

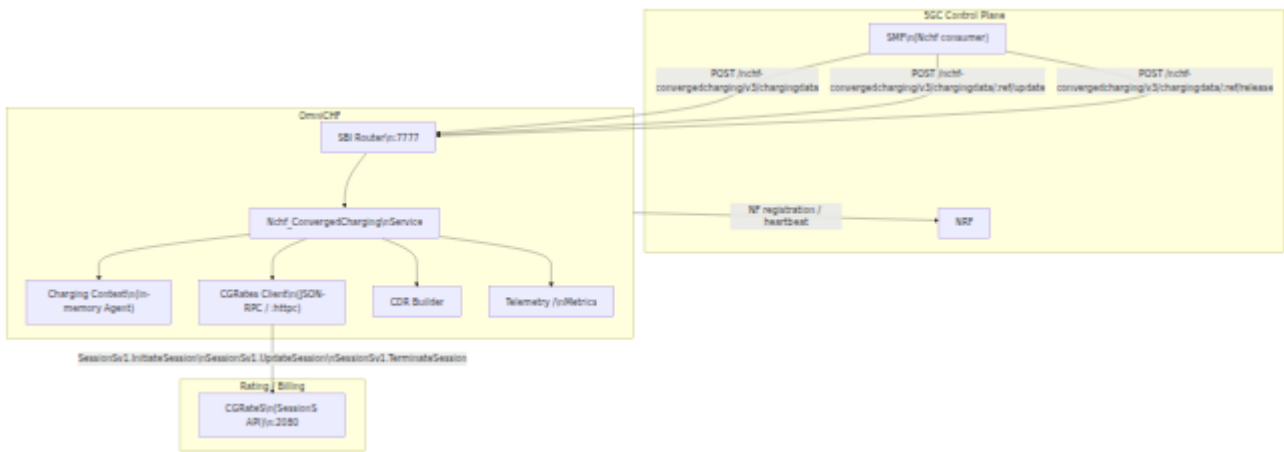
# Guía de Operaciones de OmniCHF

## Tabla de Contenidos

1. Descripción del Componente
  2. Rol de 3GPP y Referencias de Especificaciones
  3. Puntos Finales de SBI
  4. Referencia de Configuración
  5. Procedimientos Clave
  6. Observabilidad
  7. Limitaciones Conocidas
  8. Solución de Problemas
- 

## Descripción del Componente

OmniCHF implementa la función de red de Carga (CHF) definida en 3GPP TS 32.291. El CHF proporciona carga en línea y fuera de línea convergente para sesiones PDU 5G a través del servicio Nchf\_ConvergedCharging. Traduce solicitudes de carga 5G en llamadas JSON-RPC SessionS de CGRateS para autorización de crédito y gestión de sesiones, y genera Registros de Detalle de Llamadas (CDRs) al liberar la sesión.



## Ciclo de Vida de la Sesión de Carga

Cada sesión PDU se mapea a una sesión de carga, rastreada por un `chargingDataRef` (UUID). El estado de la sesión se mantiene en un Agente en memoria y no se persiste. Un reinicio pierde todo el estado de sesión activa.

Estado	Disparador	Acción de almacenamiento
Creado	POST /chargingdata	Contexto creado, se llamó a CGRateS InitiateSession
Actualizado	POST /chargingdata/:ref/update	Contexto actualizado (uso acumulado, secuencia incrementada)
Liberado	POST /chargingdata/:ref/release	CDR construido y registrado, se llamó a CGRateS TerminateSession, contexto eliminado

# Rol de 3GPP y Referencias de Especificaciones

Ítem	Referencia
Definición de NF CHF	3GPP TS 23.501 Sección 6.2.16
Servicio Nchf_ConvergedCharging	3GPP TS 32.291
Procedimiento de Creación de Datos de Carga	3GPP TS 32.291 Sección 6.1.3.2.1
Procedimiento de Actualización de Datos de Carga	3GPP TS 32.291 Sección 6.1.3.2.2
Procedimiento de Liberación de Datos de Carga	3GPP TS 32.291 Sección 6.1.3.2.3
Modelo de datos ChargingDataRequest / Response	3GPP TS 32.291 Sección 6.1.6
Formato de CDR para sesiones PDU 5G	3GPP TS 32.290
Marco común de SBI	3GPP TS 29.500
Registro de NF con NRF	3GPP TS 29.510

