

Jackson Galdino da Silveira - 2018010136
Lucas Fernandes Seraggi - 2018017674
Paulo Junior Carvalho de Paiva - 2018004783

Sistema Escolar Centralizado

Itajubá
Julho de 2023

Jackson Galdino da Silveira - 2018010136
Lucas Fernandes Seraggi - 2018017674
Paulo Junior Carvalho de Paiva - 2018004783

Sistema Escolar Centralizado

Monografia referente ao Trabalho Final de
Graduação do curso de Engenharia de Com-
putação da Universidade Federal de Itajubá

Universidade Federal de Itajubá – UNIFEI
Instituto de Engenharia de Sistemas e Tecnologia da Informação
Engenharia de Computação

Orientador: Bruno Tardiole Kuehne

Itajubá
Julho de 2023

*Este trabalho é dedicado às crianças adultas que,
quando pequenas, sonharam em se tornar cientistas.*

Agradecimentos

Nossos profundos agradecimentos ao orientador Bruno Tardiole Kuehne por sua orientação e apoio durante a execução desse trabalho e ao professor Edvard Martins de Oliveira.

Gostaríamos de agradecer também à instituição por proporcionar um ambiente acadêmico estimulante, repleto de recursos e infraestrutura de qualidade, sendo fundamentais para o desenvolvimento deste projeto.

Em especial, gostaríamos de expressar nosso reconhecimento aos professores do curso de Engenharia de Computação. Suas vastas experiências, conhecimentos e orientações foram essenciais ao longo do nosso percurso acadêmico. Somos muito gratos pela dedicação em transmitir ensinamentos, pela paciência e pelo incentivo constante. Suas contribuições enriqueceram nossa formação e foram de fundamental importância para a conclusão deste trabalho.

Agradecemos aos demais membros da equipe acadêmica e administrativa que, de diversas formas, contribuíram para o nosso crescimento pessoal e profissional durante toda a jornada na universidade.

Por fim, somos gratos a todos os colegas de curso que também trilharam essa trajetória, pelas trocas de conhecimento, apoio mútuo e momentos de aprendizado compartilhados.

*“Não vos amoldeis às estruturas deste mundo,
mas transformai-vos pela renovação da mente,
a fim de distinguir qual é a vontade de Deus:
o que é bom, o que Lhe é agradável, o que é perfeito.
(Bíblia Sagrada, Romanos 12, 2)*

Resumo

Este trabalho apresenta o desenvolvimento de uma aplicação *web* que integra um conjunto de soluções para as necessidades emergentes do ensino à distância, as quais ganharam maior destaque durante a pandemia do vírus da COVID-19. Dessa forma, busca-se ofertar serviços eficazes para as escolas, com o intuito de resolver os problemas relacionados à administração de funcionários e alunos e à ministração e consumo de conteúdos de cursos. O sistema foi implementado com base em APIs RESTful, construídas através da linguagem *TypeScript* e consumidas no *front-end* desenvolvido em *Flutter*. Os resultados obtidos demonstram a eficácia da plataforma em proporcionar uma experiência de usuário facilitada e moderna que promove um ambiente educacional de fácil utilização. Objetiva-se que o presente trabalho seja tido como referência para instituições de ensino que enfrentam desafios semelhantes, e que a aplicação possa ser aprimorada e expandida em trabalhos futuros.

Palavras-chave: sistema escolar centralizado. plataforma de ensino.

Abstract

This work presents the development of a web application that integrates a set of solutions for the emerging needs of distance learning, which gained greater prominence during the COVID-19 virus pandemic. In this way, it seeks to offer effective services to schools, with the aim of solving problems related to the administration of employees and students and the delivery and consumption of course content. The system was implemented based on RESTful APIs, built using the TypeScript language and consumed in the front-end developed in Flutter. The results obtained demonstrate the effectiveness of the platform in providing an easy and modern user experience that promotes a user-friendly educational environment. It is intended that the present work be taken as a reference for educational institutions that face similar challenges, and that the application can be improved and expanded in future works.

Keywords: centralized school system. teaching platform.

Lista de ilustrações

Figura 1 – Exemplo de estruturação de curso da plataforma <i>Moodle</i>	26
Figura 2 – Tipos de atividades e recursos disponíveis na plataforma Moodle	26
Figura 3 – Exemplo de página da plataforma <i>Canvas</i>	27
Figura 4 – Exemplo de página principal da plataforma <i>Google Classroom</i>	28
Figura 5 – Diagrama de sequência para inserção de recurso	30
Figura 6 – Tela de inserção do recurso questionário no <i>Google Classroom</i>	31
Figura 7 – Tela de inserção do recurso questionário no <i>Canvas</i>	31
Figura 8 – Tela de inserção do recurso questionário no <i>Moodle</i>	32
Figura 9 – Gráfico que representa os <i>frameworks</i> mais populares usados no <i>back-end</i> em janeiro de 2023	36
Figura 10 – Linguagens mais utilizadas pelos desenvolvedores em 2022	37
Figura 11 – Diagrama Pé-de-galinha do sistema proposto	42
Figura 12 – Página inicial do sistema	47
Figura 13 – Tela de cadastro de escolas	48
Figura 14 – Seleção de ambiente para <i>login</i>	48
Figura 15 – <i>Login</i> no ambiente de professores e alunos	49
Figura 16 – <i>Login</i> no ambiente de instituições de ensino	49
Figura 17 – Tela de recuperação de senha	50
Figura 18 – Página inicial de perfil de uma instituição de ensino	50
Figura 19 – Página de cursos de um aluno	51
Figura 20 – Visualização de um curso pelo perfil aluno	51
Figura 21 – Página de cursos para um professor	52
Figura 22 – Visualização de um curso por um professor	52
Figura 23 – Adição de recursos as disciplinas	53
Figura 24 – Botão para acesso da lista de presença e notícias do curso	53
Figura 25 – Visualização das notícias de um curso	54
Figura 26 – Cadastro de uma notícia	54
Figura 27 – Visualização da lista de presença de um curso	55
Figura 28 – Cadastro de uma nova lista de presença	55
Figura 29 – Página de turmas de uma escola	56
Figura 30 – Edição de uma turma	56
Figura 31 – Alunos de uma escola	57
Figura 32 – Cadastro de um aluno	57
Figura 33 – Professores de uma escola	58
Figura 34 – Cadastro de um professor	58
Figura 35 – Página de suporte	59

Lista de tabelas

Tabela 1 – Comparação entre os produtos similares: implementação	29
Tabela 2 – Comparação entre os produtos similares: ambiente de utilização	30

Lista de abreviaturas e siglas

ACID	Propriedades de banco de Atomicidade, Consistência, Isolamento e Durabilidade
APIs	<i>Application Programming Interfaces</i> (Interface de Programação de Aplicações)
COVID-19	<i>Corona Virus Disease</i> - 2019 (Doença do coronavírus - 2019)
CRUD	<i>Create, Read, Update and Delete</i> (Criar, Ler, Atualizar e Excluir)
DB	<i>DataBase</i> (Base de dados)
EAD	ensino à distância
HTTP	<i>Hypertext Transfer Protocol</i> (Protocolo de Transferência de Hipertexto)
IESTI	Instituto de Engenharia de Sistemas e Tecnologias da Informação
JSON	<i>JavaScript Object Notation</i> (Notação de Objeto JavaScript)
JWT	<i>JSON Web Token</i>
RESTful	<i>Representational State Transfer</i> (Transferência de estado representacional)
SQL	<i>Structured Query Language</i> (Linguagem de Consulta Estruturada)
UI	<i>User Interface</i> (Interface de Usuário)
UNIFEI	Universidade Federal de Itajubá
URI	<i>Uniform Resource Identifier</i> (Identificador Uniforme de Recurso)
URL	<i>Uniform Resource Locator</i> (Localizador Uniforme de Recursos)
Web	<i>World Wide Web</i>

Sumário

1	INTRODUÇÃO	17
1.1	Considerações Iniciais	17
1.2	Objetivo	18
1.2.1	Objetivos Específicos	19
1.3	Justificativa	20
2	REVISÃO BIBLIOGRÁFICA	21
2.1	JSON Web Token (JWT)	22
2.2	RESTful	22
2.3	Framework	23
2.4	JavaScript	23
2.5	TypeScript	23
2.6	Node	24
2.7	Express	24
2.8	Flutter	24
2.8.1	Dart	25
2.9	Trabalhos Relacionados	25
3	METODOLOGIA	33
3.1	Procedimento técnico	33
3.2	Sistema	33
3.2.1	Banco de dados	33
3.2.2	Back-end	35
3.2.3	Front-End	38
3.3	Ambiente de Desenvolvimento	39
4	DESENVOLVIMENTO	41
4.1	Objeto	41
4.2	Banco de Dados	42
4.3	Back-end	43
4.3.1	Segurança	44
4.3.2	Middlewares	44
4.3.3	Model	44
4.3.4	Router	44
4.3.5	Controller	45
4.3.6	Services	45

4.4	<i>Front-end</i>	45
4.4.1	<i>Core</i>	45
4.4.2	<i>Model</i>	45
4.4.3	<i>Widgets</i>	46
4.4.4	<i>Pages</i>	46
4.4.5	<i>Services</i>	46
4.5	<i>Deploy</i>	46
5	EXPERIMENTOS E RESULTADOS	47
5.1	Página Inicial	47
5.2	<i>Login</i>	48
5.3	Tela Inicial	50
5.4	Cursos	50
5.5	Notícias	53
5.6	Lista de Presença	54
5.7	Turmas	55
5.8	Alunos e Professores	56
5.9	Suporte	58
6	CONCLUSÃO	61
6.1	Trabalhos Futuros	61
	REFERÊNCIAS	63

1 Introdução

1.1 Considerações Iniciais

As salas de aula virtuais tornaram-se essenciais devido à pandemia sem precedentes do vírus da COVID-19, levando à mudanças significativas nos sistemas educacionais tradicionais. Para manter a continuidade na educação e cumprir as diretrizes de distanciamento social, as escolas tiveram de adotar as tecnologias digitais. À medida que se enfrenta tal situação desafiadora, é crucial integrar ferramentas computacionais para garantir a manutenção da atividade educacional.

Várias soluções baseadas na *web* passaram a ser adotadas pelas escolas durante os períodos de isolamento social, conforme aponta os resultados da pesquisa 2020 do CETIC ([CETIC.BR, 2020](#)). Algumas dessas soluções foram: gravações de videoaulas, uso de plataformas virtuais, envio de eventos por *e-mail*, ensino por plataformas *online* e utilização de grupos em aplicativos ou redes sociais. Esses métodos de ensino de emergência provocaram alterações significativas no sistema educacional, mesmo no contexto pós-pandemia.

Consoante ao Conselho Nacional de Educação ([CNE, 2018](#)), as escolas secundárias passaram recentemente por uma transformação, com 20% do total de cursos agora podendo ser concluídos remotamente. A aceitação do ensino híbrido depende fortemente do nível de dedicação e esforço do aluno ao usar sistemas de aprendizado *online*, conforme descoberto em um estudo de Chiu & Wang (2008) ([CHIU; WANG, 2008](#)). Alunos que não têm experiência com aprendizado virtual podem ter dificuldade em se ajustar às plataformas com curvas de aprendizado acentuadas.

O principal objetivo nesta situação é descobrir soluções viáveis para o ensino à distância. Isso implica na necessidade crucial de desenvolvimento de *software* para auxiliar no processo. Considerações importantes devem ser feitas sobre a correlação entre o uso da plataforma de ensino e o desempenho dos alunos, bem como o potencial das redes sociais servirem como ambientes educacionais.

Para acomodar melhor os alunos com habilidades mínimas ou inexistentes em *e-learning*, as instituições educacionais precisam incorporar plataformas instrucionais centradas em uma estrutura unificada e regulamentada. Ao fazê-la, os alunos poderão obter todos os materiais e recursos relevantes mediante um único local, removendo assim os obstáculos associados à aprendizagem de como navegar em diversas plataformas digitais e também à necessidade de uso das redes sociais, que podem ser ambientes não propícios para o propósito educacional, como meio de envolvimento nas atividades de sala de aula.

Diante desse problema, um sistema escolar integrado precisa ser desenvolvido para atender às diferentes situações existentes no campo da educação. Essa plataforma integrada permitirá que diferentes atores envolvidos no processo de aprendizagem utilizem um único ambiente para gestão acadêmica, entrega e consumo de cursos. Os professores poderão ministrar os conteúdos dos cursos, acessar as informações das turmas e dos alunos e fornecer atividades e notas. Os alunos, por outro lado, podem acessar seus cursos, atividades e visualizar suas notas. Além disso, o sistema disponibilizará um perfil dedicado às organizações, permitindo a gestão de recursos, cadastro de professores e cadastro de alunos.

Nesse contexto, a presente monografia tem como propósito apresentar o desenvolvimento e implementação de um sistema escolar centralizado que integra as necessidades emergentes do ensino à distância e fornece soluções eficazes para as instituições de ensino. Ao abordar diferentes aspectos relacionados com a integração das tecnologias digitais nas salas de aula virtuais, pretende-se facilitar a melhoria dos processos educativos no contexto pós-pandemia e garantir uma educação de qualidade para todos.

1.2 Objetivo

Com base no contexto pós-pandemia, que insere as instituições educacionais em ambientes adaptativos quanto a utilização acelerada da tecnologia nas metodologias educacionais, o presente trabalho visa a construção de uma aplicação, que atenda às necessidades das escolas e dos alunos que enfrentam os novos desafios do ensino à distância, embasada nos conceitos da *Web 2.0*. Essa, por sua vez, possui como estrutura primordial o fornecimento de serviços por meio de um *browser*. Tal abordagem foi escolhida devido aos seus pilares de sustentação que estavam conforme o sistema objetivado:

- **Acessibilidade:** o portal *web* pode ser acessado de qualquer dispositivo com conexão à *internet*, incluindo computadores *desktop*, *laptops*, *tablets* e *smartphones*. Essa acessibilidade contribui para seu uso generalizado e pelo maior número de integrantes possível;
- **Atualizações e manutenção facilitadas:** por intermédio da *web*, atualizações e manutenção podem ser realizadas centralmente no servidor, exigindo esforço mínimo dos usuários finais. Isso garante que os usuários sempre acessem a versão mais recente da aplicação;
- **Escalabilidade:** o projeto pode ser dimensionado para acomodar inúmeros usuários, pois o mesmo é hospedado em servidores, tirando proveito da infraestrutura de nuvem;

- Colaboração e compartilhamento: por se tratar de um ambiente de aprendizado é crucial que o compartilhamento contínuo de dados ou informações entre os usuários seja de fácil acesso. Dessa forma, a *internet* torna-se ideal para a possibilidade de aplicações de trabalhos em equipe ou colaboração remota.

Por conseguinte é preciso destacar que os requisitos mínimos para a utilização do sistema proposto seriam: ter algum tipo de experiência e conhecimentos básicos sobre a utilização de Sistemas Operacionais e de navegadores de *internet*. Por conta disso, o público-alvo é composto, principalmente, por escolas que abrigam cursos para os anos finais do ensino fundamental (6º ao 9º ano) e médio (1º, 2º e 3º anos), uma vez que, para os alunos de tais cursos, a probabilidade dos requisitos serem supridos é maior.

