UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE GRADUAÇÃO
DEPARTAMENTO DE ENGENHARIAS E TECNOLOGIA
CURSO BACHARELADO INTERDISCIPLINAR EM TECNOLOGIA DE INFORMAÇÃO

JOÃO GUSTAVO SOUZA LIMA

PROPOSTA DE UM SISTEMA DE TRANSCRIÇÃO BASEADO EM *LARGE LANGUAGE MODEL* PARA GERAÇÃO DE ATAS

PAU DOS FERROS
2026

JOÃO GUSTAVO SOUZA LIMA

PROPOSTA DE UM SISTEMA DE TRANSCRIÇÃO BASEADO EM *LARGE LANGUAGE MODEL* PARA GERAÇÃO DE ATAS

Monografia apresentada à Universidade Federal Rural do Semi-Árido como requisito para obtenção do título de Bacharel em Bacharelado Interdisciplinar em Tecnologia de Informação.

Orientadora: Rosana Cibely Batista Rego, Prof. Dra.

PAU DOS FERROS

2026

L732p Lima, João Gustavo Souza.
Proposta de um sistema de transcrição baseado em large language model para geração de atas / João Gustavo Souza Lima. - 2026.
68 f. : il.

Orientadora: Rosana Cibely Batista Rego.
Monografia (graduação) - Universidade Federal Rural do Semi-árido, Curso de Tecnologia da Informação, 2026.

1. Reconhecimento automático de fala. 2. Grandes modelos de linguagem. 3. Engenharia de prompts. 4. Whisper. 5. Geração de atas de reunião. I. Rego, Rosana Cibely Batista, orient. II. Título.

Ficha catalográfica elaborada por sistema gerador automático em conformidade com AACR2 e os dados fornecidos pelo) autor(a).
Biblioteca Campus Pau dos Ferros
Bibliotecária: Erlanda Maria Lopes da Silva
CRB: 15/966

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC´s) foi desenvolvido pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Bibliotecas da Universidade Federal Rural do Semi-Árido (SISBI-UFERSA), sendo customizado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC) sob orientação dos bibliotecários da instituição para ser adaptado às necessidades dos alunos dos Cursos de Graduação e Programas de Pós-Graduação da Universidade.

JOÃO GUSTAVO SOUZA LIMA

PROPOSTA DE UM SISTEMA DE TRANSCRIÇÃO BASEADO EM *LARGE LANGUAGE MODEL* PARA GERAÇÃO DE ATAS

Monografia apresentada à Universidade Federal Rural do Semi-Árido como requisito para obtenção do título de Bacharel em Bacharelado Interdisciplinar em Tecnologia de Informação.

Defendida em: 10 / 06 / 2026

BANCA EXAMINADORA

Rosana Cibely Batista Rego, Prof. Dra. (UFERSA)
Presidente

Walber José Adriano Silva, Prof. Dr. (UFERSA)
Membro Examinador

José Ferdinandy Silva Chagas, Prof. Me. (UFERSA)
Membro Examinador

DEDICATÓRIA

Dedico este trabalho à minha querida mãe, Maria Euda. Espero que, de onde estiver, você sinta orgulho da trajetória que construí sob a sua benção. Este trabalho é a materialização do futuro que você sempre desejou para mim. (*in Memoriam*)

AGRADECIMENTOS

Agradeço primeiramente a Deus.

A minha familia, Elaine, Gabriela e Clara, por todo o apoio e incentivo. Sem elas, tudo seria mais difícil.

À minha orientadora, Prof. Dra. Rosana Cibely Batista Rego, por todas as orientações e pelo suporte concedido.

Aos professores da banca examinadora, Prof. Dr. Walber José Adriano Silva e Prof. Me. José Ferdinandy Silva Chagas, pelas valiosas contribuições para este trabalho.

Ao meu colega de pesquisa Johan Pedro, por todas as discussões e colaborações durante o desenvolvimento do sistema, sendo ele o responsável pelo desenvolvimento da interface web.

Aos amigos de jornada, meu sincero agradecimento. Por cada café, cada grupo de estudo e cada conversa fora de hora que deixou os dias mais leves e os desafios acadêmicos mais suportáveis.

EPÍGRAFE

"O mundo não é um mar de rosas; é um lugar sujo, um lugar cruel, que não quer saber o quanto você é durão. Vai botar você de joelhos e você vai ficar de joelhos para sempre se você deixar. Você, eu, ninguém vai bater tão forte como a vida, mas não se trata de bater forte. Se trata de quanto você aguenta apanhar e seguir em frente, o quanto você é capaz de aguentar e continuar tentando. É assim que se consegue vencer."

(Sylvester Stallone (Rocky Balboa) no filme Rocky VI.)

RESUMO

O crescimento do uso de plataformas de videoconferência em ambientes institucionais e organizacionais intensificou a necessidade de soluções que automatizem o registro formal de reuniões. A elaboração manual de atas demanda tempo considerável e está sujeita a inconsistências decorrentes de falhas humanas, comprometendo a confiabilidade e a padronização documental. Este trabalho propõe um sistema integrado de transcrição automática e aprimoramento textual voltado à geração automatizada de atas de reuniões em língua portuguesa brasileira. A arquitetura do sistema é composta por três etapas principais: pré-processamento do sinal de áudio com supressão de ruído e conversão para o formato WAV; transcrição automática utilizando o modelo Whisper da OpenAI, executado localmente; e refinamento textual assistido pelo modelo de linguagem de grande escala DeepSeek-R1, acessado via API, por meio de técnicas de engenharia de *prompt*. O sistema foi implementado como uma API REST utilizando Python e o *framework* Django REST framework, com interface desenvolvida em React. Para validação experimental, foram utilizados vídeos de reuniões administrativas públicas da UFERSA, o desempenho dos modelos de transcrição foram avaliados pela métrica *Word Error Rate* (WER), enquanto a qualidade do refinamento textual foi medida pelas métricas ROUGE-L, SBERT e Perplexity. Os resultados indicaram que o modelo Whisper turbo obteve o melhor desempenho, com WER médio de 0,178, enquanto a estratégia de Engenharia de *Prompt* Automático (APE - *Automatic Prompt Engineering*) superou a abordagem *few-shot* no refinamento textual, alcançando ROUGE-L médio de 0,67 e SBERT de 0,95. O sistema demonstrou capacidade de corrigir erros ortográficos e gramaticais, remover disfluências típicas da fala espontânea e estruturar o conteúdo em formato documental adequado ao padrão institucional de atas, sendo disponibilizado ao usuário final como documento DOCX.

Palavras-chave: reconhecimento automático de fala; grandes modelos de linguagem; engenharia de prompts; whisper; geração de atas de reunião.

ABSTRACT

The increasing use of video conferencing platforms in institutional and organizational environments has intensified the need for solutions that automate the formal recording of meetings. Manually drafting minutes is time-consuming and subject to inconsistencies due to human error, compromising reliability and document standardization. This work proposes an integrated system for automatic transcription and text enhancement aimed at the automated generation of meeting minutes in Brazilian Portuguese. The system architecture consists of three main stages: audio signal pre-processing with noise suppression and conversion to WAV format; automatic transcription using the OpenAI Whisper model, executed locally; and text refinement assisted by the DeepSeek-R1 large-scale language model, accessed via API, using prompt engineering techniques. The system was implemented as a REST API using Python and the Django REST framework, with an interface developed in React. For experimental validation, videos of public administrative meetings at UFERSA were used. The performance of the transcription models was evaluated using the Word Error Rate (WER) metric, while the quality of text refinement was measured using the ROUGE-L, SBERT, and Perplexity metrics. The results indicated that the Whisper Turbo model obtained the best performance, with an average WER of 0.178, while the Automatic Prompt Engineering (APE) strategy outperformed the few-shot approach in text refinement, achieving an average ROUGE-L of 0.67 and an SBERT of 0.95. The system demonstrated the ability to correct spelling and grammatical errors, remove disfluencies typical of spontaneous speech, and structure the content in a document format suitable to the institutional standard for minutes, being made available to the end user as a DOCX document.

Keywords: automatic speech recognition; large Language models; prompt engineering; whisper; meeting minutes generation.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE QUADROS

SUMÁRIO

1 INTRODUÇÃO

Nos últimos anos, a intensificação do uso de plataformas de videoconferências e reuniões virtuais ampliou a necessidade de soluções que forneçam o registro adequado de informações (Graham; Roll, 2024). Nesse contexto, as atas de reuniões desempenham um papel fundamental, visto que compõem documentos formais que registram decisões, deliberações e encaminhamentos de maneira organizada e padronizada.

O avanço significativo das tecnologias de reconhecimento automático de fala (ASR – *Automatic Speech Recognition*) tem possibilitado a transcrição automática de áudio em diversos idiomas (Strauss *et al.*, 2022). Ferramentas comerciais oferecidas por empresas como Microsoft (Microsoft Azure Speech), Google (Google Speech-to-Text) e OpenAI (modelo de código aberto Whisper) permitem a transcrição de áudio com ganhos significativos em produtividade. Paralelamente, os Grandes Modelos de Linguagem (LLMs – *Large Language Models*) proporcionam desempenho de ponta em tarefas de geração de texto, sumarização, tradução automática e resposta a perguntas, possibilitando também o aprimoramento textual com adição de clareza, coesão e adequação ao contexto.

Apesar desses avanços, ainda existem desafios significativos quando se trata da língua portuguesa brasileira, sobretudo em contextos de reuniões, onde ocorrem sobreposição de falas, variações dialógicas, sotaques regionais e ruídos ambientais. Muitas das soluções disponíveis no mercado não oferecem funcionalidades de refinamento textual voltadas especificamente para a elaboração de atas, limitando sua aplicabilidade em contextos institucionais que exigem padronização documental.

Encaradas como potenciais documentos de valor jurídico, as atas têm a necessidade de se constituir como registro fiel do que ocorreu na reunião (Esquinsani, 2007). Entretanto, o processo de transcrição manual para elaboração de atas demanda tempo considerável e está sujeito a falhas humanas, como inconsistências, omissões e variações na interpretação do conteúdo.

As ferramentas de transcrição automatizada disponíveis apresentam limitações, seja na precisão da transcrição para a língua portuguesa brasileira, seja na ausência de recursos que possibilitem o aprimoramento textual e a organização adequada do conteúdo para fins oficiais.

Diante desse cenário, surge a seguinte questão de pesquisa: *Como desenvolver um sistema capaz de transcrever e aprimorar automaticamente o conteúdo de reuniões de forma eficiente, confiável e padronizada, considerando as especificidades da língua portuguesa brasileira?*

Este trabalho justifica-se pela necessidade de desenvolvimento de um sistema capaz

de integrar modelos de transcrição automática e aprimoramento textual, visando gerar atas de reuniões de forma, precisa e padronizada no idioma português brasileiro. A solução proposta visa otimizar o processo manual de elaboração de atas em diferentes setores dos contextos institucional e organizacional, contribuindo para a redução do tempo de processamento, aumento da confiabilidade do registro e melhoria da qualidade documental.

1.1 Estado da Arte

A automatização do registro de informações em reuniões constitui um problema multidisciplinar, que envolve o processamento de sinais de fala, o processamento de linguagem natural e engenharia de software. Nas últimas decadas, pesquisas nessa área como a de Toledo (2017), Rabelo (2022) e Yoshizumi *et al.* (2024) têm avançado progressivamente, partindo de sistemas simples de transcrição até abordagens que integram LLMs para refinamento textual. Esta seção apresenta os principais trabalhos que fundamentaram o desenvolvimento do sistema.

Um dos primeiros trabalhos a abordar a viabilidade do uso de sistemas ASR para a geração automatizada de documentos formais em lingua portuguesa do Brasil foi Toledo (2017), que desenvolveu um protótipo de sistema web voltado á elaboração de laudos médicos por meio de ASR. O trabalho demonstrou que sistemas ASR são viáveis para o uso em contextos institucionais, mas apontou limitações relevantes: os sistemas ASR avaliados dependem integralmente de APIs (*Application Programming Interface*) externas, são sensíveis a ruídos e variação de sotaques, e não é realizado qualquer pós-processamento textual sobre a transcrição bruta, de modo que os erros do ASR chegam diretamente ao documento final.

No campo específico da avaliação do modelo Whisper para a língua portuguesa do Brasil, Rabelo (2022) realizou um estudo de caso utilizando um dataset ultilingual TEDx, composto por mais de 150 horas de áudio em português no formato .flac acompanhadas de suas respectivas transcrições. O autor submeteu cinco arquivos de áudio totalizando 1 hora e 44 minutos ao modelo Whisper *large*, e calculou o Taxa de Erro de Palavras (WER - *Word Error Rate*) para cada arquivo. O resultado sugere que o modelo tende a performar melhor em entradas de menor duração. Apesar de confirmar a aplicabilidade do Whisper ao português, Rabelo (2022) não avança para além da transcrição bruta. Essa ausência de refinamento textual em ambos os trabalhosrepresenta a lacuna que o presente trabalho superar, ao incorporar um modelo LLM responsável por corrigir, organizar e formaliza o conteúdo antes da geração do documento.

A integração de modelos ASR e LLMs para o refinamento de transcrições em língua

portuguesa foi investigada por Yoshizumi *et al.* (2024), que propôs um sistema de aprimoramento de transcrições de chamadas telefônicas entre operadores do sistema elétrico nacional e centros de transmissão. O trabalho utilizou o Whisper combinado com o modelo *Large Language Model Meta AI* (LLaMA), orientado por *prompt engineering*, com vocabulário customizado com termos técnicos específicos do setor elétrico. Os resultados mostraram redução do WER médio de 0,5195 para 0,4867 com a adição do vocabulário e das instruções ao modelo LLaMA. Yoshizumi *et al.* (2024) demonstrou que a combinação de um modelo ASR com um modelo LLM é uma abordagem promissora, mas o trabalho é restrito a um dominio tecnico altamente específico e avaliação do refinamento exclusivamente via WER, sem metricas de qualidade semântica. O presente trabalho se distingue ao aplicar essa arquitetura a um contexto de reuniões institucionais, ao comparar sistematicamente estrategias de prompting e ao empregar métricas complementares para avaliar a qualidade do refinamento.

Em conjunto, os trabalhos citados demonstram uma prograssão na área, da simples aplicação de ASR para geração de documentos (Toledo, 2017), passando pela avaliação do Whisper (Rabelo, 2022), pela combinação de ASR e LLMs (Yoshizumi *et al.*, 2024). O presente trabalho se posiciona na interseção dessas contribuições, propondo um sistema integrado que visa unir transcrição via Whisper e aprimoramento textual orientado por prompt engineering com LLM e formatação documental padronizada, avaliado por um conjunto robusto de métricas quantitativas, com foco específico no contexto de reuniões institucionais em língua portuguesa brasileira.

1.2 Motivações

A elaboração manual de atas de reuniões em ambientes institucionais e organizacionais apresenta limitações significativas quanto à eficiência, precisão e padronização. O processo manual de transcrição demanda tempo considerável, está sujeito a inconsistências decorrentes de falhas humanas e depende da interpretação individual do redator, comprometendo a confiabilidade e a reprodutibilidade do registro (Esquinsani, 2007).

