v7.n12.2026

REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

SISTEMA DE AUDITORIA AUTOMATIZADA DE CHAMADAS DE URA (SAACU): DESENVOLVIMENTO DE UM PROTÓTIPO COM TRANSCRIÇÃO AUTOMÁTICA E MODELO DE LINGUAGEM

AUTOMATED IVR CALL AUDITING SYSTEM (SAACU): DEVELOPMENT OF A PROTOTYPE WITH AUTOMATIC SPEECH TRANSCRIPTION AND A LANGUAGE MODEL

SISTEMA AUTOMATIZADO DE AUDITORÍA DE LLAMADAS DE URA (SAACU): DESARROLLO DE UN PROTOTIPO CON TRANSCRIPCIÓN AUTOMÁTICA Y MODELO DE LENGUAJE

Gustavo Anthony de Souza[1], Guilherme Ferreira Moulin[1], Rodrigo Perkovitsch Alvares[1], João Raimundo Barreto Neto[2], Lucas Ferreira dos Santos[2], Nelson Oliveira Aguiar[3]

https://doi.org/10.47820/recima21.v7i12.9674

PUBLICADO: 12/2026

RESUMO

A auditoria de chamadas em centrais de atendimento costuma se basear em amostras, o que deixa sem análise a maior parte das ligações e raramente confronta o status de encerramento registrado pelo atendente com o que ocorreu na conversa. Este artigo relata o desenvolvimento do Sistema de Auditoria Automatizada de Chamadas de URA (SAACU) e responde à pergunta: quais requisitos de uma auditoria automatizada de chamadas um protótipo baseado em transcrição automática e modelo de linguagem consegue atender, e quais não? O protótipo, em Java e React, recebe gravações pelo Amazon S3, transcreve o áudio com o AWS Transcribe, mascara dados pessoais e solicita ao Gemini, em uma resposta estruturada, resumo, sentimento, avaliação da resolução, status declarado pelo atendente e trechos literais de evidência. O *backend* confere se as evidências existem na transcrição e só então compara o status declarado com a resolução avaliada, sinalizando divergências. Dos 15 requisitos funcionais, 10 foram implementados integralmente, 4 parcialmente e 1 ficou sem implementação. No piloto com serviços reais, as 16 chamadas simuladas foram processadas. Em relação à referência humana, o sistema acertou a resolução, a declaração do atendente e a divergência nas 16 chamadas e o sentimento em 13 (F1 macro de 0,82), com erros concentrados na troca de neutro por positivo. A taxa de erro de palavras da transcrição ficou entre 8,8% e 11,1%, e o custo estimado foi de US$ 0,008 por chamada. O piloto usou chamadas roteirizadas, com declaração explícita, avaliadas pelos próprios autores, o que limita a generalização.

PALAVRAS-CHAVE: Auditoria de chamadas. Transcrição automática de fala. Modelos de linguagem. Análise de sentimento. Centrais de atendimento.

ABSTRACT

Call auditing in contact centers usually relies on samples, which leaves most calls unanalyzed and rarely checks the closing status recorded by the agent against what happened in the conversation.

[1] Graduando em Engenharia da Computação pela Universidade São Judas Tadeu (USJT), Campus Mooca, em fase de conclusão do bacharelado.

[2] Graduando em Engenharia de Software pela Universidade São Judas Tadeu (USJT), Campus Mooca, em fase de conclusão do bacharelado.

[3] Doutor em Ciências, na área de Medicina, Tecnologia e Intervenção em Cardiologia, pela Universidade de São Paulo (USP), no Instituto Dante Pazzanese de Cardiologia. Mestre em Engenharia Elétrica, com ênfase em Inteligência Artificial Aplicada à Automação, pelo Centro Universitário FEI. Bacharel em Engenharia de Computação pela Universidade São Judas Tadeu (USJT). Docente e orientador de TCC, com experiência em Redes de Computadores, Análise de Dados e Inteligência Artificial.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

This article reports the development of the Automated IVR Call Auditing System (SAACU) and addresses the question: which requirements of automated call auditing can a prototype based on automatic speech transcription and a language model meet, and which can it not? The prototype, built in Java and React, receives recordings through Amazon S3, transcribes them with AWS Transcribe, masks personal data and asks Gemini, in one structured response, for a summary, sentiment, a resolution assessment, the agent-declared status and literal evidence excerpts. The backend checks that the evidence exists in the transcript and only then compares the declared status with the assessed resolution, flagging divergences. Of the 15 functional requirements, 10 were fully implemented, 4 were partially implemented, and 1 remained unimplemented. In the pilot with live services, all 16 simulated calls were processed. Against the human reference, the system matched resolution, agent-declared status and divergence in all 16 calls and sentiment in 13 (macro F1 of 0.82), with errors concentrated in labeling neutral customers as positive. Word error rate ranged from 8.8% to 11.1%, and the estimated cost was US$ 0.008 per call. The pilot used scripted calls with explicit declarations, rated by the authors themselves, which limits generalization.

KEYWORDS: *Call auditing. Automatic speech recognition. Language models. Sentiment analysis. Contact centers.*

RESUMEN
La auditoría de llamadas en centros de atención suele basarse en muestras y rara vez contrasta el estado de cierre registrado por el agente con la conversación. Este artículo relata el desarrollo del Sistema Automatizado de Auditoría de Llamadas de URA (SAACU) y responde a la pregunta: ¿qué requisitos de una auditoría automatizada de llamadas puede atender un prototipo basado en transcripción automática y modelo de lenguaje, y cuáles no? El prototipo, en Java y React, recibe grabaciones a través de Amazon S3, transcribe el audio con AWS Transcribe, enmascara datos personales y solicita a Gemini, en una respuesta estructurada, resumen, sentimiento, evaluación de la resolución, estado declarado por el agente y fragmentos literales de evidencia. El backend comprueba que las evidencias existan en la transcripción y solo entonces compara el estado declarado con la resolución evaluada, señalando divergencias. De los 15 requisitos funcionales, 10 se implementaron por completo, 4 parcialmente y 1 quedó sin implementar. En el piloto con servicios reales, se procesaron las 16 llamadas simuladas. Frente a la referencia humana, el sistema acertó la resolución, el estado declarado y la divergencia en las 16 llamadas y el sentimiento en 13 (F1 macro de 0,82), con errores concentrados en clasificar como positivo al cliente neutro. La tasa de error de palabras osciló entre 8,8% y 11,1%, y el costo estimado fue de US$ 0,008 por llamada. El piloto usó llamadas guionizadas con declaración explícita, evaluadas por los propios autores, lo que limita la generalización.

PALABRAS CLAVE: *Auditoría de llamadas. Transcripción automática del habla. Modelos de lenguaje. Análisis de sentimiento. Centros de atención.*

1. INTRODUÇÃO

As centrais de atendimento que operam com Unidades de Resposta Audível (URAs) recebem diariamente um volume elevado de ligações e funcionam, para muitas empresas, como o principal canal de contato com o cliente. Garantir a qualidade dessas interações e a resolução efetiva das demandas é um problema recorrente da gestão operacional.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

A inspeção por amostragem é prática consolidada de controle de qualidade em operações (SLACK; CHAMBERS; JOHNSTON, 2009). Aplicada a ligações, ela significa que monitores humanos ouvem apenas uma fração pequena das chamadas, o que motivou, já em meados dos anos 2000, sistemas que combinam reconhecimento de fala e processamento de linguagem natural para priorizar a escuta humana (ZWEIG *et al.*, 2006). Há, porém, uma falha que a amostragem não alcança: o status com que o atendente encerra o atendimento (por exemplo, "resolvido") raramente é confrontado com o conteúdo da conversa. Quando o cliente desliga sem solução, o registro operacional pode indicar o contrário.