Com isso, espera-se que o alcance das metas propostas contribua para melhorar o ensino da educação à distância (EAD), superar as limitações e barreiras impostas pela distância entre aquele que leciona e aquele que consome e promover um ambiente educacional mais eficiente, inclusivo e de qualidade.

1.2.1 Objetivos Específicos

Aliado ao montante da produção de uma solução auxiliadora do processo de aprendizagem, tal projeto também foi tido como objeto para aplicabilidade de diversos conhecimentos adquiridos durante toda a realização da graduação. Alguns importantes temas que foram visados são:

- Técnicas de linguagens de programação: desenvolvimento de estruturas de dados, manipulação de JSON (*JavaScript Object Notation*) para comunicação entre diferentes pontos, polimorfismo e programação orientada a objetos, dentre outros;
- *Design* e experiência do usuário: uso de técnicas sobre a construção de *softwares* modernos e amigáveis aos usuários finais;
- Modelagem de bancos de dados: construção de modelos de banco para o determinado desenvolvimento;
- Utilização de metodologias ágeis de desenvolvimento: gerenciamento das tarefas realizadas por meio do modelo Scrum ([SCRUM](#), 2023), que possibilita a confecção do sistema de forma iterativa e incremental. Tal método fornece uma abordagem flexível e colaborativa para gerenciamento através da divisão do projeto em uma série de iterações chamadas *sprints*, sendo períodos limitados durante os quais um trabalho específico (*task*) é concluído.

1.3 Justificativa

Conforme aponta a Pesquisa Nacional por Amostra de Domicílios relativa ao ano de 2021 (PNAD, 2021), cerca de 84,7% dos brasileiros com mais de 10 anos já possuem acesso à *internet*. Enquanto a TIC Domicílios (CETIC.BR, 2021), divulgada em agosto de 2021, demonstra um aumento de mais de 15% no número de computadores (*desktop*, portátil ou *tablet*) presentes nos domicílios analisados entre os anos de 2019 e 2020. Os dados supracitados evidenciam que o acesso às tecnologias vem ampliando-se cada vez mais cedo na vida dos brasileiros. Inserindo tais dados no contexto da EAD, fica claro o potencial de impacto do ensino remoto ou híbrido na vida dos estudantes. Sobretudo aqueles presentes no ensino fundamental II ou séries superiores.

Segundo o resumo executivo da Pesquisa TIC Educação (CETIC.BR, 2020), as soluções utilizadas durante a pandemia de COVID-19 para lecionar mesmo com a imposição do distanciamento social foram diversificadas. Aqui cabe destacar que muitas dessas formas de propagar o conhecimento ganharam notoriedade e permaneceram mesmo nos tempos pós-pandemia.

Nesse contexto, evidencia-se a utilização de diversas ferramentas presentes na *web* para estruturar todo o ecossistema que as instituições de ensino estão inseridas: desde a ministração e consumo dos cursos, até a administração dos docentes, discentes e suas turmas. Tal forma de estruturação acaba por desacelerar os processos presentes nesse âmbito, uma vez que impõe a necessidade de aprendizado e adaptação às múltiplas plataformas, cada qual com suas peculiaridades. Portanto, fica clara a necessidade de centralizar, quando possível, esses mesmos processos, para promover uma aceleração nos procedimentos do cenário educacional, uma vez que a maioria do esforço dos usuários seriam voltados à geração de qualidade do conteúdo desenvolvido e não à adaptação aos diferentes aplicativos mediadores.

Ainda segundo essa mesma pesquisa TIC Educação, cerca de 93% dos gestores de escolas apontaram que os pais ou responsáveis dos alunos não estavam preparados para apoiá-los em suas jornadas de aprendizado à distância. Tal fato evidencia que a plataforma usada como meio para consumo dos cursos pelos alunos precisa estar vinculada a uma curva de aprendizado não acentuada. De forma que os usuários finais possam desenvolver rapidamente um senso de independência na utilização do aplicativo, dispensando a necessidade de estarem acompanhados por terceiros a todo momento.

2 Revisão Bibliográfica

A pandemia da COVID-19 trouxe uma série de desafios aos modelos tradicionais de educação, com mudanças significativas impulsionadas pelo distanciamento social imposto. Essas mudanças ajudaram a expandir e consolidar as iniciativas de integração das tecnologias digitais às salas de aula virtuais, tema que vem sendo discutido nas políticas públicas desde a década de 1980 (VALENTE; ALMEIDA, 2020).

Na pesquisa realizada por Isabel Fialho, Marília Cid e Marcelo Coppi, professores e alunos de uma escola foram questionados sobre os tipos de problemas e dificuldades decorrentes da quarentena devido à COVID-19. Os alunos apontaram problemas e dificuldades mais relacionados com questões técnicas sobre a utilização de plataformas digitais e similares. Os mesmos concordam com as vantagens e as possíveis contribuições do uso das plataformas (FIALHO; CID; COPPI, 2023).

Nesse caso, a busca por métodos mais eficientes de ensino à distância leva a uma grande demanda por *softwares* que facilitem esse processo. Segundo a pesquisa TIC Educação 2020, foram adotadas diversas soluções pelas escolas, como a utilização de grupos em aplicativos ou redes sociais por 91% das instituições, a gravação de aulas em vídeo e sua disponibilização para os alunos em 79%, a realização de aulas à distância por meio de plataformas *online* em 65% das escolas, o envio de atividades e materiais por *e-mail* em 60% e o uso de plataformas virtuais em 58% (VALENTE; ALMEIDA, 2020).

Entretanto, é preciso ressaltar que o uso das redes sociais como uma ferramenta de ensino só pode ser eficaz se o ambiente dos alunos for adequado a elas. As redes sociais são espaços de convívio e aprendizagem, mas também podem ser uma fonte de distração, tornando o uso indevido dessas ferramentas como um impacto negativo na vida escolar dos alunos (RODRIGUES; BEZERRA, 2018).

Além disso, as expectativas de desempenho e o esforço necessário para utilizar a plataforma de ensino são fatores relevantes para a permanência dos alunos nos cursos à distância. Estudos mostram que alunos com menos experiência em cursos *online* podem ter mais dificuldade em aceitar plataformas com curvas de aprendizagem complexas (CHIU; WANG, 2008).

Além disso, conforme demonstra a pesquisa realizada em diversas regiões brasileiras no ano de 2023 por Ueudison Alves Guimarães, Roberto Carlos Farias de Oliveira, Maria Diva Silva Gomes, Davi Oliveira da Cruz e Leonardo Henrique Santos Mello o tópico que mais representou obstáculos aos estudantes no regime de EAD foi a dificuldade em manter a disciplina e motivação para estudar. Essa complicação foi alegada por cerca de 37,5% da totalidade dos estudantes (GUIMARAES et al., 2023).

Diante dessas questões, fica clara a necessidade do uso de plataformas instrucionais que proporcionem um ambiente centralizado e controlado, principalmente para escolas onde os alunos possuem pouca ou nenhuma experiência em *e-learning*. Este ambiente centralizado precisa dar foco a questão da facilidade de uso a fim de permitir o direcionamento dos esforços dos alunos para suas motivações e não para a complexidade da plataforma. Além de que, a aplicação também deve permitir que os usuários finais acessem todos os materiais e recursos necessários a partir de uma única fonte, eliminando assim a necessidade de aprender a utilizar diferentes ferramentas digitais ou recorrer às redes sociais para se integrar no ambiente de ensino (RODRIGUES; BEZERRA, 2018).

2.1 JSON Web Token (JWT)

JSON Web Token (JWT) (JWT, 2023) é um padrão aberto para a criação de *tokens* de acesso seguro e autenticação entre sistemas. Ele é usado para transmitir informações confiáveis entre duas partes de forma compacta e autônoma.

Um JWT é composto por três partes separadas por pontos: o cabeçalho (*header*), a carga útil (*payload*) e a assinatura (*signature*). O cabeçalho contém informações sobre o tipo do *token* e o algoritmo de criptografia usado para a assinatura. O *payload* contém os dados que deseja-se transmitir, como informações do usuário ou detalhes da autenticação. A assinatura é usada para verificar a integridade do *token* e garantir que ele não tenha sido alterado durante a transmissão.

2.2 RESTful

RESTful (RESTFULAPI, 2021) é um estilo de arquitetura que define um conjunto de princípios e restrições para projetar serviços da *web*. A palavra REST significa *Representational State Transfer*, ou seja, Transferência de Estado Representacional em português. Tal estilo foi proposto por Roy Fielding em sua dissertação de doutorado em 2000 (REDHAT, 2020).

Os serviços da *web* RESTful são baseados no protocolo HTTP (*Hypertext Transfer Protocol*) e usam métodos como: GET, POST, PUT e DELETE para manipular recursos por meio de URLs (*Uniform Resource Locators*). Esses recursos podem ser qualquer coisa que possa ser identificada por um URI (*Uniform Resource Identifier*), como um objeto, documento ou coleção de dados.

2.3 Framework

Um *framework* (NORONHA, 2023) é um conjunto de ferramentas, bibliotecas e componentes de *software* que fornecem uma estrutura para o desenvolvimento de um aplicativo ou sistema. Os *frameworks* geralmente incluem um conjunto de regras, diretrizes e convenções que ajudam os desenvolvedores a estruturar o código, lidar com tarefas comuns, interagir com outros componentes e resolver problemas específicos.

2.4 JavaScript

JavaScript (DEVDOCS, 2023a) é uma linguagem de programação amplamente utilizada para desenvolvimento de aplicações *web*. Ela foi originalmente criada para adicionar interatividade e dinamismo às páginas da *web*, permitindo que os desenvolvedores criem recursos interativos e adicionem comportamento aos elementos HTML (*Hypertext Markup Language*) (MOZILLA, 2023).

Essa linguagem de programação é classificada como de alto nível, interpretada e orientada a objetos. Ela é executada naturalmente no lado do cliente, ou seja, no navegador do usuário, permitindo a manipulação do conteúdo da página, a interação com o usuário e a comunicação com o servidor.

Além de seu uso generalizado no desenvolvimento da *web*, o *JavaScript* também é usado em outros ambientes, como dispositivos móveis, *desktop* e até mesmo no desenvolvimento de aplicativos de servidor.

2.5 TypeScript

TypeScript (DEVDOCS, 2023b) é uma linguagem de programação de código aberto desenvolvida pela *Microsoft*. Essa, por sua vez, se configura como uma extensão da linguagem *JavaScript*, que adiciona recursos de digitação estática, ou seja, permite definir os tipos de dados de variáveis, parâmetros de função e retornos de funções (XP, 2022c).

Ao contrário do *JavaScript*, que possui tipagem dinâmica, o *TypeScript* apresenta a capacidade de definir explicitamente os tipos de dados durante o desenvolvimento. Além de facilitar o trabalho em equipe, isso traz benefícios significativos como detecção de erros em tempo de compilação, melhor organização e manutenção do código. O *TypeScript* é compilado para *JavaScript* puro, o que significa que o código *TypeScript* compilado é convertido em código *JavaScript* compatível com todos os navegadores e ambientes que suportam tal linguagem (XP, 2022c).

2.6 Node

Node.js, também conhecido como *Node*, é um ambiente de tempo de execução (*runtime*) de código aberto que permite executar *JavaScript* fora de um navegador. Ele é construído sobre o motor V8 do *Google Chrome* e fornece recursos para desenvolver aplicativos de servidor escaláveis e de alto desempenho (MELO, 2021).

Ao contrário do *JavaScript* tradicional, que é executado pelo navegador, o *Node.js* permite que o *JavaScript* seja executado no servidor. Ele usa um modelo assíncrono e orientado a eventos, o que significa que pode lidar com muitas conexões simultâneas sem bloquear o processo principal.

Um dos principais recursos do *Node.js* é sua capacidade de usar E/S (Entrada/Saída) sem bloqueio. Isso ocorre através do uso de retornos de chamada e APIs assíncronas para permitir que outras operações sejam executadas enquanto aguardam os resultados, em vez de aguardar a conclusão de uma operação de E/S antes de passar para a próxima operação. Isso torna o *Node* muito eficiente e escalável, pois pode lidar com várias solicitações simultaneamente sem exigir *threads* (processos de um programa) adicionais.

2.7 Express

O *Express.js* (EXPRESS, 2023) é um *framework* rápido e minimalista para o desenvolvimento de aplicativos da *web* em *Node.js*. Ele fornece uma camada de abstração sobre o núcleo do *Node.js* que simplifica a criação de servidores e a manipulação de rotas, solicitações e respostas HTTP (XP, 2022a).

Com o *Express.js*, os desenvolvedores podem criar aplicativos da *web* com eficiência, definindo rotas para diferentes URLs e especificando como o servidor deve responder a cada uma. Também é conhecido por sua simplicidade e flexibilidade. Ele permite que os desenvolvedores criem sistemas de forma modular, com uma estrutura de código organizada e reutilizável.

2.8 Flutter

Flutter (FLUTTER, 2023) é um *framework* de código aberto criado pelo *Google* para otimizar e facilitar o desenvolvimento de aplicações *web*, *mobile* e *desktop*. Sua linguagem de programação subjacente é o *Dart* (DART, 2023). Com ele, aplicativos para *Android*, *iOS*, *Windows* e *Linux* podem ser desenvolvidos em uma única base de código, o que otimiza o processo e facilita a manutenção (XP, 2022b).

2.8.1 Dart

Dart (DART, 2023) é uma linguagem de programação desenvolvida pelo *Google* que surgiu em 2011, projetada para criar aplicativos de alto desempenho para a *web*, *desktop* e dispositivos móveis. Ela foi criada visando fornecer uma alternativa moderna e eficiente para o desenvolvimento de *software*.

É uma linguagem de tipagem estática, o que significa que as variáveis têm seus tipos definidos em tempo de compilação. No entanto, ela também possui recursos de inferência de tipo, o que permite que o compilador deduza o tipo de uma variável com base no contexto. O que permite que o código criado rode tanto no lado do cliente quanto no servidor (BARRO, 2023).

2.9 Trabalhos Relacionados

Ao realizar a revisão de trabalhos relacionados é possível identificar produtos presentes no mercado com abordagens semelhantes. Ao passo que também existem lacunas e contribuições relevantes a serem exploradas para cada um desses produtos.

Segundo os dados apresentados no portal G2 (G2, 2023), o sistema de Gestão de Aprendizagem com maior pontuação no G2 Score (G2, 2023), que se configura como um índice de classificação de produtos e fornecedores com base em avaliações coletadas de comunidades de usuários, fontes *online* e redes sociais, é o *Google Classroom* (GOOGLE, 2023). Ao mesmo tempo, para as plataformas *open-source*, as duas que mais se destacam, conforme o mesmo índice, são: *Canvas LMS* (LMS, 2023a) e *Moodle* (MOODLE, 2023). Dessa forma, ao analisar e comparar essas plataformas é possível listar algumas de suas funcionalidades:

1. **Moodle:** Essa opção consta com uma das maiores equipes de contribuidores de código aberto em todo o mundo. O *Moodle* chega com um conjunto completo de recursos que permitem não apenas as empresas, mas também educadores, criar um espaço de aprendizagem *online*, privado, repleto de ferramentas que facilitam a criação de cursos e inúmeras atividades;



Figura 1 – Exemplo de estruturação de curso da plataforma *Moodle*

Fonte: (MOODLE, 2023).

