ANAIS DO XXVII CONGRESSO DE COMPUTAÇÃO E TECNOLOGIAS DA INFORMAÇÃO

ISSN 2447-0767

03 A 07 DE NOVEMBRO DE 2025

https://ulbra-to.br/encoinfo

REALIZAÇÃO

SISTEMAS DE INFORMAÇÃO

CIÊNCIA DA COMPUTAÇÃO

ENGENHARIA DE SOFTWARE

APOIO

ULBRA PALMAS

SBC Sociedade Brasileira de Computação

ENCOINFO 2025
27º Congresso de Computação e Tecnologias da Informação

03 a 07 de Novembro de 2025
Palmas – Tocantins

ANAIS
26º CONGRESSO DE COMPUTAÇÃO E TECNOLOGIAS DA INFORMAÇÃO

Realização

Curso de Bacharelado em Sistemas de Informação
Curso de Bacharelado em Ciência da Computação
Curso de Bacharelado em Engenharia de Software

ISSN: 2447-0767

Nota: Os conceitos e a redação contidos nos resumos dos artigos são de exclusiva responsabilidade de seus autores, pois os mesmos foram transcritos na íntegra para esta publicação.

ENTIDADE MANTENEDORA
AELBRA Educação Superior - Graduação e Pós-Graduação S.A.

DIRETOR PRESIDENTE
Carlos Augusto Melke Filho

DIRETOR VICE-PRESIDENTE
Antônio Carlos Romanoski

CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS

Reitor
Marcelo Muller

Diretora Acadêmica
Parcilene Fernandes de Brito

Procuradora Institucional
Diêmy Sousa Freitas

Assessora da Reitoria
Alda Adriana Lima Gonçalves

Assessor de Comunicação
Karoliny Santiago Barbosa

Coordenador de Educação Continuada
Luiz Gustavo Santana

Secretária Geral
Driéli Drívela Cabral Araújo

Capelão
Lucas Prando

ISSN: 2447-0767

Bacharelado em Sistemas de Informação
Bacharelado em Ciência da Computação
Bacharelado em Engenharia de Software

Coordenadora dos Cursos
Parcilene Fernandes de Brito

Coordenadora Adjunta de Sistemas de Informação
Madianita Bogo Marioti

Coordenador Adjunto de Ciência da Computação
Fabiano Fagundes

Coordenador Adjunto de Engenharia de Software
Fabiano Fagundes

Coordenador de Estágios e de Trabalhos de Conclusão de Curso
Fabiano Fagundes

Coordenador do Projeto Informática & Sociedade
Madianita Bogo Marioti

Coordenadores da Fábrica de Software
Fábio Castro Araújo
Jackson Gomes de Souza

ENCOINFO 2025

Comissão Organizadora

Parcilene Fernandes de Brito, D.ra
Douglas Aquino Moreno, Esp.
Fabiano Fagundes, M.e
Fábio Castro Araújo, Esp.
Fernanda Pereira Gomes, Esp.
Jackson Gomes de Souza, M.e
Madianita Bogo Marioti, M.e

Arte e Capa
Jackson Gomes de Souza

Diagramação
Mario Matheus Pombal Rebello

Site do Evento
Fábrica de Software

Fábrica de Software
Fábio Castro Araújo
Jackson Gomes de Souza
Davi Pinheiro Teixeira
Guilherme Domiciano Silva

ISSN: 2447-0767

EQUIPE EDITORIAL

COMITÊ TÉCNICO

Anderson Iwazaki - Universidade de São Paulo
Bruno S. C. M. Vilar - Belvo
Débora Araújo - Universidade Federal do Vale do São Francisco
Edilson Ferneda - Universidade Castelo Branco
Heloise Acco Tives - IFPR campus Palmas
Jéssyka Vilela - Universidade Federal de Pernambuco
Leandro A. Pasa - Universidade Tecnológica do Paraná
Leandro Maciel Almeida - Universidade Federal de Pernambuco
Luciano de Souza Cabral - Instituto Federal de Pernambuco

ISSN: 2447-0767

SUMÁRIO

PALESTRANTE

Leonardo Tórtoro Pereira

Palestra: Jogos e carreira acadêmica: é possível?

Mini CV: Os jogos constituem uma área de pesquisa relevante não apenas na Computação, mas também em diversas outras disciplinas. Nesta palestra será apresentada a importância de uma formação acadêmica sólida e integrada às dimensões de ensino, pesquisa e extensão. O público poderá conhecer a trajetória de um grupo que, partindo de uma linha de pesquisa pouco reconhecida nas universidades estaduais de São Paulo, consolidou-se como uma equipe produtiva e de referência nacional, atualmente liderada por docentes engajados e atuantes.

Nome

Palestra: descrição

Mini CV: descrição

ISSN: 2447-0767

MINICURSOS

Treinar e executar modelos de IA com javascript

Van Neves

Introdução ao desenvolvimento de jogos 3D na Unity 6

Stefan Lucas

CyberSecAI: Introdução a Cyber Segurança com auxílio de inteligência artificial contextualizada

Mario Matheus Pombal Rebello e Marianne Lacerda Dutra e Lucas Casagrande

Aplicação FullStack com Vite(React) e FastAPI

Davi Teixeira, Guilherme Domiciano, Nicole França

Vibe Coding 101: Introdução a IA para devs

Alexandre Kavalerski

Testes Unitários com Java

Iury Felipe

Utilização de DBeaver para gerenciamento de banco de dados relacionais

Sidevalto Cipriano Capone

Agentes de IA na Prática com Langchain

Luis Fernando e Geisbelly

Alfabetização em segurança da informação: Construa sua defesa

Yasmin

ARTIGOS COMPLETOS

ISSN: 2447-0767

StudyFlow–Voice: Respostas por Voz com Whisper e Avaliação Semântica Inicial via IA em Cards de Estudo

Carlos Eduardo Costa Aleixo dos Santos[1], Jackson Gomes de Souza[1]

[1]Departamento de Computação ULBRA Palmas -
Av. Joaquim Teotônio Segurado, 1501 - Plano Diretor Sul, Palmas - TO, 77019-900

carlos.aleixo@rede.ulbra.br, jackson.souza@ulbra.br

Resumo. *Este trabalho apresenta o StudyFlow–Voice, um módulo experimental para plataformas de estudo baseado em flashcards, que permite ao aluno responder verbalmente em vez de digitar. A resposta em áudio é transcrita usando o modelo Whisper, comparada semanticamente com a resposta esperada por meio de embeddings gerados com Sentence-Transformers, e avaliada por um modelo generativo (Gemini Pro), que fornece feedback textual pedagógico. O sistema foi implementado como prova de conceito e validado com dados simulados. Uma avaliação formal está planejada, com foco em acurácia da transcrição, qualidade do julgamento semântico e experiência do usuário.*

Palavras-chave: aprendizagem ativa; respostas por voz; Whisper; IA generativa; avaliação semântica.

Abstract. *This work presents StudyFlow–Voice, an experimental module for flashcard-based study platforms that allows students to answer verbally instead of typing. Audio responses are transcribed using the Whisper model, semantically compared to expected answers through embeddings generated with Sentence-Transformers, and evaluated by a generative model (Gemini Pro), which provides pedagogical textual feedback. A proof of concept has been implemented and tested with simulated data. A formal evaluation is planned, focusing on transcription accuracy, semantic judgment quality, and user experience.*

Keywords: active learning; voice answers; Whisper; generative AI; semantic evaluation.

1. Introdução

A utilização de tecnologias de inteligência artificial (IA) no contexto educacional tem se intensificado nos últimos anos, especialmente em aplicações voltadas para a avaliação formativa e o fornecimento automático de *feedback*. Essas abordagens buscam reduzir a sobrecarga de trabalho docente e permitir intervenções mais frequentes e personalizadas durante o processo de aprendizagem.

Recentemente, pesquisadores têm demonstrado interesse crescente em sistemas de reconhecimento automático de fala (*Automatic Speech Recognition – ASR*), uma vez que a fala representa um meio natural de comunicação entre as pessoas [YU; DENG 2015]. Inicialmente restritos a sistemas simples, os mecanismos de ASR evoluíram para modelos capazes de processar linguagem natural com elevada fluência [JUANG; RABINER 2005]. O reconhecimento de fala é considerado um desafio técnico devido à variabilidade dos sinais sonoros e à necessidade de adaptação a diferentes contextos e sotaques [YU; DENG 2015]. Com o avanço de métodos de aprendizado profundo e pré-treinamento não supervisionado, como o modelo *Wav2Vec2.0*, a precisão desses
sistemas aumentou consideravelmente [BAEVSKI et al. 2020]. Entre os modelos recentes, destaca-se o *Whisper*, desenvolvido pela *OpenAI*, que apresenta robustez diante de sotaques e ruídos, além de suporte multilíngue, o que o torna promissor para aplicações educacionais em contextos diversos [RABELO 2022].

Em estratégias de aprendizagem ativa, espera-se que o estudante participe de forma mais autônoma e significativa. Nesses contextos, o uso de respostas orais pode tornar a prática mais fluida e acessível. A exigência de digitação, especialmente em dispositivos móveis ou para alunos com limitações de acessibilidade, pode introduzir barreiras que desestimulam a participação. A utilização da voz como canal de resposta favorece a expressão de ideias e contribui para o foco na compreensão de conceitos, e não apenas na reprodução literal de informações.

Entretanto, ainda são escassas as soluções voltadas à avaliação de respostas orais curtas em plataformas educacionais baseadas em flashcards. Muitas das iniciativas atuais concentram-se em tarefas discursivas mais amplas, como narrativas ou conversas abertas. Além disso, há desafios relacionados à latência da resposta, à consistência do julgamento semântico por parte dos modelos de linguagem e à experiência do usuário no uso contínuo dessas ferramentas.

Este trabalho propõe o módulo *StudyFlow–Voice*, desenvolvido para permitir respostas orais em atividades de revisão com flashcards. A funcionalidade implementa um fluxo composto por transcrição automática com *ASR* (modelo *Whisper*), análise semântica da resposta com base em embeddings e retorno textual gerado por modelo de linguagem (*Gemini Pro*). As principais contribuições incluem: (i) o desenho de um *pipeline* completo para captação de áudio, transcrição, análise e *feedback*; (ii) a elaboração de estratégias de *prompting* para orientar o julgamento pedagógico do modelo; e (iii) a integração da solução com a estrutura de cards da plataforma *StudyFlow*.

O escopo deste trabalho restringe-se à funcionalidade de resposta por voz e avaliação automatizada. Os demais módulos da plataforma, como geração de *flashcards* a partir de documentos ou recomendação de conteúdos, são apenas contextualizados e não fazem parte da implementação apresentada.

2. Fundamentos e Trabalhos Relacionados

A entrada por voz em plataformas de estudo tem ganhado relevância devido ao seu potencial para ampliar a acessibilidade e tornar a interação mais natural. Esse recurso beneficia especialmente usuários com dificuldades motoras ou que utilizam dispositivos móveis, ao mesmo tempo em que favorece o fluxo conversacional durante a aprendizagem. No entanto, a entrada oral traz desafios técnicos, como a presença de ruídos de fundo, variação de sotaques e hesitações típicas da fala espontânea. Tais fatores afetam diretamente a qualidade da transcrição e, por consequência, a avaliação automatizada.

Nesse contexto, os sistemas de reconhecimento automático de fala (*ASR*) tornaram-se essenciais. Além de transcrever a fala em texto, é necessário considerar a natureza espontânea das respostas verbais dos alunos, que podem conter repetições,
reformulações ou pausas. A literatura aponta que esses elementos dificultam a análise direta da fala transcrita e exigem estratégias de pós-processamento para viabilizar avaliações mais confiáveis [YU; DENG 2015]. Modelos modernos como o *Whisper*, da *OpenAI*, têm demonstrado avanços nesse sentido ao oferecer maior robustez diante de variações linguísticas e ambientais [RABELO 2022].

Após a transcrição, a etapa de avaliação semântica pode ser realizada com auxílio de modelos de linguagem de larga escala (*LLMs*). Esses modelos são capazes de comparar a resposta do aluno com a resposta esperada em termos conceituais, indo além da correspondência literal de palavras. Estudos recentes têm utilizado rubricas pedagógicas como guia para estruturar o

ISSN: 2447-0767

julgamento automatizado, incluindo elementos como explicações, reforço de acertos e apontamento de lacunas. Essa abordagem aproxima-se das práticas de *feedback* formativo recomendadas em ambientes educacionais.

Em sistemas baseados em *flashcards*, a possibilidade de fornecer retorno imediato após uma resposta oral representa uma forma de prática ativa com correção contínua. A literatura em psicologia da aprendizagem sugere que esse tipo de prática, quando acompanhada de *feedback* claro, promove maior retenção do conteúdo e melhora a metacognição. A entrada por voz, quando bem implementada, pode reduzir atritos e tornar a atividade mais fluida para o aluno.

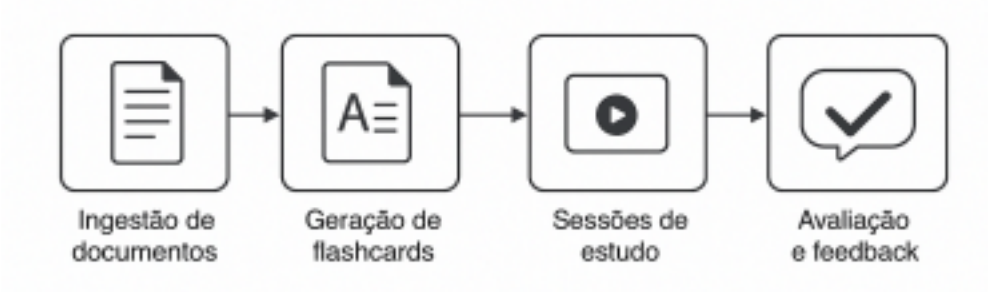
Além disso, sistemas que avaliam respostas orais já são objeto de pesquisa recente. O trabalho de Balaji et al. investiga um *pipeline* que conecta *ASR* e *LLM* para pontuar e gerar *feedback* em narrativas orais produzidas por crianças. Os autores demonstram que o sistema foi capaz de replicar com razoável coerência os julgamentos realizados por avaliadores humanos [BALAJI et al. 2024]. Outro estudo relevante é o de Wang e Chen, que conduziram um experimento controlado com estudantes de inglês como segunda língua. Os resultados indicam que o uso de reconhecimento de fala assistido por IA pode melhorar a compreensão auditiva, reduzir a ansiedade e aumentar o engajamento dos alunos [WANG; CHEN 2025].

Por fim, embora promissores, os sistemas baseados em *LLMs* ainda enfrentam limitações reconhecidas. Entre os principais desafios estão as chamadas “alucinações” dos modelos, respostas aparentemente corretas, mas sem fundamento, além do risco de enviesamento nos julgamentos e da variação de qualidade nas transcrições em função do contexto acústico. Esses pontos são discutidos por Zheng et al., que alertam para a importância de validação contínua, transparência nos critérios de avaliação e cuidados éticos na adoção dessas tecnologias [ZHENG et al. 2023].

3. Visão Geral do *StudyFlow*

O *StudyFlow* é uma plataforma educacional voltada para estudo ativo por meio de *flashcards* gerados automaticamente a partir de materiais diversos, como artigos científicos, apostilas ou anotações. O sistema organiza o processo de revisão em ciclos, nos quais o usuário interage com um conjunto de *cards* contendo uma pergunta e a resposta esperada. O objetivo pedagógico é promover a prática da recordação ativa, consolidando o entendimento de conteúdos por meio de tentativas sucessivas de resposta, com base em técnicas de aprendizagem baseadas em evidências.

O *pipeline* macro da plataforma pode ser descrito em quatro etapas principais: (i) ingestão de documentos (como PDFs, textos ou notas), (ii) geração automática de *flashcards* a partir do conteúdo



processado, (iii) realização de sessões de estudo com apresentação dos *cards* e coleta de respostas do usuário (por texto ou por voz) e (iv) avaliação da resposta, fornecimento de *feedback* e registro da tentativa no histórico do estudante, como mostra a figura 1.

Figura 1 – *Pipeline* da arquitetura do módulo de respostas

A Figura 1 ilustra o fluxo completo da plataforma, destacando o papel de cada etapa no ciclo de aprendizagem. O módulo *StudyFlow–Voice* se insere na terceira e quarta etapas desse fluxo. Durante uma sessão de estudo, ao receber um *card*, o usuário pode optar por ditar sua resposta. O áudio é capturado e processado pelo *pipeline* de voz, que inclui transcrição com o modelo *Whisper*, análise semântica com base na comparação entre *embeddings* da resposta do aluno e da resposta esperada, e geração de *feedback* textual com o apoio de um modelo generativo (*LLM*). O retorno ao usuário pode incluir reforço positivo, identificação de lacunas conceituais ou sugestões de revisão.

Cada interação com um *card* envolve três artefatos principais: (i) o próprio *card*, contendo pergunta e resposta esperada; (ii) a resposta do usuário, em formato textual (digitada ou transcrita a partir do áudio); e (iii) a avaliação automatizada, que inclui o julgamento semântico e o *feedback* gerado. Esses elementos são vinculados no banco de dados ao histórico do usuário, permitindo rastreabilidade das tentativas, análise longitudinal do desempenho e futuras estratégias adaptativas.

Este trabalho concentra-se exclusivamente no módulo de respostas orais, que visa reduzir atrito de entrada, favorecer fluidez na prática e ampliar a acessibilidade da plataforma, sem alterar a lógica pedagógica já estabelecida no ciclo de *cards*.

4. Arquitetura da Solução de Respostas por Voz

4.1. Fluxo ponta a ponta

O módulo de respostas por voz foi implementado a partir de uma arquitetura linear, com estágios claramente definidos desde a captura do áudio até a entrega do feedback ao estudante. Esse fluxo completo pode ser visualizado na **Figura 2**, que resume as etapas e componentes principais do sistema.

ISSN: 2447-0767

Figura 2 – *Pipeline* da arquitetura do módulo de respostas

A Figura 2 apresenta o funcionamento do pipeline de ponta a ponta. O processo se inicia no aplicativo do *StudyFlow*, que registra a resposta oral do usuário por meio da interface Web ou mobile. O áudio capturado é então enviado para o servidor, onde ocorre a transcrição utilizando o modelo Whisper, com suporte ao idioma português (variante brasileira).

A transcrição bruta passa por uma etapa de normalização textual, que aplica pontuação e remove elementos disfluentes (como interjeições ou palavras de preenchimento), com o objetivo de melhorar a legibilidade e a qualidade semântica da resposta. Em seguida, é montado um *prompt* estruturado, contendo a pergunta original, a resposta esperada (pré-cadastrada), a transcrição obtida e as instruções para julgamento pedagógico. Esse *prompt* é enviado a um serviço de geração de texto baseado em *IA*, que responde com uma estrutura *JSON* contendo os critérios avaliativos e um *feedback* textual.

O *JSON* retornado é validado estruturalmente antes de ser armazenado, garantindo que as chaves esperadas estejam presentes e que os dados possam ser interpretados pela aplicação. Por fim, o *feedback* é integrado ao card e apresentado ao usuário, junto com a nota ou classificação correspondente. O histórico da tentativa, incluindo áudio, transcrição, avaliação e parâmetros utilizados, é salvo no banco de dados vinculado ao perfil do estudante.

4.2. Camada *ASR* (*Whisper*)

O componente de reconhecimento de fala utiliza o modelo *Whisper*, da *OpenAI*, executado via *API* interna. O sistema é configurado para o idioma português brasileiro (pt-BR), com detecção automática de fim de fala (*endpointing*). A qualidade das transcrições foi avaliada informalmente em diferentes níveis de ruído (*SNR*), com resultados satisfatórios em condições padrão de uso com microfones embutidos em *laptops* e *smartphones*. Situações com sobreposição de fala ou ruído de fundo intenso impactam negativamente a fidelidade da transcrição, como também apontado por Gong et al. [Gong et al. 2023] e Trabelsi et al. [Trabelsi et al. 2024]. Avaliações mais recentes mostram que o desempenho do *Whisper* em português brasileiro pode variar de acordo com o perfil do falante, presença de sotaques e condições acústicas específicas [Kulkarni et al. 2024].

Após o reconhecimento, é aplicada uma correção leve à transcrição, que inclui inserção de pontuação, remoção de *fillers* (por exemplo, "éé", "tipo", "aham") e ajuste de capitalização. Essas transformações não alteram o conteúdo da resposta, mas visam facilitar o processamento posterior pelo modelo de linguagem, sem introduzir inferências adicionais.

4.3. Modelo de dados

A arquitetura de dados do módulo de voz foi desenhada para garantir rastreabilidade, eficiência na avaliação e possibilidade de análises futuras. A estrutura central envolve entidades que representam os principais elementos da interação entre aluno e sistema. A relação entre essas entidades é apresentada na Figura 3.

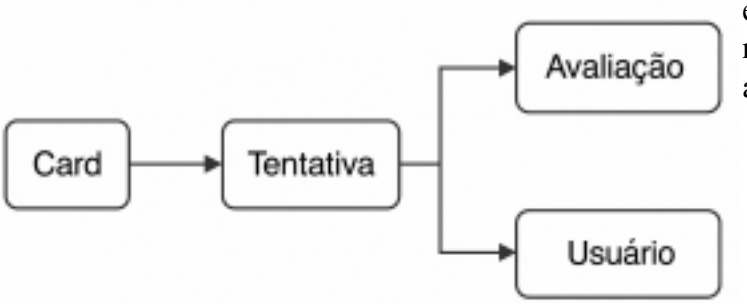


Figura 3 – Diagrama exibindo o fluxo dos dados

A Figura 3 ilustra o fluxo de dados entre os componentes centrais da aplicação. O objeto *Card* representa a unidade pedagógica, contendo a pergunta, a resposta esperada e metadados como disciplina, nível e origem. Cada tentativa de resposta do aluno é registrada na entidade Tentativa, que associa o áudio gravado, a transcrição gerada, os parâmetros utilizados no *prompt* e o tempo de resposta.

A avaliação da tentativa é armazenada em uma estrutura própria, Avaliação, que guarda o *JSON* retornado pelo modelo de linguagem, os critérios aplicados, a nota atribuída (quando presente) e o *feedback* textual. Por fim, todas essas informações são vinculadas ao Histórico do aluno, permitindo o acompanhamento longitudinal do desempenho e a geração de relatórios ou intervenções pedagógicas personalizadas.

ISSN: 2447-0767

5. Interface e Experiência do Usuário

A interface de estudo com voz foi projetada para manter a fluidez do ciclo já existente de perguntas e respostas do *StudyFlow*. Ao selecionar um card, o usuário pode optar por responder utilizando a voz, por meio de um botão dedicado à gravação. A Figura 4 apresenta essa interface com o botão visível ao usuário.

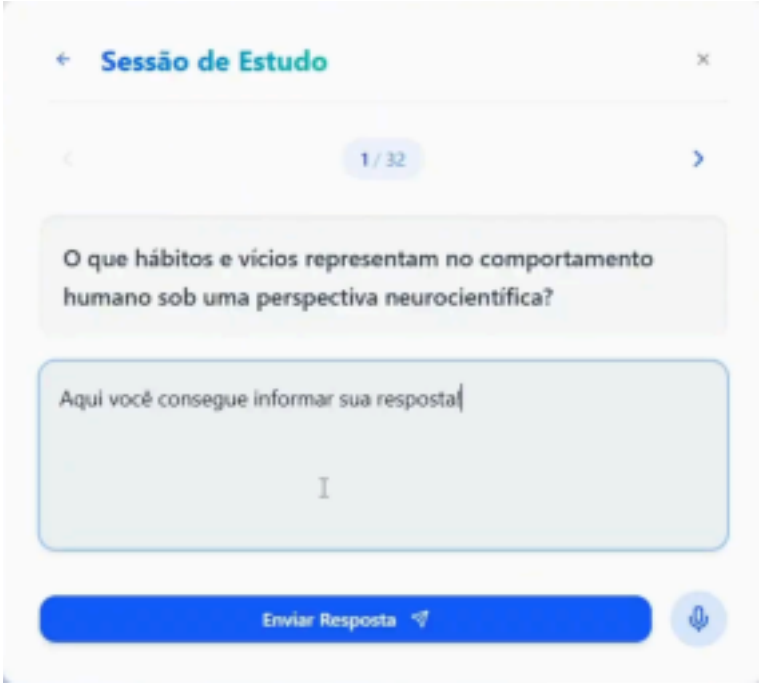


Figura 4 – Interface do card com botão de gravação

Como mostra a Figura 4, o botão de gravação é destacado na interface, proporcionando uma experiência intuitiva e acessível. Durante a captação do áudio, são exibidos indicadores visuais como ondas sonoras em movimento e ícones dinâmicos, que reforçam o estado de gravação ativa. A Figura 5 exemplifica esse momento visual.

Figura 5 – Interface do card com indicador visual ativo

A Figura 5 demonstra como a interface comunica ao usuário que a gravação está em andamento, reduzindo incertezas e aumentando a confiança na ferramenta. Após o término da fala, a transcrição gerada pelo modelo *ASR* (*Whisper*) é exibida na tela. O aluno tem a possibilidade de revisar o texto, confirmar ou editar trechos antes de prosseguir. Essa etapa permite que o usuário tenha controle sobre a entrada que será enviada para avaliação, o que contribui para maior precisão e engajamento.

Após a submissão da resposta, o sistema exibe o *feedback* avaliado com base na análise semântica. A Figura 6 mostra um exemplo da transcrição confirmada juntamente com o resultado da avaliação.

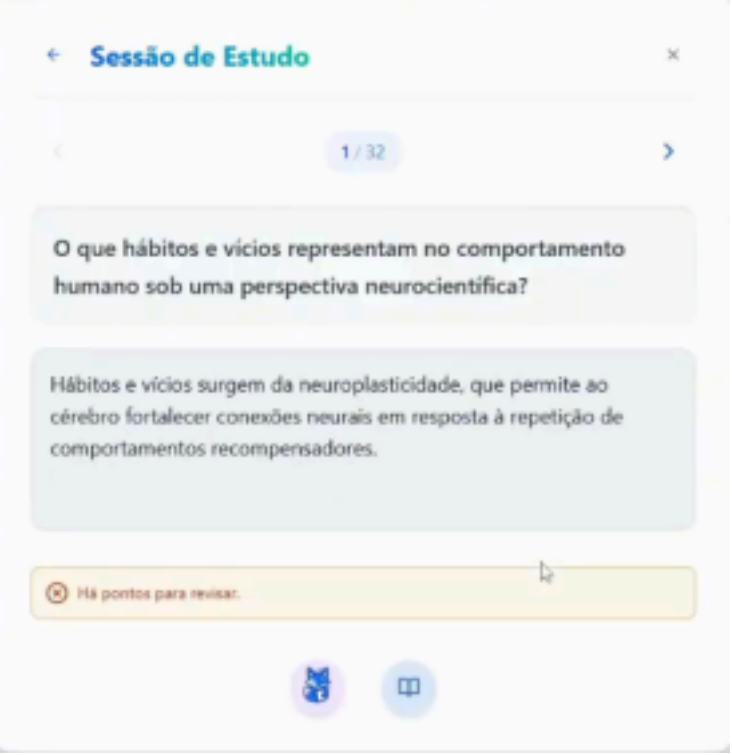


Figura 6 – *Card* mostrando resposta avaliada e a transcrição confirmada

Na sequência, a Figura 7 apresenta o feedback estruturado, que detalha o nível geral de compreensão, os critérios aplicados e eventuais lacunas conceituais detectadas na resposta do aluno.

ISSN: 2447-0767

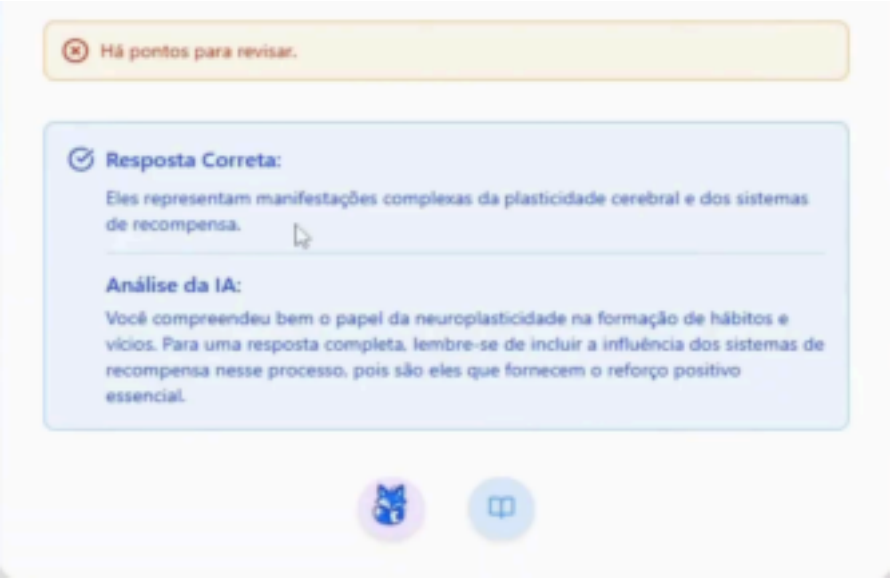


Figura 7 – *Card* mostrando *feedback* estruturado

Esse retorno ao estudante pode incluir sugestões de *follow-up* e orientações para revisão do conteúdo. A interface também contempla critérios de acessibilidade, permitindo o uso via teclado, exibição de legendas em tempo real e reenvio rápido em caso de falhas. Tais recursos foram implementados com foco em atender diferentes perfis de usuários, incluindo pessoas com deficiência motora ou limitações no uso de teclado em dispositivos móveis.

Futuramente, planeja-se a incorporação de indicadores de qualidade da transcrição, como barras de confiança por trecho para orientar o aluno sobre trechos potencialmente ambíguos ou mal interpretados pelo sistema. Esse tipo de sinalização visa contribuir para uma experiência mais transparente e informativa.

6. Experimentos Técnicos (prova de conceito)

A prova de conceito foi realizada em ambiente controlado, utilizando o modelo *Whisper large-v2*, da *OpenAI*, para transcrição automática de fala em português brasileiro. A etapa de avaliação semântica foi executada com o modelo *Gemini Pro* (versão de abril/2025), acessado por meio de *API* externa. Os parâmetros principais empregados nos *prompts* incluíram temperatura igual a 0.2 e limite de 400 *tokens* para a resposta. Esses valores foram definidos com o intuito de obter respostas mais determinísticas e adequadas ao propósito avaliativo.

Os principais parâmetros e métricas técnicas obtidas ao longo dos testes estão resumidos na Tabela 1.

Parâmetro / Métrica	**Valor**
Modelo *ASR*	*Whisper large-v2*
Modelo *LLM*	*Gemini Pro* (abr/2025)
Idioma	pt-BR
Temperature (*LLM*)	0.2
Max Tokens (*LLM*)	400

Latência total (média)	6–10 s
Latência *ASR* (média)	1,8 s
Latência *LLM* (média)	2,5–4 s
Custo por tentativa	< US$ 0,005
Taxa de *JSON* válido	100%
Taxa de edição da transcrição	12%
Repetibilidade (qualitativa)	Alta (com *prompting* e temperatura baixa)

Tabela 1 – Parâmetros técnicos avaliados

Como apresentado na Tabela 1, os tempos de resposta foram compatíveis com uma experiência interativa fluida, mesmo considerando o uso de modelos em nuvem. A latência medida entre o início da submissão da fala e o recebimento do *feedback* final apresentou variação média entre 6 e 10 segundos, considerando transcrições de até 10 segundos de fala. O tempo médio de transcrição com *Whisper* foi de 1,8 s, enquanto o tempo de resposta do modelo generativo situou-se entre 2,5 s e 4 s, dependendo da complexidade do *prompt*. O custo estimado por tentativa (considerando uso de *API* externa para *LLM*) foi inferior a US$ 0,005 em média.

Entre as tentativas registradas, observou-se uma taxa de geração de *JSON* válido de 100% nos testes, assegurada por validação sintática e por um esquema de *fallback* com reenvio automático em caso de falha. A taxa de edição manual da transcrição antes do envio definitivo foi de aproximadamente 12%, indicando que a maior parte das transcrições geradas foi aceita diretamente pelos usuários sem alterações.

Para observar o impacto de decisões de engenharia no desempenho e na consistência da avaliação, foram conduzidos testes exploratórios com variações no *prompt* e no processamento. Entre os principais ajustes avaliados, destacam-se:

- Presença ou ausência de exemplos (*few-shot prompting*): a inclusão de exemplos concretos no prompt mostrou-se útil para padronizar a estrutura do *feedback*, especialmente em respostas mais abertas.
- Limpeza de *fillers* e disfluências ligada ou desligada: a remoção de palavras de preenchimento e hesitação (como “ééé”, “tipo”, “aham”) impactou positivamente a clareza da entrada, reduzindo ambiguidades no julgamento semântico.

A repetibilidade da avaliação foi analisada de forma qualitativa, observando a consistência das respostas do modelo em tentativas sucessivas com o mesmo *input*. Nas configurações de temperatura reduzida e com *prompting* estruturado, observou-se baixa variabilidade entre execuções, o que favorece o uso pedagógico da ferramenta. No entanto, uma avaliação formal da confiabilidade ainda está planejada para fases posteriores.

7. Discussão

A integração de respostas por voz ao ambiente de estudo apresenta benefícios claros em termos de acessibilidade, fluidez e engajamento. Ao permitir que o aluno responda oralmente

ISSN: 2447-0767

aos *flashcards*, reduz-se o atrito causado pela digitação, especialmente em dispositivos móveis ou em situações em que o uso do teclado é inviável. Essa abordagem favorece uma interação mais natural, próxima da linguagem cotidiana, e estimula a verbalização ativa, importante para a consolidação do conhecimento.

Além disso, a geração imediata de *feedback* semântico contribui para o processo de aprendizagem formativa. Em vez de depender apenas de correções literais, o sistema oferece avaliações que reconhecem parcialmente corretos, apontam lacunas conceituais e sugerem reforços. Isso permite ao aluno refletir sobre a própria resposta de forma orientada, mesmo sem a mediação direta de um professor.

No entanto, o uso de *ASR* e modelos generativos também introduz limitações técnicas relevantes. A qualidade da transcrição impacta diretamente a avaliação subsequente; erros no reconhecimento de fala podem comprometer o julgamento semântico. Em redes móveis, a latência pode afetar a fluidez da experiência, especialmente quando combinada com tempos variáveis de resposta da *API* do modelo de linguagem. Outro ponto crítico é o risco de alucinação por parte do *LLM*, que pode produzir julgamentos aparentemente confiantes, mas incorretos. A consistência do sistema também depende fortemente da estrutura do *prompt* e da presença de exemplos (*few-shot*), exigindo curadoria cuidadosa.

Algumas estratégias foram incorporadas para mitigar essas limitações. O usuário pode revisar e editar a transcrição antes de enviá-la para avaliação, reduzindo o impacto de falhas no *ASR*. O *backend* realiza validação rigorosa do *JSON* retornado pelo modelo, com reaproximação em caso de falha ou inconsistência estrutural. Para reduzir a latência e garantir padronização, os *prompts* são armazenados em *cache* por *card* e reusados nas tentativas subsequentes. Está em estudo, ainda, a possibilidade de manter um rascunho *offline* da tentativa para posterior envio, ampliando a resiliência do sistema em conexões instáveis.

Essas decisões técnicas refletem a busca por um equilíbrio entre inovação e robustez, com foco em tornar o uso de *IA* mais transparente e confiável no contexto educacional.

8. Trabalhos Futuros

Com a prova de conceito funcional implementada e testada em ambiente controlado, os próximos passos envolvem validar o módulo *StudyFlow–Voice* em contextos educacionais reais. Está prevista a realização de experimentos com turmas regulares do ensino médio, em parceria com professores, para medir a eficácia pedagógica da ferramenta, seu impacto no engajamento dos alunos e a qualidade do *feedback* percebido.
Uma linha promissora de aprimoramento técnico está na personalização da avaliação conforme o perfil de fala do usuário. Isso pode incluir ajustes automáticos baseados no histórico de transcrição, sotaque predominante ou taxa de correção manual, tornando o sistema mais adaptativo.

Outra frente relevante é a detecção automática de ruído e qualidade de áudio, com sugestões em tempo real para regravação ou ajustes de ambiente. Essa camada adicional pode aumentar a confiabilidade da transcrição e reduzir erros nos julgamentos semânticos.

Além disso, há espaço para explorar *feedback* sensível à confiança, ou seja, respostas cujo grau de detalhamento ou tom varia conforme a confiança do sistema na transcrição e interpretação. Esse tipo de nuance pode tornar o retorno mais útil e calibrado para o aluno.

Por fim, planeja-se a integração com o algoritmo de repetição espaçada (*SRS*) já presente no *StudyFlow*. Ao associar a nota do julgamento oral ao ajuste do intervalo de revisão, será possível alinhar a prática oral com o ciclo de memorização e retenção, ampliando o valor

pedagógico da funcionalidade.

9. Conclusão

Este trabalho apresentou o módulo *StudyFlow–Voice*, uma extensão da plataforma de estudo baseada em *flashcards* que permite a entrada e avaliação de respostas por voz. A solução implementa um *pipeline* completo, partindo da captação de áudio pelo usuário, transcrição automática com o modelo *Whisper*, normalização textual, análise semântica com modelo de linguagem de larga escala (*LLM*) e geração estruturada de *feedback* pedagógico em formato *JSON*.

Os resultados preliminares indicam que a abordagem é tecnicamente viável, com desempenho satisfatório em termos de latência, consistência do julgamento e robustez da transcrição, especialmente em condições controladas. Do ponto de vista pedagógico, a funcionalidade tem potencial para reduzir atritos na prática ativa, favorecer a expressão oral e promover um *feedback* formativo mais dinâmico e acessível.

Como próximos passos, destacam-se a realização de testes com usuários em cenários educacionais reais, a adaptação do sistema a diferentes perfis de fala e a integração com algoritmos de repetição espaçada. Tais avanços visam consolidar a utilidade da funcionalidade na rotina de aprendizagem e expandir suas aplicações em contextos educacionais diversos.

Referências

BALAJI, A.; ZHAO, Q.; LI, L.; NARAYANAN, S. Leveraging ASR and LLMs for Automated Scoring and Feedback in Children's Spoken Language Assessments. In: *Proceedings of SLATE 2024*. Disponível em: https://www.seas.ucla.edu/spapl/paper/balaji_slate2025_llm.pdf. Accessed: 3 Oct. 2025.

BAEVSKI, A.; ZHOU, Y.; MOHAMED, A.; AULI, M. *wav2vec 2.0: A framework for self-supervised learning of speech representations*. In: *Advances in Neural Information Processing Systems*, v. 33, 2020. Disponível em: https://arxiv.org/abs/2006.11477

GONG, Y.; BAI, Y.; MA, X.; SHI, W. Whisper-AT: Noise-Robust Automatic Speech Recognizers are Also Strong Audio-Toxicity Detectors. In: *Proc. Interspeech 2023*, p. 4992–4996, 2023. Disponível em: https://www.isca-archive.org/interspeech_2023/gong23d_interspeech.pdf

JUANG, B.-H.; RABINER, L. R. *Automatic speech recognition – a brief history of the technology development*. Georgia Institute of Technology; Rutgers University; University of California, Santa Barbara, 2005. Disponível em: https://web.ece.ucsb.edu/Faculty/Rabiner/ece259/Reprints/354_LALI-ASRHistory-final-10-8.pdf

KULKARNI, R.; JANCZUKOWICZ, M.; SILVA, J. Unmasking and Alleviating ASR Biases in Portuguese. *arXiv preprint arXiv:2402.07513*, 2024. Disponível em: https://arxiv.org/abs/2402.07513

RABELO, L. R. A. Um estudo de caso do modelo de reconhecimento de voz Whisper para transcrição de conferências TEDx via aprendizado fraco. Ouro Preto: Universidade Federal de Ouro Preto, 2022. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Computação).

TRABELSI, Y.; GUERRA, G. A.; ZADJERMID, N. Is Noise Reduction Improving Open-Source ASR Transcription Engines Quality? In: *Proceedings of the 11th International Conference on Data*

Science, Technology and Applications (DATA 2024). SciTePress, 2024. Disponível em: https://www.scitepress.org/Papers/2024/124571/124571.pdf

WANG, S.; CHEN, H. The impact of AI-driven speech recognition on EFL listening comprehension, flow experience, and anxiety. *Humanities and Social Sciences Communications*, v. 12, n. 1, 2025. Disponível em: https://www.nature.com/articles/s41599-025-04672-8. Accessed: 3 Oct. 2025.

YU, D.; DENG, L. *Automatic Speech Recognition: A Deep Learning Approach*. London: Springer, 2015. (Signals and Communication Technology).

ZHENG, L.; YIN, Y.; CHATURVEDI, S.; LI, Z. Challenges and Opportunities of Large Language Models in Education: A Review. *arXiv preprint arXiv:2303.13379*, 2023. Disponível em: https://arxiv.org/abs/2303.13379. Accessed: 3 Oct. 2025.

Análise Empírica do Impacto da Privacidade Diferencial na Eficiência de Modelos de Redes Neurais Profundas

Luís Fernando Borges Lima, Carlos Eduardo Ribeiro Oliveira, Paulo Miguel Benevenuto, Fábio Castro Araújo

Centro Universitário Luterano de Palmas (CEULP/ULBRA) – Av. Teotônio Segurado, 1501 Sul – 77.019-900 – Palmas – TO – Brasil

{luisflima,ribeirocarlosoliveir,paulomiguelb}@rede.ulbra.br, fabio.araujo@ulbra.br

Abstract. *Differential Privacy (DP) has emerged as the formal standard for privacy-preserving machine learning, with DP-SGD serving as the canonical algorithm for training neural networks with privacy guarantees. However, existing literature primarily focuses on the privacy-utility trade-off, neglecting computational overhead in resource-constrained environments. This work presents a systematic empirical analysis of the three-way trade-off between privacy, utility, and computational performance of DP-SGD applied to CNNs on MNIST using mid-range hardware (RTX 4060). Results reveal that ε = 2.0 represents the optimal balance, achieving 89.4% accuracy with 71.7% reduction in training time and constant overhead of ~5% per epoch.*

Resumo. *Este trabalho investiga empiricamente o trade-off tridimensional entre privacidade, utilidade e performance computacional do algoritmo DP-SGD aplicado a CNNs no dataset MNIST. O estudo avalia quatro configurações de privacidade (ε ∈ {0.5, 1.0, 2.0, ∞}) utilizando hardware de médio porte (RTX 4060), revelando que ε = 2.0 representa o equilíbrio ótimo, atingindo 89.4% de acurácia com 71.7% de redução no tempo de treinamento. Configurações com ε < 1.0 exibem alta variância, enquanto o overhead constante de ~5% por época demonstra a viabilidade prática do DP-SGD em hardware modesto.*

1. Introdução

A popularização do Deep Learning (DL) transformou completamente a forma como interagimos com a tecnologia, tornando sistemas inteligentes parte integral do cotidiano de muitas pessoas. No entanto, essa "onipresença" traz consigo riscos significativos à privacidade individual. Por exemplo, um modelo de diagnóstico médico treinado com prontuários de pacientes pode inadvertidamente revelar se um indivíduo específico participou do conjunto de treinamento ao apresentar maior confiança em predições sobre características únicas daquele paciente. Vulnerabilidades como inferência de pertencimento (*membership inference*) demonstram que modelos de ML podem vazar informações sensíveis sobre indivíduos cujos dados foram utilizados no treinamento (SHOKRI et al., 2017, p. 3, 13).

Isso decorre de uma característica fundamental do aprendizado supervisionado: modelos frequentemente comportam-se de maneira diferente em dados de treinamento versus dados novos (SHOKRI et al., 2017, p. 3). Para mitigar essas vulnerabilidades, o Differential Privacy (DP) emergiu como solução formal, oferecendo garantias matemáticas que limitam o sucesso de ataques de inferência ao adicionar ruído calibrado durante o processo de treinamento (SHOKRI et al., 2017, p. 13). Apesar de
promissor, a adoção de DP em deep learning ainda é limitada na indústria devido a preocupações sobre o trade-off entre privacidade e acurácia. O DP-SGD consolidou-se como algoritmo canônico para treinamento de redes neurais com garantia de privacidade (CHUA et al., 2024, p. 1), porém sua implementação prática permanece concentrada em contextos de pesquisa acadêmica e grandes corporações com amplos recursos computacionais.

