

IDIOMAX - SOFTWARE PARA GESTÃO DE ESCOLAS DE IDIOMAS

LIMA, Tiago Emanuel de¹, GALVÃO, Giovane²

Centro Universitário Campo Real

Rua Comendador Norberto, 1299 - Santa Cruz – Guarapuava – PR – Brasil

{engs-tiagolima@camporeal.edu.br, prof_giovanegalvao@camporeal.edu.br}

Resumo: Com o avanço das tecnologias digitais, a demanda para plataformas online, aplicativos e sistemas de gestão que facilitam a organização pedagógica e administrativa das instituições está cada vez maior. Nesse contexto, a integração entre tecnologia e educação de idiomas torna-se essencial para otimizar processos, ampliar o acesso e melhorar a experiência de aprendizado. O presente trabalho tem como objetivo o desenvolvimento de um sistema de gestão escolar voltado para instituições de ensino de idiomas, denominado IdiomaX, que centraliza em uma única plataforma digital os processos administrativos, pedagógicos e financeiros da escola. O sistema foi projetado para automatizar tarefas rotineiras, reduzir erros humanos e otimizar o tempo de execução das atividades, promovendo maior eficiência e segurança na gestão escolar. Desenvolvido com tecnologias modernas, como React, Node.js e PostgreSQL, o IdiomaX apresenta um ambiente responsivo, escalável e seguro, que pode ser acessado de qualquer dispositivo. A avaliação final foi conduzida por meio de uma Avaliação Heurística, que validou a funcionalidade do protótipo e identificou pontos de melhoria na experiência do usuário.

Palavras-chave: Gestão escolar. Software educacional. SaaS. PWA.

Abstract: With the advancement of digital technologies, the demand for online platforms, applications, and management systems that facilitate the pedagogical and administrative organization of institutions is growing. In this context, the integration of technology and language education becomes essential to optimize processes, expand access, and improve the learning experience. This work aims to develop a school management system for language education institutions, called IdiomaX, which centralizes the school's administrative, pedagogical, and financial processes in a single

¹ Engenharia de Software, oitavo período, Centro Universitário Campo Real.

² Engenharia de Software, Professor Orientador, Centro Universitário Campo Real.

digital platform. The system is designed to automate routine tasks, reduce human error, and optimize activity execution time, promoting greater efficiency and security in school management. Developed with modern technologies such as React, Node.js, and PostgreSQL, IdiomaX offers a responsive, scalable, and secure environment that can be accessed from any device. The final evaluation was conducted through a Heuristic Evaluation, which validated the functionality of the prototype and identified points for improvement in the user experience.

Keywords: School management. Educational software. SaaS. PWA.

1. Introdução

O setor educacional nos últimos anos tem sido o alvo de muitas empresas para o desenvolvimento de plataformas, com fins de solucionar problemas dentro do contexto em que ela está inserida. Segundo o portal Terra (2024), o setor de educação movimentou R\$ 220,5 bilhões em 2023 e o crescimento previsto para 2024 é de 8% somando um total de R\$ 238,2 bilhões, sendo que R\$ 188,7 bilhões se refere somente a despesas com matrículas e mensalidades de cursos variados.

Quando uma entidade envolve um grande volume de dados, ela fica muito exposta a falhas humanas que podem acarretar danos e perdas irreversíveis, especialmente quando as atividades são realizadas manualmente, assim colocando em risco a segurança dos dados que a empresa possui, como destaca o CEO da Microsoft Nadella (2017), “*one weak link in one area can propagate*”³, apontando a propagação de falhas sendo originado pelos erros humanos.

No cenário atual, todas as empresas deveriam possuir um *software* próprio, sendo assim pode-se pensar e analisar que entidades que não adotam esse recurso, perderão competitividade de mercado. Segundo Lunardi, Dolci e Maçada (2010), “empresas que investiram em TI⁴ de forma planejada apresentaram desempenho percebido superior às demais – especialmente quanto à redução dos custos operacionais”.

Nesse cenário, a implementação de plataformas integradas se revela uma alternativa estratégica para assegurar a acurácia dos dados gerados pela escola e garantir que a empresa está seguindo um caminho alinhado com os seus objetivos, sendo competitiva no mercado atual.

³ “Um elo fraco em uma área pode se propagar”

⁴ TI: Tecnologia da Informação.

Além disso, a consolidação dos dados torna mais simples a tomada de decisões fundamentadas em informações reais, favorecendo uma administração mais eficiente e a análise crítica caso a caso.

1.1. Objetivos

O presente trabalho tem como objetivo geral o desenvolvimento de um *software* voltado à gestão de escolas de idiomas, que visa integrar os processos administrativos, pedagógicos e financeiros em uma única plataforma digital. O sistema busca automatizar tarefas cotidianas, reduzir falhas humanas e otimizar o tempo de execução das atividades escolares, promovendo uma administração mais eficiente e segura.

Além disso, o projeto pretende oferecer uma ferramenta moderna e acessível, capaz de atender diferentes perfis de usuários — como gestores, professores e alunos — de forma prática e intuitiva. Como desdobramento desse propósito, o trabalho também objetiva explorar tecnologias atuais de desenvolvimento, aplicando conceitos de escalabilidade, responsividade e segurança, a fim de proporcionar uma experiência completa e confiável aos usuários.

1.2. Organização do Trabalho

Este trabalho está estruturado em seis capítulos que abordam, de forma sequencial, as etapas teóricas e práticas do desenvolvimento do sistema. O primeiro capítulo apresenta a introdução, contextualizando o problema, a justificativa e os objetivos do estudo. O segundo capítulo trata do referencial teórico, reunindo conceitos relacionados à aprendizagem de idiomas, uso de tecnologias no ensino, modelo SaaS, arquitetura de *software* e desenvolvimento *Web*. No terceiro capítulo, é descrito o estado da arte e a metodologia de pesquisa adotada, incluindo o mapeamento sistemático e a análise dos trabalhos correlatos que embasaram a criação do sistema. O quarto capítulo explica a metodologia de desenvolvimento, detalhando o processo de planejamento, estruturação e implementação do projeto. O quinto capítulo apresenta os resultados obtidos, destacando a construção do banco de dados, da API (*Application Programming Interface*) e da aplicação *Web*, além das principais funcionalidades desenvolvidas. Por fim, o sexto capítulo reúne as considerações finais, nas quais são discutidas as contribuições do sistema IdiomaX e as possibilidades de aprimoramento em versões futuras.

2. Referencial Teórico

Este capítulo mostra o referencial teórico da pesquisa, trazendo os principais conceitos, modelos e fundamentos relacionados ao tema de interesse. A seção 2.1 apresenta como funciona a aprendizagem de idiomas. A seção 2.2 comenta sobre como a tecnologia age em conjunto com o ensino de línguas. A seção 2.3 explica como funciona o modelo de negócio SaaS. A seção 2.4 documenta qual linguagem de programação e tecnologias serão usadas no desenvolvimento *Web*. A seção 2.5 traz o conceito de PWA (*Progressive Web App*) e porque será utilizado. A seção 2.6 fala sobre a arquitetura que será utilizada no *software*. A seção 2.7 esclarece como será feito a base de dados e a API que o PWA irá consumir.

2.1. Aprendizagem de idiomas

Aprender um novo idioma vai muito além da sala de aula, ele permite conectar culturas, ampliar oportunidades e transformar carreiras. Conforme Zuckerberg (2016), *‘we’ve gone from a world of isolated communities to one global community, and we are all better off for it’*⁵, reforçando que dominar uma língua estrangeira abre caminhos para essa comunidade global. Além disso, Gates (2016) enfatiza que o aprendizado — seja de programação, idiomas ou outras habilidades, *“helps you think better... helpful in all domains”*⁶, mostrando como aprender linguagens expande a capacidade cognitiva e impacta positivamente em todas as esferas da vida.

A relação entre bilinguismo e cognição tem sido amplamente estudada por pesquisadores. Limberger e Buchweitz (2012) analisam o controle inibitório e a memória de trabalho em bilíngues, comparando pesquisas brasileiras e canadenses, e apontam que o bilinguismo pode estar associado a um melhor desempenho na resolução de conflitos.

Bialystok e Barac (2013) reforçam que bilíngues tendem a desenvolver habilidades cognitivas mais cedo e mantê-las em um nível elevado ao longo da vida, sugerindo que a experiência linguística influencia positivamente funções executivas.

Considerando todos os aspectos abordados, aprender um novo idioma revela-se uma experiência que vai além do simples domínio linguístico. Trata-se de uma ferramenta poderosa para ampliar horizontes culturais, sociais e profissionais. A fluência em outra língua não só favorece a comunicação global, mas também estimula habilidades cognitivas e ajuda a mantê-las em alto nível.

⁵ “Passámos de um mundo de comunidades isoladas para uma comunidade global, e estamos todos melhor por isso”

⁶ “Ajuda você a pensar melhor... útil em todos os domínios”

2.2. Ensino de línguas por meio de tecnologias

Nas escolas, já percebe-se que o ensino não precisa ser apenas presencial ou só digital: o equilíbrio entre os dois é o que faz a diferença (Prebianca, Finardi e Cardoso, 2015). Pesquisas como as de Prebianca, Finardi e Cardoso (2015) mostram que, quando bem utilizados, aplicativos e plataformas não são apenas "ferramentas extras", mas sim aliados que dão vida ao conteúdo. Os alunos se engajam mais porque veem sentido no que estão aprendendo, é justamente essa conexão que transforma o conhecimento em algo duradouro.

A Tabela 1 representa os resultados obtidos nos cursos de agropecuária e informática. Os autores Prebianca, Finardi e Cardoso (2015) conseguiram trazer dados sobre a(s) maior(es) contribuição(ões) de *softwares* educacionais para o aprendizado de inglês, o antes e após experimentar a abordagem híbrida durante um semestre letivo.

Tabela 1 - Resultados do antes e depois na pesquisa de Prebianca, Finardi e Cardoso (2015).

	Agropecuária		Informática	
	Antes	Após	Antes	Após
Aumento de vocabulário	21	17	18	50
Melhora da habilidade de leitura e compreensão de textos	16	38	22	28
Melhora da habilidade da fala	17	39	5	10
Melhora da habilidade de compreensão auditiva	10	52	3	6
Maior domínio de regras gramaticais	6	64	4	40

Fonte: Prebianca, Finardi e Cardoso (2015)

Observando a Tabela 1, nota-se que quase todos os temas apresentados tiveram uma boa perspectiva por conta dos alunos, sendo evidente a satisfação com o novo modelo de ensino apresentado.

2.3. SaaS (*Software as a Service*)

O SaaS é um modelo de distribuição de aplicativos que revolucionou a forma como utilizamos *softwares*, transformando a entrega, implementação e manutenção desses sistemas (Toledo, J. C. Oliveira, 2019).

Antes, era comum adquirir licenças e instalar programas manualmente. Ele permite que usuários e empresas acessem aplicações diretamente pela internet, sem necessidade de instalação local, garantindo maior praticidade e reduzindo a dependência de infraestrutura própria (MANAGEENGINE, 2025).