Suas principais funcionalidades são: gerenciamento de cursos e turmas, módulo de criação e disponibilização de materiais educacionais, fóruns de discussão, ferramentas de comunicação, atividades interativas e qualitativas, avaliação do progresso dos alunos e recursos de relatórios e análise de dados.



Figura 2 – Tipos de atividades e recursos disponíveis na plataforma Moodle

Fonte: (MOODLE, 2023).

2. **Canvas LMS:** O *Canvas LMS* é uma plataforma de aprendizagem *online* amplamente adotada em instituições de ensino de todos os níveis. O *Canvas* possui uma interface amigável, recursos avançados de colaboração e ferramentas de avaliação.

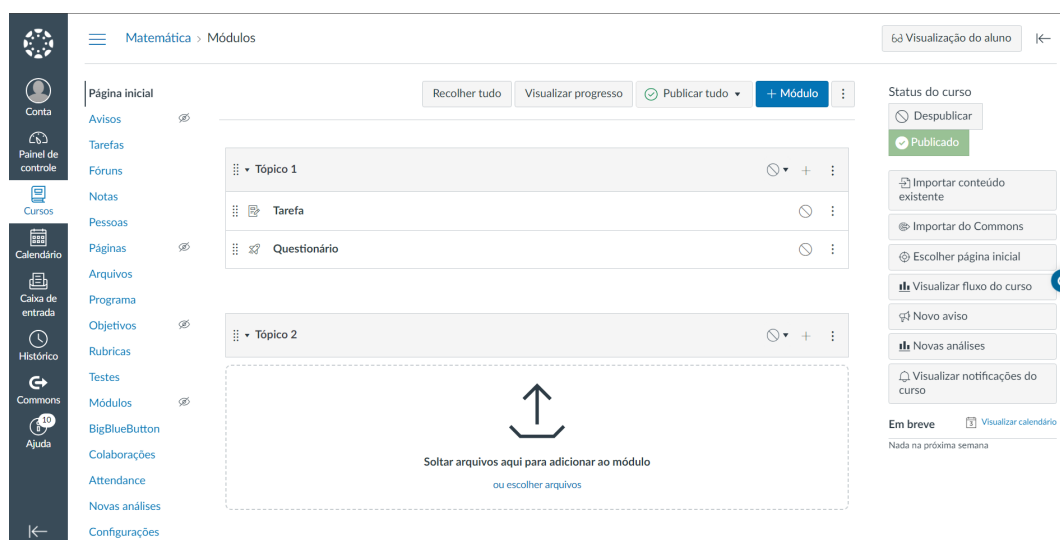


Figura 3 – Exemplo de página da plataforma *Canvas*

Fonte: (CANVA, 2023).

Suas principais funcionalidades são: gerenciamento de cursos e turmas, criação e compartilhamento de materiais educacionais, comunicação entre alunos e professores, calendário para organização de atividades e prazos, avaliação e *feedback* de desempenho dos alunos, ferramentas para monitorar o progresso dos estudantes e personalização da interface e configurações de privacidade.

3. **Google Classroom:** O *Google Classroom* é uma plataforma de aprendizagem virtual desenvolvida pelo *Google* e projetada para facilitar a comunicação e a colaboração entre professores e alunos. Ele oferece recursos como criação e distribuição de tarefas, compartilhamento de materiais, avaliações *online* e comunicação em tempo real. Com uma interface intuitiva e integrada aos aplicativos do *Google*, essa solução visa simplificar o gerenciamento de turmas e promover a interação e o engajamento dos alunos no ambiente virtual de aprendizagem.

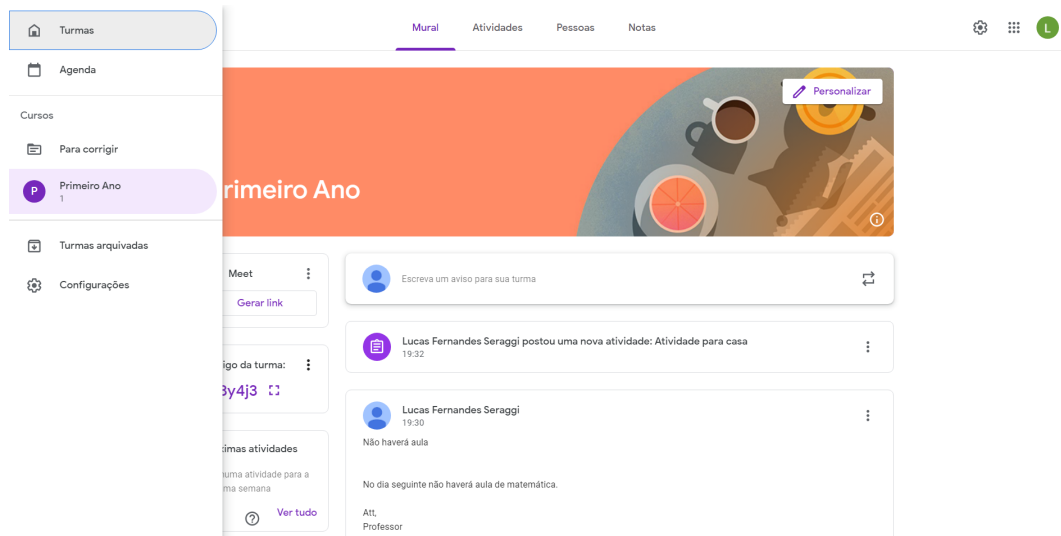


Figura 4 – Exemplo de página principal da plataforma *Google Classroom*.

Fonte: (GOOGLE, 2023).

A plataforma *Google Classroom* possui várias funcionalidades que visam facilitar a organização, comunicação e interação entre professores e alunos. Entre elas, pode-se citar: criação de turmas virtuais, compartilhamento de materiais, tarefas e atividades digitais, além da simplicidade e organização da plataforma.

Objetivando realizar a comparação entre os trabalhos similares, foram utilizadas as versões publicadas mais recentemente dos produtos *Google Classroom* e *Canvas LMS*. Já quanto ao *Moodle*, a versão 4.11 foi tida como base, uma vez que é a mesma presente no ambiente analisado, o qual está hospedado no portal *Moodle Teste* (UNIFEI, 2023) da UNIFEI. Dessa forma, ao relacionar as informações relativas a cada um dos produtos similares, é possível gerar os dados comparativos sobre implementação, presentes na Tabela 1.

Implementação			
Critérios	<i>Google Classroom</i>	<i>Canvas LMS</i>	<i>Moodle</i>
Licença	Proprietário	<i>Open Source</i>	<i>Open Source</i>
Destinatários	Todo tipo de aluno	Todo tipo de aluno	Todo tipo de aluno
Acessibilidade	Via <i>web</i> . As telas são responsivas para funcionarem tanto em dispositivos <i>desktop</i> , quanto dispositivos móveis	Os serviços são prestados através da <i>web</i> e todo o portal é responsivo conforme as dimensões das telas	Via <i>web</i> . A interface é parcialmente responsiva, de forma que se perde a referência de certos elementos ao redimensionar as páginas
Integração com outras plataformas	Possui vasta coleta de serviços da proprietária	Oferece ferramentas para integração de <i>Apps</i> externos por meio do <i>framework</i> LTI (LMS, 2023b)	Engloba o conceito de modularidade, no qual existem uma série de <i>plugins</i> que podem ser acoplados ao produto, para fornecer serviços não nativos (MOODLE, 2022)

Tabela 1 – Comparação entre os produtos similares: implementação

Partindo para o fator de ambiente de utilização, foi gerada a Tabela 2 de comparação.

Ambiente de utilização				
Critérios	<i>Google Classroom</i>	<i>Canva LMS</i>	<i>Moodle</i>	
Organização dos componentes nas telas	Possui uma identidade visual bastante característica e sucinta, na qual os elementos são dispostos de forma a gerar destaque para funcionalidades isoladas	Muitos elementos são disponibilizados em cada uma das páginas. Isso acaba por atribuir múltiplas <i>features</i> a poucas telas, porém também amplia o nível de complexidade de uso da plataforma	Existem diversos elementos em cada uma das telas, porém dispostos juntamente a ícones amigáveis ao usuário	
Temas	Torna-se necessária a instalação de extensões para realizar a mudança do tema padrão, o qual seria somente o <i>light mode</i> , no qual as cores possuem baixo contraste	Por padrão possui somente o <i>light mode</i> , sendo necessário usar ferramentas externas para realizar alterações	Consta somente com o <i>light mode</i> , porém temas de terceiros podem ser aplicados para alteração da paleta de cores vigente	

Tabela 2 – Comparação entre os produtos similares: ambiente de utilização

No que diz respeito ao consumo de serviços ofertados pelas 3 soluções dispostas, a quantidade de passos para se realizar determinada ação é bastante similar em cada uma delas. A título de exemplo, pode ser verificada na Figura 5 a sequência de passos que precisam ser realizados para inserção de um recurso dentro do módulo de um curso.

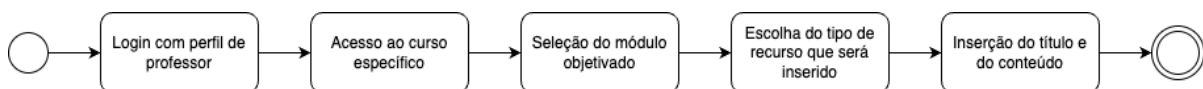
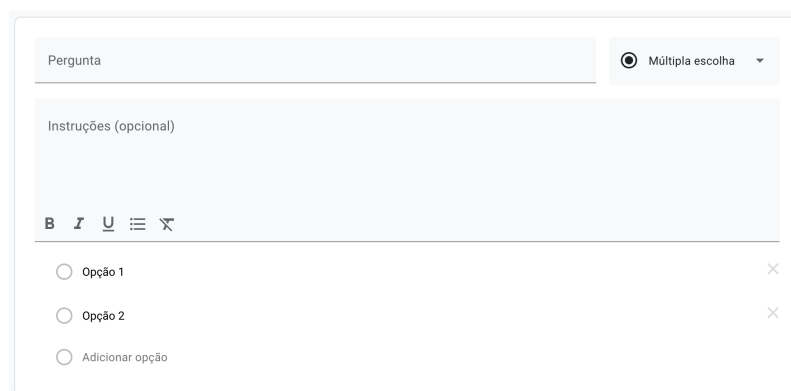


Figura 5 – Diagrama de sequência para inserção de recurso

Fonte: De autoria própria.

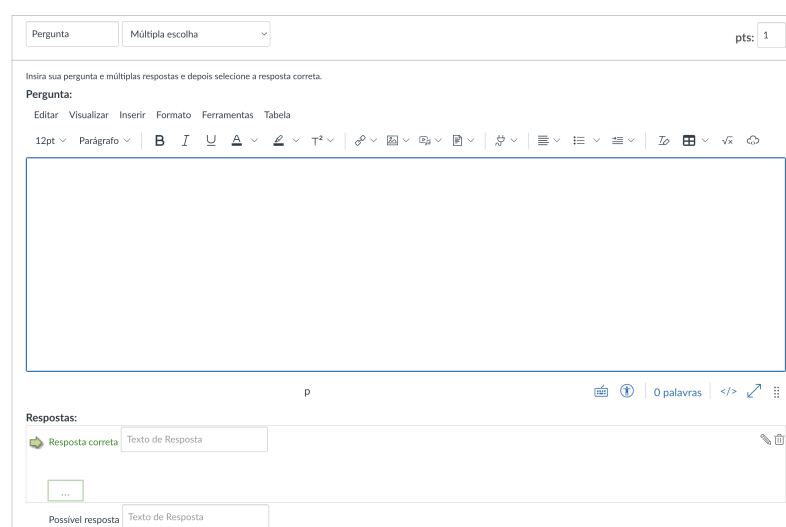
Ainda nessa temática de inserção de conteúdos nos módulos dos cursos, é perceptível algum nível de similaridade de design atribuído a cada um dos trabalhos similares, conforme é exemplificado nas Figuras 6, 7 e 8.



The screenshot shows the Google Classroom question creation interface. At the top, there is a 'Pergunta' (Question) label and a dropdown menu set to 'Múltipla escolha' (Multiple choice). Below this is a large text area for 'Instruções (opcional)' (Optional instructions) with a rich text editor toolbar. At the bottom, there are three radio button options: 'Opção 1', 'Opção 2', and 'Adicionar opção' (Add option). Each option has a small 'x' icon to its right for removal.

Figura 6 – Tela de inserção do recurso questionário no *Google Classroom*

Fonte: (GOOGLE, 2023)



The screenshot shows the Canvas question creation interface. At the top, there is a 'Pergunta' (Question) label and a dropdown menu set to 'Múltipla escolha' (Multiple choice). The points value is set to 'pts: 1'. Below this is a large text area for the question with a rich text editor toolbar. At the bottom, there is a 'Respostas:' (Answers) section with a 'Resposta correta' (Correct answer) field and a 'Possível resposta' (Possible answer) field. The 'Resposta correta' field has a green checkmark icon and the text 'Resposta correta'.

Figura 7 – Tela de inserção do recurso questionário no *Canvas*

Fonte: (LMS, 2023a)

Figura 8 – Tela de inserção do recurso questionário no *Moodle*

Fonte: ([UNIFEI, 2023](#))

Tendo em vista os pontos levantados durante essa comparação, é evidente que as soluções já presentes há anos no mercado constam com uma grande quantidade de funcionalidades. Todavia, ao limitar os escopos das mesmas para estarem conforme os requisitos de sistema da solução proposta nesse trabalho, é possível explorar alguns pontos:

- Curva de aprendizado: os produtos *Canvas* e *Moodle* constam com múltiplas opções conglomeradas em poucas telas. Isso ocasiona no aumento da dificuldade de utilização, já que prejudica a simplicidade da plataforma;
- Identidade dos temas: todas as soluções citadas fazem uso somente do *light theme*;
- Experiência do Usuário: como comentado anteriormente, as soluções *Canvas* e *Moodle* constam com diversas informações condensadas. Tal fato também acaba por comprometer a experiência do usuário, que pode se encontrar com situações de indecisão em meio a tantas opções.

Ademais, também é preciso ressaltar a ferramenta BlackBoard ([BLACKBOARD, 2023](#)) de aprendizado *online*, que prestou serventia como inspiração para a construção do sistema elaborado durante a execução do presente projeto.

3 Metodologia

3.1 Procedimento técnico

De modo a alcançar a resolução dos problemas presentes na EAD, primeiramente, foram levantados os principais requisitos para a criação do sistema e como o mesmo poderia ser utilizado no cotidiano dos usuários finais. Posteriormente foi projetada uma possibilidade de solução a qual poderia ser aplicada em um ambiente escolar.

