

Desenvolvimento de um aplicativo móvel com banco de questões com foco no Exame Nacional de Desempenho dos Estudantes (ENADE)

Aluno: João Vitor Oliveira - joaovitoroliveiradeveloper@gmail.com

Orientador: Prof. Me. Márcio Lopes Júnior - profmarciolopes@unifmg.edu.br

Resumo

O Exame Nacional de Desempenho dos Estudantes (ENADE) é um importante instrumento de avaliação da qualidade dos cursos de graduação no Brasil. Contudo, observa-se a ausência de ferramentas digitais voltadas especificamente à preparação para o exame, o que dificulta a prática e o acompanhamento do aprendizado. Este trabalho apresenta o desenvolvimento do **TrainUp**, um aplicativo móvel multiplataforma desenvolvido com foco no ENADE. O objetivo principal é auxiliar estudantes no estudo para o prova por meio da realização de simulados e do acompanhamento do desempenho individual. O aplicativo foi desenvolvido com o *framework Flutter*, integrado a um *backend* em *Node.js* com *Fastify*, *PostgreSQL* e *Prisma ORM*. Como diferencial, o projeto implementa um microserviço em *Python* capaz de extrair questões de provas anteriores, processá-las e armazená-las no banco de dados, promovendo um cadastro ágil. Espera-se que a solução proposta contribua para o aprimoramento do processo de ensino-aprendizagem, oferecendo um software moderno, escalável e acessível para o estudo direcionado ao ENADE.

Palavras-chave: Aplicativo móvel; ENADE; Ensino Superior; Simulados.

Abstract

The National Student Performance Exam (ENADE) is an important instrument for assessing the quality of undergraduate courses in Brazil. However, there is a noticeable lack of digital tools specifically designed to prepare students for the exam, which hinders practice and the monitoring of learning progress. This work presents the development of **TrainUp**, a mobile cross-platform built with a focus on ENADE exam preparation. The main goal is to assist students in their studies through the execution of mock tests and monitoring of individual performance. The application was developed using the Flutter framework, integrated with a *Node.js* backend employing *Fastify*, *PostgreSQL*, and *Prisma ORM*. As a distinctive feature, the project includes a *Python* microservice capable of extracting questions from prior exams, process and store it in database, enabling faster registration. The proposed solution is expected to contribute to the improvement of the teaching and learning process by providing a modern, scalable, and accessible software for ENADE-oriented study.

Keywords: Mobile application; ENADE; Higher Education; Mock exams.

Sumário

1 Introdução

2

2	Referencial Teórico	3
2.1	O ENADE	3
2.2	O uso de IA nos simulados	4
2.3	Arquitetura de software e microserviços	5
2.4	Segurança e privacidade de dados na LGPD	6
3	Estado da Arte	7
3.1	Plataformas Existentes	7
3.2	Aplicativos Móveis para Aprendizagem	7
3.3	Diferenciais do <i>TrainUp</i>	8
4	Metodologia	8
4.1	Etapas de Desenvolvimento	8
4.2	Arquitetura do Sistema	9
4.3	Modelagem de Dados	11
4.4	Ferramentas e Ambiente de Desenvolvimento	12
4.5	Correção automática de questões discursivas com IA	13
4.6	Síntese do Capítulo	14
5	Considerações Finais	14
6	Trabalhos Futuros	15
	Agradecimentos	16
	Lista de Figuras	
1	Arquitetura do sistema <i>TrainUp</i> . Fonte: o autor.	9
2	Pipeline de upload e processamento: PDF de questões + XLSX de respostas. Fonte: o autor.	11
3	Diagrama Entidade–Relacionamento (DER) do banco de dados do <i>TrainUp</i> . Fonte: elaborado pelo autor a partir do DBeaver (2025).	12

1. Introdução

O Exame Nacional de Desempenho dos Estudantes (ENADE) é uma avaliação aplicada pelo Ministério da Educação (MEC) aos estudantes concluintes do ensino superior, com o propósito de medir a qualidade dos cursos de graduação a partir dos resultados obtidos. O exame integra o Sistema Nacional de Avaliação da Educação Superior (SINAES) e atribui notas de 1 a 5 às instituições, considerando conteúdos programáticos, habilidades e competências desenvolvidas ao longo da formação acadêmica.

Nesse contexto, observa-se que o desempenho no ENADE influencia diretamente a reputação e os indicadores de qualidade das universidades, impactando não só a gestão institucional mas também a percepção pública por parte da sociedade. Logo, é fundamental a existência de ferramentas que auxiliem a preparação dos estudantes e possibilitem o acompanhamento de seu desempenho, contribuindo para uma aprendizagem eficiente e orientada a resultados.

Visando atender a essa necessidade, o presente trabalho apresenta o desenvolvimento do *TrainUp*, um aplicativo móvel multiplataforma desenvolvido com foco no ENADE. A

solução abrange todo o ciclo de estudo, desde o cadastro e categorização de questões de provas anteriores até a criação, execução e correção de simulados, com suporte a questões objetivas e discursivas. As questões podem ser inseridas manualmente ou obtidas de forma automatizada a partir de arquivos PDF de provas oficiais, cuja extração e validação são realizadas por meio de técnicas de processamento de linguagem natural e de imagens, sendo as respostas associadas obtidas a partir de planilhas XLSX.

Além disso, o aplicativo apresenta ao usuário uma visão do seu desempenho, como as relações entre acertos, erros e tempo médio de resolução. O software foi desenvolvido com o framework *Flutter*, integrado a um *backend* em *Node.js* utilizando *Fastify*, *PostgreSQL* e *Prisma ORM*. O sistema também conta com um microsserviço em *Python* responsável pelo processamento de PDF e XLSX para extração de questões, e com o uso de inteligência artificial para correção de respostas discursivas.

Este artigo está organizado da seguinte forma: a Seção 2 apresenta o Referencial Teórico, abordando os conceitos e fundamentos que sustentam o desenvolvimento do sistema. A Seção 3 descreve o Estado da Arte, apresentando soluções parecidas e destacando os diferenciais do *TrainUp*. A Seção 4 aborda sobre a Metodologia, detalhando as etapas, ferramentas e tecnologias utilizadas na implementação. A Seção 5 reúne as Considerações Finais, discutindo os resultados esperados e contribuições do projeto. Por fim, a Seção 6 propõe Trabalhos Futuros, indicando possibilidades de aprimoramento do software.

