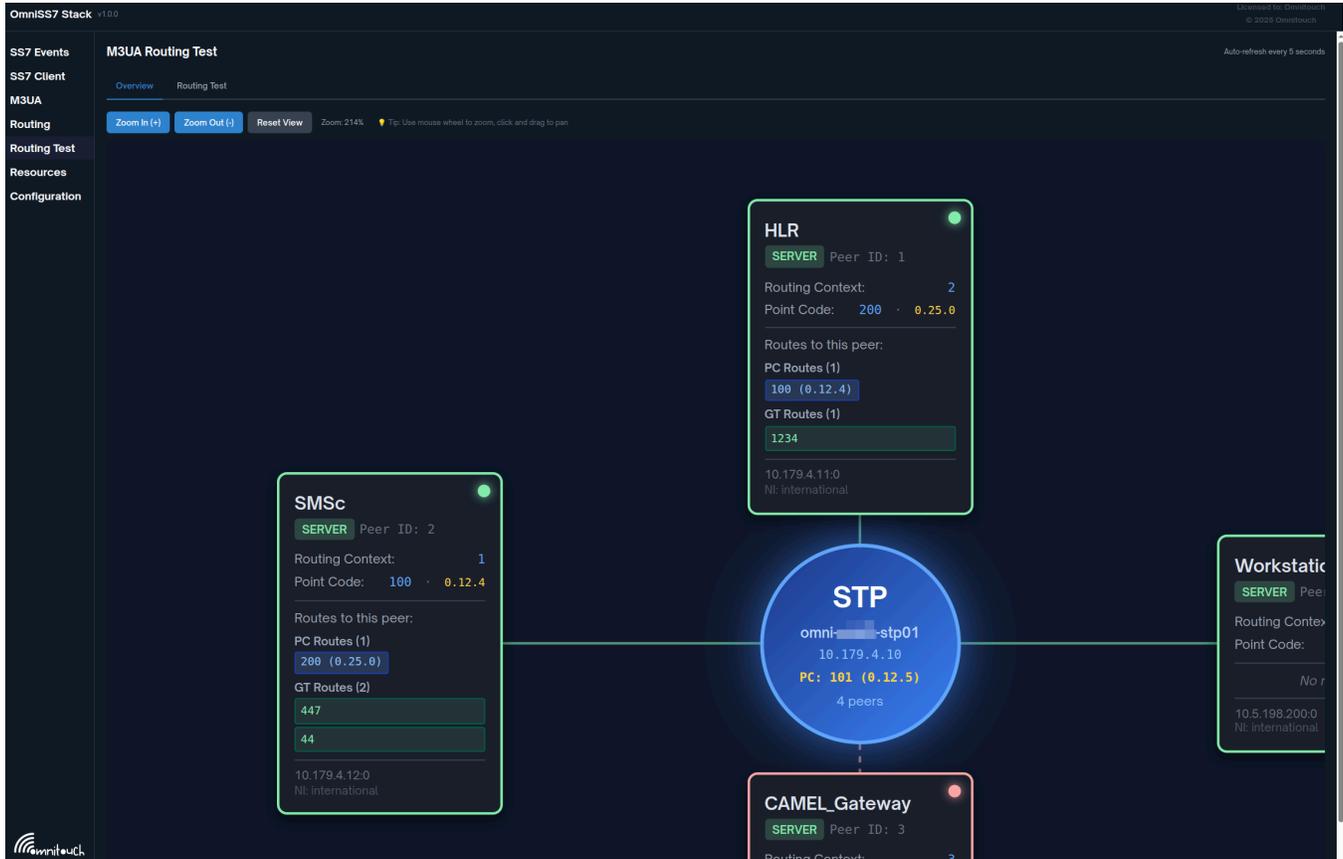




# OmniSS7 - Guía del Usuario

OmniSS7 de Omnitouch Network Services es un stack de señalización SS7 completo y de propósito general que proporciona funcionalidad flexible de elementos de red.



## Resumen de la Documentación

Esta documentación está organizada por el rol del elemento de red. Elija la guía que coincida con su implementación:

### ◆ Guías de Configuración

- [Guía STP](#) - Configuración del Punto de Transferencia de Señales
  - Enrutar tráfico SS7 entre pares de red
  - Enrutamiento de Código de Punto y Título Global
  - Balanceo de carga y ocultamiento de topología
  - **Utilice esto si está enrutando tráfico SS7 entre redes**
- [Guía del Cliente MAP](#) - Configuración del Cliente MAP
  - Conectarse como cliente M3UA para enviar solicitudes MAP
  - Consultas HLR, autenticación, información de enrutamiento
  - Soporte para el protocolo MAP genérico
  - **Utilice esto si está enviando solicitudes MAP a elementos de red**
- [Guía del Centro SMS](#) - Configuración del Centro de SMS (SMSG)
  - Enrutamiento y entrega de mensajes SMS
  - Cola de mensajes respaldada por base de datos
  - Auto-limpieza e informes de entrega
  - **Utilice esto si está operando un Centro de SMS**
- [Guía HLR](#) - Configuración del Registro de Localización de Suscriptores
  - Gestión de base de datos de suscriptores
  - Generación de vectores de autenticación
  - Actualizaciones de ubicación e información de enrutamiento
  - **Utilice esto si está operando un HLR/RSS**
- [Guía del Gateway CAMEL](#) - Configuración del Gateway CAMEL
  - Servicios de red inteligente (CAP/CAMEL)
  - Control de llamadas en tiempo real y facturación
  - Integración OCS para facturación
  - Constructor de solicitudes interactivas y monitoreo de sesiones
  - **Utilice esto si está proporcionando servicios IN o facturación en tiempo real**

### ◆ Características Comunes

- [Guía de Características Comunes](#) - Componentes Compartidos
  - Resumen y configuración de la interfaz web
  - Documentación de API
  - Monitoreo y métricas (Prometheus)
  - Mejores prácticas y solución de problemas

### ◆ Documentación de Referencia

- [Apéndice](#) - Referencia Técnica
  - Especificaciones del protocolo SS7
  - Códigos de operación MAP
  - Flujos de transacción TCAP
  - Codificaciones y formatos de caracteres

## Inicio Rápido

### 1. Resumen del Sistema

OmniSS7 puede operar en diferentes modos dependiendo de sus requisitos de red:

### 3. Configuración

OmniSS7 puede funcionar en 5 modos operativos diferentes. El archivo de configuración config/runtime.exe contiene ejemplos completos y listos para usar.

Para cambiar de modo:

1. Abra config/runtime.exe
2. Descomente la sección de configuración deseada (STP, HLR, SMSG o CAMEL GW)
3. Comente las otras secciones
4. Actualice las direcciones IP y las URL de la API según sea necesario
5. Reinicie la aplicación

-- Consulte las guías específicas del modo a continuación para obtener instrucciones completas de configuración

Ejemplos de configuraciones en runtime. exs:

Modo STP:

```
config :ommiss7,  
  map_client_enabled: true,  
  hlr_mode_enabled: false,  
  smsc_mode_enabled: false,  
  camelgw_mode_enabled: false,  
  map_client_m3ua: %{\...}
```

Modo HLR:

```
config :ommiss7,  
  map_client_enabled: false,  
  hlr_mode_enabled: true,  
  smsc_mode_enabled: false,  
  camelgw_mode_enabled: false,  
  hlr_api_base_url: "...",  
  map_client_m3ua: %{\...}
```

Modo SMS:

```
config :ommiss7,  
  map_client_enabled: true,  
  hlr_mode_enabled: false,  
  smsc_mode_enabled: true,  
  camelgw_mode_enabled: false,  
  smsc_api_base_url: "...",  
  auto_flush_enabled: true,  
  map_client_m3ua: %{\...}
```

Modo Gateway CAMEL:

```
config :ommiss7,  
  cap_client_enabled: true,  
  camelgw_mode_enabled: true,  
  ocs_enabled: true,  
  ocs_url: "http://your-ocs-server/api/charging",  
  cap_version: :v2, # Versión CAP: :v1, :v2, :v3, o :v4  
  cap_client_m3ua: %{\...}
```

#### 4. Acceso a la Interfaz Web

Navegue a <http://localhost> (o su nombre de host configurado)

## Arquitectura del Sistema

### Matriz de Características

Característica	Modo STP	Cliente MAP	Modo SMS	Modo HLR	Gateway CAMEL
Enrutamiento de Código de Punto	◊	◊	◊	◊	◊
Enrutamiento de Título Global	◊	◊	◊	◊	◊
Reescritura de SSN	◊	◊	◊	◊	◊
Soporte Multi-Peer	◊	◊	◊	◊	◊
Solicitudes MAP (Enviar)	◊	◊	◊	◊	◊
Respuestas MAP (Recibir)	◊	◊	◊	◊	◊
Gestión de Cola de SMS	◊	◊	◊	◊	◊
Auto-Limpieza de SMS	◊	◊	◊	◊	◊
Base de Datos de Suscriptores	◊	◊	◊	◊	◊
Vectores de Autenticación	◊	◊	◊	◊	◊
Actualizaciones de Ubicación	◊	◊	◊	◊	◊
Soporte CAP/CAMEL	◊	◊	◊	◊	◊
Facturación en Tiempo Real	◊	◊	◊	◊	◊
Control de Llamadas (Servicios IN)	◊	◊	◊	◊	◊
Interfaz Web	◊	◊	◊	◊	◊
API REST	◊	◊	◊	◊	◊
Métricas de Prometheus	◊	◊	◊	◊	◊

### Operaciones Comunes

#### Acceso a la Interfaz Web

- URL: <http://localhost> (o nombre de host configurado)
- API Swagger: <http://localhost/swagger>
- Métricas: <http://localhost/metrics>

#### Monitoreo

```
# Verificar estado de pares M3UA  
curl http://localhost/api/m3ua-status
```

```
# Ver métricas de Prometheus  
curl http://localhost/metrics
```

```
# Verificar salud de la aplicación  
curl http://localhost/api/health
```

#### Registros

```
# Configurar nivel de registro en config/runtime.exs  
config :logger,  
  level: :debug # Opciones: :debug, :info, :warning, :error
```

### Capacidades Clave

- **Soporte Completo del Protocolo MAP** - Operaciones de MAP Fase 2/3
- **Soporte del Protocolo CAP/CAMEL** - CAP v1/v2/v3/v4 para servicios de red inteligente
- **Serialización M3UA/CTP** - Transporte SS7 basado en IP
- **Facturación en Tiempo Real** - Integración OCS para facturación prepago/postpago
- **Cola de Mensajes en Tiempo Real** - Entrega de SMS respaldada por base de datos
- **Constructor de Solicitudes Interactivas** - Interfaz web para pruebas CAMEL/CAP
- **Monitoreo de Sesiones** - Seguimiento de sesiones de llamadas CAMEL en tiempo real
- **Documentación de API Interactiva** - Interfaz Swagger para pruebas
- **Métricas de Prometheus** - Observabilidad completa
- **Configuración Multi-rol** - STP, Cliente MAP, SMS, HLR, Gateway CAMEL

### Resumen de la Pila de Protocolo

### Ejemplos de Casos de Uso

#### Gateway de Red (STP)

Enrutar tráfico SS7 entre diferentes redes móviles

- Conectar redes de operadores
- Gateway SS7 internacional
- Balanceo de carga entre HLRs
- Traducción de Títulos Globales
- NAT SSCP (Ruteo inteligente de Títulos Globales)
- [Guía STP](#)

#### Centro SMS (SMS)

Entregar mensajes SMS a suscriptores móviles

- Entrega de MT-SMS
- Origen de MO-SMS
- Enrutamiento de SMS a Casa
- Ocultamiento de IMSI
- Cortafuegos de SMS
- Gestión de colas de mensajes
- Informes de entrega
- Integrado en OmniMessage para manejar todos los SMS MAP
- [Guía del Centro SMS](#)

#### Cliente MAP

Interactuar con cualquier elemento de red a través de MAP utilizando una simple API RESTful

- PRN / SRI / ATI / etc.
- Construya sus propias aplicaciones SS7/MAP utilizando APIs RESTful
- Gateways USSD
- Solicitudes de vectores de autenticación
- Búsquedas de IMSI/MSISDN
- Consultas de información de enrutamiento
- [Guía del Cliente MAP](#)

#### Base de Datos de Suscriptores (HLR)

Gestionar datos de suscriptores y autenticación

- Actualizaciones de ubicación
- Generación de autenticación
- Provisión de información de enrutamiento
- Integración completa en OmniHSS
- → [Guía HLR](#)

#### Plataforma de Red Inteligente (Gateway CAMEL)

Control de llamadas y facturación en tiempo real para operadores de telecomunicaciones

- Facturación de llamadas prepago/postpago
- Control de llamadas (conectar, liberar, enrutar)
- Gestión de sesiones y generación de CDR
- Constructor de solicitudes interactivas para pruebas
- → [Guía del Gateway CAMEL](#)

---

## Soporte y Recursos

### Documentación

#### Guías de Configuración del Núcleo:

- [Guía de Configuración STP](#) - Enrutamiento del Punto de Transferencia de Señales
- [Guía de Configuración del Cliente MAP](#) - Cliente del protocolo MAP
- [Guía de Configuración del Centro SMS](#) - Enrutamiento y entrega de SMS
- [Guía de Configuración HLR](#) - Base de datos de suscriptores
- [Guía de Configuración del Gateway CAMEL](#) - Red inteligente y facturación

#### Integración y Referencia:

- [Guía del Constructor de Solicitudes CAMEL](#) - Herramienta de prueba interactiva
- [Guía de Características Comunes](#) - Componentes compartidos e interfaz web
- [Referencia Técnica](#) - Especificaciones del protocolo

### Información de Contacto

**Producto:** OmniSS7

**Fabricante:** Omnitouch Network Services

**Versión de Documentación:** 2.0

**Última Actualización:** 2025

Para soporte técnico, asistencia en la implementación o consultas de ventas, comuníquese con Omnitouch Network Services.

---

*Esta documentación cubre la operación en tiempo de ejecución de OmniSS7 y la funcionalidad para el usuario final. Para instalación, desarrollo o configuración avanzada, consulte la documentación técnica.*



# Guía de la API REST

[← Volver a la Documentación Principal](#)

Esta guía proporciona documentación completa para la **API REST** de OmniSS7 y **Swagger UI**.

## Tabla de Contenidos

1. [Descripción General](#)
  2. [Configuración del Servidor HTTP](#)
  3. [Swagger UI](#)
  4. [Puntos Finales de la API](#)
  5. [Autenticación](#)
  6. [Formatos de Respuesta](#)
  7. [Manejo de Errores](#)
  8. [Métricas \(Prometheus\)](#)
  9. [Ejemplos de Solicitudes](#)
- 

## Descripción General

OmniSS7 proporciona una API REST para el acceso programático a las operaciones de MAP (Mobile Application Part). La API permite:

- Enviar solicitudes MAP (SRI, SRI-for-SM, UpdateLocation, etc.)
- Recuperar respuestas MAP
- Monitorear métricas del sistema a través de Prometheus

## Arquitectura de la API

---

## Configuración del Servidor HTTP

### Detalles del Servidor

Parámetro	Valor	Configurable
<b>Protocolo</b>	HTTP	No
<b>Dirección IP</b>	0.0.0.0 (todas las interfaces)	Solo a través de código
<b>Puerto</b>	8080	Solo a través de código
<b>Transporte</b>	Plug.Cowboy	No

**URL de Acceso:** http://[server-ip]:8080

## Habilitar/Deshabilitar el Servidor HTTP

Controla si el servidor HTTP se inicia:

```
config :omniss7,  
  start_http_server: true # Establecer en false para deshabilitar
```

**Predeterminado:** true (habilitado)

**Cuando está Deshabilitado:** El servidor HTTP no se iniciará y la API REST/Swagger UI no estará disponible.

---

## Swagger UI

La API incluye un **Swagger UI** para documentación y pruebas interactivas de la API.

### Accediendo a Swagger UI

**URL:** http://[server-ip]:8080/swagger

#### Características:

- Documentación interactiva de la API
- Funcionalidad de prueba para probar puntos finales
- Esquemas de solicitud/respuesta
- Cargas útiles de ejemplo

### Swagger JSON

La especificación OpenAPI está disponible en:

**URL:** http://[server-ip]:8080/swagger.json

#### Casos de Uso:

- Importar en Postman u otros clientes de API
  - Generar bibliotecas de cliente
  - Automatización de documentación de API
- 

## Puntos Finales de la API

Todos los puntos finales de operaciones MAP siguen el patrón: POST

/api/{operation}

## Resumen de Puntos Finales

Punto Final	Método	Propósito	Tiempo de Espera
/api/sri	POST	Enviar Información de Enrutamiento	10s
/api/sri-for-sm	POST	Enviar Información de Enrutamiento para SM	10s
/api/send-auth-info	POST	Enviar Información de Autenticación	10s
/api/MT-forwardSM	POST	SMS Móvil Terminado	10s
/api/forwardSM	POST	Reenviar SMS	10s
/api/updateLocation	POST	Actualizar Ubicación	10s
/api/prn	POST	Proporcionar Número de Roaming	10s
/metrics	GET	Métricas de Prometheus	N/A
/swagger	GET	Swagger UI	N/A
/swagger.json	GET	Especificación OpenAPI	N/A

**Nota:** Todas las solicitudes MAP tienen un **tiempo de espera de 10 segundos codificado**.

---

## SendRoutingInfo (SRI)

Recuperar información de enrutamiento para establecer una llamada a un suscriptor móvil.

**Punto Final:** POST /api/sri

**Cuerpo de la Solicitud:**

```
{  
  "msisdn": "1234567890",  
  "gmsc": "5551234567"  
}
```

**Parámetros:**

Campo	Tipo	Requerido	Descripción
msisdn	String	Sí	MSISDN de la parte llamada
gmsc	String	Sí	Título Global del MSC de Puerta

**Respuesta (200 OK):**

```
{
  "result": {
    "imsi": "001001234567890",
    "msrn": "5551234999",
    "vlr_number": "5551234800",
    ...
  }
}
```

**Error** (504 Gateway Timeout):

```
{
  "error": "timeout"
}
```

**Ejemplo de cURL:**

```
curl -X POST http://localhost:8080/api/sri \
-H "Content-Type: application/json" \
-d '{
  "msisdn": "1234567890",
  "gmsc": "5551234567"
}'
```

---

## SendRoutingInfoForSM (SRI-for-SM)

Recuperar información de enrutamiento para entregar un SMS a un suscriptor móvil.

**Punto Final:** POST /api/sri-for-sm

**Cuerpo de la Solicitud:**

```
{
  "msisdn": "1234567890",
  "service_center": "5551234567"
}
```

**Parámetros:**

<b>Campo</b>	<b>Tipo Requerido</b>	<b>Descripción</b>
msisdn	String Sí	MSISDN de destino
service_center	String Sí	Título Global del Centro de Servicio

**Respuesta** (200 OK):

```
{
```

```
"result": {
  "imsi": "001001234567890",
  "msc_number": "5551234800",
  "location_info": {...},
  ...
}
```

### Ejemplo de cURL:

```
curl -X POST http://localhost:8080/api/sri-for-sm \
-H "Content-Type: application/json" \
-d '{
  "msisdn": "1234567890",
  "service_center": "5551234567"
}'
```

---

## SendAuthenticationInfo

Solicitar vectores de autenticación para un suscriptor.

**Punto Final:** POST /api/send-auth-info

### Cuerpo de la Solicitud:

```
{
  "imsi": "001001234567890",
  "vectors": 3
}
```

### Parámetros:

Campo	Tipo	Requerido	Descripción
imsi	String	Sí	IMSI del suscriptor
vectors	Integer	Sí	Número de vectores de autenticación a generar

### Respuesta (200 OK):

```
{
  "result": {
    "authentication_sets": [
      {
        "rand": "0123456789ABCDEF...",
        "xres": "...",
        "ck": "...",
        "ik": "...",
        "autn": "..."
      }
    ]
  }
}
```

```
    }
  ],
  ...
}
}
```

### Ejemplo de cURL:

```
curl -X POST http://localhost:8080/api/send-auth-info \
-H "Content-Type: application/json" \
-d '{
  "imsi": "001001234567890",
  "vectors": 3
}'
```

---

## MT-ForwardSM

Entregar un SMS Móvil Terminado a un suscriptor.

**Punto Final:** POST /api/MT-forwardSM

### Cuerpo de la Solicitud:

```
{
  "imsi": "001001234567890",
  "destination_service_centre": "5551234567",
  "originating_service_center": "5551234568",
  "smsPDU": "0001000A8121436587F900001C48656C6C6F20576F726C64"
}
```

### Parámetros:

	<b>Campo</b>	<b>Tipo Requerido</b>	<b>Descripción</b>
imsi		String Sí	IMSI del suscriptor de destino
destination_service_centre	String Sí		GT del centro de servicio de destino
originating_service_center	String Sí		GT del centro de servicio de origen
smsPDU		String Sí	SMS TPDU en formato hexadecimal

**Nota:** smsPDU debe ser una cadena codificada en hex (mayúsculas o minúsculas).

### Respuesta (200 OK):

```
{
  "result": {
```

```
    "delivery_status": "success",
    ...
  }
}
```

### Ejemplo de cURL:

```
curl -X POST http://localhost:8080/api/MT-forwardSM \
-H "Content-Type: application/json" \
-d '{
  "imsi": "001001234567890",
  "destination_service_centre": "5551234567",
  "originating_service_center": "5551234568",
  "smsPDU": "0001000A8121436587F900001C48656C6C6F20576F726C64"
}'
```

---

## ForwardSM

Reenviar un mensaje SMS (MO-SMS del suscriptor).

**Punto Final:** POST /api/forwardSM

**Cuerpo de la Solicitud:** Igual que MT-ForwardSM

### Ejemplo de cURL:

```
curl -X POST http://localhost:8080/api/forwardSM \
-H "Content-Type: application/json" \
-d '{
  "imsi": "001001234567890",
  "destination_service_centre": "5551234567",
  "originating_service_center": "5551234568",
  "smsPDU": "0001000A8121436587F900001C48656C6C6F20576F726C64"
}'
```

---

## UpdateLocation

Notificar al HLR del cambio de ubicación del suscriptor (registro VLR).

**Punto Final:** POST /api/updateLocation

**Cuerpo de la Solicitud:**

```
{
  "imsi": "001001234567890",
  "vlr": "5551234800"
```

```
}
```

### Parámetros:

Campo	Tipo	Requerido	Descripción
imsi	String	Sí	IMSI del suscriptor
vlr	String	Sí	Dirección del Título Global del VLR

### Respuesta (200 OK):

```
{  
  "result": {  
    "hlr_number": "5551234567",  
    "subscriber_data": {...},  
    ...  
  }  
}
```

**Nota:** En modo HLR, esto activa la secuencia InsertSubscriberData (ISD) con un tiempo de espera de 10 segundos por ISD.

### Ejemplo de cURL:

```
curl -X POST http://localhost:8080/api/updateLocation \  
-H "Content-Type: application/json" \  
-d '{  
  "imsi": "001001234567890",  
  "vlr": "5551234800"  
}'
```

---

## ProvideRoamingNumber (PRN)

Solicitar MSRN (Número de Roaming de Estación Móvil) para el enrutamiento de llamadas a un suscriptor en roaming.

**Punto Final:** POST /api/prn

### Cuerpo de la Solicitud:

```
{  
  "msisdn": "1234567890",  
  "gmsc": "5551234567",  
  "msc_number": "5551234800",  
  "imsi": "001001234567890"  
}
```

### Parámetros:

<b>Campo</b>	<b>Tipo</b>	<b>Requerido</b>	<b>Descripción</b>
msisdn	String	Sí	MSISDN del suscriptor
gmsc	String	Sí	GT del MSC de Puerta
msc_number	String	Sí	Número de MSC para el suscriptor
imsi	String	Sí	IMSI del suscriptor

**Respuesta** (200 OK):

```
{
  "result": {
    "msrn": "5551234999",
    ...
  }
}
```

**Ejemplo de cURL:**

```
curl -X POST http://localhost:8080/api/prn \
-H "Content-Type: application/json" \
-d '{
  "msisdn": "1234567890",
  "gmsc": "5551234567",
  "msc_number": "5551234800",
  "imsi": "001001234567890"
}'
```

---

## Autenticación

**Estado Actual:** La API **no requiere autenticación**.

**Consideraciones de Seguridad:**

- La API está destinada para uso en redes internas/confiables
- Considerar el uso de reglas de firewall para restringir el acceso
- Para implementaciones en producción, considerar implementar middleware de autenticación

---

## Formatos de Respuesta

Todas las respuestas utilizan el formato **JSON**.

**Respuesta de Éxito**

**Estado HTTP:** 200 OK

## Estructura:

```
{
  "result": {
    // Datos de respuesta específicos de la operación
  }
}
```

## Respuesta de Error

### Estado HTTP:

- 400 Bad Request - Cuerpo de solicitud no válido
- 504 Gateway Timeout - Tiempo de espera de solicitud MAP (10 segundos)
- 404 Not Found - Punto final no válido

### Estructura:

```
{
  "error": "timeout"
}
```

o

```
{
  "error": "invalid request"
}
```

---

## Manejo de Errores

### Errores Comunes

Error	Código HTTP	Descripción	Solución
JSON Inválido	400	El cuerpo de la solicitud no es un JSON válido	Verificar la sintaxis JSON
Campos Faltantes	400	Faltan campos requeridos	Incluir todos los parámetros requeridos
Tiempo de Espera	504	La solicitud MAP excedió el tiempo de espera de 10s	Verificar conectividad M3UA, disponibilidad de HLR/VLR
No Encontrado	404	Punto final no válido	Verificar la URL del punto final

### Comportamiento de Tiempo de Espera

Todas las solicitudes MAP tienen un **tiempo de espera de 10 segundos**

## codificado:

1. Solicitud enviada al MapClient GenServer
2. Espera la respuesta hasta 10 segundos
3. Si no hay respuesta → devuelve 504 Gateway Timeout
4. Si se recibe respuesta → devuelve 200 OK con resultado

## Solución de Problemas de Tiempos de Espera:

- Verificar el estado de conexión M3UA (Interfaz Web → página M3UA)
- Verificar que el elemento de red (HLR/VLR/MSC) sea accesible
- Verificar la configuración de enrutamiento
- Revisar los registros de eventos SS7 en busca de errores

---

## Métricas (Prometheus)

La API expone métricas de Prometheus para monitoreo.

### Punto Final de Métricas

**URL:** `http://[server-ip]:8080/metrics`

**Formato:** Formato de texto de Prometheus

### Ejemplo de Salida:

```
# HELP map_requests_total Total de solicitudes MAP
# TYPE map_requests_total counter
map_requests_total{operation="sri"} 42
map_requests_total{operation="sri_for_sm"} 158
map_requests_total{operation="updateLocation"} 23

# HELP cap_requests_total Total de solicitudes CAP
# TYPE cap_requests_total counter
cap_requests_total{operation="initialDP"} 87
cap_requests_total{operation="requestReportBCSMEEvent"} 91

# HELP map_request_duration_milliseconds Duración de solicitudes/
respuestas MAP en ms
# TYPE map_request_duration_milliseconds histogram
map_request_duration_milliseconds_bucket{operation="sri",le="10"} 5
map_request_duration_milliseconds_bucket{operation="sri",le="50"} 12
map_request_duration_milliseconds_bucket{operation="sri",le="100"} 35
...

# HELP map_pending_requests Número de solicitudes MAP TID pendientes
# TYPE map_pending_requests gauge
```

```
map_pending_requests 3
```

## Métricas Disponibles

Métrica	Tipo	Etiquetas	Descripción
map_requests_total	Contador	operation	Número total de solicitudes MAP por tipo de operación
cap_requests_total	Contador	operation	Número total de solicitudes CAP por tipo de operación
map_request_duration_milliseconds	Histograma	operation	Duración de la solicitud en milisegundos
map_pending_requests	Gauge	-	Número de transacciones MAP pendientes

## Configuración de Prometheus

Agregue a su `prometheus.yml`:

```
scrape_configs:  
  - job_name: 'omniss7'  
    static_configs:  
      - targets: ['server-ip:8080']  
    metrics_path: '/metrics'  
    scrape_interval: 15s
```

---

## Ejemplos de Solicitudes

### Ejemplo en Python

```
import requests  
import json  
  
# Solicitud SRI-for-SM  
url = "http://localhost:8080/api/sri-for-sm"  
payload = {  
    "msisdn": "1234567890",  
    "service_center": "5551234567"  
}  
  
response = requests.post(url, json=payload, timeout=15)
```

```

if response.status_code == 200:
    result = response.json()
    print(f"Éxito: {result}")
elif response.status_code == 504:
    print("Tiempo de espera - sin respuesta de la red")
else:
    print(f"Error: {response.status_code} - {response.text}")

```

## Ejemplo en JavaScript

```

const axios = require('axios');

async function sendSRI() {
  try {
    const response = await axios.post('http://localhost:8080/api/
sri', {
      msisdn: '1234567890',
      gmsc: '5551234567'
    }, {
      timeout: 15000
    });

    console.log('Éxito:', response.data);
  } catch (error) {
    if (error.code === 'ECONNABORTED') {
      console.error('Tiempo de espera - sin respuesta de la red');
    } else {
      console.error('Error:', error.response?.data || error.message);
    }
  }
}

sendSRI();

```

## Ejemplo en Bash/cURL

```

#!/bin/bash

# Solicitud UpdateLocation
response=$(curl -s -w "\n%{http_code}" -X POST http://localhost:8080/
api/updateLocation \
  -H "Content-Type: application/json" \
  -d '{
    "imsi": "001001234567890",
    "vlr": "5551234800"
  }')

http_code=$(echo "$response" | tail -n 1)

```

```
body=$(echo "$response" | sed '$d')

if [ "$http_code" -eq 200 ]; then
    echo "Éxito: $body"
elif [ "$http_code" -eq 504 ]; then
    echo "Tiempo de espera - sin respuesta de la red"
else
    echo "Error $http_code: $body"
fi
```

---

## Diagramas de Flujo

### Flujo de Solicitudes de API

---

## Resumen

La API REST de OmniSS7 proporciona:

- ❖ **Operaciones MAP** - Soporte completo para SRI, SRI-for-SM, UpdateLocation, entrega de SMS, autenticación
- ❖ **Swagger UI** - Documentación y pruebas interactivas de la API
- ❖ **Métricas de Prometheus** - Monitoreo y observabilidad
- ❖ **Tiempos de Espera Codificados** - Tiempo de espera de 10 segundos para todas las solicitudes MAP
- ❖ **Servidor HTTP** - Se ejecuta en el puerto 8080 (configurable a través de `start_http_server`)

Para acceso a la Interfaz Web, consulte la [Guía de la Interfaz Web](#).

Para detalles de configuración, consulte la [Referencia de Configuración](#).

# Referencia Técnica (Apéndice)

[← Volver a la Documentación Principal](#)

Referencia técnica para los protocolos SS7 y la implementación de OmniSS7.

## Pila de Protocolos SS7

---

### Códigos de Operación MAP

Operación	Opcode	Propósito
updateLocation	2	Registrar ubicación del suscriptor
cancelLocation	3	Deregistrar del VLR
provideRoamingNumber	4	Solicitar MSRN
sendRoutingInfo	22	Consultar enrutamiento de llamadas
mt-forwardSM	44	Entregar SMS al suscriptor
sendRoutingInfoForSM	45	Consultar enrutamiento de SMS
mo-forwardSM	46	Reenviar SMS del suscriptor
sendAuthenticationInfo	56	Solicitar vectores de autenticación

---

### Tipos de Mensajes TCAP

- **BEGIN** - Iniciar transacción
  - **CONTINUE** - Medio de transacción
  - **END** - Respuesta final
  - **ABORT** - Cancelar transacción
- 

### Direccionamiento SCCP

#### Formatos de Título Global

- **E.164** - Número de teléfono internacional (por ejemplo, 447712345678)
- **E.212** - Formato IMSI (por ejemplo, 234509876543210)
- **E.214** - Formato de código de punto

#### Números de Subsistema (SSN)

- **SSN 6**: HLR

- **SSN 7:** VLR
  - **SSN 8:** MSC/SMSC
  - **SSN 9:** GMLC
  - **SSN 10:** SGSN
- 

## SMS TPDU

### Tipos de Mensajes

- **SMS-DELIVER (MT)** - De la red al móvil
- **SMS-SUBMIT (MO)** - Del móvil a la red
- **SMS-STATUS-REPORT** - Estado de entrega
- **SMS-COMMAND** - Comando remoto

### Codificaciones de Caracteres

- **GSM7** - Alfabeto GSM de 7 bits (160 caracteres por SMS)
  - **UCS2** - Unicode de 16 bits (70 caracteres por SMS)
  - **8-bit** - Datos binarios (140 bytes por SMS)
- 

## Estados M3UA

- **DOWN** - Sin conexión SCTP
  - **CONNECTING** - Conectando SCTP
  - **ASPUP\_SENT** - Esperando ACK de ASPUP
  - **INACTIVE** - ASP activo pero no en uso
  - **ASPAC\_SENT** - Esperando ACK de ASPAC
  - **ACTIVE** - Listo para tráfico
- 

## Códigos de Punto Comunes SS7

Los códigos de punto son típicamente valores de 14 bits (ITU) o 24 bits (ANSI).

### Formato de Ejemplo (ITU):

- Red: 3 bits
  - Clúster: 8 bits
  - Miembro: 3 bits
-

## Códigos de Error SCCP

- **0** - Sin traducción para la dirección
  - **1** - Sin traducción para dirección específica
  - **2** - Congestión del subsistema
  - **3** - Fallo del subsistema
  - **4** - Usuario no equipado
  - **5** - Fallo de MTP
  - **6** - Congestión de la red
  - **7** - No calificado
  - **8** - Error en el transporte del mensaje
- 

## Códigos de Error MAP

<b>Código</b>	<b>Error</b>	<b>Descripción</b>
1	unknownSubscriber	Suscriptor no en HLR
27	absentSubscriber	Suscriptor no accesible
34	systemFailure	Fallo de red
35	dataMissing	Datos requeridos no disponibles
36	unexpectedDataValue	Valor de parámetro inválido

---

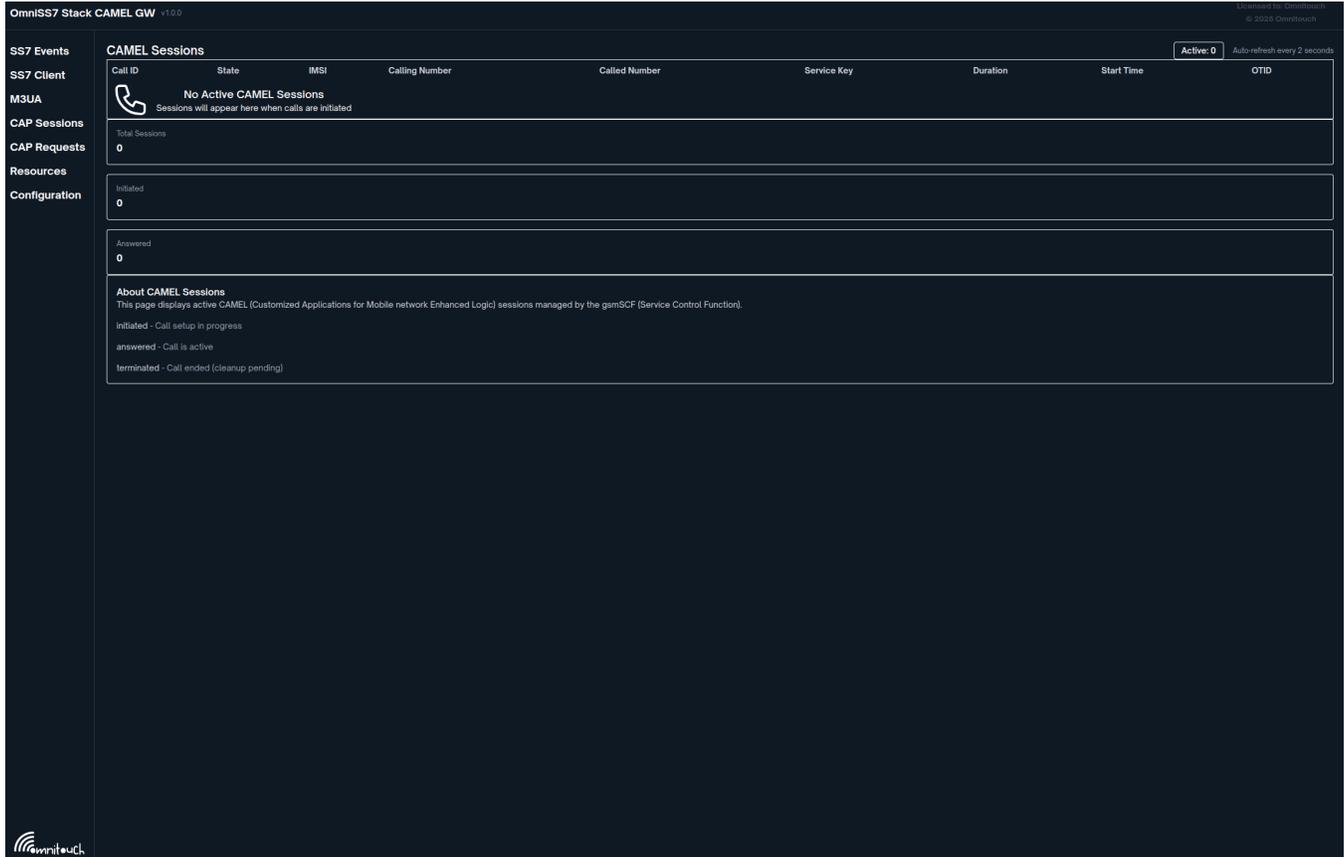
## Documentación Relacionada

- [← Volver a la Documentación Principal](#)
  - [Guía STP](#)
  - [Guía del Cliente MAP](#)
  - [Guía del Centro SMS](#)
  - [Guía HLR](#)
  - [Características Comunes](#)
-

# Guía de Configuración del Gateway CAMEL

## Descripción general

El modo **Gateway CAMEL (CAMEL GW)** transforma OmniSS7 en una plataforma de Red Inteligente (IN) que proporciona servicios de control de llamadas y de cobro en tiempo real utilizando el protocolo de Parte de Aplicación CAMEL (CAP).



