



UFOP

Universidade Federal
de Ouro Preto

**Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Computação e Sistemas**

**Desenvolvimento de uma aplicação
web para simulados preparatórios de
vestibular com enfoque no ENEM**

João Vitor de Moura Rosa

**TRABALHO DE
CONCLUSÃO DE CURSO**

**ORIENTAÇÃO:
Fernando Bernardes de Oliveira**

**Março, 2023
João Monlevade–MG**

João Vitor de Moura Rosa

Desenvolvimento de uma aplicação web para simulados preparatórios de vestibular com enfoque no ENEM

Orientador: Fernando Bernardes de Oliveira

Monografia apresentada ao curso de Sistemas de Informação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina “Trabalho de Conclusão de Curso II”.

Universidade Federal de Ouro Preto

João Monlevade

Março de 2023

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

R788d Rosa, Joao Vitor de Moura.

Desenvolvimento de uma aplicação web para simulados preparatórios de vestibular com enfoque no ENEM. [manuscrito] / Joao Vitor de Moura Rosa. - 2023.

76 f.: il.: color., tab..

Orientador: Prof. Dr. Fernando Bernardes de Oliveira.

Monografia (Bacharelado). Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Aplicadas. Graduação em Sistemas de Informação .

1. Aplicações Web. 2. Dispositivos de treinamento simulado. 3. Ensino médio - Avaliação educacional. 4. Software de aplicação - Desenvolvimento. 5. Universidades e faculdades - Vestibular. I. Oliveira, Fernando Bernardes de. II. Universidade Federal de Ouro Preto. III. Título.

CDU 004.41

Bibliotecário(a) Responsável: Flavia Reis - CRB6-2431



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
INSTITUTO DE CIÊNCIAS EXATAS E
APLICADAS
DEPARTAMENTO DE COMPUTAÇÃO E
SISTEMAS



FOLHA DE APROVAÇÃO

João Vitor de Moura Rosa

Desenvolvimento de uma aplicação web para simulados preparatórios de vestibular com enfoque no ENEM

Monografia apresentada ao Curso de Sistemas de Informação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação

Aprovada em 30 de março de 2023

Membros da banca

Prof. Dr. Fernando Bernardes de Oliveira - Orientador (Universidade Federal de Ouro Preto)
Profª. Drª. Gilda Aparecida de Assis - Avaliadora (Universidade Federal de Ouro Preto)
Prof. Dr. George Henrique Godim da Fonseca - Avaliador (Universidade Federal de Ouro Preto)

Fernando Bernardes de Oliveira, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 02/05/2023



Documento assinado eletronicamente por **Fernando Bernardes de Oliveira, PROFESSOR DE MAGISTERIO SUPERIOR**, em 02/05/2023, às 18:36, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0518017** e o código CRC **16488352**.

Este trabalho é dedicado a todos que estiveram comigo nos momentos de felicidade e de tristeza nos meus anos de graduação.

Agradecimentos

Agradeço primeiramente à minha família, em especial aos meus pais Sônia e Emerson que sempre me deram apoio e carinho, não só nos anos de graduação, mas durante toda a minha vida.

A todos os amigos e amigas que fiz durante a vida e durante a faculdade, por todo o carinho, atenção, ajuda e também por aguentar horas e horas de "choradeira".

A minha namorada Ana Clara, por todo carinho, incentivo e amor.

Aos meus professores e todos os funcionários do ICEA, sem vocês jamais teria conquistado as coisas que venho conquistando na minha jornada profissional.

Um agradecimento especial para o meu orientador Fernando, por todo o tempo que prestou, pelo apoio, acolhimento e sábios conselhos para o desenvolvimento deste trabalho.

“Talvez não tenha conseguido fazer o melhor, mas lutei para que o melhor fosse feito. Não sou o que deveria ser, mas Graças a Deus, não sou o que era antes.”

— Martin Luther King Jr. (1929 – 1968)

Resumo

Todos os anos, milhares de brasileiros se inscrevem para realizarem os vestibulares das instituições de ensino superior e buscam diferentes maneiras de se preparar, incluindo cursos preparatórios, plataformas online e estudos individuais. Este trabalho tem como objetivo criar uma plataforma gamificada para auxiliar candidatos na preparação para os vestibulares.

O crescimento significativo da educação/estudo online nos últimos anos, em parte impulsionado pela pandemia da COVID-19, tornou a criação de uma plataforma como essa ainda mais relevante e importante. A plataforma oferece uma maneira gamificada de resolver provas preparatórias e também possui uma API que fornece acesso a um banco de dados com centenas de questões de vestibulares anteriores, principalmente do ENEM.

