Referência de Configuração

Guia completo para todos os parâmetros de configuração

Visão Geral da Arquitetura

O Gateway SMPP OmniMessage é um **frontend de protocolo sem estado** que traduz mensagens SMPP para/de OmniMessage. Toda a lógica de negócios, decisões de roteamento e armazenamento de mensagens são gerenciados pelo OmniMessage Core - o gateway simplesmente:

1. **Recebe** PDUs SMPP de operadoras e clientes
2. **Traduz** para o formato OmniMessage via REST API
3. **Consulta** o OmniMessage por mensagens a serem enviadas
4. **Envia** PDUs SMPP para operadoras
5. **Relata** o status de entrega de volta ao OmniMessage

Isso é idêntico a como outros frontends do OmniMessage (Diameter, MAP, IMS) funcionam - todos são tradutores de protocolo sem estado que delegam ao OmniMessage Core.

Localização do Arquivo de Configuração

`/opt/omnimessage-smpp/config/runtime.exs`

Importante: Após alterar a configuração, reinicie o gateway:

```
sudo systemctl restart omnimessage-smpp
```

Estrutura da Configuração

O arquivo de configuração usa a sintaxe Elixir. Estrutura básica:

```
import Config

# Configurações globais
config :omnimessage_smpp,
  setting_name: value

# Bindings SMPP
config :omnimessage_smpp, :binds, [
  %{
    name: "bind_name",
    # ... configurações de bind
  }
]
```

Configurações Globais

API_BASE_URL

URL da plataforma OmniMessage Core

```
config :omnimessage_smpp,
  api_base_url: "https://omnimessage-core.example.com:8443"
```

Parâmetro	Tipo	Requerido	Padrão
`api_base_url`	String (URL)	Sim	-

Propósito: URL da plataforma OmniMessage Core. O gateway se comunica com o OmniMessage via REST API para todo o processamento de mensagens:

- **Enviar Mensagens**: Enviar mensagens SMPP recebidas para o OmniMessage para processamento
- **Recuperar Mensagens**: Consultar mensagens destinadas às operadoras SMPP
- **Relatar Status de Entrega**: Atualizar o status de entrega da mensagem de volta ao OmniMessage
- **Saúde do Sistema**: Verificações de saúde periódicas

Crítico: É aqui que o gateway obtém toda a sua "inteligência". O OmniMessage gerencia:

- ✓ Validação de mensagens e verificação de formato
- ✓ Decisões de roteamento (qual operadora usar)
- ✓ Limitação de taxa e controle de fluxo
- ✓ Validação de número
- ✓ Armazenamento e persistência de mensagens
- ✓ Lógica de tentativas de entrega
- ✓ Rastreamento de status

O gateway simplesmente traduz o formato SMPP ↔ OmniMessage.

Exemplos:

```
# HTTPS com IP
api_base_url: "https://192.168.1.100:8443"

# HTTPS com nome de host
api_base_url: "https://omnimessage-core.company.com:8443"

# HTTP (não recomendado para produção)
api_base_url: "http://192.168.1.100:8080"
```

Requisitos de Rede:

- O gateway deve ter acesso à rede do OmniMessage Core
- Use HTTPS em produção (configure `verify_ssl_peer`)
- O firewall deve permitir HTTPS de saída na porta especificada

SMPP_POLL_INTERVAL

Frequência de verificação da fila (milissegundos)

```
config :omnimessage_smpp,
  smpp_poll_interval: 100
```

Parâmetro	Tipo	Requerido	Padrão
`smpp_poll_interval`	Inteiro	Não	100

Propósito: Com que frequência (em milissegundos) cada cliente verifica a fila de mensagens.

Diretrizes:

- **Alto volume (>100 TPS)**: 100-500ms
- **Volume médio (10-100 TPS)**: 500-1000ms
- **Baixo volume (<10 TPS)**: 1000-2000ms

Variável de ambiente: `SMPP_POLL_INTERVAL`

VERIFY_SSL_PEER

Verificação de certificado SSL

```
config :omnimessage_smpp,
  verify_ssl_peer: false
```

Parâmetro	Tipo	Requerido	Padrão
`verify_ssl_peer`	Booleano	Não	false

Propósito: Se deve verificar os certificados SSL ao conectar-se à API de backend.

Valores:

- `true`: Verificar certificados (produção com certificados válidos)
- `false`: Ignorar verificação (certificados autoassinados, teste)

Variável de ambiente: `VERIFY_SSL_PEER`

SMSC_NAME

Identificador do gateway para registro

```
config :omnimessage_smpp,
  smsc_name: "smpp_gateway"
```

Parâmetro	Tipo	Requerido	Padrão
`smsc_name`	String	Não	"smpp_gateway"

Propósito: Identifica esta instância de gateway no backend da fila de mensagens.

Variável de ambiente: `SMSC_NAME`

Configuração de Bind do Cliente SMPP

Bindings de cliente são **conexões de saída** onde o gateway atua como um **ESME** (cliente) conectando-se aos servidores **SMSC** da operadora. Neste modo, o gateway inicia a conexão para enviar e receber mensagens através de operadoras externas.

Exemplo Completo de Bind do Cliente

```
config :omnimessage_smpp, :binds, [
  %{
    # Identificador único para esta conexão
    name: "vodafone_uk",

    # Modo de conexão
    mode: :client,

    # Tipo de bind SMPP
    bind_type: :transceiver,

    # Endereço do servidor SMPP da operadora
    host: "smpp.vodafone.co.uk",
    port: 2775,

    # Credenciais de autenticação
    system_id: "your_username",
    password: "your_password",

    # Campos do protocolo SMPP (opcional, definir se a operadora
exigir)
    system_type: "",
    addr_ton: 0,
    addr_npi: 0,
    address_range: "",

    # Limitação de taxa
    tps_limit: 100,

    # Frequência de verificação da fila
    queue_check_frequency: 1000,

    # Intervalo de keepalive (segundos, 0 para desativar)
    enquire_link_interval: 60,

    # Cache de mensagens (opcional)
    cache_enabled: false,
    cache_max_size: 10000,
    cache_retry_interval: 60
```

```
  }
]
```

Parâmetros de Bind do Cliente

name

Identificador único da conexão

Tipo	Requerido	Exemplo
String	Sim	`"vodafone_uk"`

Propósito: Identifica exclusivamente esta conexão SMPP.

- Usado em logs e métricas
- Deve ser único entre todos os binds
- Use nomes descritivos (operadora, região, propósito)

Convenções de nomenclatura:

- `operadora_região`: `"vodafone_uk"`, `"att_us"`
- `número_propósito`: `"marketing_1"`, `"alerts_primary"`

mode

Tipo de conexão

Tipo	Requerido	Valor
Átomo	Sim	`:client`

Propósito: Define isso como uma conexão de saída onde o gateway atua como um **ESME** conectando-se a um **SMSC** externo.

Valor fixo: Sempre `:client` para conexões de saída.

bind_type

Tipo de sessão SMPP

Tipo	Requerido	Valores Permitidos
Átomo	Sim	`:transmitter`, `:receiver`, `:transceiver`

Propósito: Define a capacidade de direção da mensagem.

Opções:

- `:transmitter` - Enviar mensagens apenas (submit_sm)
- `:receiver` - Receber mensagens apenas (deliver_sm)
- `:transceiver` - Enviar e receber (mais comum)

Recomendação: Use `:transceiver` a menos que a operadora exija um tipo específico.

host

Nome do host ou IP do servidor SMPP da operadora

Tipo	Requerido	Exemplo
String	Sim	`"smpp.carrier.com"` ou `"10.5.1.100"`

Propósito: Endereço do servidor SMPP da operadora.

Exemplos:

```
host: "smpp.vodafone.co.uk"
host: "10.20.30.40"
host: "smpp-primary.carrier.net"
```

port

Porta do servidor SMPP

Tipo	Requerido	Padrão	Faixa
Inteiro	Sim	2775	1-65535

Propósito: Porta TCP para a conexão SMPP.

Porta padrão: 2775

Exemplos:

```
port: 2775  # Padrão
port: 3000  # Personalizada
```

system_id

Nome de usuário de autenticação

Tipo	Requerido	Exemplo
String	Sim	`"company_user"`

Propósito: Nome de usuário fornecido pela operadora para autenticação.

Segurança: Proteja esta credencial - armazenada no arquivo de configuração.

password

Senha de autenticação

Tipo	Requerido	Exemplo
String	Sim	`"secret_password"`

Propósito: Senha fornecida pela operadora para autenticação.

Segurança:

- Proteja esta credencial
- Use senhas fortes
- Altere periodicamente

tps_limit

Limite de transações por segundo

Tipo	Requerido	Padrão	Faixa
Inteiro	Sim	100	1-10000

Propósito: Número máximo de mensagens por segundo a serem enviadas através desta conexão.

Diretrizes:

- Defina para 70-80% do máximo da operadora
- Previna controle de fluxo/desconexão
- Permite margem para recibos de entrega

Exemplos:

```
tps_limit: 10    # Baixo volume
tps_limit: 50    # Volume médio
tps_limit: 100   # Alto volume (mais comum)
tps_limit: 1000  # Volume muito alto
```

Cálculo:

```
Se o máximo da operadora = 100 TPS
Defina tps_limit = 70-80
Deixa 20-30 TPS de margem
```

queue_check_frequency

Intervalo de polling da fila de mensagens (milissegundos)

Tipo	Requerido	Padrão	Faixa
Inteiro	Sim	1000	100-10000

Propósito: Com que frequência verificar o backend por novas mensagens a serem enviadas.

Diretrizes:

- **Alto volume (>100 TPS)**: 500-1000ms
- **Volume médio (10-100 TPS)**: 1000-2000ms
- **Baixo volume (<10 TPS)**: 2000-5000ms

Compensações:

- Valor mais baixo = coleta de mensagens mais rápida, mais carga na API
- Valor mais alto = coleta mais lenta, menos carga na API

enquire_link_interval

Intervalo de keepalive SMPP (segundos)

Tipo	Requerido	Padrão	Faixa
Inteiro	Não	60	0-3600

Propósito: Com que frequência (em segundos) enviar PDUs enquire_link SMPP para verificar se a conexão está ativa. O servidor remoto responde com enquire_link_resp.

Diretrizes:

- **Padrão (60)**: Adequado para a maioria das operadoras
- **Valores mais baixos (15-30)**: Detecção de falhas mais rápida, mais tráfego
- **Valores mais altos (120-300)**: Menos sobrecarga, detecção de falhas mais lenta

- **0**: Desativa completamente o enquire_link (não recomendado)

Exemplos:

```
enquire_link_interval: 60   # Padrão (1 minuto)
enquire_link_interval: 30   # Keepalive agressivo
enquire_link_interval: 0    # Desativado
```

system_type

Identificador do tipo de sistema SMPP

Tipo	Requerido	Padrão	Exemplo
String	Não	`""`	`"OTP"`

Propósito: Campo do protocolo SMPP enviado durante o bind. Algumas operadoras exigem um valor específico. Deixe em branco, a menos que a operadora especifique um.

addr_ton

Tipo de Número de Endereço

Tipo	Requerido	Padrão	Faixa
Inteiro	Não	0	0-6

Propósito: Campo do protocolo SMPP que especifica o tipo de número usado na solicitação de bind.

Valores comuns:

- `0` - Desconhecido
- `1` - Internacional
- `2` - Nacional
- `5` - Alfanumérico

Defina conforme exigido pela operadora.

addr_npi

Indicador do Plano de Numeração de Endereço

Tipo	Requerido	Padrão	Faixa
Inteiro	Não	0	0-18

Propósito: Campo do protocolo SMPP que especifica o plano de numeração na solicitação de bind.

Valores comuns:

- `0` - Desconhecido
- `1` - ISDN/E.164
- `3` - Dados/X.121
- `9` - Privado

Defina conforme exigido pela operadora.

address_range

Faixa de endereços para bind

Tipo	Requerido	Padrão	Exemplo
String	Não	`""`	`"614*"`

Propósito: Campo do protocolo SMPP que especifica a faixa de endereços que este bind manipula. Usado por algumas operadoras para filtrar quais mensagens são entregues a esta conexão. Deixe em branco, a menos que a operadora especifique um valor.

enabled

Estado de ativação do peer

Tipo	Requerido	Padrão
Booleano	Não	`true`

Propósito: Controla se este peer está ativo. Peers desativados são mantidos na configuração, mas não estabelecem conexões. Útil para retirar temporariamente uma conexão do ar sem excluir sua configuração.

cache_enabled

Ativar cache local de mensagens

Tipo	Requerido	Padrão
Booleano	Não	`false`

Propósito: Quando ativado, mensagens recebidas são armazenadas localmente se a API de backend estiver inacessível, e depois entregues automaticamente quando a conectividade for restaurada. Veja MESSAGE_CACHE.md para detalhes completos.

cache_max_size

Número máximo de mensagens em cache

Tipo	Requerido	Padrão	Faixa
Inteiro	Não	10000	1-1000000

Propósito: Número máximo de mensagens a serem armazenadas em cache por bind. Quando o limite é atingido, as mensagens mais antigas são removidas (FIFO). Aplica-se apenas quando `cache_enabled` é `true`.

cache_retry_interval

Intervalo base de nova tentativa (segundos)

Tipo	Requerido	Padrão
Inteiro	Não	60

Propósito: Intervalo base em segundos antes de tentar novamente a entrega de uma mensagem em cache. Combinado com backoff exponencial (nova tentativa 0: 60s, nova tentativa 1: 120s, nova tentativa 2: 240s, etc.). Aplica-se apenas quando `cache_enabled` é `true`.

Exemplo de UI Web:

Configuração de Bind do Servidor SMPP

Bindings de servidor definem **conexões de entrada** onde o gateway atua como um **SMSC** (servidor) aceitando conexões de **ESMEs** externas (clientes). Neste modo, sistemas parceiros se conectam ao gateway para enviar e receber mensagens.

Exemplo Completo de Bind do Servidor

```
config :omnimessage_smpp, :server_binds, [
  %{
    # Identificador único para este cliente
    name: "partner_acme",

    # Credenciais esperadas do cliente
    system_id: "acme_corp",
    password: "acme_secret",

    # Tipos de bind permitidos
    allowed_bind_types: [:transmitter, :receiver, :transceiver],

    # Restrições de IP
    ip_whitelist: ["192.168.1.0/24", "10.50.1.100"],

    # Restrições de endereço de origem (vazio = permitir todos)
    source_address_whitelist: [],

    # Limitação de taxa
    tps_limit: 50,

    # Frequência de verificação da fila
    queue_check_frequency: 1000,

    # Intervalo de keepalive (segundos, 0 para desativar)
    enquire_link_interval: 60,

    # Cache de mensagens (opcional)
    cache_enabled: false,
    cache_max_size: 10000,
    cache_retry_interval: 60
  }
]
```

Parâmetros de Bind do Servidor

name

Identificador do cliente

Tipo	Requerido	Exemplo
String	Sim	`"partner_acme"`

Propósito: Identifica o cliente externo que está se conectando a você.

Convenções de nomenclatura: Use o nome do parceiro/cliente para fácil identificação.

system_id

Nome de usuário esperado do cliente

Tipo	Requerido	Exemplo
String	Sim	`"acme_corp"`

Propósito: Nome de usuário que o cliente externo deve fornecer para autenticação.

Fornecer ao cliente: Compartilhe esta credencial com seu parceiro.

password

Senha esperada do cliente

Tipo	Requerido	Exemplo
String	Sim	`"secure_password"`

Propósito: Senha que o cliente externo deve fornecer para autenticação.

Segurança:

- Use senhas fortes
- Única por cliente
- Compartilhe de forma segura com o parceiro

allowed_bind_types

Tipos de sessão permitidos

Tipo	Requerido	Padrão
Lista de Átomos	Sim	-

Propósito: Restringe quais tipos de bind o cliente pode usar.