### ¿Qué es CAMEL?

**CAMEL** (Aplicaciones Personalizadas para Lógica Mejorada de Redes Móviles) es un conjunto de estándares diseñados para trabajar en una red central GSM o en una red UMTS. Permite a los operadores proporcionar servicios que requieren control en tiempo real de las llamadas, tales como:

- **Llamadas prepago** - Verificación y cobro de saldo en tiempo real
- **Servicios de tarifa premium** - Facturación especial para servicios de valor agregado
- **Control de enrutamiento de llamadas** - Enrutamiento dinámico de destinos basado en tiempo/ubicación
- **Redes privadas virtuales** - Planes de numeración corporativo
- **Filtrado de llamadas** - Permitir/bloquear llamadas según criterios

### Versiones del Protocolo CAP

OmniSS7 CAMEL GW soporta múltiples versiones de CAP:

Versión	Fase	Características
CAP v1	Fase CAMEL 1	Control básico de llamadas, operaciones limitadas
CAP v2	Fase CAMEL 2	Operaciones mejoradas, soporte para SMS
CAP v3	Fase CAMEL 3	Soporte para GPRS, operaciones adicionales
CAP v4	Fase CAMEL 4	Características avanzadas, soporte multimedia

**Por defecto:** CAP v2 (más ampliamente desplegado)

## Arquitectura

### Ejemplo de Flujo de Llamadas

## Configuración

### Requisitos Previos

- OmniSS7 instalado y en funcionamiento
- Conectividad M3UA a MSC/GMSC (gsmSSF)
- Sistema de Cobro en Línea (OCS) con punto final API (opcional, para cobro en tiempo real)

### Habilitar el Modo Gateway CAMEL

Edita config/runtime.exe y configura la sección del Gateway CAMEL:

```
config:omni7,
# Flags de modo - Habilitar características CAP/CAMEL
cap_client_enabled: true,
camelgw_mode_enabled: true,

# Deshabilitar otros modos
map_client_enabled: false,
hlr_mode_enabled: false,
smc_mode_enabled: false,

# Configuración de Versión CAP/CAMEL
# Determina qué versión de CAP usar para solicitudes salientes y diálogo
# Opciones: :v1, :v2, :v3, :v4
cap_version: :v2,

# Integración OCS (para cobro en tiempo real)
ocs_enabled: true,
ocs_url: "http://your-ocs-server/api/charging",
ocs_timeout: 5000, # milisegundos
ocs_auth_token: "your-api-token" # Opcional, si OCS requiere autenticación

# Configuración de Conexión M3UA para CAMEL
# Conectar como ASP (Proceso de Servidor de Aplicaciones) para operaciones CAP
cap_client_m3ua: %{\n
  mode: "ASP",\n
  callback: {CapClient, :handle_payload, []},\n
  process_name: :camelgw_client_esp,\n
}

# Punto final local (sistema CAMEL GW)\n
local_ip: {10, 179, 4, 13},\n
local_port: 2905,
```

```

# Punto final remoto (MSC/GMSC - gsmSSF)
remote_ip: {10, 179, 4, 10},
remote_port: 2905,

# Parámetros M3UA
routing_context: 1,
network_appearance: 0,
asp_identifier: 13
}

```

## Configurar Páginas de Interfaz Web

La Interfaz Web incluye páginas especializadas para operaciones CAMEL:

```

config :control_panel,
  use_additional_pages: [
    {557.Web.EventsLive, "/events", "Eventos 557"},
    {557.Web.TestClientLive, "/client", "Cliente 557"},
    {557.Web.M3UAStatusLive, "/m3ua", "M3UA"},
    {557.Web.CAMELSessionsLive, "/camel_sessions", "Sesiones CAP"},
    {557.Web.CAMELRequestLive, "/camel_request", "Solicitudes CAP"}
  ],
  page_order: ["/events", "/client", "/m3ua", "/camel_sessions",
               "/camel_request", "/application", "/configuration"]

```

## Operaciones CAP Soportadas

### Operaciones Entrantes (de gsmSSF – gsmSCF)

Operación	Opcodo	Descripción	Handler
InitialDP	0	Punto de Detección Inicial - notificación de configuración de llamada	handle_initial_dp/1
EventReportBCSM	6	Evento del Modelo de Estado de Llamadas básico (respuesta, desconexión, etc.)	handle_event_report_bcs/1
ApplyChargingRequest	71	Informe de cobro desde gsmSSF	handle_apply_charging_report/1
AssistRequestInstructions	16	Solicitud de asistencia desde gsmSRF	handle_assist_request_instructions/1

### Operaciones Salientes (de gsmSCF – gsmSSF)

Operación	Opcodo	Descripción	Generador
Connect	20	Conectar llamada al número de destino	CapRequestGenerator.connect_request/2
Continue	31	Continuar el procesamiento de la llamada sin modificación	CapRequestGenerator.continue_request/1
ReleaseCall	22	Liberar/terminar la llamada	CapRequestGenerator.release_call_request/2
RequestReportBCSMEvent	23	Solicitar notificación de eventos de llamada	CapRequestGenerator.request_report_bcsme_event_request/2
ApplyCharging	35	Aplicar cobro a la llamada	CapRequestGenerator.apply_charging_request/3

## Características de la Interfaz Web

### Página de Sesiones CAMEL

URL: [http://localhost/camel\\_sessions](http://localhost/camel_sessions)

Monitorio en tiempo real de sesiones de llamadas CAMEL activas:

#### Características:

- **Lista de sesiones en vivo** - Se actualiza automáticamente cada 2 segundos
- **Detalles de la sesión** - OTID, ID de llamada, Estado, Duración
- **Versión CAP** - Muestra la versión del protocolo (CAP v1/v2/v3/v4) detectada desde InitialDP
- **Información de la llamada** - IMSI, Número A, Número B, Clave de Servicio
- **Seguimiento de estado** - Iniciada, Respondida, Terminada
- **Temporizador de duración** - Muestra en tiempo real la duración de la llamada

#### Columnas de la Tabla:

- ID de llamada, Estado, Versión, IMSI, Número Llamante, Número Llamado, Clave de Servicio, Duración, Hora de Inicio, OTID

#### Estados de la Sesión:

- **Iniciada** - InitialDP recibido, esperando respuesta
- **Respondida** - Llamada respondida, cobro en progreso
- **Terminada** - Llamada finalizada, CDR generado

**Detección de Versión CAP:** El sistema detecta automáticamente la versión del protocolo CAP desde la parte de diálogo de InitialDP y la muestra en la columna de Versión. Esto ayuda a identificar qué versión de CAP está utilizando cada MSC.

### Constructor de Solicitudes CAMEL

URL: [http://localhost/camel\\_request](http://localhost/camel_request)

Herramienta interactiva para construir y enviar solicitudes CAP.

#### Características:

- **Selector de tipo de solicitud** - InitialDP, Connect, ReleaseCall, etc.
- **Campos de formulario dinámicos** - Se adapta al tipo de solicitud seleccionado
- **Opciones SSCP/M3UA** - Configuración avanzada de direccionamiento
- **Historial de solicitudes** - Últimas 20 solicitudes con estado
- **Seguimiento de sesión** - Mantiene OTID para solicitudes de seguimiento
- **Retroalimentación en tiempo real** - Mensajes de éxito/error

#### Tipos de Solicitudes:

- InitialDP** - Iniciar nueva sesión de llamada
  - Clave de Servicio (entero)
  - Número Llamante (A-party)
  - Número Llamado (B-party)
- Connect** - Enrutar llamada al destino
  - Número de Destino
- ReleaseCall** - Terminar llamada
  - Código de Causa (16=Normal, 17=Ocupado, 31=No Especificado)
- RequestReportBCSMEvent** - Solicitar notificaciones de eventos
  - Eventos: oAnswer, oDisconnect, tAnswer, tDisconnect
- Continue** - Continuar llamada sin modificación
  - No se requieren parámetros
- ApplyCharging** - Aplicar límites de duración de llamada
  - Duración (segundos, 1-864000)
  - Liberar en Tiempo de Espera (booleano)
  - Ver [Guía del Constructor de Solicitudes CAMEL](#) para uso detallado

#### Opciones SSCP Avanzadas:

- Título Global de la Parte Llamada
- Título Global de la Parte Llamante
- SSN Llamado (por defecto: 146 = gsmSSF)
- SSN Llamante (por defecto: 146)

#### Opciones M3UA:

- OPC (Código de Punto de Origen, por defecto: 5013)
- DPC (Código de Punto de Destino, por defecto: 5011)

## Integración con OCS

### Ciclo de Vida de la Llamada con Cobro

#### 1. Inicio de Llamada (InitialDP)

Cuando el MSC envía InitialDP, CAMELGV:

- Detecta la versión CAP** - Examina la parte de diálogo para identificar CAP v1/v2/v3/v4
- Decodifica el mensaje CAP** - Extrae IMSI, números llamantes/llamados
- Llama a OCS** - API InitiateSession
- Recibe autorización** - MaxUsage (por ejemplo, 30 segundos)
- Almacena la sesión** - En SessionStore (tabla ETS) con versión CAP
- Responde al MSC** - RequestReportBCSMEvent + Continue (usando la misma versión CAP)

#### Ejemplo:

```

# Datos decodificados de InitialDP
%{
  imsi: "310150123456789",
  calling_party_number: "14155551234",
  called_party_number: "14155556789",
  service_key: 1,
  msc_address: "19216800123",
  cap_version: :v2 # Detectado desde el diálogo
}

# Respuesta de OCS
{:ok, %{max_usage: 30}} # 30 segundos autorizados

```

```
# Entrada en SessionStore
%(
  call_id: "CAMEL-48080173",
  initial_dp_data: %(.),
  cap_version: v2, # Almacenado para la generación de respuestas
  start_time: 1730246400,
  state: :initiated
)
```

## 2. Respuesta de Llamada (EventReportBCSM - oAnswer)

Cuando la llamada es respondida:

1. **Recibe el evento oAnswer** - Desde el MSC
2. **Actualiza OCS** - UpdateSession con usage=0
3. **Inicia el bucle de débito** - OCS comienza a cobrar
4. **Actualiza el estado de la sesión** - :answered en SessionStore
5. **Continúa la llamada** - Envía Continue al MSC

## 3. Actualizaciones Periódicas (Opcional)

Para llamadas largas, solicitar crédito adicional:

```
# Cada 30 segundos
OCS.Client.update_session(call_id, %(), current_usage)
```

Si MaxUsage devuelve 0, el suscriptor no tiene crédito – Enviar ReleaseCall

## 4. Terminación de Llamada (EventReportBCSM - oDisconnect)

Cuando la llamada termina:

1. **Recibe el evento oDisconnect** - Desde el MSC
2. **Calcula la duración total** - Desde el tiempo de inicio de la sesión
3. **Termina la sesión OCS** - API TerminateSession
4. **CDR generado** - Por OCS con costo final
5. **Limpia la sesión** - Elimina de SessionStore
6. **Envía ReleaseCall** - Confirma la terminación al MSC

## Análisis de CDR

Los CDR son generados por tu OCS y típicamente incluyen:

### Campos de CDR desde CAMEL:

- Account - IMSI o número llamante
- Destination - Número de la parte llamada
- OriginID - Identificador único de la llamada (CAMEL-OTID)
- Usage - Duración total de la llamada (segundos)
- Cost - Costo calculado
- IMSI - IMSI del suscriptor
- CallingPartyNumber - Parte A
- CalledPartyNumber - Parte B
- MSCAddress - Código de punto del MSC que sirve
- ServiceKey - Clave de servicio CAMEL

## Pruebas

### Pruebas Manuales con el Constructor de Solicitudes

#### 1. Navegar al Constructor de Solicitudes:

```
http://localhost/camel_request
```

#### 2. Enviar InitialDP:

- Seleccionar "InitialDP" del menú desplegable
- Clave de Servicio: 100
- Número Llamante: 14155551234
- Número Llamado: 14155556789
- Hacer clic en "Enviar Solicitud InitialDP"
- Anotar el OTID generado

#### 3. Monitorear la Sesión:

- Abrir nueva pestaña: [http://localhost/camel\\_sessions](http://localhost/camel_sessions)
- Ver sesión activa con estado "Iniciada"

#### 4. Simular Respuesta de Llamada:

- Regresar al Constructor de Solicitudes
- Seleccionar "EventReportBCSM"
- Tipo de Evento: oAnswer
- Hacer clic en "Enviar Solicitud EventReportBCSM"
- El estado de la sesión cambia a "Respondida"

#### 5. Finalizar Llamada:

- Seleccionar "ReleaseCall"
- Código de Causa: 16 (Normal)
- Hacer clic en "Enviar Solicitud ReleaseCall"
- El estado de la sesión cambia a "Terminada"

### Pruebas con MSC Real

#### Configurar Servicio CAMEL en MSC

En tu MSC/VLR, configura el servicio CAMEL:

```
# Ejemplo de configuración de MSC Huawei
ADD CAMELSERVICE;
SERVICEID=1,
SERVICEKEY=100,
GSMSCFADDR="55512341234", # Título Global de CAMELGW
DEFAULTCALLHANDLING=CONTINUE;

ADD CAMELSUBSCRIBER;
IMSI="310150123456789",
SERVICEID=1,
TRIGGERTYPE=TERMCALL;
```

#### Monitorear Registros

Observa los registros de CAMELGW para mensajes CAP entrantes:

```
# Ver registros en tiempo real
tail -f /var/log/omiss7/omiss7.log

# Filtrar eventos CAP
grep "CAP:" /var/log/omiss7/omiss7.log

# Ver registro de eventos (formato JSON)
curl http://localhost/api/events | jq '.[] | select(.map_event | startswith("CAP:"))'
```

### Pruebas de Carga

Utiliza el Constructor de Solicitudes en un bucle para pruebas de carga:

```
# Enviar 100 solicitudes InitialDP
for i in $(seq 1 100); do
  curl -X POST http://localhost/api/camel/initial_dp \
  -H "Content-Type: application/json" \
  -d '{
    "service_key": 100,
    "calling_number": "14155551234",
    "called_number": "14155556789"
  }';
  sleep 0.1
done
```

## Monitoreo y Operaciones

### Métricas de Prometheus

CAMELGW expone métricas en <http://localhost:8080/metrics>:

#### Métricas específicas de CAP:

- `cap_requests_total{operation}` - Total de solicitudes CAP por tipo de operación (por ejemplo, initialDP, requestReportBCSMEvent)

#### Métricas adicionales de MAP/API:

- `map_requests_total{operation}` - Total de solicitudes MAP por tipo de operación
- `map_request_duration_milliseconds{operation}` - Histograma de duración de solicitudes
- `map_pending_requests` - Número de transacciones MAP pendientes

#### Métricas de STP M3UA (si el modo STP está habilitado):

- `m3ua_stp_messages_received_total{peer_name,point_code}` - Mensajes recibidos de pares
- `m3ua_stp_messages_sent_total{peer_name,point_code}` - Mensajes enviados a pares
- `m3ua_stp_routing_failures_total{reason}` - Fallos de enrutamiento por razón

#### Ejemplos de consultas:

```
# Solicitudes CAP
curl http://localhost:8888/metrics | grep cap_requests_total

# Total de InitialDP recibidos
curl http://localhost:8888/metrics | grep 'cap_requests_total{operation="initialDP"}'

# Solicitudes MAP pendientes
curl http://localhost:8888/metrics | grep map_pending_requests
```

#### Comprobaciones de Salud

```
# Comprobar conectividad M3UA
curl http://localhost/api/m3ua-status

# Comprobar conectividad OCS
curl http://localhost/api/ocs-status

# Comprobar sesiones activas
curl http://localhost/api/camel/sessions/count
```

#### Configuración de Registros

Ajusta el nivel de registro en config/runtime.exs:

```
config :logger,
  level: :info, # Opciones: :debug, :info, :warning, :error

# Habilitar registro de depuración CAP
config :logger, :console,
  metadata: [:cap_operation, :otid, :call_id]
```

---

## Solución de Problemas

### Problema: No se reciben mensajes CAP

**Síntomas:** El Constructor de Solicitudes funciona, pero el MSC no envía InitialDP

#### Verificar:

- Estado del enlace M3UA: `curl http://localhost/api/m3ua-status`
- Configuración del servicio CAMEL en el MSC (Clave de Servicio, dirección gsmSCF)
- Enrutamiento SCCP (el Título Global debe enrutarse a CAMELGW)
- Reglas de firewall (permitir el puerto SCTP 2905)

#### Solución:

```
# Verificar conectividad M3UA
tcpdump -i eth0 sctp

# Comprobar si el MSC puede alcanzar CAMELGW
ss -tuln | grep 2905
```

### Problema: Errores de OCS

**Síntomas:** INSUFFICIENT\_CREDIT o errores de tiempo de espera

#### Verificar:

- OCS es accesible: `curl http://your-ocs-server/api/health`
- La cuenta tiene saldo en OCS
- Plan de calificación configurado en OCS
- Conectividad de red a OCS
- El token de autenticación es válido (si es necesario)

#### Solución:

- Verificar la configuración de la URL de OCS en runtime.exs
- Comprobar los registros de OCS en busca de errores
- Probar la API de OCS manualmente con curl
- Verificar que las reglas de firewall permitan conectividad

### Problema: Sesión no encontrada

**Síntomas:** EventReportBCSM falla con "Sesión no encontrada"

**Causa:** Desajuste de OTID o sesión expirada

#### Solución:

- Verificar OTID en los registros
- Comprobar el tiempo de espera de la sesión (por defecto, sin expiración)
- Asegurarse de que DTID coincida con OTID en los mensajes Continue/End

```
# Comprobar sesiones activas
iex> CAMELGW.SessionStore.list_sessions()
```

### Problema: Errores de decodificación

**Síntomas:** Failed to decode InitialDP en los registros

**Causa:** Desajuste de versión CAP o mensaje mal formado

#### Solución:

- Comprobar que la configuración de la versión CAP coincida con el MSC
- Verificar que la codificación ASN.1 sea correcta
- Capturar PCAP y analizar con Wireshark

```
# Capturar mensajes CAP
tcpdump -i eth0 -w cap_trace.pcap sctp port 2905

# Analizar con Wireshark (filtro: m3ua)
wireshark cap_trace.pcap
```

---

## Configuración Avanzada

### Múltiples Versiones CAP

Soporte para diferentes versiones CAP por clave de servicio:

```
config :omni7,
  cap_version_map: %{
    100 => :v2, # La Clave de Servicio 100 usa CAP v2
    200 => :v3, # La Clave de Servicio 200 usa CAP v3
    300 => :v4 # La Clave de Servicio 300 usa CAP v4
  },
  cap_version: :v2 # Por defecto
```

---

## Resumen

El modo Gateway CAMEL permite que OmniSS7 funcione como una plataforma completa de Red Inteligente con:

- ⊛ **Soporte completo del protocolo CAP** (v1/v2/v3/v4)
- ⊛ **Cobro en tiempo real** a través de la integración OCS
- ⊛ **Operaciones de control de llamadas** (Connect, Release, Continue)
- ⊛ **Gestión de sesiones** con almacenamiento ETS
- ⊛ **Frutas interactivas** a través del Constructor de Solicitudes de la Interfaz Web
- ⊛ **Monitoreo en vivo** de sesiones de llamadas activas
- ⊛ **Generación de CDR** para facturación y análisis
- ⊛ **Rendimiento y fiabilidad** listos para producción

Para información adicional:

- [Documentación del Constructor de Solicitudes CAMEL](#)
- [Referencia Técnica - Operaciones CAP](#)

---

**Producto:** Gateway CAMEL OmniSS7

**Versión de Documentación:** 1.0

**Última Actualización:** 2025-10-26

# CAMEL Request Builder - Resumen de Implementación

## Visión General

Se ha creado un nuevo componente LiveView para construir y enviar solicitudes CAMEL/CAP con fines de prueba. Esto proporciona una interfaz de usuario interactiva para crear InitialDP y otras operaciones CAMEL.

OmniSS7 Stack CAMEL GW v1.0.0 Download from Omnicalltech  
© 2025 Omnicalltech

**SS7 Events**  
**SS7 Client**  
**M3UA**  
**CAP Sessions**  
**CAP Requests**  
**Resources**  
**Configuration**

### CAMEL Request Builder

Build and send CAP/CAMEL requests for testing

ApplyCharging request sent successfully! OTID: DC6EBC13 (CAP v2)

#### Request Parameters

Request Type  
InitialDP (Initial Detection Point)

CAP/CAMEL Version (Current: V2)  
CAP v2 (OID: 0.4.0.0.1.0.50.1) - Default  
Determines the application context OID in TCAP dialogues for outgoing requests. This selection persists across form changes.

Call Duration (seconds)  
60  
Maximum call duration (1-984000 seconds, max: 10 days)

Release call when duration expires  
If unchecked, call continues but ApplyCharging/report is sent

[Show Advanced SCOP/M3UA Options](#)

[Send ApplyCharging Request](#)

---

**Current Session**  
OTID: DC6EBC13 Last Request Size: 38 bytes

---

**Request History (Last 20)**

Timestamp	Request Type	CAP Ver	OTID	Status	Size
23:56:25	ApplyCharging	v2	DC6EBC13...	sent	38 bytes
23:56:21	InitialDP	v2	DC6EBC13...	sent	74 bytes

---

**CAMEL Request Types**

- InitialDP - Initial Detection Point: Notifies gsmSCF about call setup
- Connect - Connect call to specified destination
- ReleaseCall - Release/terminate an ongoing call
- RequestReportBCSMEvent - Request notifications for call state events
- ApplyCharging - Set maximum call duration and charging parameters
- Continue - Continue call processing

## Nuevos Componentes

### 1. Constructor de Solicitudes CAMEL LiveView

#### Características:

- Interfaz de usuario basada en formularios interactivos para construir solicitudes CAMEL
- Soporte para múltiples tipos de solicitudes:
  - InitialDP - Punto de Detección Inicial (notificación de configuración de llamada)
  - Connect - Conectar llamada al destino
  - ReleaseCall - Liberar/terminar llamada
  - RequestReportBCSMEvent - Solicitar notificaciones de eventos
  - Continue - Continuar el procesamiento de la llamada
  - ApplyCharging - Aplicar límites de carga/duración a las llamadas

#### Capacidades Clave:

- Menú desplegable de selección de tipo de solicitud
- Campos de formulario dinámicos según el tipo de solicitud seleccionado
- Opciones avanzadas de SCOP/M3UA (sección colapsable)
  - Títulos Globales de Parte Llamada/Llamante
  - Configuración de SSN (Número de Subsistema)
  - Configuración de OPC/DPC (Código de Punto)
- Historial de solicitudes en tiempo real (últimas 20 solicitudes)
- Seguimiento de sesión a través de OTID
- Retroalimentación de éxito/errores
- Seguimiento del tamaño de la solicitud

Ruta: /camel\_request

### 2. EventLog Mejorado con Soporte CAMEL

#### Nuevas Funciones:

- paklog\_camel/2 - Registro de mensajes CAMEL/CAP dedicado
- lookup\_cap\_opcode\_name/1 - Búsqueda de código de operación CAP
- find\_cap\_opcode/1 - Extraer opcode CAP de JSON
- extract\_cap\_tids/1 - Extraer OTID/DTID de mensajes CAP
- format\_cap\_to\_json/1 - Convertir PDUUS CAP a formato JSON

#### Códigos de Operación CAP Soportados:

```
0 => "initialDP"  
5 => "connect"  
6 => "releaseCall"  
7 => "requestReportBCSMEvent"  
8 => "eventReportBCSM"  
10 => "continue"  
13 => "furnishChargingInformation"  
35 => "applyCharging"  
... (47 operaciones en total)
```

#### Características:

- Registro en JSON de todas las solicitudes/respuestas CAMEL
- Detección automática de acciones TCAP (Inicio/Continuar/Finalizar/Abortar)
- Extracción de direccionamiento SCCP
- Manejo de errores para mensajes malformados
- Procesamiento de tareas en segundo plano (no bloqueante)
- Evento prefijado con "CAP:" para fácil filtrado

### 3. CapClient Actualizado

#### Cambios:

- Se añadieron llamadas paklog\_camel/2 para mensajes entrantes y salientes
- Registro dual: Tanto MAP (paklog) como CAP (paklog\_camel) para compatibilidad
- Mensajes salientes registrados en sccp\_m3ua\_pakscr/2
- Mensajes entrantes registrados en handle\_payload/1

## Configuración

Las nuevas páginas LiveView se han añadido a la configuración de tiempo de ejecución:

```
# Archivo: config/runtime.exs
```

```
config :control_panel,
  use_additional_pages: [
    {557.Web.EventsLive, "/events", "Eventos 557"},
    {557.Web.TestClientLive, "/client", "Cliente 557"},
    {557.Web.M3UAStatusLive, "/m3ua", "M3UA"},
    {557.Web.HlrLinksLive, "/hlr_links", "Enlaces HLR"},
    {557.Web.CAMELSessionLive, "/camel_sessions", "Sesiones CAMEL"},
    {557.Web.CAMELRequestLive, "/camel_request", "Constructor de Solicitudes CAMEL"}
  ],
  page_order: ["/events", "/client", "/m3ua", "/hlr_links",
               "/camel_sessions", "/camel_request",
               "/application", "/configuration"]
```

## Uso

### Accediendo al Constructor de Solicitudes

1. Navega a: [https://your-server:8087/camel\\_request](https://your-server:8087/camel_request)
2. Selecciona el tipo de solicitud del menú desplegable
3. Completa los parámetros requeridos
4. Opcionalmente expande "Opciones Avanzadas de SCCP/M3UA" para ajustes finos
5. Haz clic en "Enviar Solicitud [RequestType]"

### Flujo de Solicitudes

#### InitialDP (Nueva Llamada)

1. Establecer Clave de Servicio (por ejemplo, 100)
2. Establecer Número Llamante (A-Party)
3. Establecer Número Llamado (B-Party)
4. Enviar solicitud → Genera nuevo OTID
5. OTID almacenado en la sesión para solicitudes de seguimiento

#### Solicitudes de Seguimiento (Connect, ReleaseCall, etc.)

1. Debe tener un OTID activo de InitialDP
2. La solicitud utiliza automáticamente el OTID almacenado
3. Se muestra una advertencia si no hay OTID activo

### Parámetros de Solicitud

#### InitialDP:

- Clave de Servicio (entero)
- Número Llamante (formato ISDN)
- Número Llamado (formato ISDN)

#### Connect:

- Número de Destino (donde enrutar la llamada)

#### ReleaseCall:

- Código de Causa (16 = Normal, 17 = Ocupado, 31 = No Especificado)

#### RequestReportBCSMEvent:

- Eventos BCSM (separados por comas: oAnswer, oDisconnect, etc.)

#### Continue:

- Sin parámetros (utiliza OTID activo)

#### ApplyCharging:

- Duración (segundos, 1-964000) - Duración máxima de la llamada antes de la acción
- Liberar en Tiempo de Espera (booleano) - Si liberar la llamada cuando expire la duración

### Opciones Avanzadas

#### Direccionamiento SCCP:

- GT de Parte Llamada (Título Global)
- GT de Parte Llamante
- SSN Llamado (predeterminado 146 = gsmSSF)
- SSN Llamante (predeterminado 146)

#### Códigos de Punto M3UA:

- OPC (Código de Punto de Origen, predeterminado 5013)
- DPC (Código de Punto de Destino, predeterminado 5011)

## Registro JSON

Todos los mensajes CAMEL ahora se registran en formato JSON en el registro de eventos con:

- **Dirección:** entrante/saliente
- **Acción TCAP:** Inicio/Continuar/Finalizar/Abortar
- **Operación CAP:** por ejemplo, "CAP:initialDP", "CAP:connect"
- **Direccionamiento SCCP:** Información de Parte Llamada/Llamante
- **TID:** OTID/TID para correlación
- **Mensaje Completo:** PDU CAP codificado en JSON

### Ejemplo de Entrada de Registro

```
{
  "map_event": "CAP:initialDP",
  "direction": "outgoing",
  "tcap_action": "Begin",
  "otid": "A1B2C3D4",
  "scco_called": {
    "SSN": 146,
    "GlobalTitle": {
      "Digits": "55512341234",
      "NumberingPlan": "isdn_tele",
      "NatureOfAddress_Indicator": "International"
    }
  },
  "event_message": "{ ... full CAP PDU ... }"
}
```

## Historial de Solicitudes

La interfaz de usuario muestra las últimas 20 solicitudes con:

- Marca de tiempo
- Tipo de solicitud (con insignia codificada por colores)
- OTID (primeros 8 caracteres hexadecimales)
- Estado (enviado/error)
- Tamaño del mensaje en bytes

## Seguimiento de Sesiones

### Panel de Información de Sesión Actual:

- Muestra OTID activo
- Muestra el tamaño en bytes de la última solicitud
- Visible solo cuando la sesión está activa

## Flujo de Trabajo de Pruebas

### 1. Iniciar Nueva Llamada:

- Enviar InitialDP → Obtener OTID
- El sistema crea una sesión

### 2. Controlar Llamada:

- Enviar RequestReportBCSMEvent → Solicitar notificaciones
- Enviar ApplyCharging → Establecer límite de duración de llamada (por ejemplo, 290 segundos)
- Enviar Connect → Enrutar al destino
- O Enviar ReleaseCall → Terminar

### 3. Ver Resultados:

- Revisar historial de solicitudes
- Monitorear la página de Sesiones CAMEL
- Revisar registros de eventos con prefijo "CAP:"

## ApplyCharging - Control de Duración de Llamada

### Visión General

La operación ApplyCharging permite establecer una duración máxima de llamada y opcionalmente, liberar la llamada cuando esa duración expire. Esto se utiliza típicamente para escenarios de carga prepaga o para hacer cumplir límites de tiempo en las llamadas.

### Casos de Uso

- **Carga Prepaga:** Limitar la duración de la llamada según el saldo del suscriptor
- **Facturación Basada en Tiempo:** Hacer cumplir intervalos de carga periódicos

- **Gestión de Recursos:** Prevenir que las llamadas se ejecuten indefinidamente
- **Integración OCS:** Coordinar con Sistemas de Carga en Línea para control de crédito en tiempo real

## Parámetros

### Duración (maxCallPeriodDuration)

- **Tipo:** Entero (1-864000 segundos)
- **Descripción:** Número máximo de segundos que la llamada puede ejecutarse antes de que el temporizador expire
- **Ejemplos:**
  - 60 = 1 minuto
  - 290 = 4 minutos 50 segundos (valor de prueba común)
  - 3600 = 1 hora
  - 86400 = 24 horas

### Liberar en Tiempo de Espera (releaseIfDurationExceeded)

- **Tipo:** Booleano (true/false)
- **Predeterminado:** true
- **Descripción:** Lo que sucede cuando expira la duración:
  - true: Liberar/desconectar automáticamente la llamada
  - false: Enviar notificación pero mantener la llamada activa (permite que gsmSCF tome acción)

## Estructura del Mensaje

El mensaje ApplyCharging se codifica como un TCAP Continue con:

- **TCAP:** Mensaje de Continuar (utiliza la transacción existente)
- **Opcode:** 35 (applyCharging)
- **Parámetros:** ApplyChargingArg que contiene:
  - sChargingCharacteristics: Información de carga basada en tiempo
    - LineDurationCharging: Duración máxima y bandera de liberación
  - partyToCharge: Qué parte es cargada (predeterminado: sendingSideID)

## Ejemplo de Uso

**Escenario:** Llamada prepaga con límite de 5 minutos

1. Enviar **InitialDP** para comenzar el monitoreo de la llamada

```
Clave de Servicio: 100
Llamante: 44770900123
Llamado: 44770900456
- OTID: A1B2C304
```

2. Enviar **ApplyCharging** para establecer el límite de 5 minutos

```
Duración: 300 (segundos)
Liberar en Tiempo de Espera: true
- Utiliza OTID: A1B2C304
```

3. Enviar **Connect** para completar la llamada

```
Destino: 44770900456
- Utiliza OTID: A1B2C304
```

4. Después de 5 minutos (300 segundos):

- La llamada se libera automáticamente por la red
- gsmSCF recibe notificación de desconexión

## Mejores Prácticas

1. **Siempre enviar ApplyCharging ANTES de Connect**

- Asegura que la carga esté activa cuando se conecta la llamada
- Previene segmentos de llamada no cargados

2. **Usar con RequestReportBCSMEEvent**

- Solicitar eventos oAnswer y oDisconnect
- Permite rastrear la duración real de la llamada
- Habilita la re-aplicación de carga si es necesario

3. **Establecer duraciones razonables**

- Demasiado corto: Operaciones de carga frecuentes, mala experiencia de usuario
- Demasiado largo: Riesgo de pérdida de ingresos en llamadas prepagas
- Típico: 60-300 segundos para prepago, más largo para postpago

4. **Manejar el tiempo de espera de manera adecuada**

- Si release=false, estar preparado para manejar notificaciones de expiración del temporizador
- Implementar lógica para extender la duración o liberar la llamada

## Manejo de Errores

Problemas comunes:

- **No hay OTID activo:** Debe enviar InitialDP primero
- **Duración inválida:** Debe ser de 1-864000 segundos
- **Soporte de red:** Algunas implementaciones de SSF pueden no soportar ApplyCharging
- **Precisión del temporizador:** La resolución del temporizador de la red es típicamente de 1 segundo, pero puede variar

## Monitoreo

Rastrear operaciones ApplyCharging a través de:

- **Historial de Solicitudes:** Muestra solicitudes ApplyCharging enviadas
- **Registro de Eventos:** Buscar "CAP:applyCharging"
- **Sesiones CAMEL:** Monitorear sesiones activas con carga aplicada
- **Rastreo TCAP:** Depurar problemas de codificación/decodificación

## Detalles de Implementación

### Gestión de Estado

- LiveView asigna el estado del formulario de seguimiento
- OTID almacenado en el socket asigna
- Historial de solicitudes limitado a 20 entradas
- Auto-refresco deshabilitado (solo envío manual)

### Generación de Solicitudes

- Utiliza el módulo existente CapRequestGenerator
- Construye estructuras TCAP/CAP adecuadas
- Codifica con el código: TCAPMessages
- Envuelve en SCCP a través de CapClient.sccp\_m3ua\_maker/2

### Mecanismo de Envío

- Envía a través de M3UA a :cameIgw\_client\_asp
- Utiliza el contexto de enrutamiento T
- Encapsulación automática SCCP/M3UA

### Manejo de Errores

- Validación de formularios con retroalimentación al usuario
- Manejo adecuado de OTID faltante
- Errores de análisis mostrados en la interfaz de usuario
- Fallos de codificación registrados

## Mejoras Futuras

Adiciones potenciales:

1. Plantillas/presets de solicitudes
2. Correlación y visualización de respuestas
3. Visualización del flujo de llamadas
4. Detalle de sesión en profundidad
5. Exportar historial de solicitudes
6. Pruebas de carga (solicitudes masivas)
7. Exportación PCAP de mensajes generados
8. Validación de parámetros CAP

## Notas de Integración

- Compatible con el registro MAP existente (paklog)
- Comparte la base de datos de registro de eventos con eventos MAP
- Utiliza la misma infraestructura SCCP/M3UA
- Funciona con CAMELsessionLive para monitoreo
- Se integra con el enrutamiento M3UA existente

## Archivos Modificados

- config/runtime.exs - ACTUALIZADO

## Dependencias

- Generador de Solicitudes CapRequestGenerator existente
- CapClient para envío M3UA
- M3UA.Server para transmisión de paquetes

- EventLog para registro de mensajes
- Marco Phoenix LiveView
- Panel de Control para infraestructura de UI

# Guía de Características Comunes

[— Volver a la Documentación Principal](#)

Esta guía cubre características comunes a todos los modos de operación de OmniSS7.

## Tabla de Contenidos

- 1. [Descripción General de la Interfaz Web](#)
- 2. [Documentación de la API](#)
- 3. [Monitoreo y Métricas](#)
- 4. [Muestras Prácticas](#)

## Descripción General de la Interfaz Web

La Interfaz Web es accesible a través de la dirección de su servidor web configurado.

Name	PID	Status	ASP State	Assoc/SCTP	Local	Remote	RC	24h Uptime	Actions
camelgw_client_asp	#PID=0.1322.0	Up	active	established	10.5.198.200:2985	10.179.4.10:2985	4	6.1%	

**24-Hour Availability Timeline**  
Uptime: 6.1% Total Up: 1m Total Down: 23h 58m

**Basic Information**

Name:	camelgw_client_asp	PID:	#PID=0.1322.0	Status:	Up	Mode:	ASP	ASP State:	active	Association State:	established
Routing Context:	4										

**Network Configuration**

Local IP:	10.5.198.200	Local Port:	2985	Remote IP:	10.179.4.10	Remote Port:	2985
-----------	--------------	-------------	------	------------	-------------	--------------	------

**Additional Details**  
▶ Raw Data (click to expand)

## Navegación Principal

- **Eventos** - Eventos de señalización SS7 en tiempo real y registros de mensajes
- **Aplicación** - Estado de la aplicación e información en tiempo de ejecución
- **Configuración** - Visor de configuración del sistema
- **Estado M3UA** - Conexiones de pares M3UA (modo STP)
- **Cola de SMS** - Mensajes SMS salientes (modo SMSc)

## Accediendo a la Interfaz Web

1. Abra su navegador web
2. Navegue a la dirección del hostname configurado (por ejemplo, <http://localhost>)
3. Vea el panel de estado del sistema

## Documentación de la API Swagger

Documentación interactiva de la API:

<http://your-server/swagger>

## Configuración de la Interfaz Web

Configure en `config/runtime.exs`:

```
config :control_panel,  
  # Orden de las páginas en el menú de navegación  
  page_order: ["/events", "/application", "/configuration"],  
  # Configuración del servidor web  
  web: %  
    listen_ip: "0.0.0.0", # IP a enlazar (0.0.0.0 para todas las interfaces)  
    port: 80, # Puerto HTTP (443 para HTTPS)  
    hostname: "localhost", # Nombre del servidor para la generación de URL  
    enable_tls: false, # Establecer verdadero para habilitar HTTPS  
    tls_cert: "cert.pem", # Ruta al archivo de certificado TLS  
    tls_key: "key.pem" # Ruta al archivo de clave privada TLS  
  }
```

## Parámetros de Configuración:

Parámetro	Tipo	Predeterminado	Descripción
page_order	Lista	["/events", "/application", "/configuration"]	Orden de las páginas en el menú de navegación
listen_ip	Cadena	"0.0.0.0"	Dirección IP para enlazar el servidor web
port	Entero	80	Puerto HTTP (usar 443 para HTTPS)
hostname	Cadena	"localhost"	Nombre del servidor para la generación de URL
enable_tls	Booleano	false	Habilitar HTTPS con TLS
tls_cert	Cadena	"cert.pem"	Ruta al certificado TLS (cuando TLS está habilitado)
tls_key	Cadena	"key.pem"	Ruta a la clave privada TLS (cuando TLS está habilitado)

## Configuración del Registrador

Configure el nivel de registro en `config/runtime.exs`:

```
config :logger,  
  level: :debug # Opciones: :debug, :info, :warning, :error
```

## Niveles de Registro:

- `:debug` - Información detallada de depuración
- `:info` - Mensajes informativos generales
- `:warning` - Mensajes de advertencia sobre problemas potenciales

- `:error` - Mensajes de error solamente

## Documentación de la API

### URL Base de la API

`http://your-server/api`

### Códigos de Respuesta

- **200** - Éxito
- **400** - Solicitud Incorrecta
- **504** - Tiempo de Espera del Gateway

### Especificación OpenAPI

`http://your-server/swagger.json`

## Monitoreo y Métricas

### Endpoint de Métricas de Prometheus

`http://your-server/metrics`

### Categorías Clave de Métricas

#### Métricas M3UA/SCTP:

- Cambios de estado de asociación SCTP
- Transiciones de estado ASP M3UA
- Unidades de datos de protocolo enviadas/recibidas

#### Métricas M2PA:

- Transiciones de estado del enlace (DOWN → ALIGNMENT → PROVING → READY)
- Mensajes y bytes enviados/recibidos por enlace
- Errores específicos del enlace (decodificar, codificar, SCTP)

#### Métricas STP:

- Mensajes recibidos/enviados por par
- Fallos de enrutamiento por razón
- Distribución del tráfico entre pares

#### Métricas del Cliente MAP:

- Solicitudes MAP por tipo de operación
- Histogramas de duración de solicitudes
- Medidor de transacciones pendientes

#### Métricas CAP:

- Solicitudes CAP por tipo de operación
- Operaciones del gateway CAMEL

#### Métricas SMS:

- Profundidad de la cola
- Tasas de entrega
- Mensajes fallidos

### Integración con Grafana

Las métricas de OmniSS7 son compatibles con Prometheus y Grafana.