Apesar dos avanços recentes em tecnologias de reconhecimento automático de fala e modelos de linguagem de grande escala, as soluções disponíveis apresentam deficiências na adaptação ao idioma português brasileiro, especialmente em contextos de reuniões com múltiplos falantes, ruídos ambientais e variações dialetais.

Este trabalho é motivado pela oportunidade de integrar modelos de transcrição automá-

tica e refinamento textual em uma arquitetura computacional baseada em API REST, visando automatizar a geração de atas com qualidade adequada para uso institucional. A solução proposta busca reduzir o esforço operacional, aumentar a padronização documental e melhorar a eficiência do fluxo de registro de informações em ambientes corporativos e acadêmicos. Do ponto de vista científico, o trabalho contribui com uma abordagem de integração de modelos de *deep learning* para processamento de áudio e texto, fornecendo subsídios para o desenvolvimento de sistemas similares em domínios específicos.

1.3 Objetivos

O presente trabalho tem como finalidade o desenvolvimento de um sistema integrado para transcrição automática de áudio e aprimoramento textual, utilizando modelos de reconhecimento automático de fala e modelos LLMs, especificamente o Whisper e o DeepSeek-R1.

1.3.1 Objetivo Geral

Este trabalho tem como objetivo geral o desenvolvimento e a integração de um módulo de transcrição de áudio e aprimoramento textual, voltado à geração automática de atas de reuniões, implementado em um sistema web construído com Python e Django REST framework (DRF).

1.3.2 Objetivos Específicos

Para atingir o objetivo geral proposto, este trabalho estabelece os seguintes objetivos específicos:

- Implementar um módulo de transcrição automática de áudio utilizando o modelo Whisper, com execução local;
- Integrar o modelo de linguagem DeepSeek-R1, hospedado em nuvem, para aprimoramento de texto;
- Desenvolver uma API REST em Python e DRF para a comunicação entre o *back-end* e o *front-end*;
- Avaliar o desempenho do sistema por meio de métricas quantitativas, incluindo WER, ROUGE-L (*Recall-Oriented Understudy for Gisting Evaluation - Longest Common Subsequence*), SBERT (*Sentence Bidirectional Encoder Representations*) e *Perplexity* (PPL).

1.4 Organização do Texto

Este trabalho está organizado em seis capítulos, estruturados de forma a apresentar sistematicamente o desenvolvimento da pesquisa, desde a fundamentação teórica até a análise dos resultados obtidos. A organização dos capítulos é descrita a seguir:

- No Capítulo 1 é apresentada a contextualização do problema de pesquisa, apresenta a justificativa, os objetivos e o estado da arte relacionado à transcrição automática de fala e geração de atas de reuniões;
- No Capítulo 2 é fundamentado os conceitos necessários para a compreensão do trabalho, abordando tecnologias de reconhecimento automático de fala, modelos de linguagem de grande escala, técnicas de pré-processamento de áudio e arquiteturas de sistemas baseados em API REST;
- No Capítulo 3 é descrito os procedimentos metodológicos adotados, incluindo a seleção dos modelos Whisper e DeepSeek-R1, os critérios de avaliação de desempenho e a estrutura experimental utilizada;
- No Capítulo 4 é apresentada a arquitetura do sistema proposto, detalhando os módulos de transcrição, refinamento textual e a implementação da API em Python e DRF;
- No Capítulo 5 são apresentados os experimentos realizados para validação do sistema, incluindo a análise comparativa de modelos de transcrição, métricas de desempenho (WER, ROUGE-L, SBERT e Perplexity) e discussão dos resultados quantitativos e qualitativos;
- No Capítulo 6 é sintetizada as principais contribuições do trabalho, analisa o cumprimento dos objetivos propostos, identifica limitações do sistema desenvolvido e sugere direções para trabalhos futuros.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta o referencial teórico que fundamenta o desenvolvimento do sistema de transcrição e sumarização de reuniões proposto neste trabalho. Inicialmente, são abordados os princípios matemáticos e computacionais subjacentes ao ASR, detalhando as etapas de extração de características, modelagem acústica, modelagem de linguagem, modelagem de pronúncia e decodificação. Em seguida, discute-se a evolução dos LLMs, com ênfase na arquitetura *Transformer*.

2.1 Reconhecimento Automático de Fala

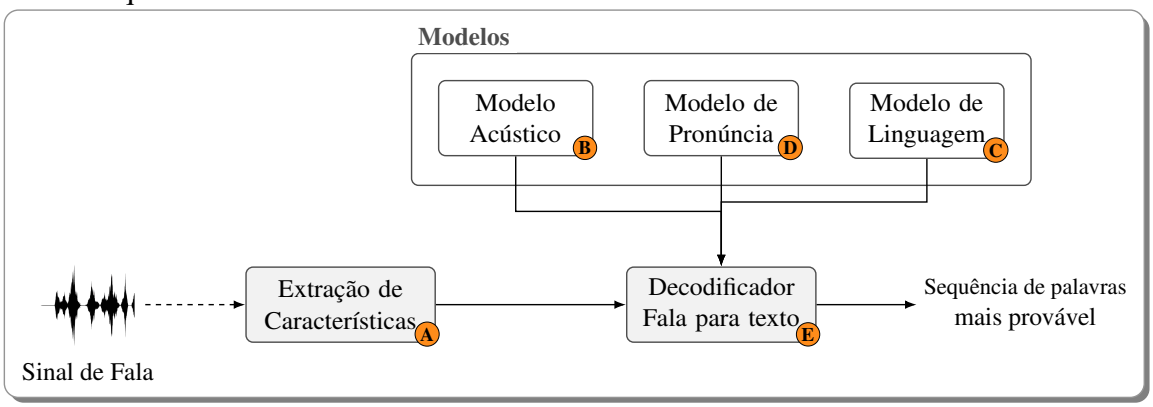
O ASR constitui um processo tecnológico que emprega algoritmos computacionais avançados para realizar a transcodificação de sinais acústicos de fala em sequências textuais estruturadas (Indurkhya; Damerau, 2010 apud Fendji *et al.*, 2022)p. 2. Formalmente, um sistema ASR pode ser definido como uma função de mapeamento $f : \mathscr{X} \rightarrow \mathscr{W}^*$, onde $\mathscr{X}$ representa o espaço de sinais acústicos contínuos e $\mathscr{W}^*$ denota o conjunto de todas as sequências possíveis sobre um vocabulário finito $\mathscr{W}$.

Conforme ilustrado na Figura 1, a arquitetura canônica de um sistema ASR é composta por cinco módulos funcionais interdependentes: (A) Extração de Características, responsável pela transformação do sinal acústico bruto em representações compactas e discriminativas; (B) Modelo Acústico, que estabelece a correspondência probabilística entre características acústicas e unidades fonéticas; (C) Modelo de Linguagem, que codifica restrições sintáticas e semânticas da língua-alvo; (D) Modelo de Pronúncia, que mapeia sequências fonêmicas em palavras do vocabulário; e (E) Decodificador, que realiza a busca pela sequência de palavras mais provável dada a evidência acústica (Fendji *et al.*, 2022).

2.1.1 Extração de Características

A extração de características constitui uma etapa crítica no *pipeline* de processamento de ASR, executando a transformação de sinais acústicos de alta dimensionalidade em representações vetoriais compactas e discriminativas que preservam informações fonéticamente relevantes (Labied; Belangour, 2021). Matematicamente, esta etapa realiza uma transformação $\phi : \mathbb{R}^T \rightarrow \mathbb{R}^{F \times N}$, onde T representa o número de amostras temporais do sinal, F é a dimensionalidade do vetor de características e N é o número de *frames* resultantes da segmentação temporal.

Figura 1 – Arquitetura do sistema ASR



Fonte: Adaptado de Fendji *et al.* (2022).

Conforme indicado no item A da Figura 1, o módulo de extração recebe o sinal acústico digital amostrado, aplica técnicas de pré-processamento para remoção de componentes espúrias (ruído de fundo, componentes *Direct Current* (DC) e realiza a conversão em vetores de características que alimentam o modelo acústico (item B). As técnicas mais consolidadas para esta finalidade incluem os Coeficientes Cepstrais de Frequência Mel (*Mel-Frequency Cepstral Coefficients* - MFCCs) e a Codificação Preditiva Linear (*Linear Predictive Coding* - LPC) (Labied; Belangour, 2021).

2.1.1.1 Coeficientes Cepstrais de Frequência Mel

Uma das técnicas mais práticas e conhecidas na área de processamento de sinais são os MFCCs (Saxena; Farooqui; Ali, 2022). A técnica baseia-se na observação de que o sistema auditivo humano não processa frequências de forma linear, mas sim seguindo uma escala perceptual (escala Mel), onde a discriminação de frequências é aproximadamente linear até 1000 Hz e logarítmica acima deste limiar.

O processo de cálculo dos MFCCs compreende as seguintes etapas algorítmicas:

(1) Segmentação temporal: O sinal de fala $s(t)$ é dividido em *frames* de curta duração, tipicamente entre 20 e 30 milissegundos, com sobreposição de 50% (10-15 ms) entre *frames* consecutivos. Esta sobreposição implementa uma janela deslizante que suaviza transições e garante continuidade na análise espectral (Basak *et al.*, 2023).

(2) Janelamento: Cada *frame* $s_i(n)$ é multiplicado por uma função janela $w(n)$ para atenuar descontinuidades nas bordas que causariam vazamento espectral (*spectral leakage*) na

análise de Fourier subsequente. A janela de Hamming é comumente empregada devido ao seu bom balanço entre resolução espectral e atenuação de lóbulos laterais, sendo definida por:

$$w(n) = 0,54 - 0,46\cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \leq n \leq N-1. \tag{2.1}$$

onde N representa o comprimento da janela (número total de amostras por *frame*) e n indexa a amostra temporal dentro do *frame*.

(3) Análise espectral: Aplica-se a Transformada Rápida de Fourier (*Fast Fourier Transform* - FFT) ao *frame* janelado para obter sua representação no domínio da frequência (Chang; Hung, 2022):

$$S_i(k) = \sum_{n=0}^{N-1} s_i(n)w(n)e^{-j2\pi kn/N}, \quad k = 0,1,\ldots,N-1. \tag{2.2}$$

onde $S_i(k)$ representa o espectro de magnitude do i-ésimo *frame*. A FFT reduz a complexidade computacional de $O(N^2)$ para $O(N\log N)$, viabilizando processamento em tempo real (Fendji *et al.*, 2022).

(4) Filtragem Mel: O espectro de potência $|S_i(k)|^2$ é mapeado na escala perceptual Mel através de um banco de filtros triangulares. A conversão entre frequência Hertz (f) e escala Mel (m) é dada por:

$$m = 2595\log_{10}\left(1 + \frac{f}{700}\right). \tag{2.3}$$

O banco de filtros Mel compreende tipicamente $M = 20$ a 40 filtros triangulares espaçados uniformemente na escala Mel, onde filtros de frequência superior apresentam maior largura de banda absoluta que filtros de frequência inferior, refletindo a resolução espectral decrescente do sistema auditivo humano em altas frequências (Labied; Belangour, 2021). A energia de cada banda Mel é calculada como:

$$E_i(m) = \sum_{k=0}^{N-1} |S_i(k)|^2 H_m(k), \quad m = 1,2,\ldots,M. \tag{2.4}$$

onde $H_m(k)$ representa a resposta em frequência do m-ésimo filtro triangular.

(5) Compressão logarítmica: Aplica-se o logaritmo às energias Mel para simular a resposta logarítmica da percepção de intensidade sonora:

$$\log E_i(m) = \log\left(\sum_{k=0}^{N-1} |S_i(k)|^2 H_m(k)\right). \tag{2.5}$$

(6) Transformada Discreta de Cosseno (DCT): Finalmente, a Transformada Discreta de Cosseno (*Discrete Cosine Transform* - DCT) é aplicada aos coeficientes log-Mel para decorrelacionar as componentes e compactar a informação nos primeiros coeficientes cepstrais (Basak *et al.*, 2023):

$$c_i(n) = \sum_{m=1}^{M} \log E_i(m) \cos\left[n\,(m-0.5)\,\frac{\pi}{M}\right], \quad n = 0,1,\ldots,C-1. \tag{2.6}$$

onde $c_i(n)$ representa o n-ésimo coeficiente MFCC do *frame* i, e C (tipicamente 12-13) indica o número de coeficientes retidos. O vetor resultante $\mathbf{c}_i = [c_i(0), c_i(1), \ldots, c_i(C-1)]^T$ constitui o vetor de características acústicas que codifica informações fonéticas relevantes de forma compacta.

2.1.1.2 Codificação Preditiva Linear

A Codificação Preditiva Linear (*Linear Predictive Coding* - LPC) representa uma abordagem de modelagem paramétrica do trato vocal baseada em análise temporal, fundamentada na teoria de produção de fala que modela o trato vocal como um filtro linear excitado por uma fonte de energia (Basak *et al.*, 2023). O modelo LPC assume que cada amostra de fala $s(n)$ pode ser aproximada por uma combinação linear das p amostras anteriores:

$$\hat{s}(n) = \sum_{k=1}^{p} a_k s(n-k). \tag{2.7}$$

onde a_k são os coeficientes de predição linear que parametrizam as características espectrais do sinal. O objetivo é determinar os coeficientes que minimizam o erro quadrático médio de predição $e(n) = s(n) - \hat{s}(n)$.

O processo de extração dos coeficientes LPC compreende as seguintes etapas:

Pré-ênfase: Aplicação de um filtro de primeira ordem $H(z) = 1 - \alpha z^{-1}$ (com $\alpha \approx 0.97$) para compensar a queda natural de alta frequência do espectro de fala e balancear a energia espectral.

Segmentação: Divisão do sinal em *frames* de 20-30 ms com sobreposição, similar ao processo MFCC.

Janelamento: Multiplicação por janela de Hamming (Equação 2.1) para suavização espectral.

Análise de autocorrelação: Cálculo da função de autocorrelação $R(k) = \sum_{n=0}^{N-1-k} s(n)s(n+k)$ seguido da resolução do sistema de equações normais de Yule-Walker através do algoritmo de Levinson-Durbin para determinação eficiente dos coeficientes a_k.

2.1.2 Modelo Acústico

O modelo acústico (Item B, Figura 1) estabelece o mapeamento probabilístico entre vetores de características acústicas e unidades linguísticas (fonemas, trifones ou estados fonéticos) (Bhatt; Jain; Dev, 2020). Formalmente, dado um vetor de observações acústicas $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T]$, o modelo acústico computa a distribuição de probabilidade $P(\mathbf{X}|Q)$, onde $Q = [q_1, q_2, \ldots, q_T]$ representa uma sequência de estados fonéticos (Benkerzaz; Elmir; Dennai, 2019).

2.1.2.1 Modelos Ocultos de Markov

Tradicionalmente, a modelagem acústica baseia-se em Modelos Ocultos de Markov (*Hidden Markov Models* - HMM), uma classe de modelos estatísticos que descrevem processos estocásticos duplamente embebidos (Eddy, 1996). Um HMM é formalmente definido pela tupla $\lambda = (\mathscr{S}, \mathscr{O}, \pi, \mathbf{A}, \mathbf{B})$, onde $\mathscr{S} = \{s_1, s_2, \ldots, s_N\}$ é o conjunto finito de N estados ocultos (não observáveis diretamente); $\mathscr{O}$ é o espaço de observações (vetores acústicos); $\pi = [\pi_1, \pi_2, \ldots, \pi_N]$ é o vetor de probabilidades iniciais, onde $\pi_i = P(q_1 = s_i)$; $\mathbf{A} = [a_{ij}]_{N \times N}$ é a matriz de probabilidades de transição, com $a_{ij} = P(q_{t+1} = s_j | q_t = s_i)$ satisfazendo $\sum_{j=1}^{N} a_{ij} = 1$; $\mathbf{B} = \{b_j(\mathbf{x})\}$ é o conjunto de densidades de probabilidade de emissão, onde $b_j(\mathbf{x}) = P(\mathbf{x}_t | q_t = s_j)$ modela a distribuição das observações acústicas em cada estado.

