

SAGAR: Sistema de Apoio à Geração Automática de Resumos com Inteligência Artificial Como Ferramenta Pedagógica Para Docentes

Rhyan Ximenes de Brito  Francisco Alexandre de A. Rocha  Francisco Gabriel Braga do Nascimento 

Campus Tianguá - CE

Campus Tianguá - CE

Campus Tianguá - CE

Instituto Federal do Ceará (IFCE)

Instituto Federal do Ceará (IFCE)

Instituto Federal do Ceará (IFCE)

Tianguá-CE, Brasil

Tianguá-CE, Brasil

Tianguá-CE, Brasil

rxbrito@gmail.com

alexandresobral2004@gmail.com

gabrielbraga@gmail.com

Abstract—This study presents SAGAR (System for Supporting Automatic Summary Generation), an Artificial Intelligence-based solution designed to support teachers in the automatic production of structured summaries from lecture-based classes. The system integrates Automatic Speech Recognition and Natural Language Processing techniques into a *pipeline* comprising audio capture, temporal segmentation, transcription, partial summarization, and aggregation, with the aim of reducing the effort required for manual preparation of post-class educational materials. The platform is intended for teacher use, and the generated summaries may be shared with students through channels external to the system. The initial evaluation was internal, exploratory, and preliminary, and was conducted by the authors in four controlled scenarios, considering three dimensions: summary coherence, transcription integrity, and functional consistency. The first two dimensions were assessed using a five-point ordinal scale, with all outputs receiving the maximum score of 5.0, whereas functional consistency was examined across five stages of the *pipeline*. A total of 20 functional checks were performed, all completed without observable failures, corresponding to a descriptive execution rate of 100% in the analyzed scenarios. These findings should be interpreted solely as preliminary evidence of system operation under controlled conditions, since the evaluation did not involve blinding, independent evaluators, real users, or inter-rater agreement measures. As a future step, the project will be submitted to a Research Ethics Committee to enable external evaluation with human participants in real educational settings.

Index Terms—Artificial Intelligence, Natural Language Processing, Automatic Speech Recognition, Text Summarization, Educational Technology.

I. INTRODUÇÃO

Ao longo das últimas décadas, a Inteligência Artificial (IA) passou por transformações significativas que redefiniram suas aplicações práticas. A consolidação de arquiteturas cada vez mais eficientes permitiu a criação de sistemas capazes de interpretar, classificar e gerar informações em diferentes formatos [1]. A introdução da arquitetura *transformer*, por exemplo, representou um marco decisivo nesse percurso, ao viabilizar modelos mais escaláveis e contextualmente sensíveis, que culminaram nos atuais modelos de linguagem de grande porte [2]. Esses avanços não apenas ampliaram o repertório de tarefas automatizáveis, mas também abriram caminho para

o uso da IA em domínios nos quais a interpretação contextual e a síntese de informações se tornam essenciais, incluindo, de forma crescente, os ambientes educacionais [3].

Nesse cenário de amadurecimento tecnológico, a incorporação da IA ao contexto educacional tem ampliado as possibilidades de apoio às atividades pedagógicas e ao trabalho docente [4]. Modelos avançados de Processamento de Linguagem Natural (PLN) possibilitam automatizar tarefas relacionadas à produção, organização e síntese de materiais educacionais, incluindo resumos e explicações complementares [5]. Ao transformar aulas, discussões e exposições orais em recursos textuais estruturados, a IA pode auxiliar o docente na organização dos conteúdos ministrados e na preparação de materiais complementares para uso posterior [6]. Tais recursos não substituem a prática pedagógica, mas podem contribuir para tornar mais eficientes os processos de registro, síntese e organização de conteúdos educacionais [7].

A dinâmica das aulas expositivas, caracterizada pela apresentação contínua e, muitas vezes, densa de informações, também impõe desafios à organização e à preparação posterior dos conteúdos ministrados. A elaboração manual de registros, sínteses e materiais de apoio pode demandar tempo adicional do docente, sobretudo quando envolve a identificação, seleção e organização dos principais conceitos apresentados durante a aula. Embora as anotações realizadas pelos discentes constituam um recurso importante para o acompanhamento das atividades acadêmicas [8], [9], a proposta deste trabalho concentra-se especificamente no apoio ao docente na produção de materiais estruturados derivados do conteúdo oral das aulas. Nesse sentido, a literatura evidencia que a automação baseada em IA apresenta potencial para reduzir a carga operacional do docente, especialmente em tarefas de processamento e síntese textual, promovendo maior eficiência sem comprometer a fidelidade do conteúdo pedagógico [10].

Nesse contexto, evidencia-se a necessidade de mecanismos capazes de auxiliar o docente no registro, na organização e na síntese dos conteúdos apresentados durante aulas expositivas. O problema central abordado neste trabalho reside na ausência de um fluxo automatizado que permita converter o

conteúdo oral das aulas em materiais textuais estruturados, reduzindo o esforço manual associado à preparação desses materiais e facilitando sua posterior utilização e disponibilização pelo docente. Dessa forma, o apoio aos discentes ocorre de maneira indireta, por meio dos materiais produzidos pelo docente e posteriormente compartilhados em canais externos à plataforma.

O presente trabalho tem como finalidade apresentar os fundamentos e o desenvolvimento da plataforma *web* denominada Sistema de Apoio à Geração Automática de Resumos (SAGAR), uma ferramenta pedagógica baseada em IA destinada a apoiar docentes na conversão de aulas expositivas em resumos textuais estruturados. A proposta integra recursos de reconhecimento automático de fala e processamento de linguagem natural, possibilitando a transcrição de conteúdos orais e sua posterior organização em sínteses estruturadas que podem ser utilizadas pelo docente como materiais educacionais complementares e, quando pertinente, posteriormente compartilhadas com os discentes por meio de canais externos à plataforma. Além da implementação do sistema, foram realizadas simulações com conteúdos das disciplinas Desenvolvimento Web 1, Desenvolvimento Web 2, Sistemas Operacionais 1 e Sistemas Operacionais 2, contemplando diferentes níveis de complexidade conceitual e técnica. Essas simulações permitiram verificar o comportamento do sistema diante de conteúdos introdutórios e avançados, avaliando sua capacidade de processar informações relevantes e gerar materiais textuais estruturados de apoio às atividades docentes.

Este trabalho está organizado em sete seções. A Seção II apresenta os trabalhos relacionados, discutindo estudos e soluções baseadas em Inteligência Artificial, reconhecimento de fala, sumarização textual e recursos digitais aplicados ao contexto educacional. A Seção III aborda a fundamentação teórica, reunindo os principais conceitos relacionados à Inteligência Artificial na Educação, ao Processamento de Linguagem Natural, à Sumarização Automática de Texto e ao Reconhecimento de Fala. A Seção IV descreve a metodologia adotada, contemplando a caracterização da pesquisa, a visão geral da solução, as tecnologias utilizadas, a arquitetura do sistema, o *pipeline* de processamento, os procedimentos de desenvolvimento e teste, os critérios de avaliação e as limitações da abordagem. A Seção V apresenta os resultados e a discussão, abrangendo os aspectos éticos, de privacidade e proteção de dados, a caracterização dos cenários simulados, os resultados técnico-funcionais, a análise comparativa dos cenários e as limitações da avaliação realizada. A Seção VI reúne as considerações finais e os trabalhos futuros, sintetizando os resultados obtidos nas simulações controladas e indicando a necessidade de avaliações posteriores com docentes e discentes em contexto educacional real, incluindo aspectos de usabilidade, qualidade dos resumos, percepção dos usuários e procedimentos éticos aplicáveis. Por fim, a Seção VII apresenta os apêndices, nos quais são disponibilizadas as telas e evidências visuais relacionadas ao funcionamento do SAGAR e aos resumos gerados nos cenários simulados.

II. TRABALHOS RELACIONADOS

A evolução das tecnologias baseadas em Inteligência Artificial (IA) e Processamento de Linguagem Natural (PLN) tem redefinido as fronteiras do suporte pedagógico. Com base na literatura recente, os esforços para otimizar a experiência de aprendizagem concentram-se em dois grandes eixos: o desenvolvimento de sistemas de transcrição e acessibilidade focados no processamento de áudio e vídeo, e a inserção de recursos digitais interativos ou gerativos no cotidiano escolar. Esta seção mapeia tais iniciativas, estabelecendo as bases conceituais para o posicionamento do SAGAR. Desse modo, verifica-se que:

[11] analisaram o uso de modelos de linguagem de grande escala no ensino de Computação, avaliando ferramentas como ChatGPT, Claude, Copilot, Gemini, Gemini-Pro e *DeepSeek* em tarefas de geração de interfaces gráficas, tradução de código e criação de recursos educacionais. A partir de mais de 350 testes, os autores produziram mais de 250 ferramentas funcionais, com taxa de sucesso de 84%. Os resultados indicaram que as *Large Language Models* (LLMs) podem acelerar a criação de materiais interativos para o ensino de Computação, especialmente em atividades envolvendo programação, grafos, questionários e modelos de aprendizado de máquina.

Já [12] propuseram o MOSALA, modelo socioafetivo baseado em *Learning Analytics*, desenvolvido para apoiar docentes no acompanhamento de discentes em Ambientes Virtuais de Aprendizagem (AVA). O modelo combina dados sociais e afetivos para identificar cenários socioafetivos e recomendar estratégias pedagógicas personalizadas. Como resultado, foram definidos 59 cenários e 354 estratégias, além de um protótipo de Mapa Socioafetivo, com potencial de aplicação em outros contextos do Ensino à Distância (EaD).

[13] avaliaram a aplicação de tecnologias de reconhecimento de fala em ambientes educativos para automatizar a transcrição de aulas e gerar legendas em tempo real, com o objetivo de melhorar a acessibilidade e o desempenho dos discentes. O estudo comparou dois métodos de implementação: legendas instantâneas durante a aula e transcrições pós-aula. Os resultados indicaram que a transcrição pós-aula, com maior taxa de precisão, promove melhorias no envolvimento dos discentes, sobretudo na compreensão do conteúdo. Esses avanços apontaram para um potencial significativo na utilização de processamento de linguagem natural em plataformas educacionais.

[14] apresentaram uma plataforma educacional baseada em IA que integra transcrição automática, sumarização textual e geração de materiais de estudo interativos, como questões de múltipla escolha e *flashcards*. A solução destacou o uso do modelo *Whisper* para transcrição em tempo real, além de algoritmos de PLN, como *TextRank* e *DistilBART*, para sintetizar conteúdos provenientes de vídeos, PDFs e anotações de aula. A plataforma buscou reduzir a carga cognitiva do discente por meio da automatização de registros e da criação de resumos estruturados, oferecendo suporte pós-aula e ampliando o acesso ao conteúdo.

[15] realizaram um estudo onde observou-se que a lousa digital como uma ferramenta educacional é capaz de tornar o ensino mais interativo, colaborativo e atrativo, favorecendo o engajamento dos discentes e o acesso a diferentes ferramentas educacionais. Contudo, apontou desafios como falta de formação docente, infraestrutura inadequada, distrações, desigualdade digital e dificuldades de integração ao currículo. Assim, sua efetividade depende de planejamento, capacitação e suporte técnico adequados.

[16] discutiram a utilização de simulações computacionais como estratégia de apoio ao ensino de Física, ressaltando sua importância na representação de fenômenos e situações experimentais que nem sempre podem ser reproduzidos em laboratórios didáticos. Segundo os autores, esse tipo de recurso contribui para a aprendizagem ao combinar elementos gráficos, animações e atividades interativas, tornando a compreensão dos conteúdos mais acessível e dinâmica. No contexto das disciplinas iniciais do curso de Engenharia da Faculdade de Engenharia Industrial (FEI), as simulações foram empregadas como complemento às aulas teóricas, evidenciando o potencial das tecnologias digitais para enriquecer o ensino presencial.

Os autores [17] analisaram a qualidade das legendas automáticas geradas pelo *software* CapCut em vídeos verticais, considerando sua adequação aos parâmetros da legendagem para surdos e ensurdecidos. Embora ferramentas baseadas em Inteligência Artificial ampliem o acesso à criação de legendas, ainda apresentam limitações relacionadas a erros de transcrição, segmentação inadequada e tempo insuficiente de leitura. Os resultados indicaram que essas falhas podem comprometer a acessibilidade digital da comunidade surda, reforçando a necessidade de critérios específicos para vídeos curtos e verticais.

