última hora)
   - SMSC de Destino
   - Prioridade
5. **Status do Frontend** (Tabela)
   - Nome do Frontend
   - Status (ativo/expirado)
   - Última vez visto
   - Contagem de mensagens (última hora)
6. **Saúde do Sistema** (Estatísticas Únicas)
   - Tempo de resposta da API (p95)
   - Tempo de consulta ao banco de dados (p95)
   - Tempo de pesquisa ENUM (p95)

Configuração de Alertas

Alertas Críticos (Resposta Imediata Necessária):

```yaml
# Nenhuma rota encontrada - mensagens não podem ser entregues
- alert: RoutingFailures
  expr: increase(sms_c_routing_failed_count[5m]) > 0
  severity: critical
  description: "{{ $value }} mensagens falharam no roteamento nos
últimos 5 minutos"

# Fila acumulando - processamento atrasando
- alert: QueueBacklog
  expr: sms_c_queue_size_pending > 10000
  severity: critical
  description: "A fila tem {{ $value }} mensagens pendentes"

# Mensagens envelhecendo - entrega presa
- alert: OldMessagesInQueue
  expr: sms_c_queue_oldest_message_age_seconds > 300
  severity: critical
  description: "A mensagem mais antiga tem {{ $value }} segundos"

# Frontend desconectado - sem caminho de entrega
- alert: FrontendDisconnected
  expr: sms_c_frontend_status_count{status="disconnected"} > 0
  severity: critical
  description: "{{ $value }} frontends desconectados"
```

Alertas de Aviso (Investigação Necessária):

```
# Taxa de sucesso na entrega caindo
- alert: LowDeliveryRate
  expr: rate(sms_c_delivery_succeeded_count[10m]) /
rate(sms_c_delivery_queued_count[10m]) < 0.90
  severity: warning
  description: "Taxa de sucesso na entrega é {{ $value }}"

# Muitas retentativas de entrega
- alert: HighRetryRate
  expr: histogram_quantile(0.95,
sms_c_delivery_succeeded_attempt_count) > 2
  severity: warning
  description: "Tentativas de entrega no 95º percentil: {{ $value
}}"

# Pesquisas ENUM lentas ou falhando
- alert: SlowEnumLookups
  expr: histogram_quantile(0.95, sms_c_enum_lookup_stop_duration)
> 5000
  severity: warning
  description: "Pesquisas ENUM levando > 5 segundos"

# Baixa taxa de acerto de cache 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: "Taxa de acerto de cache ENUM: {{ $value }}"
```

Rastreamento de Mensagens

Encontrar Mensagem Específica

Por ID da Mensagem:

1. **Interface da Web**: Navegue até `/message_queue`
2. Insira o ID da mensagem na caixa de pesquisa
3. Veja todos os detalhes e o histórico de eventos

Via API:

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

Por Número de Telefone:

1. **Interface da Web**: Navegue até `/message_queue`
2. Insira o número de telefone na caixa de pesquisa
3. Veja todas as mensagens para esse número

Rastrear Ciclo de Vida da Mensagem

Ver Histórico de Eventos:

1. **Interface da Web**: Clique na mensagem na fila, veja a seção "Eventos"
2. **API**: `GET /api/events/12345`

Sequência Comum de Eventos:

```
1. message_inserted - Mensagem criada
   ↓
2. number_translated - Números normalizados (se configurado)
   ↓
3. message_routed - Decisão de roteamento tomada
   ↓
4. charging_attempted - Verificação de cobrança (se habilitada)
   ↓
5. message_delivered - Entregue com sucesso
```

Sequência de Entrega Falhada:

```
1. message_inserted
   ↓
2. message_routed
   ↓
3. delivery_attempt_1 - Primeira tentativa falhou
   ↓
4. delivery_attempt_2 - Segunda tentativa falhou (2min de atraso)
   ↓
5. delivery_attempt_3 - Terceira tentativa falhou (4min de atraso)
   ↓
6. message_dead_letter - Limite de retentativas excedido
```

Verificar Status de Entrega

Aguardando Entrega:

- Status: "queued" (pronto para enviar agora) ou "backoff" (aguardando uma retentativa)
- deliver_after: nil/passado para `queued`, um timestamp futuro para `backoff`
- delivery_attempts: 0 para uma mensagem `queued` nova, maior para `backoff`

Mensagens Entregues:

- Status: "delivered" (entrega confirmada) ou "sent" (enviada, sem DLR ainda)
- deliver_time: Timestamp da entrega
- dest_smsc: Frontend que entregou

Mensagens Não Entregues / Falhadas:

- Status: "backoff" com altas delivery_attempts - ainda tentando (limitado a um intervalo de 30 minutos, até que o `expires` da mensagem seja alcançado)
- Status: "expired", "dropped", "balance_rejected" ou "auto_replied" - terminal
- Verifique o log de eventos para razões de falha

Roteamento de Mensagens Baseado em Localização

O SMS-C suporta recuperação de mensagens baseada em localização, permitindo que os frontends recebam automaticamente mensagens destinadas a assinantes registrados em sua localização.

Como Funciona:

Quando um frontend consulta mensagens pendentes usando `get_messages_for_smsc(smsc_name)`, o sistema retorna mensagens de duas maneiras:

1. **Roteamento Explícito** - Mensagens onde `dest_smsc` corresponde explicitamente ao nome do frontend
2. **Roteamento Baseado em Localização** - Mensagens onde:
   - `dest_smsc` é `null` (não roteado explicitamente)
   - `destination_msisdn` tem um registro de localização ativo
   - O campo `location` da localização corresponde ao nome do frontend
   - A localização não expirou

Cenário de Exemplo:

Um assinante com MSISDN `+447700900123` se registra no frontend `uk_gateway`:

```
# Assinante se registra (cria registro de localização)
POST /api/locations
{
  "msisdn": "+447700900123",
  "imsi": "234150123456789",
  "location": "uk_gateway",
  "expires": "2025-11-01T12:00:00Z"
}
```

Quando uma mensagem chega para este assinante sem roteamento explícito:

```
# Mensagem submetida sem dest_smsc
POST /api/messages
{
  "source_msisdn": "+15551234567",
  "destination_msisdn": "+447700900123",
  "message_body": "Hello",
  "source_smsc": "api"
  # Nota: dest_smsc é null
}
```

O frontend `uk_gateway` receberá automaticamente esta mensagem quando fizer a consulta:

```
# Frontend consulta mensagens
GET /api/messages/queue?smsc=uk_gateway

# Retorna a mensagem mesmo que dest_smsc seja null
# porque o assinante de destino está registrado no uk_gateway
```

Requisitos de Localização:

Para que o roteamento baseado em localização funcione:

- A tabela `locations` deve ter uma entrada para o `destination_msisdn`
- O campo `location` deve corresponder ao nome do SMSC consultado
- O timestamp `expires` deve estar no futuro

Monitoramento do Roteamento Baseado em Localização:

Verifique os registros de localização:

```
# Via API
GET /api/locations/{msisdn}

# Verifique se a localização expirou
# o campo expires deve ser > hora atual
```

Problemas Comuns:

- **Mensagem não entregue**: Verifique se a localização expirou
- **Frontend errado**: Verifique se o campo `location` corresponde ao nome do frontend esperado
- **Localização não encontrada**: O assinante pode precisar se registrar novamente

Intervenções Manuais

Tentar Novamente Mensagem Falhada:

```
# Redefinir delivery_attempts e 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"
  }'
```

Alterar Destino:

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

Excluir Mensagem Presa:

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

Gerenciamento de Rotas

Ver Rotas Atuais

Interface da Web: Navegue até `/sms_routing`

Via API:

```
# Listar todas as rotas
curl https://api.example.com:8443/api/routes
```

Verificar Uso de Rotas:

Consulta do Prometheus:

```
# Mensagens roteadas por cada rota (última hora)
increase(sms_c_routing_route_matched_count[1h])
```

Adicionar Nova Rota

Interface da Web:

1. Navegue até `/sms_routing`
2. Clique em "Adicionar Nova Rota"
3. Preencha os campos:
   - **Prefixo de Chamada**: Prefixo do número de origem (opcional)
   - **Prefixo Chamado**: Prefixo do número de destino (obrigatório para roteamento geográfico)
   - **SMSC de Origem**: Filtro do sistema de origem (opcional)
   - **SMSC de Destino**: Gateway de destino (obrigatório, a menos que seja auto-resposta/descartado)
   - **Apenas na Rede**: Restringir esta rota apenas a destinos na rede (requer Diameter/HSS)
   - **Prioridade**: Prioridade da rota (1-255, menor = maior prioridade)
   - **Peso**: Peso de balanceamento de carga (1-100)

    - **Descrição**: Descrição legível por humanos
    - **Habilitado**: Marque para ativar imediatamente
4. Clique em "Salvar Rota"

Exemplo: Rota Geográfica:

- Prefixo Chamado: `+44`
- SMSC de Destino: `uk_gateway`
- Prioridade: 50
- Peso: 100
- Descrição: "Roteamento do Reino Unido"

Exemplo: Rota Balanceada:

Crie duas rotas com os mesmos critérios, mas pesos diferentes:

Rota 1:

- Prefixo Chamado: `+44`
- SMSC de Destino: `uk_primary`
- Prioridade: 50
- Peso: 70
- Descrição: "Reino Unido primário (70%)"

Rota 2:

- Prefixo Chamado: `+44`
- SMSC de Destino: `uk_backup`
- Prioridade: 50
- Peso: 30
- Descrição: "Reino Unido de backup (30%)"

Exemplo: Rota Apenas na Rede:

Restringir um transportador externo ou aplicação interna a enviar apenas para assinantes na rede. Mensagens para destinos fora da rede são descartadas.

- SMSC de Origem: `carrier_smpp_bind`

- SMSC de Destino: `local_msc`
- Apenas na Rede: marcado
- Prioridade: 50
- Peso: 100
- Descrição: "Transportadora X - apenas na rede"

Isso requer que o Diameter/HSS esteja habilitado. Se o HSS não estiver disponível, as mensagens são descartadas (fail-closed).

Testar Rotas

Simulador de Roteamento:

1. Navegue até `/simulator`
2. Insira os parâmetros de teste:
   - Número de Chamada: `+15551234567`
   - Número Chamado: `+447700900000`
   - SMSC de Origem: (opcional)
   - Tipo de Origem: (opcional)
3. Clique em "Simular Roteamento"
4. Revise os resultados:
   - **Rota Selecionada**: Qual rota o sistema escolheu
   - **Todas as Correspondências**: Quais rotas corresponderam aos critérios
   - **Avaliação**: Por que cada rota correspondeu ou não correspondeu

Teste Antes da Produção:

- Teste todas as novas rotas no simulador
- Verifique se a rota correta é selecionada
- Verifique a ordem de prioridade
- Valide a distribuição de peso

Modificar Rota Existente

Interface da Web:

1. Navegue até `/sms_routing`
2. Encontre a rota na lista
3. Clique em "Editar"
4. Modifique os campos
5. Clique em "Salvar Rota"

Modificações Comuns:

- **Desabilitar Rota**: Desmarque "Habilitado" (remoção temporária)
- **Ajustar Peso**: Alterar distribuição de balanceamento de carga
- **Alterar Prioridade**: Reordenar avaliação da rota
- **Atualizar Destino**: Mudar para SMSC diferente

Excluir Rota

Interface da Web:

1. Navegue até `/sms_routing`
2. Encontre a rota na lista
3. Clique em "Excluir"
4. Confirme a exclusão

Aviso: Excluir rotas é permanente. Considere desabilitar em vez disso.