A metodologia usada na implementação desse trabalho incluiu a seleção de tecnologias atualmente disponíveis no mercado e apropriadas ao contexto de aplicação. Além disso, foram aplicadas técnicas de engenharia de software e design.

Os resultados sugerem uma plataforma eficiente e atraente para o público-alvo, fornecendo uma maneira interativa de estudar para os vestibulares. Além disso, a API permite que outras aplicações acessem e usem o banco de dados de questões, ampliando ainda mais as possibilidades de uso.

As considerações finais deste trabalho destacam a importância de soluções digitais no contexto da preparação para vestibulares e a relevância da gamificação como uma maneira de tornar o processo de estudo mais atraente e motivador para o usuário.

Palavras-chaves: web. vestibular. ENEM. simulado.

Abstract

This project aims to create a gamified platform to assist candidates in preparing for college entrance exams. Annually, thousands of Brazilians enroll to take entrance exams at higher education institutions and seek different ways to prepare, including courses, online platforms, and self-study.

The significant growth of online education/study in recent years, partly motivated by the COVID-19 pandemic, has made the creation of a platform even more relevant and important. The platform offers a gamified way of solving preparatory simulated exams and also features an API that provides access to a database with hundreds of questions from past college entrance exams, primarily from ENEM.

The methodology used in the implementation of this work included the selection of technologies currently present in the market and appropriate for the application context, as well as the application of software engineering and design techniques.

The results obtained present an efficient and attractive platform for the target audience, providing a fun and interactive way of studying for college entrance exams. Furthermore, the API allows other applications to access and use the question database, further expanding the possibilities of use.

The final considerations of this work highlight the importance of digital solutions in the context of preparing for college entrance exams and the relevance of gamification as a way of making the study process more attractive and motivating for the user.

Key-words: web. entrance exam. ENEM. simulation.

Lista de ilustrações

Figura 1 – Exemplo de código HTML	25
Figura 2 – Exemplo de código CSS	26
Figura 3 – Exemplo de código JavaScript	27
Figura 4 – Representação de requisição e resposta do protocolo HTTP	28
Figura 5 – Estrutura de um objeto/arquivo JSON	29
Figura 6 – Esquema não relacional simples de uma locadora de veículos	32
Figura 7 – Esquema arquitetural de microsserviços	33
Figura 8 – Exemplo de tela contruida com Material Design	37
Figura 9 – Exemplo de projeto construído no Figma	39
Figura 10 – Simulador do Brasil Escola	42
Figura 11 – Simulador do Mundo Vestibular	43
Figura 12 – Simulador Beduka	44
Figura 13 – Wireframe - Tela de inicial	48
Figura 14 – Wireframe - Modal de geração de simulados	49
Figura 15 – Wireframe - Tela de resolução de simulados	50
Figura 16 – Wireframe - Feedback	51
Figura 17 – Wireframe - Questão correta	52
Figura 18 – Estrutura de pastas para o script de carga	53
Figura 19 – Tela de login da aplicação	55
Figura 20 – Erro ao fazer login	55
Figura 21 – Tela de cadastro de usuários	57
Figura 22 – Tela inicial da aplicação	59
Figura 23 – Modal de geração de simulados	60
Figura 24 – Tela de resolução do simulado	61
Figura 25 – Finalização do simulado	62
Figura 26 – Média geral dos resultados do usuário	63
Figura 27 – Resultados das provas	64
Figura 28 – Resultados das provas - Alternativa correta	65
Figura 29 – Resultados das provas - Alternativa incorreta	66
Figura 30 – <i>Ranking</i> de usuários	67
Figura 31 – <i>Ranking</i> de usuários	67

