

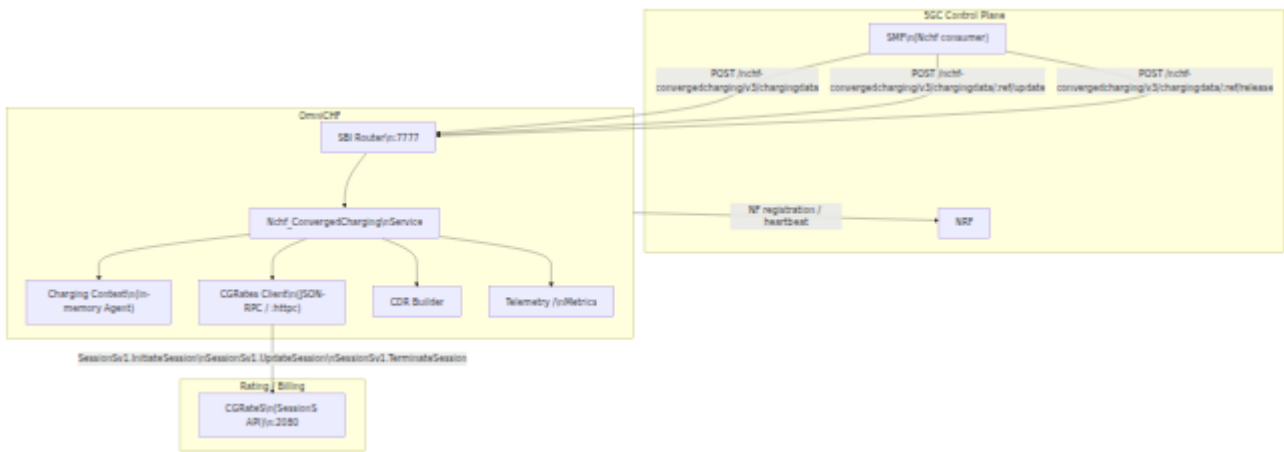
# Guia de Operações do OmniCHF

## Índice

1. [Visão Geral do Componente](#)
  2. [Referências de Papel e Especificação 3GPP](#)
  3. [Endpoints SBI](#)
  4. [Referência de Configuração](#)
  5. [Procedimentos Chave](#)
  6. [Observabilidade](#)
  7. [Limitações Conhecidas](#)
  8. [Solução de Problemas](#)
- 

## Visão Geral do Componente

OmniCHF implementa a função de rede de Função de Cobrança (CHF) definida na 3GPP TS 32.291. O CHF fornece cobrança online e offline convergente para sessões PDU 5G via o serviço Nchf\_ConvergedCharging. Ele traduz solicitações de cobrança 5G em chamadas JSON-RPC SessionS CGRateS para autorização de crédito e gerenciamento de sessão, e gera Registros de Detalhes de Chamadas (CDRs) na liberação da sessão.



## Ciclo de Vida da Sessão de Cobrança

Cada sessão PDU mapeia para uma sessão de cobrança, rastreada por um `chargingDataRef` (UUID). O estado da sessão é mantido em um Agente em memória e não é persistido. Um reinício perde todo o estado da sessão ativa.

Estado	Gatilho	Ação de armazenamento
Criado	POST /chargingdata	Contexto criado, CGRateS InitiateSession chamado
Atualizado	POST /chargingdata/:ref/update	Contexto atualizado (uso acumulado, sequência incrementada)
Liberado	POST /chargingdata/:ref/release	CDR construído e registrado, CGRateS TerminateSession chamado, contexto excluído

# Referências de Papel e Especificação 3GPP

Item	Referência
Definição de NF CHF	3GPP TS 23.501 Seção 6.2.16
Serviço Nchf_ConvergedCharging	3GPP TS 32.291
Procedimento de Criação de Dados de Cobrança	3GPP TS 32.291 Seção 6.1.3.2.1
Procedimento de Atualização de Dados de Cobrança	3GPP TS 32.291 Seção 6.1.3.2.2
Procedimento de Liberação de Dados de Cobrança	3GPP TS 32.291 Seção 6.1.3.2.3
Modelo de dados ChargingDataRequest / Response	3GPP TS 32.291 Seção 6.1.6
Formato CDR para sessões PDU 5G	3GPP TS 32.290
Estrutura comum do SBI	3GPP TS 29.500
Registro de NF com NRF	3GPP TS 29.510

## Endpoints SBI

Caminho base: `/nchf-convergedcharging/v3`

<b>Método</b>	<b>Caminho</b>	<b>Descrição</b>
POST	/chargingdata	Criar uma sessão de cobrança (Solicitação inicial). Aloca um <code>chargingDataRef</code> , inicia uma sessão CGRateS e retorna unidades concedidas.
POST	/chargingdata/{chargingDataRef}/update	Atualizar uma sessão de cobrança (Solicitação intermediária). Relata o uso atual e solicita crédito adicional.
POST	/chargingdata/{chargingDataRef}/release	Liberar uma sessão de cobrança (Solicitação final). Relata o uso final, gera CDR, termina a sessão CGRateS.