Analizando o ponto de vista da natureza, o presente trabalho pode ser classificado como uma pesquisa exploratória aplicada, uma vez que procura-se obter *insights* e possíveis ideias para alcançar a melhoria da eficiência do ensino híbrido ou totalmente à distância. Já do ponto de vista técnico esse trabalho pode ser classificado como sendo um estudo de caso, no que se refere a seleção de ferramentas que melhor se enquadrassem para auxiliar no processo de criação da plataforma final.

As metodologias utilizadas consistem em contemplar e analisar informações sobre tecnologias modernas de desenvolvimento ofertadas gratuitamente, a fim de estudar aspectos relacionados ao tema de especificação e criação de *softwares* voltados para a área educacional.

Nesse método foi buscada a aplicação prática dos conhecimentos para gerar a solução proposta que resultou no desenvolvimento de um portal *web*.

3.2 Sistema

A abordagem adotada para construir a plataforma escolar é baseada na metodologia de sistemas *web*, que consiste em duas áreas principais: *back-end* e *front-end*, além do banco de dados.

3.2.1 Banco de dados

Inicialmente, foram discutidas as principais soluções de bancos de dados ofertados pelo mercado. Com base na pesquisa de popularidade publicado pelo site *DB-Engines* ([DB-ENGINES, 2023](#)), foram destacadas as seguintes possibilidades de utilização: *Oracle*, *MySQL*, *Microsoft SQL Server*, *PostgreSQL* e *MongoDB*;

Dentre essas opções, foi dado foco para os *open source databases*, uma vez que eles melhor se encaixavam com o propósito do presente trabalho devido às suas características intrínsecas ([PERCONA, 2023](#)):

1. **Custo-benefício:** os bancos de dados de código aberto ofertam soluções gratuitas para uso e distribuição, tornando-os uma opção econômica;
2. **Suporte e desenvolvimento da comunidade:** possuem comunidades vibrantes de desenvolvedores e usuários que contribuem ativamente para seu aprimoramento, uma vez que todo desenvolvedor pode personalizar o código-fonte. Essas comunidades oferecem suporte, documentação, correções de *bugs* e atualizações, garantindo que o banco de dados permaneça robusto e atualizado.
3. **Segurança e transparência:** tais soluções são construídas em um modelo de desenvolvimento transparente. O código-fonte está disponível gratuitamente para inspeção, acelerando as análises através de buscas coletivas por vulnerabilidades, *bugs* de segurança e melhorias de desempenho.

Aliado a isso, foi preciso considerar a comparação entre os modelos de estruturação das opções de bancos, na qual aqueles conforme o padrão SQL adere a um modelo de dados tabular e estruturado e impõem definições de esquema rígidas, enquanto os NoSQL adotam um modelo de dados flexível e com *schemas* dinâmicos, permitindo armazenamento de dados escaláveis e distribuídos (IBM, 2022). Tendo isso como base, o padrão SQL tornou-se mais apropriado para a aplicação em questão, visto que as entidades pensadas sobre o ecossistema escolar poderiam ser modeladas em formato tabular de forma bastante eficiente. Juntamente, é preciso destacar o conceito de garantia de propriedades ACID (IBM, 2023) dos exemplares SQL. Esse conceito serviu como base para consolidação da escolha do banco a ser usado e se configura como um acrônimo para:

- **Atomicidade:** A atomicidade garante que uma transação seja tratada como uma unidade de trabalho única e indivisível. Isso significa que todas as alterações feitas em uma transação são confirmadas ou nenhuma delas. Não há conclusão parcial ou estado intermediário.
- **Consistência:** A consistência garante que uma transação leve o banco de dados de um estado válido para outro. Ela impõe restrições de integridade, como integridade referencial e validações de dados, para manter a precisão e a validade dos dados.
- **Isolamento:** O isolamento garante que as transações simultâneas não interfiram umas nas outras. Cada transação é executada isoladamente das outras, e seus estados intermediários não são visíveis para transações diferentes até que a transação seja confirmada.
- **Durabilidade:** A durabilidade garante que, uma vez que uma transação é confirmada, suas mudanças são permanentes e sobreviverão a qualquer falha subsequente, como falta de energia ou travamento do sistema. Os dados confirmados são armazenados de forma confiável e podem ser recuperados em caso de falha do sistema.

Portanto, respeitando as decisões de uso de *softwares open source* que estivessem consoante ao modelo SQL, as opções restantes eram: *MySQL* e *PostgreSQL*. Nesse ponto o método de escolha passou a ser experiências prévias dos integrantes da equipe vigente. Logo, optou-se pela utilização do banco *PostgreSQL* ([POSTGRESQL, 2023](#)), o qual é conhecido por sua robustez, confiabilidade e adesão aos padrões SQL. Ele oferece recursos avançados, como suporte a JSON e pesquisa de texto completo. Além disso, tal opção de *database* oferece excelente integridade de dados e mecanismos de controle de simultaneidade, tornando-o adequado para sistemas transacionais de alto volume. Sua extensibilidade permite que sejam criados tipos de dados personalizados, operadores e funções, proporcionando flexibilidade e permitindo a implementação de modelos de dados complexos.

Em conjunto ao *PostgreSQL*, para realização do gerenciamento e acesso ao banco, fez-se uso das ferramentas *DBeaver* ([DBEAVER, 2023](#)) e *PgAdmin 4* ([PGADMIN, 2023](#)).

3.2.2 *Back-end*

No que diz respeito ao *back-end*, isto é, as tecnologias que alimentam o lado do servidor da aplicação, foram construídas APIs baseada no padrão RESTful que, por sua vez, consiste na abordagem arquitetural que utiliza princípios e restrições do protocolo HTTP. Essas APIs são responsáveis pela lógica que lida com o processamento de dados, os processos de autenticação, segurança e as regras de negócios impostas.

Seguindo o modelo RESTful, as APIs são projetadas para serem simples, escaláveis, e seguir os princípios da *web*. Elas se basearam no conceito de recursos, sendo entidades identificáveis (mediante URLs) com as quais o *client-side* pode interagir. Os recursos são manipulados por meio de operações bem definidas ([RESTFULAPI, 2021](#)), com foco para as principais utilizadas: GET (obter os dados), POST (enviar os dados), PUT (atualizar o recurso) e DELETE (excluir o recurso).

Dentre as características fundamentais das APIs RESTful pode-se citar aquelas que embasaram a construção do servidor:

1. **Estado da aplicação:** A API é projetada para ser *stateless*, ou seja, cada solicitação engloba todas as informações necessárias para ser processada e não há dependência de estado entre as solicitações. Isso simplifica a escalabilidade e a manutenção da API;
2. **Utilização de URIs:** Cada recurso é identificado por uma URI exclusiva, que permite que os clientes acessem e manipulem o recurso por meio de URLs específicas;
3. **Representação de dados:** Os dados são representados em um formato padrão que, nesse caso, foi por meio do formato JSON;

4. **Status Codes adequadas:** As respostas da API incluem códigos de *status* HTTP adequados para indicar o resultado da operação e facilitar a comunicação entre sistemas. Como exemplo existem os *status* 200 para indicar que a transação ocorreu sem erros, 201 para indicar que o recurso foi criado com sucesso ou 404 para indicar que o recurso alvo não existe.

Assim, fica evidente que ao seguir o padrão RESTful, as APIs podem se beneficiar de uma arquitetura bem definida, permitindo uma comunicação eficiente entre clientes e servidores, escalabilidade e interoperabilidade.

Nessa etapa cabe ressaltar como ocorreu o processo de escolha de quais seriam as tecnologias de linguagens de programação e *frameworks* a serem utilizadas para a construção do *server-side*. Dessa forma, segundo a investigação divulgada pelo projeto de pesquisa e análise de dados *Statistics & Data* (STATISTICS; COMPANY, 2023), as 5 opções que mais se destacavam no mercado (segundo o número de estrelas recebidas pelos usuários em seus respectivos repositórios), conforme é demonstrado na Figura 9, são:

- **Laravel:** 71.903 estrelas
- **Django:** 67.941 estrelas
- **Spring:** 64.405 estralas
- **Flask:** 61.331 estrelas
- **ExpressJs** (EXPRESS, 2023): 59.132 estrelas

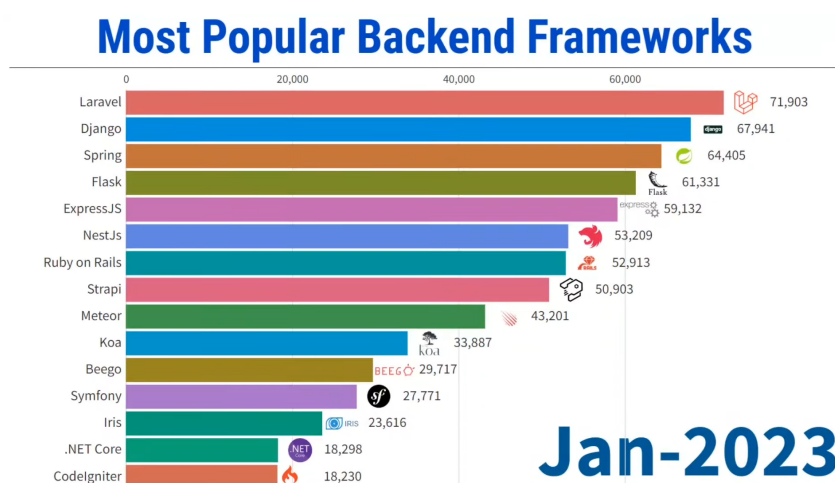


Figura 9 – Gráfico que representa os *frameworks* mais populares usados no *back-end* em janeiro de 2023

Fonte: (STATISTICS; COMPANY, 2023).

Todavia, conforme mostra a plataforma *online* alemã especializada em coleta e visualização de dados: *Statista* ([STATISTA, 2022](#)), a linguagem mais utilizada globalmente pelos desenvolvedores é a *JavaScript* (contando com 65.36% de uso pelos 71.547 entrevistados), como mostrado na Figura 10.

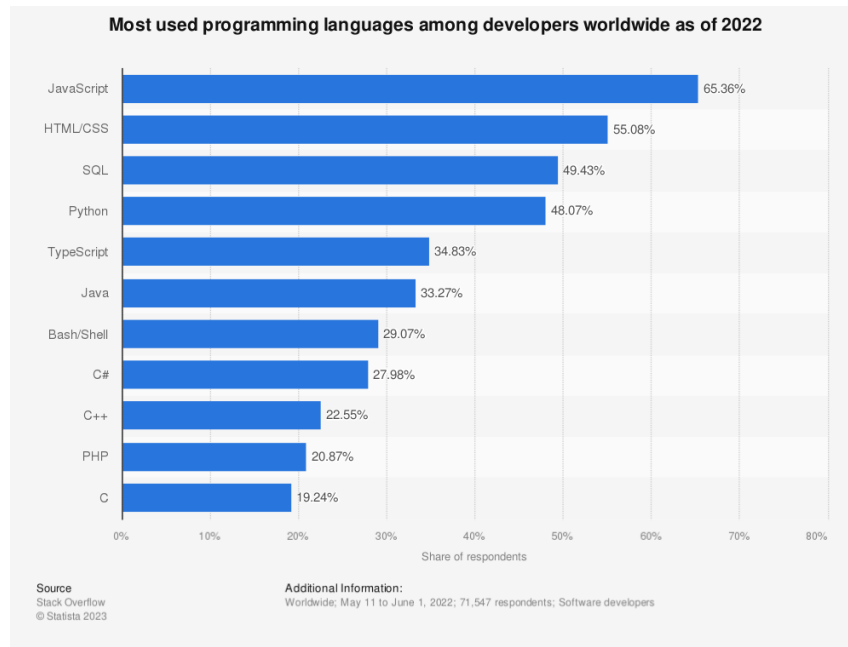


Figura 10 – Linguagens mais utilizadas pelos desenvolvedores em 2022

Fonte: ([STATISTA, 2022](#)).

Ainda nessa análise das opções de mercado é importante também destacar as características intrínsecas às linguagens de programação fortemente tipadas, as quais podem ser englobadas nos seguintes tópicos ([TECHTARGET, 2022](#)):

- Confiabilidade de código aprimorada: a tipagem forte auxilia a detectar erros relacionados ao tipo durante a fase de compilação, reduzindo a probabilidade de erros em tempo de execução. Assim, ela garante que variáveis e dados sejam usadas de maneira consistente e esperada, resultando em um código mais confiável;
- Melhor legibilidade e manutenção: as linguagens que utilizam de tal formato tornam o código autoexplicativo e fácil de entender e manusear;
- Detecção antecipada de erros: A verificação rigorosa de tipos em linguagens fortemente tipadas detecta erros no início do processo de desenvolvimento. Tal fato leva a uma identificação ágil e resolução eficiente de *bugs*;
- Melhor qualidade de código: a tipagem forte ajuda a impor boas práticas de programação, evitando conversões de tipo não intencionais ou incompatíveis. Ela obriga,

a um certo nível, os desenvolvedores a escrever um código mais limpo e confiável, resultando em maior qualidade geral nos algoritmos criados.

Portanto, combinando todos os fatos supracitados, a linguagem *TypeScript* foi a selecionada para desenvolvimento do *back-end*, uma vez que a mesma abrange os conceitos objetivados de tipagem forte e ampla utilização pelos desenvolvedores, o que resulta na ampliação da gama de *frameworks* e bibliotecas existentes para utilização na fase de criação do sistema da aplicação.

A partir de tal escolha, as decisões subsequentes se deram conforme o nível de popularidade entre a comunidade de desenvolvedores.

Logo, as APIs regentes do comportamento do sistema na totalidade foram desenvolvidas usando servidor *Node.js* (NODE, 2023). Enquanto que para intermediar essa camada por meio da abstração de código e promover um desenvolvimento ágil, a *framework Express.js* foi tida como base. Essa última se destacou por fornecer um conjunto minimalista e flexível de recursos para criar o *server-side* de aplicações *web*. O *Express.js* simplifica o processo de manipulação de solicitações HTTP, roteamento, integração de *middleware* e geração de respostas às requisições recebidas.

Por fim, para validar o funcionamento das interfaces da aplicação, o *software Postman* (POSTMAN, 2023) foi usado devido aos seus recursos que possibilitaram a execução de testes de APIs e requisições em geral. Por consequência, ele também prestou serventia como um ambiente de documentação, uma vez que concentra as informações de como as APIs projetadas operam.

3.2.3 Front-End

No *front-end*, dois pontos primordiais foram tidos como base para a definição de qual seria o *framework* utilizado para sua construção: experiências prévias de desenvolvimento, de forma a assegurar maior velocidade na criação do sistema, e conceitos de *cross-platform*, isto é, a capacidade do *client-side* poder ser acessado em diferentes plataformas, incluindo sistemas operacionais e dispositivos com dimensões diferentes. Tendo tais pilares como norte, o *framework Flutter* (FLUTTER, 2023) baseado na linguagem *Dart* (DART, 2023) foi definido como sendo a ferramenta que guiaria essa etapa do desenvolvimento.