Em sistemas ASR, cada fonema é tipicamente modelado por um HMM com arquitetura esquerda-direita (*left-to-right*) de três a cinco estados, refletindo a natureza temporal e sequencial da produção de fala (Fendji *et al.*, 2022). Para capturar efeitos de coarticulação (influência de fonemas adjacentes na realização acústica), utiliza-se modelagem dependente de contexto baseada em trifones, onde cada estado representa não apenas um fonema isolado, mas um fonema no contexto de seus vizinhos esquerdo e direito.

As densidades de emissão $b_j(\mathbf{x})$ são frequentemente modeladas por Misturas de Gaussianas (*Gaussian Mixture Models* - GMMs):

$$b_j(\mathbf{x}) = \sum_{m=1}^{M} c_{jm} \mathcal{N}(\mathbf{x}; \mu_{jm}, \Sigma_{jm}). \tag{2.8}$$

onde c_{jm} são pesos de mistura ($\sum_{m=1}^{M} c_{jm} = 1$) e $\mathcal{N}(\mathbf{x}; \mu, \Sigma)$ denota a distribuição Gaussiana multivariada com média μ e matriz de covariância Σ.

2.1.3 Modelo de Linguagem

O modelo de linguagem (*Language Model* - LM) (item C da Figura 1) desempenha papel crucial em sistemas ASR ao codificar restrições sintáticas, semânticas e estatísticas da língua-alvo, permitindo a distinção entre sequências de palavras plausíveis e implausíveis (Fendji *et al.*, 2022). Formalmente, dado um vocabulário $\mathscr{W}$ e uma sequência de palavras $W = w_1 w_2 \cdots w_L$, o modelo de linguagem fornece a distribuição de probabilidade $P(W)$ sobre todas as sequências possíveis.

Pela regra da cadeia de probabilidades, a probabilidade de uma sequência completa pode ser decomposta como:

$$P(W) = P(w_1, w_2, \ldots, w_L) = \prod_{i=1}^{L} P(w_i | w_1, \ldots, w_{i-1}). \tag{2.9}$$

2.1.3.1 Modelos N-Grama

Devido à impossibilidade de estimar confiavelmente probabilidades condicionais sobre contextos arbitrariamente longos, aplica-se a aproximação de Markov de ordem $N-1$, resultando nos modelos N-grama:

$$P(w_i | w_1, \ldots, w_{i-1}) \approx P(w_i | w_{i-N+1}, \ldots, w_{i-1}). \tag{2.10}$$

Esta aproximação assume que a probabilidade de uma palavra depende apenas das $N-1$ palavras imediatamente precedentes. Os casos mais comuns são:

- Unigrama ($N = 1$): $P(w_i)$ - independência completa entre palavras;
- Bigrama ($N = 2$): $P(w_i | w_{i-1})$ - contexto de uma palavra;

- Trigrama ($N = 3$): $P(w_i|w_{i-2}, w_{i-1})$ - contexto de duas palavras.

As probabilidades são estimadas por frequências relativas em grandes corpora textuais:

$$P(w_i|w_{i-N+1}, \ldots, w_{i-1}) = \frac{C(w_{i-N+1}, \ldots, w_{i-1}, w_i)}{C(w_{i-N+1}, \ldots, w_{i-1})}. \tag{2.11}$$

onde $C(\cdot)$ denota a função de contagem de ocorrências no corpus de treinamento. Técnicas de suavização (*smoothing*) como Laplace, Kneser-Ney ou *back-off* são empregadas para atribuir probabilidades não nulas a N-gramas não observados durante o treinamento.

2.1.4 Modelo de Pronúncia

O modelo de pronúncia (*Pronunciation Model* - PM), também denominado léxico fonético ou dicionário de pronúncia, estabelece o mapeamento entre representações ortográficas (palavras escritas) e suas realizações fonéticas (sequências de fonemas). Formalmente, o PM define uma função (potencialmente não determinística) $\Phi : \mathscr{W} \rightarrow 2^{\mathscr{P}^*}$, onde $\mathscr{W}$ é o vocabulário de palavras e $\mathscr{P}^*$ representa o conjunto de todas as sequências sobre o alfabeto fonético $\mathscr{P}$.

2.1.4.1 Abordagens para Construção de Modelos de Pronúncia

2.1.4.1.1 Abordagem baseada em dicionário

Utiliza léxicos predefinidos que mapeiam explicitamente cada palavra a uma ou mais transcrições fonéticas canônicas. Este método garante precisão para palavras conhecidas, mas não generaliza para palavras fora do vocabulário (*out-of-vocabulary* - OOV) (Fendji *et al.*, 2022).

2.1.4.1.2 Abordagem baseada em dados

Emprega técnicas de aprendizado de máquina para induzir automaticamente regras de conversão grafema-fonema a partir de exemplos anotados (Wester, 2003). Métodos incluem árvores de decisão fonológicas (*phonological decision trees*) e modelos de sequência (*Conditional Random Fields* - CRF). Mais recentemente, redes neurais passaram a capturar características mais refinadas, alcançando alto desempenho na previsão da pronúncia correta (Cheng *et al.*, 2024).

2.1.4.1.3 Abordagem baseada em conhecimento linguístico

Aplica regras fonológicas explícitas derivadas da análise linguística para gerar variantes de pronúncia contextualmente apropriadas, capturando fenômenos como assimilação, elisão, redução vocálica e outros processos fonológicos dependentes de contexto (Wester, 2003).

Em sistemas práticos, frequentemente combinam-se múltiplas abordagens: dicionários para palavras frequentes, regras fonológicas para capturar variações contextuais e modelos estatísticos para generalização a vocabulário não observado.

2.1.5 Decodificador

O decodificador (Item E, Figura 1) constitui o módulo de integração que sintetiza as informações dos componentes anteriores para produzir a transcrição final (Fendji *et al.*, 2022). Formalmente, o problema de decodificação pode ser expresso como a maximização da probabilidade *a posteriori* de uma sequência de palavras $\hat{W}$ dada a sequência de observações acústicas $\mathbf{X}$:

$$\hat{W} = \arg\max_{W \in \mathscr{W}^*} P(W|\mathbf{X}). \tag{2.12}$$

Aplicando a regra de Bayes e desconsiderando o termo de evidência $P(\mathbf{X})$ (constante para todas as hipóteses), obtém-se a formulação fundamental do decodificador ASR:
onde $P(\mathbf{X}|W)$ é fornecida pelo modelo acústico e $P(W)$ pelo modelo de linguagem.

2.1.5.1 Algoritmo de Viterbi

Devido ao espaço exponencialmente grande de hipóteses possíveis ($|\mathscr{W}|^L$ para sentenças de comprimento L), algoritmos de busca exaustiva são computacionalmente inviáveis. O algoritmo de Viterbi, baseado em programação dinâmica, explora a estrutura de Markov do problema para encontrar eficientemente a sequência de estados mais provável através de um retículo (*lattice*) de decodificação (Fendji *et al.*, 2022).

O algoritmo opera recursivamente, computando para cada estado s_j no instante t a probabilidade máxima do caminho:

$$\delta_t(j) = \max_{q_1,\ldots,q_{t-1}} P(q_1,\ldots,q_{t-1},q_t = s_j,\mathbf{x}_1,\ldots,\mathbf{x}_t|\lambda). \tag{2.13}$$

com recursão:

$$\delta_t(j) = \left[\max_i \delta_{t-1}(i) a_{ij}\right] b_j(\mathbf{x}_t). \tag{2.14}$$

A complexidade temporal é reduzida de exponencial para $O(N^2T)$, onde N é o número de estados e T o número de *frames*, tornando a decodificação tratável computacionalmente.

2.2 Modelos de Linguagem de Grande Escala

Enquanto os modelos de linguagem tradicionais, baseados em N-gramas, limitam-se a capturar dependências estatísticas de curto alcance e operam através de contagens explícitas de coocorrências, os LLMs representam um avanço paradigmático no Processamento de Linguagem Natural (PLN), fundamentado em arquiteturas de redes neurais profundas capazes de codificar representações semânticas distributivas e capturar dependências contextuais de longo alcance.

2.2.1 Arquitetura *Transformer*

A arquitetura *Transformer*, introduzida por (Vaswani *et al.*, 2017), constitui o fundamento da maioria dos LLMs contemporâneos. O modelo abandona arquiteturas recorrentes em favor de um mecanismo de atenção que permite processamento paralelo eficiente e modelagem de dependências arbitrariamente distantes.

O componente central é o mecanismo de auto-atenção multi-cabeça (*multi-head self-attention*), que computa representações contextualizadas de cada *token* ponderando sua relação com todos os demais *tokens* da sequência. Formalmente, dada uma sequência de entrada $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$, as representações de consulta (*query*), chave (*key*) e valor (*value*) são obtidas por projeções lineares (Vaswani *et al.*, 2017):

$$\mathbf{Q} = \mathbf{X}\mathbf{W}^Q, \quad \mathbf{K} = \mathbf{X}\mathbf{W}^K, \quad \mathbf{V} = \mathbf{X}\mathbf{W}^V. \tag{2.15}$$

O mecanismo de atenção escalada é então calculado como:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V}. \tag{2.16}$$

onde d_k é a dimensionalidade das representações de chave, e a normalização por $\sqrt{d_k}$ estabiliza os gradientes durante o treinamento. A função softmax converte os *scores* de atenção em distribuições de probabilidade, atribuindo pesos que indicam a relevância de cada posição para a representação contextualizada de cada *token* (Vaswani *et al.*, 2017).

2.2.2 Modelos Autorregressivos e Pré-Treinamento

LLMs modernos como GPT (*Generative Pre-trained Transformer*), Claude, DeepSeek e similares são treinados de forma autorregressiva para modelar a distribuição de probabilidade condicional:

$$P(w_1, w_2, \ldots, w_n) = \prod_{i=1}^{n} P(w_i | w_1, \ldots, w_{i-1}; \theta). \tag{2.17}$$

onde θ representa os parâmetros do modelo neural (tipicamente bilhões de pesos). O treinamento de modelos de linguagem ocorre, em geral, em duas fases complementares. Na primeira, denominada pré-treinamento não supervisionado, o modelo é exposto a vastos corpora textuais, que podem variar de centenas de bilhões a trilhões de *tokens* com o objetivo de maximizar a verossimilhança dos dados observados. Essa etapa permite que o modelo adquira um conhecimento linguístico abrangente, capturando padrões sintáticos, informações factuais e até aspectos de raciocínio causal. Na segunda fase, conhecida como ajuste fino (*fine-tuning*) e alinhamento, o modelo previamente treinado é refinado utilizando conjuntos de dados específicos, por meio de técnicas como aprendizado supervisionado direcionado a tarefas, aprendizado por reforço a partir de *feedback* humano (RLHF - *Reinforcement Learning from Human Feedback*) e outros métodos de alinhamento, com o objetivo de adequar seu comportamento a instruções e valores humanos (Naveed *et al.*, 2025).

2.2.3 Aplicação em Refinamento de Transcrições ASR

No contexto de sistemas ASR, LLMs desempenham um papel crucial no pós-processamento de transcrições brutas, oferecendo capacidades que vão além dos modelos tradicionais baseados em N-gramas. Esses modelos são capazes de realizar a correção contextual de erros ao explorar um amplo contexto semântico, identificando e corrigindo problemas de substituição, inserção e deleção por meio de uma compreensão mais profunda do significado do texto. Além disso, lidam com disfluências típicas da fala espontânea, como hesitações, repetições, falsos inícios e

marcadores discursivos promovendo sua remoção ou normalização para gerar textos mais fluentes e adequados ao registro escrito formal. Outra contribuição relevante está na segmentação e estruturação do conteúdo, permitindo organizar transcrições em unidades discursivas coerentes, como parágrafos e tópicos, bem como identificar mudanças temáticas e adaptar o texto a convenções documentais específicas. Adicionalmente, o uso de técnicas de *prompt engineering* possibilita orientar o comportamento dos LLMs por meio de instruções cuidadosamente elaboradas, incluindo o uso de exemplos (*few-shot learning*), garantindo maior controle sobre o formato, estilo e qualidade das saídas geradas. A partir dessa base teórica que integra fundamentos matemáticos e computacionais de sistemas ASR e LLMs, a próxima seção apresenta a metodologia adotada neste trabalho, detalhando os procedimentos, ferramentas e etapas utilizadas no desenvolvimento e na avaliação do sistema proposto (Ahlawat; Aggarwal; Gupta, 2025).

3 METODOLOGIA

A metodologia adotada segue um processo de experimentação e estudo de caso que compreendeu as seguintes etapas: (i) análise de requisitos e definição da arquitetura do sistema; (ii) implementação dos módulos de pré-processamento de áudio, transcrição baseada em ASR e refinamento textual assistido por LLM; (iii) integração entre *back-end* e *front-end* por meio de uma API REST; e (iv) validação experimental por meio de testes funcionais com gravações de reuniões em diferentes condições acústicas, utilizando métricas objetivas de avaliação que indicam a preservação da estrutura semântica e da fidedignidade do conteúdo transcrito.

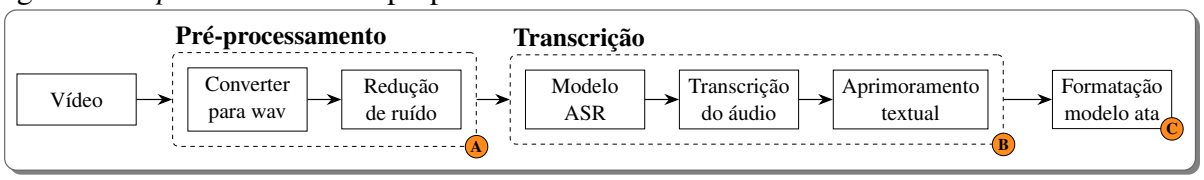
3.1 Arquitetura do Sistema

O sistema foi projetado seguindo o padrão arquitetural cliente-servidor, organizado em camadas bem definidas conforme os princípios de separação de responsabilidades (*Separation of Concerns*) e baixo acoplamento. O *back-end*, implementado com o *framework* Django REST framework, atua como núcleo de processamento responsável pela orquestração dos modelos de Inteligência Artificial, gerenciamento de estado das requisições e controle de acesso. O *front-end*, desenvolvido em React, fornece a interface gráfica para interação do usuário com o sistema.

A Figura 2 ilustra o pipeline deprocessamento do sistema, que se inicia com o recebimento de um arquivo de mídia (vídeo ou áudio) em formatos diversos. O arquivo passa por uma etapa de pré-processamento, sendo convertido para o formato de áudio WAV (*Waveform Audio File Format*) em taxa de amostragem de 16 kHz, configuração recomendada para sistemas de reconhecimento de fala (Belodedov; Fonkants; Safin, 2023). O sinal de áudio resultante é então submetido ao modelo ASR para transcrição automática, gerando uma representação textual da reunião.