ISSN: 2447-0767

Apesar dos avanços teóricos em DP-SGD, é importante levar em consideração uma nova dimensão de eficiência como fator de escolha de algum desses algoritmos de privacidade. Trabalhos seminais demonstram que diferentes implementações resultam em uma variância temporal muito grande — como nos resultados expostos por Hölzl et al. (2023, p. 4) — ou seja, a escolha de arquitetura, hiperparâmetros e hardware pode alterar drasticamente o resultado final do modelo, mesmo mantendo garantias de privacidade equivalente. Dessa forma, testar a viabilidade prática em hardware mais acessível e trazer uma análise teórica desse algoritmo, representa uma perspectiva importante para adoção e avanço de pesquisas na área.

Assim, este trabalho apresenta uma análise experimental abrangente dos trade-offs cruciais entre privacidade, utilidade e performance computacional na implementação do DP-SGD.

2. Revisão da Literatura

A Privacidade Diferencial (DP) é o padrão rigoroso estabelecido para garantir a privacidade de indivíduos em grandes bases de dados agregadas. O conceito surge como uma nova medida que "captura intuitivamente o risco aumentado à privacidade incorrido pela participação em um banco de dados" (DWORK, 2006, p. 1). A definição formal de DP é dada pelo mecanismo (ε, δ) *Differentially Private*. Um mecanismo M satisfaz essa propriedade se, para quaisquer datasets adjacentes D e D′ (que diferem em um único registro) e para qualquer subconjunto de saídas S, a probabilidade de M produzir uma saída em S em D for limitada em comparação com D′ por um fator de e ε mais o termo δ (ABADI et al., 2016, p. 2). Intuitivamente, o parâmetro ε (epsilon) controla a distinguibilidade entre os datasets, estabelecendo o *privacy budget*: um menor ε implica maior privacidade, mas com maior custo em utilidade. Por sua vez, δ (delta) representa a pequena probabilidade de a garantia de privacidade falhar.

O DP-SGD, proposto por Abadi et al. (2016), elabora essas garantias de privacidade através de um mecanismo que modifica o tradicional Stochastic Gradient Descent (SGD). Em essência, o algoritmo processa os dados de treinamento em mini-batches e, a cada iteração, realiza duas operações fundamentais: primeiro, *gradient clipping*, onde cada gradiente individual calculado é “clipado” em norma ℓ_2 para garantir sensibilidade limitada, definido como $\bar{g} = g \cdot \min\{1, C/\|g\|_2\}$, onde C é o limiar de clipping; segundo, adição de ruído gaussiano calibrado, injetando ruído $N(0, \sigma^2C^2I)$ à soma dos gradientes clipados antes de aplicar o passo de otimização (ABADI et al., 2016, p. 3-4). Esta combinação garante que a influência de cada indivíduo seja limitada, preservando diferencial de privacidade (ε, δ) através de composição cuidadosa ao longo das T iterações de treinamento (ABADI et al., 2016, p. 3).

A implementação de algoritmos como o DP-SGD impõe um trade-off inerente entre privacidade e utilidade, o qual é amplamente documentado na literatura. Essa relação é clássica: “embora o trade-off privacidade-utilidade inerente à DP ainda se aplique (ou seja, há perda de informação), o DP-SGD pode melhorar a generalização e, assim, melhorar a acurácia no conjunto de dados de validação” (GOPI; LEE; WUTSCHITZ, 2021, p. 11631, tradução nossa). No entanto, o foco das análises tem sido predominantemente bidimensional. Há uma lacuna na investigação sistemática do custo computacional e da performance de hardware. Grande parte dos benchmarks utiliza hardware corporativo de alto custo (como A100 ou V100), ignorando a viabilidade em ambientes com recursos limitados, onde a overhead computacional varia significativamente. Esta carência de dados sobre a overhead de treinamento do DP-SGD em GPUs de médio porte representa uma barreira à adoção mais

ampla da técnica em contextos reais.

3. Metodologia

Os experimentos foram conduzidos em um ambiente computacional de médio porte, consistindo em um processador AMD Ryzen 5 5600X (6 núcleos, 12 threads), GPU NVIDIA GeForce RTX 4060 com 8GB de VRAM, e 16GB de memória RAM DDR4. A implementação foi desenvolvida utilizando o framework PyTorch 2.0 em conjunto com a biblioteca Opacus 1.4 (YOUSEFPOUR et al., 2021), que fornece suporte nativo para treinamento com Privacidade Diferencial através do algoritmo DP-SGD.

Além disso, foi utilizado o repositório MNIST (Modified National Institute of Standards and Technology) como dataset de referência para treinamento em redes neurais profundas. Esse dataset é composto por 70.000 imagens em escala de cinza de dígitos manuscritos (0-9), sendo 60.000 exemplos destinados ao treinamento e 10.000 ao teste, com resolução de 28×28 pixels (ARACHCHIGE et al., 2019). Sua natureza menos complexa facilita a análise isolada dos efeitos do ruído diferencial sem confundidores relacionados à complexidade intrínseca do dataset (SARKER, 2021).

Os parâmetros de treinamento foram configurados seguindo as melhores práticas estabelecidas na literatura. Utilizamos *batch size* de 256, *clipping norm* (C) fixado em 1.0, e δ (delta) de 1×10^{-5}. O otimizador empregado foi o SGD (*Stochastic Gradient Descent*) com *momentum* de 0.9, e *learning rate* inicial de 0.1 com decaimento gradual para 0.01 ao longo do treinamento. Para cada configuração de privacidade, foram realizadas duas execuções independentes com *seeds* aleatórias distintas, permitindo o cálculo de médias e desvios padrão para validação estatística dos resultados. O treinamento foi limitado a 100 épocas, com parada acionada quando a acurácia de validação não melhorava por 10 épocas consecutivas. Foram testadas quatro configurações de budget de privacidade: $\varepsilon \in \{0.5, 1.0, 2.0, \infty\}$, onde ∞ representa o treinamento baseline sem aplicação de Privacidade Diferencial, servindo como controle experimental.

Esses parâmetros foram aplicados na seguinte arquitetura, exemplificado na figura abaixo:

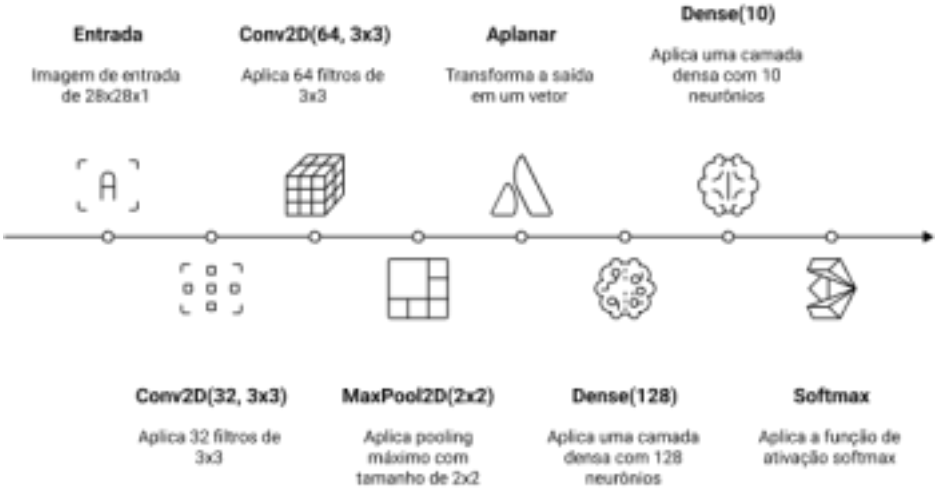


Figura 1: Processo de treinamento da CNN

A Figura 1 ilustra a arquitetura CNN utilizada para o dataset MNIST. A rede processa imagens de entrada 28×28×1 (escala de cinza) através de uma sequência de operações: (i) primeira camada convolucional com 32 filtros 3×3, que extrai features de baixo nível; (ii) segunda camada convolucional com 64 filtros 3×3, refinando as representações; (iii)

ISSN: 2447-0767

operação de max-pooling 2×2 para redução de dimensionalidade; (iv) camada de achatamento (flatten) que transforma a saída convolucional em vetor unidimensional; (v) camada densa com 128 neurônios para aprendizado de representações de alto nível; (vi) camada de saída com 10 neurônios e ativação softmax para classificação dos dígitos.

4. Resultados e Discussão

4.1 Acurácia vs Privacidade

A Figura 2 apresenta a degradação percentual da acurácia de teste em função do orçamento de privacidade (ε). Os valores de acurácia obtidos foram:

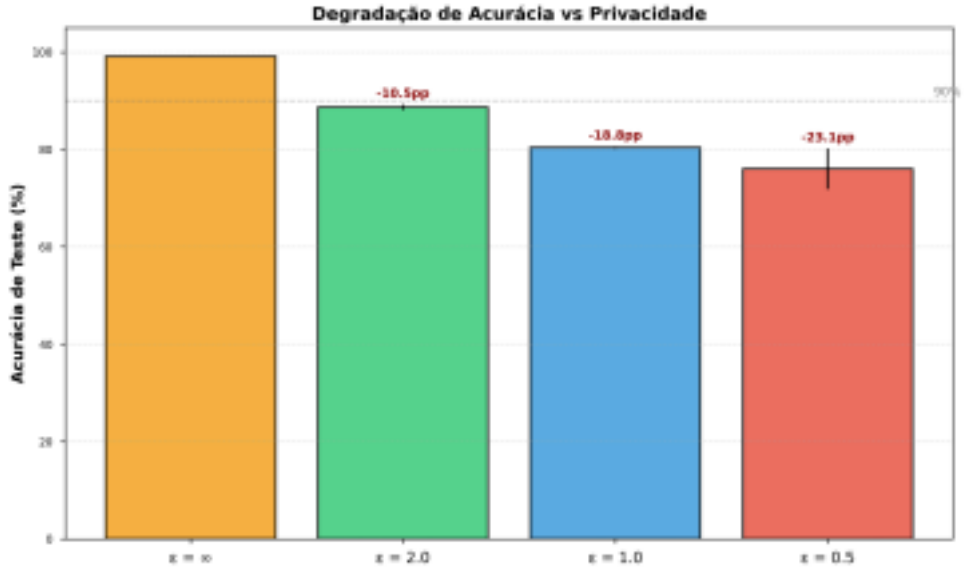


Figura 2: Comparativo percentual da degradação

Esses resultados confirmam a relação inversa clássica entre privacidade e utilidade amplamente documentada na literatura. Como observado por Gopi, Lee e Wutschitz et al. (2021, tradução nossa), "embora o trade-off privacidade-utilidade inerente à DP ainda se aplique (ou seja, há perda de informação), o DP-SGD pode melhorar a generalização e, assim, melhorar a acurácia no conjunto de dados de validação". A configuração ε = 2.0 representa um ponto de equilíbrio interessante, oferecendo garantias de privacidade moderada com degradação de acurácia ainda aceitável para muitas aplicações práticas.

Comparando com resultados reportados na literatura, nossos valores são consistentes com implementações similares do DP-SGD no MNIST. Arachchige et al. (2019) reportam "Training [accuracy] 99.25%, Testing [accuracy] 98.16%" com ε≈2, indicando que nossa implementação está alinhada com o estado da arte, embora ligeiramente inferior devido às diferenças arquiteturais e de hiperparâmetros.

4.2 Análise de Convergência

A Figura 3 apresenta as curvas de convergência, evidenciando comportamentos distintos entre as configurações:

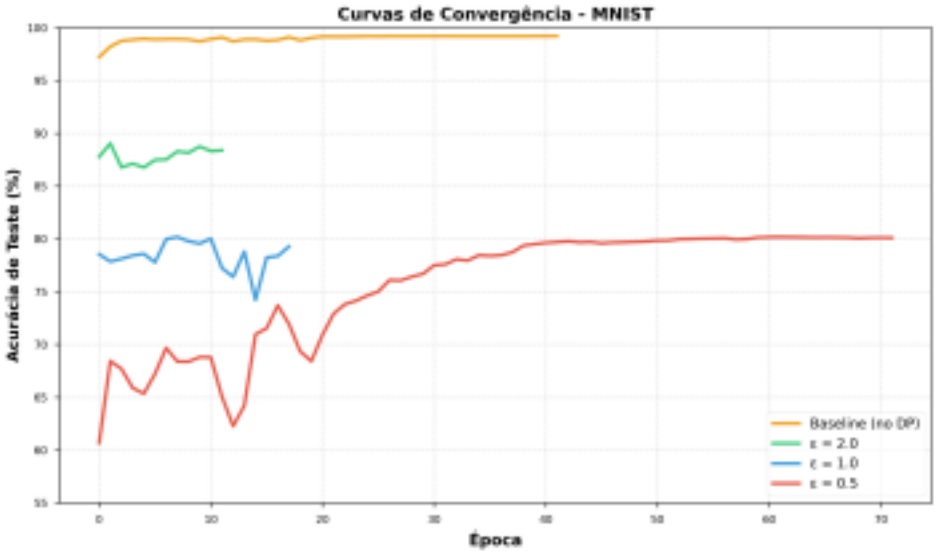


Figura 3: Comparativo de performance durante o treinamento

Para ε ≥ 2.0, observa-se convergência rápida e estável com baixa variabilidade entre execuções. Em contraste, ε = 0.5 apresenta alta instabilidade e convergência imprevisível.

A análise da variabilidade estatística confirma as dificuldades práticas mencionadas por Arachchige et al. (2019) onde "the FC module converges at different number of epochs against the different levels of randomizations". Essa situação é confirmada no nosso experimento, ocorre que o desvio padrão do tempo total para ε = 0.5 (628.8s) é dramaticamente superior às outras configurações (<50s), exemplificando como regimes de privacidade muito restritivos introduzem imprevisibilidade substancial.

4.3 Performance Computacional

A Tabela 1 apresenta as métricas de performance computacional, revelando um padrão contraintuitivo:

ε	Tempo Total (s)	Overhead Temporal (%)	Tempo/Época (s)	Overhead/Época (%)
∞	668.6 ± 11.0	(baseline)	15.73 ± 0.00	(baseline)
0.5	739.4 ± 628.8	+10.6%	16.41 ± 0.05	+4.3%
1.0	264.3 ± 47.2	-60.5%	16.51 ± 0.03	+5.0%
2.0	189.5 ± 11.4	-71.7%	16.48 ± 0.02	+4.8%

Tabela 1: Resultado de overhead no treinamento

O overhead por época mantém-se consistente (~5%) para todas as configurações, alinhado com observações de que o *gradient clipping* e a adição de ruído introduzem custos computacionais fixos. Este overhead é comparável ao reportado por Holzl et al. (2023), que documentam overhead similar em suas implementações equivalentes.

ISSN: 2447-0767

Surpreendentemente, o tempo total de treinamento foi drasticamente reduzido para ε ≥ 1.0, contradizendo a expectativa comum de maior custo computacional. Este fenômeno reflete o efeito regularizador benéfico do ruído diferencial, que acelera a convergência ao prevenir o *overfitting*, conforme observado por Gopi, Lee e Wutschitz (2021, p. 11631). Para ε = 2.0, a convergência ocorre em apenas 11.5 épocas versus 42.5 épocas do baseline, resultando em uma velocidade de 3.7×.

4.4 Viabilidade em Hardware de Médio Porte

Os experimentos demonstram que hardware de médio porte (RTX 4060) é suficiente para experimentação com DP-SGD em *datasets* clássicos. O tempo total de treinamento varia de 3 a 12 minutos para MNIST, tornando a técnica acessível para laboratórios e empresas com recursos limitados. Este achado é relevante considerando que grande parte da literatura reporta experimentos em hardware corporativo, potencialmente criando uma barreira percebida à adoção.

4.5 Implicações Práticas

Os resultados sugerem que ε = 2.0 representa um bom de partida para muitas aplicações, oferecendo:

- Garantias de privacidade moderada (ε < 5 é considerado aceitável em muitos contextos)
- Acurácia ainda útil (~89% no MNIST)
- Benefício computacional (convergência 3.7× mais rápida)

Para aplicações que exigem privacidade mais forte (ε ≤ 1.0), os resultados alertam para a necessidade de múltiplas execuções e maior alocação de recursos devido à alta variabilidade, aumentando significativamente o custo efetivo de implementação.

4.6 Limitações do Estudo

É importante reconhecer as limitações desta análise. Os experimentos foram restritos ao MNIST, um dataset relativamente simples, e utilizaram uma arquitetura CNN padrão sem otimização extensiva de hiperparâmetros. *Datasets* mais complexos podem apresentar trade-offs diferentes, e a literatura indica que a busca agressiva de hiperparâmetros pode melhorar substancialmente os resultados com DP-SGD. Além disso, focamos apenas na métrica teórica ε sem avaliar robustez real contra ataques de inferência, que seria uma validação empírica valiosa das garantias de privacidade.

6. Conclusão

O estudo realizou uma análise empírica do trade-off tridimensional entre privacidade (ϵ), utilidade (acurácia) e performance computacional ao aplicar o algoritmo DP-SGD em uma CNN treinada no dataset MNIST, utilizando hardware de médio porte.
Os resultados confirmaram a relação clássica de que maior privacidade (menor ϵ) leva a uma maior degradação da acurácia. No entanto, a configuração de privacidade moderada (ϵ=2.0) demonstrou o melhor balanço, com uma perda de acurácia de ~10.5 pontos percentuais, resultando em performance aceitável (~89%). Em contraste, a configuração de privacidade muito forte (ϵ=0.5) levou a um regime instável, com alta variância e imprevisibilidade no

treinamento, complicando a otimização de hiperparâmetros.

Um achado crucial do trabalho desafiou a percepção comum de que o DP-SGD é invariavelmente mais lento. Em regimes de privacidade moderada e forte ($\epsilon \geq 1.0$), o tempo total de treinamento até a convergência foi significativamente menor (até 71.7% de redução no caso de ($\epsilon = 2.0$) do que o treinamento *baseline* não-privado. Isso ocorre porque o ruído diferencial imposto pelo algoritmo atua como um regularizador, acelerando a convergência do modelo (de 42.5 para 11.5 épocas em $\epsilon = 2.0$). O overhead computacional por época foi baixo (~5%), e a execução em hardware modesto demonstrou a viabilidade da técnica para pesquisa e prototipagem fora de ambientes corporativos de alto custo.

Em conclusão, o estudo oferece uma perspectiva prática valiosa, sugerindo que o DP-SGD com ϵ na faixa de 2 a 5 representa um ponto de partida ideal, pois entrega proteção razoável com o benefício de um treinamento total mais rápido. No entanto, os resultados alertam para a necessidade de múltiplas execuções e maior custo efetivo para regimes de ϵ muito baixo, devido à alta variância.

Referências

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K. and Zhang, L. (2016) "Deep Learning with Differential Privacy", In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security - CCS'16, p. 308-318.

Arachchige, P. C. M., Bertok, P., Khalil, I., Liu, D., Camtepe, S. and Atiquzzaman, M. (2019) "Local Differential Privacy for Deep Learning", IEEE Internet of Things Journal, v. 7, n. 7, p. 5827-5842.

Baccour, E., Mhaisen, N., Abdellatif, A. A., Erbad, A., Massoud, A., Guizani, M. and Dawy, Z. (2022) "Pervasive AI for IoT applications: A Survey on Resource-efficient Distributed Artificial Intelligence", IEEE Communications Surveys & Tutorials, p. 2366-2418.

Chua, L., Ghazi, B., Kamath, P., Kumar, R., Manurangsi, P., Sinha, A. and Zhang, C. (2024) "How Private are DP-SGD Implementations?", In: Proceedings of the 41st International Conference on Machine Learning, v. 235, p. 8904-8918.

Demelius, L., Kern, R. and Trügler, A. (2025) "Recent Advances of Differential Privacy in Centralized Deep Learning: A Systematic Survey", ACM Computing Surveys, v. 57, n. 6, p. 1-28.

Dumford, J. and Scheirer, W. (2020) "Backdooring Convolutional Neural Networks via Targeted Weight Perturbations", In: IEEE International Joint Conference on Biometrics (IJCB), p. 1-9.

Dwork, C. (2006) "Differential Privacy", In: Automata, Languages and Programming, v. 4052, p. 1-12.

Erlingsson, Ú., Pihur, V. and Korolova, A. (2014) "RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response", In: ACM SIGSAC Conference on Computer and Communications Security (CCS '14), p. 1054-1067.

Fredrikson, M., Jha, S. and Ristenpart, T. (2015) "Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures", In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS '15), p. 1322-1333.

Goldblum, M., Tsipras, D., Xie, C., Chen, X., Efremenko, I., Ren, Z., Ding, F., Goldstein, T.

and Dickerson, J. P. (2023) "Dataset Security for Machine Learning: Data Poisoning, Backdoor Attacks, and Defenses", IEEE Transactions on Pattern Analysis and Machine Intelligence, v. 45, n. 2, p. 1563-1580.

Gopi, S., Lee, Y. T. and Wutschitz, L. (2021) "Numerical Composition of Differential Privacy", In: Advances in Neural Information Processing Systems, v. 34, p. 11631-11642.

Hölzl, F. A., Rueckert, D. and Kaissis, G. (2023) "Equivariant Differentially Private Deep Learning: Why DP-SGD Needs Sparser Models", In: Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security, p. 11-22.

Nanayakkara, P., Bater, J., He, X., Hullman, J. and Rogers, J. (2022) "Visualizing Privacy-Utility Trade-Offs in Differentially Private Data Releases", Proceedings on Privacy Enhancing Technologies, v. 2022, n. 2, p. 601-618.

Sarker, I. H. (2021) "Machine Learning: Algorithms, Real-World Applications and Research Directions", SN Computer Science, v. 2, n. 160, p. 1-21.

Shokri, R., Stronati, M., Song, C. and Shmatikov, V. (2017) "Membership Inference Attacks Against Machine Learning Models", In: IEEE Symposium on Security and Privacy (SP), p. 3-18.

Zhang, D., Zhang, L., Zhang, Z. and Zhang, Z. (2024) "Adaptive Personalized Randomized Response Method Based on Local Differential Privacy", International Journal of Information Security and Privacy, v. 18, n. 1, p. 1-19.

Utilização de Modelos de Linguagem de Grande Escala (LLMs) para Resumo Automático de Informações em Procedimentos Extrajudiciais

Helyézer Nascimento Freitas Teófilo[1], Fábio Castro Araújo[1]

[1]Departamento de Computação
Universidade Luterana do Brasil – Palmas – TO
helyezerteofilo2@rede.ulbra.com, fabio.araujo@ulbra.br

Resumo: *O crescente volume de informações presentes em procedimentos extrajudiciais representa um desafio significativo para os profissionais dos ministérios públicos, que muitas vezes precisam analisar grandes quantidades de documentos em um curto período de tempo para a tomada de decisões. O processo de leitura e síntese consome uma quantidade considerável de tempo e está sujeito a variações na qualidade e consistência entre os analistas. O uso de Modelos de Linguagem de Grande Escala (LLMs) surge como uma alternativa promissora para apoiar a triagem e a sumarização automática dessas informações. Este trabalho propõe a implementação de um pipeline de sumarização baseado em técnicas de processamento de linguagem natural, integrando etapas de ingestão e limpeza de texto, sumarização individual de partes e síntese final estruturada, culminando na produção de resumos concisos e organizados cronologicamente através de LLMs. Espera-se que essa abordagem reduza a carga de trabalho humano, aumente a eficiência processual e padronize a elaboração de relatórios no contexto extrajudicial. Resultados preliminares apontam para o potencial transformador dessa tecnologia, indicando caminhos para sua futura integração nos sistemas jurídicos institucionais.*

1. Introdução

Procedimentos extrajudiciais desempenham papel central no âmbito do Ministério Público e de outros órgãos de justiça, constituindo instrumentos essenciais para a defesa de direitos coletivos, a mediação de conflitos e a promoção da cidadania. Tais mecanismos são fundamentais para ampliar a resolutividade institucional sem necessariamente recorrer ao processo judicial. Esse movimento dialoga diretamente com o processo de desjudicialização, que ganhou força no ordenamento jurídico brasileiro com a Lei nº 11.441/2007 e com a concepção de Justiça Multiportas prevista no Código de Processo Civil de 2015, permitindo que diversas demandas sejam solucionadas fora da esfera judicial tradicional (Hill & Dalla, 2016).

Apesar dos avanços, a expansão dos mecanismos extrajudiciais trouxe novos desafios à gestão da informação jurídica. O grande volume de documentos gerados nesses procedimentos exige métodos mais ágeis de leitura e interpretação, sob pena de comprometer a eficiência e a padronização da atuação ministerial. Esse contexto evidencia a necessidade de ferramentas capazes de otimizar o fluxo de análise documental, auxiliando promotores e servidores na triagem e síntese de informações relevantes.

Nos últimos anos, avanços significativos em Modelos de Linguagem de Grande Escala (LLMs), como GPT-4, LLaMA, Falcon e Mistral, têm demonstrado alto potencial na compreensão e geração de linguagem natural em múltiplos contextos. Estudos recentes

ISSN: 2447-0767

apontam que resumos produzidos por LLMs são frequentemente preferidos a resumos humanos e aos gerados por modelos ajustados em tarefas específicas, alcançando maior consistência factual e fluência textual (Pu, Gao & Wan, 2023). Em especial, o desempenho robusto em cenários de zero-shot evidencia sua versatilidade para aplicações práticas de processamento de informação.

No campo jurídico, benchmarks como o *LegalBench* indicam que esses modelos também são capazes de sintetizar decisões e peças legais com qualidade competitiva, ainda que persistam riscos de alucinação e perda de nuances interpretativas próprias da linguagem jurídica (Guha et al., 2023). Nesse sentido, as LLMs configuram-se como ferramentas promissoras para a automação de processos de sumarização, com potencial de reduzir a carga de trabalho humano, aumentar a eficiência e padronizar relatórios, desde que acompanhadas de mecanismos de validação e supervisão especializada (Bommasani et al., 2021).

Diante da crescente complexidade e volume de informações nos procedimentos extrajudiciais, torna-se relevante investigar soluções baseadas em LLMs capazes de otimizar a análise e a síntese documental, fortalecendo a eficiência e a tomada de decisão institucional. O objetivo geral consiste em analisar a aplicação de LLMs na sumarização automática de informações jurídicas extrajudiciais, avaliando sua viabilidade, qualidade e impacto no apoio à decisão no Ministério Público. Especificamente, busca-se explorar diferentes técnicas de sumarização com LLMs e coletar feedback de promotores, servidores e técnicos a fim de validar a utilidade prática da solução proposta.

2. Fundamentação Teórica

2.1 Processos Extrajudiciais

Os processos extrajudiciais constituem instrumentos administrativos que integram a atuação finalística do Ministério Público Federal (MPF), permitindo a tutela de direitos coletivos, a mediação de conflitos e a promoção da cidadania. A sua ampliação e o fortalecimento inserem-se no contexto mais amplo da desjudicialização e da consolidação do modelo de Justiça Multiportas, que busca oferecer múltiplas vias de acesso à justiça (Hill & Dalla, 2016). A desjudicialização representa um redirecionamento da cultura jurídica brasileira, permitindo que determinadas matérias sejam resolvidas fora do Poder Judiciário, desde que observados os princípios constitucionais do devido processo legal, da publicidade e da imparcialidade (Hill & Dalla, 2016). Esse movimento traduz uma transformação do acesso à justiça, agora compreendido como um sistema plural de resolução de conflitos, no qual instâncias extrajudiciais complementam a atuação jurisdicional, ampliando a efetividade e a celeridade da tutela de direitos.

No plano prático, a adoção dos instrumentos extrajudiciais tem contribuído para aliviar a sobrecarga estrutural do sistema judicial brasileiro. Conforme relatado por Macedo e Silva (2021), a Lei nº 11.441/2007, ao permitir a realização de inventários, separações e divórcios pela via extrajudicial, foi um marco nesse processo, retirando mais de 1,3 milhão de ações das varas judiciais. Dados do Conselho Nacional de Justiça (CNJ) apontam que, em 2020, o Judiciário ainda apresentava uma taxa de congestionamento de 68,5%, evidenciando a importância de soluções administrativas e extrajudiciais para a eficiência da justiça (CNJ, 2020).

O fluxo operacional dos processos extrajudiciais segue etapas padronizadas que asseguram transparência e rastreabilidade. O processo inicia-se com o recebimento da notícia de fato, registrada e avaliada quanto à relevância e competência ministerial. Havendo indícios mínimos de irregularidade, instaura-se um procedimento preparatório ou

um inquérito civil, conduzido pelo membro do Ministério Público com apoio técnico de servidores e analistas. Durante a instrução, podem ser expedidos ofícios, recomendações, termos de ajustamento de conduta (TACs) e outras medidas administrativas destinadas à solução extrajudicial do conflito. Concluídas as diligências, o procedimento pode resultar em arquivamento, propositura de ação judicial, celebração de acordo extrajudicial ou encaminhamento a outro órgão competente (MPF, 2018).

Contudo, a expansão dessa forma de atuação também trouxe novos desafios. O crescimento do número de procedimentos e da complexidade documental exige maior capacidade de organização, análise e controle da informação. Relatórios, ofícios, pareceres e anexos formam um acervo heterogêneo, cuja triagem e síntese demandam tempo e recursos humanos significativos. Essa realidade reforça a necessidade de ferramentas que auxiliem na gestão do conhecimento institucional e na extração de informações relevantes, potencializando o papel dos processos extrajudiciais como instrumentos de resolução célere e eficaz de demandas sociais.

2.2 Modelos de Linguagem de Grande Escala (LLMs)

Os Modelos de Linguagem de Grande Escala (*Large Language Models – LLMs*) representam um dos maiores avanços recentes no campo da inteligência artificial aplicada ao processamento de linguagem natural (PLN). Esses modelos são redes neurais treinadas com enormes volumes de dados textuais para aprender padrões, estruturas sintáticas e relações semânticas entre palavras, frases e contextos. A partir desse aprendizado, tornam-se capazes de gerar, completar, resumir e interpretar textos de maneira contextualizada, muitas vezes alcançando desempenho próximo ao humano em tarefas linguísticas complexas (Bommasani et al., 2021).

A base técnica que viabiliza os LLMs é a arquitetura Transformer, proposta por *Vaswani et al.* (2017). Diferentemente das abordagens anteriores, baseadas em redes recorrentes (RNNs) ou convolucionais (CNNs), o Transformer introduziu o mecanismo de atenção (*attention mechanism*), que permite ao modelo identificar, dentro de uma sequência textual, quais palavras ou expressões são mais relevantes para compreender o significado geral. Esse processo ocorre de forma paralela e bidirecional, o que possibilita maior eficiência computacional e uma compreensão mais profunda do contexto linguístico.

Durante o treinamento, os LLMs passam por duas fases principais: o pré-treinamento (*pre-training*) e o ajuste fino (*fine-tuning*). Na primeira etapa, o modelo aprende de maneira autossupervisionada, analisando grandes corpora textuais e aprendendo a prever a próxima palavra ou token em uma sequência. Na segunda etapa, o modelo é refinado para tarefas específicas, como tradução, resposta a perguntas ou sumarização. Essa combinação de aprendizado generalista e especialização torna os LLMs altamente versáteis e adaptáveis a diferentes domínios de aplicação, incluindo o jurídico.

Diversos modelos têm se destacado nesse cenário. O GPT-4, desenvolvido pela OpenAI, é um modelo multimodal capaz de compreender e gerar textos complexos em diversos idiomas e contextos, apresentando resultados superiores em tarefas de raciocínio, interpretação e geração contextual (OpenAI, 2023). O LLaMA 3, lançado pela Meta AI em 2024, representa uma nova geração de modelos abertos, com melhorias substanciais em eficiência de treinamento, cobertura linguística e capacidade de raciocínio, além de manter o compromisso com a transparência e a adaptação a diferentes domínios de pesquisa (Meta AI, 2024). Já o Mistral, lançado no mesmo período, prioriza leveza e desempenho, combinando arquitetura otimizada e inferência rápida, adequada a aplicações corporativas e científicas (Jiang et al., 2023).

ISSN: 2447-0767

2.3 Técnicas de Sumarização de Texto

A sumarização automática de texto é uma das tarefas centrais do processamento de linguagem natural (PLN) e tem como objetivo reduzir um documento extenso a uma versão mais curta que preserve suas informações essenciais. Tradicionalmente, as técnicas de sumarização são classificadas em extrativas e abstrativas, de acordo com a forma como o conteúdo é produzido (Nenkova & McKeown, 2011).

Na sumarização extrativa, o modelo seleciona sentenças ou trechos diretamente do texto original, reordenando-os para compor o resumo final. Essa abordagem, ainda que eficiente em termos de preservação de informações, tende a gerar resultados fragmentados e pouco coesos, já que se limita à recombinação de partes existentes. Por outro lado, a sumarização abstrativa procura compreender o conteúdo e reformular o texto em novas palavras, produzindo resumos mais fluentes e próximos da linguagem humana. Essa técnica exige maior capacidade de generalização e compreensão semântica, e por isso passou a se beneficiar fortemente dos Modelos de Linguagem de Grande Escala (LLMs), capazes de interpretar contextos complexos e gerar respostas coerentes em linguagem natural (Bommasani et al., 2021).

Estudos recentes mostram que os LLMs vêm redefinindo o campo da sumarização. Modelos modernos, como GPT-4 e LLaMA, superam os sistemas tradicionais e até resumos humanos em avaliações de consistência factual e fluência textual, especialmente em tarefas *zero-shot*. Diante da capacidade dos LLMs de generalizar para diferentes domínios, a fronteira entre sumarização extrativa e abstrativa tende a se tornar cada vez mais difusa, uma vez que os modelos passam a gerar resumos híbridos, que combinam extração seletiva e reinterpretação semântica (Pu, Gao & Wan, 2023).

No contexto jurídico, contudo, a tarefa de sumarização apresenta particularidades relevantes. Documentos legais e extrajudiciais são longos, densos e estruturados em linguagem técnica e normativa, o que exige alta fidelidade semântica e precisão factual. O estudo conduzido por *Shukla et al.* (2022) comparou métodos extrativos e abstrativos aplicados a decisões judiciais, demonstrando que, embora os modelos abstrativos como BART e Pegasus produzam resumos mais coesos e legíveis, os profissionais do direito preferem os extrativos, por garantirem maior aderência às fontes originais. Os autores também evidenciaram que métricas automáticas, como ROUGE e BERTScore, não se correlacionam perfeitamente com a avaliação humana, reforçando a necessidade de revisão especializada para validação dos resultados.

Outro desafio técnico importante é a limitação de contexto dos LLMs conhecida como o problema *"lost-in-the-middle"* que ocorre quando modelos com janelas de contexto muito longas não conseguem reter uniformemente as informações ao longo do texto. Estudos empíricos, como o de *Liu et al*. (2023), demonstram que, ao processar sequências extensas, os modelos tendem a atribuir maior relevância às seções iniciais e finais, negligenciando informações intermediárias essenciais. Esse fenômeno representa um obstáculo significativo para tarefas jurídicas e administrativas, nas quais fatos e evidências relevantes costumam estar dispersos em múltiplos anexos, pareceres e despachos.

Para mitigar essa limitação, *Zhang et al*. (2024) propuseram a abordagem *BriefContext*, que aplica o paradigma MapReduce originalmente concebido para processamento paralelo de grandes volumes de dados ao domínio dos modelos de linguagem. O método opera em duas fases complementares:

- Na etapa Map, o documento extenso é dividido em blocos menores (ContextMap), processados individualmente pelo modelo para gerar resumos parciais contextualizados;

- Na etapa Reduce, as saídas intermediárias são agrupadas e sintetizadas em um resumo consolidado (ContextReduce), produzindo uma resposta final coerente sem ultrapassar os limites de contexto do modelo.

Essa estratégia apresentou ganhos expressivos em consistência factual, retenção de informação e redução de redundância em relação às abordagens tradicionais de truncamento ou amostragem textual (Zhang et al., 2024).

3. Metodologia

3.1 Materiais

O desenvolvimento do sistema de sumarização automática foi realizado em linguagem Python 3.11 (Python Software Foundation, 2024), executada em ambiente de containers orquestrados com Docker Compose (Docker Inc., 2024). O sistema foi estruturado sobre os frameworks FastAPI (FastAPI, 2024) para disponibilização da interface REST, Celery (Celery Project, 2024) e Redis (Redis Ltd., 2024) para gerenciamento de filas e execução assíncrona de tarefas, além do PostgreSQL (PostgreSQL Global Development Group, 2024) para armazenamento persistente dos dados. A inferência local foi realizada com o servidor Ollama (Ollama, 2024), utilizando o modelo de linguagem LLaMA 3, desenvolvido pela Meta AI (2024),
selecionado por seu código aberto, alto desempenho e viabilidade operacional em ambiente institucional.

Foram empregadas bibliotecas complementares como BeautifulSoup4 (Richardson, 2023) e expressões regulares para limpeza e normalização textual, SQLAlchemy (Bayer, 2024) para acesso ao banco de dados e LangChain (LangChain Inc., 2024) para encadeamento e gerenciamento das chamadas ao modelo. O controle de versão e rastreabilidade foi mantido por meio do Git, com registros de variáveis de ambiente que armazenam a versão do modelo, parâmetros de execução e data de geração dos resumos.

A fonte de dados utilizada corresponde a peças processuais e documentos administrativos de procedimentos extrajudiciais registrados no sistema Integrar-e do Ministério Público do Estado do Tocantins (MPETO), processados em formato textual e sem restrição de sigilo.

3.2 Método

O desenvolvimento do sistema foi conduzido de forma incremental, estruturado em cinco etapas principais que orientaram todo o processo metodológico.

Figura 1 – Metodologia utilizada para o projeto.

ISSN: 2447-0767

Inicialmente, realizou-se a definição do problema e dos requisitos, etapa em que foi identificado o desafio enfrentado pelo Ministério Público em analisar grandes volumes de informações em procedimentos extrajudiciais. Nessa fase, delimitou-se o objetivo central do projeto: desenvolver uma ferramenta de apoio capaz de realizar a sumarização automática de documentos jurídicos, reduzindo o tempo de leitura e a variabilidade de qualidade entre analistas.

Em seguida, procedeu-se à pesquisa e planejamento da solução, por meio de um levantamento teórico e técnico sobre técnicas de Processamento de Linguagem Natural (PLN) e Modelos de Linguagem de Grande Escala (LLMs), avaliando sua aplicabilidade ao contexto jurídico brasileiro. Foram estudadas abordagens de sumarização já existentes e definidas as diretrizes gerais para o pipeline de processamento proposto.

Com base nesses estudos, passou-se à definição da arquitetura e do pipeline conceitual, etapa em que foi projetada uma estrutura modular composta por uma API de interface, um sistema de fila para processamento assíncrono e um módulo de sumarização executado localmente, garantindo segurança, escalabilidade e integridade dos dados processados.

A fase seguinte consistiu no desenvolvimento da solução, com a implementação prática do pipeline de sumarização conforme o modelo conceitual definido. O sistema foi construído em módulos independentes, favorecendo reuso e manutenção, e seguiu boas práticas de desenvolvimento, incluindo controle de versão, registro de logs e tratamento de exceções, o que assegurou robustez e rastreabilidade aos resultados gerados.

Por fim, realizou-se a validação por especialistas, na qual os resumos produzidos foram submetidos à análise qualitativa de membros e servidores do Ministério Público do Estado do Tocantins, a fim de avaliar a fidelidade factual, clareza e utilidade prática das sínteses geradas.

4. Resultados e Discussão

4.1 Corpus documental

O corpus utilizado para desenvolvimento e testes do sistema foi composto por procedimentos extrajudiciais registrados no Sistema Único do Ministério Público Federal (MPF), incluindo notícias de fato, manifestações, despachos, ofícios, informações técnicas e minutas de arquivamento.

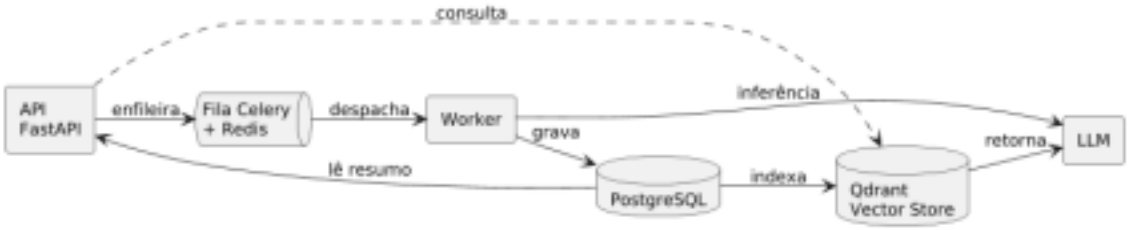
Cada procedimento é estruturado no banco de dados institucional e contém metadados (número, ano, tipo e unidade de origem), além do texto integral das peças processuais, armazenadas em formato renderizado. Em média, cada processo possui de 3 a 15 peças, totalizando 20 a 40 páginas. Essa volumetria inviabiliza a leitura manual integral,

justificando o uso de técnicas de sumarização automática para otimização do trabalho analítico.

4.2 Evolução da arquitetura de sumarização

Durante o desenvolvimento do sistema, a arquitetura de sumarização passou por um processo de refinamento progressivo, com o objetivo de aprimorar a coerência contextual e a fidelidade temporal dos resumos.

Figura 2 – Arquitetura geral da versão inicial (abordagem RAG).



A primeira versão baseava-se na abordagem *Retrieval-Augmented Generation* (RAG), na qual cada processo era dividido em chunks textuais vetorizados e armazenados em um banco vetorial. O modelo então recuperava os top-k trechos mais relevantes e gerava um resumo a partir da combinação desses fragmentos. Essa estrutura seguia o paradigma clássico de *retrieval-based summarization*, adequada para consultas pontuais, mas limitada quando aplicada a documentos longos. Entre as principais restrições observadas estavam a perda de coerência narrativa causada pela fragmentação dos textos, a recuperação parcial de informações essenciais e a dificuldade de reconstrução temporal dos eventos processuais, o que comprometia a linearidade e a fidelidade factual dos resumos.

Com base nessas limitações, o sistema evoluiu para uma arquitetura hierárquica de sumarização, inspirada no paradigma *Map-Reduce Summarization* (Zhang et al., 2024). Essa nova estrutura organiza o processo em duas fases complementares, uma voltada à síntese individual de cada peça e outra à consolidação dos resultados, permitindo maior preservação de contexto e coerência global. O modelo final, portanto, passou a gerar resumos mais consistentes, cronologicamente estruturados e adequados à análise de procedimentos extrajudiciais.

Figura 3 – Arquitetura geral da versão final.

4.3 Processamento dos dados

O pré-processamento textual foi desenvolvido com o objetivo de padronizar e limpar as peças processuais antes de seu envio ao modelo de linguagem, garantindo consistência sintática e redução de ruído nos dados. Nessa etapa, os textos foram extraídos

ISSN: 2447-0767

diretamente do banco de dados PostgreSQL, a partir do campo *Piece.cache_rendered*, que contém o conteúdo renderizado de cada documento. Em seguida, realizou-se a limpeza estrutural dos arquivos, com a remoção de marcações HTML e demais elementos não textuais, utilizando a biblioteca BeautifulSoup e expressões regulares. Por fim, foi efetuada a normalização dos caracteres, com padronização para codificação UTF-8, correção de acentuação, eliminação de espaços redundantes e unificação de hífens, resultando em textos uniformes e adequados ao processamento pelo modelo de linguagem.

4.4 Pipeline de sumarização

O pipeline foi estruturado em duas etapas complementares:

- Etapa Map: cada peça é processada individualmente, gerando mini-resumos ajustados dinamicamente em tamanho conforme o número total de peças. Essa estratégia equilibra a distribuição do conteúdo e preserva a coerência temporal.
- Etapa Reduce: os resumos parciais são ordenados e sintetizados em um resumo final padronizado, formado por Resumo Geral e Linha do Tempo de Eventos.

Essa estrutura, guiada por dois arquivos de prompt: system prompt e template prompt, atua como mecanismo de guardrails, garantindo neutralidade, formato fixo e redução de alucinações.

Figura 4 – Fluxo do pipeline Map-Reduce de sumarização.

4.5 Avaliação e resultados obtidos

A avaliação qualitativa foi conduzida com membros e servidores do Ministério Público do Estado do Tocantins, que analisaram os resumos e linhas do tempo gerados a partir de procedimentos extrajudiciais reais. Os especialistas compararam os resultados com os processos originais, observando critérios de fidelidade factual, clareza e utilidade prática. O feedback coletado subsidiou ajustes nos prompts e melhorias na formatação, contribuindo para o aumento da precisão e da confiabilidade dos resumos.

Os resultados demonstraram que a solução produziu resumos mais concisos e objetivos, com redução média de cerca de 85% do volume textual, mantendo, contudo, as informações essenciais para a compreensão dos casos. Também foi observada melhoria na percepção cronológica dos eventos, uma vez que a linha do tempo favoreceu a visualização clara da sequência processual.