2. Referencial Teórico

Esta seção apresenta os principais conceitos que fundamentam o desenvolvimento do software, abordando aspectos relacionados ao Exame Nacional de Desempenho dos Estudantes, à aplicação de tecnologias digitais no processo educacional e às tecnologias utilizadas na implementação do sistema.

2.1. O ENADE

A avaliação educacional é parte do processo de ensino e aprendizagem, funcionando não apenas como instrumento de medição, mas também como ferramenta de regulação e aprimoramento contínuo. Nesse sentido, o Exame Nacional de Desempenho dos Estudantes, integrante do Sistema Nacional de Avaliação da Educação Superior (SINAES), possibilita a análise global da qualidade dos cursos de graduação e a proposição de políticas públicas voltadas ao aperfeiçoamento da educação superior. De acordo com Costa Souza (2020), "ao combinar os resultados do exame com os indicadores de qualidade, é possível avaliar a qualidade dos cursos de graduação em larga escala, o que favorece a promoção de ações que fomentam melhorias na gestão da educação superior no Brasil".

Isso mostra o caráter formativo e diagnóstico da avaliação, cuja função vai além da certificação de resultados. Ainda segundo Costa Souza (2020), o ENADE fornece dados que, junto às avaliações internas das instituições, permitem propor ações pedagógicas e administrativas voltadas à melhoria de cursos, metodologias e currículos. A autora reforça que a participação de professores e alunos no processo avaliativo fortalece a cultura da autoavaliação e da corresponsabilidade pela qualidade do ensino.

Do ponto de vista cognitivo, pesquisas mostram que processos de avaliação frequente e a prática de recuperação de informações, conhecida como *testing effect*, favorecem a retenção de longo prazo e melhoram o desempenho em avaliações posteriores (Oliveira and Stein (2018)).

Estudos revisados pelas autoras indicam que o ato de repetir exercícios e avaliações consolida a memória de forma mais eficiente do que simplesmente reler conteúdos.

Além disso, quando a prática é acompanhada de *feedback* adequado, há maior autorregulação da aprendizagem e engajamento dos estudantes. O *feedback* permite identificar pontos de melhoria, ajustar métodos e transformar erros em oportunidades de aprendizagem, reforçando o caráter formativo da avaliação (Oliveira and Stein (2018)).

Assim, pode-se afirmar que tanto nas avaliações institucionais em larga escala, como o ENADE, quanto nos simulados usados para estudo, o principal objetivo da avaliação é oferecer um retorno aos participantes e estimular a reflexão sobre o próprio aprendizado. Essa necessidade reforça que a criação de ferramentas digitais que permitem aos alunos praticar, receber *feedback* e acompanhar seu progresso, pode facilitar o processo de aprendizagem.

2.2. O uso de IA nos simulados

A Inteligência Artificial (IA) é uma área da computação que busca reproduzir processos cognitivos humanos em sistemas computacionais. Segundo Azambuja and Silva (2024), a IA pode ser classificada em três níveis: fraca, forte e superinteligente, sendo a primeira a mais presente no cotidiano, aplicada em assistentes virtuais, sistemas de recomendação e reconhecimento de voz. No cenário educacional, destaca-se a IA generativa, principalmente os modelos de linguagem de grande porte (*Large Language Models* - LLMs), capazes de interpretar e produzir linguagem natural de forma coerente e contextualizada, permitindo a criação de funcionalidades como aplicações com correção automática de respostas discursivas e a geração de *feedbacks*.

De acordo com Lee (apud Azambuja and Silva (2024)), a IA é capaz de corrigir erros, responder dúvidas, atribuir e avaliar tarefas, além de fornecer *feedback* imediato, características ideais para o aprendizado personalizado. Essas características da IA permitem criar sistemas automáticos de correção de questões discursivas, capazes de simular o comportamento de professores ao oferecer um retorno adaptado à resposta do estudante. Os sistemas baseados em IA podem gerar comentários com orientações, promovendo o aprendizado autônomo do aluno.

Porém, o uso da IA em processos avaliativos levanta desafios éticos importantes. Azambuja and Silva (2024) destacam que os algoritmos podem reproduzir vieses presentes nos dados de treinamento. Logo, faz-se necessária a supervisão humana periódica para garantir que a devolutiva está dentro do esperado.

Além dos aspectos técnicos, Azambuja and Silva (2024) enfatizam a importância do papel docente na mediação ética. Mesmo diante do crescimento da automação, o professor é essencial, servindo como orientador do processo educativo, responsável por direcionar os estudantes na interpretação dos resultados gerados por sistemas inteligentes. Assim, cabe dizer que a correção automatizada não substitui a avaliação humana, mas certamente a complementa e a supre em situações que envolvem grande volume de respostas.

Por fim, pode-se afirmar que a aplicação de IA na avaliação educacional, com foco na correção de respostas discursivas, representa uma convergência entre avanços tecnológicos e princípios pedagógicos. Os modelos de linguagem de larga escala oferecem oportunidades para a personalização e o *feedback* imediato, mas também demandam uma abordagem crítica quanto aos vieses, à transparência e à responsabilidade ética de seu uso. A calibração e supervisão por avaliadores humanos continua sendo essencial para assegurar confiabilidade nas avaliações automatizadas.

2.3. Arquitetura de software e microsserviços

A arquitetura de software é um dos principais pontos a serem observados na organização de sistemas computacionais. Ela determina como seus componentes interagem e se estruturam. De acordo com Bosch (apud Conceição and Pinto (2021)), a arquitetura influencia diretamente o desempenho, a robustez, a capacidade de distribuição e a manutenção de um sistema, constituindo-se como um elemento determinante para o sucesso do projeto. Conceição and Pinto (2021) destacam que diferentes modelos arquiteturais oferecem formas distintas de organizar funcionalidades, sendo as arquiteturas monolíticas e de microsserviços duas das mais utilizadas no desenvolvimento de aplicações modernas.