Opções:

```
allowed_bind_types: [:transceiver]  # Apenas transceiver
allowed_bind_types: [:transmitter, :receiver]  # TX ou RX
allowed_bind_types: [:transmitter, :receiver, :transceiver]  #
Qualquer
```

Recomendação: Permita os três, a menos que precise de restrições.

ip_whitelist

Endereços IP de cliente permitidos

Tipo	Requerido	Padrão	Formato
Lista de Strings	Sim	`[]`	IPs ou notação CIDR

Propósito: Segurança - permitir apenas conexões de IPs conhecidos.

Formatos:

- IP único: `"192.168.1.100"` (automaticamente `/32`)
- Sub-rede CIDR: `"192.168.1.0/24"`, `"10.0.0.0/8"`
- Mistura de ambos: `["192.168.1.0/24", "10.50.1.100"]`

Exemplos:

```
# Permitir qualquer IP (não recomendado)
ip_whitelist: []

# IP único
ip_whitelist: ["203.0.113.50"]

# Múltiplos IPs
ip_whitelist: ["203.0.113.50", "203.0.113.51"]

# Sub-rede
ip_whitelist: ["192.168.1.0/24"]

# Misturado
ip_whitelist: ["192.168.1.0/24", "10.50.1.100", "10.60.0.0/16"]
```

Sub-redes comuns:

- `/32` - IP único (automático para IPs sem máscara)
- `/24` - 256 endereços (ex: 192.168.1.0-255)
- `/16` - 65.536 endereços (ex: 10.50.0.0-255.255)
- `/8` - 16.777.216 endereços (ex: 10.0.0.0-255.255.255.255)

source_address_whitelist

Endereços de origem permitidos

Tipo	Requerido	Padrão	Formato
Lista de Strings	Não	`[]`	Padrões exatos ou curinga

Propósito: Restringe quais endereços de origem (IDs de remetente) os clientes conectados podem usar ao enviar mensagens. Lista vazia permite todos os endereços.

Tipos de padrões:

- Correspondência exata: `"MyBrand"` corresponde apenas a "MyBrand"
- Sufixo curinga: `"614*"` corresponde a qualquer endereço que comece com "614"

Exemplos:

```
# Permitir qualquer endereço de origem
source_address_whitelist: []

# Apenas endereços específicos
source_address_whitelist: ["MyBrand", "AlertService"]

# Correspondência de prefixo curinga
source_address_whitelist: ["614*", "+61*"]

# Misturado
source_address_whitelist: ["MyBrand", "614*", "+61400000001"]
```

Mensagens com endereços de origem não permitidos são rejeitadas com `ESME_RINVSRCADR`. Veja SOURCE_ADDRESS_WHITELIST.md para detalhes completos.

tps_limit

Limite de mensagens por segundo

Igual ao `tps_limit` do bind do cliente - controla a taxa de deliver_sm para os clientes conectados.

queue_check_frequency

Intervalo de polling da fila

Igual ao `queue_check_frequency` do bind do cliente - com que frequência verificar mensagens a serem entregues a este cliente.

enquire_link_interval

Intervalo de keepalive SMPP (segundos)

Igual ao `enquire_link_interval` do bind do cliente. Controla com que frequência o servidor envia PDUs enquire_link para os clientes conectados para verificar se eles ainda estão ativos.

enabled

Estado de ativação do peer

Igual ao `enabled` do bind do cliente. Peers de servidor desativados não aceitam conex��es de entrada.

cache_enabled

Ativar cache local de mensagens

Igual ao `cache_enabled` do bind do cliente. Veja MESSAGE_CACHE.md.

cache_max_size

Número máximo de mensagens em cache

Igual ao `cache_max_size` do bind do cliente.

cache_retry_interval

Intervalo base de nova tentativa (segundos)

Igual ao `cache_retry_interval` do bind do cliente.

Exemplo de UI Web:

Configuração de Escuta do Servidor

Quando os binds do servidor estão configurados, o gateway escuta por conexões de entrada.

Exemplo Completo de Escuta

```
config :omnimessage_smpp, :listen, %{
  host: "0.0.0.0",
  port: 2775,
  max_connections: 100
}
```

Parâmetros de Escuta

host

Endereço IP para vincular

Tipo	Requerido	Padrão	Valores Comuns
String	Não	`"0.0.0.0"`	`"0.0.0.0"`, `"127.0.0.1"`

Propósito: Em qual interface de rede escutar.

Valores:

- `"0.0.0.0"` - Escutar em todas as interfaces (recomendado)
- `"127.0.0.1"` - Escutar apenas no localhost (teste)
- `"192.168.1.10"` - Escutar em IP específico

port

Porta TCP para escutar

Tipo	Requerido	Padrão	Faixa
Inteiro	Não	2775	1-65535

Propósito: Porta para conexões SMPP de entrada.

Padrão: 2775

max_connections

Número máximo de conexões simultâneas

Tipo	Requerido	Padrão	Faixa
Inteiro	Não	100	1-10000

Propósito: Limita o número total de conexões de cliente simultâneas.

Diretrizes:

- Defina com base nos clientes esperados
- Valores mais altos usam mais memória
- Típico: 10-100 conexões

Exemplos Completos de Configuração

Exemplo 1: Conexão de Operadora Única

```
import Config

config :omnimessage_smpp,
  api_base_url: "https://smsc.company.com:8443",
  verify_ssl_peer: true,
  smsc_name: "smpp_prod"

config :omnimessage_smpp, :binds, [
  %{
    name: "att_primary",
    mode: :client,
    bind_type: :transceiver,
    host: "smpp.att.com",
    port: 2775,
    system_id: "company_user",
    password: "secure_pass_123",
    tps_limit: 100,
    queue_check_frequency: 1000
  }
]
```

Exemplo 2: Múltiplas Operadoras

```
import Config

config :omnimessage_smpp,
  api_base_url: "https://smsc.company.com:8443"

config :omnimessage_smpp, :binds, [
  # América do Norte
  %{
    name: "att_us",
    mode: :client,
    bind_type: :transceiver,
    host: "smpp.att.com",
    port: 2775,
    system_id: "att_username",
    password: "att_password",
    tps_limit: 100,
    queue_check_frequency: 1000
  },

  # Europa
  %{
    name: "vodafone_uk",
    mode: :client,
    bind_type: :transceiver,
    host: "smpp.vodafone.co.uk",
    port: 2775,
    system_id: "voda_username",
    password: "voda_password",
    tps_limit: 50,
    queue_check_frequency: 1000
  }
]
```

Exemplo 3: Gateway com Binds de Servidor

```
import Config

config :omnimessage_smpp,
  api_base_url: "https://smsc.company.com:8443"

# Conexões de saída
config :omnimessage_smpp, :binds, [
  %{
    name: "upstream_carrier",
    mode: :client,
    bind_type: :transceiver,
    host: "smpp.carrier.com",
    port: 2775,
    system_id: "my_username",
    password: "my_password",
    tps_limit: 100,
    queue_check_frequency: 1000
  }
]

# Definições de clientes de entrada
config :omnimessage_smpp, :server_binds, [
  %{
    name: "partner_alpha",
    system_id: "alpha_corp",
    password: "alpha_secret",
    allowed_bind_types: [:transmitter, :receiver, :transceiver],
    ip_whitelist: ["203.0.113.0/24"],
    tps_limit: 50,
    queue_check_frequency: 1000
  },
  %{
    name: "partner_beta",
    system_id: "beta_inc",
    password: "beta_password",
    allowed_bind_types: [:transceiver],
    ip_whitelist: ["198.51.100.50"],
    tps_limit: 25,
    queue_check_frequency: 2000
  }
]
```

```
# Escuta do servidor
config :omnimessage_smpp, :listen, %{
  host: "0.0.0.0",
  port: 2775,
  max_connections: 100
}
```

Validação da Configuração

Após editar a configuração, valide antes de reiniciar:

Verificação de Sintaxe

```
# Verifique a sintaxe Elixir
/opt/omnimessage-smpp/bin/omnimessage-smpp eval "File.read!
('config/runtime.exs')"
```

Se a sintaxe for inválida, você verá um erro. Corrija antes de reiniciar.

Testar Configuração

```
# Reinicie em primeiro plano para ver erros
sudo -u omnimessage-smpp /opt/omnimessage-smpp/bin/omnimessage-
smpp console
```

Pressione `Ctrl+C` duas vezes para sair.

Variáveis de Ambiente

Todas as configurações globais podem ser substituídas por variáveis de ambiente. Defina estas em seu arquivo de unidade systemd ou ambiente de shell antes de iniciar o gateway.

Variável de Ambiente	Chave de Configuração	Padr
`API_BASE_URL`	`api_base_url`	`"https://10.17`
`SMSC_NAME`	`smsc_name`	`"smpp_gateway"`
`SMPP_POLL_INTERVAL`	`smpp_poll_interval`	`100`
`VERIFY_SSL_PEER`	`verify_ssl_peer`	`false`
`CACHE_FLUSH_INTERVAL`	`cache_flush_interval`	`10000`
`CACHE_MAX_RETRY_ATTEMPTS`	`cache_max_retry_attempts`	`10`
`CACHE_BACKOFF_MULTIPLIER`	`cache_backoff_multiplier`	`2`

Variável de Ambiente	Chave de Configuração	Padr
`MNESIA_STORAGE_TYPE`	`mnesia_storage_type`	`disc_copies`

Exemplo de substituição systemd:

```
sudo systemctl edit omnimessage-smpp
```

```
[Service]
Environment="API_BASE_URL=https://omnimessage-
core.company.com:8443"
Environment="SMSC_NAME=smpp_prod_01"
Environment="VERIFY_SSL_PEER=true"
```

Melhores Práticas de Segurança

1. **Proteja o arquivo de configuração**:

   ```
   sudo chmod 600 /opt/omnimessage-smpp/config/runtime.exs
   sudo chown omnimessage-smpp:omnimessage-smpp /opt/omnimessage-
   smpp/config/runtime.exs
   ```

2. **Use senhas fortes**:

   - Mínimo de 12 caracteres
   - Misture letras, números, símbolos

- Única por conexão

3. **Use listas brancas de IP**:
   - Sempre configure `ip_whitelist` para binds de servidor
   - Nunca use lista vazia `[]` em produção
4. **Ative a verificação SSL**:
   - Defina `verify_ssl_peer: true` com certificados válidos
5. **Rotação regular de credenciais**:
   - Altere senhas trimestralmente
   - Coordene com operadoras/parceiros

Próximos Passos

- Revise MONITORING.md para configuração de métricas
- Leia USAGE.md para gerenciar conexões
- Veja TROUBLESHOOTING.md para problemas comuns
- Volte para README.md para visão geral

Glossário

Termos e Definições

A

API (Interface de Programação de Aplicações) Interface usada para se comunicar com o sistema de backend da fila de mensagens.

Auto-Scroll Recurso na aba de Logs da interface web que rola automaticamente para mostrar as entradas de log mais recentes.

B

Backend O sistema de fila de mensagens ao qual o Gateway SMPP se conecta para recuperar e armazenar mensagens.

Bind Uma conexão SMPP entre dois sistemas. Pode ser transmissor, receptor ou transceptor.

Bind Type O tipo de sessão SMPP:

- **Transmissor**: Enviar mensagens apenas
- **Receptor**: Receber mensagens apenas
- **Transceptor**: Enviar e receber mensagens

Bind Failure Quando uma tentativa de autenticação SMPP falha, geralmente devido a credenciais incorretas ou restrições de IP.

C

CIDR (Roteamento Inter-Domínio Sem Classe) Notação para especificar intervalos de endereços IP (por exemplo, `192.168.1.0/24` representa 256 endereços IP).

Client Bind Uma conexão SMPP de saída onde o gateway atua como um **ESME** conectando-se a um **SMSC** externo (tipicamente um servidor SMPP de um operador). Neste modo, o gateway é o cliente.

Connection Status Estado atual de um bind SMPP:

- **Conectado**: Ativo e operacional
- **Desconectado**: Não conectado
- **Reconectando**: Tentando estabelecer conexão

Counter Uma métrica que só aumenta (reinicia na reinicialização do serviço), usada para totais como mensagens enviadas.

D

Data Coding Campo SMPP que especifica a codificação de caracteres da mensagem (GSM-7, UCS-2, etc.).

Deliver_SM PDU SMPP enviada por um SMSC (servidor) para entregar uma mensagem a um ESME (cliente) conectado. Usado por binds de servidor para enviar mensagens a parceiros conectados.

Delivery Failure Quando uma mensagem não pode ser entregue, indicado por uma resposta de erro do operador.

Delivery Receipt (DLR) Confirmação do operador sobre o status da entrega da mensagem.

dest_smsc Campo na fila de mensagens indicando qual conexão SMPP deve lidar com a mensagem.

Disconnection Quando uma conexão SMPP ativa é encerrada, seja intencionalmente ou devido a erro.

E

Enquire Link Mensagem de keepalive SMPP enviada periodicamente para verificar se a conexão está ativa.

ESM Class Campo SMPP que indica o tipo e os recursos da mensagem.

ESME (Entidade de Mensagem Curta Externa) Na terminologia SMPP, a aplicação cliente que se conecta a um SMSC para enviar ou receber mensagens. Quando o gateway opera em **Modo Cliente**, ele atua como um ESME conectando-se a SMSCs de operadores. Quando opera em **Modo Servidor**, aceita conexões de ESMEs externos.

Exponential Backoff Estratégia de tentativa onde o tempo de espera dobra após cada falha (1min, 2min, 4min, 8min...).

F

Firewall Sistema de segurança de rede que controla o tráfego de rede de entrada e saída.

Frontend Registration Processo pelo qual o gateway SMPP se registra com o OmniMessage Core. Um heartbeat é enviado a cada 60 segundos para manter o registro ativo. Se o gateway parar, o registro expira após 90 segundos e o OmniMessage para de rotear mensagens para ele.

G

Gateway A aplicação Gateway SMPP que faz a ponte entre a fila de mensagens e as redes móveis.

Gauge Uma métrica que pode aumentar ou diminuir, representando o valor atual (por exemplo, status da conexão).

Grafana Ferramenta de visualização popular para exibir métricas do Prometheus em painéis.

GSM-7 Codificação de caracteres padrão de 7 bits para SMS, suportando até 160 caracteres por mensagem.

H

HTTP/HTTPS Protocolos usados para comunicação web. HTTPS é a versão criptografada.

I

IP Whitelist Lista de endereços IP permitidos que podem se conectar ao gateway (recurso de segurança).

ISDN (Rede Digital de Serviços Integrados) Plano de numera��ão comumente usado para números de telefone.

J

(Sem termos)

K

Keepalive Mensagens periódicas (enquire_link) enviadas para manter a conexão e detectar falhas.

KPI (Indicador Chave de Desempenho) Valor mensurável que indica o desempenho do sistema (por exemplo, taxa de sucesso de entrega).

L

Label No Prometheus, pares chave-valor anexados a métricas para identificação (por exemplo, `bind_name="vodafone_uk"`).

LiveView Tecnologia do framework Phoenix usada para atualizações em tempo real da interface web.

M

Message Queue Sistema de backend que armazena mensagens aguardando para serem enviadas ou recebidas.

Metrics Medições quantitativas do desempenho do sistema, expostas no formato Prometheus.

MO (Origem Móvel) Mensagens enviadas de telefones móveis para o gateway (entrada).

MT (Terminado Móvel) Mensagens enviadas do gateway para telefones móveis (saída).

MSISDN (Número de Diretório Internacional de Assinante de Estação Móvel) Formato padrão para números de telefone móvel.

N

NPI (Indicador de Plano de Numeração) Campo SMPP que especifica o esquema de numeração (por exemplo, ISDN).

O

Outbound Mensagens fluindo do gateway para as redes móveis.

Inbound Mensagens fluindo das redes móveis para o gateway.

P

PDU (Unidade de Dados de Protocolo) Pacote de mensagem SMPP individual (por exemplo, submit_sm, deliver_sm).

Prometheus Sistema de monitoramento de código aberto que coleta e armazena métricas de séries temporais.