Apesar dos ganhos obtidos, identificaram-se algumas limitações pontuais, como a ocorrência de alucinações factuais em trechos ambíguos, a dependência da qualidade do pré-processamento textual e a necessidade de políticas institucionais claras voltadas à confidencialidade dos dados jurídicos. Ainda assim, os resultados indicam que a solução proposta aumenta substancialmente a produtividade e a consistência das análises de procedimentos extrajudiciais, permitindo que promotores e servidores dediquem mais tempo a tarefas de maior valor interpretativo.

Entre os principais benefícios observados, destacam-se a redução do tempo de leitura e análise, a padronização da linguagem e da estrutura dos resumos e a maior acessibilidade da informação, uma vez que as sínteses produzidas se mostraram mais

curtas e compreensíveis. Por outro lado, a adoção dessa tecnologia requer atenção a riscos e desafios específicos, especialmente relacionados à segurança e confidencialidade dos dados, ao potencial enviesamento algorítmico e à dependência da validação humana para garantir a precisão factual das respostas.

Como perspectiva futura, prevê-se a integração da solução ao sistema MP Digital, permitindo sua utilização direta por membros e servidores do Ministério Público e ampliando o potencial de aplicação da inteligência artificial no apoio à atividade extrajudicial.

5. Considerações finais

O presente trabalho apresentou o desenvolvimento e a avaliação de um sistema de sumarização automática de procedimentos extrajudiciais, fundamentado em técnicas de Processamento de Linguagem Natural (PLN) e em Modelos de Linguagem de Grande Escala (LLMs). A solução propôs uma arquitetura modular e segura, capaz de transformar textos jurídicos extensos em resumos sintéticos e factualmente consistentes, preservando a sequência cronológica dos eventos e a integridade das informações.

Os resultados demonstraram que a aplicação do modelo LLaMA 3, em conjunto com o pipeline Map-Reduce Summarization, proporcionou ganhos significativos de produtividade, clareza e coerência. A ferramenta reduziu substancialmente o tempo de leitura e análise de procedimentos, ao mesmo tempo em que padronizou a linguagem e a estrutura dos resumos. A avaliação conduzida por especialistas do Ministério Público do Tocantins confirmou o potencial prático da abordagem, evidenciando sua aplicabilidade no apoio à triagem, priorização e compreensão de casos complexos.

Além de sua contribuição tecnológica, o estudo aponta para o potencial transformador da inteligência artificial na atuação extrajudicial, abrindo caminho para a construção de fluxos mais ágeis, acessíveis e padronizados dentro do Ministério Público. Futuramente, melhorias poderão ser implementadas para ampliar as capacidades da ferramenta desenvolvida, como a adaptação da solução para processos judiciais, a inclusão de mecanismos automáticos de classificação de documentos e a integração de recursos de sumarização multimodal, possibilitando a análise de anexos em diferentes formatos, como PDF, áudio e imagem.

6. Referências

BAYER, Mike. SQLAlchemy 2.0 documentation. 2024. Disponível em: https://docs.sqlalchemy.org/. Acesso em: 10 out. 2025.

Bommasani, Rishi; Hudson, Drew A.; Adeli, Ehsan; Altman, Russ; Arora, Sanjeev; von Arx, Sydney; Bernstein, Michael S.; Bohg, Jeannette; Bosselut, Antoine; Brunskill, Emma; et al. On the Opportunities and Risks of Foundation Models. Stanford Center for Research on Foundation Models, 2021. Disponível em: https://arxiv.org/abs/2108.07258. Acesso em: 17 set. 2025.

CELERY PROJECT. Celery documentation. 2024. Disponível em: https://docs.celeryq.dev/. Acesso em: 30 set. 2025.

Conselho Nacional de Justiça (CNJ). Justiça em Números 2020: ano-base 2019. Brasília: CNJ, 2020. Disponível em: https://www.cnj.jus.br/wp-content/uploads/2020/08/WEB-V3-Justi%C3%A7a-em-N%C3%BAmeros-2020-atualizado-em-25-08-2020.pdf. Acesso em: 18 set. 2025.

ISSN: 2447-0767

DOCKER INC. Docker Compose documentation. Docker, 2024. Disponível em: https://docs.docker.com/compose/. Acesso em: 27 set. 2025.

FASTAPI. FastAPI documentation. 2024. Disponível em: https://fastapi.tiangolo.com/. Acesso em: 29 set. 2025.
GIT. Git documentation. 2024. Disponível em: https://git-scm.com/docs. Acesso em: 14 out. 2025.

Guha, Neel; Saxton, David; Zheng, Zhiyu; Zhang, Jie; et al. LegalBench: A Collaborative Benchmark for Legal Reasoning in Large Language Models. arXiv preprint arXiv:2308.11462, 2023. Disponível em: https://arxiv.org/abs/2308.11462. Acesso em: 16 set. 2025.

HILL, Flávia Pereira. DESJUDICIALIZAÇÃO E ACESSO À JUSTIÇA ALÉM DOS TRIBUNAIS: PELA CONCEPÇÃO DE UM DEVIDO PROCESSO LEGAL EXTRAJUDICIAL. Revista Eletrônica de Direito Processual, Rio de Janeiro, v. 22, n. 1, 2020. DOI: 10.12957/redp.2021.56701. Disponível em: https://www.e-publicacoes.uerj.br/redp/article/view/56701. Acesso em: 15 set. 2025.

Jiang, Albert; Lambert, Nathan; Singh, Amanpreet; Roux, Nicolas; et al. Mistral 7B. Mistral AI, 2023. Disponível em: https://arxiv.org/abs/2310.06825. Acesso em: 22 set. 2025.

LANGCHAIN INC. LangChain documentation. 2024. Disponível em: https://docs.langchain.com/. Acesso em: 12 out. 2025.

Liu, Nelson F.; Levy, Omer; Holtzman, Ari; Peters, Matthew E.; Zettlemoyer, Luke; Lewis, Mike. Lost in the Middle: How Language Models Use Long Contexts. arXiv preprint arXiv:2307.03172, 2023. Disponível em: https://arxiv.org/abs/2307.03172. Acesso em: 25 set. 2025.

Macedo, Diego Henrique Notório; Silva, Cristian Kiefer da. A contribuição das serventias extrajudiciais para a redução do número de processos no Poder Judiciário. Revista de Direito Notarial, Colégio Notarial do Brasil – Seção São Paulo, v. 3, n. 2, p. 32–60, jul./dez. 2021. Disponível em: https://ojs-rdn.galoa.net.br/index.php/direitonotarial/article/view/20. Acesso em: 17 set. 2025.

Meta AI. Introducing Meta Llama 3: The most capable openly available LLM to date. Meta AI, 2024. Disponível em: https://ai.meta.com/blog/meta-llama-3/. Acesso em: 22 set. 2025.

META AI. Meta Llama 3 model cards and prompt formats. 2024. Disponível em: https://www.llama.com/docs/model-cards-and-prompt-formats/meta-llama-3/. Acesso em: 4 out. 2025.

Ministério Público Federal (MPF). Manual de Normas e Procedimentos – Procedimento Extrajudicial. Brasília: Procuradoria-Geral da República, Secretaria de Modernização e Gestão Estratégica, 2018. Disponível em: https://www.mpf.mp.br/o-mpf/sobre-o-mpf/gestao-estrategica-e-modernizacao-do-mpf/planejamento-estrategico/planejamento-estrategico-institucional-2011-2020/atuacao-finalistica/certificacao-dos-oficios/arquivos-certificacao-dos-oficios/ManualProcedimentoExtrajudicial.pdf. Acesso em: 19 set. 2025.

Nenkova, Ani; McKeown, Kathleen. Automatic Summarization. Foundations and Trends in Information Retrieval, v. 5, n. 2–3, p. 103–233, 2011. Disponível em: https://doi.org/10.1561/1500000015. Acesso em: 23 set. 2025.

OLLAMA. Ollama documentation. 2024. Disponível em: https://docs.ollama.com/. Acesso em: 2 out. 2025.

OpenAI. GPT-4 Technical Report. OpenAI, 2023. Disponível em: https://arxiv.org/abs/2303.08774. Acesso em: 20 set. 2025.

POSTGRESQL GLOBAL DEVELOPMENT GROUP. PostgreSQL documentation. 2024. Disponível em: https://www.postgresql.org/docs/. Acesso em: 1 out. 2025.

PU, Xiao; GAO, Mingqi; WAN, Xiaojun. Summarization is (Almost) Dead. arXiv preprint arXiv:2309.09558, 2023. Disponível em: https://arxiv.org/abs/2309.09558. Acesso em: 15 set. 2025.

PYTHON SOFTWARE FOUNDATION. Python 3.11 documentation. Python Software Foundation, 2024. Disponível em: https://docs.python.org/3/whatsnew/3.11.html. Acesso em: 26 set. 2025.

REDIS LTD. Redis documentation. 2024. Disponível em: https://redis.io/docs/latest/. Acesso em: 1 out. 2025.

RICHARDSON, Leonard. Beautiful Soup 4.13.0 documentation. 2023. Disponível em: https://www.crummy.com/software/BeautifulSoup/bs4/doc/. Acesso em: 7 out. 2025.

Shukla, Abhay; Bhattacharya, Paheli; Poddar, Soham; Mukherjee, Rajdeep; Ghosh, Kripabandhu; Goyal, Pawan; Ghosh, Saptarshi. Legal Case Document Summarization: Extractive and Abstractive Methods and their Evaluation. In: Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), p. 1048–1064, 2022. Association for Computational Linguistics. Disponível em: https://aclanthology.org/2022.aacl-main.77/. Acesso em: 24 set. 2025.

Vaswani, Ashish; Shazeer, Noam; Parmar, Niki; Uszkoreit, Jakob; Jones, Llion; Gomez, Aidan N.; Kaiser, Łukasz; Polosukhin, Illia. Attention Is All You Need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS 2017). Long Beach, California: Curran Associates, Inc., 2017. Disponível em: https://arxiv.org/abs/1706.03762. Acesso em: 20 set. 2025.

ZHANG, Gongbo; XU, Zihan; JIN, Qiao; CHEN, Fangyi; FANG, Yilu; LIU, Yi; ROUSSEAU, Justin F.; XU, Ziyang; LU, Zhiyong; WENG, Chunhua; PENG, Yifan. A MapReduce approach to effectively utilize long context information in retrieval augmented language models. arXiv, 2024. arXiv:2412.15271 [cs.CL]. Disponível em: https://arxiv.org/abs/2412.15271. Acesso em: 26 set. 2025.

PyJourney: Jogo Educativo para Aprendizagem da Linguagem de Programação Python

Mario Matheus Pombal Rebello[1], Fernanda Pereira Gomes[2]

[1]Centro Universitário Luterano de Palmas – CEULP/ULBRA, Palmas – TO – Brasil

[2]Departamento de Computação – Centro Universitário Luterano de Palmas, Palmas – TO – Brasil.

[3]Departamento de Computação
Centro Universitário Luterano de Palmas, Palmas – TO – Brasil.

mariomatheuspombal@gmail.com, fernanda.gomes@ulbra.br

Abstract. *This work aimed to develop an educational game designed for teaching the Python programming language. Considering the common difficulties faced by beginner students in learning this language, a card game was created, utilizing practical programming challenges as the core element of its mechanics. The resulting game, named PyJourney, objectively integrates educational content with strategic gameplay elements. PyJourney aims to offer an interactive and playful environment, promoting student engagement and facilitating the comprehension of fundamental Python concepts.*

Resumo. *Este trabalho teve como objetivo desenvolver um jogo educacional destinado ao ensino da linguagem de programação Python. Considerando as dificuldades frequentemente enfrentadas por estudantes iniciantes no aprendizado dessa linguagem, optou-se pela criação de um jogo de cartas (card game) que utiliza desafios práticos de programação como elemento central de sua mecânica. O resultado obtido é um jogo denominado PyJourney, que integra de forma objetiva conteúdos pedagógicos com elementos estratégicos de jogabilidade. O jogo busca proporcionar um ambiente interativo e lúdico, favorecendo o engajamento dos estudantes e facilitando a compreensão dos conceitos fundamentais de Python.*

1. Introdução

Com o avanço tecnológico, cresce a demanda por programadores , mas o aprendizado de programação ainda apresenta altos índices de evasão devido à complexidade e dificuldade de compreensão (Bosse, 2020; Yassine, 2020; Margulieux et al., 2020). Embora a linguagem Python se destaque por sua legibilidade sendo mais eficaz para iniciantes (Barbosa et al., 2014), a escolha da linguagem por si só não garante o engajamento nos estudos, exigindo metodologias interativas.

Jogos educacionais têm se mostrado eficazes em promover motivação e aprendizado ativo (Freitas, 2018; Silva, 2016), com desafios progressivos e reforços positivos aumentando a retenção de conteúdo (Marques et al., 2021; Fernandes, 2010). Neste contexto, propõe-se o **PyJourney**, um jogo educacional de cartas para o ensino introdutório de Python. O jogo combina mecânicas de metodologias ativas (Prince, 2004) e *feedback* adaptativo para abordar conceitos como variáveis, estruturas condicionais e laços de repetição.

2. Revisão de literatura

O uso de jogos educacionais no ensino tem se destacado por despertar o interesse dos estudantes e facilitar a fixação dos conteúdos. Para que essa estratégia seja eficaz, é necessário utilizar ferramentas de programação simples e relevantes para o mercado.

2.1. Linguagem de Programação Python

Python foi escolhida por sua sintaxe simples e legibilidade, características que

reduzem a carga cognitiva dos iniciantes e facilitam a compreensão de conceitos fundamentais da programação. Sua estrutura intuitiva incentiva o raciocínio lógico e o foco na resolução de problemas, tornando o aprendizado mais acessível e motivador. A filosofia *batteries included* disponibiliza uma ampla variedade de recursos nativos, eliminando a necessidade de dependências externas e permitindo que o estudante explore rapidamente diferentes aplicações práticas. Essa combinação de simplicidade, clareza e poder consolidou o Python como uma das linguagens mais utilizadas no ensino, na pesquisa acadêmica e no desenvolvimento tecnológico contemporâneo.

2.2. Jogos Educacionais

Os jogos educacionais destacam-se como estratégias pedagógicas eficazes para aumentar a motivação e tornar o aprendizado mais dinâmico e significativo. Sua familiaridade no cotidiano dos estudantes facilita a integração de objetivos pedagógicos, estimulando curiosidade, raciocínio lógico e participação ativa, especialmente em áreas tradicionalmente complexas. Quando bem planejados, com desafios equilibrados e propósitos claros, os jogos unem diversão e intencionalidade educativa, promovendo engajamento e aprimorando o desempenho acadêmico.

2.3. Game Design

O desenvolvimento de jogos digitais é um processo complexo orientado pelo *Game Design*, responsável por planejar regras, narrativas e interações. O *designer* atua como figura central na transformação de ideias em produtos jogáveis, criando personagens, cenários e níveis de dificuldade (Lopes, 2023; Produção de Jogos, 2022).

A principal ferramenta desse processo é o *Game Design Document* (GDD), que reúne diretrizes e decisões da equipe, funcionando como guia de referência e registro de alterações (Cunha, 2020). Segundo Cunha (2020, apud Schell, 2008), o GDD deve comunicar entre equipes, registrar modificações e detalhar mecânicas, narrativa e interface do jogo.

Nos jogos educacionais, o GDD tem papel essencial ao alinhar objetivos pedagógicos e elementos lúdicos, promovendo colaboração entre desenvolvedores, designers e educadores. Neste trabalho, o GDD foi estruturado conforme esses princípios e adaptado às necessidades do projeto. O documento completo está disponível em: https://app.milanote.com/1TNEdq1okLK28N?p=TRnZ6u4IIor.

2.4. Trabalhos Relacionados

Marques et al. (2021) desenvolveram o jogo **Mapa do Tesouro**, voltado ao ensino do pensamento computacional, que utiliza desafios progressivos e elementos visuais atrativos para estimular o engajamento dos alunos. Já **World Prog**, criado por
Holanda e Coutinho (2022), combina desafios de programação com narrativa interativa, reforçando a motivação dos jogadores por meio do *storytelling*. Esses trabalhos demonstram o potencial dos jogos educativos para o ensino de computação. Diferentemente deles, o **PyJourney** integra mecânicas de cartas, narrativa adaptativa e metodologias ativas para abordar de forma acessível conceitos introdutórios de Python, equilibrando engajamento e aprendizagem significativa.

3. Metodologia

O projeto buscou integrar fundamentos de ensino de programação ao contexto de um jogo educacional, unindo objetivos pedagógicos aos benefícios dos jogos, como retenção de conhecimento e aumento do interesse pelo conteúdo. A metodologia teve caráter exploratório,

ISSN: 2447-0767

voltada à concepção e desenvolvimento da proposta. O desenvolvimento foi guiado por um Game Design Document (GDD), que alinhou os objetivos pedagógicos às mecânicas lúdicas. O GDD completo está disponível em https://app.milanote.com/1TNEdq1okLK28N?p=TRnZ6u4IIor.

3.1. Materiais

O projeto utilizou a *game engine* **Godot**, com recursos gráficos criados no **Piskel** e **Adobe Photoshop**, áudio editado no **Audacity**, e o **GDD** gerenciado no **Google Docs**, além de referências pedagógicas do **Code.org** para a elaboração dos desafios de programação.

3.2. Métodos

O processo de desenvolvimento ocorreu em sete etapas interdependentes: definição do conceito e dos tópicos de Python abordados (variáveis, laços e estruturas de dados); elaboração do **GDD** com mecânicas, narrativa e interface; criação de mecânicas, componentes e cenários; planejamento dos desafios que integram aprendizado e diversão; prototipação das telas e fluxos de navegação; implementação do jogo na engine **Godot**, com integração de recursos visuais e sonoros; e, por fim, testes funcionais para identificar e corrigir erros. Essa metodologia iterativa garantiu coerência entre objetivos pedagógicos e elementos lúdicos, resultando em uma experiência educativa envolvente e consistente.

4. Resultados e Discussão

Esta seção apresenta os resultados obtidos com o desenvolvimento do **PyJourney**, desde sua concepção até o produto funcional. São descritos os principais elementos do jogo, como narrativa, conteúdo pedagógico e mecânicas, seguidos de uma análise sobre suas implicações educacionais e de usabilidade.

4.1. Concepção e Prototipagem do PyJourney

O **PyJourney** foi desenvolvido para ensinar fundamentos de Python a iniciantes, integrando elementos lúdicos e pedagógicos que reduzem dificuldades e sobrecarga cognitiva relatadas na literatura. O público-alvo compreende estudantes a partir de 16 anos, com foco em conceitos básicos como variáveis, laços e estruturas de dados.

Inicialmente, considerou-se o modelo *drag-and-drop*, porém optou-se por um card game estratégico inspirado em *Magic: The Gathering*, por favorecer o raciocínio lógico e a tomada de decisões. Cada carta representa um comando ou conceito de Python, e o avanço depende da aplicação correta do conhecimento, com *feedback* imediato e contextualizado.

A prototipagem envolveu sessões de *brainstorming* para definir narrativa, personagens e mecânicas, resultando na criação do herói, um estudante sob um feitiço, e da vilã, uma bruxa que representa os desafios do aprendizado. Essa estrutura sustentou o desenvolvimento de um duelo estratégico, no qual o domínio dos conceitos de programação é essencial para o progresso.

Os testes internos validaram funcionalidades como a lógica de turnos controlada por máquina de estados finitos, o sistema de perguntas e respostas e os cálculos de ataque, vida e vitória. As falhas identificadas foram corrigidas com ajustes no sistema de estados. A versão final, desenvolvida na engine Godot, consolidou-se como uma base estável e interativa para o aprendizado de Python.

4.2. Personagens e Enredo

O **PyJourney** adota uma mecânica de duelo simplificada, na qual o estudante aprende Python por meio da resolução de desafios. O progresso do jogador está diretamente ligado ao

acerto das respostas, tornando o domínio dos conceitos de programação um elemento essencial da jogabilidade. Cada carta representa uma ação de combate cujos valores de ataque e vida interagem com os do oponente, exigindo raciocínio lógico e estratégia.

A narrativa é estruturada em torno de um duelo simbólico entre o Herói, que representa o estudante em busca de conhecimento, e a Bruxinha, antagonista que personifica os desafios e obstáculos do aprendizado. As escolhas visuais e narrativas foram pensadas para reforçar o caráter pedagógico do jogo e criar um ambiente emocionalmente estimulante. As cores e traços dos personagens refletem contrastes entre otimismo e desafio, simbolizando o processo de superação e o equilíbrio entre dificuldade e progresso na jornada do aprendizado.

4.3 Conteúdo Educacional e Desafios

A estrutura pedagógica do **PyJourney** foi projetada para introduzir gradualmente os principais conceitos de Python, em correspondência com as disciplinas iniciais de programação. Cada nível de dificuldade, **Iniciante**, **Intermediário** e **Avançado**, aborda conteúdos equivalentes aos tradicionais da linguagem, promovendo progressão lógica e contínua do aprendizado.

Essa evolução é controlada por um sistema adaptativo baseado no arquivo *questions_objective.json*, que organiza o banco de perguntas por complexidade. Durante a partida, o jogo inicia com questões do nível **Iniciante** e, à medida que a antagonista perde pontos de vida (abaixo de 15 e depois de 7), o sistema substitui automaticamente as perguntas por níveis mais desafiadores. Essa mecânica garante uma curva de aprendizado coerente, vinculando o avanço do jogador à assimilação do conteúdo.

O jogador responde a perguntas ofensivas, que liberam cartas de ataque, e defensivas, que reduzem o dano recebido. A sequência segue uma hierarquia de dificuldade com turnos controlados por uma máquina de estados finitos. Cada personagem inicia com 20 pontos de vida, e o objetivo é reduzir a vida do oponente a zero, com *feedback* visual imediato sobre o progresso.

Para evitar lacunas cognitivas, o **PyJourney** elimina a aleatoriedade na seleção das perguntas, garantindo sequência lógica e acumulativa. O conteúdo foi elaborado com base em ementas e matrizes de competências reconhecidas, assegurando alinhamento com currículos formais.

No nível **Iniciante**, aborda variáveis e tipagem, desenvolvendo habilidades como identificar e declarar variáveis e diferenciar tipos primitivos (*int, str, float, bool*). O nível **Intermediário** trabalha **estruturas condicionais** e o uso de **operadores lógicos** e relacionais em blocos *if/elif/else*. O nível **Avançado** introduz laços de repetição, estimulando o uso de *for, while, break* e *continue* na manipulação de coleções de dados. Esse mapeamento demonstra que o **PyJourney** vai além da ludicidade, configurando-se como um recurso educacional estruturado, alinhado às competências essenciais dos primeiros semestres dos cursos de tecnologia.

4.4 Mecânicas e Fluxo de Jogo

A jogabilidade do **PyJourney** é organizada em turnos controlados por uma máquina de estados finitos (*GameState*), o que garante clareza e coesão no fluxo da partida. Cada jogador inicia com 20 pontos de vida e um espaço para cartas, tendo como objetivo reduzir a vida do oponente a zero.

Cada turno começa com uma pergunta objetiva de Python. A resposta correta concede uma carta ao jogador, enquanto a incorreta encerra sua vez e transfere o controle para a

ISSN: 2447-0767

Bruxinha, que realiza sua jogada automaticamente. Durante o turno da adversária, uma nova pergunta é apresentada, e o acerto reduz pela metade o dano recebido. À medida que o jogador diminui a vida da antagonista, o sistema eleva o nível de dificuldade e introduz tópicos mais complexos. O nível **Iniciante** aborda variáveis e tipagem, com ênfase na declaração e nos tipos primitivos (*int, str, float, bool*). O nível **Intermediário** explora estruturas condicionais (*if, elif, else*) e operadores lógicos e relacionais. O nível **Avançado** trata dos laços de repetição (*for, while, break, continue*), consolidando o raciocínio algorítmico e a autonomia do aprendiz.

Os desafios seguem essa progressão. No nível **Iniciante**, o jogador responde à pergunta "O que é uma variável em Python?", cuja opção correta é "Um espaço na memória que armazena um valor". No nível **Intermediário**, a questão "Qual expressão resulta em False?" tem como resposta correta "5 < 3 or 2 > 4". No nível **Avançado**, a pergunta "Qual será a saída do código? x = 0; while x < 3: print(x); x += 1" possui como resposta correta "0 1 2". Essas perguntas estimulam o raciocínio lógico e a compreensão prática da linguagem, indo além da simples memorização.

A fase de combate ocorre quando ambos os personagens mantêm cartas em campo. O jogador ataca primeiro e, se a carta inimiga permanecer ativa, ela realiza o contra-ataque. Esse ciclo de ação e resposta mantém o duelo dinâmico e reforça a aplicação estratégica do conhecimento adquirido.

As **cartas do Herói** apresentam valores mais modestos, exigindo uso tático para vencer a Bruxinha, cujas cartas possuem atributos superiores. À medida que a antagonista perde pontos de vida, o sistema aumenta a frequência de cartas mais fortes, mantendo o desafio proporcional ao desempenho do jogador.

Quando apenas uma carta permanece em campo, ela ataca diretamente o oponente, acelerando o desenrolar do duelo. Animações e *feedbacks* visuais de dano e progresso tornam a experiência intuitiva, envolvente e orientada ao reforço positivo do aprendizado.

4.5 Desenvolvimento do PyJourney

A implementação do **PyJourney** foi estruturada em uma arquitetura modular, na qual um núcleo de controle central gerencia a lógica do jogo em etapas bem definidas. Esse controlador organiza o fluxo entre a apresentação das perguntas, a execução das ações das cartas e a alternância de turnos, garantindo consistência e previsibilidade. Essa abordagem facilita a manutenção e a expansão do sistema, pois cada módulo é responsável por uma função específica.

O gerenciamento das cartas segue a lógica de gestão de recursos inspirada em jogos de estratégia, como *Magic: The Gathering*. Quando o jogador acerta uma pergunta, recebe automaticamente uma carta utilizável; em caso de erro, o turno é encerrado. Essa mecânica reforça a relação entre domínio conceitual e progressão no jogo. De forma semelhante, durante o ataque do adversário, o jogador responde a uma pergunta de maior complexidade. As respostas corretas reduzem o dano pela metade, enquanto os erros aplicam o impacto total.

A interface foi projetada para garantir fluidez e clareza, com botões, ícones e animações que orientam o jogador de forma intuitiva. A escolha de cores e elementos visuais sustenta a narrativa e estimula o foco no aprendizado. A Figura 1 apresenta o menu principal, ponto de partida para iniciar uma nova partida ou acessar o tutorial.

Figura 1. Tela de menu do PyJourney.

Conforme ilustrado na Figura 1, a tela de menu principal do **PyJourney** apresenta, ao fundo, o cenário que conecta o reino do herói à torre sombria da Bruxinha, em tons vibrantes de azul e verde. Em primeiro plano, o protagonista surge à esquerda e a vilã à direita, compondo o contraste entre determinação e desafio. Sobre essa arte, o
painel de navegação exibe as opções “Jogar”, “Como Jogar” e “Sair”, dispostas verticalmente em tipografia clara e botões de fácil acesso. Essa composição reforça a identidade narrativa e oferece uma interface intuitiva desde o primeiro contato.

A fusão entre o duelo de cartas e os desafios de programação é refletida na interface de jogo, concebida para ser imersiva e funcional. A Figura 2 apresenta a tela principal da partida, que reúne as áreas de cartas do jogador e da Bruxinha, indicadores de pontos de vida, campo de *feedback* e painel de perguntas. Essa organização permite decisões estratégicas integradas à aplicação prática do conhecimento de Python.

Figura 2. Tela da interface de jogo do PyJourney.

Conforme apresentado na Figura 2, a interface do **PyJourney** destaca, ao centro, uma caixa de diálogo com perguntas de múltipla escolha sobre Python. Nos cantos da tela estão os avatares e respectivos pontos de vida, com o jogador no canto inferior esquerdo e a Bruxinha no superior direito, oferecendo *feedback* visual imediato sobre o progresso. A indicação de turno e as cartas em campo reforçam a mecânica de alternância e o sistema de desafios que sustentam a experiência de aprendizado. As telas de vitória e derrota substituem o campo de batalha ao final da partida, apresentando mensagens e efeitos visuais que reforçam o feedback emocional e a motivação.

ISSN: 2447-0767

O **PyJourney** atingiu seus objetivos ao integrar mecânicas de *card game* e conteúdo pedagógico de Python, promovendo uma aprendizagem ativa em que o progresso depende da aplicação correta do conhecimento. O *feedback* imediato, expresso em vantagens ou penalidades, favorece a assimilação dos conceitos de forma lúdica. Em comparação a outros jogos educativos, como Mapa do Tesouro e World Prog, diferencia-se por seu foco exclusivo em Python e por associar desempenho cognitivo à progressão estratégica.

O desenvolvimento passou por várias iterações, com ajustes na lógica de turnos, correções nos cálculos de dano e otimização das animações. A adoção de uma máquina de estados finitos trouxe previsibilidade e clareza ao fluxo do jogo, além de facilitar a manutenção e o rastreamento de erros. Testes funcionais internos confirmaram a estabilidade e a eficiência do sistema.

O projeto alcançou estabilidade operacional, com sistemas e regras funcionando conforme o planejado. Como etapa futura, recomenda-se a validação empírica com estudantes, a ampliação do banco de questões e o aprimoramento da inteligência artificial da Bruxinha, visando à aplicação do jogo em contextos educacionais reais.

4.6 Análise das Implicações Pedagógicas e de Usabilidade

O **PyJourney** é um jogo concebido para unir aprendizado e ludicidade, e suas escolhas de design apresentam implicações pedagógicas e de usabilidade que merecem análise crítica. A mecânica de consequências diretas, em que erros resultam em desvantagens como perda de turno ou dano total, foi uma decisão intencional. O objetivo não é punir o jogador, mas reforçar a narrativa de um duelo estratégico em que o conhecimento é a principal ferramenta de vitória. Essa dinâmica de risco e recompensa estimula o engajamento e transforma desafios cognitivos em ações significativas dentro do jogo.

Para evitar frustração excessiva, o sistema adota uma curva de dificuldade progressiva, ajustada ao desempenho, permitindo que o jogador desenvolva confiança antes de enfrentar tópicos mais complexos. Ainda assim, a versão atual apresenta uma limitação pedagógica relevante: o *feedback* é restrito ao resultado imediato e não explica as razões de acertos ou erros. A inclusão de um sistema explicativo que contextualize as respostas é recomendada para futuras versões, a fim de completar o ciclo de aprendizagem.

Outro ponto a ser aprimorado refere-se à usabilidade e à avaliação empírica. Ainda não foram realizados testes formais com estudantes para medir motivação, engajamento e níveis de frustração. Pesquisas futuras devem contemplar essa etapa, validando o equilíbrio entre desafio e recompensa e garantindo que a experiência mantenha coerência pedagógica sem comprometer o aspecto lúdico.

5. Considerações Finais

Este trabalho partiu do desafio enfrentado por iniciantes em Python, cuja alta taxa de evasão e reprovação evidencia a necessidade de estratégias didáticas mais envolventes. O objetivo principal foi desenvolver o jogo educacional **PyJourney**, que integra conceitos fundamentais da linguagem, como variáveis, estruturas condicionais e laços de repetição, a mecânicas de card game, promovendo um ambiente lúdico e interativo. Os eixos de design instrucional narrativo e *feedback* adaptativo orientaram a construção da experiência de aprendizado.

A metodologia seguiu etapas sequenciais, da concepção (GDD) à implementação (Godot 4), incluindo prototipação, elaboração dos desafios e testes funcionais. Esse processo, guiado por princípios de *design thinking*, assegurou coerência pedagógica e técnica.

O **PyJourney**, disponível em https://devmario.itch.io/pyjourney, baseia sua progressão na resolução de desafios de programação: perguntas corretas liberam cartas ofensivas, enquanto respostas defensivas reduzem danos. O *feedback* imediato e a máquina de estados finitos garantem fluxo claro e previsível, tornando a experiência imersiva e eficaz.

Como trabalhos futuros, recomenda-se a realização de estudos empíricos com estudantes para avaliar a eficácia pedagógica e a usabilidade do jogo, além da ampliação do banco de questões, do aprimoramento da inteligência artificial da Bruxinha e da expansão dos elementos visuais e sonoros. Em síntese, o **PyJourney** une entretenimento e aprendizado ativo, auxiliando o estudante iniciante a dominar Python de forma significativa e transformando desafios cognitivos em experiências estratégicas e engajadoras.

6. Referências

AUDACITY. **Audacity**: the world's most popular audio editing and recording app. [S. l.]: Audacity Team, 2024. Disponível em: https://www.audacityteam.org/. Acesso em: 27 maio 2024.

BARBOSA, Alexandre de A.; FERREIRA, Dyego Í. S.; COSTA, Evandro B. Influência da linguagem no ensino introdutório de programação. *In*: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 25., 2014, Dourados. **Anais** [...]. Porto Alegre: Sociedade Brasileira de Computação, 2014. p. 612-621. DOI: 10.5753/cbie.sbie.2014.612.

BOSSE, Yorah. **Padrões de dificuldades relacionadas com o aprendizado de programação**. 2020. 270 f. Tese (Doutorado em Ciência da Computação) – Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2020.

CODE.ORG. **About us**. [S. l.]: Code.org, 2024. Disponível em: https://code.org/about. Acesso em: 27 maio 2024.

CUNHA, Kátia Gomes da. **Game Design Document**: o processo de produção de jogos educacionais e os educadores. 2020. Trabalho de Conclusão de Curso (Licenciatura em Computação e Informática) – Centro Multidisciplinar de Angicos, Universidade Federal Rural do Semi-Árido, Angicos, 2020.

FERNANDES, Naraline Alvarenga. **Uso de jogos educacionais no processo de ensino e de aprendizagem**. 2010. Trabalho de Conclusão de Curso (Especialização em Mídias na Educação) – Universidade Federal do Rio Grande do Sul, Alegrete, 2010.

FREITAS, Sara de. Are games effective learning tools? A review of educational games. **Educational Technology & Society**, [S. l.], v. 21, n. 2, p. 74-84, 2018. Disponível em: https://www.jstor.org/stable/26388380. Acesso em: 9 jun. 2025.

GODOT. **Introduction to Godot**. [S. l.]: Godot Engine, 2024. Disponível em: https://docs.godotengine.org/en/stable/getting_started/introduction/introduction_to_godot.html. Acesso em: 27 maio 2024.

HOLANDA, Wallace Duarte de; COUTINHO, Jarbele Cássia da Silva. World Prog: um jogo educacional para aprendizagem de conceitos básicos de programação. **RENOTE**, Porto Alegre, v. 20, n. 1, p. 213-222, 2022. DOI: 10.22456/1679-1916.126654.

LOPES, Michele. Game designer: habilidades, salários e perspectiva de carreira. **EBAC**

Online, 2023. Disponível em: https://ebaconline.com.br/blog/game-designer-o-que-faz. Acesso em: 4 maio 2024.

LOPES, Michele. O que é Photoshop e como aprender a usar. **EBAC Online**, 2023. Disponível em: https://ebaconline.com.br/blog/o-que-e-photoshop. Acesso em: 3 jun. 2024.

MARGULIEUX, Lauren E.; MORRISON, Briana B.; DECKER, Adrienne. Reducing withdrawal and failure rates in introductory programming with subgoal labeled worked examples. **International Journal of STEM Education**, [S. l.], v. 7, n. 1, p. 1-16, 2020. DOI: 10.1186/s40594-020-00222-7.

MARQUES, Pedro *et al*. Desenvolvimento de um jogo digital educacional para o ensino de pensamento computacional concorrente. *In*: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 10.; WORKSHOP DE INFORMÁTICA NA ESCOLA, 27., 2021, [Online]. **Anais** [...]. Porto Alegre: Sociedade Brasileira de Computação, 2021. p. 458-467.

PRODUÇÃO DE JOGOS. **Como é a carreira de um game designer**. 2022. Disponível em: https://producaodejogos.com/game-designer/. Acesso em: 16 jun. 2025.

PISKEL. **Piskel**: free online pixel editor. [S. l.], [c. 2024]. Disponível em: https://www.piskelapp.com/. Acesso em: 27 maio 2024.

PRINCE, Michael. Does active learning work? A review of the research. **Journal of Engineering Education**, [S. l.], v. 93, n. 3, p. 223-231, jul. 2004. Disponível em: https://engr.ncsu.edu/wp-content/uploads/drive/1smSpn4AiHSh8z7a0MHDBwhb_JhcoLQmI/2004-Prince_AL.pdf. Acesso em: 16 jun. 2025.

SILVA, Samara Salete da. **Jogos eletrônicos**: contribuições para o processo de aprendizagem. 2016. Trabalho de Conclusão de Curso (Licenciatura em Pedagogia) - Universidade Federal da Paraíba, João Pessoa, 2016.

VALINOR, Rodrigo. Google Docs: para que serve, como usar e funções da ferramenta. **Remessa Online**, 22 nov. 2022. Disponível em: https://www.remessaonline.com.br/blog/google-docs/. Acesso em: 25 nov. 2022.

YASSINE, Alaeeddine. A serious game for teaching Python programming language. *In*: EL OUAZIZE, Y. *et al*. (ed.). **Embedded systems and artificial intelligence**. Singapore: Springer, 2020. p. 389-397. (Lecture Notes in Electrical Engineering, v. 642).

Aplicação de DeepFace e OpenFace para Identificação de Sentimentos Básicos em Vídeos de Teleconsulta

Aurea Ferreira Nascimento[1], Fábio Castro Araújo[1]
[1]Departamento de Computação
Universidade Luterana do Brasil – Palmas – TO

aureaf11@rede.ulbra.br, fabio.araujo@ulbra.br

Resumo. *Este artigo descreve a implementação integrada de OpenFace (extração de Action Units – AUs via FACS) e da biblioteca DeepFace (integração de modelos pré-treinados para análise facial) para análise de emoções básicas em vídeos de teleconsulta psicológica. O sistema executa captura, detecção/alinhamento facial, extração de AUs, inferência probabilística de emoções, persistência transacional e visualização em dashboard. O escopo é operacional: não são reportadas métricas de acurácia/calibração/latência. A proposta posiciona-se como apoio ao profissional, adicionando sinal analítico complementar sem substituir o julgamento clínico.*

Palavras-chave: *reconhecimento facial; emoções básicas; FACS; AUs; Deep Learning; telepsicologia.*

1. Introdução

Nos últimos anos, a área de visão computacional tem experimentado avanços, especialmente no reconhecimento facial e na análise de expressões. Esses progressos são resultado direto da aplicação de técnicas de aprendizado profundo (*Deep Learning*) e do desenvolvimento de arquiteturas de redes neurais convolucionais (CNNs) capazes de processar grandes volumes de dados visuais em tempo real (Goodfellow, Bengio e Courville, 2016; Li e Deng, 2020). Tal evolução tem ampliado as possibilidades de aplicação dessas tecnologias em diferentes contextos, desde sistemas de segurança até interfaces inteligentes de interação homem-máquina (Cohn e De la Torre, 2015).

Nesse cenário, bibliotecas como o *DeepFace* e o *OpenFace* têm se consolidado como ferramentas robustas para a detecção e análise de expressões faciais. O *DeepFace*, desenvolvido por Taigman et al. (2014), emprega modelos de *deep learning* capazes de reconhecer e verificar faces com alta precisão, além de realizar classificação emocional a partir de *embeddings* faciais (Schroff, Kalenichenko e Philbin, 2015; Deng et al., 2019).

Já o *OpenFace*, baseado no *Facial Action Coding System* (FACS), fornece uma análise dos movimentos musculares faciais, permitindo mapear microexpressões sutis associadas a diferentes estados emocionais (Ekman e Friesen, 1978; Baltrusaitis et al., 2018). Ambas as soluções representam contribuições significativas para a objetivação da análise emocional.

A identificação das chamadas emoções básicas alegria, tristeza, raiva, medo, surpresa e neutralidade conforme descritas por Paul Ekman (Ekman e Friesen, 1971; Ekman, 1999), tem grande relevância em áreas como psicologia, psiquiatria, educação e atendimento ao cliente. No entanto, em contextos de teleconsulta psicológica, a interpretação das emoções pode se tornar mais desafiadora devido a fatores como limitações técnicas de vídeo, ausência de contato presencial e variabilidade na qualidade da comunicação online (Luxton et al., 2011; Shore, Schneck e Mishkind, 2020). Dessa
forma, surge a necessidade de investigar se ferramentas automatizadas podem oferecer indicadores para auxiliar o profissional de saúde na análise do estado emocional de seus

ISSN: 2447-0767

pacientes.

A partir desse contexto, formula-se a seguinte questão de pesquisa: como identificar de maneira objetiva as emoções básicas expressas por pacientes em sessões de teleconsulta psicológica? Para responder a essa questão, este artigo propõe a análise de vídeos de atendimentos simulados utilizando a aplicação combinada das bibliotecas *DeepFace* e *OpenFace*. O objetivo central deste estudo é propor e descrever uma arquitetura integrada *(OpenFace + DeepFace*) para detecção de emoções básicas em teleconsulta, abrangendo *pipeline*, interfaces, requisitos operacionais e integração com *dashboard* clínico.

2. Fundamentação Teórica

A análise de emoções faciais por meio de visão computacional representa um dos campos mais promissores da inteligência artificial aplicada à saúde e ao comportamento humano. Nos últimos anos, a capacidade de interpretar expressões a partir de vídeos comuns tornou-se tecnicamente viável graças ao avanço dos modelos de aprendizado profundo, que agora alcançam níveis de sensibilidade e especificidade antes restritos a estudos laboratoriais.

No contexto da telepsicologia, a utilização dessas ferramentas oferece suporte à observação clínica em ambientes virtuais, permitindo maior rastreabilidade das reações emocionais e enriquecendo o processo terapêutico com indicadores visuais objetivos.

A fundamentação teórica deste trabalho se organiza em cinco eixos principais: as emoções básicas, os métodos de análise facial, a classificação por *Deep Learning* (*DeepFace*), o mapeamento de *Action Units* via *OpenFace* e, por fim, as aplicações dessas tecnologias na telemedicina. Esses pilares fornecem a sustentação científica para o pipeline proposto no presente estudo.

2.1 Emoções Básicas

A teoria das emoções básicas, proposta por Paul Ekman e Wallace Friesen (1971; 1978), define um conjunto de expressões universais observáveis em todas as culturas humanas. As seis emoções fundamentais alegria, medo, surpresa, tristeza, nojo e raiva apresentam padrões musculares específicos e distinguíveis. Esses padrões foram mapeados e codificados no *Facial Action Coding System* (FACS), permitindo a padronização da análise facial para fins científicos e clínicos.

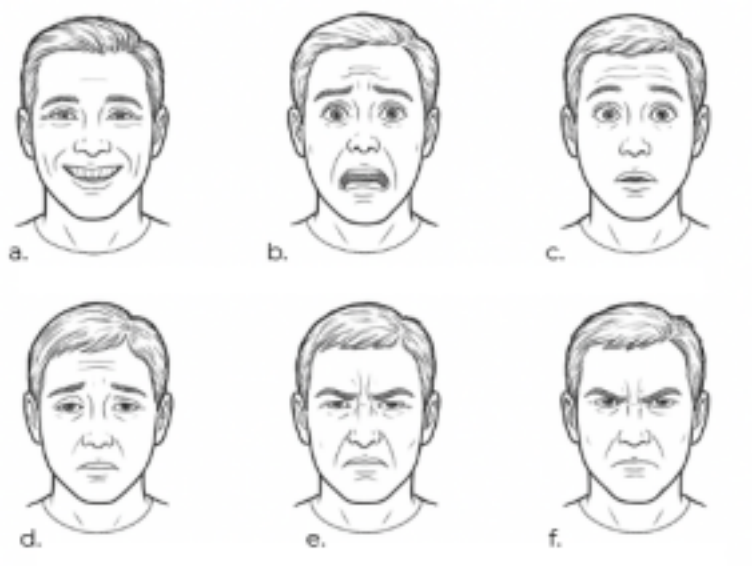


Figura 1. Expressões faciais típicas de seis emoções básicas: a. alegria, b. medo, c. surpresa, d. tristeza, e. nojo, f. raiva.

A importância dessa teoria para a computação emocional é reconhecida: a definição de categorias discretas de emoção possibilitou o treinamento de modelos supervisionados em bases rotuladas, como **FER-2013** e **AffectNet**, amplamente utilizadas para validar algoritmos de reconhecimento emocional (MOLLAHOSSEINI et al., 2017; LI; DENG, 2020).

A literatura contemporânea, no entanto, amplia a discussão sobre as limitações das categorias fixas. Pesquisadores como Barrett et al. (2019) argumentam que as emoções são construções contextuais, influenciadas por fatores sociais e culturais, e que o rosto nem sempre expressa fielmente o estado interno de uma pessoa. Essa reflexão é essencial para evitar interpretações deterministas e reforça a necessidade de leitura contextual dos dados algorítmicos.

