



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS FLORIANÓPOLIS
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Patrícia Bardini Arigoni

**VOIPTRACK QOS: SISTEMA PARA MONITORAMENTO DE QOS DE CHAMADAS
VOIP**

Florianópolis
2026

Patrícia Bardini Arigoni

**VOIPTRACK QOS: SISTEMA PARA MONITORAMENTO DE QOS DE CHAMADAS
VOIP**

Trabalho de Conclusão de Curso submetido
ao Curso de Graduação em Ciência da
Computação do Centro Tecnológico da
Universidade Federal de Santa Catarina
para obtenção do título de Bacharel em
Ciência da Computação.

Orientador: Prof. Roberto Willrich, Dr.

Florianópolis
2026

RESUMO

A tecnologia VoIP tem ganhado destaque devido à sua economia e flexibilidade na transmissão de voz via redes IP. Considerando que a qualidade dessas chamadas é fortemente influenciada por fatores como latência, *jitter* e perda de pacotes, o monitoramento eficaz desses parâmetros torna-se essencial para garantir uma comunicação estável e satisfatória. Nesse contexto, o monitoramento da qualidade de chamadas é particularmente importante para os provedores de serviço VoIP, que precisam acompanhar a qualidade das chamadas oferecidas aos seus clientes. Tendo isso em vista, o objetivo desse trabalho é implementar um protótipo chamado VoIPTrack QoS, capaz de oferecer informações sobre a qualidade das chamadas dos clientes de um provedor de serviço VoIP e recomendar ações mitigadoras, mesmo em cenários com tráfego criptografado e dispositivos móveis, superando as limitações das soluções atuais.

Palavras-chave: VoIP. QoS. Monitoramento. SIP.

ABSTRACT

VoIP technology has gained prominence due to its cost-effectiveness and flexibility in voice transmission over IP networks. Considering that the quality of these calls is strongly influenced by factors such as latency, jitter, and packet loss, effective monitoring of these parameters becomes essential to ensure stable and satisfactory communication. In this context, call quality monitoring is particularly important for VoIP service providers, who need to track the quality of calls offered to their customers. With this in mind, the objective of this work is to implement a prototype called VoIPTrack QoS, capable of providing information on the call quality of a VoIP service provider's customers and recommending mitigating actions, even in scenarios with encrypted traffic and mobile devices, overcoming the limitations of current solutions.

Keywords: VoIP. QoS. Monitoring. SIP.

LISTA DE FIGURAS

Figura 1 – Componentes SIP e Protocolos. (STALLINGS, 2014).	15
Figura 2 – Técnica de Redundant Encoding (PERKINS; HODSON; HARDMAN, 2002).	22
Figura 3 – Infraestrutura do Sistema VoIPTrack QoS. (Autorial, 2026).	30
Figura 4 – Tabelas do Banco de dados. (Autorial, 2026).	39
Figura 5 – Médias de MOS, de <i>jitter</i> , de perda de pacotes e de latência para um determinado cliente. (Autorial, 2026).	43
Figura 6 – Médias de MOS ao longo do tempo para uma determinada empresa. (Autorial, 2026).	43
Figura 7 – Chamadas simultâneas em uma determinada empresa. (Autorial, 2026).	44
Figura 8 – Usuários mais ativos em uma determinada empresa. (Autorial, 2026).	44
Figura 9 – Chamadas com MOS abaixo de 3. (Autorial, 2026).	45
Figura 10 – Alerta responsável por monitorar o MOS. (Autorial, 2026).	45
Figura 11 – Infraestutura do Ambiente de Testes. (Autorial, 2026).	46

LISTA DE ABREVIATURAS E SIGLAS

CNAME	Canonical Name
DNS	Domain Name System
FEC	Forward Error Correction
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
IP	Internet Protocol
MOS	Mean Opinion Score
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RFC	Request for Comments
ROI	Retorno sobre o investimento
RTC	Real-Time Communication
RTCP	RTP Control Protocol
RTP	Real-time Transport Protocol
SIP	Session Initiation Protocol
SRTP	Secure Real-time Transport Protocol
TCP	Transmission Control Protocol
UAC	Agente Usuário Cliente
UAS	Agente Usuário Servidor
URI	Uniform Resource Identifier
VoIP	Voice over Internet Protocol

SUMÁRIO

1	INTRODUÇÃO	8
1.1	Objetivos	9
1.1.1	Objetivo Geral	9
1.1.2	Objetivos Específicos	9
1.2	Justificativa	10
1.3	Estrutura do relatório	10
2	CONCEITOS FUNDAMENTAIS EM VOIP	12
2.1	VoIP	12
2.2	Principais protocolos utilizados no VoIP	12
2.2.1	Protocolo SIP	12
2.2.2	RTP	13
2.2.3	RTCP	14
2.3	Principais componentes de um sistema VoIP	15
3	QOS EM VOIP	17
3.1	Métricas	17
3.1.1	Confiabilidade	17
3.1.2	Atraso	17
3.1.3	<i>Jitter</i>	18
3.1.4	Vazão	18
3.1.5	Valores de referência para métricas de QoS	18
3.1.6	MOS	19
3.1.7	Fator R	20
3.2	Técnicas FEC	21
4	SOLUÇÕES ABERTAS PARA VOIP	23
4.1	Servidores SIP	23
4.1.1	Asterisk	23
4.1.2	FreeSWITCH	23
4.1.3	Kamailio	24
4.2	WebRTC: Solução para implantação agente usuário VoIP na Web	24
4.2.1	RTC	24
4.2.2	WebRTC	25
4.2.3	SIP.js	26
4.2.4	JsSIP	27

5	TRABALHOS RELACIONADOS	28
5.1	VoIPmonitor	28
5.2	SIP3	28
5.3	Dotcom-Monitor	29
6	SISTEMA VOIPTRACK QOS	30
6.1	Tecnologias Utilizadas	31
6.1.1	Docker	31
6.1.2	Rust	31
6.1.3	PostgreSQL	32
6.1.4	Grafana	32
6.2	API VoIPTrack QoS	33
6.2.1	Gestão de Empresas	33
6.2.2	Gestão de Usuários	34
6.2.3	Relatórios de Chamadas	35
6.2.4	Cálculos	36
6.2.4.1	Cálculo da Taxa de Perda de Pacotes	37
6.2.4.2	Cálculo do Fator de Degradação por Perda (lef)	37
6.2.4.3	Estimativa do Atraso Unidirecional	37
6.2.4.4	Cálculo do R-Factor	38
6.2.4.5	Conversão do R-Factor para MOS	39
6.3	Banco de Dados	39
6.4	Dashboard de Monitoramento	40
6.4.1	Controle de Acesso aos Dashboards	41
6.4.2	Painéis desenvolvidos	42
6.4.3	Sistema de Alertas	45
6.5	Ambiente de Testes	46
6.6	Testes realizados	47
6.6.1	Teste funcional do fluxo completo	47
6.6.2	Teste do dashboard com dados simulados	48
6.7	Limitações	49
7	CONCLUSÃO	50
	REFERÊNCIAS	51

1 INTRODUÇÃO

Nos últimos anos, o crescimento da Internet e o surgimento de novas tecnologias promoveram profundas transformações nas formas de conexão, trabalho e comunicação. Diversas tecnologias têm surgido a fim de possibilitar uma comunicação mais eficaz, como o VoIP (*Voice over Internet Protocol*). Com essa tecnologia, a transmissão de voz é feita por meio de redes IP (*Internet Protocol*) e não por meio de redes a comutação de circuitos do sistema de telefonia clássica.

Recentemente, a comunicação VoIP tem se tornado cada vez mais comum nos ambientes corporativos, *call centers* e até mesmo em aplicativos para uso pessoal. Como aponta Hartpence (2013), uma das principais razões para esse aumento é a significativa economia que esta tecnologia proporciona. Como a voz é transmitida digitalmente na forma de datagramas IP, as chamadas VoIP podem ser realizadas para qualquer destino conectado à Internet. Desta forma, as tarifas são evitadas ou então reduzidas significativamente. Portanto, migrar para uma arquitetura VoIP pode proporcionar um bom retorno sobre o investimento (ROI) às empresas.

Um dos fatores limitantes da tecnologia VoIP é o fato que a qualidade das chamadas depende fortemente da qualidade de serviço oferecido pela rede IP — o que envolve diversos fatores, como latência, *jitter* e perda de pacotes —, o que pode prejudicar a experiência do usuário.

Para empresas que oferecem serviços de VoIP e grandes corporações fazendo uso deste serviço, o monitoramento da Qualidade de Serviço (QoS) faz-se essencial para garantir uma comunicação VoIP eficiente, pois permite identificar e mitigar problemas que possam afetar a qualidade das chamadas VoIP. Ferramentas de monitoramento ajudam a detectar falhas e gargalos na rede, permitindo saber quais ajustes devem ser feitos a fim de melhorar a qualidade da chamada.

Implementar um sistema de monitoramento de QoS para chamadas VoIP envolve entender o tráfego em tempo real, coletar métricas específicas — latência, *jitter*, perda de pacotes, MOS (*Mean Opinion Score*) — e oferecer uma visualização clara e acessível para administradores de rede. Os benefícios de um sistema desse tipo incluem a detecção de problemas, permitindo correções rápidas sem necessidade de receber notificações dos próprios usuários do sistema. Este monitoramento permite obter subsídios para realizar medidas para otimização do desempenho da rede e melhorias contínuas da infraestrutura com decisões baseadas em evidências sólidas sobre a qualidade da rede.

Em ambientes corporativos, sistemas de monitoramento de QoS para chamadas

VoIP permitem monitorar e otimizar a comunicação entre filiais e clientes externos, assegurando que as interações comerciais não sejam comprometidas. Nos *call centers*, por exemplo, é fundamental garantir que as chamadas sejam claras e sem interrupções, o que melhora a produtividade e a satisfação do cliente.

As soluções atuais presentes no mercado realizam a análise de qualidade de chamadas VoIP a partir da captura passiva do tráfego em pontos específicos da rede. No entanto, como dependem de estar fisicamente posicionados no caminho do tráfego VoIP, sua aplicabilidade é limitada em cenários com uso de dispositivos móveis ou redes onde o tráfego não passa por um ponto central. Além disso, sua incapacidade de lidar com tráfego criptografado sem o acesso às chaves impede a extração de métricas essenciais, comprometendo a avaliação da qualidade em ambientes modernos que utilizam protocolos como SRTP (*Secure Real-time Transport Protocol*). Essas limitações motivaram o desenvolvimento da solução proposta, que supera essas restrições ao não depender da posição na rede nem do acesso ao conteúdo dos pacotes, mantendo a capacidade de análise mesmo em contextos com criptografia e mobilidade.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

O objetivo principal deste projeto de conclusão de curso é especificar e implementar um protótipo de sistema VoIPTrack QoS, para monitoramento da qualidade de chamadas VoIP e, quando da detecção de problemas de qualidade, estabelecer estratégias para orientar os usuários e provedores de serviço VoIP a realização de ações de forma a mitigar os impactos causados por problemas como *jitter*, perdas de pacotes e falta de vazão, preservando a qualidade das transmissões multimídia em tempo real.

1.1.2 Objetivos Específicos

Os objetivos específicos deste trabalho são os seguintes:

- Realizar uma revisão da literatura científica sobre conceitos básicos sobre VoIP, os protocolos envolvidos e as soluções abertas para sua implantação.
- Realizar estudo dos desafios e características das redes IP no que diz respeito à qualidade de áudio e vídeo em transmissões de tempo real.
- Especificar uma solução para o monitoramento de QoS VoIP.
- Selecionar soluções abertas para a implementar um sistema de monitoramento da qualidade de chamadas VoIP.

- Desenvolver um protótipo do sistema de monitoramento de QoS em VoIP usando as soluções selecionadas.
- Estabelecer um ambiente de testes para validação da solução proposta.

1.2 JUSTIFICATIVA

A crescente adoção da tecnologia VoIP reflete um avanço significativo na forma como as pessoas se comunicam, proporcionando uma solução eficiente e econômica. No entanto, apesar das suas vantagens em comparação com os sistemas tradicionais de telefonia, o VoIP também apresenta desafios relacionados à qualidade da transmissão de voz. A variação de fatores como latência, variação de atraso (*jitter*) e perda de pacotes pode impactar negativamente a estabilidade das chamadas.

Esses fatores podem prejudicar a experiência do usuário, resultando em chamadas com cortes de áudio, atrasos e, em casos mais extremos, quedas de chamadas. Em ambientes corporativos, esses problemas podem comprometer a produtividade, o relacionamento com clientes e a imagem da empresa. Empresas que dependem dessa tecnologia para o atendimento ao cliente, por exemplo, enfrentam o risco de causar insatisfação, podendo resultar em perda de contratos, diminuição da confiança do cliente e queda da reputação da marca.

Portanto, monitorar a qualidade das chamadas VoIP torna-se essencial para garantir o bom funcionamento dos sistemas de comunicação nas organizações. A implementação de um sistema de monitoramento eficaz permite a identificação de falhas e a correção de problemas, sem que os próprios usuários precisem relatar as ocorrências. Além disso, essa monitoração ajuda a evitar problemas que podem gerar prejuízos financeiros para empresas que dependem do atendimento telefônico para realizar vendas, prestar suporte técnico ou outras interações com clientes.

A justificativa para o desenvolvimento e implementação de um sistema de monitoramento de QoS para VoIP está diretamente ligada à necessidade de assegurar que o serviço de comunicação seja o mais robusto e eficiente possível. Garantir a qualidade das chamadas VoIP é crucial para assegurar o sucesso das empresas que dependem dessa tecnologia para operações do dia a dia, seja no atendimento aos clientes, seja na comunicação interna.

1.3 ESTRUTURA DO RELATÓRIO

O presente trabalho está organizado como segue. No Capítulo 2, são apresentados os conceitos fundamentais em VoIP, detalhando os principais elementos, protocolos e funcionamento da tecnologia. O Capítulo 3 trata da Qualidade de Serviço em VoIP,

explorando métricas objetivas e subjetivas, bem como técnicas FEC (*Forward Error Correction*) para garantir a qualidade das chamadas. No Capítulo 4, são abordadas soluções abertas para VoIP, tanto no nível de servidores SIP (*Session Initiation Protocol*) quanto de agentes usuários. O Capítulo 5 analisa trabalhos relacionados, com foco em soluções existentes para monitoramento de qualidade em chamadas VoIP. Já o Capítulo 6 apresenta a implementação do Sistema VoIPTrack QoS. Por fim, o Capítulo 7 expõe a conclusão.

2 CONCEITOS FUNDAMENTAIS EM VOIP

Este capítulo apresenta uma revisão dos conceitos fundamentais da tecnologia VoIP, abordando os principais protocolos e componentes que compõem uma solução VoIP, essenciais para a compreensão do presente trabalho.

2.1 VOIP

VoIP, ou Voz sobre Protocolo de Internet em português, é uma tecnologia que permite realizar chamadas de voz e outras comunicações multimídia — como vídeo e mensagens — utilizando as redes IP, como a própria Internet, em vez da rede telefônica tradicional (*PSTN - Public Switched Telephone Network*).

Essa tecnologia oferece uma série de benefícios que a tornam uma alternativa atraente às chamadas telefônicas convencionais. Uma das maiores vantagens é a redução de custos, especialmente em chamadas de longa distância, que geralmente são gratuitas ou muito mais baratas via VoIP. Além disso, a flexibilidade e mobilidade são destacadas por [Kurose e Ross \(2013\)](#), pois o VoIP permite realizar e receber chamadas de qualquer lugar com acesso à Internet, usando diferentes dispositivos como computadores, smartphones e telefones IP.

O VoIP é amplamente utilizado em diversos cenários. Para o uso pessoal, aplicativos como *WhatsApp*, *Meet* e *Zoom* utilizam VoIP para chamadas de voz e vídeo entre usuários. Em ambientes corporativos, o VoIP é essencial em sistemas como as centrais telefônicas IP (PABX IP) e em *call centers*. Como apontam [Tanenbaum e Feamster \(2019\)](#), o VoIP transformou as comunicações corporativas, oferecendo maior economia e adaptabilidade.

2.2 PRINCIPAIS PROTOCOLOS UTILIZADOS NO VOIP

No VoIP, a comunicação é viabilizada por diversos protocolos. Dentre os mais importantes, destacam-se o SIP, RTP e RTCP.

2.2.1 Protocolo SIP

O protocolo de comunicação SIP (*Session Initiation Protocol*) foi desenvolvido pela *Internet Engineering Task Force* (IETF) ([ROSENBERG et al., 2002](#)) e é amplamente utilizado em sistemas de telefonia IP. Ele é o protocolo utilizado para estabelecer, modificar e encerrar sessões multimídia entre dois usuários em uma rede IP. O SIP

opera de forma semelhante ao HTTP (*Hypertext Transfer Protocol*), pois ambos são baseados em texto e usam um modelo de requisição/resposta.