Aqui cabe destacar alguns pontos cruciais integrados ao *Flutter* que o tem destacado dentre as várias ferramentas de construção de aplicações existentes (STACKIFY, 2023):

- UI (*user interface*) rápida e reativa: o *Flutter* utiliza uma abordagem exclusiva chamada *widgets* para criar a interface do usuário. Ele engloba uma arquitetura reativa que permite renderização e atualizações rápidas, gerando interfaces de usuário suaves e responsivas;

- *Hot Reload*: o recurso de *hot reload* permite que as alterações feitas no código sejam refletidas imediatamente no aplicativo sem precisar reiniciá-lo;
- Quantidade e personalização de *widgets*: o *Flutter* fornece um conjunto abrangente de *widgets* personalizáveis que permitem criar uma UI visualmente atraente e consistente em diferentes plataformas;
- Desempenho: o mecanismo de renderização do *Flutter*, chamado *Skia*, garante gráficos e animações de alto desempenho. A estrutura foi projetada para aproveitar a aceleração de *hardware* subjacente, resultando em aplicativos suaves e responsivos.
- Documentação de alta qualidade: *Flutter* oferece documentação abrangente, incluindo guias, amostras e referências de APIs.

Inerente a escolha do *framework*, a linguagem *Dart* ([DART, 2023](#)) foi tida como entidade mãe para a construção dessa etapa do projeto, uma vez que ela também é empregada na construção do próprio *Flutter*.

O *design* da interface do usuário foi inicialmente criado pelo *software Figma* ([FIGMA, 2023](#)), permitindo a visualização e o planejamento do visual da plataforma.

3.3 Ambiente de Desenvolvimento

Ambas as áreas (*front-end* e *back-end*) do projeto foram construídas usando o editor de texto *Visual Studio Code* ([VSCODE, 2023](#)), que fornece um ambiente de desenvolvimento eficiente e suporta diversas linguagens de programação.

Já o controle de versionamento do código gerado durante o desenvolvimento foi feito pelo *git* e hospedado através do *GitHub*, permitindo o controle de versão, colaboração e gerenciamento eficiente do projeto.

Essa abordagem foi escolhida por sua eficiência e flexibilidade no desenvolvimento de *web systems*, fornecendo uma estrutura poderosa e extensível para a construção da plataforma escolar. A utilização de ferramentas modernas e amplamente adotadas no mercado facilita o desenvolvimento ágil e eficiente do projeto.

Durante a construção da plataforma, boas práticas de programação como modularização, reutilização de código e testes foram aplicadas para garantir a qualidade e manutenibilidade do produto final.

Por fim, a metodologia adotada possibilitou a criação de uma plataforma escolar centralizada e funcional, atendendo aos objetivos propostos e oferecendo uma experiência de ensino à distância eficiente, intuitiva e de qualidade.

4 Desenvolvimento

O objetivo deste capítulo é demonstrar como foi realizada e estruturada a aplicação *web* para gerenciamento das atividades e conteúdo ministrados em sala de aula.

4.1 Objeto

No âmbito da criação de uma solução facilitadora da educação à distância para as instituições de ensino, o presente trabalho propõe o desenvolvimento de uma aplicação *web* que visa englobar os diferentes papéis existentes no cenário educacional e administrativo de tais instituições. Os principais papéis tidos como base para a criação das funcionalidades pensadas para suprir os requisitos de utilização foram:

- Administrador: tal cargo seria voltado para a conta da instituição em si, de forma que, por meio dela, seria possível realizar a manutenção e cadastro de usuários (alunos e professores) e de turmas, as quais englobam os devidos usuários, agrupando-os;
- Professor: esse cargo, por sua vez, seria o responsável pelo fornecimento dos conteúdos dos devidos cursos ministrados, atribuição de notas e fichas de presença aos discentes, postagem de lembretes e acesso às informações abrangentes sobre turmas e alunos de maneira eficiente e organizada;
- Aluno: por fim, o aluno seria aquele que consumiria os conteúdos ofertados, participando ativamente por meio da entrega das atividades, estudando os conteúdos ofertados e verificando suas notas e presenças.

Objetivando a melhor experiência do usuário com a plataforma, sobretudo no que se refere a facilidade do uso e o *design* moderno, a plataforma deve estar conforme as seguintes técnicas de *design*:

- Apelo visual: a interface deve ser agradável, consistente e atraente. De forma a incentivar o usuário a permanecer no site e transmitir o conceito de facilidade de uso;
- Utilidade: O serviço ofertado deve ser sempre funcional, de acordo com requisitos de cada um dos 3 principais papéis;
- Credibilidade: A plataforma deve estar livre de erros, atender às expectativas e gerar gratificação;

- Natureza intuitiva: A interface do usuário deve garantir usabilidade e ser intuitiva. Os usuários finais devem ser capazes de entender como usar o produto e aproveitá-lo ao máximo sem precisar de suporte.
- Desempenho técnico: Os usuários não devem sentir frustrações ao usar o produto.

4.2 Banco de Dados

Com base nos requisitos levantados para cada um dos perfis de usuários finais, foi gerado o seguinte diagrama que serve como fonte para a construção da relação entre as entidades:

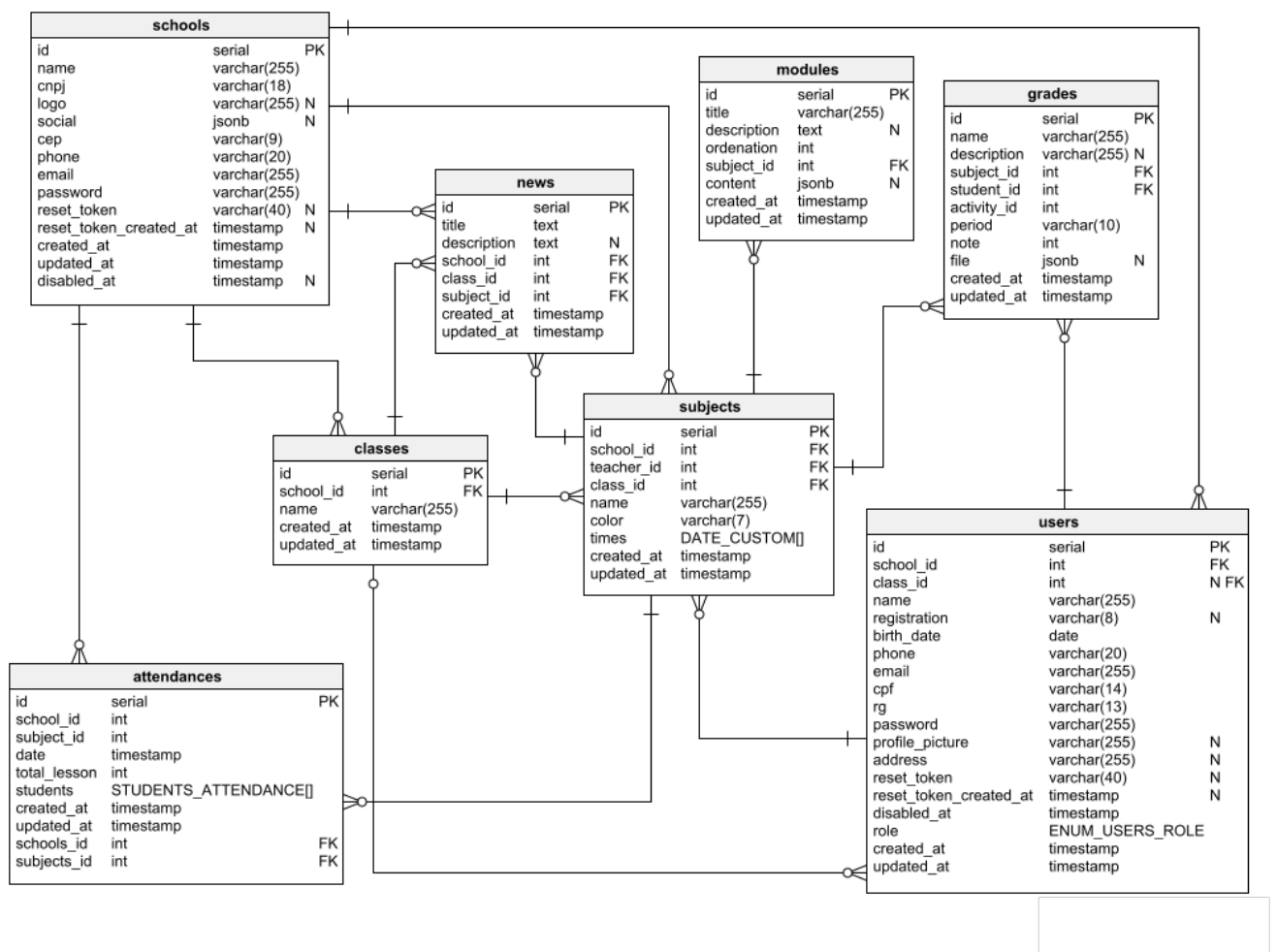


Figura 11 – Diagrama Pé-de-galinha do sistema proposto

Fonte: De autoria própria.

No diagrama presente na Figura 6 é possível identificar, primeiramente, as entidades (representadas pelas tabelas) e as suas relações (representadas pelas conexões entre as

tabelas) que compõem o ambiente escolar no qual o sistema está introduzido. Nesse contexto as regras de negócio são inseridas para a regência e definições de comportamentos das entidades:

- *Schools*: essas entidades seriam consideradas as bases para a construção das demais, uma vez que representam as escolas como instituições, sendo que somente é possível construir os ecossistemas escolares, quando essas entidades já estiverem inseridas na base de dados;
- *Users*: aqui se encaixam os professores e os alunos que, no modelo construído, possuem permissões de visualização similares, porém diferem nas permissões de criação, edição ou exclusão de recursos. Dessa forma, o sistema utiliza o campo *role* justamente para realizar a distinção entre esses dois tipos de perfis;
- *Classes*: engloba as turmas, análogas ao que seriam os 1ºs, 2ºs e 3ºs anos do ensino médio, por exemplo. Esse tipo de entidade é usado para agrupar os professores e alunos que fazem parte de anos escolares equivalentes. O mesmo ocorre para as disciplinas (*subjects*) vinculadas a determinada turma;
- *Subjects*: correlatas ao que seriam as disciplinas, as *subjects* são utilizadas para associar, pela devida temática, as demais entidades do ambiente escolar para promover os respectivos serviços atrelados ao curso;
- *Modules*: são as subdivisões presentes nas *subjects*, de forma que os cursos possam ser melhor organizados mediante tópicos de conteúdos bem definidos;
- *News*: compõem as entidades usadas como lembretes e avisos aos alunos ou como base de informações aos professores sobre eventos ocorrentes durante o período de formação escolar;
- *Grades*: aqui se encontram os conteúdos avaliativos, ou seja, aqueles que possuem o objetivo de instigar a realização de algum exercício por parte do aluno;
- *Attendances*: essas entidades são análogas às listas de chamada, isto é, são objetos que armazenam as informações de presença de discentes em certas datas.

4.3 Back-end

Como já mencionado, foi anteriormente utilizado um servidor *Node.js* ([NODE](#), 2023) com *Express.js* ([EXPRESS](#), 2023) para servir o *front-end* da aplicação e fornecer APIs com rotas seguras e chamadas rápidas.

4.3.1 Segurança

Para garantir acesso seguro a todos os usuários e escalabilidade da plataforma optou-se pela utilização do padrão JWT ([JWT, 2023](#)). Por meio dele, após a validação do *e-mail* e senha no ato da realização do *login*, a API gera um *token* que contém as principais informações do usuário que acabou de acessar o sistema. Algumas informações seriam: id de usuário, id da respectiva escola e a data de expiração desse *token*. Após isso, sempre que o usuário fizer qualquer requisição ao servidor, o *token* será validado, e em caso positivo a requisição será realizada.

4.3.2 Middlewares

Middlewares são funções intermediárias que podem ser executadas antes e em qualquer momento entre o recebimento de uma requisição até a entrega de uma resposta. Os *middlewares* tem como o principal propósito a autenticação, autorização, validação dos dados e geração de *logs* ([REDHAT, 2023](#)).

Desde modo, com exceção da rota de *login* e de cadastro da entidade escola, as demais requisições passam, primeiramente, por um *middleware* de autenticação que valida o *token* JWT e coloca todas as informações pertinentes dentro do *header* da requisição, para serem utilizadas posteriormente. Em casos de *token* inválido ou expirado é retornada uma mensagem de erro com o *status code* 401.

Após autenticado, o próximo passo é verificar se o usuário logado tem permissão para realizar aquela operação. No sistema existem 3 níveis de autorização, sendo eles: escola, professor e estudante. Tendo a autorização necessária, a ação solicitada é executada, caso contrário é retornada uma mensagem de erro juntamente ao *status code* 403.

4.3.3 Model

Além de englobar todos os atributos de uma entidade que está presente no banco de dados, os modelos se configuram como uma metodologia eficiente de manipulação de dados dos recursos da aplicação, uma vez que, os mesmos são construídos para conter, em seus métodos, um CRUD completo da entidade.

4.3.4 Router

Os *routers* são um mapeamento de todas as rotas da API. Eles englobam chamadas para suas respectivas entidades de *controller*. Além disso, podem conter chamadas para *middlewares* específicos.

4.3.5 Controller

Dentro do *controller* a solicitação é interpretada e validada. Posteriormente a isso, normalmente realiza-se uma interação com o respectivo modelo, cujo objetivo é retornar os dados referentes a consulta. Em seguida, tais dados são tratados e, por fim, são retornados ao usuário solicitante.

4.3.6 Services

Com o intuito de realizar uma tarefa específica que pode ser utilizada em múltiplas áreas do algoritmo, os *services* são classes que realizam uma única função e englobam, internamente, toda a sua regra de negócio. Esse tipo de serviço foi utilizado em três ocasiões no projeto: para acesso direto ao banco de dados, acesso ao serviço de *storage* do *Firebase* e envio de *e-mails*;

A classe *DatabaseConnection*, realiza a configuração inicial e abstrai a conexão com o *database*, desta forma os *models* não precisam se preocupar em abrir ou fechar as conexões com o banco de dados. Para otimizar o processo, é utilizado o conceito de *debounce*, no qual somente após um certo período em inatividade, a conexão com o banco é fechada.

Também foi utilizada a *FirebaseConfig* que realiza a configuração e *login* com os servidores do *Google*, enquanto que a classe *FirebaseStorage* tem a responsabilidade de gerenciar os arquivos do *storage*. E por último, a *EmailSender* para envios de *e-mails* automáticos.

4.4 Front-end

4.4.1 Core

Na pasta *core* estão localizados os arquivos de configurações globais, como as cores e fontes utilizados em todo o projeto, bem como as configurações globais de estilizações de *widgets* nativos do *Flutter* (FLUTTER, 2023).

4.4.2 Model

Assim como no *back-end*, dentro dos *models* construídos nessa etapa, se localizam todos os atributos e métodos que, desta vez, não refletem diretamente no banco. Referentes a esses métodos, *fromMap()* e *toMap()* são exemplos que realizam a ação de converter os dados que transacionam nas chamadas das APIs.

4.4.3 Widgets

Dentro do *Flutter* todos os componentes que compõem a tela se configuram como *widgets*. Esses elementos podem ser segregados em dois tipos: *stateful* (realizam o armazenamento de estados) e os *stateless* (não contem a função de guardar os estados do determinado *widget*). Assim sendo, o repositório do projeto foi estruturado para conter pastas que abriguem todos os componentes utilizados para compor as telas.