Q

Queue Lista de mensagens aguardando para serem processadas ou enviadas.

Queue Check Frequency Com que frequência (em milissegundos) o gateway verifica o backend em busca de novas mensagens.

Queue Worker Componente que recupera mensagens da fila e as envia via SMPP.

R

Rate Limiting Controle da taxa de mensagens para cumprir as restrições do operador. Veja TPS.

Receiver Tipo de bind SMPP que apenas recebe mensagens (deliver_sm).

Reconnect Restabelecendo uma conexão SMPP desconectada.

Retry Tentando enviar uma mensagem falhada novamente, geralmente com backoff exponencial.

S

Sequence Number Identificador numérico único atribuído a cada PDU SMPP dentro de uma sessão. Usado para corresponder solicitações com suas respostas (por exemplo, correspondendo um submit_sm com seu submit_sm_resp).

Server Bind Configuração que permite que **ESMEs** externos (clientes) se conectem ao gateway. Neste modo, o gateway atua como um **SMSC** (servidor) aceitando conexões de entrada de sistemas parceiros.

Session Conexão SMPP ativa entre dois sistemas.

source_smsc Campo na fila de mensagens indicando qual bind de servidor deve entregar a mensagem a seus clientes conectados via deliver_sm.

SMPP (Protocolo de Mensagem Curta Ponto a Ponto) Protocolo padrão da indústria para troca de mensagens SMS entre sistemas.

SMSC (Centro de Serviço de Mensagem Curta) Na terminologia SMPP, o componente do servidor que aceita conexões de ESMEs (clientes) e lida com o roteamento e entrega de mensagens SMS. Quando o gateway opera em **Modo Servidor**, ele atua como um SMSC aceitando conexões de ESMEs externos.

SSL/TLS Protocolos de criptografia para comunicação segura.

Submit_SM PDU SMPP para submeter uma mensagem para entrega.

Submit_SM_Resp Resposta SMPP ao submit_sm, indicando sucesso ou falha.

System ID Nome de usuário usado para autenticação SMPP.

T

Telemetry Coleta e transmissão automatizada de métricas do sistema.

TON (Tipo de Número) Campo SMPP que especifica o formato do número (por exemplo, internacional, nacional).

TPS (Transações Por Segundo) Limite de taxa para o máximo de mensagens por segundo através de uma conexão.

Transceiver Tipo de bind SMPP que pode enviar e receber mensagens (mais comum).

Transmitter Tipo de bind SMPP que apenas envia mensagens (submit_sm).

Throughput Taxa de processamento de mensagens, tipicamente medida em mensagens por segundo.

U

UCS-2 Codificação de caracteres Unicode de 16 bits para SMS, suportando até 70 caracteres por mensagem.

Uptime Duração que uma conexão ou serviço tem estado continuamente operacional.

V

Validity Period Limite de tempo para a tentativa de entrega da mensagem antes da expiração.

W

Web Dashboard Interface de usuário baseada em navegador para monitorar e gerenciar o gateway.

Whitelist Veja IP Whitelist e Source Address Whitelist.

X

(Sem termos)

Y

(Sem termos)

Z

(Sem termos)

Referência Rápida de Acrônimos

Acrônimo	Termo Completo
API	Interface de Programação de Aplicações
CIDR	Roteamento Inter-Domínio Sem Classe
DLR	Confirmação de Entrega
ESME	Entidade de Mensagem Curta Externa
GSM	Sistema Global para Comunicações Móveis
HTTP	Protocolo de Transferência de Hipertexto
HTTPS	Protocolo de Transferência de Hipertexto Seguro
IP	Protocolo de Internet
ISDN	Rede Digital de Serviços Integrados
KPI	Indicador Chave de Desempenho
MO	Origem Móvel
MSISDN	Número de Diretório Internacional de Assinante de Estação Móvel
MT	Terminado Móvel
NPI	Indicador de Plano de Numeração
PDU	Unidade de Dados de Protocolo
SMPP	Protocolo de Mensagem Curta Ponto a Ponto

Acrônimo	Termo Completo
SMSC	Centro de Serviço de Mensagem Curta
SMS	Serviço de Mensagem Curta
SSL	Camada de Sockets Segura
TLS	Segurança da Camada de Transporte
TON	Tipo de Número
TPS	Transações Por Segundo
UCS	Conjunto de Caracteres Codificados Universalmente
UI	Interface de Usuário
URL	Localizador Uniforme de Recursos

Documentação Relacionada

- **README.md** - Visão geral do sistema e como começar
- **CONFIGURATION.md** - Parâmetros de configuração explicados
- **USAGE.md** - Operações do dia a dia
- **MONITORING.md** - Métricas e monitoramento
- **TROUBLESHOOTING.md** - Resolução de problemas

Cache de Mensagens SMPP

Visão Geral

O Cache de Mensagens SMPP é uma camada de persistência local que permite que o gateway SMPP continue aceitando mensagens de entrada mesmo quando a API de backend está indisponível. As mensagens são armazenadas localmente no Mnesia e entregues automaticamente à API quando a conectividade é restaurada, utilizando uma lógica de tentativa inteligente com retrocesso exponencial.

Recursos

- **Aceitação Resiliente de Mensagens** - Continue aceitando mensagens SMPP durante interrupções da API
- **Armazenamento Persistente** - Usa Mnesia com `:disc_copies` para durabilidade em reinicializações
- **Tentativa Automática** - Trabalhos em segundo plano tentam automaticamente a entrega com retrocesso exponencial
- **Configuração por Bind** - Ative/desative o cache de forma independente para cada bind SMPP
- **Proteção contra Transbordamento** - Evicção FIFO quando o cache atinge o limite de tamanho configurado
- **Retenção de Mensagens Falhadas** - Mensagens que falharam permanentemente são mantidas para revisão manual
- **Monitoramento em Tempo Real** - Painel LiveView com estatísticas e métricas do cache
- **Métricas Prometheus** - Exportação completa de métricas para monitoramento e alertas

Arquitetura

Fluxo de Mensagens

Com Cache Habilitado

```
Cliente SMPP → submit_sm → Servidor Gateway
                                ↓
                        Cache no Mnesia (resposta imediata)
                                ↓
                        CacheFlushWorker (em segundo plano)
                                ↓
                          API de Backend
```

Com Cache Desabilitado

```
Cliente SMPP → submit_sm → Servidor Gateway
                                ↓
                          API de Backend (direto)
```

Componentes

1. **Módulo MessageCache** (`lib/sms_c/smpp/message_cache.ex`)
    - Lógica central de cache
    - Manipulação de transbordamento
    - Funções de consulta para LiveView e trabalhadores
2. **CacheFlushWorker** (`lib/sms_c/smpp/cache_flush_worker.ex`)
    - GenServer por bind com cache habilitado
    - Polls por mensagens prontas para tentativa
    - Implementa retrocesso exponencial
    - Marca mensagens que falharam permanentemente
3. **Tabela Mnesia** (`:smpp_message_cache`)

- Armazenamento persistente com `:disc_copies`
- Indexada por bind_name, next_retry_at e status
- Sobrevive a reinicializações da aplicação

Configuração

Configurações Globais

Edite `config/runtime.exs`:

```
config :omnimessage_smpp,
  # Com que frequência os trabalhadores de flush poll por
mensagens (milissegundos)
  cache_flush_interval: 10_000,

  # Máximo de tentativas de retry antes de marcar como
failed_permanent
  cache_max_retry_attempts: 10,

  # Multiplicador de retrocesso exponencial
  cache_backoff_multiplier: 2,

  # Tipo de armazenamento Mnesia (:disc_copies ou :ram_copies)
  mnesia_storage_type: :disc_copies
```

Configuração por Bind

Cada bind SMPP (cliente ou servidor) pode ser configurado de forma independente:

```
config :omnimessage_smpp, :binds, [
  %{
    name: "my_smpp_bind",
    mode: :server,
    system_id: "username",
    password: "password",

    # Configuração do cache
    cache_enabled: true,         # Habilitar cache (padrão: false)
    cache_max_size: 10_000,      # Máx. mensagens a serem
armazenadas (padrão: 10.000)
    cache_retry_interval: 60     # Intervalo base de retry em
segundos (padrão: 60)
  }
]
```

Variáveis de Ambiente

```
# Configurações globais do cache
CACHE_FLUSH_INTERVAL=10000              # Intervalo de poll de
flush (ms)
CACHE_MAX_RETRY_ATTEMPTS=10             # Máx. tentativas antes de
falha permanente
CACHE_BACKOFF_MULTIPLIER=2              # Multiplicador de
retrocesso exponencial
MNESIA_STORAGE_TYPE=disc_copies         # Tipo de armazenamento
Mnesia
```

Comportamento de Retry

Retrocesso Exponencial

Quando a entrega da mensagem falha, o intervalo de retry dobra a cada tentativa:

```
Intervalo base: 60 segundos
Multiplicador de retrocesso: 2

Tentativa 0: 60s   (1 minuto)
Tentativa 1: 120s  (2 minutos)
Tentativa 2: 240s  (4 minutos)
Tentativa 3: 480s  (8 minutos)
Tentativa 4: 960s  (16 minutos)
Tentativa 5: 1920s (32 minutos)
...
Tentativa 9: 30,720s (8.5 horas)
```

Falha Permanente

Após 10 tentativas falhadas (por padrão), as mensagens são marcadas como `failed_permanent` e:

- Permanecem no cache para revisão manual
- Param de ser tentadas automaticamente
- Aparecem na seção "Falhas Permanentes" do painel do cache
- Podem ser tentadas novamente ou limpas manualmente

Transições de Status

```
:pending → :delivering → SUCESSO (deletado do cache)
                      → FALHA → :pending (retry com retrocesso)
                               → :failed_permanent (após
tentativas máximas)
```

Monitoramento

Painel LiveView

Acesse o painel do cache em `http://your-server:4000/smpp` → aba "Cache de Mensagens"

Cartões de Resumo:

- Total em Cache - Todas as mensagens atualmente no cache
- Entrega Pendente - Mensagens aguardando retry
- Falhas Permanentes - Mensagens que excederam as tentativas máximas

Tabela por Bind:

- Nome do bind
- Contagem de mensagens em cache
- Quebra de Pendentes / Falhadas
- Tamanho máximo do cache
- Percentual de utilização (com barra de progresso visual)

Métricas Prometheus

```
# Tamanho atual do cache por bind
smpp_cache_size{bind_name="my_bind",mode="server"} 42

# Total de mensagens entregues com sucesso
smpp_cache_delivered_total{bind_name="my_bind"} 1234

# Total de tentativas de retry
smpp_cache_retry_total{bind_name="my_bind"} 56

# Total de falhas permanentes
smpp_cache_permanent_failures_total{bind_name="my_bind"} 2

# Total de eventos de transbordamento (mensagens descartadas)
smpp_cache_overflow_total{bind_name="my_bind"} 0
```

Mensagens de Log

```
INFO  Mensagem -123456789 armazenada para my_smpp_bind
INFO  Mensagem em cache -123456789 entregue com sucesso, ID da
API: 99999
WARN  Falha ao entregar mensagem -123456789 (tentativa 3/10),
próxima tentativa em 2026-02-01 12:34:56Z
ERROR Mensagem -123456789 excedeu tentativas máximas (10),
marcando como failed_permanent
WARN  Transbordamento de cache para my_smpp_bind: mensagem mais
antiga deletada
```

Operações

Habilitar/Desabilitar Cache

Via UI LiveView

1. Navegue até `http://your-server:4000/smpp`
2. Vá para a aba "Client Peers" ou "Server Peers"
3. Edite o bind
4. Altere a caixa de seleção "Cache Enabled"
5. Salve as alterações

Via Console IEx

```
# Habilitar cache para um bind
SmsC.SMPPConfig.update_server_peer("my_bind", "username",
"password",
  cache_enabled: true,
  cache_max_size: 10_000,
  cache_retry_interval: 60
)

# Desabilitar cache
SmsC.SMPPConfig.update_server_peer("my_bind", "username",
"password",
  cache_enabled: false
)
```

Monitorar Status do Cache

```
# Obter resumo global
SmsC.SMPP.MessageCache.get_cache_summary()
# => %{total_cached: 42, pending: 40, failed: 2}

# Obter detalhamento por bind
SmsC.SMPP.MessageCache.get_cache_by_bind()
# => [
#   %{bind_name: "bind1", total: 30, pending: 28, failed: 2,
max_size: 10_000},
#   %{bind_name: "bind2", total: 12, pending: 12, failed: 0,
max_size: 10_000}
# ]

# Contar mensagens para um bind específico
SmsC.SMPP.MessageCache.count_cached_messages("my_bind")
# => 42
```

Intervenções Manuais

Limpar Mensagens Falhadas

```
# Obter todas as mensagens falhadas para um bind
{:atomic, failed_messages} = :mnesia.transaction(fn ->
  :mnesia.match_object({:smpp_message_cache, :_, "my_bind", :_,
:_, :_, :_, :_, :_, :failed_permanent})
end)

# Deletá-las
Enum.each(failed_messages, fn {_, {bind_name, msg_id}, _, _, _, _,
_, _, _, _} ->
  SmsC.SMPP.MessageCache.delete_cache_record(bind_name, msg_id)
end)
```

Forçar Retry de Mensagem Falhada

```
# Resetar uma mensagem failed_permanent para pending
SmsC.SMPP.MessageCache.update_cache_record("my_bind", -123456, %{
  status: :pending,
  retry_count: 0,
  next_retry_at: DateTime.utc_now(),
  last_error: nil
})
```

Resolução de Problemas

Cache Cheio / Eventos de Transbordamento

Sintoma: métrica `cache_overflow_total` aumentando, mensagens mais antigas sendo descartadas

Causa: Limite de tamanho do cache atingido

Soluções:

1. Aumentar `cache_max_size` para o bind
2. Investigar por que a entrega da API está falhando (verificar logs da API, rede)
3. Limpar manualmente mensagens falhadas antigas

4. Verificar se o intervalo de flush está muito lento

Mensagens Não Sendo Entregues

Sintoma: Mensagens presas no status `:pending`

Possíveis Causas:

1. **API está fora do ar**
   - Verificar disponibilidade da API
   - Verificar logs da API de backend
   - Verificar conectividade de rede
2. **next_retry_at está no futuro**
   - Mensagens serão tentadas novamente quando `next_retry_at` for alcançado
   - Verificar cronograma de retrocesso exponencial
3. **Trabalhador de flush não está em execução**

```
# Verificar se os trabalhadores estão em execução
Supervisor.which_children(SmsC.SMPP.Supervisor)
```

4. **Cache desabilitado**
   - Verificar `cache_enabled: true` na configuração do bind

Contagens de Retry Altas

Sintoma: Muitas mensagens com altos valores de `retry_count`

Investigação:

```
# Encontrar mensagens com altas contagens de retry
{:atomic, messages} = :mnesia.transaction(fn ->
  :mnesia.match_object({:smpp_message_cache, :_, "my_bind", :_,
:_, :_, :_, :_, :_, :_})
end)

high_retry = Enum.filter(messages, fn {_, _, _, _, _, _,
retry_count, _, _, _} ->
  retry_count >= 5
end)

# Verificar last_error para cada uma
Enum.each(high_retry, fn {_, _, _, msg_id, _, _, retry_count, _,
last_error, _} ->
  IO.puts("Mensagem #{msg_id}: #{retry_count} tentativas, erro: #
{inspect(last_error)}")
end)
```

Espaço em Disco do Mnesia

Sintoma: Espaço em disco se esgotando

Verificar diretório do Mnesia:

```
ls -lh Mnesia.*
du -sh Mnesia.*
```

Limpar:

1. Limpar mensagens falhadas antigas (ver Intervenções Manuais acima)
2. Reduzir `cache_max_size` por bind
3. Habilitar transbordamento de cache (garantir evicção FIFO adequada)

Considerações de Desempenho

Uso de Memória

- Cada mensagem em cache usa aproximadamente 500-1000 bytes (dependendo do tamanho da mensagem)
- 10.000 mensagens ≈ 5-10 MB de memória
- Com `:disc_copies`, os dados também são gravados em disco