Esta transcrição bruta é posteriormente encaminhada a um modelo LLM especializado, que executa operações de pós-processamento textual incluindo: (i) correção de erros ortográficos e gramaticais resultantes de imprecisões do ASR; (ii) remoção de hesitações, repetições e vícios de linguagem característicos da fala espontânea; (iii) segmentação lógica do conteúdo em seções temáticas; e (iv) formatação adequada ao padrão documental de atas. Ao final do *pipeline* de processamento, o sistema retorna um documento estruturado em formato DOCX (*Office Open XML*), representando a ata final gerada de forma automatizada e pronta para revisão humana.

Figura 2 – *Pipeline* do sistema proposto



Fonte: Autoria própria (2026).

3.1.1 Pré-processamento de Áudio

O módulo inicial do *pipeline* de processamento (Item A da Figura 2) consiste na etapa de pré-processamento e normalização do sinal de áudio. Esta fase é fundamental para garantir a qualidade dos dados de entrada e maximizar a taxa de acerto do modelo ASR nas etapas subsequentes.

O sistema aceita como entrada arquivos de mídia em diversos formatos contêineres (MP4, AVI, MKV, MOV) e codificações de áudio (MP3, WAV, AAC, OGG). Independentemente do formato original, todos os arquivos são processados pela biblioteca *pydub*, que realiza a demultiplexação (separação das trilhas de áudio e vídeo) e a transcodificação para o formato WAV com as seguintes especificações técnicas: codificação PCM (*Pulse Code Modulation*) linear de 16 bits, taxa de amostragem de 16 kHz e configuração mono (canal único). A escolha deste formato fundamenta-se na preservação da integridade do sinal, uma vez que o áudio é armazenado sem compressão com perdas, ao contrário de codecs como MP3 ou AAC que utilizam algoritmos psicoacústicos de compressão destrutiva (Belodedov; Fonkants; Safin, 2023).

Após a conversão, o sinal de áudio submete-se a uma etapa de supressão de ruído. Esta fase visa mitigar interferências acústicas comuns em ambientes de videoconferência e reuniões presenciais, tais como ruídos de ventilação, digitação em teclados, reverberação de salas, artefatos de compressão de rede e outros tipos de ruído de fundo. O objetivo principal é elevar a relação sinal-ruído (*Signal-to-Noise Ratio* - SNR), melhorando significativamente a inteligibilidade da fala e, consequentemente, a acurácia do modelo ASR.

Devido à heterogeneidade dos ambientes de gravação e às condições acústicas variáveis, o sistema processa sinais de fala contaminados por uma ampla gama de artefatos sonoros, tornando inviável a definição de um perfil de ruído estático ou predeterminado. Por esta razão, adotou-se uma abordagem de redução de ruído não estacionário, implementada através de técnicas de filtragem espectral adaptativa. Diferentemente das abordagens estacionárias clássicas,

que requerem um segmento inicial de ruído puro para estimação do perfil de interferência, os algoritmos não estacionários estimam e atenuam as componentes ruidosas de forma dinâmica e contínua ao longo de todo o sinal, adaptando-se às variações temporais do ruído simultaneamente à locução (Sainburg; Gentner, 2021). Esta estratégia mostra-se particularmente eficaz em cenários de reuniões remotas, onde o ruído apresenta características não constantes ao longo do tempo.

3.1.2 Transcrição Automática de Fala

Concluída a etapa de pré-processamento, o sinal de áudio normalizado é encaminhado ao módulo de transcrição automática, que utiliza o modelo Whisper, um sistema ASR de código aberto desenvolvido pela OpenAI, baseado em arquitetura *Transformer* e treinado em um conjunto massivo de dados multilíngues. O Whisper implementa uma abordagem *sequence-to-sequence* para conversão de fala em texto, demonstrando robustez significativa em relação a variações de sotaque, ruído de fundo e diferentes condições acústicas.

O Whisper disponibiliza seis modelos pré-treinados de diferentes capacidades computacionais, variando desde o *tiny* (39 milhões de parâmetros) até o *large* (1550 milhões de parâmetros), além do modelo *turbo* (809 milhões de parâmetros), uma versão otimizada para inferência mais rápida. Cada modelo apresenta um *trade-off* distinto entre acurácia de transcrição, consumo de memória e latência de processamento. Após análise experimental detalhada de desempenho, descrita no Capítulo 4, foram selecionados dois modelos para integração ao sistema: *small* (244 milhões de parâmetros) e *turbo* (809 milhões de parâmetros). Esta estratégia de disponibilização de múltiplos modelos permite que o usuário final selecione a configuração que melhor equilibra a velocidade de processamento e a precisão da transcrição de acordo com suas necessidades específicas e restrições de recursos computacionais.

O processo de transcrição retorna não apenas o texto transcrito, mas também metadados temporais (*timestamps*) que indicam o momento de início e término de cada segmento transcrito, informação potencialmente útil para navegação e indexação do conteúdo.

3.1.3 Refinamento Textual Assistido por LLM

Após a geração da transcrição bruta, o texto resultante é submetido a um módulo de pós-processamento e refinamento linguístico baseado em LLM. Esta etapa é essencial para corrigir imperfeições inerentes aos sistemas ASR, que frequentemente apresentam três categorias

principais de erros: substituições (palavras transcritas incorretamente), omissões (palavras não detectadas) e inserções (palavras adicionadas erroneamente) (Errattahi; Hannani; Ouahmane, 2018). Além disso, a fala espontânea em reuniões contém características linguísticas inadequadas para documentos formais, como hesitações ("é...", "hum..."), falsos inícios, repetições, marcadores discursivos coloquiais e construções sintáticas fragmentadas.

Para realizar este refinamento, o sistema utiliza o modelo DeepSeek-R1, um LLM de grande capacidade acessado via API em ambiente de nuvem. A interação com o modelo ocorre através da técnica de *prompt engineering*, onde um *prompt* estruturado e detalhado é construído especificando as tarefas de processamento desejadas. Este *prompt* inclui: (i) instruções explícitas sobre os tipos de correções a serem realizadas; (ii) exemplos de transformações esperadas (*few-shot learning*); (iii) especificações sobre o formato de saída; e (iv) restrições quanto à preservação do conteúdo semântico original. A engenharia do *prompt* e sua validação experimental são detalhadas no Capítulo 5.

O modelo LLM processa a transcrição aplicando técnicas de compreensão de linguagem natural para: corrigir erros ortográficos e gramaticais, normalizar o texto removendo disfluências, segmentar logicamente o conteúdo em tópicos coerentes, e reestruturar o discurso oral em formato textual adequado ao registro formal de atas.

3.1.4 Formatação e Padronização Documental

Após a conclusão das etapas de transcrição e refinamento textual, o fluxo de processamento avança para o módulo de formatação e estruturação documental (Item C da Figura 2). Esta fase constitui o estágio final do *pipeline* antes da entrega do artefato ao usuário final.

O objetivo central deste módulo é garantir a conformidade do documento gerado com os padrões documentais institucionais da Universidade Federal Rural do Semi-Árido (UFERSA), assegurando que a ata apresente estrutura organizacional consistente e formatação adequada às normas vigentes. A implementação utiliza a biblioteca python-docx, que fornece uma interface programática para criação e manipulação de documentos no formato DOCX, padrão amplamente adotado para documentos de texto estruturados.

O processo de formatação compreende as seguintes operações: (i) inserção de cabeçalho institucional contendo identificação do órgão, departamento e data da reunião; (ii) aplicação de estilos tipográficos padronizados conforme normas documentais (fontes, tamanhos, espaçamentos e alinhamentos); (iii) numeração automática de parágrafos e tópicos quando aplicável; e (iv)

geração de rodapé com metadados do documento.

O documento resultante é armazenado temporariamente no servidor e disponibilizado ao usuário através de um *endpoint* de *download*, permitindo a obtenção do arquivo DOCX que pode ser posteriormente editado, revisado ou arquivado conforme os procedimentos institucionais estabelecidos.

3.2 Ferramentas e Tecnologias Utilizadas

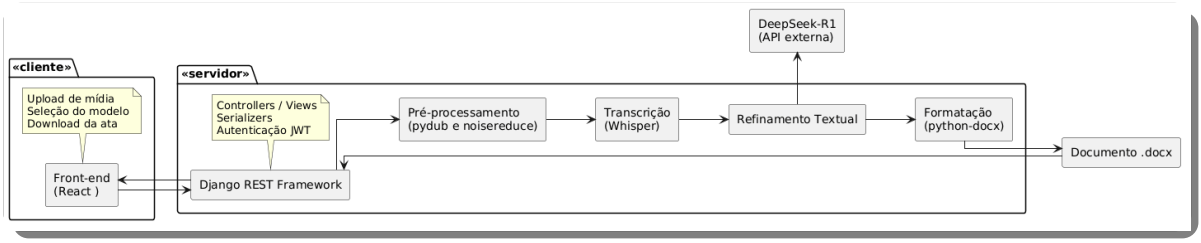
O desenvolvimento do sistema foi realizado utilizando Python (versão 3.12.3), juntamente com o *framework* Django (versão 5.1.1) e sua extensão Django REST framework (versão 3.15.2) responsável pela implementação da API REST e pelo gerenciamento das requisições HTTP (*Hypertext Transfer Protocol*). Para manipulação e processamento de arquivos de áudio, será empregada a biblioteca pydub (versão 0.25.1) e a noisereduce (versão 3.0.3). A geração e formatação de documentos no formato .docx serão realizadas por meio da biblioteca python-docx (versão 1.1.2). Para a transcrição automática de áudio, será utilizado o modelo Whisper (versão 1.1.10), executado localmente. O aprimoramento textual será realizado por meio do modelo DeepSeek-R1 (versão 0528), acessado por meio de requisição à API disponibilizada em ambiente de nuvem. No próximo capítulo, serão detalhados os procedimentos de implementação e integração do sistema. Foi feito o uso do google colab para a execução dos testes.

4 DESENVOLVIMENTO DO SISTEMA

Este capítulo detalha a implementação técnica da solução proposta, descrevendo as etapas de configuração do ambiente de desenvolvimento, a estruturação das rotas de comunicação baseadas em arquitetura REST (*Representational State Transfer*), os mecanismos de registro e autenticação de usuários, bem como a integração dos modelos de Inteligência Artificial e a interface com o *front-end*.

A Figura 3 apresenta uma visão geral dos módulos que compõem o sistema, destacando as tecnologias utilizadas em cada componente.

Figura 3 – Módulos do sistema e tecnologias utilizadas



Fonte: Autoria própria (2026).

Essa organização permitiu desacoplar as responsabilidades do sistema em componentes independentes, facilitando a manutenção do código, a substituição de tecnologias específicas e a incorporação de novas funcionalidades sem impactos significativos nos demais módulos da aplicação.

4.1 Configuração do Ambiente e Inicialização do Projeto

A configuração inicial do ambiente de desenvolvimento fundamentou-se na linguagem Python versão 3.10 ou superior, utilizando o *framework* Django como base para a construção do *back-end*. O projeto foi estabelecido seguindo o padrão de arquitetura modular de aplicações (*apps*) nativo do *framework*, que consiste em módulos independentes e coesos responsáveis por funções específicas dentro do sistema. Nesse contexto, implementou-se a aplicação denominada *transcricao*, responsável por orquestrar o fluxo completo de processamento de áudio e texto, conforme ilustrado na Figura 2.

Para viabilizar o funcionamento da API, o arquivo de configurações globais (*settings.py*) foi ajustado para integrar as dependências necessárias ao projeto. O módulo *transcricao* foi

registrado na lista *INSTALLED_APPS*, permitindo o reconhecimento das *views* e rotas pelo *framework*. Adicionalmente, configurou-se o pacote *django-cors-headers* para gerenciar o mecanismo CORS (*Cross-Origin Resource Sharing*), definindo explicitamente os *endpoints* do *front-end* com permissão de acesso através da diretiva *CORS_ALLOWED_ORIGINS*, garantindo assim a segurança nas requisições entre diferentes domínios e prevenindo vulnerabilidades relacionadas a requisições não autorizadas.

No âmbito da segurança e controle de acesso, adotou-se o padrão JWT (*JSON Web Token*), definido na RFC 7519, como mecanismo de autenticação *stateless*. Essa escolha permite a geração de *tokens* criptograficamente seguros que definem as permissões de acesso dos usuários aos *endpoints* da API, assegurando que apenas requisições devidamente autorizadas sejam processadas. A utilização dessa abordagem contribui para a escalabilidade, caso novas funcionalidades sejam implementadas futuramente, como controle de permissões por perfil ou integração com serviços externos, a autenticação baseada em tokens possibilita a definição de regras de acesso específicas para cada tipo de usuário, mantendo a combatibilidade com a arquitetura atual. A implementação foi realizada através da biblioteca *djangorestframework-simplejwt*, que fornece um par de *tokens*: o *access token* (com tempo de expiração reduzido) e o *refresh token* (com validade estendida). Por fim, para garantir as boas práticas de segurança da informação, utilizou-se um arquivo de variáveis de ambiente (*.env*) para o armazenamento seguro de informações sensíveis, como chaves de API, *secret keys* e credenciais de acesso a serviços de nuvem, evitando a exposição dessas informações no controle de versão.

4.2 Implementação das views

As *views* do Django são classes ou funções definidas no arquivo *views.py*, que representam o núcleo lógico do sistema, responsáveis por receber as requisições HTTP e retornar respostas apropriadas ao cliente. Nas *views* ocorrem todas as etapas descritas no fluxo de processamento apresentado na Figura 2, incluindo validação de dados, invocação de serviços externos e formatação de respostas. Neste trabalho, utilizou-se o padrão de classe *APIView* fornecido pelo DRF, que oferece uma abstração de alto nível e adequação superior na construção de APIs RESTful, permitindo a definição explícita de métodos HTTP (GET, POST, PUT, DELETE) e facilitando a implementação de conceitos REST como *stateless communication* e representação de recursos.

A *view* principal é a *TranscribeView*, que implementa o método *post* para processar

requisições de transcrição. Quando uma requisição é enviada pelo *front-end*, o sistema de roteamento do Django identifica a *view* correspondente através da configuração definida no arquivo *urls.py* da aplicação. Uma vez identificada a rota, a requisição é encaminhada à *view* correspondente, que executa o processamento do áudio e retorna a transcrição como resposta. Adicionalmente, o método *delete* foi implementado para possibilitar o cancelamento de operações de transcrição em andamento, permitindo ao usuário interromper o processamento quando necessário. Este método retorna uma resposta estruturada através do objeto *Response* do DRF, incluindo o código de estado HTTP 200 (*OK*) e os dados formatados em JSON (*JavaScript Object Notation*), seguindo o padrão de comunicação REST.