## Mejores Prácticas

### Recomendaciones de Seguridad

- 1. Aislamiento de Red**
  - Desplegar en VLAN dedicada
  - Reglas de firewall para restringir el acceso
  - Permitir SCTP solo desde direcciones conocidas
- 2. Seguridad de la Interfaz Web**
  - Habilitar TLS para producción
  - Usar proxy inverso con autenticación
  - Restringir a IPs de gestión
- 3. Seguridad de la API**
  - Implementar limitación de tasa
  - Usar claves de API o OAuth
  - Registrar todas las solicitudes para auditoría

### Ajuste de Rendimiento

- 1. Límites de TPS**
  - Configurar TPS apropiado
  - Monitorear carga del sistema
  - Ajustar buffers SCTP
- 2. Optimización de Base de Datos**
  - Agregar índices
  - Archivar mensajes antiguos
  - Monitorear el grupo de conexiones
- 3. Ajuste de M3UA**
  - Ajustar intervalos de latido SCTP
  - Configurar valores de tiempo de espera
  - Usar múltiples enlaces para redundancia

### Monitoreo y Alertas

#### Métricas Clave:

- Estado de conexión M3UA
- Tasa de éxito de solicitudes MAP
- Tiempos de respuesta de la API
- Profundidad de la cola de mensajes

#### Umbral de Alerta:

- M3UA inactivo > 1 minuto
- Tasa de tiempo de espera de MAP > 10%
- Profundidad de la cola > 1000
- Tasa de error de la API > 5%

## Referencia Completa de Configuración

### Todos los Parámetros de Configuración

Esta sección proporciona una referencia completa de todos los parámetros de configuración disponibles en todos los modos de operación.

#### Configuración del Registrador (:logger)

```
config :logger,  
  level: :debug # :debug | :info | :warning | :error
```

#### Configuración de la Interfaz Web (:control\_panel)

```
config :control_panel,  
  page_order: ["/events", "/application", "/configuration"],  
  web: %{\br/>    listen_ip: "0.0.0.0",  
    port: 80,  
    hostname: "localhost",  
    enable_tls: false,  
    tls_cert: "cert.pem",  
    tls_key: "key.pem"  
  }
```

Parámetro	Tipo	Requerido	Predeterminado	Descripción
page_order	Lista de Cadenas	No	[ "/events", "/application", "/configuration" ]	Orden de las páginas en el menú de navegación
web.listen_ip	Cadena	Si	"0.0.0.0"	Dirección IP para enlazar el servidor web
web.port	Entero	Si	80	Número de puerto HTTP/HTTPS
web.hostname	Cadena	Si	"localhost"	Nombre del servidor
web.enable_tls	Booleano	No	false	Habilitar HTTPS
web.tls_cert	Cadena	Si TLS está habilitado	"cert.pem"	Ruta del certificado TLS
web.tls_key	Cadena	Si TLS está habilitado	"key.pem"	Ruta de la clave privada TLS

### Configuración STP M3UA (:omnis7)

```
config :omnis7,
  m3ua_stp: %{
    enabled: false,
    local_ip: {127, 0, 0, 1},
    local_port: 2905
  }
  enable_gt_routing: true,
  m3ua_peers: [...],
  m3ua_routes: [...],
  m3ua_gt_routes: [...]
```

Parámetro	Tipo	Requerido	Predeterminado	Descripción
m3ua_stp.enabled	Booleano	Si	false	Habilitar modo STP al iniciar
m3ua_stp.local_ip	Tupla	Si	{127, 0, 0, 1}	IP para enlazar para M3UA entrante
m3ua_stp.local_port	Entero	Si	2905	Puerto SCTP para M3UA
enable_gt_routing	Booleano	No	false	Habilitar enrutamiento de Título Global

### Parámetros del Par M3UA:

Parámetro	Tipo	Requerido	Descripción
peer_id	Entero	Si	Identificador único del par
name	Cadena	Si	Nombre descriptivo del par
role	Atomo	Si	:client o :server
local_ip	Tupla	Si :client	IP local para enlazar
local_port	Entero	Si :client	Puerto local (0 para dinámico)
remote_ip	Tupla	Si	IP del par remoto
remote_port	Entero	Si :client	Puerto del par remoto
routing_context	Entero	Si	Contexto de enrutamiento M3UA
point_code	Entero	Si	Código de punto SS7
network_indicator	Atomo	No	:international o :national

### Parámetros de Ruta M3UA:

Parámetro	Tipo	Requerido	Descripción
dest_pc	Entero	Si	Código de punto de destino
peer_id	Entero	Si	Par a través del cual enrutarse
priority	Entero	Si	Prioridad de la ruta (menor = mayor prioridad)
network_indicator	Atomo	No	:international o :national

### Parámetros de Ruta GT M3UA:

Parámetro	Tipo	Requerido	Descripción
gt_prefix	Cadena	Si	Prefijo de Título Global a coincidir
peer_id	Entero	Si	Par de destino
priority	Entero	Si	Prioridad de la ruta
description	Cadena	No	Descripción de la ruta para registro
source_ssn	Entero	No	Coincidir solo si el SSN de origen coincide
dest_ssn	Entero	No	Reescribir el SSN de destino a este valor

### Configuración del Cliente MAP (:omnis7)

```
config :omnis7,
  map_client_enabled: false,
  map_client_m3ua: %{
    mode: "ASP",
    callback: {MapClient, :handle_payload, []},
    process_name: :map_client_asp,
    local_ip: {10, 0, 0, 100},
    local_port: 2905,
    remote_ip: {10, 0, 0, 1},
    remote_port: 2905,
    routing_context: 1
  }
```

Parámetro	Tipo	Requerido	Predeterminado	Descripción
map_client_enabled	Booleano	Si	false	Habilitar modo cliente MAP
map_client_m3ua.mode	Cadena	Si	"ASP"	Modo de conexión M3UA ("ASP" o "SGP")
map_client_m3ua.callback	Tupla	Si	{MapClient, :handle_payload, []}	Manejador de callback de mensajes
map_client_m3ua.process_name	Atomo	Si	:map_client_asp	Nombre del proceso registrado
map_client_m3ua.local_ip	Tupla	Si	-	Dirección IP local
map_client_m3ua.local_port	Entero	Si	2905	Puerto SCTP local
map_client_m3ua.remote_ip	Tupla	Si	-	IP remota de STP/SGP
map_client_m3ua.remote_port	Entero	Si	2905	Puerto SCTP remoto
map_client_m3ua.routing_context	Entero	Si	-	Contexto de enrutamiento M3UA

### Configuración del Centro de SMS (:omnis7)

```
config :omnis7,
  auto_flush_enabled: false,
  auto_flush_interval: 10_000,
  auto_flush_dest_smsc: nil,
  auto_flush_tps: 10
```

Parámetro	Tipo	Requerido	Predeterminado	Descripción
auto_flush_enabled	Booleano	No	false	Habilitar el vaciado automático de la cola de SMS
auto_flush_interval	Entero	No	10000	Intervalo de sondeo de la cola (milisegundos)
auto_flush_dest_smsc	Cadena	Nil	nil	Filtrar por SMSC de destino (nil = todos)
auto_flush_tps	Entero	No	10	Máx. transacciones por segundo

### Configuración de la API HTTP (:omnis7)

El backend de SMS ahora utiliza la API HTTP en lugar de conexiones directas a la base de datos.

```
config :omnis7,
  smsc_api_base_url: "https://10.5.198.200:8443",
  frontend_name: "omni-smsc01" # Opcional: predeterminado a hostname_SMSC
```

### Parámetros de la API:

Parámetro	Tipo	Requerido	Predeterminado	Descripción
smc_api_base_url	Cadena	Si	"https://10.5.198.200:8443"	URL base para la API del backend de SMS
frontend_name	Cadena	No	"{hostname}_SMSC"	Identificador del frontend para registro

### Endpoints de la API Utilizados:

- POST /api/frontends - Registrar esta instancia de frontend con el backend
- POST /api/messages\_raw - Insertar nuevos mensajes SMS
- GET /api/messages - Recuperar cola de mensajes (con encabezado smsc)
- PATCH /api/messages/{id} - Marcar mensaje como entregado
- PUT /api/messages/{id} - Actualizar estado del mensaje
- POST /api/events - Agregar seguimiento de eventos
- GET /api/status - Endpoint de verificación de salud

### Registro del Frontend:

El sistema se registra automáticamente con la API del backend al iniciar y se vuelve a registrar cada 5 minutos. El registro incluye:

- Nombre y tipo de frontend (SMSC)
- Nombre del host
- Tiempo de actividad en segundos
- Detalles de configuración (formato JSON)

### Notas de Configuración:

- La verificación SSL está deshabilitada por defecto para certificados autofirmados
- Las solicitudes HTTP tienen un tiempo de espera de 5 segundos
- Todas las marcas de tiempo están en formato ISO 8601
- La API utiliza JSON para los cuerpos de solicitud/respuesta

## Documentación Relacionada

- [= Volver a la Documentación Principal](#)
- [Guía STP](#)
- [Guía del Cliente MAP](#)
- [Guía del Centro de SMS](#)
- [Guía HTTP](#)

OmniSS7 por Omnitouch Network Services

# Referencia de Configuración

[← Volver a la Documentación Principal](#)

Este documento proporciona una referencia completa para todos los parámetros de configuración de OmniSS7.

## Tabla de Contenidos

- [1. Descripción General](#)
- [2. Banderas de Modo Operativo](#)
- [3. Parámetros de Modo HLR](#)
- [4. Parámetros de Modo SMSc](#)
- [5. Parámetros de Modo STP](#)
- [6. Parámetros de NAT de Título Global](#)
- [7. Parámetros de Conexión M3UA](#)
- [8. Parámetros del Servidor HTTP](#)
- [9. Parámetros de Base de Datos](#)
- [10. Valores Codificados](#)

---

## Descripción General

La configuración de OmniSS7 se gestiona a través de `config/runtime.exs`. El sistema admite tres modos operativos:

- **Modo STP** - Punto de Transferencia de Señales para enrutamiento
- **Modo HLR** - Registro de Ubicación del Hogar para gestión de suscriptores
- **Modo SMSc** - Centro de SMS para entrega de mensajes

**Archivo de Configuración:** `config/runtime.exs`

---

## Banderas de Modo Operativo

Controla qué características están habilitadas.

Parámetro	Tipo	Predeterminado	Descripción	Modos
<code>map_client_enabled</code>	Booleano	false	Habilitar cliente MAP y conectividad M3UA	Todos
<code>hlr_mode_enabled</code>	Booleano	false	Habilitar características específicas de HLR	HLR
<code>smc_mode_enabled</code>	Booleano	false	Habilitar características específicas de SMSc	SMSc

**Ejemplo:**

```
config :omniss7,  
  map_client_enabled: true,  
  hlr_mode_enabled: true,  
  smc_mode_enabled: false
```

---

## Parámetros de Modo HLR

Configuración para el modo HLR (Registro de Ubicación del Hogar).

## Configuración de la API HLR

Parámetro	Tipo	Predeterminado	Requerido	Descripción
hlr_api_base_url	Cadena	-	Sí	URL del punto final de la API HLR del backend (verificación SSL codificada como deshabilitada)
hlr_service_center_gt_address	Cadena	-	Sí	Dirección de Título Global HLR devuelta en las respuestas de UpdateLocation
smc_service_center_gt_address	Cadena	-	Sí	Dirección GT del SMC devuelta en las respuestas SRI-for-SM

### Ejemplo:

```
config :omniss7,  
  hlr_api_base_url: "https://10.180.2.140:8443",  
  hlr_service_center_gt_address: "55512341111",  
  smc_service_center_gt_address: "55512341112"
```

## Mapeo MSISDN ↔ IMSI

Configuración para la generación sintética de IMSI a partir de MSISDNs. Para una explicación técnica detallada del algoritmo de mapeo, consulte [Mapeo MSISDN ↔ IMSI en la Guía HLR](#).

Parámetro	Tipo	Predeterminado	Requerido	Descripción
hlr_imsi_plmn_prefix	Cadena	"50557"	No	Prefijo PLMN (MCC+MNC) para la generación sintética de IMSI
hlr_msisdn_country_code	Cadena	"61"	No	Prefijo del código de país para el mapeo inverso IMSI→MSISDN
hlr_msisdn_nsn_offset	Entero	0	No	Desplazamiento en MSISDN donde comienza el NSN (típicamente la longitud del código de país)
hlr_msisdn_nsn_length	Entero	9	No	Longitud del Número Nacional de Suscriptor a extraer de MSISDN

### Ejemplo (código de país de 2 dígitos):

```
config :omniss7,  
  hlr_imsi_plmn_prefix: "50557", # MCC 505 + MNC 57  
  hlr_msisdn_country_code: "99", # Ejemplo de código de país de 2 dígitos  
  hlr_msisdn_nsn_offset: 2, # Omitir el código de país de 2 dígitos  
  hlr_msisdn_nsn_length: 9 # Extraer NSN de 9 dígitos
```

### Ejemplo (código de país de 3 dígitos):

```
config :omniss7,  
  hlr_imsi_plmn_prefix: "50557", # MCC 505 + MNC 57  
  hlr_msisdn_country_code: "999", # Ejemplo de código de país de 3 dígitos  
  hlr_msisdn_nsn_offset: 3, # Omitir el código de país de 3 dígitos  
  hlr_msisdn_nsn_length: 8 # Extraer NSN de 8 dígitos
```

**Importante:** Establezca `nsn_offset` a la longitud de su código de país para extraer correctamente el NSN. Por ejemplo:

- Código de país "9" (1 dígito) → `nsn_offset: 1`
- Código de país "99" (2 dígitos) → `nsn_offset: 2`
- Código de país "999" (3 dígitos) → `nsn_offset: 3`

## Configuración de InsertSubscriberData (ISD)

Configuración para los datos de aprovisionamiento de suscriptores enviados a los VLR durante

UpdateLocation. Para una explicación detallada de la secuencia ISD y el flujo de mensajes, consulte [Configuración de InsertSubscriberData en la Guía HLR](#).

Parámetro	Tipo	Predeterminado	Requerido	Descripción
isd_network_access_mode	Átomo	:packetAndCircuit	No	Tipo de acceso a la red: :packetAndCircuit, :packetOnly, o :circuitOnly
isd_send_ss_data	Booleano	true	No	Enviar ISD #2 con datos de Servicios Suplementarios
isd_send_call_barring	Booleano	true	No	Enviar ISD #3 con datos de Bloqueo de Llamadas

**Ejemplo:**

```
config :omniss7,
  isd_network_access_mode: :packetAndCircuit,
  isd_send_ss_data: true,
  isd_send_call_barring: true
```

**Configuración CAMEL**

Configuración para el enrutamiento de llamadas inteligentes basado en CAMEL. Para una explicación detallada de la integración CAMEL y las claves de servicio, consulte [Integración CAMEL en la Guía HLR](#).

Parámetro	Tipo	Predeterminado	Requerido	Descripción
camel_service_key	Entero	11_110	No	Clave de servicio CAMEL para respuestas SRI
camel_trigger_detection_point	Átomo	:termAttemptAuthorized	No	Punto de detección de CAMEL: :termAttemptAuthorized, :tBusy, :tNoAnswer, :tAnswer
camel_gsmcf_gt_address	Cadena (usa GT llamado)		No	Título Global gsmSCF predeterminado para respuestas CAMEL (puede ser anulado por GT NAT)

**Ejemplo:**

```
config :omniss7,
  camel_service_key: 11_110,
  camel_trigger_detection_point: :termAttemptAuthorized
```

**Prefijos de VLR de Hogar**

Configuración para distinguir entre suscriptores de hogar y de roaming. Para una explicación detallada de la detección de hogar/roaming y las operaciones PRN, consulte [Manejo de Suscriptores de Roaming en la Guía HLR](#).

Parámetro	Tipo	Predeterminado	Requerido	Descripción
home_vlr_prefixes	Lista	["5551231"]	No	Prefijos GT de VLR considerados como red "hogar"

**Ejemplo:**

```
config :omniss7,
  home_vlr_prefixes: ["5551231", "5551234"]
```

**Parámetros de Modo SMSc**

Configuración para el modo Centro de SMS.

## Configuración de la API SMSc

Parámetro	Tipo	Predeterminado	Requerido	Descripción
smsc_api_base_url	Cadena	-	<b>Sí</b>	URL del punto final de la API SMSc del backend (verificación SSL codificada como deshabilitada)
smsc_name	Cadena	"{hostname}_SMSc"	No	Identificador SMSc para el registro en el backend
smsc_service_center_gt_address	Cadena	-	<b>Sí</b>	Dirección de Título Global del Centro de Servicio

### Ejemplo:

```
config :omniss7,  
  smsc_api_base_url: "https://10.179.3.219:8443",  
  smsc_name: "ipsmgw",  
  smsc_service_center_gt_address: "55512341112"
```

**Nota:** El registro en el frontend ocurre cada **5 minutos** (codificado) a través del módulo SMS.FrontendRegistry.

## Configuración de Auto-Limpieza

Parámetro	Tipo	Predeterminado	Requerido	Descripción
auto_flush_enabled	Booleano	true	No	Habilitar el procesamiento automático de la cola de SMS
auto_flush_interval	Entero	10_000	No	Intervalo de procesamiento de la cola en milisegundos
auto_flush_dest_smsc	Cadena	-	<b>Sí</b>	Nombre del SMSc de destino para auto-limpieza
auto_flush_tps	Entero	10	No	Tasa de procesamiento de mensajes (transacciones/segundo)

### Ejemplo:

```
config :omniss7,  
  auto_flush_enabled: true,  
  auto_flush_interval: 10_000,  
  auto_flush_dest_smsc: "ipsmgw",  
  auto_flush_tps: 10
```

---

## Parámetros de Modo STP

Configuración para el modo Punto de Transferencia de Señales M3UA. Para una configuración de enrutamiento detallada y ejemplos, consulte la [Guía de Configuración STP](#).

### Servidor STP Autónomo

Parámetro	Tipo	Predeterminado	Requerido	Descripción
m3ua_stp.enabled	Booleano	false	No	Habilitar servidor STP M3UA autónomo
m3ua_stp.local_ip	Tupla	{127, 0, 0, 1}	No	Dirección IP para escuchar conexiones
m3ua_stp.local_port	Entero	2905	No	Puerto para escuchar
m3ua_stp.point_code	Entero	-	<b>Sí</b> (si está habilitado)	Código de punto SS7 de este STP

### Ejemplo:

```
config :omniss7,
  m3ua_stp: %{
    enabled: true,
    local_ip: {10, 179, 4, 10},
    local_port: 2905,
    point_code: 100
  }
}
```

## Enrutamiento de Título Global

Parámetro	Tipo	Predeterminado	Requerido	Descripción
enable_gt_routing	Booleano	false	No	Habilitar enrutamiento GT además del enrutamiento PC

### Ejemplo:

```
config :omniss7,
  enable_gt_routing: true
```

## Parámetros de NAT de Título Global

La Traducción de Dirección de Red de Título Global permite diferentes GTs de respuesta según el prefijo de la parte que llama. Para una explicación detallada y ejemplos, consulte la [Guía de NAT de Título Global](#).

Parámetro	Tipo	Predeterminado	Requerido	Descripción
gt_nat_enabled	Booleano	false	No	Habilitar/deshabilitar la función de NAT GT
gt_nat_rules	Lista de Mapas	[]	<b>Sí</b> (si está habilitado)	Lista de mapeos de prefijo a GT

**Formato de Regla:** Cada regla en `gt_nat_rules` debe ser un mapa con:

- `calling_prefix`: Prefijo de cadena para coincidir con el GT que llama
- `response_gt`: Título Global a utilizar en las respuestas

### Ejemplo:

```
config :omniss7,
  gt_nat_enabled: true,
  gt_nat_rules: [
    # Cuando se llama desde GT que comienza con "8772", responder con "55512341112"
    %{calling_prefix: "8772", response_gt: "55512341112"},
    # Cuando se llama desde GT que comienza con "8773", responder con "55512341111"
    %{calling_prefix: "8773", response_gt: "55512341111"},
    # Fallback predeterminado (prefijo vacío coincide con todos)
    %{calling_prefix: "", response_gt: "55512311555"}
  ]
```

**Véase También:** [Guía de NAT GT](#) para un uso y ejemplos detallados.

## Parámetros de Conexión M3UA

Configuración de conexión M3UA para el modo cliente MAP. Para un uso y ejemplos detallados, consulte la [Guía del Cliente MAP](#).

Parámetro	Tipo	Predeterminado	Requerido	Descripción
map_client_m3ua.mode	Cadena-	-	<b>Sí</b>	Modo de conexión: "ASP" o "SGP"
map_client_m3ua.callback	Tupla	-	<b>Sí</b>	Módulo/función de callback:

Parámetro	Tipo	Predeterminado	Requerido	Descripción
map_client_m3ua.process_name	Átomo	-	<b>Sí</b>	{MapClient, :handle_payload, []} Nombre del proceso para el registro
map_client_m3ua.local_ip	Tupla	-	<b>Sí</b>	Dirección IP local para enlazar
map_client_m3ua.local_port	Entero	2905	<b>Sí</b>	Puerto SCTP local
map_client_m3ua.remote_ip	Tupla	-	<b>Sí</b>	Dirección IP remota STP/SGW
map_client_m3ua.remote_port	Entero	2905	<b>Sí</b>	Puerto SCTP remoto
map_client_m3ua.routing_context	Entero	-	<b>Sí</b>	ID de contexto de enrutamiento M3UA

### Ejemplo:

```
config :omniss7,
  map_client_m3ua: %{
    mode: "ASP",
    callback: {MapClient, :handle_payload, []},
    process_name: :hlr_client_asp,
    local_ip: {10, 179, 4, 11},
    local_port: 2905,
    remote_ip: {10, 179, 4, 10},
    remote_port: 2905,
    routing_context: 1
  }
}
```

---

## Parámetros del Servidor HTTP

Configuración para el servidor HTTP de la API REST.

Parámetro	Tipo	Predeterminado	Requerido	Descripción
start_http_server	Booleano	true	No	Habilitar/deshabilitar el servidor HTTP (puerto 8080)

**Valores Codificados** (no configurables):

- **IP:** 0.0.0.0 (todas las interfaces)
- **Puerto:** 8080
- **Transporte:** Plug.Cowboy

### Ejemplo:

```
config :omniss7,
  start_http_server: true # Establecer en false para deshabilitar
```

**Puntos finales de la API:**

- API REST: `http://[server-ip]:8080/api/*`
- Interfaz Swagger: `http://[server-ip]:8080/swagger`
- Métricas de Prometheus: `http://[server-ip]:8080/metrics`

Consulte la [Guía de API](#) para más detalles.

---

## Parámetros de Base de Datos

Configuración para la persistencia de la base de datos Mnesia.

Parámetro	Tipo	Predeterminado	Requerido	Descripción
mnesia_storage_type	Átomo	:disc_copies	No	Tipo de almacenamiento de Mnesia: :disc_copies o :ram_copies

### Ejemplo:

```
config :omniss7,
  mnesia_storage_type: :disc_copies # Producción
  # mnesia_storage_type: :ram_copies # Solo para pruebas
```

### Tipos de Almacenamiento:

- :disc\_copies - Almacenamiento en disco persistente (sobrevive a reinicios) - **Recomendado para producción**
- :ram\_copies - Solo en memoria (se pierde en reinicio) - Solo para pruebas

### Tablas de Mnesia:

- m3ua\_peer - Conexiones de pares M3UA
- m3ua\_route - Rutas de Código de Punto
- m3ua\_gt\_route - Rutas de Título Global

**Ubicación:** directorio Mnesia.{node\_name}/

---

## Valores Codificados

Los siguientes valores están **codificados en el código fuente** y no se pueden cambiar a través de la configuración.

### Tiempos de Espera

Valor	Impacto	Solución Alternativa
Tiempo de espera de solicitud MAP: <b>10 segundos</b>	Todas las operaciones MAP caducan después de 10s	Modificar el código fuente
Tiempo de espera ISD: <b>10 segundos</b>	Cada mensaje ISD caduca después de 10s	Modificar el código fuente

### Servidor HTTP

Valor	Impacto	Solución Alternativa
IP HTTP: <b>0.0.0.0</b>	El servidor escucha en todas las interfaces	Modificar el código fuente
Puerto HTTP: <b>8080</b>	La API REST se ejecuta en el puerto 8080	Modificar el código fuente

### Verificación SSL

Valor	Impacto	Solución Alternativa
API HLR SSL: <b>deshabilitado</b>	La verificación SSL siempre está deshabilitada	Modificar el código fuente
API SMS Sc SSL: <b>deshabilitado</b>	La verificación SSL siempre está deshabilitada	Modificar el código fuente

### Intervalos de Registro

Valor	Impacto	Solución Alternativa
Registro en el frontend: <b>5 minutos</b>	SMS Sc se registra con el backend cada 5 min	Modificar el código fuente

## Auto-Actualización de la Interfaz Web

Página	Intervalo
Gestión de Enrutamiento	5 segundos
Suscriptores Activos	2 segundos

---

## Ejemplos de Configuración

### Configuración Mínima de HLR

```
config :omniss7,
  map_client_enabled: true,
  hlr_mode_enabled: true,
  smsc_mode_enabled: false,

  hlr_api_base_url: "https://10.180.2.140:8443",
  hlr_service_center_gt_address: "55512341111",
  smsc_service_center_gt_address: "55512341112",

  map_client_m3ua: %{
    mode: "ASP",
    callback: {MapClient, :handle_payload, []},
    process_name: :hlr_client_asp,
    local_ip: {10, 179, 4, 11},
    local_port: 2905,
    remote_ip: {10, 179, 4, 10},
    remote_port: 2905,
    routing_context: 1
  }
```

### Configuración Mínima de SMSc

```
config :omniss7,
  map_client_enabled: true,
  hlr_mode_enabled: false,
  smsc_mode_enabled: true,

  smsc_api_base_url: "https://10.179.3.219:8443",
  smsc_name: "ipsmgw",
  smsc_service_center_gt_address: "55512341112",

  auto_flush_enabled: true,
  auto_flush_interval: 10_000,
  auto_flush_dest_smsc: "ipsmgw",
  auto_flush_tps: 10,

  map_client_m3ua: %{
    mode: "ASP",
    callback: {MapClient, :handle_payload, []},
    process_name: :stp_client_asp,
    local_ip: {10, 179, 4, 12},
    local_port: 2905,
    remote_ip: {10, 179, 4, 10},
    remote_port: 2905,
    routing_context: 1
  }
```

### STP con Servidor Autónomo

```
config :omniss7,
  map_client_enabled: true,
```

```
hlr_mode_enabled: false,
smc_mode_enabled: false,

enable_gt_routing: true,
mnesia_storage_type: :disc_copies,

m3ua_stp: %{
  enabled: true,
  local_ip: {10, 179, 4, 10},
  local_port: 2905,
  point_code: 100
},

map_client_m3ua: %{
  mode: "ASP",
  callback: {MapClient, :handle_payload, []},
  process_name: :stp_client_asp,
  local_ip: {10, 179, 4, 10},
  local_port: 2906,
  remote_ip: {10, 179, 4, 11},
  remote_port: 2905,
  routing_context: 1
}
```

---

## Resumen

### Total de Parámetros de Configuración: 32

#### Por Categoría:

- Modo Operativo: 3 parámetros
- Modo HLR: 13 parámetros
- Modo SMSc: 7 parámetros
- Modo STP: 5 parámetros
- Conexión M3UA: 8 parámetros
- Servidor HTTP: 1 parámetro
- Base de Datos: 1 parámetro

#### Parámetros Requeridos (deben ser establecidos):

- `hlr_api_base_url` (modo HLR)
- `hlr_service_center_gt_address` (modo HLR)
- `smc_api_base_url` (modo SMSc)
- `smc_service_center_gt_address` (modo SMSc/HLR)
- Todos los parámetros `map_client_m3ua.*`
- `m3ua_stp.point_code` (si STP está habilitado)

---

## Documentación Relacionada

- [Guía HLR](#) - Configuración específica de HLR
- [Guía SMSc](#) - Configuración específica de SMSc
- [Guía STP](#) - Configuración de enrutamiento STP
- [Guía API](#) - Referencia de API REST
- [Guía de Interfaz Web](#) - Documentación de la interfaz web

# Guía de NAT de Título Global

## Descripción general

La Traducción de Dirección de Título Global (GT NAT) es una función que permite a OmniSS7 responder con diferentes direcciones de Título Global basadas en el prefijo GT de la parte que llama, el prefijo GT de la parte llamada, o una combinación de ambos. Esto es esencial cuando se opera con múltiples Títulos Globales y se necesita asegurar que las respuestas utilicen el GT correcto basado en qué red o par está llamando y/o cuál GT llamaron.

## Novedades (GT NAT Mejorado)

La función GT NAT ha sido mejorada con potentes nuevas capacidades:

### Nuevas características

1. **Coincidencia de Prefijo de Parte Llamada:** Las reglas ahora pueden coincidir en `called_prefix` además de `calling_prefix`
2. **Coincidencia Combinada:** Las reglas pueden coincidir en ambos prefijos de llamada Y de llamada simultáneamente
3. **Priorización Basada en Peso:** Las reglas ahora utilizan un campo `weight` (menor = mayor prioridad) en lugar de solo la longitud del prefijo
4. **Coincidencia Flexible:** Ahora puedes crear reglas con:
  - Solo prefijo de llamada
  - Solo prefijo de parte llamada
  - Ambos prefijos de llamada y parte llamada
  - Ninguno (regla comodín/de respaldo)

### Nuevo Formato de Regla

#### Campos requeridos:

- `weight`: Prioridad entera (menor = mayor prioridad)
- `response_gt`: El GT con el que responder

#### Campos opcionales (se recomienda al menos uno para coincidencias específicas):

- `calling_prefix`: Coincidir en el prefijo GT de la parte que llama
- `called_prefix`: Coincidir en el prefijo GT de la parte llamada

#### Ejemplo:

```
gt_nat_rules: [  
  # Regla específica con ambos prefijos - mayor prioridad  
  %{calling_prefix: "8772", called_prefix: "555", weight: 1,  
  response_gt: "111111"},  
  
  # Reglas específicas - prioridad media  
  %{calling_prefix: "8772", weight: 10, response_gt: "222222"},  
  %{called_prefix: "555", weight: 10, response_gt: "333333"},  
  
  # Comodín de respaldo - menor prioridad  
  %{weight: 100, response_gt: "999999"}  
]
```

## Casos de uso

### Operación Multi-Rede

Cuando tienes múltiples redes pares y cada una espera respuestas de un GT específico:

- **Red A** llama a tu GT 111111 y espera respuestas de 111111
- **Red B** llama a tu GT 222222 y espera respuestas de 222222

Sin GT NAT, necesitarías instancias separadas o enrutamiento complejo. Con GT NAT, una sola instancia de OmniSS7 puede manejar esto de manera inteligente.

### Escenarios de Roaming

Cuando operas como un HLR o SMSr con acuerdos de roaming:

- Los suscriptores de la **red de origen** utilizan GT 555000
- **Socio de roaming 1** utiliza GT 555001
- **Socio de roaming 2** utiliza GT 555002

GT NAT asegura que cada socio reciba respuestas del GT correcto al que están configurados para enrutar.

### Pruebas y Migración

Durante migraciones de red o pruebas:

- Migrar gradualmente el tráfico del antiguo GT al nuevo GT
- Mantener ambos GT durante el período de transición
- Enrutar respuestas basadas en qué GT utilizó el llamador

# Cómo funciona

## Flujo de Traducción de Direcciones

1. **Solicitud Entrante:** OmniSS7 recibe un mensaje SCCP con:
  - GT de Parte Llamada: 55512341112 (tu GT)
  - GT de Parte Llamante: 877234567 (su GT)
2. **Búsqueda de GT NAT:** El sistema verifica el GT de llamada 877234567 contra las reglas de prefijo configuradas
3. **Coincidencia de Prefijo:** Encuentra el prefijo coincidente más largo (por ejemplo, 8772 coincide con 877234567)
4. **Selección de GT de Respuesta:** Utiliza `response_gt` de la regla coincidente (por ejemplo, 55512341112)
5. **Respuesta Enviada:** La respuesta SCCP utiliza:
  - GT de Parte Llamada: 877234567 (invertido - su GT)
  - GT de Parte Llamante: 55512341112 (GT NAT'd)

## Tipos de Respuesta Afectados

GT NAT se aplica a múltiples capas de la pila SS7:

### Capa SCCP (Todas las Respuestas)

- Direcciones GT de Parte Llamada/Llamante en todos los mensajes de respuesta
- Acknowledgments de ISD (InsertSubscriberData)
- Respuestas de UpdateLocation
- Respuestas de error

### Capa MAP (Específica de Operación)

- **Respuestas SRI-for-SM:** `networkNode-Number` (dirección GT de SMS)C)
- **UpdateLocation:** `hlr-Number` en respuestas
- **InsertSubscriberData:** GT de HLR en mensajes ISD

## Configuración

### Configuración Básica

Agrega a `config/runtime.exs`:

```

config :omniss7,
  # Habilitar GT NAT
  gt_nat_enabled: true,

  # Definir reglas de GT NAT
  gt_nat_rules: [
    # Regla 1: Llamadas desde el prefijo "8772" obtienen respuesta de
    "55512341112"
    %{calling_prefix: "8772", response_gt: "55512341112"},

    # Regla 2: Llamadas desde el prefijo "8773" obtienen respuesta de
    "55512341111"
    %{calling_prefix: "8773", response_gt: "55512341111"},

    # Regla predeterminada (prefijo vacío coincide con todo)
    %{calling_prefix: "", response_gt: "55512311555"}
  ]

```

## Parámetros de Configuración

Para referencia completa de configuración, consulta [Parámetros de NAT de Título Global en la Referencia de Configuración](#).

Parámetro	Tipo	Requerido	Descripción
gt_nat_enabled	Booleano	Sí	Habilitar/deshabilitar la función GT NAT
gt_nat_rules	Lista de Mapas	Sí (si está habilitado)	Lista de reglas de coincidencia de prefijos

## Formato de Regla

Cada regla es un mapa con las siguientes claves:

```

%{
  calling_prefix: "8772",      # (Opcional) Prefijo para coincidir
con el GT de llamada
  called_prefix: "555",      # (Opcional) Prefijo para coincidir
con el GT de llamada
  weight: 10,                # (Requerido) Valor de prioridad
(menor = mayor prioridad)
  response_gt: "55512341112" # (Requerido) GT a utilizar en las
respuestas
}

```

## Campos de Regla:

- **calling\_prefix** (Opcional): Prefijo de cadena para coincidir con el GT de llamada entrante

- La coincidencia se realiza mediante `String.starts_with?/2`
- La cadena vacía "" o nil actúa como comodín (coincide con cualquier GT de llamada)
- Se puede omitir para coincidir con cualquier GT de llamada
- **called\_prefix** (Opcional): Prefijo de cadena para coincidir con el GT de llamada entrante
  - La coincidencia se realiza mediante `String.starts_with?/2`
  - La cadena vacía "" o nil actúa como comodín (coincide con cualquier GT de llamada)
  - Se puede omitir para coincidir con cualquier GT de llamada
- **weight** (Requerido): Valor de prioridad entero
  - Peso menor = mayor prioridad (procesado primero)
  - Debe ser  $\geq 0$
  - Se utiliza como criterio de ordenación principal para las reglas de coincidencia
- **response\_gt** (Requerido): La dirección de Título Global a utilizar en las respuestas
  - Debe ser una cadena de número E.164 válida
  - Debe coincidir con uno de tus GT configurados

**Al menos uno de `calling_prefix` o `called_prefix` debe ser especificado para el enrutamiento específico. Ambos se pueden omitir para una regla comodín/de respaldo.**

## Lógica de Coincidencia de Reglas

Las reglas se evalúan por **peso primero (ascendente)**, luego por **especificidad combinada de prefijo**:

### Algoritmo de Coincidencia:

1. Filtrar reglas donde todos los prefijos especificados coinciden
  - Si `calling_prefix` está configurado, debe coincidir con el GT de llamada
  - Si `called_prefix` está configurado, debe coincidir con el GT de llamada
  - Si ambos están configurados, ambos deben coincidir
  - Si ninguno está configurado, la regla actúa como un comodín
2. Ordenar las reglas coincidentes por:
  - **Primario**: Peso (ascendente - valores más bajos primero)
  - **Secundario**: Longitud de prefijo combinada (descendente - más largo = más específico)
3. Retornar la primera regla coincidente

## Ejemplos:

```
# Ejemplo de reglas
gt_nat_rules: [
  # Peso 1: Mayor prioridad - coincide con ambos prefijos
  %{calling_prefix: "8772", called_prefix: "555", weight: 1,
  response_gt: "111111"},

  # Peso 10: Prioridad media - reglas específicas
  %{calling_prefix: "8772", weight: 10, response_gt: "222222"}, #
Solo llamada
  %{called_prefix: "555", weight: 10, response_gt: "333333"}, #
Solo llamada

  # Peso 100: Menor prioridad - comodín de respaldo
  %{weight: 100, response_gt: "444444"} # Coincide con todo
]

# Ejemplos de coincidencia:
# Llamada: "877234567", Llamada: "555123" -> "111111" (peso 1, ambos
coinciden)
# Llamada: "877234567", Llamada: "999999" -> "222222" (peso 10, solo
llamada)
# Llamada: "999999999", Llamada: "555123" -> "333333" (peso 10, solo
llamada)
# Llamada: "999999999", Llamada: "888888" -> "444444" (peso 100,
comodín)
```

## Ejemplos

### Ejemplo 1: Dos Socios de Red

**Escenario:** Operas un SMSc con dos socios de red. Cada uno espera respuestas de un GT diferente.

```
config :omniss7,
  gt_nat_enabled: true,

  # GT predeterminado de SMSc (utilizado cuando GT NAT está
deshabilitado o no coincide ninguna regla)
  smsc_service_center_gt_address: "5551000",

  # Reglas de GT NAT para socios
  gt_nat_rules: [
    # Socio A (prefijo 4412) espera respuestas del GT 5551001
    %{calling_prefix: "4412", weight: 10, response_gt: "5551001"},

    # Socio B (prefijo 4413) espera respuestas del GT 5551002
```

```
%{calling_prefix: "4413", weight: 10, response_gt: "5551002"},  
# Predeterminado: usar GT estándar de SMSc (comodín de respaldo)  
%{weight: 100, response_gt: "5551000"}  
]
```

## Flujo de Tráfico:

Solicitud SRI-for-SM entrante de 44121234567:  
GT Llamada: 5551001 (tu GT que utiliza el Socio A)  
GT Llamante: 44121234567 (GT del Socio A)

Búsqueda de GT NAT:  
"44121234567" coincide con el prefijo "4412"  
GT de respuesta seleccionado: "5551001"

Respuesta SRI-for-SM a 44121234567:  
GT Llamada: 44121234567 (invertido)  
GT Llamante: 5551001 (NAT'd)  
networkNode-Number: 5551001 (en respuesta MAP)

## Ejemplo 2: HLR con GTs Regionales

**Escenario:** HLR nacional con diferentes GTs por región.

```
config :omniss7,  
  gt_nat_enabled: true,  
  hlr_service_center_gt_address: "555000", # GT HLR predeterminado  
  
  gt_nat_rules: [  
    # VLRs de la región norte (prefijo 5551)  
    %{calling_prefix: "5551", weight: 10, response_gt: "555100"},  
  
    # VLRs de la región sur (prefijo 5552)  
    %{calling_prefix: "5552", weight: 10, response_gt: "555200"},  
  
    # VLRs de la región oeste (prefijo 5553)  
    %{calling_prefix: "5553", weight: 10, response_gt: "555300"},  
  
    # Predeterminado para otras regiones (comodín)  
    %{weight: 100, response_gt: "555000"}  
  ]
```

## Ejemplo 3: Escenario de Migración

**Escenario:** Migrando gradualmente del antiguo GT al nuevo GT.

```
config :omniss7,
```

```

gt_nat_enabled: true,
hlr_service_center_gt_address: "123456789", # Antiguo GT
(predeterminado)

gt_nat_rules: [
  # Redes migradas (ya actualizaron sus configuraciones)
  %{calling_prefix: "555", weight: 10, response_gt: "987654321"},
# Nuevo GT
  %{calling_prefix: "666", weight: 10, response_gt: "987654321"},
# Nuevo GT

  # Todos los demás aún utilizan el antiguo GT (comodín)
  %{weight: 100, response_gt: "123456789"} # Antiguo GT
]

```

## Ejemplo 4: Coincidencia de Prefijo de Parte Llamada (NUEVO)

**Escenario:** Tienes múltiples GTs para diferentes servicios y deseas responder con el GT correcto basado en cuál GT fue llamado.

```

config :omniss7,
  gt_nat_enabled: true,

  gt_nat_rules: [
    # Cuando llaman a tu GT de SMS (5551xxx), responde con ese GT
    %{called_prefix: "5551", weight: 10, response_gt: "555100"},

    # Cuando llaman a tu GT de Voz (5552xxx), responde con ese GT
    %{called_prefix: "5552", weight: 10, response_gt: "555200"},

    # Cuando llaman a tu GT de Datos (5553xxx), responde con ese GT
    %{called_prefix: "5553", weight: 10, response_gt: "555300"},

    # Respaldo predeterminado
    %{weight: 100, response_gt: "555000"}
  ]

```

### Flujo de Tráfico:

Solicitud entrante al GT Llamado: 555100 (tu GT de SMS)  
 GT Llamante: 441234567 (cualquier llamador)

Búsqueda de GT NAT:  
 GT Llamado "555100" coincide con el prefijo "5551"  
 GT de respuesta seleccionado: "555100"

La respuesta utiliza el GT Llamante: 555100 (coincide con lo que llamaron)

## Ejemplo 5: Coincidencia Combinada de Prefijos de Llamada + Parte Llamada (AVANZADO)

**Escenario:** Diferentes socios llaman a diferentes GTs y deseas un control detallado.

```
config :omniss7,
  gt_nat_enabled: true,

  gt_nat_rules: [
    # Socio A llamando a tu GT de SMS - mayor prioridad (peso 1)
    %{calling_prefix: "4412", called_prefix: "5551", weight: 1,
response_gt: "555101"},

    # Socio B llamando a tu GT de SMS - mayor prioridad (peso 1)
    %{calling_prefix: "4413", called_prefix: "5551", weight: 1,
response_gt: "555102"},

    # Cualquiera llamando a tu GT de SMS - prioridad media (peso 10)
    %{called_prefix: "5551", weight: 10, response_gt: "555100"},

    # Socio A llamando a cualquier GT - prioridad media (peso 10)
    %{calling_prefix: "4412", weight: 10, response_gt: "555200"},

    # Respaldo predeterminado - baja prioridad (peso 100)
    %{weight: 100, response_gt: "555000"}
  ]
```

### Ejemplos de Coincidencia:

```
# Socio A llama al GT de SMS
Llamada: "441234567", Llamada: "555100"
→ Coincide con la regla de peso 1 (ambos prefijos) → "555101"

# Socio A llama al GT de Voz
Llamada: "441234567", Llamada: "555200"
→ Coincide con la regla de peso 10 (solo llamada) → "555200"

# Llamador desconocido llama al GT de SMS
Llamada: "999999999", Llamada: "555100"
→ Coincide con la regla de peso 10 (solo llamada) → "555100"

# Llamador desconocido llama al GT de Voz
Llamada: "999999999", Llamada: "555200"
→ Coincide con el comodín de peso 100 → "555000"
```

# Modos Operativos

GT NAT funciona en todos los modos operativos de OmniSS7:

## Modo HLR

GT NAT afecta:

- Respuestas de UpdateLocation (GT de HLR en respuesta)
- Mensajes de InsertSubscriberData (GT de HLR como parte llamante)
- Respuestas de SendAuthenticationInfo
- Respuestas de Cancel Location

Para más información sobre operaciones de HLR, consulta la [Guía de Configuración de HLR](#).