A arquitetura monolítica caracteriza-se pela construção de todo o sistema como um único bloco de software, no qual os componentes estão integrados em uma única aplicação, sendo compilados, implantados e executados de forma conjunta. Wigmore (2016 apud Conceição and Pinto (2021)) explica que, nesse modelo, a atualização de qualquer módulo exige a recompilação e implantação de toda a aplicação. Ainda assim, sistemas monolíticos podem apresentar vantagens, como maior simplicidade de desenvolvimento, facilidade de testes e implantação unificada (Newman, 2020 apud Conceição and Pinto (2021)). Contudo, à medida que o sistema cresce, esse modelo tende a se tornar difícil de compreender e manter, limitando a evolução do software e aumentando os riscos associados a alterações (Richardson, 2014 apud Conceição and Pinto (2021)).

Por outro lado, a arquitetura de microsserviços propõe a separação da aplicação em serviços independentes, cada um responsável por uma funcionalidade específica. Segundo Conceição and Pinto (2021), cada microsserviço pode ser desenvolvido, implantado e escalado de forma autônoma, utilizando as tecnologias mais adequadas ao seu domínio. Atkisson (2017 apud Conceição and Pinto (2021)) define os microsserviços como pequenos componentes distribuídos que se comunicam por meio de protocolos leves, como HTTP, formando um ecossistema modular capaz de se adaptar rapidamente às mudanças de requisitos.

As vantagens dessa abordagem incluem a maior produtividade das equipes, a possibilidade de escalabilidade individual dos serviços, menos propensão a falhas e flexibilidade tecnológica (Ivory, 2021 apud Conceição and Pinto (2021)). Como apontam Conceição and Pinto (2021), a independência dos módulos permite que falhas localizadas não comprometam o funcionamento de toda a aplicação, contribuindo para maior confiabilidade do sistema. Esse modelo favorece também a computação paralela e a implantação contínua, acelerando o ciclo de desenvolvimento.

No entanto, a adoção de microsserviços também apresenta desafios significativos. Entre eles, destacam-se a necessidade de testes mais complexos, depuração mais complexa, aumento da latência devido à comunicação entre serviços e exigência de monitoramento e registro de *logs* de maneira centralizada (Atkisson, 2017 apud Conceição and Pinto (2021)). Conceição and Pinto (2021) enfatizam que a migração para microsserviços envolve não apenas mudanças técnicas, mas também transformações organizacionais, exigindo equipes capacitadas e processos de desenvolvimento bem estabelecidos para evitar aumento excessivo e desnecessário da complexidade.

Nesse contexto, a arquitetura de microsserviços tem sido amplamente adotada em aplicações modernas que demandam alta escalabilidade, resiliência e modularidade, especialmente quando integradas com tecnologias de mensageria e comunicação assíncrona, como o uso de filas. Assim, a escolha entre uma abordagem monolítica ou baseada em microsserviços deve considerar fatores como volume de requisições, necessidades de evolução, complexidade do domínio e capacidade organizacional de gerir sistemas distribuídos.

2.4. Segurança e privacidade de dados na LGPD

A segurança da informação é indispensável para garantir a proteção de dados pessoais e o funcionamento confiável de sistemas computacionais. De acordo com Carvalho et al. (2024), a segurança da informação envolve a adoção de práticas, políticas e controles voltados à preservação dos ativos informacionais, garantindo a mitigação de riscos e a continuidade das operações. Os autores afirmam que a crescente digitalização de serviços e a popularização de dispositivos móveis aumentaram significativamente a exposição dos usuários a ameaças cibernéticas, tornando indispensável a implementação de mecanismos de proteção robustos.

Segundo a ABNT NBR ISO 27002:2013 (apud Carvalho et al. (2024)), a segurança da informação baseia-se em três pilares: confidencialidade, integridade e disponibilidade. A confidencialidade assegura que apenas indivíduos autorizados tenham acesso a determinados dados; a integridade garante que as informações não sejam alteradas indevidamente; e a disponibilidade assegura que os dados estejam acessíveis quando necessários. Esses princípios orientam o desenvolvimento de aplicações que lidam com informações pessoais, especialmente em contextos que envolvem autenticação, armazenamento e comunicação em rede.

Com o aumento dos crimes cibernéticos ao longo da última década, principalmente durante a pandemia de Covid-19, período no qual houve aumento de 175% nas fraudes digitais, segundo levantamento do G1 (GLOBO, 2022 apud Carvalho et al. (2024)), a necessidade de políticas efetivas de segurança tornou-se ainda mais evidente. Ataques como *phishing*, engenharia social, *keyloggers* e *backdoors* representam riscos constantes aos sistemas de informação, exigindo atenção dos desenvolvedores e gestores de tecnologia.

Nesse cenário, a Lei Geral de Proteção de Dados (LGPD), instituída pela Lei nº 13.709/2018, define diretrizes para o tratamento de dados pessoais no Brasil. Conforme Carvalho et al. (2024), a LGPD visa proteger os direitos fundamentais de liberdade, privacidade e livre desenvolvimento da personalidade, reconhecendo a assimetria existente entre titulares e organizações, reforçando a necessidade de proteger os usuários quanto ao uso indevido de seus dados em um ambiente marcado pela intensificação das tecnologias digitais.

Carvalho et al. (2024) enfatizam que a conformidade com a LGPD requer a adoção de medidas técnicas e organizacionais capazes de assegurar a confidencialidade, integridade e proteção dos dados tratados. Entre essas medidas, destacam-se o consentimento explícito do titular, a transparência sobre as finalidades do tratamento, a prevenção contra acessos indevidos e a notificação obrigatória de incidentes de segurança. As organizações também devem implementar mecanismos de governança, auditorias regulares e capacitação de equipes, reforçando uma cultura de proteção de dados.

A LGPD classifica os dados pessoais em categorias como dados sensíveis, dados anonimizados e dados pseudonimizados, cada um demandando níveis específicos de proteção. Além disso, estabelece direitos fundamentais ao titular, como acesso, correção, portabilidade, anonimização e eliminação de dados, bem como informação sobre compartilhamento e possibilidade de revogação do consentimento.

Para sistemas digitais, como plataformas móveis, Carvalho et al. (2024) citam práticas essenciais para garantir segurança e boa experiência do usuário, incluindo autenticação forte (como múltiplos fatores), proteção contra engenharia social, comunicação segura, criptografia de dados armazenados e transitados, além do fornecimento de informações claras sobre privacidade e permissões solicitadas pelo aplicativo.