Para sintetizar o panorama do estado da arte e evidenciar a lacuna científica que o SAGAR visa preencher, a Tabela I apresenta uma análise comparativa entre a solução proposta e os principais trabalhos correlatos da literatura, mapeando suas funcionalidades, tecnologias, métodos de avaliação e limitações identificadas.

III. FUNDAMENTAÇÃO TEÓRICA

A fundamentação teórica deste trabalho está organizada em quatro subseções principais, que apresentam os conceitos necessários para compreender a construção e o funcionamento do SAGAR. Inicialmente, discute-se a aplicação da Inteligência Artificial na Educação, destacando seu papel no apoio ao ensino, na personalização da aprendizagem e na automação de tarefas pedagógicas. Em seguida, aborda-se o Processamento de Linguagem Natural (PLN), área responsável por permitir que sistemas computacionais interpretem, organizem e gerem textos em linguagem humana. Na sequência, são apresentadas as técnicas de Sumarização Automática de Texto (SAT), fundamentais para a produção de resumos estruturados a partir de conteúdos extensos. Por fim, discute-se o Reconhecimento de Fala (RF), tecnologia essencial para a conversão de aulas orais em texto, constituindo a etapa inicial do fluxo de processamento adotado pelo SAGAR.

A. Inteligência Artificial na Educação

A presença da Inteligência Artificial (IA) na educação vem redesenhando práticas pedagógicas, processos de aprendizagem e formas de gestão, como mostram [18] ao mapear, em perspectiva recente, a consolidação de aplicações voltadas à personalização, à avaliação automatizada e à análise de dados educacionais.

Os autores [3], [18], [19] destacam que sistemas tutores inteligentes, plataformas adaptativas e motores de recomendação de conteúdo possibilitam ajustar ritmo, dificuldade e recursos às necessidades individuais dos discentes, rompendo com o modelo homogêneo de ensino ainda predominante. Em linha semelhante, [19] evidencia como tais tecnologias ampliam a autonomia discente, automatizam rotinas de correção e acompanhamento e liberam o docente para tarefas de maior densidade conceitual e relacional. Na mesma direção, a discussão sobre educação e IA em contexto universitário ressalta que, diante de máquinas capazes de operar sobre grande parte do conhecimento explícito, a formação superior precisa deslocar o foco para competências subjetivas, éticas e críticas, sob pena de reduzir-se a mero treinamento técnico.

A aprendizagem personalizada apoiada em IA, descrita por [18], apoia-se na coleta e análise intensiva de dados de desempenho e trajetórias de estudo, permitindo intervenções precoces em casos de dificuldades e risco de evasão. [19] aprofunda esse ponto ao discutir mineração de dados e análise preditiva para orientar decisões pedagógicas baseadas em evidências, destacando ainda que tecnologias como leitores de tela, reconhecimento de voz e interfaces adaptativas ampliam oportunidades para discentes com deficiência. [20] acrescentam o papel de *chatbots* especializados e sistemas generativos como tutores permanentes, redefinindo o ecossistema de recursos didáticos disponíveis.

Contudo, [18] alertam para riscos de viés e aprofundamento de desigualdades quando decisões pedagógicas se apoiam em modelos opacos, enquanto [19] descrevem um ponto de tensão em que lógicas algorítmicas deslocam a centralidade humana, empurrando a educação para uma racionalidade instrumental orientada por métricas de desempenho. Já [20] destacam a necessidade de uma alfabetização tecnológica contínua de docentes e discentes para que não se tornem usuários passivos desses sistemas. No plano formativo, os estudos convergem ao defender que a integração da IA exige formação docente crítica e ética, e que seu potencial de democratizar o acesso e personalizar percursos depende de governança robusta e comprometimento com a formação integral dos sujeitos [18], [19], [20].

B. Processamento de Linguagem Natural

O Processamento de Linguagem Natural (PLN) é uma subárea da Inteligência Artificial (IA) dedicada ao desenvolvimento de métodos computacionais capazes de analisar, interpretar, representar e gerar informações expressas em linguagem humana. Suas aplicações abrangem diferentes tarefas, como classificação textual, extração de informações, tradução automática, compreensão de diálogos, geração de texto e

TABLE I
ANÁLISE COMPARATIVA ENTRE OS TRABALHOS RELACIONADOS E A PROPOSTA DESTE ARTIGO

Trabalho	Principais Funcionalidades	Tecnologias de Base	Procedimento de Avaliação	Escopo / Limitações
[11]	Geração de interfaces, tradução de código e recursos educacionais.	LLMs: ChatGPT, Claude, Copilot, Gemini e DeepSeek.	Mais de 350 testes, com produção de mais de 250 ferramentas e taxa de sucesso de 84%.	Não contempla transcrição ou sumarização de aulas.
[12]	Identificação de cenários socioafetivos e recomendação de estratégias pedagógicas.	<i>Learning Analytics</i> e AVA.	Definição de 59 cenários, 354 estratégias e protótipo de mapa socioafetivo.	Direcionado ao acompanhamento socioafetivo, sem processamento de áudio ou sumarização.
[13]	Transcrição automática de aulas e geração de legendas.	Reconhecimento Automático de Fala (ASR).	Comparação entre legendas em tempo real e transcrições pós-aula.	Concentra-se em transcrição e acessibilidade, sem geração de resumos estruturados.
[14]	Transcrição, sumarização, questões de múltipla escolha e <i>flashcards</i> .	<i>Whisper</i> , <i>TextRank</i> e <i>DistilBART</i> .	Demonstração da integração das funcionalidades da plataforma.	Solução ampla para diferentes tipos de conteúdo, não restrita a aulas expositivas.
[15]	Apoio ao ensino por meio de lousas digitais interativas.	Lousas digitais e recursos educacionais interativos.	Análise de benefícios e desafios de utilização no contexto educacional.	Dependência de infraestrutura, capacitação docente e integração curricular.
[16]	Simulação computacional de fenômenos físicos para apoio ao ensino.	Simulações, animações e recursos gráficos.	Aplicação como complemento às aulas teóricas de Engenharia.	Não contempla processamento de fala ou geração automatizada de materiais textuais.
[17]	Geração automática de legendas para vídeos.	Ferramentas de IA do <i>CapCut</i> .	Análise da transcrição, segmentação e tempo de leitura das legendas.	Apresenta erros de transcrição, segmentação inadequada e limitações no tempo de leitura.
Proposta deste artigo	Transcrição e transformação de aulas expositivas em resumos textuais estruturados.	ASR, PLN, LLM, React.js, Node.js e Firebase.	Verificação técnico-funcional em cenários controlados, com análise das etapas do fluxo de processamento.	Dependência de Internet, perfil restrito a docentes, ausência de recursos de acessibilidade e validação restrita a cenários controlados.

sumarização. Com o avanço das arquiteturas baseadas em *transformers* e dos Grandes Modelos de Linguagem (*Large Language Models* – LLMs), essas tarefas passaram a ser executadas com maior capacidade de representação contextual e geração textual [21].

Ainda de acordo com [21], os LLMs podem ser empregados por meio de diferentes paradigmas, incluindo estratégias que não exigem ajuste dos parâmetros do modelo, como *zero-shot* e *few-shot*, bem como abordagens que envolvem *fine-tuning*. Essa flexibilidade permite que modelos previamente treinados sejam utilizados em diferentes tarefas de PLN a partir de instruções fornecidas em linguagem natural, reduzindo a necessidade de desenvolvimento de modelos específicos para cada problema.

Entre as aplicações relevantes do PLN destaca-se a sumarização automática de textos, cujo objetivo consiste em produzir representações mais concisas de conteúdos extensos, preservando as informações essenciais e o sentido global do texto [21]. Nesse contexto, técnicas de processamento e geração de linguagem possibilitam identificar informações relevantes, reduzir redundâncias e reorganizar o conteúdo de forma coerente, aspectos particularmente importantes no tratamento de documentos extensos e grandes volumes de informação textual.

O avanço das arquiteturas neurais e dos Grandes Modelos de Linguagem ampliou as possibilidades de compreensão e geração textual, permitindo que modelos previamente treinados sejam utilizados em tarefas como síntese, reorganização e

geração de conteúdos a partir de instruções em linguagem natural [21]. Na sumarização automática, essa evolução favoreceu especialmente as abordagens abstrativas, capazes de produzir novas formulações textuais preservando as informações centrais do conteúdo original, consolidando o PLN como base para o processamento automatizado de informações textuais.

C. Sumarização Automática de Texto

A sumarização automática de texto constitui uma tarefa central do Processamento de Linguagem Natural (PLN), voltada à produção de versões condensadas de documentos sem perda das informações essenciais e do sentido global. Sua relevância aumenta em cenários de sobrecarga informacional, nos quais é necessário acessar rapidamente os conteúdos mais importantes de grandes volumes textuais [22], [23]. Tradicionalmente, distinguem-se as abordagens extrativa, que seleciona sentenças do texto-fonte, e abstrativa, que reformula o conteúdo em novas sentenças; abordagens híbridas combinam ambas as estratégias para equilibrar fidelidade e fluidez [23]–[25].

Historicamente, a área evoluiu de métodos estatísticos clássicos para técnicas baseadas em aprendizado de máquina, redes neurais e, mais recentemente, Grandes Modelos de Linguagem (*Large Language Models* - LLMs). O trabalho de Luhn constitui um dos marcos iniciais, com o uso da frequência de termos para seleção de sentenças, influenciando métodos posteriores [22], [25], [26]. Técnicas extrativas baseadas em ponderação, regressão e programação linear ainda

permanecem relevantes em cenários de menor custo computacional [27], [28].

Com o avanço das Redes Neurais Profundas e da arquitetura *transformer*, a pesquisa passou a enfatizar modelos abstrativos e pré-treinados, capazes de representar relações semânticas mais complexas e produzir resumos linguisticamente mais fluidos [2], [24]. Os LLMs ampliaram essa capacidade ao permitir a execução de tarefas por meio de instruções em linguagem natural, utilizando estratégias como *zero-shot*, *few-shot* ou ajuste do modelo, o que favorece sua aplicação em síntese e reorganização textual [21]. Apesar disso, permanecem desafios relacionados a custo computacional, controle da geração e factualidade [25], [28].

No contexto de transcrições extensas, técnicas de PLN podem reduzir progressivamente o volume textual, identificar informações relevantes e reorganizá-las em estruturas mais adequadas à consulta posterior. No SAGAR, esse processamento ocorre após o reconhecimento de fala: as transcrições são submetidas a uma etapa intermediária de sumarização e, posteriormente, à agregação e geração do resumo final estruturado, permitindo transformar o conteúdo oral das aulas em representações textuais condensadas e organizadas.

A avaliação desses sistemas envolve tanto métricas automáticas quanto análise humana. ROUGE-L e BERTScore são frequentemente empregados para comparar resumos gerados com referências, enquanto a avaliação humana permanece importante para analisar aspectos como coerência, coesão, relevância e utilidade [28], [29]. O pré-processamento textual também pode contribuir para redução de ruídos e organização dos dados [22].

Apesar dos avanços, a sumarização automática ainda apresenta limitações relacionadas à adaptação a diferentes domínios, à disponibilidade de recursos para o português, à necessidade de avaliações mais rigorosas e ao risco de alucinações ou inconsistências em modelos generativos [22], [24], [25]. Assim, a evolução da área evidencia uma transição de métodos predominantemente extrativos para abordagens mais semânticas, generativas e sensíveis ao contexto [25], [28].

D. Reconhecimento de Fala

O Reconhecimento Automático de Fala (*Automatic Speech Recognition* – ASR) compreende o conjunto de técnicas computacionais destinadas à conversão de sinais de fala em representações textuais. Historicamente, esses sistemas evoluíram a partir de abordagens fundamentadas em processamento de sinais e modelos estatísticos para arquiteturas baseadas em aprendizado profundo, capazes de aprender representações acústicas e linguísticas diretamente a partir de grandes volumes de dados [30], [31]. Essa evolução tem permitido avanços significativos na transcrição de fala contínua e na capacidade de lidar com diferentes locutores, idiomas e condições acústicas.

Entre as abordagens recentes destaca-se o modelo Whisper, apresentado por [32], desenvolvido a partir de aproximadamente 680 mil horas de dados de áudio supervisionados de forma multilíngue e multitarefa. O modelo emprega uma

arquitetura baseada em *Transformer* e foi projetado para executar tarefas como reconhecimento de fala, tradução e identificação de idioma. A diversidade e a escala dos dados utilizados em seu treinamento contribuíram para sua capacidade de generalização em diferentes conjuntos de dados e cenários, inclusive sem necessidade de ajuste específico para cada domínio.