Os fornecedores de SaaS normalmente oferecem um modelo baseado em assinatura que reduz os custos iniciais de *software* tradicional, como licenças, instalação ou gerenciamento de infraestrutura. Também não há necessidade de investir em recursos computacionais adicionais para executar o *software*, pois o fornecedor gerencia tudo nos servidores dele (AWS, 2024).

Além disso, o SaaS oferece atualizações automáticas, eliminando a necessidade de instalação manual de novas versões do *software*. Os provedores cuidam das melhorias e correções, garantindo que o sistema esteja sempre atualizado e funcionando corretamente (MANAGEENGINE, 2025).

O Quadro 1 disponibiliza os modelos de cobrança que um SaaS pode oferecer.

Quadro 1 - Modelos de cobrança de SaaS.

Modelo	Descrição
Faturamento com preços fixos	Essa é a forma mais simples de cobrança, em que os clientes pagam uma tarifa fixa pelo acesso ao serviço.
Faturamento por usuário	Nesse modelo, os clientes pagam com base no número de usuários que têm na plataforma. Isso é mais comum em serviços business-to-business (B2B), em que cada membro da equipe de uma organização pode precisar de uma conta.
Faturamento escalonado	Aqui, o provedor de SaaS oferece diferentes níveis de preços, cada um com seu próprio conjunto de recursos. As empresas costumam usar esse modelo para atender clientes de diferentes portes e necessidades.
Faturamento por uso	Os clientes são cobrados com base na quantidade de um determinado serviço que usam. O faturamento por uso é comum em setores como a computação em nuvem, onde os custos são baseados no armazenamento ou no uso de dados.

Modelo	Descrição
Freemium	Nesse modelo, os serviços básicos são gratuitos e os clientes têm a opção de pagar por recursos ou funções avançadas. Frequentemente visto em produtos de SaaS business-to-consumer (B2C), o objetivo desse modelo é dedicar os usuários antes de fazer upselling.
Híbrido	Esse modelo combina diferentes elementos dos outros modelos de cobrança. Uma empresa SaaS pode oferecer uma tarifa fixa para o serviço básico e, em seguida, cobrar com base no uso ou recursos adicionais.

Fonte: STRIPE, (2024)

2.4. PWA (*Progressive Web App*)

Os aplicativos PWA são aplicações que são criadas através de tecnologias *Web* e que podem ser instaladas e executadas em todos os dispositivos, a partir de uma base de código (Microsoft, 2025). Da mesma forma que uma aplicação nativa, um PWA pode ser instalado no dispositivo, integrando-se a ele e a outras aplicações instaladas, além de poder operar offline e em segundo plano (MDN *Web Docs*, 2025).

2.5. Desenvolvimento *Web*

Ao longo dos anos, o desenvolvimento *Web* passou por transformações significativas, ampliando as possibilidades para a criação de sites e aplicativos modernos, ágeis e escaláveis. Esse campo não se limita apenas à construção de páginas estáticas, mas abrange a elaboração de experiências digitais que sejam interativas e responsivas, capazes de funcionar adequadamente em diversos dispositivos e plataformas.

No início da *Web*, o design das páginas era pensado para tamanhos de tela fixos (MDN *Web Docs*, 2025). Entretanto, com a popularização de diferentes tipos de dispositivos, como *smartphones*, *tablets*, *notebooks* e *desktops*, a necessidade de se adaptar a múltiplas resoluções tornou-se fundamental. A responsividade deixou de ser um diferencial para se tornar uma exigência básica no desenvolvimento de aplicações.

Além da criação de aplicações robustas, é imprescindível garantir que o processo de desenvolvimento seja fluido e organizado. O avanço das ferramentas disponíveis para os desenvolvedores facilitou consideravelmente a adoção do *Responsive Web Design* (RWD),

tornando mais acessível a criação de interfaces que se ajustem automaticamente às diversas dimensões de tela.

No Quadro 2 contém as principais tecnologias para o desenvolvimento do projeto *Web*:

Quadro 2 - Tecnologias que serão usadas no projeto *Web*.

Tecnologia	Descrição
React	O React é conhecido como uma biblioteca de JavaScript utilizado para a criação de <i>User Interfaces</i> (UI) (Ebac Online, 2025).
Vite	Vite é uma ferramenta de construção que visa proporcionar uma experiência de desenvolvimento mais rápida e enxuta para projetos <i>Web</i> modernos (VITE, 2025).
Shadcn UI	O shadcn UI é um conjunto de componentes prontos que permitem total personalização (SHADCN/UI, 2025).
ESLint	Mantêm o código padronizado e limpo, facilitando a colaboração em equipe (CWI Software, 2025).
Tanstack Query (antigo React Query)	É uma biblioteca para busca de dados que facilita a busca, o armazenamento em cache, a sincronização e a atualização do estado do servidor em aplicativos da <i>Web</i> (TANSTACK, 2025).
Axios	Axios é um cliente HTTP baseado em promessas para o node.js e para o navegador. É isomórfico (pode rodar no navegador e no node.js com a mesma base de código) (AXIOS, 2025).
Tailwind CSS	O Tailwind CSS funciona escaneando todos os seus arquivos HTML, componentes JavaScript e quaisquer outros modelos em busca de nomes de classes, gerando os estilos correspondentes e depois gravando-os em um arquivo CSS estático (TAILWIND CSS, 2025).
Zod	Zod é uma biblioteca de validação que prioriza o TypeScript. Com

Tecnologia	Descrição
	ela, você pode definir esquemas para validar dados, desde uma string simples até um objeto aninhado complexo (ZOD, 2025).

Fonte: Autor, 2025.

2.6. API (Application Programming Interface)

API 's são mecanismos que permitem que dois componentes de *software* se comuniquem usando um conjunto de definições e protocolos. Por exemplo, a aplicação para a previsão do tempo em seu telefone traz os dados exibidos por meio de uma API e assim consegue mostrar atualizações meteorológicas diárias no telefone (AWS, 2024).

No desenvolvimento de aplicativos *Web*, a API geralmente é utilizada para buscar, criar, atualizar e deletar dados da aplicação, o famoso CRUD (*Create, Read, Update and Delete*) (Miranda, 2024).

No Quadro 3 é apresentado as tecnologias que serão utilizadas para o desenvolvimento da API.

Quadro 3 - Tecnologias que serão usadas na API.

Tecnologia	Descrição
Zod	Zod é uma biblioteca de validação que prioriza o TypeScript. Com ela, você pode definir esquemas para validar dados, desde uma string simples até um objeto aninhado complexo (ZOD, 2025).
ESLint	Mantêm o código padronizado e limpo, facilitando a colaboração em equipe (CWI Software, 2025).
Node.js	Node.js é um ambiente de execução multi-plataforma em JavaScript que permite aos desenvolvedores produzirem aplicações para rede e server-side usando o JavaScript (MDN <i>Web Docs</i> , 2025).
PostgreSQL	O PostgreSQL é um poderoso sistema de banco de dados objeto-relacional de código aberto que utiliza a linguagem SQL (POSTGRESQL GLOBAL DEVELOPMENT GROUP, 2025).

Stripe	Gerenciamento das cobranças de SaaS para as empresas (Stripe, 2024).
--------	--

Fonte: Autor, 2025

2.7. Arquitetura de *Software*

A arquitetura de *software* define o que é o sistema em termos de componentes computacionais, os relacionamentos entre estes componentes, os padrões que guiam a sua composição e restrições. Além da escolha dos algoritmos e estruturas de dados, a arquitetura envolve: decisões sobre as estruturas que formarão o sistema, controle, protocolos de comunicação, sincronização e acesso a dados, atribuição de funcionalidade a elementos do sistema, distribuição física dos elementos, escalabilidade, desempenho, e outros atributos de qualidade (TRUECHANGE, 2025).

Existem diferentes tipos de arquiteturas para API's, como: microserviços, DDD (*Domain-Driven Design*), *Clean Architecture*, entre outras, mas a arquitetura que vai ser utilizado no App será a MVC (*Model-View-Controller*). O Quadro 4 explica o modelo MVC.

Quadro 4 - Modelo MVC.

<i>Model</i>	Define quais dados o aplicativo deve conter. Se o estado desses dados mudar, o modelo geralmente notificará a <i>view</i> (para que a exibição possa mudar conforme necessário) e, às vezes, o <i>controller</i> (se uma lógica diferente for necessária para controlar a visualização atualizada) (MDN <i>Web Docs</i> , 2025).
<i>View</i>	Define como os dados do aplicativo devem ser exibidos (MDN <i>Web Docs</i> , 2025).
<i>Controller</i>	Contém a lógica que atualiza o <i>model</i> e/ou a visualização em resposta à entrada dos usuários do aplicativo (MDN <i>Web Docs</i> , 2025).

Fonte: Autor, 2025

A organização do *software* é a chave para o sucesso de uma aplicação, seguindo esse princípio, a arquitetura geral a ser utilizada é baseada em *monorepo*. Segundo Stanley Gomes (2021), *monorepo* é simplesmente a definição de arquitetura em que você utiliza apenas um repositório para todas as partes da aplicação.

Seguindo essa lógica, o desenvolvimento do App será acelerado, pois tudo estará centralizado, também fazendo com que o compartilhamento de módulos entre projetos seja possível sem a utilização de serviços externos.

2.8. Métodos de Avaliação de Usabilidade

É essencial avaliar a usabilidade de um sistema para assegurar que, além de ser funcional, ele seja também eficiente, eficaz e proporcione uma boa experiência ao usuário final. Na parte de Interação Humano-Computador (IHC), há uma variedade de métodos para realizar essa avaliação. Uma das estratégias mais consolidadas e populares é a Avaliação Heurística.

A Avaliação Heurística é um método de inspeção da usabilidade desenvolvido por Jakob Nielsen, que dispensa a necessidade de participação direta dos usuários finais (NIELSEN; MOLICH, 1990). Em vez disso, especialistas em usabilidade analisam a interface do sistema, confrontando-a com um conjunto de princípios de usabilidade já reconhecidos, chamados de "heurísticas". O propósito é detectar falhas de usabilidade no design da interface para que possam ser corrigidas.

Nielsen estabeleceu dez heurísticas de usabilidade gerais, as quais servem como um guia amplamente adotado no projeto de interfaces. A Figura 1, apresentada logo abaixo, fornece uma representação visual das dez heurísticas.

Figura 1 – As 10 Heurísticas de Usabilidade de Nielsen:



Fonte: O autor (2025).

A aplicação desta metodologia ao projeto IdiomaX é mostrada no Capítulo 5, que contém os resultados obtidos no projeto.

3. Estado da Arte

Nas décadas recentes, a aplicação de tecnologias digitais para objetivos educacionais passou de tendência a uma realidade firmemente estabelecida em instituições de ensino em

todos os níveis. Segundo Prebianca; Finardi; Cardoso (2023), “Observou-se que 91,7% da amostra analisada acreditam na contribuição desse recurso tecnológico para a aprendizagem de inglês”.

Esse capítulo expõe um mapeamento sistemático (MS) sobre a utilização de métodos de aprendizado que utilizam da tecnologia. Além de explorar o que as escolas já utilizam em seu dia a dia, esse estudo tem como objetivo maior estudar como ou porque as escolas utilizam ou não utilizam *softwares* educacionais.