### Configuración:

```
config :omniss7,
  hlr_mode_enabled: true,
  hlr_service_center_gt_address: "5551234567", # GT HLR
predeterminado

  gt_nat_enabled: true,
  gt_nat_rules: [
    %{calling_prefix: "331", weight: 10, response_gt: "5551234568"},
# Francia
    %{calling_prefix: "44", weight: 10, response_gt: "5551234569"},
# Reino Unido
    %{weight: 100, response_gt: "5551234567"} # Comodín
predeterminado
  ]
```

## Modo SMSc

GT NAT afecta:

- Respuestas SRI-for-SM (campo networkNode-Number) - consulta [Detalles de SRI-for-SM](#)
- Acknowledgments de MT-ForwardSM

Para más información sobre operaciones de SMSc, consulta la [Guía de Configuración de SMSc](#).

### Configuración:

```
config :omniss7,
  smsc_mode_enabled: true,
```

```

smc_service_center_gt_address: "5559999", # GT SMC
predeterminado

gt_nat_enabled: true,
gt_nat_rules: [
  %{calling_prefix: "1", weight: 10, response_gt: "5559991"}, #
América del Norte
  %{calling_prefix: "44", weight: 10, response_gt: "5559992"}, #
Reino Unido
  %{calling_prefix: "86", weight: 10, response_gt: "5559993"}, #
China
  %{weight: 100, response_gt: "5559999"} # Comodín predeterminado
]

```

## Modo Gateway CAMEL

GT NAT afecta:

- Todas las respuestas a nivel SCCP (GT de gsmSCF como Parte Llamante)
- Respuestas de operaciones CAMEL/CAP (InitialDP, EventReportBCSM, etc.)
- Acknowledgments de RequestReportBCSMEvent
- Respuestas de ApplyCharging
- Respuestas de Continue

### Configuración:

```

config :omniss7,
  camelgw_mode_enabled: true,
  camel_gsmscf_gt_address: "55512341112", # GT gsmSCF predeterminado

  gt_nat_enabled: true,
  gt_nat_rules: [
    %{calling_prefix: "555", weight: 10, response_gt: "55512341111"},
# Red A
    %{calling_prefix: "666", weight: 10, response_gt: "55512311555"},
# Red B
    %{weight: 100, response_gt: "55512341112"} # Comodín
predeterminado
  ]

```

**Caso de Uso:** Cuando operas como un gsmSCF (Función de Control de Servicio) para múltiples redes, cada gsmSSF de la red puede esperar respuestas de un GT específico de gsmSCF. GT NAT asegura que se utilice el GT correcto basado en qué gsmSSF está llamando.

# Registro y Depuración

## Habilitar Registro de GT NAT

GT NAT incluye registro automático de todas las traducciones:

```
# En los registros, verás:  
[info] GT NAT [respuesta SRI-for-SM]: GT Llamante 877234567 -> GT de  
Respuesta 55512341112  
[info] GT NAT [UpdateLocation ISD]: GT Llamante 331234567 -> GT de  
Respuesta 55512341111  
[info] GT NAT [respuesta MAP BEGIN]: GT Llamante 441234567 -> GT de  
Respuesta 55512311555
```

El campo de contexto muestra dónde se aplicó el NAT:

- "respuesta SRI-for-SM" - En el controlador SRI-for-SM
- "UpdateLocation ISD" - En mensajes de InsertSubscriberData
- "UpdateLocation END" - En respuesta de UpdateLocation END
- "respuesta MAP BEGIN" - Respuestas MAP BEGIN genéricas
- "ACK ISD" - Acknowledgment de ISD
- "respuesta de error HLR" - Respuesta de error del HLR
- "respuesta CAMEL" - Respuestas de operaciones CAMEL/CAP (gsmSCF)

## Validación

El sistema valida la configuración de GT NAT al inicio:

```
# Verificar configuración de GT NAT  
iex> GtNat.validate_config()  
{:ok, [  
  %{calling_prefix: "8772", weight: 10, response_gt: "55512341112"},  
  %{calling_prefix: "8773", weight: 10, response_gt: "55512341111"}  
]}
```

```
# Verificar si está habilitado  
iex> GtNat.enabled?()  
true
```

```
# Obtener todas las reglas  
iex> GtNat.get_rules()  
[  
  %{calling_prefix: "8772", weight: 10, response_gt: "55512341112"},  
  %{calling_prefix: "8773", weight: 10, response_gt: "55512341111"}  
]
```

## Pruebas de GT NAT

Prueba la lógica de GT NAT programáticamente:

```
# Probar traducción con GT de llamada solo (called_gt es nil)
iex> GtNat.translate_response_gt("877234567", nil, "default_gt")
"55512341112"

# Probar traducción con ambos GT de llamada y llamada
iex> GtNat.translate_response_gt("877234567", "555123", "default_gt")
"55512341112"

# Probar con registro (GT llamado nil)
iex> GtNat.translate_response_gt_with_logging("877234567", nil,
"default_gt", "test")
# Registros: GT NAT [test]: GT Llamante 877234567 -> GT de Respuesta
55512341112
"55512341112"

# Probar con registro (ambos GTs)
iex> GtNat.translate_response_gt_with_logging("877234567", "555123",
"default_gt", "test")
# Registros: GT NAT [test]: GT Llamante 877234567, GT Llamado 555123
-> GT de Respuesta 55512341112
"55512341112"

# Probar sin coincidencia (devuelve predeterminado)
iex> GtNat.translate_response_gt("999999999", "888888", "default_gt")
"default_gt"
```

## Solución de Problemas

### Problema: GT NAT No Funciona

#### Verificación 1: ¿Está habilitado?

```
iex> Application.get_env(:omniss7, :gt_nat_enabled)
true # Debería ser true
```

#### Verificación 2: ¿Están configuradas las reglas?

```
iex> Application.get_env(:omniss7, :gt_nat_rules)
[%{calling_prefix: "8772", response_gt: "55512341112"}, ...] #
Debería devolver lista
```

**Verificación 3: Verificar registros** Busca "GT NAT" en los registros para ver si se están realizando traducciones.

## Problema: GT Incorrecto en Respuestas

**Síntoma:** Las respuestas utilizan una dirección GT inesperada

**Causa:** La coincidencia de prefijos de regla podría ser demasiado amplia o la regla predeterminada está capturando tráfico

**Solución:** Revisar pesos y prefijos de las reglas:

```
# MALO: Comodín con bajo peso (captura todo primero)
gt_nat_rules: [
  %{weight: 1, response_gt: "111111"},           # ¡Esto coincide
  con todo primero!
  %{calling_prefix: "8772", weight: 10, response_gt: "222222"} #
  Nunca alcanzado
]

# BUENO: Reglas específicas con peso más bajo, comodín con peso más
alto
gt_nat_rules: [
  %{calling_prefix: "8772", weight: 10, response_gt: "222222"}, #
  Específico, bajo peso
  %{weight: 100, response_gt: "111111"} # Comodín, alto peso
  (respaldo)
]
```

## Problema: GT NAT No Aplicado a Tipo de Mensaje Específico

**Síntoma:** Algunas respuestas utilizan GT NAT'd, otras no

**Cobertura Actual:**

- ♦ GT de Llamada SCCP (todas las respuestas)
- ♦ Respuestas SRI-for-SM (networkNode-Number)
- ♦ Mensajes de UpdateLocation ISD (GT de HLR)
- ♦ Respuestas de UpdateLocation END
- ♦ Acknowledgments de ISD
- ♦ Respuestas MAP BEGIN

Si un tipo de mensaje específico no está utilizando GT NAT, puede que aún no esté implementado. Verifica el código fuente o contacta con soporte.

## Consideraciones de Rendimiento

### Rendimiento de Búsqueda

GT NAT utiliza coincidencia de prefijos simple con complejidad  $O(n)$  donde  $n$  es el

número de reglas.

### Consejos de rendimiento:

- Mantener el recuento de reglas por debajo de 100 para un mejor rendimiento
- Utilizar prefijos específicos para reducir el recuento de reglas
- La regla predeterminada (prefijo vacío) debe ser la última

### Benchmark (sistema típico):

- 10 reglas: < 1µs por búsqueda
- 50 reglas: < 5µs por búsqueda
- 100 reglas: < 10µs por búsqueda

### Uso de Memoria

Cada regla requiere ~100 bytes de memoria:

- 10 reglas ≈ 1 KB
- 100 reglas ≈ 10 KB

## Mejores Prácticas

### 1. Siempre Incluir una Regla de Respaldo Comodín

```
gt_nat_rules: [  
  {%calling_prefix: "8772", weight: 10, response_gt: "111111"},  
  {%calling_prefix: "8773", weight: 10, response_gt: "222222"},  
  {%weight: 100, response_gt: "default_gt"} # Siempre tener un  
comodín con alto peso  
]
```

### 2. Utilizar Prefijos y Pesos Significativos

```
# BUENO: Prefijos claros y específicos con pesos apropiados  
{%calling_prefix: "331", weight: 10, response_gt: "..."} # Francia  
{%calling_prefix: "44", weight: 10, response_gt: "..."} # Reino  
Unido  
  
# MALO: Prefijos demasiado amplios o pesos confusos  
{%calling_prefix: "3", weight: 5, response_gt: "..."} #  
Demasiados países  
{%calling_prefix: "331", weight: 100, response_gt: "..."} # El peso  
debería ser más bajo para reglas específicas
```

### 3. Documentar Tus Reglas

```
gt_nat_rules: [  
  # Socio XYZ - red del Reino Unido (rango GT: 4412xxxxxxx)  
  # Peso 10: Prioridad estándar del socio  
  %{calling_prefix: "4412", weight: 10, response_gt: "5551001"},  
  
  # Socio ABC - red de Francia (rango GT: 33123xxxxxx)  
  # Peso 10: Prioridad estándar del socio  
  %{calling_prefix: "33123", weight: 10, response_gt: "5551002"}  
]
```

### 4. Probar Antes de Desplegar

```
# Probar en iex antes de desplegar  
iex> GtNat.translate_response_gt("44121234567", nil, "default")  
"5551001" # Resultado esperado  
  
# Probar con GT llamado  
iex> GtNat.translate_response_gt("44121234567", "555123", "default")  
"5551001" # Resultado esperado
```

### 5. Monitorear Registros

Habilitar el registro de nivel INFO para ver todas las traducciones de GT NAT en producción.

## Integración con Otras Funciones

### Modo STP

GT NAT funciona independientemente del enrutamiento STP. STP enruta basado en códigos de punto y GTs de destino, mientras que GT NAT maneja la dirección de respuesta.

Para más información sobre enrutamiento STP, consulta la [Guía de Configuración de STP](#).

### Integración CAMEL

GT NAT está **totalmente integrado** con operaciones CAMEL/CAP:

#### Capa SCCP:

- GT de Parte Llamante en todas las respuestas CAMEL
- Aplicado automáticamente basado en el GT de gsmSSF entrante

## Configuración:

- `camel_gsmSCF_gt_address` - GT gsmSCF predeterminado (opcional)
- Si no está configurado, utiliza el GT de Parte Llamada de la solicitud entrante
- Las reglas de GT NAT anulan el predeterminado basado en el prefijo de la parte que llama

## Ejemplo:

```
# Cuando gsmSSF 555123456 llama a tu gsmSCF
# Entrante: Llamado=55512341112, Llamante=555123456
# Búsqueda de GT NAT: "555" -> response_gt="55512341111"
# Respuesta: Llamado=555123456, Llamante=55512341111
```

## Balanceo de Carga

GT NAT se puede combinar con balanceo de carga M3UA para una gestión avanzada del tráfico.

# Guía de Migración

## Habilitar GT NAT en un Sistema Existente

### 1. Preparar Configuración

```
# Agregar a runtime.exs (mantener deshabilitado inicialmente)
config :omniss7,
  gt_nat_enabled: false, # Comenzar deshabilitado
  gt_nat_rules: [
    # Tus reglas aquí con pesos
    %{calling_prefix: "877", weight: 10, response_gt: "111111"},
    %{weight: 100, response_gt: "999999"} # Comodín de respaldo
  ]
```

### 2. Probar Configuración

```
# Validar que la configuración compile
mix compile

# Probar en iex
iex -S mix
iex> GtNat.validate_config()
```

### 3. Habilitar en Staging

```
gt_nat_enabled: true # Cambiar a true
```

#### 4. **Monitorear Registros**

```
tail -f log/omniss7.log | grep "GT NAT"
```

#### 5. **Desplegar en Producción**

- Desplegar durante la ventana de mantenimiento
- Monitorear las primeras 24 horas de cerca
- Tener un plan de retroceso listo (configurar `gt_nat_enabled: false`)

## **Soporte**

Para problemas o preguntas:

- Verificar registros para mensajes "GT NAT"
- Validar configuración con `GtNat.validate_config()`
- Revisar la sección de solución de problemas de esta guía
- Contactar con el soporte de OmniSS7 con extractos de registros

# Guía de Configuración de HLR

[— Volver a la Documentación Principal](#)

Esta guía proporciona la configuración para usar OmniSS7 como un **Registro de Ubicación del Hogar (HLR/HSS)** con **OmniHSS** como la base de datos de suscriptores en el backend.

## Integración de OmniHSS

El modo HLR de OmniSS7 funciona como un **frontend de señalización SS7** que se conecta con **OmniHSS**, un servidor de suscriptores en el hogar (HSS) completo. Esta arquitectura separa las preocupaciones:

- **OmniSS7 (Frontend HLR):** Maneja toda la señalización del protocolo SS7/MAP, el enrutamiento SCCP y la comunicación de red
- **OmniHSS (Backend HSS):** Gestiona los datos de suscriptores, la autenticación, la provisión y características avanzadas

### ¿Por qué OmniHSS?

OmniHSS proporciona gestión de suscriptores de grado de operador con características que incluyen:

- **Soporte Multi-IMSI:** Cada suscriptor puede tener múltiples IMSIs asociadas con un solo MSISDN para roaming internacional, cambio de red y provisión de eSIM
- **Autenticación Flexible:** Soporte para algoritmos de autenticación tanto Milenage (3G/4G/5G) como COMP128 (2G)
- **Seguimiento de Sesiones de Circuito y Paquete:** Seguimiento independiente de registros de red CS (conmutación de circuitos) y PS (conmutación de paquetes)
- **Provisionamiento Avanzado:** Perfiles de servicio personalizables, servicios suplementarios y datos de suscripción CAMEL
- **Diseño API-First:** API HTTP RESTful para integración con sistemas de facturación, CRM y provisión
- **Actualizaciones en Tiempo Real:** Seguimiento de ubicación, gestión de sesiones y generación de vectores de autenticación

Todos los datos de suscriptores, credenciales de autenticación y configuraciones de servicio se almacenan y gestionan en OmniHSS. OmniSS7 consulta a OmniHSS a través de llamadas API HTTPS para responder a operaciones MAP como UpdateLocation, SendAuthenticationInfo y SendRoutingInfo.

**Importante:** El modo HLR de OmniSS7 es un **frontend de señalización solamente**. Toda la lógica de gestión de suscriptores, algoritmos de autenticación, reglas de provisión y operaciones de base de datos son manejadas por OmniHSS. Esta guía cubre la configuración del protocolo SS7/MAP en OmniSS7. Para información sobre la provisión de suscriptores, configuración de autenticación, perfiles de servicio y operaciones administrativas, **consulte la documentación de OmniHSS**.

### Soporte Multi-IMSI

OmniHSS admite nativamente configuraciones **Multi-IMSI**, permitiendo que un solo suscriptor (identificado por MSISDN) tenga múltiples IMSIs. Esto permite:

- **Perfiles de Roaming Internacional:** Diferentes IMSIs para diferentes regiones para reducir costos de roaming
- **eSIM Multi-Perfil:** Múltiples perfiles de red en un solo dispositivo compatible con eSIM
- **Cambio de Red:** Cambio sin problemas entre redes sin cambiar MSISDN
- **Coordinación de Doble SIM:** Coordinación entre múltiples SIMs físicas o virtuales
- **Pruebas y Desarrollo:** Múltiples IMSIs de prueba apuntando al mismo suscriptor

### Cómo funciona:

- Cada IMSI tiene sus propias credenciales de autenticación (KI, OPc, algoritmo)
- Cada IMSI puede tener registros de sesión de circuito y paquete independientes
- Los servicios y perfiles de suscriptores pueden ser compartidos o personalizados por IMSI
- OmniSS7 consulta a OmniHSS por IMSI, y OmniHSS devuelve los datos de suscriptor apropiados
- Los sistemas de facturación pueden rastrear el uso por IMSI mientras asocian todas las IMSIs a una sola cuenta

### Ejemplo de escenario Multi-IMSI:

```
Suscriptor MSISDN: +1-555-123-4567
├─ IMSI 1: 310260123456789 (Red Nacional de EE. UU. - autenticación Milenage)
├─ IMSI 2: 208011234567890 (Perfil de Roaming de Francia - autenticación Milenage)
└─ IMSI 3: 440201234567891 (Perfil de Roaming del Reino Unido - autenticación COMP128)
```

Las tres IMSIs pueden ser utilizadas independientemente para el registro en la red, pero todas pertenecen a la misma cuenta de suscriptor. OmniHSS gestiona el mapeo IMSI-suscriptor y asegura la autenticación y provisión adecuadas para cada IMSI.

OmniSS7 Stack v1.0.0

Download for OmniStack  
© 2025 Omnicouch

SS7 Events **Active Subscribers** Total: 0 Auto-refresh every 2 seconds

SS7 Client

M3UA

HLR Links

Active Subscribers **Clear All Subscribers**

Resources

IMSI	VLR Number	MSC Number	Updated At	Duration
 No Active Subscribers Subscribers will appear here when they send UpdateLocation requests.				

## Tabla de Contenidos

1. [Integración de OmniHSS](#)
  2. [Soporte Multi-IMSI](#)
  3. [¿Qué es el Modo HLR?](#)
  4. [Habilitando el Modo HLR](#)
  5. [Base de Datos de Suscriptores](#)
  6. [Vectores de Autenticación](#)
  7. [Actualizaciones de Ubicación](#)
  8. [Integración CAMEL](#)
  9. [Manejo de Suscriptores en Roaming](#)
  10. [Operaciones HLR](#)
- [Mapeo de Campos de Respuesta](#)
    - [SendRoutingInfo \(SRU\)](#)
    - [UpdateLocation \(ULN\)](#)
    - [SendRoutingInfoForSM](#)
  - [Resumen de Fuentes de Campo](#)

## ¿Qué es el Modo HLR?

El **Modo HLR** permite que OmniSS7 funcione como un Registro de Ubicación del Hogar para:

- **Gestión de Suscriptores:** Almacenar y gestionar datos de suscriptores
- **Autenticación:** Generar vectores de autenticación para acceso a la red
- **Seguimiento de Ubicación:** Procesar actualizaciones de ubicación de VLRs
- **Información de Enrutamiento:** Proporcionar información de enrutamiento para llamadas y SMS

## Arquitectura HLR

---

### Habilitando el Modo HLR

OmniSS7 puede operar en diferentes modos. Para usarlo como un HLR, necesitas habilitar el modo HLR en la configuración.

#### Cambiando a Modo HLR

El config/runtime.exe de OmniSS7 contiene tres modos operativos preconfigurados. Para habilitar el modo HLR:

1. **Abrir** config/runtime.exe
2. **Encontrar** las tres secciones de configuración (líneas 53-174):
  - Configuración 1: Modo STP (líneas 53-85)
  - Configuración 2: Modo HLR (líneas 87-123)
  - Configuración 3: Modo SMS (líneas 125-174)
3. **Comentar** la configuración actualmente activa (agregar # a cada línea)
4. **Descomentar** la configuración HLR (quitar # de las líneas 87-123)
5. **Personalizar** los parámetros de configuración según sea necesario
6. **Reiniciar** la aplicación: iex -S mix

#### Configuración del Modo HLR

La configuración completa del HLR se ve así:

```
config :omniSS7,
# Flags de modo - Habilitar características HLR solamente
map_client_enabled: true,
hlr_mode_enabled: true,
smsc_mode_enabled: false,

# Configuración de la API Backend de OmniHSS
hlr_api_base_url: "https://10.180.2.140:8443",

# Dirección GT del Centro de Servicio HLR para operaciones SMS
hlr_service_center_gt_address: "1234567890",

# Configuración de Mapeo MSISDN - IMSI
# Ver: sección de Mapeo MSISDN - IMSI para detalles
hlr_imsi_plmn_prefix: "50557",
hlr_msisdn_country_code: "61",
hlr_msisdn_nsn_offset: 0,
hlr_msisdn_nsn_length: 9,

# Configuración de InsertSubscriberData
# Modo de Acceso a la Red: :packetAndCircuit, :packetOnly, o :circuitOnly
isd_network_access_mode: :packetAndCircuit,

# Enviar ISD #2 (datos de Servicios Suplementarios)
isd_send_ss_data: true,

# Enviar ISD #3 (datos de Barrido de Llamadas)
isd_send_call_barring: true,

# Configuración CAMEL (para respuestas SendRoutingInfo)
# Clave de Servicio para la iniciación del servicio CAMEL
camel_service_key: 11_110,

# Punto de Detección de Disparador CAMEL
# Opciones: :termAttemptAuthorized, :tBusy, :tNoAnswer, :tAnswer
camel_trigger_detection_point: :termAttemptAuthorized,

# Prefijos de VLR en el Hogar
# Lista de prefijos de dirección VLR que se consideran red "local"
# Si el VLR del suscriptor comienza con uno de estos prefijos, usar respuesta SRI estándar
# De lo contrario, el suscriptor está en roaming y necesitamos enviar PRN para obtener MSRN
home_vlr_prefixes: ["123456*"],

# Configuración de Conexión M3UA
# Conectarse como ASP para recibir operaciones MAP (UpdateLocation, SendAuthInfo, etc.)
map_client_m3ua: %{
  mode: "ASP",
  callback: {MapClient, :handle_payload, []},
  process_name: :hlr_client_asp,
  # Punto final local (sistema HLR)
  local_ip: {10, 179, 4, 11},
  local_port: 2905,
  # Punto final remoto STP
  remote_ip: {10, 179, 4, 10},
  remote_port: 2905,
  routing_context: 1
}
```

OmniSS7 Stack 1.0.0 Licensed to: Omnitouch  
© 2023 Omnitouch

SS7 Events

SS7 Client

M3UA

HLR Links

Active

Subscribers

Resources

Configuration

### HLR API Connection Status

Last updated: 2025-10-26 00:00:54 UTC Check now

DOWN

HLR HTTP API

Latency: 46.14 ms

Version/Role: -

Host: https://10.180.2.140:8443

Details: {:tls\_alert, {:bad\_certificate, ~c"TLS client: In state wait\_cert\_cr at ssl\_handshake.erl:2178 generated CLIENT ALERT: Fatal - Bad Certificate\n selfsigned\_peer"}}

**Subscriber Lookup**

Lookup Type: IMSI Value

IMSI:  Lookup

**Generate Authentication Vectors**

IMSI:  Generate

## Parámetros de Configuración a Personalizar

Para una referencia completa de todos los parámetros de configuración, consulte la [Referencia de Configuración](#).

Parámetro	Tipo	Predeterminado	Descripción	Ejemplo
hlr_api_base_url	String	Requerido	OmniHSS punto final de la API backend	"https://10.179.3.219:8443"
hlr_service_center_gt_address	String	Requerido	Dirección GT HLR utilizada en respuestas UpdateLocation	"5551234568"
smsc_service_center_gt_address	String	Requerido	Dirección GT SMSC devuelta en respuestas SRI-for-SM	"5551234567"
hlr_smsc_alert_gts	Lista	1	Lista de GTs SMSc para enviar alertServiceCenter después de UpdateLocation	"15559876543", "15559876544"
hlr_alert_location_expiry_seconds	Entero	172800	Tiempo de expiración de ubicación (segundos) cuando SMSc recibe alertServiceCenter	86400
hlr_imsi_plmn_prefix	String	"90557"	Prefijo PLMN (MCC+MNC) para mapeo MSISDN-IMSI (ver <a href="#">MSISDN =&gt; IMSI Mapping</a> )	"001001"
hlr_msisdn_country_code	String	"61"	Prefijo de código de país para mapeo inverso IMSI-MSISDN (ver <a href="#">MSISDN =&gt; IMSI Mapping</a> )	"1"
hlr_msisdn_nsn_offset	Entero	0	Desplazamiento en MSISDN para extracción de NSN (ver <a href="#">MSISDN =&gt; IMSI Mapping</a> )	0
hlr_msisdn_nsn_length	Entero	9	Longitud del Número Nacional de Suscriptor a extraer (ver <a href="#">MSISDN =&gt; IMSI Mapping</a> )	10
isd_network_access_mode	Atomo	:packetAndCircuit	Modo de acceso a la red para InsertSubscriberData	:packetOnly
isd_send_ss_data	Booleano	true	Enviar ISD #2 con datos de Servicios Suplementarios	false
isd_send_call_barring	Booleano	true	Enviar ISD #3 con datos de Barring de Llamadas	false
camel_service_key	Entero	11_110	Clave de servicio CAMEL para respuestas SendRoutingInfo	100
camel_trigger_detection_point	Atomo	:termAttemptAuthorized	Punto de detección de disparador CAMEL	:tBusy
home_vlr_prefixes	Lista	["5551231"]	Lista de prefijos de dirección VLR considerados red "local"	["555123"]
local_ip	Tupla	Requerido	Dirección IP de su sistema HLR	{10, 179, 4, 12}
local_port	Entero	2905	Puerto SCTP local	2905
remote_ip	Tupla	Requerido	Dirección IP STP para conectividad SS7	{10, 179, 4, 10}
remote_port	Entero	2905	Puerto SCTP remoto	2905
routing_context	Entero	1	ID de contexto de enrutamiento M3UA	1

## Qué Ocurre Cuando se Habilita el Modo HLR

Cuando hlr\_mode\_enabled: true, la interfaz web mostrará:

- **Eventos SS7** - Registro de eventos
- **Cliente SS7** - Pruebas de operación MAP
- **M3UA** - Estado de conexión
- **Enlaces HLR** - Estado de la API HLR + gestión de suscriptores – *Específico de HLR*
- **Recursos** - Monitoreo del sistema
- **Configuración** - Visor de configuración

Las pestañas **Enrutamiento**, **Prueba de Enrutamiento** y **Enlaces SMSc** estarán ocultas.

## Notas Importantes

- **Configuración Requerida:** El parámetro hlr\_service\_center\_gt\_address es **obligatorio**. La aplicación no podrá iniciarse si no está configurado.
- **Backend de OmniHSS:** La API de OmniHSS debe ser accesible en la hlr\_api\_base\_url configurada
- **Tiempo de Espera de Solicitud API:** Todas las solicitudes API de OmniHSS tienen un **tiempo de espera codificado de 5 segundos**
- **Tiempo de Espera de Solicitud MAP:** Todas las solicitudes MAP (SRI, UpdateLocation, SendAuthInfo, etc.) tienen un **tiempo de espera codificado de 10 segundos**
- **Tiempo de Espera ISD:** Cada mensaje InsertSubscriberData (ISD) es una secuencia UpdateLocation tiene un **tiempo de espera codificado de 10 segundos**
- Se requiere conexión M3UA al STP para recibir operaciones MAP
- Después de cambiar de modo, debes reiniciar la aplicación para que los cambios surtan efecto
- **Interfaz Web:** Consulta la [Guía de Interfaz Web](#) para información sobre el uso de la interfaz web
- **Acceso API:** Consulta la [Guía API](#) para documentación de la API REST y acceso a Swagger UI

## Base de Datos de Suscriptores

OmniHSS gestiona todos los datos de suscriptores incluyendo identidades, credenciales de autenticación, perfiles de servicio e información de ubicación. OmniSS7 recupera estos datos a través de llamadas API RESTful.

### Modelo de Suscriptor de OmniHSS

OmniHSS almacena información completa del suscriptor:

- **Múltiples IMSIs por suscriptor:** Soporte para configuraciones Multi-IMSI (eSIM, perfiles de roaming, cambio de red)
- **Credenciales de autenticación:** Selección de Ki, OPc y algoritmo (Milenage o COMP128)
- **Perfiles de servicio:** Categoría de suscriptor, servicios permitidos, parámetros de QoS
- **Seguimiento de ubicación:** Seguimiento independiente de VLR/MSc; actual (sesión de circuito) y SGSN/GGSN (sesión de paquete)
- **Datos de suscripción CAMEL:** Claves de servicio, puntos de activación y direcciones gsmSCF
- **Servicios suplementarios:** Desvío de llamadas, barring, espera, configuraciones CLIP/CLIR
- **Estado administrativo:** Habilitado/deshabilitado, restricciones de servicio, fechas de expiración

## Vectores de Autenticación

### Generar Vectores de Autenticación

OmniHSS genera vectores de autenticación utilizando los algoritmos Milenage o COMP128 basados en el método de autenticación configurado de cada suscriptor. Cuando OmniSS7 recibe solicitudes MAP de `sendAuthenticationInfo`:

1. OmniSS7 extrae el IMSI de la solicitud MAP
2. OmniSS7 llama a la API de OmniHSS para generar vectores de autenticación
3. OmniHSS recupera las credenciales Ki y OPc del suscriptor
4. OmniHSS genera el número solicitado de vectores (RAND, XRES, CK, IK, AUTN)
5. OmniSS7 codifica los vectores en formato MAP y los devuelve al VLR/SGSN solicitante

## Integración de la API de OmniHSS

OmniSS7 se comunica con OmniHSS a través de la API REST HTTPS para recuperar información de suscriptores, actualizar datos de ubicación y generar vectores de autenticación:

```
config :omniss7,  
  hlr_api_base_url: "https://omnihss-server:8443"
```

Cuando OmniSS7 recibe operaciones MAP de la red SS7, consulta a OmniHSS para:

- **Recuperar datos de suscriptor** por IMSI o MSISDN
- **Generar vectores de autenticación** utilizando credenciales Ki/OpC almacenadas
- **Actualizar la ubicación de la sesión de circuito** cuando los suscriptores realizan UpdateLocation
- **Verificar el estado del suscriptor** y los derechos de servicio

## Actualizaciones de Ubicación

### Procesamiento de Actualización de Ubicación

Al recibir solicitudes MAP de **updateLocation**, OmniSS7 coordina con OmniHSS para registrar al suscriptor en un nuevo VLR:

1. **Extraer información de ubicación** de la solicitud UpdateLocation (IMSI, nuevo GT de VLR, nuevo GT de MSC)
2. **Consultar a OmniHSS** para verificar que el suscriptor existe y está habilitado
3. **Actualizar la sesión de circuito** en OmniHSS con la nueva ubicación VLR/MSC
4. **Enviar mensajes InsertSubscriberData (ISD)** para provisionar al suscriptor en el nuevo VLR
5. **Devolver la respuesta UpdateLocation** al VLR (incluye GT HLR de hlr\_service\_center\_gt\_address)
6. **Enviar alertServiceCenter** a los GTs SMSc configurados (si hlr\_smsc\_alert\_gts está poblado)

**Nota:** El parámetro de configuración hlr\_service\_center\_gt\_address especifica el Título Global del HLR que se devuelve en las respuestas UpdateLocation. Esto permite que el VLR/MSC identifique y enrute los mensajes de vuelta a este HLR.

### Integración del Centro de Servicio de Alerta

Después de una actualización de ubicación exitosa, el HLR puede notificar específicamente a los sistemas SMSc que un suscriptor ahora es alcanzable enviando mensajes **alertServiceCenter** (código de operación MAP 64). Para información sobre cómo el SMSc maneja estas alertas, consulte [Manejo del Centro de Servicio de Alerta en la Guía SMSc](#).

### Configuración

Configure la lista de Títulos Globales SMSc a notificar:

```
config :omniss7,  
  # Lista de GTs SMSc para enviar alertServiceCenter después de UpdateLocation  
  hlr_smsc_alert_gts: [  
    "15559878543",  
    "15559878544"  
  ],  
  # Tiempo de expiración de ubicación cuando SMSc recibe alertServiceCenter (predeterminado: 48 horas)  
  hlr_alert_location_expiry_seconds: 172800
```

### Diagrama de Flujo

### Comportamiento

Cuando un suscriptor realiza UpdateLocation:

1. HLR envía alertServiceCenter a **cada** GT SMSc en la lista hlr\_smsc\_alert\_gts
2. El mensaje incluye el MSISDN del suscriptor
3. HLR utiliza hlr\_service\_center\_gt\_address como el GT del llamante
4. Direccionamiento SCCP, SSN de llamada=6 (HLR), SSN llamado=8 (SMSc)

El SMSc recibe la alerta y:

- **Elimina el prefijo TON/NPI** del MSISDN (por ejemplo, "19123123213" → "123123213")
- **Marca al suscriptor** como alcanzable en su base de datos de ubicación (a través de POST a /api/location)
- **Establece el campo user\_agent** al GT del HLR al llamar a la API (para rastrear qué HLR envió la alerta)
- **Establece el tiempo de expiración de ubicación** basado en hlr\_alert\_location\_expiry\_seconds
- **Rastrea al suscriptor** en el Rastreador de Suscriptores del SMSc para monitorear

### Pruebas

Utilice la página **Suscriptores Activos** en la interfaz web para enviar manualmente mensajes alertServiceCenter para pruebas:

1. Navegue a la pestaña "Suscriptores Activos"
2. Encuentre la sección "Probar Centro de Servicio de Alerta"
3. Ingrese MSISDN, GT SMSc y GT HLR (los valores predeterminados se completan automáticamente desde la configuración)
  - GT SMSc predeterminado es la primera entrada en hlr\_smsc\_alert\_gts
  - GT HLR predeterminado es hlr\_service\_center\_gt\_address
4. Haga clic en "Enviar alertServiceCenter"

Esto es útil para probar el manejo de alertas del SMSc sin requerir un flujo completo de UpdateLocation. El formulario utiliza validación phx-blur para evitar mostrar errores mientras se escribe.

### Configuración de InsertSubscriberData (ISD)

Después de una actualización de ubicación exitosa, el HLR envía datos de provisión de suscriptor al VLR utilizando mensajes **InsertSubscriberData (ISD)**. La configuración de ISD permite personalizar qué datos se envían y cómo.

Para referencia de parámetros de configuración, consulte [Configuración ISD en la Referencia de Configuración](#).