Exportar/Importar Rotas

Exportar Rotas (Backup):

1. Navegue até `/sms_routing`
2. Clique em "Exportar Rotas"
3. Salve o arquivo JSON

Importar Rotas:

1. Navegue até `/sms_routing`
2. Clique em "Importar Rotas"
3. Selecione o arquivo JSON

4. Escolha o modo de importação:
    - **Mesclar**: Adicionar às rotas existentes
    - **Substituir**: Excluir todas e importar

Casos de Uso:

- Backup antes de grandes mudanças
- Copiar rotas entre ambientes
- Recuperação de desastres
- Versionamento de configuração

Gerenciamento de Frontend

Monitorar Conexões de Frontend

Interface da Web: Navegue até `/frontend_status`

Verifique:

- Todos os frontends esperados estão "ativos"
- Os horários da última vez vistos são recentes (< 90 segundos)
- Nenhum frontend expirado inesperado

Via API:

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

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

Investigar Desconexões

Frontend Expirado:

1. Verifique os logs do frontend para erros

2. Verifique a conectividade de rede com o SMS-C
3. Confirme se o frontend está em execução
4. Verifique a lógica de registro do frontend (deve se registrar a cada 60s)

Registro Não Aparecendo:

1. Verifique se o frontend está chamando POST `/api/frontends/register`
2. Verifique os logs da API para erros de registro
3. Verifique o formato do payload JSON
4. Teste o registro manualmente com curl

Exemplo 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 Histórico do Frontend

Interface da Web:

1. Navegue até `/frontend_status`
2. Encontre o frontend na lista
3. Clique em "Histórico"
4. Revise registros de registros anteriores

Via API:

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

Casos de Uso:

- Investigar a confiabilidade da conexão
- Rastrear padrões de tempo de atividade do frontend
- Identificar mudanças de configuração

Gerenciamento de Tradução de Números

As regras de tradução de números são gerenciadas via `config/runtime.exs`. Mudanças requerem reinício da aplicação.

Ver Regras de Tradução Ativas

Verifique o arquivo de configuração:

```
cat config/runtime.exs | grep -A 20 "translation_rules:"
```

Tarefas Comuns de Tradução

Adicionar Código de País a Números Locais:

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: "Adicionar +1 a números de 10 dígitos dos EUA",
  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: "Converter prefixo 00 para +",
  enabled: true
}
```

Remoção de Código Específico do Transportador:

```
%{
  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: "Remover código de transportadora do transportador
A",
  enabled: true
}
```

Testar Regras de Tradução

Após alterações de configuração:

1. Reinicie a aplicação para carregar novas regras
2. Envie uma mensagem de teste com origem/destino que deve corresponder
3. Verifique o log de eventos para o evento `number_translated`
4. Verifique se os números foram transformados corretamente

Desabilitar Regra de Tradução

Defina `enabled: false` na regra:

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

Reinicie a aplicação.

Manutenção do Sistema

Manutenção do Banco de Dados

Verificar Tamanho do Banco de Dados:

Use suas ferramentas de gerenciamento de banco de dados para monitorar o tamanho do armazenamento de CDR:

- **MySQL/MariaDB**: Consulte `information_schema.tables` para o tamanho do banco de dados
- **PostgreSQL**: Use a função `pg_database_size()` ou o comando `\l+` no psql

Limpar Registros CDR Antigos:

Os registros CDR devem ser arquivados e purgados periodicamente com base em sua política de retenção:

- Configure arquivamento automático com base nos requisitos de negócios (tipicamente 30-90 dias no banco de dados operacional)
- Arquive registros mais antigos para um data warehouse ou armazenamento frio
- Exclua registros arquivados do banco de dados operacional em lotes para evitar contenção de bloqueio

Otimizar Tabelas:

Periodicamente otimize as tabelas do banco de dados para manter o desempenho:

- **MySQL/MariaDB**: Execute o comando `OPTIMIZE TABLE` durante períodos de baixo tráfego
- **PostgreSQL**: Execute `VACUUM ANALYZE` regularmente (ou habilite autovacuum)

Executar Semanalmente durante o período de baixo tráfego para manter desempenho ideal.

Manutenção do Banco de Dados Mnesia

Verificar Tamanho da Tabela Mnesia:

```
# No console IEx
:mnesia.table_info(:sms_route, :size)
:mnesia.table_info(:translation_rule, :size)
```

Backup das Tabelas Mnesia:

```
# Exportar rotas (Interface da Web)
# Navegue até /sms_routing
# Clique em "Exportar Rotas"

# Ou via backup Mnesia
:mnesia.backup("/var/backups/sms_c/mnesia_backup.bup")
```

Restaurar Mnesia:

```
# Via importação da Interface da Web
# Ou restaurar backup:
:mnesia.restore("/var/backups/sms_c/mnesia_backup.bup", [])
```

Rotação de Logs

Configure logrotate para logs da aplicação:

```
# /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 Aplicação

Reinício Graceful (zero downtime em cluster):

```
# Reinicie um nó de cada vez
systemctl restart omnimessage

# Aguarde o nó se juntar ao cluster
# Repita para cada nó
```

Reinício de Emergência (todos os nós):

```
systemctl restart omnimessage
```

Após o Reinício:

- Verifique se todos os frontends se reconectam
- Verifique o Prometheus para continuidade das métricas
- Monitore logs para erros

- Verifique se o processamento de mensagens é retomado

Backup e Recuperação

O que Fazer Backup

1. **Arquivos de Configuração**:
    - `config/runtime.exs`
    - `config/config.exs`
    - `config/prod.exs` (se existir)
2. **Tabelas de Roteamento** (Mnesia):
    - Exportar via Interface da Web
    - Ou comando de backup Mnesia
3. **Banco de Dados SQL CDR**:
    - Backup completo diário
    - Backups de log de transações (contínuos)
4. **Certificados TLS**:
    - `priv/cert/*.crt`
    - `priv/cert/*.key`

Procedimentos de Backup

Backup Diário de Configuração:

```
#!/bin/bash
# /opt/omnimessage/scripts/backup_config.sh

BACKUP_DIR="/var/backups/sms_c/$(date +%Y%m%d)"
mkdir -p $BACKUP_DIR

# Backup de configuração
cp -r /opt/omnimessage/config $BACKUP_DIR/

# Backup de certificados
cp -r /opt/omnimessage/priv/cert $BACKUP_DIR/

# Definir permissões
chmod 600 $BACKUP_DIR/cert/*

echo "Backup de configuração concluído: $BACKUP_DIR"
```

Backup do Banco de Dados:

```
#!/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

# Backup do banco de dados SQL CDR
# MySQL/MariaDB: Use mysqldump com --single-transaction para
consistência
# PostgreSQL: Use pg_dump -F c para formato personalizado

# Estrutura de exemplo (adapte ao seu banco de dados):
# - Use a ferramenta de backup apropriada (mysqldump, pg_dump)
# - Habilite backups seguros para consistência
# - Comprimir saída para economizar espaço
# - Configurar período de retenção (por exemplo, 30 dias)

# Remover backups antigos
find $BACKUP_DIR -name "sms_c_*.gz" -mtime +30 -delete

echo "Backup do banco de dados concluído: sms_c_${DATE}"
```

Backup da Tabela de Roteamento:

```
#!/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 via API
curl https://api.example.com:8443/api/routes/export \
  > $BACKUP_DIR/routes_${DATE}.json

echo "Backup das rotas concluído: routes_${DATE}.json"
```

Agendar Backups (crontab):

```
# Diário às 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
```

Procedimentos de Recuperação

Restaurar Configuração:

```
# Parar aplicação
systemctl stop omnimessage

# Restaurar arquivos de configuração
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 aplicação
systemctl start omnimessage
```

Restaurar Banco de Dados SQL CDR:

Use ferramentas de restauração apropriadas para seu banco de dados:

- **MySQL/MariaDB**: Descomprimir e canalizar para o cliente mysql
- **PostgreSQL**: Usar pg_restore com dumps de formato personalizado

Importante: Pare a aplicação SMS-C antes de restaurar o banco de dados para evitar conflitos de dados.

Restaurar Tabelas de Roteamento:

1. Navegue até a Interface da Web `/sms_routing`
2. Clique em "Importar Rotas"
3. Selecione o arquivo JSON de backup

4. Escolha o modo "Substituir"
5. Confirme a importação

Planejamento de Capacidade

Monitorar Tendências de Crescimento

Tendência de Volume de Mensagens:

Consulta do Prometheus (média de 30 dias):

```
avg_over_time(sms_c_message_received_count[30d])
```