Dessa forma, a segurança e privacidade de dados constituem aspectos indispensáveis no desenvolvimento de soluções digitais modernas, especialmente aquelas que fazem uso de dados pessoais sensíveis ou realizam autenticação de usuários. A implementação de práticas

alinhadas à LGPD não apenas garante conformidade legal, mas também contribui para a confiança, a usabilidade e a proteção dos usuários em ambientes digitais cada vez mais complexos e suscetíveis a riscos.

3. Estado da Arte

Esta seção visa apresentar soluções e abordagens existentes relacionadas à preparação para exames de larga escala, simulados digitais e correção automática, destacando as lacunas que motivaram o desenvolvimento do *TrainUp*. A análise conta com plataformas bastante utilizadas no Brasil, aplicativos móveis de aprendizagem e serviços de correção automática, seguida de uma comparação sistemática e dos diferenciais da solução proposta.

3.1. Plataformas Existentes

No Brasil, observa-se a predominância de plataformas voltadas ao ENEM e vestibulares, com forte ênfase em bancos de questões, simulados e relatórios de desempenho. A Estuda.com, por exemplo, disponibiliza um repositório de itens, simulados e estatísticas de desempenho, com funcionalidades como correção por cartão-resposta e recursos de estudo adaptativos. Já a Geekie atua com avaliações e simulados preparatórios, oferecendo matrizes de avaliação, relatórios e análises que apoiam escolas e estudantes na preparação para o exame. Apesar de consolidadas, essas plataformas não são orientadas especificamente ao ENADE e, quando oferecem correção automática, esta concentra-se em *redações* (geralmente em língua portuguesa padrão) ou em métricas objetivas, não em respostas discursivas técnico-científicas por área de formação.

Em paralelo, há serviços internacionais com *feedback* automatizado, como o *Write & Improve*, desenvolvido pela Universidade de Cambridge, que fornece devolutivas instantâneas vinculadas ao *Common European Framework of Reference* (CEFR) para aprendizagem de inglês. Tais soluções ilustram o estado da técnica em correção automática de produções textuais, embora não cubram o escopo disciplinar e linguístico exigido por avaliações em português com foco em áreas específicas do conhecimento, como é o caso do ENADE.

Por fim, cabe citar que as fontes oficiais de provas e gabaritos do ENADE são publicadas pelo Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP) em seus portais, constituindo o principal acervo *ground truth* para estudos e desenvolvimento de soluções de prática e análise. No entanto, esse material é fornecido em formatos estáticos (PDFs e cadernos), sem um repositório estruturado de questões, o que limita seu uso direto em aplicações digitais sem um processo prévio de extração e normalização.

3.2. Aplicativos Móveis para Aprendizagem

Aplicativos educacionais atuais priorizam experiências de estudo contínuas, com simulações de prova, análises de desempenho e algum grau de personalização. Em geral, tais soluções organizam conteúdo por trilhas, oferecem simulados periódicos e relatórios sintéticos, e, em alguns casos, integram *feedback* automatizado para produções escritas. Entretanto, é menos frequente a presença de um fluxo ponta a ponta para exames específicos como o ENADE, combinando: (i) ingestão/extração automática de questões a partir de PDFs oficiais e XLSX de respostas, (ii) correção automática de questões discursivas e (iii) análise de desempenho alinhados às competências do exame.

3.3. Diferenciais do *TrainUp*

O *TrainUp* surge para preencher lacunas identificadas na literatura e no mercado ao:

- Focar explicitamente o ENADE, usando provas e gabaritos oficiais como fonte primária de itens;
- Implementar um pipeline de extração (PDF + XLSX $\rightarrow$ questões) com normalização de enunciados, alternativas e metadados, reduzindo o custo de curadoria;
- Oferecer simulados com itens objetivos e discursivos, com correção automática de discursivas baseada em LLMs e rubricas;
- Unificar uma camada móvel (app) com recursos de estudo e acompanhamento, mantendo *feedbacks* rápidos e histórico de desempenho.

4. Metodologia

Esta seção descreve o processo de desenvolvimento do sistema, detalhando as etapas executadas, a arquitetura adotada, os modelos de dados utilizados e as principais ferramentas que compõem o ambiente de implementação. A metodologia integra práticas de engenharia de software, princípios arquiteturais e tecnologias modernas voltadas ao desenvolvimento de aplicações móveis e distribuídas.

4.1. Etapas de Desenvolvimento

O desenvolvimento do *TrainUp* foi conduzido de forma incremental e iterativa, seguindo um processo dividido em etapas sucessivas. Inicialmente, realizou-se o levantamento dos requisitos funcionais e não funcionais, com foco na análise das necessidades do usuário e das limitações presentes em soluções existentes. Em seguida, foi elaborada a arquitetura do sistema, contemplando a divisão entre aplicativo móvel, *backend* monolítico e microserviço de extração de questões.

Na etapa de implementação, o aplicativo foi desenvolvido em *Flutter*, garantindo suporte multiplataforma e integração com o *backend* por meio do *Dio*, uma biblioteca que lida com o fluxo de requisições HTTP. Paralelamente, o *backend* em *Node.js* foi responsável pela camada de negócios, autenticação, gerenciamento de usuários e armazenamento de dados. O microserviço em Python foi implementado em fluxo independente, consumindo mensagens provenientes do RabbitMQ e realizando a extração e processamento automático de questões a partir de arquivos PDF e XLSX.

Por fim, as etapas de testes, validação e otimização incluíram ajustes de interface e verificação de consistência dos dados.