## Puntos Finales de SBI

Ruta base: `/nchf-convergedcharging/v3`

Método	Ruta	Descripción
POST	/chargingdata	<p>Crear una sesión de carga (Solicitud inicial). Asigna un <code>chargingDataRef</code>, inicia una sesión de CGRateS y devuelve unidades concedidas.</p>
POST	/chargingdata/{chargingDataRef}/update	<p>Actualizar una sesión de carga (Solicitud interina). Informa el uso actual y solicita crédito adicional.</p>
POST	/chargingdata/{chargingDataRef}/release	<p>Liberar una sesión de carga (Solicitud final). Informa el uso final, genera CDR, termina la sesión de CGRateS.</p>

## ChargingDataRequest – Campos Clave

Campo	Tipo	Usado en	Descripción
<code>subscriberIdentifier</code>	string	Crear, Actualizar, Liberar	SUPI (por ejemplo, <code>imsi-999700000000001</code> ). Usado como identificador de cu de CGRateS.
<code>nfConsumerIdentification</code>	object	Crear	Información del consumidor de NF. Fuente de respaldo para SUPI si <code>subscriberIdentifier</code> está ausente.
<code>pDU SessionChargingInformation</code>	object	Crear, Actualizar, Liberar	Detalles de la sesión PDU: DNN, S-NSSA tipo de RAT, QoS, II tipo de sesión PDU
<code>multipleUnitUsage</code>	array	Actualizar, Liberar	Contenedores de u reportados. El <code>usedUnitContaine</code> del primer element usa para la extracc de volumen y duración.
<code>requestType</code>	string	Todos	<code>INITIAL_REQUEST</code> , <code>UPDATE_REQUEST</code> , o <code>TERMINATION_REQUI</code>

## ChargingDataResponse – Campos Clave

Campo	Tipo	Presente	Descripción
<code>invocationSequenceNumber</code>	integer	Crear, Actualizar	Número de secuencia para esta respuesta. Codificado como <code>1</code> en Crear (ver CHF-M1). Incrementado en cada Actualización.
<code>invocationResult</code>	object	Crear, Actualizar	Siempre <code>{"resultCode": "SUCCESS"}</code> en el camino feliz.
<code>sessionId</code>	string	Crear, Actualizar	El <code>chargingDataRef</code> (UUID) asignado para esta sesión.
<code>multipleUnitInformation</code>	array	Crear, Actualizar	Unidades concedidas. Contiene una entrada con <code>resultCode</code> , <code>grantedUnit</code> (totalVolume, time), y <code>ratingGroup</code> (codificado como <code>1</code> , ver CHF-L2).

---

## Referencia de Configuración

Todos los parámetros se establecen a través del entorno de la aplicación (típicamente `config/runtime.exs`).

```
config :omnichf,  
  sbi_scheme:      "http",  
  sbi_addr:        "127.0.0.14",  
  sbi_port:        7777,  
  nrf_uri:         "http://127.0.0.10:7777",  
  mcc:             "999",  
  mnc:             "70",  
  heartbeat_interval: 10_000,  
  cgrates_enabled: false,  
  cgrates_url:     "http://localhost:2080/jsonrpc",  
  cgrates_tenant:  "cgrates.org",  
  cgrates_timeout: 5000
```

## Tabla de Parámetros

Parámetro	Predeterminado	Tipo	
sbi_scheme	"http"	string	Esquema de transporte
sbi_addr	"127.0.0.14"	string	Dirección de servidor HTTP
sbi_port	7777	integer	Puerto de escucha HTTP
nrf_uri	"http://127.0.0.10:7777"	string	URI utilizada para registrar la identidad
mcc	"999"	string	Código de país de la red de suscripción de usuario (NF e)
mnc	"70"	string	Código de red de suscripción de usuario (NF e)
heartbeat_interval	10_000	integer (ms)	Intervalo de solicitud de mantenimiento de la red (NRF)