### Secuencia ISD

El HLR puede enviar hasta 3 mensajes ISD secuenciales:

1. **ISD #1** (Siempre enviado) - Datos básicos del suscriptor:
  - IMSI
  - MSISDN
  - Categoría del suscriptor
  - Estado del suscriptor (serviceGranted)
  - Lista de servicios de portadora
  - Lista de telesestos
  - Modo de acceso a la red
2. **ISD #2** (Opcional) - Datos de Servicios Suplementarios (SS):
  - Configuraciones de desvío de llamadas (incondicional, ocupado, sin respuesta, no alcanzable)
  - Espera de llamadas
  - Retención de llamadas
  - Servicio de múltiples partes
  - Estado y características del servicio suplementario
3. **ISD #3** (Opcional) - Datos de Barring de Llamadas:
  - Barring de todas las llamadas salientes (BAOC)
  - Barring de llamadas internacionales salientes (BOIC)
  - Datos de restricción de acceso

### Opciones de Configuración

```
# Configuración de InsertSubscriberData  
# Modo de Acceso a la Red: :packetAndCircuit, :packetOnly, o :circuitOnly  
isd_network_access_mode: :packetAndCircuit,  
# Enviar ISD #2 (datos de Servicios Suplementarios)  
isd_send_ss_data: true,  
# Enviar ISD #3 (datos de Barring de Llamadas)  
isd_send_call_barring: true,
```

### Modo de Acceso a la Red

El parámetro isd\_network\_access\_mode controla qué tipo de acceso a la red se permite al suscriptor:

Valor	Descripción	Caso de Uso
:packetAndCircuit	Tanto conmutación de paquetes (GPRS/LTE) como conmutación de circuitos (voz)	Predeterminado - Suscriptores de servicio completo
:packetOnly	Solo conmutación de paquetes (datos/LTE)	Tarjetas SIM solo de datos, dispositivos IoT
:circuitOnly	Solo conmutación de circuitos (voz/SMS)	Dispositivos heredados, planes solo de voz

### Controlando Mensajes ISD

Puedes controlar qué mensajes ISD se envían según tus requisitos de red:

**Enviar todos los ISDs** (Predeterminado - Conjunto completo de características):

```
isd_send_ss_data: true,  
isd_send_call_barring: true,
```

**Enviar solo datos básicos del suscriptor** (Provisionamiento mínimo):

```
isd_send_ss_data: false,  
isd_send_call_barring: false,
```

**Enviar datos básicos + servicios suplementarios** (Sin barring de llamadas):

```
isd_send_ss_data: true,  
isd_send_call_barring: false,
```

### Ejemplo de Flujo ISD

Cuando se recibe UpdateLocation:

```
HLR → HLR: UpdateLocation (INICIO)  
HLR → VLR: InsertSubscriberData #1 (CONTINUAR) - Datos básicos  
HLR → HLR: ISD #1 ACK (CONTINUAR)
```

```
HLR - VLR: InsertSubscriberData #2 (CONTINUAR) - Datos SS [si está habilitado]
VLR - HLR: ISD #2 ACK (CONTINUAR)
HLR - VLR: InsertSubscriberData #3 (CONTINUAR) - Barring de llamadas [si está habilitado]
VLR - HLR: ISD #3 ACK (CONTINUAR)
HLR - VLR: Respuesta UpdateLocation (FIN)
```

Si `isd_send_ss_data` o `isd_send_call_barring` están configurados como `false`, esos mensajes ISD se omiten, y el `UpdateLocation` FIN se envía antes.

#### Mejores Prácticas

- **Configuración Predeterminada:** Utilice `packetAndCircuit` y habilite todos los ISDs para máxima compatibilidad
- **IoT/M2M:** Utilice `packetOn` y desactive datos SS/barring de llamadas para dispositivos solo de datos
- **Interoperabilidad:** Algunos VLRs más antiguos pueden no soportar todos los servicios suplementarios - desactive `isd_send_ss_data` si encuentra problemas
- **Rendimiento:** Desactivar ISDs no utilizados reduce la sobrecarga de mensajes y acelera las actualizaciones de ubicación

## Integración CAMEL

### Configuración CAMEL para SendRoutingInfo

Al responder a solicitudes de **SendRoutingInfo** (SRI) de un GMSC (Gateway MSC), el HLR puede instruir al GMSC para que invoque servicios CAMEL para enrutamiento inteligente de llamadas y control de servicios.

Para referencia de parámetros de configuración, consulte [Configuración CAMEL en la Referencia de Configuración](#).

#### ¿Qué es CAMEL?

CAMEL (Aplicaciones Personalizadas para Lógica Mejorada de Redes Móviles) es un protocolo que permite servicios de red inteligentes en redes GSM/UMTS. Permite a los operadores de red implementar servicios de valor añadido como:

- Facturación prepago
- Filtrado y barring de llamadas
- Redes Privadas Virtuales (VPN)
- Servicios de tarifa premium
- Desvío de llamadas con lógica personalizada
- Servicios basados en ubicación

#### Opciones de Configuración

```
# Configuración CAMEL (para respuestas SendRoutingInfo)
# Clave de Servicio para la iniciación del servicio CAMEL
camel_service_key: 11_110,

# Punto de Detección de Disparador CAMEL
# Opciones: :termAttemptAuthorized, :tBusy, :tNoAnswer, :tAnswer
camel_trigger_detection_point: :termAttemptAuthorized,
```

#### Clave de Servicio

La `camel_service_key` identifica qué servicio CAMEL debe ser invocado en el `gsmSCF` (Función de Control de Servicio). Este es un identificador numérico configurado en tu red:

Clave de Servicio	Caso de Uso Típico
11_110	Control de llamada prepago terminante (predeterminado)
300	Servicio prepago de origen
200	Desvío de llamadas con lógica personalizada
300	Servicio de Red Privada Virtual (VPN)
Personalizado	Servicios específicos del operador

#### Ejemplo de Configuración:

```
# Para control de llamada prepago terminante
camel_service_key: 11_110,

# Para servicio VPN
camel_service_key: 300,
```

#### Punto de Detección de Disparador

El `camel_trigger_detection_point` especifica cuándo debe activarse el servicio CAMEL durante la configuración de la llamada:

Punto de Detección	Descripción	Cuándo se Activa
:termAttemptAuthorized	Intento de llamada autorizado (predeterminado)	Antes de que la llamada sea enrutada al suscriptor
:tBusy	Ocupado terminante	Cuando el suscriptor está ocupado
:tNoAnswer	Sin respuesta terminante	Cuando el suscriptor no responde
:tAnswer	Respuesta terminante	Cuando el suscriptor responde la llamada

#### Ejemplos de Configuración:

**Control prepago estándar** (disparador antes del enrutamiento):

```
camel_trigger_detection_point: :termAttemptAuthorized,
```

**Manejo personalizado de ocupado** (disparador cuando está ocupado):

```
camel_trigger_detection_point: :tBusy,
```

**Facturación basada en respuesta** (disparador en respuesta):

```
camel_trigger_detection_point: :tAnswer,
```

#### Respuesta SRI con CAMEL

Cuando está configurado, las respuestas `SendRoutingInfo` incluyen información de suscripción CAMEL:

```
GMSC - HLR: SendRoutingInfo (INICIO)
HLR - GMSC: Respuesta SRI (FIN) con:
- IMSI
- Número de VLR
- Estado del suscriptor
- Información de enrutamiento CAMEL:
  * Clave de Servicio: 11_110
  * Dirección gsmSCF: <dirección configurada>
  * Punto de Detección de Disparador: termAttemptAuthorized
  * Manejo de Llamadas por Defecto: continueCall
```

GMSC contacta a `gsmSCF` en el punto de disparo para ejecutar el servicio CAMEL.

#### Mejores Prácticas

- **Redes de Producción:** Utilice claves de servicio estandarizadas acordadas con su proveedor de `gsmSCF`
- **Pruebas:** Utilice `:termAttemptAuthorized` para pruebas más exhaustivas
- **Servicios Prepagos:** La clave de servicio `11_110` es un estándar común de la industria para llamadas terminantes prepagos
- **Manejo de Respaldo:** `defaultCallHandling: :continueCall` asegura que las llamadas continúen si el `gsmSCF` no está disponible

## Manejo de Suscriptores en Roaming

### Detección de VLR en el Hogar vs VLR en Roaming

Cuando el HLR recibe una solicitud de **SendRoutingInfo** (SRI), necesita determinar si el suscriptor está en un VLR "local" (dentro de su red) o en un VLR en roaming (visitando otra red). El comportamiento difiere según esta determinación:

Para referencia de parámetros de configuración, consulte [Perfiles de VLR en el Hogar en la Referencia de Configuración](#).

- **VLR en el Hogar:** Devolver respuesta SRI estándar con información de enrutamiento CAMEL
- **VLR en Roaming:** Enviar una solicitud de Proveer Número en Roaming (PRN) para obtener un MSRN, luego devolverlo en la respuesta SRI

#### Configuración

```
# Prefijos de VLR en el Hogar
# Lista de prefijos de dirección VLR que se consideran red "local"
# Si la dirección VLR del suscriptor comienza con uno de estos prefijos, usar respuesta SRI estándar
# De lo contrario, el suscriptor está en roaming y necesitamos enviar PRN para obtener MSRN
home_vlr_prefixes: ["55123"];
```

#### Ejemplo de Configuración:

```
# Un solo operador de red local
home_vlr_prefixes: ["55123"];

# Múltiples operadores de red locales (por ejemplo, diferentes regiones o subsidiarias)
home_vlr_prefixes: ["55123", "55124", "55125"];
```

#### Cómo Funciona

##### 1. Flujo de Suscriptor Local (Estándar)

Cuando la dirección VLR del suscriptor comienza con un prefijo local configurado:

```
GMSC - HLR: SendRoutingInfo (MSISDN: "1234567890")
HLR consulta la API backend para datos del suscriptor
HLR verifica la dirección VLR: "551234567"
HLR determina: VLR comienza con "55123" - Red local
HLR - GMSC: Respuesta SRI con información de enrutamiento CAMEL:
- IMSI
- Número de VLR: "551234567"
- Dirección gsmSCF (MSC): "551234501"
- Clave de servicio CAMEL: 11_110
- Punto de detección de disparador: termAttemptAuthorized
```

##### 2. Flujo de Suscriptor en Roaming (PRN Requerido)

Cuando la dirección VLR del suscriptor NO coincide con ningún prefijo local:

GMSC → HLR: SendRoutingInfo (MSISDN: "1234567890")  
HLR consulta la API backend para datos del suscriptor  
HLR verifica la dirección VLR: "49170123456"  
HLR determina: VLR no comienza con "555123" → Roaming  
HLR → MSC: ProvideRoamingNumber (PRN):  
- MSISDN: "1234567890"  
- IMSI: "999999876543210"  
- Número MSC: "49170123456"  
- Dirección GMSC: "5551234501"  
MSC → HLR: Respuesta PRN con MSRN: "49170999888777"  
HLR → GMSC: Respuesta SRI con información de enrutamiento:  
- IMSI  
- Número de VLR: "49170123456"  
- Número en Roaming (MSRN): "49170999888777"

## Diferencias en la Estructura de Respuesta

### Respuesta SRI de Suscriptor Local

```
{
  "imsi": "999999876543210",
  "extendedRoutingInfo": {
    "camelRoutingInfo": {
      "gsmcCAMELSubscriptionInfo": {
        "t-CSI": {
          "serviceKey": 11110,
          "gsmSCF-Address": "5551234501",
          "defaultCallHandling": "continueCall",
          "t-BsmTriggerDetectionPoint": "itemAttemptAuthorized"
        }
      }
    }
  },
  "subscriberInfo": {
    "locationInformation": {"vlr-number": "5551234567"},
    "subscriberState": {"notProvidedFromVLR": NULL}
  }
}
```

### Respuesta SRI de Suscriptor en Roaming

```
{
  "imsi": "999999876543210",
  "extendedRoutingInfo": {
    "routingInfo": {
      "roamingNumber": "49170999888777" # MSRN de PRN
    }
  },
  "subscriberInfo": {
    "locationInformation": {"vlr-number": "49170123456"},
    "subscriberState": {"notProvidedFromVLR": NULL}
  }
}
```

## Operación de Proveer Número en Roaming (PRN)

### Estructura de Solicitud PRN

La solicitud PRN enviada al MSC/VLR contiene:

Campo	Fuente	Descripción
MSISDN	Solicitud SRI	Número de teléfono del suscriptor
IMSI	API HLR	IMSI del suscriptor
Número MSC	API HLR	MSC que atiende al suscriptor en roaming (serving_msc)
Dirección GMSC	Solicitud SRI	GMSC que realiza la solicitud SRI original
Número de Referencia de Llamada	Estático	Identificador de referencia de llamada
Fases CAMEL Soportadas	Estático	Fases CAMEL soportadas por GMSC

### Manejo de Respuesta PRN

El HLR espera una respuesta PRN que contenga:

- **MSRN** (Número de Estación Móvil en Roaming): Un número temporal asignado por la red visitada para enrutar la llamada

### Manejo de Errores:

- Si PRN se agota → Devuelve error 27 (Suscriptor Ausente) en la respuesta SRI
- Si PRN falla → Devuelve error 27 (Suscriptor Ausente) en la respuesta SRI
- Si no se puede extraer MSRN → Devuelve error 27 (Suscriptor Ausente) en la respuesta SRI

## Ejemplos de Configuración

### Operador de Red Local Único

```
# Todas las direcciones VLR que comienzan con "555123" se consideran locales
home_vlr_prefixes: ["555123"],
```

- VLR 5551234567 → Local (respuesta CAMEL)
- VLR 5551235001 → Local (respuesta CAMEL)
- VLR 49170123456 → Roaming (PRN + respuesta MSRN)

### Operador Multi-Región

```
# Múltiples redes locales en diferentes regiones
home_vlr_prefixes: ["555123", "555124", "555125"],
```

- VLR 5551234567 → Local (región 1)
- VLR 5552341234 → Local (región 2)
- VLR 5553411111 → Local (región 3)
- VLR 44201234567 → Roaming (internacional)

### Configuración de Pruebas

Para probar la funcionalidad PRN, configure una lista vacía para tratar todos los VLRs como en roaming:

```
# Todos los VLRs se tratan como roaming (para probar el flujo PRN)
home_vlr_prefixes: [],
```

### Mejores Prácticas

- **Selección de Prefijos:** Utilice el prefijo único más corto que identifique los VLRs de su red (por ejemplo, código de país + código de red)
- **Múltiples Prefijos:** Incluya todos los prefijos de VLR en su red, incluidos diferentes regiones y subsidiarias
- **Acuerdos de Roaming:** Asegúrese de que PRN esté soportado adecuadamente por las redes asociadas en roaming
- **Pruebas:** Pruebe a fondo tanto los escenarios locales como los de roaming antes de la implementación en producción
- **Monitoreo:** Monitoree las tasas de tiempo de espera de PRN para identificar problemas de conectividad con socios de roaming

## Solución de Problemas

**Síntoma:** Todos los suscriptores tratados como roaming

- **Causa:** home\_vlr\_prefixes no configurado o los prefijos no coinciden con las direcciones VLR
- **Solución:** Verifique las direcciones VLR en su base de datos y actualice los prefijos según sea necesario

**Síntoma:** Solicitudes PRN que se agotan

- **Causa:** Problemas de conectividad de red con el MSC/VLR asociado en roaming
- **Solución:** Verifique el enrutamiento M3UA/SCCP a las direcciones MSC remotas

**Síntoma:** MSRN inválido en la respuesta SRI

- **Causa:** El formato de respuesta PRN del socio en roaming no coincide con la estructura esperada
- **Solución:** Revise los registros de respuesta PRN y ajuste extract\_msrn\_from\_prn/1 si es necesario

## Operaciones HLR

### Operaciones MAP Soportadas

- updateLocation (Código de operación 2) - Registrar ubicación VLR
- sendAuthenticationInfo (Código de operación 56) - Generar vectores de autenticación
- sendRoutingInfo (Código de operación 22) - Proporcionar MSRN para llamadas con soporte CAMEL
- sendRoutingInfoForSMS (Código de operación 45) - Proporcionar GT de MSC para SMS
- cancelLocation (Código de operación 3) - Desregistrar del antiguo VLR
- insertSubscriberData (Código de operación 7) - Empujar perfil de suscriptor

### Mapeo de Campos de Respuesta

Esta sección detalla de dónde proviene cada campo en las respuestas HLR.

#### Respuesta SendRoutingInfo (SRI)

**Propósito:** Proporciona información de enrutamiento para llamadas entrantes a un suscriptor.

El HLR proporciona dos tipos de respuesta diferentes según si el suscriptor está en un VLR local o en roaming:

#### Respuesta de Suscriptor Local (Enrutamiento CAMEL)

Utilizado cuando la dirección VLR del suscriptor comienza con un valor configurado en home\_vlr\_prefixes.

#### Estructura de Respuesta:

Campo	Fuente	Descripción	Ejemplo
IMSI	API OmniHSS	IMSI del suscriptor de la base de datos OmniHSS	"999999876543210"
Número de VLR	runtime.exe	Identificador de servicio CAMEL (camel_service_key)	"5551234567"
Estado del Suscriptor	Estático	Siempre notProvidedFromVLR	:notProvidedFromVLR
extendedRoutingInfo	-	Tipo: camelRoutingInfo	-
Dirección gsmSCF	API OmniHSS	MSC que atiende al suscriptor (circuit_session_assigned_msc)	"5551234501"
Clave de Servicio	runtime.exe	Identificador de servicio CAMEL (camel_service_key)	11_110
Punto de Detección de Disparador	runtime.exe	Cuándo activar CAMEL (camel_trigger_detection_point)	:termAttemptAuthorized
Manejo de Capacidades CAMEL	Estático	Nivel de soporte de fase CAMEL	3
Manejo de Llamadas por Defecto	Estático	Respaldo si gsmSCF no está disponible	:continueCall

#### Respuesta de Suscriptor en Roaming (Enrutamiento MSRN)

Utilizado cuando la dirección VLR del suscriptor NO coincide con ningún valor configurado en home\_vlr\_prefixes.

#### Estructura de Respuesta:

Campo	Fuente	Descripción	Ejemplo
IMSI	API OmniHSS	IMSI del suscriptor de la base de datos OmniHSS	"999999876543210"
Número de VLR	API OmniHSS	VLR actual que atiende al suscriptor (circuit_session_assigned_vlr)	"49170123456"
Estado del Suscriptor	Estático	Siempre notProvidedFromVLR	:notProvidedFromVLR
extendedRoutingInfo	-	Tipo: routingInfo	-
Número en Roaming (MSRN)	Respuesta PRN MSRN obtenido de la solicitud ProvideRoamingNumber		"49170999888777"

#### Lógica de Decisión de Enrutamiento:

- OmniSS7 recibe solicitud SendRoutingInfo
- OmniSS7 consulta datos del suscriptor desde la API OmniHSS
- OmniSS7 verifica la dirección VLR contra home\_vlr\_prefixes:

Si VLR comienza con prefijo local:  
- Devolver información de enrutamiento CAMEL (flujo de suscriptor local)

Si VLR NO coincide con ningún prefijo local:  
- Enviar ProvideRoamingNumber (PRN) al MSC  
- Extraer MSRN de la respuesta PRN  
- Devolver información de enrutamiento con MSRN (flujo de suscriptor en roaming)

#### Flujo de Datos:

- OmniSS7 consulta a OmniHSS por información del suscriptor
- OmniHSS devuelve IMSI, ubicación actual VLR/MSR y estado del suscriptor
- OmniSS7 utiliza estos datos para construir la respuesta MAP

#### Requisitos de Configuración:

```
# En runtime.exe
home_vlr_prefixes: ["555123"], # Lista de prefijos de VLR locales
```

#### Respuestas de Error:

- Si serving\_vlr y serving\_msc son null: Devuelve error 27 (Suscriptor Ausente)
- Si no se encuentra al suscriptor: Devuelve error 1 (Suscriptor Desconocido)
- Si la solicitud PRN se agota (caso de roaming): Devuelve error 27 (Suscriptor Ausente)
- Si la respuesta PRN es inválida (caso de roaming): Devuelve error 27 (Suscriptor Ausente)

#### Respuesta UpdateLocation con InsertSubscriberData

**Propósito:** Registra al suscriptor en el nuevo VLR y provisiona los datos del suscriptor.

#### Respuesta UpdateLocation FIN

Campo	Fuente	Descripción	Ejemplo
Número HLR	runtime.exe	Título Global de este HLR (hlr_service_center_gt_address)	"5551234568"
Tipo de Mensaje TCAP	Estático	Respuesta final después de todos los ISDs	FIN

#### InsertSubscriberData #1 (Datos Básicos del Suscriptor)

Campo	Fuente	Descripción	Ejemplo
IMSI	Solicitud	De la solicitud UpdateLocation	"999999876543210"
MSISDN	API OmniHSS	Número de teléfono del suscriptor de OmniHSS	"555123456"
Categoría	Estático	Categoría del suscriptor	"n" (0x0A)
Estado del Suscriptor	Estático	Estado del servicio	:serviceGranted
Lista de Servicios de Portadora	Estático	Servicios de portadora soportados	<<1;31>>
Lista de Teleservicios	Estático	Teleservicios soportados	<<1;17>>, "1", ""
Modo de Acceso a la Red	runtime.exe	Acceso a paquete/circuito (isd_network_access_mode):packetAndCircuit	

#### InsertSubscriberData #2 (Servicios Suplementarios) - Opcional

Campo	Fuente	Descripción	Controlado Por
SS Provisionados	Estático	Datos de servicios suplementarios	isd_send_ss_data: true
Desvío de Llamadas	Estático	Configuraciones de desvío (incondicional, ocupado, sin respuesta, no alcanzable)	Config habilitado
Espera de Llamadas	Estático	Estado del servicio de espera de llamadas	Config habilitado
Servicio de Múltiples Partes	Estático	Soporte para conferencias	Config habilitado

#### ISD #2 incluye:

- Desvío de llamadas incondicional (código SS 21)
- Desvío de llamadas en ocupado (código SS 41)
- Desvío de llamadas en sin respuesta (código SS 42)
- Desvío de llamadas en no alcanzable (código SS 62)
- Espera de llamadas (código SS 43)
- Servicio de múltiples partes (código SS 51)
- Servicios CLIP/CLIR

#### InsertSubscriberData #3 (Barring de Llamadas) - Opcional

Campo	Fuente	Descripción	Controlado Por
Información de Barring de Llamadas	Estático	Configuraciones de barring de llamadas	isd_send_call_barring: true
BAOC	Estático	Barring de Todas las Llamadas Salientes (código SS 146)	Config habilitado
BOIC	Estático	Barring de Llamadas Internacionales Salientes (código SS 147)	Config habilitado
Datos de Restricción de Acceso	Estático	Restricciones de acceso a la red	Config habilitado

#### Control de Secuencia ISD:

- ISD #1: Siempre enviado - Contiene datos esenciales del suscriptor
- ISD #2: Enviado solo si isd\_send\_ss\_data: true en runtime.exe
- ISD #3: Enviado solo si isd\_send\_call\_barring: true en runtime.exe

#### Respuesta SendRoutingInfoForSM (SRI-for-SM)

**Propósito:** Proporciona información de enrutamiento MSC/SMSC para la entrega de SMS. Cuando un SMSC necesita entregar un SMS a un suscriptor, envía una solicitud SRI-for-SM al HLR para determinar dónde enrutar el mensaje.

#### Estructura de Respuesta:

Campo	Fuente	Descripción	Cómo Generado	Ejemplo
IMSI	Calculado	IMSI sintético derivado de MSISDN	PLMN_PREFIX + zero_padded_MSISDN	"001001555123456"
Número de Nodo de Red	runtime.exe	Dirección GT SMSC para enrutamiento de SMS	smc_service_center_gt_address	"5551234567"

#### Parámetros de Configuración (desde runtime.exe):

```
# Dirección del Centro de Servicio GT (devuelta en respuestas SRI-for-SM)
# Esto indica al SMSC solicitante dónde enviar mensajes MT-ForwardSM
smc_service_center_gt_address: "5551234567", # Requerido

# Configuración de Mapeo MSISDN -> IMSI
# Prefijo PLMN: MCC (001 = Red de Prueba) + MNC (01 = Operador de Prueba)
hlr_imsi_plmn_prefix: "001001", # Único parámetro de configuración necesario!
```

#### Mapeo MSISDN -> IMSI

##### Parámetros de Configuración:

Estos parámetros controlan cómo OmniSS7 genera IMSIs sintéticos a partir de MSISDNs para respuestas SRI-for-SM:

- hlr\_imsi\_plmn\_prefix: El prefijo MCC+MNC que se utilizará al construir IMSIs sintéticos (por ejemplo, "50557" para MCC=505, MNC=57)
- hlr\_imsi\_nsn\_country\_code: Código de país que se antepone al hacer el mapeo inverso IMSI->MSISDN (por ejemplo, "61" para Australia, "1" para EE. UU./Canadá)
- hlr\_imsi\_nsn\_offset: Posición de carácter donde comienza el Número Nacional de Suscriptor (NSN) dentro del MSISDN (típicamente 0 si el MSISDN no incluye código de país, o longitud del código de país si lo incluye)
- hlr\_imsi\_nsn\_length: Número de dígitos a extraer del MSISDN como NSN

Para detalles adicionales de configuración, consulte [Mapeo MSISDN -> IMSI en la Referencia de Configuración](#).

#### ¿Por qué se Necesita el Mapeo MSISDN a IMSI?

El protocolo MAP para [SendRoutingInfoForSM \(SRI-for-SM\)](#) requiere que el HLR devuelva un IMSI (Identidad Internacional de Suscriptor Móvil) en su respuesta. Sin embargo, el SMSC solicitante solo conoce el MSISDN (número de teléfono).

En una red tradicional:

- El SMSC envía SRI-for-SM con el MSISDN de destino (por ejemplo, "5551234567")
- El HLR debe buscar al suscriptor en su base de datos para encontrar su IMSI
- El HLR devuelve el IMSI en la respuesta SRI-for-SM
- El SMSC luego utiliza este IMSI al enviar MT-ForwardSM al MSC/VLR

#### Enfoque de OmniSS7 - IMSIs Sintéticos:

En lugar de mantener una base de datos completa de suscriptores con mapeos MSISDN a IMSI, OmniSS7 utiliza un esquema de codificación simple para **calcular** IMSIs sintéticos directamente a partir del MSISDN. Este enfoque proporciona dos beneficios clave:

1. **Privacidad:** Los IMSIs reales de los suscriptores almacenados en la base de datos HLR nunca se exponen en las respuestas SRI-for-SM enviadas a través de la red SS7
2. **Simplicidad:** No es necesario consultar la base de datos HLR para búsquedas de IMSI durante las operaciones SRI-for-SM: el IMSI se calcula sobre la marcha a partir del MSISDN

#### Cómo Funciona:

Los MSISDNs se codifican directamente en la porción del suscriptor del IMSI (los dígitos después de MCC+MNC):

```
IMSI = PLMN_PREFIX + zero_padded_MSISDN
```

Donde:

- **PLMN\_PREFIX:** MCC + MNC (por ejemplo, "001001" para Red de Prueba)
- **MSISDN:** Todos los dígitos numéricos del número de teléfono
- **Zero Padding:** Rellenado a la izquierda con ceros para llenar el IMSI a exactamente 15 dígitos

#### Ejemplo Paso a Paso:

```
# Configuración
plmn_prefix = "001001" # MCC 001 + MNC 01

# Entrada: MSISDN de la solicitud SRI-for-SM (decodificado TBCC)
msisdn = "555123456" # 9 dígitos

# Paso 1: Calcular el espacio disponible para el número de suscriptor
subscriber_digits = 15 - String.length("001001") # = 9 dígitos

# Paso 2: Rellenar el MSISDN con ceros a la izquierda para llenar la porción del suscriptor
padded_msisdn = String.pad_left("555123456", 9, "0") # = "555123456" (sin relleno necesario)

# Paso 3: Concatenar prefijo PLMN + MSISDN relleno
imsi = "001001" <-> "555123456" # = "001001555123456" (exactamente 15 dígitos)
```

#### Ejemplos Completos:

MSISDN de Entrada	Prefijo PLMN	Porción de Suscriptor Disponibles	MSISDN Rellenado	IMSI Final	Notas
"555123456"	"001001" (6)	9	"555123456"	"001001555123456"	Ajuste exacto, sin relleno
"99"	"001001" (6)	9	"000000099"	"001001000000999"	Rellenado a la izquierda con ceros
"999999999"	"001001" (6)	9	"999999999"	"001001999999999"	Ajuste exacto
"91123456789"	"001001" (6)	9	"555123456"	"001001555123456"	Demasiado largo, se mantienen los 9 dígitos más a la derecha

#### Manejo de Casos Especiales:

- **MSISDNs Cortos:** Rellenados a la izquierda con ceros (por ejemplo, "99" → "000000099")
- **MSISDNs Largos:** Se mantienen los dígitos más a la derecha, se recortan los más a la izquierda (por ejemplo, "91123456789" → "555123456")
- **Longitud de IMSI:** Siempre exactamente 15 dígitos

#### Manejo de Mapeo Inverso (IMSI → MSISDN):

El SMSc puede revertir este mapeo para convertir IMSIs de nuevo a MSISDNs:

```
# Entrada: IMSI de la respuesta SRI-for-SM
imsi = "001001555123456"

# Paso 1: Eliminar prefijo PLMN
plmn_prefix = "001001"
subscriber_portion = String.slice(imsi, 6, 9) # = "555123456"

# Paso 2: Eliminar ceros a la izquierda para obtener el MSISDN real
msisdn = String.replace_leading(subscriber_portion, "0", "") # = "555123456"
```

#### Ejemplos de Mapeo Inverso:

IMSI de Entrada	Prefijo PLMN	Porción de Suscriptor	Eliminar Ceros a la Izquierda	MSISDN Final
"001001555123456"	"001001"	"555123456"	"555123456"	"555123456"
"001001000000999"	"001001"	"000000099"	"99"	"99"
"001001999999999"	"001001"	"999999999"	"999999999"	"999999999"

#### Propiedades de Este Mapeo:

- **Determinístico:** El mismo MSISDN siempre produce el mismo IMSI
- **Reversible:** Se puede convertir de nuevo de IMSI a MSISDN
- **Configuración Mínima:** Solo requiere `hlr_imsi_plmn_prefix`
- **Protección de Privacidad:** Los IMSIs reales nunca se exponen
- **Sin Consulta a la Base de Datos:** Cálculo rápido, no se necesitan llamadas a la API
- **Siempre 15 Dígitos:** El IMSI siempre tiene exactamente 15 dígitos

#### Manejo de Entrada MSISDN:

Cuando el HLR recibe una solicitud SRI-for-SM, el MSISDN pasa por decodificación TBCC:

1. **Decodificación TBCC:** Convertir TBCC binario a cadena (puede incluir prefijo TON/NPI como "91")
2. **Extraer Dígitos:** Mantener solo dígitos numéricos, eliminar cualquier carácter no numérico
3. **Normalizar:** Si es más largo que el espacio disponible, tomar los dígitos más a la derecha; si es más corto, rellenar con ceros a la izquierda
4. **Codificar:** Concatenar prefijo PLMN + MSISDN normalizado

#### Consideraciones de Seguridad:

Los IMSIs sintéticos devueltos en las respuestas SRI-for-SM son puramente para fines de enrutamiento. NO son los IMSIs reales almacenados en la base de datos de suscriptores HLR. Esto proporciona una capa adicional de protección de privacidad, ya que los IMSIs reales de los suscriptores solo se exponen cuando es absolutamente necesario (por ejemplo, durante `UpdateLocation` o `SendAuthenticationInfo` que requieren vectores de autenticación reales).

#### Flujo de Respuesta:

1. SMSc → HLR: Solicitud SRI-for-SM  
- MSISDN (TBCC): "91123456789" (incluye TON/NPI)
2. Procesamiento HLR:  
- Decodificación TBCC: "91123456789"  
- Extraer dígitos: "91123456789" (11 dígitos)  
- Ajustar a 9 dígitos: "555123456" (9 más a la derecha)  
- Agregar PLMN: "001001" + "555123456" = "001001555123456"  
- Obtener GT SMSc de la configuración: "5551234567"
3. HLR → SMSc: Respuesta SRI-for-SM  
- IMSI: "001001555123456" (sintético, siempre 15 dígitos)  
- Número de Nodo de Red: "5551234567" (dónde enviar MT-ForwardSM)
4. SMSc envía MT-ForwardSM a "5551234567" con IMSI "001001555123456"

#### Configuración:

Los siguientes parámetros se utilizan en runtime.exs:

```
# Prefijo PLMN: MCC (001 = Red de Prueba) + MNC (01 = Operador de Prueba)
hlr_imsi_plmn_prefix: "001001",

# Extracción de NSN (si los MSISDN incluyen código de país)
hlr_msisdn_country_code: "1", # Utilizado para mapeo inverso (IMSI-MSISDN)
hlr_msisdn_nsn_offset: 1, # Saltar 1 dígito del código de país
hlr_msisdn_nsn_length: 10 # Extraer 10 dígitos de NSN
```

#### Configuración de Extracción de NSN:

Si sus MSISDN incluyen el código de país (por ejemplo, "68988000088" en lugar de solo "88000088"), debe configurar la extracción de NSN:

- `hlr_msisdn_nsn_offset`: Posición donde comienza el NSN (típicamente la longitud de su código de país)
- `hlr_msisdn_nsn_length`: Número de dígitos en el NSN

#### Ejemplos:

Ejemplo	Código de País	MSISDN de Ejemplo	nsn_offset	nsn_length	NSN Extraído
Código de País de 1 dígito "9"	"9551234567"	1	10	"5551234567"	
Código de País de 2 dígitos "99"	"99412345678"	2	9	"412345678"	
Código de País de 3 dígitos "999"	"99988000088"	3	8	"88000088"	

#### Cómo Funciona:

1. **MSISDN → IMSI:** Extraer NSN del MSISDN, rellenar con ceros a la izquierda, concatenar con prefijo PLMN

```
MSISDN: "99988000088"
NSN: String.slice("99988000088", 3, 8) = "88000088"
MSISDN Rellenado: "088000088" (9 dígitos)
IMSI: "547050" + "088000088" = "547050088000088"
```

2. **IMSI → MSISDN:** Eliminar prefijo PLMN, eliminar ceros a la izquierda, anteponer código de país

```
IMSI: "547050088000088"
Porción del Suscriptor: "088000088"
Eliminar ceros: "88000088"
MSISDN: "+999" + "88000088" = "+99988000088"
```

## Requisitos de API: Ninguno - SRI-for-SM utiliza valores calculados y configuración solamente. No se requieren llamadas a la API backend.

#### Resumen de Fuentes de Campo

Tipo de Fuente	Descripción	Ejemplos
API OmnIHSS	Datos dinámicos de la base de datos de suscriptores OmnIHSS IMSI, MSISDN, VLR/MSV que atiende desde <code>circuit_session</code>	
runtime.exs	Parámetros de configuración de OmnIHSS7	<code>smc_service_center_gt_address</code> , <code>camel_service_key</code> , <code>isd_network_access_mode</code>
Estático	Valores codificados en el generador de respuestas	Estado del suscriptor, servicios de portadora, códigos SS
Solicitud	Campos extraídos de la solicitud MAP entrante	IMSI de <code>UpdateLocation</code> , MSISDN de SRI
Calculado	Valores derivados utilizando lógica	IMSI sintético en SRI-for-SM ( <code>hlr_imsi_prefix</code> + NSN)

#### Dependencias de Configuración

Requerido en runtime.exs:

- hlr\_service\_center\_gt\_address - Utilizado en respuestas UpdateLocation
- smsc\_service\_center\_gt\_address - Utilizado en respuestas SRI-for-SM (dónde se deben enviar MF-ForwardSM)

**Opcional en runtime.exe** (con valores predeterminados):

- camel\_service\_key - Predeterminado: 11\_118
- camel\_trigger\_detection\_point - Predeterminado: :termAttemptAuthorized
- isd\_network\_access\_mode - Predeterminado: :packetAndCircuit
- isd\_send\_ss\_data - Predeterminado: true
- isd\_send\_call\_barring - Predeterminado: true
- hlr\_imsi\_plmn\_prefix - Predeterminado: "001001" (prefijo PLMN para mapeo MSISDN->IMSI)

**Requerido desde OmniHSS:**

OmniHSS debe proporcionar puntos finales de API REST para:

- Búsqueda de suscriptores por IMSI y MSISDN
- Actualizaciones de ubicación de sesiones de circuito (asignación VLR/MSC)
- Generación de vectores de autenticación
- Consultas sobre el estado del suscriptor y el perfil de servicio

---

## Documentación Relacionada

**Documentación de OmniSS7:**

- [= Volver a la Documentación Principal](#)
- [Guía de Características Comunes](#)
- [Guía del Cliente MAP](#)
- [Referencia Técnica](#)
- [Referencia de Configuración](#)

**Documentación de OmniHSS:** Para gestión de suscriptores, provisión, configuración de autenticación y operaciones administrativas, consulte la **documentación del producto OmniHSS**. OmniHSS contiene toda la lógica de la base de datos de suscriptores, algoritmos de autenticación, reglas de provisión de servicios y capacidades de gestión Multi-IMSI.

---

**OmniSS7** por Omnitouch Network Services

# Guía de Configuración del Cliente MAP

[— Volver a la Documentación Principal](#)

Esta guía proporciona una configuración detallada para usar OmniSS7 como un **Cliente MAP** para enviar solicitudes del protocolo MAP a elementos de red.

## Tabla de Contenidos

- [1. ¿Qué es el Modo Cliente MAP?](#)
- [2. Habilitando el Modo Cliente MAP](#)
- [3. Operaciones MAP Disponibles](#)
- [4. Enviando Solicitudes a través de la API](#)
- [5. Métricas y Monitoreo](#)
- [6. Solución de Problemas](#)

## ¿Qué es el Modo Cliente MAP?

Modo Cliente MAP permite que OmniSS7 se conecte como un **Proceso de Servidor de Aplicaciones (ASP)** a un par M3UA (STP o SGP) y envíe/reciba mensajes MAP (**Parte de Aplicación Móvil**) para servicios como:

- **Consultas HLR:** SRI (Enviar Información de Enrutamiento), SRI-for-SM, Información de Autenticación
- **Actualizaciones de Ubicación:** Actualizar Ubicación, Cancelar Ubicación
- **Gestión de Suscriptores:** Prover Número de Roaming (PRN), Insertar Datos del Suscriptor

## Arquitectura de Red

## Habilitando el Modo Cliente MAP

Edita config/runtime.exe y configura los ajustes del cliente MAP. Para una referencia completa de configuración, consulta [Parámetros de Conexión M3UA en la Referencia de Configuración](#).