3.1. Metodologia de pesquisa

A metodologia utilizada neste estudo foi o MS, apresentado e explicado por Kitchenham (2007). Essa estratégia possibilita a identificação, análise de publicações científicas relevantes e categorização delas. A execução do MS neste trabalho incluiu a definição das questões de pesquisa, a criação das strings de busca, a escolha dos artigos, a extração e organização dos dados, além da posterior análise e conclusão sobre os resultados.

A realização deste mapeamento se justifica pela necessidade de embasamento teórico e prático para o desenvolvimento do App proposto para gestão de escolas de idiomas, que surge como uma solução às limitações observadas em aplicativos que já estão no mercado.

Dessa forma, a metodologia adotada valida a originalidade do projeto e reforça sua relevância no mercado diante das demandas crescentes por ferramentas tecnológicas que sejam eficientes no setor pedagógico.

A busca foi limitada a trabalhos publicados a partir de 2008. Este período foi considerado, pois alguns artigos dessa época analisaram o impacto da tecnologia no desempenho organizacional de empresas.

O protocolo completo utilizado na condução do mapeamento sistemático encontra-se descrito no apêndice A.

3.2. Resultados do mapeamento sistemático

O MS realizado nesta pesquisa teve como objetivo identificar e organizar as principais evidências científicas relacionadas ao uso de tecnologias educacionais, com ênfase em sistemas de gestão, ensino de idiomas e *softwares* educacionais. A partir da leitura crítica de quatro artigos previamente selecionados, foi possível extrair informações relevantes que subsidiam o desenvolvimento do sistema proposto, além de observar lacunas e oportunidades de inovação.

Nos artigos apresentados e estudados, pudemos perceber que o uso de *softwares* educacionais podem ser uma ferramenta estratégica na gestão acadêmica, contribuindo para o planejamento, acompanhamento e avaliação de programas educacionais, também que os estudantes reconhecem a tecnologia como uma aliada importante no processo de aprendizagem de línguas, por fim, foi validado que o uso estratégico da tecnologia da informação pode melhorar significativamente os processos internos, a tomada de decisões e a competitividade, especialmente em micro e pequenas empresas.

3.2.1. Respostas para as Questões

Com base no MS feito, foi possível responder às questões de pesquisa levantadas no início deste trabalho:

Q1: Qual a demanda observada no contexto educacional que justifica o desenvolvimento de um PWA para gestão escolar?

A principal demanda é a centralização dos dados e comunicação entre diferentes perfis de usuários dentro da instituição, também a necessidade de uma solução acessível, responsiva e multiplataforma, que funcione em diferentes dispositivos. O PWA une o melhor da *Web* com funcionalidades nativas (acesso offline, notificações), sendo ideal para ambientes com conectividade limitada e exigência de flexibilidade (MDN *Web Docs*, 2025).

Q2: Quais os principais benefícios operacionais e financeiros que instituições de ensino podem obter com a adoção de um sistema?

Entre os benefícios operacionais estão a automatização de processos, redução de erros humanos e organização da operação da escola. Financeiramente, sistemas SaaS reduzem custos com infraestrutura e suporte, otimizam recursos humanos e aumentam a eficiência da gestão (AWS, 2024; STRIPE, 2024).

Q3: Quais são as vantagens de centralizar os processos acadêmicos e administrativos em uma única plataforma digital no ambiente escolar?

A centralização promove padronização, integração de dados, melhor comunicação entre usuários e maior segurança. Além disso, facilita decisões estratégicas e garante acessibilidade e escalabilidade com menores custos (TRUECHANGE, 2025).

3.3. Tendências, comparações e diferencial

Com a transformação digital, cresce a adoção de sistemas para a gestão escolar. Tendências como mobilidade, automação e integração de processos em uma única plataforma vêm ganhando espaço. Instituições buscam soluções acessíveis, seguras e responsivas, acompanhando avanços como computação em nuvem e inteligência artificial aplicadas à educação.

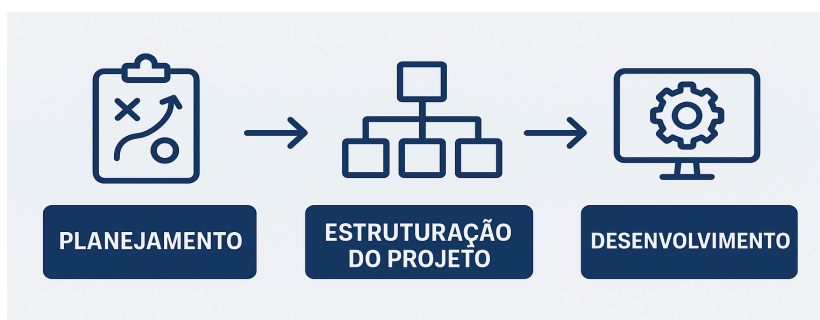
Comparando com métodos tradicionais baseados em papéis e planilhas, plataformas digitais oferecem mais agilidade, controle e confiabilidade. Enquanto processos manuais são lentos e propensos a erros, sistemas integrados permitem gestão em tempo real, relatórios dinâmicos e melhor tomada de decisão.

O diferencial dos PWA 's está na leveza e praticidade: funcionam como apps, sem necessidade de instalação. São acessíveis em qualquer dispositivo e funcionam até offline. Além disso, permitem personalização conforme as demandas da instituição, o que os torna mais eficientes que sistemas genéricos.

4. Metodologia de Desenvolvimento

A metodologia de desenvolvimento adotada baseia-se em um processo sistemático de levantamento e análise de requisitos, no qual busca-se compreender as necessidades, expectativas e restrições do usuário antes do início do desenvolvimento do sistema, tendo como objetivo principal o desenvolvimento de uma solução tecnológica voltada ao gerenciamento da instituição. O processo metodológico adotado envolveu três etapas fundamentais: planejamento, estruturação do projeto e desenvolvimento, como mostra a Figura 2:

Figura 2 - Fluxo do processo metodológico.



Fonte: Autor, 2025

4.1. Planejamento

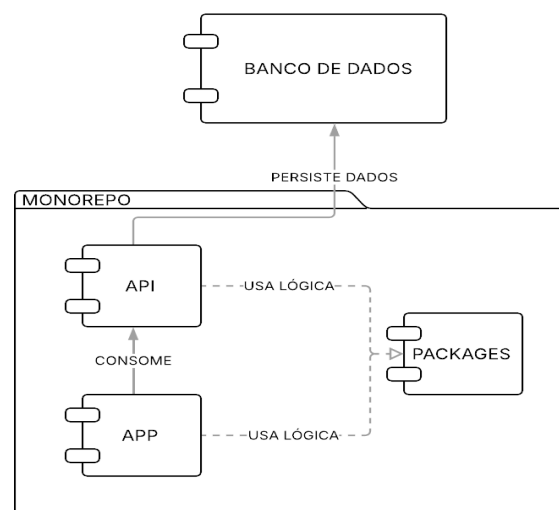
Na fase de planejamento, foram definidos os requisitos e regras de negócio do sistema, a partir da análise das demandas observadas em escolas locais. Essa análise permitiu identificar a necessidade de uma plataforma *all-in-one*, reunindo em um único ambiente recursos de gestão, ensino e acompanhamento pedagógico, com o objetivo de promover a autonomia do estudante, o controle interno na instituição e a confiabilidade dos dados.

4.2. Estruturação do projeto

Durante a etapa de estruturação do projeto, definiu-se o fluxo de desenvolvimento do *software*. Nesse momento, foram tomadas decisões importantes quanto à organização dos arquivos, padrões de codificação, divisão de responsabilidades entre os módulos e utilização de boas práticas para garantir que o código permaneça legível e fácil de manter. O foco foi criar uma estrutura escalável, capaz de suportar futuras evoluções do sistema sem comprometer a qualidade.

O primeiro passo da construção do sistema foi definir o tamanho do escopo, possibilitando uma visão clara da complexidade e extensão do desenvolvimento, então optou-se pelas tecnologias React para a construção da interface do usuário, Zod para validação de dados, Node.js e Fastify para a API, e PostgreSQL como banco de dados relacional. Após o escopo da aplicação definido, determinamos que os módulos API e APP seriam mantidos no mesmo projeto, com o objetivo de compartilhar o módulo de autorização e validação dos dados, criando assim um *monorepo* (Stanley Gomes, 2021). A Figura 3 mostra como foi definida a estrutura de pastas na raiz do projeto:

Figura 3 - Estrutura de pastas na raiz do projeto.



Fonte: Autor, 2025

Com a pasta “API” contendo todas as rotas que manipulam o banco de dados, a pasta “APP” que contém toda a parte de interface, controle de acesso e fluxos de navegação do sistema, a pasta “PACKAGES”, que é onde está os módulos de autorização e tipagens, que são compartilhados entre APP e a API e por fim o banco de dados, que é onde mantemos todos os dados do sistema.

4.3. Desenvolvimento

A fase de desenvolvimento teve como objetivo, transformar o planejamento e a estruturação definidos anteriormente em uma solução funcional. O processo de desenvolvimento foi conduzido de forma iterativa, com foco na organização do código, escalabilidade e reutilização de componentes, garantindo a consistência entre as camadas da aplicação. Os pontos mais importantes dessa fase foram: A instalação das dependências, criação dos módulos, configuração do banco de dados, integração do projeto com o servidor do Stripe, implementação da comunicação do APP com a API e o desenvolvimento dos módulos de autenticação e autorização.

Com essas etapas bem definidas e desenvolvidas, o IdiomaX foi capaz de ser construído de uma forma organizada, escalável e eficiente. A estrutura adotada permitiu o reaproveitamento de código e garantiu que as funcionalidades pudessem ser implementadas com facilidade.

5. Resultados

Esta seção descreve o funcionamento geral do sistema, regras de negócio e suas principais características.

5.1. Funcionamento do sistema

O IdiomaX foi projetado para gerenciar todo o ciclo administrativo, pedagógico e pedagógico de uma escola de idiomas, desde o cadastro inicial da instituição até o acompanhamento acadêmico dos alunos. O acesso à plataforma inicia-se na página principal, onde o responsável pela instituição seleciona a opção de criação de conta e realiza o cadastro de seus dados pessoais, tornando-se automaticamente o super usuário da instituição. Em seguida, é solicitado as informações da empresa, como nome, CNPJ (Cadastro Nacional da Pessoa Jurídica), endereço e contatos institucionais, preparando o ambiente para uso.

Após a identificação da instituição, o sistema direciona o administrador para a contratação do plano de assinatura, que pode ser mensal ou anual. Esse processo ocorre de

forma integrada ao Stripe, permitindo o registro do cartão de crédito e liberando automaticamente um período de teste de 14 dias. Além disso, toda gestão financeira é centralizada através de *webhooks*, que garantem a sincronização contínua entre Stripe e banco de dados, mantendo as informações de planos, assinaturas, pagamentos e valores sempre atualizadas sem intervenção manual. Com a conta habilitada, o gestor realiza seu primeiro login e passa a configurar a estrutura acadêmica da escola.