Taxa de Crescimento do Banco de Dados:

```sql
-- Crescimento de dados mensal
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 Capacidade

Uso de CPU:

- **Normal**: < 50% em média
- **Alto**: > 70% sustentado
- **Crítico**: > 90%

Uso de Memória:

- **Normal**: < 70% do disponível

- **Alto**: > 80%
- **Crítico**: > 90%

Uso de Disco:

- **Normal**: < 60% cheio
- **Alto**: > 75%
- **Crítico**: > 85%

Profundidade da Fila:

- **Normal**: < 1000 pendentes
- **Alto**: > 5000 pendentes
- **Crítico**: > 10.000 pendentes

Recomendações de Escalonamento

Quando Escalonar Verticalmente (Atualizar Recursos):

- CPU consistentemente > 70%
- Memória consistentemente > 80%
- Gargalo de nó único

Quando Escalonar Horizontalmente (Adicionar Nós):

- CPU > 50% em todos os nós
- Volume de mensagens > 5.000 msg/sec
- Distribuição geográfica necessária
- Alta disponibilidade requerida

Escalonamento de Banco de Dados:

- Réplicas de leitura para consultas de relatórios
- Otimização de pooling de conexões
- Otimização de índices
- Particionar tabelas grandes por data

Resposta a Incidentes

Níveis de Severidade

Crítico (Resposta Imediata):

- Nenhuma mensagem sendo entregue
- Todos os frontends desconectados
- Banco de dados indisponível
- API completamente fora do ar

Alto (Resposta dentro de 1 hora):

- Taxa de sucesso na entrega < 80%
- Múltiplos frontends desconectados
- Falhas de roteamento > 10%
- Fila de backlog crescendo

Médio (Resposta dentro de 4 horas):

- Um único frontend desconectado
- Taxa de sucesso na entrega 80-95%
- Processamento de mensagens lento
- Pesquisas ENUM falhando

Baixo (Resposta dentro de 24 horas):

- Degradação de desempenho menor
- Problema em uma única rota
- Alertas de aviso não críticos

Checklist de Incidentes

1. Avaliar Severidade:

- Verifique os alertas do Prometheus
- Revise as métricas do painel

- Verifique o status da fila de mensagens
- Verifique as conexões do frontend

2. Reunir Informações:

- Mudanças de configuração recentes?
- Implantações recentes?
- Status de dependências externas (OCS, DNS)?
- Mensagens de erro nos logs?

3. Ações Imediatas:

- Pare mudanças em andamento
- Reverta implantações recentes se suspeitar que são a causa
- Habilite logs detalhados se necessário
- Notifique as partes interessadas

4. Investigação:

- Revise logs da aplicação
- Verifique o uso de recursos do sistema
- Examine o desempenho do banco de dados
- Teste dependências externas

5. Resolução:

- Aplique a correção
- Teste no simulador
- Implante em produção
- Monitore para melhorias

6. Pós-Incidente:

- Documente a causa raiz
- Atualize monitoramento/alertas
- Implemente medidas preventivas
- Atualize runbooks

Incidentes Comuns

Alta Pendência na Fila:

1. Verifique a taxa de sucesso na entrega
2. Verifique se os frontends estão conectados e consultando
3. Verifique o desempenho do banco de dados
4. Revise o Prometheus para gargalos
5. Considere aumentar o tamanho do lote/intervalo

Falhas de Roteamento:

1. Revise a configuração de roteamento
2. Teste no simulador de roteamento
3. Verifique se há rotas ausentes
4. Verifique se a rota catch-all existe
5. Verifique logs de eventos para razões de falha

Desconexões de Frontend:

1. Verifique o status do sistema do frontend
2. Verifique a conectividade de rede
3. Revise logs do frontend
4. Teste registro manual da API
5. Verifique regras de firewall

Processamento de Mensagens Lento:

1. Verifique o desempenho de consultas ao banco de dados
2. Revise a configuração do trabalhador de lotes
3. Verifique se há recursos adequados (CPU/Memória)
4. Verifique se há atrasos nas pesquisas ENUM
5. Revise o desempenho do sistema de cobrança

Para procedimentos detalhados de solução de problemas, veja o Guia de Solução de Problemas.

Roteamento de SMS

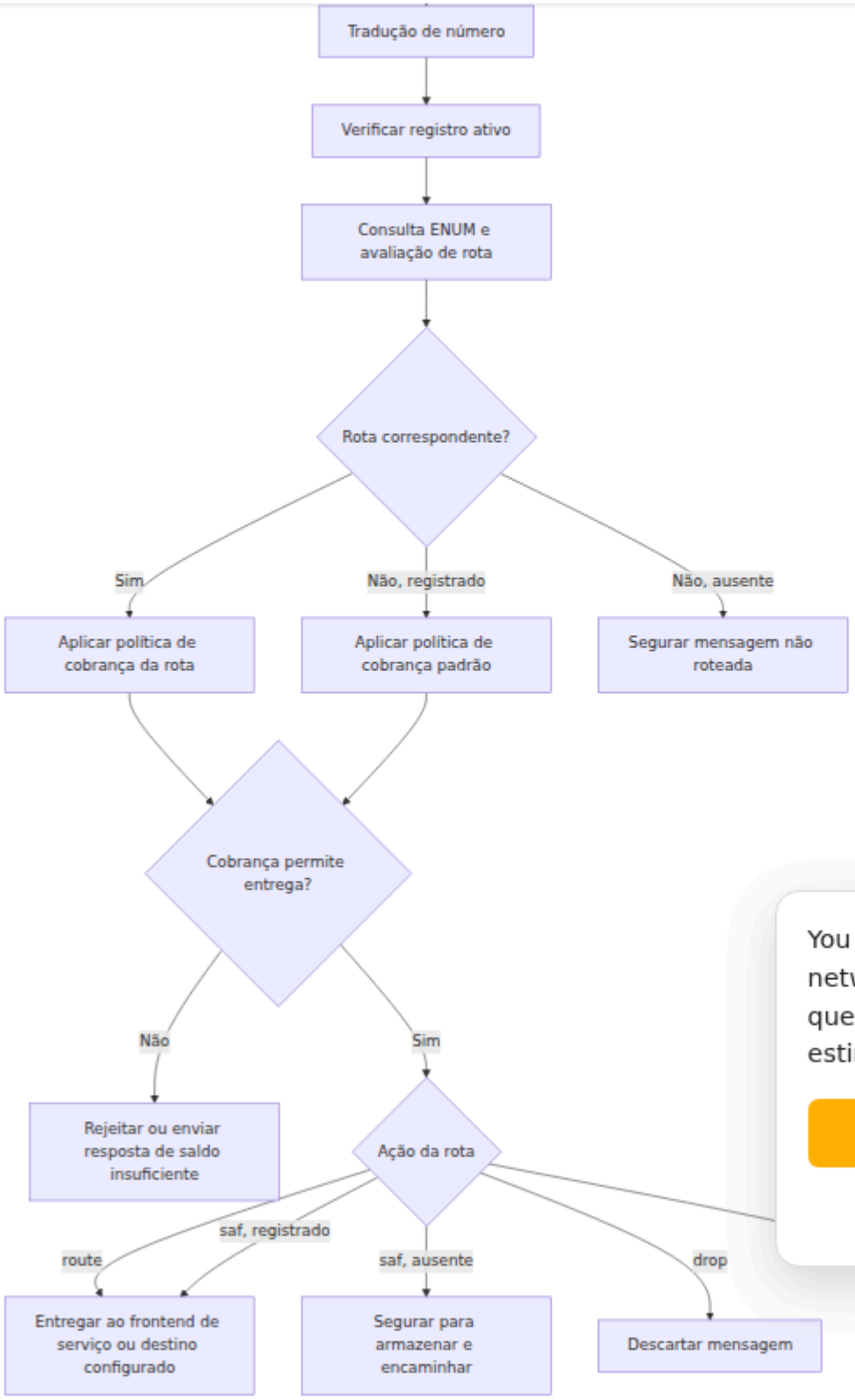
← Voltar ao Índice da Documentação

Visão Geral

O mecanismo de roteamento SMS-C determina para onde cada mensagem vai após entrar no sistema. O Mnesia armazena as rotas. Você pode modificá-las em tempo de execução via API ou painel de controle. As rotas suportam correspondência regex, ordenação por prioridade, balanceamento de carga ponderado, correspondência de resultados ENUM (DNS NAPTR) e correspondência on-net/off-net baseada em HSS.

Pipeline de Processamento de Mensagens

Cada mensagem segue este pipeline desde a submissão até o roteamento:



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

O status on-net/off-net é um critério de correspondência (`originating_network` / `terminating_network`). Veja Correspondência On-Net / Off-Net.

Política e Ordem de Entrega

1. Verifique se o assinante de destino tem um registro ativo.
2. Avalie as rotas habilitadas e selecione a rota correspondente. Registre os resultados da correspondência no histórico da mensagem.
3. Aplique a política de cobrança da rota selecionada antes da entrega. Se nenhuma rota corresponder a um assinante registrado, aplique a política de cobrança padrão.
4. Execute a ação da rota. Para `route` e `saf`, um registro ativo seleciona o frontend de serviço. Uma mensagem `saf` aguarda se o assinante estiver ausente. As ações `drop` e `auto_reply` também se aplicam a assinantes registrados.

Cobrança e Eventos de Mensagem

Um registro ativo não isenta a cobrança. A política `charged` tem estes valores:

Valor	Comportamento
`yes`	Solicitar autorização OCS para o remetente.
`no`	Pular a solicitação OCS.
`default`	Usar a configuração `default_charging_enabled`.

Novas mensagens sem um SMSC de destino têm status `routing` até que a avaliação da rota e a cobrança terminem. Frontends não podem buscá-las durante esta etapa. Atualizações de registro não podem liberá-las antecipadamente. Respostas automáticas e submissões com um SMSC de destino explícito mantêm seu comportamento de entrega existente.

O histórico da mensagem registra estes eventos de cobrança:

Evento	Significado
`charging_requested`	O SMSC solicita autorização OCS.
`charging_succeeded`	O OCS autoriza a cobrança.
`charging_failed`	O OCS recusa a solicitação ou retorna um erro. A descrição indica se a entrega prossegue.
`charging_skipped`	A política selecionada desativa a cobrança.

Crédito insuficiente rejeita a mensagem ou envia a resposta configurada. Um erro OCS segue a configuração `on_charging_error`. Com `allow`, a entrega pode prosseguir após um erro. O evento registra isso como uma falha com entrega permitida, em vez de uma cobrança bem-sucedida.

Modelo de Dados da Rota

Campo	Tipo	Descrição
`route_id`	Inteiro	Identificador único gerado automaticamente
`calling_regex`	String	Regex para número de origem (`nil` = curinga)
`called_regex`	String	Regex para número de destino (`nil` = curinga)
`source_smsc`	String	Nome do SMSC de origem (`nil` = curinga)
`dest_smsc`	String	SMSC de destino. Necessário para `route`; opcional para `saf`; não utilizado por `drop`/`auto_reply`
`source_type`	String	Filtrar por tipo de origem: `ims`, `circuit_switched`, `smpp` ou `nil`
`enum_domain`	String	Rótulo de domínio ENUM para a rota (`nil` = nenhum). Armazenado na rota e adiciona especificidade, mas a consulta DNS real usa a configuração global `enum_domains` - veja Consulta ENUM
`enum_result_domain`	String	Critério de correspondência: a rota corresponde apenas se um resultado ENUM NAPTR resolver para este domínio (`nil` = curinga)

Campo	Tipo	Descrição
`action`	String	Ação da rota: `route`, `drop`, `auto_reply`, `saf`
`auto_reply_message`	String	Texto da resposta (necessário quando a ação é `auto_reply`)
`charged`	String	Cobrança: `yes`, `no`, `default`
`weight`	Inteiro	Peso de balanceamento de carga (1–100)
`priority`	Inteiro	Prioridade (1–255, menor = maior prioridade)
`description`	String	Rótulo legível por humanos
`enabled`	Booleano	Alternância ativa/inativa
`originating_network`	Atom	Critério de correspondência no status HSS do MSISDN de origem: `:any` | `:on_net` | `:off_net` (padrão `:any`)
`terminating_network`	Atom	Critério de correspondência no status HSS do MSISDN de destino: `:any` | `:on_net` | `:off_net` (padrão `:any`)

Ações da Rota

O campo `action` tem exatamente quatro valores válidos: `:route`, `:drop`, `:auto_reply`, `:saf`.

`route` (Padrão)

Entrega a mensagem ao `dest_smsc`. Esta é a ação de roteamento padrão.

```
{
  "action": "route",
  "called_regex": "^61",
  "dest_smsc": "australia-gw",
  "priority": 100,
  "description": "Roteie números australianos para o gateway AU"
}
```

`drop`

Descarta a mensagem. Nenhum `dest_smsc` necessário.

```
{
  "action": "drop",
  "called_regex": "^900",
  "priority": 5,
  "description": "Bloquear números de tarifa premium"
}
```

`auto_reply`

Envia `auto_reply_message` de volta ao originador. A mensagem original é marcada como respondida automaticamente.

```
{
  "action": "auto_reply",
  "called_regex": "^911$",
  "auto_reply_message": "Emergência: SMS para 911 não é suportado. Por favor, disque 911.",
  "priority": 1,
  "description": "Resposta automática 911"
}
```

`saf`

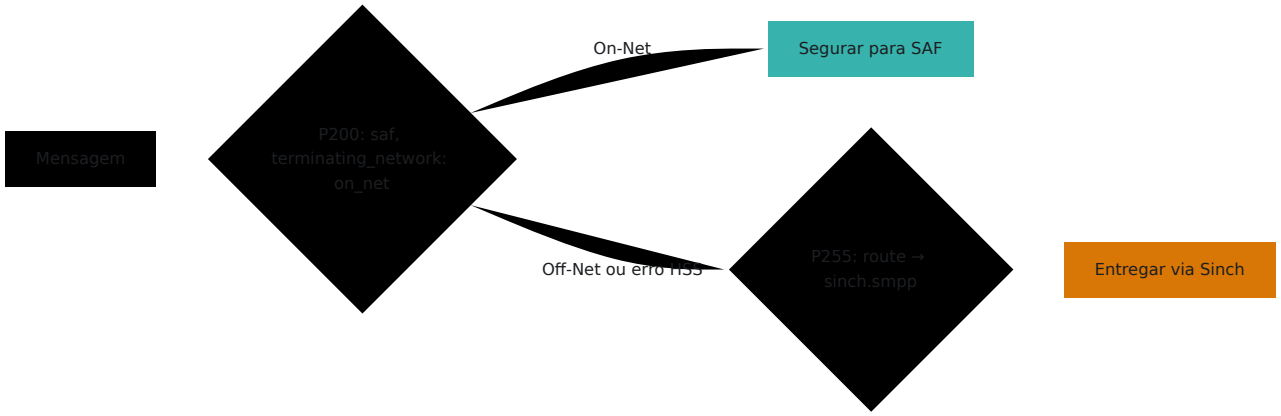
Armazenar e encaminhar. A mensagem é mantida e entregue posteriormente, quando o assinante de destino se registrar. `dest_smsc` é opcional (se definido,

é usado quando a mensagem retida é finalmente entregue).

`saf` em si é **incondicional** - não realiza uma consulta HSS. Para segurar *apenas* assinantes on-net para armazenar e encaminhar, combine `saf` com um critério de correspondência `terminating_network: :on_net` (veja abaixo) e forneça uma rota de fallback off-net.