Os serviços de transcrição automática (*Speech-to-Text*) e os modelos de linguagem de grande escala (*Large Language Models*, LLMs), baseados na arquitetura Transformer (VASWANI *et al.*, 2017), tornam possível converter o áudio em texto e extrair dele informações estruturadas em larga escala. Ao mesmo tempo, esses modelos podem gerar conteúdo não sustentado pela entrada (JI *et al.*, 2023) e apresentam vieses quando usados como avaliadores (ZHENG *et al.*, 2023; WANG *et al.*, 2024), o que exige cuidado ao usá-los em uma tarefa de auditoria.

Este artigo relata o desenvolvimento do Sistema de Auditoria Automatizada de Chamadas de URA (SAACU), um protótipo que transcreve gravações, mascara padrões de dados pessoais, extrai por meio do modelo Gemini o resumo, o sentimento, a avaliação da resolução e o status declarado pelo atendente, e sinaliza divergências entre o status declarado e a resolução avaliada. A comparação é feita pelo *backend*, e não pelo modelo, e só ocorre quando o modelo devolve trechos literais da transcrição que sustentam a declaração e a avaliação.

1.1. Problema e pergunta de pesquisa

O problema tratado é a impossibilidade prática de auditar manualmente todas as chamadas e, em particular, de verificar se o status declarado pelo atendente corresponde ao que aconteceu na conversa. A pergunta principal é quais requisitos de uma auditoria automatizada de chamadas de URA um protótipo baseado em transcrição automática e modelo de linguagem consegue atender. A avaliação complementar investiga o desempenho da transcrição e das classificações em chamadas simuladas. Para responder a ela, executou-se um piloto com 16 chamadas simuladas, avaliado em relação a uma referência humana.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

1.2. Objetivos

O objetivo geral é desenvolver e verificar um protótipo de *software* que receba gravações de chamadas, transcreva-as, proteja padrões de dados pessoais na transcrição, extraia por modelo de linguagem o resumo, o sentimento, a resolução e o status declarado pelo atendente, e sinalize divergências entre o status declarado e a resolução avaliada. Os objetivos específicos são:

a) especificar os requisitos funcionais e não funcionais e modelar o sistema por diagramas UML de casos de uso, de sequência e de classes;

b) implementar o fluxo de recebimento, transcrição, mascaramento, análise e persistência das chamadas;

c) implementar a comparação entre o status declarado pelo atendente e a resolução avaliada, condicionada à existência de evidências literais na transcrição;

d) disponibilizar as chamadas auditadas em interface *web*, com consulta individual, filtragem, exportação e indicadores;

e) verificar a situação de cada requisito funcional na versão avaliada, indicando a evidência correspondente;

f) executar um piloto com chamadas simuladas e avaliar as transcrições e as classificações do sistema em relação a uma referência humana.

1.3. Justificativa

Um sistema que processa todas as gravações recebidas pode ampliar a cobertura da auditoria sem multiplicar o tempo de escuta humana, e a comparação entre o status declarado e a conversa ataca justamente a falha que a amostragem não detecta. Sob a ótica da Engenharia de Computação, a integração de serviços gerenciados de transcrição e de um modelo de linguagem levanta três questões que o trabalho documenta: como validar a saída de um modelo probabilístico antes de usá-la em uma regra de negócio, como tratar dados pessoais conforme a Lei Geral de Proteção de Dados Pessoais (BRASIL, 2018) e como tornar o resultado reproduzível. As contribuições são uma arquitetura em que o modelo extrai e o código decide, um quadro de requisitos verificados na versão avaliada e um protocolo de avaliação replicável.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

2. REFERENCIAL TEÓRICO

2.1. Fundamentos técnicos

O SAACU adota um monólito organizado em camadas. A separação entre apresentação, aplicação, domínio, infraestrutura e persistência favorece a coesão e o baixo acoplamento (BUSCHMANN *et al.*, 1996; EVANS, 2003), e o estilo em camadas é adequado a sistemas pequenos, com equipe única, em que a simplicidade de implantação pesa mais que a escalabilidade independente de componentes (RICHARDS; FORD, 2020). A camada de aplicação concentra a orquestração em serviços, no padrão *Service Layer* (FOWLER, 2002), e as integrações externas ficam isoladas por *adapters* (GAMMA *et al.*, 1994). A linguagem Java foi escolhida pela tipagem estática e pela maturidade do ecossistema (GOSLING *et al.*, 2014).

O Reconhecimento Automático de Fala (ASR) converte áudio em texto por meio de modelos estatísticos (JUANG; RABINER, 2006), e as redes neurais profundas reduziram substancialmente seus erros (HINTON *et al.*, 2012). O uso do AWS Transcribe se insere no modelo de computação em nuvem, que oferece serviços gerenciados e elásticos (ARMBRUST *et al.*, 2010). A análise textual se apoia no Processamento de Linguagem Natural (JURAFSKY; MARTIN, 2021) e nos LLMs. Componentes baseados em aprendizado de máquina introduzem dependências externas pouco visíveis, como mudanças de comportamento do serviço e de dados de entrada (SCULLEY *et al.*, 2015); isolar o fornecedor atrás de um *adapter* reduz o custo de substituí-lo e preserva a capacidade de a arquitetura evoluir de forma incremental (FORD; PARSONS; KUA, 2017). Quanto aos dados pessoais, a LGPD (BRASIL, 2018) e o princípio de *Privacy by Design* (CAVOUKIAN, 2011) orientam a proteção desde a concepção do sistema, e não apenas na persistência.

2.2. Auditoria automatizada de chamadas: trabalhos relacionados

A busca foi feita em novembro de 2026 no Google Acadêmico, na ACL Anthology, no arXiv e na biblioteca digital da Sociedade Brasileira de Computação (SBC OpenLib), com combinações dos termos *call center quality monitoring*, *speech analytics*, *contact center* AND *large language models*, *LLM-as-a-judge*, *sentiment analysis* AND *telephone conversations* e "reconhecimento de fala" AND português, para o período de 2006 a 2025. Foram priorizados trabalhos com avaliação quantitativa e, para o português, estudos com fala espontânea.

Zweig *et al.* (2006) descrevem um sistema que transcreve chamadas de suporte técnico e combina regras sobre a transcrição com um modelo estatístico para ordenar as chamadas e direcionar a escuta humana às mais críticas. Fu *et al.* (2022) anotaram 23 mil enunciados de

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

chamadas telefônicas de uma central de atendimento para classificar o sentimento por entidade e relatam que as transcrições automáticas trazem erros de reconhecimento e hesitações que dificultam a tarefa; o melhor modelo alcançou F1 de cerca de 74,7. Esses trabalhos tratam do inglês e não comparam o registro do atendente com a conversa.

Com os LLMs, a avaliação de qualidade passou a ser delegada ao próprio modelo. Ingle *et al.* (2024) avaliaram oito modelos no julgamento de conversas entre atendentes e clientes e observaram que planos de avaliação gerados por modelos maiores elevam o F1 macro dos menores em até 18,95%. Shin *et al.* (2025) mostraram que afirmações geradas por LLMs sobre conversas de atendimento podem não ser sustentadas pelo texto e construíram um conjunto de 410 afirmações rotuladas, com concordância entre anotadores de 0,82 e melhor F1 de 0,86. Em avaliações abertas, LLMs concordam com preferências humanas em mais de 80% dos casos, mas são sensíveis à posição das respostas, à extensão do texto e às próprias saídas (ZHENG *et al.*, 2023; WANG *et al.*, 2024). Esses limites, somados à geração de conteúdo não sustentado (JI *et al.*, 2023), justificam que a decisão final de uma auditoria não fique a cargo do modelo.