Parámetro	Predeterminado	Tipo	
<code>cgrates_enabled</code>	<code>false</code>	boolean	Inter para CGR <code>false</code> toda CGR devu conc prec 86,4 Esta proc haya CGR
<code>cgrates_url</code>	<code>"http://localhost:2080/jsonrpc"</code>	string	URL JSON insta Solo <code>cgrates</code> <code>true</code>
<code>cgrates_tenant</code>	<code>"cgrates.org"</code>	string	Nom de C en t a la el ca Deb inqu en C
<code>cgrates_timeout</code>	<code>5000</code>	integer (ms)	Tien la sc las l de C

Parámetro	Predeterminado	Tipo	
			llam verit utiliz min 3000 bloq de s

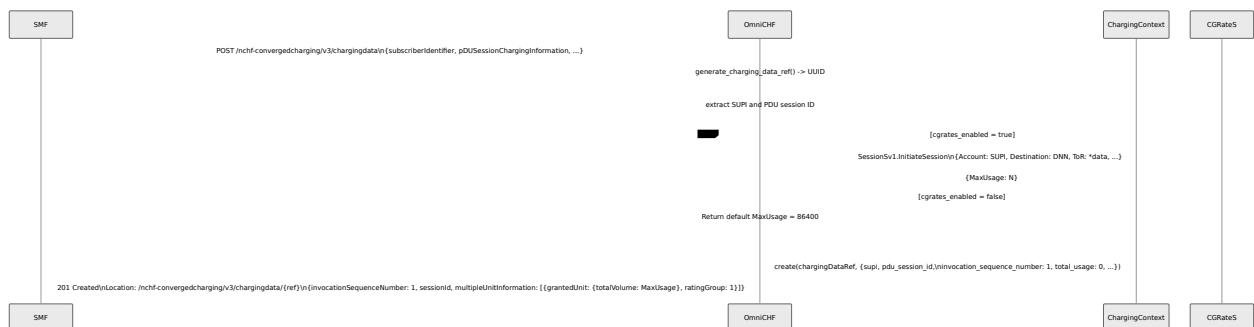
## Notas de Integración de CGRateS

Cuando `cgrates_enabled` es `false`, OmniCHF opera en **modo de bypass**: se aceptan todas las solicitudes de carga y se conceden 86,400 unidades (tiempo o volumen) sin ninguna calificación o autorización. Esto es adecuado para pruebas de laboratorio e integración cuando CGRateS no está disponible.

La comunicación de CGRateS utiliza el cliente HTTP `:httpc` incorporado de Erlang (ver limitación CHF-M5). Este cliente no admite agrupamiento de conexiones. Bajo alta carga, cada solicitud de CGRateS abre y cierra una nueva conexión HTTP, lo que puede convertirse en un cuello de botella.

## Procedimientos Clave

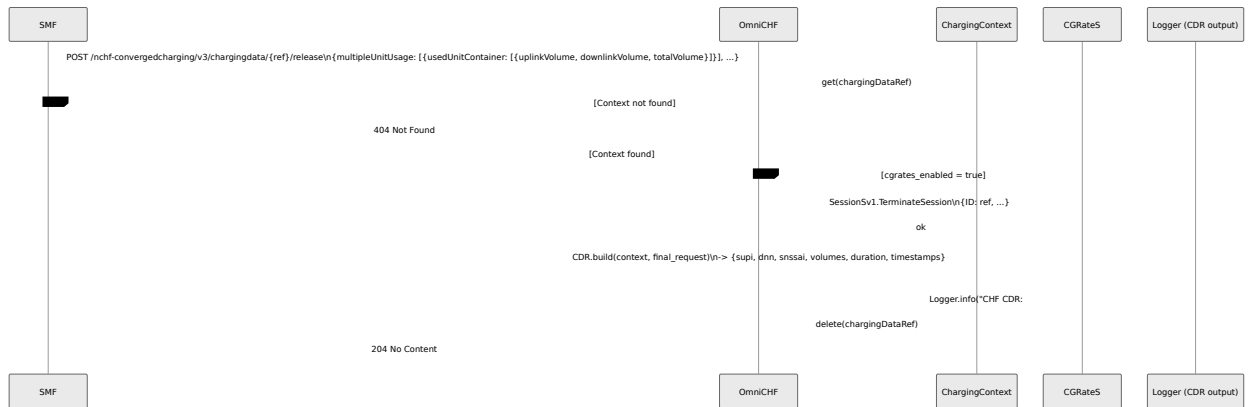
### Crear Sesión de Carga (Inicial)