No ambiente administrativo, o gestor possui acesso integral às funcionalidades do sistema. O primeiro passo consiste no cadastro dos cursos, incluindo informações como carga horária, valor de matrícula, frequência mínima exigida e limite de notas. A partir dos cursos, ele cria os níveis de formação e as disciplinas pertencentes a cada um. A organização pedagógica é complementada pela configuração de salas e turmas, onde são definidos dias de aula, horários, professores responsáveis e número de vagas. Quando um aluno é matriculado, o sistema gera automaticamente suas parcelas financeiras conforme as regras definidas no curso, evitando cálculos manuais e reduzindo riscos de erro.

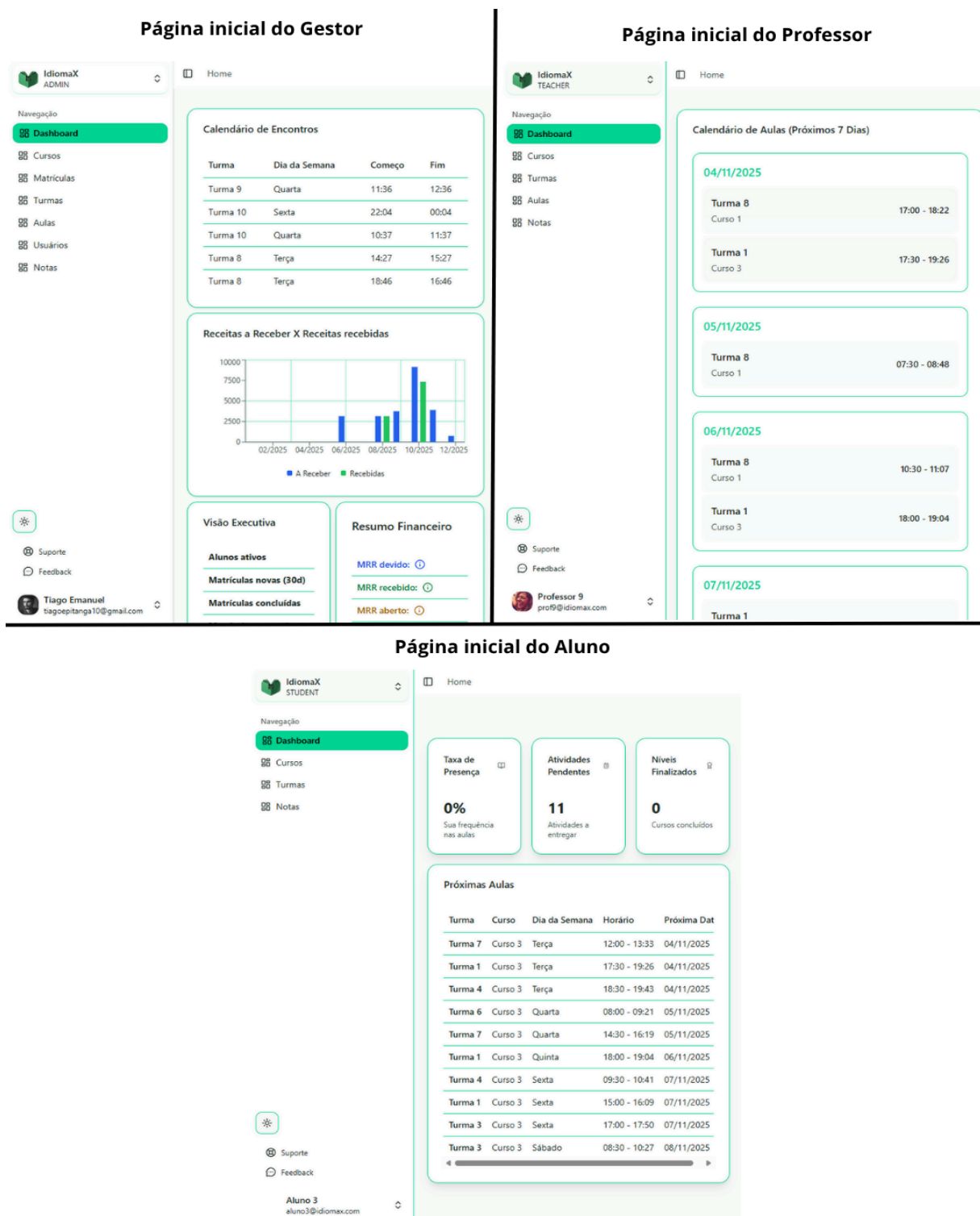
O IdiomaX também contempla operações administrativas essenciais, como edição, desativação ou exclusão de registros, controle de professores, troca de turmas, trancamento de matrícula e emissão de histórico escolar. Todas as ações refletem instantaneamente nos relatórios institucionais, garantindo consistência e rastreabilidade dos dados. O gestor pode ainda atribuir papéis de acesso aos usuários, distinguindo perfis de aluno, professor ou administrador, o que garante controle de permissões e segurança da informação.

No módulo pedagógico, voltado ao professor, a plataforma disponibiliza ferramentas para o registro diário de aulas, incluindo conteúdo ministrado, frequência dos alunos, datas e horários. As aulas são registradas de forma sequencial, permitindo acompanhamento cronológico do planejamento. O professor também pode criar atividades avaliativas e exames, definir valores de nota e registrar resultados. Essas informações ficam acessíveis para o gestor e para o aluno, promovendo transparência e organização acadêmica.

O aluno, por sua vez, acessa o sistema com login individual e visualiza em seu painel todas as informações pertinentes à sua vida escolar. É possível acompanhar notas, faltas, atividades, materiais virtuais, mensalidades pagas e pendentes, além de solicitar reposição de aulas quando necessário. O estudante também pode atualizar parte de seus dados pessoais, como e-mail e telefone, fortalecendo a autonomia dentro do ambiente acadêmico.

A Figura 4 apresenta as principais telas em que ocorre todo o processo explicado.

Figura 4 - Principais telas que compõem o sistema.



Fonte: Autor, 2025

5.2. Modelagem dos Dados

O desenvolvimento do BD (Banco de dados) foi pensado e planejado para poder suportar e suprir as necessidades das funcionalidades do IdiomaX, a tecnologia escolhida foi o

Postgres, então, realizando o mapeamento do BD no arquivo `schema.prisma`, conseguimos replicar nosso modelo criado no Postgres, alinhando as tipagens e instâncias criadas do cliente prisma com o BD. A Figura 5 mostra como é feito o mapeamento, usando a tabela de empresas como exemplo:

Figura 5 - Estrutura da tabela de empresas no `schema.prisma`.

```
model companies {
  id          String @id @default(uuid()) @db.VarChar(256)
  name        String @db.VarChar(256)
  cnpj        String @unique @db.VarChar(256)
  phone       String @unique @db.VarChar(256)
  email       String? @db.VarChar(256)
  logo_16x16_url String? @db.VarChar(1024)
  logo_512x512_url String? @db.VarChar(1024)
  social_reason String? @db.VarChar(256)
  state_registration String? @db.VarChar(256)
  tax_regime   String? @db.VarChar(256)
  address      String @db.VarChar(256)
  owner_id     String @db.VarChar(256)
  created_at   DateTime @default(now())
  created_by   String @db.VarChar(256)
  updated_at   DateTime @updatedAt
  updated_by   String @db.VarChar(256)
  active       Boolean @default(true)

  creator users? @relation("CompaniesCreatedBy", fields: [created_by], references: [id])
  updater  users? @relation("CompaniesUpdatedBy", fields: [updated_by], references: [id])
}
```

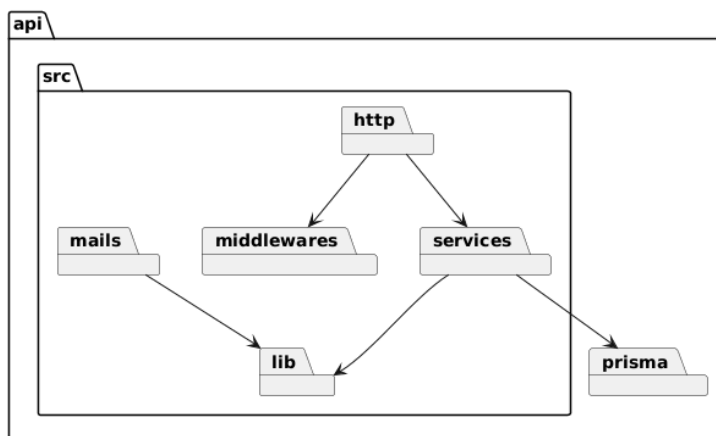
Fonte: Autor, 2025

Após o schema do BD feito, precisamos adicionar uma variável ambiente “DATABASE_URL” contendo a URL do BD e rodar o comando “pnpm db:migrate:dev” (comando configurado apenas para esse projeto) no terminal, assim o prisma cria as migrations necessárias, conecta com o banco de dados e aplica as alterações.

Para entendermos como as assinaturas são controladas pelo sistema juntamente ao Stripe (Stripe, 2025), segue a Figura 6, onde mostra a estruturação das tabelas compartilhadas com o *webhook* do Stripe.

Com a análise da Figura 6, pode-se perceber que as assinaturas são controladas por empresa, ou seja, cada empresa deve possuir uma única assinatura para que o *software* seja habilitado.

Figura 7 - Estrutura do módulo API.



Fonte: Autor, 2025

A pasta Prisma é onde fica as configurações do BD, o `schema.prisma`, que já foi citado anteriormente, o script de *seed*, que insere dados pré-definidos ao rodar o comando “`pnpm db:seed`” e também a pasta *migrations*, que guarda o versionamento do BD.

Na pasta *src* (source), considerada o coração do projeto, estão localizados os arquivos responsáveis pelos registros e pela lógica do servidor. Dentro dela, a pasta *controllers* tem a função de processar as requisições dos usuários. A Figura 8 apresenta a estrutura básica dessa funcionalidade.

Em *src*, o arquivo “`server.ts`”, que mapeia todas as rotas da aplicação; *lib*, armazena-se funções utilitárias; *middlewares*, guarda funções que intermediam requisições do cliente e API; *mails*, pasta que armazena modelos de e-mail, como o email que é enviado após o usuário solicitar uma troca de senha; *services* (serviços), nela são instanciadas e configuradas dependências como a do Stripe, para operações de pagamento e gerenciamento de assinaturas, e o Prisma, para interação com o banco de dados.

Também na API, contamos com uma documentação gerada automaticamente pelo *Swagger*, onde declaramos um escopo básico em cada rota para gerá-la, é o que mostra na Figura 9.

Como mostra a Figura 9, podemos organizar as requisições por *tags*, dar uma descrição para cada uma no *summary*, definir uma segurança, no caso estamos usando o *bearer*, que armazena o *token de login*, o *response* (resposta da requisição) e *body* (corpo da requisição), ambos fazem o papel de tipar entradas e saídas de dados, nesse caso, estamos utilizando a biblioteca *Zod* para isso.