Além disso, a análise moderna das emoções considera sua dimensão temporal e dinâmica. Cada expressão apresenta três fases: início, pico e recuperação que variam conforme a intensidade e a individualidade de cada sujeito. A observação dessas variações permite identificar não apenas emoções momentâneas, mas também padrões afetivos recorrentes, relevantes em avaliações psicológicas.

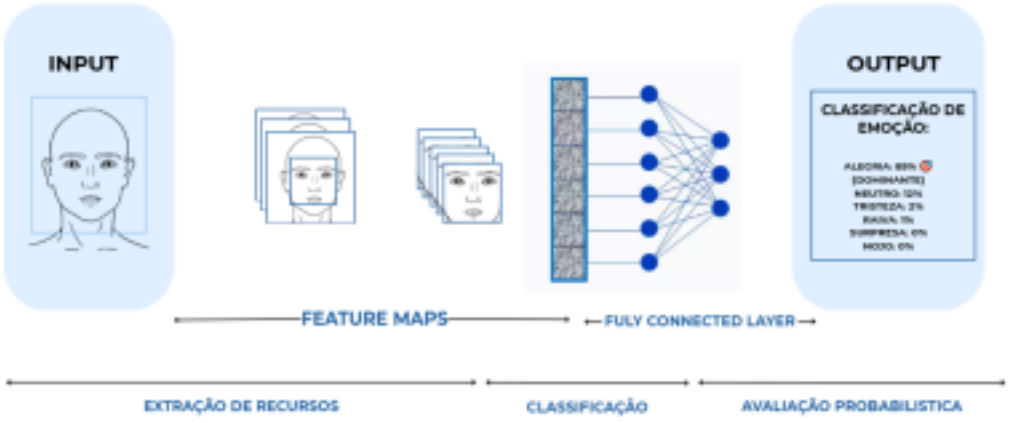
Dessa forma, a análise facial automatizada deve ser vista como ferramenta complementar ao olhar clínico, possibilitando medições de parâmetros sutis e fornecendo informações adicionais ao raciocínio profissional.

2.2 Análise de Expressões Faciais

Com o amadurecimento da visão computacional, a análise de expressões faciais tornou-se um dos campos mais desenvolvidos do reconhecimento de padrões. As metodologias podem ser agrupadas em duas abordagens principais: geométrica (baseada em *landmarks*) e baseada em aprendizado profundo *(Deep Learning*).

A abordagem geométrica identifica pontos-chave , olhos, sobrancelhas, nariz e boca e mensura distâncias e ângulos entre eles, garantindo interpretabilidade. Contudo,
apresenta limitações sob iluminação irregular, oclusões parciais (mãos, cabelo, óculos) ou ângulos fora do plano frontal (ZENG et al., 2009).

Já a abordagem baseada em *Deep Learning* utiliza redes neurais convolucionais (CNNs) para aprender representações hierárquicas diretamente das imagens, dispensando extração manual de características. As CNNs se mostram mais robustas frente a ruído, variação de pose e condições ambientais, sendo o método predominante nas pesquisas atuais



(GOODFELLOW; BENGIO; COURVILLE, 2016; LI; DENG, 2020).

ISSN: 2447-0767

Figura 2. Arquitetura Conceitual de uma CNN. (Fluxo: Entrada → Camadas Convolucionais → Classificação)

Nos sistemas modernos, ambas as técnicas são frequentemente combinadas: os *landmarks* garantem estabilidade e rastreabilidade da face, enquanto as CNNs operam sobre regiões faciais alinhadas, produzindo *embeddings* que alimentam classificadores de emoção. Essa combinação é adotada neste estudo, integrando o rastreamento geométrico via *OpenFace* e a classificação probabilística via *DeepFace*.

2.3 DeepFace: A Classificação por Deep Learning

O *DeepFace*, desenvolvido pelo Facebook AI Research (TAIGMAN et al., 2014), foi um marco no reconhecimento facial por aprendizado profundo, alcançando desempenho próximo ao humano na verificação de identidade (*LFW ~97,35%*). Sua implementação em *Python*, mantida por Serengil e Özpınar (2020), expandiu o escopo para reconhecimento, verificação, análise de atributos e classificação emocional.
O fluxo geral de processamento segue as etapas:

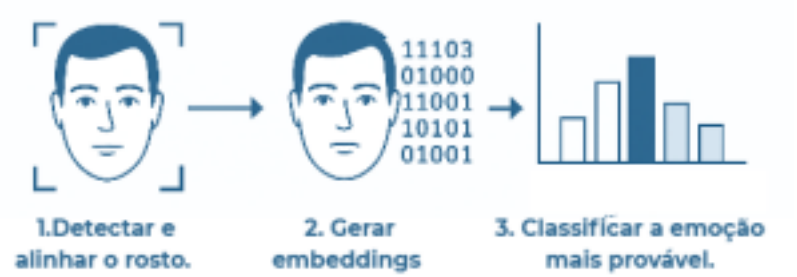


Figura 3. Fluxograma de Processamento.

O *DeepFace* suporta múltiplos *backbones*, entre eles:

- **VGG-Face** (PARKHI et al., 2015); **FaceNet** (SCHROFF et al., 2015);
- **ArcFace** (DENG et al., 2019);
- **DeepID** (SUN et al., 2014).

Essas arquiteturas, originalmente projetadas para verificação de identidade, podem ser adaptadas para análise de emoção mediante bases rotuladas como **FER-2013**. Em *benchmarks* públicos, a acurácia na classificação de emoções varia amplamente conforme dataset e protocolo experimental, geralmente inferior aos resultados de verificação facial (MOLLAHOSSEINI et al., 2017; LI; DENG, 2020).

Assim, qualquer aplicação clínica deve ser precedida por calibração contextual e validação experimental. A arquitetura modular do *DeepFace* permite combinar diferentes CNNs e ajustar sensibilidade e latência, tornando-o útil em protótipos clínicos exploratórios.

2.4 OpenFace e o Facial Action Coding System (FACS)

O *OpenFace* é um *toolkit* de código aberto para análise de comportamento facial, baseado no *Facial Action Coding System* (EKMAN; FRIESEN, 1978). Ele realiza detecção de *landmarks*, rastreamento de movimento, extração de *Action Units* (AUs) e estimativa de pose e olhar.

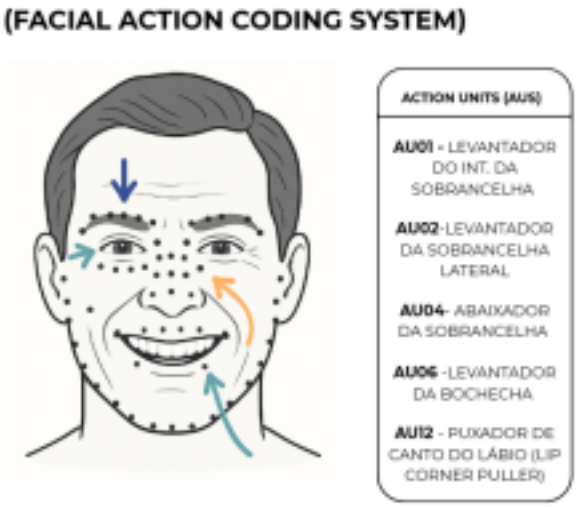


Figura 4: Ilustração do FACS e Action Units (AUs).

Diferentemente do *DeepFace*, o *OpenFace* não classifica emoções diretamente, mas identifica quais músculos foram ativados e com que intensidade. Cada emoção está associada a um conjunto de AUs, como:

Alegria: AU6 (orbicular dos olhos) e AU12 (elevação dos cantos da boca); **Tristeza**: AU1 (levantamento interno das sobrancelhas), AU4 (franzir), AU15 (cantos da boca abaixados); **Raiva**: AU4 + AU5 + AU7 + AU23.

Essas combinações permitem mapear microexpressões — movimentos rápidos (≈40–200 ms) — que revelam respostas emocionais involuntárias (Ekman, 2003) De acordo com Baltrušaitis et al. (2016, 2018), o OpenFace fornece rastreamento facial em tempo real e detecção de AUs com desempenho dependente de pose, iluminação
e base de dados. Sua confiabilidade é reduzida em chamadas de vídeo com iluminação e ângulos variáveis.

Por esse motivo, este estudo combina o *OpenFace* (extração objetiva de AUs) e o *DeepFace* (inferência probabilística de emoções), equilibrando aplicabilidade e abrangência analítica.

2.5 Aplicações de Análise Facial na Telemedicina

A pandemia de COVID-19 acelerou a adoção da telemedicina, consolidando plataformas de videoconferência como parte do atendimento psicológico remoto. Nesse contexto, a comunicação não verbal permanece essencial, mas sofre limitações técnicas que dificultam a leitura emocional.

A integração de sistemas de análise facial oferece suporte promissor, permitindo registrar expressões e visualizar tendências emocionais ao longo das sessões. Ferramentas como *DeepFace* e *OpenFace* possibilitam:

- Monitorar variações emocionais durante o atendimento;
- Identificar momentos de tensão, ansiedade ou desânimo;
- Gerar históricos visuais do comportamento emocional;
- Acompanhar respostas expressivas a estímulos terapêuticos.

Esses recursos podem enriquecer o planejamento terapêutico, desde que pautados por princípios éticos e legais. O tratamento de dados biométricos requer consentimento explícito, finalidade legítima e segurança da informação, conforme a Lei Geral de Proteção de Dados (LGPD – Lei nº 13.709/2018).

Além disso, é fundamental implementar auditorias de viés, avaliando consistência de desempenho entre grupos populacionais distintos idade, gênero, etnia e variações de iluminação (RHUE, 2019; TERHÖRST et al., 2022).

ISSN: 2447-0767

A interpretação dos resultados deve ser sempre técnica e humana: os algoritmos indicam probabilidades e padrões, mas a atribuição de significado e contexto cabe exclusivamente ao profissional de saúde.

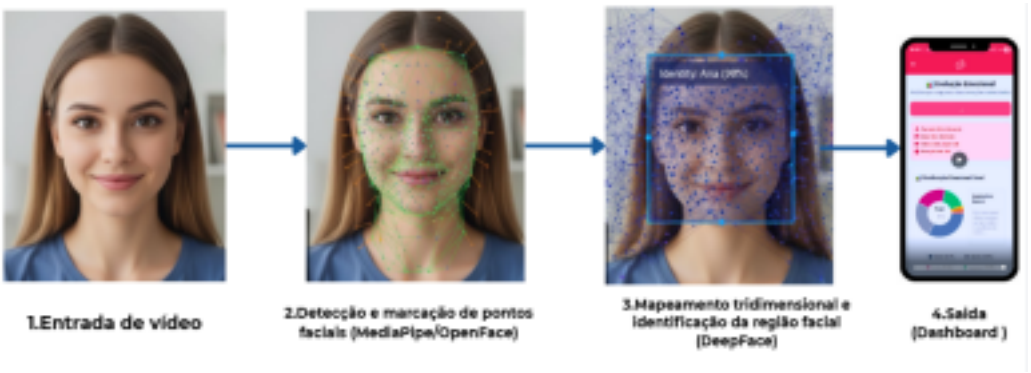


Figura 5: Diagrama da Solução Proposta (DeepFace + OpenFace).

3. Metodologia

Esta seção descreve o processo de implementação do sistema de análise de expressões faciais voltado à teleconsulta psicológica, detalhando os materiais utilizados, o fluxo de etapas, o pipeline computacional e a integração entre os componentes.

A metodologia adotada tem caráter aplicado e exploratório, uma vez que busca demonstrar a viabilidade técnica *end-to-end* do modelo, sem mensuração de métricas quantitativas de desempenho.

A abordagem é descritiva-operacional, centrada na apresentação do funcionamento do sistema e de suas potencialidades como ferramenta de apoio ao profissional de saúde mental.

3.1 Materiais

O ambiente experimental foi estruturado com base em ferramentas amplamente utilizadas em visão computacional e análise facial. O conjunto de materiais empregados inclui bibliotecas de software, infraestrutura de hardware e frameworks de visualização e persistência de dados. As Bibliotecas de Visão Computacional, OpenCV (*Open Source Computer Vision Library*), foram utilizadas para captura e manipulação de fluxos de vídeo, pré-processamento de quadros (*frames*) e integração com a *webcam*, e o *MediaPipe*, biblioteca desenvolvida pelo Google, foi responsável pela detecção facial, rastreamento de *landmarks* e alinhamento geométrico do rosto. Essas ferramentas asseguraram a qualidade da entrada visual, corrigindo variações de iluminação e posicionamento. Para a Análise Facial, o *OpenFace* foi empregado para extração de *Action Units* (AUs) e métricas relacionadas ao *Facial Action Coding System* (FACS), fornecendo dados interpretáveis sobre intensidade e ativação muscular. O *DeepFace* foi utilizado para classificação probabilística das emoções básicas, integrando modelos pré treinados (VGG-Face, *ArcFace*, *FaceNet* e DeepID) com bases de referência como FER
2013 e *AffectNet*. A Infraestrutura de Armazenamento utilizou o *Supabase*, plataforma *backend* as a service baseada em banco de dados relacional *PostgreSQL*, empregada para armazenar registros de emoções, *timestamps* e intensidades, com acesso seguro via API REST e autenticação JWT. A Interface de Visualização foi desenvolvida com *React Native*, *framework* para desenvolvimento de aplicações móveis multiplataforma, implementando o *dashboard* interativo com visualizações gráficas (linha, pizza e mapa de calor), com o auxílio de SVG e D3.js, bibliotecas auxiliares para renderização vetorial de alta precisão, integradas à camada de visualização do aplicativo. O *Hardware* Utilizado consistiu em um computador pessoal com processador Intel Core i5, GPU dedicada NVIDIA GTX 1650, 8 GB de RAM e

sistema operacional Windows 11, e uma *Webcam* HD (30 fps) utilizada para captura das expressões faciais durante as sessões simuladas. O conjunto de componentes operou de forma estável e responsiva, entregando processamento conforme os requisitos do experimento.

3.2 Métodos

O método adotado segue uma sequência lógica de etapas encadeadas, compondo um pipeline integrado que parte da aquisição de vídeo e termina na visualização analítica dos resultados.

Figura 6. Arquitetura geral do sistema: Captura de vídeo → Detecção e alinhamento (OpenCV/MediaPipe) → Extração de AUs (OpenFace) → Classificação emocional (DeepFace) → Persistência (Supabase) → Visualização (React Native

O processo metodológico empregado para validar a viabilidade técnica do sistema de análise facial e demonstrar o potencial de integração das bibliotecas DeepFace e *OpenFace* iniciou-se com a Aquisição de Dados de Vídeo (Etapa 1). Nela, a captura das imagens foi realizada por meio de webcam em um ambiente controlado, simulando uma teleconsulta psicológica. Os vídeos foram gravados em formato MP4, com resolução 720p e taxa de 30 *frames* por segundo (fps), buscando um equilíbrio entre qualidade e custo computacional. Durante as gravações, os participantes expressaram as seis emoções básicas (alegria, tristeza, raiva, medo, surpresa e nojo) para testar a estabilidade do rastreamento facial. Em seguida, na Detecção e Alinhamento Facial (Etapa 2), cada frame foi processado usando OpenCV e *MediaPipe* para localizar e alinhar a face. Esta etapa é crucial para a centralização e padronização da região facial, o que reduz o ruído e as variações de pose. O *MediaPipe* Face Mesh foi utilizado para fornecer até 468 landmarks, servindo como referência para um alinhamento geométrico de alta precisão. Na Extração de Features Comportamentais (Etapa 3), as imagens alinhadas foram enviadas ao *OpenFace*, que executou a análise muscular por meio das Action Units (AUs). Cada AU recebeu um valor de intensidade e um sinal binário de ativação, o que possibilitou a identificação de micro expressões (com duração inferior a 200 ms). Esses dados ofereceram um mapa detalhado da atividade muscular, de grande valor interpretativo para estudos psicológicos. Paralelamente, ocorreu a Classificação de Emoções (Etapa 4), onde os frames processados foram analisados pelo DeepFace. Este executou a classificação emocional utilizando modelos pré-treinados (VGG-Face + FER-2013), resultando em um vetor de probabilidades das seis emoções básicas, associado ao respectivo timestamp. Os resultados de ambas as análises foram direcionados à Persistência de Dados em Tempo Real (Etapa 5), onde foram armazenados no *Supabase*. Cada registro incluía o timestamp (momento da detecção), a emoção predita, a probabilidade, a intensidade (AUs) e um identificador único de sessão (idSessão). A segurança e a integridade dos registros foram garantidas por meio de autenticação via JWT e controle de acesso. Por fim, na Visualização Interativa dos Resultados (Etapa 6), os dados foram consumidos por um aplicativo móvel (*React Native*), que exibiu as informações de forma visual e dinâmica,
principalmente por meio de um Gráfico de Pizza que mostrava a proporção das emoções durante a sessão.

A adoção de um método *end-to-end* baseia-se em sua adequação a um contexto clínico exploratório, no qual a prioridade é a funcionalidade coerente e reprodutível, e não a mensuração estatística formal de desempenho.

Ao integrar bibliotecas *open source,* o sistema assegura transparência, baixo custo e replicabilidade, características essenciais para pesquisas acadêmicas em saúde. O uso do *Supabase* como camada de persistência permite futura expansão para análises, correlacionando séries temporais de emoções com contextos de fala ou discurso, fortalecendo o potencial clínico e analítico do modelo.

ISSN: 2447-0767

4. Resultados e Discussão

Esta seção apresenta os resultados obtidos com a execução do pipeline proposto, além de discutir suas implicações técnicas, operacionais e éticas. O foco principal está na funcionalidade da arquitetura integrada (OpenCV + MediaPipe + OpenFace + DeepFace + Supabase + React Native) e na análise crítica dos resultados observados durante as sessões simuladas de teleconsulta psicológica.

O

pipeline foi estruturado em módulos, permitindo substituições pontuais sem comprometer o fluxo central. A comunicação entre os módulos ocorre via APIs locais em formato JSON, garantindo a interoperabilidade entre o backend (Python) e o frontend (React Native). A detecção facial é executada com uma média de 30 fps, enquanto a inferência emocional opera entre 10 e 12 fps, assegurando fluidez em tempo quase real, mesmo em hardware intermediário.

Figura 7. Arquitetura e Comunicação.

O diagrama na Figura 7 detalha a arquitetura do sistema, ilustrando como os módulos interagem entre si. A comunicação entre os módulos ocorre via APIs locais, com dados transmitidos em formato JSON entre o backend em Python (*OpenCV, MediaPipe, OpenFace, DeepFace)* e o frontend em *React Native*. Os resultados processados são armazenados no Supabase, permitindo acesso e visualização dos dados pelo psicólogo através de um aplicativo móvel.

A arquitetura do sistema foi projetada para capturar e analisar as emoções faciais durante as teleconsultas psicológicas. O processo começa com a captura do vídeo, realizada pelo *OpenCV*, que acessa a câmera do dispositivo e captura os frames do paciente. Em seguida, os dados são enviados ao *API Gateway*, que organiza e direciona as informações para o processamento.

No processamento, o *MediaPipe* detecta os pontos-chave faciais e realiza o alinhamento da face. O *OpenFace* extrai as Action Units (AUs), enquanto o *DeepFace* analisa as expressões faciais e inferir as emoções predominantes do paciente. Esses dados

são organizados em um objeto JSON e armazenados no Supabase, garantindo acesso eficiente e em tempo real. O pós-processamento ocorre no frontend, onde os dados armazenados são visualizados por meio de um dashboard interativo desenvolvido em *React Native* com SVG e D3.js. Isso permite que o psicólogo monitore a evolução emocional do paciente ao longo da sessão.

Essa arquitetura modular e eficiente permite que o sistema realize a captura, análise, armazenamento e visualização das emoções do paciente de forma fluida e precisa, apoiando a prática psicológica em contextos remotos. A modularidade do pipeline garante que partes do sistema possam ser substituídas sem afetar o funcionamento central, proporcionando flexibilidade e facilidade de manutenção.

Figura 8. Fluxo operacional

Ele foi desenvolvido em *React Native*, com apoio das bibliotecas SVG e D3.js, e teve como objetivo apresentar de forma visual os dados processados pelo sistema. As informações de cada sessão (emoção predominante, intensidade das Action Units, e tempo de detecção) foram armazenadas no *Supabase* e carregadas automaticamente no aplicativo, permitindo ao profissional acompanhar as variações emocionais.

Durante a análise qualitativa dos resultados, observou-se que emoções de maior intensidade, como alegria e surpresa, apresentaram correspondência consistente entre as *Action Units* detectadas pelo *OpenFace* e as probabilidades de emoção calculadas pelo *DeepFace*. Essa coerência reforça a complementaridade entre os dois modelos e demonstra a eficácia do *pipeline* proposto para fins de observação emocional em teleconsultas.

Os testes realizados buscaram avaliar diferentes aspectos do sistema: a fluidez da execução, a consistência dos dados emocionais gerados, a estabilidade do rastreamento facial e a coerência entre predições sucessivas. Durante os testes, foi possível observar que o sistema manteve um bom desempenho mesmo em condições de hardware intermediário, com detecção facial operando em média a 30 fps e a inferência emocional entre 10 e 12 fps.

Nos testes qualitativos, emoções de maior intensidade, como alegria e surpresa, apresentaram uma correspondência consistente entre as Action Units detectadas pelo OpenFace e as probabilidades de emoção calculadas pelo DeepFace. Isso reforça a complementaridade entre os dois modelos e demonstra a eficácia do pipeline para fins de observação emocional em teleconsultas psicológicas.

Contudo, alguns desafios surgiram durante os testes, principalmente em relação às variações de iluminação e ângulo de captura, que afetaram a estabilidade da detecção facial em certas situações. Essas variações impactaram principalmente a detecção de emoções de baixa ativação, como tristeza e neutralidade. Apesar dessas limitações, o sistema manteve um desempenho adequado e forneceu dados úteis para interpretação clínica exploratória.

ISSN: 2447-0767

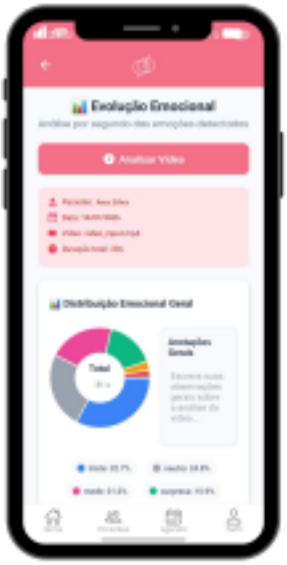


Figura 9. Dashboard de visualização.

Em síntese, o *dashboard* implementado demonstrou potencial para apoiar a prática psicológica em contextos remotos, fornecendo indicadores visuais claros e acessíveis sobre o comportamento emocional do paciente, sem substituir o olhar clínico do profissional.

5. Considerações Finais

O estudo demonstrou a viabilidade técnica e operacional de um *pipeline* integrado para análise automatizada de expressões faciais em teleconsulta psicológica, articulando *OpenCV*, *MediaPipe*, *OpenFace*, *DeepFace*, *Supabase* e *React Native* em fluxo contínuo da captura à visualização. O sistema executou as etapas de detecção/alinhamento facial, extração de *Action Units* e classificação probabilística de emoções com responsividade adequada ao uso exploratório, preservando rastreabilidade (AUs + probabilidades por *timestamp*) e interoperabilidade entre *backend* em *Python* e *dashboard* móvel. Observou se coerência prática entre picos emocionais mais intensos (ex.: alegria, surpresa) e padrões de AUs, com utilidade para revisão temporal de momentos críticos da sessão. Do ponto de vista clínico-operacional, a solução agrega sinal objetivo complementar ao julgamento profissional, sem pretensão de substituí-lo. Limitações relevantes permanecem: ausência de *ground truth* sincronizado e, portanto, de métricas formais (acurácia, F1, AUC, calibração); sensibilidade a condições de captura (iluminação, pose, oclusões, qualidade de câmera) e *dataset shift* frente a bases de treino (FER-2013, AffectNet).

6. Referências

BALTRUŠAITIS, T.; ROBINSON, P.; MORENCY, L.-P. *OpenFace: An Open Source Facial Behavior Analysis Toolkit*. IEEE WACV, 2016.
BALTRUŠAITIS, T.; ZADEH, A.; LIM, Y. C.; MORENCY, L.-P. *OpenFace 2.0: Facial Behavior Analysis Toolkit*. IEEE FG, 2018.

DENG, J.; GUO, J.; XUE, N.; ZAFEIRIOU, S. *ArcFace: Additive Angular Margin Loss for Deep Face Recognition*. CVPR, 2019.

EKMAN, P.; FRIESEN, W. V. *Facial Action Coding System (FACS): Manual*. 1978.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. MIT Press, 2016.

GOODFELLOW, I. J. et al. *Challenges in Representation Learning: FER-2013*. ICML Challenge, 2013.

GOOGLE. *MediaPipe Face Mesh — Documentação Técnica*. Disponível em: https://developers.google.com/mediapipe. Acesso em: 12 out. 2025.

INTEL. *OpenCV Documentation*. Disponível em: https://docs.opencv.org. Acesso em: 12 out. 2025.

LI, S.; DENG, W. *Deep Facial Expression Recognition: A Survey*. IEEE Transactions on Affective Computing, 13(3), 1195–1215, 2020.

META. *React Native — Getting Started*. Disponível em: https://reactnative.dev/docs/getting-started. Acesso em: 12 out. 2025.

MOLLAHOSSEINI, A.; HASANI, B.; MAHOOR, M. H. *AffectNet: A Database for Facial Expression, Valence, and Arousal in the Wild*. IEEE Transactions on Affective Computing, 10(1), 18–31, 2019.

SUPABASE. *Supabase Docs (Auth, Database, Realtime)*. Disponível em: https://supabase.com/docs. Acesso em: 12 out. 2025.

SERENGIL, S.; ÖZPINAR, A. *DeepFace (GitHub Repository)*. Disponível em: https://github.com/serengil/deepface. Acesso em: 12 out. 2025.

ISSN: 2447-0767

BIOS — O Código da Vida: Implementação Web Nativa de um Jogo de Desenvolvimento do Pensamento Computacional

Lucas Anselmo Pires Rosa[1], Henrique Dias Silva[1], Jackson Gomes de Souza[1]

[1]Departamento de Computação – Universidade Luterana do Brasil – Palmas – TO {lucasapr,henrique.dias}@rede.ulbra.br, jackson.souza@ulbra.br

Abstract. *This paper presents the migration of the educational game BIOS — The Code of Life to a native web version, aiming to broaden access to computational thinking education and reduce installation barriers. The study is based on the pillars of computational thinking—decomposition, pattern recognition, abstraction, and algorithms—integrated into a game-based learning approach. The implementation employs open web technologies such as HTML5, WebGL, and PWA, enabling direct browser execution and multiplatform compatibility. The paper outlines the pedagogical design, web architecture, and telemetry mechanisms supporting the system, as well as a proposed plan for learning and usability evaluation.*

Resumo. *O presente trabalho apresenta a migração do jogo educacional BIOS — O Código da Vida para uma versão web nativa, com o objetivo de ampliar o acesso ao ensino do pensamento computacional e reduzir barreiras de instalação. A pesquisa fundamenta-se nos pilares do pensamento computacional — decomposição, reconhecimento de padrões, abstração e algoritmos — integrados a uma proposta pedagógica baseada em jogos. A implementação utiliza tecnologias abertas, como HTML5, WebGL e PWA, permitindo execução direta em navegadores e compatibilidade multiplataforma. O artigo descreve o design pedagógico, a arquitetura técnica e os mecanismos de telemetria que sustentam o novo ambiente, além de propor um plano de avaliação de aprendizagem e usabilidade.*

1. Introdução

No contexto educacional atual, a alfabetização computacional consolidou-se como competência essencial do século XXI, equiparável às habilidades tradicionais de leitura e escrita [Wing 2006] e [Yadav et al. 2017]. Ela vai além do uso de tecnologias digitais, abrangendo a capacidade de aplicar princípios da ciência da computação para resolver problemas de forma lógica e criativa. O pensamento computacional, definido por Wing (2006), envolve processos como decomposição, reconhecimento de padrões, abstração e elaboração de algoritmos, pilares fundamentais para a resolução de desafios em diferentes contextos [Brackmann, 2017] e [Grover e Pea 2013].

A integração desses conceitos nas práticas pedagógicas tem se mostrado eficaz para promover a alfabetização digital e desenvolver a autonomia intelectual dos estudantes. Desde a educação básica, a compreensão desses fundamentos potencializa o raciocínio lógico e a organização do pensamento [Oliveira, Macêdo e Veronese 2014]. Nesse cenário, jogos educacionais digitais destacam-se como ferramentas valiosas, pois unem ludicidade, engajamento e desafio em ambientes interativos [Guarda e Goulart 2018] e [Michel, Pires e

Pessoa 2019].

Jogos baseados em pensamento computacional podem favorecer a aprendizagem ativa e contextualizada, permitindo ao estudante construir conhecimento por meio da experimentação e da resolução de problemas. Estudos indicam que o uso dessas ferramentas amplia a motivação e a participação dos alunos, ao mesmo tempo em que facilita a compreensão dos conceitos fundamentais da área [Guarda e Goulart 2018] e [Michel, Pires e Pessoa 2019].

Apesar de seu potencial, a adoção de jogos digitais enfrenta desafios técnicos, como limitações de acesso, compatibilidade e necessidade de instalação. Tecnologias web, como HTML5, WebGL e *Progressive Web Apps* (PWAs), são alternativas que permitem a execução direta em navegadores web e ampliam o alcance desses recursos [Garaizar e Guenaga 2014].

O projeto "BIOS — O Código da Vida" teve início com um mini game desenvolvido em Unity para dispositivos móveis [Silva e Souza 2024]. Inspirado em abordagens de ensino baseadas na decomposição e nos princípios fundamentais do pensamento computacional, o jogo apresenta desafios que exploram a resolução de problemas por meio de uma narrativa simbólica e enigmas progressivos, como os encontrados em [Brackmann 2017].

O presente trabalho realizou a migração do BIOS para a web, tornando-o acessível diretamente em navegadores web e eliminando a necessidade de instalação. A portabilidade amplia o público-alvo, facilita atualizações e permite a coleta de dados sobre o desempenho dos jogadores. O objetivo foi evidenciar como tecnologias abertas podem fortalecer a alfabetização computacional e ampliar o impacto de iniciativas educacionais. Também se busca demonstrar ganhos em acessibilidade, manutenção, escalabilidade e integração pedagógica.

As principais contribuições deste trabalho incluem: (i) o design pedagógico alinhado aos pilares do pensamento computacional (PC); (ii) a definição de uma arquitetura web modular com camadas de renderização, entrada, áudio e estado; (iii) a implementação de mecanismos de telemetria e balanceamento dinâmico de dificuldade; e (iv) a elaboração de um plano de avaliação com métricas de desempenho, engajamento e satisfação.

2. Fundamentos e Trabalhos Relacionados

2.1. Pensamento Computacional

O pensamento computacional (PC) consolidou-se como uma competência transversal para a resolução sistemática de problemas, o desenho de sistemas e a compreensão de comportamentos, ancorado em conceitos fundamentais da ciência da computação [Wing 2006]. Na literatura educacional, o PC tem se expandido como um campo de investigação voltado à integração de seus princípios nos currículos da educação básica, com ênfase em estratégias de ensino e em
evidências de aprendizagem associadas [Grover e Pea 2013]. Em síntese, o PC não se restringe à programação: trata-se de um conjunto de práticas cognitivas que estrutura o raciocínio para criar soluções eficazes e generalizáveis.

Os pilares clássicos do PC, decomposição, reconhecimento de padrões, abstração e algoritmos, funcionam de modo integrado: decompõe-se o problema em subpartes, identificam-se regularidades, modelam-se aspectos essenciais e, por fim, elabora-se um

ISSN: 2447-0767

procedimento finito para executar a solução. Esses elementos aparecem de forma recorrente como base conceitual em revisões e materiais acadêmicos, oferecendo um vocabulário comum para desenho de atividades e avaliação [Brackmann 2017], [Grover e Pea 2013] e [Wing 2006].

Para explicitar e avaliar competências, diferentes taxonomias e frameworks têm sido propostos. O modelo de Brennan e Resnick (2012) organiza o PC em três dimensões complementares: conceitos (p.ex., sequência, repetição, paralelismo), práticas (p.ex., iterar, depurar, remixar) e perspectivas (identidade e agência computacional), ampliando o foco para além da sintaxe de código. Em paralelo, o Computational Thinking Scale (CTScale) [Korkmaz, Çakir e Özden 2017] oferece um instrumento psicométrico com fatores como criatividade, pensamento algorítmico, cooperatividade, pensamento crítico e resolução de problemas, permitindo mensurar facetas do PC em contextos escolares.

No plano pedagógico, micro-desafios, tarefas curtas e conceituais, e minijogos têm se mostrado estratégias eficazes para introduzir e praticar habilidades de PC. O Bebras Challenge é exemplo consolidado: tarefas de 3–5 minutos mapeiam conceitos informáticos e habilidades de PC, podendo ser integradas ao currículo como atividades desencadeadoras [Dagienė e Sentance 2016]. Materiais unplugged, como Happy Maps (Code.org), operacionalizam algoritmos e sequenciamento em atividades graduais, com feedback imediato. Em ambientes digitais, minijogos específicos, como WAlgor [Michel, Pires e Pessoa 2019], demonstram potencial de engajamento e de prática deliberada de planejamento e depuração em contextos lúdicos.

Assim, um referencial de ensino que combine pilares conceituais, uma taxonomia clara de habilidades e um ecossistema de micro-desafios e minijogos favorece o alinhamento entre objetivos de aprendizagem, desenho de atividades e instrumentos de avaliação.

Na prática, o docente pode: (i) mapear objetivos dos pilares (ex.: “abstração por meio de modelagem visual”); (ii) adotar frameworks (Brennan–Resnick; CTScale) para planejar e monitorar progressões; e (iii) utilizar micro-desafios (Bebras, Happy Maps) e minijogos (WAlgor) para prática distribuída e avaliação formativa ao longo do curso.

2.2. Jogos Educacionais

Os jogos educacionais, ou serious games, têm se mostrado ferramentas eficazes no ensino de computação e pensamento computacional, por estimularem o engajamento, a motivação e a aprendizagem ativa. Quando bem alinhados aos objetivos pedagógicos, promovem maior motivação intrínseca e melhor desempenho dos estudantes [Hamari et al. 2016] e [Plass, Homer
e Kinzer 2015]. No ensino de programação, essas experiências tornam conceitos abstratos, como lógica e algoritmos, mais concretos e acessíveis, incentivando a persistência e o raciocínio lógico [Plass, Homer e Kinzer 2015] e [Gee 2003].

O engajamento dos alunos nesses jogos está ligado à teoria do flow de Csikszentmihalyi (1990), que descreve o estado de imersão obtido pelo equilíbrio entre desafio e habilidade. Jogos que ajustam dinamicamente sua dificuldade mantêm o interesse e evitam frustração, favorecendo a concentração e o aprendizado significativo [Kiili 2005] e [Bellotti et al. 2009].

De acordo com Shute (2008), o feedback imediato favorece a aprendizagem ao

permitir que erros sejam corrigidos prontamente, evitando sua consolidação e promovendo o progresso contínuo do aprendiz. Além disso, contribui para reduzir a incerteza e a sobrecarga cognitiva, favorecendo a motivação e o desempenho, especialmente em tarefas procedurais como programação e resolução de problemas.

O scaffolding (andaimagem ou suporte instrucional) também desempenha papel central na aprendizagem por jogos. De acordo com Gee (2003) e Yu et al. (2023), o uso de níveis graduais de dificuldade, tutoriais e pistas contextuais auxilia o estudante a progredir com autonomia, ajustando o suporte conforme seu avanço. Esse equilíbrio entre ajuda e desafio potencializa o engajamento e a retenção de conhecimento.

Em síntese, os serious games integram flow, feedback e scaffolding em um ambiente de aprendizagem ativo e motivador. Sua aplicação no ensino de pensamento computacional favorece o desenvolvimento de habilidades como abstração, resolução de problemas e raciocínio lógico, consolidando uma abordagem prazerosa e eficaz para a formação em computação.

2.3. Tecnologias para jogos

O uso de jogos digitais no ensino de computação e pensamento computacional (PC) tem se ampliado devido ao seu potencial de promover engajamento e aprendizagem ativa [Silva e Souza 2024]. A escolha da tecnologia de desenvolvimento impacta diretamente fatores como desempenho, acessibilidade e portabilidade. Entre as abordagens mais comuns, destacam-se os jogos móveis nativos, criados em engines como Unity, e os jogos web, desenvolvidos com tecnologías abertas como HTML5, Canvas, WebGL, WebAudio e IndexedDB [Cavalcante e Pereira 2018] e [Cantor e Jones 2012]. Cada modelo apresenta vantagens específicas conforme o contexto pedagógico.

O Unity é amplamente empregado em jogos educacionais por possibilitar desenvolvimento multiplataforma a partir de um único projeto, exportável para Android, iOS, Web e desktop [Cavalcante e Pereira 2018]. Sua estrutura integra módulos de física, renderização e áudio com forte suporte comunitário [Alves 2021]. Silva e Souza (2024) evidenciam seu uso na criação de mini games voltados ao ensino de PC, destacando o desempenho e a imersão proporcionados pela engine. Contudo, a necessidade de instalação e manutenção pode limitar o
uso em escolas com infraestrutura restrita, ainda que a execução offline seja vantajosa em locais com baixa conectividade.

O desenvolvimento web nativo, por sua vez, consolidou-se como alternativa acessível e multiplataforma. Tecnologias como Canvas e WebGL permitem gráficos acelerados por GPU diretamente no navegador, enquanto WebAudio e IndexedDB garantem som e persistência local de dados [Cantor e Jones 2012]. Essa abordagem elimina etapas de instalação, facilitando o acesso via link e reduzindo barreiras técnicas [AV Studios 2021]. Frameworks como Phaser e PixiJS ampliam a viabilidade de jogos leves voltados à lógica e programação [Orlova 2018], o que é ideal em projetos educacionais de larga escala.

Quanto ao desempenho, os jogos nativos ainda superam os web em tarefas gráficas complexas, explorando melhor o hardware e o processamento paralelo [Unity Technologies 2023]. Entretanto, melhorias recentes nas APIs web e o suporte a compilação via WebAssembly reduziram significativamente essa diferença, tornando os jogos web adequados

ISSN: 2447-0767

para atividades 2D e desafios lógicos típicos do ensino de PC [Alves 2021] e [Cavalcante e Pereira 2018]. Assim, a escolha tecnológica deve equilibrar performance e alcance, optando-se pelo Unity em aplicações imersivas e pela Web em experiências amplas e acessíveis.

Em termos de acessibilidade, os jogos web são mais inclusivos por funcionarem em qualquer navegador, sem necessidade de instalação [AV Studios 2021]. Além disso, utilizam linguagens amplamente difundidas, como JavaScript, reduzindo custos e curva de aprendizado [Orlova 2018]. Já o Unity, embora mais exigente tecnicamente, possibilita experiências imersivas e suporte a simulações complexas [Alves 2021].

Conclui-se que ambas as abordagens são relevantes no ensino de computação: os jogos nativos oferecem desempenho e imersão superiores, enquanto os jogos web ampliam o alcance e a acessibilidade. A integração entre essas tecnologias, por exemplo, exportações Unity via WebGL, constitui uma solução híbrida eficiente, equilibrando qualidade técnica e democratização do acesso à aprendizagem por meio de jogos digitais.

2.4. Trabalhos correlatos

O uso de jogos digitais para PC como recurso educacional tem se mostrado eficaz no ensino de pensamento computacional (PC), por unir interatividade, desafio e aprendizagem ativa [Wing 2006] e [Prensky 2012]. Em ambientes escolares, os jogos para computador são acessíveis e exploram plenamente o hardware disponível, permitindo experiências mais fluidas e integradas aos currículos de computação. Esses ambientes lúdicos possibilitam que os estudantes pratiquem os pilares do PC, decomposição, reconhecimento de padrões, abstração e algoritmos, de forma intuitiva, em atividades que valorizam o erro e a experimentação como partes naturais do processo de aprendizagem.

Entre os trabalhos nacionais, o Projeto Logicamente [Guarda e Goulart 2018] destaca-se por utilizar jogos digitais e atividades desplugadas para ensinar lógica e programação no ensino básico. O projeto promove desafios de raciocínio e criptografia em formato de jogo, aumentando o engajamento e a compreensão dos alunos sobre conceitos computacionais. De modo semelhante, Michel, Pires e Pessoa (2019) desenvolveram o WAlgor, um jogo de estratégia no estilo tower defense em que o jogador precisa planejar ações, reconhecer padrões e criar algoritmos para defender seu território. Ambos os projetos demonstram como jogos para PC podem integrar os fundamentos do pensamento computacional em contextos escolares de forma prazerosa e eficaz.

Nos minigames e puzzles algorítmicos, a proposta pedagógica baseia-se em desafios curtos e progressivos. Plataformas como Code.org e o projeto Hello Ruby [Liukas 2015] exemplificam o uso de atividades interativas para introduzir noções de lógica e algoritmos por meio de histórias e quebra-cabeças visuais. Essas experiências reduzem barreiras iniciais à programação textual e reforçam o aprendizado por meio da resolução de problemas concretos. No Brasil, Fávaro (2023) apresentou o jogo Zumbi Mind, que adapta desafios lógicos para um formato de tabuleiro digital, estimulando a criação de sequências de instruções e o raciocínio algorítmico, com resultados positivos em motivação e desempenho dos alunos.

Os trabalhos analisados demonstram que jogos educacionais para PC, especialmente os baseados em minigames e puzzles de lógica, são eficazes no desenvolvimento de habilidades de pensamento computacional. Esses jogos combinam entretenimento e

aprendizado, promovendo engajamento, resolução de problemas e autonomia do aluno. Tal abordagem fundamenta o projeto BIOS, que segue a mesma linha dos estudos revisados ao propor um mini game multiplataforma voltado à prática dos pilares do PC, unindo interatividade, narrativa e objetivos pedagógicos.

3. Narrativa, Mecânicas e Mapeamento Pedagógico

A narrativa do jogo acompanha a trajetória de Richard, que após um evento marcante busca retomar o controle sobre sua vida. O enredo é estruturado em ciclos de reflexão, desafio e superação, simbolizando o processo de reconstrução pessoal e cognitiva do personagem. Cada fase representa um novo passo em sua recuperação, articulando a progressão emocional e o avanço nas habilidades de resolução de problemas.

As mecânicas centrais baseiam-se em puzzles curtos e sequenciais, que exigem observação, planejamento e tomada de decisão. O jogador deve organizar elementos em ordem lógica, identificar padrões e ajustar suas ações por meio de tentativas sucessivas, recebendo feedback imediato que orienta o aprendizado. A progressão por capítulos e o uso de insígnias e conquistas reforçam a sensação de avanço e domínio das tarefas.

O design pedagógico do jogo mapeia suas mecânicas aos pilares do pensamento computacional. A decomposição aparece na divisão das tarefas complexas em etapas menores; o reconhecimento de padrões, na identificação de regularidades nas sequências e regras; a abstração, na capacidade de focar apenas nos elementos essenciais; e os algoritmos, na execução ordenada das ações até alcançar a solução. O feedback formativo, por meio de dicas graduais e
tentativas ilimitadas, transforma o erro em parte do processo de aprendizado, fortalecendo a autonomia e o raciocínio lógico do jogador.

4. Arquitetura e Implementação Web

O projeto adota uma arquitetura em camadas que organiza as responsabilidades entre lógica de jogo, controle de cenas, persistência e áudio, equilibrando desempenho e flexibilidade. Essa estrutura permite gerenciar de forma eficiente as etapas da experiência, desde menus e cenas narrativas até fases jogáveis e telas de resultado, com transições contínuas e facilidade para expansão futura. A persistência mantém o registro de variáveis globais, como moedas, vidas e progresso, além de calcular pontuação e fornecer feedback contextual. O sistema de áudio ajusta sons e efeitos dinamicamente, reforçando a imersão e a resposta às ações do jogador.

A exibição adapta-se às capacidades gráficas do dispositivo, otimizando o uso de imagens e animações leves para garantir fluidez e compatibilidade. As interações reconhecem toque e clique de forma unificada, com respostas visuais imediatas que indicam acertos e erros, assegurando acessibilidade em múltiplas plataformas. O progresso é controlado por uma estrutura central que registra o desempenho, gera indicadores e gerencia os estados de vitória ou derrota. Os dados são mantidos apenas durante a sessão, o que é adequado ao caráter experimental do protótipo.