## ChargingDataRequest – Campos Chave

Campo	Tipo	Usado em	Descrição
<code>subscriberIdentifier</code>	string	Criar, Atualizar, Liberar	SUPI (por exemplo, <code>imsi-999700000000001</code> ). Usado como identificador de con CGRateS.
<code>nfConsumerIdentification</code>	objeto	Criar	Informações do consumidor NF. Fonte de fallback para SUPI se <code>subscriberIdentifier</code> estiver ausente.
<code>pDUSessionChargingInformation</code>	objeto	Criar, Atualizar, Liberar	Detalhes da sessão PDU: DNN, S-NSSAI, tipo RAT, QoS, ID e t da sessão PDU.
<code>multipleUnitUsage</code>	array	Atualizar, Liberar	Contêineres de uso relatados. O <code>usedUnitContainer</code> primeiro elemento é usado para extração de volume e duração.
<code>requestType</code>	string	Todos	<code>INITIAL_REQUEST</code> , <code>UPDATE_REQUEST</code> ou <code>TERMINATION_REQUE</code>

## ChargingDataResponse – Campos Chave

Campo	Tipo	Presente	Descrição
<code>invocationSequenceNumber</code>	integer	Criar, Atualizar	Número de sequência para esta resposta. Codificado como <code>1</code> na Criação (veja CHF-M1). Incrementado em cada Atualização.
<code>invocationResult</code>	objeto	Criar, Atualizar	Sempre <code>{"resultCode": "SUCCESS"}</code> no caminho feliz.
<code>sessionId</code>	string	Criar, Atualizar	O <code>chargingDataRef</code> (UUID) alocado para esta sessão.
<code>multipleUnitInformation</code>	array	Criar, Atualizar	Unidades concedidas. Contém uma entrada com <code>resultCode</code> , <code>grantedUnit</code> (totalVolume, time) e <code>ratingGroup</code> (codificado como <code>1</code> , veja CHF-L2).