As mensagens SIP podem ser encaminhadas usando diferentes protocolos da camada de transporte, como o UDP ou TCP. Essa flexibilidade é muito importante, pois permite que o SIP seja adaptado a diversas redes e tecnologias.

A RFC 3261 (ROSENBERG et al., 2002) define os seguintes métodos SIP:

- *REGISTER*: Utilizado por um Agente Usuário (descrito na Seção 2.3) para se registrar junto a um servidor SIP, informando seu endereço IP atual. Isso permite que o servidor saiba onde o usuário está localizado na rede para encaminhar chamadas corretamente.
- *INVITE*: Inicia o processo de estabelecimento de uma sessão de comunicação (como uma chamada VoIP). Por exemplo, quando um usuário realiza uma chamada, seu dispositivo envia uma mensagem SIP *INVITE* ao servidor, solicitando a criação de uma sessão com o destinatário. Essa mensagem pode incluir informações como *codecs* suportados e parâmetros de mídia.
- *ACK*: Confirma o recebimento da resposta final a um *INVITE*, especialmente após uma resposta *200 OK*. É essencial para completar o *handshake* da chamada. Sem o *ACK*, a sessão não é considerada estabelecida.
- *CANCEL*: Enviado para cancelar uma requisição pendente, como um *INVITE* que ainda não foi atendido. Por exemplo, se o chamador decidir encerrar a tentativa de ligação antes que ela seja aceita, o dispositivo envia um *CANCEL*.
- *BYE*: Encerra uma sessão SIP já estabelecida. Quando um dos participantes de uma chamada deseja finalizar a conversa, ele envia uma mensagem *BYE*, encerrando a comunicação.
- *OPTIONS*: Permite a consulta das capacidades de um Agente Usuário ou Servidor Proxy, sem necessariamente iniciar uma sessão. Pode ser usado, por exemplo, para verificar se um dispositivo está disponível e quais *codecs* ou métodos ele suporta.

2.2.2 RTP

O Protocolo de Transporte em Tempo Real (RTP - *Real-time Transport Protocol*) (SCHULZRINNE et al., 2003) é um protocolo fundamental para a comunicação de dados multimídia que exigem entrega em tempo real. No VoIP, o RTP é utilizado para o transporte de pacotes de voz. Ele foi projetado especificamente para lidar com as complexidades da transmissão de áudio e vídeo pelas redes IP, na qual atrasos e

perdas de pacotes podem comprometer significativamente a qualidade da experiência do usuário.

Ao contrário de protocolos como o TCP (*Transmission Control Protocol*), que prioriza a entrega confiável e ordenada de dados, garantindo que todos os pacotes cheguem e na sequência correta, o RTP foca na entrega em tempo real. Isso pois as aplicações multimídia conversacionais, como o VoIP, preferem descartar um pacote atrasado a esperar por sua retransmissão, pois um pacote de áudio ou vídeo que chega muito tarde é, na prática, inútil em cenários de comunicação em tempo real.

2.2.3 RTCP

O RTP é geralmente utilizado em conjunto com o protocolo RTCP (*RTP Control Protocol*). Enquanto o RTP transporta os dados de mídia em si, o RTCP é responsável por fornecer informações de controle, monitoramento e *feedback* sobre a qualidade da transmissão.

A RFC 3550 ([SCHULZRINNE et al., 2003](#)) aponta quatro papéis desempenhados pelo RTCP:

- Permitir que os receptores do tráfego multimídia enviem relatórios aos remetentes informando sobre a qualidade da transmissão, essencial para detectar problemas na rede.
- Manter identificação persistente dos participantes (CNAME – *Canonical Name*), possibilitando que os receptores associem múltiplos fluxos de mídia a um único usuário.
- Controle da taxa de envio de pacotes RTCP, para garantir a escalabilidade em grandes grupos.
- Transporte opcional de informações mínimas de controle de sessão, útil em sessões com pouco controle de participantes, em que estes entram e saem livremente.

Sem o RTCP, as sessões RTP não teriam conhecimento das condições da rede necessárias para oferecer a qualidade de VoIP necessária à comunicação. Se as aplicações VoIP não usarem este protocolo, os emissores do tráfego de voz não têm *feedback* sobre a qualidade da entrega, e os receptores não são capazes de informar sobre problemas como perda de pacotes ou *jitter* excessivo.

2.3 PRINCIPAIS COMPONENTES DE UM SISTEMA VOIP

Segundo [Stallings \(2014\)](#), uma rede SIP pode ser vista como uma infraestrutura baseada no modelo cliente/servidor, composta por agentes usuários e servidores SIP, além de outros elementos que colaboram na sinalização e no estabelecimento das sessões. De acordo com a especificação da RFC 3261 ([ROSENBERG et al., 2002](#)), as definições de cliente e servidor são:

- **Cliente:** Qualquer componente da rede capaz de inicializar uma sessão de comunicação com o protocolo SIP. Nem todos os clientes precisam interagir diretamente com um usuário final para oferta de um serviço de voz. Exemplos incluem agentes usuário clientes e proxies.
- **Servidor:** Elemento responsável por receber requisições SIP e fornecer as respostas adequadas. Servidores podem assumir diferentes papéis, como *proxies*, servidores registradores, servidores de redirecionamento ou agentes usuário servidores.

A Figura 1 mostra como alguns componentes SIP se relacionam, bem como os protocolos que são empregados.

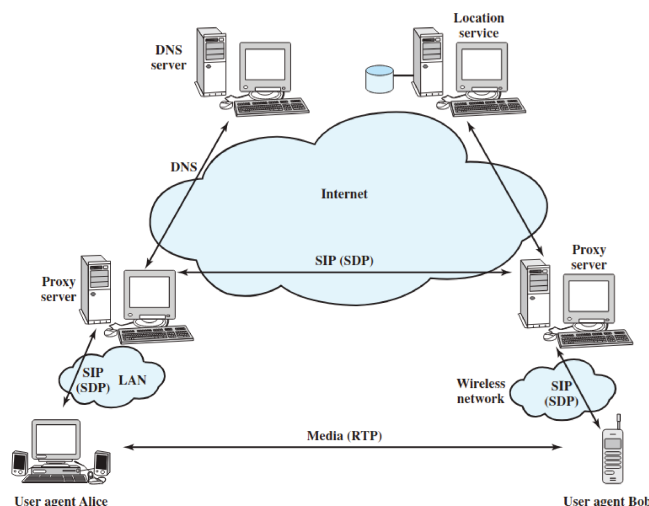


Figura 1 – Componentes SIP e Protocolos. ([STALLINGS, 2014](#)).

Os principais elementos que integram uma rede SIP típica incluem:

- **Agente Usuário:** Presente nos terminais finais SIP, esse componente desempenha duas funções:
 - *Agente Usuário Cliente (UAC):* Responsável por iniciar requisições SIP. Na Figura 1 são os UAC Alice e Bob, que estabelecem uma sessão SIP para se comunicarem;

- *Agente Usuário Servidor (UAS)*: Recebe essas requisições e responde a elas, aceitando, rejeitando ou redirecionando a requisição.
- **Servidor Proxy**: Um servidor intermediário que funciona simultaneamente como cliente e servidor, encaminhando solicitações em nome de outros clientes. Sua principal função é o roteamento eficiente de mensagens para o destino correto, além de poder aplicar políticas como controle de permissões. O *proxy* pode interpretar e modificar certas partes das mensagens SIP antes de redirecioná-las. Na Figura 1, existem dois Servidores *Proxy*, um no domínio VoIP da instituição do usuário Alice, e outro na do usuário Bob.
- **Servidor de Redirecionamento**: Atua no processo de estabelecimento de chamadas, fornecendo ao originador da chamada o endereço do destinatário. Ele responde ao cliente com um novo URI (*Uniform Resource Identifier*), de forma semelhante à resolução iterativa de nomes no DNS (*Domain Name System*).
- **Servidor Registrador**: Este servidor processa requisições do tipo *REGISTER*, registrando os dados recebidos (como o endereço SIP e o IP correspondente) em um serviço de localização específico ao domínio.

3 QOS EM VOIP

A QoS em VoIP está relacionada ao nível de qualidade oferecida pela rede IP para o tráfego de dados. Sem garantias de QoS adequadas, as chamadas VoIP podem sofrer problemas como atrasos, interrupções e/ou distorções, comprometendo a comunicação.

3.1 MÉTRICAS

Conforme apontam [Tanenbaum e Feamster \(2019\)](#), a QoS se baseia em quatro requisitos básicos, que servem como métricas objetivas: a confiabilidade, o atraso, o *jitter* e a vazão. Além desses parâmetros técnicos, existe também o MOS, uma métrica subjetiva de qualidade de chamada calculada com base no desempenho desses requisitos fundamentais.

3.1.1 Confiabilidade

Refere-se à probabilidade de que um pacote de dados seja entregue com sucesso ao seu destino, sem erros ou perdas. É a garantia de que a informação enviada realmente chegará onde deveria.

Em comunicações de voz via VoIP, taxas de perda elevadas comprometem a qualidade da chamada, gerando ruídos e cortes na fala. Isso afeta a compreensão e a fluidez da conversa. Quanto maior a confiabilidade da rede, melhor a qualidade da voz percebida pelo usuário.

3.1.2 Atraso

Mede o tempo que um pacote de dados leva para trafegar da origem ao destino. Esse tempo inclui o processamento nos dispositivos de rede, o tempo de propagação do sinal pelo meio físico e o tempo de enfileiramento nos dispositivos.

Para a maioria das aplicações, um pequeno atraso é aceitável. Porém, para comunicações interativas, como chamadas telefônicas via VoIP e videoconferências, o retardo excessivo é um grande problema ([TANENBAUM; FEAMSTER, 2019](#)). Se o atraso de ida e volta (*RTT - Round-Trip Time*) for muito alto, os participantes da conversa podem começar a se sobrepôr, tendo dificuldades em manter a comunicação.

3.1.3 Jitter

A variação de atraso, mais conhecida como *jitter*, é a variação no retardo entre a chegada de pacotes consecutivos. O *jitter* ocorre quando esses pacotes não chegam em intervalos regulares, alguns chegam mais rápido e outros mais devagar do que o esperado.

Essa variação pode causar problemas significativos para aplicações em tempo real. Em uma chamada VoIP, por exemplo, se os pacotes chegam em um ritmo inconsistente, o áudio pode soar distorcido ou apresentar interrupções. Para mitigar o *jitter*, os dispositivos receptores geralmente utilizam buffers de *jitter* que armazenam os pacotes temporariamente e os liberam em um ritmo constante. Neste caso, *jitters* grandes refletem em atrasos significativos devido ao aumento do tempo de bufferização.

3.1.4 Vazão

A vazão refere-se à quantidade de dados que pode ser transmitida por uma rede em um determinado período de tempo, geralmente medida em bits por segundo (*bps*). Ela não é uma métrica de qualidade diretamente percebida pelo usuário como as outras três, mas é um requisito fundamental da QoS, pois afeta diretamente a capacidade de atender aos outros requisitos.

Uma vazão insuficiente pode ocorrer em caso de congestionamento na rede, o que resulta em aumento do atraso (pacotes esperando em filas nos roteadores), maior *jitter* (pela variação nos tempos de espera) e maior perda de pacotes (devido ao descarte de pacotes nos roteadores devido ao *overflow* das filas). Para garantir uma boa qualidade em aplicações de voz e vídeo, é essencial que haja largura de banda suficiente disponível para todo o tráfego crítico.

3.1.5 Valores de referência para métricas de QoS

Embora métricas como atraso, *jitter* e perda de pacotes sejam medidas continuamente durante uma chamada VoIP, sua interpretação depende de valores de referência amplamente adotados na literatura e em recomendações técnicas. Esses limites permitem avaliar se a qualidade observada é adequada para comunicações em tempo real.

O atraso é um dos fatores que mais influenciam a percepção da qualidade da comunicação. De acordo com a recomendação ITU-T G.114 ([International Telecommunication Union, 2003](#)), atrasos inferiores a 150 ms são considerados aceitáveis para a maioria das aplicações de voz. Valores entre 150 ms e 400 ms ainda permitem a comunicação, porém podem causar dificuldades na conversação devido ao aumento do

tempo de resposta entre os interlocutores. Atrasos superiores a 400 ms são geralmente considerados inadequados para chamadas interativas.

Em relação ao *jitter*, recomenda-se que sua variação permaneça inferior a 20 ou 30 ms para garantir uma reprodução contínua do áudio ([International Telecommunication Union, 2001](#)). Valores superiores podem aumentar a necessidade de bufferização e ocasionar interrupções ou distorções perceptíveis durante a chamada.

A perda de pacotes também possui impacto significativo na qualidade percebida. Em condições normais de operação, perdas inferiores a 1% são praticamente imperceptíveis para *codecs* modernos. Entre 1% e 3%, a qualidade tende a permanecer satisfatória ([International Telecommunication Union, 2001](#)), principalmente quando são utilizados *codecs* robustos ou mecanismos de correção de erros. Perdas superiores a 5% normalmente resultam em degradação perceptível da qualidade da voz, podendo comprometer a inteligibilidade da comunicação.

A Tabela 1 resume os valores de referência adotados para essas métricas.

Métrica	Boa	Aceitável	Ruim
Atraso (ms)	< 150	150–400	> 400
Jitter (ms)	< 20	20–30	> 30
Perda de pacotes (%)	< 1	1–3	> 5

Tabela 1 – Valores de referência para métricas de QoS em chamadas VoIP. ([International Telecommunication Union, 2001](#)).

Os valores apresentados servem como referência para interpretação das métricas coletadas pelo sistema VoIPTrack QoS e complementam a avaliação realizada por meio do MOS, que considera simultaneamente diferentes fatores que influenciam a experiência do usuário.

3.1.6 MOS

O MOS é uma pontuação que varia numa escala de 1 (pior qualidade) a 5 (melhor qualidade). No início de sua adoção era obtido por meio da avaliação humana da qualidade de uma chamada de voz, sendo subjetiva. Ela representa o quão satisfatória a comunicação é para o usuário final.

Os provedores de serviços utilizam ferramentas para monitorar continuamente as métricas objetivas de QoS e correlacionam esses dados com os valores de MOS esperados. Assim, o MOS atua como um medidor da qualidade de uma comunicação VoIP.

A Tabela 2 apresenta a interpretação qualitativa dos diferentes níveis de MOS, relacionando cada faixa de pontuação à experiência perceptível do usuário final.

Nota	Qualidade percebida	Significado
5	Excelente	Sem degradações perceptíveis.
4	Boa	Leves degradações perceptíveis, porém sem comprometer a experiência.
3	Moderada	Degradações audíveis/visuais, mas sem perda de inteligibilidade
2	Pobre	Presença de interrupções que impactam a compreensão.
1	Ruim	Comunicação inviável.

Tabela 2 – Relação entre pontuação MOS e experiência do usuário. ([International Telecommunication Union, 1996](#)).

3.1.7 Fator R

Atualmente, com o avanço das tecnologias de comunicação, surgiram métodos objetivos para estimar o MOS de forma automática, com base em parâmetros técnicos da rede e do áudio. Um dos principais modelos utilizados é o modelo *E-model*, definido pela recomendação ITU-T G.107 ([International Telecommunication Union, 2015](#)), que introduz o Fator R. Esse fator é um índice numérico que varia de 0 a 100 e leva em conta diversos fatores técnicos que afetam a experiência do usuário.

A pontuação R é calculada por meio da seguinte fórmula:

$$R = R_o - I_s - I_d - I_{e-eff} + A \quad (3.1)$$

, em que:

- *R_o*: representa a qualidade básica da conexão, ou seja, o quão limpa é a transmissão sem considerar degradações mais complexas. Leva em conta, por exemplo, ruídos do ambiente, ruído da linha e outras interferências básicas.
- *I_s*: representa os prejuízos simultâneos ao sinal de voz, como ecos, distorções de volume e ruídos adicionais que ocorrem ao mesmo tempo que o usuário está falando.
- *I_d*: corresponde aos prejuízos causados pelo atraso na comunicação, como latência e eco causado por longos tempos de ida e volta (*round trip*).
- *I_{e-eff}*: é o fator de degradação causado por *codecs* de baixa taxa de bits. Também leva em conta perdas de pacotes na rede, que são comuns em transmissões VoIP, como em redes móveis e Internet instável.
- *A*: é usado para compensar algumas degradações quando o usuário aceita uma qualidade inferior em troca de outras conveniências — como mobilidade ou custo mais baixo. Por exemplo, em uma chamada de celular em movimento, o usuário pode tolerar mais perdas do que em uma chamada fixa de alta qualidade.