Uso de CPU

- Trabalhadores de flush poll a cada 10 segundos por padrão (configurável)
- Processamento em lote (100 mensagens por ciclo) reduz sobrecarga
- Entrega concorrente (máx. 10 chamadas de API simultâneas por trabalhador)

I/O de Disco

- `:disc_copies` grava no disco em cada transação
- Para throughput muito alto (>1000 msg/sec), considere:
  - Usar `:ram_copies` (perde persistência)
  - Aumentar intervalos de flush
  - Escalar horizontalmente

Limites Recomendados

Cenário	cache_max_size	cache_flush_interval
Baixo volume (<100 msg/sec)	10.000	10.000ms
Volume médio (100-500 msg/sec)	50.000	5.000ms
Alto volume (>500 msg/sec)	100.000	3.000ms

Cenários de Recuperação

Reinicialização da Aplicação

1. O Mnesia carrega automaticamente tabelas `:disc_copies` do disco
2. Mensagens em cache permanecem intactas
3. Trabalhadores de flush reiniciam e continuam o processamento

Migração de Banco de Dados

Ao atualizar de uma versão sem suporte a cache:

1. A migração adiciona automaticamente campos de cache aos binds existentes
2. Valores padrão: `cache_enabled: false`, `cache_max_size: 10_000`, `cache_retry_interval: 60`
3. Sem perda de dados
4. Tabela de cache criada na primeira execução

Recuperação de Interrupção da API

1. Mensagens se acumulam no cache durante a interrupção
2. Quando a API se recupera, os trabalhadores de flush entregam automaticamente
3. Mensagens mais antigas são entregues primeiro (FIFO)
4. O retrocesso exponencial evita sobrecarga da API durante a recuperação

Melhores Práticas

1. **Habilitar Cache por Padrão** - Previne perda de mensagens durante interrupções
2. **Monitorar Métricas** - Configurar alertas em `cache_permanent_failures_total` e `cache_overflow_total`

3. **Dimensionar Apropriadamente** - Definir `cache_max_size` com base na duração esperada da interrupção
4. **Revisar Mensagens Falhadas** - Verificar regularmente mensagens failed_permanent em busca de padrões
5. **Testar Failover** - Simular interrupções da API para verificar o comportamento do cache
6. **Ajustar Intervalos de Retry** - Ajustar com base nos padrões de tempo de recuperação da API
7. **Usar Armazenamento Persistente** - Manter `mnesia_storage_type: disc_copies` em produção

Veja Também

- Referência de Configuração
- Monitoramento e Métricas
- Resolução de Problemas

Guia de Monitoramento e Métricas

Referência completa para monitorar o SMPP Gateway

Visão Geral

O SMPP Gateway expõe métricas no formato Prometheus para monitorar a saúde da conexão, a taxa de mensagens e o desempenho do sistema.

Crítico: Como o gateway é sem estado e depende do OmniMessage Core, **a conectividade do OmniMessage é a métrica mais importante a ser monitorada**. Monitore ambos:

1. **Métricas do SMPP Gateway** - Saúde em nível de protocolo
2. **Métricas da API OmniMessage** - Conectividade e saúde do backend

Endpoint de Métricas

URL: `http://your-server:4000/metrics`

Formato: Formato de texto Prometheus

Acesso: Aberto para localhost por padrão (configure o firewall para acesso remoto)

Teste Rápido

```
curl http://localhost:4000/metrics
```

Métricas Disponíveis

Todas as métricas são prefixadas com `smpp_` e incluem rótulos para identificação.

Métricas de Licença

omnimessage_smpp_license_status

Tipo: Gauge
Descrição: Status atual da licença
Valores:

- `1` = Licença válida
- `0` = Licença inválida/expirada

Rótulos: Nenhum

Exemplo:

```
omnimessage_smpp_license_status 1
```

Uso:

- Alerta quando o valor é 0 (licença inválida)
- Quando a licença é inválida, o processamento da fila de saída para, mas os binds SMPP permanecem conectados
- A interface da Web permanece acessível para solução de problemas

Nome do Produto: `omnimessage_smpp`

Notas:

- Quando a licença é inválida (`license_status == 0`), o gateway para de processar filas de saída
- Os binds SMPP (tanto cliente quanto servidor) permanecem conectados e aceitam solicitações de bind

- Mensagens de entrada ainda são recebidas, mas não processadas
- A interface e o monitoramento permanecem acessíveis, independentemente do status da licença

Exemplo de Alerta:

```
- alert: SMPP_License_Invalid
  expr: omnimessage_smpp_license_status == 0
  for: 1m
  labels:
    severity: critical
  annotations:
    summary: "Licença do SMPP Gateway inválida ou expirada"
    description: "O status da licença é inválido - o processamento
de mensagens de saída está bloqueado"
```

Métricas de Status de Conexão

smpp_connection_status

Tipo: Gauge
Descrição: Status atual da conexão do bind SMPP
Valores:

- `1` = Conectado
- `0` = Desconectado

Rótulos:

- `bind_name` - Nome da conexão (ex: "vodafone_uk")
- `mode` - Tipo de conexão ("client" ou "server")
- `host` - Host remoto (apenas modo cliente)
- `port` - Porta remota (apenas modo cliente)
- `bind_type` - Tipo de bind SMPP (apenas modo cliente)
- `system_id` - ID do sistema utilizado

Exemplo:

```
smpp_connection_status{bind_name="vodafone_uk",mode="client",host="sm
1
```

Uso:

- Alerta quando o valor é 0 (desconectado)
- Acompanhe a porcentagem de tempo de atividade da conexão
- Monitore a frequência de reconexão

Contadores de Mensagens

smpp_messages_sent_total

Tipo: Counter
Descrição: Número total de mensagens enviadas através do bind SMPP
Unidade: Mensagens

Rótulos: Mesmo que connection_status

Exemplo:

```
smpp_messages_sent_total{bind_name="vodafone_uk",mode="client",...}
150234
```

Uso:

- Calcule a taxa de mensagens (mensagens/segundo)
- Acompanhe o volume diário/mensal
- Compare o throughput real com o esperado

smpp_messages_received_total

Tipo: Counter
Descrição: Número total de mensagens recebidas através do bind SMPP
Unidade: Mensagens

Rótulos: Mesmo que connection_status

Exemplo:

```
smpp_messages_received_total{bind_name="partner_acme",mode="server",..
45123
```

Uso:

- Monitore o volume de mensagens de entrada
- Acompanhe o tráfego originado por dispositivos móveis (MO)
- Alerta sobre mudanças inesperadas no volume

Métricas de Entrega

smpp_delivery_failures_total

Tipo: Counter
Descrição: Número total de falhas na entrega de mensagens
Unidade: Falhas

Rótulos: Mesmo que connection_status

Exemplo:

```
smpp_delivery_failures_total{bind_name="vodafone_uk",mode="client",..
234
```

Uso:

- Calcule a taxa de sucesso de entrega
- Alerta sobre altas taxas de falha
- Identifique conexões problemáticas

Cálculo da Taxa de Sucesso:

```
success_rate = (messages_sent - delivery_failures) / messages_sent
* 100
```

Métricas de Operação de Bind

smpp_bind_success_total

Tipo: Counter
Descrição: Número total de operações de bind bem-sucedidas
Unidade: Tentativas de bind

Exemplo:

```
smpp_bind_success_total{bind_name="vodafone_uk",...} 45
```

Uso:

- Acompanhe a estabilidade do bind
- Monitore o sucesso da autenticação

smpp_bind_failures_total

Tipo: Counter
Descrição: Número total de operações de bind falhadas
Unidade: Tentativas de bind

Exemplo:

```
smpp_bind_failures_total{bind_name="vodafone_uk",...} 3
```

Uso:

- Alerta sobre falhas de autenticação
- Identifique problemas de credenciais
- Acompanhe problemas de conexão com a operadora

Métricas de Evento de Conexão

smpp_connection_attempts_total

Tipo: Counter
Descrição: Número total de tentativas de conexão
Unidade: Tentativas

Exemplo:

```
smpp_connection_attempts_total{bind_name="vodafone_uk",...} 48
```

Uso:

- Acompanhe a rotatividade de conexões
- Monitore a frequência de reconexões

smpp_disconnection_total

Tipo: Counter
Descrição: Número total de desconexões
Unidade: Desconexões

Exemplo:

```
smpp_disconnection_total{bind_name="vodafone_uk",...} 3
```

Uso:

- Alerta sobre desconexões frequentes
- Identifique problemas de rede
- Acompanhe a estabilidade da conexão

Métricas de Enquire Link

smpp_enquire_link_sent_total

Tipo: Counter
Descrição: Número total de PDUs enquire_link enviados para verificar a vivacidade da conexão
Unidade: PDUs

Rótulos: Mesmo que connection_status

Exemplo:

```
smpp_enquire_link_sent_total{bind_name="vodafone_uk",mode="client",..
1440
```

Uso:

- Acompanhe a atividade de keepalive
- Compare com recebidos para detectar falhas unidirecionais

smpp_enquire_link_received_total

Tipo: Counter
Descrição: Número total de PDUs enquire_link_resp recebidos do par remoto
Unidade: PDUs

Rótulos: Mesmo que connection_status

Exemplo:

```
smpp_enquire_link_received_total{bind_name="vodafone_uk",mode="client
1438
```

Uso:

- Detecte pares não responsivos (enviados >> recebidos)
- Monitore a saúde da conexão além do status simples

Métricas de Uptime

smpp_uptime_seconds

Tipo: Gauge
Descrição: Uptime atual do bind SMPP em segundos
Unidade: Segundos

Exemplo:

```
smpp_uptime_seconds{bind_name="vodafone_uk",...} 86400
```

Uso:

- Acompanhe a estabilidade da conexão
- Calcule a porcentagem de uptime
- Alerta sobre reinicializações recentes

Métricas de Cache de Mensagens

Essas métricas estão disponíveis quando o cache de mensagens está habilitado em um ou mais binds. Veja MESSAGE_CACHE.md para detalhes de configuração do cache.

smpp_cache_size

Tipo: Gauge
Descrição: Número atual de mensagens no cache local por bind
Unidade: Mensagens

Rótulos:

- `bind_name` - Nome da conexão
- `mode` - Tipo de conexão ("client" ou "server")

Exemplo:

```
smpp_cache_size{bind_name="partner_acme",mode="server"} 42
```

Uso:

- Monitore a utilização do cache
- Alerta quando se aproximar de `cache_max_size`

smpp_cache_delivered_total

Tipo: Counter
Descrição: Número total de mensagens em cache entregues com sucesso à API backend
Unidade: Mensagens

Exemplo:

```
smpp_cache_delivered_total{bind_name="partner_acme"} 1234
```

smpp_cache_retry_total

Tipo: Counter
Descrição: Número total de tentativas de retry para mensagens em cache
Unidade: Tentativas

Exemplo:

```
smpp_cache_retry_total{bind_name="partner_acme"} 56
```

smpp_cache_permanent_failures_total

Tipo: Counter
Descrição: Número total de mensagens que excederam o número máximo de tentativas de retry e foram marcadas como falhas permanentes
Unidade: Mensagens

Exemplo:

```
smpp_cache_permanent_failures_total{bind_name="partner_acme"} 2
```

Uso:

- Alerta quando > 0 (requer revisão manual)

smpp_cache_overflow_total

Tipo: Counter
Descrição: Número total de eventos de overflow de cache onde a mensagem mais antiga foi removida para dar espaço
Unidade: Eventos

Exemplo:

```
smpp_cache_overflow_total{bind_name="partner_acme"} 0
```

Uso:

- Alerta quando aumentando (cache muito pequeno ou interrupção da API muito longa)

Métricas de Saúde da API OmniMessage

Enquanto o gateway em si expõe métricas relacionadas ao SMPP, **a saúde da API OmniMessage é crítica**. Você também deve monitorar:

Das Métricas OmniMessage (se disponíveis)

- `omnimessage_api_requests_total` - Total de solicitações da API do gateway
- `omnimessage_api_request_duration_seconds` - Tempos de resposta da API
- `omnimessage_queue_depth` - Mensagens pendentes na fila OmniMessage

Dos Logs do Gateway (se métricas não expostas)

Procure por esses padrões para detectar problemas na API:

- "api.*connection refused" - Não é possível alcançar o OmniMessage
- "api.*timeout" - OmniMessage não respondendo
- "api.*http 503" - OmniMessage temporariamente fora do ar
- "api.*parse error" - Problema no formato da resposta

Configuração do Prometheus

Configuração Básica de Scrape

Adicione a `/etc/prometheus/prometheus.yml`:

```yaml
scrape_configs:
  - job_name: 'omnimessage-smpp'
    scrape_interval: 15s
    static_configs:
      - targets: ['your-server:4000']
        labels:
          environment: 'production'
          service: 'omnimessage-smpp'
```

Múltiplos Gateways

```yaml
scrape_configs:
  - job_name: 'omnimessage-smpp-instances'
    scrape_interval: 15s
    static_configs:
      - targets:
        - 'smpp-gw-1:4000'
        - 'smpp-gw-2:4000'
        - 'smpp-gw-3:4000'
        labels:
          environment: 'production'
```

Descoberta de Serviço

Usando descoberta baseada em arquivos:

```
scrape_configs:
  - job_name: 'omnimessage-smpp-instances'
    file_sd_configs:
      - files:
        - '/etc/prometheus/targets/smpp-*.json'
```

Arquivo `/etc/prometheus/targets/smpp-production.json`:

```
[
  {
    "targets": ["smpp-gw-1:4000", "smpp-gw-2:4000"],
    "labels": {
      "environment": "production",
      "datacenter": "us-east"
    }
  }
]
```

Painéis de Dashboard do Grafana

Painéis de Dashboard de Exemplo

Painel de Status de Conexão

Consulta:

```
smpp_connection_status{job="omnimessage-smpp"}
```

Visualização: Stat
Limites:

- Vermelho: valor < 1 (desconectado)
- Verde: valor == 1 (conectado)

Painel de Taxa de Mensagens

Consulta:

```
rate(smpp_messages_sent_total{job="omnimessage-smpp"}[5m])
```

Visualização: Gráfico
Unidade: mensagens/segundo
Legenda: `{{bind_name}}`

Painel de Taxa de Sucesso de Entrega

Consulta:

```
100 * (1 - (
  rate(smpp_delivery_failures_total{job="omnimessage-smpp"}[5m])
  /
  rate(smpp_messages_sent_total{job="omnimessage-smpp"}[5m])
))
```

Visualização: Gauge
Unidade: Porcentagem (0-100)
Limites:

- Vermelho: < 95%
- Amarelo: 95-98%
- Verde: > 98%

Painel de Uptime da Conexão

Consulta:

```
smpp_uptime_seconds{job="omnimessage-smpp"} / 3600
```

Visualização: Stat
Unidade: Horas

Regras de Alerta

Regras de Alerta do Prometheus

Salve em `/etc/prometheus/rules/smpp-alerts.yml`:

```yaml
groups:
  - name: smpp_gateway
    interval: 30s
    rules:
      # Conexão caída
      - alert: SMPPConnectionDown
        expr: smpp_connection_status == 0
        for: 2m
        labels:
          severity: critical
        annotations:
          summary: "Conexão SMPP {{ $labels.bind_name }} está
caída"
          description: "Conexão {{ $labels.bind_name }} está
desconectada há mais de 2 minutos."

      # Alta taxa de falha
      - alert: SMPPHighFailureRate
        expr: |
          (
            rate(smpp_delivery_failures_total[5m])
            /
            rate(smpp_messages_sent_total[5m])
          ) > 0.05
        for: 5m
        labels:
          severity: warning
        annotations:
          summary: "Alta taxa de falha de entrega em {{
$labels.bind_name }}"
          description: "A taxa de falha de entrega é {{ $value |
humanizePercentage }} em {{ $labels.bind_name }}."

      # Falhas de bind
      - alert: SMPPBindFailures
        expr: increase(smpp_bind_failures_total[10m]) > 3
        labels:
          severity: warning
        annotations:
          summary: "Múltiplas falhas de bind em {{
$labels.bind_name }}"
          description: "{{ $labels.bind_name }} falhou ao bindar
{{ $value }} vezes nos últimos 10 minutos."
```

```
      # Nenhuma mensagem enviada (quando esperado)
      - alert: SMPPNoTraffic
        expr: rate(smpp_messages_sent_total[10m]) == 0
        for: 30m
        labels:
          severity: warning
        annotations:
          summary: "Nenhuma mensagem enviada em {{
$labels.bind_name }}"
          description: "{{ $labels.bind_name }} não enviou nenhuma
mensagem por 30 minutos."

      # Desconexões frequentes
      - alert: SMPPFrequentDisconnections
        expr: increase(smpp_disconnection_total[1h]) > 5
        labels:
          severity: warning
        annotations:
          summary: "Desconexões frequentes em {{ $labels.bind_name
}}"
          description: "{{ $labels.bind_name }} desconectou {{
$value }} vezes na última hora."

      # OmniMessage API inacessível
      - alert: OmniMessageAPIUnreachable
        expr: |
          count(count_over_time({job="omnimessage-smpp"} |=
"api.*connection refused"[5m])) > 0
        for: 1m
        labels:
          severity: critical
        annotations:
          summary: "API OmniMessage está inacessível"
          description: "O SMPP Gateway não consegue alcançar a API
OmniMessage. Verifique a configuração de API_BASE_URL e a
conectividade de rede."

      # Timeouts da API OmniMessage
      - alert: OmniMessageAPITimeout
        expr: |
          count(count_over_time({job="omnimessage-smpp"} |=
"api.*timeout"[5m])) > 5
        for: 2m
```

```
        labels:
          severity: warning
        annotations:
          summary: "API OmniMessage está com timeout"
          description: "Múltiplos timeouts da API detectados.
OmniMessage pode estar lento ou sobrecarregado."

      # Nenhum fluxo de mensagem (problema na API)
      - alert: NoMessageFlow
        expr: rate(smpp_messages_sent_total[10m]) == 0 and
rate(smpp_messages_received_total[10m]) == 0
        for: 30m
        labels:
          severity: warning
        annotations:
          summary: "Nenhum fluxo de mensagem detectado - verifique
a conectividade do OmniMessage"
          description: "Nenhuma mensagem enviada ou recebida por
30 minutos. Verifique a conectividade da API OmniMessage e o
status da fila."
```

Carregue as regras em `prometheus.yml`:

```
rule_files:
  - '/etc/prometheus/rules/smpp-alerts.yml'
```

Monitoramento do Dashboard Web

A interface da web integrada fornece monitoramento em tempo real sem o Prometheus.

Acesso

URL: `https://your-server:8087`

Página de Status Ao Vivo

Navegação: SMPP → Status Ao Vivo

Recursos:

- Status de conexão em tempo real
- Contadores de mensagens
- Uptime da conexão
- Controles manuais de reconexão/desconexão
- Atualização automática a cada 5 segundos

Uso:

- Verificação rápida de status
- Intervenção manual
- Solução de problemas em tempo real

O dashboard exibe:

- **Total de Binds**: Contagem combinada de todas as conexões de cliente e servidor
- **Binds de Cliente**: Conexões de saída para operadoras (mostrando contagem de conectados/desconectados)

- **Binds de Servidor**: Conexões de entrada de parceiros (mostrando contagem de ativos/aguardando)
- **Servidor Escutando**: Configuração do socket de servidor de entrada (host, porta, conexões máximas)

Monitoramento de Logs

Logs do Sistema

Ver logs:

```
# Acompanhe os logs em tempo real
sudo journalctl -u omnimessage-smpp -f

# Últimas 100 linhas
sudo journalctl -u omnimessage-smpp -n 100

# Desde um horário específico
sudo journalctl -u omnimessage-smpp --since "1 hour ago"

# Filtrar por nível
sudo journalctl -u omnimessage-smpp -p err
```

Logs da Interface Web

Navegação: Aba de logs na interface web

Recursos:

- Streaming de logs em tempo real
- Filtrar por nível (debug, info, warning, error)
- Pesquisar logs
- Pausar/retomar
- Limpar logs

A visualização dos logs permite que você:

- **Filtrar por Nível**: Selecione o nível de log (Todos, Debug, Info, Warning, Error)
- **Pesquisar**: Encontre entradas de log específicas por conteúdo de texto
- **Auto-rolagem**: Habilitar/desabilitar rolagem automática à medida que novos logs chegam
- **Pausar/Retomar**: Pausar atualizações de log para revisar entradas específicas
- **Limpar**: Limpar todos os logs exibidos

Indicadores-Chave de Desempenho (KPIs)

Saúde da Conexão

Métrica: Porcentagem de uptime da conexão

```
avg_over_time(smpp_connection_status[24h]) * 100
```

Meta: > 99.9%

Taxa de Entrega de Mensagens

Métrica: Mensagens entregues por segundo

```
rate(smpp_messages_sent_total[5m])
```

Meta: Corresponde ao volume esperado

Taxa de Sucesso de Entrega

Métrica: Porcentagem de entregas bem-sucedidas

```
100 * (1 - rate(smpp_delivery_failures_total[5m]) /
rate(smpp_messages_sent_total[5m]))
```

Meta: > 98%

Estabilidade do Bind

Métrica: Tentativas de bind por hora

```
rate(smpp_bind_success_total[1h]) * 3600
```

Meta: < 10 por hora (indica conexão estável)

Melhores Práticas de Monitoramento

1. Configure Alertas

- Configure alertas do Prometheus para métricas críticas
- Use PagerDuty/OpsGenie para alertas 24/7

- Teste os alertas regularmente

2. Crie Dashboards

- Construa dashboards do Grafana para cada gateway
- Inclua todas as conexões em um único dashboard
- Adicione painéis de planejamento de capacidade

3. Revisões Regulares

- Revise as métricas semanalmente
- Identifique tendências e padrões
- Planeje ajustes de capacidade

4. Documente Linhas de Base

- Registre volumes normais de mensagens
- Documente taxas de TPS esperadas
- Anote horários/dias de pico

5. Correlacione com o Backend

- Monitore métricas da API do backend
- Acompanhe o fluxo de mensagens de ponta a ponta
- Identifique gargalos

Solução de Problemas com Métricas

Problemas de Conexão

Verifique: `smpp_connection_status`

- Valor 0 = Revise logs, verifique rede, verifique credenciais
- Mudanças frequentes = Instabilidade na rede

Baixas Taxas de Entrega

Verifique: `smpp_delivery_failures_total`

- Alta taxa = Verifique status da operadora, revise formato da mensagem
- Compare entre conexões = Identifique operadora problemática

Baixo Throughput

Verifique: taxa de `smpp_messages_sent_total`

- Abaixo do esperado = Verifique limites de TPS, disponibilidade da fila
- Verifique métricas da API do backend

Problemas de Bind

Verifique: `smpp_bind_failures_total`

- Aumentando = Problemas de autenticação, problemas de credenciais
- Verifique system_id e senha na configuração

Documentação Relacionada

- **CONFIGURATION.md** - Configurar configurações de monitoramento
- **USAGE.md** - Procedimentos operacionais
- **TROUBLESHOOTING.md** - Resolver problemas
- **README.md** - Visão geral e início rápido

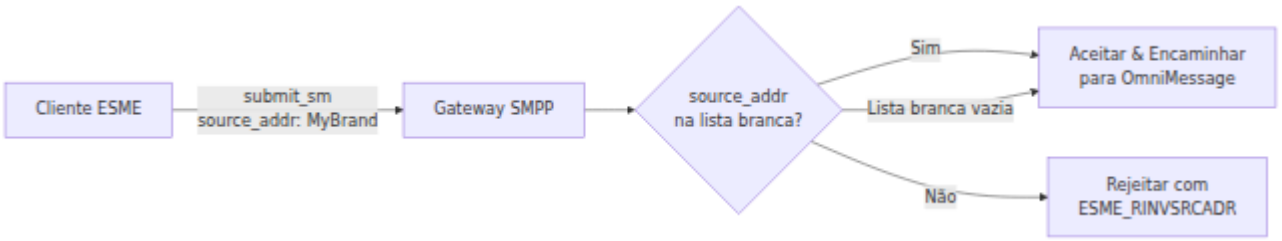
Lista Branca de Endereços de Origem

Controle por par sobre quais endereços de origem (source_addr) um cliente SMPP pode usar ao enviar mensagens.

Visão Geral

Quando um ESME externo (cliente) envia um PDU `submit_sm` através do Gateway SMPP, o PDU inclui um campo `source_addr` representando o endereço de origem (CLI / ID do remetente). Por padrão, clientes autenticados podem usar qualquer endereço de origem. O recurso de Lista Branca de Endereços de Origem permite que os operadores restrinjam quais endereços de origem cada par de servidor pode usar.

Isso segue o mesmo padrão da Lista Branca de IP existente: quando a lista branca está vazia, todos os valores são permitidos. Quando preenchida, apenas os endereços de origem correspondentes são aceitos.



Regras de Correspondência

A correspondência de endereços de origem suporta dois modos:

Correspondência Exata

O endereço de origem deve corresponder exatamente à entrada da lista branca. A correspondência é **sensível a maiúsculas e minúsculas**.

Entrada da Lista Branca	Endereço de Origem	Resultado
`MyBrand`	`MyBrand`	Permitido
`MyBrand`	`mybrand`	Rejeitado
`MyBrand`	`MyBrands`	Rejeitado
`+61400000001`	`+61400000001`	Permitido

Correspondência com Curinga (Prefixo)

Anexe `*` a uma entrada da lista branca para corresponder a qualquer endereço de origem que comece com o prefixo antes do `*`.

Entrada da Lista Branca	Endereço de Origem	Resultado
`614*`	`61400000001`	Permitido
`614*`	`61412345678`	Permitido
`614*`	`61500000001`	Rejeitado
`+614*`	`+61400000001`	Permitido
`My*`	`MyBrand`	Permitido
`My*`	`MyCompany`	Permitido

Múltiplas Entradas

Quando várias entradas estão configuradas, o endereço de origem é permitido se corresponder a **qualquer** entrada na lista branca.

Exemplo de lista branca: `MyBrand, 614*, +61400000001`

Endereço de Origem	Correspondências	Resultado
`MyBrand`	`MyBrand` (exato)	Permitido
`61412345678`	`614*` (curinga)	Permitido
`+61400000001`	`+61400000001` (exato)	Permitido
`OtherBrand`	Nenhuma	Rejeitado
`61500000001`	Nenhuma	Rejeitado

Tratamento de Erros

Quando um `submit_sm` é rejeitado devido a uma violação da lista branca de endereços de origem, o gateway responde com:

Campo	Valor
PDU	`submit_sm_resp`
Status do Comando	`0x0000000A`
Nome do Erro	`ESME_RINVSRCADR` (Endereço de Origem Inválido)
ID da Mensagem	Vazio

Um aviso é registrado com o endereço de origem rejeitado e o nome do par:

```
Servidor SMPP: Rejeitado submit_sm de partner_acme - source_addr
'UnauthorisedBrand' não está na lista branca
```

Configuração

Via Interface Web

1. Navegue até **SMPP > Servidores Peers**
2. Clique em **Editar** no par alvo (ou **Adicionar Novo Servidor Peer**)
3. Localize o campo **Lista Branca de Endereços de Origem** (abaixo da Lista Branca de IP)
4. Insira padrões separados por vírgula:

   ```
   MyBrand,614*,+61400000001
   ```

5. Clique em **Salvar**

As alterações entram em vigor imediatamente para novos PDUs `submit_sm` em conexões existentes.

Via Arquivo de Configuração

Adicione `source_address_whitelist` à configuração de vinculação do servidor em `runtime.exs`:

```
config :omnimessage_smpp, :server_binds, [
  %{
    name: "partner_acme",
    system_id: "acme_corp",
    password: "secure_password",
    allowed_bind_types: [:transmitter, :receiver, :transceiver],
    ip_whitelist: ["203.0.113.0/24"],
    source_address_whitelist: ["MyBrand", "614*", "+61400000001"],
    tps_limit: 50,
    queue_check_frequency: 1000
  }
]
```

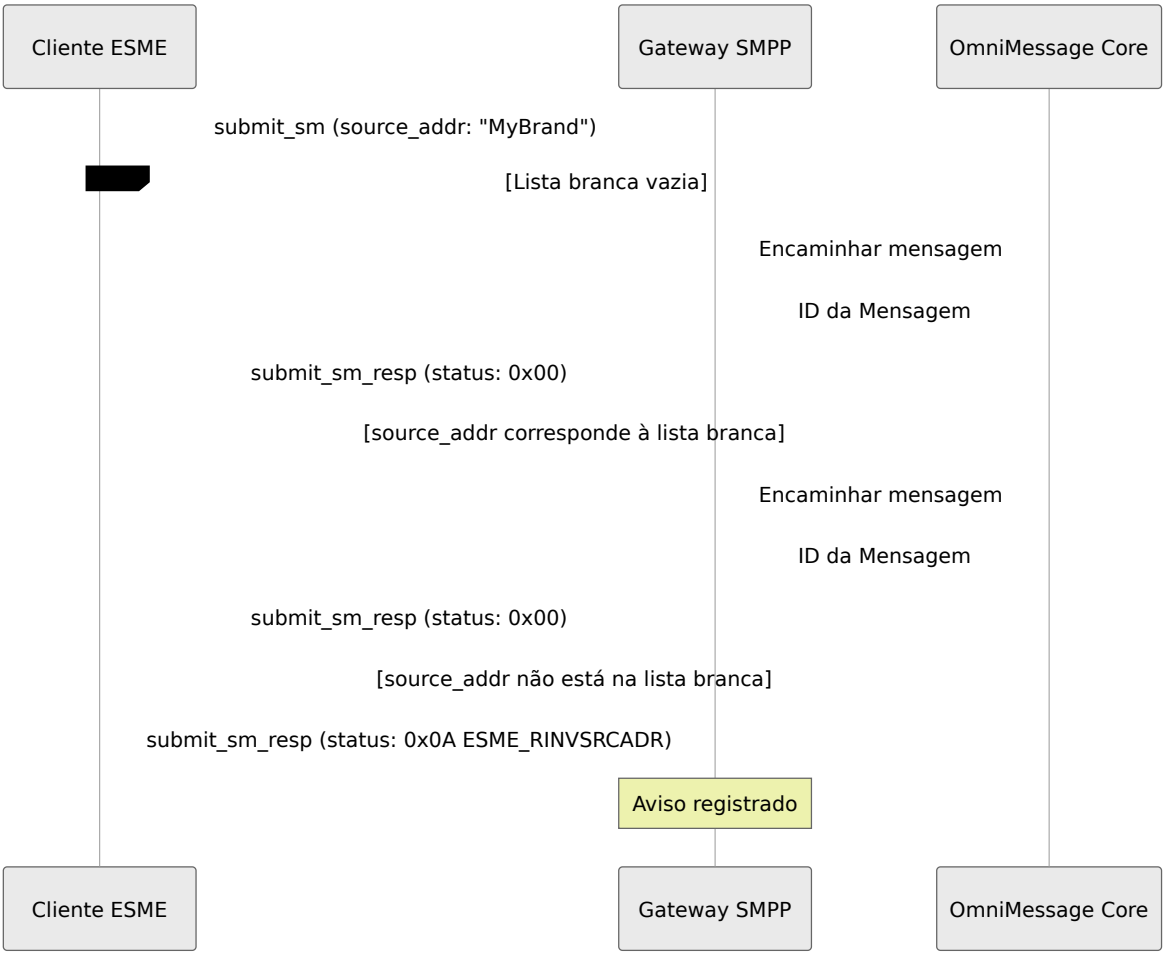
Parâmetros

Parâmetro	Tipo	Obrigatório	Padrão	Descri
`source_address_whitelist`	Lista de strings	Não	`[]` (permitir todos)	Lista de p de endere de origem permitido Suporta correspon exata e cu no final (`* Lista vazi permite t os endere origem.