Ao dividir a tela em diversos componentes garante-se uma melhor qualidade de código visto que esses componentes poderão ser reutilizados em outras partes do sistema. Além disso, ao dividir em componentes menores também reduz-se a complexidade do código, visto que cada componente passa a ter menos responsabilidades.

4.4.4 Pages

Mesmo sendo compostas apenas por *widgets*, as *pages* são estruturas primárias para a construção de todas as *views* do sistema.

4.4.5 Services

Os *services*, por sua vez, são a única forma da aplicação se comunicar com o mundo exterior via requisições HTTP. Neles são configuradas rotas nas quais se pretende acessar os dados necessários para que as *pages* alcancem seus objetivos de fornecimento dos serviços ao usuário. Nesse contexto é importante citar que tais dados são convertidos em modelos *MAP* ou JSON para poderem ser interpretados corretamente pelo servidor ou pelo *client-side*.

4.5 Deploy

No intuito de disponibilizar o sistema na *internet*, buscaram-se algumas plataformas que oferecem esse serviço de hospedagem de forma gratuita. Para salvar as imagens dos perfis e documentos fez-se uso do *Firebase Storage* ([FIREBASE, 2023a](#)) que consta com até 5GB de armazenamento gratuito e o *Firebase Hosting* ([FIREBASE, 2023b](#)) para hospedar o site. Já no que se refere ao *back-end*, a ferramenta *fly.io* que oferece, de forma gratuita, um núcleo de processamento compartilhado e 256MB de RAM, foi utilizada, bem como o *ElephantSQL* ([ELEPHANTSQL, 2023](#)) para armazenamento do banco SQL.

5 Experimentos e Resultados

Nesse capítulo são apresentadas as funcionalidades disponibilizadas na aplicação *web* desenvolvida.

5.1 Página Inicial

A página inicial é o primeiro contato ao acessar a aplicação *web*. Exemplificada na Figura 12, ela expõe alguns critérios informativos do sistema para que novas instituições possam conhecê-lo e verificar seus pontos fortes.

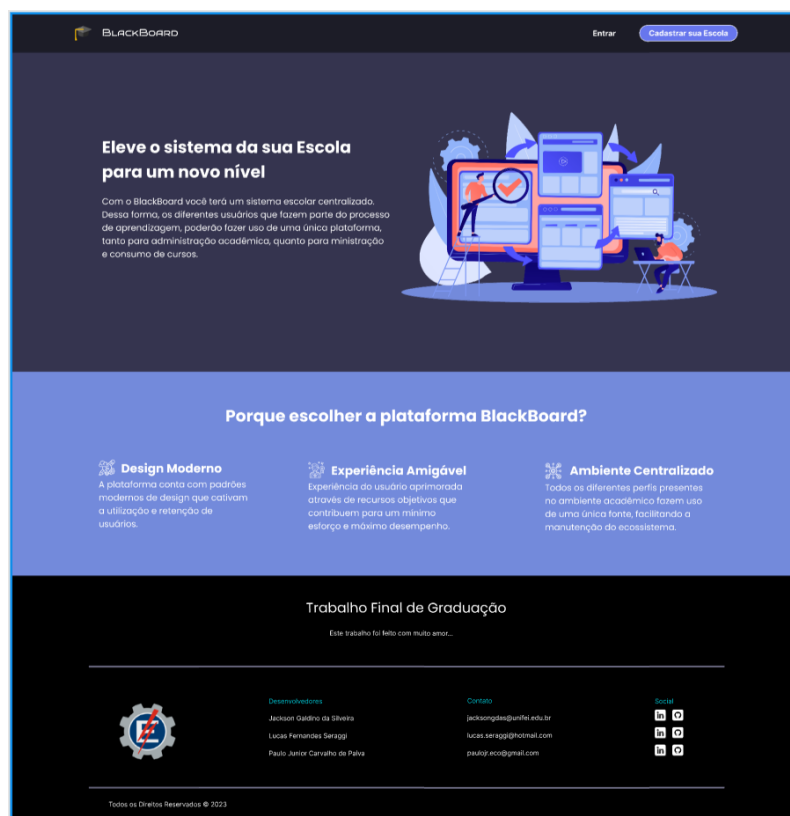


Figura 12 – Página inicial do sistema

Fonte: De autoria própria.

Selecionando o botão de "Cadastrar sua Escola", a página, demonstrada na Figura 13, contendo o cadastro de escola aparecerá, para que, posteriormente, seja iniciada a construção das entidades que compõe o ambiente escolar.

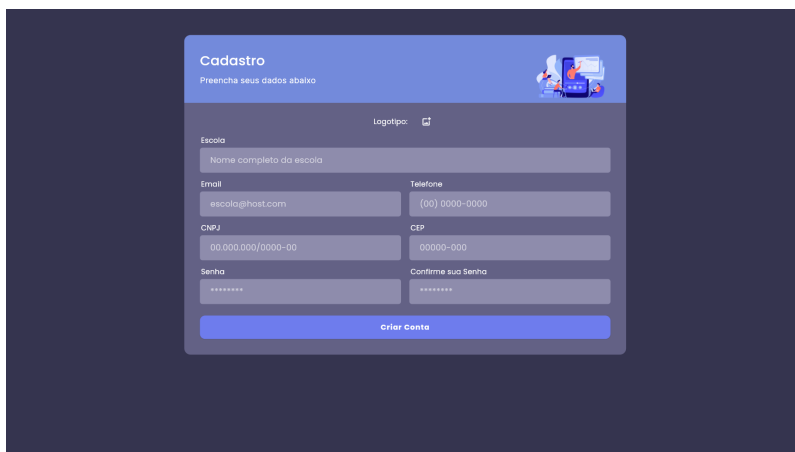
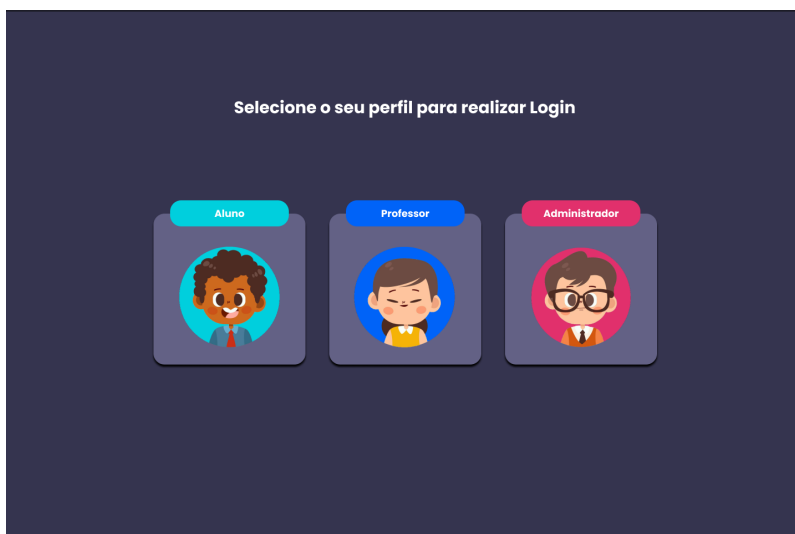
A imagem mostra a interface de cadastro de escolas. No topo, há um cabeçalho azul com o título "Cadastro" e o subtítulo "Preencha seus dados abaixo". À direita do cabeçalho, há um ícone de uma escola. Abaixo do cabeçalho, há um campo "Logotipo:" com um ícone de uma escola. O formulário principal contém campos para "Escola" (Nome completo da escola), "Email" (escola@host.com), "Telefone" ((00) 0000-0000), "CNPJ" (00.000.000/0000-00), "CEP" (00000-000), "Senha" e "Confirme sua Senha". Um botão azul "Criar Conta" está na base do formulário.

Figura 13 – Tela de cadastro de escolas

Fonte: De autoria própria.

Dessa vez, ao selecionar "Entrar" o sistema irá redirecionar o usuário para a tela de seleção de ambiente, conforme a Figura 14, para realizar o *login*:

Figura 14 – Seleção de ambiente para *login*

Fonte: De autoria própria.

5.2 Login

Para obter acesso às funcionalidades desenvolvidas aos perfis presentes num sistema escolar é necessário que o usuário realize seu *login* na plataforma. Nesse contexto, as credenciais utilizadas foram: *e-mail* e senha, sendo que o primeiro campo citado foi tido como chave única, de forma que o mesmo *e-mail* somente poderia ser utilizado em uma

única instituição de ensino. Além disso, cada ambiente (instituição ou aluno/professor) possuem páginas de *login* diferentes que podem ser verificadas nas Figuras 15 e 16.

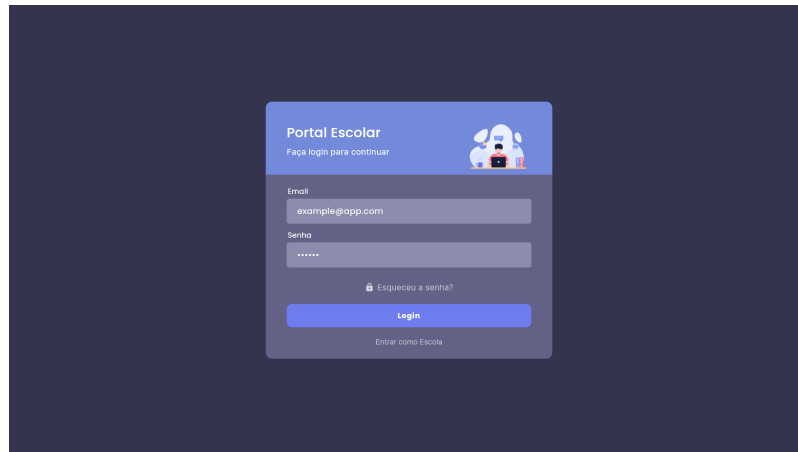


Figura 15 – *Login* no ambiente de professores e alunos

Fonte: De autoria própria.

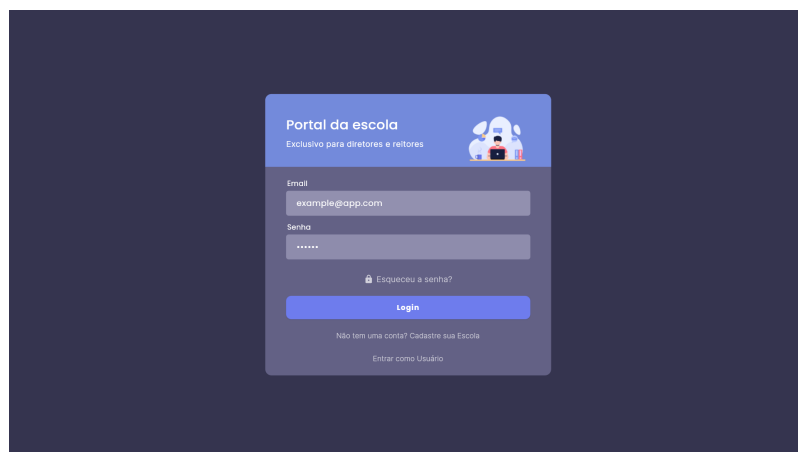


Figura 16 – *Login* no ambiente de instituições de ensino

Fonte: De autoria própria.

Em qualquer um dos casos é realizada a verificação para uso apenas de *e-mails* válidos e existentes no sistema. No caso do usuário precisar recuperar a senha, ele pode acessar a página de recuperação de senha, exemplificada na Figura 17, ao selecionar o botão "Esqueceu a senha":



Figura 17 – Tela de recuperação de senha

Fonte: De autoria própria.

5.3 Tela Inicial

Para o ambiente de escolas, a página inicial remete ao perfil da instituição, presente na Figura 18.

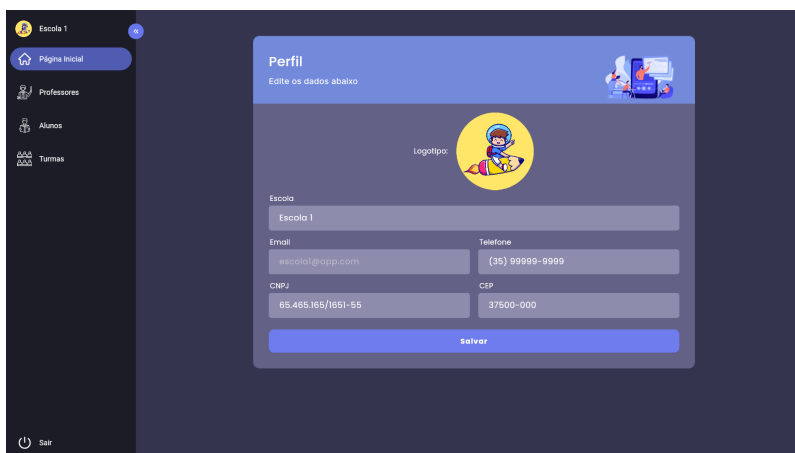


Figura 18 – Página inicial de perfil de uma instituição de ensino

Fonte: De autoria própria.

5.4 Cursos

Concordante à Figura 19, para o perfil de aluno, é possível acessar a área de cursos para visualizar tarefas, atividades, notícias e presença dos cursos cujo aluno esteja cadastrado.

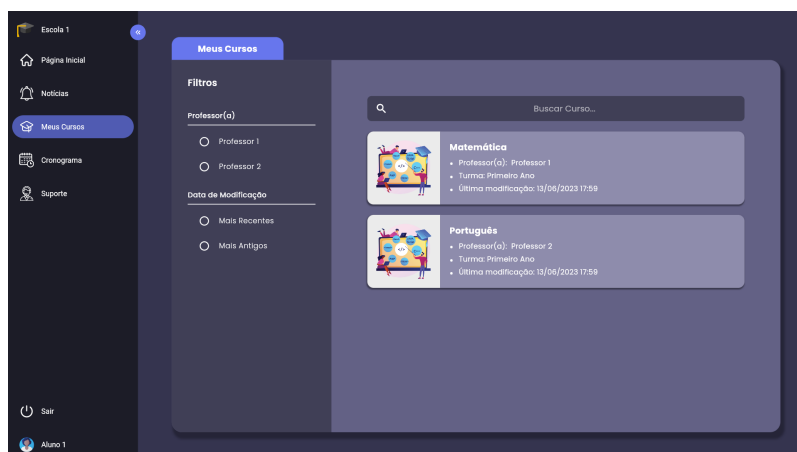


Figura 19 – Página de cursos de um aluno

Fonte: De autoria própria.

Ao selecionar o curso que deseja inspecionar, a tela de visualização será exibida com detalhes dos módulos criados pelos professores, atividades e outros recursos. Tal exibição pode ser verificada pela Figura 20.