Para o português do Brasil, o corpus CORAA reúne fala espontânea e preparada validada manualmente para avaliação de ASR (CANDIDO JUNIOR *et al.*, 2023). Em fala espontânea com o sotaque paulistano, o menor WER (*word error rate*) obtido por Lima *et al.* (2024) foi de 24,22%, e as falhas concentraram-se em nomes próprios e vocabulário de domínio. Como o áudio telefônico tem banda estreita e ruído, erros dessa ordem podem se propagar para a análise, o que torna a medição do WER parte necessária da avaliação. O Quadro 1 resume os trabalhos e indica o que o SAACU acrescenta: a aplicação ao português, a comparação entre o status declarado e a resolução avaliada e a verificação, fora do modelo, das evidências que sustentam essa comparação.

Quadro 1. Trabalhos relacionados e contribuição do SAACU

Trabalho	Dados e idioma	Abordagem	Avaliação relatada
Zweig *et al.* (2006)	Chamadas de suporte técnico; inglês	ASR, regras e modelo estatístico para ordenar chamadas para escuta humana	Demonstração de sistema
Fu *et al.* (2022)	23 mil enunciados de chamadas telefônicas; inglês	Sentimento por entidade sobre transcrições automáticas	F1 ≈ 74,7

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Trabalho	Dados e idioma	Abordagem	Avaliação relatada
Ingle *et al.* (2024)	Conversas de central de atendimento (base interna); inglês	Oito LLMs como avaliadores de qualidade, com planos de avaliação	Ganho de até 18,95% no F1 macro
Shin *et al.* (2025)	410 afirmações sobre conversas sintéticas; inglês	Factualidade de afirmações geradas por LLM	Melhor F1 = 0,86; concordância entre anotadores de 0,82
Lima *et al.* (2024)	239 h de fala espontânea; português	ASR com Wav2Vec2 e Distil-Whisper	Menor WER = 24,22%
SAACU (este trabalho)	Chamadas simuladas; português	AWS Transcribe e Gemini; comparação entre status declarado e resolução avaliada, com evidência verificada no *backend*	Quadro 2 e Tabela 1

Fonte: elaborado pelos autores (2026).

3. METODOLOGIA

Trata-se de pesquisa aplicada, de natureza tecnológica. O protótipo foi verificado em duas frentes. Na primeira, cada requisito funcional foi classificado como implementado quando o código realizava o comportamento especificado e havia teste automatizado ou verificação reproduzível, como parcial quando apenas parte do comportamento existia, ou como não implementado quando faltava código correspondente. Na segunda, executou-se o piloto com chamadas simuladas descrito na Seção 3.5.

3.1. Desenvolvimento e versão avaliada

A especificação definiu 15 requisitos funcionais (RF01 a RF15) e 12 não funcionais, desdobrados em histórias de usuário e organizados em backlog (COHN, 2004). O backend foi desenvolvido em Java 21 com Spring Boot 4.1, e o frontend, em React com Vite. O código é distribuído sob a licença MIT, e o endereço do repositório foi omitido para avaliação cega. A versão do backend e do frontend usada no piloto está registrada no commit 2439fc89e1a4b2e00968dc1938165e24d141cd5a.

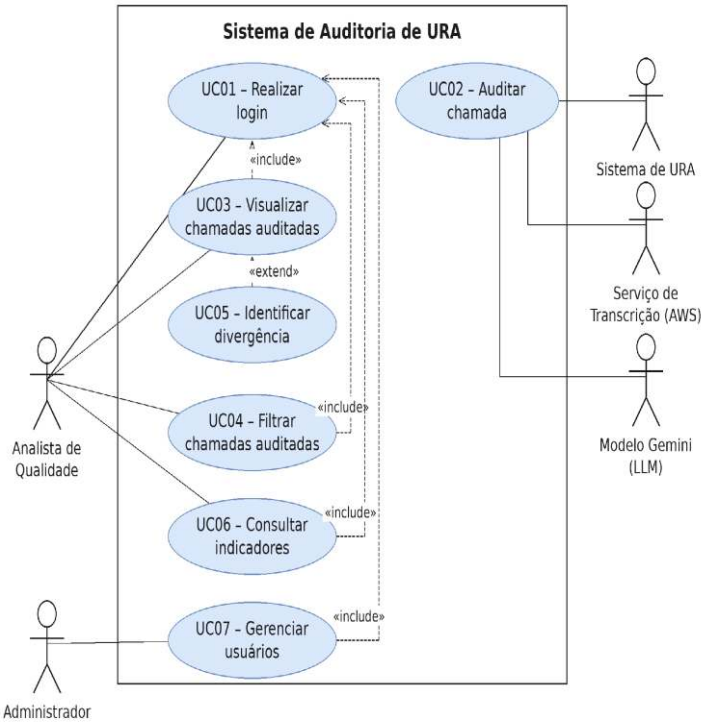
3.2. Modelagem e fluxo de processamento

Os diagramas apresentados neste artigo (Figuras 1 a 4) representam o escopo projetado na especificação; o Quadro 2 (Seção 4.1) indica o que desse escopo está implementado. A Figura 1 apresenta os casos de uso. O Analista de Qualidade consulta, filtra e acompanha as

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

chamadas; o Administrador gerencia usuários; o caso de uso central, UC02 – Auditar chamada, é executado automaticamente quando uma gravação chega, com a participação do Sistema de URA, do serviço de transcrição e do modelo Gemini. Na versão avaliada, o papel do Sistema de URA é cumprido pelo depósito das gravações em um prefixo do Amazon S3, pois não há integração direta com uma URA, e o UC07 (Gerenciar usuários) não foi implementado.