Migração

Os pares de servidor existentes são migrados automaticamente quando o gateway é iniciado. Pares criados antes que esse recurso fosse adicionado recebem uma lista branca vazia (todos os endereços de origem permitidos), preservando o comportamento existente.

Fluxo de Validação



Exemplos

Restringir a uma Única Marca

Permitir apenas mensagens do ID do remetente `AcmeCorp`:

```
AcmeCorp
```

Permitir um Intervalo de Números Australianos

Permitir qualquer número de celular australiano (começando com `614`):

```
614*
```

Alfanumérico e Numérico Misturado

Permitir um nome de marca e um intervalo de números:

```
AcmeCorp,614*,+61290000001
```

Permitir Todos (Padrão)

Deixe o campo vazio para permitir qualquer endereço de origem. Este é o comportamento padrão.

Resolução de Problemas

Mensagens Rejeitadas com ESME_RINVSRCADR

Sintomas: O parceiro relata `submit_sm_resp` com status do comando `0x0A`.

Possíveis causas:

- O endereço de origem não corresponde a nenhuma entrada na lista branca
- A entrada da lista branca tem um erro de digitação ou padrão incorreto
- Incompatibilidade de maiúsculas e minúsculas (a correspondência é sensível a maiúsculas e minúsculas)
- O padrão de curinga é muito restritivo

Resolução:

1. Verifique a Lista Branca de Endereços de Origem do par de servidor na Interface Web
2. Compare o endereço de origem rejeitado com cada entrada da lista branca
3. Adicione o endereço de origem ausente ou ajuste o padrão de curinga

4. Verifique se a correspondência de maiúsculas e minúsculas é exata para entradas não curinga

Lista Branca Não Tendo Efeito

Sintomas: Mensagens aceitas apesar do endereço de origem não corresponder à lista branca.

Possíveis causas:

- A lista branca está vazia (permite todos por padrão)
- O ESME está conectado a um par de servidor diferente
- Alteração do arquivo de configuração ainda não aplicada (requer reinício)

Resolução:

1. Verifique se a lista branca está preenchida (não vazia) na Interface Web
2. Verifique a qual par de servidor o ESME está vinculado no Status Ao Vivo
3. Se estiver usando o arquivo de configuração, reinicie o serviço

---

Documentação Relacionada

- **Referência de Configuração** - Documentação completa dos parâmetros do par de servidor
- **Guia de Uso** - Gerenciando conexões SMPP
- **Resolução de Problemas** - Procedimentos gerais de resolução de problemas

Guia de Solução de Problemas

Problemas comuns e soluções

Problemas de Conectividade do OmniMessage

Como o SMPP Gateway é sem estado e depende inteiramente do OmniMessage Core, os problemas de conectividade com o OmniMessage são os mais críticos.

Sintomas de Desconexão do OmniMessage

- **Sem mensagens de saída**: A fila aumenta, mensagens não estão sendo enviadas
- **Sem mensagens de entrada**: Parceiros não conseguem enviar mensagens
- **Timeouts**: Chamadas de API com timeout ou travadas
- **Logs mostram**: "Conexão recusada", "Timeout", "HTTP 503", "Conexão redefinida"

Diagnóstico

1. Verifique a Disponibilidade do OmniMessage:

```
# Testar conectividade
curl -k -v https://omnimessage-
core.example.com:8443/api/system/health

# Testar a partir do host do gateway especificamente
ssh gateway-server 'curl -k https://omnimessage-
core.example.com:8443/api/system/health'
```

2. Verifique a URL da API Configurada:

```
# Revisar a configuração
grep -A1 'api_base_url' /opt/omnimessage-smpp/config/runtime.exs

# Verificar conectividade de rede
ping omnimessage-core.example.com
nc -zv omnimessage-core.example.com 8443
```

3. Verifique os Logs do Gateway para Erros de API:

```
# Procurar por erros relacionados à API
sudo journalctl -u omnimessage-smpp -f | grep -i
'api\|omnimessage\|connect'

# Pesquisar logs por erros recentes
sudo journalctl -u omnimessage-smpp -n 200 | grep -i error
```

Soluções

Se o OmniMessage estiver fora do ar:

1. Contate a equipe de operações do OmniMessage
2. Mensagens pendentes se acumularão na fila
3. O gateway continuará tentando (veja `SMPP_POLL_INTERVAL`)
4. Verifique a página de status do OmniMessage ou monitoramento

Se o OmniMessage estiver ativo, mas o gateway não conseguir alcançá-lo:

1. Verifique se as regras do firewall permitem HTTPS de saída
2. Verifique a resolução DNS: `nslookup omnimessage-core.example.com`
3. Verifique o roteamento de rede: `traceroute omnimessage-core.example.com`
4. Verifique os certificados SSL se estiver usando HTTPS

Se a URL da API estiver mal configurada:

1. Edite `/opt/omnimessage-smpp/config/runtime.exs`
2. Verifique se `api_base_url` está correto (deve ser HTTPS para produção)
3. Reinicie o gateway: `sudo systemctl restart omnimessage-smpp`

Problemas de Conexão

A Conexão Não Estabelece

Sintomas:

- Status mostra "Desconectado" (vermelho)
- Nenhum bind bem-sucedido nos logs
- Tentativas de conexão repetidas

Causas Possíveis & Soluções:

1. Problemas de Conectividade de Rede

Verifique:

```
# Testar resolução DNS
nslookup smpp.carrier.com

# Testar conectividade
ping -c 3 smpp.carrier.com

# Testar porta
telnet smpp.carrier.com 2775
# ou
nc -zv smpp.carrier.com 2775
```

Soluções:

- Se o DNS falhar: Use o endereço IP em vez do nome do host na configuração
- Se o ping falhar: Verifique as regras do firewall, contate o operador

- Se a porta falhar: Verifique o número da porta correta, verifique o firewall

2. Credenciais Incorretas

Verifique:

- Logs mostram "bind failed" ou "authentication error"
- Web UI: SMPP → Client Peers → verifique system_id e password

Soluções:

- Confirme as credenciais com o operador
- Verifique se há erros de digitação (sensível a maiúsculas)
- Atualize a configuração e reconecte

3. IP Não Está na Lista Branca

Verifique:

- Conexão rejeitada imediatamente
- Logs do operador mostram IP não autorizado

Soluções:

- Confirme o IP público do seu gateway:

  ```
  curl ifconfig.me
  ```

- Solicite ao operador adicionar o IP à lista branca
- Verifique se o IP não mudou (IP dinâmico)

4. Firewall Bloqueando

Verifique:

```
# Verifique se a porta está aberta
sudo iptables -L -n | grep 2775

# Verifique UFW (Ubuntu/Debian)
sudo ufw status | grep 2775

# Verifique firewalld (RHEL/CentOS)
sudo firewall-cmd --list-ports | grep 2775
```

Soluções:

```
# Ubuntu/Debian
sudo ufw allow out 2775/tcp

# RHEL/CentOS
sudo firewall-cmd --permanent --add-port=2775/tcp
sudo firewall-cmd --reload
```

A Conexão Continua Caindo

Sintomas:

- Conexão estabelecida, mas desconecta frequentemente
- Métrica `smpp_disconnection_total` aumentando
- Logs mostram reconexões repetidas

Causas Possíveis & Soluções:

1. Instabilidade na Rede

Verifique:

```
# Monitorar perda de pacotes
ping -c 100 smpp.carrier.com | grep loss

# Verifique erros de rede
netstat -s | grep -i error
```

Soluções:

- Contate o operador sobre problemas de rede
- Verifique com o ISP se é do seu lado
- Considere conexão/rota de backup

2. Timeout de Enquire Link

Verifique:

- Logs mostram "enquire_link timeout"
- Conexão cai após períodos de inatividade

Soluções:

- O timeout padrão é de 30 segundos
- Verifique se a rede permite pacotes keepalive
- Verifique firewalls agressivos que desconectam conexões ociosas

3. Limite de TPS Excedido

Verifique:

- Alta taxa de mensagens no momento da desconexão
- Operador limitando mensagens

Soluções:

- Revise a configuração `tps_limit`
- Reduza o TPS para 70-80% do máximo do operador
- Espalhe o tráfego por múltiplos binds

4. Problemas no Servidor do Operador

Verifique:

- Verifique o status do serviço do operador
- Contate o suporte do operador

Soluções:

- Aguarde o operador resolver
- Configure um operador de backup se disponível

Problemas de Entrega de Mensagens

Mensagens Não Estão Sendo Enviadas

Sintomas:

- Mensagens presas na fila
- `smpp_messages_sent_total` não aumentando
- Conexão mostra conectada

Causas Possíveis & Soluções:

1. Roteamento de dest_smsc Incorreto

Verifique:

- Web UI → Queue → Verifique o campo `dest_smsc` da mensagem
- Compare com o nome da conexão em SMPP → Live Status

Soluções:

- Mensagens são roteadas com base no campo `dest_smsc`
- Verifique se o backend está configurando o `dest_smsc` correto
- Se `dest_smsc` for NULL, verifique o roteamento padrão

2. Mensagens Agendadas para o Futuro

Verifique:

- Web UI → Queue → Verifique o campo `deliver_after`

- Mensagens com timestamp futuro não serão enviadas ainda

Explicação:

- O sistema de retry define `deliver_after` para mensagens falhadas
- Mensagens aguardam até esse horário antes de tentar novamente

Soluções:

- Aguarde o horário agendado
- Se urgente, contate a equipe de backend para redefinir o timestamp

3. Limite de TPS Muito Baixo

Verifique:

- Grande acúmulo na fila
- Mensagens sendo enviadas muito lentamente

Soluções:

- Aumente o `tps_limit` na configuração
- Verifique se o operador pode lidar com uma taxa maior
- Veja CONFIGURATION.md

4. Worker da Fila Não Está Executando

Verifique:

- Status do serviço
- Logs para erros

Soluções:

```
# Reiniciar serviço
sudo systemctl restart omnimessage-smpp

# Verificar logs
sudo journalctl -u omnimessage-smpp -f
```

Alta Taxa de Falhas na Entrega

Sintomas:

- `smpp_delivery_failures_total` aumentando
- Logs mostram "submit_sm_resp" com status de erro
- Mensagens não chegam aos destinatários

Causas Possíveis & Soluções:

1. Números de Destino Inválidos

Verifique:

- Logs para códigos de erro específicos
- Revise o formato do destino da mensagem

Códigos de Erro Comuns:

- `0x0000000B` - Destino inválido
- `0x00000001` - Comprimento da mensagem inválido
- `0x00000003` - Comando inválido

Soluções:

- Valide o formato do número (E.164 recomendado)
- Verifique se o número inclui o código do país
- Verifique com os requisitos do operador

2. Conteúdo da Mensagem Inválido

Verifique:

- Comprimento da mensagem
- Caracteres especiais
- Codificação

Soluções:

- GSM-7: Máx 160 caracteres
- UCS-2: Máx 70 caracteres
- Remova caracteres não suportados
- Verifique as configurações de codificação

3. Rejeição do Operador

Verifique:

- Códigos de erro específicos do operador
- Padrões em mensagens rejeitadas

Soluções:

- Contate o operador para saber o motivo da rejeição
- Pode ser necessário filtragem de conteúdo
- Verifique padrões de spam/abuso

4. Mensagens Expiradas

Verifique:

- Timestamp de `expires` da mensagem
- Tempo de tentativa de entrega

Soluções:

- Aumente o período de validade da mensagem
- Reduza o atraso de retry para mensagens sensíveis ao tempo

Problemas na Web UI

Não Consegue Acessar o Painel da Web

Sintomas:

- O navegador não consegue conectar a https://your-server:8087

- Timeout ou conexão recusada

Causas Possíveis & Soluções:

1. Serviço Não Está Executando

Verifique:

```
sudo systemctl status omnimessage-smpp
```

Soluções:

```
# Se parado, inicie-o
sudo systemctl start omnimessage-smpp

# Verifique logs para erros
sudo journalctl -u omnimessage-smpp -n 50
```

2. Firewall Bloqueando a Porta 8087

Verifique:

```
sudo ufw status | grep 8087
# ou
sudo firewall-cmd --list-ports | grep 8087
```

Soluções:

```
# Ubuntu/Debian
sudo ufw allow 8087/tcp

# RHEL/CentOS
sudo firewall-cmd --permanent --add-port=8087/tcp
sudo firewall-cmd --reload
```

3. Problemas com Certificado SSL

Verifique:

- O navegador mostra aviso de segurança
- Certificado expirado ou inválido

Soluções:

- Aceitar exceção de segurança (se autoassinado)
- Instalar certificado SSL válido
- Verifique se os arquivos do certificado existem:

```
ls -l /opt/omnimessage-smpp/priv/cert/
```

4. URL Errada

Verifique:

- Verifique usando HTTPS (não HTTP)
- Verifique o IP/hostname do servidor correto
- Verifique a porta 8087

Web UI Mostra Erros

Sintomas:

- A página carrega, mas mostra erros
- Funções não funcionam
- Dados não estão sendo exibidos

Soluções:

1. **Limpar Cache do Navegador**:
    - Ctrl+F5 (atualização forçada)
    - Limpar cache e cookies do navegador
2. **Verificar Console do Navegador**:
    - Pressione F12
    - Verifique a aba Console para erros de JavaScript

- Relate ao suporte se erros encontrados

3. **Tentar Navegador Diferente**:
   - Teste no Chrome, Firefox, Edge
   - Isolar problemas específicos do navegador
4. **Verificar Logs do Serviço**:

```
sudo journalctl -u omnimessage-smpp -f
```

Problemas de Métricas

Métricas do Prometheus Não Disponíveis

Sintomas:

- `curl http://localhost:4000/metrics` falha
- Prometheus não consegue coletar métricas
- Resposta vazia ou de erro

Causas Possíveis & Soluções:

1. Serviço Não Está Executando

Verifique:

```
sudo systemctl status omnimessage-smpp
```

Soluções:

```
sudo systemctl start omnimessage-smpp
```

2. Porta Não Acessível

Verifique:

```
# Testar localmente
curl http://localhost:4000/metrics

# Testar remotamente
curl http://your-server-ip:4000/metrics
```

Soluções:

- Se local funciona, mas remoto não: Verifique o firewall
- Abra a porta 4000 no firewall para o servidor Prometheus

3. Endpoint Errado

Verifique:

- O endpoint é `/metrics` (não `/prometheus` ou `/stats`)
- A porta é 4000 (não 8087)

Métricas Mostram Valores Inesperados

Sintomas:

- Contadores redefinidos para zero
- Medidores mostram valores errados
- Métricas ausentes para alguns binds

Soluções:

1. **Reinício do Serviço Redefine Contadores**:
    - Contadores são redefinidos no reinício do serviço
    - Isso é comportamento normal
    - Use `increase()` ou `rate()` nas consultas do Prometheus
2. **Novos Binds Não Aparecendo**:

- Métricas só aparecem após o primeiro evento
- Envie uma mensagem de teste para popular métricas
- Verifique se o bind está habilitado e conectado

3. **Métricas Obsoletas**:
   - Binds antigos podem ainda aparecer nas métricas
   - Reinicie o serviço para limpar entradas obsoletas
   - Ou use rotulagem do Prometheus para filtrar

Problemas de Desempenho

Alto Uso de CPU

Verifique:

```
top -p $(pgrep -f omnimessage-smpp)
```

Causas Possíveis:

- Volume de mensagens muito alto
- Muitas conexões
- Problema de configuração

Soluções:

- Verifique se a taxa de mensagens está dentro da capacidade
- Revise limites de TPS
- Contate o suporte se o uso de CPU for alto de forma sustentada

Alto Uso de Memória

Verifique:

```
ps aux | grep omnimessage-smpp
```

Causas Possíveis:

- Grande fila de mensagens na memória
- Vazamento de memória (raro)