---

## Referência de Configuração

Todos os parâmetros são definidos via o ambiente da aplicação (tipicamente `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
```

# Tabela de Parâmetros

Parâmetro	Padrão	Tipo	
sbi_scheme	"http"	string	Esquema de transporte
sbi_addr	"127.0.0.14"	string	Endereço de serviço vinculado
sbi_port	7777	integer	Porta de serviço escuta
nrf_uri	"http://127.0.0.10:7777"	string	URI de usuário NFE
mcc	"999"	string	Código de usuário em nome de serviço
mnc	"70"	string	Código de usuário em nome de serviço
heartbeat_interval	10_000	integer (ms)	Intervalo de solicitação de heartbeat

Parâmetro	Padrão	Tipo	
<code>cgrates_enabled</code>	<code>false</code>	boolean	Inter para CGR <code>false</code> char são conc 86.4 reto com proc instá estiv
<code>cgrates_url</code>	<code>"http://localhost:2080/jsonrpc"</code>	string	URL RPC CGR ape <code>cgrates</code> <code>true</code>
<code>cgrates_tenant</code>	<code>"cgrates.org"</code>	string	Non CGR toda API : cam corr inqu no C
<code>cgrates_timeout</code>	<code>5000</code>	integer (ms)	Tem solic char do C Char

Parâmetro	Padrão	Tipo	
			verifi usar min 3000 bloq de s

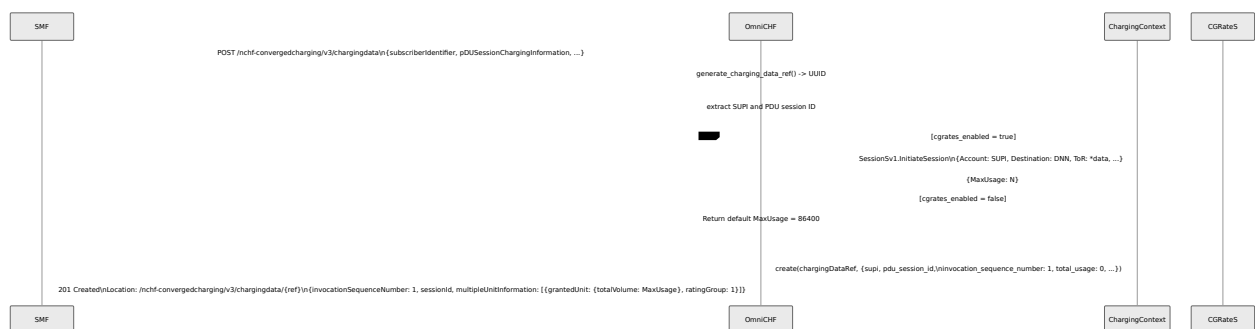
## Notas de Integração do CGRateS

Quando `cgrates_enabled` é `false`, o OmniCHF opera em **modo de bypass**: todas as solicitações de cobrança são aceitas e concedidas 86.400 unidades (tempo ou volume) sem qualquer classificação ou autorização. Isso é adequado para testes em laboratório e integração quando o CGRateS não está disponível.

A comunicação com o CGRateS utiliza o cliente HTTP embutido `:httpc` do Erlang (veja a limitação CHF-M5). Este cliente não suporta pooling de conexão. Sob alta carga, cada solicitação CGRateS abre e fecha uma nova conexão HTTP, o que pode se tornar um gargalo.

## Procedimentos Chave

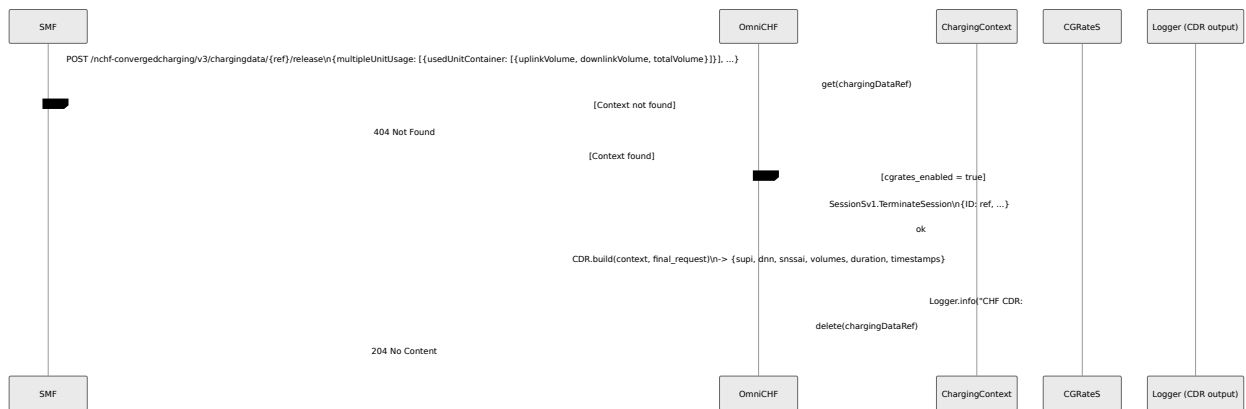
### Criar Sessão de Cobrança (Inicial)



# Atualizar Sessão de Cobrança (Intermediária)



# Liberar Sessão de Cobrança (Final)



# Mapeamento de Eventos do CGRateS

OmniCHF mapeia campos de cobrança 5G para campos de eventos SessionS do CGRateS da seguinte forma:

<b>Campo CGRateS</b>	<b>Fonte</b>	
Account	subscriberIdentifier (SUPI)	Reverte p nfConsum
Subject	subscriberIdentifier (SUPI)	O mesmo
Destination	pduSessionInformation.dnnId ou .dnn	Nome da
ToR	pduSessionInformation.pduType	Sempre *
RequestType	requestType	Sempre n
Usage	usedUnitContainer.totalVolume ou soma de uplink+downlink ou time	O primeir
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	Padrão: "
PDUSessionID	pduSessionInformation.pduSessionID	Campo de
PDUSessionType	pduSessionInformation.pduType	Padrão: "

## **Campos CDR**

Os CDRs são construídos na liberação da sessão e emitidos para o log da aplicação no nível INFO. O mapa CDR contém:

<b>Campo</b>	<b>Fonte</b>
<code>record_type</code>	Estático: <code>"5G_PDU_SESSION"</code>
<code>supi</code>	Contexto de cobrança
<code>dnn</code>	<code>pduSessionInformation.dnnId</code> ou <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 cobrança
<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> (ou uplink+downlink)
<code>duration</code>	<code>usedUnitContainer.time</code> (ou diferença de relógio se zero)
<code>start_time</code>	Timestamp <code>created_at</code> da sessão
<code>end_time</code>	Relógio de parede no momento da liberação
<code>charging_data_ref</code>	UUID da sessão

# Observabilidade

## Eventos de Telemetria

Evento	Medidas	Tags	Descrição
<code>[ :omnichf, :charging, :initial ]</code>	<code>count</code>	<code>supi</code>	Disparado em cada solicitação de Criação
<code>[ :omnichf, :charging, :update ]</code>	<code>count</code>	<code>ref</code>	Disparado em cada solicitação de Atualização