4.2.1 Camada de serialização

Para assegurar a integridade e a conformidade dos dados recebidos, uma camada de serialização foi implementada utilizando o componente *Serializer* do DRF. Esta camada desempenha um papel duplo fundamental: realiza a conversão de dados complexos para tipos nativos do Python (*marshalling*) e vice-versa (*unmarshalling*), além de verificar se os dados recebidos no *body* da requisição correspondem aos tipos e formatos esperados, conforme definidos em cada classe de serialização. Essa validação é executada no recebimento da requisição, antes que os dados alcancem as camadas de lógica de negócio do sistema, impedindo que dados inválidos ou malformados penetrem nas camadas subsequentes. Essa abordagem otimiza o consumo de recursos do servidor e reduz o acoplamento entre as camadas de apresentação e de negócio, seguindo o princípio de separação de responsabilidades (*Separation of Concerns*).

A *view TranscribeView* possui um *serializer* associado denominado *TranscriptionSerializer*, que garante que toda requisição de transcrição contenha um arquivo válido de mídia, restringe o campo *model* a um conjunto finito de opções pré-definidas (*whitelist approach*) e permite informar o departamento de forma opcional para fins de formatação do documento final.

Código-fonte 1 – Serializer da View TranscribeView

```
1 class TranscriptionSerializer(serializers.Serializer):
2     media = serializers.FileField(required=True)
3     model = serializers.ChoiceField(
4         choices=["small","turbo"],
5         default="small"
6     )
7     departamento = serializers.CharField(required=False,
        allow_blank=True)
```

A linha 1 declara a classe `TranscriptionSerializer` que herda de `serializers.Serializer`, estabelecendo assim um *serializer* personalizado para a *view*. Esta classe define três campos principais:

- **media**: Definido como `serializers.FileField(required=True)`, este campo é obrigatório e deve conter um arquivo de mídia (áudio ou vídeo). Caso o cliente não forneça este campo, a validação falhará automaticamente, retornando um erro HTTP 400 (*Bad Request*).
- **model**: Especificado como `serializers.ChoiceField(choices=["small","turbo"], default="small")`, este campo aceita exclusivamente dois valores pré-definidos para o modelo de transcrição: *small* ou *turbo*. Se o cliente não especificar este campo, o valor padrão *small* será utilizado. Esta restrição previne a utilização de modelos não suportados pelo sistema.
- **departamento**: Declarado como `serializers.CharField(required=False, allow_blank=True)`, este é um campo de texto opcional utilizado para especificar o departamento que deverá ser incluído no cabeçalho do documento de ata gerado.

Código-fonte 2 – Lógica de validação e extração de dados do Serializer

```
serializer = TranscriptionSerializer(data=request.data)
serializer.is_valid(raise_exception=True)

audio_file = serializer.validated_data["media"]
selected_model = serializer.validated_data["model"]
departamento = serializer.validated_data.get("departamento"
    )
```

O Código 2 ilustra a aplicação prática da lógica de validação. Na primeira linha, instancia-se o *TranscriptionSerializer* passando os dados da requisição (`request.data`). Em seguida, invoca-se o método `is_valid(raise_exception=True)`, que executa todas as validações definidas no *serializer*. O parâmetro `raise_exception=True` instrui o DRF a lançar automaticamente uma exceção *ValidationError* em caso de falha na validação, o que resulta no retorno imediato de uma resposta HTTP 400 (*Bad Request*) ao cliente, contendo os detalhes dos erros de validação. Após a validação bem-sucedida, os dados limpos e validados ficam disponíveis no atributo `validated_data`, de onde são extraídos os valores para processamento posterior.

4.3 Arquitetura de roteamento e endpoints

O sistema de roteamento foi projetado seguindo os princípios da arquitetura REST, garantindo que cada *endpoint* representa um recurso específico ou uma ação claramente definida sobre esse recurso. A configuração das URLs é modularizada e dividida em dois níveis hierárquicos: o nível global (definido no arquivo *urls.py* do projeto) e o nível de aplicação (definido no arquivo *urls.py* de cada *app*). Esta separação proporciona melhor organização do código e facilita a manutenibilidade do sistema, seguindo o princípio de modularização de software.

Os quadros a seguir detalham os *endpoints* da API e suas respectivas funcionalidades. Todos os caminhos listados têm como prefixo base a URL `/api/v1/`, definida no nível global de configuração. A inclusão da versão (*v1*) na URL segue as boas práticas de versionamento de APIs, permitindo futura evolução do sistema sem quebrar compatibilidade com clientes existentes.

Quadro 1 – Endpoints da API para Autenticação e Gestão de Usuários

Método	URI	**Descrição**
POST	/auth/register/	Realiza o cadastro de novos usuários no sistema.
POST	/auth/login/	Autentica o usuário e retorna os tokens JWT.
POST	/auth/token/refresh/	Renova o token de acesso expirado.

Fonte: Autoria própria (2026).

Quadro 2 – Endpoints da API para o Módulo de Transcrição

Método	URI	**Descrição**
POST	/transcriptions/	Envia o vídeo para transcrição e aprimoramento.
DELETE	/transcriptions/cancel/	Interrompe uma operação de transcrição em andamento.
GET	/transcriptions/download/<filename>/	Realiza o download do documento da ata gerada.
POST	/transcriptions/summary/	Gera o resumo ou ata formatada com base na transcrição bruta.

Fonte: Autoria própria (2026).

Enquanto os *endpoints* de autenticação são públicos para permitir o registro e *login* de usuários, as rotas do módulo de transcrição são protegidas pela classe de permissão *IsAuthenticated* do DRF. Esta proteção garante que apenas usuários autenticados, com *tokens* JWT válidos no cabeçalho HTTP (*Authorization: Bearer* <*token*>), possam acessar os recursos de transcrição, implementando assim um controle de acesso baseado em autenticação (*authentication-based access control*).

4.4 Interface do sistema e documento final

Para ilustrar o funcionamento prático do sistema desenvolvido, apresenta-se nesta seção o ambiente de interface com o usuário, bem como o documento final gerado pelo processo descrito na Figura 2.

4.4.1 Interface do usuário

Na Figura 4 é ilustrada a tela inicial do sistema, onde é necessária a autenticação do usuário para acessar as funcionalidades.

Figura 4 – Interface do sistema web: Tela de Login

Fonte: Autoria própria (2026).

Se o usuário ainda não for cadastrado, ele pode acessar a tela de cadastro, ilustrada na Figura 5, onde é necessário informar um nome de usuário, matrícula, email e senha para criar uma conta no sistema.

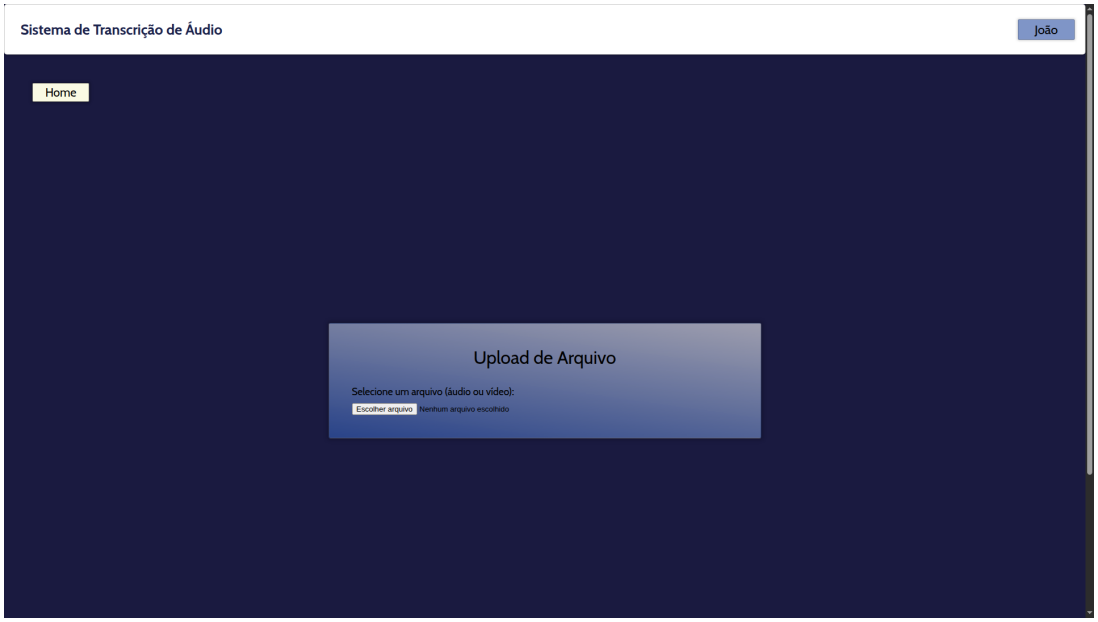
Figura 5 – Interface do sistema web: Tela de Cadastro

Fonte: Autoria própria (2026).

Após a autenticação, o usuário é redirecionado para a tela inicial, ilustrada na Figura 6,

onde pode acessar as funcionalidades do sistema, como o upload de arquivos de áudio ou vídeo para transcrição e geração da ata.

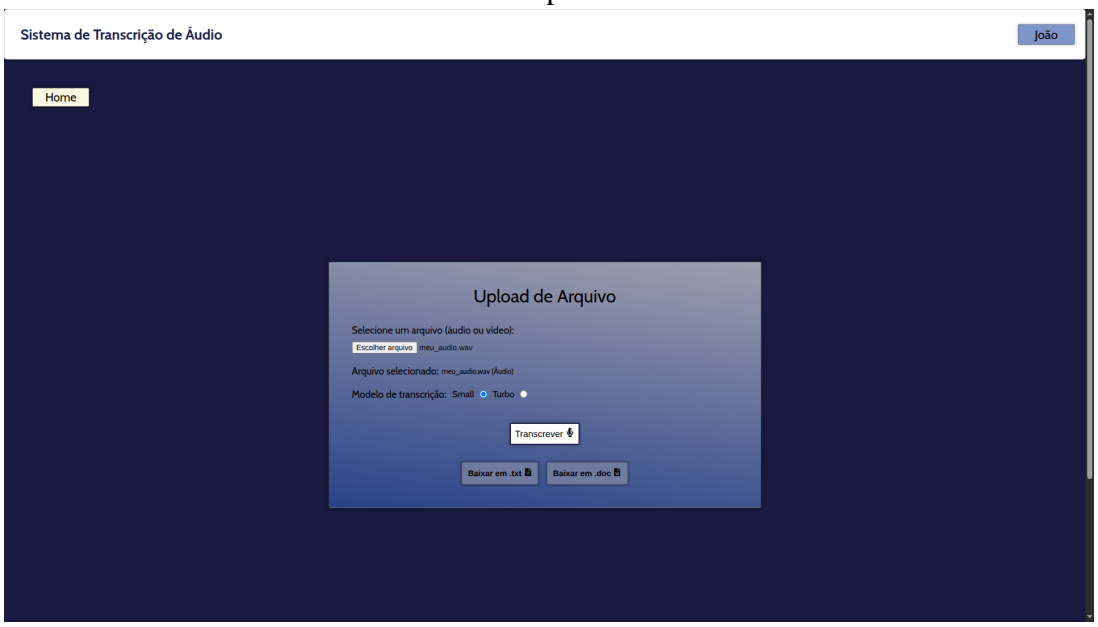
Figura 6 – Interface do sistema web: Tela de Inicial



Fonte: Autoria própria (2026).

Na Figura 7 é ilustrada a tela de *upload*, onde o usuário pode selecionar um arquivo de áudio ou vídeo, escolher o modelo de transcrição (*small* ou *turbo*). Após o envio, o sistema processa o arquivo e gera a ata correspondente. O sistema disponibiliza dois formatos de saída, um arquivo de texto simples e um documento formatado seguindo o modelo de ata padrão, ambos prontos para *download*.

Figura 7 – Interface do sistema web: Tela de Upload



Fonte: Autoria própria (2026).

4.4.2 Documento Final

O Apêndice A (p. 62) ilustra o documento final gerado automaticamente pelo sistema e pronto para revisão humana. O documento segue o modelo de ata institucional apresentado no Anexo A (p. 66).

5 EXPERIMENTOS

Este capítulo apresenta os experimentos realizados para validar o sistema de transcrição e geração de atas. Os testes foram estruturados para avaliar o desempenho das variantes do modelo Whisper e a capacidade de refinamento do modelo DeepSeek-R1 sob condições de uso.

5.1 Configuração Experimental

Os experimentos foram conduzidos no *Google Colab*, que oferece uma capacidade de processamento em nuvem com GPU NVIDIA Tesla T4, com 16 GB de memória GDDR6 e 8.1 TFLOPS de processamento. Como base de dados, foi utilizado trechos de vídeos públicos do canal oficial da UFERSA no YouTube.

Para avaliar o desempenho do sistema em cada estágio, utilizaram-se as seguintes métricas:

- **WER:** métrica padrão para sistemas ASR que calcula a taxa de erro de palavras, comparando a transcrição gerada com uma referência (Roy, 2021).
- **Métricas de Refinamento Textual:** para avaliar a qualidade do refinamento textual feito pelo modelo DeepSeek-R1, empregaram-se:
  - **ROUGE-L:** Encontra a maior sequência de palavras que aparecem na mesma ordem tanto no texto candidato quanto no texto de referência (Nguyen *et al.*, 2024), permitindo medir a preservação do conteúdo;
  - **SBERT:** Utilizada para buscar a similaridade semântica entre os textos (Reimers; Gurevych, 2019), garantindo que o sentido da reunião foi preservado;
  - **Perplexity:** Quantifica a incerteza do modelo (Cooper; Scholak, 2024). Utilizado para avaliar quanto o modelo ficou confuso com o prompt de refinamento.

5.2 Estudo de Caso

Para validar o sistema, utilizou-se como estudo de caso a trechos de gravações de reuniões administrativa, escolhidos por contar com múltiplos participantes, diversidade acústica e condições de ruidos típicos de videoconferência. O conjunto de dados totaliza 58 minutos e 55 segundos de material.

5.2.1 Descrição do Cenário

O cenário consiste em reuniões virtuais, realizadas via plataforma de comunicação remota, e presenciais, ambas transmitidas via YouTube. As gravações apresentam duração variando de 3 a 14 minutos, com número de participantes distinto entre os vídeos, com diferentes qualidades de áudio, ocorrências de falas sobrepostas, interrupções e oscilações eventuais na transmissão, caracterizando condições reais de uso do sistema.

5.2.2 Procedimento Experimental

O procedimento experimental para avaliação dos modelos Whisper foi estruturado em etapas sequenciais. Inicialmente, foram definidos os sete vídeos a serem processados, cada um acompanhado de sua respectiva transcrição de referência elaborada manualmente.

O pipeline de avaliação seguiu o seguinte fluxo, aplicado a cada modelo da família Whisper: (i) Carregamento do modelo na GPU; (ii) transcrição dos sete vídeos; (iii) registro do tempo de processamento; (iv) cálculo do WER em relação à transcrição de referência; e (v) salvamento incremental dos resultados.

Essa etapa foi repetida 10 vezes para cada modelo, garantindo a consistência dos resultados e permitindo a análise estatística das métricas obtidas. A média e o desvio padrão do WER e do tempo de processamento foram calculados para cada modelo, fornecendo uma visão abrangente do desempenho de cada variante.

5.2.3 Análise de Desempenho dos Modelos de Transcrição

Esta seção apresenta e discute os resultados obtidos na avaliação comparativa dos modelos Whisper.