Uma análise comparativa entre o desenvolvimento móvel e a abordagem web mostrou que plataformas nativas oferecem desempenho superior, mas com maiores custos e complexidade. A versão web, por sua vez, destaca-se pela leveza, rápida distribuição e

ISSN: 2447-0767

manutenção simplificada, utilizando ferramentas amplamente acessíveis e de baixo custo. Suas limitações concentram-se no suporte parcial a funcionalidades offline e na dependência de soluções externas para coleta de dados. Essas diferenças estão resumidas na Tabela 1, que apresenta os principais aspectos comparativos entre ambas as abordagens.

Tabela 1 – Comparativo técnico entre Unity (móvel) e Web (Phaser + React + Vercel)

Critério	Unity (Móvel)	Web (Phaser + React + Vercel)
Desempenho	Superior – compilação nativa, acesso direto à GPU e execução estável em 60 FPS.	Moderado – execução via engine JavaScript e dependente das otimizações do navegador.
Build Size	Maior (50–150 MB em média).	Reduzido (2–5 MB, com *assets* carregados sob demanda).
Tooling	Completo – editor visual, profiler e animação integrados.	Fragmentado – uso de VS Code, DevTools e bibliotecas externas.
Distribuição	Tradicional – depende de lojas e aprovação.	Instantânea – deploy imediato via Vercel e acesso por link direto.
Compatibilidade de Hardware	Limitada a Android 5.1+ e iOS 12+.	Ampla – roda em qualquer dispositivo com navegador moderno.
Manutenção	Complexa – versões depreciadas e recompilações frequentes.	Simplificada – updates via npm e deploy contínuo.
Curva de Aprendizado	Moderada/alta – exige domínio de C# e da engine Unity.	Moderada – utiliza JavaScript/TypeScript e APIs familiares da web.
Custos de Desenvolvimento	Médio/alto – licenças, taxas e requisitos de hardware.	Baixo – ferramentas e hospedagem gratuitas.

Capacidades Offline	Nativas – jogo funcional após instalação.	Limitadas – dependem de implementação via PWA.
Analytics e Telemetria	Integrados ao Unity e Firebase.	Externos – via Google Analytics ou Vercel Analytics.

5. Design de Fases e Balanceamento

O design do jogo foi estruturado em arcos narrativos que acompanham a jornada do personagem principal, unindo progressão narrativa e aumento gradual da complexidade dos desafios. A experiência se organiza em etapas que vão do prólogo introdutório ao desfecho avaliativo, guiando o jogador por fases de aprendizado, consolidação e domínio. A curva de dificuldade evolui de tarefas simples para desafios que exigem análise, planejamento e otimização de caminhos, aplicando os pilares do pensamento computacional, decomposição, reconhecimento de
padrões, abstração e algoritmos. O sistema de pontuação e feedback reforça o aprendizado ao oferecer avaliações percentuais, estrelas e mensagens adaptativas que estimulam a autorregulação.

Portanto o design adota matrizes bidimensionais para definir áreas navegáveis e obstáculos, incorporando elementos que reduzem frustração, como múltiplas tentativas, feedback imediato, checkpoints narrativos e dicas visuais. Embora ainda falte um editor visual e integração analítica, a estrutura modular e o gerenciamento centralizado de estado garantem consistência e expansão futura, sustentando um balanceamento progressivo e pedagógico.

6. Experimentos Técnicos

A prova de conceito avaliou a estabilidade e o desempenho da versão web do BIOS — O Código da Vida em diferentes dispositivos e navegadores. O sistema ajusta-se automaticamente a cada ambiente, mantendo fluidez e responsividade em computadores, tablets e celulares, inclusive de baixo desempenho.

Os resultados mostraram execução estável, baixo consumo de memória e resposta imediata às ações do jogador. O tempo de carregamento foi curto, beneficiado pelo uso de imagens otimizadas e compressão de recursos, o que reduziu significativamente o tamanho do jogo e facilitou o acesso em conexões mais lentas.

A aplicação manteve boa jogabilidade e adaptação visual em todos os testes, confirmando que a versão web é leve, estável e acessível, adequada para uso educacional em diferentes contextos sem necessidade de instalação. Essa flexibilidade consolida o BIOS Web como uma alternativa viável e inclusiva para o ensino do pensamento computacional.

7. Discussão

A versão web do BIOS — O Código da Vida ampliou o acesso e simplificou o uso do jogo

ISSN: 2447-0767

em ambientes escolares. Por rodar diretamente no navegador, elimina etapas de instalação e compatibilidade, tornando-se ideal para laboratórios e dispositivos modestos. O acesso por link e as atualizações automáticas reduzem custos e facilitam o uso em sala de aula.

Mesmo com pequenas limitações de desempenho e suporte offline, o uso de tecnologias abertas como HTML5, WebGL e PWA garante boa estabilidade e fluidez. No aspecto pedagógico, o jogo fortalece o papel do professor como mediador e se integra facilmente a atividades alinhadas à BNCC, estimulando os pilares do pensamento computacional, decomposição, padrões, abstração e algoritmos, de forma prática e motivadora.

8. Trabalhos Futuros

O projeto pode avançar com estudos de aprendizagem e experiência do usuário para avaliar o impacto do jogo em contextos reais. Também há potencial para criar um editor de fases voltado a professores, permitindo a personalização de desafios e maior integração pedagógica. Outras
melhorias incluem a coleta de métricas por pilar, expansão da acessibilidade, suporte a interações assíncronas e internacionalização. A evolução natural é tornar o BIOS uma PWA completa, com sincronização em nuvem e ferramentas analíticas que reforcem seu papel como plataforma aberta de ensino.

9. Conclusão

O BIOS — O Código da Vida se firmou como uma solução viável e acessível para o ensino do pensamento computacional. A versão web nativa mostrou-se leve, estável e adequada para diferentes contextos escolares, mantendo o foco em desafios curtos e feedback contínuo. O projeto comprova que tecnologias web abertas podem sustentar jogos educacionais de qualidade, com baixo custo e fácil distribuição. Além de sua contribuição técnica, o BIOS apoia práticas pedagógicas alinhadas à BNCC, promovendo uma aprendizagem ativa e significativa. Como próximos passos, prevê-se ampliar os recursos analíticos, evoluir o suporte PWA e criar ferramentas complementares, consolidando o BIOS Web como uma plataforma expansível para o ensino de computação e raciocínio lógico.

Referências

Alves, G. B. (2021) Desenvolvimento de jogos 2D no Unity para ensino de lógica de programação. Trabalho de Conclusão de Curso (Engenharia de Computação). Em Universidade Federal do Ceará, Fortaleza. Disponível em: https://repositorio.ufc.br/handle/riufc/58827. Acesso em: 2 out. 2025.

Aquino, E. M. e Diniz, F. V. S. e Barbosa, G. F. e Santos, C. A. (2024). Em busca do prêmio nobel: um jogo digital multiplataforma e multitemático. Disponível em: https://www.researchgate.net/publication/379413476_EM_BUSCA_DO_PREMIO_NOBEL_UM_JOGO_DIGITAL_MULTIPLATAFORMA_E_MULTITEMATICO_IN_SEARCH_OF_THE_NOBEL_PRIZE_A_MULTI-PLATFORM_AND_MULTI-_THEMATICAL_DIGITAL _GAME. Acesso em: 27 set. 2025.

AV studios. (2021). Why choose HTML5 games over native apps. Disponível em:

https://avstudios.com/what-we-do/educational-games/. Acesso em: 1 out. 2025.

Bellotti, F. e Berta, R. e Gloria, A. e Primavera, L. (2009). Adaptive Experience Engine for Serious Games. IEEE Transactions on Computational Intelligence and AI in Games, v. 1, n. 4, p. 264–280, dez. DOI: https://doi.org/10.1109/TCIAIG.2009.2035923. Acesso em: 4 out. 2025.

Brackmann, C. P. (2017). Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica. Tese (Doutorado em Informática na Educação) – Universidade Federal do Rio Grande do Sul, Porto Alegre. Disponível em: https://lume.ufrgs.br/handle/10183/172208. Acesso em: 27 set. 2025.

Brennan, K. e Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In: AERA. American Educational Research Association, Vancouver, Disponível em: https://scratched.gse.harvard.edu/ct/files/AERA2012.pdf. Acesso em: 27 set. 2025.

Cantor, M. e Brandon, J. (2012). WebGL: Up and Running. Sebastopol: O'Reilly Media.

Cavalcante, F. e Pereira, R. (2018). Comparativo entre game engines como etapa inicial para o desenvolvimento de um jogo de educação financeira. In: Anais do Workshop de Computação Aplicada, CEUR Workshop Proceedings, v. 2185, p. 110–118. Disponível em: https://ceur-ws.org/Vol-2185/CtrlE_2018_paper_110.pdf. Acesso em: 1 out. 2025.

Chen, C. e Lin, J. e Chang, H. (2022). Designing Scaffolding Mechanisms in Serious Games to Support Computational Thinking. Education Sciences (MDPI), v. 12, n. 8, p. 551. Disponível em: https://www.mdpi.com/2227-7102/12/8/551. Acesso em: 4 out. 2025.

Csikszentmihalyi , M. (1990). Flow: The Psychology of Optimal Experience. New York: Harper & Row.

Fávaro, A. L. O. (2023). Zumbi Mind: um jogo de tabuleiro para desenvolvimento de habilidades de pensamento computacional. Dissertação (Mestrado Profissional em Mídia e Tecnologia) – UNESP, Bauru. Disponível em: https://repositorio.unesp.br/entities/publication/132bc33c-9f4c-4cb3-98a4-ed0438d54f63. Acesso em: 30 set. 2025.

Garaizar, P. e Guenaga, M. (2014). A multimodal learning analytics view of HTML5 APIs: technical benefits and privacy risks. In: Proceedings of the International Conference on Learning Analytics and Knowledge (LAK 2014), p. 98–102. Disponível em: https://www.researchgate.net/publication/267266823_A_Multimodal_Learning_Analytics_View_of_HTML5_APIs_Technical_Benefits_and_Privacy_Risks. Acesso em: 30 set. 2025.

Gee, J. P. (2003). What Video Games Have to Teach Us About Learning and Literacy. New York: Palgrave Macmillan. Disponível em: https://www.researchgate.net/publication/220686314_What_Video_Games_Have_to_Teach_Us_About_Learning_and_Literacy. Acesso em: 5 out. 2025.

Grover, S. e Pea, R. D. (2013). *Computational Thinking in K–12: A Review of the State of the Field*. Educational Researcher, 42(1), 38–43. Disponível em: https://www.researchgate.net/publication/258134754_Computational_Thinking_in_K-12_A_Review_of_the_State_of_the_Field. Acesso em: 5 out. 2025.

ISSN: 2447-0767

Guarda, G.; Goulart, I. (2018). Jogos lúdicos sob a ótica do pensamento computacional: experiências do Projeto Logicamente. In: Anais do SBIE, Fortaleza, CE. Porto Alegre: SBC, 2018. p. 486–495. DOI: 10.5753/cbie.sbie.2018.486. Disponivel em: https://www.researchgate.net/publication/328735814_Jogos_Ludicos_sob_a_otica_do_Pensamento_Computacional_Experiencias_do_Projeto_Logicamente. Acesso em: 28 set. 2025.

Hamari, J. e Shernoff, D. e Rowe, E. e Coller, B. e Asbell-Clarke, J. e Edwards, T. (2016). Challenging games help students learn: An empirical study on engagement, flow and immersion in game-based learning. Computers in Human Behavior, v. 54, p. 170–179. DOI: https://doi.org/10.1016/j.chb.2015.07.045. Acesso em: 4 out. 2025.

Hoffmann, L. F. e Barbosa, D. N. F. e Martins, P. R. S. (2016). *Aprendizagem baseada em jogos digitais educativos para o ensino da matemática – um estudo-piloto a partir da utilização do Erudito*. Teknos: Revista Científica, 16(2), p. 38. Disponível em: https://doi.org/10.25044/25392190.820. Acesso em: 5 out. 2025.

Kiili, K. (2005). Digital game-based learning: Towards an experiential gaming model. The Internet and Higher Education, v. 8, n. 1, p. 13–24. DOI: https://doi.org/10.1016/j.iheduc.2004.12.001. Acesso em: 5 out. 2025.

Liukas, L. (2015). Hello Ruby: adventures in coding. Nova York: Feiwel & Friends.

Michel, F. e Pires, F. e Pessoa, M. (2019). WAlgor: um jogo de tower defense para o desenvolvimento do pensamento computacional e apresentação de algoritmos computacionais. In: Anais dos Workshops do VIII Congresso Brasileiro de Informática na Educação (CBIE 2019), Brasília, DF. Porto Alegre: SBC. p. 514–523. Disponível em: https://walgprog.gp.utfpr.edu.br/assets/files/articles/S1A3-article.pdf. Acesso em: 30 set. 2025.

Oliveira, G. A. A. de e Macêdo, J. A. e Veronese, G. L. (2014). GrubiBots Educacional: jogo para o ensino de algoritmos na educação básica. In: Anais do XXV SBIE, p. 592–601. Disponível em: http://milanesa.ime.usp.br/rbie/index.php/sbie/article/view/2988. Acesso em: 2 out. 2025.

Orlova, A. (2018). Educational Web Game for Learning Programming. Bachelor's Thesis – South-Eastern Finland University of Applied Sciences, Kotka. Disponível em: https://www.theseus.fi/bitstream/10024/158487/1/Orlova_Anna.pdf. Acesso em: 1 out. 2025.

Plass, J. L. e Homer, B. D. e Kinzer, C. K. (2015). Foundations of game-based learning. Educational Psychologist, v. 50, n. 4, p. 258–283. DOI: https://doi.org/10.1080/00461520.2015.1122533. Acesso em: 28 set. 2025.

Prensky, M. (2012). Aprendizagem baseada em jogos digitais. São Paulo: Senac.

Shute, V. (2008). Focus on formative feedback. Review of Educational Research, v. 78, n. 1, p. 153–189. DOI: https://doi.org/10.3102/0034654307313795. Acesso em: 5 out. 2025.

Silva, H. D. e de Souza, J. G. (2024). Criação de um Minigame para o Desenvolvimento de Habilidades do Pensamento Computacional. In: ENCOINFO - Congresso de Computação e Tecnologias da Informação, 26. Palmas - TO. Anais [...]. Palmas - TO: CEULP/ULBRA. p. 39 - 50. ISSN e-ISSN: 2447-0767 versão online. Disponível em: https://ulbra-to.br/encoinfo/edicoes/2024/artigos/criacao-de-um-minigame-para-o-desenvo

lvi
mento-de-habilidades-do-pensamento-computacional/. Acesso em: 06 out. 2025.

Unity Technologies. (2023). Unity Manual – WebGL Performance Considerations. 2023. Disponível em: https://docs.unity3d.com/Manual/webgl-performance.html. Acesso em: 2 out. 2025.

Van Der Spek, E. D. e Wouters, P. e Van Oostendorp, H. (2011). A taxonomy of feedback types for serious games. Computers & Education, v. 57, n. 4, p. 2372–2385. DOI: https://doi.org/10.1016/j.compedu.2011.06.006. Acesso em: 5 out. 2025.

Wing, J. M. (2006). Computational Thinking. Communications of the ACM, v. 49, n. 3, p. 33–35. PDF: https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf. Acesso em: 5 out. 2025.

Wouters, P. e Van Der Spek, E. e Oostendorp, H. (2013). Current Practices in Serious Game Research: A Review from a Learning Outcomes Perspective. In: Games-Based Learning Advancements for Multi-Sensory Human Computer Interfaces, IGI Global, p. 232–250. Acesso em: https://www.researchgate.net/profile/H-Oostendorp/publication/46707445_Current_Practices_
in_Serious_Game_Research_A_Review_from_a_Learning_Outcomes_Perspective/links/0912
f50d0f32d5cf33000000/Current-Practices-in-Serious-Game-Research-A-Review-from-a-Lear ning-Outcomes-Perspective.pdf. Acesso em: 5 out. 2025.

YU, S. et al. (2023). A systematic literature review of teacher scaffolding in game-based learning in primary education. Computers and Education: Artificial Intelligence, v. 5. Disponível em: https://www.sciencedirect.com/science/article/pii/S1747938X23000398. Acesso em: 5 out. 2025.

Yadav, A. e Mayfield, C. e Zhou, N. e Hambrusch, S. e Korb, J. (2017). Computational Thinking in Teacher Education. Disponível em: https://www.researchgate.net/publication/316446592_Computational_Thinking_in_Teacher_E ducation. Acesso em: 5 out. 2025.

ISSN: 2447-0767

Assistente Generativo para Terapeutas baseado em Arquitetura RAG

Anne Ingrid Chagas Vieira Oliveira[1], Jackson Gomes de Souza[1]

[1]Centro Universitario Luterano de Palmas (CEULP/ULBRA) ´
Avenida Teotonio Segurado 1501 Sul, Palmas - TO, CEP 77.019-900, Caixa Postal nº 85 ˆ

Abstract. This work presents a generative assistant for therapists based on the Retrieval-Augmented Generation (RAG) architecture. The solution integrates automatic transcription, speaker diarization, and contextual retrieval to sup port the analysis of clinical sessions. It aims to reduce documentation overload and optimize access to therapeutic records while ensuring compliance with data protection laws. The system combines technical accuracy, traceability, and ethi cal use of AI in mental health contexts.

Resumo. Este trabalho apresenta um assistente generativo para terapeutas, de senvolvido com base na arquitetura Retrieval-Augmented Generation (RAG). A soluc¸ao integra transcric¸ ˜ ao autom ˜ atica, diarizac¸ ´ ao e recuperac¸ ˜ ao contextual de ˜ informac¸oes para apoiar a an ˜ alise de sess ´ oes cl ˜ ´ınicas. O sistema busca reduzir a sobrecarga documental e otimizar o acesso a registros terapeuticos, mantendo ˆ conformidade com a LGPD. A aplicac¸ao combina precis ˜ ao t ˜ ecnica, rastreabili- ´ dade e etica no uso de IA em sa ´ ude mental. ´

1. Introduc¸ao˜

A documentac¸ao cl ˜ ´ınica tem sido reconhecida como um dos prin cipais fatores associados ao *burnout* entre profissionais da saude ´ [Agency for Healthcare Research and Quality 2023, Gaffney et al. 2022]. Estu dos indicam que medicos e terapeutas dedicam horas significativas ao preenchi- ´ mento de prontuarios eletr ´ onicos, muitas vezes estendendo o expediente para al ˆ em´ do horario cl ´ ´ınico, o que afeta sua saude mental e a qualidade do atendimento ´ [Budd 2023, Kruse et al. 2022, Tajirian et al. 2020]. Esse fenomeno ˆ e particularmente ´ cr´ıtico em contextos de saude mental, onde os registros s ´ ao extensos e predominantemente ˜ narrativos, exigindo alto esforc¸o cognitivo [Asgari et al. 2024, Vazquez 2025].

O avanc¸o das tecnologias de teleatendimento em saude mental ampliou o ´ acesso a servic¸os psicologicos e psiqui ´ atricos, especialmente durante a pandemia de ´ COVID-19, quando o isolamento social impulsionou o uso de plataformas digitais para continuidade dos atendimentos. A *telepsychology* mostrou-se eficaz e com paravel ´ a terapia presencial, permitindo a manutenc¸ ` ao de v ˜ ´ınculos e a expansao do ˜ cuidado a populac¸oes antes subatendidas [Bulkes et al. 2022, Greenwood et al. 2022, ˜ Association 2023, Abraham et al. 2021]. Nesse cenario, cresce o interesse por ferramen- ´ tas que reduzam o tempo gasto com registros manuais, sem comprometer a precisao e a ˜ confidencialidade das informac¸oes. ˜

Entre psicologos, a sobrecarga documental ´ e agravada pelo volume de atendi- ´ mentos e pela necessidade de manter registros cont´ınuos de evoluc¸ao e

intervenc¸ ˜ oes ˜
[Hammond et al. 2018]. Segundo o Censo da Psicologia de 2022, realizado pelo Con selho Federal de Psicologia, a maioria das(os) profissionais afirmou trabalhar entre 36 e 50 horas semanais, enquanto o segundo grupo mais numeroso declarou jornadas entre 21 e 35 horas semanais [Conselho Federal de Psicologia 2022].

Diante desse contexto, tecnologias baseadas em inteligencia artificial generativa ˆ tem se mostrado promissoras na reduc¸ ˆ ao da carga documental. Ferramentas de transcric¸ ˜ ao˜ automatica e modelos de linguagem de grande escala (LLMs) v ´ em sendo aplicados ˆ a` elaborac¸ao de notas cl ˜ ´ınicas e sumarizac¸ao de sess ˜ oes [Wang and Zhang 2024]. Estudos ˜ recentes apontam ganhos de produtividade e maior satisfac¸ao profissional com o uso de ˜ assistentes de documentac¸ao automatizados [Zhan et al. 2025]. ˜

Entretanto, a aplicac¸ao de modelos generativos em contextos terap ˜ euticos ˆ ainda impoe desafios, como a preservac¸ ˜ ao da privacidade, a precis ˜ ao sem in- ˜ ferencias indevidas e a adequac¸ ˆ ao˜ a linguagem psicol ` ogica [Blease and Rodman 2024, ´ Chen and Esmaeilzadeh 2024, Torous and Blease 2024]. Tambem se destaca a ne- ´ cessidade de rastreabilidade e de vinculac¸ao das respostas a evid ˜ encias verific ˆ aveis ´ [Adhikary et al. 2024]. Estrategias de engenharia de ´ *prompt* contribuem para orien tar os modelos, mas se mostram limitadas quando o historico de interac¸ ´ oes ˜ e extenso ´ [Sahoo et al. 2024].

Esses desafios motivam a adoc¸ao de abordagens que combinem recuperac¸ ˜ ao˜ e gerac¸ao de linguagem natural, como a arquitetura ˜ *Retrieval-Augmented Generation* (RAG). Este trabalho propoe o desenvolvimento de um assistente generativo baseado em ˜ RAG, voltado a apoiar psicologos e terapeutas na consulta e interpretac¸ ´ ao de informac¸ ˜ oes ˜ cl´ınicas. A soluc¸ao atua como recurso complementar, reduzindo o tempo de busca e pro- ˜ movendo uma visao contextualizada do paciente. ˜

O sistema integra uma interface em linha do tempo e um chat interativo, que per mitem a navegac¸ao cronol ˜ ogica entre sess ´ oes e consultas assistidas a partir de ˜ *prompts* estruturados. Alem disso, incorpora mecanismos de conformidade com a Lei Geral de ´ Protec¸ao de Dados (LGPD) e diretrizes de seguranc¸a e rastreabilidade. ˜

2. Fundamentos e Trabalhos Relacionados

Estudos recentes tem explorado o uso de sistemas de Reconhecimento Autom ˆ atico de ´ Fala (ASR) e diarizac¸ao para apoiar a an ˜ alise de interac¸ ´ oes cl ˜ ´ınicas. O modelo *Whisper* (OpenAI, 2022) destaca-se pela robustez a sotaques, ru´ıdo e multiplos idiomas, mostrando ´ boa generalizac¸ao em contextos terap ˜ euticos [Zolnoori et al. 2024, Adedeji et al. 2024]. ˆ Ja a ´ *pyannote.audio* vem sendo amplamente empregada para segmentar o audio em ´ turnos de fala e identificar locutores por meio de *speaker embeddings* e *clustering* [O'Shaughnessy 2025, Medaramitta 2021]. Essa combinac¸ao tem se mostrado essencial ˜ em abordagens voltadas a documentac¸ ` ao automatizada de sess ˜ oes e ˜ a construc¸ ` ao de sis- ˜ temas de apoio cl´ınico baseados em IA.

Durante a pandemia, o volume de sessoes remotas aumentou abruptamente, exi- ˜ gindo que as plataformas de telepsicologia nao s ˜ o garantissem a continuidade do cuidado, ´ mas tambem a documentac¸ ´ ao automatizada para que cl ˜ ´ınicos pudessem preservar qua lidade, rastreabilidade e eficiencia [Sablone et

ISSN: 2447-0767

al. 2024]. Neste contexto, a combinac¸ ˆ ao˜ de ASR/diarizac¸ao com fluxos automatizados de registro, sob a forma de transcric¸ ˜ ao, ˜
identificac¸ao de interlocutores e estruturac¸ ˜ ao de fala, tornou-se parte cr ˜ ´ıtica da infraestru tura de teleatendimento, permitindo que terapeutas e instituic¸oes pudessem dedicar mais ˜ tempo a intervenc¸ ` ao cl ˜ ´ınica e menos aos processos administrativos.

Contudo, a transcric¸ao isolada n ˜ ao˜ e suficiente para apoiar decis ´ oes cl ˜ ´ınicas ou sintetizar o historico de forma significativa. Nesse contexto, abordagens de ´ *Retrieval Augmented Generation* (RAG) tem se destacado como alternativas promissoras ao com- ˆ binar a recuperac¸ao de informac¸ ˜ oes relevantes com a gerac¸ ˜ ao de linguagem natural con- ˜ textualizada [Amugongo et al. 2025]. No RAG, a indexac¸ao converte documentos e ˜ transcric¸oes em vetores de ˜ *embeddings* semanticos; a recuperac¸ ˆ ao seleciona os fragmen- ˜ tos mais relevantes diante de uma consulta; e a gerac¸ao produz respostas com base nos ˜ textos recuperados. Esse fluxo reduz a dependencia da “mem ˆ oria” interna do modelo, ´ melhora a rastreabilidade das respostas e ajuda a mitigar alucinac¸oes, aspectos cruciais ˜ em contextos cl´ınicos [Yang et al. 2024].

Conforme Zhang (2025), “a etapa de indexac¸ao constitui um passo importante ˜ para o alto desempenho em sistemas baseados em RAG”, pois sua estruturac¸ao impacta ˜ diretamente as fases de recuperac¸ao e gerac¸ ˜ ao. Essa etapa define como os dados s ˜ ao or- ˜ ganizados, armazenados e posteriormente acessados. Para lidar com bases crescentes e heterogeneas, os documentos s ˆ ao segmentados em partes menores e semanticamente co- ˜ esas, denominadas *chunks*. Cada *chunk* preserva o sentido local do texto e permite gerar representac¸oes vetoriais mais precisas, reduzindo o risco de dispers ˜ ao sem ˜ antica. Se- ˆ gundo Koshorek (2018), a segmentac¸ao textual pode ser tratada como uma tarefa supervi- ˜ sionada, na qual o modelo identifica limites significativos entre unidades de informac¸ao, ˜ aumentando a granularidade da indexac¸ao e a qualidade das consultas contextuais. ˜

Aplicac¸oes m ˜ edicas e psicol ´ ogicas baseadas em RAG v ´ em sendo investigadas, ˆ por exemplo, Gargari (2025) demonstrou ganhos expressivos em precisao de respostas ˜ e transparencia quando comparadas a LLMs convencionais. Outros estudos, como o de ˆ Bernardi (2023), destacam a importancia da qualidade dos dados cl ˆ ´ınicos e da governanc¸a de informac¸ao para garantir infer ˜ encias confi ˆ aveis e rastre ´ aveis. Essa literatura sustenta ´ a adoc¸ao da RAG como componente central de sistemas de suporte ˜ a decis ` ao em sa ˜ ude, ´ desde que combinada a mecanismos de auditoria e controle de acesso.

Alem dos aspectos t ´ ecnicos, o design de interac¸ ´ ao e a experi ˜ encia do usu ˆ ario (UX) ´ sao decisivos para a aceitac¸ ˜ ao de ferramentas cl ˜ ´ınicas baseadas em IA. Estudos indicam que interfaces empaticas, acess ´ ´ıveis e centradas no usuario aumentam o engajamento e ´ reduzem o abandono em contextos de saude mental [Vial et al. 2022, Kaveladze 2022]. ´ Em aplicac¸oes terap ˜ euticas, a interface deve minimizar a carga cognitiva dos profissionais ˆ e transmitir confianc¸a, clareza e controle sobre os dados — fatores essenciais em sistemas de IA explicavel e centrada no cuidado [Donoso-Guzm ´ an et al. 2025]. ´

Outro aspecto essencial envolve as implicac¸oes ˜ eticas e legais do uso de inte- ´ ligencia artificial em sa ˆ ude. A Lei Geral de Protec¸ ´ ao de Dados (LGPD) estabelece ˜ princ´ıpios de finalidade, necessidade e seguranc¸a para o tratamento de dados sens´ıveis. Em ambito internacional, frameworks recentes destacam transpar ˆ encia, responsabili- ˆ dade e nao malefic ˜ encia como pilares para o uso ˆ

etico da IA em contextos cl ´ ´ınicos [Saeidnia et al. 2024, Tavory 2024]. Assim, aplicac¸oes em psicologia e psiquiatria de- ˜ vem adotar praticas de anonimizac¸ ´ ao, consentimento informado e controle de acesso, ˜
garantindo o uso etico e audit ´ avel dos dados terap ´ euticos. ˆ

Em s´ıntese, a literatura revisada indica que a integrac¸ao entre ASR, diarizac¸ ˜ ao˜ e RAG representa um caminho promissor para a documentac¸ao automatizada em sa ˜ ude ´ mental. Contudo, o sucesso dessas soluc¸oes depende tanto da robustez t ˜ ecnica quanto ´ do respeito as dimens ` oes humanas e ˜ eticas do cuidado. O trabalho proposto alinha-se a ´ essa perspectiva, buscando equilibrar inovac¸ao tecnol ˜ ogica, confiabilidade informacional ´ e responsabilidade social na aplicac¸ao da IA em contextos cl ˜ ´ınicos.

3. Cenarios de Uso e Requisitos ´

Esta sec¸ao descreve os contextos de aplicac¸ ˜ ao da soluc¸ ˜ ao proposta, as personas envolvidas ˜ e os principais requisitos tecnicos que orientaram seu desenvolvimento. ´

3.1. Personas

Foram definidos tres perfis principais de interac¸ ˆ ao com o sistema: o terapeuta, respons ˜ avel ´ por conduzir as sessoes, consultar registros anteriores e utilizar o assistente generativo ˜ como apoio a an ` alise e elaborac¸ ´ ao de notas; o paciente, que participa das sess ˜ oes e tem ˜ suas falas transcritas para posterior consulta; e o perfil de *compliance*/gestao, voltado ˜ a` supervisao de pol ˜ ´ıticas institucionais, requisitos legais e metricas de uso. ´

3.2. Cenarios de Uso ´

A soluc¸ao foi projetada para apoiar o terapeuta em diferentes momentos do fluxo te- ˜ rapeutico: na revis ˆ ao pr ˜ e-sess ´ ao, permitindo o acesso a resumos e menc¸ ˜ oes relevantes de ˜ atendimentos anteriores; na consulta tematica, em que o assistente identifica padr ´ oes de ˜ comportamento, sentimentos ou topicos recorrentes ao longo do hist ´ orico do paciente; e ´ no apoio a elaborac¸ ` ao de notas, oferecendo rascunhos estruturados ap ˜ os a sess ´ ao, sempre ˜ sujeitos a revis ` ao humana. ˜

3.3. Requisitos Funcionais

Os requisitos funcionais (RFs) definem o comportamento e as funcionalidades espec´ıficas que o sistema deve executar. A Tabela 1 a seguir apresenta a lista detalhada dos requisitos funcionais identificados para a soluc¸ao proposta. ˜

Tabela 1. Requisitos Funcionais da Soluc¸ao˜

ID	Requisito	Descric¸ao˜
RF01	Ingestao de ˜ audio/v ´ ´ıdeo	Permitir o envio de arquivos de sessoes cl ˜ ´ınicas em audio ou v ´ ´ıdeo para processamento.
RF02	Transcric¸ao e ˜ diarizac¸ao˜	Converter automaticamente o audio em texto e ´ identificar os respectivos

ISSN: 2447-0767

		falantes.
RF03	Visualizac¸ao temporal ˜	Exibir as transcric¸oes sincronizadas com o tempo ˜ da sessao, permitindo navegac¸ ˜ ao entre trechos. ˜
RF04	Assistente generativo (RAG)	Oferecer um assistente com gerac¸ao de respostas ˜ baseadas em recuperac¸ao de contexto. ˜
RF05	Consulta contextual	Permitir consultas interativas via chat com base no conteudo transcrito e armazenado. ´

Os requisitos funcionais apresentados na Tabela 1 descrevem as principais ca pacidades que o sistema deve oferecer para apoiar o trabalho cl´ınico. Esses requisitos abrangem desde a ingestao de dados multimodais (˜ audio e v ´ ´ıdeo) ate o uso de t ´ ecnicas de ´ gerac¸ao aumentada por recuperac¸ ˜ ao (RAG). Al ˜ em disso, incluem funcionalidades volta- ´ das a transcric¸ ` ao autom ˜ atica, organizac¸ ´ ao temporal das sess ˜ oes e interac¸ ˜ ao por meio de ˜ consultas em linguagem natural, assegurando suporte eficiente a an ` alise e documentac¸ ´ ao˜ terapeutica. ˆ

3.4. Requisitos Nao Funcionais ˜

Os requisitos nao funcionais da soluc¸ ˜ ao abrangem dimens ˜ oes de qualidade, como privaci- ˜ dade, seguranc¸a, desempenho e disponibilidade. A arquitetura foi projetada para garantir conformidade com a LGPD, aplicando criptografia em repouso e em transito, al ˆ em de ´ autenticac¸ao segura e controle de acesso. O sistema tamb ˜ em busca manter baixa lat ´ encia ˆ nas consultas, alta disponibilidade em cenarios de m ´ ultiplas requisic¸ ´ oes e registro de logs ˜ auditaveis para rastreabilidade e versionamento das interac¸ ´ oes, assegurando a confiabili- ˜ dade e a integridade das informac¸oes cl ˜ ´ınicas.

4. Arquitetura da Soluc¸ao˜

4.1. Visao geral ˜

A arquitetura da soluc¸ao foi projetada para estabelecer um fluxo estruturado e escal ˜ avel de ´ processamento de dados cl´ınicos. O diagrama geral da arquitetura, apresentado na Figura 1, representa a interac¸ao entre os principais componentes do sistema, evidenciando o ˜ percurso dos dados desde a ingestao inicial at ˜ e a gerac¸ ´ ao da resposta final. Essa vis ˜ ao˜ integrada permite compreender como os modulos se comunicam e como os servic¸os de ´ inteligencia artificial s ˆ ao incorporados ao fluxo para garantir efici ˜ encia, rastreabilidade e ˆ conformidade com diretrizes regulatorias. ´

Figura 1. Diagrama Geral da Arquitetura

Fonte: Elaborado pelo autor (2025)

A Figura 1 ilustra o *pipeline* sequencial do sistema. O fluxo inicia com a ingestao˜ de arquivos de audio e v ´ ´ıdeo dos atendimentos, seguida pela transcric¸ao autom ˜ atica e ´ diarizac¸ao das falas. Em seguida, os dados s ˜ ao normalizados, segmentados temporal- ˜ mente e estruturados para armazenamento eficiente.

Apos a preparac¸ ´ ao dos dados, ocorre a montagem do ˜ *prompt*, que integra as transcric¸oes, os metadados cl ˜ ´ınicos relevantes e a pergunta do usuario, sendo ent ´ ao envi- ˜
ado a um servic¸o de IA generativa. Por fim, o pos-processamento extrai trechos relevan- ´ tes, gera destaques na linha do tempo da sessao e prepara a resposta para a interface do ˜ usuario. ´

4.2. Transcric¸ao e Diarizac¸ ˜ ao˜

A fase de transcric¸ao e diarizac¸ ˜ ao˜ e essencial para converter registros de sess ´ oes cl ˜ ´ınicas em representac¸oes textuais estruturadas e alinhadas temporalmente. Para a transcric¸ ˜ ao˜ automatica de fala, ´ e utilizado o modelo ´ *Whisper*, um sistema de ASR (Automatic Speech Recognition) baseado em redes neurais profundas, capaz de lidar com variac¸oes fon ˜ eticas ´ do portugues brasileiro, sotaques regionais e ru ˆ ´ıdo de fundo moderado.

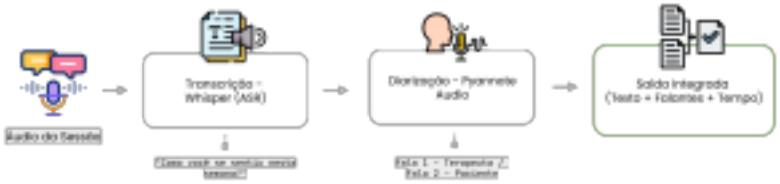


Figura 2. Etapa de Pre-Processamento de Sess ´ ao Terap ˜ eutica ˆ

Fonte: Elaborado pelo autor (2025).

Conforme ilustrado na Figura 2, o processo inicia-se com a ingestao do ˜ audio da ´ sessao. Em seguida, o ˜ audio ´ e submetido ao modelo ´ *Whisper* para a gerac¸ao da transcric¸ ˜ ao˜ textual e a biblioteca ` *pyannote.audio* para a diarizac¸ao de falantes. ˜

A *pyannote.audio* utiliza tecnicas de ´ *speaker embedding* e *clustering* para segmen tar a fala em blocos temporalmente delimitados e identificar automaticamente o locutor em cada trecho. Essa segmentac¸ao organiza os turnos de fala com identificac¸ ˜ ao de falante ˜ e *timestamps* precisos, que posteriormente

ISSN: 2447-0767

sao enriquecidos com metadados cl ˜ ´ınicos.

A estruturac¸ao detalhada desses dados ˜ e apresentada na sec¸ ´ ao seguinte, garantindo ˜ que tanto a contextualizac¸ao quanto a sequ ˜ encia temporal do di ˆ alogo sejam preservadas ´ ao longo do *pipeline* de processamento.

4.3. Modelo de Dados

A representac¸ao das sess ˜ oes foi estruturada de forma hier ˜ arquica, visando rastreabilidade ´ e integrac¸ao com o m ˜ odulo de recuperac¸ ´ ao sem ˜ antica do sistema. Cada sess ˆ ao˜ e tratada ´ como uma unidade independente, contendo informac¸oes b ˜ asicas sobre o atendimento e ´ uma lista de transcric¸oes correspondentes aos turnos de fala. A Tabela 2 apresenta a ˜ estrutura geral adotada para esse modelo de dados.

Como mostra a Tabela 2, o modelo foi projetado para preservar a sequencia tem- ˆ poral das falas e manter a associac¸ao entre conte ˜ udo e emissor. Essa estrutura facilita ´ consultas contextuais e analises longitudinais, al ´ em de servir de base para o processo de ´ indexac¸ao sem ˜ antica da arquitetura RAG. Cada transcric¸ ˆ ao˜ e convertida em um vetor de ´ *embedding* e armazenada com seus metadados, permitindo que o sistema recupere tre chos relevantes conforme o contexto da consulta e produza respostas fundamentadas em evidencias cl ˆ ´ınicas verificaveis. ´

Tabela 2. Estrutura de Dados de uma Sessao Cl´ınica ˜

Campo	Descric¸ao˜
id	Identificador unico da sess ´ ao cl ˜ ´ınica.
dataSessao	Data em que o atendimento foi realizado.
idPaciente	Codigo identificador do paciente. ´
idTerapeuta	Codigo identificador do terapeuta respons ´ avel. ´
transcricoes[]	Lista de turnos de fala, representando o dialogo. ´
timestamp	Momento exato do in´ıcio da fala dentro da sessao. ˜
autor	Indica o emissor da fala (paciente ou terapeuta).
mensagem	Texto transcrito automaticamente pelo modelo Whisper.

4.4. Seguranc¸a e LGPD

A arquitetura incorpora medidas de seguranc¸a e conformidade com a Lei Geral de Protec¸ao de Dados (LGPD), considerando o tratamento de informac¸ ˜ oes sens ˜ ´ıveis de saude. Todas as transcric¸ ´ oes e metadados cl ˜ ´ınicos sao protegidos por criptografia ˜ simetrica e assim ´ etrica (RSA), assegurando confidencialidade em repouso e em tr ´ ansito. ˆ

O acesso ao sistema utiliza autenticac¸ao JWT, restringindo o uso ˜ as func¸ `

oes e ˜ dados autorizados. As operac¸oes s ˜ ao registradas em logs seguros e audit ˜ aveis, sem ´ exposic¸ao de conte ˜ udo sens ´ ´ıvel em texto claro. As pol´ıticas de autenticac¸ao, autorizac¸ ˜ ao˜ e seguranc¸a seguem um modelo modular e escalavel, favorecendo manutenc¸ ´ ao e evoluc¸ ˜ ao˜ da protec¸ao de dados. ˜

5. Arquitetura da Soluc¸ao baseada em RAG ˜

A arquitetura da soluc¸ao foi concebida com base nos princ ˜ ´ıpios de *Retrieval-Augmented Generation* (RAG), estruturando um fluxo voltado a an ` alise e consulta de registros ´ cl´ınicos transcritos. O sistema integra diferentes modulos de processamento, permitindo ´ que o terapeuta acesse informac¸oes contextualizadas e rastre ˜ aveis a partir do hist ´ orico de ´ atendimentos.

A adoc¸ao da RAG neste projeto possibilitou o uso controlado de dados sens ˜ ´ıveis, garantindo que as respostas geradas pelo modelo se apoiem em fragmentos documen tais reais, preservando a precisao sem ˜ antica e o contexto cl ˆ ´ınico de origem. A Figura 3 apresenta a visao geral da arquitetura implementada. As subsec¸ ˜ oes seguintes descrevem ˜ o funcionamento das tres etapas principais, indexac¸ ˆ ao, recuperac¸ ˜ ao e enriquecimento, e ˜ gerac¸ao, e sua contribuic¸ ˜ ao para a construc¸ ˜ ao de respostas contextualizadas. ˜

Figura 3. Arquitetura RAG da soluc¸ao proposta. ˜

Fonte: Elaborado pelo autor (2025), adaptado de [Databricks 2025].

5.1. Etapa de Indexac¸ao (˜ *Indexing*)

No contexto do presente projeto, a etapa de indexac¸ao foi respons ˜ avel por preparar e ´ armazenar as transcric¸oes dos atendimentos processadas pelo modelo Whisper. Cada ˜ sessao, ap ˜ os passar pela diarizac¸ ´ ao, ˜ e convertida em fragmentos menores de texto (´ *chunks*) por meio da ferramenta *RecursiveCharacterTextSplitter*, a fim de otimizar a granularidade da busca.

Os fragmentos resultantes sao ent ˜ ao vetorizados utilizando o modelo de ˜ *embed dings*, implementado na biblioteca *SentenceTransformers*. Esses vetores, juntamente com os metadados de origem (identificador da sessao, autor e carimbo temporal), s ˜ ao armaze- ˜ nados no banco vetorial *ChromaDB*. Essa abordagem permite que cada trecho da sessao˜ cl´ınica seja indexado de forma semantica, garantindo rastreabilidade e precis ˆ ao durante o ˜ processo de recuperac¸ao posterior. ˜

5.2. Etapa de Recuperac¸ao e Enriquecimento (˜ *Retrieval & Augmentation*)

Durante a etapa de Recuperac¸ao e Enriquecimento, o sistema utiliza o reposit ˜ orio veto- ´ rial previamente construído para localizar os fragmentos de transcric¸ao mais relevantes ˜ em relac¸ao˜ a consulta do terapeuta. Essa busca ` e executada por meio da func¸ ´ ao˜ *simila rity search* da biblioteca *LangChain*, que aplica o metodo dos ´ *k nearest neighbors* sobre os vetores armazenados no *ChromaDB*.

Os fragmentos mais proximos s ´ ao integrados ˜ a consulta original,

ISSN: 2447-0767

formando o ` *prompt* contextualizado enviado ao modulo de gerac¸ ´ ao de linguagem natural. Esse pro- ˜ cesso garante que o modelo baseie suas respostas em evidencias cl ˆ ´ınicas espec´ıficas do paciente, e nao apenas em seu conhecimento pr ˜ e-treinado. ´

5.3. Etapa de Gerac¸ao (˜ *Generation*)

Na fase final da arquitetura, o modelo de linguagem e respons ´ avel por processar o ´ *prompt* enriquecido e gerar a resposta textual. Nesta implementac¸ao, empregou-se o modelo ˜ *google/gemma-2b*, um LLM de codigo aberto, executado por meio da ´ *Ollama API* de forma ass´ıncrona, utilizando a biblioteca *httpx*. Essa abordagem permite escalabilidade no processamento das requisic¸oes, reduzindo o tempo de resposta e favorecendo a execuc¸ ˜ ao˜ paralela de multiplas consultas. ´

Apos a gerac¸ ´ ao bruta da resposta, o sistema realiza um tratamento adicional para ˜ estruturar a sa´ıda antes da apresentac¸ao ao usu ˜ ario. Esse processo, implementado no ´ *front end*, e respons ´ avel por organizar e exibir de forma clara o conte ´ udo retornado pela API. ´ Nele, ocorre a extrac¸ao e a formatac¸ ˜ ao das refer ˜ encias correspondentes aos fragmentos ˆ recuperados, garantindo que cada trecho citado na resposta esteja devidamente vinculado a sua origem textual. Dessa forma, evita-se a apresentac¸ ` ao de respostas n ˜ ao verific ˜ aveis e ´ assegura-se a rastreabilidade entre o conteudo produzido e o contexto cl ´ ´ınico utilizado.