No fim, o valor R resume todas essas variáveis em um único número de 0 a 100, que pode ser convertido numa estimativa objetiva da qualidade percebida da chamada, aproximando-se do MOS. De acordo com o ITU-T G.107 ([International Telecommunication Union, 2015](#)), essa conversão pode ser feita da seguinte forma:

Para $R < 0$:

$$MOS = 1 \quad (3.2)$$

Para $0 < R < 100$:

$$MOS = 1 + 0.035 \times R + R \times (R - 60) \times (100 - R) \times 7 \times 10^{-6} \quad (3.3)$$

Para $R > 100$:

$$MOS = 4.5 \quad (3.4)$$

3.2 TÉCNICAS FEC

FEC são técnicas para aprimorar a QoS em VoIP, especialmente em ambientes onde a perda de pacotes é uma preocupação. Em vez de depender da retransmissão de pacotes perdidos — um processo que introduziria atrasos grandes na comunicação em tempo real —, as técnicas FEC atuam de forma proativa, adicionando informações redundantes aos dados de voz originais antes mesmo que sejam enviados.

O processo básico de uma técnica FEC envolve três etapas:

1. Codificação: O transmissor aplica um algoritmo de codificação FEC aos dados originais, gerando bits de redundância e transmitindo tudo junto.
2. Transmissão pelo canal: Os dados codificados (dados + *bits* de correção) são enviados. Durante a transmissão, podem ocorrer erros devido a ruídos ou interferências.
3. Decodificação e correção: O receptor usa os bits extras de correção para detectar e corrigir os erros, reconstruindo os dados originais.

Um exemplo de técnica FEC comumente utilizada em VoIP é o *Redundant Encoding*, também conhecido como redundância de pacotes de voz. Essa abordagem envia, junto com o pacote de voz atual, uma versão compactada do pacote anterior ([PERKINS; HODSON; HARDMAN, 2002](#)). Dessa forma, se o pacote atual for perdido, mas o próximo pacote chegar a tempo, ainda será possível reconstruir (mesmo que com qualidade reduzida) o conteúdo do pacote perdido a partir dessa redundância.

A Figura 2 ilustra um cenário em que quatro pacotes de voz são transmitidos com redundância. Durante a transmissão, ocorre a perda do pacote 3. No entanto,

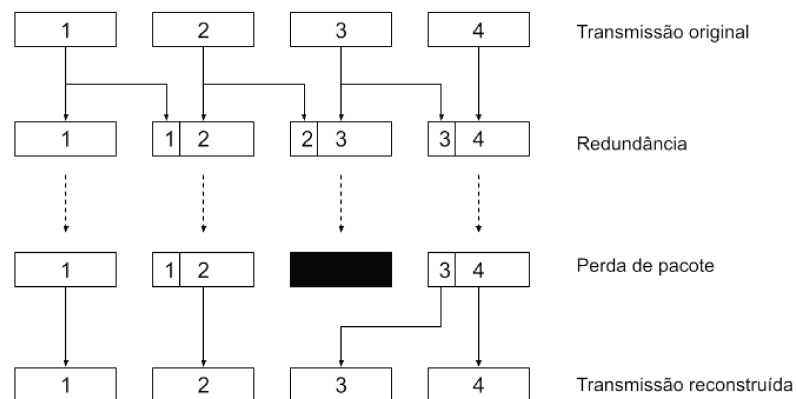


Figura 2 – Técnica de Redundant Encoding ([PERKINS; HODSON; HARDMAN, 2002](#)).

graças à informação redundante incluída no pacote 4, é possível reconstruir o conteúdo perdido.

4 SOLUÇÕES ABERTAS PARA VOIP

Neste capítulo são apresentadas soluções abertas amplamente utilizadas para implementação de sistemas de comunicação VoIP. O objetivo é explorar plataformas que oferecem, de forma gratuita e com código aberto, recursos essenciais para o funcionamento de uma infraestrutura de telefonia baseada em IP. Essas soluções abrangem desde servidores SIP, responsáveis pelo controle das sessões de comunicação, até os agentes usuários, que realizam a interface com o usuário final.

4.1 SERVIDORES SIP

Um servidor SIP é um componente fundamental em redes de telefonia IP, sendo responsável por gerenciar o estabelecimento, manutenção e encerramento de sessões de comunicação, como chamadas de voz e vídeo, entre dois ou mais dispositivos na rede.

4.1.1 Asterisk

O Asterisk ([Sangoma Technologies, 2026](#)) é uma das soluções mais conhecidas e amplamente utilizadas no mundo da telefonia IP. Ele é um PBX (*Private Branch Exchange*) IP *open source* que permite montar uma central telefônica completa utilizando apenas software e um servidor convencional. Oferece suporte ao protocolo SIP e a vários outros protocolos de comunicação, permitindo estabelecer chamadas de voz e vídeo, URAs (unidades de resposta audível), correio de voz, filas de atendimento, gravação de chamadas, entre outros recursos. Desde a versão 11, o Asterisk suporta WebRTC (descrito na Seção [4.2.2](#)) e WebSocket, usados para viabilizar a comunicação em tempo real diretamente a partir de navegadores web.

Uma das grandes vantagens do Asterisk é sua flexibilidade: ele permite a criação de *dialplans* personalizados utilizando uma linguagem própria. Isso facilita a implementação de soluções adaptadas a necessidades específicas, como filtros de chamadas e menus interativos.

4.1.2 FreeSWITCH

O FreeSWITCH ([SignalWire, 2024](#)) surgiu como uma alternativa moderna ao Asterisk, com foco em desempenho, modularidade e escalabilidade. Foi criado por ex-desenvolvedores do próprio Asterisk com o objetivo de superar algumas das suas limitações de arquitetura. Assim como o Asterisk, o FreeSWITCH é *open source*, mas sua estrutura é mais orientada a módulos e mais preparada para lidar com grandes

volumes de chamadas simultâneas, tornando-o atrativo para operadoras de VoIP e plataformas de comunicação em nuvem.

O FreeSWITCH também suporta SIP, além de WebRTC. Ele pode atuar como PBX, servidor de mídia, *gateway* VoIP, ou ainda ser integrado a outras soluções como parte de uma infraestrutura de comunicação distribuída.

4.1.3 Kamailio

Ao contrário do Asterisk e do FreeSWITCH, o Kamailio ([KAMAILIO, 2025](#)) não processa mídia, e sim atua exclusivamente na sinalização SIP. Ele é um servidor SIP de alto desempenho, projetado para ser um Servidor de Redirecionamento, *Proxy* ou Registrador (definidos na Seção 2.3) em grandes infraestruturas de comunicação. Seu papel principal é rotear chamadas, autenticar usuários, controlar sessões e aplicar regras de segurança, sem se envolver no transporte de áudio ou vídeo.

Por sua arquitetura leve e altamente eficiente, o Kamailio é amplamente utilizado por operadoras de telefonia IP, provedores de SIP Trunking (tecnologia que permite a realização de chamadas telefônicas pela Internet, substituindo troncos telefônicos tradicionais por conexões SIP) e empresas que precisam escalar sua infraestrutura VoIP.

É comum que o Kamailio seja usado em conjunto com o FreeSWITCH ou o Asterisk. Enquanto esses últimos cuidam da parte multimídia da chamada, o Kamailio atua na borda da rede, gerenciando o tráfego SIP de forma eficiente, segura e escalável.

4.2 WEBRTC: SOLUÇÃO PARA IMPLANTAÇÃO AGENTE USUÁRIO VOIP NA WEB

Esta seção apresenta a tecnologia WebRTC, a principal solução aberta voltada à implementação de Agentes Usuários VoIP em ambientes Web. Conforme já descrito na Seção 2.3, estes são os elementos finais da rede responsáveis por iniciar, receber e gerenciar chamadas de voz e vídeo. Além do WebRTC são apresentadas duas bibliotecas JavaScript que facilitam o desenvolvimento de soluções VoIP na Web.

4.2.1 RTC

A Comunicação em Tempo Real (RTC) compreende um conjunto de tecnologias que possibilitam a troca instantânea de áudio, vídeo e dados entre usuários por meio de redes IP. Nos últimos anos, a expansão do acesso à Internet banda larga e das redes móveis impulsionou significativamente a adoção dessas aplicações, tornando-as essenciais tanto para atividades pessoais quanto profissionais. Sua relevância tornou-se ainda mais evidente durante a pandemia de COVID-19, quando as medidas de distanciamento social levaram milhões de pessoas a dependerem de plataformas de

comunicação em tempo real para trabalho, educação e interação social. Atualmente, diversas soluções de RTC coexistem no mercado, incluindo aplicações amplamente utilizadas como *Google Meet*, *Skype*, *WhatsApp* e *Telegram*, que oferecem diferentes funcionalidades e modelos de uso para comunicação síncrona entre indivíduos e organizações (NISTICò et al., 2020).

4.2.2 WebRTC

O WebRTC (*Web Real-Time Communication*) (World Wide Web Consortium (W3C), 2025) é uma tecnologia de código aberto que permite a comunicação em tempo real — áudio, vídeo e troca de dados — por meio dos navegadores Web, sem necessidade de *plugins*. Ele foi projetado para viabilizar aplicações multimídia interativas em tempo real, diretamente no navegador, reduzindo a complexidade da comunicação ponto a ponto entre usuários.

O WebRTC atua no Agente Usuário da arquitetura VoIP, conforme apresentado na Seção 2.3. Nesta solução, o próprio navegador assume esse papel, executando tanto a captura e o envio da mídia quanto o controle de sessão quando integrado a uma camada de sinalização. Atualmente, o WebRTC é suportado nativamente pelos principais navegadores, como Google Chrome, Mozilla Firefox, Safari e Microsoft Edge.

A capacidade de comunicação *peer-to-peer* (P2P) do WebRTC simplifica a implementação de chamadas VoIP em aplicações Web, pois elimina a necessidade de servidores intermediários para o fluxo de mídia. No entanto, antes que essa conexão direta possa acontecer, algumas etapas são necessárias.

1. **Negociação de Sinalização:** Antes de iniciar a comunicação, os participantes precisam trocar informações para a parametrização da sessão. Isso inclui detalhes como *codecs* de áudio/vídeo, endereços IPs e portas de rede a serem usadas. Esta parametrização é feita por meio de um processo de sinalização, que pode usar qualquer protocolo escolhido pela aplicação (por exemplo, WebSocket, SIP ou XMPP). Durante essa etapa, cada lado utiliza o protocolo SDP (*Session Description Protocol*) para descrição dos parâmetros da sessão.
2. **Estabelecimento da Conexão:** Após a troca de sinalização, o WebRTC utiliza o protocolo ICE (*Interactive Connectivity Establishment*) para descobrir a melhor rota de comunicação entre os peers. Ele pode usar servidores STUN, para descobrir o IP público de cada peer, e TURN, para retransmitir mídia quando conexões diretas não são possíveis devido a NAT ou *firewalls*.
3. **Depois da negociação e da abertura dos chamados canais ICE,** os dados de mídia começam a fluir diretamente entre os navegadores usando o RTP (*Real-time*

Transport Protocol). ([Mozilla, 2025](#))

Para possibilitar todas essas funcionalidades, a API do WebRTC disponibiliza três métodos que atuam em conjunto para estabelecer e gerenciar a comunicação em tempo real:

- O *getUserMedia()*, pertencente ao objeto *MediaDevices*, permite acessar dispositivos de entrada, como microfone e câmera, capturando áudio e vídeo do usuário mediante sua autorização.
- O *RTCPeerConnection* é o núcleo da comunicação *peer-to-peer*. Ele gerencia a troca de informações de sessão (SDP), coleta candidatos de conexão (ICE) e estabelece a conexão segura entre os navegadores.
- O *RTCDataChannel* possibilita a troca de dados arbitrários entre os peers, como mensagens de texto, arquivos ou comandos, funcionando de maneira similar a um WebSocket, mas de forma direta entre navegadores.

Além disso, o WebRTC também oferece o método *getStats()* ([Mozilla, 2024](#)), da interface *RTCPeerConnection*, que permite obter métricas detalhadas da conexão em tempo real. Por meio dele, os desenvolvedores podem monitorar estatísticas como perda de pacotes, largura de banda utilizada, *jitter*, latência, entre outros parâmetros importantes para avaliar e otimizar a qualidade da chamada.

4.2.3 SIP.js

O SIP.js ([SIP.js, 2022](#)) é uma biblioteca JavaScript que permite que aplicações Web atuem como clientes SIP completos, utilizando WebRTC para mídia e SIP sobre WebSocket para sinalização. Criado sobre a especificação RFC 3261 do SIP, o SIP.js permite que desenvolvedores integrem facilmente chamadas VoIP em tempo real dentro de aplicações Web.

O SIP.js transforma o navegador em um Usuário Agente completo: ele pode registrar um ramal em um servidor SIP, iniciar e receber chamadas, lidar com mensagens SIP (como *INVITE*, *BYE*, *REGISTER*), e utilizar WebRTC para transmitir áudio e vídeo.

É amplamente usado em sistemas de telefonia integrada à Web, como CRMs (*Customer Relationship Management*) com *click-to-call* e portais de atendimento. Ele é mais flexível do que soluções prontas como *softphones*, pois permite controle total da interface e da lógica da chamada, tudo via código.

4.2.4 JsSIP

O JsSIP ([JsSIP](#), 2023) é outra biblioteca JavaScript *open source* que permite implementar um Usuário Agente completo no navegador, utilizando SIP sobre WebSocket para sinalização e WebRTC para a mídia. Ele é bastante semelhante ao SIP.js em relação a suas funcionalidades. Com JsSIP, também é possível transformar qualquer navegador moderno em um *softphone* SIP baseado na Web, sem necessidade de *plugins*.

5 TRABALHOS RELACIONADOS

Nesta seção são apresentadas e analisadas algumas soluções já existentes para o monitoramento da qualidade de chamadas em sistemas VoIP. O objetivo é destacar as principais características técnicas, vantagens e limitações de cada ferramenta, servindo como base para comparação com a solução proposta neste trabalho.

5.1 VOIPMONITOR

O VoIPmonitor ([VoIPmonitor, 2025](#)) é uma solução de monitoramento de qualidade de chamadas VoIP que possui código aberto, com uma interface gráfica comercial opcional. O monitoramento é realizado através da captura direta de pacotes na rede, funcionando como um *sniffer* que analisa o tráfego em tempo real. A ferramenta apresenta ao administrador informações como o *Mean Opinion Score* (MOS), *jitter*, atraso, perda de pacotes, além de relatórios detalhados por chamada e sessão.

O VoIPmonitor pode ser configurado para o envio de notificações caso determinados limiares de qualidade de chamada sejam excedidos.

Um fator limitante do VoIPmonitor é que ele precisa ser instalado em algum ponto de rede no caminho do tráfego das sessões VoIP, o que em alguns casos não é possível, principalmente devido ao uso de dispositivos móveis para realização das chamadas. Além disso, em ambientes onde o tráfego VoIP é criptografado (como com SRTP - *Secure Real-time Transport Protocol*), o VoIPmonitor não consegue analisar o conteúdo dos pacotes nem métricas de qualidade como *jitter*, latência e MOS, a menos que tenha acesso às chaves de descryptografia.

5.2 SIP3

O SIP3 ([SIP3, 2025](#)) é uma plataforma *open source* dedicada ao monitoramento de qualidade de serviços VoIP. A QoS é medida ao processar dados de sinalização (SIP) e mídia (RTP) enviados por sensores de rede. Esses sensores capturam os pacotes de chamadas em tempo real e os encaminham ao SIP3, geralmente via protocolo HEP. Com base nesses dados, o SIP3 calcula métricas de QoS, apresentando-as ao administrador por meio de *dashboards* interativos.

A ferramenta permite o envio de notificações e alertas aos usuários por meio da interface gráfica ou integração via API.

Um fator limitante do SIP3 é que ele, assim como o VoIPmonitor, depende da captura passiva do tráfego VoIP, o que exige que este esteja instalado em um ponto

da rede por onde as sessões de voz passem. Em cenários onde as chamadas são realizadas por dispositivos móveis ou em arquiteturas distribuídas isso pode ser inviável, já que o tráfego não necessariamente passa por um ponto central de monitoramento. Além disso, em ambientes onde o tráfego está criptografado, o SIP3 também não consegue inspecionar o conteúdo dos pacotes a menos que tenha acesso às chaves de descryptografia, o que não é possível em muitos ambientes.