### Configuración Básica

```
config :omniss7,
# Habilitar modo Cliente MAP
map_client_enabled: true,

# Conexión M3UA para el Cliente MAP (se conecta como ASP a STP/SGP remoto)
map_client_m3ua: %{
  mode: "ASP", # Modo M3UA: "ASP" (cliente) o "SGP" (servidor)
  callback: (MapClient, :handle_payload, []), # Callback para mensajes entrantes
  process_name: :map_client_esp, # Nombre del proceso registrado
  local_ip: {10, 0, 0, 100}, # Dirección IP local
  local_port: 2905, # Puerto SCTP local
  remote_ip: {10, 0, 0, 1}, # IP STP/SGP remoto
  remote_port: 2905, # Puerto STP/SGP remoto
  routing_context: 1 # Contexto de enrutamiento M3UA
}
```

### Ejemplo de Configuración para Producción

```
config :omniss7,
# Habilitar Cliente MAP para producción
map_client_enabled: true,

# Conexión M3UA de producción
map_client_m3ua: %{
  mode: "ASP",
  callback: (MapClient, :handle_payload, []),
  process_name: :map_client_esp,
  local_ip: {10, 0, 0, 100},
  local_port: 2905,
  remote_ip: {10, 0, 0, 1}, # IP STP de producción
  remote_port: 2905,
  routing_context: 1
}

config :control_panel,
web: %{
  listen_ip: "0.0.0.0",
  port: 443,
  hostname: "ss7-gateway.example.com",
  enable_tls: true,
  tls_cert: "/etc/ssl/certs/gateway.crt",
  tls_key: "/etc/ssl/private/gateway.key"
}
```

## Operaciones MAP Disponibles

### MAP Tester API 1.0.0 OAS 3.0

fwaggar.json

#### default

POST	/api/MT-forwardSM	MT-forwardSM MAP Request (For sending SMS for delivery by remote MSC/SMSC)
POST	/api/deliverPDU	Utility: Build SMS-DELIVER TPDU from originating address + GSM7
POST	/api/forwardSM	forwardSM MAP Request
POST	/api/prn	ProvideRoamingNumber (PRN) MAP Request
POST	/api/send-auth-info	SendAuthenticationInfo MAP Request
POST	/api/sendSM	Utility: Perform SRI-for-SM + MT-forwardSM from MSISDN and GSM7
POST	/api/sri	SendRoutingInfo MAP Request
POST	/api/sri-for-sm	SendRoutingInfoForSM MAP Request
POST	/api/updateLocation	UpdateLocation MAP Request
GET	/metrics	Prometheus metrics
GET	/swagger.json	OpenAPI spec

#### Schemas

AuthInfoRequest >

ErrorResponse >

ForwardSMRequest >

PRNRequest >

SMSDeliverPDURequest >

SMSDeliverPDUResponse >

SRIForSMRequest >

SRIRequest >

SRIResponse >

SendsMRRequest >

UpdateLocationRequest >

### 1. Enviar Información de Enrutamiento para SM (SRI-for-SM)

Consulta el HLR para determinar el MSC que atiende la entrega de SMS. Para información detallada sobre cómo el HLR procesa las solicitudes SRI-for-SM, consulta [SRI-for-SM en la Guía HLR](#).

**Endpoint de la API:** POST /api/sri-for-sm

**Solicitud:**

```
{
  "msisdn": "447712345678",
  "serviceCenter": "447999123456"
}
```

**Respuesta:**

```
{
  "result": {
    "smsi": "234509876543210",
    "locationInfoWithLMSI": {
      "networkNode-Number": "447999555111"
    }
  }
}
```

**Ejemplo de cURL:**

```
curl -X POST http://localhost/api/sri-for-sm \
-H "Content-Type: application/json" \
-d '{
  "msisdn": "447712345678",
  "serviceCenter": "447999123456"
}'
```

### 2. Enviar Información de Enrutamiento (SRI)

Consulta el HLR para obtener información de enrutamiento de llamadas de voz.

**Endpoint de la API:** POST /api/sri

**Solicitud:**

```
{
  "msisdn": "447712345678",
  "gmsc": "447999123456"
}
```

**Respuesta:**

```
{
  "result": {
    "smsi": "234509876543210",
    "extendedRoutingInfo": {
      "routingInfo": {
        "roamingNumber": "447999555222"
      }
    }
  }
}
```

### 3. Probeer Número de Roaming (PRN)

Solicita un número de roaming temporal (MSRN) del MSC que atiende.

**Endpoint de la API:** POST /api/prn

**Solicitud:**

```
{
  "msisdn": "447712345678",
  "gmsc": "447999123456",
  "msc_number": "44799955111",
  "imsi": "234509876543210"
}
```

#### 4. Enviar Información de Autenticación

Solicita vectores de autenticación del HLR para la autenticación del suscriptor.

**Endpoint de la API:** POST /api/send-auth-info

**Solicitud:**

```
{
  "imsi": "234509876543210",
  "vectors": 5
}
```

**Respuesta:**

```
{
  "result": {
    "authenticationSetList": [
      {
        "rand": "0123456789ABCDEF0123456789ABCDEF",
        "xres": "ABCDEF0123456789",
        "ck": "0123456789ABCDEF0123456789ABCDEF",
        "ik": "FEDCBA9876543210FEDCBA9876543210",
        "autn": "0123456789ABCDEF0123456789ABCDEF"
      }
    ]
  }
}
```

#### 5. Actualizar Ubicación

Registra la ubicación actual de un suscriptor con el HLR. Para información detallada sobre el procesamiento de UpdateLocation y las secuencias de InsertSubscriberData, consulta [Actualizaciones de Ubicación en la Guía HLR](#).

**Endpoint de la API:** POST /api/updateLocation

**Solicitud:**

```
{
  "imsi": "234509876543210",
  "vlr": "44799955111"
}
```

#### Resumen de Operaciones MAP

### Enviando Solicitudes a través de la API

#### Usando Swagger UI

La interfaz Swagger UI proporciona una interfaz interactiva para enviar solicitudes SS7.

**Acceso a Swagger UI:**

1. Navega a <http://your-server/swagger>
2. Explora los endpoints de la API disponibles
3. Haz clic en cualquier endpoint para expandir sus detalles

**Enviando una Solicitud:**

1. Haz clic en el endpoint que desees usar (por ejemplo, /api/sri-for-sm)
2. Haz clic en el botón "Try it out"
3. Completa los parámetros requeridos en el cuerpo de la solicitud
4. Haz clic en "Execute"
5. Visualiza la respuesta a continuación

**Códigos de Respuesta de la API**

- **200** - Éxito, resultado devuelto en el cuerpo de la respuesta
- **400** - Solicitud incorrecta, parámetros inválidos
- **504** - Tiempo de Espera del Gateway, sin respuesta de la red SS7 dentro de 10 segundos

### Métricas del Cliente MAP

**Métricas Disponibles**

**Métricas de Solicitud:**

- map\_requests\_total - Número total de solicitudes MAP enviadas
  - Etiquetas: operation (valores: sri, sri\_for\_sm, prn, authentication\_info, etc.)
- map\_request\_errors\_total - Número total de errores en solicitudes MAP
  - Etiquetas: operation
- map\_request\_duration\_milliseconds - Histograma de duraciones de solicitudes MAP
  - Etiquetas: operation
- map\_pending\_requests - Número actual de solicitudes MAP pendientes (gauge)

**Ejemplos de Consultas Prometheus**

```
# Total de solicitudes SRI-for-SM en la última hora
increase(map_requests_total{operation="sri_for_sm"}[1h])

# Tiempo de respuesta promedio para solicitudes SRI
rate(map_request_duration_milliseconds_sum{operation="sri"}[5m]) /
rate(map_request_duration_milliseconds_count{operation="sri"}[5m])

# Tasa de errores para todas las operaciones MAP
sum(rate(map_request_errors_total[5m])) by (operation)

# Solicitudes pendientes actuales
map_pending_requests
```

### Solución de Problemas del Cliente MAP

**Problema: Tiempo de Espera en Solicitudes**

**Síntomas:**

- La API devuelve 504 Gateway Timeout
- Sin respuesta del HLR/MSC

**Verificaciones:**

1. Verifica que la conexión M3UA esté ACTIVA:  
# En la consola IEX  
:sys.get\_state(:map\_client\_osp)
2. Verifica la conectividad de red al STP
3. Verifica el contexto de enrutamiento y la dirección SCCP
4. Revisa los registros en busca de errores SCCP

**Problema: Errores SCCP**

**Síntomas:**

- La API devuelve respuestas de error SCCP
- Los registros muestran mensajes de "servicio unitdata SCCP"

**Códigos de Error SCCP Comunes:**

- **Sin Traducción:** Título Global no encontrado en la tabla de enrutamiento del STP
- **Fallo de Subsistema:** Subsistema de destino (HLR SSN 6) no está disponible
- **Fallo de Red:** Congestión o fallo de red

**Soluciones:**

- Contacta al administrador del STP para verificar la configuración de enrutamiento
- Verifica que el Título Global de destino sea accesible
- Verifica si el subsistema de destino está operativo

---

## Documentación Relacionada

- [- Volver a la Documentación Principal](#)
- [Guía de Funciones Comunes](#) - Interfaz Web, API, Monitoreo
- [Guía SIP](#) - Configuración de enrutamiento
- [Guía del Centro SMS](#) - Entrega de SMS
- [Referencia Técnica](#) - Especificaciones del protocolo

---

OmniSS7 por Omnitouch Network Services

# Guía de Configuración del Centro de SMS (SMSc)

[← Volver a la Documentación Principal](#)

Esta guía proporciona una configuración detallada para usar OmniSS7 como un **Centro de SMS (SMSc)** en el frontend con **OmniMessage** como la plataforma de almacenamiento y entrega de mensajes en el backend.

## Integración de OmniMessage

El modo SMSc de OmniSS7 funciona como un **frontend de señalización SS7** que se conecta con **OmniMessage**, una plataforma de SMS de calidad de operador. Esta arquitectura separa las preocupaciones:

- **OmniSS7 (Frontend SMSc):** Maneja toda la señalización del protocolo SS7/MAP, enrutamiento SCCP y comunicación de red
- **OmniMessage (Backend SMS):** Gestiona el almacenamiento de mensajes, la cola, la lógica de reintentos, el seguimiento de entregas y las decisiones de enrutamiento

### ¿Por qué OmniMessage?

OmniMessage proporciona capacidades de mensajería SMS de calidad de operador con características que incluyen:

- **Gestión de Cola de Mensajes:** Almacenamiento persistente con lógica de reintentos configurable y encolado por prioridad
- **Seguimiento de Entregas:** Estado de entrega en tiempo real, informes de entrega (DLR) y seguimiento de razones de fallo
- **Soporte Multi-SMSc:** Múltiples instancias de frontend pueden conectarse a un solo backend de OmniMessage para balanceo de carga y redundancia
- **Inteligencia de Enrutamiento:** Reglas de enrutamiento avanzadas basadas en destino, remitente, contenido del mensaje y hora del día
- **Limitación de Tasa:** Controles de TPS (transacciones por segundo) por ruta para prevenir congestión en la red
- **Diseño API-Primero:** API HTTP RESTful para integración con sistemas de facturación, portales de clientes y aplicaciones de terceros
- **Análítica y Reportes:** Estadísticas de volumen de mensajes, tasas de éxito de entrega y métricas de rendimiento

Todos los datos de mensajes, estado de entrega y configuraciones de enrutamiento se almacenan y gestionan en OmniMessage. OmniSS7 consulta a OmniMessage a través de llamadas API HTTPS para recuperar mensajes pendientes, actualizar el estado de entrega y registrarse como un frontend activo.

**Importante:** El modo SMSc de OmniSS7 es un **frontend de señalización solamente**. Toda la lógica de enrutamiento de mensajes, gestión de colas, algoritmos de reintentos, seguimiento de entregas y reglas de negocio son manejadas por OmniMessage. Esta guía cubre la configuración del protocolo SS7/MAP en OmniSS7. Para información sobre enrutamiento de mensajes, configuración de colas, informes de entrega, limitación de tasa y analítica, **consulte la documentación de OmniMessage**.

## Tabla de Contenidos

1. [Integración de OmniMessage](#)
  2. [¿Qué es el Modo de Centro de SMS?](#)
  3. [Habilitando el Modo SMSc](#)
  4. [Configuración de API HTTP](#)
  5. [Flujos de Mensajes SMS](#)
  6. [Prevención de Bucles](#)
  7. [Seguimiento de Suscriptores SMSc](#)
  8. [Configuración de Auto-Flushing](#)
  9. [Métricas y Monitoreo](#)
  10. [Resolución de Problemas](#)
-

# ¿Qué es el Modo de Centro de SMS?

**Nota:** Esta sección cubre la configuración de señalización SS7 de OmniSS7 solamente. Para reglas de enrutamiento de mensajes, gestión de colas, seguimiento de entregas y configuración de lógica de negocio, consulte la **documentación del producto OmniMessage**.

**Modo de Centro de SMS** permite a OmniSS7 funcionar como un SMSc para:

- **Entrega MT-SMS:** Entrega de SMS terminados en móviles a suscriptores
- **Manejo de MO-SMS:** Recepción y enrutamiento de SMS originados en móviles
- **Encolado de Mensajes:** Cola de mensajes respaldada por base de datos con lógica de reintentos
- **Auto-Flushing:** Entrega automática de SMS desde la cola
- **Informes de Entrega:** Seguimiento del estado de entrega de mensajes

## Arquitectura del Centro de SMS

---

### Habilitando el Modo SMSc

OmniSS7 puede operar en diferentes modos. Para usarlo como un SMSc, necesita habilitar el modo SMSc en la configuración.

#### Cambiando al Modo SMSc

El config/runtime.exs de OmniSS7 contiene tres modos operativos preconfigurados. Para habilitar el modo SMSc:

1. **Abrir** config/runtime.exs
2. **Encontrar** las tres secciones de configuración (líneas 53-204):
  - Configuración 1: Modo STP (líneas 53-95)
  - Configuración 2: Modo HLR (líneas 97-142)
  - Configuración 3: Modo SMSc (líneas 144-204)
3. **Comentar** cualquier otra configuración activa (agregar # a cada línea)
4. **Descomentar** la configuración SMSc (eliminar # de las líneas 144-204)
5. **Personalizar** los parámetros de configuración según sea necesario
6. **Reiniciar** la aplicación: `iex -S mix`

#### Configuración del Modo SMSc

La configuración completa del SMSc se ve así:

```
config :omniss7,  
  # Flags de modo - Habilitar características de STP + SMSc  
  # Nota: map_client_enabled es verdadero porque SMSc necesita capacidades de enrutamiento  
  map_client_enabled: true,  
  hlr_mode_enabled: false,  
  smsc_mode_enabled: true,  
  
  # Configuración de API del Backend de OmniMessage  
  smsc_api_base_url: "https://10.179.3.219:8443",  
  # Identificación del SMSc para registro con el backend  
  smsc_name: "ipsmgw",  
  # Dirección GT del Centro de Servicio para operaciones de SMS  
  smsc_service_center_gt_address: "5551234567",  
  
  # Configuración de Auto Flush (procesamiento de cola de SMS en segundo plano)  
  auto_flush_enabled: true,  
  auto_flush_interval: 10_000,  
  auto_flush_dest_smsc: "ipsmgw",  
  auto_flush_tps: 10,  
  
  # Configuración de Conexión M3UA  
  # Conectarse como ASP para enviar/recibir operaciones MAP SMS  
  map_client_m3ua: %}
```

```

mode: "ASP",
callback: {MapClient, :handle_payload, []},
process_name: :stp_client_asp,
# Punto final local (sistema SMSc)
local_ip: {10, 179, 4, 12},
local_port: 2905,
# Punto final remoto STP
remote_ip: {10, 179, 4, 10},
remote_port: 2905,
routing_context: 1
}

config :control_panel,
  use_additional_pages: [
    {SS7.Web.EventsLive, "/events", "Eventos SS7"},
    {SS7.Web.TestClientLive, "/client", "Cliente SS7"},
    {SS7.Web.M3UAStatusLive, "/m3ua", "M3UA"},
    {SS7.Web.RoutingLive, "/routing", "Enrutamiento"},
    {SS7.Web.RoutingTestLive, "/routing_test", "Prueba de Enrutamiento"},
    {SS7.Web.SmscLinksLive, "/smsc_links", "Enlaces SMSc"}
  ],
  page_order: ["/events", "/client", "/m3ua", "/routing", "/routing_test", "/smsc_links",
"/application", "/configuration"]

```

## Parámetros de Configuración a Personalizar

Para una referencia completa de todos los parámetros de configuración, consulte la [Referencia de Configuración](#).

Parámetro	Tipo	Predeterminado	Descripción	Ejemplo
smsc_api_base_url	String	<i>Requerido</i>	Endpoint de API del backend de OmniMessage Su	"https://10.179.3.219:8443"
smsc_name	String	"{hostname}_SMSc"	identificador SMSc para registro	"ipsmgw"
smsc_service_center_gt_address	String	<i>Requerido</i>	Título Global del Centro de Servicio	"5551234567"
auto_flush_enabled	Boolean	true	Habilitar procesamiento automático de colas	false
auto_flush_interval	Integer	10_000	Intervalo de procesamiento de colas en milisegundos	5_000
auto_flush_dest_smsc	String	<i>Requerido</i>	Nombre del SMSc de destino para auto-flush	"ipsmgw"
auto_flush_tps	Integer	10	Tasa de procesamiento de mensajes (transacciones/segundo)	20
local_ip	Tuple	<i>Requerido</i>	Dirección IP de su sistema SMSc	{10, 179, 4, 12}
local_port	Integer	2905	Puerto SCTP local	2905
remote_ip	Tuple	<i>Requerido</i>	Dirección IP de STP para	{10, 179, 4, 10}

Parámetro	Tipo	Predeterminado	Descripción	Ejemplo
remote_port	Integer	2905	conectividad SS7 Puerto SCTP remoto	2905
routing_context	Integer	1	ID de contexto de enrutamiento M3UA	1

## ¿Qué Ocurre Cuando se Habilita el Modo SMSc?

Cuando `smsc_mode_enabled: true` y `map_client_enabled: true`, la interfaz web mostrará:

- **Eventos SS7** - Registro de eventos
- **Cliente SS7** - Pruebas de operación MAP
- **M3UA** - Estado de conexión
- **Enrutamiento** - Gestión de tabla de rutas (STP habilitado)
- **Prueba de Enrutamiento** - Pruebas de rutas (STP habilitado)
- **Enlaces SMSc** - Estado de API SMSc + gestión de cola de SMS ← *Específico de SMSc*
- **Recursos** - Monitoreo del sistema
- **Configuración** - Visor de configuración

La pestaña **Enlaces HLR** estará oculta.

## Notas Importantes

- El modo SMSc requiere `map_client_enabled: true` para capacidades de enrutamiento
- **Backend de OmniMessage**: La API del backend de OmniMessage debe ser accesible en la `smsc_api_base_url` configurada
- **Registro de Frontend**: El sistema se registra automáticamente con OmniMessage cada **5 minutos** a través del módulo `SMS.FrontendRegistry`
- **Tiempo de Espera de Solicitud API**: Todas las solicitudes API de OmniMessage tienen un **tiempo de espera codificado de 5 segundos**
- **Tiempo de Espera de Solicitud MAP**: Todas las solicitudes MAP (SRI-for-SM, MT-ForwardSM, etc.) tienen un **tiempo de espera codificado de 10 segundos**
- El auto-flush procesa automáticamente la cola de SMS en segundo plano
- La conexión M3UA a STP es necesaria para enviar/recibir operaciones MAP SMS
- Después de cambiar de modo, debe reiniciar la aplicación para que los cambios surtan efecto
- **Interfaz Web**: Consulte la [Guía de Interfaz Web](#) para información sobre el uso de la interfaz web
- **Acceso a API**: Consulte la [Guía de API](#) para la documentación de API REST y acceso a Swagger UI

## Configuración de API HTTP

### Configuración del Backend de OmniMessage

OmniSS7 se comunica con OmniMessage a través de la API REST HTTPS para gestionar la entrega de mensajes, rastrear el estado de los suscriptores y registrarse como un frontend activo:

```
config :omniss7,
  # URL base de la API de OmniMessage
  smsc_api_base_url: "https://10.5.198.200:8443",
  # Identificador de nombre del SMSC para registro (por defecto a hostname_SMSc si está vacío)
  smsc_name: "omni-smsc01",
  # Dirección GT del Centro de Servicio para operaciones de SMS
  smsc_service_center_gt_address: "5551234567"
```

### Parámetros de Configuración:

Parámetro	Tipo Requerido	Predeterminado	Descripción
<code>smsc_api_base_url</code>	String Sí	"https://localhost:8443"	URL base para la API de OmniMessage
<code>smsc_name</code>	String No	"" (usa	Identificador SMSC para

Parámetro	Tipo Requerido	Predeterminado	Descripción
		"{hostname}_SMSc")	registro y gestión de colas
smsc_service_center_gt_address	StringNo	"5551234567"	Dirección GT del Centro de Servicio devuelta en respuestas SRI-for-SM. Esto indica a otros elementos de la red dónde enrutar los mensajes MT-ForwardSM. Consulte la <a href="#">Guía SRI-for-SM</a> para más detalles.

## Registro de Frontend

El sistema se registra automáticamente con OmniMessage al inicio y **se vuelve a registrar cada 5 minutos** a través del módulo SMS.FrontendRegistry. Esto permite a OmniMessage:

- Rastrear frontends activos para balanceo de carga
- Monitorear tiempo de actividad y estado de salud
- Recopilar información de configuración
- Gestionar el enrutamiento de SMS distribuido a través de múltiples frontends

### Detalles de Implementación:

- **Intervalo de Registro:** 5 minutos (codificado)
- **Proceso:** Iniciado automáticamente cuando `smsc_mode_enabled: true`

### Carga Útil de Registro:

```
{
  "frontend_name": "omni-smsc01",
  "configuration": "{...}",
  "frontend_type": "SS7",
  "hostname": "smsc-server01",
  "uptime_seconds": 12345
}
```

**Nota:** El nombre del frontend se toma del parámetro de configuración `smsc_name`. Si no está configurado, por defecto es `"{hostname}_SMSc"`.

## Comunicación de API de OmniMessage

Cuando OmniSS7 recibe operaciones MAP de la red SS7 o procesa la cola de mensajes, se comunica con OmniMessage para:

- **Registrarse como un frontend activo** e informar el estado de salud
- **Enviar mensajes originados en móviles (MO)** recibidos de suscriptores
- **Recuperar mensajes terminados en móviles (MT)** de la cola para entrega
- **Actualizar el estado de entrega** con informes de éxito/fallo
- **Consultar información de enrutamiento** para el reenvío de mensajes

Endpoint	Método	Propósito	Cuerpo de Solicitud
/api/frontends	POST	Registrar instancia de frontend	{"frontend_name": "...", "frontend_type": "SMSc", "hostname": "...", "uptime_seconds": ...}
/api/messages_raw	POST	Insertar nuevo mensaje SMS	{"source_msisdn": "...", "source_smsc": "...", "message_body": "...}"
/api/messages	GET	Obtener cola de mensajes	Header: smsc: <smsc_name>
/api/messages/{id}	PATCH	Marcar mensaje como entregado	{"deliver_time": "...", "dest_smsc": "...}"
/api/messages/{id}	PUT	Actualizar estado del mensaje	{"dest_smsc": null}

Endpoint	Método	Propósito	Cuerpo de Solicitud
/api/locations	POST	Insertar/actualizar ubicación del suscriptor	{ "msisdn": "...", "imsi": "...", "location": "...", "ims_capable": true, "csfb": false, "expires": "...", "user_agent": "...", "ran_location": "...", "imei": "...", "registered": "..." }
/api/events	POST	Agregar seguimiento de eventos	{ "message_id": ..., "name": "...", "description": "..." }
/api/status	GET	Chequeo de salud	-

## Formato de Respuesta de API

Todas las respuestas de API utilizan el formato JSON con las siguientes convenciones:

- **Respuestas de éxito:** HTTP 200-201 con cuerpo JSON que contiene datos de resultado
- **Respuestas de error:** HTTP 4xx/5xx con detalles de error en el cuerpo de respuesta
- **Tiempos:** Formato ISO 8601 (por ejemplo, "2025-10-21T12:34:56Z")
- **IDs de Mensaje:** Identificadores enteros o de cadena

## Módulos de Cliente de API

El sistema SMS consta de tres módulos principales:

### 1. SMS.APIClient

Módulo principal del cliente API que proporciona toda la comunicación de API HTTP con OmniMessage:

- frontend\_register/4 - Registrar frontend con OmniMessage
- insert\_message/3 - Insertar mensaje SMS en crudo (versión de 3 parámetros compatible con Python)
- insert\_location/9 - Insertar/actualizar datos de ubicación del suscriptor
- get\_message\_queue/2 - Recuperar mensajes pendientes de la cola
- mark\_dest\_smsc/3 - Marcar mensaje como entregado o fallido
- add\_event/3 - Agregar seguimiento de eventos para mensajes
- flush\_queue/2 - Procesar mensajes pendientes (SRI-for-SM + MT-forwardSM)
- auto\_flush/2 - Bucle de procesamiento continuo de cola

### 2. SMS.FrontendRegistry

Maneja el registro periódico del frontend con el backend:

- Se registra automáticamente al inicio
- Se vuelve a registrar cada 5 minutos
- Utiliza smsc\_name de la configuración (vuelve a usar el nombre de host)
- Recopila información de configuración y tiempo de actividad del sistema

### 3. SMS.Utils

Funciones utilitarias para operaciones SMS:

- generate\_tp\_scts/0 - Generar marca de tiempo de SMS en formato TPDU

## Flujos de Mensajes SMS

### Flujo de SMS Entrante (Originado en Móvil)

### Flujo de SMS Saliente (Terminado en Móvil)

#### Pasos Clave Explicados:

- **Solicitud SRI-for-SM:** El SMSc consulta al HLR con el MSISDN de destino para determinar dónde enrutar el mensaje SMS. El HLR responde con:

- Un IMSI sintético (calculado a partir del MSISDN por privacidad) - consulte [MSISDN ↔ IMSI Mapping](#)
  - La dirección GT del SMSC (número de nodo de red) donde se debe enviar el MT-ForwardSM
  - Para detalles completos sobre cómo funciona esto, consulte [SRI-for-SM en la Guía HLR](#)
- **Solicitud MT-forwardSM:** Una vez que se obtiene la información de enrutamiento, el SMSc envía el mensaje SMS real al MSC/VLR que atiende al suscriptor

## Estructura del SMS TPDU

---

## Manejo del Centro de Servicio de Alertas

El SMSc puede recibir mensajes **alertServiceCenter** del HLR para rastrear el estado de alcanzabilidad de los suscriptores.

Para información sobre cómo el HLR envía mensajes alertServiceCenter, consulte [Integración del Centro de Servicio de Alertas en la Guía HLR](#).

### ¿Qué es alertServiceCenter?

Cuando un suscriptor realiza un UpdateLocation en el HLR (es decir, se registra con un nuevo VLR/MSC), el HLR puede notificar a los sistemas SMSc que el suscriptor ahora es alcanzable enviando un mensaje **alertServiceCenter** (código de operación MAP 64).

### Configuración

El tiempo de expiración de la ubicación se configura en el HLR:

```
config :omniss7,
  # Tiempo de expiración de la ubicación cuando SMSc recibe alertServiceCenter
  (predeterminado: 48 horas)
  hlr_alert_location_expiry_seconds: 172800
```

### Comportamiento

Cuando el SMSc recibe un mensaje alertServiceCenter:

1. **Decodificar MSISDN:** Extraer el MSISDN del suscriptor del mensaje (formato TBCD)
2. **Eliminar prefijo TON/NPI:** Quitar prefijos comunes como "19", "11", "91" (por ejemplo, "19123123213" → "123123213")
3. **Calcular IMSI:** Generar IMSI sintético utilizando el mismo mapeo que SRI-for-SM
4. **POST a /api/location:** Actualizar la base de datos de ubicación con:
  - msisdn: Número de teléfono del suscriptor (limpiado)
  - imsi: IMSI sintético
  - location: Nombre del SMSc (por ejemplo, "ipsmgw")
  - expires: Tiempo actual + hlr\_alert\_location\_expiry\_seconds
  - csfb: true (suscriptor alcanzable a través de Circuit-Switched Fallback)
  - ims\_capable: false (esto es registro CS de 2G/3G, no IMS/VoLTE)
  - user\_agent: GT del HLR que envió la alerta (para seguimiento)
  - ran\_location: "SS7"
5. **Rastrear en el Rastreador de Suscriptores SMSc:** Registrar al suscriptor con GT de HLR, estado=activo, contadores de mensajes en 0
6. **Enviar ACK:** Responder al HLR con acuse de recibo de alertServiceCenter

### Manejo de Suscriptores Ausentes

Cuando el SMSc intenta entregar un mensaje y recibe un error de "suscriptor ausente" durante SRI-for-SM (para más información sobre SRI-for-SM, consulte [SRI-for-SM en la Guía HLR](#)):

1. **Detectar ausencia:** SRI-for-SM devuelve error absentSubscriberDiagnosticSM
2. **Expirar ubicación:** POST a /api/location con expires=0 para marcar al suscriptor como inalcanzable
3. **Agente de usuario:** Establecer en "SS7\_AbsentSubscriber" para identificar la fuente
4. **Actualizar rastreador:** Marcar al suscriptor como fallido en el Rastreador de Suscriptores SMSc

Esto asegura que la base de datos de ubicación y el rastreador reflejen con precisión el estado de alcanzabilidad del suscriptor.

## Diagrama de Flujo

### Endpoint de API

#### POST /api/location

```
{
  "msisdn": "15551234567",
  "imsi": "001010123456789",
  "location": "ipsmgw",
  "ims_capable": false,
  "csfb": true,
  "expires": "2025-11-01T12:00:00Z",
  "user_agent": "15551111111",
  "ran_location": "SS7",
  "imei": "",
  "registered": "2025-10-30T12:00:00Z"
}
```

**Nota:** El campo `user_agent` contiene el GT del HLR que envió el `alertServiceCenter`, permitiendo al SMSc rastrear qué HLR está proporcionando actualizaciones de ubicación.

Para suscriptores ausentes, `expires` se establece en el tiempo actual (expiración inmediata).

---

## Prevención de Bucles

El SMSc implementa **prevención automática de bucles** para evitar bucles de enrutamiento de mensajes infinitos cuando los mensajes se originan en redes SS7.

### Por qué es Importante la Prevención de Bucles

Cuando el SMSc recibe mensajes SMS originados en móviles (MO) de la red SS7, los inserta en la cola de mensajes con un campo `source_smsc` que identifica su origen (por ejemplo, "SS7\_GT\_15551234567"). Sin prevención de bucles, estos mensajes podrían ser:

1. Recibidos de la red SS7 → Encolados con `source_smsc` conteniendo "SS7"
2. Recuperados de la cola → Procesados para entrega
3. Enviados de vuelta a la red SS7 → Creando un bucle

### Cómo Funciona

El SMSc detecta y previene automáticamente los bucles durante el procesamiento de mensajes:

### Implementación

Al procesar mensajes de la cola, el SMSc verifica el campo `source_smsc`:

- **Si `source_smsc` contiene "SS7":**
  - El mensaje se omite
  - Se agrega un evento: "Prevención de Bucles" con una descripción que explica la razón de la omisión
  - El mensaje se marca como fallido a través de una solicitud PUT
  - Se registra con nivel de advertencia
- **De lo contrario:**
  - El mensaje se procesa normalmente
  - Las operaciones SRI-for-SM y MT-ForwardSM continúan

## Valores de SMSC de Origen

Los mensajes pueden tener varios valores de source\_smsc:

Origen	Valor de Ejemplo	Acción
Red SS7 (MO-FSM)	"SS7_GT_15551234567"	<b>Omitido</b> - Prevención de bucles
API Externa/SMPP	"ipsmgw" o "api_gateway"	Procesado normalmente
Otro SMSC	"smsc-node-01"	Procesado normalmente

## Seguimiento de Eventos

Cuando un mensaje se omite debido a la prevención de bucles, se registra un evento:

```
{
  "message_id": 12345,
  "name": "Prevención de Bucles",
  "description": "Mensaje omitido - source_smsc 'SS7_GT_15551234567' contiene 'SS7',
previniendo bucle de mensaje"
}
```

Este evento es visible en:

- **Interfaz Web:** Página de Eventos SS7 (/events)
- **Base de Datos:** Tabla de events a través de la API
- **Registros:** Entradas de registro de nivel de advertencia

## Configuración

La prevención de bucles está **siempre habilitada** y no puede ser deshabilitada. Esta es una característica de seguridad crítica para prevenir interrupciones en la red debido a bucles de mensajes.

## Escenario de Ejemplo

**Escenario:** Suscriptor móvil envía SMS a través de la red SS7

1. Teléfono móvil → MSC/VLR → SMSc (a través de MO-ForwardSM)
2. SMSc recibe MO-FSM de GT 15551234567
3. SMSc inserta en la cola: source\_smsc = "SS7\_GT\_15551234567"
4. Auto-flush recupera el mensaje de la cola
5. SMSc detecta "SS7" en source\_smsc → OMITIR
6. Evento registrado: "Prevención de Bucles"
7. Mensaje marcado como fallido
8. No se envía SRI-for-SM ni MT-ForwardSM (bucle prevenido)

Sin la prevención de bucles, el paso 8 enviaría el mensaje de vuelta a la red SS7, potencialmente creando un bucle infinito.

---

## Seguimiento de Suscriptores SMSc

El SMSc incluye un GenServer de **Rastreador de Suscriptores** que mantiene el estado en tiempo real de los suscriptores basado en mensajes alertServiceCenter y en intentos de entrega de mensajes.

### Propósito

El rastreador proporciona:

- **Monitoreo de alcanzabilidad:** Qué suscriptores son actualmente alcanzables
- **Seguimiento de HLR:** Qué HLR envió el alertServiceCenter para cada suscriptor
- **Contadores de mensajes:** Número de mensajes enviados/recibidos por suscriptor
- **Seguimiento de fallos:** Marcar suscriptores como fallidos cuando los intentos de entrega fallan
- **Visibilidad en la Interfaz Web:** Panel en tiempo real que muestra todos los suscriptores rastreados

## Información Rastreada

Para cada suscriptor, el rastreador almacena:

<b>Campo</b>	<b>Descripción</b>	<b>Ejemplo</b>
msisdn	Número de teléfono del suscriptor (clave)	"15551234567"
imsi	IMSI del suscriptor	"001010123456789"
hlr_gt	GT de HLR que envió alertServiceCenter	"15551111111"
messages_sent	Conteo de mensajes MT-FSM enviados	5
messages_received	Conteo de mensajes MO-FSM recibidos	2
status	:active o :failed	:active
updated_at	Marca de tiempo Unix de la última actualización	1730246400

## Transiciones de Estado

### Comportamiento

#### Cuando se recibe alertServiceCenter:

- Crear o actualizar la entrada del suscriptor
- Establecer status = :active
- Registrar GT de HLR
- Reiniciar o preservar contadores de mensajes

#### Cuando SRI-for-SM tiene éxito:

- Incrementar el contador messages\_sent
- Actualizar la marca de tiempo updated\_at

#### Cuando SRI-for-SM falla:

- Establecer status = :failed
- Mantener en el rastreador para monitoreo

#### Cuando el suscriptor es eliminado:

- Eliminar de la tabla ETS
- Ya no aparece en la Interfaz Web

## Interfaz Web - Página de Suscriptores SMSc

**Ruta:** /smc\_subscribers **Auto-refresco:** Cada 2 segundos

**Nota:** Esta página solo está disponible cuando se ejecuta en modo SMSc. Después de descomentar la configuración SMSc en config/runtime.exs, debe reiniciar la aplicación para que la ruta esté disponible.

La página de **Suscriptores SMSc** proporciona monitoreo en tiempo real de todos los suscriptores rastreados:

### Características

#### 1. Tabla de Suscriptores

- MSISDN, IMSI, HLR GT
- Contadores de mensajes enviados/recibidos
- Insignia de estado (Activo/Fallido) con codificación de colores
- Marca de tiempo de última actualización y duración
- Botón de eliminación para suscriptores individuales

#### 2. Estadísticas Resumidas

- Total de suscriptores rastreados
- Conteo de suscriptores activos
- Conteo de suscriptores fallidos
- Número de HLR únicos

### 3. Acciones

- Limpiar Todo: Eliminar todos los suscriptores rastreados
- Eliminar: Eliminar suscriptor individual

#### Vista de Ejemplo

Suscriptores Rastreado SMSc			Total: 3	
MSISDN	IMSI	HLR GT	Msgs S/R	Estado
15551234567	001010123456789	15551111111	5/2	● Activo
15559876543	001010987654321	15551111111	0/0	● Activo
15551112222	001010111222233	15552222222	3/1	○ Fallido

Resumen: Total: 3 | Activos: 2 | Fallidos: 1 | HLR Únicos: 2

#### Funciones de API

El rastreador expone estas funciones para acceso programático:

```
# Llamado cuando se recibe alertServiceCenter
SMSc.SubscriberTracker.alert_received(msisdn, imsi, hlr_gt)

# Incrementar contadores de mensajes
SMSc.SubscriberTracker.message_sent(msisdn)
SMSc.SubscriberTracker.message_received(msisdn)

# Marcar como fallido (fallo de SRI-for-SM)
SMSc.SubscriberTracker.mark_failed(msisdn)

# Eliminar del seguimiento
SMSc.SubscriberTracker.remove_subscriber(msisdn)

# Funciones de consulta
SMSc.SubscriberTracker.get_active_subscribers()
SMSc.SubscriberTracker.get_subscriber(msisdn)
SMSc.SubscriberTracker.count_subscribers()
SMSc.SubscriberTracker.clear_all()
```

#### Integración

El rastreador está automáticamente integrado con:

- **Manejador de alertServiceCenter:** Llama a alert\_received/3 en la actualización de ubicación exitosa
- **Manejador de SRI-for-SM:** Incrementa messages\_sent en el enrutamiento exitoso
- **Manejador de suscriptor ausente:** Llama a mark\_failed/1 cuando el suscriptor está ausente
- **Errores de suscriptor desconocido:** Llama a mark\_failed/1 cuando SRI-for-SM falla

---

## Configuración de Auto-Flushing de Cola de SMS

El servicio de **Auto-Flushing** procesa automáticamente los mensajes SMS pendientes.

Para referencia de parámetros de configuración, consulte [Configuración de Auto-Flushing en la Referencia de Configuración](#).

#### Configuración

```
config :omnis7,
  auto_flush_enabled: true,          # Habilitar/deshabilitar auto-flush
  auto_flush_interval: 10_000,     # Intervalo de consulta en milisegundos
```

```
auto_flush_dest_smsc: nil,          # Filtro: nil = todos
auto_flush_tps: 10                  # Máximo de transacciones por segundo
```

## Cómo Funciona

1. **Consulta:** Cada `auto_flush_interval` milisegundos, consulta la API para mensajes pendientes
2. **Filtrado:** Opcionalmente filtrar por `auto_flush_dest_smsc`
3. **Limitación de Tasa:** Procesar hasta `auto_flush_tps` mensajes por ciclo
4. **Entrega:** Para cada mensaje:
  - Enviar **SRI-for-SM** (Enviar Información de Enrutamiento para Mensaje Corto) al HLR para obtener información de enrutamiento
    - El HLR devuelve un IMSI sintético calculado a partir del MSISDN
    - El HLR devuelve la dirección GT del SMSC donde se debe enviar el MT-ForwardSM
    - Consulte [Detalles de SRI-for-SM en la Guía HLR](#) para la documentación completa
  - En caso de éxito, enviar **MT-forwardSM** al MSC/VLR
  - Actualizar el estado del mensaje a través de la API (entregado/fallido)
  - Agregar seguimiento de eventos a través de la API

◊ **Profundización Técnica:** Para una explicación completa de cómo funciona SRI-for-SM, incluyendo el mapeo de MSISDN a IMSI, configuración de dirección GT del centro de servicio y la generación de IMSI sintético que preserva la privacidad, consulte la [sección SRI-for-SM en la Guía de Configuración HLR](#).