O comportamento do modelo foi configurado para operar de maneira anal´ıtica e propositiva, de modo que as respostas apresentem nao apenas informac¸ ˜ oes diretas, mas ˜ tambem interpretac¸ ´ oes complementares e sugest ˜ oes de acompanhamento cl ˜ ´ınico, sempre dentro dos limites das evidencias contidas nos fragmentos recuperados. Assim, o sistema ˆ entrega uma resposta final que combina precisao, contextualizac¸ ˜ ao e transpar ˜ encia quanto ˆ as fontes empregadas. `

6. Interface e Interac¸ao˜

A interface do sistema foi concebida para apoiar o trabalho do terapeuta com foco em seguranc¸a e conformidade. A estrutura privilegia organizac¸ao de pacientes e sess ˜ oes, ˜ apresentac¸ao de indicadores derivados do processamento das sess ˜ oes e a interac¸ ˜ ao por ˜ meio de um chat com o assistente generativo.

6.1. Visao Geral da Interface ˜

A aplicac¸ao apresenta tr ˜ esˆ areas principais: ´ (i) painel de pacientes e sessoes; ˜ (ii) painel de metadados e indicadores derivados das sessoes; e ˜ (iii) chat com o assistente generativo. A visualizac¸ao prioriza exibir apenas informac¸ ˜ oes tratadas e autorizadas, conforme as ˜ pol´ıticas de acesso definidas. A Figura 4 apresenta a interface dessas funcionalidades na visao geral do sistema. ˜

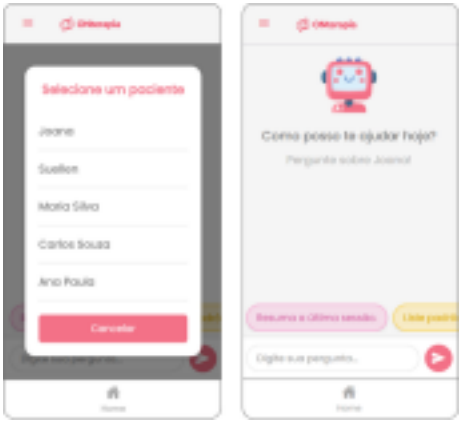


Figura 4. Visao Geral da Interface do Sistema ˜

Fonte: Elaborado pelo autor (2025).

A Figura 4 mostra duas telas principais da aplicac¸ao. ˜ A esquerda, ` e exibido o ´ painel de selec¸ao de pacientes dispon ˜ ´ıveis para consulta. A direita, observa-se o ambiente ` de chat do assistente generativo, apresentado em seu estado inicial, antes do in´ıcio da interac¸ao. Tamb ˜ em s ´ ao apresentados ˜ *chips* de sugestao de consulta, gerados automatica- ˜ mente a partir do conteudo das sess ´ oes. ˜

6.2. Painel de Metadados e Indicadores

O painel de metadados foi desenvolvido para fornecer uma visao sint ˜ etica e n ´ ao sens ˜ ´ıvel das sessoes registradas. A Figura 5 apresenta a interface dessa funcionalidade, que re ˜ une ´ informac¸oes resumidas sobre cada sess ˜ ao, como data, durac¸ ˜ ao e etiquetas tem ˜ aticas. Esse ´ painel integra-se ao ambiente de analise, permitindo a visualizac¸ ´ ao de padr ˜ oes e m ˜ etricas ´ sem expor transcric¸oes literais ou conte ˜ udo cl ´ ´ınico confidencial.

A Figura 5 ilustra o conjunto de indicadores e elementos do painel. Na parte supe rior, sao exibidos os indicadores gerais e o gr ˜ afico que representa a distribuic¸ ´ ao dos temas ˜ das sessoes. Na parte inferior, encontra-se o painel de metadados, que consolida dados de- ˜ rivados dos processos de extrac¸ao e anonimizac¸ ˜ ao, apresentando apenas informac¸ ˜ oes su- ˜ marizadas das sessoes. Essa estrutura possibilita o acompanhamento da evoluc¸ ˜ ao cl ˜ ´ınica sem a necessidade de acesso direto aos conteudos sens ´ ´ıveis das sessoes. ˜

Figura 5. Interface do Painel de Metadados e acompanhamento de sessoes ˜

ISSN: 2447-0767

Fonte: Elaborado pelo autor (2025).

6.3. Chat com o Assistente Generativo

O modulo de chat constitui o n ´ ucleo da aplicac¸ ´ ao, respons ˜ avel por intermediar a interac¸ ´ ao˜ entre o terapeuta e o assistente generativo. A Figura 6 apresenta a interface do chat, na qual o usuario pode formular consultas em linguagem natural sobre as sess ´ oes processadas ˜ e visualizar as respostas geradas pelo modelo com base nos dados tratados.

Figura 6. Interface de Chat com o Assistente Generativo

Fonte: Elaborado pelo autor (2025).

A Figura 6 mostra o ambiente de conversa entre terapeuta e sistema. Cada interac¸ao exibe a mensagem enviada pelo psic ˜ ologo, a resposta do modelo e os ´ *chips* de referencia localizados abaixo das respostas. ˆ

6.4. Acessibilidade e Experiencia do Usu ˆ ario ´

O design segue princ´ıpios de clareza e acessibilidade como contraste adequado, agrupa mento visual por blocos informacionais e suporte a navegac¸ao por teclado. A apresentac¸ ˜ ao˜ foi pensada para facilitar a leitura de indicadores e a execuc¸ao de ac¸ ˜ oes r ˜ apidas, mantendo ´ controle sobre o n´ıvel de detalhe exibido em conformidade com a LGPD.

7. Discussao˜

A adoc¸ao da arquitetura RAG neste sistema mostrou-se adequada ao contexto cl ˜ ´ınico, por possibilitar a combinac¸ao entre gerac¸ ˜ ao de linguagem natural e acesso a informac¸ ˜ oes ˜ previamente registradas nas sessoes terap ˜ euticas. Essa abordagem permite que o modelo ˆ produza respostas contextualizadas a partir de trechos reais dos dialogos, reduzindo a ´ dependencia exclusiva do conhecimento adquirido durante o treinamento. ˆ

Entre os benef´ıcios observados, destaca-se a capacidade do modelo em preservar o contexto semantico das interac¸ ˆ oes cl ˜ ´inicas, o que pode facilitar o acompanhamento longitudinal de pacientes e a revisao de sess ˜ oes de forma mais organizada. A estrutura ˜ modular da arquitetura tambem favorece a manutenc¸ ´ ao e a substituic¸ ˜ ao independente dos ˜ componentes, como o modelo de *embeddings* ou a LLM, sem comprometer o fluxo geral do sistema. Alem disso, a utilizac¸ ´ ao de uma base vetorial local oferece maior controle ˜ sobre os dados e

contribui para a protec¸ao da privacidade em ambientes cl ˜ ´ınicos.

Apesar dos avanc¸os proporcionados, algumas limitac¸oes permanecem. A quali- ˜ dade e consistencia dos dados de origem s ˆ ao fatores cr ˜ ´ıticos em sistemas de IA em saude, ´ pois dados imprecisos ou incompletos comprometem a confiabilidade da inferencia. ˆ [Bernardi et al. 2023]. Nas etapas iniciais de transcric¸ao e diarizac¸ ˜ ao, ˜ e importante con- ´ siderar que os modelos utilizados, como *Whisper* e *pyannote.audio*, operam com taxas de acuracia vari ´ aveis, podendo introduzir pequenas imprecis ´ oes na separac¸ ˜ ao de falan- ˜ tes ou na transcric¸ao textual. Esses desvios, ainda que sutis, podem afetar a indexac¸ ˜ ao˜ e, consequentemente, a relevancia dos fragmentos recuperados. Al ˆ em disso, dados in- ´ completos ou mal segmentados comprometem a contextualizac¸ao, e fragmentos extensos ˜ podem ultrapassar o limite de tokens do modelo, reduzindo a eficiencia na gerac¸ ˆ ao de ˜ respostas. A literatura recente tambem destaca o risco de ´ *retrieval collapse*, situac¸ao˜ em que o sistema tende a recuperar fragmentos semelhantes, diminuindo a diversi dade informacional e a abrangencia do conte ˆ udo apresentado [Gargari and Habibi 2025]. ´ Tais fenomenos reforc¸am a import ˆ ancia de mecanismos de diversificac¸ ˆ ao de busca e ˜ reponderac¸ao sem ˜ antica. ˆ

Outros desafios incluem o custo computacional de manutenc¸ao da base veto- ˜ rial, especialmente quando ha atualizac¸ ´ ao cont ˜ ´ınua dos dados cl´ınicos, e a necessidade de garantir transparencia e auditabilidade em todo o processo. Mesmo com registros ˆ de referencia e logs seguros, ainda ˆ e necess ´ ario garantir que o modelo n ´ ao produza ˜ interpretac¸oes que possam ser entendidas como julgamento cl ˜ ´ınico ou diagnostico. Esses ´ fatores reforc¸am a importancia de governanc¸a sobre o uso da tecnologia e de diretrizes ˆ eticas bem definidas. ´

Do ponto de vista pratico, a adoc¸ ´ ao da RAG em cl ˜ ´ınicas e servic¸os de saude men- ´ tal pode contribuir para otimizar o registro e a recuperac¸ao de informac¸ ˜ oes, auxiliando o ˜ profissional no acompanhamento de pacientes e na analise de evoluc¸ ´ ao terap ˜ eutica. Con- ˆ tudo, sua aplicac¸ao requer infraestrutura adequada, controle de acesso e capacitac¸ ˜ ao dos ˜ usuarios para interpretac¸ ´ ao cr ˜ ´ıtica dos resultados.

Por fim, destaca-se a importancia de alinhar o desenvolvimento e a utilizac¸ ˆ ao de ˜ sistemas baseados em RAG a princ´ıpios de etica, transpar ´ encia e confiabilidade. A utili- ˆ dade dessa arquitetura depende nao apenas da qualidade t ˜ ecnica, mas tamb ´ em da forma ´ como e integrada ao processo cl ´ ´ınico, garantindo que as respostas geradas complementem, e nao substituam, o julgamento humano [Solanki et al. 2023]. ˜

8. Conclusao˜

Este trabalho propos e implementou um assistente generativo baseado na arquitetura ˆ *Retrieval-Augmented Generation* (RAG), voltado a an ` alise de sess ´ oes cl ˜ ´ınicas. A soluc¸ao˜ foi estruturada em tres etapas principais, que operam de forma sequencial para combinar ˆ evidencias textuais com a produc¸ ˆ ao de respostas em linguagem natural. ˜

A aplicac¸ao da RAG mostrou-se adequada a cen ˜ arios em que a rastreabilidade e a ´ fidelidade as fontes s ` ao requisitos fundamentais, permitindo que cada resposta seja susten- ˜ tada por fragmentos espec´ıficos de registros cl´ınicos. O sistema desenvolvido demonstra a viabilidade tecnica de utilizac¸ ´ ao de modelos de linguagem em contextos cl ˜ ´ınicos sob controle de dom´ınio,

ISSN: 2447-0767

assegurando transparencia no processo de infer ˆ encia e limitac¸ ˆ ao no ˜ escopo das respostas.

Como desdobramento futuro, pretende-se realizar a validac¸ao do sistema com te- ˜ rapeutas em ambiente controlado, a fim de avaliar sua aplicabilidade pratica e aprimorar ´ os modulos de compress ´ ao sem ˜ antica e diarizac¸ ˆ ao. Tamb ˜ em est ´ ao previstas melhorias na ˜ arquitetura RAG, com a incorporac¸ao de agentes especializados e integrac¸ ˜ ao a prontu ˜ arios ´ eletronicos, visando ampliar a precis ˆ ao das respostas e reforc¸ar a conformidade ˜ etica e a ´ seguranc¸a das informac¸oes processadas. ˜

9. Referencias ˆ

Referencias ˆ

Abraham, A., Jithesh, A., Doraiswamy, S., Al-Khawaga, N., Mamtani, R., and Cheema, S. (2021). Telemental health use in the covid-19 pandemic: a scoping review and evidence gap mapping. *Frontiers in Psychiatry*, 12:748069.

Adedeji, A., Joshi, S., and Doohan, B. (2024). The sound of healthcare: Improving medical transcription asr accuracy with large language models. *arXiv preprint ar Xiv:2402.07658*.

Adhikary, P. K., Srivastava, A., Kumar, S., Singh, S. M., Manuja, P., Gopinath, J. K., Krishnan, V., Gupta, S. K., Deb, K. S., and Chakraborty, T. (2024). Exploring the efficacy of large language models in summarizing mental health counseling sessions: benchmark study. *JMIR Mental Health*, 11:e57306.

Agency for Healthcare Research and Quality (2023). Physician burnout. Acesso em: 15 out. 2025.

Amugongo, L. M., Mascheroni, P., Brooks, S., Doering, S., and Seidel, J. (2025). Re trieval augmented generation for large language models in healthcare: A systematic review. *PLOS Digital Health*, 4(6):e0000877.

Asgari, E., Kaur, J., Nuredini, G., Balloch, J., Taylor, A. M., Sebire, N., Robinson, R., Peters, C., Sridharan, S., and Pimenta, D. (2024). Impact of electronic health record use on cognitive load and burnout among clinicians: narrative review. *JMIR Medical Informatics*, 12:e55499.

Association, A. P. (2023). Telehealth and telepsychology. *APA*. Available at: https://www.apa.org/practice/telehealth-telepsychology.

Bernardi, F. A., Alves, D., Crepaldi, N., Yamada, D. B., Lima, V. C., and Rijo, R. (2023). Data quality in health research: integrative literature review. *Journal of Medical Inter net Research*, 25:e41446.

Blease, C. and Rodman, A. (2024). Generative artificial intelligence in mental healthcare: an ethical evaluation. *Current Treatment Options in Psychiatry*, 12(1):5.

Budd, J. (2023). Burnout related to electronic health record use in primary care. *Journal of Primary Care & Community Health*, 14.

Bulkes, N. Z., Davis, K., Kay, B., and Riemann, B. C. (2022). Comparing efficacy of te lehealth to in-person mental health care in intensive-treatment-seeking adults. *Journal of Psychiatric Research*, 145:347–352.

Chen, Y. and Esmaeilzadeh, P. (2024). Generative ai in medical practice: in-depth ex ploration of privacy and security challenges. *Journal of Medical Internet Research*,

26:e53008.

Conselho Federal de Psicologia (2022). Censopsi 2022: Resultados da maior pesquisa sobre o exerc´ıcio profissional da psicologia brasileira. Acesso em: 22 out. 2025.

Databricks (2025). Rag (retrieval-augmented generation) on databricks. Acesso em: 19 out. 2025.

Donoso-Guzman, I., Sirka Kacaf ´ ´ırkova, K., Szymanski, M., Jacobs, A., Parra, D., and ´ Verbert, K. (2025). A systematic review of user-centred evaluation of explainable ai in healthcare. *arXiv Preprint*.

Gaffney, A., Woolhandler, S., Cai, C., et al. (2022). Medical documentation burden among us office-based physicians in 2019: A national study. *JAMA Internal Medi cine*, 182(5):564–566.

Gargari, O. K. and Habibi, G. (2025). Enhancing medical ai with retrieval-augmented generation: A mini narrative review. *Digital Health*, 11:20552076251337177.

Greenwood, H., Krzyzaniak, N., Peiris, R., Clark, J., Scott, A. M., Cardona, M., Griffith, R., and Glasziou, P. (2022). Telehealth versus face-to-face psychotherapy for less common mental health conditions: systematic review and meta-analysis of randomized controlled trials. *JMIR Mental Health*, 9(3):e31780.

Hammond, T. E., Crowther, A., and Drummond, S. (2018). A thematic inquiry into the burnout experience of australian solo-practicing clinical psychologists. *Frontiers in Psychology*, 8:1996.

Kaveladze, B. T. (2022). User experience, engagement, and popularity in mental health apps. *JMIR*. Available via PMC.

Kruse, C. S., Mileski, M., Dray, G., Johnson, Z., Shaw, C., and Shirodkar, H. (2022). Phy sician burnout and the electronic health record leading up to and during the first year of covid-19: systematic review. *Journal of medical Internet research*, 24(3):e36200.

Medaramitta, R. (2021). Evaluating the performance of using speaker diarization for speech separation of in-person role-play dialogues.

O'Shaughnessy, D. (2025). Speaker diarization: A review of objectives and methods. *Applied Sciences*, 15(4):2002.

Sablone, S., Giordano, C., Messina, I., et al. (2024). Telepsychology revolution in the mental health care delivery: a global overview of emerging clinical and legal issues. *Frontiers in Psychology*, 15:1326667.

Saeidnia, H. R., Hashemi Fotami, S. G., Lund, B., and Ghiasi, N. (2024). Ethical consi derations in artificial intelligence interventions for mental health and well-being: En suring responsible implementation and impact. *Social Sciences*, 13(7):381.

Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., and Chadha, A. (2024). A syste matic survey of prompt engineering in large language models: Techniques and appli cations. *arXiv preprint arXiv:2402.07927*.

Solanki, P., Grundy, J., and Hussain, W. (2023). Operationalising ethics in artificial intel ligence for healthcare: a framework for ai developers. *AI and Ethics*, 3(1):223–240.

Tajirian, T., Stergiopoulos, V., Strudwick, G., Sequeira, L., Sanches, M., Kemp, J.,

ISSN: 2447-0767

Rama moorthi, K., Zhang, T., and Jankowicz, D. (2020). The influence of electronic health record use on physician burnout: cross-sectional survey. *Journal of medical Internet research*, 22(7):e19274.

Tavory, T. (2024). Regulating ai in mental health: ethics of care perspective. *JMIR Mental Health*, 11(1):e58493.

Torous, J. and Blease, C. (2024). Generative artificial intelligence in mental health care: potential benefits and current challenges. *World Psychiatry*, 23(1):1.

Vazquez, Y. (2025). *Perceived Impacts of Electronic Health Record (EHR) Documenta tion Burden on Clinicians' Job Dissatisfaction*. PhD thesis, Capella University.

Vial, S., Boudhraa, S., and Dumont, M. (2022). Human-centered design approaches in ˆ digital mental health interventions: exploratory mapping review. *JMIR Mental health*, 9(6):e35591.

Wang, D. and Zhang, S. (2024). Large language models in medical and healthcare fields: applications, advances, and challenges. *Artificial intelligence review*, 57(11):299.

Yang, R., Ning, Y., Keppo, E., Liu, M., Hong, C., Bitterman, D. S., Ong, J. C. L., Ting, D. S. W., and Liu, N. (2024). Retrieval-augmented generation for generative artificial intelligence in medicine. *arXiv preprint arXiv:2406.12449*.

Zhan, J., Moore, D., Lu, Y., and Abbasi, H. (2025). Inspired spine smart universal re source identifier (suri): An adaptive ai framework for transforming multilingual speech into structured medical reports. *Cureus*, 17(3).

Zolnoori, M., Vergez, S., Xu, Z., Esmaeili, E., Zolnour, A., Anne Briggs, K., Scroggins, J. K., Hosseini Ebrahimabad, S. F., Noble, J. M., Topaz, M., et al. (2024). Decoding disparities: evaluating automatic speech recognition system performance in transcri bing black and white patient verbal communication with nurses in home healthcare. *JAMIA open*, 7(4):ooae130.

Ambientes Interativos para Aprendizagem de Maquina: Potencialidades e Limitac¸oes de Jupyter Lab e Google Colab

Martony Demes da Silva[1]

[1]Universidade Federal do Piau´ı (UFPI)
Centro de Educac¸ao Aberta e a Distancia (CEAD) ˜
Teresina – Piau´ı – Brasil

martony.silva@ufpi.edu.br

Abstract. The advancement of Artificial Intelligence (AI), particularly Machine Learning, has reshaped education by demanding environments that foster prac tice and reproducibility. However, challenges such as infrastructure limitations, local configuration difficulties, and restrictions of cloud-based platforms persist. In this context, Jupyter Lab and Google Colab emerge as central alternatives, each with specific strengths and limitations that directly impact the learning process. This article presents a comparative analysis of these tools, highligh ting pedagogical implications, opportunities for democratization, and pathways for innovation in Machine Learning education.

Resumo. O avanc¸o da Inteligencia Artificial (IA), especialmente da Aprendiza- ˆ gem de Maquina, tem transformado o ensino ao exigir ambientes que favorec¸am ´ a pratica e a reprodutibilidade. Contudo, persistem desafios como limitac¸ ´ oes ˜ de infraestrutura, dificuldades de configurac¸ao local e restric¸ ˜ oes de platafor- ˜ mas em nuvem. Nesse contexto, Jupyter Lab e Google Colab emergem como alternativas centrais, cada uma com potencialidades e limitac¸oes que afetam ˜ diretamente o processo formativo. Este artigo apresenta uma analise compa- ´ rativa dessas ferramentas, destacando implicac¸oes pedag ˜ ogicas, possibilidades ´ de democratizac¸ao e caminhos para inovac¸ ˜ ao no ensino de Aprendizagem de ˜ Maquina. ´

1. Introduc¸ao˜

O avanc¸o da Inteligencia Artificial (IA) e, em especial, da Aprendizagem de M ˆ aquina, tem ´ transformado o ensino e a pratica de modelos computacionais inteligentes. T ´ ecnicas como ´ redes neurais convolucionais (CNNs) e recorrentes (RNNs) exigem nao apenas conheci- ˜ mentos conceituais solidos, mas tamb ´ em ambientes computacionais capazes de integrar ´ codigo, documentac¸ ´ ao, visualizac¸ ˜ ao e experimentac¸ ˜ ao pr ˜ atica [Gent et al. 2020]. ´

ISSN: 2447-0767

Apesar do crescimento do uso de notebooks interativos no ensino, persistem desa fios significativos. Estudantes e professores enfrentam limitac¸oes de hardware, restric¸ ˜ oes ˜ de tempo de execuc¸ao em ambientes em nuvem e dificuldades na configurac¸ ˜ ao local de ˜ ferramentas como Jupyter Lab. Em contrapartida, ambientes como Google Colab demo cratizam o acesso a GPUs e TPUs gratuitas, mas dependem de conectividade estavel e ´ tem restric¸ ˆ oes de sess ˜ ao, o que pode prejudicar experimentos longos e cont ˜ ´ınuos [Matias 2024, Seebut et al. 2024].

Essas limitac¸oes evidenciam a necessidade de an ˜ alises comparativas que orientem ´ praticas pedag ´ ogicas e escolhas de ferramentas, considerando o perfil do curso, a infraes- ´ trutura dispon´ıvel e os objetivos de aprendizagem. Estudos recentes destacam tanto o po tencial pedagogico de Colab em contextos de ensino remoto e aplicac¸ ´ oes em IA [Brocker ˜ et al. 2022,Pinto et al. 2024], quanto as oportunidades de reprodutibilidade, customizac¸ao˜ e suporte a Learning Analytics oferecidas pelo Jupyter Lab [Valle Torre et al. 2025, Cai et al. 2025].

Nesse cenario, este artigo prop ´ oe uma an ˜ alise comparativa de Jupyter Lab e Go- ´ ogle Colab no contexto educacional, com foco no ensino de Aprendizagem de Maquina. ´ Busca-se identificar potencialidades, limitac¸oes e implicac¸ ˜ oes pedag ˜ ogicas, fornecendo ´ recomendac¸oes pr ˜ aticas para a integrac¸ ´ ao desses ambientes em disciplinas de IA. Este ˜ estudo contribui para orientar decisoes de docentes e pesquisadores, al ˜ em de sugerir ca- ´ minhos para futuras pesquisas voltadas a democratizac¸ ` ao do ensino de Aprendizagem de ˜ Maquina e ´ a adoc¸ ` ao de metodologias h ˜ ´ıbridas que combinem acessibilidade, reprodutibi lidade e inovac¸ao pedag ˜ ogica [Amoudi and Tbaishat 2023, Temel et al. 2025]. ´

2. Trabalhos Relacionados

O uso de notebooks interativos como Jupyter Lab e Google Colab no ensino de programac¸ao, ci ˜ encia de dados e disciplinas correlatas tem recebido crescente atenc¸ ˆ ao em ˜ pesquisas nacionais e internacionais. Tais estudos discutem aspectos como usabilidade, potencial pedagogico, reprodutibilidade e limitac¸ ´ oes t ˜ ecnicas. ´

No contexto nacional, destacam-se investigac¸oes que exploram o Colab em dife- ˜ rentes cenarios educacionais. Silva [Silva 2021] avaliou a utilidade percebida e a facili- ´ dade de uso da ferramenta em cursos de Ciencia de Dados. Ferreira et al. [Ferreira and ˆ Nacif 2023] apresentaram experiencias de ensino de arquiteturas de computadores e es- ˆ calonamento em GPUs utilizando Python no Colab, enquanto outro trabalho dos mesmos autores [Ferreira et al. 2023] relatou a adoc¸ao da plataforma em disciplinas introdut ˜ orias ´ de computac¸ao. Ainda, Hayashi e

Prado [Hayashi et al. 2021] propuseram um laborat ˜ orio ´ virtual com dados reais, mostrando como os notebooks podem ampliar a experimentac¸ao˜ remota, e Baptista et al. [Baptista 2021] demonstraram a aplicabilidade do Colab em areas ´ nao diretamente relacionadas ˜ a computac¸ ` ao, como F ˜ ´ısico-Qu´ımica.

No cenario internacional, observa-se uma ampliac¸ ´ ao das discuss ˜ oes sobre ˜ integrac¸ao entre ferramentas, metodologias de ensino h ˜ ´ıbridas e reprodutibilidade. Tru ong et al. [Brocker et al. 2022] exploraram abordagens de baixo custo para ensino de matematica aplicada em ambientes baseados em notebooks. Torre et al. [Valle Torre ´ et al. 2025] destacaram o potencial do Jupyter Lab integrado a Learning Analytics para suporte ao acompanhamento e personalizac¸ao do aprendizado. Al ˜ em disso, Lle- ´ rena et al. [Amoudi and Tbaishat 2023] e Yavuz et al. [Temel et al. 2025] abordaram a importancia de notebooks interativos na democratizac¸ ˆ ao do ensino de IA, incluindo ˜ comparac¸oes entre diferentes plataformas educacionais e sua adoc¸ ˜ ao em larga escala. ˜

A Tabela 1 sintetiza os principais estudos encontrados, destacando os ambientes utilizados, areas de aplicac¸ ´ ao e contribuic¸ ˜ oes centrais. ˜

Tabela 1. S´ıntese de trabalhos relacionados ao uso de notebooks no ensino

Autores	Ambiente	Area de ´Aplicac¸ao˜	Contribuic¸oes ˜
Silva (2021) [Silva 2021]	Google Colab	Ciencia de Dados ˆ	Avaliac¸ao de usabilidade e ˜ percepc¸ao de utilidade em ˜ cursos introdutorios. ´
Ferreira et al. (2023) [Ferreira and Nacif 2023]	Google Colab	Arquitetura de Computadores	Ensino de pipelining e escalo namento em GPUs.
Ferreira et al. (2023) [Ferreira et al. 2023]	Google Colab	Computac¸ao Ge- ˜ ral	Estrategias did ´ aticas para dis- ´ ciplinas de computac¸ao. ˜
Hayashi & Prado (2021) [Hayashi et al. 2021]	Google Colab	Laboratorios Vir- ´ tuais	Uso de dados reais para praticas remotas. ´
Baptista et al. (2021) [Baptista 2021]	Google Colab	Ciencias Exatas ˆ	Ensino de F´ısico-Qu´ımica com notebooks.

ISSN: 2447-0767

Truong et al. (2024) [Brocker et al. 2022]	Notebooks Inte rativos	Matematica Apli- ´ cada	Estrategias de baixo custo ´ para ensino remoto.
Torre et al. (2025) [Valle Torre et al. 2025]	Jupyter Lab + LA	Ciencia de Dados ˆ / IA	Integrac¸ao com Le- ˜ arning Analytics para personalizac¸ao. ˜
Llerena et al. (2024) [Amoudi and Tbaishat 2023]	Colab e outros	IA / Educac¸ao˜	Democratizac¸ao e ˜ comparac¸oes entre plata- ˜ formas.
Yavuz et al. (2025) [Temel et al. 2025]	Jupyter/Colab	Ensino de IA	Avaliac¸ao em larga escala de ˜ notebooks interativos.

Esses trabalhos demonstram que, embora o Google Colab se destaque pela aces sibilidade e facilidade de uso, o Jupyter Lab apresenta vantagens em reprodutibilidade, customizac¸ao e integrac¸ ˜ ao com sistemas de an ˜ alise de aprendizagem. A literatura sugere ´ que a escolha do ambiente deve considerar nao apenas a infraestrutura dispon ˜ ´ıvel, mas tambem os objetivos pedag ´ ogicos e a escalabilidade das pr ´ aticas de ensino. ´

3. Metodologia

A presente pesquisa adota como referencial metodologico a abordagem de estudo de caso ´ aliada a an ` alise comparativa, sustentada pela revis ´ ao bibliogr ˜ afica e pela coleta de da- ´ dos emp´ıricos. Essa estrategia metodol ´ ogica, conforme orientac¸ ´ oes de Yin [Yin 2018] ˜ e Gil [Gil 2019], possibilita compreender em profundidade fenomenos educacionais, ao ˆ integrar evidencias documentais, aplicac¸ ˆ ao pr ˜ atica e percepc¸ ´ oes de atores diretamente en- ˜ volvidos. Dessa forma, busca-se alinhar a investigac¸ao te ˜ orica com a pr ´ atica pedag ´ ogica, ´ garantindo maior consistencia na interpretac¸ ˆ ao dos resultados. ˜

3.1. Pesquisa bibliografica ´

A revisao da literatura concentrou-se em artigos e trabalhos que discutem o uso de ambi- ˜ entes interativos no ensino de Inteligencia Artificial e Aprendizagem de M ˆ aquina. Estudos ´ como o de Truong [Brocker et al. 2022] evidenciam o potencial do Google Colab para democratizar o acesso a recursos computacionais em contextos educacionais, enquanto Torre [Valle Torre et al. 2025] destaca as oportunidades de personalizac¸ao e integrac¸ ˜ ao˜ de Learning Analytics proporcionadas pelo Jupyter Lab. Esses achados reforc¸am a re levancia de se analisar comparativamente ambos os ambientes, n ˆ ao apenas do ponto de ˜ vista tecnico, mas sobretudo pedag ´ ogico, no apoio ao ensino e ´

a aprendizagem. `

3.2. Estudo de caso

O estudo aplicado foi realizado em uma disciplina de Aprendizagem de Maquina em n ´ ´ıvel de graduac¸ao. Os estudantes foram divididos em dois grupos: um utilizando o Jupyter ˜ Lab em ambiente local e outro utilizando o Google Colab. Ambos os grupos receberam as mesmas atividades praticas, consistindo no treinamento de modelos de redes neurais ´ convolucionais (CNNs) para classificac¸ao de imagens em um dataset de dom ˜ ´ınio educa cional. O objetivo foi observar como as caracter´ısticas de cada ferramenta influenciam aspectos como fluidez do trabalho, dificuldades tecnicas enfrentadas, tempo de execuc¸ ´ ao˜ e qualidade dos resultados obtidos.

3.3. Coleta de percepc¸oes ˜

Para complementar a analise, foram aplicados question ´ arios estruturados e entrevistas se- ´ miestruturadas com alunos e professores participantes. Os instrumentos buscaram captar percepc¸oes relacionadas a: ˜

- Engajamento – motivac¸ao e interesse dos estudantes durante o uso das ferramen- ˜ tas;
- Usabilidade – clareza da interface, facilidade de configurac¸ao e execuc¸ ˜ ao das ˜ atividades;
- Impacto na aprendizagem – percepc¸ao de ganho de autonomia, aprofundamento ˜ conceitual e aplicabilidade pratica. ´

Essa coleta possibilitou a triangulac¸ao entre dados quantitativos, provenientes dos ˜ questionarios, e qualitativos, obtidos por meio das entrevistas semiestruturadas, fortale- ´ cendo a robustez das conclusoes. A Tabela 2 apresenta os resultados por turma, conside- ˜ rando a evasao de 20%, o n ˜ umero de participantes restantes e os percentuais de aprovac¸ ´ ao˜ para os criterios de engajamento, usabilidade e impacto na aprendizagem, acompanhados ´ de comentarios sintetizados sobre a percepc¸ ´ ao geral dos alunos. ˜

Tabela 2. Resultados por turma considerando evasao de 20%. ˜

Turma	Alunos inici ais	Evasao (20%) ˜	Participante s	Engajame nto (%)	Usabilid ade (%)	Impacto Aprendiza gem (%)	Comentarios ´
T1	15	3	12	78	70	85	Resultados refletempercepc¸ao geral; Colab ˜ motivou mais, Jupyter Lab aprofundou.
T2	12	2	10	78	70	85	Observac¸oes semelhantes; ˜ engajamento inicial maior no Colab.
T3	18	4	14	78	70	85	Estrategia h ´ ´ıbrida eficaz; alu nos evolu´ıram do Colab para Jupyter Lab.

ISSN: 2447-0767

Observa-se, a partir da Tabela 2, que a evasao foi relativamente homog ˜ enea entre ˆ as turmas e que os percentuais de aprovac¸ao se mant ˜ em consistentes, refletindo padr ˆ oes si- ˜ milares de engajamento e percepc¸ao de impacto pedag ˜ ogico. Para uma vis ´ ao consolidada ˜ do desempenho nos ambientes Jupyter Lab e Google Colab, a Tabela 3 reune os resulta- ´ dos gerais dos questionarios, mostrando que o Google Colab apresentou maior aceitac¸ ´ ao˜ inicial em termos de engajamento e usabilidade, enquanto o Jupyter Lab se destacou no aprofundamento tecnico e na promoc¸ ´ ao de autonomia dos estudantes em projetos mais ˜ complexos.

Tabela 3. Resultados reunidos dos questionarios sobre engajamento, usabili- ´ dade e impacto na aprendizagem.

Criterio ´	Jupyter Lab	Google Colab	Comentarios gerais ´
Engajamento	78%	92%	Alunos mais motivados ini cialmente no Colab; Jupyter Lab melhor em projetos com plexos.
Usabilidade	70%	95%	Colab e mais intuitivo e ´ rapido; Jupyter Lab exige ´ configurac¸ao pr ˜ evia. ´
Impacto na aprendiza gem	85	80	Jupyter Lab favorece apro fundamento tecnico; Colab ´ facilita a pratica r ´ apida e ´ compreensao inicial. ˜

Dessa forma, a analise conjunta das duas tabelas evidencia que a escolha do am- ´ biente de ensino deve considerar tanto o perfil da turma quanto a complexidade das ati vidades, sugerindo que uma estrategia h ´ ´ıbrida, combinando Colab e Jupyter Lab, pode maximizar engajamento, aprendizagem e autonomia.

Alem dos dados quantitativos, as entrevistas permitiram captar percepc¸ ´ oes mais ˜ detalhadas. Entre os alunos, destacaram-se comentarios como: ´

• *”No Colab, consegui comec¸ar a treinar meu modelo de classificac¸ao imediata- ˜ mente, sem me preocupar com instalac¸ao de bibliotecas.”* ˜ – Aluno A • *“O Jupyter Lab me permitiu instalar pacotes espec´ıficos e personalizar os parametros do meu projeto de rede neural, o que foi essencial para meu aprendi- ˆ zado avanc¸ado.”* – Aluno B
• *“Gostei de poder compartilhar meus notebooks no Colab*

com meus colegas de forma simples, facilitando a colaborac¸ao em grupo.” ˜ – Aluno C

Do ponto de vista dos professores, os depoimentos reforc¸aram a complementari dade dos ambientes:

- *”O Colab e excelente para atividades introdut ´ orias; os alunos rapidamente se ´ engajam e compreendem conceitos basicos de Aprendizagem de M ´ aquina” ´* . – Professor X
- *”Para projetos mais complexos, recomendo o Jupyter Lab, pois ele permite maior controle e reprodutibilidade das experiencias”. ˆ* – Professor Y
- *“Uma estrategia h ´ ´ıbrida, iniciando com Colab e depois migrando para Jupyter Lab, tem se mostrado eficaz pedagogicamente.”* – Professor Z

A combinac¸ao dos dados quantitativos e qualitativos evidencia que **a escolha do ˜ ambiente deve considerar o perfil dos alunos, a complexidade das atividades e os objetivos pedagogicos**. Em particular, a estrat ´ egia h ´ ´ıbrida mostrou-se promissora, permitindo que os estudantes iniciem com maior acessibilidade e engajamento no Colab e evoluam para o aprofundamento tecnico e autonomia proporcionados pelo Jupyter Lab. ´

3.4. Criterios de comparac¸ ´ ao˜

A comparac¸ao entre Jupyter Lab e Google Colab foi organizada a partir de crit ˜ erios ´ tecnicos e pedag ´ ogicos, sintetizados no Quadro 4. ´

Tabela 4. Criterios de comparac¸ ´ ao entre Jupyter Lab e Google Colab. Fonte: ˜ Autor.

Criterio ´	Jupyter Lab	Google Colab
Facilidade de acesso	Requer instalac¸ao local; pode ˜ demandar conhecimento tecnico. ´	Acesso direto via navegador; nao requer instalac¸ ˜ ao. ˜
Desempenho computa cional	Depende do hardware dis pon´ıvel localmente; limitado em maquinas de baixo de- ´ sempenho.	Oferece GPUs/TPUs gratui tas; restric¸oes de tempo de ˜ execuc¸ao. ˜
Colaborac¸ao e compar- ˜ tilhamento	Integrac¸ao com GitHub; ˜ exige configurac¸ao. ˜	Integrac¸ao nativa com Goo- ˜ gle Drive; compartilhamento simplificado.
Customizac¸ao˜	Flex´ıvel para instalac¸ao de ˜ pacotes adicionais e fra	Suporte a frameworks popu lares, mas com restric¸oes para ˜

ISSN: 2447-0767

	meworks.	instalac¸oes avanc¸adas. ˜
Acessibilidade e in clusao digital ˜	Limitada em contextos de baixo acesso a infraestrutura.	Favorece inclusao pelo acesso ˜ em nuvem, desde que haja in ternet estavel. ´
Impacto pedagogico ´	Estimula maior autonomia e reprodutibilidade dos experi mentos.	Facilita engajamento inicial e reduz barreiras tecnicas. ´

4. Resultados

A analise dos dados obtidos por meio de question ´ arios e entrevistas revelou diferenc¸as sig- ´ nificativas no uso de Jupyter Lab e Google Colab no ensino de Aprendizagem de Maquina, ´ tanto em termos de experiencia t ˆ ecnica quanto pedag ´ ogica. ´

Os alunos destacaram que o Google Colab proporcionou maior engajamento inicial, especialmente para atividades introdutorias. Muitos relataram que n ´ ao precisa- ˜ ram se preocupar com instalac¸ao de softwares ou configurac¸ ˜ ao de bibliotecas, podendo ˜ focar imediatamente na experimentac¸ao pr ˜ atica. Por exemplo, durante o treinamento de ´ modelos simples de classificac¸ao, 85% dos estudantes indicaram que conseguiram iniciar ˜
as atividades sem dificuldades tecnicas, o que contribuiu para motivac¸ ´ ao e confianc¸a na ˜ disciplina.

Em contraste, o Jupyter Lab foi valorizado em projetos mais complexos e de longo prazo, onde a personalizac¸ao do ambiente e a liberdade para instalar pacotes es- ˜ pec´ıficos foram determinantes. Alguns alunos mencionaram que ao desenvolver redes neurais convolucionais mais profundas, conseguiram ajustar parametros avanc¸ados e ex- ˆ plorar ferramentas de analise de performance que n ´ ao estavam dispon ˜ ´ıveis no Colab.

As entrevistas com professores reforc¸aram essas percepc¸oes. Eles observaram ˜ que o Colab facilita o acompanhamento remoto e a colaborac¸ao˜ , sendo ideal para tarefas de grupo e atividades rapidas de laborat ´ orio. J ´ a o **Jupyter Lab permitiu explorar ´ conceitos avanc¸ados com maior controle do ambiente**, favorecendo a reprodutibilidade dos experimentos e o desenvolvimento de autonomia dos alunos.

A analise do impacto na aprendizagem indicou que a ´ estrategia h ´ ´ıbrida mostrou se pedagogica e eficaz: os estudantes iniciam com Colab para ganhar familiaridade e ´ motivac¸ao, e depois migram para Jupyter Lab quando precisam de maior complexidade ˜ e personalizac¸ao. Essa transic¸ ˜ ao gradual contribuiu para o aprofundamento conceitual ˜ e a aplicac¸ao pr ˜

atica de t ´ ecnicas de Aprendizagem de M ´ aquina, al ´ em de permitir que ´ professores ajustassem a metodologia conforme o perfil da turma.

Em termos de usabilidade, ambos os ambientes receberam avaliac¸oes positivas, ˜ porem com nuances: o Colab destacou-se pela simplicidade e acessibilidade, enquanto o ´ Jupyter Lab foi apontado como mais robusto e flex´ıvel para experimentos avanc¸ados. Esse contraste sugere que a escolha da ferramenta deve considerar a experiencia pr ˆ evia ´ dos alunos, a complexidade das atividades e os objetivos pedagogicos ´ , reforc¸ando a relevancia de uma abordagem combinada. ˆ

5. Conclusao˜

O estudo comparativo evidenciou que Jupyter Lab e Google Colab possuem carac ter´ısticas complementares que podem ser estrategicamente exploradas no ensino de Aprendizagem de Maquina. A an ´ alise das turmas, considerando evas ´ ao, engajamento, ˜ usabilidade e impacto na aprendizagem, indicou que o Google Colab favorece a inclusao˜ digital e o engajamento inicial, especialmente em contextos com limitac¸oes de hardware, ˜ enquanto o Jupyter Lab proporciona maior autonomia, reprodutibilidade e aprofunda mento tecnico em projetos mais complexos. ´

Os dados coletados por meio de questionarios e entrevistas reforc¸am que os estu- ´ dantes valorizam a facilidade de uso e o acesso imediato do Colab, mas reconhecem o po tencial do Jupyter Lab para atividades avanc¸adas e customizac¸ao do ambiente. Essa com- ˜ plementaridade sugere que uma abordagem h´ıbrida — iniciando com Colab para praticas ´ introdutorias e avanc¸ando para Jupyter Lab em projetos mais elaborados — maximiza ´ tanto o aprendizado quanto a motivac¸ao dos alunos. ˜

Em s´ıntese, os resultados indicam que a escolha do ambiente deve considerar o perfil da turma, a complexidade das atividades e a infraestrutura dispon´ıvel. A pesquisa contribui para orientar docentes na selec¸ao e integrac¸ ˜ ao de ferramentas de ensino, ao ˜ mesmo tempo em que abre caminhos para estudos futuros sobre estrategias pedag ´ ogicas ´ h´ıbridas que equilibrem acessibilidade, desempenho e profundidade conceitual.

Referencias ˆ

Amoudi, G. and Tbaishat, D. (2023). Interactive notebooks for achieving learning outco mes in a graduate course: a pedagogical approach. *Education and Information Tech nologies*, 28(12):1523–1540.

Baptista, L. (2021). Usando python e o google colab para ensinar f´ısico-qu´ımica. In *Anais do IV Seminario de Boas Pr ´ aticas de Ensino e Aprendizagem – SBPEA, EEL-USP ´* .

Brocker, A., Judel, S., and Schroeder, U. (2022). Integration of gamification and learning analytics in jupyter. *Zeitschrift fur*

ISSN: 2447-0767

E-Learning (eleed) ¨ . Projeto conceitual: aplicar gamificac¸ao e analytics em Jupyter, especialmente para iniciantes em programac¸ ˜ ao. ˜

Cai, Z., Davis, R., Marietan, R., Tormey, R., and Dillenbourg, P. (2025). Jupyter analy- ´ tics: A toolkit for collecting, analyzing, and visualizing distributed student activity in jupyter notebooks. In *SIGCSE Technical Symposium Series (SIGCSE TS) 2025*. Ferra menta para coleta, analise e visualizac¸ ´ ao de dados de interac¸ ˜ ao em Jupyter em cursos ˜ STEM.