Figura 8 - Exemplo de controller (*Sign-in*).

```
export async function SignIn(app: FastifyInstance) {
  async (request, reply) => {
    const { username, password } = request.body;

    const user = await prisma.users.findUnique({
      where: { username },
      include: {
        member_on: {
          include: {
            company: true,
          }
        }
      }
    });

    if (!user || !(await bcrypt.compare(password, user.password))) {
      throw new BadRequestError('Credenciais inválidas');
    }

    const token = app.jwt.sign({
      sub: user.id,
    });

    reply.status(200).send({
      token,
      message: 'Bem vindo!',
    });
  },
};
```

Fonte: Autor, 2025

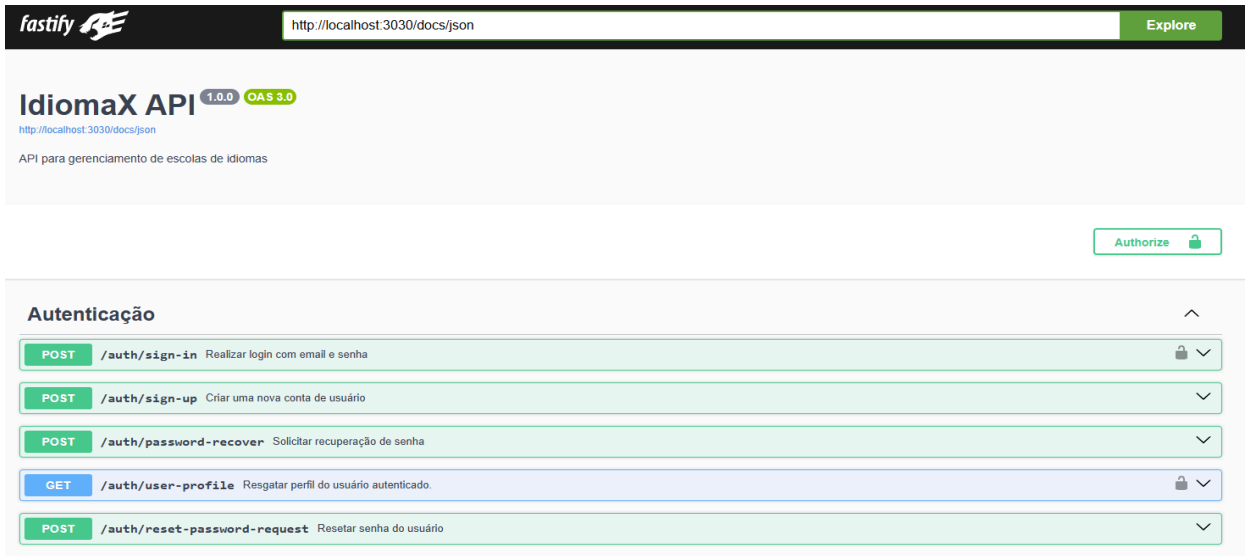
Figura 9 - Exemplo de escopo para geração de documentação do *Swagger*.

```
schema: {
  tags: ['Autenticação'],
  summary: 'Realizar login com email e senha',
  security: [{ bearerAuth: [] }],
  response: {
    200: SignInApiResponse,
  },
  body: SignInApiRequest
},
```

Fonte: Autor, 2025

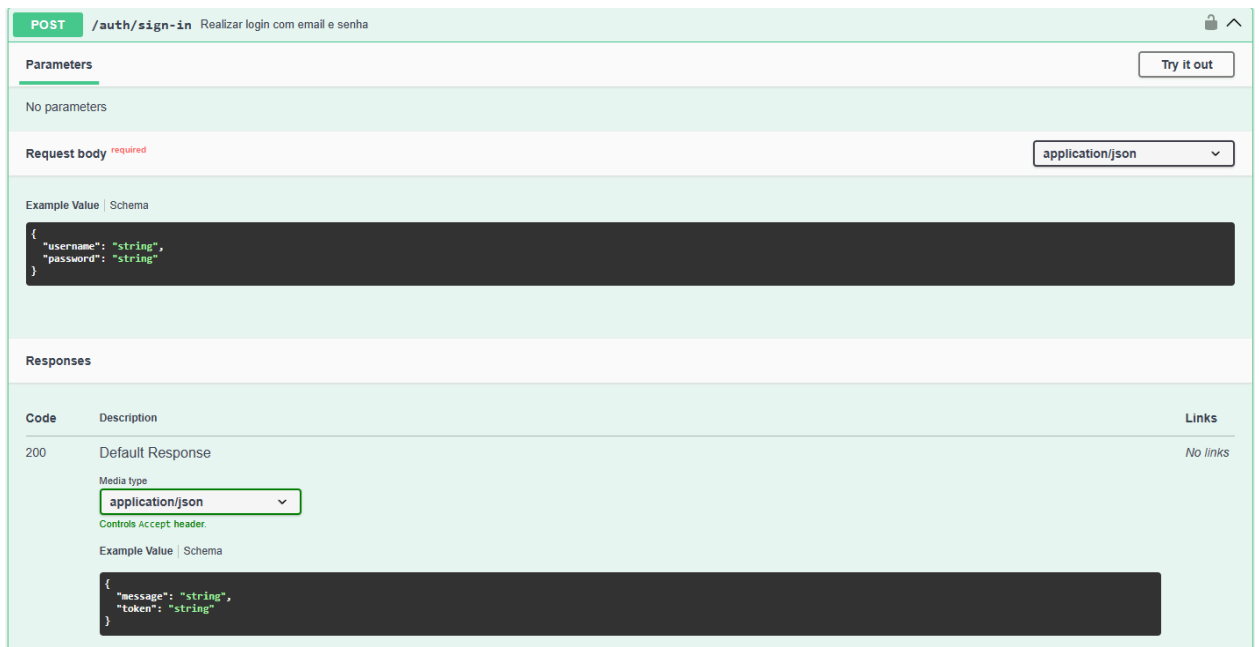
Após essa configuração, o *Swagger* gera uma documentação completa de cada rota, mencionando as *URL*'s, o *body* cadastrado e as possíveis respostas, como mostra as Figuras 10 e 11:

Figura 10 - UI do *Swagger*.



Fonte: Autor, 2025

Figura 11 - Detalhes da rota *Sign-in*.



Fonte: Autor, 2025

5.4. Desenvolvimento do Aplicativo

Após o desenvolvimento do BD e da API, deu-se início à construção do aplicativo *web*, responsável por toda a camada de interação com o usuário.

A estrutura do app segue o padrão de rotas públicas e privadas, conforme o tipo de autorização do usuário. As rotas públicas englobam as páginas de *login*, cadastro e recuperação de senha, enquanto as privadas são acessíveis apenas após a autenticação. Essas rotas são controladas por um token JWT (*JSON Web Token*), que é validado pelo backend após cada requisição. Essa organização garante a segurança e o isolamento de informações sensíveis.

Primeiramente, para receber o acesso ao sistema, é preciso realizar o cadastro de um usuário, uma empresa e selecionar o plano de assinatura, vale acrescentar que, após escolhido o plano, é gerado uma assinatura de teste, onde o usuário ganha 14 dias grátis para testar a plataforma. A Figura 12 mostra as telas responsáveis por esse fluxo.

Vale ressaltar que, após a escolha do plano, o usuário é redirecionado para uma tela de agradecimento e pode clicar em um botão que o redireciona para a tela inicial da plataforma, já que, nesse ponto do fluxo de cadastro ele já está autenticado.

A interface foi construída utilizando o Shadcn como biblioteca de estilização, permitindo uma padronização visual consistente entre todos os módulos. Elementos como botões, tabelas, formulários e modais seguem o mesmo padrão de espaçamento, cores e tipografia, proporcionando uma experiência coesa e intuitiva ao usuário. A Figura D.1, no apêndice D, dá um exemplo de padronização de telas, usando as telas de gerenciamento de cursos e aulas.

Foi criado também o painel administrativo do administrador, onde ele pode ter uma visão privilegiada da instituição, tendo acesso ao calendário de encontros, taxa de presença dos alunos, taxa de inadimplência, receitas e despesas mensais, taxa de ocupação das turmas e muito mais. A Figura D.2, no apêndice D, mostra uma pequena parte do painel do administrador.

As funções dos gestores incluem, cadastrar cursos, níveis, disciplinas, turmas, usuários, além de vincular usuários às turmas, o APP também oferece recursos para a gestão de matrículas e controle financeiro. A Figura D.3, no apêndice D, mostra como é o formulário para matricular um aluno.

Os professores possuem um painel com funcionalidades voltadas ao acompanhamento pedagógico. É possível registrar aulas, marcar presenças, lançar notas. A Figura D.4, no apêndice D, mostra como ficou a tela de registrar aulas.

Figura 12 - Fluxo de cadastro do usuário.

1 - Cadastro do super usuário

Cadastro do SUPER usuário
Depois você poderá cadastrar mais usuários admin em sua empresa.

Nome: João da Silva | Email: tiagoepitanga10@gmail.com

CPF: 12345458901 | Gênero: Masculino | Telefone: 11999999999

Data de nascimento: 01/01/2000 | Endereço: Rua das Flores, 123

Usuário: tiago10 | Senha: ***** | Confirmação da senha: *****

Já possuo uma conta

2 - Cadastro da instituição

Cadastro da Instituição
Preencha os dados da instituição.

Nome da empresa: Empresa Teste

CNPJ: 12345458000199 | Endereço: Av. Paulista, 1000

Telefone: 42984066200

Já possuo uma conta

3 - Seleção do plano de assinatura

Plano Mensal
R\$ 79.90/mês

- Gestão completa de estudantes
- Controle de matrículas e mensalidades
- Dashboard administrativo avançado
- Organização de cursos e turmas
- Controle de presença e frequência
- Gestão de professores e horários
- Sistema de tarefas e materiais
- Controle de acesso com permissões

Plano Anual
10% de desconto
R\$ 862.92/ano

- Gestão completa de estudantes
- Controle de matrículas e mensalidades
- Dashboard administrativo avançado
- Organização de cursos e turmas
- Controle de presença e frequência
- Gestão de professores e horários
- Sistema de tarefas e materiais
- Controle de acesso com permissões

Fonte: Autor, 2025

Os alunos, por sua vez, possuem acesso ao painel acadêmico, onde podem visualizar suas notas, faltas, histórico escolar e parcelas financeiras, apresentado na Figura D.5, no apêndice D.

O sistema também oferece feedback imediato em todas as ações do usuário, como confirmações de sucesso, alertas de erro e validações de campos em formulários. Essa abordagem melhora a comunicação entre o sistema e o usuário, reduzindo o índice de erros

operacionais. A Figura D.6, no apêndice D, mostra um formulário uma tentativa de submissão com campos em branco.

5.4. Avaliação do Produto

Para aferir a usabilidade do sistema, empregou-se o método de Avaliação Heurística, em estrita conformidade com os preceitos teóricos estabelecidos na seção 2.8. O autor deste estudo assumiu a função de avaliador especialista, procedendo à inspeção da interface do aplicativo com o intuito de verificar aderência ou violação às 10 Heurísticas de Usabilidade propostas por Jakob Nielsen. O principal escopo dessa análise foi a identificação precoce de eventuais pontos de fricção (*friction points*) e a prospecção de oportunidades de aprimoramento no design da interação.