```
{
  "action": "saf",
  "terminating_network": "on_net",
  "priority": 200,
  "description": "Segurar assinantes on-net para armazenar e
encaminhar"
}
```

Configuração Típica: Retenção On-Net + Fallback Off-Net



A rota P200 só *corresponde* quando o destino é confirmado on-net, então assinantes on-net são mantidos para SAF. Números off-net (e erros HSS - falha fechada) falham nessa rota e caem na P255 de fallback. Isso garante que assinantes on-net nunca vazem para o gateway off-net.

Consulta ENUM

ENUM permite que decisões de roteamento dependam de uma consulta DNS E.164 NAPTR do número de destino. Isso é típico para portabilidade de números ou roteamento entre operadoras, onde o DNS (não seu HSS) é a autoridade sobre a qual operadora um número pertence.

Como funciona

1. **Habilitado globalmente.** ENUM só é executado quando `enum_enabled: true` e `enum_domains` não está vazio. É uma única consulta realizada **uma vez por mensagem, antes da tabela de rotas ser avaliada** - não por rota.
2. **Consulta.** O número chamado é convertido para o formato ENUM reverso (`+15551234` → `4.3.2.1.5.5.5.1.e164.arpa`) e uma consulta NAPTR é emitida. Cada domínio configurado em `enum_domains` é testado **em ordem**, parando no primeiro que retorna registros.
3. **Extração do domínio de resultado.** Cada registro NAPTR retornado é analisado e reduzido a um `result_domain` - os dois últimos rótulos do campo `replacement`, ou do host em uma reescrita `sip:user@host` (`sip:+15551234@ims.example.com` → `example.com`).
4. **Correspondência.** Uma rota com `enum_result_domain` definida corresponde apenas se **qualquer** registro NAPTR retornado tiver `result_domain` igual a essa string exatamente. `enum_result_domain: nil` é um curinga (sempre corresponde, mesmo quando nenhuma consulta ENUM foi realizada).
5. **Cache.** Os resultados são armazenados em ETS por 15 minutos por `(número, domínio)`. Falhas também são armazenadas, para evitar sobrecarregar o DNS em falhas repetidas.

Derivando on-net/off-net do ENUM

ENUM não retorna um status booleano on/off-net - retorna um domínio. Você expressa "on-net" apontando o `enum_result_domain` de uma rota para seu próprio domínio IMS/operadora e dando a tudo mais uma rota de fallback. Contraste isso com a correspondência HSS (abaixo), que retorna uma resposta explícita on-net/off-net.

O que o ENUM *não* faz

- O resultado NAPTR bruto (`enum_result`) é usado **apenas** como uma chave de correspondência de rota. Ele **não** é usado para reescrever o URI de destino ou escolher um próximo salto - para onde uma mensagem

correspondente vai é inteiramente `route.dest_smsc`. O pipeline de entrega ao vivo descarta o resultado ENUM bruto após o roteamento.

- `enum_domain` em uma rota **não** é usado para escopo da consulta DNS (a consulta sempre usa a lista global `enum_domains`) e **não** é um critério de correspondência. Ele apenas contribui para pontuação de especificidade.

Configuração

O comportamento DNS do ENUM (`enum_enabled`, `enum_domains`, `enum_dns_servers`, `enum_timeout`) é configurado globalmente - veja CONFIGURAÇÃO.md → Configuração ENUM.

Correspondência On-Net / Off-Net

Uma **consulta Diameter Sh no HSS** responde se um destino (ou originador) está on-net. As rotas apresentam isso como dois critérios de correspondência. Estes **não** são flags booleanos e **não** são uma ação de rota. Eles são critérios de valor enum que o roteador avalia junto com as verificações regex/SMSC/tipo. Uma rota só corresponde se satisfizer seus critérios de rede.

`originating_network` / `terminating_network`

Cada um recebe um dos três valores:

Valor	Significado
`:any`	Sem verificação HSS; corresponde tanto on-net quanto off-net (padrão)
`:on_net`	A rota corresponde apenas se o MSISDN for confirmado **on-net** (HSS retorna sucesso, código de resultado 2001)
`:off_net`	A rota corresponde apenas se o MSISDN for confirmado **off-net** (HSS retorna *Usuário Desconhecido*, código de resultado 5001)

- `originating_network` verifica o número **chamador**; `terminating_network` verifica o número **chamado**.
- **Casos de uso**: restringir uma rota para entregar apenas a assinantes on-net (`terminating_network: :on_net`); restringir um bind SMPP ou operadora externa para aceitar apenas originadores on-net (`originating_network: :on_net`); enviar um fallback apenas off-net via um gateway externo (`terminating_network: :off_net`).

Comportamento de falha fechada

Qualquer valor diferente de `:any` requer uma resposta definitiva do HSS. Se a resposta não for definitiva, a rota **não corresponde** (e o mecanismo tenta a próxima rota):

- Diameter desabilitado → a rota não corresponde
- HSS inacessível / timeout / erro de transporte → a rota não corresponde
- HSS retorna um erro diferente de *Usuário Desconhecido* → a rota não corresponde

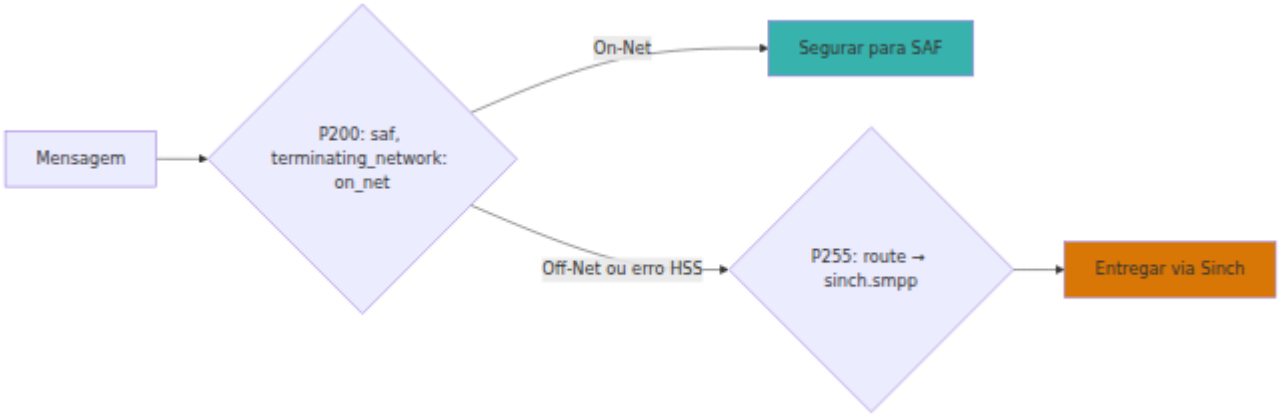
Observe que mesmo `:off_net` falha fechada: corresponde apenas a um explícito *Usuário Desconhecido* (5001), não a uma falha de transporte. Isso evita o vazamento de tráfego para um gateway off-net quando o HSS está simplesmente fora do ar.

Cache de Resultados HSS

As consultas HSS são caras (uma viagem de ida e volta Diameter). O sistema armazena resultados por MSISDN para evitar consultas redundantes.

- **TTL do Cache**: 2 horas (configurável via `hss_cache_ttl_seconds`)
- **Consulta preguiçosa**: o HSS é consultado apenas quando uma rota sendo avaliada tem `originating_network` ou `terminating_network` definido como `:on_net` ou `:off_net`. Rotas que deixam ambos como `:any` nunca acionam uma consulta.
- **Compartilhado entre rotas**: se várias rotas precisarem do status do mesmo MSISDN, apenas uma consulta HSS ocorre

- **Re-rota forçada**: o cache é ignorado (uma nova consulta é feita) quando uma mensagem é explicitamente re-roteada



Registro de Avaliação de Rota

Cada rota avaliada para uma mensagem é registrada como um evento `route_evaluated` nessa mensagem. O log inclui:

- ID da rota, ação e prioridade
- Status de aprovação/reprovação para cada critério de correspondência (incluindo `enum_result` e as verificações HSS `orig_network`/`term_network`)
- Detalhe do que foi comparado (padrão regex vs número, nome do SMSC vs origem, rede requerida vs status HSS real, etc.)

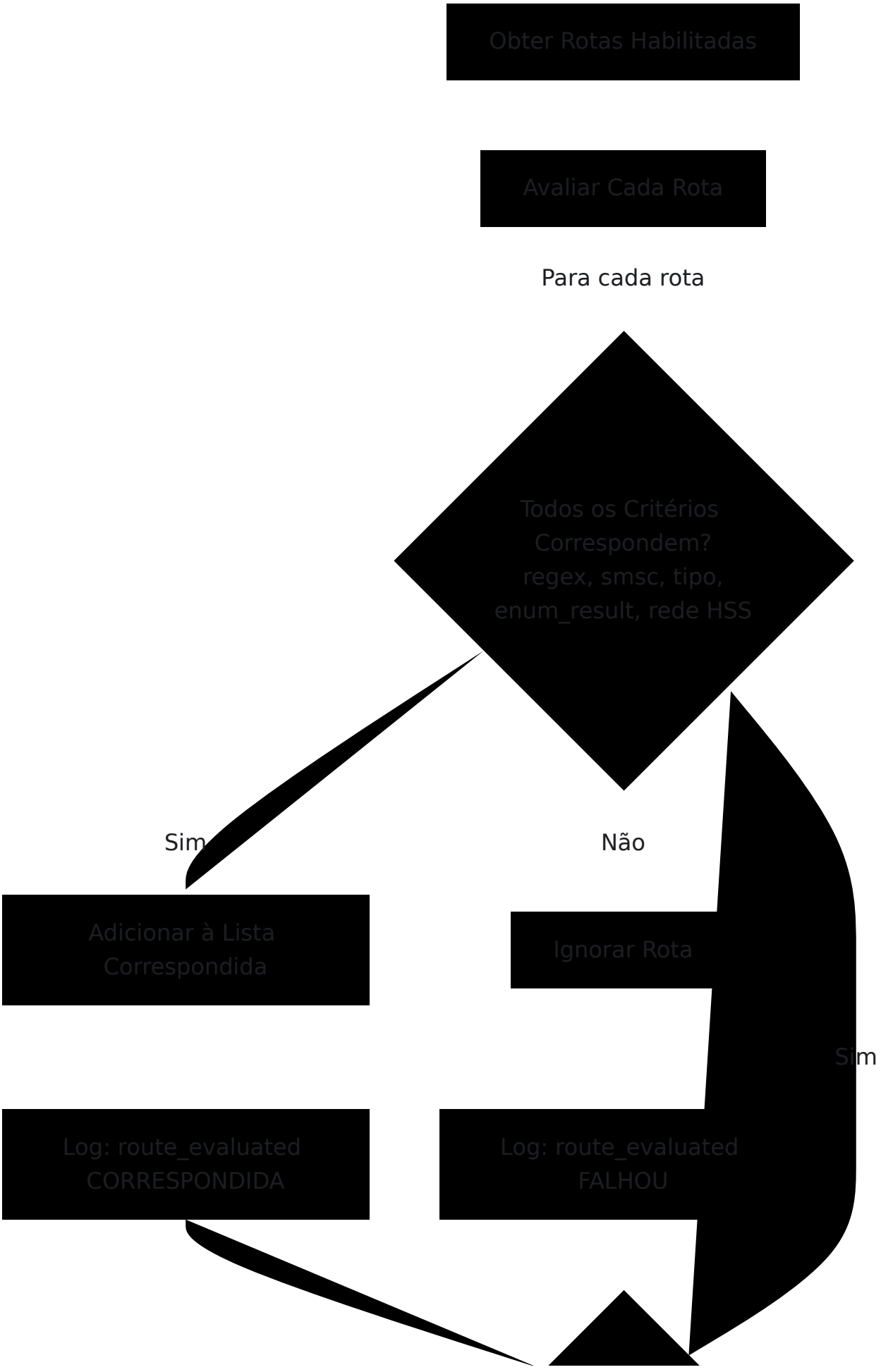
Exemplo de entradas de log de eventos:

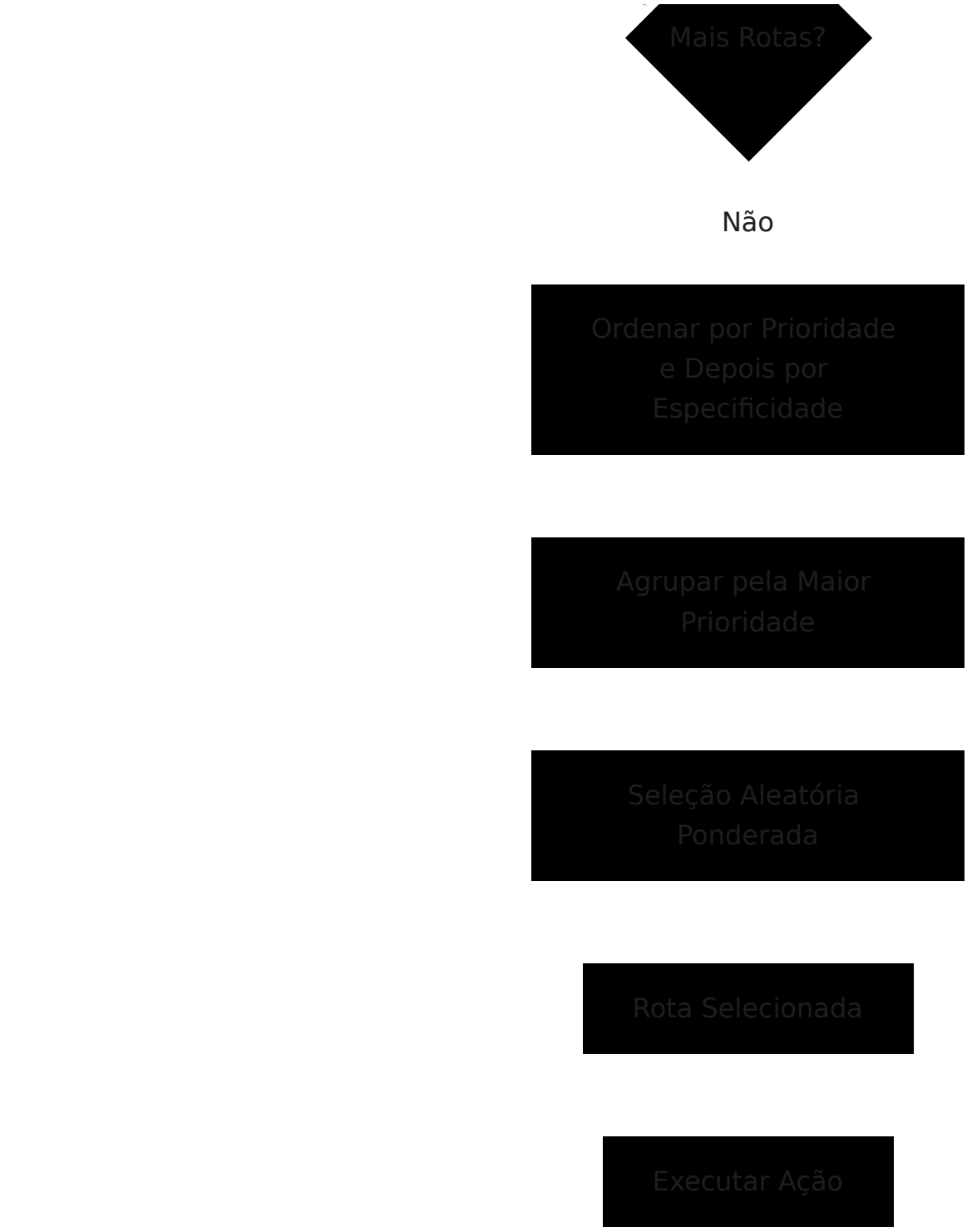
```
Rota #4 [saf] CORRESPONDIDA: P200 "Segurar on-net" |
  calling_regex: ✓ (curinga), called_regex: ✓ (curinga),
  source_smsc: ✓ (curinga), source_type: ✓ (curinga),
  enum_result: ✓ (curinga),
  term_network: ✓ (terminando requer on_net, +15551234 está on-
net)