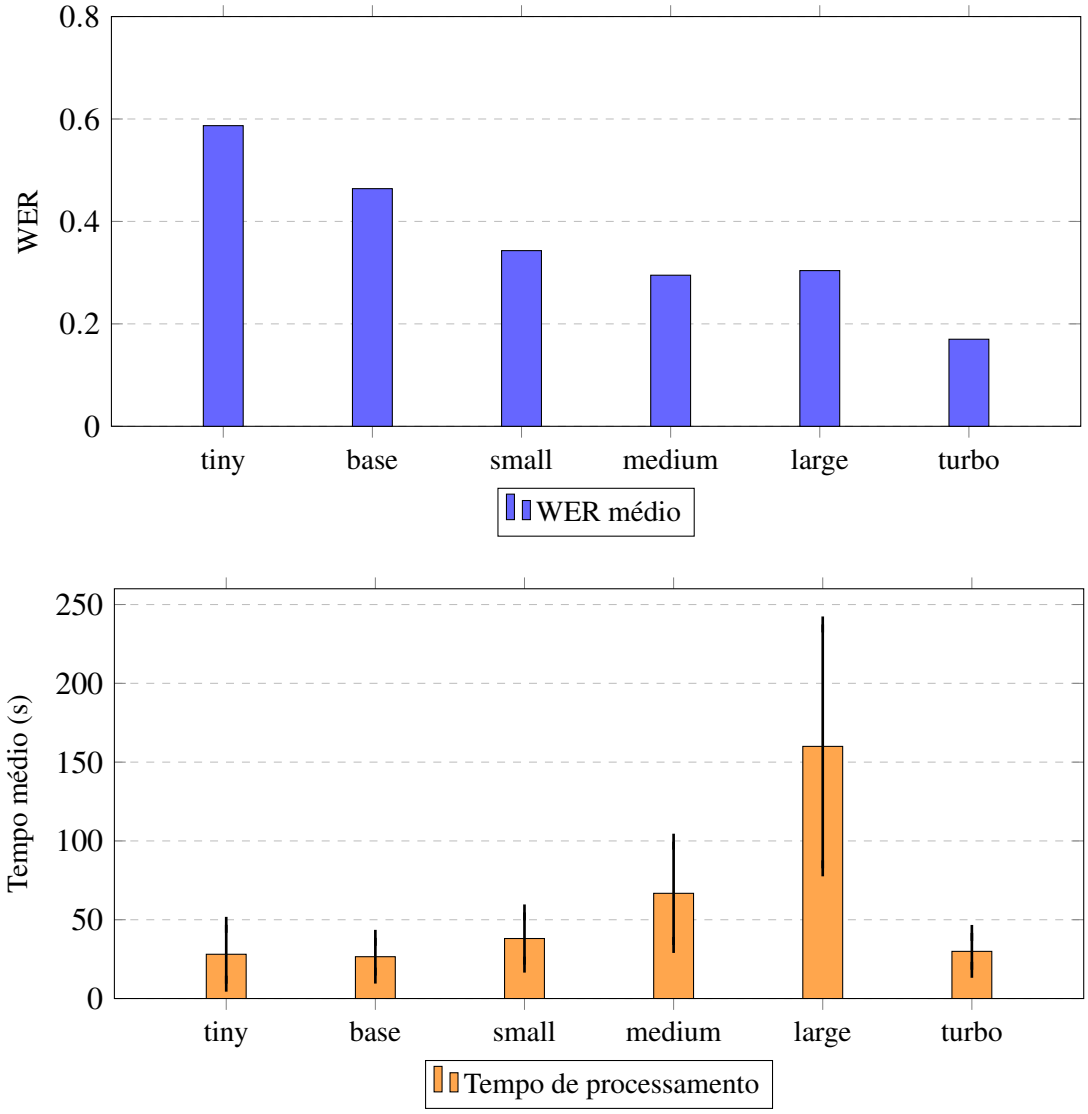
A Tabela 1 apresenta os resultados quantitativos obtidos na avaliação dos seis modelos Whisper. A Figura 8 ilustra graficamente a relação entre o WER e o tempo de processamento para cada modelo avaliado.

Tabela 1 – Resultado dos modelos de transcrição considerando WER e duração.

Modelo	**Média WER**	**Tempo Médio (s)**
tiny	0,587	$28{,}08 \pm 18{,}66$
base	0,464	$26{,}53 \pm 11{,}98$
small	0,343	$38{,}07 \pm 16{,}56$
medium	0,295	$66{,}75 \pm 32{,}77$
large	0,304	$159{,}99 \pm 77{,}37$
turbo	0,170	$29{,}92 \pm 11{,}69$

Fonte: Autoria própria (2026).

Figura 8 – Comparação de desempenho dos modelos Whisper: WER e tempo de processamento.



Fonte: Autoria própria (2026).

Ao comparar os resultados de cada modelo, os de menor capacidade, *tiny* e *base*, registra-

ram os maiores índices de erro, 0,587 e 0,464, sugerindo que podem ter dificuldades em capturar detalhes específicos, embora tenham apresentado as menores médias de tempo de processamento, $28{,}08 \pm 18{,}66$s e $26{,}53 \pm 11{,}98$s por vídeo.

O modelo *small* ocupou uma posição intermediária, com WER de 0,343 e tempo de $38{,}07 \pm 16{,}56$s por vídeo, mostrando um equilíbrio entre precisão e tempo.

Os modelos *medium* e *large*, apresentaram WER de 0,295 e 0,304, respectivamente, com os tempos de processamento significativamente maiores, $66{,}75 \pm 32{,}56$s e $159{,}99 \pm 77{,}37$s por vídeo. Este comportamento evidencia que, para o contexto de transcrição de áudio, o aumento do tamanho do modelo não necessariamente se traduz em melhorias proporcionais na precisão. A comparação entre os modelos evidencia que o aumento no número de parâmetros do modelo *large* em relação ao *medium* não resultou em melhoria significativa na precisão.

O modelo *turbo* obteve o melhor desempenho em termos de precisão, com WER de 0,170, indicando que aproximadamente 17% das palavras transcritas apresentaram algum tipo de erro. O tempo médio de transcrição por vídeo foi de $29{,}92 \pm 11{,}69$ s, o que representa um desempenho competitivo em relação aos demais modelos.

Os modelos *tiny* e *base*, embora apresentem tempos de processamento atrativos, demonstraram limitações significativas em termos de precisão. Para aplicações onde a fidelidade da transcrição é crítica, como a geração de atas de reuniões, estes modelos podem não ser adequados. O modelo *small* apresentou um desempenho intermediário que pode ser justificável em cenários com restrições de recursos computacionais, desde que se aceite uma taxa de erro moderada.

O modelo *turbo* destaca-se como a opção mais equilibrada, combinando a menor taxa de erro com tempo de processamento moderado. A superioridade do modelo *turbo* em ambas as métricas (precisão e velocidade) o torna a escolha preferencial para a implementação do sistema proposto neste trabalho, juntamente com o modelo *small*, dando ao usuário a opção de escolha entre os dois modelos.

Vale destacar que os valores de desvio padrão do tempo de processamento apresentaram-se relativamente elevados em todos os modelos, como observado nos modelos *medium* ($66{,}75 \pm 32{,}77$s) e *large* ($159{,}99 \pm 77{,}37$s). Esse comportamento é esperado, uma vez que o conjunto de vídeos utilizados nos experimentos possui durações variadas, fazendo com que vídeos mais longos demandem naturalmente um tempo de processamento maior, o que aumenta a dispersão dos valores em torno da média.

5.2.4 Refinamento com LLM

Esta seção descreve a metodologia adotada para o refinamento utilizando LLMs, bem como os experimentos conduzidos para avaliar sua eficácia.

Um *prompt* é uma entrada baseada em texto que é fornecida a um modelo de linguagem para orientar sua saída (Marvin *et al.*, 2024). A configuração dos prompts constitui um aspecto fundamental para o sucesso do refinamento textual, Marvin *et al.* (2024) afirma que a eficácia do modelo é fortemente influenciada pela qualidade do *prompt* usado para guiá-lo.

Foram exploradas duas estratégias de engenharia de *prompts*, a *few-shot*, que consiste em um prompt elaborado para que um modelo de linguagem execute uma nova tarefa (Ahmed *et al.*, 2023), e *Engenharia de prompts automáticos(Automatic Prompt Engineering* - APE), implica no uso de um *meta-prompt* que guia a LLM a inspecionar o *prompt* atual, fornecer *feedback* e, em seguida, gerar um *prompt* atualizado (Ye *et al.*, 2024).

Para a construção de *prompts* eficazes, é necessário seguir as seguintes etapas: (i) Definir claramente o objetivo, (ii) Fornecer exemplos específicos de como o texto deve ser melhorado, (iii) Incluir instruções sobre o formato e utilizar uma linguagem clara e concisa para evitar ambiguidades, (iv) Testar e iterar os *prompts* com base no *feedback* do modelo para otimizar os resultados (Marvin *et al.*, 2024).

Quadro 3 – Estrutura e conteúdo do *prompt* de aprimoramento.

Prompt	**(i) Objetivo**	**(ii) Exemplos**	**(iii) Formato**
"Você é um assistente que corrige e aprimora transcrições de áudio. Recebe a transcrição e retorna ela aprimorada seguindo os seguintes pontos: Preservar sentido. Não invente informações que não estejam no texto. Corrija ortografia e gramática sem alterar o sentido. Organize em parágrafos apropriados. A resposta deve ser apenas a transcrição aprimorada. em vez de falar '**Transcrição Aprimorada:**' ou similar, não adicione nada."	Você é um assistente que corrige e aprimora transcrições de áudio.	Preservar sentido. Não invente informações que não estejam no texto; Corrija ortografia e gramática sem alterar o sentido;	Organize em parágrafos apropriados; A resposta deve ser apenas a transcrição aprimorada, sem '**Transcrição Aprimorada:**' ou similar, não adicione nada.
"Crie um *prompt* conciso para aprimorar transcrições de áudio. Posteriormente irei enviar uma transcrição e preciso de um *prompt* para aprimorar a transcrição. O *prompt* deve instruir a correção ortográfica e gramatical, organização em parágrafos, preservação do sentido original, e a saída deve ser apenas a transcrição aprimorada, sem texto adicional como 'Transcrição Aprimorada:'. Você Deve ter como saída apenas o *prompt*, sem '***Prompt*:**'"	Crie um *prompt* conciso para aprimorar transcrições de áudio;	O prompt deve instruir a correção ortográfica e gramatical, organização em parágrafos, preservação do sentido original;	a saída deve ser apenas a transcrição aprimorada, sem texto adicional como "Transcrição Aprimorada:"; Você Deve ter como saída apenas o prompt, sem "***Prompt*:**"

Fonte: Autoria própria (2026).

O Quadro 3 define os *prompts* utilizado nos teste e cada etapa da sua construção. O primeiro *prompt* consistiu na elaboração manual baseado na técnica *few-shot*; o segundo utilizou o método de *automatic prompt engineering*, onde orienta a LLM a gerar sua própria instrução de refinamento.

A definição do papel como assistente e as restrições de formato são fundamentais para mitigar problemas comuns de redudância e perda de coesão. Essas definições foram quantificadas por meio das métricas ROUGE-L, SBERT e Perplexity, com o mesmo número de amostras do teste anterior, utilizando suas transcrições de referência e as geradas pelo Whisper com o modelo turbo.

O processo de refinamento é feito com a submissão dos *prompts* junto à transcrição

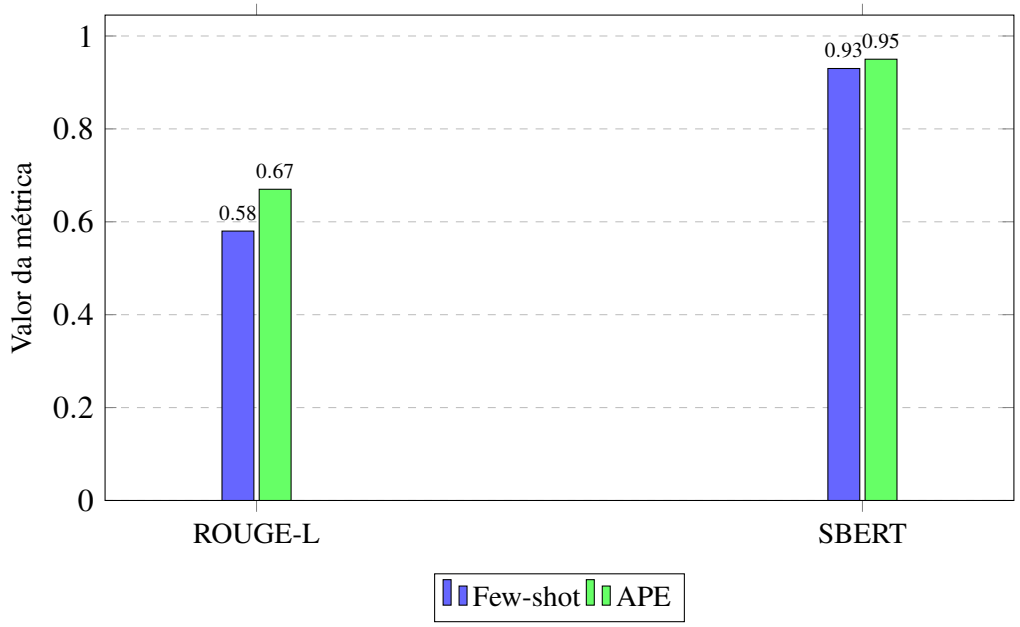
bruta à LLM, executando-se de forma independente para cada uma das duas estratégias de *prompting*. Os textos resultantes de ambas as abordagens foram comparados com as transcrições de referência.

Tabela 2 – Resultados comparativos das métricas ROUGE-L, SBERT e Perplexity.

Vídeo	Estratégia de Prompt	ROUGE-L	SBERT	PPL
1	Few-shot	0,4331	0,0927	83.1159
	APE	0,2169	0,9169	82.0795
2	Few-shot	0,6887	0,9263	36.6548
	APE	0,7778	0,9313	41.0778
3	Few-shot	0,9296	0,9516	45.1597
	APE	0,9781	0,9941	49.3394
4	Few-shot	0,2234	0,8134	50.7816
	APE	0,4778	0,9273	57.7451
5	Few-shot	0,4912	0,973	67.2236
	APE	0,6781	0,9673	61.676
6	Few-shot	0,8125	0,9565	61.4375
	APE	0,9808	0,9878	52.4411
7	Few-shot	0,5353	0,9771	59.1019
	APE	0,6355	0,9821	54.8900

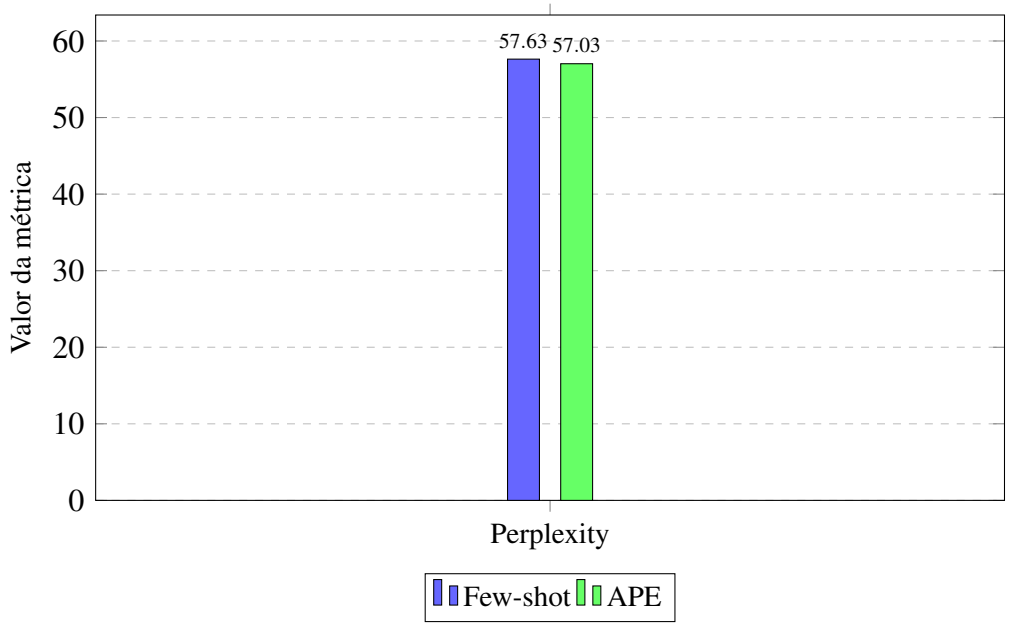
Fonte: Autoria própria (2026).