4.2. Arquitetura do Sistema

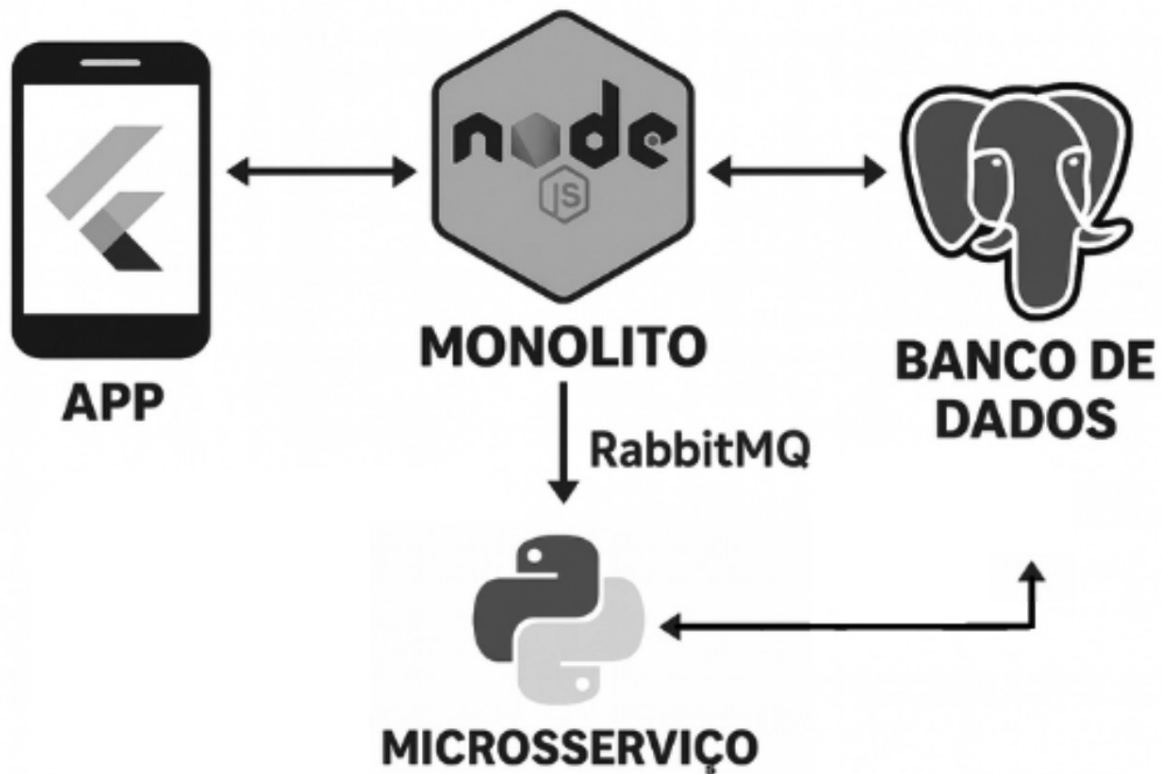


Figura 1: Arquitetura do sistema *TrainUp*. Fonte: o autor.

A arquitetura do sistema foi estruturada em três componentes principais:

Aplicativo móvel

O aplicativo móvel foi desenvolvido com *Flutter*, um *framework* multiplataforma amplamente adotado para a criação de aplicações nativas a partir de uma única base de código. Para o gerenciamento de estado e controle do fluxo da aplicação, utiliza-se o *GetX*, um padrão que integra gerenciamento de estado reativo, injeção de dependências e navegação. Os dados são armazenados localmente por meio do *IsarDB*, um banco de dados local orientado a objetos e otimizado para alto desempenho, o que possibilita o funcionamento parcial da aplicação em modo *offline*. A comunicação com o *backend* ocorre por meio de requisições HTTP realizadas com a biblioteca *Dio*, empregando autenticação baseada em *JSON Web Tokens* (JWT) para garantir a segurança das operações.

Backend monolítico

O *backend* foi implementado em *Node.js*, uma ferramenta consolidada para o desenvolvimento de aplicações servidoras, utilizando o *Fastify* na construção de APIs HTTP. A aplicação foi estruturada como um monólito, de modo a simplificar a orquestração das rotinas de negócio e o ciclo de desenvolvimento e implantação. O acesso ao banco de dados PostgreSQL é realizado por meio do *Prisma ORM*, que provê uma camada de abstração para manipulação de dados e garantia de integridade relacional. Para o envio de notificações por *e-mail*,

emprega-se a biblioteca *Nodemailer*. Além disso, o *backend* é responsável pela comunicação com o *RabbitMQ*, um sistema de mensageria assíncrona, enviando mensagens a uma fila que aciona o microsserviço dedicado à extração e processamento de dados.

As funcionalidades centrais incluem:

- autenticação e autorização via token JWT;
- CRUD de usuários, questões, simulados, gestão acadêmica;
- correção de questões discursivas via API da OpenAI e API do Gemini para contingência;
- integração com o Amazon S3 para leitura e armazenamento de arquivos.

Microsserviço de extração

O microsserviço de extração foi desenvolvido em *Python* e não expõe rotas REST, realizando toda a sua comunicação de forma assíncrona por meio do *RabbitMQ*. Ao receber uma notificação proveniente da fila de mensagens, o serviço inicia o processamento consultando o banco de dados para recuperar o registro associado, obtendo as URLs dos arquivos vinculados e realizando o download deles a partir do *Amazon S3*. Durante essa etapa, o estado do registro é atualizado de modo a refletir o progresso do processamento no aplicativo.

Em seguida, o serviço executa a extração de texto estruturado a partir dos documentos PDF utilizando a biblioteca *PyMuPDF*, adequada para o processamento desse tipo de arquivo. Paralelamente, imagens e trechos gráficos presentes nos documentos são tratados por meio da biblioteca *Pillow*, sendo posteriormente enviados ao *Amazon S3* para armazenamento e referência. No caso das planilhas de respostas em formato XLSX, o conteúdo é interpretado e validado pela biblioteca *openpyxl* de acordo com a estrutura esperada, permitindo a associação das respostas às respectivas questões. Após essas etapas, o conteúdo obtido passa por uma validação inicial e por um processo de estruturação, visando garantir a consistência e a organização dos dados.

Na etapa subsequente, o conteúdo textual extraído é submetido a um processo de normalização semântica por meio de modelos de inteligência artificial, utilizando prioritariamente a API da *OpenAI* e, de forma contingencial, a API do *Google Gemini*, com o objetivo de padronizar as informações e torná-las adequadas para uso posterior na aplicação. Por fim, as informações resultantes do processamento, provenientes tanto dos documentos quanto das planilhas, são persistidas no banco de dados *PostgreSQL*, utilizando a biblioteca *psycopg* para a comunicação com o sistema gerenciador de banco de dados.

Ressalta-se que o microsserviço emprega bibliotecas auxiliares, como o *boto3* para a comunicação com o *Amazon S3* e o *httpx* para a realização de chamadas HTTP externas, compondo a infraestrutura necessária para a integração entre os diferentes serviços e recursos utilizados no processo de extração.