A aplicação metódica resultou na catalogação de um conjunto de problemas de usabilidade, cuja documentação integral, incluindo descrições detalhadas, classificação de severidade e as respectivas recomendações para correção, encontra-se consolidada no Quadro B.1, no Apêndice B.

Os problemas classificados com severidade média constituem os desafios mais proeminentes, pois comprometem a fluidez da experiência sem necessariamente inviabilizar a tarefa. Foram identificados atributos relevantes, tais como a dificuldade ou o caminho não intuitivo para a edição de uma matrícula. Tal achado representa uma violação direta da heurística de "Controle e liberdade do usuário". Adicionalmente, verificou-se a presença de inconsistência no padrão visual empregado em elementos como ícones para a diferenciação de botões na *sidebar*, o que compromete significativamente a heurística de "Consistência e padrões". Embora não sejam impeditivos para a funcionalidade básica do *software*, esses elementos quebram a curva de aprendizado e podem mitigar a percepção de profissionalismo e confiabilidade da solução.

Para a solução desses problemas, propõe-se padronizar os ícones da interface para reforçar a consistência visual, ajustar o gerenciamento de cache após exclusões para garantir atualização imediata dos dados, incluir redirecionamentos nos cards de desempenho e tornar o processo de edição de matrículas mais intuitivo, aprimorando controle, previsibilidade e eficiência na interação do usuário.

A Avaliação Heurística revelou-se uma ferramenta de valor inestimável para a validação do IdiomaX, culminando na geração de um backlog robusto de melhorias de *User Experience* (UX). A implementação dessas ações corretivas em ciclos de desenvolvimento

subsequentes é fundamental para assegurar o refinamento contínuo e a otimização da qualidade final do produto.

6. Considerações finais

O principal objetivo deste trabalho foi desenvolver um *software* voltado à gestão de escolas de idiomas, integrando os processos administrativos, pedagógicos e financeiros em uma única plataforma digital. Esse objetivo foi alcançado de forma satisfatória, sendo que, o sistema automatiza tarefas, reduz falhas humanas e otimiza o tempo de execução das atividades escolares.

O *software* permitiu constatar que o uso de sistemas centralizados favorece a otimização de processos administrativos e pedagógicos, promovendo maior controle, transparência e integração entre docentes e discentes. Além disso, o *IdiomaX* mostrou-se uma ferramenta adaptável a diferentes contextos educacionais, podendo ser ampliado ou customizado conforme as necessidades institucionais.

Por fim, destaca-se que o IdiomaX conseguiu estabelecer uma base sólida para futuras versões e aprimoramentos, devido à abrangência da proposta do projeto, que permite evoluir em múltiplos aspectos. As próximas versões podem contar com a criação de novos perfis de usuário, como financeiro e secretário, trazendo um controle maior sobre as permissões dos usuários, também uma inteligência artificial para conversação e treino de idiomas, sendo ela destinada para os alunos, trazendo mais um diferencial para as escolas que utilizam o IdiomaX.

REFERÊNCIAS

AWS. *Software* como serviço (SaaS). 2024. Acesso em: 13 jun. 2025.

AXIOS. Introdução. acesso em 15 jun. 2025.

BIALYSTOK, E.; BARAC, R. Correlação entre fatores de experiência linguística bilíngue e controle inibitório. *Letras Hoje*, v. 53, n. 1, 2013. Acesso em: 9 jun. 2025.

CWI SOFTWARE. ESLint + Prettier: a dupla perfeita para produtividade e padronização de código. *CWI Software*, 2025. Acesso em: 10 jun. 2025.

EBAC ONLINE. React: o que é e como funciona. Ebac Online, 2025. Acesso em: 10 jun. 2025.

GATES, Bill. Facebook F8: 6 major takeaways from Mark Zuckerberg's keynote. Inverse, 12 abr. 2016. (Citação de Gates no keynote do F8). Acesso em: 8 jun. 2025.

KITCHENHAM, B. Guidelines for performing systematic literature reviews in *software engineering*. EBSE Technical Report, v. 2, n. 1, 2007.

LIMBERGER, J.; BUCHWEITZ, A. Estudos sobre a relação entre bilinguismo e cognição: o controle inibitório e a memória de trabalho. *Letrônica*, v. 5, n. 3, 2012. Acesso em: 9 jun. 2025.

LUNARDI, Guilherme Lerch; DOLCI, Pietro Cunha; MAÇADA, Antônio Carlos Gastaud. Adoção de tecnologia de informação e seu impacto no desempenho organizacional: um estudo realizado com micro e pequenas empresas. *Revista de Administração – RAUSP*, São Paulo, v. 45, n. 1, p. 5–17, mar. 2010. DOI: 10.1590/S0080-21072010000100001. Acesso em: 25 ago. 2025.

MANAGEENGINE. SaaS models and management overview, 2025.

MDN WEB DOCS. Progressive Web Apps. MDN Web Docs, 2025. Acesso em: 10 jun. 2025.

Microsoft. Visão geral dos aplicativos web progressivos (PWAs). Microsoft Edge Developer, 28 jan. 2025. Acesso em: 13 jun. 2025.

MIRANDA, Luiz. CRUD: o que é e como funciona na programação. *Revista Quero Bolsa*, 25 jan. 2024. Acesso em: 13 jun. 2025.

Nadella, Satya. Satya Nadella: governments shouldn't weaponize *software* bugs. *Business Insider*, 10 out. 2017. Acesso em: 07 jun. 2025.

NIELSEN, Jakob; MOLICH, Rolf. Heuristic evaluation of user interfaces. In: PROCEEDINGS OF THE SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 1990, Seattle. Anais [...]. New York: ACM, 1990. p. 249-256.

PEREIRA, F. M.; REIS, R. M. C. dos; SILVA, N. da. O *software* de gestão educacional como contribuição tecnológica à gestão de um programa. Caderno Pedagógico, [S. l.], v. 22, n. 4, p. e13839, 2025. DOI: 10.54033/cadpedv22n4-008.

POSTGRESQL GLOBAL DEVELOPMENT GROUP. About. PostgreSQL, 25 set. 2025. Acesso em: 10 out. 2025.

PREBIANCA, G.; FINARDI, K. R.; CARDOSO, A. R. Ensino-aprendizagem em contextos híbridos: o que pensam os alunos sobre o uso da tecnologia em aulas de inglês. Revista Brasileira de Linguística Aplicada, 2015.

SHADCN/UI. Documentação do shadcn/ui. shadcn/ui, 2025. Acesso em: 10 jun. 2025.

STANLEY, Gomes. Monorepo? O que é? Devo usar? Dev.to, 2025. Acesso em: 13 jun. 2025.

STRIPE. SaaS billing 101: what businesses need to know. Stripe, 2024. Acesso em: 13 jun. 2025.

TAILWIND CSS. Como usar o Vite com Tailwind CSS. Tailwind CSS, 2025. Acesso em: 11 jun. 2025.

TANSTACK. TanStack Query – React: visão geral. TanStack, 2025. Acesso em: 11 jun. 2025.

TERRA. Educação deve movimentar cerca de R\$ 10,1 bilhões este ano no Rio Grande do Sul. Terra, 2025. Acesso em: 7 jun. 2025.

Toledo, J. C. Oliveira, L. C. de; DIAS, L. M. A. *Software* como Serviço (SaaS): uma abordagem inovadora para entrega de aplicações. Revista Gestão & Tecnologia, v. 19, n. 1, p. 150–167, 2019. Acesso em: 13 jun. 2025.

TRUECHANGE. Tipos de arquiteturas de *software*. Blog TrueChange, 2025. Acesso em: 13 jun. 2025.

VITE. Guia do Vite. Vite, 2025. Acesso em: 10 jun. 2025.

ZOD. TypeScript-first schema validation with static type inference. Zod, 2025. Acesso em: 11 jun. 2025.

ZUCKERBERG, Mark. Mark Zuckerberg is the world's youngest elder statesman. Time, 13 maio 2016. Acesso em: 8 jun. 2025.

Apêndice A - PROTOCOLO DO MAPEAMENTO SISTEMÁTICO

1. Questões de pesquisa

No Quadro A.1 é mostrado as questões de pesquisa (QP) utilizadas:

Quadro A.1 - Questões de pesquisa.

QP1	Qual a demanda observada no contexto educacional que justifica o desenvolvimento de um PWA para gestão escolar?
QP2	Quais os principais benefícios operacionais e financeiros que instituições de ensino podem obter com a adoção de um sistema?
QP3	Quais são as vantagens de centralizar os processos acadêmicos e administrativos em uma única plataforma digital no ambiente escolar?

Fonte: Autor, 2025

2. Strings de busca

Para garantir a abrangência e a relevância dos artigos encontrados, foram definidas *strings* de buscas, que foram elaboradas com base nos principais termos relacionados ao tema deste estudo. Essas *strings* combinam palavras-chave em português e inglês para garantir o máximo de assertividade nas buscas.

No Quadro A.2 é mostrado as *strings* de busca utilizadas para chegar aos resultados.

Quadro A.2 - Strings de busca.

1	("SaaS" OR "software como serviço" OR "plataforma integrada" OR "sistema de gestão escolar online" OR "Sistema de gestão") AND ("gestão educacional" OR "escola digital" OR "tecnologia na educação" OR "plataforma acadêmica" OR "Escolas de inglês") AND ("automação escolar" OR "dados escolares" OR "comunicação entre escola e pais" OR "Gestão escolar")
2	("Software educacional" AND "psicopedagogia" AND "educação infantil")
3	"software educacional"

Fonte: Autor, 2025

3. Análise dos artigos

Para a realização da análise dos materiais encontrados ao longo da pesquisa, foi necessário definir critérios de inclusão (CI) e exclusão (CE) que orientassem a seleção dos estudos mais relevantes e voltados para escolas de idiomas. A adoção desses critérios contribuiu para tornar o processo de avaliação mais objetivo e coerente com os objetivos do trabalho, permitindo uma triagem mais eficiente dos conteúdos analisados.

Os CI utilizados para fazer a pesquisa foram:

- Artigos após a 2005;
- Artigos em português ou inglês;
- Artigos completos.

Após as escolhas dos artigos, novamente foram expostos a uma nova análise utilizando os CE, que são:

- Artigos que eram privados;
- Artigos que saíram fora do tema abordado;

4. Classificação e extração dos dados

Para a realização desta pesquisa, foram selecionados e analisados quatro artigos científicos com temáticas complementares, relacionados ao uso de tecnologias na educação, especialmente voltadas para o ensino de idiomas, gestão educacional e avaliação de *softwares* educacionais. Os 4 artigos foram classificados pela sua área de abordagem, o Quadro A.3 mostra a classificação.

Quadro A.3 - Artigos escolhidos para análise.

Artigo 1	O software de gestão educacional como contribuição tecnológica à gestão de um programa	Gestão educacional e sistemas administrativos
Artigo 2	Ensino-aprendizagem em contextos híbridos: o que pensam os alunos sobre o uso da tecnologia em aulas de inglês	Tecnologia no ensino de idiomas e percepção discente
Artigo 3	Uma proposta de método para a avaliação de softwares educacionais através de uma visão psicopedagógica	Avaliação de softwares educacionais (método e critérios)

Artigo 4	Adoção de tecnologia de informação e seu impacto no desempenho organizacional	Impacto da TI na performance de micro e pequenas organizações
----------	---	---

Fonte: Autor, 2025.