5.3 DOTCOM-MONITOR

Dentre as soluções comerciais para o monitoramento de chamadas VoIP destaca-se o Dotcom-Monitor ([Dotcom-Monitor, 2025](#)). Ele realiza o monitoramento da QoS e desempenho de serviços de rede e aplicações (incluindo VoIP/SIP) por meio de testes ativos e sensores distribuídos globalmente, em vez de capturar pacotes diretamente na rede interna da empresa. A fim de realizar o monitoramento ativo, o sistema gera tráfego de teste (ex: SIP INVITE, chamadas RTP simuladas, HTTP requests, etc.) a partir de sensores externos na Internet e mede o desempenho do ponto de vista do usuário final. Com isso, ele coleta métricas como tempo de estabelecimento da chamada, tempo de resposta SIP, MOS estimado, *jitter*, latência, perda de pacotes (em chamadas simuladas com RTP).

Além disso, esta ferramenta oferece alertas instantâneos aos usuários, que podem ser recebidos por e-mail, SMS ou até mesmo via chamada de voz.

Por se basear em monitoramento ativo com tráfego sintético, o Dotcom-Monitor não analisa chamadas reais de usuários, o que limita sua capacidade de identificar problemas específicos que ocorrem em ambientes internos. Além disso, seu escopo é voltado principalmente à disponibilidade e desempenho sob perspectiva externa, podendo não refletir com precisão a experiência real dos usuários finais em redes locais ou privadas.

6 SISTEMA VOIPTRACK QOS

O sistema VoIPTrack QoS proposto neste trabalho tem como objetivo receber dados coletados, armazenar e visualizar métricas relacionadas à qualidade de chamadas VoIP. Para isso, foi implementada uma arquitetura distribuída composta por um serviço de coleta de relatórios de qualidade de chamadas VoIP, exposto por meio de uma API REST, responsável pela recepção de relatórios de QoS, um banco de dados para persistência das métricas de qualidade, e uma ferramenta para visualização e monitoramento dos resultados. Essa organização permite a separação de responsabilidades entre os componentes, facilitando a escalabilidade e a manutenção do sistema.

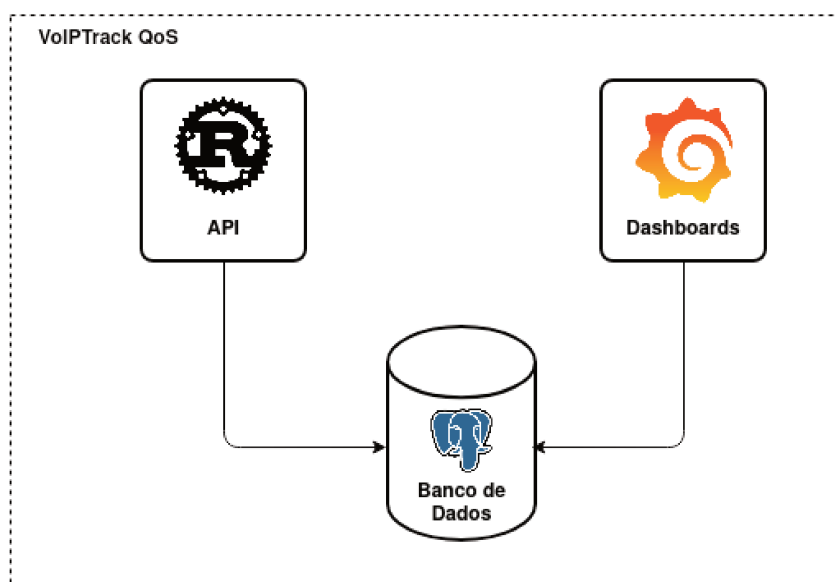


Figura 3 – Infraestrutura do Sistema VoIPTrack QoS. (Autoral, 2026).

Esse sistema é voltado para organizações que dependem da comunicação por Voz sobre IP como parte essencial de suas operações. Ele pode ser utilizado tanto por provedores de serviços VoIP, que ofertam soluções de comunicação para empresas, quanto por corporações que operam sua própria infraestrutura de telefonia IP para comunicação interna ou atendimento a clientes.

A finalidade do sistema VoIPTrack QoS é permitir que essas organizações acompanhem de forma contínua a qualidade das chamadas realizadas, garantindo maior controle sobre a experiência dos usuários finais. Ao coletar e centralizar métricas de qualidade de chamada, o sistema fornece uma base concreta para avaliação do desempenho do serviço.

Com isso, administradores passam a ter uma visão clara e estruturada da qualidade das chamadas, por meio de painéis visuais que apresentam essas métricas

de forma gráfica e intuitiva. Essa visualização facilita a identificação de padrões, falhas recorrentes e degradações no serviço, permitindo agir rapidamente.

Entre os principais benefícios, destaca-se a capacidade de apoiar ações corretivas — ao detectar problemas já existentes — e ações preventivas, ao antecipar situações que possam impactar negativamente as chamadas. Dessa forma, evita-se que falhas cheguem a afetar diretamente os usuários, reduzindo interrupções, melhorando a eficiência operacional e contribuindo para a manutenção de um padrão consistente de qualidade nas comunicações.

6.1 TECNOLOGIAS UTILIZADAS

O desenvolvimento do sistema VoIPTrack QoS e do ambiente de testes demandou a utilização de diferentes tecnologias, cada uma responsável por atender requisitos específicos relacionados à implementação, armazenamento de dados, monitoramento e execução dos experimentos.

6.1.1 Docker

O Docker ([Docker Inc., 2025](#)) é uma plataforma de código aberto que simplifica o desenvolvimento, a distribuição e a execução de aplicações. Ele permite que softwares sejam empacotados junto com todas as suas dependências em um formato padrão chamado contêiner ([BASHARIRAD; BHATTI; AHMADI, 2017](#)). Essa abordagem garante que a aplicação funcione de maneira consistente em diferentes sistemas operacionais e infraestruturas, reduzindo problemas relacionados a incompatibilidades de ambiente.

Diferentemente das máquinas virtuais tradicionais, os contêineres compartilham o *kernel* do sistema operacional hospedeiro, resultando em menor consumo de recursos computacionais e tempo de inicialização reduzido ([BOETTIGER, 2015](#)). Essas características tornam o Docker uma solução adequada para ambientes de desenvolvimento, testes e implantação de aplicações distribuídas.

Neste trabalho, o Docker foi utilizado para isolar os componentes do sistema. A administração dos serviços foi realizada por meio do Docker Compose, permitindo a definição e execução conjunta dos contêineres responsáveis pela API, banco de dados, *dashboards* de monitoramento e clientes WebRTC.

6.1.2 Rust

Rust ([Rust Project Developers, 2026b](#)) é uma linguagem de programação voltada para o desenvolvimento de sistemas, criada com a proposta de oferecer uma alternativa moderna a linguagens como C++. Ela prioriza a segurança de memória sem abrir mão do desempenho, utilizando conceitos como abstrações de custo zero, que permitem

que recursos de alto nível não comprometam a eficiência da execução (BUGDEN; ALAHMAR, 2022).

Outro ponto importante é o seu ecossistema bem estruturado. Rust possui um conjunto amplo de bibliotecas reutilizáveis, conhecidas como *crates* (Rust Project Developers, 2026a), que podem ser facilmente integradas aos projetos por meio da ferramenta oficial Cargo, facilitando significativamente o desenvolvimento e a gestão de dependências.

No contexto deste trabalho, Rust foi utilizada para o desenvolvimento da API responsável pela coleta e armazenamento dos relatórios de qualidade de chamada. A escolha da linguagem foi motivada pelas características previamente mencionadas.

6.1.3 PostgreSQL

PostgreSQL é um sistema gerenciador de banco de dados relacional de código aberto amplamente utilizado em aplicações que demandam confiabilidade, integridade dos dados e alto desempenho (JUBA; VANNAHME; VOLKOV, 2015).

Além de sua conformidade com os padrões SQL, o PostgreSQL disponibiliza recursos de controle transacional e consistência de dados, tornando-o uma solução adequada para aplicações que necessitam armazenar grandes volumes de informações de forma segura.

Neste trabalho, o PostgreSQL foi utilizado para armazenar as métricas coletadas durante as chamadas VoIP.

6.1.4 Grafana

Grafana é uma plataforma de código aberto para visualização e monitoramento de dados que permite a criação de *dashboards* interativos a partir de diferentes fontes de dados e em diversos tipos de visualização, como gráficos de linhas, áreas e barras (ELRADI, 2025). Além disso, oferece suporte a funcionalidades de alertas, possibilitando o envio de notificações quando determinados limites ou eventos críticos são atingidos.

A plataforma também disponibiliza uma API HTTP (Grafana Labs, 2025) com *endpoints* para administração e automação de recursos, incluindo o gerenciamento de organizações, usuários, *dashboards*, fontes de dados, permissões, entre outros.

No sistema desenvolvido para este trabalho, o Grafana foi utilizado para o gerenciamento de organizações e usuários por meio de sua API, para visualizar as métricas armazenadas no PostgreSQL — permitindo o acompanhamento do desempenho das comunicações VoIP — e para o envio de alertas para situações específicas que exigissem atenção.

6.2 API VOIPTRACK QOS

Para o armazenamento e gerenciamento dos relatórios de qualidade das chamadas, foi desenvolvida uma API REST utilizando a linguagem Rust e o *framework* Actix Web. A arquitetura do sistema é organizada em três camadas principais: *handlers*, responsáveis pelo tratamento das requisições HTTP; *services*, responsáveis pela lógica de negócio e integração com serviços externos; e *models*, responsáveis pela definição das estruturas de dados utilizadas.

A API VoIPTrack QoS é organizada por recursos de negócio, seguindo princípios REST, onde cada conjunto de *endpoints* representa uma entidade do domínio. As entidades definidas nesta API são: empresas (*companies*), usuários (*users*) e relatórios de qualidade (*call-reports*).

Os endpoints desta API utilizam serviços fornecidos pela API do Grafana (descrita na Seção 6.1.4). Dessa forma, a API REST VoIPTrack QoS também atua como uma camada de integração transparente com o Grafana, gerenciando organizações e usuários automaticamente. Assim, sempre que esses recursos são criados, atualizados ou removidos na API VoIPTrack QoS, as alterações correspondentes são propagadas para o Grafana, eliminando a necessidade de alterá-los manualmente por meio de sua interface gráfica.

6.2.1 Gestão de Empresas

Esta seção descreve os *endpoints* relacionados à gestão de empresas no sistema. As operações disponibilizadas permitem o cadastro, consulta, atualização e remoção de empresas. Os *endpoints* relacionados à entidade empresas são:

- *POST /company*: Cria uma nova empresa no sistema, além de criar uma organização correspondente no Grafana e armazenar seu identificador.
- *PUT /company?id={id}*: Atualiza os dados de uma empresa existente com base no identificador informado.
- *GET /company?id={id}*: permite obter os dados de uma empresa específica com base no identificador informado.
- *GET /companies*: Retorna a lista completa de empresas cadastradas no sistema.
- *DELETE /company?id={id}*: Remove uma empresa do sistema e também exclui a organização correspondente no Grafana.

Os *endpoints* acima, com exceção do *GET /company?id={id}* e do *GET /companies*, consomem serviços expostos pela API do Grafana. Essa integração tem como

objetivo garantir que cada empresa cadastrada no sistema corresponda a uma organização dentro do Grafana. Dessa forma, ao criar, atualizar ou remover uma empresa, as alterações também são refletidas no sistema externo, mantendo consistência entre as plataformas.

No processo de criação de empresas, é realizada uma requisição HTTP do tipo *POST* para o *endpoint* */api/orgs* do Grafana, responsável pela criação de uma nova organização. O identificador retornado por essa operação é armazenado no banco de dados e vinculado à empresa. Em caso de falha na criação da organização, a operação é interrompida, evitando inconsistências entre os sistemas.

Da mesma forma, na remoção de empresas, é realizada uma requisição HTTP do tipo *DELETE* para o *endpoint* */api/orgs/orgId*, garantindo que a organização correspondente também seja excluída do Grafana.

6.2.2 Gestão de Usuários

Esta seção apresenta os *endpoints* responsáveis pelo gerenciamento de usuários no sistema. As operações incluem criação, consulta, atualização e remoção de usuários, sendo que cada usuário está associado a uma empresa específica. Os *endpoints* para a gerência de usuários são:

- *POST /user*: Cria um novo usuário, cria o usuário correspondente no Grafana e o associa à organização da empresa.
- *PUT /user?id={id}&company_id={company_id}*: Atualiza os dados de um usuário existente com base no identificador e empresa informados.
- *GET /user?id={id}&company_id={company_id}*: Permite a obtenção dos dados de um usuário específico com base no identificador do usuário e da empresa.
- *GET /users?company_id={company_id}*: Retorna uma lista contendo todos os usuários pertencentes a uma determinada empresa.
- *GET /users*: Retorna a lista de todos os usuários cadastrados no sistema.
- *DELETE /user?id={id}&company_id={company_id}*: Remove um usuário do sistema e também o exclui do Grafana.

Assim como no módulo de empresas, este conjunto de funcionalidades também consomem serviços expostos pela API do Grafana para a integração dos sistemas.

Na criação de usuários, o sistema utiliza o *endpoint* */api/admin/users* do Grafana, responsável pelo cadastro de usuários na plataforma. Após a criação, o usuário recebe um identificador interno do Grafana, que é armazenado no sistema.

Em seguida, o usuário é automaticamente associado à organização correspondente da empresa por meio do *endpoint* `/api/orgs/orgId/users`, garantindo que ele tenha acesso aos recursos adequados dentro do Grafana. Nessa etapa, também é definido o nível de permissão do usuário dentro da organização.

Na remoção de usuários, é realizada uma requisição ao *endpoint* `/api/admin/users/userId`, garantindo que o usuário seja removido tanto do sistema interno quanto do Grafana, mantendo consistência entre as plataformas.

6.2.3 Relatórios de Chamadas

Esta seção descreve os *endpoints* relacionados ao gerenciamento de relatórios de chamadas. Esses relatórios armazenam métricas de qualidade de chamadas VoIP, como *jitter*, RTT e perda de pacotes, além do cálculo do indicador MOS.

As operações incluem criação, consulta, atualização e remoção de relatórios, sendo todas realizadas diretamente no banco de dados do sistema. Diferentemente dos módulos anteriores, este componente não possui integração com sistemas externos, concentrando-se na análise e armazenamento de métricas de rede.

Os *endpoints* principais relacionados aos relatórios de chamadas são:

- *POST /call-report*: Responsável por receber um relatório de chamada enviado pelo cliente ao término da comunicação. Antes do armazenamento, os dados passam por um processo de validação para garantir a integridade e a consistência das informações. Caso a validação seja bem-sucedida, é realizado o cálculo do fator R (definido na Seção 3.1.7) com estes dados recebidos e, em seguida, o cálculo do MOS. Após adicionar este último ao relatório, o conteúdo é persistido no banco de dados.
- *PUT /call-report?id={id}*: Atualiza os dados de um relatório de chamada existente.
- *GET /call-report?id={id}*: Retorna um relatório de chamada específico com base no identificador informado.
- *GET /call-reports*: Retorna a lista completa de relatórios de chamadas cadastrados no sistema.
- *DELETE /call-report?id={id}*: Remove um relatório de chamada do sistema com base no identificador informado.

Os relatórios de chamada armazenados por meio da VoIPTrack QoS contêm métricas de desempenho da chamadas VoIP, como *jitter*, perda de pacotes, *codec* e MOS, servindo como base para a análise dos indicadores de QoS das chamadas

VoIP realizadas pelo sistema. Além disso, também contém informações à respeito dos usuários participantes da chamada.

O cliente responsável pelo envio desses relatórios é um agente usuário VoIP baseado em WebRTC executado diretamente no navegador Web. Durante a chamada, o navegador disponibiliza estatísticas da conexão por meio do método *RTCPeerConnection.getStats()*, permitindo a coleta de métricas. Ao término da chamada, essas métricas são agrupadas e enviadas para a API por meio de uma requisição HTTP do tipo POST. A Listagem 6.1 apresenta um exemplo do corpo da requisição enviada pelo cliente WebRTC ao endpoint */call-report*.

Listing 6.1 – Exemplo de corpo da requisição POST para envio de um relatório de chamada

```
{
  "call_id": "c03c85d6-3d57-41dc-ae37-0f6bfe0a66fb",
  "caller": "1001",
  "callee": "1002",
  "company_id": "2ef9a8e9-9ef2-49ef-8c37-c11d765b29df",
  "call_started_at": "2026-05-14T14:32:18Z",
  "call_ended_at": "2026-05-14T14:35:47Z",
  "packets_received": 15284,
  "packets_lost": 37,
  "jitter": 4.83,
  "rtt": 26.74,
  "codec": "opus"
}
```

Nesse exemplo, são enviados os identificadores da chamada, dos participantes e da empresa, os instantes de início e término da comunicação, bem como as métricas de qualidade coletadas durante a chamada, que servirão de base para o cálculo do fator R e do MOS realizado pela API.