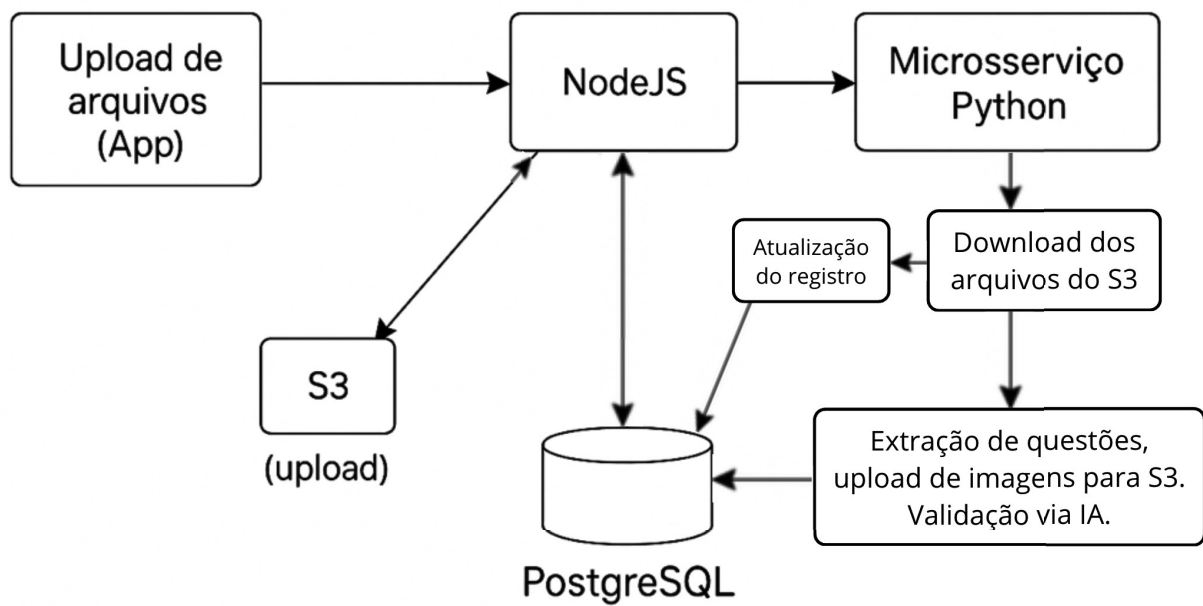


Figura 2: Pipeline de upload e processamento: PDF de questões + XLSX de respostas. *Fonte: o autor.*

4.3. Modelagem de Dados

A modelagem de dados do sistema foi projetada para organizar informações referentes a usuários, disciplinas, cursos, provas, questões, simulados e resultados, e uploads de prova realizados. O SGBD PostgreSQL foi adotado por sua estabilidade, suporte a dados relacionais e capacidade de escalar em ambientes distribuídos. Entre todas as entidades, destacam-se:

- **Usuário:** informações de acesso, perfil e histórico;
- **Questão:** enunciado, alternativas, categoria e tipo (objetiva ou discursiva);
- **Simulado:** conjunto de questões que compõem um simulado.
- **Resultado:** acertos, erros, tempo de execução e correções discursivas.

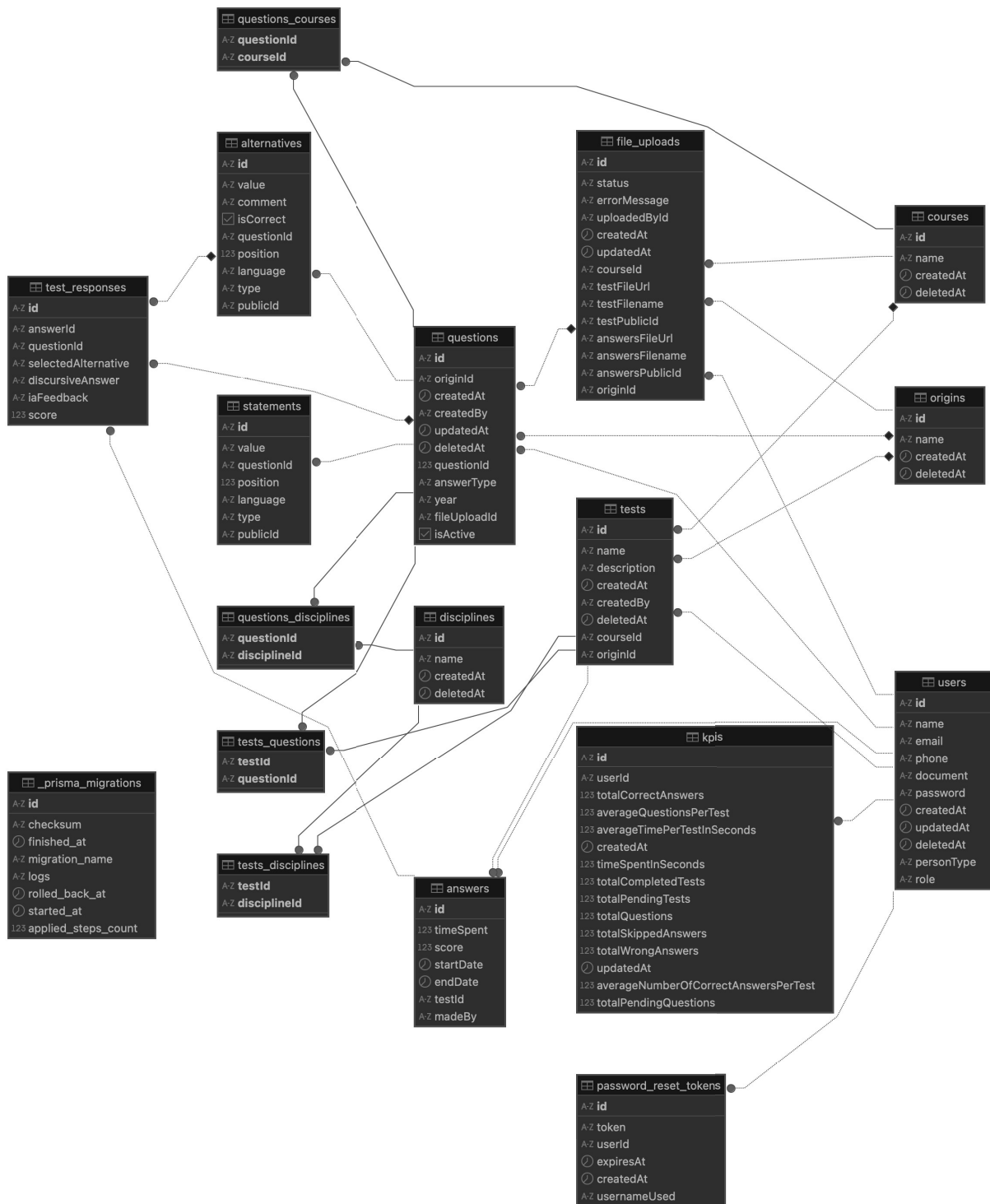


Figura 3: Diagrama Entidade-Relacionamento (DER) do banco de dados do TrainUp. Fonte: elaborado pelo autor a partir do DBaiver (2025).