Apesar dos avanços obtidos com modelos de larga escala, o reconhecimento automático de fala permanece sujeito a desafios relacionados às características acústicas do ambiente, ruídos, variações de pronúncia, diversidade linguística e extensão dos áudios processados [31]. No caso de conteúdos de longa duração, estudos baseados no Whisper também apontam limitações relacionadas à segmentação, precisão temporal e processamento sequencial, motivando estratégias específicas para divisão e alinhamento dos trechos de áudio [33]. Assim, o desempenho de sistemas de reconhecimento de fala depende não apenas do modelo empregado, mas também das condições de captura e das estratégias adotadas para o processamento do sinal de áudio.

IV. METODOLOGIA

A Figura 1 apresenta o fluxo metodológico adotado no trabalho, descrevendo as etapas envolvidas na transformação do áudio de uma aula em um resumo estruturado. O processo inicia-se com a captura do áudio e sua segmentação temporal em blocos fixos de aproximadamente 9,5 minutos. Cada segmento é posteriormente submetido à transcrição automática por meio de técnicas de Reconhecimento de Fala (RF) e à sumarização parcial com técnicas de Processamento de Linguagem Natural (PLN). Os resumos parciais resultantes são integrados na etapa de agregação, na qual o conteúdo é organizado, refinado e unificado, produzindo o resumo final estruturado.

A. Caracterização da Pesquisa

A pesquisa caracteriza-se, quanto à sua natureza, como aplicada, por estar voltada ao desenvolvimento do SAGAR, uma plataforma *web* baseada em Inteligência Artificial destinada ao registro, processamento e organização de conteúdos provenientes de aulas expositivas. A solução integra técnicas de Reconhecimento de Fala (RF) e Processamento de Linguagem Natural (PLN) para realizar captura de áudio, segmentação temporal, transcrição, sumarização e agregação de conteúdos. Quanto aos objetivos, o estudo apresenta caráter descritivo e exploratório, ao documentar a arquitetura, as tecnologias e o fluxo de processamento da solução, bem como investigar a integração dessas técnicas em um sistema computacional voltado ao contexto educacional.

Quanto aos procedimentos, o trabalho caracteriza-se como uma pesquisa de desenvolvimento tecnológico, envolvendo a concepção, implementação e verificação técnico-funcional do sistema por meio de simulações controladas com conteúdos de diferentes disciplinas e níveis de complexidade. Essas simulações constituem uma verificação interna e preliminar,

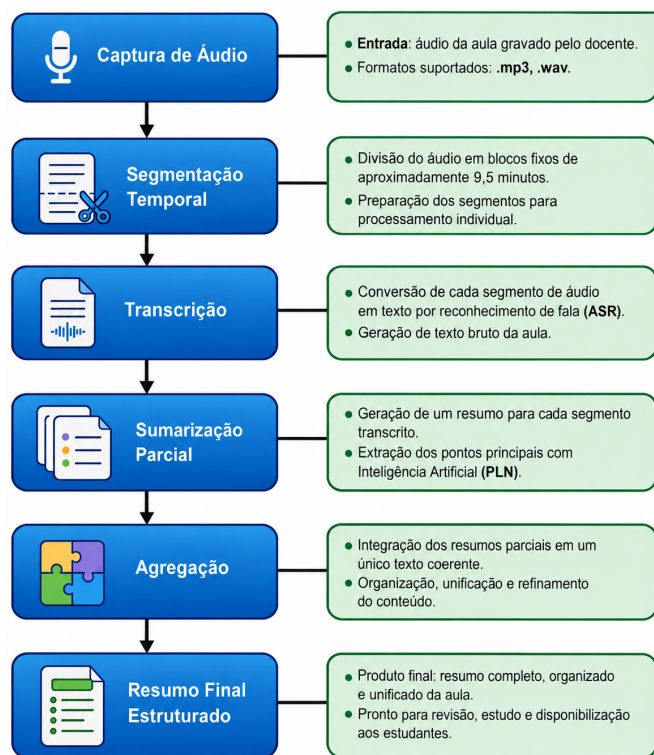


Fig. 1. Fluxo Metodológico

conduzida pelos próprios autores, sem participação de docentes, discentes ou avaliadores externos. Assim, o estudo não se caracteriza como pesquisa experimental, pois não contempla hipótese experimental, grupo de controle, manipulação de variáveis ou comparação com *baseline*, de modo que os resultados obtidos devem ser interpretados como evidências do funcionamento técnico da solução nos cenários simulados, e não de sua efetividade pedagógica em contexto real.

B. Visão Geral da Solução

A plataforma construída consiste em uma solução *web* desenvolvida para apoiar o processo de aprendizagem ao transformar conteúdos orais de aula em resumos estruturados. Por meio da gravação de áudio realizada pelo docente, o sistema utiliza serviços de IA para gerar automaticamente transcrições e resumos, permitindo que os discentes revisitem o conteúdo em formato textual. A ferramenta é capaz de capturar, armazenar e disponibilizar os materiais gerados.

C. Tecnologias Utilizadas

O desenvolvimento da solução foi baseado em uma arquitetura moderna de aplicações *web*, integrando tecnologias tanto no *front-end* quanto no *back-end*, além de serviços em nuvem e APIs de Inteligência Artificial. No lado do servidor, foi utilizada a plataforma *Node.js*, que possibilita a execução de *JavaScript* no *back-end* com alto desempenho e suporte a operações assíncronas. Para estruturar a aplicação e gerenciar requisições HTTP, foi empregado o *framework Express.js*, permitindo a criação de uma API RESTful organizada, escalável

e alinhada às boas práticas de desenvolvimento. Essa API atua como intermediária entre a interface do usuário, o banco de dados e os serviços externos.

No *front-end*, a interface foi desenvolvida com *React.js*, uma biblioteca que permite a construção de interfaces dinâmicas por meio de componentes reutilizáveis, proporcionando maior organização e escalabilidade da aplicação. A comunicação entre *front-end* e *back-end* ocorre via requisições HTTP, utilizando bibliotecas como *Axios*. Para armazenamento e gerenciamento de dados, foi adotado o *Firebase*, incluindo o *Firestore* como banco de dados NoSQL, o *Firebase Authentication* para controle de acesso e o *Firebase Hosting* para disponibilização da aplicação. Além disso, a solução integra serviços externos como a *OpenAI Platform*, responsável pela transcrição e geração automática de resumos com base em técnicas de Processamento de Linguagem Natural, e a *Google Classroom Application Programming Interface* (GCAP), que possibilita a integração com ambientes educacionais. Toda a infraestrutura do *back-end* é hospedada na *Google Cloud Platform* (GCP), garantindo escalabilidade, disponibilidade e desempenho ao sistema.

D. Arquitetura do Sistema

A arquitetura foi desenvolvida com tecnologias modernas que trabalham em conjunto para oferecer uma plataforma *Web* segura, escalável e integrada com Inteligência Artificial, que seja capaz de transformar gravações de aulas em transcrições e resumos automáticos.

Como pode ser observado na Figura 2, a arquitetura do SAGAR segue o modelo cliente-servidor em camadas, organizado em três componentes principais: *front-end*, *back-end* e serviços em nuvem. No *front-end*, desenvolvido em *React.js*, ocorre toda a interação com o usuário, permitindo o envio de gravações, visualização de transcrições e acesso aos resumos gerados. Essa camada se comunica com o servidor por meio de requisições HTTP, garantindo uma interface dinâmica e responsiva.

No *back-end*, implementado com *Node.js* e *Express.js*, está concentrada a lógica de negócio do sistema. Essa camada recebe as requisições do *front-end*, processa os dados, gerencia autenticação e integra serviços externos, como APIs de Inteligência Artificial para transcrição e sumarização. Já os serviços em nuvem, como *Firebase* e *Google Cloud Platform*, são responsáveis pelo armazenamento, autenticação e escalabilidade da aplicação. Essa organização modular permite que o sistema seja flexível, escalável e de fácil manutenção, além de possibilitar a integração eficiente entre os diferentes componentes tecnológicos.

E. Pipeline de Processamento

O *pipeline* de processamento foi estruturado como um fluxo sequencial de transformação de dados, no qual o conteúdo capturado em formato de áudio é progressivamente convertido em um resumo textual estruturado. O processo inicia-se com a captura do áudio da aula, realizada pelo docente por meio da interface do sistema. Durante essa captura, o sinal

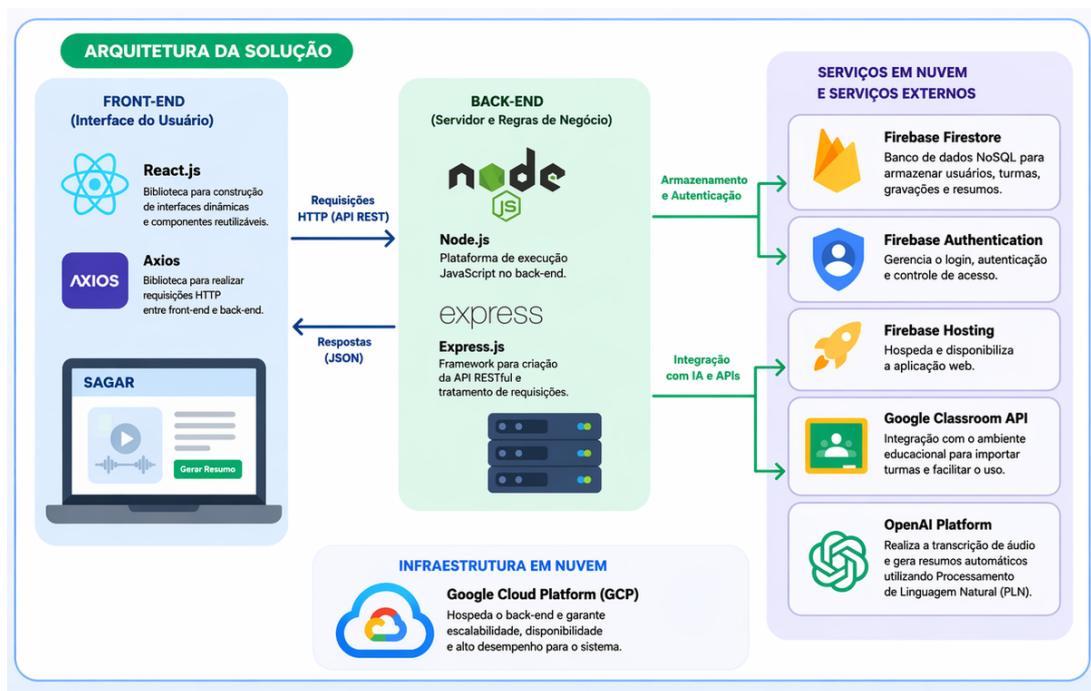


Fig. 2. Arquitetura do SAGAR

sonoro é dividido automaticamente em blocos temporais fixos de aproximadamente 9,5 minutos, caracterizando a etapa de segmentação temporal.

A segmentação do áudio em blocos independentes tem como objetivo reduzir os riscos de perda de dados e de sobrecarga de memória, além de permitir o processamento individual dos segmentos. Após sua geração, cada bloco de áudio é encaminhado à etapa de transcrição automática, na qual técnicas de reconhecimento de fala são empregadas para converter o conteúdo oral em texto. As transcrições resultantes permanecem associadas aos respectivos segmentos e são mantidas na mesma ordem temporal da gravação, preservando a sequência do conteúdo abordado durante a aula.

Na etapa seguinte, cada transcrição é processada individualmente para a geração de um resumo parcial, destinado a sintetizar as principais informações presentes no respectivo trecho da aula. Posteriormente, os resumos parciais são encaminhados à etapa de agregação, na qual são reunidos, organizados e refinados em um único conteúdo textual. Como resultado, o sistema produz um resumo final estruturado e unificado, que preserva a coerência global do conteúdo e apresenta, de forma organizada, os principais pontos abordados durante a aula. O *pipeline* representado na Figura 3 constitui o núcleo funcional da solução proposta, permitindo a automatização da transformação do conteúdo educacional oral em material textual de apoio ao estudo e à revisão.

Conforme apresentado na Figura 3, o *pipeline* de processamento organiza, de forma sequencial, as etapas necessárias para transformar uma aula gravada em áudio em um resumo textual estruturado. O processo inicia-se com a entrada do arquivo de áudio, em formatos como .mp3 ou .wav, que

é inicialmente dividido em blocos temporais fixos de aproximadamente 9,5 minutos. Cada segmento de áudio é, então, submetido individualmente à etapa de transcrição automática, na qual técnicas de reconhecimento de fala são empregadas para converter o conteúdo oral em texto, preservando-se a sequência temporal da gravação.