<code>[ :omnichf, :charging, :release ]</code>	<code>count</code>	<code>ref</code>	Disparado em cada solicitação de Liberação
<code>[ :omnichf, :charging, :creates ]</code>	<code>count</code>	<code>result (success/failure)</code>	Resultado da operação de Criação
<code>[ :omnichf, :charging, :updates ]</code>	<code>count</code>	<code>result</code>	Resultado da operação de Atualização
<code>[ :omnichf, :charging, :releases ]</code>	<code>count</code>	<code>result</code>	Resultado da operação de Liberação
<code>[ :omnichf, :cgrates, :request ]</code>	<code>count, duration_ms</code>	<code>operation, result</code>	Por chamada JSON-RPC do CGRateS
<code>[ :omnichf, :cgrates, ]</code>	<code>status (1/0)</code>	—	Saúde da conectividade

Evento	Medidas	Tags	Descrição
<code>:health]</code>			do CGRateS
<code>[:omnichf, :sessions, :active]</code>	<code>count</code>	—	Medida: sessões de cobrança ativas
<code>[:omni5g, :nrf, :registration]</code>	<code>status (1/0)</code>	<code>nf_type</code>	Status de registro do NRF

## Métricas do Prometheus

### Métricas de Cobrança

Métrica	Tipo	Tags	Descrição
<code>omni_chf.charging.initial.count</code>	contador	supi	Criações de sessão de cobrança
<code>omni_chf.charging.update.count</code>	contador	ref	Atualizações de sessão de cobrança
<code>omni_chf.charging.release.count</code>	contador	ref	Liberações de sessão de cobrança
<code>omni_chf.charging.create.total</code>	contador	result	Total de criações de sessão de cobrança
<code>omni_chf.charging.update.total</code>	contador	result	Total de atualizações de sessão de cobrança
<code>omni_chf.charging.release.total</code>	contador	result	Total de liberações de sessão de cobrança
<code>omni_chf.sessions.active.count</code>	medidor	--	Número de sessões de cobrança ativas

## Métricas do CGRateS

Métrica	Tipo	Tags	D
<code>omni_chf.cgrates.calls.count</code>	contador	<code>method,</code> <code>result</code>	Ch JSO CGI
<code>omni_chf.cgrates.latency.milliseconds</code>	medidor	--	Lat cha CGI
<code>omni_chf.cgrates.health</code>	medidor	--	Saú cor do (1= 0=i
<code>omni_chf.cgrates.requests.total</code>	contador	<code>operation,</code> <code>result</code>	Tota soli JSO CGI
<code>omni_chf.cgrates.request.duration_ms</code>	distribuição	<code>operation</code>	Dur soli CGI ms 5, 1 100 500 250

## Métricas do NRF

Métrica	Tipo	Tags	Descrição
<code>omni_chf.nrf.registration.status</code>	medidor	<code>nf_type</code>	Status de registro do NRF (1=registrado, 0=não registrado)

### Métricas da VM BEAM

<b>Métrica</b>	<b>Tipo</b>	<b>Descrição</b>
<code>beam.memory.total</code>	medidor	Total de memória BEAM em bytes
<code>beam.memory.processes</code>	medidor	Memória usada por processos Erlang
<code>beam.memory.processes_used</code>	medidor	Memória realmente usada por processos
<code>beam.memory.system</code>	medidor	Memória do sistema
<code>beam.memory.atom</code>	medidor	Total de memória de átomos
<code>beam.memory.atom_used</code>	medidor	Memória de átomos usada
<code>beam.memory.binary</code>	medidor	Memória binária
<code>beam.memory.code</code>	medidor	Memória de código
<code>beam.memory.ets</code>	medidor	Memória da tabela ETS
<code>beam.processes.count</code>	medidor	Número de processos Erlang
<code>beam.ports.count</code>	medidor	Número de portas Erlang
<code>beam.atom.count</code>	medidor	Número de átomos
<code>beam.vm.uptime</code>	medidor	Tempo de atividade da VM em segundos

## Padrões de Log

Nível	Padrão	Significado
info	CHF Create: ref=<UUID> supi=<SUPI> pdu_session=<N>	Criação bem-sucedida iniciada
info	CHF Update: ref=<UUID>	Solicitação de atualização recebida
info	CHF Release: ref=<UUID>	Solicitação de liberação recebida