# Actualizar Sesión de Carga (Interina)



# Liberar Sesión de Carga (Final)



# Mapeo de Eventos de CGRateS

OmniCHF mapea campos de carga 5G a campos de eventos SessionS de CGRateS de la siguiente manera:

<b>Campo de CGRateS</b>	<b>Fuente</b>	
Account	subscriberIdentifier (SUPI)	Recae en nfConsum
Subject	subscriberIdentifier (SUPI)	Igual que
Destination	pduSessionInformation.dnnId o .dnn	Nombre d
ToR	pduSessionInformation.pduType	Siempre e PDU
RequestType	requestType	Siempre r
Usage	usedUnitContainer.totalVolume o suma de uplink+downlink o time	El primer
OriginID	chargingDataRef	UUID únic
OriginHost	estático "OmniCHF"	
SUPI	subscriberIdentifier	Campo de
DNN	pduSessionInformation.dnnId	Campo de
S-NSSAI_SST	pduSessionInformation.sNSSAI.sst	Campo de
S-NSSAI_SD	pduSessionInformation.sNSSAI.sd	Campo de
5QI	pduSessionInformation.qoSInformation.5qI	Campo de
RATType	pduSessionInformation.ratType	Predeterm
PDUSessionID	pduSessionInformation.pduSessionID	Campo de
PDUSessionType	pduSessionInformation.pduType	Predeterm

## **Campos de CDR**

Los CDRs se construyen al liberar la sesión y se emiten al registro de la aplicación a nivel INFO. El mapa de CDR contiene:

<b>Campo</b>	<b>Fuente</b>
<code>record_type</code>	Estático: <code>"5G_PDU_SESSION"</code>
<code>supi</code>	Contexto de carga
<code>dnn</code>	<code>pduSessionInformation.dnnId</code> o <code>.dnn</code>
<code>snsai</code>	{sst, sd} de <code>pduSessionInformation.sNSSAI</code>
<code>qos_5qi</code>	<code>pduSessionInformation.qoSInformation.5qi</code>
<code>rat_type</code>	<code>pduSessionInformation.ratType</code>
<code>pdu_session_id</code>	Contexto de carga
<code>pdu_session_type</code>	<code>pduSessionInformation.pduType</code>
<code>volume_uplink</code>	<code>usedUnitContainer.uplinkVolume</code>
<code>volume_downlink</code>	<code>usedUnitContainer.downlinkVolume</code>
<code>volume_total</code>	<code>usedUnitContainer.totalVolume</code> (o uplink+downlink)
<code>duration</code>	<code>usedUnitContainer.time</code> (o diferencia de reloj si es cero)
<code>start_time</code>	Marca de tiempo <code>created_at</code> de la sesión
<code>end_time</code>	Reloj de pared en el momento de la liberación
<code>charging_data_ref</code>	UUID de la sesión

# Observabilidad

## Eventos de Telemetría

Evento	Mediciones	Etiquetas	Descripción
<code>[ :omnichf, :charging, :initial]</code>	<code>count</code>	<code>supi</code>	Disparado en cada solicitud de Creación
<code>[ :omnichf, :charging, :update]</code>	<code>count</code>	<code>ref</code>	Disparado en cada solicitud de Actualización
<code>[ :omnichf, :charging, :release]</code>	<code>count</code>	<code>ref</code>	Disparado en cada solicitud de Liberación
<code>[ :omnichf, :charging, :creates]</code>	<code>count</code>	<code>result (success/failure)</code>	Resultado de la operación de creación
<code>[ :omnichf, :charging, :updates]</code>	<code>count</code>	<code>result</code>	Resultado de la operación de actualización
<code>[ :omnichf, :charging, :releases]</code>	<code>count</code>	<code>result</code>	Resultado de la operación de liberación
<code>[ :omnichf, :cgrates, :request]</code>	<code>count, duration_ms</code>	<code>operation, result</code>	Por cada llamada JSON-RPC de CGRateS