---

## Métricas SMSc

### Métricas Disponibles

#### Métricas de Cola de SMS:

- `smc_queue_depth` - Número actual de mensajes pendientes
- `smc_messages_delivered_total` - Total de mensajes entregados con éxito
- `smc_messages_failed_total` - Total de mensajes que fallaron en la entrega
- `smc_delivery_duration_milliseconds` - Histograma de tiempos de entrega

#### Consultas de Ejemplo:

```
# Profundidad actual de la cola
smc_queue_depth

# Tasa de éxito de entrega (últimos 5 minutos)
rate(smc_messages_delivered_total[5m]) /
(rate(smc_messages_delivered_total[5m]) + rate(smc_messages_failed_total[5m]))

# Tiempo promedio de entrega
rate(smc_delivery_duration_milliseconds_sum[5m]) /
rate(smc_delivery_duration_milliseconds_count[5m])
```

---

## Resolución de Problemas SMSc

### Problema: Mensajes No Entregados

#### Verificaciones:

1. Verifique que el auto-flush esté habilitado
2. Verifique la conexión a la base de datos
3. Monitoree los registros en busca de errores
4. Verifique que la conexión M3UA esté ACTIVA
5. Verifique los límites de TPS

### Problema: Alta Profundidad de Cola

#### Causas Posibles:

- Límite de TPS demasiado bajo
- Problemas de tiempo de espera del HLR
- Problemas de conectividad de red
- Números de destino inválidos

**Soluciones:**

- Aumentar `auto_flush_tps`
  - Verificar disponibilidad del HLR
  - Revisar registros de mensajes fallidos
- 

## API MT-forwardSM

### Enviar SMS a través de API

**Endpoint de API:** POST /api/MT-forwardSM

**Solicitud:**

```
{
  "imsi": "234509876543210",
  "destination_serviceCentre": "447999555111",
  "originating_serviceCenter": "447999123456",
  "smsPDU": "040B917477218345F600001570301857140C0BD4F29C0E9281C4E1F11A"
}
```

**Respuesta:**

```
{
  "result": "success",
  "message_id": "12345"
}
```

---

## Documentación Relacionada

**Documentación de OmniSS7:**

- [← Volver a la Documentación Principal](#)
- [Guía de Configuración HLR](#) - Configuración y operaciones en modo HLR
  - [Detalles Técnicos de SRI-for-SM](#) - Documentación completa sobre el mapeo de MSISDN a IMSI y configuración del centro de servicio
- [Guía de Características Comunes](#) - Interfaz Web, API, Monitoreo
- [Guía del Cliente MAP](#) - Operaciones MAP
- [Referencia Técnica](#) - Especificaciones de protocolo

**Documentación de OmniMessage:** Para la configuración de enrutamiento de mensajes, gestión de colas, seguimiento de entregas, limitación de tasa y analítica, consulte la **documentación del producto OmniMessage**. OmniMessage contiene toda la lógica de enrutamiento de mensajes, algoritmos de reintentos de colas, manejo de informes de entrega y motor de reglas de negocio.

---

OmniSS7 por Omnitouch Network Services

# Guía de Configuración de M3UA STP

[Volver a la Documentación Principal](#)

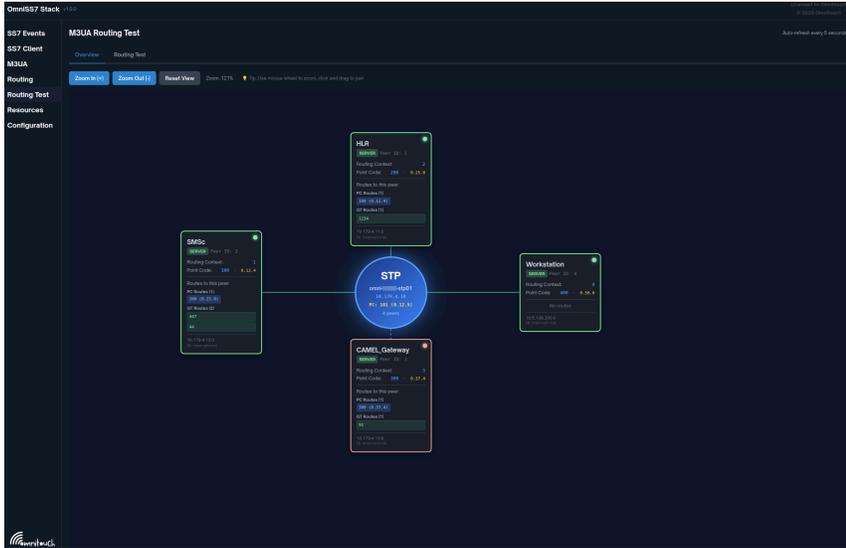
Esta guía proporciona una configuración detallada para usar OmniSS7 como un Punto de Transferencia de Señales (STP).

## Tabla de Contenidos

1. [Qué es un STP?](#)
2. [Roles de Red STP](#)
3. [Habilitando el Modo STP](#)
4. [Diagrama de Red STP](#)
5. [Funciones del STP](#)
  - [Entrenamiento de Mensajes](#)
  - [Transferencia de Pruebas](#)
  - [Distribución de Carga](#)
  - [Puerto de Enlace de Red](#)
  - [Ocultamiento de Topología](#)
6. [Diagrama de Red STP](#)
7. [Funciones del STP](#)
8. [Diagrama de Red STP](#)
9. [Funciones del STP](#)
10. [Diagrama de Red STP](#)
11. [Diagrama de Red STP](#)
12. [Diagrama de Red STP](#)

## ¿Qué es un Punto de Transferencia de Señales (STP)?

Un Punto de Transferencia de Señales (STP) es un elemento crítico de la red en redes de señalización SS7 e IP que envía mensajes de señalización entre nodos de la red.



### Funciones del STP

- **Entrenamiento de Mensajes:** Rutea el tráfico de señalización SS7 basado en el Código de Punto de Destino (PC) o Título Global (GT)
- **Transferencia de Pruebas:** Conecta redes SS7 tradicionales con redes NGN/NGN-IP basadas en IP.
- **Distribución de Carga:** Distribuye el tráfico entre múltiples destinos utilizando enrutamiento basado en prioridades.
- **Puerto de Enlace de Red:** Conecta diferentes redes de señalización y proveedores de servicios.
- **Ocultamiento de Topología:** Puede reemplazar direcciones para ocultar la topología interna de la red.

### Diagrama de Red STP

### Roles de Red STP Explicados

#### ASP (Proceso de Servidor de Aplicaciones)

- **Red:** Cliente que se conecta a un SCP/STP remoto
- **Dirección:** Conexión remota
- **Caso de Uso:** Su STP se conecta al STP de una red socio

#### SGP (Proceso de Puerta de Enlace de Señalización)

- **Red:** Servidor que acepta conexiones de ASPs
- **Dirección:** Conexión entrante
- **Caso de Uso:** Rutas socio se conectan a su STP

#### AS (Servidor de Aplicaciones)

- **Definición:** Aportación lógica de uno o más ASPs
- **Propósito:** Proporcionar redundancia y distribución de carga
- **Caso de Uso:** Múltiples ASPs que sirven el mismo destino

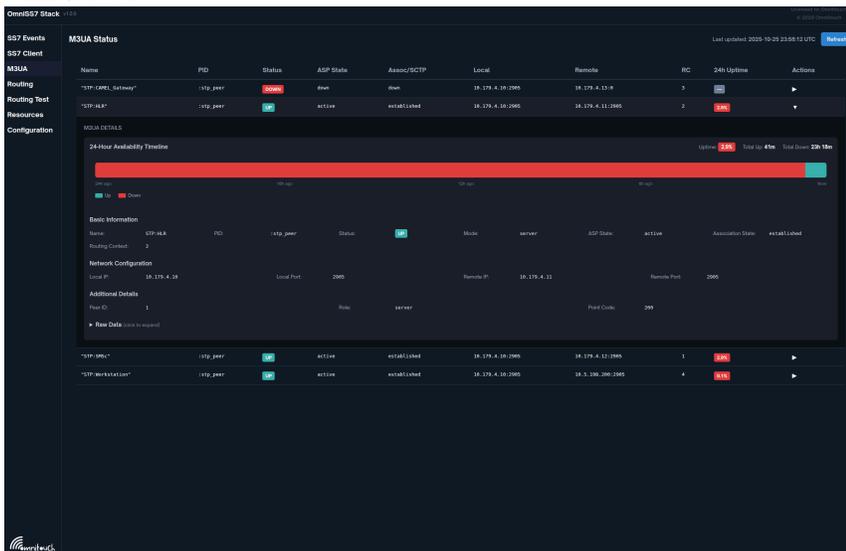
### Habilitando el Modo STP

OmniSS7 puede operar en diferentes modos. Para usarlo como un STP, necesita habilitar el modo STP en la configuración.

#### Cambiando al Modo STP

El archivo `config/route/asn` de OmniSS7 contiene tres modos operativos preconfigurados. Para habilitar el modo STP:

1. **Abra** `config/route/asn`
2. **Encuentra** las tres secciones de configuración (líneas 53-174):
  - Configuración 1: Modo STP (líneas 53-65)
  - Configuración 2: Modo HLR (líneas 87-125)
  - Configuración 3: Modo SMSC (líneas 125-174)
3. **Comentar** la configuración actualmente activa (agregar # a cada línea)
4. **Descomentar** la configuración STP (eliminar # de las líneas 53-65)
5. **Personalizar** los parámetros de configuración según sea necesario
6. **Reinicie** la aplicación: `ss7 -s -s6`



### Configuración del Modo STP

La configuración completa de STP se ve así:

```
config complete(), # modo de modo - Habilitar solo características STP mode_client_enabled: true, hlr_mode_enabled: false, stp_mode_enabled: false, work_mode_enabled: false,
```

```

# Configuración de Conexión M3UA
# Conectar como RSP (Proceso de Servidor de Aplicaciones) a STP/S0W remoto
mg_client_name M3
mode RSP
context: (mg_client, channel_playback, 11),
process_name: (mg_client, mg),
# Puerto Local (RFC 1828)
local_ip: (10, 179, 4, 10),
local_port: 2000,
# Puerto Final remoto STP/S0W
remote_ip: (10, 179, 4, 11),
remote_port: 2000,
routing_context: 1
}

```

#### Parámetros de Configuración a Personalizar

Para una referencia completa de todos los parámetros de configuración, consulte la [Referencia de Configuración](#).

Parámetro	Tipo	Predefinido	Descripción	Ejemplo
mg_client_enabled	Booleano	true	Habilitar capacidades de cliente MAP y enrutamiento	true
local_ip	Tupla	Dependiente	Dirección IP de su sistema	(10, 179, 4, 10)
local_port	Entero	2000	Puerto de STP local	2000
remote_ip	Tupla	Dependiente	Dirección IP remota de STP/S0W	(10, 179, 4, 11)
remote_port	Entero	2000	Puerto de STP remoto	2000
routing_context	Entero	1	ID de contexto de enrutamiento M3UA	1
enabled_routing	Booleano/Falso	true	Habilita enrutamiento de Tuba Global (además del enrutamiento por PC) true	true

#### ¿Qué Sucede Cuando se Habilita el Modo STP?

Cuando mg\_client\_enabled = true, la interfaz web monitorea:

- **Eventos SS7** - Registro de eventos
- **Cliente SS7** - Pruebas de operación MAP
- **M3UA** - Estado de conexión
- **Enrutamiento** - Consulta de estado de rutas - Específico de STP
- **Prueba de Enrutamiento** - Pruebas de ruta - Específico de STP
- **Escaneo** - Monitoreo del sistema
- **Configuración** - Visual de configuración

Las pestañas **Enlaces HLR** y **Enlaces SMS** estarán ocultas.

#### Notas Importantes

- El protocolo STP (protocolo IP 133) debe estar permitido a través de los firewalls
- El puerto M3UA predefinido es 2000 (estándar de la industria)
- Asegúrese de que haya suficientes recursos del sistema para manejar el tráfico de enrutamiento
- **Persistencia de Enrutamiento** - Todas las rutas configuradas a través de la interfaz web o API se almacenan en la base de datos **Mnesia** y sobreviven a los reinicios
- **Función de Configuración** - Las rutas de rutinas, así como el inicio y el fin de las rutas de Mnesia
- **Deposito de cambios de estado** debe reiniciar la aplicación para que los cambios surtiran efecto
- **Interfaz Web** - Consulte la [Guía de Interfaz Web](#) para gestionar rutas a través de la interfaz web
- **Acceso a API** - Consulte la [Guía de API](#) para la documentación de la API REST y acceso a Swagger UI

#### Modo STP Autónomo

Además de las capacidades de enrutamiento STP disponibles cuando mg\_client\_enabled = true, puede ejecutar un **servidor STP M3UA autónomo** que escuche conexiones entrantes.

#### Habilitando STP Autónomo

Agregue esta configuración a config/runtline.exe:

```

config: (mg_client,
  mg_client_name: M3,
  enabled: true,
  local_ip: (127, 0, 0, 1), # Dirección IP para escuchar
  local_port: 2000, # Puerto para escuchar
  point_code: 100 # Código de punto de este STP
)

```

#### Parámetros de Configuración STP

Parámetro	Tipo	Predefinido	Descripción	Ejemplo
enabled	Booleano/Falso	true	Habilitar servidor STP autónomo	true
local_ip	Tupla	(127, 0, 0, 1)	Dirección IP para escuchar conexiones (E, 0, 0, 0)	(127, 0, 0, 1)
local_port	Entero	2000	Puerto para escuchar	2000
point_code	Entero	100	Código de punto STP de este STP	100

#### Cómo Usar STP Autónomo

- **Enrutamiento Puro** - Cuando solo necesita enrutamiento M3UA sin funcionalidad de cliente MAP
- **STP Control** - true, crea un servidor de enrutamiento control para múltiples elementos de red
- **Arquitectura de Hub** - Conectar múltiples HLRs, MSCs y SMSGs a través de un STP control

**Nota:** Puede habilitar tanto mg\_client\_name como mg\_client\_enabled simultáneamente si necesita tanto conexiones salientes como funcionalidad STP entrante.

#### Persistencia de Tablas de Enrutamiento (Mnesia)

Todas las tablas de enrutamiento (pares, rutas de Código de Punto y rutas de Tuba Global) se almacenan en una **base de datos Mnesia** para persistencia.

#### Cómo Funciona el Enrutamiento

1. **Rutas de rutinas.exe** - Las rutas definidas en config/runtline.exe bajo mg\_client\_name, mg\_client\_enabled y mg\_client\_enabled se cargan al inicio de la aplicación
2. **Rutas de la Interfaz Web** - Las rutas agregadas a través de la [Interfaz Web de Enrutamiento de la Interfaz Web](#) se almacenan en Mnesia
3. **Punto de Inicio** - Al reiniciar, las rutas de rutinas.exe se fusionan con las rutas almacenadas en Mnesia (en duplicado)
4. **Persistencia** - Todas las rutas configuradas a través de la Interfaz Web **sobreviven a los reinicios de la aplicación**

#### Tipo de Almacenamiento de Mnesia

Consulte cómo se almacenan las tablas de enrutamiento. Para más detalles sobre la configuración de la base de datos, consulte [Configuración de Base de Datos en la Referencia de Configuración](#).

```

config: (mg_client,
  mnesia_storage_type: 'disk_copies' # 0: ram_copies para pruebas
)

```

Tipo de Almacenamiento	Descripción	Persistencia	Caso de Uso
disk_copies	Almacenamiento replicado en disco (predefinido <b>Sobrevive a los reinicios</b> )	Entonces de producción	Pruebas, desarrollo
ram_copies	Solo en memoria	Perdido al reiniciar	Pruebas, desarrollo

#### Ubicación de la Base de Datos Mnesia

Mnesia almacena tablas de enrutamiento en el directorio Mnesia de la aplicación:

- **Ubicación Mnesia (node\_name)** (por ejemplo, mnesia-node@prohost1)
- **Tablas mg\_client\_name** (por ejemplo, mg\_client\_name)

#### Gestionando Rutas

Hay tres operaciones para gestionar rutas:

1. **Runtline.exe** - Configuración estática cargada al inicio
2. **Interfaz Web** - Gestión interactiva de rutas (consulte la [Guía de Interfaz Web](#))
3. **API REST** - Gestión programática de rutas (consulte la [Guía de API](#))

**Mejor Práctica:** Use **runtline.exe** para la configuración base y la Interfaz Web para cambios dinámicos de rutas durante la operación.

#### Configurando Pares M3UA

Los pares representan puntos finales de conexión M3UA (entre STPs, HLRs, MSCs, SMSGs). Agregue pares a config/runtline.exe.

#### Ejemplo de Configuración de Pares

```

config: (mg_client,
  # Conexión saliente al STP Socio (rol: cliente)
  %
  peer_id: 1, # Identificador único
  name: "Partner_STP_Mgmt", # Nombre descriptivo
  role: "client", # rol para salientes, server para entrantes
  local_ip: (10, 0, 0, 1), # IP local para salir
  local_port: 0, # 0 = asignación de puerto dinámica
  remote_ip: (10, 0, 0, 10), # IP del par remoto
  remote_port: 2000, # Puerto del par remoto
  routing_context: 1, # Contexto de enrutamiento M3UA
  point_code: 100, # Código de punto de este par
  network_indicator: international # international o national
)
# Conexión al HLR Local (rol: cliente)
%
peer_id: 2,
name: "Local_HLR",
role: "client",
local_ip: (10, 0, 0, 1),
local_port: 0,
remote_ip: (10, 0, 0, 20),
remote_port: 2000,
routing_context: 1,
point_code: 200,
network_indicator: international
)
# Conexión entrante desde MSC Remoto (rol: server)
# Para rol: server, STP espera conexión entrante
%
peer_id: 3,
name: "Remote_MSC",
role: "server",
remote_ip: (10, 0, 0, 30), # Aceptar conexión entrante
remote_port: 2000, # IP de origen esperada
routing_context: 3, # Puerto de origen esperada (0 = aceptar de cualquier puerto)
)

```

```
peer_id: 300,
network_indicator: international
}
}
# Conexión entrante con puerto de origen dinámico (sin filtrado de puerto)
}
peer_id: 4,
name: "Dynamic_Client",
role: "server",
remote_ip: (10, 0, 0, 40), # IP de origen esperada
remote_port: 2, # P = aceptar conexiones de cualquier puerto de origen
routing_contact: 4,
point_code: 400,
network_indicator: international
}
}
```

The screenshot shows the Omnis57 Stack interface. The top section displays 'MSUA Status' for a peer named 'STP-4A'. It includes a table with columns for Name, PID, Status, ASP State, Assoc./SCTP, Local, Remote, RC, 24h Uptime, and Actions. The peer is shown as 'Up' and 'Open'. Below this, there is a '24-Hour Availability Timeline' bar chart showing 100% uptime. The 'Basic Information' section shows the peer name 'STP-4A', role 'server', and various configuration parameters like routing\_contact (2) and network\_configuration. The 'Network Configuration' section shows local and remote IP addresses and ports. The 'Additional Details' section shows peer ID (1) and role (server). The 'Raw Data' section shows a table of raw data entries for the peer.

### Parámetros de Configuración de Pares

Parámetro	Tipo	Requisito	Descripción
peer_id	Entero	SI	Identificador numérico único para el par
name	Cadena de texto	SI	Nombre legible por humanos para registros y monitoreo
role	Enumero	SI	<Client>: Indica rol de cliente (servidor)
local_ip	Tupla	SI (cliente)	Dirección IP local para enlace
local_port	Entero	SI (cliente)	Puerto local (IP para dinámico)
remote_ip	Tupla	SI	Dirección IP del par remoto
remote_port	Entero	SI	Puerto del par remoto (0 para entrante = aceptar de cualquier puerto de origen)
routing_contact	Entero	SI	Identificador de contacto de enrutamiento MSUA
point_code	Entero	SI	Código de punto SST de este par
network_indicator	Enumero	NO	:international o :national

### Filtrado de Puerto de Origen para Conexiones Entrantes

Para conexiones entrantes (rol: 'server'), el parámetro remote\_port controla el filtrado de puerto de origen:

- Puerto Específico** (por ejemplo, remote\_port: 2905): Solo aceptar conexiones de ese puerto de origen exacto
  - Proporciona seguridad adicional al reducir el puerto de origen
  - Usar cuando el par remoto utiliza un puerto de origen fijo
- Cualquier Puerto** (remote\_port: 0): Aceptar conexiones de cualquier puerto de origen
  - Un cuando el par remoto utiliza puertos de origen dinámicos/variables
  - Solo valida la dirección IP de origen
  - Más flexible pero ligeramente menos seguro

Ejemplo:

```
# Aceptar solo de 10.5.198.200:2905 (puerto específico)
}
peer_id: 1,
name: "Static_Peer",
role: "server",
remote_ip: (10, 5, 198, 200),
remote_port: 2905,
# ... otra configuración
}
# Aceptar de 10.5.198.200 con cualquier puerto de origen
}
peer_id: 2,
name: "Flexible_Peer",
role: "server",
remote_ip: (10, 5, 198, 200),
remote_port: 0, # Aceptar de cualquier puerto de origen
# ... otra configuración
}
```

### Soporte del Protocolo M2PA

Omnis57 soporta tanto MSUA como M2PA para el transporte de señalización SST.

#### ¿Qué es M2PA?

M2PA (Caja de Adaptación Peer-to-Peer del Usuario MTP2) es un protocolo estandarizado por IETF (RFC 4165) para transportar mensajes MTP3 de SST sobre redes IP utilizando SCTP.

#### MSUA vs M2PA: Diferencias Clave

Característica	MSUA	M2PA
Arquitectura	Cliente/Servidor (AS/PSGW)	Peer-to-Peer
Canales de Llamada	Puente entre SST e IP	Enlace directo punto a punto
Gestión de Estado de Enlace	Nivel de aplicación (AS/PSW/M2PA)	Enlace MTP2 (Administración, Prueba, Límite)
Número de Interfaz	Sin interconexión intermedia	RSVP/RSN de 24 bits para entrada ordenada
Diagrama Típico	Puerto de enlace SST=IP-SST	Enlaces de señalización directos entre nodos RFC 4666

#### Orientación sobre Selección de Protocolo

Recomendación: Usar MSUA por defecto. Solo usar M2PA cuando sea específicamente requerido.

#### ¿Cuándo Usar MSUA (Recomendado)

MSUA es el protocolo recomendado para la mayoría de los despliegues:

- Despliegues SIP**: Implementaciones estándar de puertos de transferencia de señalización
- Funciones de Puerto de Enlace**: Conectar entes SST con enrutamiento basado en IP
- Conectores de Elementos de Red**: Conectar HLR, MSC, SMSC, y otros elementos de red a un STP
- Puerto de Enlace de Señalización (SES)**: Puerto de enlace central que acepta conexiones de múltiples Servidores de Aplicaciones
- Diagrama de Enlace**: Diagrama de interconexión que muestra interconexión
- Redes Multi-proveedor**: Estándar de la industria ampliamente adoptado (RFC 4666)

Usar MSUA para conectar elementos de red (HLR, MSC, SMSC, VLR, etc.) a un STP.

#### ¿Cuándo Usar M2PA (Solo Casos Especiales)

M2PA solo debe usarse en escenarios específicos:

- Enlaces STP-to-STP**: Conexiones directas punto a punto entre Puertos de Transferencia de Señal en una red multi-STP
- Reemplazo de TDM Legacy**: Reemplazar enlaces TDM SST tradicionales cuando el sistema remoto requiere específicamente M2PA
- Compatibilidad MTP2 Requerida**: Si conectarse a sistemas legacy que exigen puertos de estado de enlace estilo MTP2
- Requisito de Soporte**: Cuando un socio o interconexión requiere específicamente el protocolo M2PA

Importante: No use M2PA para conectar elementos de red (HLR, MSC, SMSC) a un STP - use MSUA en su lugar. M2PA está diseñado para interconexiones STP-to-STP donde ambos lados operan como nodos de enrutamiento.

#### Configurando Pares M2PA

Los pares M2PA se configuran de la misma manera que los pares MSUA, con un parámetro adicional protocol.

#### Configuración de Par M2PA

Agregue pares M2PA a su configuración en config/routing/asn (¡sí, comparten la misma sección de configuración a pesar de ser diferentes protocolos):

#### Parámetros Clave para M2PA:

Parámetro	Valor	Descripción
protocol	m2pa	Especifica el protocolo M2PA (predeterminado a msua si se omite)
role	<Client>   <Server>	Dirección de conexión
local_ip	Entero	Puerto SCTP local (el puerto estándar de M2PA es 3565)
remote_ip	Entero	Puerto SCTP remoto del puerto estándar de M2PA es 3565
point_code	Entero	Código de punto
remote_port	Entero	Código de punto del par remoto (específico de M2PA)

Nota: M2PA utiliza el puerto 3565 como estándar de la industria (diferente del puerto 2905 de MSUA).

#### Estados de Enlace M2PA

Los enlaces M2PA progresan a través de varios estados durante la inicialización:

- Abajo**: No se ha establecido conexión
- Abriendo**: Fase de interconexión inicial (~1 segundo)
- Prueba**: Verificación de calidad de enlace (~1 segundo)
- Límite**: Enlace activo y listo para tráfico

La progresión del estado de enlace asegura señalización confiable antes de que se intercambie tráfico.

#### Gestionando Pares M2PA a través de la Interfaz Web

La página de Enrutamiento en la Interfaz Web proporciona soporte completo para gestionar pares M2PA:

- Navegar** a la página de Enrutamiento
- Subir** la pestaña "Pares"
- Hacer clic** en "Agregar Nuevo Par"
- Editar** M2PA (RFC 4165) del menú desplegable de Protocolo
- Completar** la configuración del par:
  - Número de Par (identificador descriptivo)
  - Protocolo M2PA
  - Rol (cliente o servidor)
  - Código de Punto (en PC)
  - Dirección IP de Anfitrión
  - Puertos locales/remotos (específicamente 3565 para M2PA)
  - Indicador de Red (internacional o nacional)
- Hacer clic** en "Guardar Par"

La tabla de pares muestra el tipo de protocolo con configuración de colores:

- Azul**: pares MSUA
- Verde**: pares M2PA

#### Comportamiento de Enrutamiento M2PA

Los pares M2PA se integran sin problemas con el sistema de enrutamiento de Omnis57:

- Rutas de Código de Punto**: Funcionan de manera idéntica para M2PA y MSUA
- Rutas de Tráfico Global**: Tratamiento apropiado en enlaces M2PA

- **Prioridad de Ruta:** Los puros MZPA y MEUA pueden mostrarse en las mismas tablas de enrutamiento
- **Reserva de Mensajes:** Los mensajes pueden llegar por MZPA y ser enrutados a MEUA, y viceversa

### Métricas MZPA

MZPA proporciona métricas completas de Promesas para monitorizar la salud del enlace y el tráfico:

#### Métricas de Tráfico:

- `m2a_messages_sent_total` - Total de mensajes MZPA enviados por enlace
- `m2a_messages_received_total` - Total de mensajes MZPA recibidos por enlace
- `m2a_bytes_sent_total` - Total de bytes enviados a través de MZPA
- `m2a_bytes_received_total` - Total de bytes recibidos a través de MZPA

Todas las métricas de tráfico están etiquetadas por: `Link_name, peer_id, adjacent_pc`

#### Métricas de Estado de Enlace:

- `m2a_link_state_change_total` - Transiciones de estado de enlace (ABAJO → ALINEACION → PREUBA → LISTO)
- `linkup`: `Link_name, from_state, to_state`

#### Métricas de Error:

- `m2a_errors_total` - Total de errores por tipo
- `disconnect_error` - Fallos en la descodificación de mensajes MZPA
- `encode_error` - Fallos en la codificación de mensajes MZPA
- `sctp_send_error` - Fallos en la transmisión SCTP
- `linkup`: `Link_name, error_type`

#### Métricas de Acceso:

- Punto final de Promesas: `http://peer-server:8888/metrics`
- Las métricas se registran automáticamente al inicio de la aplicación

### Mejores Prácticas MZPA

1. **Selección de Puerto:** Use el puerto 3565 para MZPA (estándar de la industria)
2. **Monitoreo de Estado:** Monitoree los cambios de estado de enlace a través de métricas
3. **Reglas de Firewall:** Asegúrese de que SCTP (puerto 3565) esté permitido
4. **Código de Punto:** Asegúrese de que los códigos de punto avanzados estén configurados correctamente en ambos lados
5. **Indicador de Red:** Debe coincidir entre pares (internacional o nacional)
6. **Pruebas:** Use la página de Pruebas de Enrutamiento para verificar la conectividad después de la configuración

## Configurando el Enrutamiento por Código de Punto

El enrutamiento por Código de Punto dirige mensajes basados en el Código de Punto de Destino (DPC) en el encabezado MTP3.

### Entendiendo los Códigos de Punto en la Pila de Protocolo SS7

Los códigos de punto existen en diferentes capas de la pila de protocolos SS7. Comprender esta distinción es importante:

#### Capas de la Pila de Protocolo:



#### Dos Tipos de Códigos de Punto:

1. **Códigos de Punto de Capa MTP3** (Usados para Enrutamiento):
  - Utilizados en la etiqueta de enrutamiento MTP3 (DPC, DPC\_SLS)
  - Presentes en el parámetro de Datos de Protocolo de MEUA (etiqueta S28)
  - Presentes en mensajes de Datos de Usuario de MZPA
  - **STP utiliza estos valores DPC para decisiones de enrutamiento**
  - Estos mensajes deben ser entregados finalmente al mensaje.
2. **Códigos de Punto de Capa MEUA** (Usados para Gestión de Red):
  - Presentes en mensajes de gestión MEUA (DUNA, DANA, SCUN, DUPL)
  - Indican códigos de punto dedicados para el estado de la red
  - Indican a los pares que destino están disponibles/disponibles
  - No se utilizan para enrutamiento por código de punto de datos de usuario

#### Cómo Funciona el Enrutamiento STP:

- **Para mensajes MEUA DATA:** STP extrae el mensaje MTP3 del parámetro de Datos de Protocolo (etiqueta S28), que contiene la etiqueta de enrutamiento MTP3 (DPC, DPC\_SLS). El DPC de la capa MTP3 se utiliza para buscar rutas.
- **Para mensajes de Datos de Usuario:** STP extrae el mensaje MTP3 del campo de datos de usuario MEUA, luego lee el DPC de la etiqueta de enrutamiento MTP3.
- **Mensajes de gestión MEUA:** Los mensajes de gestión de la red (DUNA, DANA, SCUN) contienen códigos de punto dedicados a nivel de MEUA para indicar el estado de la red entre pares.

### Rutas Básicas de Código de Punto

Agrege rutas a `config/routing`, así:

```

config name1;
  m3ua routes;
  # Enrutar todo el tráfico para PC 100 al par 1 (STP South)
  #
  dest_pc: 100;                # Código de punto de destino
  peer_id: 1;                 # Par a través del cual enrutar
  priority: 1;                # Prioridad (menor = mayor prioridad)
  network_indicator: international;
  # mask: 14                  # Opcional: predeterminado a 14 (coincidencia exacta)
  #
  # Enrutar todo el tráfico para PC 200 al par 2 (MLR local)
  #
  dest_pc: 200;
  peer_id: 2;
  priority: 1;
  network_indicator: international;
  #
  # Ejemplo de balanceo de carga: PC 300 con rutas primaria y de respaldo
  #
  dest_pc: 300;
  peer_id: 3;                 # Ruta primaria
  priority: 1;
  network_indicator: international;
  #
  dest_pc: 300;
  peer_id: 4;                 # Ruta de respaldo (número de prioridad más alto)
  priority: 2;
  network_indicator: international;
  #
  #

```

**Nota:** El campo `mask` es opcional y predeterminado a 14 (coincidencia exacta). Solo especifique `mask` cuando necesite enrutamiento basado en rangos (consulte la sección de Mascaras de Código de Punto a continuación).

### Lógica de Enrutamiento

1. STP recibe un mensaje MEUA DATA o un mensaje de Datos de Usuario MZPA
2. STP extrae el **mensaje MTP3** del campo de Datos de Protocolo (MEUA) o del campo de Datos de Usuario (MZPA)
3. STP lee el **Código de Punto de Destino (DPC)** de la etiqueta de enrutamiento MTP3
4. Busca en la lista de rutas coincidentes los códigos de DPC coincidentes
5. Si existen múltiples rutas, selecciona la ruta con **máscara más específica** (valor de máscara más alto), luego **número de prioridad más bajo**
6. Envía el mensaje MTP3 al destino de Usuario MZPA para el par de destino
7. Busca el mensaje al par correspondiente
8. Si el par seleccionado está inactivo, selecciona la siguiente ruta de mayor prioridad

The screenshot shows the Omnis7 Stack STP configuration interface. The main section is 'MSUA Routing Management'. Under 'Routing Test', there is a table showing the results of a test:

OT Profile	Source SSN	TC/NPM	Peer ID	Peer	Dest SSN	Priority	Description	Action
91	any	-	3	MSC_South (3)	presence	1	India numbers	[Disable] [SR] [Details]
124	any	-	1	STP_West (1)	presence	1	US numbers	[Disable] [SR] [Details]
447	any	-	2	STP_East (2)	presence	1	UK mobile numbers	[Disable] [SR] [Details]
44	any	-	2	STP_East (2)	presence	1	UK numbers	[Disable] [SR] [Details]

Below the table, there are sections for 'Common SSN Values' and 'DROPPED Routes - Preventing Routing Loops'. The 'DROPPED Routes' section includes a warning: 'DROPPED routes peer\_id=0 identify discard traffic instead of forwarding it. This is useful for preventing routing loops in complex network topologies.'

### Mascaras de Código de Punto

Las máscaras de punto tienen valores de 14 bits (rango 0-16383). No defectos, las máscaras coinciden exactamente con un solo código de punto (máscara /14). Sin embargo, puede usar **mascaras de código de punto** para crear rutas que coincidan con **rangos** de códigos de punto.

#### Entendiendo Mascaras

Las máscaras específicas cuántos bits más significativos cubren coincidan con el PC de destino de la ruta y el DPC del mensaje entrante. Los bits restantes pueden ser cualquier valor, creando un rango de códigos de punto coincidentes.

#### Tabla de Referencia de Mascaras:

Máscara/Código de Punto Coincidentes	Caso de Uso
/14	1 PC (coincidencia exacta)
/13	2 PCs
/12	4 PCs
/11	8 PCs
/10	16 PCs
/9	32 PCs
/8	64 PCs
/7	128 PCs
/6	256 PCs
/5	512 PCs
/4	1,024 PCs
/3	2,048 PCs
/2	4,096 PCs
/1	8,192 PCs
/0	16,384 PCs (toda posibilidad de coincidencia de respuesta)

#### Ejemplos de Mascaras de Código de Punto

**Nota:** El campo `mask` es **opcional** en todos los ejemplos. Si se omite, predetermina a 14 (coincidencia exacta).

#### Ejemplo 1: Código de Punto Único (Comportamiento Predeterminado)

# Sin campo de máscara (recomendado para un solo PC)

```

#1
dest_pc: 1000,
peer_id: 1,
priority: 1,
network_indicator: international
# La máscara predefinida es 14 - Coincide: Solo PC 1000
# Máscara explícita (mismo resultado)
dest_pc: 1000,
peer_id: 1,
priority: 1,
mask: 14,
network_indicator: international
# Coincidencia exacta explícita
# Coincide: Solo PC 1000

```

#### Ejemplo 2: Rango Pequeño

```

#1
dest_pc: 1000,
peer_id: 2,
priority: 1,
mask: 16,
network_indicator: international
# Coincide 4 PCs
# Coincide: PC 1000, 1001, 1002, 1003

```

#### Ejemplo 3: Rango Medio

```

#1
dest_pc: 1000,
peer_id: 3,
priority: 1,
mask: 8,
network_indicator: international
# Coincide 64 PCs
# Coincide: PC 1000-1063 (64 códigos de punto consecutivos)

```

#### Ejemplo 4: Ruta Predefinida de Respaldo

```

#1
dest_pc: 0,
peer_id: 4,
priority: 10,
network_indicator: international
# Baja prioridad (número alto)
# Coincide con todos los PCs
# Coincide: Todos los códigos de punto (0-16383)
# Nota: como ruta de captura predefinida con baja prioridad

```

#### Combinando Rutas Específicas y Rutas enmascaradas

Puede combinar rutas específicas con rutas enmascaradas para un enrutamiento flexible:

```

config omit17,
# Ruta específica para PC 100 (toma precedencia)
#1
dest_pc: 1000,
peer_id: 1,
priority: 1,
network_indicator: international
# La máscara predefine a 14 (coincidencia exacta)
#2
dest_pc: 1000,
peer_id: 2,
priority: 1,
mask: 8,
network_indicator: international
# Coincide 64 PCs
# Ruta predefinida/de respaldo para todos los demás PCs
#3
dest_pc: 0,
peer_id: 3,
priority: 10,
network_indicator: international
# Baja prioridad
# Coincide con todos los PCs

```

#### Decisión de Enrutamiento para DPC 1000:

- Coincide con la ruta de máscara /14 (PC 1000 exactamente) - **Seleccionada** (más específica)
- También coincide con la ruta de máscara /8 (rango PC 1000-1063) - **Ignorada** (menos específica)
- También coincide con la ruta de máscara /8 (todos los PCs) - **Ignorada** (menos específica)

#### Decisión de Enrutamiento para DPC 1013:

- No coincide con la ruta de máscara /14 (solo PC 1000)
- Coincide con la ruta de máscara /8 (rango PC 1000-1063) - **Seleccionada** (coincidencia más específica)
- También coincide con la ruta de máscara /8 (todos los PCs) - **Ignorada** (menos específica)

#### Decisión de Enrutamiento para DPC 1000:

- No coincide con la ruta de máscara /14
- No coincide con la ruta de máscara /8
- Coincide con la ruta de máscara /8 (todos los PCs) - **Seleccionada** (única coincidencia, ruta de respaldo)

#### Mejores Prácticas

- Omitir mask para Destinoes Únicas:** Para coincidencias exactas de códigos de punto, omita completamente el campo mask (predeterminado a /14)
- Usar /8 Específicamente Solo Cuando Sea Necesario:** Solo especificar mask: 14 cuando necesite dirigir tráfico en la documentación o al mostrar con rutas de rango
- Usar Máscaras de Rango para Bloques de Red:** Rutas segmentadas completas de red o partes específicas con máscaras /8 a /13
- Usar /8 como Respaldo:** Como una ruta predefinida con baja prioridad para capturar tráfico no coincidente
- El Más Específico Gana:** El motor de enrutamiento siempre selecciona primero la ruta coincidente más específica (valor de máscara más alto)
- Precedencia como Breaker:** Si múltiples rutas toman la misma máscara, el número de precedencia más bajo gana

## Configurando el Enrutamiento por Título Global (GT)

El enrutamiento por Título Global permite el **enrutamiento basado en contenido** utilizando números de teléfono o valores DNSI en lugar de códigos de punto. Para una traducción avanzada de direcciones de Título Global basada en el perfil que **Barra/Flamenco**, consulte la [Guía de NAT de Título Global](#).