info	CHF CDR: %{...}	CDR emitido na liberação
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 ativo
warning	CHF Update: unknown ref=<UUID>	Atualização para sessão inexistente
warning	CHF Release: unknown ref=<UUID>	Liberação para sessão inexistente
error	CHF Create failed: <reason>	Falha na operação de criação

<b>Nível</b>	<b>Padrão</b>	<b>Significado</b>
error	CHF Update failed: <reason>	Falha na operação de atualização
error	CGRateS InitiateSession failed for <ref>: <reason>	Erro do CGRateS na criação
error	CGRateS HTTP error <status>: <body>	Não-200 do CGRateS
error	CGRateS HTTP request failed: <reason>	Erro de rede para o CGRateS

---

# Limitações Conhecidas

ID	Severidade	Descrição
CHF-M1	Médio	<code>invocationSequenceNumber</code> é codificado como <code>1</code> na resposta de Criação (Inicial). De acordo com a TS 32.291, o número de sequência deve começar em 1 e ser incrementado nas respostas subsequentes, o que acontece na Atualização. O problema é que, se um consumidor enviar uma Criação com um <code>invocationSequenceNumber</code> na solicitação, esse valor não é inspecionado ou validado.
CHF-M3	Médio	<code>invocationTimeStamp</code> está ausente do <code>ChargingDataResponse</code> . De acordo com a TS 32.291, este campo é obrigatório no corpo da resposta. Consumidores rigorosos que exigem este campo receberão uma resposta incompleta.
CHF-M5	Médio	O cliente CGRateS usa o cliente HTTP <code>:httpc</code> do Erlang em vez do Finch. <code>:httpc</code> não suporta pooling de conexão; cada chamada JSON-RPC abre e fecha uma conexão TCP. Sob carga (muitas sessões de cobrança simultâneas), a latência da chamada CGRateS aumentará e a sobrecarga de configuração da conexão se tornará significativa. Monitore <code>omni_chf.cgrates.request.duration_ms</code> .
CHF-L1	Baixo	Nenhum campo <code>triggers</code> é incluído no <code>ChargingDataResponse</code> . De acordo com a TS 32.291, os gatilhos podem instruir o SMF a enviar uma atualização intermediária em eventos específicos (limite de volume, limite de tempo, mudança de QoS). Sem gatilhos, o SMF usa sua própria política local para determinar quando enviar atualizações.

ID	Severidade	Descrição
CHF-L2	Baixo	<p><code>ratingGroup</code> em <code>multipleUnitInformation</code> é codificado como 1. Implementações reais normalmente têm vários grupos de classificação por sessão PDU (um por fluxo de dados de serviço). Todo uso é atribuído ao grupo de classificação 1, independentemente dos valores de <code>ratingGroup</code> nos <code>multipleUnitUsage</code> da solicitação.</p>
CHF-L3	Baixo	<p>Nenhum <code>chargingId</code> no formato 3GPP é gerado. De acordo com a TS 32.290, os registros de cobrança devem carregar um <code>chargingId</code> que correlacione com o ID da sessão PDU atribuído pelo SMF. O UUID <code>charging_data_ref</code> é usado em vez disso, o que pode causar problemas de correlação em sistemas de faturamento downstream que esperam o formato <code>chargingId</code> 3GPP.</p>
CHF-L4	Baixo	<p>O registro CDR está faltando os campos <code>chargingID</code> e <code>recordingEntity</code> exigidos pela TS 32.290. Sistemas de mediação ou faturamento downstream que esperam esses campos precisarão tolerar sua ausência ou ser configurados para tratá-los como opcionais.</p>
CHF-L5	Baixo	<p>A cobrança offline e a saída de arquivo CDR não estão implementadas. Os CDRs são emitidos apenas para o log da aplicação via <code>Logger.info</code>. Não há saída de CDR baseada em arquivo, nenhuma codificação ASN.1 e nenhuma transferência para um gateway de domínio de faturamento. Para cobrança offline em produção, um coletor de logs (por exemplo, Fluentd, Vector) deve coletar as linhas de log CDR do CHF e transformá-las para o sistema de faturamento.</p>