Na etapa seguinte, cada transcrição é processada individualmente para a geração de um resumo parcial, no qual são sintetizadas as principais informações presentes no respectivo trecho da aula. Posteriormente, os resumos parciais são reunidos, organizados e refinados na etapa de agregação, resultando em um resumo final estruturado e unificado, adequado para apoio ao estudo e à revisão dos conteúdos abordados durante a aula.

Para assegurar a reprodutibilidade do procedimento técnico, detalham-se a seguir os modelos, critérios, parâmetros e *prompts* empregados em cada etapa do fluxo de processamento de dados do SAGAR:

- 1) **Segmentação Temporal do Áudio:** a captura do sinal sonoro não ocorre de forma contínua em um arquivo único para mitigar riscos de perda de dados e sobrecarga de memória. Adota-se um critério de segmentação temporal fixa, disparado por um temporizador contínuo a cada 9,5 minutos (570.000 ms), gerando blocos discretos em formato .mp3. Ao encerramento da sessão, os segmentos são consolidados para processamento paralelo.
- 2) **Transcrição com Whisper-1:** cada segmento de áudio é enviado de forma assíncrona e paralela à API de Reconhecimento de Fala, utilizando o modelo Whisper-1 da OpenAI. O *pipeline* força o parâmetro de idioma para o português (`language: 'pt'`) e aplica temper-

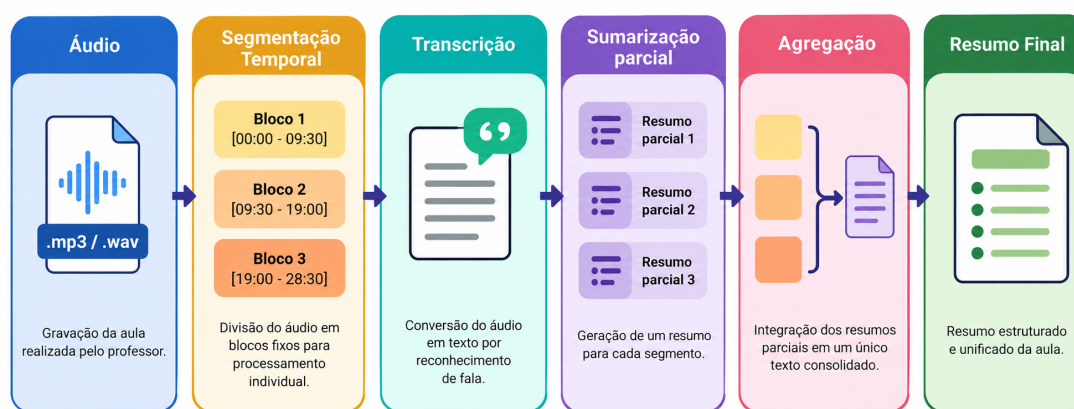


Fig. 3. Pipeline de Processamento: Áudio – Segmentação Temporal – Transcrição – Sumarização Parcial – Agregação – Resumo Final.

atura igual a 0.0, buscando reduzir a variabilidade da saída e favorecer maior consistência na transcrição. Essa configuração, contudo, não elimina a possibilidade de erros de reconhecimento decorrentes das características acústicas do áudio. O texto bruto final é gerado pela concatenação espacial dos blocos textuais resultantes. O processamento automatizado utiliza infraestrutura externa disponibilizada por meio de API. Dessa forma, uma eventual utilização do sistema com dados pessoais reais deverá considerar as condições de tratamento e retenção estabelecidas pelo provedor, bem como os requisitos da Lei Geral de Proteção de Dados Pessoais (LGPD) aplicáveis ao compartilhamento e à eventual transferência internacional de dados.

- 3) **Sumarização Parcial (Redução):** reduzir o volume textual processado nas etapas subsequentes e preservar as informações essenciais de trechos mais extensos, as transcrições com mais de 1.000 caracteres são submetidas a uma etapa intermediária de sumarização utilizando o modelo *gpt-3.5-turbo*, configurado com temperatura 0.3. Nessa etapa, o modelo recebe o seguinte *prompt* de sistema: "Você é um assistente acadêmico. Resuma o trecho da aula em, no máximo, 100 palavras, preservando os conceitos principais, definições e exemplos relevantes, sem acrescentar informações que não estejam presentes no texto original." A adoção do *gpt-3.5-turbo* nesta etapa está associada à natureza localizada e intermediária da operação, cujo objetivo é reduzir individualmente o volume textual de cada segmento antes da agregação. Diferentemente dessa etapa, restrita à condensação dos trechos transcritos, a etapa subsequente requer integração global das informações e geração de uma saída estruturada em múltiplas dimensões, sendo empregado, para essa finalidade, o *gpt-4o-mini*. Essa configuração corresponde à separação funcional das etapas do pipeline e não decorre de uma comparação experimental de desempenho entre os modelos.

- 4) **Agregação e Sumarização Estruturada Final:** os

sumários parciais são unificados em um único bloco discursivo e submetidos ao modelo *gpt-4o-mini*, configurado com temperatura 0.3 e parametrizado pelo *prompt*: "Você é um assistente acadêmico. Resuma esta transcrição de aula, mantendo a precisão acadêmica e produzindo um resumo principal de até 900 palavras." Para reduzir variações na estrutura da saída e padronizar o material produzido, utiliza-se um esquema JSON (*Structured Outputs*), que organiza a resposta em quatro dimensões delimitadas:

- **Resumo Principal:** síntese textual orientada a um limite máximo de 900 palavras, com marcações semânticas em HTML;
- **Marcos Cronológicos:** lista de acontecimentos narrados em sentenças de até 10 palavras;
- **Conceitos-Chave:** até 5 termos fundamentais definidos sinteticamente, com limite de 5 palavras por definição;
- **Exemplos Práticos:** casos aplicados discutidos em aula, limitados a 30 palavras cada.

F. Desenvolvimento da Ferramenta

Esta subseção apresenta, de forma integrada, os principais aspectos relacionados ao desenvolvimento da ferramenta, evidenciando tanto a abordagem metodológica adotada quanto as decisões técnicas que fundamentam sua implementação. Inicialmente, destaca-se o uso do desenvolvimento incremental, que possibilitou a evolução contínua do sistema por meio de entregas sucessivas e validações constantes.

Na sequência, são descritas as estruturas dos projetos de *back-end* e *front-end*, organizadas de maneira modular para favorecer a manutenção, escalabilidade e reutilização de código. Também são abordados elementos essenciais ao funcionamento da aplicação, como a configuração do servidor, a definição de rotas e o uso de *middlewares*, responsáveis pelo processamento das requisições e pela aplicação de regras de segurança.

Por fim, são apresentados os mecanismos de construção da interface, com ênfase na utilização de componentes *hooks*

e no contexto de autenticação, que centraliza o controle de acesso dos usuários. Em conjunto, esses elementos demonstram a integração entre as diferentes camadas do sistema, contribuindo para uma solução robusta, segura e eficiente.

1) *Desenvolvimento Incremental*: o desenvolvimento da ferramenta seguiu uma abordagem incremental, alinhada a práticas ágeis de Engenharia de *Software*, permitindo a evolução progressiva do sistema por meio de entregas sucessivas e funcionais. Cada incremento foi planejado para introduzir novas funcionalidades de forma controlada, possibilitando validação contínua, identificação de falhas e refinamento do código ao longo do processo. Essa estratégia favoreceu maior estabilidade da aplicação, além de permitir adaptações rápidas frente a novos requisitos ou ajustes arquiteturais. O uso de controle de versão com *Git* e repositórios remotos contribuiu para o rastreamento das mudanças, organização das etapas de desenvolvimento e integração segura entre diferentes módulos do sistema.

2) *Estrutura de Diretórios*: o *back-end* foi estruturado de forma modular, seguindo os princípios do padrão arquitetural *Model-View-Controller* (MVC), o que promove separação de responsabilidades e facilita a manutenção do código. A organização em diretórios específicos para *controllers*, *services*, rotas e integrações externas permite maior clareza na estrutura do projeto e escalabilidade da aplicação. Essa divisão também contribui para o reaproveitamento de código e para a evolução independente dos componentes, tornando o sistema mais robusto e preparado para futuras expansões.

3) *Configuração do Servidor*: a configuração do servidor foi realizada utilizando o *framework Express.js*, que fornece uma base leve e eficiente para o gerenciamento de requisições HTTP. O servidor foi preparado para lidar com diferentes tipos de requisições, incluindo tratamento de dados em formato *JavaScript Object Notation* (JSON), integração com serviços externos e aplicação de políticas de segurança, como *Cross-Origin Resource Sharing* (CORS). Além disso, foram definidos pontos de entrada e rotas principais, garantindo que o sistema possa inicializar corretamente e responder às solicitações do *front-end*. A inicialização do servidor em uma porta específica e sua execução contínua asseguram a disponibilidade da *Application Programming Interface* (API) para comunicação com os demais componentes do sistema.

4) *Rotas do Middlewares*: a definição das rotas no *back-end* segue uma organização modular, em que cada conjunto de funcionalidades (como usuários, turmas e gravações) possui seu próprio roteador. As rotas são responsáveis por receber requisições, validar os dados de entrada e encaminhar o processamento para os serviços apropriados. Complementarmente, os *middlewares* desempenham um papel fundamental ao aplicar regras transversais, como autenticação, tratamento de erros e validação de *tokens*. O uso de *middleware* para verificação de autenticação garante que apenas usuários autorizados tenham acesso a recursos protegidos, reforçando a segurança e integridade do sistema.

5) *Estrutura de Diretórios do Projeto Front-end*: o *front-end* foi desenvolvido com base em uma estrutura modu-

lar típica de aplicações *React.js*, organizada em diretórios que representam responsabilidades específicas, como páginas, componentes, serviços, *hooks* e contexto. Essa organização facilita o desenvolvimento colaborativo, melhora a legibilidade do código e permite maior reutilização de elementos da interface. A separação em camadas também contribui para a escalabilidade da aplicação, possibilitando a inclusão de novas funcionalidades sem comprometer a estrutura existente.

6) *Componentes e Hooks*: a construção da interface foi baseada no conceito de componentização, no qual cada elemento da aplicação é desenvolvido como uma unidade independente e reutilizável. Componentes representam partes específicas da interface, como formulários, painéis e botões, enquanto os *hooks* são utilizados para gerenciamento de estado e efeitos colaterais. Além dos *hooks* nativos do *React*, foram implementados *hooks* personalizados para encapsular lógicas específicas, como controle de temporização e manipulação de recursos do navegador. Essa abordagem proporciona maior organização do código, facilita a manutenção e melhora a eficiência no desenvolvimento de interfaces dinâmicas.

7) *Contexto de Autenticação*: o gerenciamento de autenticação no *front-end* foi implementado por meio de um contexto global, responsável por centralizar o estado do usuário e disponibilizar informações de autenticação para toda a aplicação. Esse contexto controla o fluxo de *login*, *logout* e persistência de sessão, integrando-se aos serviços de autenticação do *Firebase*. Além disso, permite a proteção de rotas privadas, garantindo que apenas usuários autenticados possam acessar determinadas funcionalidades do sistema. A centralização dessas informações contribui para maior segurança, consistência e facilidade de gerenciamento do estado da aplicação.

G. Procedimentos de Teste

Foram realizados testes unitários com intuito de validar individualmente os módulos do sistema, assegurando que cada função, classe ou serviço executasse corretamente o comportamento esperado. Esses testes contemplaram *endpoints* da API, integração com o banco de dados e comunicação com as APIs de Inteligência Artificial, utilizando o *framework Jest* para execução e verificação dos resultados. Além dos testes unitários, também foram realizadas quatro simulações funcionais com conteúdos das disciplinas Desenvolvimento Web 1, Desenvolvimento Web 2, Sistemas Operacionais 1 e Sistemas Operacionais 2. Essas simulações tiveram como finalidade verificar o comportamento do sistema em diferentes cenários de uso, avaliando a gravação, a transcrição, a organização das informações e a geração dos resumos. A execução regular dos testes e das simulações contribuiu para a identificação de falhas, validação do fluxo principal da aplicação e aumento da confiabilidade do sistema, alinhando-se às boas práticas de desenvolvimento de *software*.

H. Critérios de Avaliação

A avaliação foi conduzida a partir de três critérios centrais: coerência dos resumos, integridade da transcrição e

consistência do sistema. No que se refere à coerência, o sistema utiliza um processo em múltiplas etapas, no qual o áudio é segmentado temporalmente, cada segmento é transcrito individualmente e as transcrições resultantes são posteriormente sintetizadas de forma progressiva até a geração de um resumo final unificado. Essa estratégia contribui para a organização lógica das ideias e para a preservação do contexto ao longo do texto, reduzindo problemas como fragmentação temática e perda de continuidade, comuns em abordagens mais simplificadas de sumarização.