Apêndice B – Resultados da Avaliação Heurística

Este apêndice tem por objetivo detalhar os resultados da Avaliação Heurística, a qual foi conduzida no sistema IdiomaX, conforme referenciado na Seção 5. O Quadro B.1 apresenta a lista dos problemas de usabilidade mais relevantes que foram identificados, indicando a(s) heurística(s) de Nielsen violada(s), a respectiva classificação de severidade e as recomendações de melhoria para o design da interface.

Quadro B.1 – Principais Problemas de Usabilidade Identificados na Avaliação Heurística

Problema de Usabilidade Identificado	Heurística(s) Violada(s)	Severidade	Descrição e Recomendação
Os ícones da <i>sidebar</i> não mostram nenhum padrão com a função dele.	Consistência e padrões.	Média	Escolher ícones que exemplificam a função do botão, por exemplo: no botão de <i>dashboard</i> exibir um ícone de gráfico.
Ao excluir uma matrícula, o sistema redireciona o usuário para a listagem de matrículas, mas a matrícula excluída ainda era mostrada, sendo preciso o usuário reiniciar a página para normalização.	Visibilidade do status do sistema	Média	Após o usuário ser redirecionado, invalidar o cache da query que é responsável por trazer dados das matrículas existentes.
A seção que mostra as piores taxas de assiduidade	Flexibilidade e eficiência de uso	Média	Implementar um link de redirecionamento para cada

Problema de Usabilidade Identificado	Heurística(s) Violada(s)	Severidade	Descrição e Recomendação
não possui um botão para redirecionamento para o gestor visualizar somente a turma escolhida			turma que aparecer nesta seção.

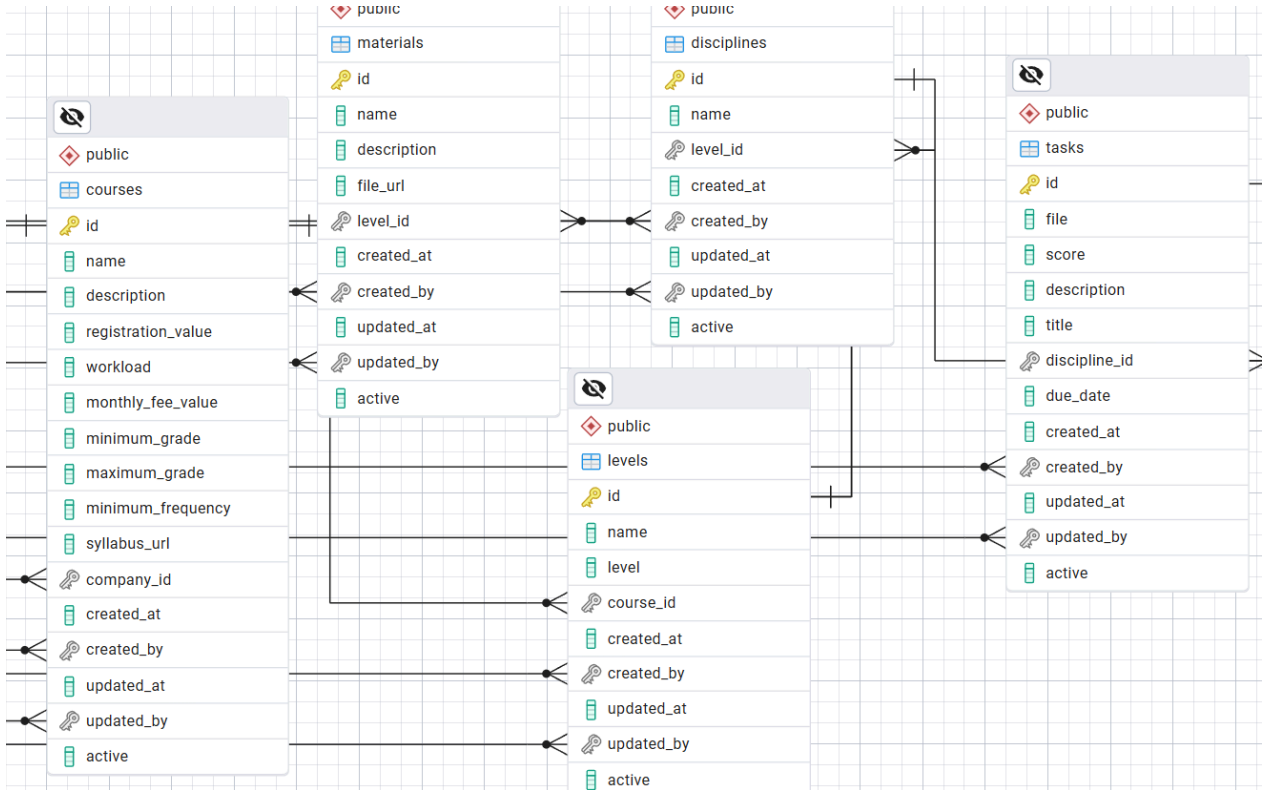
Fonte: Autor, 2025

Concluimos que algumas das regras foram violadas, evidenciando a necessidade de ajustes no design e na interação do sistema IdiomaX. As inconsistências observadas, como a falta de padronização dos ícones, a ausência de atualização imediata de informações e a limitação na navegação entre componentes, indicam oportunidades claras de aprimoramento da usabilidade.

A aplicação das recomendações propostas — como a padronização de ícones, o aprimoramento do feedback do sistema e o aumento da flexibilidade de navegação — contribuirá significativamente para tornar o sistema mais coerente, eficiente e agradável de utilizar, alinhando-se às boas práticas de design de interface e às heurísticas de Nielsen.

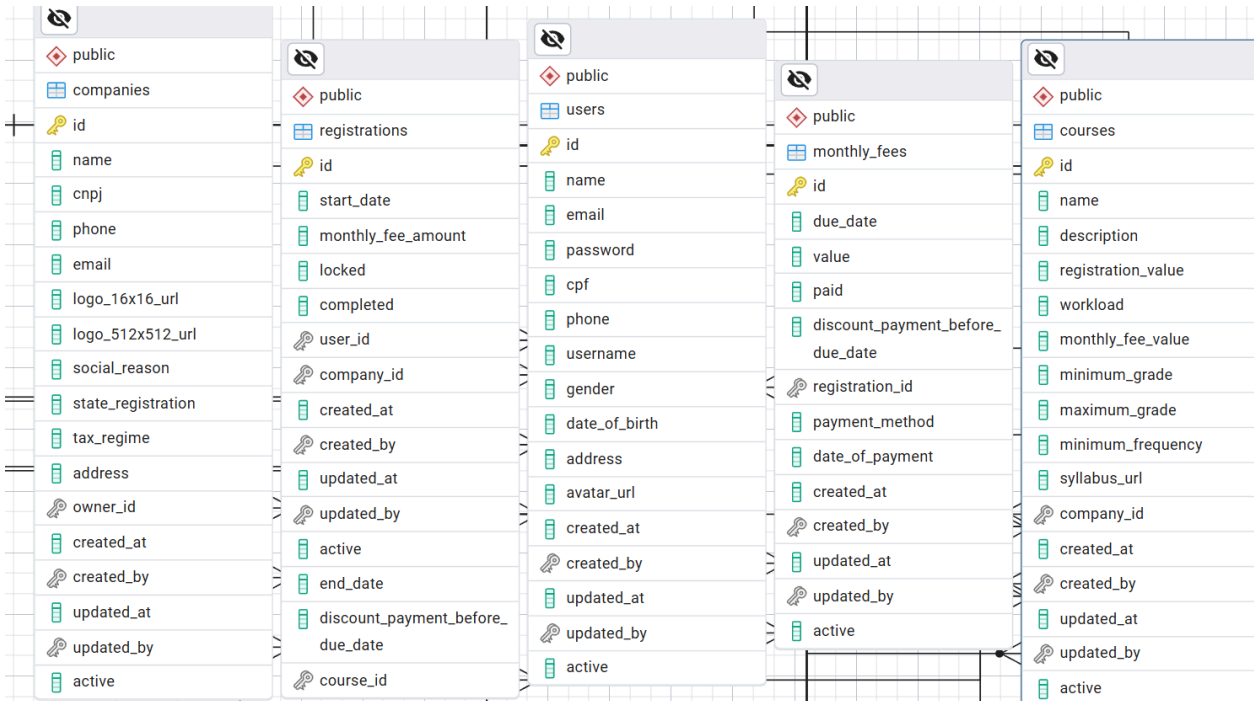
Apêndice C – Diagramas do Banco de Dados

Figura C.1 - Tabelas no módulo de cursos.



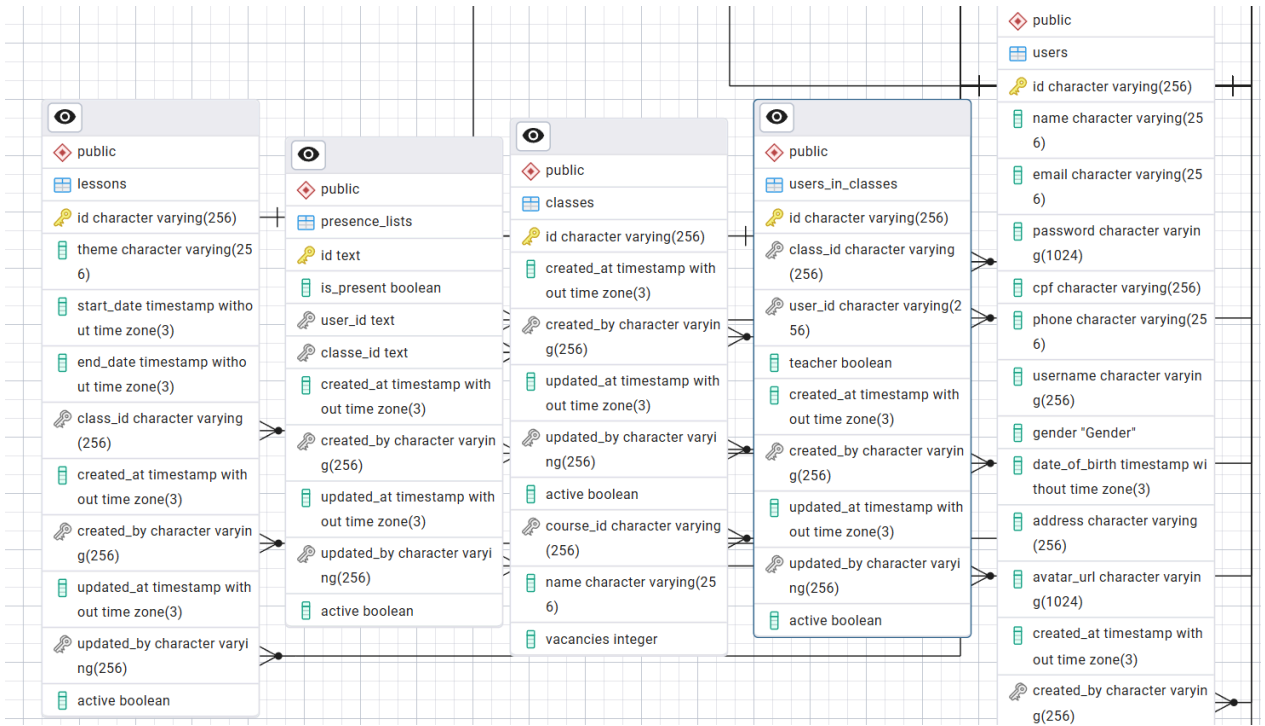
Fonte: Autor, 2025

Figura C.2 - Tabelas no módulo de matrículas.