Figura 1. Diagrama de casos de uso do SAACU

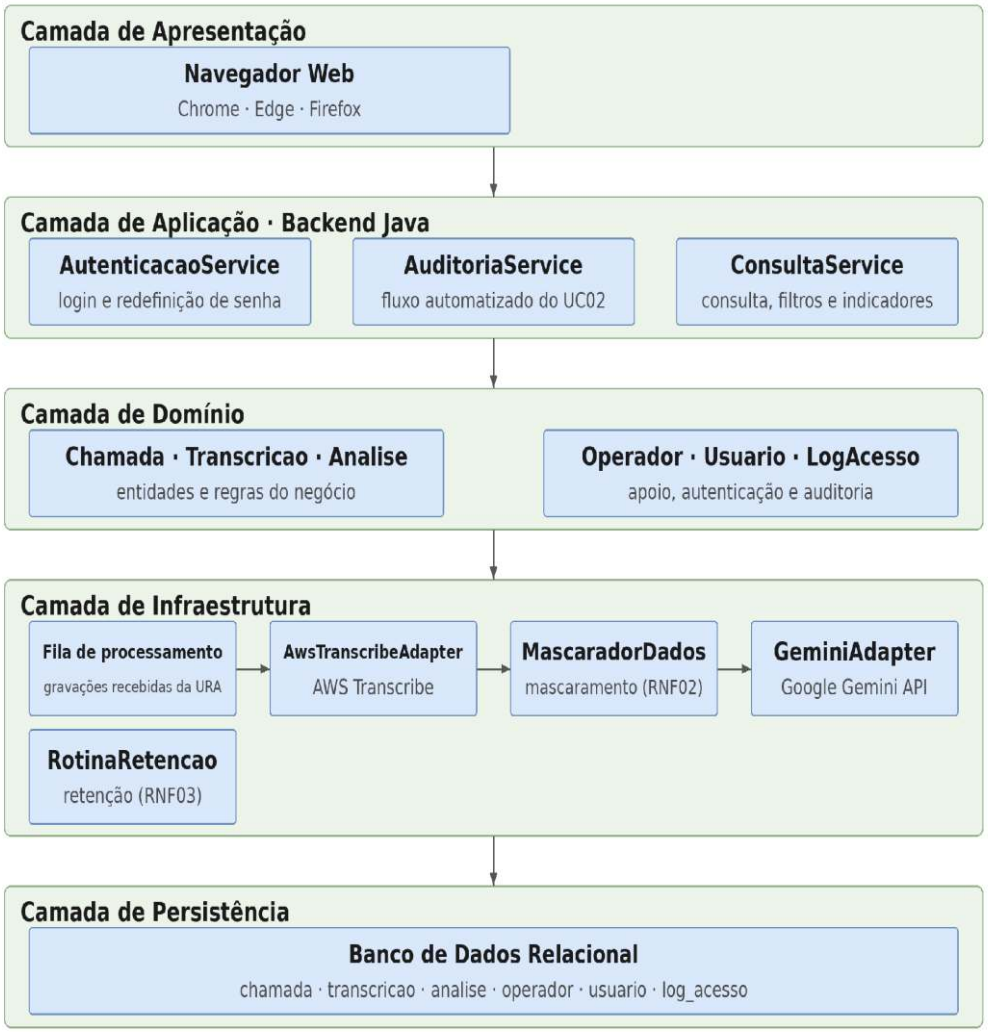


Fonte: elaborado pelos autores (2026).

A Figura 2 mostra a arquitetura lógica em camadas. A interface React, executada no navegador, consome a API REST do *backend*; os serviços de aplicação orquestram o fluxo; as entidades de domínio representam chamada, transcrição e análise; os *adapters* de infraestrutura encapsulam o AWS Transcribe, o mascaramento e a API do Gemini. Duas diferenças em relação ao projeto merecem registro: a "fila de processamento" foi implementada como registro das entradas no banco de dados, alimentado por verificação periódica do S3 com até duas transcrições simultâneas, e a rotina de retenção (RNF03) não foi implementada. A persistência usa o banco H2 em arquivo criptografado com AES, com MySQL como alternativa configurável.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Figura 2. Arquitetura lógica em camadas do SAACU

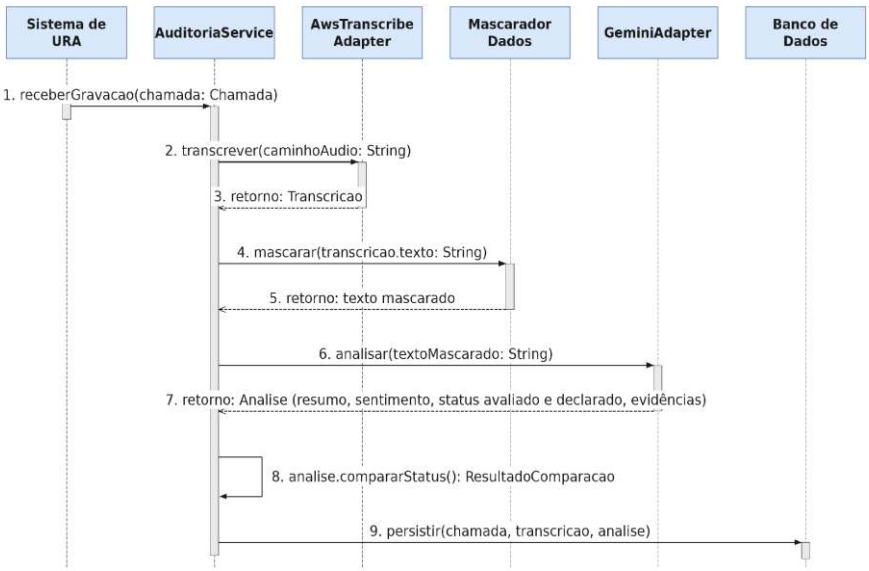


Fonte: elaborado pelos autores (2026).

A Figura 3 detalha a sequência de processamento. A gravação pode chegar por *upload multipart* em *endpoint* autenticado ou pela verificação periódica do prefixo de entrada no S3, configurada para 30 segundos no piloto. O sistema transcreve o áudio, mascara o texto, envia a transcrição protegida ao Gemini, valida a resposta, compara o status declarado com a resolução avaliada e persiste o resultado. Em caso de falha recuperável na análise, o sistema preserva a transcrição e programa nova tentativa sem refazer a transcrição. Erros classificados como permanentes recebem o estado FALHA_ANALISE_IA.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Figura 3. Diagrama de sequência do processamento de uma chamada



Fonte: elaborado pelos autores (2026).

3.3. Configuração dos serviços externos

A transcrição usou *jobs* assíncronos do AWS Transcribe em português do Brasil (pt-BR), na região us-east-2. Os arquivos deste piloto estavam no formato MP3. Como a configuração não inclui diarização, o modelo infere a atribuição das falas a partir do conteúdo.

A análise usou o método generateContent da API do Gemini com o identificador de modelo gemini-3.6-flash. O código não define a temperatura nem registra a revisão interna do modelo retornada pelo serviço. O esquema JSON restringe as classes de sentimento, resolução e status declarado. O *prompt* consta no Apêndice A e solicita que a resolução seja avaliada independentemente da declaração do atendente, com trechos de evidência copiados da transcrição. Antes da persistência, a resposta passa por verificações de completude e consistência (BATINI *et al.*, 2009).

Antes da persistência e do envio ao Gemini, a transcrição passa por mascaramento baseado em expressões regulares para CPF, telefone, endereço de *e-mail*, número de cartão e dados precedidos de rótulo. O atendente é identificado quando se apresenta explicitamente e o nome completo corresponde a um atendente cadastrado. O acesso exige autenticação por sessão, com proteção contra CSRF, senhas armazenadas com bcrypt, bloqueio da conta após cinco falhas consecutivas por 15 minutos e redefinição de senha por *token* de uso único válido por 30 minutos.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

3.4. Verificação dos requisitos

Cada requisito funcional foi associado às classes que o implementam e aos testes que exercitam seu comportamento. A suíte de *backend* usa serviços simulados nos testes que dependem de AWS Transcribe, Gemini e envio de e-mail. O resultado da execução automatizada está na Seção 4.3.

3.5. Protocolo de avaliação com chamadas simuladas

O conjunto ampliado contém 21 roteiros de chamadas fictícias com classificações esperadas. O piloto de 13 de novembro de 2026 usou as 16 gravações correspondentes aos roteiros 01 a 16, em MP3. O gabarito previa seis chamadas resolvidas, seis parcialmente resolvidas e quatro não resolvidas. As 16 incluíam declaração explícita do atendente, com seis divergências previstas entre a declaração e a resolução. Os cenários 17 a 21 ficaram fora deste lote. As gravações foram realizadas por um dos autores, que autorizou o uso das gravações no trabalho.