Em relação à integridade da transcrição, o sistema adota mecanismos destinados a favorecer a fidelidade entre o conteúdo oral e o texto gerado. A segmentação do áudio em blocos e sua recomposição ordenada contribui para preservar a sequência discursiva da aula, mantendo a progressão lógica do conteúdo apresentado pelo docente. Esse cuidado é essencial, pois a qualidade dos resumos depende diretamente da precisão da transcrição, sendo fundamental evitar distorções que comprometam a interpretação do conteúdo pedagógico.

Quanto à consistência do sistema, o sistema foi desenvolvido com base em uma arquitetura modular e em boas práticas de Engenharia de *Software*, o que favorece a estabilidade e a confiabilidade da aplicação. A separação entre camadas, o uso de testes unitários e o desenvolvimento incremental contribuem para a manutenção e evolução do sistema de forma controlada. Assim, nos cenários simulados, a solução apresentou execução técnico-funcional sem falhas observáveis, resultado restrito às condições da avaliação interna realizada.

A avaliação foi realizada uma avaliação interna, exploratória e preliminar do funcionamento do SAGAR, conduzida pelos próprios autores em cenários controlados. O procedimento teve como finalidade verificar o comportamento inicial do sistema e identificar eventuais falhas antes de uma avaliação independente com usuários e avaliadores externos. Portanto, os resultados obtidos nesta etapa não devem ser interpretados como evidência conclusiva de qualidade pedagógica ou de desempenho do sistema em situações reais de ensino.

A análise contemplou três dimensões: coerência dos resumos, integridade das transcrições e consistência funcional do sistema. Para as duas primeiras dimensões, foi utilizada uma escala ordinal de cinco pontos, operacionalizada da seguinte forma: (1) muito insatisfatório, com comprometimento substancial do conteúdo; (2) insatisfatório, com erros relevantes que prejudicam a compreensão; (3) parcialmente satisfatório, com preservação geral do conteúdo, mas presença de inconsistências perceptíveis; (4) satisfatório, com pequenas inadequações que não comprometem a compreensão ou o contexto; e (5) plenamente satisfatório, com adequada organização textual, preservação do contexto e fidelidade ao conteúdo de referência. Para fins descritivos, resultados iguais ou superiores a 4 foram considerados adequados dentro deste protocolo interno.

A consistência funcional foi verificada nas etapas de captura, segmentação temporal, transcrição, sumarização e geração do resultado final. Cada etapa recebeu valor 1 quando executada integralmente sem falha observável e valor 0 quando

ocorreu erro que impedisse sua conclusão. Considerando os quatro cenários simulados e as cinco etapas avaliadas em cada cenário, foram realizadas 20 verificações funcionais. Todas foram concluídas sem falhas observadas, resultando, neste conjunto específico de simulações, em taxa descritiva de execução funcional de 100%.

Como a avaliação foi realizada pelos próprios autores, sem cegamento, avaliadores independentes ou cálculo de concordância interavaliadores, os resultados estão sujeitos a viés de avaliação e devem ser interpretados exclusivamente como evidências preliminares de funcionamento em ambiente controlado. Assim, esta etapa não constitui validação externa do SAGAR. Como etapa subsequente, o projeto será submetido à apreciação de um Comitê de Ética em Pesquisa (CEP), de modo a viabilizar a realização de estudos envolvendo participantes humanos, incluindo avaliadores independentes e usuários reais da plataforma. Após a devida aprovação ética, pretende-se conduzir uma avaliação externa com aplicação de rubrica previamente definida, mensuração da concordância entre avaliadores e participação de docentes e discentes, com o objetivo de produzir evidências mais robustas sobre a qualidade das transcrições, dos resumos gerados e da utilização do SAGAR em contextos educacionais reais.

I. Limitações da Abordagem

O sistema depende fortemente de serviços externos em nuvem, como o *Firebase* para hospedagem do *front-end* e o *Google Cloud Platform* (GCP) para o *back-end*. Além disso, a plataforma utiliza APIs como a *OpenAI Platform* e a *Google Classroom API*. Essa dependência implica que qualquer indisponibilidade, alteração nas políticas de uso, mudança de preços ou limitação na taxa de requisições pode impactar diretamente a operação do sistema, representando uma restrição significativa à sua autonomia.

Como trata-se de uma plataforma *web* que depende de armazenamento em nuvem e de APIs externas, é imprescindível que os usuários tenham conexão estável com a *Internet*. Isso restringe o uso em locais com conectividade limitada ou instável e impede que o sistema funcione *offline*, mesmo parcialmente.

Ressalta-se que o sistema foi desenvolvido exclusivamente para docentes, sem suporte para discentes ou outros tipos de usuários. Essa restrição limita a escalabilidade funcional do sistema e define claramente o escopo de uso e as regras de interface, mas também reduz a complexidade do projeto.

V. RESULTADOS E DISCUSSÃO

Os resultados apresentados neste trabalho foram obtidos a partir de simulações controladas realizadas no SAGAR, contemplando quatro cenários: Desenvolvimento *Web* 1, Desenvolvimento *Web* 2, Sistemas Operacionais 1 e Sistemas Operacionais 2. As simulações foram conduzidas com conteúdos de diferentes níveis de complexidade, abrangendo desde temas introdutórios, como HTML, CSS e conceitos básicos da *web*, até conteúdos relacionados a aplicações *web* dinâmicas, integração com banco de dados, gerenciamento de memória,

concorrência, sincronização de processos e sistemas de arquivos. Nos quatro cenários, foi verificada a execução do fluxo composto pelas etapas de captura de áudio, transcrição automática, processamento do conteúdo e geração de resumos estruturados. Os resultados desta etapa são apresentados a seguir com foco na caracterização técnico-funcional do sistema nos cenários simulados.

A. Aspectos Éticos, Privacidade e Proteção de Dados

As simulações apresentadas neste estudo foram conduzidas em ambiente controlado, utilizando gravações produzidas especificamente para os cenários de avaliação do SAGAR, sem participação de turmas reais ou coleta de dados pessoais de discentes. Dessa forma, nesta etapa da pesquisa não foram realizadas intervenções ou coletas envolvendo participantes humanos. A utilização do SAGAR com dados pessoais reais deverá estar amparada por base legal adequada prevista na LGPD, definida de acordo com a finalidade do tratamento, a natureza dos dados processados, o contexto institucional e as responsabilidades dos agentes de tratamento.

Entretanto, a utilização do SAGAR em contexto educacional real poderá envolver a captação de vozes e de informações mencionadas durante aulas presenciais, caracterizando operações de tratamento de dados pessoais. Nesse cenário, a utilização do sistema deverá observar os princípios e requisitos estabelecidos pela Lei Geral de Proteção de Dados Pessoais (LGPD), incluindo finalidade, adequação, necessidade, transparência, segurança e prevenção.

Antes da gravação de aulas reais, os participantes deverão ser informados sobre a finalidade da coleta, os dados processados, o fluxo de tratamento, o eventual compartilhamento com provedores externos e os procedimentos de retenção e eliminação. Identificadores pessoais presentes nas transcrições deverão ser removidos ou pseudonimizados sempre que tecnicamente possível, e o acesso aos registros deverá ser restrito aos agentes autorizados.

Considerando a utilização de serviços externos de Inteligência Artificial, eventual aplicação do SAGAR com dados reais deverá também observar os requisitos aplicáveis à transferência internacional de dados e as políticas institucionais de proteção da informação.

B. Caracterização dos Cenários Simulados

Os quatro cenários foram definidos de modo a submeter o fluxo de processamento a conteúdos com características distintas. Em Desenvolvimento Web 1, foram utilizados conteúdos introdutórios relacionados à estruturação de documentos HTML, aplicação de estilos com CSS e conceitos básicos de funcionamento da web. Em Desenvolvimento Web 2, o conteúdo contemplou aplicações dinâmicas, integração com banco de dados, uso de *frameworks* e programação no lado servidor.

Nos cenários relacionados a Sistemas Operacionais, foram utilizados conteúdos compostos por conceitos sequenciais e interdependentes. Sistemas Operacionais 1 contemplou gerenciamento de processos, gerenciamento de memória e escalonamento de CPU, enquanto Sistemas Operacionais 2 incluiu

gerenciamento de memória, concorrência, sincronização de processos, comunicação entre processos e organização de sistemas de arquivos. Dessa forma, as simulações permitiram observar o funcionamento do mesmo fluxo técnico diante de diferentes durações de áudio, volumes textuais e conteúdos abordados.

C. Resultados Técnico-Funcionais

A Tabela II apresenta os dados obtidos durante o processamento dos quatro cenários simulados. As gravações tiveram duração entre 45 e 90 minutos e foram divididas em cinco a dez segmentos. O volume resultante das transcrições variou de aproximadamente 5.800 a 12.000 palavras, enquanto os tempos registrados para o processamento completo situaram-se entre 38 e 65 segundos.

Além das medidas de processamento, foi aplicado o protocolo de avaliação interna definido na metodologia, contemplando a coerência dos resumos, a integridade das transcrições e a consistência funcional do sistema. Para os dois primeiros critérios, utilizou-se uma escala de 1 a 5, sendo 1 correspondente ao menor nível da escala e 5 ao maior. Nos quatro cenários simulados, foi atribuído grau 5,0 às saídas analisadas, considerando a organização textual, a preservação do contexto e a fidelidade ao conteúdo utilizado nas simulações.

A consistência funcional foi verificada nas etapas de captura, segmentação temporal, transcrição, sumarização e geração do resultado final. Considerando as cinco etapas nos quatro cenários, foram realizadas 20 verificações, todas concluídas sem ocorrência de falha durante as simulações. Esses resultados correspondem, portanto, a 20 execuções concluídas em 20 verificações realizadas.

Ressalta-se que essa avaliação foi conduzida pelos próprios autores e deve ser interpretada como uma verificação interna e preliminar da solução. Não houve, nesta etapa, participação de docentes, discentes ou avaliadores externos, de modo que os resultados não permitem inferências sobre usabilidade, aceitação ou efetividade pedagógica em contexto educacional real.

D. Análise Comparativa dos Cenários

A comparação entre os cenários evidencia variação no volume de dados processados. Sistemas Operacionais 1 apresentou o menor tempo de processamento, com 38 segundos, associado a uma gravação de 45 minutos, cinco segmentos e aproximadamente 5.800 palavras transcritas. Em contrapartida, Sistemas Operacionais 2 apresentou os maiores valores de duração, segmentação e volume textual, com 90 minutos de áudio, dez segmentos e aproximadamente 12.000 palavras, correspondendo também ao maior tempo registrado, de 65 segundos.

Entre os cenários de Desenvolvimento Web, o aumento da duração de 50 para 60 minutos foi acompanhado pelo crescimento do número de segmentos, de seis para sete, e do volume da transcrição, de aproximadamente 6.500 para 8.200 palavras. O tempo registrado passou de 42 para 48 segundos. Esses dados permitem descrever a variação observada no

TABLE II
RESULTADOS DO PROCESSAMENTO NOS QUATRO CENÁRIOS SIMULADOS

Cenário	Duração do Áudio	Segmentos	Volume da Transcrição	Tempo de Processamento
Desenvolvimento Web 1	50 min	6	≈ 6.500 palavras	42 s
Desenvolvimento Web 2	60 min	7	≈ 8.200 palavras	48 s
Sistemas Operacionais 1	45 min	5	≈ 5.800 palavras	38 s
Sistemas Operacionais 2	90 min	10	≈ 12.000 palavras	65 s

processamento conforme as características das entradas utilizadas, sem que os quatro casos constituam um experimento de desempenho ou uma análise de escalabilidade.

Para além da execução do fluxo técnico, os resumos e as transcrições foram examinados internamente segundo os critérios definidos no protocolo metodológico, relacionados à preservação do contexto conceitual, à organização das informações e à manutenção da terminologia técnica. Essa análise teve caráter interno e preliminar, sendo realizada no contexto das próprias simulações.