Evento	Mediciones	Etiquetas	Descripción
<code>[ :omnichf, :cgrates, :health]</code>	<code>status</code> (1/0)	—	Salud de conectividad de CGRateS
<code>[ :omnichf, :sessions, :active]</code>	<code>count</code>	—	Medida: sesiones de carga activas
<code>[ :omni5g, :nrf, :registration]</code>	<code>status</code> (1/0)	<code>nf_type</code>	Estado de registro de NRF

## Métricas de Prometheus

### Métricas de Carga

Métrica	Tipo	Etiquetas	Descripción
<code>omni_chf.charging.initial.count</code>	contador	supi	Creaciones de sesiones de carga
<code>omni_chf.charging.update.count</code>	contador	ref	Actualizaciones de sesiones de carga
<code>omni_chf.charging.release.count</code>	contador	ref	Liberaciones de sesiones de carga
<code>omni_chf.charging.create.total</code>	contador	result	Total de creaciones de sesiones de carga
<code>omni_chf.charging.update.total</code>	contador	result	Total de actualizaciones de sesiones de carga
<code>omni_chf.charging.release.total</code>	contador	result	Total de liberaciones de sesiones de carga
<code>omni_chf.sessions.active.count</code>	medidor	--	Número de sesiones de carga activas

## Métricas de CGRateS

Métrica	Tipo	Etiquetas	De
<code>omni_chf.cgrates.calls.count</code>	contador	<code>method,</code> <code>result</code>	Lla JSO CGI
<code>omni_chf.cgrates.latency.milliseconds</code>	medidor	--	Lat llar CGI
<code>omni_chf.cgrates.health</code>	medidor	--	Sal cor de (1= 0=
<code>omni_chf.cgrates.requests.total</code>	contador	<code>operation,</code> <code>result</code>	Tot soli JSO CGI
<code>omni_chf.cgrates.request.duration_ms</code>	distribución	<code>operation</code>	Dur la s CGI ms 5, 1 50, 250 100

## Métricas de NRF

Métrica	Tipo	Etiquetas	Descripción
omni_chf.nrf.registration.status	medidor	nf_type	Estado de registro de NRF (1=registrado 0=no)

## Métricas de BEAM VM

<b>Métrica</b>	<b>Tipo</b>	<b>Descripción</b>
<code>beam.memory.total</code>	medidor	Memoria total de BEAM en bytes
<code>beam.memory.processes</code>	medidor	Memoria utilizada por procesos de Erlang
<code>beam.memory.processes_used</code>	medidor	Memoria realmente utilizada por procesos
<code>beam.memory.system</code>	medidor	Memoria del sistema
<code>beam.memory.atom</code>	medidor	Memoria total de átomos
<code>beam.memory.atom_used</code>	medidor	Memoria de átomos utilizada
<code>beam.memory.binary</code>	medidor	Memoria binaria
<code>beam.memory.code</code>	medidor	Memoria de código
<code>beam.memory.ets</code>	medidor	Memoria de tabla ETS
<code>beam.processes.count</code>	medidor	Número de procesos de Erlang
<code>beam.ports.count</code>	medidor	Número de puertos de Erlang
<code>beam.atom.count</code>	medidor	Número de átomos
<code>beam.vm.uptime</code>	medidor	Tiempo de actividad de VM en segundos

# Patrones de Registro

Nivel	Patrón	Significado
info	CHF Create: ref=<UUID> supi=<SUPI> pdu_session=<N>	Creación exitosa iniciada
info	CHF Update: ref=<UUID>	Solicitud de actualización recibida
info	CHF Release: ref=<UUID>	Solicitud de liberación recibida
info	CHF CDR: %{...}	CDR emitido en la liberación
info	Initiating CGRateS session for <ref>, account: <SUPI>	CGRateS InitiateSession enviado
info	CGRateS authorized <N> units for session <ref>	Crédito concedido
info	CGRateS session <ref> terminated successfully	CGRateS TerminateSession OK
warning	CGRateS integration disabled, returning default authorization	Modo de bypass activo
warning	CHF Update: unknown ref=<UUID>	Actualización para sesión no existente
warning	CHF Release: unknown ref=<UUID>	Liberación para sesión no existente
error	CHF Create failed: <reason>	Fallo en la operación de creación