6.2.4 Cálculos

Esta seção descreve como é determinado o MOS das chamadas incluso nos relatórios. A avaliação da qualidade de chamadas de voz foi realizada por meio de uma adaptação simplificada do *E-model* ([International Telecommunication Union, 2015](#)), amplamente utilizado para estimar a percepção de qualidade em sistemas VoIP. A API VoIPTrack QoS utiliza as métricas coletadas pelo cliente WebRTC — como perda de pacotes, RTT, *jitter* e *codec* utilizado — para estimar o fator de degradação causado pela perda de pacotes (*lef*), calcular o R-Factor e, por fim, convertê-lo para o valor de MOS.

6.2.4.1 Cálculo da Taxa de Perda de Pacotes

Inicialmente, é calculada a porcentagem de perda de pacotes da chamada. Para isso, utiliza-se o número de pacotes perdidos e o número de pacotes recebidos durante a sessão:

$$P_{loss} = \frac{PacketsLost}{PacketsLost + PacketsReceived} \times 100 \quad (6.1)$$

, em que:

- *PacketsLost*: representa o total de pacotes perdidos.
- *PacketsReceived*: representa o total de pacotes recebidos.

O resultado é expresso em porcentagem.

6.2.4.2 Cálculo do Fator de Degradação por Perda (*I_{ef}*)

A influência das perdas de pacotes na qualidade percebida é representada pelo fator *I_{ef}*, calculado a partir dos parâmetros específicos do *codec* utilizado:

$$I_{ef} = I_e + (95 - I_e) \frac{P_{loss}}{P_{loss} + B_{pl}} \quad (6.2)$$

, em que:

- *I_e*: é o fator de degradação intrínseca do *codec*.
- *B_{pl}*: representa a robustez do *codec* frente à perda de pacotes.
- *P_{loss}*: é a porcentagem de perda calculada anteriormente.

Para a realização dos cálculos, foram utilizados valores estimados para *I_e* e *B_{pl}* presentes na Tabela 3, definidos com base nas características típicas de cada *codec* conforme referências técnicas do modelo *E-Model* ([International Telecommunication Union, 2015](#)). Esses valores podem variar dependendo da implementação, mas são amplamente usados como aproximação em análises de qualidade de voz.

6.2.4.3 Estimativa do Atraso Unidirecional

O atraso unidirecional (*d*) foi estimado utilizando o tempo de ida e volta (*RTT*) e o *jitter* observado na conexão:

$$d = \frac{RTT}{2} + jitter \quad (6.3)$$

, em que:

Codec	<i>le</i>	<i>Bpl</i>
audio/pcmu	0	34
audio/pcma	0	34
audio/g728	7	17
audio/g729	4	20
audio/iLBC	10	34
audio/BV16	5	25

Tabela 3 – Valores estimados para *le* e *Bpl* para diferentes *codecs*. ([Cable Television Laboratories, Inc., 2004](#)).

- *RTT*: corresponde ao *Round Trip-Time* medido em milissegundos.
- *jitter*: representa a variação temporal entre pacotes consecutivos.

Como as estatísticas disponibilizadas pelo WebRTC não fornecem diretamente o atraso unidirecional da comunicação, este trabalho utiliza uma aproximação baseada na metade do tempo de ida e volta (RTT) somada ao *jitter* observado durante a sessão. Essa estimativa procura representar o atraso efetivamente percebido pelo receptor considerando tanto o tempo médio de propagação quanto a variação temporal dos pacotes.

6.2.4.4 Cálculo do R-Factor

O fator *R* é então calculado utilizando a seguinte equação:

$$R = 94.2 - 0.024d + 0.11(d - 177.3)H(d - 177.3) - I_{ef} \quad (6.4)$$

em que:

- *d*: é o atraso unidirecional em milissegundos.
- *lef*: é o fator de degradação devido ao *codec* e às perdas.
- *H(x)*: é a função degrau de Heaviside:

$$H(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases} \quad (6.5)$$

O valor final do R-Factor é limitado ao intervalo de 0 a 100.

Diferentemente da fórmula completa do *E-model* apresentada na Seção 3.1.7, a implementação desenvolvida considera constantes os termos *R0*, *Is* e *A*, assumindo condições normais de operação da rede e dos equipamentos. Dessa forma, apenas os efeitos do atraso e da degradação causada pelo *codec* e pela perda de pacotes

são considerados variáveis durante o cálculo do R-Factor. Essa simplificação reduz a complexidade computacional e permite estimar a qualidade da chamada utilizando exclusivamente as métricas disponibilizadas pelo WebRTC.

6.2.4.5 Conversão do R-Factor para MOS

Após o cálculo do R-Factor, este é convertido para a escala MOS, que representa a qualidade percebida pelo usuário final:

$$MOS = 1 + 0.035R + 7 \times 10^{-6}R(R - 60)(100 - R) \quad (6.6)$$

O resultado é limitado ao intervalo de 1.0 a 4.5 e avaliado conforme descrito na Tabela 2.

6.3 BANCO DE DADOS

O banco de dados foi modelado com base em três tabelas principais: *companies*, *users* e *reports*, que representam, respectivamente, as empresas cadastradas no sistema, seus funcionários e os relatórios de qualidade das chamadas realizadas entre estes.

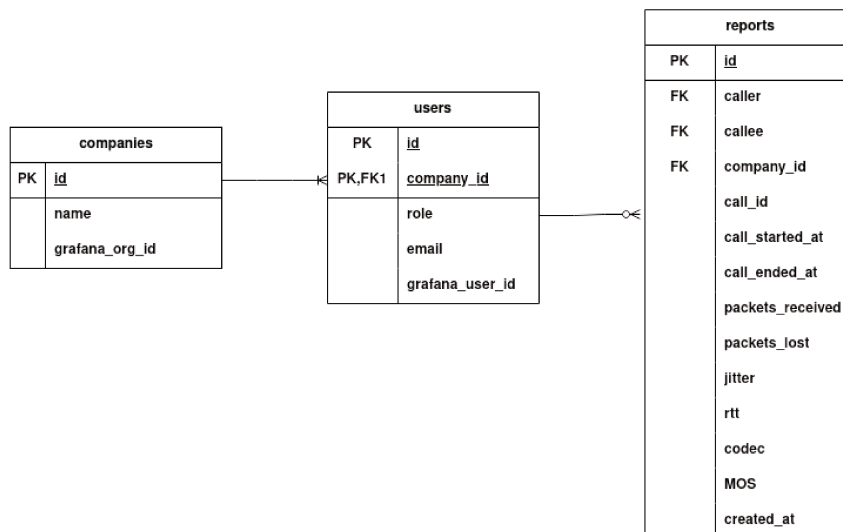


Figura 4 – Tabelas do Banco de dados. (Autorial, 2026).

A tabela *companies* é responsável por armazenar as informações das empresas. Ela possui como chave primária o atributo *id*, do tipo UUID, gerado automaticamente por meio da função *gen_random_uuid()* do PostgreSQL, e o atributo *name*, que armazena o nome da empresa. Essa tabela atua como entidade central para a organização dos dados em um ambiente multiempresa.

A tabela *users* contém os colaboradores pertencentes às empresas cadastradas. Seus principais atributos são *id*, *email*, *role* e *company_id*. A chave primária é composta

pelos campos *id* e *company_id*, garantindo a unicidade de um funcionário dentro de uma determinada empresa. O atributo *role* define o papel desempenhado pelo usuário no sistema, podendo assumir os valores *client*, *admin* ou *provider*, conforme restrição imposta por uma cláusula *CHECK* da tabela. O campo *company_id* é uma chave estrangeira que referencia a tabela *companies*, estabelecendo uma relação de um para muitos, na qual uma empresa pode possuir diversos colaboradores.

A tabela *reports* armazena os relatórios gerados após as chamadas VoIP. Cada registro possui um identificador único (*id*) e informações sobre os participantes da ligação, representados pelos campos *caller* (originador) e *callee* (destinatário). Além disso, são armazenadas métricas relacionadas à qualidade da chamada, como *packets_received*, *packets_lost*, *jitter* e *MOS*, bem como a data de criação do relatório (*created_at*). O campo *company_id* identifica a empresa à qual o relatório pertence.

Os relacionamentos entre as tabelas são garantidos por meio de chaves estrangeiras. O atributo *company_id* da tabela *reports* referencia a tabela *companies*, permitindo associar cada relatório a uma empresa específica. Os campos *caller* e *callee*, em conjunto com *company_id*, referenciam a chave primária da tabela *users*, assegurando que os usuários envolvidos na chamada estejam cadastrados na mesma empresa do relatório.

Conforme o diagrama da Figura 4, uma empresa pode possuir vários colaboradores e vários relatórios, caracterizando relacionamentos do tipo 1:N entre *companies* e *users*, e entre *companies* e *reports*. Além disso, cada colaborador pode participar de múltiplos relatórios, tanto como originador da chamada quanto como destinatário, resultando em relacionamentos 1:N entre *users* e *reports*. As restrições de integridade referencial impedem a criação de relatórios associados a usuários inexistentes e garantem a consistência dos dados armazenados.

Por fim, foram definidas regras de integridade referencial para garantir a consistência dos dados durante operações de exclusão. Quando uma empresa é removida do sistema, todos os usuários e relatórios associados a ela são automaticamente excluídos por meio de um mecanismo de deleção em cascata, evitando a permanência de registros órfãos no banco de dados. Em contrapartida, a exclusão de usuários que possuam relatórios vinculados não é permitida, uma vez que essa restrição foi estabelecida para preservar o histórico das chamadas registradas e manter a integridade das informações armazenadas.

6.4 DASHBOARD DE MONITORAMENTO

Para facilitar a análise dos indicadores de qualidade das chamadas, foi utilizada a plataforma Grafana (descrita na Seção 6.1.4) como ferramenta de visualização dos

dados armazenados pelo sistema. A partir dos relatórios coletados ao término de cada comunicação e armazenados no banco de dados por meio da API de relatórios, o Grafana permite a criação de painéis interativos para acompanhamento das métricas de QoS.

Os painéis desenvolvidos permitem acompanhar graficamente indicadores como latência, *jitter*, perda de pacotes, largura de banda e duração das chamadas. A partir dessas informações, é possível identificar padrões de comportamento, analisar tendências e detectar eventuais degradações na qualidade das comunicações VoIP.

Esses painéis foram construídos manualmente por meio da interface gráfica do Grafana. Durante sua configuração, foram definidas as consultas SQL ao banco de dados PostgreSQL, os painéis de visualização, as variáveis de filtragem e as demais opções necessárias para o monitoramento das métricas de qualidade. Após sua implementação, toda a configuração foi exportada para um arquivo no formato JSON, recurso nativo do Grafana que permite reproduzir a mesma estrutura em outras instâncias da plataforma. Esse arquivo é disponibilizado no repositório do projeto, conforme indicado no Apêndice A.

Além da exibição individual das métricas, o Grafana permite a agregação e filtragem dos dados com base em diferentes critérios, facilitando a comparação entre chamadas e a realização de análises históricas.

A utilização do Grafana contribui para a interpretação dos dados coletados, transformando informações brutas em representações visuais de fácil compreensão.

6.4.1 Controle de Acesso aos Dashboards

Com o objetivo de garantir que cada usuário tenha acesso apenas às informações relevantes ao seu perfil, foi adotado um mecanismo de controle de acesso baseado em papéis (*Role-Based Access Control – RBAC*). Para isso, a tabela de usuários do banco de dados possui o atributo *role*, responsável por definir o nível de acesso de cada usuário ao sistema.

São suportados três tipos de usuários:

- *Client*: representa os clientes finais da plataforma. Usuários com esse perfil podem visualizar exclusivamente os relatórios e métricas referentes às chamadas realizadas por eles próprios.
- *Admin*: representa os administradores de uma empresa cliente. Esse perfil possui acesso aos relatórios de todas as chamadas realizadas pelos usuários pertencentes à sua empresa, permitindo uma visão consolidada do desempenho das comunicações da organização.

- *Provider*: representa os administradores da empresa provedora do serviço. Usuários com esse perfil possuem acesso global aos dados do sistema, podendo visualizar os relatórios e métricas de chamadas de clientes pertencentes a todas as empresas cadastradas na plataforma.

O controle de acesso é implementado diretamente no Grafana por meio das consultas utilizadas nos painéis. Durante a execução das consultas, o sistema identifica o nome de usuário autenticado no Grafana e realiza uma consulta à tabela de usuários para obter o valor do atributo *role* correspondente. Com base nessa informação, são aplicadas condições de filtragem que determinam quais registros podem ser retornados para cada usuário. Esse mecanismo garante que as informações exibidas nos *dashboards* estejam de acordo com as permissões de cada perfil, sem a necessidade de manter painéis completamente distintos para cada tipo de usuário.

A integração entre o banco de dados, a API de relatórios e o Grafana permite que os *dashboards* sejam apresentados de acordo com o papel do usuário autenticado. Dessa forma, cada perfil tem acesso apenas aos dados necessários para suas atividades, ao mesmo tempo em que a empresa provedora mantém uma visão abrangente do desempenho geral.

6.4.2 Painéis desenvolvidos

Com o objetivo de visualizar as principais métricas de qualidade de serviço, foi desenvolvido um *dashboard* no Grafana reunindo os indicadores mais relevantes para o monitoramento das chamadas VoIP. A Figura 5, por exemplo, apresenta painéis, no qual são exibidos as métricas de latência, *jitter*, perda de pacotes e MOS.

A organização dessas métricas em uma única visualização permite uma análise correlacionada dos indicadores de desempenho da rede. Dessa forma, torna-se possível observar, por exemplo, como variações no *jitter* ou na perda de pacotes impactam diretamente o valor do MOS, facilitando a identificação das causas de degradação na qualidade das chamadas. Além disso, os valores apresentados podem ser comparados aos limites de referência descritos na Seção 3.1.5, permitindo avaliar se as condições observadas durante as chamadas encontram-se dentro dos parâmetros recomendados para aplicações VoIP.

Os gráficos são apresentados em função do tempo, possibilitando o acompanhamento da evolução das métricas dentro de um intervalo configurável no Grafana. Esse recurso permite a análise tanto de eventos pontuais quanto de tendências ao longo do tempo, contribuindo para a detecção de padrões recorrentes de instabilidade.

Foi desenvolvido um painel para análise da evolução do MOS ao longo do tempo, conforme Figura 6. Esse painel permite uma visualização mais detalhada da qualidade



Figura 5 – Médias de MOS, de *jitter*, de perda de pacotes e de latência para um determinado cliente. (Autorial, 2026).

percebida das chamadas, destacando variações ao longo do período selecionado. Por meio dessa abordagem, torna-se possível identificar momentos de degradação contínua ou intermitente, bem como avaliar o impacto de eventos na infraestrutura sobre a experiência do usuário final. De forma similar, foram desenvolvidos painéis com a evolução da latência, do *jitter* e da perda de pacotes.

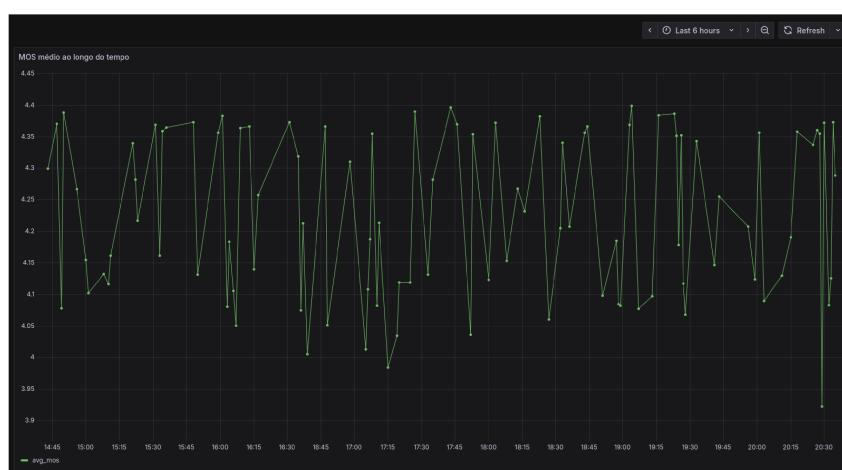


Figura 6 – Médias de MOS ao longo do tempo para uma determinada empresa. (Autorial, 2026).

A fim de monitorar a quantidade de chamadas simultâneas, foi implementado o painel apresentado na Figura 7. Esse painel apresenta o número de chamadas ativas em um determinado intervalo de tempo, permitindo avaliar o nível de utilização do sistema. A análise desse indicador é importante para identificar possíveis situações de sobrecarga, nas quais o aumento no volume de chamadas pode influenciar

negativamente a qualidade do serviço.