Rota #3 [route] FALHOU: P255 "Fallback Sinch" → sinch.smpp |
  calling_regex: ✓ (curinga), called_regex: ✓ (curinga),
  source_smsc: ✓ (curinga), source_type: ✗ (ims vs
circuit_switched),
  enum_result: ✓ (curinga), term_network: ✓ (qualquer)
```

Esses eventos são visíveis no log de atividade de mensagens do painel de controle e via API de Eventos.

Algoritmo de Avaliação de Rota


Obter Rotas Habilitadas
Avaliar Cada Rota
Para cada rota
Todos os Critérios Correspondem? regex, smsc, tipo, enum_result, rede HSS
Sim
Não
Adicionar à Lista Correspondida
Ignorar Rota
Sim
Log: route_evaluated CORRESPONDIDA
Log: route_evaluated FALHOU



A verificação on-net/off-net do HSS é parte da **correspondência de critérios** (uma rota que falha simplesmente não corresponde) - não há uma etapa separada de HSS pós-seleção.

Pontuação de Especificidade

Quando várias rotas correspondem, a especificidade determina qual é preferida dentro do mesmo nível de prioridade:

Critério	Pontos
Comprimento do padrão `called_regex`	comprimento × 100
Comprimento do padrão `calling_regex`	comprimento × 50
`source_smsc` especificado	+25
`enum_result_domain` especificado	+15
`source_type` especificado	+10
`enum_domain` especificado	+5

Padrões mais longos e mais específicos sempre vencem sobre curingas.

Configuração

Carregando Rotas da Configuração

Rotas definidas em `config/runtime.exs` sob `:sms_routes` são carregadas na primeira inicialização quando a tabela de roteamento está vazia. Reinicializações subsequentes preservam rotas adicionadas via API.

```
config :sms_c, :sms_routes, [
  %{
    called_regex: ~r/^15555/,
    dest_smsc: "example-gw",
    action: :route,
    priority: 100,
    description: "Roteamento nacional de exemplo"
  },
  %{
    action: :saf,
    terminating_network: :on_net,
    priority: 200,
    description: "Segurar assinantes on-net para armazenar e
encaminhar"
  },
  %{
    action: :auto_reply,
    called_regex: ~r/^911$/,
    auto_reply_message: "Emergência: Por favor, disque 911.",
    priority: 1,
    description: "Resposta automática 911"
  }
]
```

Parâmetros de Configuração da Rota

Parâmetro	Tipo	Necessário	Padrão	Desc
`action`	Atom/String	Não	`:route`	Ação da r `:route`, `: :auto_re`
`calling_regex`	Regex	Não	`nil`	Padrão re número d `nil` corre todos
`called_regex`	Regex	Não	`nil`	Padrão re número d `nil` corre todos
`source_smsc`	String	Não	`nil`	Filtrar por SMSC de `nil` corre todos
`dest_smsc`	String	Condicional	`nil`	Necessári `route`. Op para `saf`. utilizado p `drop`/`aut`
`source_type`	Atom	Não	`nil`	Filtrar por origem: `: :circuit_ :smpp`
`enum_domain`	String	Não	`nil`	Rótulo de ENUM par (adiciona

Parâmetro	Tipo	Necessário	Padrão	Desc
				especifici escopo da
`enum_result_domain`	String	Não	`nil`	Correspon apenas se resultado NAPTR res este domí correspon
`auto_reply_message`	String	Condicional	`nil`	Necessári a ação é `:auto_re`
`charged`	Atom	Não	`:default`	Política de cobrança: `:no`, `:def`
`weight`	Inteiro	Não	`100`	Peso de balancear carga (1–1
`priority`	Inteiro	Não	`100`	Prioridade avaliação menor = r
`description`	String	Não	`""`	Descrição por huma
`enabled`	Booleano	Não	`true`	Se a rota
`originating_network`	Atom	Não	`:any`	Requisito HSS do MS origem: `:` `:on_net`,

Parâmetro	Tipo	Necessário	Padrão	Desc
`terminating_network`	Atom	Não	`:any`	Requisito HSS do MS destino: `:` `:on_net`,

Configuração do Cache HSS

Parâmetro	Tipo	Padrão	Descrição
`hss_cache_ttl_seconds`	Inteiro	`7200`	Tempo de vida para resultados de consulta HSS em cache (segundos). Definido em `runtime.exs`

API

Criar Rota

```
POST /api/routes
```

```
{
  "action": "saf",
  "terminating_network": "on_net",
  "priority": 200,
  "description": "Segurar assinantes on-net para armazenar e encaminhar"
}
```

Listar Rotas

```
GET /api/routes
```

Atualizar Rota

```
PUT /api/routes/:id
```

Deletar Rota

```
DELETE /api/routes/:id
```

Importar/Exportar

```
POST /api/routes/import
GET /api/routes/export
```

A importação suporta modos `merge` (adicionar ao existente) e `replace` (limpar e substituir tudo).

Solução de Problemas

Nenhuma Rota Encontrada

Sintomas: Mensagem permanece pendente com `dest_smsc: null`, log de eventos mostra `sms_routing_failed`

Possíveis causas:

- Nenhuma rota habilitada corresponde aos critérios da mensagem
- Padrões regex da rota não correspondem ao formato do número (por exemplo, código de país ausente)

- Cada rota candidata tem um requisito `terminating_network`/`originating_network` que o HSS não atendeu (off-net, ou um erro HSS sob falha fechada), e nenhuma rota de fallback `:any` existe

Resolução:

1. Verifique o log de eventos da mensagem - cada rota avaliada é registrada com aprovação/reprovação por critério (incluindo a verificação da rede HSS)
2. Use o simulador de roteamento no painel de controle para testar com os mesmos números
3. Certifique-se de que uma rota de fallback curinga (`originating_network: :any`, `terminating_network: :any`) exista em um número de prioridade alto (por exemplo, 255)

Assinante On-Net Roteado Off-Net

Sintomas: Mensagens para assinantes on-net são entregues via um gateway externo em vez de serem mantidas/entregues localmente

Possíveis causas:

- Nenhuma rota com `terminating_network: :on_net` configurada para capturar destinos on-net
- A rota on-net tem um número de prioridade *maior* (menor precedência) do que a rota de fallback off-net
- Falha na consulta HSS (verifique `hss_dip_error` / telemetria Sh) - sob falha fechada a rota on-net não corresponde e o tráfego cai na rota de fallback
- Diameter não habilitado na configuração

Resolução:

1. Adicione uma rota com `terminating_network: :on_net` em um número de prioridade mais baixo do que o fallback off-net
2. Verifique se o Diameter está habilitado e se os pares HSS estão conectados via o painel de controle

3. Verifique o status do cache HSS e os resultados recentes da consulta no log de eventos

Cache HSS Obsoleto

Sintomas: Assinante portado ou provisionado, mas ainda roteando com base no antigo status on-net/off-net

Resolução:

1. Re-roteie a mensagem via API - a re-rota forçada ignora o cache HSS
2. Aguarde a expiração do TTL do cache (padrão 2 horas)
3. Reduza `hss_cache_ttl_seconds` se atualizações mais rápidas forem necessárias

TROUBLESHOOTING

```
---
last_update:
  date: "2026-08-20T00:30:03Z"
---

# Guia de Solução de Problemas do SMS-C

**[← Voltar ao Índice da Documentação](../README.md)**

Guia abrangente para diagnosticar e resolver problemas comuns do SMS-C.