As gravações foram depositadas em dois lotes no prefixo de entrada do Amazon S3. O registro de execução contém o identificador de cada chamada, o horário de registro do objeto no S3 (LastModified), o horário de recebimento no *backend*, a conclusão, o estado final, a transcrição mascarada e os campos da análise. Para formar a referência humana, dois autores transcreveram os 16 áudios de forma independente, e dois avaliadores classificaram as chamadas. O primeiro avaliador atribuiu a cada chamada o sentimento do cliente (POSITIVO, NEUTRO ou NEGATIVO), a resolução e o status declarado pelo atendente. O segundo, autor de uma das transcrições, registrou por escrito se o cliente ficou satisfeito ou insatisfeito e se o status declarado correspondia ao ocorrido, com justificativa, da qual foram extraídas a resolução e a declaração. Como a escala de sentimento do segundo avaliador é binária, a concordância entre os avaliadores nessa dimensão foi medida apenas na distinção entre negativo e não negativo. A concordância foi calculada pelo kappa de Cohen (COHEN, 1960; LANDIS; KOCH, 1977). Os avaliadores são autores do trabalho e conheciam os roteiros.

Com a referência humana, calcularam-se a acurácia e o F1 macro do sistema para sentimento, resolução e status declarado, além da acurácia do resultado da comparação. A taxa de erro de palavras (WER) comparou a transcrição registrada pelo sistema, na qual o mascaramento não alterou nenhum trecho, com cada transcrição humana, após normalização de caixa, pontuação, acentuação, numerais e hesitações (JURAFSKY; MARTIN, 2021). A WER entre as duas transcrições humanas foi calculada para estimar a variação da própria referência. O tempo foi medido do registro do objeto no S3 e do recebimento pelo *backend* até a conclusão.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

O custo por chamada foi estimado pela duração dos áudios e pela tarifa de transcrição em lote da AWS (AMAZON WEB SERVICES, 2026) e pelos *tokens* de entrada e de saída do Gemini, incluídos os de raciocínio, às tarifas do plano pago padrão (GOOGLE, 2026).

4. RESULTADOS E DISCUSSÃO

4.1. Atendimento aos requisitos

O Quadro 2 apresenta a situação de cada requisito funcional na versão avaliada e responde à pergunta principal. Dos 15 requisitos, 10 estão implementados, 4 parcialmente e 1 não foi implementado. Entre os sete requisitos classificados como essenciais na especificação, apenas o RF02 é parcial.

Quadro 2. Situação dos requisitos funcionais na versão avaliada

RF	Requisito (resumo)	Situação	Evidência
RF01	Login com credenciais cadastradas	Implementado	SecurityConfig; AuthenticationIntegrationTest
RF02	Recebimento automático das gravações da URA	Parcial	S3AudioIngestionService (verificação periódica do S3); S3AudioIngestionTest. Sem integração direta com URA
RF03	Transcrição pelo serviço da AWS	Implementado	AwsTranscribeService; AwsStoredAudioTest
RF04	*Prompt* estruturado único ao Gemini	Implementado	GeminiClient; GeminiClientTest; GeminiRetryTest
RF05	Persistência dos resultados	Implementado	CallAuditStore (H2 com AES ou MySQL); CallAuditPersistenceTest
RF06	Visualização em tabela	Implementado	GET /api/v1/calls paginado; TabelaChamadas.jsx
RF07	Filtros por data, sentimento, resolução e operador	Parcial	Apenas filtro por operador (API e BarraFiltros.jsx)
RF08	Sinalização de divergência entre status declarado e avaliado	Implementado	ResultadoComparacao; GeminiAnalysisResult; AudioDeclarationComparisonTest
RF09	Exibição do resumo	Implementado	PainelDetalhe.jsx
RF10	Consulta individual completa	Implementado	GET /api/v1/calls/{id}; PainelDetalhe.jsx
RF11	Exportação dos resultados	Parcial	exportarCsv.js; exporta só a página exibida
RF12	Painel com indicadores	Implementado	PaginaDashboard.jsx; cálculo no navegador

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

RF	Requisito (resumo)	Situação	Evidência
RF13	Gestão de usuários pelo Administrador	Não implementado	Só há o usuário inicial criado por configuração
RF14	Redefinição de senha por *e-mail*	Implementado	PasswordResetService; PasswordResetServiceTest
RF15	Notificação de gravação inválida	Parcial	Estado GRAVACAO_INVALIDA registrado e exibido; sem notificação ativa

Fonte: elaborado pelos autores a partir do commit 2439fc8 (2026).

As limitações parciais têm causas distintas. No RF02, a gravação chega por um repositório intermediário, e não diretamente da URA; a integração real depende de um fornecedor de URA. No RF07, a API filtra apenas por operador, e a interface não aplica os demais filtros no navegador porque eles atuariam só sobre a página carregada e dariam um resultado incorreto. No RF11, o arquivo CSV reproduz apenas a página exibida. No RF15, a gravação inválida é registrada e aparece na tabela como "Gravação inválida", mas ninguém é notificado ativamente. O RF12 está implementado, porém os indicadores são calculados no navegador a partir de todas as páginas da listagem, solução adequada ao volume do protótipo, mas que exigirá um *endpoint* de agregação com volumes maiores. Indicadores agregados como esses são o recurso típico das ferramentas de inteligência de negócios para acompanhamento gerencial (CHAUDHURI; DAYAL; NARASAYYA, 2011), e o painel segue a recomendação de apresentar primeiro a visão geral e depois o detalhe (SHNEIDERMAN, 1996; FEW, 2006).

Em relação ao projeto, a principal mudança está na distribuição das responsabilidades de consulta. O serviço de consulta projetado reuniria no *backend* a listagem com filtro completo, a exportação e os indicadores; na versão avaliada, a listagem paginada por operador está na API, enquanto exportação e indicadores foram implementados no *frontend*. A RotinaRetencao, que eliminaria áudios e transcrições após o prazo definido (RNF03), não existe no código, e a gestão de perfis de usuário (RF13) se limita ao usuário inicial.

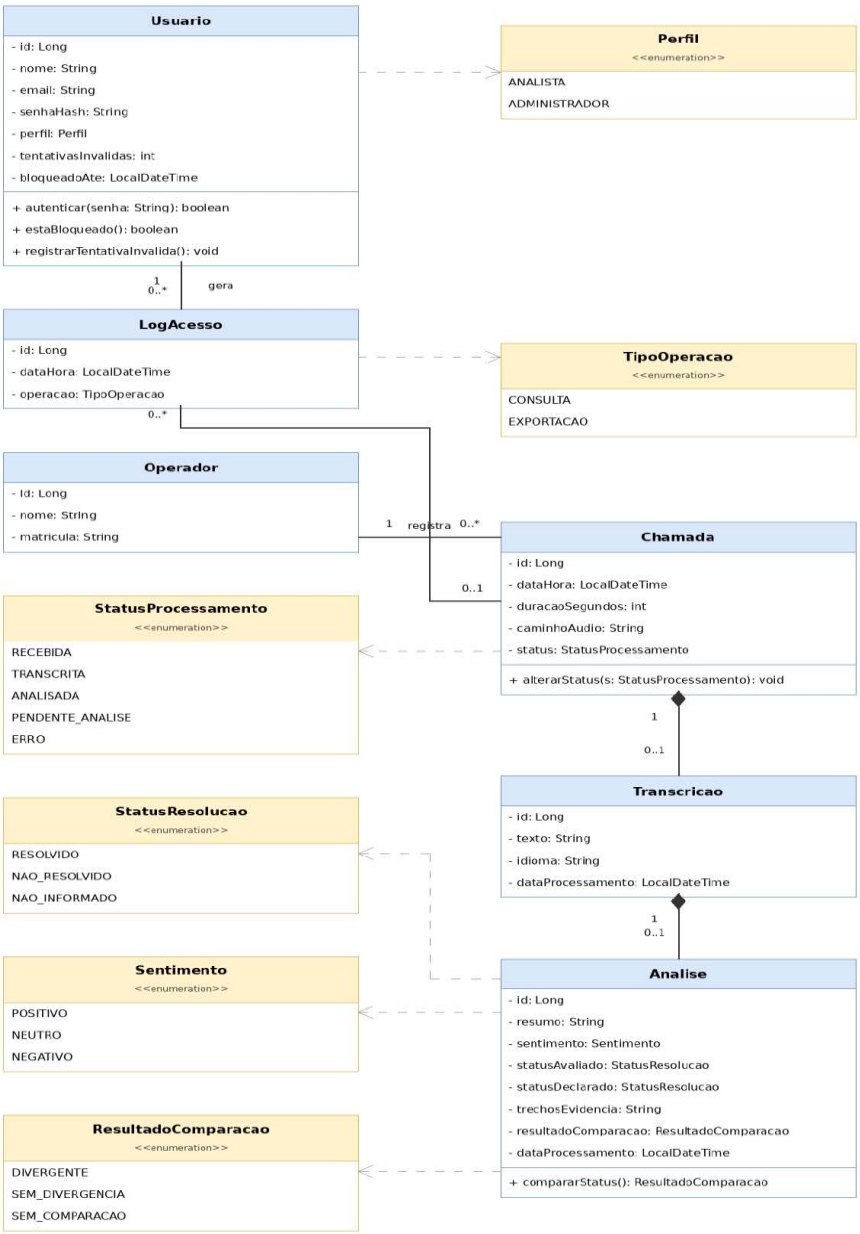
4.2. Comparação entre o status declarado e a resolução avaliada