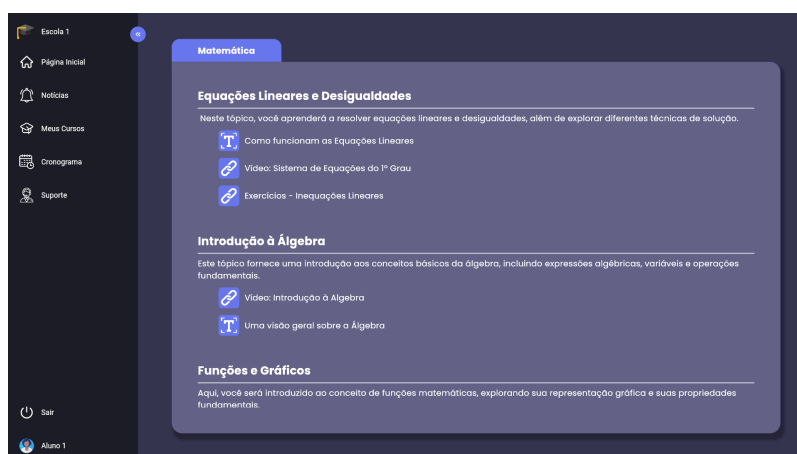


Figura 20 – Visualização de um curso pelo perfil aluno

Fonte: De autoria própria.

Já para o perfil de professor, além de conseguir visualizar as mesmas páginas que os alunos, ele também tem acesso à criação, edição e exclusão dos recursos que estão cadastrados, conforme mostra a Figura 21.

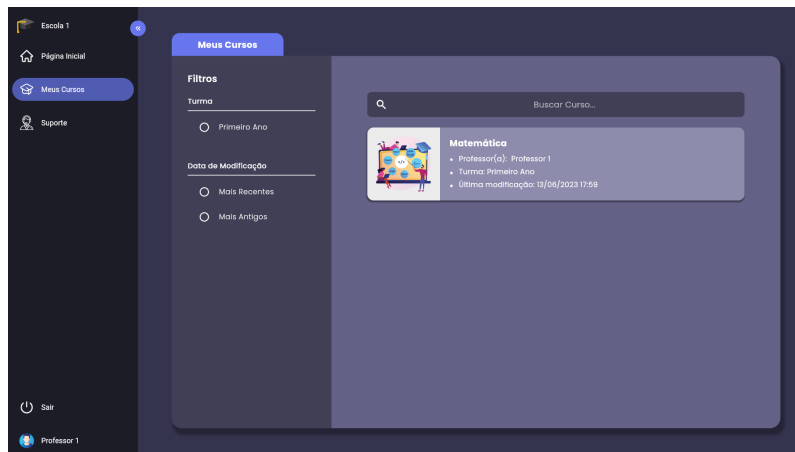


Figura 21 – Página de cursos para um professor

Fonte: De autoria própria.

A fim de organizar os conteúdos de aulas apresentados e adicionar atividades de forma organizada, o professor poderá criar diversos módulos dentro de suas matérias e adicionar conteúdo. Na Figura 22 é demonstrada a disposição de módulos em um curso.

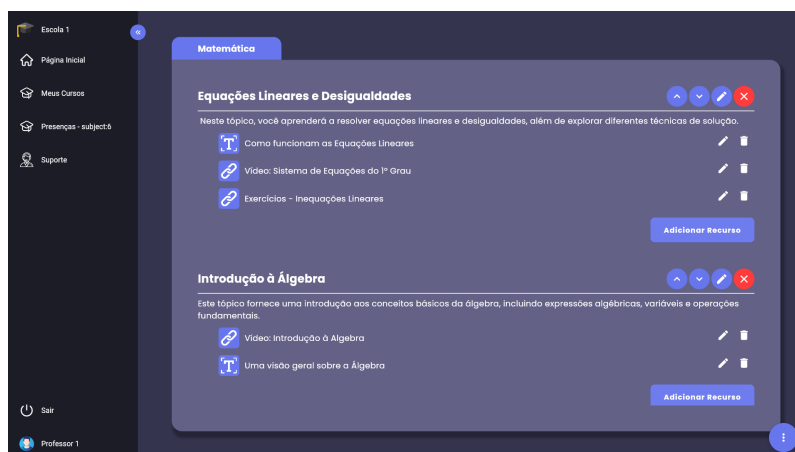


Figura 22 – Visualização de um curso por um professor

Fonte: De autoria própria.

O professor contará com alguns tipos de recursos para enriquecer seus cursos. Dentre eles, há o recurso **Link**, no qual o professor poderá colocar qualquer tipo de *site* com explicações ou apresentações, como um vídeo de determinado conteúdo. Paralelo a isso, o professor também poderá adicionar um **Texto**, assim como **Anexar um arquivo**, contendo uma atividade ou resumos de suas aulas. O *pop-up* de escolha desses recursos pode ser verificado pela Figura 23.



Figura 23 – Adição de recursos as disciplinas

Fonte: De autoria própria.

Além disso, o professor tem a possibilidade de criar novas notícias em um curso específico ou gerenciar a presença dos alunos cadastrados no curso ao clicar no botão (:) para abrir as opções "Notícias" e "Lista de Presença", conforme a Figura 24.

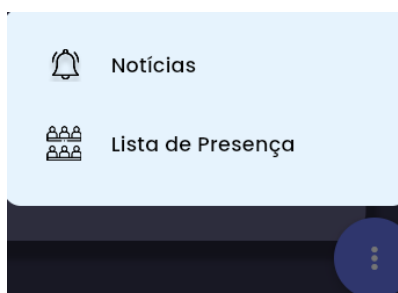


Figura 24 – Botão para acesso da lista de presença e notícias do curso

Fonte: De autoria própria.

5.5 Notícias

Como comentado anteriormente e concordante às Figuras 25 e 26, ao selecionar o botão de notícias na área de visualização do curso como um professor, a página de notícias será acessada permitindo gerenciar as notícias já existentes ou criar novas:

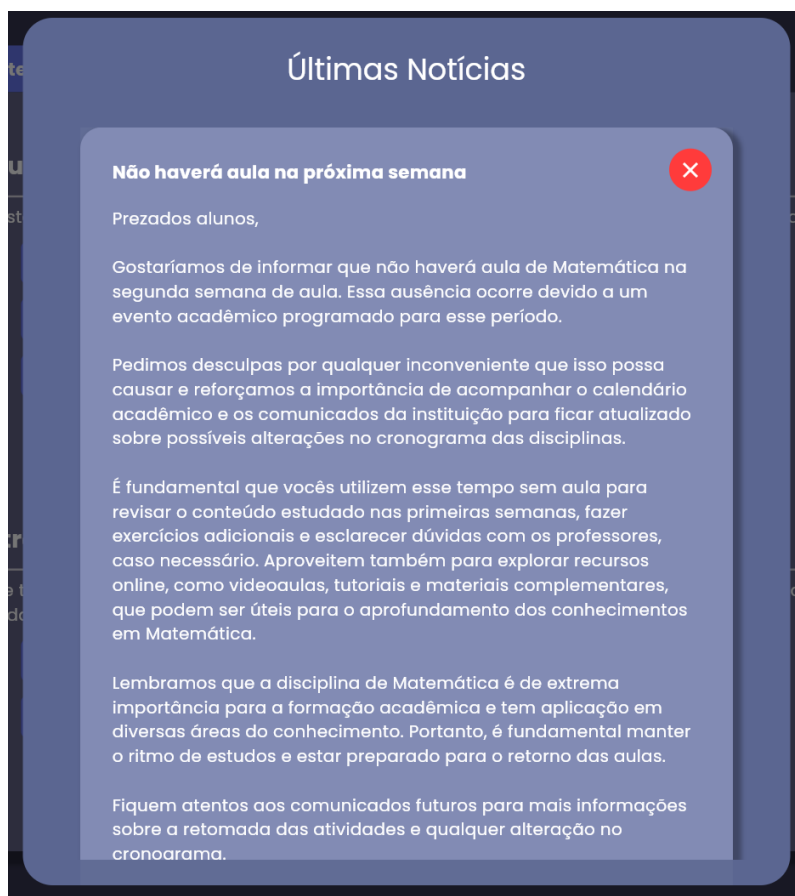


Figura 25 – Visualização das notícias de um curso

Fonte: De autoria própria.



Adicionar Notícia

Título*

Descrição*

Cancelar Confirmar

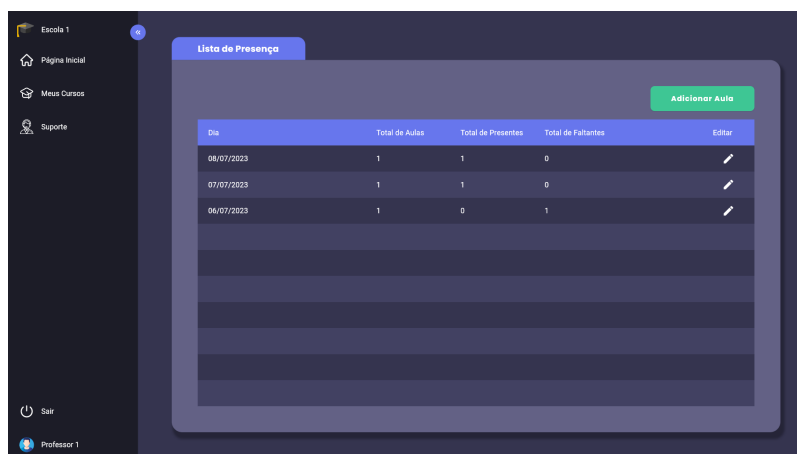
Figura 26 – Cadastro de uma notícia

Fonte: De autoria própria.

5.6 Lista de Presença

A fim de garantir que os alunos estejam acompanhando regularmente as aulas, o professor pode utilizar de uma lista de presença para registrar a frequência ou ausências

dos alunos em suas aulas. Dessa forma, permitindo que o professor saiba se o aluno está acompanhando os conteúdos no final do bimestre ou semestre. Além disso, o docente pode usar essa função para controlar a quantidade de aulas ministradas em determinado dia. Tal *feature* pode ser verificada através das Figuras 27 e 28.



Dia	Total de Aulas	Total de Presenças	Total de Faltantes	Editar
08/07/2023	1	1	0	
07/07/2023	1	1	0	
06/07/2023	1	0	1	

Figura 27 – Visualização da lista de presença de um curso

Fonte: De autoria própria.



Nome	Matrícula	<input checked="" type="radio"/> Presença	<input type="radio"/> Falta
Aluno 1	20230000	<input checked="" type="radio"/> Presença	<input type="radio"/> Falta

Figura 28 – Cadastro de uma nova lista de presença

Fonte: De autoria própria.

5.7 Turmas

Para o ambiente de uma escola, exclusivamente, a página de turmas é utilizada para criar as classes e atribuir a elas seus respectivos cursos. Sendo que, ao cadastrar ou editar uma turma é necessário fornecer um nome e suas disciplinas da estrutura curricular, como pode ser visto nas Figuras 29 e 30.

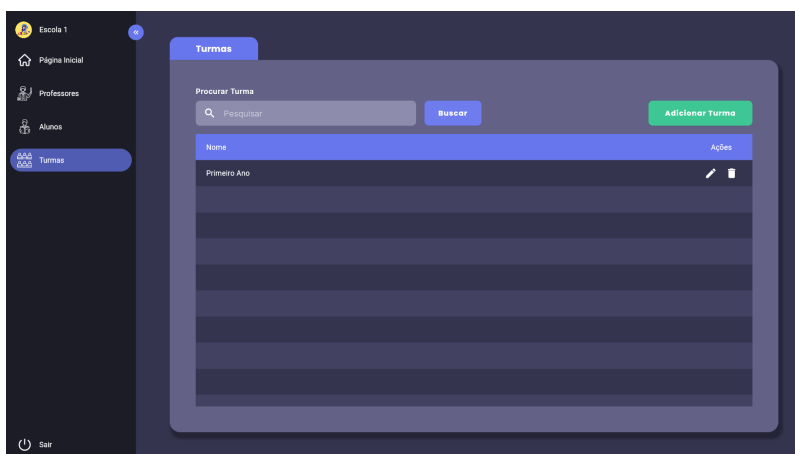


Figura 29 – Página de turmas de uma escola

Fonte: De autoria própria.

Editar Turma
Preencha com os dados abaixo

Nome
Primeiro Ano

Disciplina 1

Nome Matemática	Cor #6200D9	Professor Professor 1
Dia da semana Nenhum	Horário de Início Horário de Início	Horário de Término Horário de Término

Disciplina 2

Nome Português	Cor #FFE800	Professor Professor 2
Dia da semana Nenhum	Horário de Início Horário de Início	Horário de Término Horário de Término

+

Figura 30 – Edição de uma turma

Fonte: De autoria própria.

5.8 Alunos e Professores

Para cadastrar os docentes e discentes na plataforma de uma instituição, desta vez como um perfil de administrador, haverá a área "Professores" e "Alunos", onde pode-se

criar, editar e excluir esses tipos de usuário, conforme é demonstrado nas Figuras 31, 32, 33 e 34 .

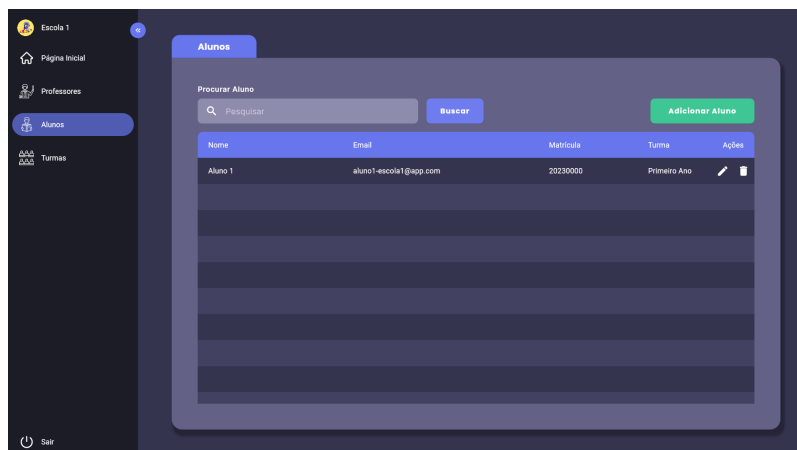


Figura 31 – Alunos de uma escola

Fonte: De autoria própria.

Figura 32 – Cadastro de um aluno

Fonte: De autoria própria.

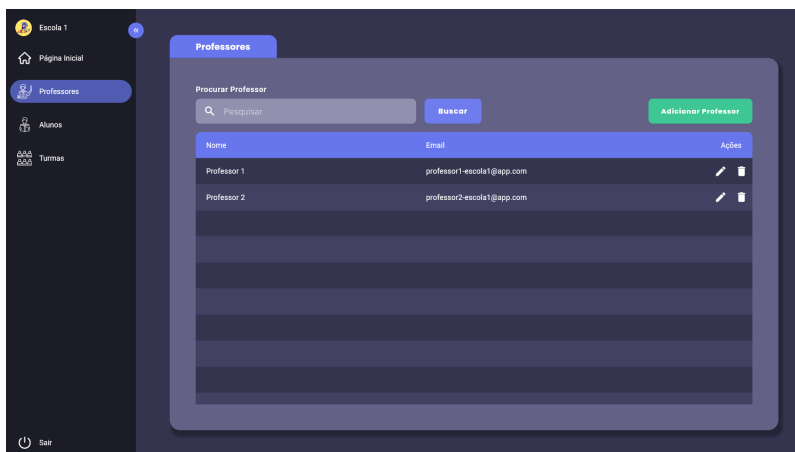


Figura 33 – Professores de uma escola

Fonte: De autoria própria.