## Índice

- [Ferramentas de Diagnóstico](#ferramentas-de-diagnóstico)
- [Problemas de Entrega de Mensagens](#problemas-de-entrega-de-mensagens)
- [Problemas de Roteamento](#problemas-de-roteamento)
- [Problemas de Desempenho](#problemas-de-desempenho)
- [Problemas de Banco de Dados](#problemas-de-banco-de-dados)
- [Problemas de Conexão com o Frontend](#problemas-de-conexão-com-o-frontend)
- [Problemas de Cobrança/Faturamento](#problemas-de-cobrançafaturamento)
- [Problemas de Pesquisa ENUM](#problemas-de-pesquisa-enum)
- [Problemas de Cluster](#problemas-de-cluster)
- [Problemas de API](#problemas-de-api)
- [Problemas de Interface Web](#problemas-de-interface-web)
- [Problemas de Recursos do Sistema](#problemas-de-recursos-do-sistema)

## Ferramentas de Diagnóstico

### Verificação Rápida de Saúde

```bash
# 1. Verificar status da API
curl https://api.example.com:8443/api/status

# 2. Verificar endpoint de métricas do Prometheus
curl https://api.example.com:9568/metrics | grep sms_c

# 3. Verificar logs da aplicação
```
```

```
tail -f /var/log/sms_c/application.log

# 4. Verificar status do processo
systemctl status omnimessage

# 5. Verificar conectividade do banco de dados 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"
```

Análise de Logs

Ver Erros Recentes:

```
# Últimas 100 entradas de log de nível de erro
tail -1000 /var/log/sms_c/application.log | grep "\[error\]"

# Procurar por padrões de erro específicos
grep "routing_failed" /var/log/sms_c/application.log

# Encontrar erros de banco de dados SQL
grep -i "database\|sql\|ecto" /var/log/sms_c/application.log |
grep error
```

Monitorar Logs em Tempo Real:

```
# Seguir logs com filtro
tail -f /var/log/sms_c/application.log | grep -E "
(error|warning|critical)"
```

Consultas de Métricas

Verificar Taxa de Processamento de Mensagens:

```
# Mensagens por segundo
rate(sms_c_message_received_count[5m])

# Taxa de sucesso de entrega
rate(sms_c_delivery_succeeded_count[5m]) /
rate(sms_c_delivery_queued_count[5m])
```

Verificar Status da Fila:

```
# Profundidade atual da fila
sms_c_queue_size_pending

# Idade da mensagem mais antiga (segundos)
sms_c_queue_oldest_message_age_seconds
```

Verificar Desempenho do Sistema:

```
# Latência de processamento de mensagens (p95)
histogram_quantile(0.95,
sms_c_message_processing_stop_duration_bucket)

# Latência de roteamento (p95)
histogram_quantile(0.95, sms_c_routing_stop_duration_bucket)
```

Problemas de Entrega de Mensagens

Mensagens Não Sendo Entregues

Sintomas:

- Mensagens presas no status "queued"/"backoff"
- Alta contagem de mensagens não entregues
- Sem notificações de entrega

Passos de Diagnóstico:

1. **Verificar Conexões do Frontend**:

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

Esperado: Lista de frontends ativos
Problema: Lista vazia ou frontends esperados ausentes

2. **Verificar Fila de Mensagens**:

Acesse a Interface Web: `/message_queue`

- Filtrar por status: "queued" ou "backoff"
- Verificar valor de `dest_smsc`
- Verificar se `deliver_after` não está no futuro

3. **Verificar Roteamento**:

Acesse a Interface Web: `/simulator`

- Testar com parâmetros de mensagem reais
- Verificar se a rota corresponde e se o destino está correto

4. **Verificar Polling do Frontend**:

Revisar logs do sistema do frontend:

- O frontend está consultando `/api/messages`?
- O frontend está enviando o cabeçalho `smsc` corretamente?

Soluções:

Nenhum Frontend Conectado:

```
# Verificar status do sistema do frontend
systemctl status frontend_service

# Verificar se o frontend pode alcançar a API
curl -k https://api.example.com:8443/api/status

# Registrar frontend manualmente
curl -X POST https://api.example.com:8443/api/frontends/register \
  -H "Content-Type: application/json" \
  -d '&#123;
    "frontend_name": "test_gateway",
    "frontend_type": "smpp",
    "ip_address": "10.0.1.50"
  &#125;'
```

Mensagens Roteadas para o SMSC Errado:

- Revisar configuração de roteamento
- Verificar prioridades de rota
- Testar no simulador de roteamento
- Verificar se o nome do frontend corresponde a `dest_smsc` nas mensagens

Mensagens Agendadas para o Futuro:

- Verificar timestamp de `deliver_after`
- Redefinir se necessário:

```
curl -X PATCH https://api.example.com:8443/api/messages/12345 \
  -H "Content-Type: application/json" \
  -d '&#123;"deliver_after": "2025-10-30T12:00:00Z"&#125;'
```

Mensagens Falhando com Retentativas

Sintomas:

- Contador de `delivery_attempts` aumentando
- Mensagens com alta contagem de tentativas (> 3)
- Atrasos de backoff exponencial

Passos de Diagnóstico:

1. **Verificar Log de Eventos**:

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

Procurar por:

- Eventos de falha de entrega
- Descrições de erro
- Timestamps de retentativa

2. **Verificar Logs do Frontend**:

- Por que o frontend está falhando na entrega?
- Erros de rede?
- Erros de protocolo?
- Sistema downstream indisponível?

Soluções:

Problemas Temporários de Rede:

- Aguardar retentativa (automático)
- Monitorar para entrega bem-sucedida

Falhas Persistentes:

```
# Roteamento para gateway alternativo
curl -X PATCH https://api.example.com:8443/api/messages/12345 \
  -H "Content-Type: application/json" \
  -d '&#123;"dest_smsc": "backup_gateway"&#125;'

# Redefinir contador de retentativas
curl -X PATCH https://api.example.com:8443/api/messages/12345 \
  -H "Content-Type: application/json" \
  -d '&#123;"delivery_attempts": 0, "deliver_after": "2025-10-
30T12:00:00Z"&#125;'
```

Número de Destino Inválido:

- Verificar formato do número
- Verificar regras de tradução de números
- Excluir mensagem se realmente inválida

Mensagens de Carta Morta

Sintomas:

- `deadletter: true` na mensagem
- Mensagens além do tempo de expiração
- Status ainda "queued"/"backoff" (entregas falhadas não são dead-lettered; elas param de ser atendidas uma vez que ultrapassam `expires`)

Passos de Diagnóstico:

1. **Encontrar Mensagens de Carta Morta**:

Acesse a Interface Web: `/message_queue`

- Filtrar por status expirado
- Verificar timestamps de expiração

2. **Verificar Por Que Expirou**:

- Revisar log de eventos
- Verificar histórico de tentativas de entrega
- Verificar se o roteamento foi bem-sucedido

Soluções:

Estender Expiração:

```
# Adicionar 24 horas à expiração
curl -X PATCH https://api.example.com:8443/api/messages/12345 \
  -H "Content-Type: application/json" \
  -d '&amp;#123;"expires": "2025-10-31T12:00:00Z", "deadletter":
false&amp;#125;'
```

Problemas de Roteamento

Nenhuma Rota Encontrada

Sintomas:

- Erro: `no_route_found`
- Métrica `sms_c_routing_failed_count` aumentando
- Log de eventos mostra "routing_failed"

Passos de Diagnóstico:

1. **Verificar se as Rotas Existem**:

Acesse a Interface Web: `/sms_routing`

- Verificar se as rotas estão configuradas
- Verificar se pelo menos uma rota está habilitada

2. **Testar Roteamento**:

Acesse a Interface Web: `/simulator`

- Inserir parâmetros de mensagem (número chamador, número chamado, SMSC de origem)
- Revisar resultados de avaliação
- Verificar por que as rotas não corresponderam

3. **Verificar Critérios de Rota**:

- Prefixos correspondem ao necessário?

- Filtro de SMSC de origem muito restritivo?
- Todas as rotas desabilitadas?

Soluções:

Nenhuma Rota Configurada:

Adicionar rota catch-all:

```
Prefixo Chamador: (vazio)
Prefixo Chamado: (vazio)
SMSC de Origem: (vazio)
SMSC de Destino: default_gateway
Prioridade: 255
Peso: 100
Habilitado: ✓
Descrição: Rota padrão catch-all
```

Rotas Muito Específicas:

Adicionar rota mais ampla:

```
Prefixo Chamado: +
SMSC de Destino: international_gateway
Prioridade: 200
Peso: 100
Habilitado: ✓
Descrição: Catch-all internacional
```

Todas as Rotas Desabilitadas:

- Habilitar rotas apropriadas via Interface Web
- Verificar se a configuração não desabilitou acidentalmente as rotas

Rota Errada Selecionada

Sintomas:

- Mensagens roteadas para destino inesperado

- Gateway errado recebendo tráfego
- Balanceamento de carga não distribuindo como esperado

Passos de Diagnóstico:

1. **Usar Simulador de Roteamento**:

Acesse a Interface Web: `/simulator`

- Testar com parâmetros de mensagem reais
- Revisar seção "Todas as Correspondências"
- Verificar prioridades e pontuações de especificidade

2. **Verificar Prioridades de Rota**:

- Número menor = prioridade mais alta
- Rotas avaliadas em ordem de prioridade
- Dentro da mesma prioridade, pesos se aplicam

3. **Verificar Especificidade da Rota**:

Pontuação de especificidade:

- Prefixo chamado mais longo: +100 pontos por caractere
- Prefixo chamador mais longo: +50 pontos por caractere
- SMSC de origem especificado: +25 pontos
- Tipo de origem especificado: +10 pontos
- Domínio ENUM especificado: +15 pontos

Soluções:

Ajustar Prioridades:

Tornar rota específica de maior prioridade:

```
Rota Premium:
  Prefixo Chamado: +1555
  Prioridade: 10 (alta prioridade)

Rota Geral:
  Prefixo Chamado: +1
  Prioridade: 50 (prioridade mais baixa)
```

Ajustar Pesos:

Mudar distribuição de balanceamento de carga:

```
Primário (70%):
  Peso: 70

Backup (30%):
  Peso: 30
```

Adicionar Rota Mais Específica:

Substituir rota geral para caso específico:

```
Rota Específica:
  Prefixo Chamado: +15551234
  SMSC de Destino: dedicated_gateway
  Prioridade: 1

Rota Geral:
  Prefixo Chamado: +1
  SMSC de Destino: general_gateway
  Prioridade: 50
```

Resposta Automática Não Funcionando

Sintomas:

- Rota de resposta automática configurada, mas não acionando
- Nenhuma mensagem de resposta sendo enviada

- Log de eventos sem evento de resposta automática

Passos de Diagnóstico:

1. **Verificar Configuração da Rota**:

- `action: :auto_reply`
- `auto_reply_message` contém texto
- Rota está habilitada
- Rota corresponde aos critérios da mensagem

2. **Testar no Simulador**:

- Verificar se a rota é selecionada
- Verificar indicação de "auto_reply"

3. **Verificar Log de Eventos**:

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

Soluções:

Rota Não Correspondendo:

- Ampliar critérios (remover filtros)
- Verificar prioridade (deve ser maior que rotas normais)
- Verificar status habilitado

Mensagem Não Definida:

Editar rota, adicionar mensagem:

```
Resposta Automática: ✓
Mensagem de Resposta Automática: "Obrigado pela sua mensagem.
Responderemos em breve."
```

Prioridade Errada:

Rotas de resposta automática devem ter alta prioridade (número baixo):

```
Rota de Resposta Automática:
  Prioridade: 10

Rota Normal:
  Prioridade: 50
```

Problemas de Desempenho

Alta Latência de Processamento de Mensagens

Sintomas:

- `sms_c_message_processing_stop_duration` p95 > 1000ms
- Respostas lentas da API
- Fila se acumulando

Passos de Diagnóstico:

1. **Verificar Latências dos Componentes**:

```
# Latência de roteamento
histogram_quantile(0.95, sms_c_routing_stop_duration_bucket)

# Latência de pesquisa ENUM
histogram_quantile(0.95, sms_c_enum_lookup_stop_duration_bucket)

# Latência de cobrança
histogram_quantile(0.95, sms_c_charging_succeeded_duration_bucket)

# Latência de entrega
histogram_quantile(0.95, sms_c_delivery_succeeded_duration_bucket)
```

2. **Verificar Recursos do Sistema**:

```
# Uso da CPU
top -b -n 1 | grep sms_c

# Uso da memória
ps aux | grep beam.smp
```

Soluções:

Roteamento Lento (Muitas rotas):

- Reduzir número de rotas habilitadas
- Combinar rotas semelhantes
- Otimizar critérios de rota

Pesquisas ENUM Lentas:

- Verificar latência do servidor DNS
- Aumentar timeout
- Usar servidores DNS mais rápidos/próximos
- Desabilitar ENUM se não necessário

Cobrança Lenta:

- Verificar desempenho do OCS
- Aumentar timeout do OCS
- Desabilitar cobrança se não necessário
- Usar cobrança assíncrona

Banco de Dados Lento:

- Aumentar tamanho do pool de conexões
- Adicionar índices
- Otimizar consultas
- Atualizar recursos do banco de dados

Mudanças de Configuração:

```
# config/config.exs
# Aumentar tamanho do lote para throughput
config :sms_c,
  batch_insert_batch_size: 200,
  batch_insert_flush_interval_ms: 200

# Aumentar pool de banco de dados
config :sms_c, SmsC.Repo,
  pool_size: 50
```

Baixo Throughput de Mensagens

Sintomas:

- Processamento < 100 msg/sec
- Usando API assíncrona, mas ainda lenta
- Altos tempos de resposta da API

Passos de Diagnóstico:

1. **Verificar Trabalhador de Lote**:

```
# No console de produção (iex)
SmsC.Messaging.BatchInsertWorker.stats()
```

Procurar por:

- `current_queue_size` próximo ao máximo
- `flush_errors` > 0
- `last_flush_duration_ms` muito alto

2. **Verificar Gargalos**:

```
# Tempo de consulta do banco de dados
ecto_pools_query_time

# Tempo de fila do pool de conexões
ecto_pools_queue_time
```

Soluções:

Gargalo no Banco de Dados:

Aumentar tamanho do pool:

```
config :sms_c, SmsC.Repo,
  pool_size: 50  # Aumentar de 20
```

Configuração de Lote:

Ajustar para throughput:

```
config :sms_c,
  batch_insert_batch_size: 200,  # Lotes maiores
  batch_insert_flush_interval_ms: 200  # Intervalo mais longo
```

Usar Endpoint Assíncrono:

```
# Alto throughput: usar /create_async
curl -X POST
https://api.example.com:8443/api/messages/create_async

# NÃO: /api/messages (síncrono)
```

Acúmulo de Fila Crescendo

Sintomas:

- `sms_c_queue_size_pending` aumentando
- Idade da mensagem mais antiga aumentando

- Processamento não consegue acompanhar a taxa de entrada

Passos de Diagnóstico:

1. **Verificar Taxa de Entrada vs Entrega**:

```
# Taxa de entrada
rate(sms_c_message_received_count[5m])

# Taxa de entrega
rate(sms_c_delivery_succeeded_count[5m])
```

2. **Verificar Capacidade do Frontend**:

- Os frontends estão consultando com frequência suficiente?
- Os frontends estão processando mensagens rapidamente o suficiente?
- Algum erro no frontend?

3. **Verificar Taxa de Sucesso de Entrega**:

```
rate(sms_c_delivery_succeeded_count[5m]) /
rate(sms_c_delivery_attempted_count[5m])
```

Soluções:

Frontends Não Consultando:

- Verificar conectividade do frontend
- Verificar intervalo de polling (deve ser de 5 a 10 segundos)
- Reiniciar serviços do frontend

Frontends Muito Lentos:

- Adicionar mais instâncias de frontend
- Otimizar processamento do frontend
- Aumentar concorrência do frontend

Alta Taxa de Retentativas:

- Investigar falhas de entrega
- Corrigir problemas downstream
- Roteamento para gateways alternativos

Pico Temporário:

- Aguardar a fila esvaziar
- Monitorar até normal
- Considerar upgrades de capacidade se recorrente

Problemas de Banco de Dados

Falhas de Conexão

Sintomas:

- Erro: "unable to connect to database"
- API retornando erros 500
- Aplicação não inicia

Passos de Diagnóstico:

1. **Verificar Status do Banco de Dados SQL CDR**:

```
# MySQL/MariaDB
systemctl status mysql

# PostgreSQL
systemctl status postgresql

# Testar conectividade (MySQL/MariaDB)
mysql -u sms_user -p -h db.example.com -e "SELECT 1"

# Testar conectividade (PostgreSQL)
psql -U sms_user -h db.example.com -d sms_c_prod -c "SELECT 1"
```

2. **Verificar Rede**:

```
# Ping no host do banco de dados
ping db.example.com

# Verificar conectividade da porta (MySQL/MariaDB: 3306,
PostgreSQL: 5432)
telnet db.example.com 3306
# ou
telnet db.example.com 5432
```

3. **Verificar Credenciais**:

```
# Verificar variáveis de ambiente
echo $DB_USERNAME
echo $DB_HOSTNAME
echo $DB_PORT

# Tentar conexão manual com as mesmas credenciais (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
```

Soluções:

Banco de Dados Fora do Ar:

```
# Iniciar banco de dados (MySQL/MariaDB)
systemctl start mysql

# Iniciar banco de dados (PostgreSQL)
systemctl start postgresql
```

Credenciais Erradas:

Atualizar configuração:

```
export DB_USERNAME=correct_user
export DB_PASSWORD=correct_password

# Reiniciar aplicação
systemctl restart omnimessage
```

Problema de Rede:

- Verificar regras de firewall
- Verificar grupos de segurança (nuvem)
- Verificar conectividade VPN/rede

Pool de Conexões Exaurido:

Aumentar tamanho do pool:

```
config :sms_c, SmsC.Repo,
  pool_size: 50  # Aumentar do valor atual
```

Consultas Lentas

Sintomas:

- Tempo de consulta do banco de dados alto
- Respostas lentas da API
- Fila do pool de conexões se acumulando

Passos de Diagnóstico:

1. **Verificar Log de Consultas Lentas**:

```sql
-- MySQL/MariaDB: Habilitar log de consultas lentas
SET GLOBAL slow_query_log = 'ON';
SET GLOBAL long_query_time = 1;  -- Logar consultas > 1 segundo

-- Visualizar consultas lentas (MySQL/MariaDB)
SELECT * FROM mysql.slow_log ORDER BY query_time DESC LIMIT 10;

-- PostgreSQL: Habilitar log de consultas lentas em
postgresql.conf
-- log_min_duration_statement = 1000  # milissegundos
-- Então verificar logs do PostgreSQL
```

2. **Verificar Índices Ausentes**:

```sql
-- Verificar índices da tabela
SHOW INDEX FROM message_queues;

-- Índices esperados:
-- - source_smsc
-- - dest_smsc
-- - send_time
-- - inserted_at
```

3. **Verificar Estatísticas da Tabela**:

```sql
-- Tamanhos das tabelas (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';

-- Tamanhos das tabelas (PostgreSQL)
-- SELECT schemaname, tablename,
--
pg_size_pretty(pg_total_relation_size(schemaname||'.'||tablename))
AS size
-- FROM pg_tables WHERE schemaname = 'public';
```

Soluções:

Índices Ausentes:

```
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);
```

Fragmentação de Tabela:

```
-- MySQL/MariaDB
OPTIMIZE TABLE message_queues;
OPTIMIZE TABLE frontend_registrations;

-- PostgreSQL
-- VACUUM ANALYZE message_queues;
-- VACUUM ANALYZE frontend_registrations;
```

Demasiados Dados:

Limpar registros antigos:

```
-- Excluir mensagens entregues com mais de 30 dias
DELETE FROM message_queues
WHERE status = 'delivered'
AND deliver_time < DATE_SUB(NOW(), INTERVAL 30 DAY)
LIMIT 10000;
```

Espaço em Disco Cheio

Sintomas:

- Erro: "Disk full"
- Não é possível escrever no banco de dados

- Aplicação trava

Passos de Diagnóstico:

1. **Verificar Uso do Disco**:

```
df -h

# Verificar diretório do banco de dados SQL (MySQL/MariaDB)
du -sh /var/lib/mysql

# Verificar diretório do banco de dados SQL (PostgreSQL)
du -sh /var/lib/postgresql
```

2. **Encontrar Arquivos Grandes**:

```
# Encontrar maiores arquivos (MySQL/MariaDB)
find /var/lib/mysql -type f -exec du -h &amp;#123;&amp;#125; + | sort -rh
| head -20

# Encontrar maiores arquivos (PostgreSQL)
find /var/lib/postgresql -type f -exec du -h &amp;#123;&amp;#125; + | sort
-rh | head -20

# Verificar arquivos de log
du -sh /var/log/sms_c/*
```

Soluções:

Limpar Dados Antigos:

```
-- Excluir mensagens antigas
DELETE FROM message_queues
WHERE inserted_at < DATE_SUB(NOW(), INTERVAL 90 DAY)
LIMIT 100000;
```

Rotacionar Logs:

```
# Forçar logrotate
logrotate -f /etc/logrotate.d/sms_c

# Limpar arquivos de log antigos
find /var/log/sms_c -name "*.log.*" -mtime +30 -delete
```

Expandir Disco:

- Redimensionar volume (nuvem)
- Adicionar novo disco e estender volume
- Mover dados para disco maior

Problemas de Conexão com o Frontend

Frontend Não Aparecendo como Ativo

Sintomas:

- Status do frontend mostra "expired"
- Frontend não na lista ativa
- Mensagens não estão sendo entregues ao frontend

Passos de Diagnóstico:

1. **Verificar Registro**:

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

2. **Verificar Logs do Frontend**:

- O frontend está chamando `/api/frontends/register`?
- Algum erro na API?
- Frequência de registro (deve ser a cada 60s)

3. **Verificar Logs da API**:

```
grep "frontend.*register" /var/log/sms_c/application.log | tail
-20
```

Soluções:

Frontend Não Registrando:

Testar 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;'
```

Se bem-sucedido, o problema está no código/configuração do frontend.

Registro Expirando:

Frontends expiram após 90 segundos. Garantir registro a cada 60 segundos:

```
# O frontend deve chamar registrar a cada 60 segundos
while True:
    register_with_smsc()
    time.sleep(60)
```

Problemas de Rede:

- Verificar firewall entre frontend e API
- Verificar resolução de DNS
- Testar com curl do servidor do frontend

Frontend Conectando/Desconectando Repetidamente

Sintomas:

- Status do frontend alternando entre ativo/expirado
- Alta contagem de registros na história
- Conexão instável

Passos de Diagnóstico:

1. **Verificar Saúde do Frontend**:

- O processo do frontend está estável?
- Algum travamento ou reinício?
- Problemas de recursos (CPU/memória)?

2. **Verificar Estabilidade da Rede**:

```
# Verificar perda de pacotes
ping -c 100 api.example.com

# Verificar resets de conexão
netstat -s | grep -i reset
```

3. **Verificar Tempo de Registro**:

- Muito frequente? (a cada poucos segundos)
- Muito infrequente? (> 90 segundos)

Soluções:

Frontend Instável:

- Corrigir problemas da aplicação do frontend
- Aumentar recursos do frontend
- Verificar logs do frontend para erros

Problemas de Rede:

- Verificar conectividade intermitente
- Revisar logs de firewall
- Verificar verificações de saúde do balanceador de carga

Intervalo de Registro Errado:

Corrigir intervalo:

```
REGISTRATION_INTERVAL = 60  # segundos
```

```
O carregamento é realizado sobre **Diameter Ro** (Controle de
Crédito). Observe que o carregamento **falha abertamente**: um
erro OCS ou uma pilha Diameter não configurada registra uma falha,
mas ainda entrega a mensagem. Apenas uma resposta definitiva de
fora de balanço rejeita uma mensagem (status `:balance_rejected`).

**Sintomas**:
- `sms_c_charging_failed_count` aumentando
- O log de eventos mostra "charging_failed" ou "balance_rejected"
- Mensagens marcadas com status `:balance_rejected` (fora de
balanço), ou
  avisos de log sobre a pilha Diameter não estar habilitada

**Passos de Diagnóstico**:

1. **Verifique o estado do par Diameter Ro** (CER/CEA estabelecido
para o DRA/OCS):
```bash
# Em um console remoto
:diameter.service_info(:omnimessage, :connections)
```

2. **Confirme que a pilha está habilitada e que a aplicação `:ro` está registrada**:

```
grep -E "diameter_enabled|application_name: :ro"
config/runtime.exs
```
```

Uma linha de log `Charging requested but the Diameter Ro stack is not enabled/configured` significa que o carregamento está sendo pulado (mensagens permitidas) - habilite a pilha.