Ferreira, R., Canesche, M., and Penha, J. (2023). Google colab para ensino de computac¸ao. In ˜ *Anais Estendidos do Simposio Brasileiro de Educac¸ ´ ao em Computac¸ ˜ ao˜ (EDUCOMP)*.

Ferreira, R. and Nacif, J. A. M. (2023). Teaching software pipelining and scheduling on gpus with python in google colab. *International Journal of Computer Architecture Education*, 12(2):20–29.

Gent, I. P., Ferreira, J., et al. (2020). Jupyter for teaching data science. In *Proceedings of the 2020 ACM Technical Symposium on Computer Science Education (SIGCSE)*, pages XX–XX, United States. ACM. Cloud-based tools for teaching, removing local installation barriers; comparing environments.

Gil, A. C. (2019). *Metodos e T ´ ecnicas de Pesquisa Social ´* . Atlas, Sao Paulo, 7 edition. 7ª ˜ edic¸ao. ˜

Hayashi, V. T., Martins, D. O., Arakaki, R., Teixeira, J. C., Angelico, B., and Hayashi, ´ F. H. (2021). Laboratorio virtual com google colab para o ensino de engenharia. In ´ *Anais do Congresso Brasileiro de Educac¸ao em Engenharia (COBENGE) ˜* .

Matias, A. d. S. (2024). O uso do google colab na disciplina de calculo num ´ erico: ´ uma analise das suas potencialidades para o ensino e aprendizagem de matem ´ atica. ´ Trabalho de Conclusao de Curso (Graduac¸ ˜ ao) – Universidade Estadual do Maranh ˜ ao˜ (UEMA). Acesso: 2025-09-18.

Pinto, J. S., Costa, E. A. T., Almeida, E. P. B. d., and Soares, A. d. S. (2024). Uso do software jupyter notebook como ferramenta de ensino e aprendizagem de matematica. ´ In *Anais da SNCT 2024, IFBA – Campus Vitoria da Conquista ´* .

Seebut, S., Wongsason, P., and Kim, D. (2024). Combining gpt and colab as learning tools for students to explore the numerical solutions of difference equations. *Eurasia Journal of Mathematics, Science and Technology Education*, 20(1):em2377.

Silva, M. D. d. (2021). Aplicac¸ao da ferramenta google colaboratory no ensino de ci ˜ encias ˆ de dados. *Anais do Simposio Brasileiro de Sistemas Colaborativos (SBSC) ´* .

Temel, G. Y., Barenthien, J., and Padubrin, T. (2025). Using jupyter notebooks as di gital assessment tools: An empirical examination of student teachers' attitudes and skills towards digital assessment. *Education and Information Technologies*, 30:18621– 18650.

Valle Torre, M., van der Velden, T., Specht, M., and Oertel, C. (2025). Jelai: Integra ting ai and learning analytics in jupyter notebooks. *arXiv preprint arXiv:2505.17593*. Plataforma experimental integrando LA + tutor IA em Jupyter; arquitetura modular e casos de uso de interac¸ao estudantil. ˜

Yin, R. K. (2018). *Case Study Research and Applications: Design and Methods*. SAGE Publications, Los Angeles, 6 edition. Sixth edition.

ISSN: 2447-0767

DaeLink: Job opportunity for people with special needs

Alex E. dos Santos, Danilo S. Soares, Endrigo G. B. de Oliveira, Andreza M. de S. Rocha, Jeferson R. de Lima.

Análise de Desenvolvimento de Sistemas - Fatec da Zona Leste

Alex.santos129@fatec.sp.gov.br, Danilo.soares14@fatec.sp.gov.br, Endrigo.oliveira@fatec.sp.gov.br, andreza.rocha@fatec.sp.gov.br, jeferson.lima10@fatec.sp.gov.br

Abstract. This work addresses the inclusion of people with disabilities (PWDs) in the labor market through a digital platform that promotes connectivity and professional integration. Despite existing quota laws, companies often do not meet them due to prejudice and low demand. The project studies the development of a platform to ease job inclusion and strengthen opportunities for PWDs. Theoretical support includes analyzing the needs of both PWDs and companies. Results show that a prototype system forming a website and an app can help companies fill vacancies and find qualified candidates. Thus, the project promotes equal opportunities and effective inclusion of PWDs in the workforce.

1. Introduction

Integrating individuals with special needs (PWN) into the labor market through digital connectivity platforms is a significant and pressing concern. Despite many social initiatives aimed at integrating PWNs into society, the number of PWNs employed in companies stays disproportionately low, as shown by the insufficient fulfillment of established quotas. This persistent exclusion of PWNs from the labor market underscores the continuous need for more robust measures to ease their inclusion. (INTERNATIONAL DISABILITY ALLIANCE, 2022). Therefore, this study aims to develop a system that promotes connectivity more efficiently through a website and a mobile application to enhance the integration between companies and their PWNs. In this context, the labor market participation and formalization rates of people with special needs aged 14 and over are significantly lower than those of people without special needs. The labor market participation rate of people with special needs is 23.8%, while the formalization rate is 34.3%. The rates for people without special needs are 66.3% and 50.9%, respectively (IBGE, 2022). One of the primary constraints impeding the inclusion of PWNs in the labor market is prejudice. Many companies remain reluctant to hire individuals with special needs, often due to a lack of awareness about these professionals' skills and abilities. (CNN, 2021).

It is, therefore, imperative to find solutions that help the integration between companies and PWN, thereby increasing opportunities and hiring these professionals through modern technology to promote a more inclusive society. Considering these considerations, it becomes pertinent to inquire why the quotas for PWNs in the labor market remain unfulfilled and prove how a digital platform for professionals can ease the integration of these individuals into companies. The hypothesis is that using a digital system specifically designed to connect companies and people with special needs can increase the filling rate in the labor market, easing the recruitment process, and overcoming current barriers, such as prejudice. This study aims to develop a digital platform that enables the integration of people with special needs into the labor market, promoting a more inclusive work environment, and increasing the number of quota positions.

The first stage involved a bibliographic review of the inclusion of PWNs in the labor

market, emphasizing inclusion studies and tools to find how digital platforms can ease the hiring of PWNs. Based on the findings of this review, the quantitative method and case study were used to gather all data and analyze specific problems, employing both inductive and deductive methods, as set forth by Lakatos and Marconi (2017), along with PEREIRA et al. (2018) and GIL (2002). The research will address several seminal authors in the field, including CNN (2022), which analyzes the importance of using technology to enhance inclusion. The authors will be referenced throughout the article to provide theoretical support for developing the platform and its potential solutions for the inclusion challenge. The development was divided into three sections: the web, a mobile application, and the recommendation system. React was selected for web development due to its straightforward part creation and processing capabilities (SCHMITZ; GEORGII, 2015). React Native was employed for mobile app development due to its native and cross-platform compatibility with Android and iOS (ESCUDELARIO; PINHO, 2020). The recommendation system used Python for its extensive toolset, including Scikit-Learn for machine learning (MENEZES, 2014).

2. Theoretical Foundation

This chapter abstracts all the stages of the theoretical foundation for understanding this article and presents concepts and technologies. It aims to prove all the theoretical underpinnings of the DAELink platform.

2.1. Challenges of Inclusion in the labor market for people with special needs

In Brazil, around 18.6 million people in Brazil between two and more have some type of disability, and inclusion is a challenge that persists in Brazil due to the lack of accessibility and adequate support (G1, 2023).

Highlighted by a comparatively inadequate perspective about other candidates, along with challenges rooted in internalized societal prejudices that hinder adaptive processes in new circumstances, both in general and corporate environments (RIBEIRO; DELLATORRE, 2021).

These special and structural barriers are also reflected in labor market statistics. According to data from the Brazilian Institute of Geography and Statistics (IBGE, 2022), people with special needs face greater difficulties entering the workforce. Their participation rate is only 28.3%, compared to 66.3% for others, and unemployment and income inequality are significantly higher (CNN, 2022). This inequality particularly affects young people and those working in agricultural and domestic sectors.

2.2. Legislation and solutions for businesses and people with special needs

Brazil sets up that companies with one hundred employees or more are needed to fill 2% to 5% with people with special needs, known as law quotas, according to Article 1 of Law No 8,213 of July 24, 1991:

> Article 1: Social Security, using contributions, aims to ensure its beneficiaries' indispensable means of maintenance due to incapacity, involuntary unemployment, advanced age, length of service, family burdens, and imprisonment or death of those on whom they depended economically. (BRAZIL,1991, OUR TRANSLATION)

Although the law has been in force for about thirty years, its full enforcement remains limited due to legislative gaps and the lack of professional qualification among people with special needs (SANTOS NETO, 2020).

ISSN: 2447-0767

Recently, more companies have recognized the benefits of including people with disabilities with strong engagement, talent, and work ethics. As a result, organizations are taking proactive steps to recruit and keep these professionals, aiming to reduce turnover and increase productivity (WINIARSKI, 2024). However, the lack of accessible systems still creates barriers to integration, highlighting the need for a digital platform that promotes inclusion and connectivity.

2.3. Daelink system for companies to fill their vacancies for people with special needs

The project focuses on creating a system for web and mobile applications based on JavaScript and with a cloud database that allows business users to connect in a simplified way and thus fill their vacancies through a recommendation system, chat, and the availability of vacancies. The tools described below were used to achieve this.

Figure 1 – Process page website

2.4 Next.js

A React framework for full-stack web applications that serves as a facilitator thanks to its built-in tools, such as simple route mapping and several performance optimizations. It provides automatic configuration for low-level tasks and its own engine libraries and utilities that help create more dynamic and functional React projects. (Next)

2.5. React Native

React Native is a platform-based tool that enables the creation of hybrid applications running on iOS (Apple) and Android. It was created by Facebook in 2013. React Native can be defined as an open-source framework that aims to develop native applications; that is, there is a web layer as an interface, but the native application itself (CASA GRANDE; TANAKA, 2023).

2.6. Expo

Expo, along with React's create-react-app package, provides the necessary structure to develop an application, offering an environment that simplifies the creation of mobile applications (ESCUDELARIO; PINHO, 2020). Expo is a tool used in mobile development with React Native that allows easy access to some native APIs without needing to install additional dependencies or change native code (ROCKETSEAT, 2020). This makes the development process faster and more accessible for developers.

2.7. Python

Python is a highly efficient programming language because programs constrain Dewar lines of code, helping to build “clean” code, obtaining a quick understanding, and debugging (MATTHERS, 2016).

Python often needs added packages that are not included in the Anaconda distribution. One such package manager is Pip, a tool that manages and installs Python packages (MCKINNEY, 2018). In this sense, two types, Conda and Pip, serve different purposes. Conda provides general package management for various languages in the Conda environment, and Pip offers services specifically for Python (MUELLER, 2020).

2.8. Machine Learning

Machine Learning uses data filtering to create current information, generate noteworthy results, and enable intelligent decision-making through the data generated (KNEUSEL, 2024). Technology constantly evolves, and machine learning has become crucial for advancing various commercial areas, which have been adopted by today's most prominent companies, such as Netflix (DOMINGOS, 2017).

2.9. Firebase

Firebase Database is an effective tool for database creation, working through real-time data updates, and cloud-based storage. These features allow multi-platform projects to perform efficiently while offering compatibility with other Google Systems (FIREBASE).

2.10. UML

The Unified Modeling Language is a visual representation that helps to understand the system in its logical parts. It is an accepted international standard for software (GUEDES, 2018).

Used to carry out a complete project, it needs to be able to be changed later and allow a better understanding between customers and developers (PEREIRA, 2011). In general terms, the diagrams that make up the UML approach the entire project in different technical ways to obtain a better result in its completion (GUEDES, 2018).

2.11. SpringBoot

Spring Boot is a framework that simplifies the creation of stand-alone, production-ready applications built on top of the Spring ecosystem. According to the official documentation, its central purpose is to reduce the configuration complexity by offering an opinionated approach to the Spring platform and third-party libraries, allowing developers to start projects quickly and efficiently (SpringBoot).

3. Methodology

For the construction of the project, multiple methodologies were used for a better elaboration of the work, analyzing fundamental points that concern the compliance of people with special needs in the job market and the practical part of companies in this integration. Since these aspects are highlighted by Gil (2002), a solid structure based on bibliographic surveys and quantitative research reworked by the case study provides an adequate perspective on a given subject.

A quantitative study is a method used to analyze problems with data and statistics. We are following the relevance of quantitative data to understand the distribution of opportunities (PEREIRA et al., 2018). By employing this method, we analyze patterns, show statistics, and

ISSN: 2447-0767

the impact of including people with special needs in the job market.

The case study is a methodological process related to an in-depth analysis of the specific phenomenon, examining its multiple dimensions and associated factors to find the root causes of a given problem (GIL, 2017). This type of study is essential for clarifying typical issues in research and practice. It investigates the discrepancy in the job market experienced by people with special needs, helping find the principles contributing to this inequality (GIL, 2002).

Finally, a bibliography study provides an investigative and theoretical analysis of a given theme using articles and bibliographic sources (GIL, 2017). This method analyzes the perspective of researchers in the field to elucidate the members of the job market and their interconnection with individuals with special needs.

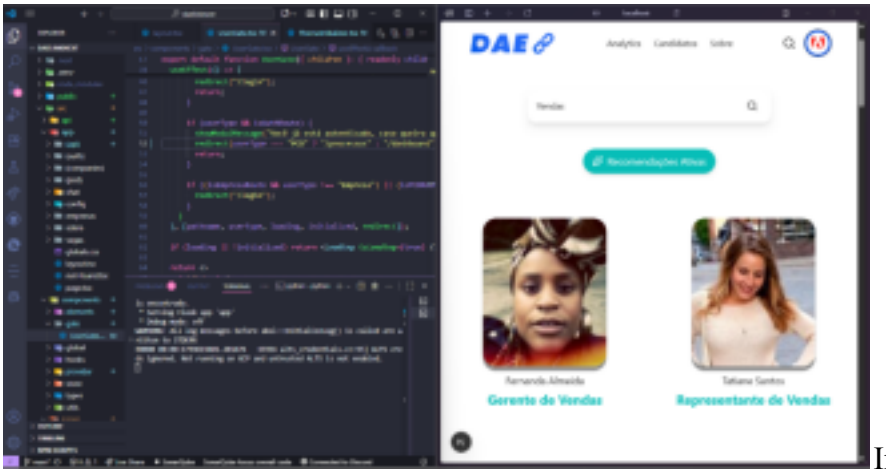
3.1. Technology Use

The technologies mentioned in the preceding chapter are used for formatting and developing the project. Each tool is essential due to its efficacy in constructing the system. React has been used as a library to create the user interface and integrate with the Firebase database.

Python was employed with machine learning to develop the DaeLink recommendation system. The system enables companies to find individuals with special needs registered on the platform efficiently and automatically.

Figure 2 shows the recommendation system with an example of a "Sales" vacancy and the terminal that proves the similarity between the vacancies.

Figure 2 – Screen and terminal on system recommendation



In mobile development, we use React Native with Expo and Firebase, enabling the application to provide access to web and mobile platforms. The mobile application has been designed for people with special needs. It allows them to view and manage the vacancies for which they have applied.

Using the Unified Modelling Language (UML) was relevant to the system's planning, providing its construction through use of cases, sequences, activities, and state machines. The diagrams were of significant value in easing the organization and execution of the project in a clear and structured manner.

In addition to the technologies, the project method was based on the approach proposed by Lakatos and Marconi (2017), Gil (2002), and Pereira et al. (2018). The method provides a systematic approach to defining aims, problem statements, and procedures. This approach eased the organization of the DaeLink development process, integrating quantitative and qualitative analysis to address the challenges faced by individuals with special needs in the job market.

4. Result and discussion

During Daelink's development, several challenges related to accessibility and functionality appeared. The project aimed to simplify hiring through adaptable and inclusive technology. Discussions centered on social and legal aspects, which guided the creation of a systematic adaptation method. As a result, Daelink became an integrated system for managing candidate information and improving recruitment. It was built with React Next.js (web), React Native (app), and Firebase for data storage, alongside Python and other technologies supporting its intelligent recommendation system. Overall, the project offered valuable learning in development and digital social accessibility.

The project is significant concerning social inclusion in the job market, addressing substantial gaps in access to professional opportunities for people with special needs (PWN). By offering a platform that centralizes information and simplifies the inclusion process, Daelink looks to meet the growing need for practical, accessible solutions that promote professional inclusion for people with special needs.

5. Final Consideration

Throughout the project, meaningful results contributed to the advancement of inclusion. The project was designed to automate and ease the filing of remaining job positions for people with special needs, promoting greater inclusion of people with special needs (PWN) in the job market.

Through a detailed analysis of the main barriers faced by both companies and PWN, it was possible to develop an innovative solution that connects these two groups and improves the hiring process based on legal, social, and accessibility criteria. While Brazilian laws have a great responsibility for social inclusion, while research on the subject is being conducted, a lack of enforcement of these laws can create a massive gap in the labor market. Another aggravating factor is society, which is still adapting to all
inclusive practices.

DaeLink differentiates itself by integrating an inclusive digital environment that eases the lives of people with disabilities (PWDs). It offers an adapted platform for document submission and job applications through a simplified selection process with exclusive features. Ultimately, DaeLink demonstrates strong potential to transform the landscape of social and professional inclusion by increasing the participation of people with disabilities in the job market and encouraging companies to adopt more inclusive practices, thereby contributing to a fairer world. In the future, DaeLink could be expanded through partnerships with public policies to reach more regions and enhance its platform with innovative solutions designed for PWDs.

References

BRAZIL. *Law No. 8,213 of July 24, 1991. Establishes social security benefit plans and provides other measures. On the Purpose and Basic Principles of Social Security.* Available at: https://www.planalto.gov.br/ccivil_03/leis/l8213cons.htm. Accessed on: June 7, 2024.

CASA GRANDE, C.; TANAKA, S. *Comparison between the performance of smartphone applications developed in Flutter and React Native: an analysis using sorting algorithms.* Revista Terra & Cultura: Cadernos de Ensino e Pesquisa, v. 39, special issue, p. 7–17, 2023. Available at: http://periodicos.unifil.br/index.php/Revistateste/article/view/2796/2559. Accessed on: May 5, 2024.

CNN Brasil. *IBGE releases an unprecedented study on disability and social inequalities in Brazil: the survey reveals statistics on labor market insertion, income profiles, access to*

ISSN: 2447-0767

education and health services, and housing characteristics for people with disabilities. São Paulo, September 21, 2022. Available at: https://www.cnnbrasil.com.br/nacional/ibge-divulga-estudo-inedito-sobre deficiencia-e-desigualdades-sociais-no-brasil/. Accessed on: July 4, 2024.

CNN Brasil. *Quota law for people with disabilities turns 30 this Saturday.* 2021. Available at: https://www.cnnbrasil.com.br/nacional/lei-de-cotas-para-pessoas-com deficiencia-faz-30-anos-neste-sabado/. Accessed on: June 15, 2024.

DOMINGOS, Pedro. *The Master Algorithm: How the Quest for the Ultimate Machine Learning Algorithm Will Recreate Our World*. São Paulo: Novatec Editora, 2017. 344 p.

ESCUDELARIO, Bruna de Freitas; PINHO, Diego Martins de. *React Native: Mobile Application Development with React*. São Paulo: Casa do Código, 2020. 189 p.

FIREBASE. *Learn the Fundamentals.* [S.l.]: Firebase, 2024. Available at: https://firebase.google.com/docs?hl=pt-br. Accessed on: June 7, 2024.

G1. *Brazil has 18.6 million people with disabilities, approximately 8.9% of the population, according to the IBGE.* 2023. Available at: https://g1.globo.com/economia/noticia/2023/07/07/brasil-tem-186-milhoes-de pessoas-com-deficiencia-cerca-de-89percent-da-populacao-segundo-ibge.ghtml. Accessed on: May 14, 2024.

GIL, Antônio Carlos. How to Develop Research Projects. 4th ed. São Paulo: Atlas, 2002. 175 p.

GIL, Antônio Carlos. How to Develop Research Projects. 6th ed. São Paulo: Atlas, 2017. 192 p.

GUEDES, Gilleanes T. A. *UML 2:* A Practical Approach. 3rd ed. São Paulo: Novatec Editora, 2018. 496 p.

INTERNATIONAL DISABILITY ALLIANCE. *Equalizing Access to the Labor Market.* [S.l.], 2022. Available at: https://www.internationaldisabilityalliance.org/sites/default/files/ida_equalizing_acce ss_to_the_labor_market.pdf. Accessed on: September 26, 2024.

IBGE - Unemployment and informality are higher among people with disabilities | News Agency, October 24, 2022. Available at: https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de noticias/noticias/34977-desemprego-e-informalidade-sao-maiores-entre-as-pessoas com-deficiencia. Accessed on: May 15, 2024.

KNEUSEL, Ronald T. *How Artificial* Intelligence Works: From Magic to Science. São Paulo: Novatec Editora, 2024. 256 p.

LAKATOS, Eva Maria; MARCONI, Marina de Andrade. *Fundamentals of Scientific Methodology*. 8th ed. São Paulo: Atlas, 2017. 368 p.

MATTHERS, Eric. *Python* Crash Course: A Practical, Project-Based Introduction to Programming. São Paulo: Novatec Editora, 2016. 656 p.

MCKINNEY, Wes. *Python for Data Analysis: Data Processing with Pandas, NumPy, and Jupyter*. São Paulo: Novatec Editora, 2018. 616 p.

MENEZES, Nilo Ney Coutinho. *Introduction to Programming with Python: Algorithms and Programming Logic for Beginners*. 2nd ed. São Paulo: Novatec Editora, 2014. 328 p.

MUELLER, John Paul. *Getting Started Programming in Python:* For Dummies. 2nd ed. Rio de Janeiro: Alta Books, 2020. 391 p.

NEXT.JS. *Next.js Documentation*. Available at: https://nextjs.org/docs. Accessed on: October 4, 2025.

PEREIRA, Adriana Soares et al. Scientific Research Methodology. [S.l.]: [s.n.], 2018. 119 p.

PEREIRA, Luiz Antônio Demoraes. *Systems Analysis and Modeling with the UML: with tips and solved exercises*. Rio de Janeiro: Author's Edition, 2011. 282 p.

RIBEIRO, L.; PINHEIRO, S.; DELLATORRE, F. *Challenges of Including People with Disabilities in the Job Market: a study on the perception of those involved.* 2015. Available at: https://www.uricer.edu.br/site/pdfs/perspectiva/148_537.pdf. Accessed on: October 20, 2024.

ROCKETSEAT. Expo and React Native: the union that is transforming mobile development. Rocketseat Blog, October 9, 2020. Available at: https://blog.rocketseat.com.br/expo-react-native/. Accessed on: June 9, 2024.

SANTOS NETO. *Difficult Insertion of People with Disabilities in the Job Market.* Campinas, September 23. 2020. Available at: https://unicamp.br/unicamp/ju/noticias/2020/09/23/dificil-insercao-de-pessoas-com deficiencia-no-mercado-de-trabalho/. Accessed on: September 17, 2024.

SCHMITZ, Daniel; GEORGII, Daniel Pedrinha. *React – A Beginner's Guide: Master the JavaScript library used by Facebook and Instagram*. São Paulo: Leanpub, 2015. 51 p.

SPRING. *Spring Boot Documentation*. Available at: https://docs.spring.io/spring boot/index.html. Accessed on: October 4, 2025.

WINIARSKI, Diane. *How People with Disabilities Make a Positive Impact on the Workplace.* 2024. Available at: https://www.forbes.com/sites/dianewiniarski/2024/01/30/how-people-with disabilities-make-a-positive-impact-in-the-workplace/. Accessed: October 19, 2024.

ISSN: 2447-0767

Desenvolvimento de um Chatbot para Anamnese Psicológica em Sistema de Atendimento Online Utilizando o Framework Rasa

Maria Fernanda Rocha Tolentino[1], Fábio Castro Araújo[1]

[1]Departamento de Computação ULBRA Palmas -
Av. Joaquim Teotônio Segurado, 1501 - Plano Diretor Sul, Palmas - TO, 77019-900

mariafernandarocha@rede.ulbra.br, fabio.araujo@ulbra.br

Resumo. *O crescimento do atendimento psicológico online evidenciou a necessidade de aprimorar a etapa inicial de coleta de informações do paciente. Este artigo apresenta o desenvolvimento de um chatbot voltado à condução da anamnese psicológica, com o objetivo de automatizar e padronizar essa fase do processo clínico. A solução foi construída utilizando o framework Rasa e estruturada a partir da análise comparativa de seis modelos de anamnese, resultando em um roteiro organizado em blocos temáticos e acompanhado da definição dos tipos de resposta. A metodologia envolveu a configuração dos módulos do Rasa, integração com banco de dados e aplicação de diretrizes de privacidade e consentimento. Os resultados indicam a viabilidade de automatizar o processo inicial e sugerem potencial para reduzir o tempo de atendimento e tornar mais consistente a coleta de registros, a partir de testes funcionais. O estudo discute o impacto do chatbot no fluxo de trabalho do psicólogo, suas limitações linguísticas e as possibilidades de expansão para outras áreas da saúde.*

1. Introdução

O aumento da procura por atendimentos psicológicos nos últimos anos tornou-se um desafio para profissionais da saúde mental. Durante o período de isolamento social, o atendimento online emergiu como alternativa viável, ainda que com limitações impostas pelas normas do Conselho Federal de Psicologia. Estudos indicam que, apesar de eficiente para o contexto, essa modalidade trouxe também sobrecarga aos profissionais, escassez de recursos humanos e a necessidade de adaptação das práticas clínicas [DE ARAUJO PASSOS et al. 2021].

Esse cenário indica a utilidade de soluções tecnológicas voltadas à organização do fluxo de atendimento e à continuidade do cuidado. O presente trabalho está vinculado ao projeto OnTerapia, uma plataforma de teleatendimento psicológico desenvolvida para aprimorar a comunicação entre pacientes e profissionais. O sistema prevê funcionalidades como videoconferência para realização de sessões, transcrição automática com tecnologia *Whisper*, chatbot para condução da anamnese, mecanismos de criptografia e armazenamentos no IPFS, controles de acesso e módulos de análise emocional. No contexto deste artigo, focamos no módulo de anamnese desenvolvido sobre o Rasa.

Entre os diversos processos que compõem o atendimento psicológico, destaca-se a anamnese como um dos pontos mais sensíveis e que mais demanda tempo do profissional. Tradicionalmente, esse processo é conduzido pelo profissional durante a primeira consulta, demandando tempo considerável e apresentando variações na qualidade e na padronização das informações registradas. A anamnese possui papel central para a compreensão do paciente, pois, como aponta [Ramos 2011, p. 97], "a anamnese é um momento crucial do diagnóstico, por meio dessa entrevista questões relativas à história de vida do paciente, bem como normas, preconceitos, expectativas, padrões familiares e a circulação dos afetos e do conhecimento ficam evidenciados". Esse caráter abrangente evidencia a necessidade de sistematização e consistência na coleta dessas informações.

Para enfrentar esse desafio, este projeto propõe o desenvolvimento de um *chatbot* voltado à realização da anamnese inicial de pacientes em um sistema de atendimento psicológico online. A solução foi construída utilizando o *framework* Rasa, que permite a criação de agentes conversacionais personalizados e com capacidade de integração a fluxos clínicos digitais. O objetivo principal é disponibilizar uma ferramenta segura, capaz de coletar dados essenciais sobre histórico, queixas, sintomas e contexto familiar, fornecendo subsídios padronizados para o trabalho do psicólogo.

A escolha dessa solução baseia-se em três objetivos principais: a padronização das informações obtidas, a redução do tempo destinado à anamnese durante a consulta e a melhoria da experiência do paciente, que pode interagir de forma mais confortável e gradual com a plataforma. Além disso, a automação do processo fortalece a consistência dos registros clínicos e potencializa a integração com sistemas institucionais de saúde.

O estudo delimita-se à anamnese inicial, compreendendo dados básicos sobre histórico pessoal e familiar, sintomas relatados e experiências anteriores de tratamento, sem avançar para etapas posteriores do acompanhamento clínico. A ênfase recai sobre a aplicação do Rasa como base tecnológica e sobre a integração ao sistema de atendimento online desenvolvido.

O artigo está organizado em seis seções principais. A Seção 2 apresenta a fundamentação teórica sobre a anamnese psicológica, o uso de *chatbots* na saúde e as principais características do *framework* Rasa. A Seção 3 descreve a metodologia adotada, abordando a arquitetura do sistema e o processo de desenvolvimento do protótipo. A Seção 4 detalha os resultados obtidos com a implementação do *chatbot*, enquanto a Seção 5 discute os resultados obtidos e suas implicações práticas. A Seção 6 reúne as conclusões e propõe possíveis trabalhos futuros. Por fim, são apresentadas as referências utilizadas ao longo da pesquisa.

2. Fundamentação Teórica

2.1. Atendimento psicológico online

O atendimento psicológico mediado por tecnologias digitais consolidou-se como prática regulamentada e reconhecida no Brasil, acompanhando o crescimento da demanda por serviços de saúde mental. Em 2000, o Conselho Federal de Psicologia instituiu a Comissão Nacional de Credenciamento e Fiscalização dos Serviços de Psicologia pela Internet, por meio da Resolução nº 06/2000. Cinco anos depois, a Resolução nº 12/2005 passou a normatizar a orientação psicológica e outros serviços mediados por
computador, revogando a normativa anterior e estabelecendo parâmetros para o exercício ético do atendimento remoto [Gonçalves and Neto 2023]. Essa regulamentação assegurou a expansão do atendimento psicológico online, que traz benefícios como a ampliação do acesso a pacientes geograficamente distantes, a flexibilização de horários e a redução de barreiras logísticas.

Além da praticidade, o atendimento online contribui para a democratização do acesso à saúde mental, sobretudo em regiões com escassez de profissionais. No entanto, ele também impõe desafios relacionados à preservação da privacidade, à segurança da informação e à manutenção do vínculo terapêutico em ambientes virtuais. Tais fatores demandam estratégias que conciliem eficiência tecnológica e sensibilidade humana, promovendo uma prática ética e acolhedora. Nesse cenário, sistemas digitais integrados, como o OnTerapia, tornam-se aliados valiosos para a gestão de atendimentos e o suporte à prática clínica, especialmente quando incorporam recursos de automação e inteligência artificial.

ISSN: 2447-0767

2.2. Anamnese em psicologia

Nesse contexto, destaca-se a anamnese como etapa inicial e essencial do processo diagnóstico em psicologia. Ao iniciar o psicodiagnóstico, é fundamental a coleta de informações aprofundadas sobre o avaliando, abrangendo áreas relevantes de sua vida e os motivos que o levaram a buscar atendimento. Para esse levantamento, que subsidia a formulação de hipóteses diagnósticas e a escolha de técnicas a serem utilizadas, os psicólogos realizam uma entrevista detalhada sobre a história de vida do paciente. Esse procedimento, denominado entrevista de anamnese, constitui recurso central que fundamenta todo o processo de avaliação psicológica [HUTZ et al., 2016]. A sistematização dessa etapa é essencial para garantir consistência nos registros clínicos e oferecer ao profissional uma base sólida para a condução do atendimento.

A anamnese também representa o primeiro contato significativo entre paciente e psicólogo, momento em que se estabelece o vínculo terapêutico e se avaliam aspectos emocionais, comportamentais e contextuais. Por envolver dados pessoais e sensíveis, requer acolhimento, escuta ativa e rigor ético em sua condução. A transformação digital da psicologia, portanto, tem impulsionado o desenvolvimento de instrumentos que auxiliam na coleta e organização dessas informações, buscando equilibrar a precisão técnica com a empatia necessária à prática clínica. Nesse sentido, o uso de chatbots aplicados à anamnese surge como alternativa promissora para padronizar a coleta de dados sem perder o foco no cuidado humanizado.

2.3. Chatbots e inteligência artificial

A incorporação de tecnologias inteligentes ao campo da saúde tem impulsionado o uso de *chatbots* como facilitadores na coleta de dados e na interação inicial com pacientes. Segundo Aquino e Adaniya (2018), "um *chatterbot* é um sistema de conversação por computador que interage com usuários humanos por meio de uma linguagem conversacional natural. Os primeiros *chatterbots* eram aplicados restritamente a estudos acadêmicos. Atualmente, eles são considerados alternativas capazes de desempenhar o papel de facilitadores em diversas aplicações como, por exemplo, uso pedagógico, comercial, social, ensino a distância entre outros". Em linhas gerais, os *chatbots* podem
ser baseados em regras, respondendo a fluxos previamente estruturados, ou em aprendizado de máquina, empregando inteligência artificial para interpretar linguagem, adaptar-se ao contexto e gerar respostas mais dinâmicas.

Na área da saúde, essas ferramentas vêm sendo aplicadas em triagem de sintomas, acompanhamento de tratamentos, lembretes de medicação e apoio à saúde mental, oferecendo benefícios como otimização do tempo clínico, redução da sobrecarga profissional e ampliação do acesso dos pacientes a orientações iniciais. No caso específico da anamnese psicológica, o uso de *chatbots* possibilita padronizar a coleta de informações, aumentar a eficiência do atendimento e contribuir para a integração de dados a sistemas de gestão em saúde, tornando o processo mais ágil e consistente.

3. Trabalhos Relacionados

O estudo de Johann (2021) apresenta um chatbot voltado ao treinamento de estudantes de Psicologia para a condução de entrevistas psicopatológicas simuladas. Desenvolvido com a linguagem AIML (*Artificial Intelligence Markup Language*) e o interpretador *Program-Y,* o sistema foi integrado a uma aplicação web que media a interação entre usuário e chatbot. O modelo foi avaliado por profissionais e estudantes, que relataram facilidade de uso, boa compreensão das respostas e conforto durante a interação. Os resultados indicaram que o

chatbot é mais eficaz na etapa inicial da entrevista clínica, englobando o histórico, o comportamento e as demandas do paciente, demonstrando potencial como ferramenta de apoio teórico-prático no ensino de psicopatologia.

De forma complementar, Grandi (2024) desenvolveu um simulador virtual de clínica médica com um paciente virtual alimentado por inteligência artificial generativa (Google Gemini), voltado ao treinamento de estudantes de medicina e enfermagem em anamnese clínica. O ambiente, criado na plataforma *Unity*, integra IA generativa e banco de dados de patologias e sintomas, permitindo diálogos naturais e realistas. O projeto relata ter atingido seus objetivos, apontando que a IA generativa pode aproximar a interação do contexto clínico no cenário de simulação, tornando o aprendizado mais imersivo e contribuindo para o desenvolvimento das habilidades de comunicação diagnóstica dos estudantes.

Por sua vez, Diógenes, De Paula e Jorge (2022) propuseram um protocolo de revisão de escopo sobre o uso de chatbots para promoção do autocuidado em saúde mental de profissionais da saúde durante a pandemia de COVID-19. O estudo ressalta que agentes conversacionais podem oferecer apoio psicológico contínuo, anônimo e acessível, auxiliando na redução de estresse e ansiedade. Em conjunto, os três trabalhos evidenciam a expansão dos chatbots na saúde, desde o ensino e simulação clínica até o suporte emocional, e fundamentam a proposta deste estudo, que aplica a tecnologia conversacional ao contexto da anamnese psicológica, buscando otimizar a coleta de dados e fortalecer o vínculo inicial entre paciente e profissional.

4. Metodologia

4.1. Materiais

O principal recurso tecnológico adotado para o desenvolvimento do *chatbot* foi o *framework* Rasa, uma plataforma *open source* amplamente utilizada na construção de assistentes conversacionais. Diferentemente de interações tradicionais do tipo FAQ *(Frequently Asked Questions)*, o Rasa se baseia em conversas naturais, considerando o contexto anterior e lidando de forma flexível com desvios no diálogo [Sharma and Joshi 2020] (tradução nossa). Sua arquitetura é composta por módulos que atuam de forma integrada: o Rasa NLU (*Natural Language Understanding*), que pode ser comparado ao "ouvido" do sistema, responsável por interpretar a linguagem natural e identificar intenções e entidades, e o Rasa Core, equivalente ao "cérebro", que decide os próximos passos do diálogo com base nas entradas do usuário [Sharma and Joshi 2020](tradução nossa). Além desses, o *Domain* estabelece intenções, *slots*, respostas e fluxos de conversa, enquanto o módulo de *Actions* permite a criação de ações personalizadas, incluindo integrações externas e persistência de dados.

A escolha do Rasa se justifica por vantagens relevantes: trata-se de uma ferramenta de código aberto, com alto nível de customização, ampla documentação e suporte para integração com APIs e sistemas externos. Como apontam [Sharma and Joshi 2020] (tradução nossa), sua extensibilidade e licença aberta o tornam mais versátil do que outras plataformas corporativas de *chatbots*, como *Dialogflow* e *Microsoft Bot*. Esses aspectos permitiram adequar o assistente às necessidades específicas da anamnese psicológica, tanto na padronização das perguntas quanto na coleta e armazenamento das respostas dos pacientes.

No contexto deste trabalho, o Rasa foi utilizado para estruturar a coleta de informações referentes à anamnese inicial. O módulo *Actions* foi fundamental para implementar funcionalidades como salvar anamneses personalizadas, armazenar respostas dos pacientes, buscar anamneses por código de identificação e validar acessos durante o login do usuário.

ISSN: 2447-0767

Além disso, o projeto utilizou o *Domain* para definir perguntas padrão relacionadas a dados pessoais, histórico de saúde e queixa principal, enquanto o NLU foi responsável por compreender comandos e intenções básicas dos usuários.

De for essamento da linguagem natural ii) a definição de regras e respost meio do módulo *Actions*. Essa es ibilita a integração futura com sis onais.

4.2. Mé

A Figu envolvimento do chatbot, distribu oleto de pesquisa, análise, implementação e validação do sistema.

Figura 1 – Etapas metodológicas do desenvolvimento do chatbot de anamnese psicológica.

Conforme ilustrado na Figura 1, o desenvolvimento do trabalho seguiu um ciclo iterativo, fundamentado em encontros semanais com o orientador para discutir a proposta e alinhar o escopo do projeto. No Passo 1, foram realizadas pesquisas exploratórias sobre *frameworks* conversacionais capazes de subsidiar a construção do *chatbot*. Após a análise de diferentes alternativas, e considerando a viabilidade técnica e a orientação acadêmica, optou-se pelo uso do *framework* Rasa, escolhido por sua flexibilidade, capacidade de customização e integração com sistemas externos.

No Passo 2, iniciou-se uma etapa primordial voltada ao estudo de modelos de anamnese psicológica já consolidados. Foi elaborado um levantamento comparativo de seis modelos distintos, em formato de planilha, no qual foram listadas suas perguntas e identificadas aquelas que apareciam com maior frequência. A partir dessa análise, construiu-se uma Proposta de Anamnese Unificada, contemplando perguntas e tipos de respostas sugeridas. Essa proposta foi submetida à validação do orientador e da psicóloga Elisa Lopes, assegurando pertinência clínica e adequação ao contexto de uso.

Com a validação concluída, avançou-se para o Passo 3 de desenvolvimento, na qual o projeto

foi estruturado dentro do Rasa. Nesse estágio, o *Domain* foi configurado para contemplar as intenções, *slots* e respostas correspondentes às perguntas da anamnese. O NLU foi ajustado para interpretar comandos básicos, enquanto o módulo de *Actions* foi programado para gerenciar funcionalidades essenciais, como salvar anamneses personalizadas, registrar respostas, validar códigos de acesso e listar informações previamente coletadas.

O Passo 4 correspondeu à fase de testes, na qual foram realizados experimentos com diferentes fluxos de perguntas e respostas. Nessa fase, foram avaliados cenários de entrada textual livre e opções de resposta categorizadas, incluindo formatos booleanos, de modo a verificar a consistência das interações, a precisão na coleta de dados e a robustez do sistema frente a diferentes possibilidades de entrada do usuário.

Os resultados dos testes confirmaram o funcionamento correto do fluxo de coleta e registro das anamneses, evidenciando o correto funcionamento do fluxo proposto nos testes realizados em compreender diferentes tipos de respostas e realizar as integrações necessárias com serviços de banco de dados e armazenamento. A seguir, apresentam-se os resultados obtidos com a aplicação da metodologia proposta.

5. Resultados e Discussões

O desenvolvimento do *chatbot* para anamnese psicológica teve como ponto de partida um processo detalhado de pesquisa e sistematização de formulários clínicos utilizados por psicólogos em atendimentos presenciais. Essa etapa inicial foi registrada em planilhas comparativas que reuniram seis modelos distintos de anamnese, extraídos de materiais acadêmicos e profissionais. Cada modelo foi analisado quanto à estrutura, linguagem e profundidade das perguntas, permitindo identificar convergências e lacunas entre eles. As perguntas foram classificadas de acordo com seus temas centrais, como dados pessoais, histórico clínico, hábitos, emoções, relações interpessoais e expectativas em relação à terapia.

A partir dessa análise, elaborou-se a Proposta de Anamnese Unificada, consolidada em uma planilha própria que reuniu os itens mais recorrentes dos modelos estudados. A síntese dessa comparação encontra-se no Quadro 1.

				Estado	**Profissão**	**Escolaridade**	**Endereço**
Acolhimento Psicológico	X	X	-	X	-	-	X
Anamnese Ocupacional	X	X	X	X	X	-	X
Psicoterapia Breve (UFAM)	X	X	X	X	X	X	-
SCID-5 (DSM-5)	-	-	-	-	-	-	-
Ficha Anamnese Simples	X	X	-	-	X	X	X
Anamnese Psicológica Adulto	X	X	-	X	X	X	X

Quadro 1 – Síntese comparativa dos modelos de anamnese analisados.
Fonte: elaboração própria (2025).

Essa proposta foi revisada pelo orientador e validada pela psicóloga Elisa Lopes, que colaborou com ajustes de redação e sequência lógica, assegurando coerência clínica e

ISSN: 2447-0767

acolhimento durante a coleta de dados. O resultado foi um roteiro dividido em blocos temáticos, com perguntas abertas e fechadas, acompanhadas da definição do tipo de resposta mais adequado (texto livre, sim/não ou numérico). A distribuição dos blocos temáticos e dos respectivos tipos de resposta está na Tabela 1. Esse instrumento serviu como base para a configuração do fluxo de diálogo no *framework* Rasa, definindo a sequência de interações e os parâmetros de validação das respostas durante o atendimento simulado.

Categoria	**Pergunta**	**Tipo de Resposta**	**Observações**
Identificação	Nome completo	Texto	Campo livre para resposta
Identificação	Nome de preferência	Texto	Campo livre para resposta
Identificação	Idade	Número	Em anos
Identificação	Estado civil	Texto	Campo livre para resposta
Identificação	Trabalha atualmente	Booleano	Sim / Não
Identificação	Sobre a Profissão	Texto	Campo livre para resposta
Identificação	Estuda atualmente	Booleano	Sim / Não
Identificação	Sobre o Estudo	Texto	Campo livre para resposta
Identificação	Motivo Atendimento	Texto	Campo livre para resposta
Identificação	Tempo Situacao	Texto	Campo livre para resposta
Identificação	Buscou Ajuda Anterior	Texto	Campo livre para resposta
Identificação	Experiencia Ajuda	Texto	Campo livre para resposta

Tabela 1 – Blocos temáticos e tipos de resposta adotados no protótipo.
Fonte: elaboração própria (2025).

Durante a fase de implementação, os dados da anamnese unificada foram adaptados para os arquivos de configuração do Rasa. O arquivo *domain.yml* recebeu a estrutura das perguntas, definindo intenções, respostas e tipos de *slots*, enquanto o *nlu.yml* foi utilizado para treinar o modelo de linguagem natural com exemplos de expressões dos usuários. Nos arquivos *stories.yml* e *rules.yml*, foram descritas as sequências de interação e as condições de transição entre as perguntas, assegurando que o *chatbot* conduzisse o diálogo de maneira natural e linear. O arquivo *actions.py* foi configurado com ações personalizadas responsáveis por controlar o comportamento do sistema e registrar as respostas do usuário.

A partir dessa estrutura, o *chatbot* tornou-se capaz de conduzir uma entrevista psicológica inicial, desde a saudação e o consentimento do paciente até a coleta das informações principais. O fluxo de interação segue uma sequência lógica baseada nos blocos temáticos definidos na pesquisa inicial, com maior padronização e coerência no processo. As respostas são processadas em tempo real, e o sistema valida o tipo de entrada conforme o formato esperado, evitando interrupções ou respostas inválidas.

Os testes funcionais foram realizados diretamente no terminal do Rasa, utilizando o comando *rasa shell*. Nessa etapa, foram executadas simulações completas de diálogo para avaliar o comportamento do sistema diante de diferentes padrões de resposta. O *chatbot* demonstrou compreender corretamente as intenções de início de conversa, consentimento e encerramento, além de seguir a ordem adequada das perguntas. A Figura 2 apresenta um exemplo de interação completa registrada durante os testes realizados no terminal do Rasa.