<b>Nivel</b>	<b>Patrón</b>	<b>Significado</b>
error	CHF Update failed: <reason>	Fallo en la operación de actualización
error	CGRateS InitiateSession failed for <ref>: <reason>	Error de CGRateS en la creación
error	CGRateS HTTP error <status>: <body>	No-200 de CGRateS
error	CGRateS HTTP request failed: <reason>	Error de red a CGRateS

---

# Limitaciones Conocidas

ID	Severidad	Descripción
CHF-M1	Media	<code>invocationSequenceNumber</code> está codificado como <code>1</code> en la respuesta de Crear (Inicial). Según TS 32.291, el número de secuencia debe comenzar en 1 e incrementarse en respuestas posteriores, lo que hace en Actualización. El problema es que si un consumidor envía un Crear con un <code>invocationSequenceNumber</code> en la solicitud, ese valor no se inspecciona ni valida.
CHF-M3	Media	<code>invocationTimeStamp</code> está ausente de <code>ChargingDataResponse</code> . Según TS 32.291, este campo es obligatorio en el cuerpo de la respuesta. Consumidores estrictos que requieren este campo recibirán una respuesta incompleta.
CHF-M5	Media	El cliente de CGRateS utiliza el cliente HTTP <code>:httpc</code> de Erlang en lugar de Finch. <code>:httpc</code> no admite agrupamiento de conexiones; cada llamada JSON-RPC abre y cierra una conexión TCP. Bajo carga (muchas sesiones de carga concurrentes), la latencia de llamada a CGRateS aumentará y la sobrecarga de configuración de conexión se vuelve significativa. Monitorear <code>omni_chf.cgrates.request.duration_ms</code> .
CHF-L1	Baja	No se incluye un campo <code>triggers</code> en <code>ChargingDataResponse</code> . Según TS 32.291, los disparadores pueden instruir al SMF para que envíe una actualización interina en eventos específicos (umbral de volumen, umbral de tiempo, cambio de QoS). Sin disparadores, el SMF utiliza su propia política local para determinar cuándo enviar actualizaciones.

ID	Severidad	Descripción
CHF-L2	Baja	<p><code>ratingGroup</code> en <code>multipleUnitInformation</code> está codificado como 1. Las implementaciones reales suelen tener múltiples grupos de calificación por sesión PDU (uno por flujo de datos de servicio). Todo el uso se atribuye al grupo de calificación 1 independientemente de los valores de <code>ratingGroup</code> en el <code>multipleUnitUsage</code> de la solicitud.</p>
CHF-L3	Baja	<p>No se genera un <code>chargingId</code> en el formato 3GPP. Según TS 32.290, los registros de carga deben llevar un <code>chargingId</code> que correlacione con el ID de sesión PDU asignado por el SMF. En su lugar se utiliza el UUID <code>charging_data_ref</code>, lo que puede causar problemas de correlación en sistemas de facturación posteriores que esperan el formato de <code>chargingId</code> 3GPP.</p>
CHF-L4	Baja	<p>El registro CDR carece de los campos <code>chargingID</code> y <code>recordingEntity</code> requeridos por TS 32.290. Los sistemas de mediación o facturación posteriores que esperan estos campos deberán tolerar su ausencia o configurarse para tratarlos como opcionales.</p>
CHF-L5	Baja	<p>La carga fuera de línea y la salida de archivos CDR no están implementadas. Los CDRs se emiten solo al registro de la aplicación a través de <code>Logger.info</code>. No hay salida de CDR basada en archivos, no hay codificación ASN.1 y no hay transferencia a una puerta de enlace de dominio de facturación. Para la carga fuera de línea en producción, un transportador de registros (por ejemplo, Fluentd, Vector) debe recopilar las líneas de registro CDR de CHF y transformarlas para el sistema de facturación.</p>

# Solución de Problemas

## 404 en Actualización o Liberación

El `chargingDataRef` no coincide con ninguna sesión activa en memoria.