Fonte: Autor, 2025

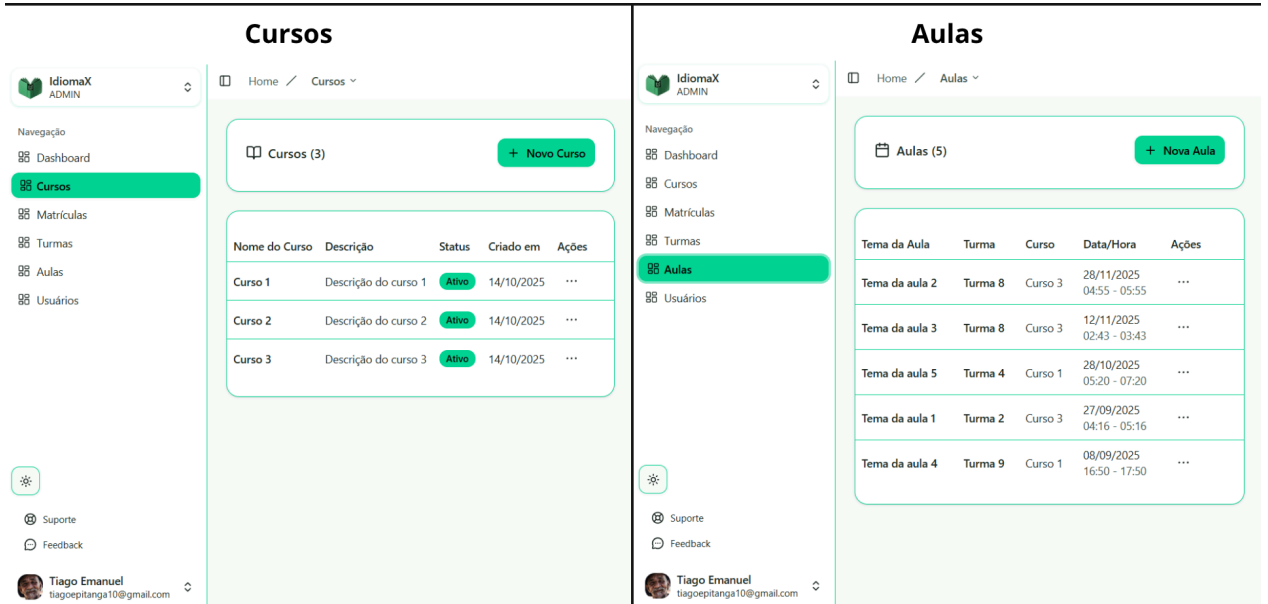
Figura C.3 - Tabelas do módulo de aulas.



Fonte: Autor, 2025

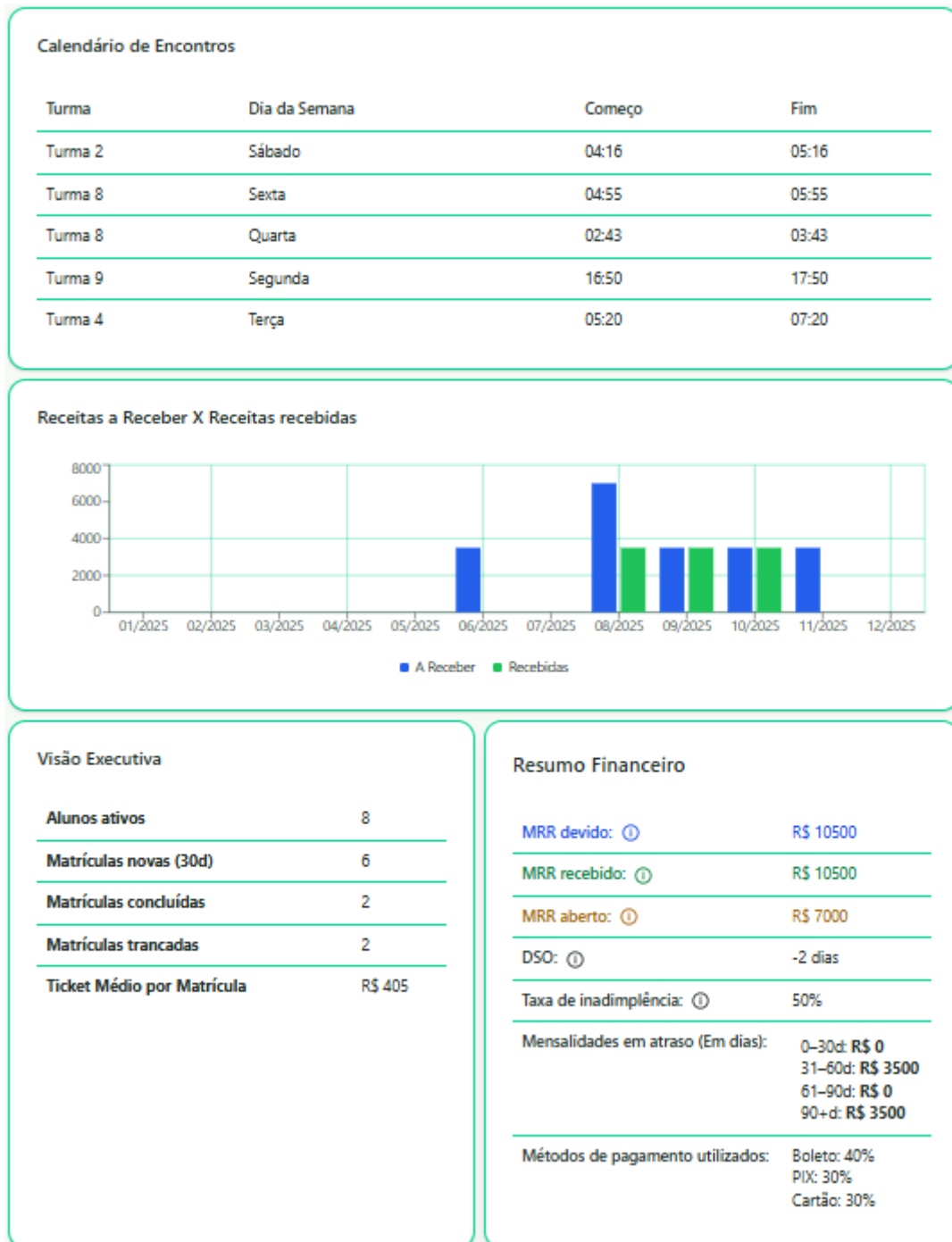
Apêndice D – Figuras do Desenvolvimento do Aplicativo

Figura D.1 - Exemplo de padronização de telas.



Fonte: Autor, 2025

Figura D.2 - Parte do painel do administrador.



Fonte: Autor, 2025

Figura D.3 - Formulário para matricular um aluno.

 **Nova Inscrição**

Preencha os campos abaixo para criar uma nova inscrição de estudante.

Estudante *	Curso *	Data de Início *
<input type="text" value="Selecione o estudante..."/>	<input type="text" value="Selecione um curso"/>	<input type="text" value="dd/mm/yyyy"/>
Valor da Mensalidade (R\$) *	Valor do Desconto de Pagamento Antecipado (R\$) *	
<input type="text" value="0.00"/>	<input type="text" value="0.00"/>	

 **Matricular**

Fonte: Autor, 2025

D.4 - Tela para registrar aulas.

IdiomaX
TEACHER

Navegação

- Cursos
- Turmas
- Aulas**

Home / Aulas

← Criar Nova Aula

Preencha os dados para criar uma nova aula. A lista de presença será gerada automaticamente com todos os alunos da turma selecionada.

Tema da Aula

⊗ Tema da aula é obrigatório

Turma

Data/Hora de Início

Data/Hora de Fim

⊕ **Lista de Presença Automática**

Ao criar a aula, uma lista de presença será gerada automaticamente com todos os alunos da turma selecionada. Você poderá marcar as presenças na tela de edição da aula.

Cancelar **Criar Aula**

🔧

📧 Suporte

💬 Feedback

👤 Professor 9
prof9@idiomax.com

Fonte: Autor, 2025

Figura D.5 - Dashboard do aluno.

IdiomaX STUDENT

Home

Navegação

- Dashboard
- Cursos
- Turmas
- Notas

Taxa de Presença **87%**
Sua frequência nas aulas

Atividades Pendentes **13**
Atividades a entregar

Níveis Finalizados **2**
Cursos concluídos

Próximas Aulas

Turma	Curso	Dia da Semana	Horário	Próxima Data
Turma 10	Curso 1	Domingo	14:00 - 15:42	26/10/2025
Turma 9	Curso 1	Domingo	16:00 - 16:57	26/10/2025
Turma 8	Curso 1	Domingo	17:30 - 18:31	26/10/2025
Turma 10	Curso 1	Segunda	13:30 - 14:37	27/10/2025
Turma 2	Curso 1	Segunda	16:30 - 17:42	27/10/2025
Turma 9	Curso 1	Terça	09:30 - 10:36	28/10/2025
Turma 8	Curso 1	Terça	17:00 - 18:22	28/10/2025
Turma 2	Curso 1	Terça	20:30 - 21:14	28/10/2025
Turma 8	Curso 1	Quarta	07:30 - 08:48	29/10/2025
Turma 8	Curso 1	Quinta	10:30 - 11:07	30/10/2025


Aluno 1
aluno1@idiomax.com

Fonte: Autor, 2025

Figura D.6 - *Dashboard* do aluno.

Cadastrar usuário

Preencha os dados para cadastrar um novo usuário



Tipo de Usuário <input type="text" value="Selecione o tipo"/>	Nome <input type="text" value="Nome completo"/>	Email <input type="text" value="Email"/>
⊗ Selecione uma função válida	⊗ Nome deve ter pelo menos 2 caracteres	⊗ Email inválido
CPF <input type="text" value="CPF"/>	Gênero <input type="text" value="Selecione o gênero"/>	Telefone <input type="text" value="Telefone"/>
⊗ CPF deve ter exatamente 11 dígitos	⊗ Selecione o gênero	⊗ Telefone muito curto
Data de nascimento <input type="text" value="dd/mm/aaaa"/>	Endereço <input type="text" value="Endereço completo"/>	Usuário <input type="text" value="Nome de usuário"/>
⊗ Data de nascimento é obrigatória	⊗ Endereço muito curto	⊗ Nome de usuário deve ter pelo menos 3 caracteres

Senha ⓘ

Cadastrar ⊕

Fonte: Autor, 2025