A Figura 4 apresenta o modelo de domínio. A classe Analise reúne a resolução avaliada, o status declarado pelo atendente, os trechos de evidência e o resultado da comparação, que assume três valores: DIVERGENTE, SEM_DIVERGENCIA e SEM_COMPARACAO. A implementação segue esse modelo, com três ajustes: há dois trechos de evidência, um para a declaração e outro para a resolução; a resolução avaliada admite PARCIALMENTE_RESOLVIDO, e o valor NAO_INFORMADO ficou restrito ao status declarado; e os estados de processamento foram detalhados, incluindo GRAVACAO_INVALIDA e

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

FALHA_TRANSCRICAO. Em todos os casos, o Gemini extrai a declaração e as evidências, e o *backend* decide a divergência.

Figura 4. Diagrama de classes: modelo de domínio



Fonte: elaborado pelos autores (2026).

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

As regras implementadas são as seguintes. A declaração precisa ser explícita e atribuível ao atendente; perguntas, despedidas e promessas de solução futura não contam. A resolução é avaliada pelo conteúdo da conversa, independentemente da declaração. Cada evidência precisa aparecer literalmente na transcrição mascarada, admitidas apenas diferenças de espaços, quebras de linha e normalização Unicode; resposta incoerente ou com trecho inexistente é rejeitada, e a chamada volta ao estado PENDENTE_DE_ANALISE. Sem declaração explícita ou sem evidência de resolução, o resultado é SEM_COMPARACAO, que se distingue de SEM_DIVERGENCIA. Com declaração e as duas evidências, status iguais produzem SEM_DIVERGENCIA e status diferentes, DIVERGENTE. A comparação é, portanto, uma regra de negócio explícita, declarada e testável, mantida fora do componente probabilístico (ROSS, 2003).

Essa divisão de responsabilidades responde aos limites discutidos na Seção 2.2. Exigir trechos literais e conferi-los no código reduz o risco de o modelo justificar sua classificação com conteúdo inexistente (JI *et al.*, 2023), e a decisão determinística evita que a divergência dependa dos vieses do modelo como avaliador (ZHENG *et al.*, 2023; WANG *et al.*, 2024). A verificação, porém, comprova apenas que o trecho existe, e não que foi interpretado corretamente nem que foi dito pelo atendente; sem diarização, a atribuição das falas continua sendo uma inferência do modelo. Por isso, a qualidade da comparação só pode ser estabelecida pela avaliação com referência humana.

4.3. Verificação automatizada

Em 10 de novembro de 2026, no mesmo *commit* 2439fc8 usado no piloto, o Maven contabilizou 74 casos na suíte de backend, distribuídos em 13 classes, com Java 21.0.11. Houve 73 aprovações, zero falhas, zero erros e um teste de integração com MySQL ignorado. A suíte verifica autenticação, ingestão pelo S3, processamento de áudio, persistência, identificação do atendente, comparação e tratamento de falhas. Os resultados da execução com AWS Transcribe e Gemini estão na Seção 4.4.

4.4. Avaliação com chamadas simuladas

As 16 chamadas foram concluídas. A Tabela 1 compara as saídas do SAACU com a referência humana. Em resolução, status declarado e resultado da comparação, o sistema coincidiu com a referência nas 16 chamadas, e os dois avaliadores concordaram entre si em todas elas ($\kappa = 1{,}0$). O *backend* marcou seis chamadas como divergentes e dez sem divergência.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Tabela 1. Resultados do piloto com 16 chamadas simuladas

Dimensão	Medida	Resultado	Referência ou unidade
Execução	Chamadas concluídas	16 de 16	Lote do piloto
Sentimento	Acurácia; F1 macro	13 de 16 (81,3%); 0,82	Avaliador humano (três classes)
Resolução	Acurácia; F1 macro	16 de 16 (100%); 1,00	Dois avaliadores (κ = 1,0)
Status declarado	Acurácia; F1 macro	16 de 16 (100%); 1,00	Dois avaliadores (κ = 1,0)
Comparação	Acurácia	16 de 16 (100%)	Referência humana
Transcrição	WER	8,8% e 11,1%	Duas transcrições humanas (7,6% entre elas)
Tempo	Mediana do S3 à conclusão	81,9 s	Duração média dos áudios: 38,1 s
Custo	Estimativa por chamada	US$ 0,008	AWS US$ 0,004 + Gemini US$ 0,004

Fonte: registro de execução, avaliação humana e transcrições humanas (2026).

Os áudios têm duração média de 38,1 s, de 30,5 a 42,1 s. Do registro do objeto no S3 à conclusão, a mediana foi de 81,9 s, cerca de duas vezes a duração da ligação; do recebimento pelo *backend* à conclusão, foi de 78,6 s. O maior tempo, 557,9 s, ocorreu no áudio 08, cuja análise foi retomada automaticamente após falhas do Gemini, com a transcrição já persistida.

No sentimento, as três discordâncias seguiram o mesmo padrão: nas chamadas 01, 04 e 11, o sistema atribuiu POSITIVO a clientes que o avaliador classificou como NEUTRO. Nas três, o cliente agradece ou elogia o atendimento ao final, o que sugere que o modelo dá peso excessivo ao encerramento cordial em relação ao tom predominante da ligação. Na classe NEGATIVO, o sistema acertou as cinco chamadas. O gabarito dos roteiros coincidiu com o avaliador em apenas 11 das 16 chamadas nessa dimensão; nas chamadas 02 e 06, em que o gabarito previa NEUTRO, o avaliador concordou com a classificação POSITIVO do sistema. Por isso, adotou-se a referência humana, e não o gabarito, para o sentimento. Na distinção entre negativo e não negativo, os dois avaliadores concordaram nas 16 chamadas (κ = 1,0), mas a separação entre positivo e neutro contou com um único avaliador.