Causas:

1. OmniCHF se reinició entre Crear y Actualizar/Liberar — todo el estado de la sesión está en memoria y se pierde en el reinicio.
2. El SMF envió el `chargingDataRef` incorrecto en la ruta.
3. Se envió previamente una Liberación para esta sesión, lo que eliminó el contexto.

Verifique los registros para `CHF Update: unknown ref=` o `CHF Release: unknown ref=` para confirmar.

## 500 en Crear con CGRateS habilitado

La llamada a CGRateS falló. Verifique:

1. ¿Está `cgrates_url` apuntando a una instancia de CGRateS accesible?
2. ¿Es correcto `cgrates_tenant`? Nombres de inquilinos desajustados hacen que CGRateS devuelva una respuesta de error.
3. Verifique la métrica `omni_chf.cgrates.health` (1=arriba, 0=abajo).
4. Revise los registros para `CGRateS InitiateSession failed for <ref>: <reason>` — la razón será una de: `{:cgrates_error, message}`, `{:http_error, status}`, o `{:http_error, reason}`.
5. Verifique que el servicio `SessionSv1` de CGRateS esté habilitado en la configuración de CGRateS.

## Fallo en la verificación de salud de CGRateS pero el demonio está en ejecución

La verificación de salud (a través de `SessionSv1.GetActiveSessions`) utiliza un tiempo de espera máximo de 3 segundos. Si CGRateS responde lentamente, la verificación de salud puede fallar mientras el servicio está técnicamente

disponible. Verifique `cgrates_timeout` — el límite es `min(cgrates_timeout, 3000)`. También confirme que `cgrates_url` use HTTP (no HTTPS) a menos que TLS esté configurado.

## CDRs no aparecen o están incompletos

Los CDRs se escriben en el registro de la aplicación a nivel INFO (ver limitación CHF-L5). Para capturar CDRs:

1. Asegúrese de que el nivel de registro de la aplicación esté configurado en `info` o inferior.
2. Filtre las líneas de registro que contengan `"CHF CDR:"` para su posterior procesamiento.
3. Tenga en cuenta que los CDRs carecen de `chargingID`, `recordingEntity` (CHF-L4), y tendrán `ratingGroup: 1` para todos los flujos de datos de servicio (CHF-L2).

## Alta latencia en llamadas a CGRateS

Debido a que CGRateS utiliza `:httpc` sin agrupamiento de conexiones (CHF-M5), la latencia aumenta bajo carga. Para diagnosticar:

1. Verifique el histograma `omni_chf.cgrates.request.duration_ms` para la latencia p99.
2. Si la latencia es alta bajo carga concurrente, reduzca el número de sesiones de carga concurrentes o considere implementar OmniCHF detrás de un equilibrador de carga con múltiples instancias.
3. Como solución alternativa, establezca `cgrates_timeout` en un valor inferior al tiempo de respuesta peor caso esperado de CGRateS para evitar que las llamadas lentas a CGRateS bloqueen el grupo de procesos de Elixir.

## El recuento de sesiones activas no disminuye después de las liberaciones

Si `omni_chf.sessions.active.count` permanece elevado después de que las sesiones deberían haber sido liberadas:

1. Verifique las respuestas 404 en las llamadas de Liberación — si el SMF recibe 404, puede que no reintente y el SMF considera que la sesión ha sido liberada mientras que OmniCHF aún puede tener el contexto.
2. En el caso inverso, si OmniCHF se reinició y perdió el estado de la sesión, los contextos se han ido, pero el SMF aún puede enviar solicitudes de Actualización/Liberación que resulten en 404. Este es un comportamiento esperado.

## Registro de NRF no mantenido

Verifique la métrica `omni_chf.nrf.registration.status`. Si muestra 0:

1. Verifique que `nrf_uri` sea correcto y que el NRF sea accesible desde `sbi_addr` de OmniCHF.
2. Verifique que `mcc` y `mnc` coincidan con la configuración PLMN del NRF.
3. Revise los registros de la aplicación al inicio para errores de registro de NRF.