O formulário 'Cadastrar novo Professor' apresenta o título e a instrução 'Preencha com os dados abaixo'. No canto superior direito, há uma ilustração de pessoas trabalhando em computadores. O formulário é dividido em campos para: 'Nome' (com o placeholder 'Nome completo'), 'Email' (com o placeholder 'Email'), 'Data de Nascimento', 'Telefone' (com o placeholder '(00) 0000-0000'), 'CPF' (com o placeholder '000.000.000-00'), 'RG' (com o placeholder 'documento'), 'Endereço' (com o placeholder 'Rua A, 123'), 'Senha' e 'Confirme sua Senha' (ambos com placeholders de pontos). Um botão azul 'Criar Conta' está na base do formulário.

Figura 34 – Cadastro de um professor

Fonte: De autoria própria.

5.9 Suporte

Por fim, a página de suporte (exemplificada na Figura 35) tem a simples função de direcionar usuários como alunos e professores aos principais meios de comunicação com seus respectivos administradores da escola.

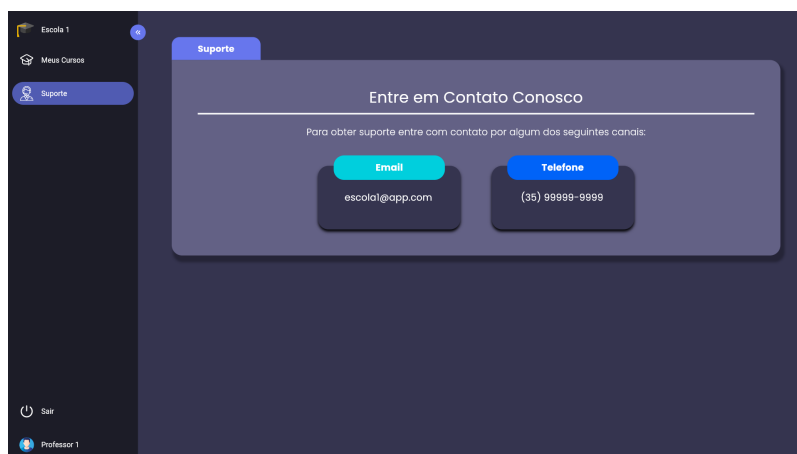


Figura 35 – Página de suporte

Fonte: De autoria própria.

6 Conclusão

A integração das tecnologias digitais nas salas de aula virtuais revela-se crucial, evidenciando a importância de plataformas educativas que correspondam às necessidades dos alunos e das instituições de ensino.

Nesse sentido, com base no objetivo deste trabalho de construir um sistema escolar centralizado que abranja diferentes modalidades no espectro educacional, conclui-se que a aplicação projetada é eficaz no fornecimento de serviços para administração acadêmica, ministração e consumo de cursos.

À medida que esse sistema escolar centralizado cresce, a necessidade de uma plataforma educacional com uma curva de aprendizado suave e uma interface amigável pode ser atendida. A centralização do sistema atua como um facilitador e é fundamental para o sucesso e aceitação de professores e alunos. Ao fornecer uma solução unificada, o *Mediation System* ajuda a superar os desafios do ensino à distância, proporcionando às escolas e alunos uma gestão acadêmica eficiente e um processo de aprendizagem mais eficaz.

A possibilidade de customização conforme as necessidades específicas da instituição é outro diferencial significativo, permitindo uma adaptação mais precisa. O apelo à identidade moderna e a otimização da experiência do usuário por meio de técnicas de UX/UI também contribuem para a superioridade do sistema proposto. Essas características se combinam para formar uma solução abrangente e inovadora que atende de forma eficiente e satisfatória as necessidades do ensino à distância. A partir dos resultados alcançados, pode-se concluir que o projeto possui amplo potencial para melhorar a qualidade do ensino virtual e proporcionar uma experiência educacional mais rica e personalizada.

6.1 Trabalhos Futuros

Durante o desenvolvimento do sistema escolar integrado, foram identificadas algumas oportunidades de implementação e melhoria para enriquecer ainda mais a plataforma e melhorar a experiência do usuário. Uma das implementações futuras é a inclusão da funcionalidade de *chat* para facilitar a comunicação direta entre professores, alunos e a escola.

Além disso, é importante ampliar a coletânea de recursos de atividades e avaliações disponíveis. Uma variedade de formatos de avaliação, como atividades práticas, projetos colaborativos e questionários interativos, ajudará a obter avaliações mais abrangentes e personalizadas.

Outra melhoria objetivada é a criação de uma seção específica para *upload* e visualização de videoaulas. Esse recurso fornecerá aos professores uma maneira eficiente de fornecer conteúdo audiovisual que enriquece a experiência de aprendizado dos alunos. A integração com outras plataformas externas também é uma possibilidade interessante. Ao integrar bibliotecas digitais, ferramentas de produtividade ou sistemas de gerenciamento de tarefas, os usuários poderão acessar mais recursos e aproveitar a sinergia entre diferentes ferramentas acadêmicas.

Além da implementação e melhoria citadas, uma sugestão relevante para melhorar o sistema escolar abrangente é adicionar o *light theme*, para que o usuário possa optar pela opção que mais lhe for confortável.

Por fim, considera-se o desenvolvimento de recursos de inteligência artificial para detecção de plágio de documentos. Por meio de algoritmos de análise de texto, semelhanças e padrões suspeitos podem ser identificados, promovendo assim a integridade acadêmica e originalidade do trabalho dos alunos.

Referências

- BARRO, B. B. Conheça a linguagem dart e entenda as polêmicas que a envolvem. 2023. Disponível em: <<https://www.hostinger.com.br/tutoriais/linguagem-dart#:~:text=Dart%20%C3%A9%20uma%20linguagem%20de,do%20cliente%20quanto%20no%20servidor.>>. Citado na página 25.
- BLACKBOARD. Blackboard help center. 2023. Disponível em: <<https://help.blackboard.com/pt-br>>. Citado na página 32.
- CANVA. Canva instructure. 2023. Disponível em: <<https://canvas.instructure.com/>>. Citado na página 27.
- CETIC.BR. Resumo executivo tic educação 2020. 2020. Disponível em: <https://www.cetic.br/media/docs/publicacoes/2/20211124200731/resumo_executivo_tic_educacao_2020.pdf>. Citado 2 vezes nas páginas 17 e 20.
- CETIC.BR. Tic domicílios 2021. 2021. Disponível em: <https://cetic.br/media/analises/tic_domicilios_2021_coletiva_imprensa.pdf>. Citado na página 20.
- CHIU, C.-M.; WANG, E. T. Understanding web-based learning continuance intention: The role of subjective task value. *Information Management*, v. 45, n. 3, p. 194–201, 2008. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0378720608000293>>. Citado 2 vezes nas páginas 17 e 21.
- CNE. Resolução cne/ceb. 2018. Disponível em: <<https://decentro.educacao.sp.gov.br/resolucao-seduc-de-18-3-2020/>>. Citado na página 17.
- DART. Dart documentation. 2023. Disponível em: <<https://dart.dev/guides>>. Citado 4 vezes nas páginas 24, 25, 38 e 39.
- DB-ENGINES. Db-engines ranking. 2023. Disponível em: <<https://db-engines.com/en/ranking>>. Citado na página 33.
- DBEAVER. Dbeaver documentation. 2023. Disponível em: <<https://dbeaver.io/>>. Citado na página 35.
- DEVDOCS. Javascript docs. 2023. Disponível em: <<https://devdocs.io/javascript/>>. Citado na página 23.
- DEVDOCS. Typescript docs. 2023. Disponível em: <<https://devdocs.io/typescript/>>. Citado na página 23.
- ELEPHANTSQL. Elephantsql documentation. 2023. Disponível em: <<https://www.elephantsql.com/docs/index.html>>. Citado na página 46.
- EXPRESS. Express documentation. 2023. Disponível em: <<https://expressjs.com/>>. Citado 3 vezes nas páginas 24, 36 e 43.
- FIALHO, I.; CID, M.; COPPI, M. Vantagens e dificuldades na utilização de plataformas e tecnologias digitais por professores e alunos. 2023. Disponível em: <<https://doi.org/10.1590/S1413-24782023280050>>. Citado na página 21.

FIGMA. Figma website. 2023. Disponível em: <<https://www.figma.com/>>. Citado na página 39.

FIREBASE. Cloud storage para firebase. 2023. Disponível em: <<https://firebase.google.com/docs/storage?hl=pt-br>>. Citado na página 46.

FIREBASE. Firebase hosting documentation. 2023. Disponível em: <<https://firebase.google.com/docs/hosting?hl=pt-br>>. Citado na página 46.

FLUTTER. Flutter documentation. 2023. Disponível em: <<https://docs.flutter.dev/>>. Citado 3 vezes nas páginas 24, 38 e 45.

G2. G2 research scoring methodologies. 2023. Disponível em: <<https://research.g2.com/methodology/scoring#:~:text=G2%20scores%20products%20and%20vendors,Presence%20scores%20in%20real%20time.>> Citado na página 25.

G2. The top 20 learning management systems. 2023. Disponível em: <https://www.g2.com/categories/learning-management-system-lms?tab=highest_rated>. Citado na página 25.

GOOGLE. Google classroom. 2023. Disponível em: <<https://classroom.google.com/>>. Citado 3 vezes nas páginas 25, 28 e 31.

GUIMARAES, U. A. et al. Ead e ensino superior brasileiro: Políticas públicas e uso de tics. 2023. Disponível em: <<https://recima21.com.br/index.php/recima21/article/view/3386/2442>>. Citado na página 21.

IBM. Sql vs. nosql databases: What's the difference? 2022. Disponível em: <<https://www.ibm.com/cloud/blog/sql-vs-nosql>>. Citado na página 34.

IBM. Acid properties of transactions. 2023. Disponível em: <<https://www.ibm.com/docs/en/cics-ts/5.4?topic=processing-acid-properties-transactions>>. Citado na página 34.

JWT. Introduction to json web tokens. 2023. Disponível em: <<https://jwt.io/introduction>>. Citado 2 vezes nas páginas 22 e 44.

LMS, C. Home page. 2023. Disponível em: <<https://www.instructure.com/pt-br/canvas>>. Citado 2 vezes nas páginas 25 e 31.

LMS, C. What are external apps (lti tools)? 2023. Disponível em: <<https://community.canvaslms.com/t5/Canvas-Basics-Guide/What-are-External-Apps-LTI-Tools/ta-p/57>>. Citado na página 29.

MELO, D. O que é node.js? 2021. Disponível em: <<https://tecnoblog.net/responde/o-que-e-node-js-guia-para-iniciantes/>>. Citado na página 24.

MOODLE. Moodle coding documentation. 2022. Disponível em: <<https://docs.moodle.org/dev/Coding#Plugins>>. Citado na página 29.

MOODLE. Moodle unifei. 2023. Disponível em: <<https://moodle.unifei.edu.br/>>. Citado 2 vezes nas páginas 25 e 26.

MOZILLA. O que é javascript? 2023. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First_steps/What_is_JavaScript>. Citado na página 23.

NODE. Node documentation. 2023. Disponível em: <<https://nodejs.org/pt-br/docs>>. Citado 2 vezes nas páginas 38 e 43.

NORONHA, C. B. O que é um framework. 2023. Disponível em: <<https://balta.io/blog/o-que-e-um-framework#oqueumframework>>. Citado na página 23.

PERCONA. The ultimate guide to open source databases. 2023. Disponível em: <<https://www.percona.com/blog/ultimate-guide-open-source-databases>>. Citado na página 33.

PGADMIN. Pgadmin documentation. 2023. Disponível em: <<https://www.pgadmin.org/>>. Citado na página 35.

PNAD. Internet já é acessível em 90,02021. Disponível em: <<https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/34954-internet-ja-e-acessivel-em-90-0-dos-domicilios-do-pais-em-2021#:~:text=Em%202021%2C%20entre%20os%20183,a%20rede%20p%C3%ABblica%20de%20ensino.>> Citado na página 20.

POSTGRESQL. Postgresql documentation. 2023. Disponível em: <<https://www.postgresql.org/about/>>. Citado na página 35.

POSTMAN. Postman documentation. 2023. Disponível em: <<https://learning.postman.com/docs/introduction/overview/>>. Citado na página 38.

REDHAT. Api rest. 2020. Disponível em: <<https://www.redhat.com/pt-br/topics/api/what-is-a-rest-api>>. Citado na página 22.

REDHAT. What is middleware? 2023. Disponível em: <<https://www.redhat.com/en/topics/middleware/what-is-middleware#:~:text=Middleware%20and%20APIs-,Overview,applications%2C%20data%2C%20and%20users.>> Citado na página 44.

RESTFULAPI. What is rest. 2021. Disponível em: <<https://restfulapi.net/>>. Citado 2 vezes nas páginas 22 e 35.

RODRIGUES, I. A. D. A.; BEZERRA, M. M. As redes sociais e a gestão da indisciplina escolar: um estudo de revisão bibliográfica. 2018. Disponível em: <<https://editorarealize.com.br/artigo/visualizar/46496>>. Citado 2 vezes nas páginas 21 e 22.

SCRUM. What is scrum? 2023. Disponível em: <<https://www.scrum.org/learning-series/what-is-scrum>>. Citado na página 19.

STACKIFY. Should you use flutter in your next project? 2023. Disponível em: <<https://stackify.com/should-you-use-flutter-in-your-next-project/>>. Citado na página 38.

STATISTA. Most used programming languages among developers worldwide as of 2022. 2022. Disponível em: <<https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/>>. Citado na página 37.

STATISTICS; COMPANY, D. Most popular backend frameworks. *Statistics and Data*, 2023. Disponível em: <<https://statisticsanddata.org/data/most-popular-backend-frameworks-2012-2023/>>. Citado na página 36.

TECHTARGET. Strongly typed programming language. 2022. Disponível em: <https://www.techtarget.com/whatis/definition/strongly-typed>. Citado na página 37.

UNIFEI, M. T. Login page. 2023. Disponível em: <https://moodleteste.unifei.edu.br/login/index.php>. Citado 2 vezes nas páginas 28 e 32.

VALENTE, J.; ALMEIDA, M. E. Políticas de tecnologia na educação no brasil: Visão histórica e lições aprendidas. *education policy analysis archives*, v. 28, p. 94, 06 2020. Citado na página 21.

VSCODE. Vscod documentation. 2023. Disponível em: <https://code.visualstudio.com/docs>. Citado na página 39.

XP. Entenda o que é e para que serve o express.js. 2022. Disponível em: <https://blog.xpeducacao.com.br/express-javascript/>. Citado na página 24.

XP. Flutter: o que é, para que serve e se vale a pena aprender. 2022. Disponível em: <https://blog.xpeducacao.com.br/flutter/#:~:text=Para%20que%20serve%20o%20Flutter,processo%20e%20facilita%20a%20manuten%C3%A7%C3%A3o.>> Citado na página 24.

XP. Typescript: entenda o que é e para que serve a linguagem. 2022. Disponível em: <https://blog.xpeducacao.com.br/typescript/>. Citado na página 23.