Figura 9 – Comparação das estratégias de prompting: médias das métricas ROUGE-L e SBERT



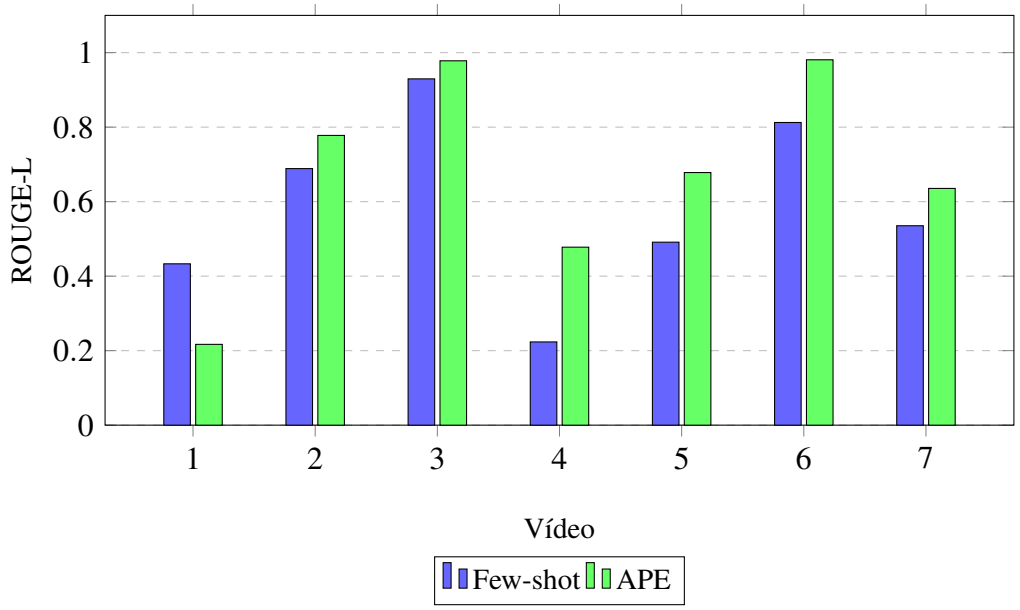
Fonte: Autoria própria (2026).

Figura 10 – Comparação das estratégias de prompting: média da métrica Perplexity.



Fonte: Autoria própria (2026).

Figura 11 – Comparação detalhada por vídeo: ROUGE-L para estratégias Few-shot e APE.



Fonte: Autoria própria (2026).

Os resultados quantitativos da etapa de refinamento, apresentados na Tabela 2, revelam o comparativo entre as duas estratégias de *prompting* avaliadas. Observou-se que os resultados do *prompt* gerado pela LLM apresentaram médias superiores em ambas as métricas. As Figuras 9, 10 e 11 apresentam uma visualização comparativa do desempenho das estratégias.

Em relação ao ROUGE-L, a estratégia APE obteve média de 0,67 contra 0,58 da abordagem *few-shot*, indicando uma maior preservação da estrutura sequencial do texto de referência. No SBERT, a média do APE foi de 0,95, ligeiramente superior à média de 0,93 obtida pelo *few-shot*, o que sugere que o modelo manteve o sentido semântico das frases de forma mais consistente.

Quanto à PPL, a diferença foi pequena, 57,63 para o APE e 57,03 do *few-shot*. Essa proximidade entre os valores indica que, independentemente da estratégia de *prompting* adotada, o modelo apresentou níveis similares de incerteza na geração do texto, sugerindo que as construções sintáticas do domínio avaliado ainda representam um desafio considerável para o modelo.

De maneira geral, os resultados indicam que a abordagem APE foi capaz de produzir *prompts* mais eficazes do que a estratégia *few-shot* construída manualmente, com ganhos consistentes nas métricas de similaridade textual. Contudo, a ausência de diferença expressiva na Perplexidade sugere que o principal fator limitante não está na estratégia de *prompting*, mas sim na capacidade do modelo em lidar com a complexidade linguística. Esses achados reforçam o potencial da otimização automática de *prompts* como alternativa viável à engenharia manual, ao mesmo tempo em que evidenciam a necessidade de investigações futuras voltadas à redução da incerteza do modelo na geração de texto especializado.

5.3 Discussão dos Resultados

Os experimentos conduzidos neste capítulo permitem avaliar o desempenho do sistema proposto em suas duas etapas principais: a transcrição automática de fala e o refinamento textual por meio de LLM.

Na etapa de transcrição, os resultados evidenciaram que o modelo *turbo* se destacou como a opção mais equilibrada entre precisão e eficiência computacional, registrando o menor WER médio (0,170) com tempo de processamento equivalente ao dos modelos mais leves. Em contraste, o modelo *large*, apesar de possuir maior número de parâmetros, não apresentou ganhos proporcionais em relação ao *medium*, o que indica que o aumento da capacidade do modelo não implica necessariamente em melhoria de desempenho no contexto avaliado. O modelo *small*, por sua vez, mostrou-se uma alternativa viável para cenários com restrições computacionais, dado o equilíbrio entre precisão moderada e tempo reduzido.

Na etapa de refinamento, a estratégia APE demonstrou vantagem sobre a abordagem *few-*

shot nas métricas de similaridade textual, com ganhos em ROUGE-L e SBERT, indicando maior preservação da estrutura sequencial e do sentido semântico do texto. A proximidade dos valores de PPL entre as duas estratégias sugere que o modelo enfrenta desafios no domínio linguístico das reuniões administrativas, independentemente da abordagem de *prompting* adotada.

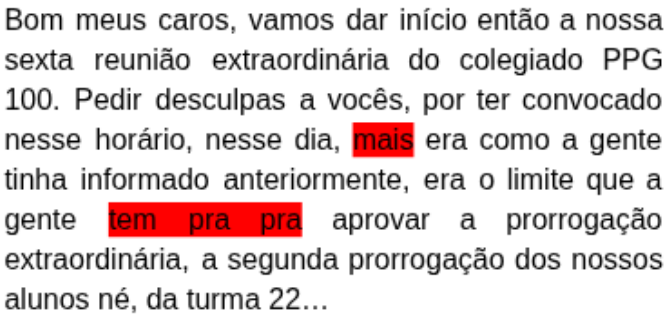
A Figura 12 ilustra, de forma qualitativa, as três etapas do processamento textual aplicado a um trecho do Vídeo 1: o texto de referência, a transcrição gerada pelo *Whisper* e o texto resultante do refinamento pela LLM, com os erros e correções destacados.

No texto de referência, observa-se a presença de um erro gramatical o uso de “mais” no lugar de “mas”, ocorrência comum em transcrições manuais. Na saída do *Whisper*, são visíveis erros de transcrição que chegam a comprometer informações do conteúdo original, como a substituição de “era o li- mite que a gente” por “não era o limite que a gente”. Após o refinamento pela LLM, esses erros foram corrigidos e a escrita foi reorganizada de forma mais coesa e formal, demonstrando a eficácia da etapa de refinamento na melhoria da qualidade textual das transcrições.

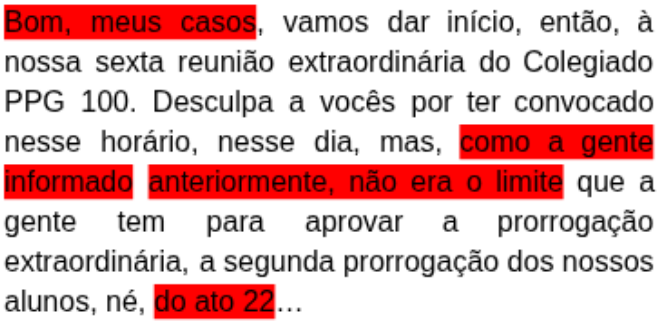
Os resultados foram obtidos a partir de sete vídeos do canal oficial de transmissão da UFERSA no YouTube, totalizando 58 minutos e 55 segundos de material, conferindo ao conjunto de dados com características próximas ao contexto de uso pretendido do sistema. O domínio específico das reuniões administrativas influencia diretamente os resultados obtidos, uma vez que a presença de termos institucionais, siglas e estruturas de fala características desse contexto impõem desafios adicionais ao modelo de transcrição, o que pode explicar parte dos erros registrados pelo WER, especialmente nos modelos de menor capacidade. Da mesma forma, o refinamento textual é impactado pelo nível de formalidade e pela terminologia própria do domínio, fatores que contribuem para os valores de PPL observados.

No que diz respeito à capacidade de generalização da abordagem proposta, é importante considerar que tanto o modelo *Whisper* quanto o *DeepSeek-R1* foram utilizados sem ajuste fino específico para o domínio, o que indica que os resultados obtidos refletem o desempenho de modelos de propósito geral aplicados a um contexto especializado. Essa característica sugere que a abordagem tem potencial de ser transferida para outros domínios, tais como reuniões de conselhos municipais, comitês acadêmicos ou ambientes corporativos, desde que as condições acústicas sejam comparáveis. No entanto, domínios com vocabulário muito específico ou alto grau de informalidade podem demandar adaptações no *prompt* de refinamento ou, em casos mais exigentes, o ajuste fino do modelo de transcrição.

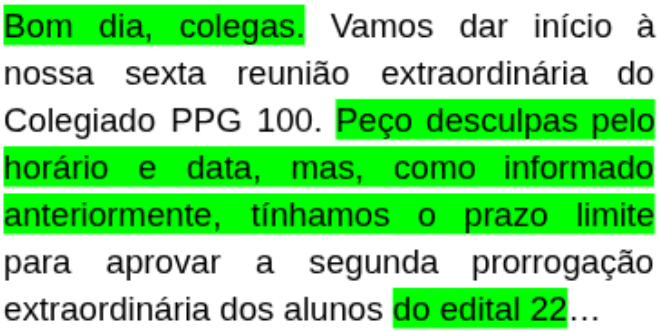
Figura 12 – Comparação qualitativa entre as etapas de processamento (Vídeo 1).



(a) Texto de referência.



(b) Whisper(*turbo*).



(c) Resultado do refinamento.

Fonte: Autoria própria (2026).

6 CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo o desenvolvimento de um sistema integrado para transcrição automática de reuniões e aprimoramento textual voltado à geração automatizada de atas de reuniões em língua portuguesa do Brasil. O problema de pesquisa que motivou a investigação estava na ausência de soluções que, ao mesmo tempo, transcrevessem de forma eficiente o conteúdo de reuniões institucionais e o transformassem em documentos formais padronizados. Foi realizado um levantamento de ferramentas disponíveis no mercado voltadas à transcrição automática e geração de documentos, para fins de geração automatica de documentos não foi identificada nenhuma solução, ja para transcrição automática, foram identificadas algumas soluções no mercado, contudo, a comparação direta não foi viável por duas razões principais, a maioria das soluções identificadas adota modelos proprietários ou não divulga publicamente qual tecnologia de ASR é empregada, e parte delas opera sob planos pagos, inviabilizando o acesso para fins de avaliação. Diante disso, a avaliação do sistema proposto foi conduzida de forma isolada, com base em métricas objetivas e transcrições de referência elaboradas manualmente, o que preserva a validade dos resultados dentro do escopo definido.

Conlui-se que o objetivo geral foi alcançado, o sistema demonstrou capacidade de processar arquivos de mídia em diferentes formatos, transcrever seu conteúdo e gerar documentos estruturados no formato DOCX, aderentes ao padão institucional da UFERSA.

Em relação aos objetivos especificos, todos foram cumpridos no desenvolvimento. O módulo de transcrição automática foi implementado utilizando o modelo Whisper executado localmente, tendo o modelo *turbo* se destacado como a escolha mais equilibrada entre precisão e eficiência, com WER médio de 0,170 e tempo de processamento médio de 30,92 segundos por vídeo. O modelo DeeepSeek-R1 foi integrado via API em nuvem para o refinamento textual, e sua aplicação demonstrou ganhos mensuráveis na qualidade das transcrições, com a estrategia de APE superando a abordagem *few-shot* construída manualmente, alcançando ROUGE-L médio de 0,67 e SBERT médio de 0,95. A API REST foi desenvolvida com êxito em Python e Django REST framework, viabilizando a comunicação entre o *back-end* e o *front-end*. Por fim, o desempenho do sistema foi avaliado pelas metricas especificadas.

Do ponto de vista das contribuições, o trabalho oferece três dimensões complementares. No plano técnico, demonstra que a combinação de um modelo ASR de código aberto com um LLM orientado por *prompt engineering* constitui uma arquitetura viável e eficaz para a automação de documentos formais em português brasileiro. No plano prático, um sistema que entrega ao

usuário final uma ata gerada de forma automatizada, reduzindo significativamente o esforço operacional envolvido no registro de reuniões institucionais. A comparação sistemática entre estratégias de *prompting* e o emprego de métricas complementares de avaliação do refinamento textual fornecem subsídios metodológicos no plano científico para investigações futuras na área de pós-processamento de transcrições com LLM.

6.1 Limitações e Trabalhos Futuros

Apesar dos resultados positivos, o sistema apresenta limitações que devem ser consideradas. A base de avaliação, composta por sete vídeos totalizando aproximadamente 59 minutos, representa uma amostra restrita, o que limita a generalização dos resultados para outros contextos, perfis acústicos e variações linguísticas. Adicionalmente, o sistema não realiza a identificação e separação automática de falantes, o que representa uma lacuna para a geração de atas que exigem a atribuição de falas a participantes identificados nominalmente. A dependência de uma API externa para o modelo LLM também impõe variáveis de latência, custo operacional e disponibilidade que podem impactar a adoção do sistema em ambientes com restrições de conectividade ou orçamento.

Quanto à confiabilidade dos modelos, cabe uma avaliação mais cautelosa. O modelo Whisper, embora validado na literatura, foi originalmente treinado predominância de dados na língua inglesa, o que implica limitações para o português brasileiro refletidas nos valores de WER obtidos. No estágio de refinamento textual, os valores da metrica PPL registrados indicam que o modelo LLM enfrenta dificuldades na geração do texto dentro do domínio do trabalho, o que representa um fator limitante para a confiabilidade do documento final em contextos que exigem precisão. Por essa razão, recomenda-se que o conteúdo gerado pelo sistema seja sempre submetido à revisão humana antes de sua utilização em caráter oficial. Reconhecer essas limitações é essencial para situar adequadamente o alcance do sistema proposto e orientar os esforços de aprimoramento em trabalhos futuros.