E. Limitações da Avaliação

Os resultados obtidos devem ser interpretados à luz do caráter preliminar da avaliação e das condições controladas em que as simulações foram realizadas. Nesse estágio, os achados permitem examinar o funcionamento técnico do *pipeline* implementado e o comportamento do sistema nos cenários definidos, mas não possibilitam estabelecer conclusões sobre usabilidade, aceitação, experiência dos usuários, aprendizagem ou efetividade pedagógica em situações educacionais reais. Também não foram realizadas comparações com sistemas de referência ou *baselines*, nem empregadas métricas objetivas fundamentadas em referências humanas para mensurar a qualidade da transcrição e da sumarização. Essas limitações delimitam o alcance das conclusões e constituem aspectos relevantes para etapas posteriores de validação.

Para além da verificação do fluxo técnico de execução, as transcrições e os resumos produzidos nas simulações foram analisados segundo os critérios estabelecidos no protocolo metodológico. Nos quatro cenários avaliados - Desenvolvimento Web 1, Desenvolvimento Web 2, Sistemas Operacionais 1 e Sistemas Operacionais 2 - os textos obtiveram grau 5,0, valor máximo da escala adotada, considerando a preservação do contexto conceitual, a organização das ideias e a manutenção da terminologia técnica dos conteúdos processados. Esses resultados, entretanto, correspondem a uma avaliação interna e controlada, não devendo ser interpretados como evidência de desempenho junto a usuários reais.

A validação com docentes e discentes constitui, portanto, uma etapa subsequente da pesquisa. Por envolver participantes humanos, essa fase deverá ser realizada somente após a submissão e a aprovação do respectivo protocolo por Comitê de Ética em Pesquisa (CEP), observando-se os procedimentos aplicáveis a pesquisas dessa natureza. A partir dessa etapa, será possível incorporar instrumentos específicos para investigar aspectos como usabilidade, percepção dos participantes, utilidade do sistema e sua aplicação em contextos educacionais

reais, bem como ampliar a avaliação mediante métricas objetivas e referências produzidas por avaliadores humanos.

Outro aspecto a ser considerado refere-se à confiabilidade do conteúdo gerado. Embora o SAGAR tenha produzido sínteses consideradas coerentes nos cenários avaliados, modelos generativos não garantem fidelidade factual absoluta, podendo ocorrer omissões, reformulações imprecisas ou inclusão de informações não suficientemente sustentadas pela transcrição original. Dessa forma, os resumos devem ser compreendidos como materiais auxiliares de apoio ao ensino e à revisão de conteúdos, e não como registros oficiais das aulas. Recomenda-se, assim, a revisão docente antes de sua disponibilização aos discentes, enquanto estudos posteriores deverão avaliar de maneira sistemática a fidelidade factual e a correspondência entre as transcrições, os conteúdos originais e os resumos produzidos.

Para complementar essa análise e permitir a comparação do comportamento do sistema entre as disciplinas avaliadas, a Tabela III consolida as métricas operacionais registradas durante as simulações controladas. O tempo de processamento apresentado corresponde ao fluxo completo de ponta a ponta (*end-to-end*), executado em uma arquitetura parcialmente assíncrona. Para cada segmento de áudio de 9,5 minutos, as etapas de transcrição, realizadas pelo modelo *whisper-1*, e de redução preliminar do conteúdo textual, utilizando o *gpt-3.5-turbo*, são organizadas de modo a explorar processamento concorrente entre os segmentos, contribuindo para a redução do tempo total de execução. Após a conclusão e consolidação desses resultados intermediários, a etapa de agregação e sumarização estruturada final, realizada com o modelo *gpt-4o-mini*, é executada de forma subsequente, produzindo o resumo final correspondente ao conteúdo processado.

VI. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Os resultados apresentados neste trabalho foram obtidos a partir de simulações controladas realizadas no SAGAR, contemplando quatro cenários: Desenvolvimento Web 1, Desenvolvimento Web 2, Sistemas Operacionais 1 e Sistemas Operacionais 2. As simulações foram conduzidas com conteúdos de diferentes níveis de complexidade, abrangendo desde temas introdutórios, como HTML, CSS e conceitos básicos da web, até conteúdos relacionados a aplicações web dinâmicas, integração com banco de dados, gerenciamento de memória, concorrência, sincronização de processos e sistemas de arquivos. Nos quatro cenários, foi verificada a execução do fluxo composto pelas etapas de captura de áudio, transcrição automática, processamento do conteúdo e geração de resumos

TABLE III
MÉTRICAS DE PROCESSAMENTO E AVALIAÇÃO QUALITATIVA DO SAGAR POR CENÁRIOS SIMULADOS

Cenário	Duração do Áudio	Segmentos ¹	Volume Transcrição	Tempo Médio Pro- cessamento	Resumo Final	Avaliação qualitativa
Desenvolvimento Web 1	50 min	6	≈ 6.500 palavras 40.000 caracteres	42 s	800 palavras	Satisfatória: identificação clara dos conceitos de HTML e CSS.
Desenvolvimento Web 2	60 min	7	≈ 8.200 palavras 50.000 caracteres	48 s	780 palavras	Satisfatória: preservação das terminologias relacionadas aos <i>frameworks</i> .
Sistemas Operacionais 1	45 min	5	≈ 5.800 palavras 36.000 caracteres	38 s	820 palavras	Satisfatória: organização lógica de tópicos, como escalonamento de processos.
Sistemas Operacionais 2	90 min	10	≈ 12.000 palavras 75.000 caracteres	65 s	850 palavras	Satisfatória: preservação de conceitos complexos e interdependentes.

¹A segmentação foi configurada com janelas fixas de aproximadamente 9,5 minutos por trecho de áudio.

estruturados. Os resultados obtidos forneceram evidências preliminares sobre o funcionamento técnico-funcional do SAGAR nos cenários simulados, considerando as condições controladas adotadas nesta etapa da pesquisa.

Como trabalhos futuros, pretende-se realizar uma avaliação do SAGAR com docentes e discentes em contexto educacional real, considerando aspectos de usabilidade, qualidade dos resumos e percepção dos usuários. Essa etapa será precedida pela elaboração de protocolo específico de pesquisa e, por sua submissão à instância de apreciação ética competente antes do recrutamento e da coleta de dados dos participantes. Também serão definidos previamente os procedimentos relacionados à informação e consentimento dos participantes, proteção da privacidade, retenção e eliminação dos registros de áudio, anonimização ou pseudonimização dos dados e eventual transferência internacional de dados pessoais.

VII. APÊNDICES

Esta seção apresenta materiais complementares do SAGAR, incluindo telas da aplicação e evidências visuais dos cenários simulados, com o objetivo de ilustrar as principais funcionalidades da plataforma e complementar os resultados discutidos ao longo do trabalho.

A Figura 4 apresenta a tela destinada ao cadastro de novos usuários. A interface disponibiliza os campos “Nome”, “E-mail” e “Senha”, necessários para a criação da conta, além do botão “Cadastrar”. Para usuários que já possuem registro, também é disponibilizado um *link* de acesso à tela de autenticação.

A Figura 5 apresenta a tela de autenticação do SAGAR, por meio da qual usuários previamente cadastrados podem acessar a plataforma utilizando e-mail e senha. A interface também disponibiliza opções para recuperação de senha, criação de uma nova conta e autenticação por meio de uma conta *Google*.

Após a autenticação, o usuário é direcionado à tela inicial, apresentada na Figura 6. Essa interface constitui o ponto de acesso às principais funcionalidades da plataforma e apresenta o menu de navegação com as opções “Turmas”, “Conta” e “Sair”, além de informações introdutórias sobre a finalidade do SAGAR.

A Figura 7 apresenta a área destinada ao gerenciamento das turmas cadastradas. Nessa interface, o usuário pode visualizar

as turmas existentes, acessar seus respectivos conteúdos, iniciar gravações e realizar o cadastro de uma nova turma. O sistema também disponibiliza a possibilidade de importação de turmas por meio do *Google Sala de Aula*.

A Figura 8 detalha as opções de gerenciamento disponíveis para cada turma cadastrada. Por meio do menu acionado pelo ícone de três pontos verticais, o usuário pode editar ou excluir uma turma. Também são disponibilizadas as opções “Abrir” e “Gravar”, destinadas, respectivamente, ao acesso à turma e ao início de uma nova gravação.

A Figura 9 apresenta os registros utilizados nos cenários simulados, correspondentes às disciplinas Desenvolvimento Web 1, Desenvolvimento Web 2, Sistemas Operacionais 1 e Sistemas Operacionais 2. Para cada registro, a interface disponibiliza acesso à gravação, à transcrição e ao resumo gerado pelo sistema, permitindo consultar os diferentes artefatos produzidos durante o processamento das aulas.

A Figura 10 apresenta a interface utilizada para a gravação de uma aula, exemplificada pela disciplina Desenvolvimento Web 1. O painel central disponibiliza o controle para iniciar a captura do áudio e um cronômetro para acompanhamento de sua duração. A opção “Gravações” permite consultar os registros previamente realizados. Essa estrutura é utilizada de forma padronizada nas demais turmas cadastradas no sistema.

A Figura 11 apresenta um exemplo de resumo gerado para a disciplina Desenvolvimento Web 1. O conteúdo é organizado em seções estruturadas, como “Resumo”, “Marcos” e “Conceitos Chave”, permitindo a consulta dos principais tópicos identificados no conteúdo processado. A interface também disponibiliza as opções “Fechar” e “Copiar”, possibilitando encerrar a visualização ou copiar o material gerado.

A Figura 12 apresenta o resumo produzido para a disciplina Desenvolvimento Web 2. Nesse cenário, o conteúdo contempla temas relacionados ao desenvolvimento de aplicações web, incluindo integração com bancos de dados, utilização de *frameworks*, autenticação, consumo de APIs e *deploy* de aplicações.

A Figura 13 apresenta o resumo correspondente à disciplina Sistemas Operacionais 1. Entre os conteúdos identificados estão conceitos relacionados a processos, estados de processos, gerenciamento de memória e algoritmos de escalonamento de CPU.

Por fim, a Figura 14 apresenta o resumo produzido para a

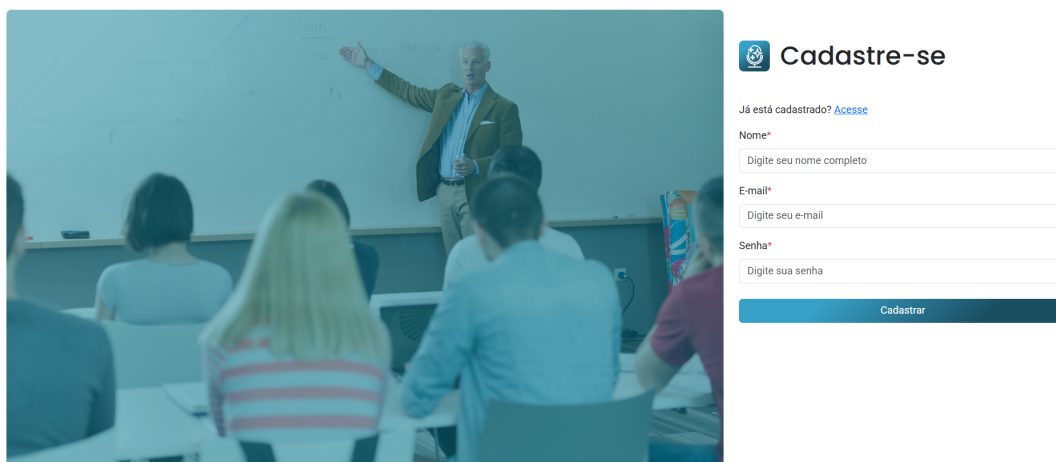


Fig. 4. Tela de cadastro de usuário no SAGAR.

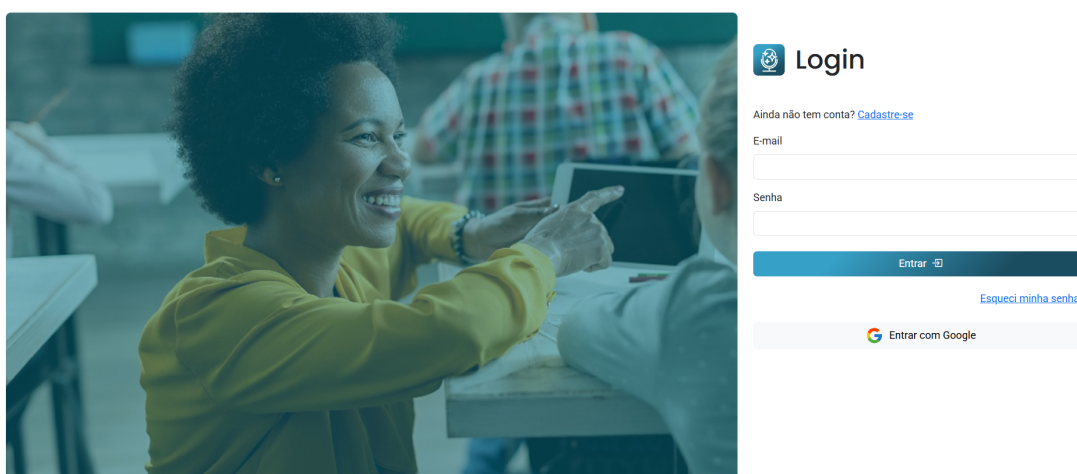


Fig. 5. Tela de autenticação do SAGAR.