# Solução de Problemas

## 404 na Atualização ou Liberação

O `chargingDataRef` não corresponde a nenhuma sessão ativa em memória.

Causas:

1. OmniCHF reiniciado entre Criação e Atualização/Liberar — todo o estado da sessão está em memória e é perdido no reinício.
2. O SMF enviou o `chargingDataRef` errado no caminho.
3. Uma Liberação foi enviada anteriormente para esta sessão, o que excluiu o contexto.

Verifique os logs para `CHF Update: unknown ref=` ou `CHF Release: unknown ref=` para confirmar.

## 500 na Criação com CGRateS habilitado

A chamada CGRateS falhou. Verifique:

1. O `cgrates_url` está apontando para uma instância CGRateS acessível?
2. O `cgrates_tenant` está correto? Nomes de inquilinos incompatíveis fazem com que o CGRateS retorne uma resposta de erro.
3. Verifique a métrica `omni_chf.cgrates.health` (1=ativo, 0=inativo).
4. Revise os logs para `CGRateS InitiateSession failed for <ref>: <reason>` — a razão será uma das: `{:cgrates_error, message}`, `{:http_error, status}`, ou `{:http_error, reason}`.
5. Verifique se o serviço `SessionSv1` do CGRateS está habilitado na configuração do CGRateS.

## Verificação de saúde do CGRateS falhando, mas o daemon está em execução

A verificação de saúde (via `SessionSv1.GetActiveSessions`) usa um tempo limite de 3 segundos. Se o CGRateS estiver lento para responder, a verificação de saúde pode falhar enquanto o serviço está tecnicamente disponível.

Verifique `cgrates_timeout` — o limite é `min(cgrates_timeout, 3000)`. Também confirme se o `cgrates_url` usa HTTP (não HTTPS) a menos que TLS esteja configurado.

## CDRs não aparecendo ou incompletos

Os CDRs são escritos no log da aplicação no nível INFO (veja a limitação CHF-L5). Para capturar CDRs:

1. Certifique-se de que o nível de log da aplicação esteja definido como `info` ou inferior.
2. Filtre as linhas de log contendo `"CHF CDR:"` para processamento posterior.
3. Observe que os CDRs estão faltando `chargingID`, `recordingEntity` (CHF-L4), e terão `ratingGroup: 1` para todos os fluxos de dados de serviço (CHF-L2).

## Alta latência de chamadas do CGRateS

Como o CGRateS usa `:httpc` sem pooling de conexão (CHF-M5), a latência aumenta sob carga. Para diagnosticar:

1. Verifique o histograma `omni_chf.cgrates.request.duration_ms` para latência p99.
2. Se a latência for alta sob carga concorrente, reduza o número de sessões de cobrança concorrentes ou considere implantar o OmniCHF atrás de um balanceador de carga com várias instâncias.
3. Como uma solução alternativa, defina `cgrates_timeout` para um valor inferior ao tempo de resposta esperado do CGRateS para evitar que chamadas lentas do CGRateS bloqueiem o pool de processos Elixir.

## Contagem de sessões ativas não diminuindo após liberações

Se `omni_chf.sessions.active.count` permanecer elevado após as sessões terem sido liberadas:

1. Verifique as respostas 404 nas chamadas de Liberação — se o SMF receber 404, pode não tentar novamente e o SMF considera a sessão liberada enquanto o OmniCHF pode ainda ter o contexto.
2. No caso inverso, se o OmniCHF reiniciou e perdeu o estado da sessão, os contextos se foram, mas o SMF pode ainda enviar solicitações de Atualização/Liberar que resultam em 404. Este é o comportamento esperado.

## Registro do NRF não mantido

Verifique a métrica `omni_chf.nrf.registration.status`. Se estiver em 0:

1. Verifique se o `nrf_uri` está correto e se o NRF é acessível a partir do `sbi_addr` do OmniCHF.
2. Verifique se `mcc` e `mnc` correspondem à configuração PLMN do NRF.
3. Revise os logs da aplicação na inicialização para erros de registro do NRF.