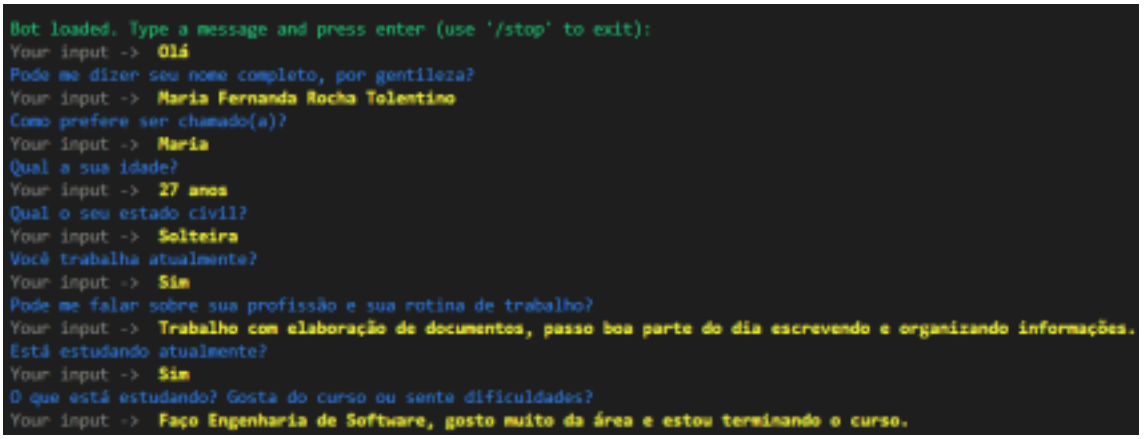


Figura 2 – Interação entre o usuário e o *chatbot* de anamnese psicológica desenvolvido com o *framework* Rasa.

Além dos testes no terminal, a aplicação foi integrada a uma interface visual que reproduz a experiência do paciente durante o preenchimento da anamnese. A Figura 3 apresenta o ambiente gráfico do *chatbot*, onde o usuário responde às perguntas de forma sequencial e intuitiva, com uma interface que privilegia a clareza e a acessibilidade.

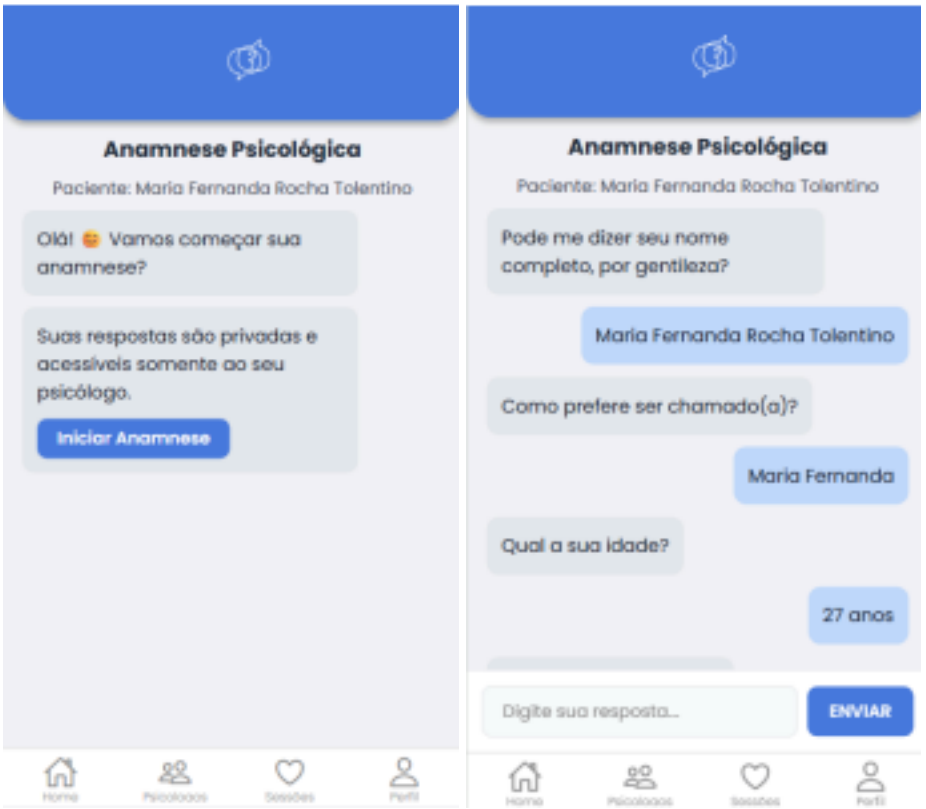


Figura 3 – Interface visual do *chatbot* de anamnese psicológica.

Os resultados obtidos apontaram viabilidade técnica do uso do Rasa na estruturação de entrevistas clínicas automatizadas. Nas simulações, o sistema operou de forma estável e clara nas interações, o que sugere potencial de uso como ferramenta de apoio ao atendimento psicológico online.

ISSN: 2447-0767

A partir desses resultados, observa-se que a aplicação desenvolvida também traz contribuições significativas para o aprimoramento do processo de anamnese psicológica, ao introduzir uma interface automatizada que auxilia o psicólogo na coleta inicial de informações do paciente. O chatbot funciona como um recurso complementar, voltado à otimização do tempo e à padronização dos registros preliminares. A automação dessa etapa pode possibilitar que o profissional inicie o atendimento com um panorama prévio das informações básicas, favorecendo o direcionamento da escuta e o planejamento das intervenções.

A utilização do *framework* Rasa mostrou-se adequada para esse propósito, pois oferece flexibilidade na definição das intenções, fluxos e respostas do diálogo, além de possibilitar a criação de ações personalizadas. Essa arquitetura permitiu que o *chatbot* conduzisse o diálogo de forma estruturada, mantendo a coerência entre as perguntas e a sequência dos temas. A padronização obtida com esse modelo contribui para maior consistência na coleta de dados clínicos, especialmente em ambientes de atendimento online.

Durante o desenvolvimento, foram observadas algumas limitações. Embora o sistema compreenda comandos básicos e realize a transição entre perguntas com precisão, ele apresenta restrições diante de variações linguísticas, gírias ou expressões ambíguas. Outro ponto importante é a necessidade de definir previamente o tipo de resposta esperado para cada pergunta. Essa configuração, que diferencia respostas em texto livre, sim ou não e numéricas, garante a integridade dos dados e a validação das entradas, mas pode limitar a espontaneidade do paciente em determinadas situações. Ainda assim, essa estrutura serve como base para a evolução do sistema, permitindo a aplicação de validações específicas conforme o tipo de dado coletado.

A aplicação também inclui um módulo voltado ao profissional de psicologia, que possibilita a personalização completa do roteiro de anamnese. Esse módulo permite adicionar novas perguntas, editar ou excluir itens e escolher o tipo de resposta mais adequado para cada caso, conforme demonstrado na Figura 4. Essa funcionalidade amplia o potencial de uso do sistema, adaptando-o a diferentes linhas teóricas da Psicologia e também a outras áreas da saúde. Profissionais como médicos, fisioterapeutas e nutricionistas poderiam ajustar o conteúdo das perguntas e o fluxo de interação de acordo com as necessidades específicas de suas consultas, mantendo a mesma estrutura conversacional desenvolvida.

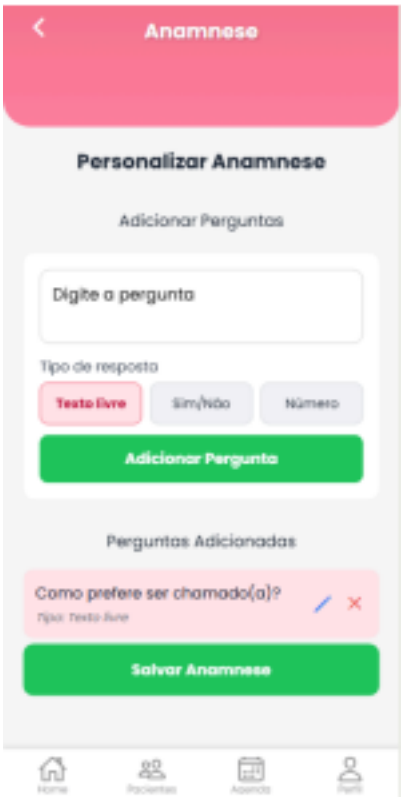


Figura 4 – Módulo de personalização de perguntas da anamnese desenvolvido para o psicólogo.

Com esses resultados, o *chatbot* demonstra potencial para expansão e adaptação, preservando o equilíbrio entre padronização técnica e flexibilidade profissional. Apesar das limitações linguísticas e da dependência de respostas pré-definidas, o sistema mostrou-se funcional, coerente e alinhado às boas práticas de acolhimento e organização de dados no contexto de teleatendimento psicológico.

6. Considerações Finais

O desenvolvimento do chatbot de anamnese psicológica demonstrou a viabilidade de utilizar o *framework* Rasa como base para a automação da coleta inicial de informações em atendimentos psicológicos online. A aplicação apresentou estabilidade e coerência nas interações. Em testes funcionais, o desenho do fluxo e a padronização mostraram potencial para otimizar o tempo de atendimento e aumentar a consistência dos registros.

Os resultados indicam que agentes conversacionais podem apoiar o trabalho do psicólogo nas etapas iniciais, desde que empregados como ferramenta complementar sob supervisão profissional, sem comprometer o vínculo humano essencial à prática clínica. O modelo desenvolvido consolidou uma estrutura funcional, adaptável e compatível com diferentes contextos de atendimento remoto.

Como trabalhos futuros, destacam-se a ampliação do vocabulário do chatbot, o aprimoramento da compreensão de diferentes formas de expressão e variações linguísticas, a integração ao modelo de inteligência artificial do sistema OnTerapia para análise de conteúdo das respostas dos pacientes e a realização de estudos de usabilidade com amostras maiores, voltados à avaliação do impacto clínico do uso da ferramenta.

O trabalho reafirma o potencial de tecnologias conversacionais como instrumentos complementares de apoio à prática psicológica, desde que aplicadas com rigor ético,

ISSN: 2447-0767

validação profissional e supervisão técnica adequadas.

7. Referências

DE ARAUJO PASSOS, Alana Gabriela et al. O aumento das doenças psicossomáticas durante a pandemia e dificuldades no atendimento psicológico. *Research, Society and Development*, v. 10, n. 8, p. e10710817004–e10710817004, 2021.

DE OLIVEIRA AQUINO, Victor Hugo; DA COSTA ADANIYA, Mario Henrique Akihiko. Desenvolvimento e aplicações de Chatbot. *Revista Terra & Cultura: Cadernos de Ensino e Pesquisa*, v. 34, p. 56–68, 2018.

DIOGENES, Carina Nogueira; DE PAULA, Milena Lima; JORGE, Maria Salete Bessa. Chatbot como instrumento de promoção do autocuidado em saúde mental para profissionais da saúde que atuam na linha de frente durante a pandemia de covid-19: Protocolo de revisão de escopo. *RECIMA21 – Revista Científica Multidisciplinar*, v. 3, n. 11, p. e3112119–e3112119, 2022.

GONÇALVES, Charlisson Mendes; NETO, João Leite Ferreira. O atendimento psicológico on-line: Uma revisão sistemática da literatura. *Revista Foco*, v. 16, n. 5, p. e1723–e1723, 2023.

GRANDI, Luigi Marson. *Simulador virtual de clínica médica – anamnese: inteligência artificial generativa*. Orientador: Ivando Severino Diniz. 2024. 67 p. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Controle e Automação) – Instituto de Ciência e Tecnologia, Universidade Estadual Paulista, Sorocaba, 2024.

HUTZ, Claudio Simon et al. *Psicodiagnóstico: avaliação psicológica*. Artmed Editora, 2016.

JOHANN, Guilherme Alexandre dos Santos. *Entrevista simulada mediante um chatbot para investigação psicopatológica*. 2021.

RAMOS, Maria Inês Paton. A entrevista de anamnese sob a ótica do referencial teórico psicodramático: uma contribuição para a psicopedagogia. *Revista Psicopedagogia*, v. 28, n. 85, p. 97–102, 2011.

SHARMA, Rakesh Kumar; JOSHI, Manoj. An analytical study and review of open source chatbot framework, rasa. *Int. J. Eng. Res.*, v. 9, n. 06, p. 1011–1014, 2020.

StructLive: desenvolvimento de uma plataforma extensível para o ensino de Estruturas de Dados

Raphael Henrique Scheffler Ferreira[1], Fabiano Fagundes[2], Jackson Gomes de Souza[2]

[1]Departamento de Computação
Universidade Luterana do Brasil – Palmas – TO

raphaelscheffler47@rede.ulbra.br, fabiano.fagundes@ulbra.br,
jackson.souza@ulbra.br

Resumo: *Este trabalho detalha o desenvolvimento da StructLive, uma plataforma educacional projetada para o ensino de Estruturas de Dados por meio de uma abordagem prática, visual e interativa. O projeto foi concebido com foco em modularidade e escalabilidade, permitindo que novos conteúdos e funcionalidades possam ser adicionados progressivamente. Diferentemente de soluções estáticas, a StructLive foi desenhada como uma estrutura-base, capaz de suportar múltiplos módulos, animações e recursos de avaliação. A metodologia contemplou o uso de tecnologias modernas, como Next.js, TypeScript e Supabase. Os resultados demonstram que a arquitetura proposta oferece um ambiente intuitivo e passível de expansão, adequado tanto para uso acadêmico quanto profissional.*

1. Introdução

O ensino de Estruturas de Dados é um desafio recorrente nos cursos de Ciência da Computação e Engenharia de Software, dada a complexidade dos conceitos e a dificuldade de visualização prática. Plataformas que apresentam exemplos animados e atividades interativas podem facilitar a aprendizagem, tornando o processo mais envolvente e acessível. Embora existam ferramentas de visualização, muitas são limitadas a um conjunto fixo de conteúdos, carecem de interatividade genuína ou possuem arquiteturas que dificultam sua expansão e manutenção. Essa rigidez resulta em obsolescência tecnológica e pedagógica, criando uma lacuna por soluções que sejam não apenas eficazes, mas também evolutivas.

O projeto StructLive surge como resposta a essa lacuna, com um diferencial estratégico, priorizando a engenharia da plataforma em si: um núcleo preparado para receber incrementos contínuos. Assim, conteúdos como listas, pilhas, filas, árvores, além de exercícios, testes e animações, podem ser adicionados gradualmente, sem comprometer a arquitetura geral.

O objetivo do projeto é desenvolver uma plataforma educacional para o ensino de Estruturas de Dados, com foco em modularidade, escalabilidade e interatividade. Para isso, foram trabalhados os seguintes objetivos: projetar e implementar uma arquitetura de software extensível, capaz de suportar múltiplos módulos de ensino independentes; desenvolver recursos de visualização dinâmica que permitam aos estudantes observar as operações sobre as estruturas de dados em tempo real; implementar um ambiente interativo onde os discentes possam submeter algoritmos e analisar sua execução passo a passo; validar a arquitetura proposta através da implementação completa de um módulo de ensino inicial (lista dinâmica simplesmente encadeada).

O foco deste artigo recai especificamente sobre o desenvolvimento da arquitetura de software e sua primeira implementação, com o objetivo de validar o modelo proposto. Dessa

ISSN: 2447-0767

forma, a pesquisa concentra-se na definição e construção de uma arquitetura extensível, capaz de suportar múltiplos módulos de ensino independentes, e na implementação inicial de um módulo referente à lista dinâmica simplesmente encadeada, que servirá como prova de conceito para avaliar a adequação, a robustez e a potencialidade da solução arquitetural.

2. Fundamentação Teórica

2.1. Estruturas de Dados

As Estruturas de Dados constituem um dos pilares centrais da Ciência da Computação e da Engenharia de Software, pois definem as formas pelas quais as informações são organizadas, manipuladas e armazenadas em memória. Segundo Cormen et al. (2009), a escolha da estrutura correta é tão importante quanto a elaboração do algoritmo, uma vez que afeta diretamente o desempenho e a escalabilidade da aplicação. Em sistemas de grande porte, como bancos de dados e redes, a eficiência no gerenciamento de dados é um fator decisivo para o sucesso do software.

O estudo das Estruturas de Dados abrange uma variedade de modelos, como listas, pilhas, filas, árvores e grafos. Pilhas (LIFO) são utilizadas em algoritmos de retrocesso e compilação, enquanto filas (FIFO) são essenciais em sistemas de escalonamento de processos. Árvores balanceadas permitem operações eficientes de busca, inserção e remoção, fundamentais para sistemas de arquivos e bancos de dados (GOODRICH; TAMASSIA; GOLDWASSER, 2011). Grafos, por sua vez, são indispensáveis em problemas de redes e inteligência artificial.

Apesar de sua importância fundamental, a aprendizagem de Estruturas de Dados é notoriamente complexa. Knuth (1997) já apontava a dificuldade dos estudantes em internalizar a manipulação de ponteiros e a alocação dinâmica de memória. Essa dificuldade é ampliada quando o conteúdo é apresentado de forma puramente textual, desprovido de representações visuais que materializam o comportamento dinâmico das estruturas. É precisamente nesse ponto que a visualização computacional se torna uma ferramenta pedagógica poderosa, pois, como demonstram estudos de Shaffer (2001), ela aumenta significativamente a capacidade de compreensão e aplicação prática dos algoritmos.

2.2. Engenharia de Software

A Engenharia de Software provê a disciplina e as metodologias para construir sistemas complexos de forma sistemática e controlada (SOMMERVILLE, 2011). Princípios como modularidade, baixo acoplamento e reuso de software são cruciais para a manutenibilidade e longevidade de um projeto (PRESSMAN; MAXIM, 2016; KRUEGER, 1992). Um sistema modular, onde componentes com responsabilidades bem definidas interagem através de interfaces claras, é inerentemente mais fácil de compreender, testar e evoluir.

Nesse sentido, a arquitetura de software atua como o elo entre a teoria da Engenharia de Software e sua aplicação prática, definindo a estrutura organizacional do sistema e os padrões de interação entre seus componentes. Uma arquitetura bem projetada permite que a evolução de funcionalidades ocorra de forma controlada, preservando a integridade do sistema ao longo do tempo. Conforme destacam Bass, Clements e Kazman (2013), decisões arquiteturais tomadas nas fases iniciais de um projeto têm impacto direto na qualidade, escalabilidade e capacidade de manutenção do produto final. Assim, investir em uma arquitetura sólida não é apenas uma escolha técnica, mas também estratégica, pois viabiliza o crescimento sustentável do software e reduz o custo de adaptação frente a mudanças tecnológicas e pedagógicas futuras.

No contexto de plataformas acadêmicas, que frequentemente sofrem com a descontinuidade, a aplicação rigorosa desses princípios é um diferencial estratégico. A abordagem adotada na StructLive, ao priorizar uma arquitetura extensível desde sua concepção, visa transformar o projeto de uma ferramenta isolada em uma infraestrutura educacional sustentável, capaz de se adaptar a novas demandas pedagógicas e tecnológicas sem a necessidade de refatorações custosas.

2.3. Arquiteturas Extensíveis

Uma arquitetura de software é considerada extensível quando permite a adição de novas funcionalidades com impacto mínimo sobre a base de código existente. uma arquitetura extensível caracteriza-se pela capacidade de adaptação controlada, em que novas funcionalidades podem ser incorporadas por meio de mecanismos previamente planejados, como interfaces bem definidas, pontos de extensão e padrões de abstração. Essa propriedade está diretamente relacionada à flexibilidade e à manutenibilidade do sistema, uma vez que reduz o risco de regressões e facilita a evolução incremental sem comprometer a integridade do software. Em síntese, a extensibilidade reflete o equilíbrio entre estabilidade estrutural e abertura à inovação (BASS; CLEMENTS; KAZMAN, 2013).

A extensibilidade é alcançada através de um projeto deliberado que define "pontos de extensão" e utiliza padrões de projeto, como Factory Method e Strategy, para desacoplar componentes (GAMMA et al., 1995). Aderir ao princípio Aberto/Fechado (aberto para extensão, fechado para modificação) de Martin (2003) é fundamental. A aplicação prática do conceito de extensibilidade demanda não apenas uma estrutura de código bem organizada, mas também decisões arquiteturais que antecipem a evolução do sistema. Isso implica projetar componentes com responsabilidades isoladas e interfaces claras, permitindo que novas funcionalidades sejam incorporadas sem comprometer as existentes.

Em projetos educacionais digitais, essa característica é particularmente relevante, pois o conteúdo e as demandas pedagógicas estão em constante transformação. Assim, uma arquitetura extensível garante que o sistema possa acompanhar inovações tecnológicas e metodológicas sem a necessidade de reescritas significativas, assegurando a sustentabilidade técnica e pedagógica da plataforma ao longo do tempo.

Na StructLive, a extensibilidade não é um mero detalhe técnico, mas o pilar central do projeto. A plataforma foi desenhada para que cada estrutura de dados seja um módulo autocontido, conectando-se ao núcleo do sistema por meio de interfaces bem definidas. Isso garante que a adição de um módulo de "Árvores AVL" no futuro não exija qualquer alteração no módulo já existente de "Listas Encadeadas", promovendo um crescimento orgânico e sustentável.

2.4. Plataformas Educacionais Interativas e Aprendizagem Visual

Plataformas educacionais interativas alteram a dinâmica do aprendizado, transformando o estudante de um receptor passivo em um agente ativo na construção do conhecimento (BONWELL; EISON, 1991). A integração de tecnologia digital com práticas pedagógicas, conforme defende Moran (2015), potencializa o engajamento e a autonomia. No ensino de programação, a aprendizagem visual é particularmente eficaz. Segundo a Teoria da Aprendizagem Multimídia de Mayer (2009), a apresentação de informações através de canais visuais e textuais simultaneamente reduz a carga cognitiva e facilita a criação de modelos

ISSN: 2447-0767

mentais robustos.

A adoção de recursos interativos e visuais no ensino de Computação amplia significativamente o potencial de aprendizagem, pois transforma a teoria em prática observável. Quando o estudante pode testar hipóteses, manipular elementos e visualizar os resultados de suas ações em tempo real, o processo de aprendizado torna-se mais ativo, exploratório e significativo. Esse tipo de interação estimula a autonomia intelectual e favorece a consolidação do raciocínio lógico, uma vez que o aluno compreende não apenas o resultado final, mas também o encadeamento das operações que o produz.

Recursos como simulações em tempo real e *feedback* automatizado são decisivos para uma aprendizagem eficaz (BOUD; MOLLOY, 2013). Ao permitir que o aluno escreva um código para inserir um nó em uma lista e veja imediatamente uma animação que representa essa operação, a StructLive conecta o simbólico (código) ao concreto (visualização), reforçando a compreensão de causa e efeito. Essa abordagem é análoga a exemplos de sucesso como o jogo The Farmer Was Replaced (2023), onde a programação se torna uma ferramenta para resolver problemas em um ambiente simulado e visual.

3. Metodologia

O desenvolvimento da StructLive seguiu uma abordagem metodológica estruturada, combinando um processo de desenvolvimento de software iterativo com uma seleção criteriosa de tecnologias para atender aos requisitos de interatividade, escalabilidade e manutenibilidade.

3.1 Processo de Desenvolvimento

Para a materialização do projeto, foram empregados os seguintes artefatos tecnológicos, escolhidos por sua modernidade, robustez e ecossistema de suporte, assim como demonstrado na Figura 1.

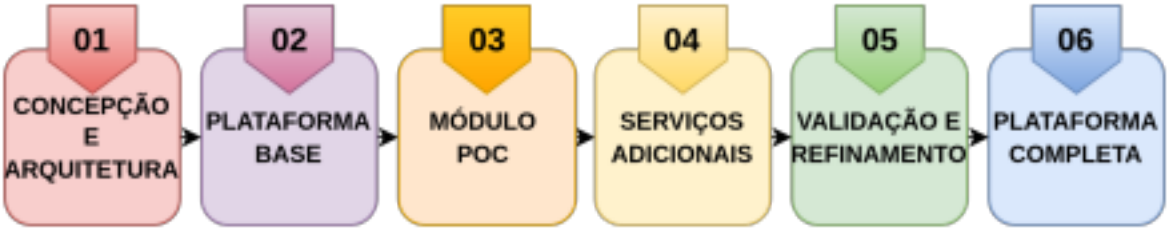


Figura 1. Processo de Desenvolvimento

1. Fase de Concepção e Arquitetura: Definição dos requisitos funcionais e não funcionais, com ênfase na extensibilidade. Nesta fase, foi projetada a arquitetura em camadas e modelada a estrutura base dos módulos educacionais e suas interfaces de comunicação.
2. Desenvolvimento da Plataforma Base: Implementação das funcionalidades centrais e transversais, como o sistema de autenticação de usuários, a navegação principal, a camada de persistência de dados e os componentes de UI reutilizáveis.
3. Implementação do Primeiro Módulo (Prova de Conceito): Desenvolvimento completo do módulo de listas simplesmente encadeadas para validar as decisões arquitetônicas,

testar as interfaces e servir como modelo para futuros módulos.

4. Integração de Serviços Adicionais: Acoplamento de serviços como o de Inteligência Artificial para feedback, demonstrando a capacidade de extensão da plataforma. 5. Validação e Refinamento: Execução de testes unitários e funcionais contínuos para garantir a estabilidade e a qualidade do código.

3.2 Tecnologias Utilizadas

Para a materialização do projeto, foram empregados os seguintes artefatos tecnológicos, escolhidos por sua modernidade, robustez e ecossistema de suporte:

- *Framework Next.js com React e TypeScript*: Escolhido para o desenvolvimento do frontend devido à sua capacidade de renderização híbrida (SSR e SSG), ecossistema robusto e à segurança de tipos fornecida pelo TypeScript, essencial para a manutenibilidade de projetos complexos.
- *Plataforma Supabase com banco de dados PostgreSQL:* Utilizada como solução de Backend-as-a-Service para gerenciar a autenticação, o armazenamento de dados em um banco PostgreSQL e a criação de APIs, acelerando o desenvolvimento sem sacrificar a escalabilidade
- *API do modelo de linguagem Google Gemini:* Integrada para prover um sistema de feedback adaptativo. A técnica de Geração Aumentada por Recuperação (RAG) é utilizada para contextualizar a IA com o problema do aluno, gerando dicas mais precisas
- *Bibliotecas Tailwind CSS e shadcn/ui:* Adotadas para a estilização da interface, permitindo a construção rápida de um design responsivo, consistente e acessível através de uma abordagem baseada em utilitários e componentes.
- *GitHub e Vitest:* O GitHub foi utilizado para o controle de versionamento do código fonte, facilitando a colaboração, enquanto o Vitest foi a ferramenta de escolha para a automação de testes unitários, garantindo a confiabilidade do código.

4. Resultados e Discussão

O primeiro passo no desenvolvimento da StructLive, seguindo a proposta de uma arquitetura extensível, foi a definição de um núcleo estável, responsável por funcionalidades comuns como autenticação (NextAuth), persistência de dados (Supabase) e navegação. Para viabilizar a inclusão incremental de funcionalidades, foram definidos pontos de extensão claros na camada de módulos educacionais, onde cada estrutura de dados segue um padrão pré
estabelecido de pastas e arquivos.

A plataforma adota contratos e interfaces, como o *ActivityContract*, que determina como cada atividade deve ser inicializada, renderizada e validada. Isso garante que módulos distintos sigam o mesmo padrão e possam se conectar ao sistema de feedback automatizado sem ajustes na base. Padrões de projeto como *Factory Method*, *Observer* e *Strategy* foram aplicados para
reforçar a extensibilidade, permitindo a criação de novas atividades e a variação entre tipos de estruturas (ex: lista simples e duplamente encadeada) de forma desacoplada

4.1 Arquitetura da Plataforma

ISSN: 2447-0767

O StructLive foi arquitetado em um modelo de camadas lógicas, promovendo a separação de responsabilidades e o baixo acoplamento entre os componentes do sistema.

- *Camada de Interface de Usuário (UI):* desenvolvida em React e Next.js, esta camada é responsável por toda a interação com o usuário. Ela contém os componentes visuais, as animações das estruturas de dados e o editor de código. A reatividade da interface garante que qualquer alteração no estado de uma estrutura, comandada pelo usuário, seja refletida visualmente em tempo real.
- *Camada de Módulos Educacionais:* o coração pedagógico da plataforma. Cada estrutura de dados (ex: Lista, Pilha) é implementada como um módulo independente que encapsula sua própria lógica de negócio, componentes de visualização específicos e atividades interativas. Eles se comunicam com o restante do sistema através de contratos de interface predefinidos.
- *Camada de Serviços da Aplicação:* contém a lógica de negócio transversal, como gerenciamento de sessão de usuário (NextAuth), comunicação com o backend e a orquestração de chamadas para serviços externos, como a API de IA.
- *Camada de Persistência:* abstraída pelo Supabase, é responsável por toda a comunicação com o banco de dados PostgreSQL, gerenciando o armazenamento de perfis de usuário, progresso em atividades e conteúdo das estruturas.
- *Camada de Integração com IA:* Um serviço desacoplado que recebe o contexto de uma atividade e o código do aluno, interage com a API do Google Gemini e retorna um feedback formatado para ser exibido na UI.

O fluxo de dados típico se inicia quando o usuário executa uma operação no editor de código. A UI captura essa ação, o módulo educacional correspondente processa a lógica da estrutura de dados, atualiza seu estado e, em seguida, a camada de UI renderiza novamente a visualização para refletir a mudança.

4.2 Estrutura Padrão dos Módulos

Para garantir a extensibilidade e a manutenibilidade, foi definida uma estrutura de diretórios e arquivos padrão para cada módulo educacional:

- *page.tsx*: atua como o ponto de entrada do módulo, apresentando o conteúdo teórico, a descrição das atividades e a navegação interna.
- *activity.tsx*: componente central que contém a lógica da atividade interativa, incluindo o editor de código, a área de visualização e o controle de estado da estrutura de dados. ● *rag_contexts.ts*: arquivo que armazena os contextos específicos (explicações, exemplos de erros comuns) a serem enviados para a IA (via RAG) para a geração de feedback contextualizado.
- *api/*: diretório contendo as rotas de API específicas do módulo, utilizadas para operações de backend como salvar o progresso de uma atividade.

Essa padronização assegura que a adição de um novo módulo (ex: Pilhas) seja um processo previsível e de baixo atrito para os desenvolvedores.

4.3 Implementação do Primeiro Módulo: Listas

A plataforma StructLive foi concebida para oferecer uma experiência de aprendizado imersiva e interativa sobre estruturas de dados. Sua página inicial, apresentada na Figura 2,

comunica claramente a proposta de valor do projeto: aprender de forma visual e prática, transformando conceitos teóricos em simulações dinâmicas.

Figura 2. Página inicial da plataforma StructLive, destacando a abordagem visual e prática

O módulo de listas simplesmente encadeadas foi o primeiro a ser completamente implementado, servindo como uma prova de conceito para a arquitetura da StructLive. Ele permite que o usuário realize operações de inserção, remoção e busca, visualizando cada passo da manipulação dos ponteiros e nós.

A interface de aprendizado, como exemplificado na Figura 3, é o coração da experiência. Nela, o usuário interage diretamente com a estrutura de dados. A tela é dividida de forma a facilitar a compreensão: à esquerda, o código-fonte em Python da operação selecionada (neste caso, remover_fim); no painel central, uma representação gráfica e animada da lista encadeada, onde cada nó e ponteiro é claramente visível; e, por fim, os controles de execução e um *log* que descreve cada passo realizado pelo algoritmo.

Figura 3. Interface de visualização da operação "Remover Fim" em uma lista simplesmente

ISSN: 2447-0767

encadeada.

A implementação bem-sucedida deste módulo validou os seguintes pontos: ● Eficácia da arquitetura: o modelo de módulos independentes se provou funcional, permitindo o desenvolvimento focado sem impactar o núcleo da plataforma, o que decorre do fato de que a implementação do primeiro módulo, dedicado às listas simplesmente encadeadas, pôde ser realizada de forma autônoma e isolada, sem necessidade de modificações no núcleo da plataforma. Isso demonstra que os contratos de interface e os pontos de extensão previamente definidos funcionaram conforme o planejado, garantindo baixo acoplamento entre as camadas do sistema. Além disso, o módulo foi integrado ao ambiente principal preservando a coerência funcional e visual da aplicação, evidenciando que a arquitetura proposta suporta a adição incremental de novos componentes sem comprometer a estabilidade global. Esse resultado confirma que o modelo modular concebido é eficaz tanto do ponto de vista da engenharia de software, pela manutenibilidade e reuso, quanto do ponto de vista pedagógico, ao permitir a rápida expansão do conteúdo educacional de forma sustentável. ● Validade dos contratos de interface: foi confirmada durante o processo de integração do módulo de listas simplesmente encadeadas aos serviços centrais da plataforma, como autenticação de usuários e persistência de dados. As interfaces previamente definidas garantiram uma comunicação estável e padronizada entre os componentes, permitindo que o módulo interagisse com o sistema principal sem dependências diretas do código interno de cada serviço. Esse comportamento comprova que os contratos foram corretamente especificados e implementados segundo os princípios de encapsulamento e baixo acoplamento, assegurando que futuras extensões ou substituições de serviços possam ocorrer sem impactos estruturais. Dessa forma, a arquitetura demonstrou confiabilidade, previsibilidade e reuso efetivo, atributos essenciais para plataformas de natureza modular e evolutiva como a StructLive.

- Viabilidade da visualização interativa: a combinação de React para o controle de estado e bibliotecas de renderização, como visto na prática na Figura 2, demonstrou ser performática e eficaz para criar uma experiência de aprendizado fluida e sem gargalos. Isso foi comprovado pela capacidade do sistema de representar, em tempo real, as transformações ocorridas nas estruturas de dados conforme as ações do usuário. A utilização do React para o gerenciamento de estado permitiu uma atualização eficiente da interface a cada operação, enquanto as bibliotecas de renderização gráfica adotadas garantiram animações suaves e responsivas, sem perdas de desempenho perceptíveis. Essa combinação tecnológica assegurou uma sincronização precisa entre o código executado e sua representação visual, elemento essencial para a compreensão conceitual das operações sobre listas encadeadas. Além do desempenho técnico, a experiência de uso evidenciou que a interação direta com os elementos visuais contribui para um aprendizado mais intuitivo e engajador, validando a proposta pedagógica da StructLive e confirmando a adequação da arquitetura para suportar visualizações dinâmicas em módulos futuros.

O desenvolvimento deste módulo confirmou que a arquitetura da StructLive é robusta e está preparada para escalar, acomodando futuras estruturas de dados, além de exercícios, testes e animações, com complexidade crescente.

5. Considerações Finais

Este trabalho descreveu o desenvolvimento da plataforma StructLive, uma solução educacional interativa que utiliza Next.js, React e TypeScript para o ensino de Estruturas de Dados de forma visual e modular. O projeto visou fornecer uma ferramenta escalável e de fácil manutenção, focada em superar as dificuldades de abstração no aprendizado de conceitos complexos da computação por meio de uma arquitetura extensível.

Durante o desenvolvimento, foram abordados desde o planejamento e modelagem da arquitetura do sistema até a implementação de uma interface de usuário reativa com componentes padronizados e a integração com serviços de backend via Supabase. A aplicação de princípios de Engenharia de Software e a definição de uma estrutura de módulos padronizada foram fundamentais para garantir a funcionalidade e a sustentabilidade da plataforma a longo prazo.

A implementação e os testes do primeiro módulo de listas simplesmente encadeadas demonstraram que a StructLive atende aos objetivos propostos, permitindo a adição de novos conteúdos de forma desacoplada e validando a eficácia da arquitetura. No entanto, foi identificado que, para uma avaliação completa do impacto pedagógico, é necessária a realização de testes de usabilidade com turmas de estudantes para coletar dados quantitativos e qualitativos sobre a experiência de aprendizado.

A plataforma oferece uma solução prática e moderna para o ensino de Estruturas de Dados, com a vantagem de ser projetada para evoluir continuamente. Futuramente, melhorias poderão ser implementadas, como a expansão para outras estruturas (pilhas, filas, árvores e grafos) e o desenvolvimento de um painel de acompanhamento para professores, refinando ainda mais as capacidades da StructLive para atender às necessidades específicas do ambiente acadêmico.

6. Referências

BASS, L.; CLEMENTS, P.; KAZMAN, R. Software Architecture in Practice. 3. ed. Addison Wesley Professional, 2013.

BONWELL, C. C.; EISON, J. A. Active Learning: Creating Excitement in the Classroom. ASHE-ERIC Higher Education Report No. 1, 1991.

BOUD, D.; MOLLOY, E. Feedback in Higher and Professional Education: Understanding It and Doing It Well. Routledge, 2013.

CORMEN, T. H. et al. Introduction to Algorithms. 3. ed. MIT Press, 2009.

GAMMA, E. et al. Design Patterns: Elements of Reusable Object-Oriented Software. Addison Wesley, 1995.

GOODRICH, M. T.; TAMASSIA, R.; GOLDWASSER, M. H. Data Structures and Algorithms in Java. 5. ed. John Wiley & Sons, 2011.

KRUEGER, C. W. Software Reuse. ACM Computing Surveys, v. 24, n. 2, p. 131-183, jun. 1992.

KNUTH, D. E. The Art of Computer Programming, Volume 1: Fundamental Algorithms. 3. ed. Addison-Wesley, 1997.

ISSN: 2447-0767

MARTIN, R. C. Agile Software Development, Principles, Patterns, and Practices. Prentice Hall, 2003.

MAYER, R. E. Multimedia Learning. 2. ed. Cambridge University Press, 2009.

MORAN, J. M. Metodologias Ativas para uma Aprendizagem Mais Profunda. Em: Bacich, L.; Moran, J. (Orgs.). Metodologias Ativas para uma Educação Inovadora. Penso, 2015.

PRENSKY, M. Digital Game-Based Learning. Paragon House, 2007.

PRESSMAN, R. S.; MAXIM, B. R. Engenharia de Software: Uma Abordagem Profissional. 8. ed. McGraw-Hill, 2016.

SEDGEWICK, R.; WAYNE, K. Algorithms. 4. ed. Addison-Wesley Professional, 2011.

SHAFFER, C. A. A practical introduction to data structures and algorithm analysis. Journal of Computing Sciences in Colleges, v. 16, n. 4, p. 88-96, 2001.

SOMMERVILLE, I. Engenharia de Software. 9. ed. Pearson Prentice Hall, 2011.

THE FARMER WAS REPLACED. Desenvolvido por Ad ad Studio, 2023. Disponível em: https://store.steampowered.com/app/2432020/The_Farmer_Was_Replaced/.

PROGRAMAÇÃO

Dia 03/11/2025

19h	Abertura
19h30	Palestra de abertura: **Jogos e carreira acadêmica: é possível?** Palestrante: **Leonardo Tórtoro Pereira – DOUTOR- IGCE.**
21h	Coffee-break

Dia 04/11/2025

19h Minicursos

Minicurso 1: Treinar e executar modelos de IA com javascript
Ministrante: Van Neves

Minicurso 2: Introdução ao desenvolvimento de jogos 3D na Unity 6
Ministrante: Stefan Lucas

Minicurso 3: CyberSecAI: Introdução a Cyber Segurança com auxílio de inteligência artificial contextualizada
Ministrantes: Mario Matheus Pombal Rebello e Marianne Lacerda Dutra e Lucas Casagrande

Minicurso 4: Aplicação FullStack com Vite(React) e FastAPI
Ministrantes: Davi Teixeira, Guilherme Domiciano, Nicole França

Minicurso 5: Vibe Coding 101: Introdução a IA para devs
Ministrantes: Alexandre Kavalerski

Minicurso 6: Testes Unitários com Java
Ministrantes: Iury Felipe

Minicurso 7: Utilização de DBeaver para gerenciamento de banco de dados relacionais
Ministrantes: Sidevalto Cipriano Capone

Minicurso 8: Agentes de IA na Prática com Langchain
Ministrantes: Luis Fernando e Geisbelly

Minicurso 9: Alfabetização em segurança da informação: Construa sua defesa
Ministrantes: Yasmin

21h Coffee-break

Dia 05/11/2025

19h Minicursos

Minicurso 1: Treinar e executar modelos de IA com javascript
Ministrante: Van Neves

ISSN: 2447-0767

	Minicurso 2: Introdução ao desenvolvimento de jogos 3D na Unity 6 Ministrante: Stefan Lucas **Minicurso 3: Desenvolvimento Full Stack com NextJS** Ministrante: Mario Matheus Pombal Rebello, Henrique Crispin de Aguiar, Marianne Lacerda Dutra **Minicurso 4: CyberSecAI: Introdução a Cyber Segurança com auxílio de inteligência artificial contextualizada** Ministrantes: Mario Matheus Pombal Rebello e Marianne Lacerda Dutra e Lucas Casagrande **Minicurso 5: Aplicação FullStack com Vite(React) e FastAPI** Ministrantes: Davi Teixeira, Guilherme Domiciano, Nicole França **Minicurso 6: Vibe Coding 101: Introdução a IA para devs** Ministrantes: Alexandre Kavalerski **Minicurso 7: Testes Unitários com Java** Ministrantes: Iury Felipe **Minicurso 8: Utilização de DBeaver para gerenciamento de banco de dados relacionais** Ministrantes: Sidevalto Cipriano Capone **Minicurso 9: Agentes de IA na Prática com Langchain** Ministrantes: Luis Fernando e Geisbelly **Minicurso 10: Alfabetização em segurança da informação: Construa sua defesa** Ministrantes: Yasmin
21h	Coffee-break

Dia 06/11/2025

19h – 22h	Sessões Técnicas
19h	**StudyFlow–Voice: Respostas por Voz com Whisper e Avaliação Semântica Inicial via IA em Cards de Estudo** Carlos Aleixo, Jackson Gomes de Souza
19h13	**Análise Empírica do Impacto da Privacidade Diferencial na Eficiência de Modelos de Redes Neurais Profundas** Luís Fernando Borges Lima, Carlos Eduardo Ribeiro, Paulo Miguel, Fabio Araujo
19h26	**Utilização de Modelos de Linguagem de Grande Escala (LLMs) para Resumo Automático de Informações em Procedimentos Extrajudiciais** Helyezer Teofilo, Fabio Araujo
19h39	**PyJourney: Jogo Educativo para Aprendizagem da Linguagem de Programação Python** Mario Rebello, Fernanda Pereira Gomes

19h52	**Aplicação de DeepFace e OpenFace para Identificação de Sentimentos Básicos em Vídeos de Teleconsulta** Aurea Nascimento, Fabio Araujo
20h05	**BIOS — O Código da Vida: Implementação Web Nativa de um Jogo de Desenvolvimento do Pensamento Computacional** Lucas Anselmo, Henrique Dias Silva, Jackson Gomes de Souza
20h18	**Assistente Generativo para Terapeutas baseado em Arquitetura RAG** Anne Oliveira, Jackson Gomes de Souza
20h31	**Ambientes Interativos para Aprendizagem de Máquina: Potencialidades e Limitações de Jupyter Lab e Google Colab** Martony Demes Silva
20h44	**Arquitetura Actor-Critic com Memória Causal para Agentes Baseados em Modelos de Linguagem** Lucas Vinicius Oliveira Cardoso, Marianne Theodoro, Parcilene Fernandes de Brito
20h57	**DaeLink: Job opportunity for people with special needs** Alex Santos, Danilo Santos Soares, Endrigo, Adreza Maria de Souza Rocha, Jeferson Roberto de Lima
21h10	**Desenvolvimento de um Chatbot para Anamnese Psicológica em Sistema de Atendimento Online Utilizando o Framework Rasa** Maria Fernanda Rocha, Fabio Castro Araujo
21h23	**StructLive: desenvolvimento de uma plataforma extensível para o ensino de Estruturas de Dados** Raphael Henrique Scheffler Ferreira, Fabiano Fagundes, Jackson Gomes de Souza
21h30	Coffee-break

Dia 07/11/2025

19h	Abertura
19h30	**Mesa redonda com egressos dos cursos do departamento de computação** Participantes: **Ricardo Almeida, Darley Passarin, Alexandre Kavalerski**
20h30	Encerramento
21h	Coffee-break

ISSN: 2447-0767

ANAIS DO XXVII CONGRESSO DE COMPUTAÇÃO E TECNOLOGIAS DA INFORMAÇÃO

ISSN 2447-0767

03 A 07 DE NOVEMBRO DE 2025

https://ulbra-to.br/encoinfo

REALIZAÇÃO

SISTEMAS DE INFORMAÇÃO

CIÊNCIA DA COMPUTAÇÃO

ENGENHARIA DE SOFTWARE

APOIO

ULBRA PALMAS

SBC Sociedade Brasileira de Computação