3. **Verifique se o perfil de classificação** corresponde ao contexto de classificação configurado no OCS:

```
grep rating_profile config/runtime.exs
```

Soluções:

OCS / DRA Inacessível:

- Verifique a acessibilidade do IP e se o DRA tem este nó provisionado como um par Diameter (Origin-Host/Realm).
- Confirme que o CCR `Service-Context-Id` (`rating_profile`) corresponde ao perfil de classificação SMS do OCS.

Tratamento de fora de balanço (`config :sms_c, :charging`):

```
config :sms_c, :charging,
  out_of_balance_action: :reject,    # ou :auto_reply
  out_of_balance_reply_text: "...",
  rating_profile: "sms@sms-c"
```

Desabilitar Carregamento Temporariamente:

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

Reinicie a aplicação.

Problemas de Conta:

- Verifique se a conta existe no OCS
- Verifique se a conta tem saldo
- Verifique se os planos de classificação estão configurados

Carregamento Muito Lento

Sintomas:

- `sms_c_charging_succeeded_duration` p95 > 500ms
- Processamento de mensagens lento quando o carregamento está habilitado
- Rápido quando o carregamento está desabilitado

Passos de Diagnóstico:

1. **Verifique a Latência do Carregamento**:

```
histogram_quantile(0.95, sms_c_charging_succeeded_duration_bucket)
```

2. **Verifique o desempenho do OCS / DRA** (viagem de ida e volta Diameter Ro). Alta `sms_c_charging_succeeded_duration` geralmente aponta para latência de classificação do OCS ou um par DRA lento/flutuante.

3. **Verifique a Latência da Rede para o DRA**:

```
# Ping no host DRA / OCS
ping -c 10 dra01.example.com
```

Soluções:

OCS Lento:

- Otimize a configuração do OCS
- Adicione recursos ao OCS
- Use um mecanismo de classificação mais rápido

Latência da Rede:

- Implemente o OCS mais próximo do SMS-C
- Use um caminho de rede direto
- Evite VPN/túneis se possível

Timeout Muito Baixo:

Aumente o timeout:

```
config :sms_c,
  ocs_timeout: 5000  # 5 segundos
```

Problemas de Busca ENUM

Buscas ENUM Falhando

Sintomas:

- `sms_c_enum_lookup_stop_duration` mostrando falhas
- O log de eventos mostra erros ENUM
- Rotas com `enum_result_domain` não correspondendo

Passos de Diagnóstico:

1. **Verifique a Configuração ENUM**:

```
grep -A 10 "enum_" config/runtime.exs
```

2. **Teste a Conectividade DNS**:

```
# Teste o servidor DNS
dig @8.8.8.8 e164.arpa

# Teste a 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 o Servidor DNS**:

```
# O DNS personalizado é acessível?
ping 10.0.1.53

# Teste a porta
nc -zv 10.0.1.53 53
```

Soluções:

Servidor DNS Inacessível:

Use 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
  ]
```

Domínio ENUM Errado:

Atualize o domínio:

```
config :sms_c,
  enum_domains: ["e164.arpa"]  # Use o domínio padrão
```

Timeout Muito Curto:

Aumente o timeout:

```
config :sms_c,
  enum_timeout: 10000  # 10 segundos
```

Desabilitar ENUM (se não necessário):

```
config :sms_c,
  enum_enabled: false
```

Problemas de Cache ENUM

Sintomas:

- Taxa de acerto de cache baixa (< 70%)
- Tamanho do cache crescendo sem limites
- Uso de memória alto

Passos de Diagnóstico:

1. **Verifique as Estatísticas do Cache**:

```
# Taxa de acerto do cache
rate(sms_c_enum_cache_hit_count[5m]) /
(rate(sms_c_enum_cache_hit_count[5m]) +
rate(sms_c_enum_cache_miss_count[5m]))

# Tamanho do cache
sms_c_enum_cache_size_size
```

2. **Verifique o Padrão de Tráfego**:

- Os números estão se repetindo?
- O TTL do cache é apropriado?

Soluções:

Baixa Taxa de Acerto (Esperado):

- Tráfego para números únicos (normal)
- Monitore, mas não alarme se < 70%

Cache Crescendo:

Limpe o cache através da página de Teste NAPTR ou reinicie a aplicação.

Alto Uso de Memória:

- Esperado com cache grande
- Monitore a memória total do sistema

- Considere ajustar o TTL

Problemas de Cluster

Nó Não Consegue Entrar no Cluster

Sintomas:

- Um único nó em execução
- Consultas de cluster retornando apenas resultados locais
- Erros de distribuição Erlang

Passos de Diagnóstico:

1. **Verifique os Nomes dos Nós**:

```
# No console IEx
Node.self()
# Esperado: :sms@node1.example.com

Node.list()
# Esperado: Lista de outros nós
```

2. **Verifique o Cookie Erlang**:

```
# Verifique o arquivo de cookie
cat ~/.erlang.cookie

# Verifique se é o mesmo em todos os nós
```

3. **Verifique a Rede**:

```
# Os nós conseguem se alcançar?
ping node2.example.com

# Verifique as portas
nc -zv node2.example.com 4369
nc -zv node2.example.com 9100-9200
```

Soluções:

Cookie Não Correspondente:

Defina o mesmo cookie em todos os nós:

```
export ERLANG_COOKIE=same_secret_value_here

# Ou atualize ~/.erlang.cookie
echo "same_secret_value_here" > ~/.erlang.cookie
chmod 400 ~/.erlang.cookie
```

Firewall Bloqueando:

Abra as portas necessárias:

```
# EPMD
iptables -A INPUT -p tcp --dport 4369 -j ACCEPT

# Distribuição Erlang
iptables -A INPUT -p tcp --dport 9100:9200 -j ACCEPT
```

Problemas de DNS:

Use endereços IP em vez de nomes de host:

```
config :sms_c,
  cluster_nodes: [
    :"sms@10.0.1.10",
    :"sms@10.0.1.11"
  ]
```

Cérebro Dividido do Cluster

Sintomas:

- Nós em execução, mas desconectados
- Dados diferentes em nós diferentes
- Inconsistências no Mnesia

Passos de Diagnóstico:

1. **Verifique a Conectividade dos Nós**:

```
# Em cada nó (IEx)
Node.list()
```

2. **Verifique o Mnesia**:

```
:mnesia.system_info(:running_db_nodes)
```

Soluções:

Reconectar Nós:

```
# Pare todos os nós
systemctl stop omnimessage

# Inicie um nó primeiro
systemctl start omnimessage  # No nó1

# Aguarde até que ele inicie completamente, então inicie os outros
systemctl start omnimessage  # No nó2
systemctl start omnimessage  # No nó3
```

Inconsistência no Mnesia:

- Exporte rotas do nó correto
- Pare todos os nós
- Exclua o diretório Mnesia

- Inicie os nós
- Importe rotas

Problemas de API

API Não Respondendo

Sintomas:

- Timeout de conexão
- Conexão recusada
- Sem resposta

Passos de Diagnóstico:

1. **Verifique o Processo da API**:

```
# A aplicação está em execução?
systemctl status omnimessage

# Verifique as portas em escuta
netstat -tlnp | grep 8443
```

2. **Verifique o Firewall**:

```
# Verifique iptables
iptables -L -n | grep 8443

# Teste a conectividade local
curl -k https://localhost:8443/api/status
```

3. **Verifique a Configuração TLS**:

```
# Verifique se o certificado existe
ls -l priv/cert/server.crt priv/cert/server.key

# Verifique a validade do certificado
openssl x509 -in priv/cert/server.crt -noout -dates
```

Soluções:

Aplicação Não Executando:

```
systemctl start omnimessage
```

Firewall Bloqueando:

```
# Permitir porta da API
iptables -A INPUT -p tcp --dport 8443 -j ACCEPT
```

Problemas de Certificado:

Gere um novo certificado (veja o Guia de Configuração).

Porta Errada:

Verifique a configuração:

```
grep "port:" config/runtime.exs
```

API Retornando Erros 500

Sintomas:

- Erro Interno do Servidor
- Código de status 500
- Erro nos logs

Passos de Diagnóstico:

1. **Verifique os Logs da Aplicação**:

```
tail -100 /var/log/sms_c/application.log | grep "\[error\]"
```

2. **Verifique o Banco de Dados**:

```
mysql -u sms_user -p -e "SELECT 1"
```

3. **Verifique os Recursos**:

```
# Memória
free -h

# CPU
top -b -n 1

# Disco
df -h
```

Soluções:

Banco de Dados Indisponível:

- Inicie o banco de dados
- Corrija a configuração de conexão

Fora de Memória:

- Reinicie a aplicação
- Aumente a memória do sistema
- Verifique por vazamentos de memória

Erro na Aplicação:

- Verifique o erro específico nos logs
- Corrija o problema de configuração
- Reinicie a aplicação

Problemas na Interface Web

Não Consigo Acessar a Interface Web

Sintomas:

- Timeout de conexão
- 404 Não Encontrado
- Página não carrega

Passos de Diagnóstico:

1. **Verifique o Status da Aplicação**:

```
systemctl status omnimessage
```

2. **Verifique a Porta**:

```
netstat -tlnp | grep 80
```

3. **Verifique a URL**:

- Nome do host correto?
- Porta correta?
- HTTP vs HTTPS?

Soluções:

Porta Errada:

Verifique a configuração:

```
grep "control_panel" config/runtime.exs
```

Acesse na porta correta (padrão: 80 ou 4000).

Aplicação Não Executando:

```
systemctl start omnimessage
```

Firewall:

```
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

LiveView Não Atualizando

Sintomas:

- Página carrega, mas não atualiza
- Dados estão desatualizados
- Erros de WebSocket no console do navegador

Passos de Diagnóstico:

1. **Verifique o Console do Navegador**:

- Abra as Ferramentas de Desenvolvedor (F12)
- Procure por erros de WebSocket
- Verifique a aba de rede para requisições falhadas

2. **Verifique a Configuração do Proxy**:

Se estiver usando proxy reverso, assegure suporte a WebSocket:

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

Soluções:

WebSocket Bloqueado:

- Configure o proxy para WebSocket
- Verifique o firewall
- Verifique extensões do navegador

Atualize a Página:

- Atualização forçada (Ctrl+F5)
- Limpe o cache do navegador

Problemas de Recursos do Sistema

Alto Uso de CPU

Sintomas:

- CPU consistentemente > 80%
- Sistema lento
- Aplicação não responsiva

Passos de Diagnóstico:

1. **Verifique o Processo**:

```
top -b -n 1 | grep beam.smp
```

2. **Verifique as Métricas**:

```
# Taxa de processamento de mensagens
rate(sms_c_message_received_count[5m])

# Operações de roteamento
rate(sms_c_routing_route_matched_count[5m])
```

Soluções:

Tráfego Alto:

- Escale horizontalmente (adicione nós)
- Escale verticalmente (adicione CPU)

Roteamento Ineficiente:

- Reduza o número de rotas
- Otimize os critérios de rota

Muitas Buscas ENUM:

- Verifique a taxa de acerto do cache
- Considere desabilitar se não necessário

Alto Uso de Memória

Sintomas:

- Uso de memória > 90%
- Aplicação trava
- Erros de falta de memória

Passos de Diagnóstico:

1. **Verifique a Memória**:

```
free -h

ps aux | grep beam.smp
```

2. **Verifique os Tamanhos do Cache**:

```
sms_c_enum_cache_size_size
```

Soluções:

Cache ENUM Muito Grande:

- Limpe o cache

- Reduza o TTL
- Desabilite ENUM se não necessário

Fila de Lotes Crescendo:

```
# Verifique as estatísticas do trabalhador (IEx)
SmsC.Messaging.BatchInsertWorker.stats()
```

Se a fila estiver grande, limpe manualmente ou reinicie.

Adicionar Memória:

- Escale verticalmente
- Adicione swap (temporariamente)

Vazamento de Memória:

- Reinicie a aplicação
- Relate o problema para investigação

Para assistência adicional, consulte:

- Guia de Operações - Procedimentos diários
- Guia de Configuração - Opções de configuração
- Guia de Métricas - Configuração de monitoramento
- Logs da aplicação - `/var/log/sms_c/application.log`