4.4. Ferramentas e Ambiente de Desenvolvimento

O desenvolvimento integrou um conjunto de ferramentas distribuídas em três camadas aliadas à infraestrutura:

Aplicativo móvel

- **Flutter 3.27.3** (Dart 3.6.1): desenvolvimento multiplataforma;
- **GetX 4.7.2**: gerenciamento de estado e rotas, injeção de dependências;
- **IsarDB 3.1.0**: banco de dados local;
- **Dio 5.8.0**: cliente HTTP para comunicação com o servidor.

Backend monolítico

- **Node.js 22.20.0 + Fastify 5.2.0**: servidor e rotas REST;
- **TypeScript 5.7.2**: tipagem estática;
- **Prisma ORM 6.1.0**: integração com PostgreSQL;
- **Amazon S3 (AWS SDK v3.913.0)**: armazenamento de arquivos PDF e XLSX, e imagens de questões extraídas.
- **OpenAI SDK 6.7.0**: avaliação de questões discursivas;

Microserviço Python

- **Python 3.13.2**: *runtime* principal;
- **PyMuPDF 1.24.9**: leitura e extração de PDF;
- **OpenPyXL 3.1.5**: leitura e extração de XLSX;
- **Pillow 10.4.0**: tratamento de imagens;
- **Psycopg 3.2.12**: comunicação com PostgreSQL;
- **Pika 1.3.2**: consumo do RabbitMQ;
- **Boto3 1.34.0**: integração com S3;
- **HTTPTX 0.27.0**: chamadas assíncronas externas.

Infraestrutura

- **RabbitMQ 3.13**: comunicação assíncrona entre serviços;
- **Docker 27.3.1**: contêineres e ambiente replicável;
- **PostgreSQL 14.19**: persistência e integridade dos dados.

4.5. Correção automática de questões discursivas com IA

A correção de respostas discursivas é realizada por modelos de linguagem (LLMs). O serviço recebe o enunciado (*statement*), a resposta do estudante (*userAnswer*) e uma resposta de referência (*expectedAnswer*), retornando como saída uma nota contida no intervalo $[0, 1]$ e um feedback textual curto.

Modelo e parametrização. O fluxo padrão utiliza o modelo `gpt-4o-mini` (OpenAI) com *temperature* = 0,2 e *max_tokens* = 300, garantindo maior consistência nas notas. Em caso de falha de chamada ou saída inválida, aplica-se *fallback* para o `gemini-1.5-flash` (Google), mantendo o mesmo prompt de avaliação. A resposta do modelo é instruída a retornar somente um objeto JSON com os campos `score` (número real em $[0, 1]$) e `feedback` (string curta).

Rubrica e esquema de saída. A rubrica utilizada no *prompt* mapeia níveis qualitativos para o escore: (i) 1,0 para resposta completa e precisa; (ii) 0,5–0,9 para resposta parcialmente correta; (iii) 0,1–0,4 para resposta fraca; (iv) 0,0 para resposta incorreta ou fora do assunto. O serviço implementa validação de formato (JSON) e saneamento, rejeitando retornos com campos ausentes ou tipos inválidos.

Controles de qualidade e reprocessamento. Para reduzir variações e erros de formatação: (i) instruções explícitas para proibir texto extra fora do JSON; (ii) validação do JSON contra *schema* e até duas tentativas de reprocessamento;

Privacidade e segurança. Somente dados estritamente necessários são enviados ao LLM (enunciado, resposta-aluno e referência), sem informações pessoais identificáveis.

4.6. Síntese do Capítulo

O desenvolvimento do *TrainUp* integrou processos de engenharia de software, práticas arquiteturais, modelagem de dados e tecnologias modernas voltadas ao desenvolvimento distribuído. A combinação entre aplicativo móvel, *backend* centralizado e microserviço de extração permitiu construir uma solução capaz de realizar ingestão automática de questões, simulação de provas e avaliação inteligente das respostas. As ferramentas e bibliotecas apresentadas forneceram suporte técnico para garantir escalabilidade, segurança e eficiência.

5. Considerações Finais

O desenvolvimento do *TrainUp* permitiu a construção de uma solução tecnológica integrada voltada à preparação para o Exame Nacional de Desempenho dos Estudantes (ENADE), contemplando desde a ingestão e organização de provas oficiais até a realização de simulados e a correção automatizada de questões. A aplicação foi fundamentada em conceitos de avaliação educacional, práticas de *feedback* formativo e abordagens modernas de Inteligência Artificial, articulando aspectos pedagógicos, computacionais e legais.

Do ponto de vista técnico, o projeto demonstrou que a combinação entre um aplicativo móvel, um *backend* centralizado e um microserviço especializado em extração de questões é capaz de atender às demandas de escalabilidade, modularidade e desempenho exigidas em um contexto educacional. A utilização das ferramentas mencionadas no capítulo anterior mostrou-se ideal para a construção de um ecossistema robusto e eficiente. Além disso, o uso de modelos de linguagem para a correção automatizada de discursivas evidenciou o potencial da IA generativa na ampliação do acesso ao feedback imediato e personalizado.

Em relação ao impacto educacional, o *TrainUp* facilitará o estudo voltado ao ENADE, oferecendo um ambiente de prática contínua, acompanhamento de desempenho e apoio à aprendizagem individual. Ao estruturar provas e questões de maneira sistemática, o sistema reduz barreiras comuns no processo de preparação, como a indisponibilidade de bases organizadas e a ausência de retornos individualizados sobre o desempenho do estudante.