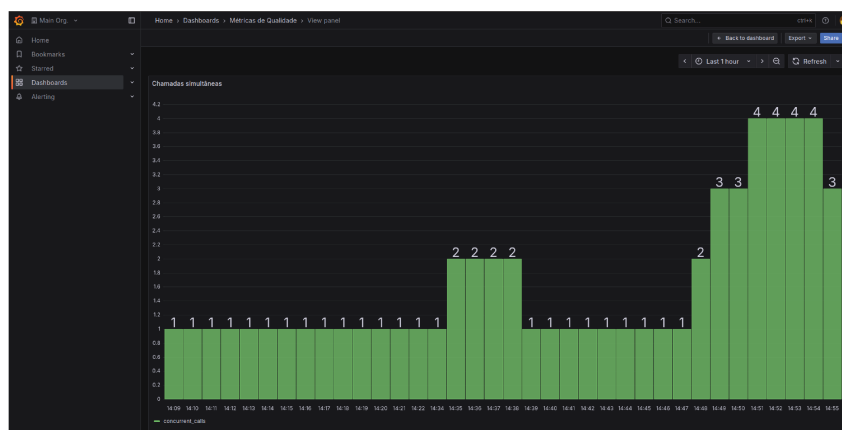


Figura 7 – Chamadas simultâneas em uma determinada empresa. (Autorial, 2026).

Ainda no contexto de análise de utilização do sistema, foi desenvolvido o painel apresentado na Figura 8. Ele apresenta os dez usuários que possuem os maiores volumes de chamadas na empresa.

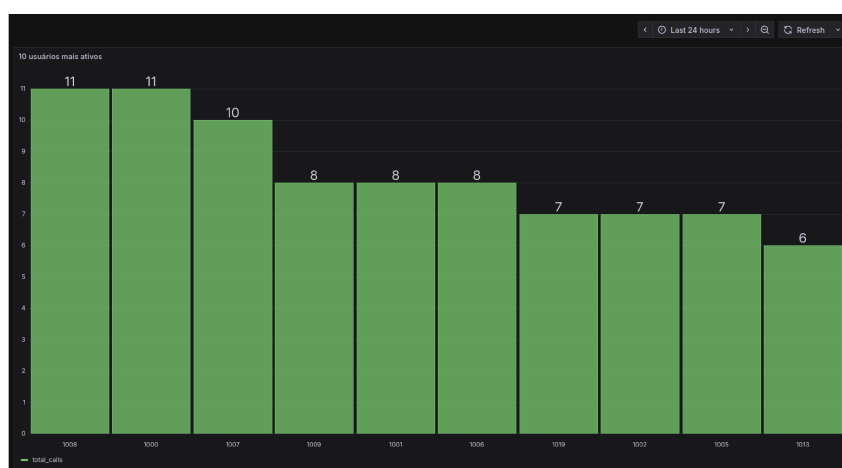


Figura 8 – Usuários mais ativos em uma determinada empresa. (Autorial, 2026).

Por fim, visando facilitar o monitoramento da qualidade das chamadas VoIP, o painel apresentado na Figura 9 foi implementado. Ele é responsável por apresentar o conteúdo do relatório de todas as chamadas que obtiveram MOS menor que 3. Assim, é possível identificar padrões e causas por meio da correlação com as outras métricas do relatório.

A combinação desses painéis proporciona uma visão abrangente do sistema, permitindo não apenas a análise da qualidade das chamadas, mas também a correlação entre o desempenho da rede e o volume de utilização. Assim, o *dashboard* desenvolvido contribui significativamente para o monitoramento contínuo e para a tomada de decisões relacionadas à manutenção e otimização da infraestrutura de comunicação.

call_id	caller	callee	mos	jitter	rt	packets_lost
00e982d3-6b2a-4c67-992a-34c32068928c	1008	1014	2.63	117	252	487
feb51e6-5eca-40f0-81f7-3adb582e3b45	1000	1005	2.46	127	460	425
1616da79-7dee-4d6e-9b0b-a0b6787a594	1002	1003	2.60	100	293	460
a0ba0c7-c732-4573-9793-d68877a4ba64	1002	1017	2.92	140	238	571
17787aa8-adb9-45f3-98b2-9008b33250a8	1006	1003	2.82	162	278	518
c28862d0-3afe-46d9-8626-608b3576fd	1018	1007	2.18	147	284	312
a0145898-5220-489a-8af8-131673e5025	1012	1005	2.31	104	250	297
eb82b902-1552-4678-8dc8-25a67317d6e4	1009	1008	2.30	166	305	565
0d968f0e-d98e-40b1-ad9b-78aee4d3b5be	1014	1012	2.01	123	281	454
brd547df-9422-49a8-bc9e-55bd2669424e	1000	1001	2.89	184	340	478
c6c5a7c2-b838-46a9-ba1b-f8c96b20a66	1002	1001	3.00	195	606	581
cafe7d95-b081-47db-a18e-d1e4d6d7b67	1002	1009	2.58	177	484	463
f6738f15-27f0-4fec-bc14-2af38fc5c6d	1009	1008	1.98	84	411	444
ac184d0-4436-4526-8770-62ac7bde4234	1002	1001	2.33	178	252	399
9377b65f-8a08-4522-443e-3f6073b0f45	1000	1007	2.70	185	401	519
fe4f92c1-9bfe-44b5-9a38-a9b23a258965	1005	1007	2.16	80	488	470
bde8fa7b-41c7-4fde-af6d-883557c6b754	1004	1001	2.47	190	444	424
7edfbbab-9347-4c16-873f-fa2847b0d1f9	1005	1008	2.69	127	543	481
58a6a081-7aad-442a-8241-10c02a4af633	1004	1005	2.05	198	323	529
aac22227-2817-4d78-bf30-b20b6997e95	1000	1007	2.01	117	264	398

Figura 9 – Chamadas com MOS abaixo de 3. (Aural, 2026).

6.4.3 Sistema de Alertas

Além da visualização dos indicadores de QoS, o Grafana também é utilizado como ferramenta de geração de alertas. Por meio da definição de regras baseadas nas métricas coletadas nas chamadas, a plataforma é capaz de identificar automaticamente situações que indiquem degradação da qualidade da comunicação.

No caso deste presente trabalho, foi adicionada uma regra de alerta que monitora especificamente o indicador MOS das chamadas registradas, conforme mostrado na Figura 10. O sistema avalia uma janela temporal correspondente à última hora e verifica a ocorrência de chamadas com degradação significativa de qualidade — aqui definida para valores de MOS abaixo de 3. O alerta é disparado quando são identificados pelo menos três registros de chamadas que ultrapassam esse limiar dentro desse intervalo de tempo, caracterizando uma condição recorrente de baixa qualidade de serviço e não apenas um evento isolado.

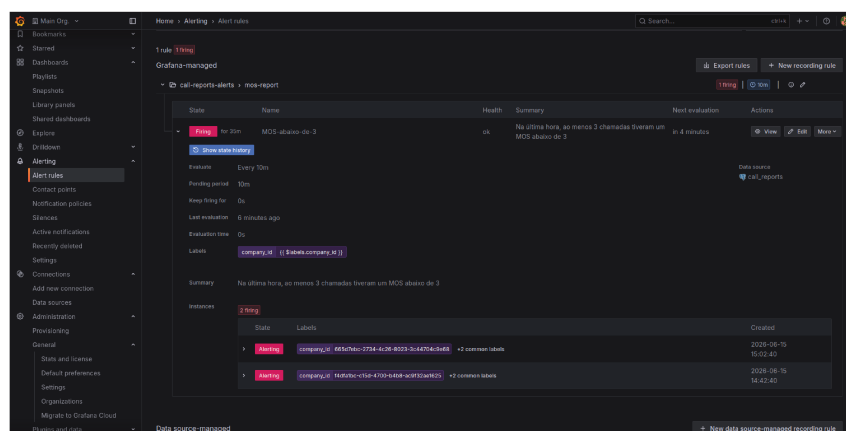


Figura 10 – Alerta responsável por monitorar o MOS. (Aural, 2026).

Além do critério temporal e do limiar de qualidade, os alertas são separados por empresa por meio do atributo *company_id*, utilizado como *label* na consulta. Essa

separação permite que cada organização seja monitorada de forma independente, garantindo que problemas de qualidade sejam atribuídos corretamente ao respectivo contexto organizacional.

Quando a condição definida pela regra é satisfeita, o Grafana gera automaticamente uma notificação. Essas notificações podem ser direcionadas a diferentes perfis de usuários, como administradores do sistema ou responsáveis pela empresa afetada, dependendo da configuração do canal de alerta. Dessa forma, o sistema possibilita uma resposta rápida a problemas de degradação de qualidade, contribuindo para a manutenção da estabilidade e confiabilidade do serviço de comunicação.

6.5 AMBIENTE DE TESTES

Para a realização dos experimentos foi desenvolvido um ambiente de testes baseado em contêineres Docker. A infraestrutura é composta por cinco serviços principais: dois clientes WebRTC, a API de coleta e armazenamento de métricas, um banco de dados PostgreSQL e uma instância Grafana para visualização dos resultados. A comunicação entre os componentes ocorre conforme o fluxo ilustrado na Figura 11.

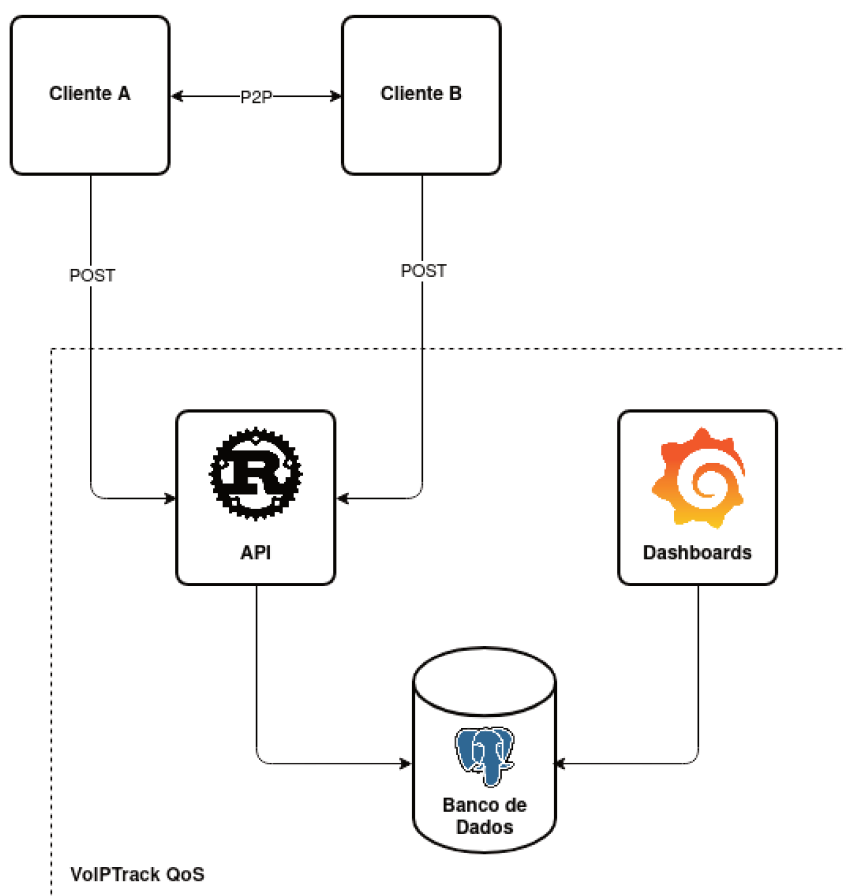


Figura 11 – Infraestrutura do Ambiente de Testes. (Autorial, 2026).

Cada contêiner desempenha uma função específica na arquitetura proposta,

permitindo a execução isolada dos componentes e a reprodução consistente dos experimentos.

Os contêineres Cliente A e Cliente B atuam como participantes da comunicação WebRTC, sendo responsáveis pelo estabelecimento da conexão e pela coleta das estatísticas disponibilizadas nativamente pela API WebRTC. Durante a comunicação, o navegador mantém informações atualizadas sobre diversos parâmetros da conexão, incluindo *jitter*, perda de pacotes, RTT, bytes transmitidos e recebidos, codec utilizado e outras estatísticas relacionadas ao fluxo RTP.

Após o encerramento da chamada, essas informações são agrupadas pelo cliente e enviadas automaticamente à API VoIPTrack QoS por meio de uma requisição HTTP POST para o *endpoint* `/call-report`. A API então valida os dados recebidos, calcula o MOS e armazena o relatório para posterior visualização nos dashboards do Grafana.

A API de relatórios é responsável por receber as métricas produzidas pelos clientes por meio de requisições HTTP e realizar a persistência dessas informações no banco de dados.

O serviço PostgreSQL armazena todas as métricas coletadas durante os experimentos, permitindo consultas posteriores para análise e visualização dos resultados.

Por fim, o Grafana foi utilizado para a construção de *dashboards* e painéis de monitoramento, consumindo os dados armazenados no PostgreSQL e possibilitando a análise visual do comportamento do sistema durante os testes.

6.6 TESTES REALIZADOS

Esta seção apresenta os testes realizados para validar o funcionamento do sistema VoIPTrack QoS. Os experimentos tiveram como objetivo verificar se os componentes desenvolvidos executam corretamente as funcionalidades especificadas, desde a coleta das métricas de qualidade pelo cliente WebRTC até o armazenamento dos relatórios e sua posterior visualização nos painéis do Grafana.

Os testes foram conduzidos em um ambiente controlado, utilizando a infraestrutura descrita na Figura 11. Durante os experimentos, foram realizadas chamadas VoIP entre usuários cadastrados no sistema, possibilitando a geração de relatórios de qualidade e a validação do fluxo completo da solução.

6.6.1 Teste funcional do fluxo completo

Os testes funcionais tiveram como objetivo verificar se o sistema atende aos requisitos especificados, validando o correto funcionamento de cada etapa do processo

de monitoramento da qualidade das chamadas VoIP.

Para isso, foram realizados experimentos de simulação de chamadas no ambiente de testes. Durante cada chamada, o cliente WebRTC coletou automaticamente as métricas de qualidade disponibilizadas pelo navegador. Ao término da comunicação, essas métricas foram enviadas à API VoIPTrack QoS por meio de uma requisição HTTP do tipo *POST*.

Após o recebimento dos dados, foi verificado que a API realizou corretamente a validação do relatório, o cálculo do fator R e do valor estimado de MOS, bem como o armazenamento das informações no banco de dados PostgreSQL. Em seguida, confirmou-se que os relatórios ficaram imediatamente disponíveis para consulta nos dashboards do Grafana, permitindo a visualização das métricas coletadas e dos indicadores de qualidade da chamada.

Os resultados obtidos demonstraram que o fluxo completo da solução operou conforme o esperado, desde a coleta das métricas no cliente até sua apresentação gráfica, validando o funcionamento dos principais componentes do sistema proposto.

6.6.2 Teste do dashboard com dados simulados

Além dos testes realizados com chamadas VoIP reais no ambiente de testes, foi desenvolvido um *script* para popular automaticamente o banco de dados PostgreSQL com relatórios de chamadas simulados. O objetivo desse experimento foi disponibilizar um volume maior de dados para avaliar o comportamento do *dashboard* implementado no Grafana.

O *script* gera registros contendo valores para as principais métricas monitoradas pelo sistema, como *jitter*, latência, perda de pacotes, duração da chamada e MOS, simulando diferentes cenários de qualidade de comunicação. Esses registros são inseridos diretamente no banco de dados, permitindo que os painéis do Grafana sejam avaliados sem a necessidade de realizar um grande número de chamadas reais.

Com esse conjunto de dados, foi possível verificar o correto funcionamento das consultas SQL utilizadas pelos painéis, dos filtros de visualização, dos gráficos e das demais funcionalidades de monitoramento implementadas. O experimento também permitiu avaliar a apresentação das informações em diferentes períodos de tempo e identificar possíveis inconsistências na configuração dos painéis durante o desenvolvimento.

6.7 LIMITAÇÕES

Embora a arquitetura proposta apresente vantagens em relação às abordagens baseadas na captura passiva de tráfego vistas na Seção 5, ela possui algumas limitações. A principal delas é a necessidade de integração entre o sistema VoIP da empresa e a API VoIPTrack QoS para o envio dos relatórios de qualidade das chamadas.

Como a coleta das métricas é realizada diretamente pelo agente usuário por meio das estatísticas disponibilizadas pelo WebRTC, é necessário que o cliente VoIP implemente a lógica responsável por obter essas informações e transmiti-las à API ao término de cada chamada. Dessa forma, a utilização do VoIPTrack QoS depende da possibilidade de modificar ou estender o cliente utilizado pela organização.