Fig. 6. Tela inicial do SAGAR.

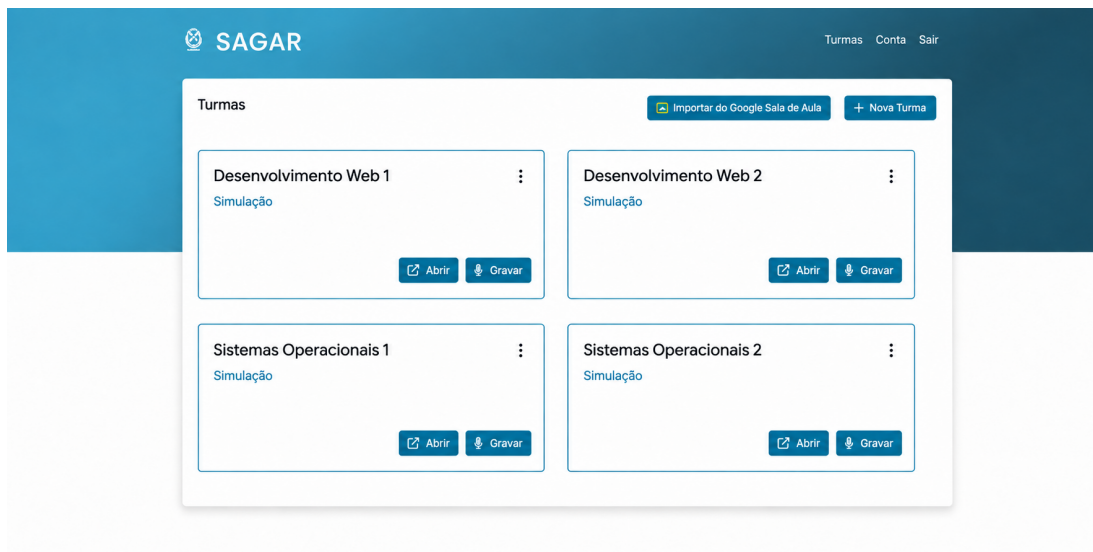


Fig. 7. Tela de turmas cadastradas no SAGAR.

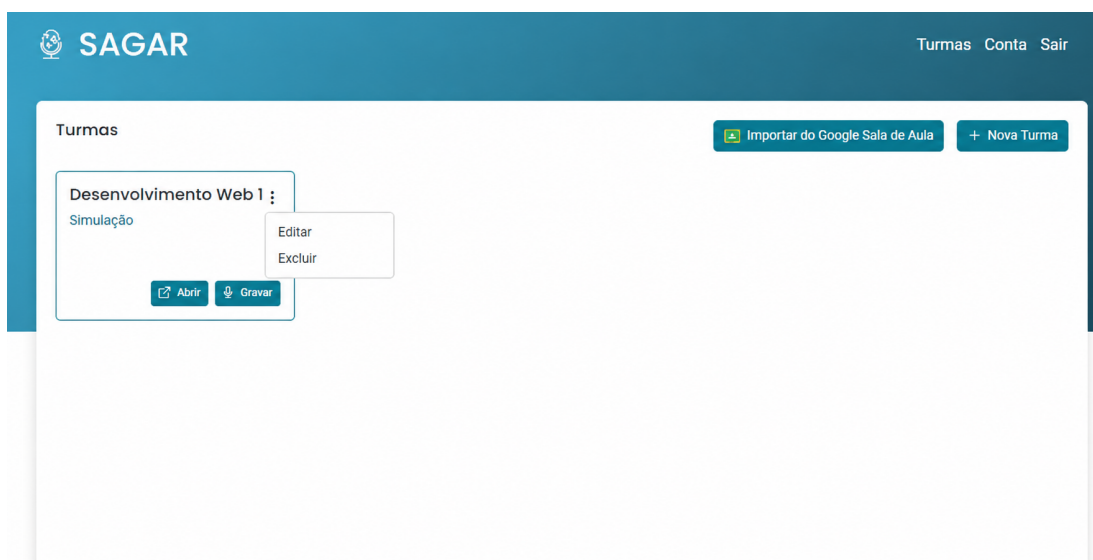


Fig. 8. Opções de gerenciamento de uma turma no SAGAR.

disciplina Sistemas Operacionais 2, contemplando conteúdos como paginação, segmentação, memória virtual, concorrência, sincronização de processos, comunicação entre processos, *deadlocks* e sistemas de arquivos.


SAGAR

[Turmas](#)
[Conta](#)
[Sair](#)

[Turmas](#) » Desenvolvimento Web 1, Desenvolvimento Web 2, Sistemas Operacionais 1, Sistemas Operacionais 2

Gravar

Título	Date	Gravação	Transcrição	Resumo	
Desenvolvimento Web 1	TER, 15/09/2026 09:10:07 - 10:00:07	Gravação	Transcrição	Resumo	⋮
Desenvolvimento Web 2	TER, 15/09/2026 15:27:33 - 16:27:33	Gravação	Transcrição	Resumo	⋮
Sistemas Operacionais 1	QUA, 16/09/2026 07:20:17 - 08:05:17	Gravação	Transcrição	Resumo	⋮
Sistemas Operacionais 2	QUA, 16/09/2026 14:20:35 - 15:50:35	Gravação	Transcrição	Resumo	⋮

Fig. 9. Registros dos cenários simulados no SAGAR.

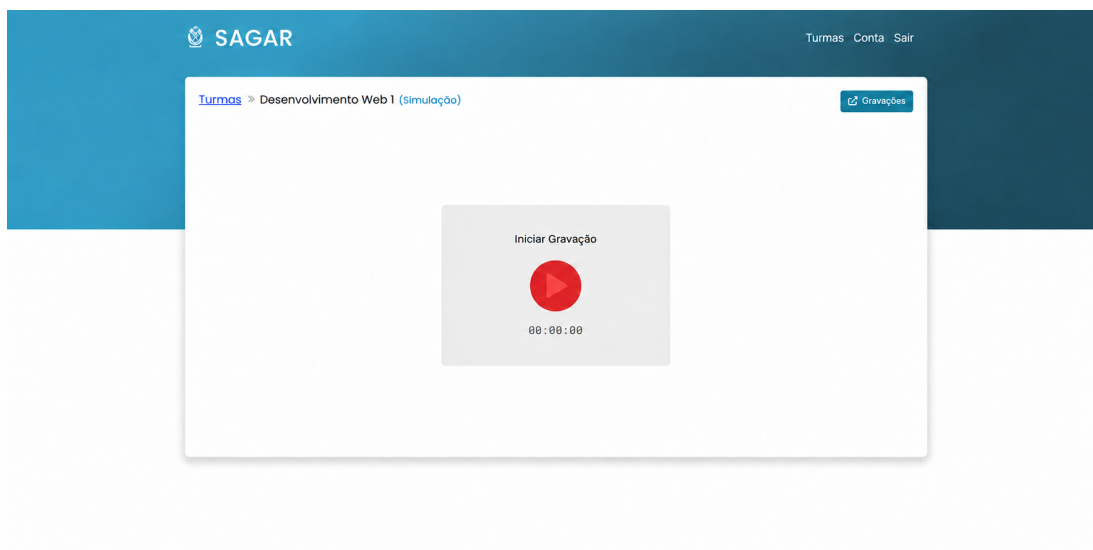


Fig. 10. Tela de gravação de áudio da disciplina Desenvolvimento Web 1.

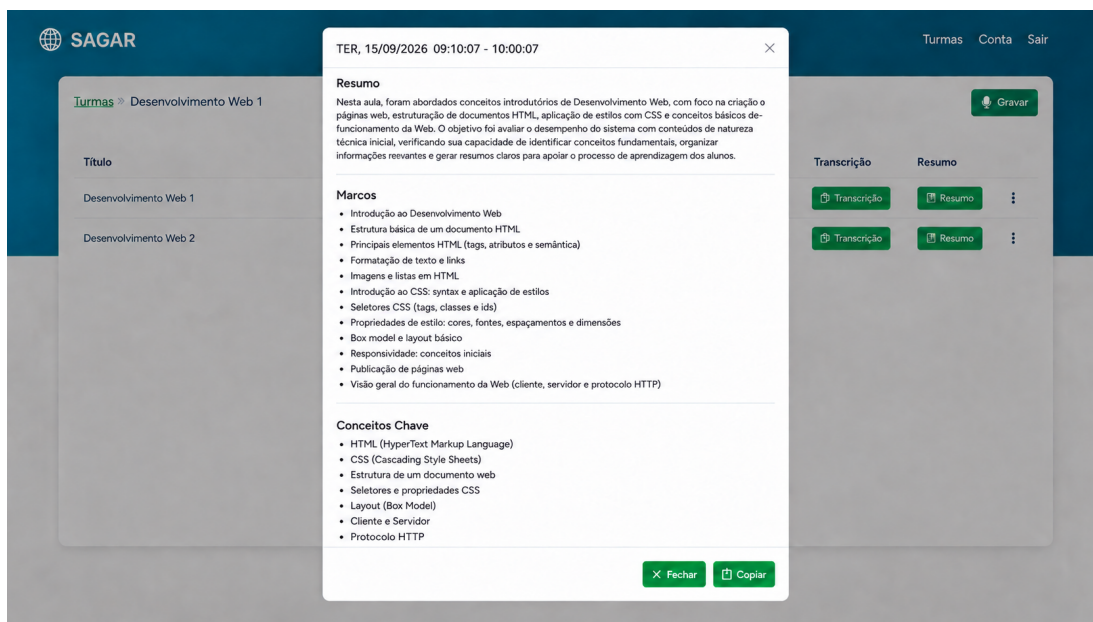


Fig. 11. Resumo gerado para a disciplina Desenvolvimento Web 1.

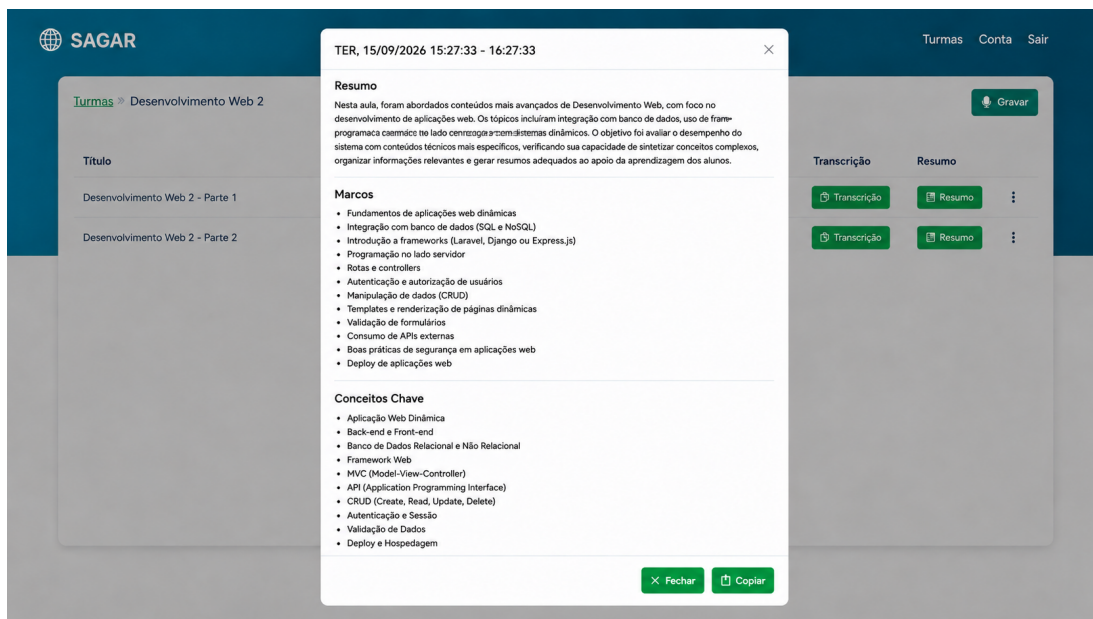


Fig. 12. Resumo gerado para a disciplina Desenvolvimento Web 2.