Soluções:

- Reinicie o serviço para limpar a memória
- Verifique o tamanho da fila de mensagens
- Contate o suporte se a memória crescer continuamente

Processamento Lento de Mensagens

Sintomas:

- Mensagens demoram para serem enviadas
- Fila se acumulando
- Baixa taxa de mensagens

Verifique:

1. Limites de TPS - podem ser muito restritivos
2. `queue_check_frequency` - pode estar muito alto
3. Tempo de resposta da API do backend - pode estar lento
4. Latência de rede para o operador

Soluções:

- Aumente o TPS se o operador permitir
- Diminua `queue_check_frequency` para polling mais rápido
- Otimize a API do backend
- Verifique a latência da rede

Problemas de Configuração

Erros de Sintaxe no Arquivo de Configuração

Sintomas:

- O serviço não inicia após a alteração da configuração
- Logs mostram "syntax error" ou "parse error"

Verifique:

```
# Validar sintaxe Elixir
/opt/omnimessage-smpp/bin/omnimessage-smpp eval "File.read!
('config/runtime.exs')"
```

Erros Comuns:

- Vírgula faltando entre entradas do mapa
- Aspas não correspondentes (" vs ')
- Colchetes ou chaves não correspondentes
- Faltando `import Config` no topo

Soluções:

- Restaurar a partir do backup
- Rever cuidadosamente a sintaxe
- Usar editor de texto com destaque de sintaxe Elixir

Alterações Não Estão Tendo Efeito

Sintomas:

- Configuração modificada, mas sem mudança no comportamento
- Configurações antigas ainda ativas

Soluções:

```
# Alterações na configuração requerem reinício
sudo systemctl restart omnimessage-smpp

# Verifique se o reinício foi bem-sucedido
sudo systemctl status omnimessage-smpp

# Verifique logs para erros
sudo journalctl -u omnimessage-smpp -n 50
```

Recuperação de Emergência

Falha Completa do Sistema

Etapas:

1. **Verifique a saúde básica do sistema**:

   ```
   # Espaço em disco
   df -h

   # Memória
   free -h

   # Carga da CPU
   uptime
   ```

2. **Verifique o status do serviço**:

   ```
   sudo systemctl status omnimessage-smpp
   ```

3. **Revise logs recentes**:

   ```
   sudo journalctl -u omnimessage-smpp -n 200
   ```

4. **Tente reiniciar o serviço**:

```
sudo systemctl restart omnimessage-smpp
```

5. **Se o reinício falhar**:
   - Verifique a sintaxe da configuração
   - Verifique se os certificados SSL existem
   - Verifique permissões de arquivos
   - Revise logs para erro específico
6. **Restaure a partir do backup** (se necessário):

```
# Restaurar configuração
sudo cp /opt/omnimessage-smpp/config/runtime.exs.backup \
  /opt/omnimessage-smpp/config/runtime.exs

# Reiniciar
sudo systemctl restart omnimessage-smpp
```

7. **Contate o suporte** se não resolver

Obtendo Ajuda

Informações a Coletar

Antes de contatar o suporte, colete:

1. **Versão**: `cat /opt/omnimessage-smpp/VERSION`
2. **Logs Recentes**:

```
sudo journalctl -u omnimessage-smpp -n 200 > /tmp/smpp-logs.txt
```

3. **Configuração** (sanitizar senhas):

```
sudo cp /opt/omnimessage-smpp/config/runtime.exs
/tmp/config.exs
# Edite /tmp/config.exs para remover senhas antes de enviar
```

4. **Saída de Métricas**:

```
curl http://localhost:4000/metrics > /tmp/metrics.txt
```

5. **Informações do Sistema**:

```
uname -a > /tmp/system-info.txt
free -h >> /tmp/system-info.txt
df -h >> /tmp/system-info.txt
```

Contate o Suporte

- **Email**: support@omnitouch.com
- **Telefone**: +61 XXXX XXXX (24/7)
- **Inclua**: Todas as informações acima

Documentação Relacionada

- **USAGE.md** - Procedimentos operacionais normais
- **CONFIGURATION.md** - Referência de configuração
- **MONITORING.md** - Monitoramento e métricas
- **README.md** - Visão geral do sistema

Guia de Operações

Procedimentos operacionais do dia a dia

Dependência Crítica: OmniMessage Core

IMPORTANTE: O Gateway SMPP do OmniMessage não pode funcionar sem acesso ao OmniMessage Core. Todo o processamento de mensagens acontece no OmniMessage - o gateway é apenas um tradutor de protocolo.

Se o OmniMessage se tornar indisponível:

- □ Novas mensagens não podem ser enviadas
- □ Mensagens pendentes não podem ser recuperadas
- □ O status de entrega não pode ser reportado
- □ O sistema parece travar ou expirar

Verifique a Saúde do OmniMessage:

```
# Testar conectividade da API
curl -k https://omnimessage-
core.example.com:8443/api/system/health

# Verificar URL da API configurada nos logs
grep api_base_url /opt/omnimessage-smpp/config/runtime.exs
```

Operações Diárias

Verificação de Saúde Matinal

Realize essas verificações no início de cada dia:

1. **Acessar o Painel da Web**

- URL: `https://your-server:8087`
- Verifique se o painel carrega corretamente

2. **Verificar Status da Conexão**

- Navegue até: SMPP → Status Ao Vivo
- Verifique se todas as conexões mostram "Conectado" (verde)
- Anote quaisquer binds desconectados

3. **Revisar Métricas de Mensagens**

- Navegue até: aba Fila
- Verifique se as contagens de mensagens são razoáveis
- Verifique se não há acúmulo inesperado na fila

4. **Verificar Logs do Sistema**

- Navegue até: aba Logs
- Procure por mensagens de erro (vermelho)
- Anote quaisquer padrões de aviso

5. **Revisar Métricas do Prometheus**

- `curl http://localhost:4000/metrics`
- Ou verifique os painéis do Grafana
- Verifique se as taxas de mensagens estão normais

Monitoramento Contínuo

Configure alertas para:

- Falhas de conexão (> 2 minutos fora)
- Altas taxas de falha de entrega (> 5%)
- Sem tráfego por períodos prolongados
- Desconexões frequentes

Veja MONITORING.md para configuração de alertas.

Compreendendo o Roteamento de Mensagens

O gateway roteia mensagens entre o OmniMessage Core e conexões SMPP usando dois campos principais:

- **`dest_smsc`** — Roteia mensagens de saída para **binds de cliente**. Quando o OmniMessage coloca uma mensagem na fila com `dest_smsc: "vodafone_uk"`, o bind de cliente do gateway chamado `vodafone_uk` a pega e a envia via SMPP `submit_sm`.
- **`source_smsc`** — Roteia mensagens de entrada para **binds de servidor**. Quando o OmniMessage coloca uma mensagem na fila com `source_smsc: "partner_acme"`, o gateway a entrega aos clientes conectados ao bind de servidor chamado `partner_acme` via SMPP `deliver_sm`.

Distinção chave: Binds de cliente enviam PDUs `submit_sm` (o gateway é o ESME enviando para um transportador). Binds de servidor enviam PDUs `deliver_sm` (o gateway é o SMSC entregando a um ESME conectado).

Registro de Frontend

O gateway se registra automaticamente com o OmniMessage Core para que o backend saiba quais conexões SMPP estão disponíveis para roteamento de mensagens.

- **Nome de registro**: Controlado pela configuração `smsc_name` (padrão: `"smpp_gateway"`, env: `SMSC_NAME`)
- **Heartbeat**: Enviado a cada 60 segundos para manter o registro ativo
- **Expiração**: O registro expira no backend após 90 segundos sem um heartbeat
- **Registro por bind**: Cada par habilitado é registrado individualmente usando o formato `{hostname}_{peer_name}`

Se o gateway parar ou perder conectividade com o OmniMessage Core, seus registros expiram e o backend para de roteá-las para ele.

Solução de Problemas: Se as mensagens não estão sendo roteadas para o gateway, verifique:

1. Logs para entradas "frontend_register"
2. Se o `smsc_name` corresponde ao que o OmniMessage espera
3. Conectividade de rede com o OmniMessage Core (`api_base_url`)

Gerenciando Conexões SMPP

Como os Pares SMPP São Configurados

As conexões SMPP (pares) podem ser configuradas usando **dois métodos**:

Método 1: UI Web (Recomendado)

- **Vantagem**: As mudanças entram em vigor imediatamente, sem necessidade de reinício
- **Localização**: SMPP → Abas de Pares de Cliente / Pares de Servidor

- **Operações**: Adicionar, editar, excluir pares
- **Persistência**: Armazenado no banco de dados Mnesia
- **Melhor para**: Operações do dia a dia, testes, mudanças rápidas

Método 2: Arquivo de Configuração

- **Vantagem**: Configuração como código, controle de versão
- **Localização**: `/opt/omnimessage-smpp/config/runtime.exs`
- **Operações**: Definir pares na configuração Elixir
- **Persistência**: Baseada em arquivo, sobrevive a reinicializações
- **Requer**: Reinício do serviço após mudanças
- **Melhor para**: Configuração inicial, infraestrutura como código

Nota: As mudanças na UI Web são armazenadas separadamente e sobrescrevem as configurações do arquivo de configuração.

Veja CONFIGURATION.md para referência do arquivo de configuração.

Adicionando uma Nova Conexão de Cliente

Objetivo: Configurar o gateway para agir como um **ESME** (cliente) conectando-se ao **SMSC** (servidor) de um transportador

Preparação: Reúna informações do transportador:

- Nome do host/IP do servidor SMPP
- Número da porta (geralmente 2775)
- ID do sistema (nome de usuário)
- Senha
- Tipo de bind (geralmente transceiver)
- Limite de TPS

Escolha um dos seguintes métodos:

Opção A: Via UI Web (Recomendado)

Vantagens: Efeito imediato, sem necessidade de reinício

Passos:

1. **Navegar até Pares de Cliente**:
   - Abra a UI Web: `https://your-server:8087`
   - Navegue até: SMPP → Pares de Cliente
2. **Adicionar Novo Par**:
   - Clique em "Adicionar Novo Par de Cliente"
   - Preencha o formulário:
     - **Nome**: `vodafone_uk` (identificador único)
     - **Host**: `smpp.vodafone.co.uk`
     - **Porta**: `2775`
     - **ID do Sistema**: `your_username`
     - **Senha**: `your_password`
     - **Tipo de Bind**: `Transceiver`
     - **Limite de TPS**: `100`
     - **Frequência de Verificação da Fila**: `1000`
   - Clique em "Salvar"
3. **Conexão Estabelece Automaticamente**:
   - O gateway tenta imediatamente a conexão
   - Navegue até: SMPP → Status Ao Vivo

- O status deve mudar para "Conectado" (verde) dentro de 10-30 segundos
- Verifique a aba de Logs para mensagem de bind bem-sucedida

4. **Testar Fluxo de Mensagens**:
   - Navegue até: aba Fila
   - Envie uma mensagem de teste com `dest_smsc` correspondente ao nome do bind
   - Monitore no Status Ao Vivo para transmissão
   - Verifique a confirmação de entrega

Opção B: Via Arquivo de Configuração

Vantagens: Infraestrutura como código, controle de versão

Passos:

1. **Editar Arquivo de Configuração**:

```
sudo nano /opt/omnimessage-smpp/config/runtime.exs
```

2. **Adicionar Novo Bind à Configuração**:

```
config :omnimessage_smpp, :binds, [
  # Binds existentes...

  # Adicionar novo bind
  %{
    name: "vodafone_uk",
    mode: :client,
    bind_type: :transceiver,
    host: "smpp.vodafone.co.uk",
    port: 2775,
    system_id: "your_username",
    password: "your_password",
    tps_limit: 100,
    queue_check_frequency: 1000
  }
]
```

3. **Salvar e Reiniciar o Serviço**:

```
# Salvar arquivo (Ctrl+X, Y, Enter no nano)

# Reiniciar serviço
sudo systemctl restart omnimessage-smpp
```

4. **Verificar Conexão**:
   - Navegue até: SMPP → Status Ao Vivo
   - Encontre nova conexão
   - O status deve ser "Conectado" (verde)
   - Verifique logs para bind bem-sucedido
5. **Testar Fluxo de Mensagens**:
   - Navegue até: aba Fila
   - Envie uma mensagem de teste com `dest_smsc` correspondente ao novo nome do bind
   - Monitore no Status Ao Vivo para transmissão
   - Verifique a confirmação de entrega

Adicionando um Bind de Servidor

Objetivo: Configurar o gateway para agir como um **SMSC** (servidor) aceitando conexões de **ESMEs** externas (clientes parceiros)

Preparação:

1. **Gerar Credenciais**:
    - Crie um ID de sistema único: `partner_name`
    - Crie uma senha forte
    - Documente e compartilhe de forma segura com o parceiro
2. **Obter Informações do Parceiro**:
    - Endereços IP de origem do parceiro
    - Volume de mensagens esperado (para limite de TPS)
    - Tipos de bind necessários

Escolha um dos seguintes métodos:

Opção A: Via UI Web (Recomendado)

Vantagens: Efeito imediato, sem necessidade de reinício

Passos:

1. **Navegar até Pares de Servidor**:
    - Abra a UI Web: `https://your-server:8087`
    - Navegue até: SMPP → Pares de Servidor
2. **Adicionar Novo Par de Servidor**:
    - Clique em "Adicionar Novo Par de Servidor"
    - Preencha o formulário:
        - **Nome**: `partner_acme` (identificador único)
        - **ID do Sistema**: `acme_corp`
        - **Senha**: `secure_password_123`

        - **Tipos de Bind Permitidos**: Selecione todos (Transmissor, Receptor, Transceiver)
        - **Lista de IP Permitidos**: `203.0.113.0/24` (separados por vírgula para múltiplos)
        - **Limite de TPS**: `50`
        - **Frequência de Verificação da Fila**: `1000`
    - Clique em "Salvar"

3. **Gateway Pronto para Conexão**:
    - O par de servidor agora está ativo e aguardando conexão do parceiro
    - Nenhum reinício necessário
4. **Compartilhar Informações com o Parceiro**:
    - Endereço IP do gateway
    - Porta: `2775`
    - ID do Sistema: `acme_corp`
    - Senha: `secure_password_123`
    - Tipo de Bind: Conforme configurado
5. **Aguardar Conexão do Parceiro**:
    - Navegue até: SMPP → Status Ao Vivo
    - Observe a conexão de entrada

- Verifique o sucesso da autenticação
- Verifique se o IP corresponde à lista de permitidos

Opção B: Via Arquivo de Configuração

Vantagens: Infraestrutura como código, controle de versão

Passos:

1. **Editar Arquivo de Configuração**:

```
sudo nano /opt/omnimessage-smpp/config/runtime.exs
```

2. **Adicionar Bind de Servidor e Configuração de Escuta**:

```
# Adicionar à lista de server_binds
config :omnimessage_smpp, :server_binds, [
  # Binds de servidor existentes...

  # Adicionar novo bind de servidor
  %{
    name: "partner_acme",
    system_id: "acme_corp",
    password: "secure_password_123",
    allowed_bind_types: [:transmitter, :receiver,
:transceiver],
    ip_whitelist: ["203.0.113.0/24"],
    tps_limit: 50,
    queue_check_frequency: 1000
  }
]

# Garantir que a configuração de escuta exista (apenas
necessário uma vez)
config :omnimessage_smpp, :listen, %{
  host: "0.0.0.0",
  port: 2775,
  max_connections: 100
}
```

3. **Salvar e Reiniciar o Serviço**:

```
sudo systemctl restart omnimessage-smpp
```

4. **Compartilhar Informações com o Parceiro**:

   - Endereço IP do gateway
   - Porta: `2775`
   - ID do Sistema: `acme_corp`
   - Senha: `secure_password_123`
   - Tipo de Bind: Conforme configurado

5. **Aguardar Conexão do Parceiro**:

   - Navegue até: SMPP → Status Ao Vivo
   - Observe a conexão de entrada
   - Verifique o sucesso da autenticação
   - Verifique se o IP corresponde à lista de permitidos

Modificando Conexão Existente

Objetivo: Atualizar parâmetros de conexão (limites de TPS, senhas, lista de IP permitidos, etc.)