A WER da transcrição automática foi de 8,8% em relação à primeira transcrição humana e de 11,1% em relação à segunda, com variação por chamada de 2,9% a 15,6% na primeira comparação. As duas transcrições humanas divergiram entre si em 7,6%, valor da mesma ordem, o que indica que parte das diferenças vem da própria referência, como omissão de artigos, grafias e trechos não literais, e não do serviço de transcrição. Os valores ficam abaixo dos 24,22% relatados por Lima *et al.* (2024) para fala espontânea, o que é esperado para fala

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

lida de roteiro. Alguns erros, contudo, alteram o sentido. Na chamada 03, "fiquei muito insatisfeito" foi transcrito como "fiquei muito satisfeito". Na chamada 05, "sumiu" virou "assumiu", e o trecho com o erro foi devolvido pelo Gemini como evidência e aceito pela verificação do *backend*, que confirma a presença do trecho na transcrição, e não no áudio.

Os 16 áudios somam 10,2 min. Pela tarifa de transcrição em lote publicada pela AWS para a região US East (N. Virginia), de US$ 0,006 por minuto (AMAZON WEB SERVICES, 2026), o lote custaria cerca de US$ 0,061; como o uso ficou dentro do nível gratuito mensal do serviço, o painel da AWS registrou US$ 0,00. No Gemini, o lote consumiu 12.690 *tokens* de entrada e 15.380 de saída, o que corresponde a US$ 0,067 pelas tarifas do plano pago padrão (GOOGLE, 2026). O custo estimado foi, portanto, de cerca de US$ 0,13 para o lote, ou US$ 0,008 por chamada. A rejeição de arquivos inválidos não foi testada neste piloto.

4.5. Proteção de dados e limitações

O mascaramento reduz a exposição de formatos previsíveis, mas não reconhece nomes sem rótulo nem números falados por extenso. Além disso, ele atua sobre o texto: o áudio original chega ao S3 e ao AWS Transcribe sem tratamento, e a saída bruta do serviço fica no armazenamento gerenciado da AWS. Como a rotina de retenção não foi implementada, o sistema não elimina esses dados ao fim do tratamento. Essas medidas não equivalem a anonimização e, isoladamente, não estabelecem conformidade com a LGPD (BRASIL, 2018).

O piloto usou chamadas roteirizadas, com declarações explícitas do atendente e áudio de boa qualidade, cenário favorável que explica em parte a concordância total em resolução e declaração. Os resultados não se estendem a chamadas reais, com ruído, interrupções e fala sobreposta. Os avaliadores e os autores das transcrições são autores do trabalho e conheciam os roteiros, o que tende a elevar a concordância entre eles. A distinção entre sentimento positivo e neutro contou com um único avaliador, e o tamanho da amostra limita a precisão das métricas por classe.

5. CONSIDERAÇÕES FINAIS

Este trabalho desenvolveu um protótipo de auditoria automatizada de chamadas e respondeu à pergunta sobre quais requisitos ele atende. Na versão avaliada, o protótipo atende integralmente 10 dos 15 requisitos funcionais, entre eles o recebimento por *upload*, a transcrição, a análise estruturada pelo Gemini, a persistência criptografada, a consulta e a sinalização de divergências entre o status declarado pelo atendente e a resolução avaliada, que era a função

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

central do projeto. Atende parcialmente o recebimento automático, que depende de um repositório intermediário, os filtros, a exportação e a notificação de gravações inválidas, e não implementa a gestão de usuários.

No piloto, os serviços reais processaram as 16 chamadas simuladas. Em relação à referência humana, o sistema acertou resolução, declaração e divergência em todas as chamadas e o sentimento em 13 de 16, com tendência a classificar como positivo o cliente neutro que agradece ao final. A transcrição apresentou WER entre 8,8% e 11,1%, e o custo estimado foi de US$ 0,008 por chamada. Esses resultados mostram que o fluxo funciona em condições controladas; a robustez em chamadas reais ainda precisa ser medida.

A principal contribuição técnica é a separação entre extração e decisão: o modelo de linguagem extrai a declaração, a avaliação e as evidências, e o código confere as evidências e decide a divergência. Os trabalhos futuros incluem a integração direta com uma plataforma de URA, filtros e agregações no servidor, a diarização das falas, a rotina de retenção de dados e a avaliação com chamadas reais anonimizadas, submetida à apreciação ética, para medir o desempenho em condições operacionais.

CONTRIBUIÇÃO DOS AUTORES

Omitida para avaliação cega.

DECLARAÇÃO DE USO DE INTELIGÊNCIA ARTIFICIAL

O SAACU utiliza o Gemini (Google) no fluxo de análise, conforme a Seção 3.3. Na preparação do manuscrito, foram utilizados Claude (Anthropic) para apoio à revisão, organização do texto, busca de referências e inspeção do código, e Codex (OpenAI) para consolidar evidências do piloto e revisar a apresentação dos resultados. Os autores são responsáveis pela conferência e aprovação do conteúdo.

REFERÊNCIAS

AMAZON WEB SERVICES. **Amazon Transcribe pricing**. 2026. Disponível em: https://aws.amazon.com/transcribe/pricing/. Acesso em: 15 nov. 2026.

ARMBRUST, Michael; FOX, Armando; GRIFFITH, Rean; JOSEPH, Anthony D.; KATZ, Randy; KONWINSKI, Andy; LEE, Gunho; PATTERSON, David; RABKIN, Ariel; STOICA, Ion; ZAHARIA, Matei. A view of cloud computing. **Communications of the ACM**, v. 53, n. 4, p. 50-58, 2010.

BATINI, Carlo; CAPPIELLO, Cinzia; FRANCALANCI, Chiara; MAURINO, Andrea. Methodologies for data quality assessment and improvement. **ACM Computing Surveys**, v. 41, n. 3, art. 16, 2009. DOI: 10.1145/1541880.1541883.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

BRASIL. Lei nº 13.709, de 14 de agosto de 2018. Lei Geral de Proteção de Dados Pessoais (LGPD). Brasília, DF: Presidência da República, 2018.

BUSCHMANN, Frank; MEUNIER, Regine; ROHNERT, Hans; SOMMERLAD, Peter; STAL, Michael. **Pattern-Oriented Software Architecture**: a system of patterns. Chichester: John Wiley & Sons, 1996. v. 1.

CANDIDO JUNIOR, Arnaldo; CASANOVA, Edresson; SOARES, Anderson da Silva; OLIVEIRA, Frederico Santos de; OLIVEIRA, Lucas; FERNANDES JUNIOR, Ricardo Corso; SILVA, Daniel Peixoto Pinto da; FAYET, Fernando Gorgulho; CARLOTTO, Bruno Baldissera; GRIS, Lucas Rafael Stefanel; ALUÍSIO, Sandra Maria. CORAA ASR: a large corpus of spontaneous and prepared speech manually validated for speech recognition in Brazilian Portuguese. **Language Resources and Evaluation**, v. 57, n. 3, p. 1139-1171, 2023. DOI: 10.1007/s10579-022-09621-4.

CAVOUKIAN, Ann. **Privacy by Design**: the 7 foundational principles. Toronto: Information and Privacy Commissioner of Ontario, 2011.

CHAUDHURI, Surajit; DAYAL, Umeshwar; NARASAYYA, Vivek. An overview of business intelligence technology. **Communications of the ACM**, v. 54, n. 8, p. 88-98, 2011. DOI: 10.1145/1978542.1978562.