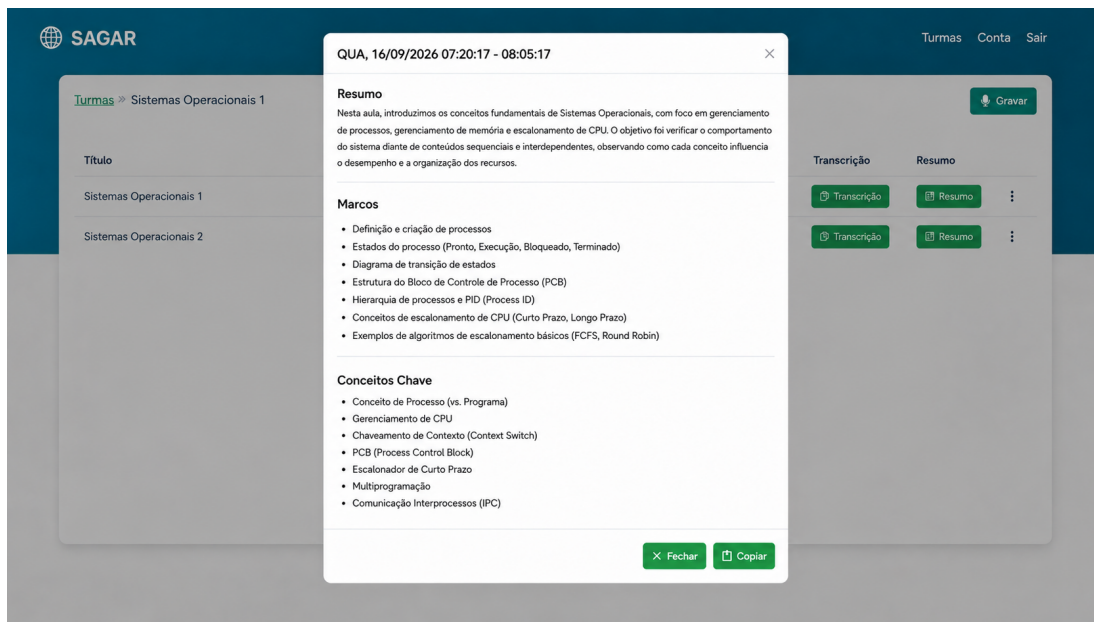


Fig. 13. Resumo gerado para a disciplina Sistemas Operacionais 1.

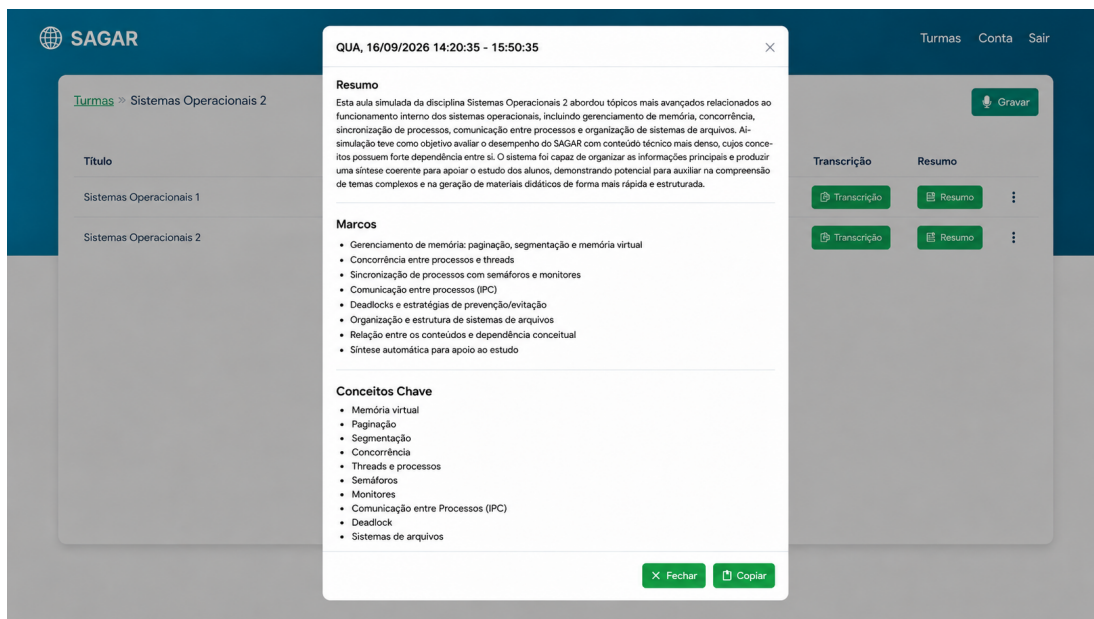


Fig. 14. Resumo gerado para a disciplina Sistemas Operacionais 2.

REFERENCES

- [1] I. D. Mienye and T. G. Swart, "A comprehensive review of deep learning: Architectures, recent advances, and applications," *Information*, vol. 15, no. 12, p. 755, 2024.
- [2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
- [3] J. Garzón, E. Patiño, and C. Marulanda, "Systematic review of artificial intelligence in education: Trends, benefits, and challenges," *Multimodal Technologies and Interaction*, vol. 9, no. 8, p. 84, 2025.
- [4] M. Y. Mustafa, A. Tlili, G. Lampropoulos, R. Huang, P. Jandrić, J. Zhao, S. Salha, L. Xu, S. Panda, Kinshuk, *et al.*, "A systematic review of literature reviews on artificial intelligence in education (aied): A roadmap to a future research agenda," *Smart Learning Environments*, vol. 11, no. 1, p. 59, 2024.
- [5] D. Zhao, "The impact of ai-enhanced natural language processing tools on writing proficiency: An analysis of language precision, content summarization, and creative writing facilitation," *Education and Information Technologies*, vol. 30, no. 6, pp. 8055–8086, 2025.
- [6] C. R. d. A. C. Barbosa, "Transformações no ensino-aprendizagem com o uso da inteligência artificial: revisão sistemática da literatura," *RECIMA21-Revista Científica Multidisciplinar-ISSN 2675-6218*, vol. 4, no. 5, pp. e453103–e453103, 2023.
- [7] I. M. de Vasconcelos, A. J. R. Ambrozio, A. B. B. Almeida, R. A. C. da Silva, G. B. de Sá, M. C. V. Segura, R. Walter, and A. A. G. R. de Oliveira, "O papel da inteligência artificial na educação: Ferramenta de suporte ou substituta?," *LUMEN ET VIRTUS*, vol. 15, no. 43, pp. 7918–7933, 2024.
- [8] N. Kala and A. Ayas, "Effect of instructional design based on cognitive load theory on students' performances and the indicators of element interactivity," *Journal of Turkish Science Education*, vol. 20, no. 3, pp. 468–489, 2023.
- [9] I. I. Salame, M. Tuba, and M. Nujhat, "Note-taking and its impact on learning, academic performance, and memory," *International Journal of Instruction*, vol. 17, no. 3, pp. 599–616, 2024.
- [10] X. Tan, G. Cheng, and M. H. Ling, "Artificial intelligence in teaching and teacher professional development: A systematic review," *Computers and Education: Artificial Intelligence*, vol. 8, p. 100355, 2025.
- [11] H. C. Araújo, M. O. Lisboa, P. H. C. Pereira, I. d. C. Freitas, M. L. B. Villela, and R. Ferreira, "Desenvolvimento de ferramentas educacionais para computação com apoio de modelos generativos de linguagem," *Revista Brasileira de Informática na Educação*, vol. 34, pp. 279–313, mar 2026.
- [12] J. M. Akazaki, L. S. R. Machado, and P. A. Behar, "MOSALA: Modelo socioafetivo baseado em Learning Analytics para auxiliar professores no acompanhamento de estudantes em um ambiente virtual de aprendizagem," *Revista Brasileira de Informática na Educação*, vol. 34, pp. 598–625, may 2026.
- [13] R. Ranchal, T. Taber-Doughty, Y. Guo, K. Bain, H. Martin, J. P. Robinson, and B. S. Duerstock, "Using speech recognition for real-time captioning and lecture transcription in the classroom," *IEEE Transactions on Learning Technologies*, vol. 6, no. 4, pp. 299–311, 2013.
- [14] A. Aher, R. Sawant, R. Patil, S. Sabat, and T. Panchal, "Mindscape: An ai-powered study companion integrating natural language processing algorithms," in *2025 3rd International Conference on Communication, Security, and Artificial Intelligence (ICCSAI)*, vol. 3, pp. 2116–2121, IEEE, 2025.
- [15] E. L. Ecker, L. W. Costa, and L. M. O. de Barros Cardoso, "Uso da lousa digital em sala de aula: benefícios e desafios," *Revista Ibero-Americana de Humanidades, Ciências e Educação*, vol. 11, no. 6, pp. 1997–2006, 2025.
- [16] I. Yamamoto and V. B. Barbeta, "Simulações de experiências como ferramenta de demonstração virtual em aulas de teoria de física," *Revista Brasileira de Ensino de Física*, vol. 23, pp. 215–225, 2001.
- [17] A. da Silva Pereira and A. F. Seoane, "A adequação das legendas automáticas do aplicativo capcut em vídeos verticais para legendagem para surdos e ensurdecidos," *Revista Linguagem em Foco*, vol. 17, no. 3, pp. 168–187, 2025.
- [18] A. J. S. d. Souza, A. B. Possato, J. P. Curtulo, D. R. d. Silva, A. C. d. Santos, T. Ferla, M. E. d. Oliveira Junior, and E. H. d. S. Goncalves, "Ia na educação: Tendências e desafios recentes," *Cuadernos de Educación y Desarrollo*, vol. 17, no. 5, pp. 1–18, 2025.
- [19] D. M. A. d. A. Pestana dos Santos, "Inteligência artificial na educação contemporânea," *Revista Processando o Saber*, vol. 17, pp. 51–73, 2025.
- [20] C. C. d. Azambuja and G. Ferreira da Silva, "Novos desafios para a educação na era da inteligência artificial," *Filosofia Unisinos*, vol. 25, no. 1, pp. 1–16, 2024.
- [21] L. Qin *et al.*, "Large language models meet nlp: A survey," *Frontiers of Computer Science*, vol. 0, no. 0, pp. 1–35, 2025.
- [22] M. M. R. Silva, "Sumarização automática de textos: desafios e avanços em técnicas de processamento de linguagem natural," 2023. Trabalho de Conclusão de Curso em Engenharia de Computação.
- [23] A. Nenkova and K. McKeown, "A survey of text summarization techniques," in *Mining Text Data*, pp. 43–76, Springer, 2012.
- [24] M. Zhang, G. Zhou, W. Yu, N. Huang, and W. Liu, "A comprehensive survey of abstractive text summarization based on deep learning," *Computational Intelligence and Neuroscience*, vol. 2022, p. 7132226, 2022.
- [25] J. Lei *et al.*, "A survey of automatic text summarization with large language model based methods," *arXiv preprint arXiv*, 2025. Survey on ATS and LLM-based summarization methods.
- [26] H. P. Luhn, "The automatic creation of literature abstracts," *IBM Journal of Research and Development*, vol. 2, pp. 159–165, Apr. 1958.
- [27] L. Sodré and H. de Oliveira, "Avaliando algoritmos de regressão para sumarização automática de textos em português do brasil," in *Anais do XVI Encontro Nacional de Inteligência Artificial e Computacional*, pp. 634–645, SBC, 2019.
- [28] P. H. Paiola, G. L. Garcia, D. S. Jodas, J. V. M. Correia, L. A. Sugi, and J. P. Papa, "Recognasumm: A novel brazilian summarization dataset," in *Proceedings of the 16th International Conference on Computational Processing of Portuguese*, pp. 575–579, 2024.
- [29] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," *arXiv preprint arXiv:1904.09675*, 2019.
- [30] F. O. Runstein, *Sistema de reconhecimento de fala baseado em redes neurais artificiais*. PhD thesis, University of Campinas, Brazil, 1998.
- [31] H. Kheddar, M. Hemis, and Y. Himeur, "Automatic speech recognition using advanced deep learning approaches: A survey," *Information Fusion*, vol. 109, p. 102422, 2024.
- [32] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proceedings of the 40th International Conference on Machine Learning*, vol. 202, pp. 28492–28518, PMLR, 2023.
- [33] M. Bain, J. Huh, T. Han, and A. Zisserman, "Whisperx: Time-accurate speech transcription of long-form audio," in *Proceedings of INTER-SPEECH 2023*, pp. 4489–4493, 2023.