O projeto também evidenciou desafios relevantes. Entre eles, destacam-se a complexidade do processamento automático de questões discursivas, a necessidade de calibração e validação contínua do desempenho dos modelos de IA e os cuidados necessários para garantir a segurança e privacidade dos dados conforme a Lei Geral de Proteção de Dados (LGPD). Tais desafios reforçam a importância de evoluções futuras, especialmente no refinamento dos modelos avaliativos e na ampliação das funcionalidades do sistema. Somado a isso, cabe dizer que lidar com arquivos PDF e garantir a melhor extração possível foi outro desafio superado.

Em resumo, o *TrainUp* representa um avanço no estado da arte das ferramentas digitais de preparação para o ENADE, oferecendo uma solução escalável, acessível e pedagogicamente fundamentada. Os resultados alcançados demonstram a viabilidade técnica e a relevância educacional da solução, além de abrir caminho para aprimoramentos e extensões que poderão ampliar ainda mais o alcance e o impacto do sistema.

6. Trabalhos Futuros

Embora o *TrainUp* já ofereça um conjunto sólido de funcionalidades voltadas à prática de questões do ENADE, existem melhorias que tornarão o software ainda mais completo. Entre as possibilidades de expansão, destacam-se: (i) um módulo de turmas para docentes e (ii) a geração automática de relatórios personalizados em PDF; (iii) a integração com modelos de IA locais; (iv) a implementação de mecanismos de gamificação; e (v) estudos comparativos com grupos de controle.

Módulo de turmas para docentes

Desenvolver um ambiente no qual professores possam criar turmas, cadastrar alunos, acompanhar o progresso individual e coletivo e elaborar simulados compartilhados. Essa funcionalidade amplia o escopo do sistema para uso institucional, viabilizando análises comparativas, monitoramento contínuo e identificação de dificuldades comuns.

Relatórios em PDF

Gerar relatórios em PDF com análises detalhadas (distribuição de acertos/erros, tempo de resolução, progresso ao longo do tempo e pontos fortes/fracos). Os relatórios atendem a estudantes (autoavaliação e planejamento) e docentes (diagnóstico e aprimoramento), reduzindo trabalho manual e ampliando a acessibilidade às informações.

Integração com IA/LLM local

Investigar o uso de modelos executados localmente para reduzir custos variáveis e dependência de nuvem. Exemplos incluem *embeddings* e classificação semântica locais para correção de respostas discursivas. Pretende-se avaliar arquiteturas leves executáveis no servidor da aplicação, mantendo *fallback* para provedores em nuvem quando necessário. A estratégia inclui:

- Pipelines locais de similaridade semântica e cobertura de conceitos para correção de discursivas;
- Cache e balanceamento entre execução local e remota conforme carga, latência e custo;
- Métricas de concordância com avaliadores humanos para calibrar limiares de aceitação.

Gamificação

Introduzir elementos de engajamento (ranking, conquistas e desafios diários) com notificações para reativação de estudo. O desenho inicial contempla:

- Sistema de pontos (XP), níveis e *badges* por metas atingidas;
- Rankings por curso e desafios diários/semanais configuráveis;
- Notificações *push* (por exemplo, retomada de trilhas e/ou de *streaks*);
- Dashboards de progresso e metas personalizadas para diferentes perfis de estudante.

Estudos comparativos com grupos de controle

Planejar experimentos controlados para avaliar impacto pedagógico do *TrainUp*. A proposta inclui:

- Desenhos A/B (grupo com gamificação/IA vs. grupo controle), acompanhando desempenho por período;
- Métricas: variação de acurácia por área, tempo de resolução, retenção (*streaks*) e questionário de usabilidade (SUS);
- Análises estatísticas (efeito, significância e tamanho de efeito) e relatórios técnicos para subsidiar decisões de produto.

Em resumo, tais melhorias aproximam o *TrainUp* de um ecossistema completo de apoio ao ensino superior, ampliando o impacto educacional por meio de recursos para docentes, relatórios automatizados, IA local com foco em eficiência e custo, ferramentas de engajamento e avaliação rigorosa por estudos controlados.

Agradecimentos

Agradeço ao Prof. Me. Márcio Lopes Júnior pela orientação ao longo do projeto, à banca examinadora pelas contribuições e apontamentos realizados durante a avaliação, e aos participantes dos testes de usabilidade, que forneceram *feedbacks* essenciais para o aprimoramento do software. Agradeço também ao curso de Ciência da Computação do UNIFOR–MG pelo apoio institucional.

Referências

- Azambuja, C.C.d., Silva, G.F.d., 2024. Novos desafios para a educação na era da inteligência artificial. *Filosofia Unisinos* 25, 1–16. Disponível em: <https://www.scielo.br/j/fun/a/jWKkyjpRzxjm6c85yCKv4MN/?format=html&lang=pt>. Acesso em: 14 set. 2025.
- Carvalho, I.F., Ramos, R.G.G., de Souza, A.F., Piantino, L.F.M., 2024. Aplicação da segurança da informação e lgpd para a experiência e usabilidade dos usuários em aplicativos móveis. *Revista Sociedade Científica* 7, 1717–1738. Disponível em: <https://journal.scientificsociety.net/index.php/sobre/article/view/377>. Acesso em: 19 out. 2025.

- Conceição, M.T.d., Pinto, G.S., 2021. Arquitetura de micros-serviços. Interface Tecnológica 18, 53–64. Disponível em: https://revista.fatectq.edu.br/interfacetecnologica/pt_BR/article/view/1186. Acesso em: 26 set. 2025.
- Costa Souza, M.G.d.M., 2020. O enade enquanto política de avaliação da educação superior. Revista Encantar - Educação, Cultura e Sociedade 2, 1–11. Disponível em: <https://www.revistas.uneb.br/index.php/encantar/article/view/9260>. Acesso em: 01 ago. 2025.
- Oliveira, L.H., Stein, L.M., 2018. A autorregulação, avaliação e promoção da aprendizagem por meio da prática de recuperação da memória. Associação Brasileira de Psicologia Escolar e Educacional (ABRABEE) 22, 1–8. Disponível em: <https://www.scielo.br/j/pee/a/jQZfmHL5v3PqLpw8kNG8FtM/?format=html&lang=pt>. Acesso em: 28 ago. 2025.