Essa característica pode dificultar a adoção da solução em ambientes que utilizam softphones proprietários ou aplicações de terceiros que não disponibilizam mecanismos para extração das estatísticas de qualidade ou para integração com sistemas externos. Nesses cenários, seria necessário desenvolver adaptações específicas ou utilizar um cliente compatível com a arquitetura proposta.

7 CONCLUSÃO

O presente trabalho abordou o crescente papel das tecnologias de comunicação baseadas em IP, com destaque para o VoIP, evidenciando seus benefícios, desafios e a importância da garantia de qualidade nas transmissões em tempo real. A substituição gradual dos sistemas tradicionais de telefonia por soluções baseadas em rede IP demonstra uma tendência consolidada, impulsionada principalmente pela redução de custos e pela flexibilidade proporcionada.

Entretanto, conforme discutido ao longo deste estudo, a qualidade das chamadas VoIP está diretamente associada às condições da rede, sendo impactada por fatores como latência, *jitter* e perda de pacotes. Esses elementos tornam evidente a necessidade de mecanismos eficientes de monitoramento de Qualidade de Serviço, especialmente em ambientes corporativos e aplicações críticas, como *call centers*.

Diante desse cenário, este trabalho atingiu seu objetivo principal ao especificar e implementar o protótipo do sistema VoIPTrack QoS, uma solução capaz de coletar, processar e visualizar métricas relevantes de qualidade de chamadas. A arquitetura proposta, baseada em componentes distribuídos mostrou-se adequada para garantir escalabilidade, organização e facilidade de manutenção.

Um dos principais diferenciais da solução desenvolvida está na sua independência em relação ao posicionamento na rede e ao acesso ao conteúdo dos pacotes, o que a torna aplicável em cenários modernos que envolvem mobilidade e criptografia, como o uso de SRTP. Essa característica supera limitações comuns em ferramentas tradicionais de monitoramento passivo, ampliando o campo de aplicação do sistema.

Além disso, a integração com o Grafana possibilitou a criação de dashboards interativos e sistemas de alerta, permitindo não apenas a visualização das métricas, mas também a identificação proativa de problemas de qualidade.

Os testes realizados em ambiente controlado validaram a capacidade do sistema em coletar e apresentar dados relevantes, bem como em identificar cenários de degradação da qualidade. Dessa forma, a solução proposta se mostra útil como ferramenta de apoio à tomada de decisão, permitindo ações corretivas rápidas e contribuindo para a melhoria contínua da infraestrutura de rede.

REFERÊNCIAS

BASHARIRAD, B.; BHATTI, H.; AHMADI, M. An introduction to docker and analysis of its performance. *IJCSNS International Journal of Computer Science and Network Security*, v. 173, p. 8, 03 2017. Cited on page 31.

BOETTIGER, C. An introduction to docker for reproducible research. *SIGOPS Oper. Syst. Rev.*, Association for Computing Machinery, New York, NY, USA, v. 49, n. 1, p. 71–79, jan. 2015. ISSN 0163-5980. Disponível em: <<https://doi.org/10.1145/2723872.2723882>>. Cited on page 31.

BUGDEN, W.; ALAHMAR, A. *Rust: The Programming Language for Safety and Performance*. 2022. Disponível em: <<https://arxiv.org/abs/2206.05503>>. Cited on page 32.

Cable Television Laboratories, Inc. *PacketCable 1.5 Specifications: Audio/Video Codecs*. 2004. Acesso em: 10 jun. 2026. Disponível em: <<https://account.cablelabs.com/server/alfresco/08957a5c-c2e6-44ac-8d7a-bdbb34a7c232>>. Cited on page 38.

Docker Inc. *Docker Documentation*. 2025. Acesso em: 15 jun. 2026. Disponível em: <<https://docs.docker.com/>>. Cited on page 31.

Dotcom-Monitor. *Dotcom-Monitor: Advanced Website Monitoring Solutions*. 2025. Acesso em: 24 jun. 2025. Disponível em: <<https://www.dotcom-monitor.com/>>. Cited on page 29.

ELRADI, M. D. Prometheus grafana: A metrics-focused monitoring stack. *Journal of Computer Allied Intelligence(JCAI, ISSN: 2584-2676)*, v. 3, n. 3, p. 28–39, Jun. 2025. Disponível em: <<https://fringeglobal.com/ojs/index.php/jcai/article/view/prometheus-grafana-a-metrics-focused-monitoring-stack>>. Cited on page 32.

Grafana Labs. *Grafana HTTP API Reference*. 2025. Acesso em: 15 jun. 2026. Disponível em: <<https://grafana.com/docs/grafana/latest/developer-resources/api-reference/http-api/>>. Cited on page 32.

HARTPENCE, B. *Packet guide to Voice over IP*. [S.l.]: Beijing; Cambridge; Farnham; Koln; Sebastopol; Tokyo: O'reilly, 2013. Cited on page 8.

International Telecommunication Union. *Methods for subjective determination of transmission quality*. 1996. ITU-T Recommendation P.800. Accessed: 2025-06-15. Disponível em: <<https://www.itu.int/rec/T-REC-P.800-199608-I/en>>. Cited on page 20.

International Telecommunication Union. *ITU-T Recommendation G.1010: End-user multimedia QoS categories*. Geneva, Switzerland, 2001. Cited on page 19.

International Telecommunication Union. *ITU-T Recommendation G.114: One-way transmission time*. Geneva, Switzerland, 2003. Cited on page 18.

International Telecommunication Union. *The E-model: a computational model for use in transmission planning*. 2015. ITU-T Recommendation G.107. Accessed: 2025-06-15.

Disponível em: <<https://www.itu.int/rec/T-REC-G.107-201506-I/en>>. Cited 4 times on pages 20, 21, 36 e 37.

JsSIP. *JsSIP - The JavaScript SIP Library*. 2023. Acesso em: 26 jun. 2025. Disponível em: <<https://jssip.net/>>. Cited on page 27.

JUBA, S.; VANNAHME, A.; VOLKOV, A. *Learning PostgreSQL: Create, develop and manage relational databases in real world applications using PostgreSQL*. Packt Publishing, 2015. ISBN 9781783989195. Disponível em: <<https://books.google.com.br/books?id=jfKoCwAAQBAJ>>. Cited on page 32.

KAMAILIO. *The Kamailio SIP Server Project – The Open Source SIP Server*. 2025. Acesso em: 30 jun. 2025. Disponível em: <<https://www.kamailio.org/>>. Cited on page 24.

KUROSE, J. F.; ROSS, K. W. *Redes de computadores e a internet: uma abordagem top-down*. 6. ed. São Paulo: Pearson, 2013. ISBN 978-8581436777. Cited on page 12.

Mozilla. *RTCPeerConnection.getStats()* – Web APIs | MDN. 2024. Acesso em: 05 jul. 2025. Disponível em: <<https://developer.mozilla.org/en-US/docs/Web/API/RTCPeerConnection/getStats>>. Cited on page 26.

Mozilla. *WebRTC API*. 2025. Acesso em: 19 jun. 2025. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API>. Cited on page 26.

NISTICò, A. et al. A comparative study of rtc applications. In: *2020 IEEE International Symposium on Multimedia (ISM)*. [S.l.: s.n.], 2020. p. 1–8. Cited on page 25.

PERKINS, C.; HODSON, O.; HARDMAN, V. A survey of packet loss recovery techniques for streaming audio. In: *Readings in Multimedia Computing and Networking*. San Francisco: Morgan Kaufmann, 2002, (The Morgan Kaufmann Series in Multimedia Information and Systems). p. 607–615. ISBN 978-1-55860-651-7. Disponível em: <<https://www.sciencedirect.com/science/article/pii/B978155860651750139X>>. Cited 3 times on pages 4, 21 e 22.

ROSENBERG, J. et al. *SIP: Session Initiation Protocol*. 2002. RFC 3261. Accessed: 2025-06-17. Disponível em: <<https://datatracker.ietf.org/doc/html/rfc3261>>. Cited 3 times on pages 12, 13 e 15.

Rust Project Developers. *crates.io: Rust Package Registry*. 2026. Acesso em: 15 jun. 2026. Disponível em: <<https://crates.io/>>. Cited on page 32.

Rust Project Developers. *The Rust Programming Language*. 2026. Acesso em: 15 jun. 2026. Disponível em: <<https://www.rust-lang.org/>>. Cited on page 31.

Sangoma Technologies. *Home Asterisk*. 2026. Acesso em: 9 jun. 2026. Disponível em: <<https://www.asterisk.org/>>. Cited on page 23.

SCHULZRINNE, H. et al. *RTP: A Transport Protocol for Real-Time Applications*. 2003. RFC 3550. Accessed: 2025-06-17. Disponível em: <<https://datatracker.ietf.org/doc/html/rfc3550>>. Cited 2 times on pages 13 e 14.

SignalWire. *FreeSWITCH*. 2024. Acesso em: 30 jun. 2025. Disponível em: <https://signalwire.com/freeswitch>. Cited on page 23.

SIP3. *SIP3 - Monitor your VoIP and RTC traffic real-time*. 2025. Acesso em: 24 jun. 2025. Disponível em: <https://www.sip3.io/>. Cited on page 28.

SIP.js. *SIP Signaling JavaScript Library for WebRTC Developers*. 2022. Acesso em: 26 jun. 2025. Disponível em: <https://sipjs.com/>. Cited on page 26.

STALLINGS, W. *Data and Computer Communications*. 10. ed. Harlow, Essex: Pearson, 2014. ISBN 9780133506488. Cited 2 times on pages 4 e 15.

TANENBAUM, A. S.; FEAMSTER, N. *Computer Networks*. 6. ed. [S.l.]: Pearson, 2019. ISBN 9780135408001. Cited 2 times on pages 12 e 17.

VoIPmonitor. *VoIPmonitor® - VoIP monitoring software - quality analyzer - WAV recorder*. 2025. Acesso em: 24 jun. 2025. Disponível em: <https://www.voipmonitor.org/>. Cited on page 28.

World Wide Web Consortium (W3C). *WebRTC: Real-Time Communication in Browsers*. 2025. Acesso em: 24 jun. 2025. Disponível em: <https://www.w3.org/TR/webrtc/>. Cited on page 25.

APÊNDICE A - CÓDIGO FONTE

O código fonte do Sistema VoIPTrack QoS, bem como as configurações realizadas no Grafana neste presente trabalho, estão disponíveis no repositório e podem ser acessados pelo link <https://github.com/patbardini/voiptrack-qos>

APÊNDICE B - ARTIGO

Artigo no formato especificado pela Sociedade Brasileira de Computação (SBC).

VoIPTrack QoS: Sistema para Monitoramento de QoS de Chamadas VoIP

Patrícia Bardini Arigoni¹

¹Departamento de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)
Florianópolis – SC – Brasil

patricia.bardini@grad.ufsc.br

Abstract. *This work aims to implement a prototype called VoIPTrack QoS, capable of providing information on the call quality of a VoIP service provider's customers and recommending mitigating actions, even in scenarios with encrypted traffic and mobile devices, overcoming the limitations of current solutions. The proposed system continuously monitors quality-related metrics, enabling providers to identify performance degradation, diagnose potential causes, and take proactive actions to improve the user experience. By eliminating the need for traffic interception or access to encrypted media streams, the solution broadens the applicability of VoIP quality monitoring in modern communication environments.*

Resumo. *Este trabalho tem como objetivo implementar um protótipo chamado VoIPTrack QoS, capaz de fornecer informações sobre a qualidade das chamadas dos clientes de um provedor de serviços VoIP e recomendar ações de mitigação, mesmo em cenários com tráfego criptografado e dispositivos móveis, superando as limitações das soluções atuais. O sistema proposto monitora continuamente métricas relacionadas à qualidade, permitindo que os provedores identifiquem a degradação do desempenho, diagnostiquem as causas potenciais e tomem medidas proativas para melhorar a experiência do usuário. Ao eliminar a necessidade de interceptação de tráfego ou acesso a fluxos de mídia criptografados, a solução amplia a aplicabilidade do monitoramento da qualidade de VoIP em ambientes de comunicação modernos.*

1. Introdução

A tecnologia *Voice over Internet Protocol* (VoIP) consolidou-se como uma das principais soluções para comunicação de voz em redes IP, sendo amplamente utilizada em ambientes corporativos, provedores de serviços de telefonia e aplicações de comunicação em tempo real [Hartpence 2013, Kurose and Ross 2013]. Em comparação com a telefonia tradicional baseada em comutação de circuitos, o VoIP oferece maior flexibilidade, facilidade de integração com outros serviços e redução significativa de custos operacionais, fatores que impulsionaram sua adoção em diferentes cenários [Hartpence 2013, Tanenbaum and Feamster 2019].

Apesar dessas vantagens, a qualidade das chamadas VoIP depende diretamente das condições da rede utilizada para o transporte dos pacotes de voz. Diferentemente das redes telefônicas convencionais, as redes IP não garantem, por padrão, requisitos de Qualidade de Serviço (*Quality of Service – QoS*), tornando as chamadas suscetíveis a

problemas como latência elevada, variação de atraso (*jitter*), perda de pacotes e limitações de largura de banda. Esses fatores podem comprometer a inteligibilidade da comunicação e degradar a experiência percebida pelos usuários.

Nesse contexto, o monitoramento contínuo da QoS torna-se uma atividade essencial para organizações que dependem da comunicação VoIP. A coleta e análise de métricas como latência, *jitter*, perda de pacotes e *Mean Opinion Score* (MOS) permitem identificar degradações no serviço, diagnosticar suas possíveis causas e subsidiar ações corretivas antes que problemas sejam percebidos pelos usuários finais.

Entretanto, as principais soluções de monitoramento disponíveis atualmente baseiam-se na captura passiva do tráfego de rede [VoIPmonitor 2025, SIP3 2025], exigindo que o sistema de monitoramento esteja posicionado no caminho das chamadas. Essa abordagem apresenta limitações em cenários modernos, especialmente quando há utilização de dispositivos móveis, arquiteturas distribuídas ou tráfego criptografado por protocolos como o *Secure Real-time Transport Protocol* (SRTP) [Baugher et al. 2004], nos quais o acesso ao conteúdo dos pacotes pode não ser possível.

Diante desse cenário, este trabalho apresenta o VoIPTrack QoS, um sistema para monitoramento da qualidade de chamadas VoIP baseado na coleta de estatísticas diretamente dos agentes usuários, dispensando a inspeção do tráfego de mídia. A solução utiliza uma API REST para armazenamento dos relatórios de qualidade, um banco de dados PostgreSQL para persistência das informações e dashboards desenvolvidos no Grafana para visualização das métricas e geração de alertas. Dessa forma, o sistema permite acompanhar a qualidade das chamadas mesmo em ambientes com tráfego criptografado e usuários móveis, ampliando o conjunto de cenários nos quais o monitoramento de QoS pode ser realizado.

2. Conceitos Fundamentais

Esta seção apresenta os principais conceitos necessários para a compreensão da solução proposta.

2.1. VoIP

O VoIP é uma tecnologia que permite a transmissão de voz por meio de redes baseadas no Protocolo de Internet (IP), substituindo a infraestrutura tradicional de telefonia comutada. Nesse modelo, o sinal de voz é convertido em dados digitais, segmentado em pacotes e transmitido pela rede até o destinatário, onde é reconstruído para reprodução [Hartpence 2013, Kurose and Ross 2013]. Essa abordagem possibilita que voz, vídeo e outros tipos de mídia compartilhem a mesma infraestrutura de comunicação, reduzindo custos e simplificando a integração entre diferentes serviços.

A adoção do VoIP cresceu significativamente devido às vantagens econômicas e operacionais em relação aos sistemas convencionais de telefonia. Além da redução dos custos de comunicação, especialmente em chamadas de longa distância, a tecnologia oferece maior flexibilidade, permitindo que usuários realizem chamadas a partir de computadores, smartphones ou telefones IP conectados à Internet. Atualmente, o VoIP é amplamente utilizado em centrais telefônicas IP, provedores de telefonia, *call centers* e aplicações de comunicação em tempo real.

2.2. Protocolos utilizados no VoIP

O funcionamento de uma comunicação VoIP depende da cooperação entre protocolos responsáveis pela sinalização da chamada, transporte da mídia e monitoramento da qualidade da comunicação.

O principal protocolo de sinalização é o *Session Initiation Protocol* (SIP), definido pela RFC 3261 [Rosenberg et al. 2002]. O SIP é responsável por estabelecer, modificar e finalizar sessões multimídia entre usuários. Durante o processo de comunicação, mensagens como REGISTER, INVITE, ACK e BYE permitem registrar usuários em um servidor SIP, iniciar chamadas, confirmar o estabelecimento da sessão e encerrá-la. O protocolo não transporta o áudio da chamada, sendo utilizado exclusivamente para controlar o ciclo de vida da sessão.