#### Requisitos Previos

- Habilitar el enrutamiento GT: enable\_gt\_routing: true en config/runtime.xml

#### Configuración de Rutas GT

```

config omit17,
# Habilitar enrutamiento GT
enable_gt_routing: true,
# Ruta GT:
#1
# Definir todos los números del Reino Unido (prefijo 44) al par 1
gt_prefix: "44",
peer_id: 1,
priority: 1,
description: "Números del Reino Unido"
# Enrutar números de EE. UU. (prefijo 1) al par 2
gt_prefix: "1",
peer_id: 2,
priority: 1,
description: "Números de EE. UU."
# Ruta más específica: números móviles del Reino Unido que comienzan con 447
gt_prefix: "447",
peer_id: 3,
priority: 1,
description: "Números móviles del Reino Unido"
# Enrutamiento específico de SSN (opcional)
gt_prefix: "555",
source_ssn: 8,
peer_id: 4,
mask_len: 8,
priority: 1,
description: "Tráfico SMS para prefijo 61"

```

#### Lógica de Enrutamiento GT

El algoritmo de enrutamiento GT sigue este proceso de decisión:

##### Paso de Enrutamiento:

- Coincidencia de Prefijo Más Largo:** El STP encuentra todas las rutas GT desde el prefijo coincide con el inicio del Título Global
  - Ejemplo: GT "44712345678" coincide tanto con "44" como con "447", pero "447" gana (coincidencia más larga)
- Coincidencia de SSN (Opcional):**
  - Si se especifica source\_ssn, la ruta solo coincide cuando el SSN de la Parte Llamada del SCCP es igual a ese valor
  - Si source\_ssn es nil, la ruta coincide con cualquier SSN (omitido)

- Coincidencia de TTNPNNAI (Opcional):**
  - Si se especifican source, TT, source\_npi o dest\_nai, las rutas deben coincidir con esos indicadores
  - Las rutas se actúan como comodín (coinciden con cualquier valor)
- Selección Basada en Especificidad:**
  - Las rutas con criterios de coincidencia más específicos ganan sobre los comodín
  - Orden de prioridad: Longitud de Prefijo GT > SSN > TT > NPI > NAI > Número de Prioridad
- Rescritura de Indicadores (Opcional):**
  - Si se especifican dest\_ssn, dest\_tt, dest\_npi o dest\_nai, el STP rescribe esos indicadores
  - Útil para normalización de planillas e implementación de red
- Recada al Código de Punto:**
  - Si no coincide ninguna ruta GT, el STP recae al enrutamiento por Código de Punto utilizando el DPC

#### Enrutamiento Avanzado GT: Tipo de Traducción, NPI y NAI

Además de la coincidencia de prefijo GT y SSN, el STP admite el enrutamiento y la transformación basados en indicadores de Títulos Global SCCP:

- Tipo de Traducción (TT)** Identifica el plan de numeración y el tipo de dirección
- Indicador de Plan de Numeración (NPI)** Define el plan de numeración (por ejemplo, ISDN, Datos, Telex)
- Indicador de Naturaleza de Dirección (NAI)** Especifica el formato de dirección (por ejemplo, Internacional, Nacional, Suscriptor)

#### Coincidencia (Indicadores de Origen)

Las rutas pueden coincidir en los indicadores de mensajes entrantes:

- source: TT: Coincidir mensajes con Tipo de Traducción específico
- source\_nai: Coincidir mensajes con Indicador de Plan de Numeración específico
- source\_npi: Coincidir mensajes con Indicador de Naturaleza de Dirección específico
- value\_nai = comodín (coincide con cualquier valor)

#### Transformación (Indicadores de Destino)

Las rutas pueden rescribir indicadores al reenviar:

- dest\_tt: Transformar Tipo de Traducción a nuevo valor
- dest\_nai: Transformar Indicador de Plan de Numeración a nuevo valor
- dest\_npi: Transformar Indicador de Naturaleza de Dirección a nuevo valor
- value\_nai = preservar valor original (sin transformación)

#### Selección Basada en Especificidad

Cuando múltiples rutas coinciden, se selecciona la ruta más específica utilizando este orden de prioridad:

- Coincidencia de prefijo GT más largo
- SSN específico de origen sobre SSN comodín
- TT específico de origen sobre TT comodín
- NPI específico de origen sobre NPI comodín
- NAI específico de origen sobre NAI comodín
- Número de prioridad más bajo

#### Ejemplos de Configuración

```

config remote;
enable_gt_routing: true;

# Ejemplo 1: Coincidir y transformar Tipo de Traducción
#
gt_prefix: "44";
peer_id: 1;
source_ssn: 8; # Coincidir TT=0 (Desconocido)
dest_tt: 3; # Transformar a TT=3 (Nacional)
priority: 1;
description: "Número del Reino Unido: transformación TT 0->3";
}

# Ejemplo 2: Coincidir NPI específico y transformar NAI
#
gt_prefix: "11";
peer_id: 2;
source_npi: 1; # Coincidir NPI=1 (ISDN/Telefono)
source_nai: 4; # Coincidir NAI=4 (Internacional)
dest_nai: 3; # Transformar a NAI=3 (Nacional)
priority: 1;
description: "Número de EE. UU.: NAI Internacional->Nacional";
}

# Ejemplo 3: Enrutamiento combinado de SSN e indicadores
#
gt_prefix: "11P";
source_ssn: 8; # Coincidir tráfico SMS
source_tt: 0; # Coincidir TT=0
dest_ssn: 6; # Rescribir SSN a NLR
dest_tt: 2; # Transformar a TT=2
dest_npi: 1; # Establecer NPI=1 (ISDN)
dest_nai: 4; # Establecer NAI=4 (Internacional)
peer_id: 3;
priority: 1;
description: "SMS francés: normalización completa";
}

# Ejemplo 4: TT comodín, NPI específico
#
gt_prefix: "44P";
source_tt: 0; # Coincidir cualquier TT (comodín)
source_npi: 6; # Coincidir NPI=6 (Telex)
dest_npi: 1; # Transformar a NPI=1 (ISDN)
peer_id: 4;
priority: 1;
description: "Normalización de red de datos alemana";
}

```

#### Valores Comunes de TTNPNNAI

##### Tipo de Traducción (TT):

- 0 = Desconocido
- 1 = Internacional
- 2 = Nacional
- 3 = Especifico de Red

##### Indicador de Plan de Numeración (NPI):

- 0 = Desconocido
- 1 = ISDN/Telefono (E.164)
- 2 = Datos (X.121)
- 4 = Telex (E.16)
- 6 = Móvil Terrestre (E.212)

##### Indicador de Naturaleza de Dirección (NAI):

- 0 = Desconocido
- 1 = Número de Suscriptor
- 2 = Reservado para Uso Nacional
- 3 = Número Significativo Nacional
- 4 = Número Internacional

##### Ejemplo de Decisión de Enrutamiento

Para un mensaje entrante con:

- GT: "44712345678"
- SSN: 8
- TT: 0
- NPI: 1
- NAI: 4

Con estas rutas configuradas:

```

# Ruta A: TT comodín
%gt_prefix: "44?"; peer_id: 1; priority: 1;

# Ruta B: TT específico + NPI
%gt_prefix: "44?"; source_tt: 0; peer_id: 2; priority: 1;

# Ruta C: TT específico + NPI
%gt_prefix: "44?"; source_tt: 0; source_npi: 1; peer_id: 3; priority: 1;

```

**Resultado:** Se selecciona la Ruta C (más específica, coincide GT + TT + NPI). El mensaje se reenvía con indicadores transformados según los valores de dest\_tt, dest\_npi, dest\_nai de la Ruta C.

#### Ejemplos de Enrutamiento GT

GT	Llamado SSN de Origen TTNPNNAI	Ruta Coincidente	Razón
44712345678	- - - - -	44? - par 0	Coincidencia de prefijo más largo
441234567890	- - - - -	44? - par 1	Coincidencia de prefijo, sin ruta más específica
1212551234	- - - - -	1? - par 2	Coincidencia de prefijo, sin número de EE. UU.
5558812345678	- - - - -	55? (SSN 8) - par 4	Coincidencia de GT + SSN, rescribe SSN a 8
5558812345678	- - - - -	55? (comodín SSN) - par 3	Coincidencia de GT, sin rescritura de SSN
441234567890	0 1 4	44? (TT=0) - par 1	Coincidencia de GT + TT, transforma TT a 3
1212551234	0 1 4	1? (TT=0, NPI=1, NAI=4)	Más específico: coincidencia GT+TT+NPNNAI

#### Cases de Uso Comunes para Enrutamiento TTNPNNAI

- Normalización de Interconexión de Redes:**
  - Diferentes redes pueden usar diferentes convenciones de indicadores
  - Transformar indicadores en el punto de interconexión para asegurar compatibilidad
  - Ejemplo: la red asociada con TT=0 para internacional, se red sea TT=1

## 2. Conversión de Protocolo

- Convertir entre planes de numeración al conectar entre diferentes tipos de red
- Ejemplo: Ruta de red móvil (NPN=6) a PSTN (NPN=1)

## 3. Estandarización de Formato de Dirección

- Normalizar todo el tráfico entrante para usar valores NAJ consistentes
- Ejemplo: Convertir todos los formatos internacionales (NA=4) a formato nacional (NA=3) para enrutamiento interno

## 4. Enrutamiento Específico de Proveedor

- Rutas basadas en tipo de indicación a diferentes proveedores de servicios
- Ejemplo: IT=0 rutas a Carrier A, IT=1 rutas a Carrier B

## 5. Integración de Sistemas Legados

- Los sistemas modernos pueden usar diferentes valores de indicador que los sistemas legados
- Transformar en el STP para mantener compatibilidad hacia atrás

## Características de Gestión de Rutas

### Deshabilitando Rutas

Las rutas pueden deshabilitarse temporalmente sin eliminarlas. Esto es útil para pruebas, mantenimiento o gestión de tráfico.

### Flag Habilitado

Tanto las rutas de Código de Punto como las de Título Global admiten un flag `enabled` opcional:

```
config omnisc17
show routes |
# Ruta activa
#
dest_pc 100
peer_id 1
priority 1
network_indicator international
enabled true # Ruta activa (predeterminado si se omite)
}

# Ruta deshabilitada (no evaluada durante el enrutamiento)
#
dest_pc 200
peer_id 2
priority 1
network_indicator international
enabled false # Ruta deshabilitada
}

}

show gt_routes |
# Ruta GT deshabilitada
#
gt_prefix "44"
peer_id 1
priority 1
description "Número del Reino Unido - temporalmente deshabilitado"
enabled false
}

}
```

### Comportamiento Predeterminado

- Si `enabled` no se especifica, las rutas predeterminan a `enabled: true`
- Las rutas deshabilitadas se omiten completamente durante la búsqueda de rutas
- Use la interfaz Web para obtener rutas activadas/ocultas sin editar la configuración

### Causa de Uso

- Probar cambios en el flujo de tráfico
- Ventanas de mantenimiento temporales
- Pruebas A/B de diferentes caminos de enrutamiento
- Implementación gradual de nuevas rutas

## Rutas DROP - Previendo Bucles de Enrutamiento

Las rutas DROP (con `peer_id: 0`) descartan silenciosamente el tráfico en lugar de reenviarlo. Esto previene bucles de enrutamiento y permite un filtrado avanzado de tráfico.

### Configurando Rutas DROP

```
config omnisc17
show routes |
# Ruta DROP para código de punto específico
#
dest_pc 999
peer_id 0 # peer_id=0 significa DROP
priority 1
network_indicator international
}

}

show gt_routes |
# Ruta DROP para prefijo GT
#
gt_prefix "999"
peer_id 0 # peer_id=0 significa DROP
priority 0 # Bloquear rango de prueba
description "Bloquear rango de prueba"
}

}
```

### Cómo Funcionan las Rutas DROP

Cuando un mensaje coincide con una ruta DROP:

1. El motor de enrutamiento identifica `peer_id: 0`
2. El mensaje es **silenciosamente descartado** (no reenviado)
3. Se genera un **registro INFO**: "Ruta DROP coincidente para OPC 999" o "Ruta DROP coincidente para GT 999"
4. La búsqueda de enrutamiento devuelve [error, -dropped]

**Importante:** El tráfico descartado se registra a nivel INFO para monitoreo y solución de problemas.

### Caso de Uso Común: Listado de Prefijos

Uno de los usos más poderosos de las rutas DROP es el **listado de prefijos**: permitiendo solo números específicos dentro de un gran rango mientras se bloquean todos los demás.

### El Patrón

1. Crear una ruta DROP para todo el prefijo con **número de prioridad alta** (por ejemplo, 99)
2. Crear rutas de permit específicas para números individualizados con **números de prioridad bajos** (por ejemplo, 1)
3. Dado que los números de prioridad más bajos se evalúan primero, las rutas permitidas coinciden antes que la ruta DROP
4. Cualquier número que no esté explícitamente permitido se captura en la ruta DROP

### Ejemplo de Configuración

Tiene un prefijo GT 1234 que representa un rango de 10,000 números (1234000000 - 1234999999), pero solo desea mostrar 3 números específicos: 1234550000, 1234550006 y 1234112222.

```
config omnisc17
show routes |
# Ruta DROP con PRIORIDAD ALTA (evaluada al final)
#
gt_prefix "1234"
peer_id 0 # DROP
priority 99 # Número alto = baja prioridad = evaluada al final
description "Bloquear todo los 1234's excepto números permitidos"
}

# Rutas de permitir específicas con NÚMEROS DE PRIORIDAD BAJOS (evaluadas primero)
#
gt_prefix "1234550000"
peer_id 1 # Ruta al par 1
priority 1 # Número bajo = alta prioridad = evaluada primero
description "Número permitido 1"
}

gt_prefix "1234550006"
peer_id 1
priority 1
description "Número permitido 2"
}

gt_prefix "1234112222"
peer_id 1
priority 1
description "Número permitido 3"
}

}
```

### Comportamiento de Enrutamiento:

GT Entrante	Rutas Calificadas	Ruta Seleccionada	Acción
123450000	1234112222 (prioridad 1) + 1234550000 (más específica, menor prioridad)	1234112222	Enrutado al par 1
123455000	1234550000 (prioridad 1) + 1234112222 (prioridad 1)	1234550000	Enrutado al par 1
1234550006	1234550006 (prioridad 1) + 1234112222 (prioridad 1)	1234550006	Enrutado al par 1
123411222	1234112222 (prioridad 1) + 1234112222 (más específica, mayor prioridad)	1234112222	Enrutado al par 1
1234999999	1234112222 (prioridad 1) + 1234112222 (más específica)	1234112222	Destruído + registrado
1234000000	1234112222 (prioridad 1) + 1234112222 (más específica)	1234112222	Destruído + registrado

### Resultado:

- Solo 3 números específicos se enrutan al par 1
- Todos los demás números 1234's son silenciosamente descartados
- Todo el tráfico descartado se registra para monitoreo

### Registros Generados:

- [INFO] Ruta DROP coincidente para GT 1234999999
- [INFO] Ruta DROP coincidente para OPC 1000

### Rutas DROP para Códigos de Punto

El mismo patrón de listado funciona para el enrutamiento por Código de Punto:

```
config omnisc17
show routes |
# DROP todo el rango /8 (64 códigos de punto: 1000-1063)
#
dest_pc 1000
peer_id 0
priority 99
mask 8
network_indicator international
}

# Permitir códigos de PC específicos
#
dest_pc 1020 peer_id 1 priority 1 network_indicator international
dest_pc 1028 peer_id 1 priority 1 network_indicator international
dest_pc 1030 peer_id 1 priority 1 network_indicator international
}

}
```

**Resultado:** Solo los PCs 1020, 1028 y 1030 se enrutan. Todos los demás PCs en el rango 1000-1063 son descartados.

### Monitoreo de Rutas DROP

#### Verifique los Registros:

```
# Monitorear tráfico descartado
tail -f logs/app-log | grep "Ruta DROP coincidente"

# Salida esperado:
[INFO] Ruta DROP coincidente para GT 1234999999
[INFO] Ruta DROP coincidente para OPC 1000
```

#### A través de la Interfaz Web:

- Navegue a la pantalla **Registros del Sistema**
- Filtre por nivel INFO
- Busque "Ruta DROP coincidente"

#### Mejores Prácticas:

1. • Monitoree los registros regularmente para asegurarse de que las rutas DROP no bloqueen tráfico legítimo
2. • Use campos de `description` para documentar por qué se descartan rutas
3. • Use números de prioridad más altos para rutas `deny` y para asegurarse de que las rutas de captura
4. • Pruebe el comportamiento de la ruta DROP antes de implementarla en producción
5. • Configure alertas para eventos inesperados en el tráfico descartado

## Enrutamiento Avanzado: Enrutamiento y Reescritura Basados en SSN

### Números de Subsistema (SSN)

Los Números de Subsistema identifican la capa de aplicación:

- **SSN 6:** HLR (Departamento de Ubicación de Hooper)
- **SSN 7:** HLR (Departamento de Ubicación de Hooper)
- **SSN 8:** CMC (Centro de Consultas Móvil) CMC (Centro de SMS)
- **SSN 9:** CMC (Centro de Ubicación Móvil de Puerta de Entrada)

### Ejemplo de Enrutamiento Basado en SSN

Enrutar tráfico SMS a diferentes HLR según el prefijo del número:

```
show gt_routes |
# Dirigir SMS para números del Reino Unido al HLR del Reino Unido, reescribir SSN de 8 (SMS) a 6 (HLR)
#
gt_prefix "44"
peer_id 0 # Coincidir SSN entrante 8 (SMS)
peer_id 6 # Reescribir a SSN 6 (HLR)
dest_pc 10
priority 1
description: "SMS del Reino Unido a HLR"
```

1. • **SSN 6:** HLR (Departamento de Ubicación de Hooper)
2. • **SSN 7:** HLR (Departamento de Ubicación de Hooper)
3. • **SSN 8:** CMC (Centro de Consultas Móvil) CMC (Centro de SMS)
4. • **SSN 9:** CMC (Centro de Ubicación Móvil de Puerta de Entrada)

```

},
# Enviar tráfico de voz para números del Reino Unido (506 6) sin reescritura
%{
  source_ssn: 6,          # Coincidir SSN entrante 6 (HLR)
  peer_id: 1,           # Sin reescritura de SSN
  dest_ssn: 511,
  srvc_ssn: 1,
  description: "Tráfico de voz del Reino Unido"
}
}

```

## Probando la Configuración de Enrutamiento STP

Después de configurar pares y rutas, verifique su configuración:

### 1. Verificar Estado de Pares

A través de la interfaz Web:

- Navegue a la página de estado MSLA
- Verifique que los pares muestran Estado: **ACTIVO**

A través de la Consola CLI:

```
# Obtener todos los estados de pares
R00A-STP-peer_status
```

```
# Salida esperada:
{
  "peer_id": 1, name: "Partner_STP_West", status: inactive, point_code: 100, ...},
  "peer_id": 2, name: "Local_HLR", status: active, point_code: 200, ...}
}

```

### 2. Probar el Enrutamiento por Código de Punto

```
# Enviar mensaje MSLA de prueba a DPC 100
test_payload = ech_7_1; dev # Carga #11 de prueba
R00A-STP-route_by_pc(100, test_payload, 0)
```

```
# Verificar registros para decisión de enrutamiento
# Registro esperado: "Enrutando mensaje: DPC->... -> DPC=100 a través del par 1"
```

### 3. Probar el Enrutamiento por Título Global

```
# Buscar ruta GT manualmente
R00Routing.lookup_peer_by_gt("44771345678")
```

```
# Salida esperada:
# [{"rt": [{"rtua_peer": 3, "rt_Mobile_Peer": "..."}, null]
```

```
# Buscar ruta GT con SSN
R00Routing.lookup_peer_by_gt("55881234567", 0)
```

```
# Salida esperada con reescritura de SSN:
# [{"rt": [{"rtua_peer": 4, "rt_HLR_Peer": "..."}, 0]
```

### 4. Monitorear Métricas de Enrutamiento

Acceda a las métricas de Prometheus en #metrics

Métricas clave:

```
# Mensajes recibidos por par
#hu_otp_messages_received_total{peer_name="Partner_STP_West",point_code="100"} 1523
```

```
# Mensajes enviados por par
#hu_otp_messages_sent_total{peer_name="Local_HLR",point_code="200"} 1408
```

```
# Fallos de enrutamiento
#hu_otp_routing_failures_total{reason="no_route"} 0
#hu_otp_routing_failures_total{reason="no_route"} 2
```

## Métricas y Monitoreo STP

### Métricas Disponibles

#### Métricas de Tráfico por Par

- #hu\_otp\_messages\_received\_total - Total de mensajes recibidos de cada par
  - Etiquetas: peer\_name, point\_code
- #hu\_otp\_messages\_sent\_total - Total de mensajes enviados a cada par
  - Etiquetas: peer\_name, point\_code

\*\*Métricas de Fallos de Enrutamiento



# Guía de la Interfaz de Usuario Web

[← Volver a la Documentación Principal](#)

Esta guía proporciona documentación completa para el uso de la **Interfaz de Usuario Web** de OmniSS7 (interfaz Phoenix LiveView).

## Tabla de Contenidos

1. [Descripción General](#)
  2. [Acceso a la Interfaz de Usuario Web](#)
  3. [Página de Gestión de Enrutamiento](#)
  4. [Página de Suscriptores Activos](#)
  5. [Operaciones Comunes](#)
  6. [Comportamiento de Auto-Actualización](#)
- 

## Descripción General

La Interfaz de Usuario Web de OmniSS7 es una aplicación de **Phoenix LiveView** que proporciona capacidades de monitoreo y gestión en tiempo real. Las páginas disponibles dependen de cuál modo operativo esté activo (STP, HLR o SMSc).

## Arquitectura de la Interfaz de Usuario Web

### Configuración del Servidor

- **Protocolo:** HTTPS
- **Puerto:** 443 (configurado en config/runtime.exs)
- **IP Predeterminada:** 0.0.0.0 (escucha en todas las interfaces)
- **Certificados:** Ubicados en priv/cert/

**URL de Acceso:** https://[server-ip]:443

---

## Acceso a la Interfaz de Usuario Web

### Requisitos Previos

1. **Certificados SSL:** Asegúrese de que los certificados SSL válidos estén presentes en priv/cert/:

- omnitouch.crt - Archivo de certificado
- omnitouch.pem - Archivo de clave privada

2. **Aplicación en Ejecución:** Inicie la aplicación con `iex -S mix`
3. **Cortafuegos:** Asegúrese de que el puerto 443 esté abierto para el tráfico HTTPS

## Páginas Disponibles por Modo

Página	Modo STP	Modo HLR	Modo SMSc	Descripción
<b>Eventos SS7</b>	◇	◇	◇	Registro de eventos y captura de mensajes SCCP
<b>Cliente SS7</b>	◇	◇	◇	Pruebas de operación MAP manual
<b>M3UA</b>	◇	◇	◇	Estado de conexión M3UA
<b>Enrutamiento</b>	◇	◇	◇	Gestión de la tabla de enrutamiento M3UA
<b>Prueba de Enrutamiento</b>	◇	◇	◇	Pruebas y validación de rutas
<b>Enlaces HLR</b>	◇	◇	◇	Estado de la API HLR y gestión de suscriptores
<b>Suscriptores Activos</b>	◇	◇	◇	Seguimiento de ubicación de suscriptores en tiempo real (HLR)
<b>Enlaces SMSc</b>	◇	◇	◇	Estado de la API SMSc y gestión de colas
<b>Suscriptores SMSc</b>	◇	◇	◇	Seguimiento de suscriptores en tiempo real (SMSc)
<b>Aplicación</b>	◇	◇	◇	Recursos del sistema y monitoreo
<b>Configuración</b>	◇	◇	◇	Visor de configuración

## Página de Gestión de Enrutamiento

**Página:** /routing

**Modos:** STP, SMSc

**Auto-Actualización:** Cada 5 segundos

La página de Gestión de Enrutamiento proporciona una interfaz con pestañas para gestionar las tablas de enrutamiento M3UA.

### Diseño de la Página

#### Pestaña de Peers

Gestione las conexiones de pares M3UA (otros STPs, HLRs, MSCs, SMSCs).

## Columnas de la Tabla de Peers

Columna	Descripción	Ejemplo
<b>ID</b>	Identificador único del par	1
<b>Nombre</b>	Nombre del par legible por humanos	"STP_West"
<b>Rol</b>	Rol de conexión	client, server, stp
<b>Point Code</b>	Código de punto SS7 del par	100
<b>Remoto</b>	IP:Puerto remoto	10.0.0.10:2905
<b>Estado</b>	Estado de conexión	active, aspup, down
<b>Acciones</b>	Botones de Editar/Eliminar	-

## Agregar un Par

1. **Haga clic** en la pestaña Peers
2. **Complete** los campos del formulario:
  - **Peer ID:** Se genera automáticamente si se deja vacío
  - **Peer Name:** Nombre descriptivo (requerido)
  - **Role:** Seleccione client, server o stp
  - **Point Code:** Código de punto SS7 (requerido)
  - **Local IP:** Dirección IP de su sistema
  - **Local Port:** 0 para asignación de puerto dinámica
  - **Remote IP:** Dirección IP del par
  - **Remote Port:** Puerto del par (típicamente 2905)
  - **Routing Context:** ID de contexto de enrutamiento M3UA
  - **Network Indicator:** international o national
3. **Haga clic** en "Add Peer"

**Persistencia:** El par se guarda inmediatamente en Memoria y sobrevive al reinicio.

## Editar un Par

1. **Haga clic** en el botón "Edit" en la fila del par
2. **Modifique** los campos del formulario según sea necesario
3. **Haga clic** en "Update Peer"

**Nota:** Si cambia el Peer ID, el par antiguo se elimina y se crea uno nuevo.

## Eliminar un Par

1. **Haga clic** en el botón "Delete" en la fila del par
2. **Confirme** la eliminación (todas las rutas que usen este par también serán eliminadas)

## Indicadores de Estado del Par

Estado	Color	Descripción
active	Verde	El par está conectado y enrutando mensajes
aspup	Amarillo	ASP está activo pero aún no está en funcionamiento
down	Rojo	El par está desconectado

---

## Pestaña de Rutas de Código de Punto

Configure reglas de enrutamiento basadas en Códigos de Punto de destino.

### Columnas de la Tabla de Rutas

Columna	Descripción	Ejemplo
<b>Destino PC</b>	Código de punto objetivo (formato zone.area.id)	1.2.3 (100)
<b>Máscara</b>	Máscara de subred para coincidencia de PC	/14 (exacto), /8 (rango)
<b>Peer ID</b>	Par objetivo para esta ruta	1
<b>Peer Name</b>	Nombre del par objetivo	"STP_West"
<b>Prioridad</b>	Prioridad de la ruta (1 = más alta)	1
<b>Red</b>	Indicador de red	international
<b>Acciones</b>	Botones de Editar/Eliminar	-

### Agregar una Ruta de Código de Punto

1. **Haga clic** en la pestaña "Point Code Routes"
2. **Complete** los campos del formulario:
  - **Destino Point Code:** Ingrese como zone.area.id (por ejemplo, 1.2.3) o como entero (0-16383)
  - **Máscara:** Seleccione la máscara /14 para coincidencia exacta, valores más bajos para rangos
  - **Peer ID:** Seleccione el par objetivo del menú desplegable
  - **Prioridad:** Ingrese la prioridad (1 = más alta, predeterminado)
  - **Network Indicator:** Seleccione international o national
3. **Haga clic** en "Add Route"

**Formato de Código de Punto:** Puede ingresar códigos de punto en dos formatos:

- **Formato 3-8-3:** zone.area.id (por ejemplo, 1.2.3)
- **Formato Entero:** 0-16383 (por ejemplo, 1100)

El sistema convierte automáticamente entre formatos.

## Comprender las Máscaras

Los códigos de punto son valores de 14 bits (0-16383). La máscara especifica cuántos bits más significativos deben coincidir:

Máscara	PCs Coincidentes	Caso de Uso
/14	1 (coincidencia exacta)	Ruta a un destino específico
/13	2 PCs	Rango pequeño
/8	64 PCs	Rango mediano
/0	Todos los 16,384 PCs	<b>Ruta predeterminada/de respaldo</b>

### Ejemplos:

- PC 1000 /14 → Coincide solo con PC 1000
- PC 1000 /8 → Coincide con PC 1000-1063 (64 PCs consecutivos)
- PC 0 /0 → Coincide con todos los códigos de punto (ruta predeterminada)

## Tarjeta de Referencia de Máscara de Código de Punto

La página web incluye una referencia interactiva que muestra todos los valores de máscara y sus rangos.

---

## Pestaña de Rutas de Títulos Globales

Configure reglas de enrutamiento basadas en direcciones de Títulos Globales SCCP.

**Requisito:** El enrutamiento de Títulos Globales debe estar habilitado en la configuración:

```
config :omniss7,  
  enable_gt_routing: true
```

## Columnas de la Tabla de Rutas

Columna	Descripción	Ejemplo
<b>GT Prefix</b>	Prefijo GT del partido llamado (vacío = respaldo)	"1234", ""
<b>Source SSN</b>	Coincidencia en SSN del partido llamado (opcional)	6 (HLR), any
<b>Peer ID</b>	Par objetivo	1
<b>Peer</b>	Nombre del par	"HLR_West (1)"
<b>Dest SSN</b>	Reescribir SSN al reenviar (opcional)	6, preserve
<b>Prioridad</b>	Prioridad de la ruta	1
<b>Descripción</b>	Descripción de la ruta	"Números de

Columna	Descripción	Ejemplo
<b>Acciones</b>	Botones de Editar/Eliminar	EE.UU."
		-

## Agregar una Ruta de Título Global

1. **Haga clic** en la pestaña "Global Title Routes"
2. **Complete** los campos del formulario:
  - **GT Prefix:** Deje vacío para la ruta de respaldo, o ingrese dígitos (por ejemplo, "1234")
  - **Source SSN:** Opcional - filtrar por SSN del partido llamado
  - **Peer ID:** Seleccione el par objetivo
  - **Dest SSN:** Opcional - reescribir SSN al reenviar
  - **Prioridad:** Prioridad de la ruta (1 = más alta)
  - **Descripción:** Descripción legible por humanos
3. **Haga clic** en "Add Route"

**Rutas de Respaldo:** Si el GT Prefix está vacío, la ruta actúa como una captura para GTs que no coinciden con ninguna otra ruta.

## Valores Comunes de SSN

La página incluye una tarjeta de referencia con valores comunes de SSN:

SSN	Elemento de Red
6	HLR (Registro de Ubicación de Hogar)
7	VLR (Registro de Ubicación de Visitantes)
8	MSC (Centro de Conmutación Móvil)
9	EIR (Registro de Identidad de Equipos)
10	AUC (Centro de Autenticación)
142	RANAP
145	gsmSCF (Función de Control de Servicio)
146	SGSN

## Reescritura de SSN

- **Source SSN:** Coincidencia en el SSN del Partido Llamado en mensajes entrantes
- **Dest SSN:** Si se establece, reescribe el SSN del Partido Llamado al reenviar
  - Vacío = preservar el SSN original
  - Valor = reemplazar con este SSN

**Caso de Uso:** Rutar mensajes con SSN=6 (HLR) a un par, y reescribir a SSN=7 (VLR) en el lado saliente.

---

## Persistencia de la Tabla de Enrutamiento

Todas las rutas se almacenan en Mnesia y sobreviven a los reinicios de la aplicación.

### Cómo Persisten las Rutas

1. **Cambios en la Interfaz de Usuario Web:** Todas las operaciones de agregar/editar/eliminar se guardan inmediatamente en Mnesia
2. **Reinicio de la Aplicación:** Las rutas se cargan desde Mnesia al iniciar
3. **Fusión de runtime.exs:** Las rutas estáticas de config/runtime.exs se fusionan con las rutas de Mnesia (sin duplicados)

### Prioridad de Ruta

Cuando múltiples rutas coinciden con un destino:

1. **Más Específico Primero:** Valores de máscara más altos (más específicos) tienen prioridad
2. **Campo de Prioridad:** Números de prioridad más bajos enrutados primero (1 = mayor prioridad)
3. **Estado del Par:** Solo se utilizan rutas a pares active

---

## Página de Suscriptores Activos

**Página:** /subscribers

**Modo:** Solo HLR

**Auto-Actualización:** Cada 2 segundos

Muestra el seguimiento en tiempo real de los suscriptores que han enviado solicitudes de UpdateLocation.

### Características de la Página

#### Columnas de la Tabla de Suscriptores

Columna	Descripción	Ejemplo
<b>IMSI</b>	IMSI del suscriptor	"50557123456789"
<b>Número VLR</b>	Dirección GT actual del VLR	"555123155"
<b>Número MSC</b>	Dirección GT actual del MSC	"555123155"
<b>Actualizado En</b>	Última marca de tiempo de UpdateLocation	"2025-10-25 14:23:45 UTC"
<b>Duración</b>	Tiempo desde el registro	"2h 15m 34s"

## Resumen de Estadísticas

Cuando hay suscriptores presentes, se muestra una tarjeta de resumen:

- **Total Activo:** Número total de suscriptores registrados
- **VLRs Únicos:** Número de direcciones VLR distintas
- **MSCs Únicos:** Número de direcciones MSC distintas

## Limpiar Suscriptores

**Botón Limpiar Todo:** Elimina todos los suscriptores activos del rastreador.

**Confirmación:** Requiere confirmación antes de limpiar (no se puede deshacer).

**Caso de Uso:** Limpiar registros de suscriptores obsoletos después del mantenimiento o pruebas de la red.

## Auto-Actualización

La página se actualiza automáticamente cada **2 segundos** para mostrar actualizaciones en tiempo real de los suscriptores.

---

## Suscriptores SMSc

**Página:** /smc\_subscribers

**Modo:** Solo SMSc

**Auto-Actualización:** Cada 2 segundos

Muestra el seguimiento en tiempo real de los suscriptores basado en mensajes alertServiceCenter recibidos de HLRs, estado de entrega de mensajes y seguimiento de fallos.

## Características de la Página

### Columnas de la Tabla de Suscriptores

Columna	Descripción	Ejemplo
<b>MSISDN</b>	Número de teléfono del suscriptor	"15551234567"
<b>IMSI</b>	IMSI del suscriptor	"001010123456789"
<b>HLR GT</b>	HLR GT que envió alertServiceCenter	"15551111111"
<b>Msgs Enviados</b>	Conteo de mensajes MT-FSM enviados	5
<b>Msgs Recibidos</b>	Conteo de mensajes MO-FSM recibidos	2

Columna	Descripción	Ejemplo
<b>Estado</b>	Active o Failed (codificado por color)	● Active
<b>Última Actualización</b>	Marca de tiempo de la última actualización	"2025-10-30 14:23:45 UTC"
<b>Duración</b>	Tiempo desde la última actualización	"15m 34s"

## Indicadores de Estado

- **Active** (Verde): El suscriptor es alcanzable, la última alertServiceCenter se recibió con éxito
- **Failed** (Rojo): El último intento de entrega falló (error SRI-for-SM o suscriptor ausente)

## Resumen de Estadísticas

Cuando hay suscriptores presentes, se muestra una tarjeta de resumen:

- **Total Rastreado:** Número total de suscriptores rastreados
- **Activos:** Número de suscriptores con estado activo
- **Fallidos:** Número de suscriptores con estado fallido
- **HLRs Únicos:** Número de HLRs distintos que envían alertas

## Gestión de Suscriptores

**Botón Eliminar:** Elimina un suscriptor individual del seguimiento.

**Botón Limpiar Todo:** Elimina todos los suscriptores rastreados.

**Confirmación:** Limpiar Todo requiere confirmación antes de limpiar (no se puede deshacer).

### Caso de Uso:

- Eliminar entradas obsoletas después de problemas en la red
- Limpiar datos de prueba después del desarrollo
- Monitorear qué HLRs están enviando alertas

## Contadores de Mensajes

El rastreador incrementa automáticamente los contadores:

- **Mensajes Enviados:** Incrementado cuando SRI-for-SM tiene éxito y se envía MT-FSM
- **Mensajes Recibidos:** Incrementado cuando se recibe MO-FSM del suscriptor

## Auto-Actualización

La página se actualiza automáticamente cada **2 segundos** para mostrar actualizaciones en tiempo real de suscriptores y estados.

---

## Operaciones Comunes

### Búsqueda y Filtrado

Actualmente, la Interfaz de Usuario Web no incluye funcionalidad de búsqueda/filtrado incorporada. Para encontrar rutas específicas:

1. Use la función de búsqueda de su navegador (Ctrl+F / Cmd+F)
2. Busque nombres de pares, códigos de punto o prefijos GT

### Operaciones Masivas

Para realizar cambios masivos en las rutas:

1. **Opción 1:** Use la [API REST](#) para acceso programático
2. **Opción 2:** Edite `config/runtime.exs` y reinicie la aplicación
3. **Opción 3:** Use la Interfaz de Usuario Web para cambios individuales en las rutas

### Exportar/Importar

**Nota:** La Interfaz de Usuario Web actualmente no admite la exportación o importación de tablas de enrutamiento. Las rutas son:

- Almacenadas en archivos de base de datos Mnesia
- Configuradas en `config/runtime.exs`

Para hacer una copia de seguridad de las rutas:

1. **Mnesia:** Haga una copia de seguridad del directorio `Mnesia.{node_name}/`
  2. **Configuración:** Control de versiones de `config/runtime.exs`
- 

## Comportamiento de Auto-Actualización

Diferentes páginas tienen diferentes intervalos de actualización:

Página	Intervalo de Actualización	Razón
Gestión de	5 segundos	Los cambios de ruta son poco

Página	Intervalo de Actualización	Razón
Enrutamiento		frecuentes
Suscriptores Activos	2 segundos	El estado de los suscriptores cambia con frecuencia
Estado M3UA	Varía según la página	Monitoreo del estado de conexión

**Conexión WebSocket:** Todas las páginas utilizan conexiones WebSocket de Phoenix LiveView para actualizaciones en tiempo real.

**Interrupción de Red:** Si se pierde la conexión WebSocket, la página intentará reconectarse automáticamente.

## Solución de Problemas

### Página No Cargando

1. **Verifique el Certificado HTTPS:** Asegúrese de que `priv/cert/omnitouch.crt` y `.pem` estén presentes
2. **Verifique el Puerto 443:** Verifique que las reglas del cortafuegos permitan tráfico HTTPS
3. **Aplicación en Ejecución:** Confirme que la aplicación esté en ejecución con `ix -S mix`
4. **Consola del Navegador:** Verifique si hay errores de certificado SSL (advertencias de certificado autofirmado)

### Rutas No Persistentes

1. **Verifique el Almacenamiento de Mnesia:** Verifique `mnesia_storage_type: :disc_copies` en la configuración
2. **Directorio de Mnesia:** Asegúrese de que el directorio de Mnesia sea escribible
3. **Verifique los Registros:** Busque errores de Mnesia en los registros de la aplicación

### Auto-Actualización No Funciona

1. **Conexión WebSocket:** Verifique la consola del navegador para errores de WebSocket
2. **Red:** Verifique la conexión de red estable
3. **Recargar Página:** Intente refrescar la página (F5)

# Documentación Relacionada

- [Guía STP](#) - Configuración de enrutamiento detallada
  - [Guía HLR](#) - Gestión de suscriptores
  - [Guía API](#) - API REST para acceso programático
  - [Referencia de Configuración](#) - Todos los parámetros de configuración
- 

## Resumen

La Interfaz de Usuario Web de OmniSS7 proporciona gestión intuitiva y en tiempo real de tablas de enrutamiento y seguimiento de suscriptores:

- ◇ **Actualizaciones en Tiempo Real** - La auto-actualización mantiene los datos actualizados
- ◇ **Almacenamiento Persistente** - Mnesia asegura que las rutas sobrevivan a los reinicios
- ◇ **Interfaz Basada en Roles** - Las páginas se adaptan al modo operativo (STP/HLR/SMSc)
- ◇ **Gestión Interactiva** - Agregar, editar, eliminar rutas sin reinicio
- ◇ **Monitoreo de Estado** - Estado de conexión y de pares en vivo

Para operaciones avanzadas o automatización, consulte la [Guía API](#).