Compreendidas essas restrições, os trabalhos futuros descritos a seguir visam suprir as lacunas identificadas e ampliar a aplicabilidade do sistema em cenários mais diversos. Recomenda-se a incorporação de diarização de falantes, a fim de enriquecer o documento final com a identificação dos participantes. Sugere-se ainda a expansão da base de avaliação com gravações de diferentes contextos institucionais e condições acústicas, bem como a investigação de modelos LLM executados localmente como alternativa à dependência de APIs externas, acompanhada de

uma análise comparativa da PPL entre os modelos, a fim de identificar aqueles que apresentam menor incerteza na geração de texto em domínio especializado. Por fim, estudos voltados à avaliação da aceitação e usabilidade do sistema por parte dos usuários finais contribuiriam para validar sua adequação às necessidades reais do ambiente institucional.

6.2 DECLARAÇÃO DE USO DE INTELIGÊNCIA ARTIFICIAL

Em conformidade com as diretrizes de integridade científica e boas práticas acadêmicas recomendadas pela CAPES, os autores declaram que utilizaram ferramentas de inteligência artificial, incluindo *Google Gemini 3.5 Flash*, exclusivamente para auxiliar na melhoria da linguagem e legibilidade deste manuscrito. O uso dessas ferramentas foi realizado sob supervisão humana, e os autores revisaram e editaram cuidadosamente o conteúdo, assumindo total responsabilidade pelo texto final publicado.

REFERÊNCIAS

AHLAWAT, H.; AGGARWAL, N.; GUPTA, D. Automatic speech recognition: A survey of deep learning techniques and approaches. **International Journal of Cognitive Computing in Engineering**, Elsevier, v. 6, p. 201–237, 2025. DOI: https://doi.org/10.1016/j.ijcce.2024.12.007.

AHMED, T.; PAI, K. S.; DEVANBU, P. *et al.* Improving few-shot prompts with relevant static analysis products. **arXiv preprint arXiv:2304.06815**, Apr, 2023. DOI:https://doi.org/10.48550/arXiv.2304.06815.

BASAK, S.; AGRAWAL, H.; JENA, S. *et al.* Challenges and limitations in speech recognition technology: A critical review of speech signal processing algorithms, tools and systems. **CMES-Computer Modeling in Engineering and Sciences**, Tech Science Press, 2023. DOI: https://doi.org/10.32604/cmes.2022.021755.

BELODEDOV, M. V.; FONKANTS, R. V.; SAFIN, R. R. Development of an algorithm for optimal encoding of wav files using genetic algorithms. In: IEEE. **2023 5th international youth conference on radio electronics, electrical and power engineering (REEPE)**. [S.l.], 2023. v. 5, p. 1–6. DOI: https://doi.org/10.1109/reepe57272.2023.10086837.

BENKERZAZ, S.; ELMIR, Y.; DENNAI, A. A study on automatic speech recognition. **Journal of Information Technology Review**, v. 10, n. 3, p. 77–85, 2019. DOI: https://doi.org/10.6025/jitr/2019/10/3/77-85.

BHATT, S.; JAIN, A.; DEV, A. Acoustic modeling in speech recognition: a systematic review. **International Journal of Advanced Computer Science and Applications**, Science and Information (SAI) Organization Limited, v. 11, n. 4, 2020. DOI: https://doi.org/10.14569/ijacsa.2020.0110455.

CHANG, L.-C.; HUNG, J.-W. A preliminary study of robust speech feature extraction based on maximizing the probability of states in deep acoustic models. **Applied System Innovation**, v. 5, n. 4, 2022. ISSN 2571-5577. DOI: https://doi.org/10.3390/asi5040071. Disponível em: https://www.mdpi.com/2571-5577/5/4/71.

CHENG, S.; ZHU, P.; LIU, J. *et al.* A survey of grapheme-to-phoneme conversion methods. **Applied Sciences**, v. 14, n. 24, 2024. ISSN 2076-3417. DOI: https://doi.org/10.3390/app142411790. Disponível em: https://www.mdpi.com/2076-3417/14/24/11790.

COOPER, N.; SCHOLAK, T. Perplexed: Understanding when large language models are confused. **arXiv preprint arXiv:2404.06634**, 2024. DOI: https://doi.org/10.48550/arXiv.2404.06634.

EDDY, S. R. Hidden markov models. **Current opinion in structural biology**, Elsevier, v. 6, n. 3, p. 361–365, 1996. DOI: https://doi.org/10.1016/S0959-440X(96)80056-X.

ERRATTAHI, R.; HANNANI, A. E.; OUAHMANE, H. Automatic speech recognition errors detection and correction: A review. **Procedia Computer Science**, Elsevier, v. 128, p. 32–37, 2018. DOI: https://doi.org/10.1016/j.procs.2018.03.005.

ESQUINSANI, R. S. S. As atas de reuniões enquanto fontes para a história da educação: pautando a discussão a partir de um estudo de caso. **Educação Unisinos**, Universidade do Vale do Rio dos Sinos, v. 11, n. 2, p. 103–110, 2007.

FENDJI, J. L. K. E.; TALA, D. C. M.; YENKE, B. O. *et al.* Automatic speech recognition using limited vocabulary: A survey. **Applied Artificial Intelligence**, Taylor & Francis, v. 36, n. 1, p. 2095039, 2022. DOI: https://doi.org/10.1080/08839514.2022.2095039.

GRAHAM, C.; ROLL, N. Evaluating openai's whisper asr: Performance analysis across diverse accents and speaker traits. **JASA Express Letters**, v. 4, 02 2024. DOI: https://doi.org/10.1121/10.0024876.

INDURKHYA, N.; DAMERAU, F. An overview of modern speech recognition xuedong huang and li deng. **Handbook of natural language processing**, Chapman and Hall/CRC, p. 363–90, 2010. DOI: https://doi.org/10.1201/9781420019329.ch12.

LABIED, M.; BELANGOUR, A. Automatic speech recognition features extraction techniques: A multi-criteria comparison. **International Journal of Advanced Computer Science and Applications**, Science and Information (SAI) Organization Limited, v. 12, n. 8, 2021. DOI: https://doi.org/10.14569/ijacsa.2021.0120821.

MARVIN, G.; RAUDHA, N. H.; JJINGO, D. *et al.* Prompt engineering in large language models. In: ____. [S.l.: s.n.], 2024. p. 387–402. ISBN 978-981-99-7999-8. DOI: https://doi.org/10.1007/978-981-99-7962-2_30.

NAVEED, H.; KHAN, A. U.; QIU, S. *et al.* A comprehensive overview of large language models. **ACM Transactions on Intelligent Systems and Technology**, ACM New York, NY, v. 16, n. 5, p. 1–72, 2025. DOI: https://doi.org/10.48550/arXiv.2307.06435.

NGUYEN, H.; CHEN, H.; POBBATHI, L. *et al.* A comparative study of quality evaluation methods for text summarization. **arXiv preprint arXiv:2407.00747**, 2024. DOI: https://doi.org/10.48550/arXiv.2407.00747.

RABELO, L. R. A. **Um estudo de caso do modelo de reconhecimento de voz Whisper para transcrição de conferências TEDx via aprendizado fraco**. Dissertação (Trabalho de Conclusão de Curso (Graduação em Engenharia de Software)) — Universidade Federal do Ceará, Quixadá, 2022. Disponível em: https://http://repositorio.ufc.br/handle/riufc/73941.

REIMERS, N.; GUREVYCH, I. Sentence-bert: Sentence embeddings using siamese bert-networks. In: **Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)**. [S.l.: s.n.], 2019. p. 3982–3992. DOI: https://doi.org/10.18653/v1/d19-1410.

ROY, S. **Semantic-WER: A Unified Metric for the Evaluation of ASR Transcript for End Usability**. 2021. DOI: https://doi.org/10.48550/arXiv.2106.02016. Disponível em: https://arxiv.org/abs/2106.02016.

SAINBURG, T.; GENTNER, T. Q. Toward a computational neuroethology of vocal communication: From bioacoustics to neurophysiology, emerging tools and future directions. **Frontiers in Behavioral Neuroscience**, Volume 15 - 2021, 2021. ISSN 1662-5153. DOI: https://doi.org/10.3389/fnbeh.2021.811737. Disponível em: https://www.frontiersin.org/journals/behavioral-neuroscience/articles/10.3389/fnbeh.2021.811737.

SAXENA, D.; FAROOQUI, A.; ALI, S. Extricate features utilizing mel frequency cepstral coefficient in automatic speech recognition system. **Int. J. Eng. Manuf**, v. 12, n. 6, p. 14–21, 2022. DOI: https://doi.org/10.5815/ijem.2022.06.02.

STRAUSS, E.; JÚNIOR, M. V. B.; GONZAGA, V. D. *et al.* Inteligência artificial e reconhecimento de voz na unidade de resposta audível (ura). **Projectus**, v. 7, n. 1, p. 69–89, 2022. DOI: https://doi.org/10.15202/25254146.2022v7n1p68.

TOLEDO, T. F. d. **Desenvolvimento de um protótipo de sistema web para elaboração de laudos médicos utilizando sistemas de reconhecimento automático de fala**. Dissertação (Dissertação (Mestrado em Engenharia Elétrica e Computação)) — Universidade Estadual do Oeste do Paraná, Foz do Iguaçu, 2017. Disponível em: https://tede.unioeste.br/handle/tede/3399. Disponível em: https://tede.unioeste.br/handle/tede/3399.

VASWANI, A.; SHAZEER, N.; PARMAR, N. *et al.* Attention is all you need. **Advances in neural information processing systems**, Long Beach, CA, v. 30, n. 1, p. 5998–6008, 2017.

WESTER, M. Pronunciation modeling for asr–knowledge-based and data-derived methods. **Computer Speech & Language**, Elsevier, v. 17, n. 1, p. 69–85, 2003. DOI: https://doi.org/10.1016/s0885-2308(02)00030-x.

YE, Q.; AHMED, M.; PRYZANT, R. *et al.* Prompt engineering a prompt engineer. In: **Findings of the Association for Computational Linguistics: ACL 2024**. [S.l.: s.n.], 2024. p. 355–385. DOI: https://doi.org/10.18653/v1/2024.findings-acl.21.

YOSHIZUMI, V. H.; LOPES, S.; SPATTI, D. H. *et al.* Aprimoramentos de performance de modelos llm para aplicações de transcrição de texto em sistemas elétricos de potência. **Anais**, 2024.

APÊNDICE A – ATA GERADA PELO SISTEMA

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
CENTRO MULTIDISCIPLINAR DE PAU DOS FERROS
DEPARTAMENTO DE ENGENHARIAS E TECNOLOGIA
BR 226, KM 405, s/n, São Geraldo - Pau dos Ferros/RN
Telefone: (84) 3317-8512 e-mail: decen@ufersa.edu.br

ATA DA 6 Reunião Extraordinária do Colegiado do PPgCEM

Vamos dar início, então, à nossa sexta reunião extraordinária do Colégio do PPG 100. Peço desculpas por convocá-los neste horário e dia, mas, como informado anteriormente, este era o prazo para aprovar a segunda prorrogação extraordinária do nosso edital de atividades. Considerando que na sexta-feira será facultativo o ponto, preciso aprovar hoje para enviar amanhã à secretaria. Apesar da greve, encaminharei a convocação formal. Como a secretaria está em greve, não participará, mas providenciarei a inserção dos dados junto à Procuradoria para garantir que fiquem cobertos.

O objetivo desta reunião é aprovar a prorrogação dos alunos, com pauta única: deliberar sobre as solicitações de Aurivan Gomes, Hidro cleiton, Indy de Pamir, José Ilângio e Teresa Tavares. Sem discussão, peço que se manifestem no chat com "favorável", "contra" ou "abstenção". Iniciaremos pela solicitação do Lucas, que teve problemas de saúde. Ele solicitou três meses adicionais (originalmente seis, por engano), com justificativa médica relacionada a uma condição (provavelmente "Bornout" ou esgotamento). Seu orientador, Rodrigo Collares, endossa a necessidade para conclusão da defesa. Sem discussão, deliberem via chat.

Aprovada a solicitação do Lucas, passamos a Aurivan Gomes, orientando da professora Patrícia. Ele alega atrasos devido à greve, mas afirma estar pronto para defesa. Patrícia confirma: o trabalho está concluído, restando apenas agendar banca. Deliberem no chat. Seguimos para Ítalo, que também justificou atrasos por infraestrutura, mas já tem a qualificação aprovada e texto de defesa pronto. Patrícia reitera: está apto. Deliberem.

Prosseguimos com Indy de Pamir, orientando do professor Júlio. Ele enfrentou problemas com equipamentos de um projeto FAPERGS, necessitando de três meses para refazer experimentos. Júlio detalha: parte dos testes foi realizada na estação da APA, e as análises finais estão em andamento. Sem objeções, deliberem. Por fim, Teresa Tavares, também orientanda de Patrícia, atrasou-se devido à falta de insumos (gás) e a greve agravou a situação. Patrícia explica: o fornecedor atrasa reiteradamente a entrega, alegando questões contratuais. Deliberem.

Com todas as solicitações aprovadas (Aurivan, Ítalo, Indy e Teresa), formalizaremos as prorrogações junto à secretaria amanhã, visando garantir os históricos. Agradeço a presença de todos e encerro esta reunião extraordinária. Desejo um bom feriado de Corpus Christi e até breve.

APÊNDICE B – LINKS E DISPONIBILIDADE DAS REUNIÕES ANALISADAS

Abaixo são apresentados os links de acesso público e o status de disponibilidade das gravações em vídeo referentes às reuniões do colegiado e conselhos utilizadas como base de dados neste trabalho:

1. 6ª Reunião Extraordinária do Colegiado do PPgCEM: Indisponível
2. CONSUNI – 8ª Reunião Ordinária de 2025: https://www.youtube.com/watch?v=SlH2abdhGRA&t=182s
3. CONSEPE – 1ª Reunião Extraordinária de 2025: Indisponível
4. 2º Sessão da 1ª Reunião Extraordinária do CONSAD 2024: https://www.youtube.com/live/3Q-dU9Mgtqs?si=xVG9RLK4zWCahzcB
5. CONSUNI – Terceira Sessão da 2ª Reunião Extraordinária de 2025: https://youtu.be/RNhCHcVIavM?list=PLxU7YP4gWyX9rj0oS0FX4DpKbNsutdpRo
6. CONSEPE – 9ª Reunião Ordinária de 2025: https://www.youtube.com/watch?v=Y9aUJ9W187w&t=7s
7. CONSEPE – 10ª Reunião Ordinária de 2025: https://www.youtube.com/watch?v=8Oy8vANLa9s&t=2154s

ANEXO A – MODELO DE ATA INSTITUCIONAL

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
CENTRO MULTIDISCIPLINAR DE PAU DOS FERROS
DEPARTAMENTO DE ENGENHARIAS E TECNOLOGIA
BR 226, KM 405, s/n, São Geraldo - Pau dos Ferros/RN
Telefone: (84) 3317-8512 e-mail: decen@ufersa.edu.br

TÍTULO