Escolha um dos seguintes métodos:

Opção A: Via UI Web (Recomendado)

Vantagens: Efeito imediato, sem necessidade de reinício

Passos:

1. **Navegar até Pares**:

   - Abra a UI Web: `https://your-server:8087`
   - Para conexões de cliente: SMPP → Pares de Cliente
   - Para conexões de servidor: SMPP → Pares de Servidor

2. **Editar Par**:
    - Encontre o par a ser modificado
    - Clique no botão "Editar"
    - Atualize os parâmetros desejados:
        - Mudanças comuns: Limite de TPS, senha, lista de IP permitidos, host/porta
    - Clique em "Salvar"
3. **Mudanças Aplicam-se Imediatamente**:
    - A conexão reconecta automaticamente com as novas configurações
    - Nenhum reinício do serviço necessário
    - Navegue até: SMPP → Status Ao Vivo para verificar
4. **Verificar Mudanças**:
    - Verifique se a conexão é estabelecida com sucesso
    - Monitore a aba de Logs para erros
    - Teste o fluxo de mensagens, se aplicável

Opção B: Via Arquivo de Configuração

Vantagens: Infraestrutura como código, controle de versão

Passos:

1. **Editar Arquivo de Configuração**:

```
sudo nano /opt/omnimessage-smpp/config/runtime.exs
```

2. **Modificar Parâmetros de Bind**:
    - Encontre o bind na lista `:binds` ou `:server_binds`
    - Atualize os parâmetros desejados:
        - Mudanças comuns: Limite de TPS, senhas, lista de IP permitidos, host/porta
    - Exemplo:

```
%{
  name: "vodafone_uk",
  # ... outros params
  tps_limit: 150,  # Alterado de 100
  password: "new_password"  # Senha atualizada
}
```

3. **Salvar e Reiniciar o Serviço**:

```
sudo systemctl restart omnimessage-smpp
```

4. **Verificar Mudanças**:
   - Navegue até: SMPP → Status Ao Vivo
   - Verifique se a conexão é estabelecida com sucesso
   - Monitore logs para erros
   - Teste o fluxo de mensagens

Removendo uma Conexão

Objetivo: Descomissionar uma conexão SMPP

Passos:

1. **Notificar Partes Interessadas**:
   - Informar o transportador/parceiro
   - Coordenar janela de inatividade
2. **Desconectar via UI Web**:
   - Navegue até: SMPP → Status Ao Vivo
   - Encontre a conexão
   - Clique em "Dropar Conexão"
   - Confirme a ação
3. **Remover Configuração**:

- Navegue até: SMPP → Pares de Cliente/Servidor
- Encontre a conexão
- Clique em "Excluir"
- Confirme a remoção

4. **Verificar Remoção**:
   - Verifique o Status Ao Vivo - a conexão deve ter desaparecido
   - Revise logs para um desligamento limpo

Habilitando e Desabilitando Conexões

Objetivo: Retirar temporariamente uma conexão do ar sem excluir sua configuração

Os pares têm um campo `enabled` que controla se estão ativos. Pares desabilitados mantêm toda a sua configuração, mas não estabelecem ou aceitam conexões.

Via UI Web:

1. Navegue até: SMPP → Pares de Cliente ou Pares de Servidor
2. Encontre o par a ser desabilitado
3. Clique em "Editar"
4. Desmarque a caixa "Habilitado"
5. Clique em "Salvar"

A conexão será desconectada imediatamente. Para reabilitar, repita os passos e marque a caixa novamente.

Casos de uso:

- Janelas de manutenção planejadas do transportador
- Pausar temporariamente uma conexão de parceiro durante uma investigação
- Desabilitar uma conexão enquanto aguarda novas credenciais

Comportamento da Conexão

Lógica de Reconexão

Quando um bind de cliente desconecta inesperadamente, o gateway tenta automaticamente reconectar:

- **Intervalo de tentativa**: A cada 30 segundos
- **Inicialização escalonada**: Quando múltiplos binds iniciam simultaneamente (por exemplo, após uma reinicialização do serviço), as conexões são escalonadas com atrasos de 500ms entre cada bind para evitar sobrecarregar a rede
- **Inicialização resiliente**: Se um transportador não estiver acessível na inicialização do gateway, o gateway inicia com sucesso e tenta a conexão em segundo plano

Enquire Link (Keepalive)

O gateway envia periodicamente PDUs SMPP `enquire_link` para verificar se as conexões estão ativas:

- **Intervalo padrão**: 60 segundos (configurável por bind via `enquire_link_interval`)
- **Desabilitar**: Defina `enquire_link_interval: 0` (não recomendado)
- **Detecção de falhas**: Se o par remoto parar de responder ao enquire_link, a conexão é considerada morta e a reconexão começa

Monitore a saúde do enquire link via métricas do Prometheus `smpp_enquire_link_sent_total` e `smpp_enquire_link_received_total`. Um aumento na diferença entre enviado e recebido indica problemas de conexão.

Limitação de Taxa de TPS

Cada bind aplica seu `tps_limit` usando uma janela deslizante por segundo:

- As mensagens são contadas dentro de cada janela de 1 segundo

- Quando o limite é atingido, o trabalhador da fila pausa até o próximo segundo
- No máximo 100 mensagens podem estar em trânsito (aguardando resposta) por bind a qualquer momento
- A janela é redefinida automaticamente no início de cada novo segundo

Se você observar baixa taxa de transferência, verifique se:

1. `tps_limit` está definido alto o suficiente para seu tráfego
2. `queue_check_frequency` é baixo o suficiente para manter o pipeline alimentado
3. O transportador está respondendo às mensagens prontamente (respostas lentas reduzem a taxa de transferência efetiva)

Gerenciando o Fluxo de Mensagens

Verificando a Fila de Mensagens

Objetivo: Monitorar mensagens pendentes

Passos:

1. **Acessar Fila**:
    - Navegue até: aba Fila
    - Veja a lista de mensagens pendentes

2. **Verificar Detalhes da Mensagem**:
    - Clique na linha da mensagem
    - Revise:
        - Número de destino
        - Corpo da mensagem
        - SMSC alvo (dest_smsc)
        - Tentativas de entrega
        - Status
3. **Pesquisar Mensagem Específica**:
    - Use o filtro de pesquisa
    - Filtre por destino, conteúdo ou SMSC

Solução de Problemas de Mensagens Presas

Sintomas: Mensagens não estão sendo entregues

Passos:

1. **Verificar Status da Conexão**:
    - Navegue até: SMPP → Status Ao Vivo
    - Verifique se a conexão alvo está conectada
    - Se desconectada, veja Reconectando

2. **Verificar Detalhes da Mensagem**:
    - Navegue até: aba Fila
    - Encontre a mensagem presa
    - Verifique se o campo `dest_smsc` corresponde ao nome da conexão
    - Verifique o timestamp `deliver_after` (agendamento de nova tentativa)
3. **Verificar Tentativas de Entrega**:
    - Muitas tentativas = falhas repetidas
    - Verifique logs para mensagens de erro
    - Pode indicar formato inválido ou rejeição do transportador
4. **Intervenção Manual** (se necessário):
    - Contate o transportador para verificar o problema
    - Pode ser necessário cancelar e reenviar a mensagem
    - Verifique com a equipe de backend sobre problemas na fila

---

Solução de Problemas de Conexão

Reconectando um Bind

Sintomas: Conexão mostra "Desconectada" (vermelho)

Passos:

1. **Verificar Conectividade de Rede**:

```
ping -c 3 carrier-smpp-server.com
telnet carrier-smpp-server.com 2775
```

2. **Verificar Logs para Erros**:
    - Navegue até: aba Logs

- Filtrar: Nível de erro
- Procure por falhas de autenticação, timeouts de rede

3. **Verificar Credenciais**:
   - Navegue até: SMPP → Pares de Cliente/Servidor
   - Verifique se `system_id` e senha estão corretos
   - Contate o transportador se não tiver certeza
4. **Reconexão Manual**:
   - Navegue até: SMPP → Status Ao Vivo
   - Encontre o bind desconectado
   - Clique no botão "Reconectar"
   - Aguarde 10-30 segundos
   - Verifique se o status muda para "Conectado"
5. **Se a Reconexão Falhar**:
   - Verifique regras de firewall
   - Verifique se o servidor do transportador está operacional
   - Contate o suporte do transportador
   - Veja TROUBLESHOOTING.md

Tratando Falhas de Autenticação

Sintomas: Falhas repetidas de bind nos logs

Causas:

- Nome de usuário/senha incorretos
- IP não está na lista permitida do transportador
- Conta suspensa/expirada

Passos:

1. **Verificar Credenciais**:
    - Navegue até: SMPP → Pares de Cliente
    - Verifique novamente `system_id` e senha
    - Confirme com o transportador
2. **Verificar Lista de IP Permitidos**:
    - Confirme o IP do seu gateway com o transportador
    - Solicite ao transportador verificar a lista de IPs permitidos
3. **Verificar Status da Conta**:
    - Verifique se a conta está ativa
    - Verifique se há contratos expirados
    - Contate a cobrança do transportador
4. **Atualizar Configuração**:
    - Se as credenciais mudaram, atualize na UI Web
    - Clique em "Reconectar" para tentar novamente com as novas credenciais

---

Monitoramento e Alerta

Verificando Métricas do Prometheus

Verificação rápida:

```
curl http://localhost:4000/metrics | grep smpp_connection_status
```

Saída esperada:

```
smpp_connection_status{bind_name="vodafone_uk",...} 1
smpp_connection_status{bind_name="att_us",...} 1
```

Todos os valores devem ser `1` (conectado).

Respondendo a Alertas

Alerta de Conexão Fora:

1. Verifique UI Web → SMPP → Status Ao Vivo
2. Tente reconexão manual
3. Verifique logs para erros
4. Contate o transportador se a interrupção for prolongada
5. Veja TROUBLESHOOTING.md

Alerta de Alta Taxa de Falha:

1. Verifique logs para padrões de erro
2. Revise mudanças recentes na configuração
3. Contate o transportador sobre rejeições
4. Verifique conformidade do formato da mensagem

Alerta de Sem Tráfego:

1. Verifique se a fila do backend tem mensagens
2. Verifique se o roteamento `dest_smsc` está correto

3. Verifique se os limites de TPS não estão muito restritivos
4. Revise a configuração de `queue_check_frequency`

Procedimentos de Manutenção

Manutenção Rotineira

Realizar mensalmente:

1. **Revisar Métricas**:
    - Analisar tendências de volume de mensagens
    - Verificar taxas de sucesso de entrega
    - Identificar oportunidades de otimização
2. **Atualizar Documentação**:
    - Documentar quaisquer mudanças de configuração
    - Atualizar informações de contato
    - Anotar janelas de manutenção do transportador
3. **Auditoria de Credenciais**:
    - Revisar todas as senhas SMPP
    - Planejar rotação de credenciais
    - Verificar se as listas de IP estão atualizadas
4. **Planejamento de Capacidade**:
    - Revisar taxas de mensagens de pico
    - Verificar contra limites de TPS
    - Planejar para crescimento

Reinício do Serviço

Quando necessário:

- Após mudanças no arquivo de configuração
- Após atualizações do sistema
- Durante solução de problemas

Passos:

```
# Verificar status atual
sudo systemctl status omnimessage-smpp

# Reiniciar serviço
sudo systemctl restart omnimessage-smpp

# Verificar reinício
sudo systemctl status omnimessage-smpp

# Verificar logs
sudo journalctl -u omnimessage-smpp -n 50
```

Verificar via UI Web:

1. Acesse o painel (pode levar 30-60 segundos para ficar online)
2. Navegue até: SMPP → Status Ao Vivo
3. Aguarde que todas as conexões sejam estabelecidas (1-2 minutos)
4. Verifique logs para erros

Backup de Configuração

Faça backup de arquivos críticos antes de mudanças:

```
# Backup da configuração
sudo cp /opt/omnimessage-smpp/config/runtime.exs \
  /opt/omnimessage-smpp/config/runtime.exs.backup.$(date +%Y%m%d)

# Backup de certificados
sudo tar -czf /tmp/smpp-certs-$(date +%Y%m%d).tar.gz \
  /opt/omnimessage-smpp/priv/cert/
```

Restaurar se necessário:

```
# Restaurar configuração
sudo cp /opt/omnimessage-smpp/config/runtime.exs.backup.YYYYMMDD \
  /opt/omnimessage-smpp/config/runtime.exs

# Reiniciar serviço
sudo systemctl restart omnimessage-smpp
```

Procedimentos de Emergência

Interrupção Completa do Serviço

Passos:

1. **Verificar status do serviço**:

   ```
   sudo systemctl status omnimessage-smpp
   ```

2. **Se o serviço parou, inicie-o**:

   ```
   sudo systemctl start omnimessage-smpp
   ```

3. **Verifique logs para razão do crash**:

   ```
   sudo journalctl -u omnimessage-smpp -n 100
   ```

4. **Se não iniciar**:

   - Verifique erros de sintaxe na configuração
   - Verifique se os certificados SSL existem
   - Verifique espaço em disco: `df -h`
   - Verifique memória: `free -h`

5. **Contate o suporte** se não resolver

Solicitações de Desconexão de Emergência do Transportador

Passos:

1. **Dropar conexão imediatamente**:
    - Navegue até: SMPP → Status Ao Vivo
    - Encontre a conexão afetada
    - Clique em "Dropar Conexão"
2. **Documentar razão**:
    - Anote o nome do transportador
    - Registre hora e razão
    - Salve a correspondência
3. **Investigar o problema**:
    - Verifique padrões recentes de mensagens
    - Revise logs para erros
    - Identifique a causa raiz
4. **Coordenar resolução**:
    - Trabalhe com o transportador
    - Implemente correções
    - Teste antes de reconectar

Pico de Volume Alto

Sintomas: Tráfego de mensagens inesperadamente alto

Passos:

1. **Verificar limites de TPS**:
    - Navegue até: SMPP → Status Ao Vivo
    - Verifique se as conexões não estão sendo limitadas

- Pode ser necessário aumentar temporariamente os limites de TPS

2. **Monitorar estabilidade do transportador**:
   - Observe desconexões
   - Verifique taxas de sucesso de entrega
3. **Coordenar com o backend**:
   - Verifique se a origem da mensagem é legítima
   - Pode ser necessário implementar limitação de taxa a montante
4. **Escalar se necessário**:
   - Pode ser necessário instâncias adicionais do gateway
   - Contate o suporte para conselhos sobre escalonamento

Melhores Práticas

Lista de Verificação Diária

- [ ] Verificar se todas as conexões SMPP estão conectadas
- [ ] Revisar logs de erro para quaisquer problemas
- [ ] Monitorar fila de mensagens para acúmulo
- [ ] Verificar painéis do Prometheus/Grafana
- [ ] Verificar taxas de sucesso de entrega > 98%

Tarefas Semanais

- [ ] Revisar tendências de métricas
- [ ] Verificar anomalias de padrões
- [ ] Testar procedimentos de recuperação de desastres
- [ ] Atualizar documentação conforme necessário
- [ ] Revisar e reconhecer alertas

Tarefas Mensais

- [ ] Auditoria de credenciais
- [ ] Revisão de planejamento de capacidade
- [ ] Atualizar contatos do transportador
- [ ] Revisar e otimizar configurações de TPS
- [ ] Fazer backup de arquivos de configuração

Documentação Relacionada

- **CONFIGURATION.md** - Configurar conexões e configurações
- **SOURCE_ADDRESS_WHITELIST.md** - Restringir endereços de origem por par de servidor
- **MONITORING.md** - Configurar alertas do Prometheus
- **TROUBLESHOOTING.md** - Resolver problemas comuns
- **README.md** - Visão geral do sistema