Após o estabelecimento da sessão SIP, a transmissão da voz é realizada pelo *Real-time Transport Protocol* (RTP) [Schulzrinne et al. 2003]. Esse protocolo foi desenvolvido especificamente para aplicações multimídia em tempo real, priorizando baixa latência em detrimento da confiabilidade absoluta da entrega dos pacotes. Dessa forma, pacotes que chegam com atraso excessivo são descartados, evitando que retransmissões comprometam a fluidez da comunicação.

Complementando o RTP, o *Real-Time Transport Control Protocol* (RTCP) [Schulzrinne et al. 2003] fornece informações de controle e estatísticas sobre a qualidade da transmissão. Por meio de relatórios periódicos, o RTCP disponibiliza métricas como perda de pacotes, *jitter*, atraso e quantidade de pacotes transmitidos, permitindo que aplicações monitorem continuamente as condições da comunicação. Essas informações são amplamente utilizadas por sistemas de monitoramento de QoS, auxiliando na identificação de degradações durante as chamadas.

2.3. WebRTC

O *Web Real-Time Communication* (WebRTC) é um conjunto de tecnologias e APIs que possibilita a comunicação de áudio, vídeo e dados em tempo real diretamente entre navegadores, sem a necessidade de plugins ou aplicações externas [World Wide Web Consortium (W3C) 2025, Mozilla 2025]. Padronizado pelo W3C, o WebRTC tornou-se uma das principais soluções para implementação de aplicações VoIP baseadas na Web.

Embora o WebRTC seja responsável pelo transporte da mídia entre os participantes, ele não define um protocolo de sinalização [World Wide Web Consortium (W3C) 2025]. Assim, tecnologias como SIP sobre WebSocket podem ser utilizadas para estabelecer as sessões de comunicação, enquanto o WebRTC assume a transmissão da mídia utilizando protocolos como RTP e mecanismos de segurança baseados em DTLS e SRTP.

O estabelecimento de uma conexão WebRTC ocorre em três etapas principais. Inicialmente, os participantes trocam informações de sinalização contendo descrições da sessão (SDP). Em seguida, o protocolo *Interactive Connectivity Establishment* (ICE) identifica a melhor rota de comunicação entre os dispositivos, podendo utilizar servidores STUN e TURN [Mozilla 2025] para atravessar dispositivos NAT e *firewalls*. Após essa negociação, inicia-se a transmissão direta da mídia entre os participantes.

Além de disponibilizar APIs para captura de áudio e vídeo e gerenciamento das conexões, o WebRTC oferece o método *getStats()* [Mozilla 2024], pertencente à interface *RTCPeerConnection*, que permite acessar estatísticas detalhadas da comunicação em tempo real. Entre as informações disponibilizadas estão latência, *jitter*, perda de pacotes, taxa de transmissão e quantidade de bytes enviados e recebidos. Essas métricas constituem uma importante fonte de informações para avaliação de QoS, sendo utilizadas neste trabalho para alimentar o sistema VoIPTrack QoS sem a necessidade de inspeção passiva do tráfego da rede.

3. QoS em VoIP

A Qualidade de Serviço (QoS) em sistemas VoIP corresponde à capacidade da rede em fornecer condições adequadas para o transporte de tráfego multimídia em tempo real. Diferentemente de aplicações convencionais de transferência de dados, comunicações de voz são sensíveis a atrasos, variações no tempo de entrega dos pacotes e perdas durante a transmissão. Dessa forma, a avaliação da QoS baseia-se em métricas objetivas que permitem identificar degradações na comunicação e estimar a qualidade percebida pelo usuário.

Uma das principais métricas é a latência, que representa o tempo necessário para que um pacote percorra o caminho entre origem e destino. Em chamadas de voz, atrasos elevados prejudicam a conversação, ocasionando sobreposição de falas e reduzindo a naturalidade da comunicação. De acordo com a recomendação ITU-T G.114 [International Telecommunication Union 2003], atrasos inferiores a 150 ms são considerados adequados para a maioria das aplicações de voz em tempo real.

Outra métrica importante é o *jitter*, definido como a variação do atraso entre pacotes consecutivos. Em redes IP, os pacotes podem percorrer caminhos distintos ou sofrer tempos diferentes de enfileiramento, chegando ao destino em intervalos irregulares. Essa variação pode provocar interrupções, distorções e falhas na reprodução do áudio. Para minimizar seus efeitos, aplicações VoIP normalmente utilizam *buffers* de *jitter*, que armazenam temporariamente os pacotes antes de sua reprodução.

A perda de pacotes ocorre quando parte dos pacotes transmitidos não chega ao destino devido a congestionamentos, falhas de enlace ou descarte em equipamentos de rede. Embora *codecs* modernos possuam mecanismos para reduzir o impacto dessas perdas, valores elevados comprometem significativamente a inteligibilidade da fala. Em condições normais, perdas inferiores a 1% são praticamente imperceptíveis, enquanto perdas superiores a 5% tendem a causar degradação perceptível da qualidade da comunicação.

Para representar a experiência do usuário de forma mais intuitiva, utiliza-se o MOS, uma métrica que expressa a qualidade percebida da chamada em uma escala de 1 a 5, conforme a Tabela 1.

Atualmente, o MOS pode ser estimado automaticamente a partir do *E-model*, padronizado pela recomendação ITU-T G.107 [International Telecommunication Union 2015], que considera fatores como atraso, perdas de pacotes e características dos *codecs* para calcular o chamado Fator R. A partir desse índice é possível estimar objetivamente a qualidade percebida da comunicação sem a necessidade de avaliações subjetivas realizadas por usuários.

Nota	Qualidade percebida	Significado
5	Excelente	Sem degradações perceptíveis.
4	Boa	Leves degradações perceptíveis, porém sem comprometer a experiência.
3	Moderada	Degradações audíveis/visuais, mas sem perda de inteligibilidade
2	Pobre	Presença de interrupções que impactam a compreensão.
1	Ruim	Comunicação inviável.

Tabela 1. Relação entre pontuação MOS e experiência do usuário. [International Telecommunication Union 1996].

4. Sistema VoIPTrack QoS

O VoIPTrack QoS é um sistema desenvolvido para monitorar continuamente a qualidade de chamadas VoIP por meio da coleta de estatísticas diretamente dos agentes usuários. Diferentemente das soluções tradicionais, que dependem da captura passiva do tráfego em pontos estratégicos da rede, a abordagem proposta utiliza as informações disponibilizadas pela API do WebRTC para coletar métricas de qualidade nos próprios clientes. Com isso, o sistema elimina a necessidade de inspeção do tráfego de mídia e possibilita o monitoramento de chamadas em ambientes distribuídos, com usuários móveis e comunicações protegidas por protocolos de criptografia, como o SRTP.

4.1. Arquitetura do Sistema

A arquitetura do VoIPTrack QoS é composta por três componentes principais, conforme ilustrado na Figura 1: uma API REST, responsável por receber e processar os relatórios enviados pelos clientes; um banco de dados PostgreSQL, utilizado para armazenar as métricas coletadas; e o Grafana, empregado para visualização dos dados e geração de alertas.

A API foi desenvolvida na linguagem Rust [Rust Project Developers 2026, Bugden and Alahmar 2022] utilizando o *framework* Actix Web, escolhido devido ao seu elevado desempenho e segurança de memória, características importantes para aplicações de monitoramento em tempo real. O armazenamento das métricas é realizado no PostgreSQL [Juba et al. 2015], enquanto o Grafana [Grafana Labs 2025, Elradi 2025] é utilizado para construção dos *dashboards* e gerenciamento de alertas. Todos os componentes são executados em contêineres Docker [Docker Inc. 2025, Basharirad et al. 2017], simplificando a implantação e a manutenção do sistema.

4.2. Coleta e Processamento das Métricas

Durante uma chamada, o cliente WebRTC coleta periodicamente estatísticas disponibilizadas pelo navegador, incluindo métricas como latência, *jitter*, perda de pacotes e *codec* utilizado. Essas informações são organizadas em objetos JSON e enviadas para a API REST por meio de requisições HTTP do tipo POST.

Após receber os relatórios, a API realiza o processamento das métricas, calcula indicadores derivados, como o MOS estimado, e armazena os resultados no banco de

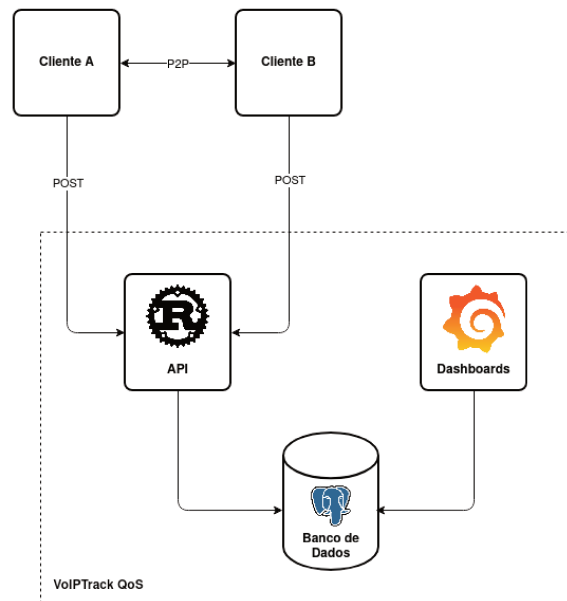


Figura 1. Arquitetura do sistema VoIPTrack QoS. (Autorial, 2026).

dados. Dessa forma, as informações ficam disponíveis para consultas históricas e acompanhamento contínuo da qualidade das chamadas.

4.3. Dashboards e Monitoramento

As métricas armazenadas são disponibilizadas por meio de *dashboards* desenvolvidos no Grafana, permitindo o monitoramento da qualidade das comunicações em tempo real. Os painéis apresentam indicadores como latência, *jitter*, perda de pacotes e MOS, possibilitando tanto a análise individual de chamadas quanto a visualização consolidada do desempenho das chamadas ao longo do tempo.

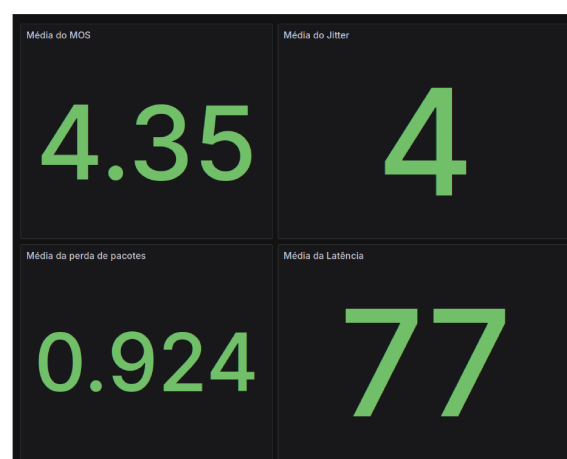


Figura 2. Médias de MOS, de *jitter*, de perda de pacotes e de latência para um determinado cliente. (Autorial, 2026).

A Figura 2 apresenta um dos painéis desenvolvidos, no qual são exibidas as principais métricas de QoS. A organização dessas informações em uma única interface permite correlacionar os diferentes indicadores e identificar, por exemplo, como aumentos

na latência, no *jitter* ou na perda de pacotes refletem diretamente na redução do MOS. Os gráficos são apresentados em função do tempo, possibilitando o acompanhamento da evolução das métricas e a identificação de degradações pontuais ou recorrentes.



Figura 3. Médias de MOS ao longo do tempo para uma determinada empresa. (Autorial, 2026).

Além do painel consolidado, foram desenvolvidos painéis específicos para análise detalhada das métricas. A Figura 3 apresenta a evolução temporal do MOS, permitindo uma visualização mais detalhada da qualidade percebida das chamadas, destacando variações ao longo do período selecionado. Por meio dessa abordagem, torna-se possível identificar momentos de degradação contínua ou intermitente, bem como avaliar o impacto de eventos na infraestrutura sobre a experiência do usuário final. De forma similar, foram desenvolvidos painéis com a evolução da latência, do *jitter* e da perda de pacotes.

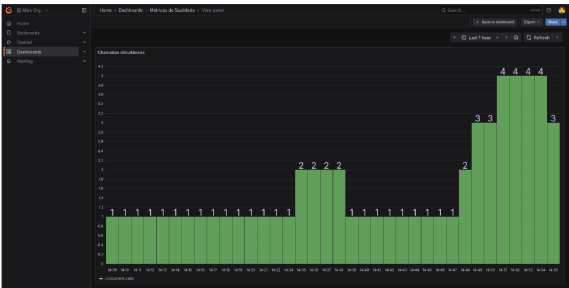


Figura 4. Chamadas simultâneas em uma determinada empresa. (Autorial, 2026).

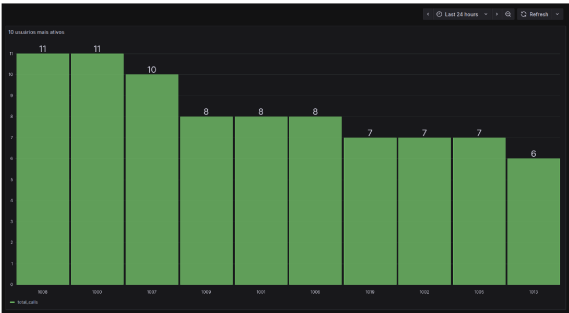


Figura 5. Usuários mais ativos em uma determinada empresa. (Autorial, 2026).

O sistema também disponibiliza painéis voltados ao acompanhamento operacional da infraestrutura. A Figura 4 apresenta a quantidade de chamadas simultâneas ao longo

por obter as informações disponibilizadas pelo WebRTC e transmiti-las ao sistema. Dessa forma, a adoção da solução pode ser limitada em ambientes que utilizam *softphones* proprietários ou aplicações de terceiros que não permitam a extração dessas estatísticas ou sua integração com sistemas externos.

Referências

- Basharirad, B., Bhatti, H., and Ahmadi, M. (2017). An introduction to docker and analysis of its performance. *IJCSNS International Journal of Computer Science and Network Security*, 173:8.
- Baughner, M., McGrew, D., Naslund, M., Carrara, E., and Norrman, K. (2004). The secure real-time transport protocol (srtp). RFC 3711.
- Bugden, W. and Alahmar, A. (2022). Rust: The programming language for safety and performance.
- Docker Inc. (2025). Docker documentation. Acesso em: 15 jun. 2026.
- Elradi, M. D. (2025). Prometheus grafana: A metrics-focused monitoring stack. *Journal of Computer Allied Intelligence(JCAI, ISSN: 2584-2676)*, 3(3):28–39.
- Grafana Labs (2025). Grafana http api reference. Acesso em: 15 jun. 2026.
- Hartpence, B. (2013). *Packet guide to Voice over IP*. Beijing; Cambridge; Farnham; Koln; Sebastopol; Tokyo: O'reilly.
- International Telecommunication Union (1996). Methods for subjective determination of transmission quality. ITU-T Recommendation P.800. Accessed: 2025-06-15.
- International Telecommunication Union (2003). ITU-T Recommendation G.114: One-way transmission time. Recommendation G.114, International Telecommunication Union, Geneva, Switzerland.
- International Telecommunication Union (2015). The e-model: a computational model for use in transmission planning. ITU-T Recommendation G.107. Accessed: 2025-06-15.
- Juba, S., Vannahme, A., and Volkov, A. (2015). *Learning PostgreSQL: Create, develop and manage relational databases in real world applications using PostgreSQL*. Packt Publishing.
- Kurose, J. F. and Ross, K. W. (2013). *Redes de computadores e a internet: uma abordagem top-down*. Pearson, São Paulo, 6 edition.
- Mozilla (2024). Rtcpeerconnection.getstats() – web apis — mdn. Acesso em: 05 jul. 2025.
- Mozilla (2025). Webrtc api. Acesso em: 19 jun. 2025.
- Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and Schooler, E. (2002). SIP: Session initiation protocol. RFC 3261. Accessed: 2025-06-17.
- Rust Project Developers (2026). The rust programming language. Acesso em: 15 jun. 2026.
- Schulzrinne, H., Casner, S., Frederick, R., and Jacobson, V. (2003). RTP: A transport protocol for real-time applications. RFC 3550. Accessed: 2025-06-17.

SIP3 (2025). Sip3 - monitor your voip and rtc traffic real-time. Acesso em: 24 jun. 2025.

Tanenbaum, A. S. and Feamster, N. (2019). *Computer Networks*. Pearson, 6 edition.

VoIPmonitor (2025). Voipmonitor® - voip monitoring software - quality analyzer - wav recorder. Acesso em: 24 jun. 2025.

World Wide Web Consortium (W3C) (2025). Webrtc: Real-time communication in browsers. Acesso em: 24 jun. 2025.