COHEN, Jacob. A coefficient of agreement for nominal scales. **Educational and Psychological Measurement**, v. 20, n. 1, p. 37-46, 1960.

COHN, Mike. **User Stories Applied**: for agile software development. Boston: Addison-Wesley, 2004.

EVANS, Eric. **Domain-Driven Design**: tackling complexity in the heart of software. Boston: Addison-Wesley, 2003.

FEW, Stephen. **Information Dashboard Design**: the effective visual communication of data. Sebastopol: O'Reilly Media, 2006.

FORD, Neal; PARSONS, Rebecca; KUA, Patrick. **Building Evolutionary Architectures**: support constant change. Sebastopol: O'Reilly Media, 2017.

FOWLER, Martin. **Patterns of Enterprise Application Architecture**. Boston: Addison-Wesley, 2002.

FU, Xue-Yong; CHEN, Cheng; LASKAR, Md Tahmid Rahman; GARDINER, Shayna; HIRANANDANI, Pooja; BHUSHAN TN, Shashi. Entity-level sentiment analysis in contact center telephone conversations. **arXiv**, 2210.13401, 2022.

GAMMA, Erich; HELM, Richard; JOHNSON, Ralph; VLISSIDES, John. **Design Patterns**: elements of reusable object-oriented software. Reading: Addison-Wesley, 1994.

GOOGLE. Gemini Developer API pricing. Google AI for Developers, 2026. Disponível em: https://ai.google.dev/gemini-api/docs/pricing. Acesso em: 13 nov. 2026.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

GOSLING, James; JOY, Bill; STEELE, Guy; BRACHA, Gilad; BUCKLEY, Alex. **The Java Language Specification**: Java SE 8 Edition. Upper Saddle River: Addison-Wesley, 2014.

HINTON, Geoffrey; DENG, Li; YU, Dong; DAHL, George E.; MOHAMED, Abdel-rahman; JAITLY, Navdeep; SENIOR, Andrew; VANHOUCKE, Vincent; NGUYEN, Patrick; SAINATH, Tara N.; KINGSBURY, Brian. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. **IEEE Signal Processing Magazine**, v. 29, n. 6, p. 82-97, 2012.

INGLE, Digvijay Anil; SACHDEVA, Aashraya; SAHU, Surya Prakash; SATI, Mayank; GEORGE, Cijo; VEPA, Jithendra. Probing the depths of language models' contact-center knowledge for quality assurance. In: CONFERENCE ON EMPIRICAL METHODS IN NATURAL LANGUAGE PROCESSING: INDUSTRY TRACK, 2024, Miami. **Proceedings** [...]. Miami: Association for Computational Linguistics, 2024. p. 790-804. DOI: 10.18653/v1/2024.emnlp-industry.60.

JI, Ziwei; LEE, Nayeon; FRIESKE, Rita; YU, Tiezheng; SU, Dan; XU, Yan; ISHII, Etsuko; BANG, Ye Jin; MADOTTO, Andrea; FUNG, Pascale. Survey of hallucination in natural language generation. **ACM Computing Surveys**, v. 55, n. 12, art. 248, p. 1-38, 2023.

JUANG, Biing-Hwang; RABINER, Lawrence R. Speech recognition, automatic: history. In: BROWN, Keith (ed.). **Encyclopedia of Language and Linguistics**. 2. ed. Oxford: Elsevier, 2006. p. 806-819.

JURAFSKY, Daniel; MARTIN, James H. **Speech and Language Processing**: an introduction to natural language processing, computational linguistics, and speech recognition. 3. ed. (rascunho). Stanford: Stanford University, 2021. Disponível em: https://web.stanford.edu/~jurafsky/slp3/. Acesso em: 12 nov. 2026.

LANDIS, J. Richard; KOCH, Gary G. The measurement of observer agreement for categorical data. **Biometrics**, v. 33, n. 1, p. 159-174, 1977.

LIMA, Rodrigo; LEAL, Sidney E.; CANDIDO JUNIOR, Arnaldo; ALUÍSIO, Sandra M. A large dataset of spontaneous speech with the accent spoken in São Paulo for automatic speech recognition evaluation. In: BRAZILIAN CONFERENCE ON INTELLIGENT SYSTEMS (BRACIS), 13., 2024, Belém. **Anais** [...]. Porto Alegre: Sociedade Brasileira de Computação, 2024. p. 33-47.

RICHARDS, Mark; FORD, Neal. **Fundamentals of Software Architecture**: an engineering approach. Sebastopol: O'Reilly Media, 2020.

ROSS, Ronald G. **Principles of the Business Rule Approach**. Boston: Addison-Wesley, 2003.

SCULLEY, D.; HOLT, G.; GOLOVIN, D.; DAVYDOV, E.; PHILLIPS, T.; EBNER, D.; CHAUDHARY, V.; YOUNG, M.; CRESPO, J.-F.; DENNISON, D. Hidden technical debt in machine learning systems. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, 28., 2015, Montreal. **Proceedings** [...]. 2015. p. 2503-2511.

SHIN, Hagyeong; MATHARU, Navjot; DALAL, Binoy Robin; MUIR, Ryan; WONG, Samuel W. K.; BIALYNICKA-BIRULA, Iwona; YANG, Xingwei. FECT: factuality evaluation of interpretive AI-generated claims in contact center conversation transcripts. In: ACM SIGKDD CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, 31., 2025, Toronto. **Proceedings** [...]. New York: ACM, 2025.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

SHNEIDERMAN, Ben. The eyes have it: a task by data type taxonomy for information visualizations. In: IEEE SYMPOSIUM ON VISUAL LANGUAGES, 1996, Boulder. **Proceedings** [...]. 1996. p. 336-343.

SLACK, Nigel; CHAMBERS, Stuart; JOHNSTON, Robert. **Administração da produção**. 3. ed. São Paulo: Atlas, 2009.

VASWANI, Ashish; SHAZEER, Noam; PARMAR, Niki; USZKOREIT, Jakob; JONES, Llion; GOMEZ, Aidan N.; KAISER, Łukasz; POLOSUKHIN, Illia. Attention is all you need. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, 30., 2017, Long Beach. **Proceedings** [...]. 2017. p. 5998-6008.

WANG, Peiyi; LI, Lei; CHEN, Liang; CAI, Zefan; ZHU, Dawei; LIN, Binghuai; CAO, Yunbo; KONG, Lingpeng; LIU, Qi; LIU, Tianyu; SUI, Zhifang. Large language models are not fair evaluators. In: ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS, 62., 2024, Bangkok. **Proceedings** [...]. Bangkok: Association for Computational Linguistics, 2024. p. 9440-9450. DOI: 10.18653/v1/2024.acl-long.511.

ZHENG, Lianmin; CHIANG, Wei-Lin; SHENG, Ying; ZHUANG, Siyuan; WU, Zhanghao; ZHUANG, Yonghao; LIN, Zi; LI, Zhuohan; LI, Dacheng; XING, Eric P.; ZHANG, Hao; GONZALEZ, Joseph E.; STOICA, Ion. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, 36., 2023, New Orleans. **Proceedings** [...]. 2023.

ZWEIG, Geoffrey; SIOHAN, Olivier; SAON, George; RAMABHADRAN, Bhuvana; POVEY, Daniel; MANGU, Lidia; KINGSBURY, Brian. Automated quality monitoring for call centers using speech and NLP technologies. In: HUMAN LANGUAGE TECHNOLOGY CONFERENCE OF THE NAACL: DEMONSTRATIONS, 2006, New York. **Proceedings** [...]. New York: Association for Computational Linguistics, 2006. p. 292-295.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.