Lista de tabelas

Tabela 1 – Tabela comparativa entre linguagens.	35
---	----

Lista de abreviaturas e siglas

API *Application Programming Interface*

CSS *Cascading Style Sheets*

DBMS *Data Base Management System*

ENEM Exame Nacional do Ensino Médio

FIES Fundo de Financiamento ao Estudante do Ensino Superior

Fuvest Fundação Universitária para o Vestibular

HTML *HyperText Markup Language*

HTTP *HyperText Transfer Protocol*

IDEB Índice de Desenvolvimento da Educação Básica

INEP Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira

JSON *JavaScript Object Notation*

JWT *JSON Web Token*

PROUNI Programa Universidade Para Todos

REST *Representational State Transfer*

SGBD Sistema Gerenciador de Bancos de Dados

UNICAMP Universidade Estadual de Campinas

URI *Uniform Resource Identifiers*

Sumário

1	INTRODUÇÃO	16
1.1	Problema	16
1.2	Objetivos	17
1.2.1	Objetivo geral	17
1.2.2	Objetivos específicos	17
1.3	Organização do trabalho	18
2	FUNDAMENTAÇÃO TEÓRICO-METODOLÓGICA	19
2.1	Vestibular e ingresso no ensino superior	19
2.1.1	ENEM	20
2.1.2	Fuvest	21
2.1.3	UNICAMP	21
2.2	Gameificação	22
2.3	Definições e conceitos da web	22
2.3.1	Web	23
2.3.2	Website	23
2.3.3	Sistemas web	23
2.4	Desenvolvimento web	24
2.4.1	HTML	24
2.4.2	CSS	25
2.4.3	JavaScript	26
2.4.4	HTTP	27
2.4.4.1	Cabeçalho da requisição	28
2.4.4.2	Corpo da Requisição	28
2.4.4.3	Métodos da Requisição	29
2.4.4.4	Códigos de resposta	30
2.4.5	Bancos de dados	30
2.4.5.1	Bancos não relacionais (NoSQL)	31
2.4.6	API	32
2.4.7	Microserviços	33
2.5	Bibliotecas	34
2.6	Ferramentas e tecnologias utilizadas	34
2.6.1	TypeScript	34
2.6.2	React	35
2.6.2.1	MUI	36

2.6.3	Node.js	37
2.6.3.1	Express.js	38
2.6.4	MongoDB	38
2.6.5	Figma	39
2.6.6	Git e GitHub	39
2.6.7	Docker	40
2.7	Considerações	40
3	TRABALHOS CORRELATOS	41
3.1	Simulador do Brasil Escola	41
3.2	Simulador Mundo Vestibular	42
3.3	Simulador Beduka	44
3.4	Considerações	45
4	DESENVOLVIMENTO E RESULTADOS	46
4.1	Wireframe da aplicação	47
4.2	Script de carregamento de questões	52
4.3	API de questões e geração de simulados	53
4.4	API principal	54
4.5	Interface de usuário e detalhamento dos requisitos	54
4.5.1	Login do usuário	54
4.5.2	Cadastro de usuário	56
4.5.3	Tela inicial	58
4.5.4	Geração dos simulados	59
4.5.4.1	Simulados comuns	60
4.5.4.2	Simulados ranqueados	60
4.5.5	Resolução dos simulados	61
4.5.6	Tela de resultados	62
4.5.6.1	Detalhamento do simulado resolvido	64
4.5.6.2	Ranking de usuários	66
4.5.7	Inserção de questões feitas de maneira manual (Administrador)	67
4.6	Testes e resultados	68
4.7	Considerações	69
5	CONCLUSÃO	70
5.1	Trabalhos Futuros	70
	REFERÊNCIAS	72

APÊNDICES	73
APÊNDICE A – FORMULÁRIO DE EXPERIÊNCIA DO USUÁRIO E FEEDBACKS	74

1 Introdução

Com o advento da tecnologia, houve uma modernização no modo de vida das pessoas, ocasionando na substituição de antigos hábitos por novas práticas que empregam soluções digitais. Nesse contexto, o uso de software se tornou parte integrante de nossas vidas, sendo utilizado em diversas atividades, tais como viagens, pedido de comida, interação social e, especificamente, no âmbito acadêmico, como suporte aos estudos.

Além disso, a tecnologia também permitiu a modernização de processos antes realizados manualmente, como a realização de aulas e atividades complementares. O presente trabalho visa proporcionar uma solução para complementar o processo de estudo de candidatos que irão prestar vestibular, oferecendo uma base para estudos a partir de questões de vestibulares passados, bem como, estabelecer um ranqueamento dos usuários de acordo com o seu desempenho.

A utilização de tecnologias atuais e bem estabelecidas no mercado é fundamental para a implementação deste trabalho, pois garante que a aplicação será desenvolvida de forma eficiente e com boa qualidade. Além disso, a utilização de técnicas de engenharia de *software* e *design*, contribui para a criação de um produto que seja fácil de usar, intuitivo e que atenda às expectativas dos usuários.

1.1 Problema

Todos os anos, uma grande quantidade de estudantes brasileiros se inscrevem em vestibulares com o objetivo de ingressar em instituições de ensino superior. É comum que esses candidatos busquem diferentes formas de se preparar para as provas, incluindo a preparação intelectual, mental e até física. Isso se deve ao fato de que muitos vestibulares são compostos por provas que demandam muitas horas para serem completadas.

Uma das formas mais utilizadas por esses candidatos para complementar sua preparação é o estudo de questões antigas desses exames. Além de ser uma estratégia eficiente para o estudo, essa prática também permite que o candidato tenha maior familiaridade com os formatos de questões que serão apresentados na prova. A familiaridade com o formato de questões é importante pois pode influenciar diretamente no tempo disponível para resolução das questões e na eficiência do desempenho do candidato durante a prova.

A plataforma descrita no presente trabalho permite que os usuários gerarem simulados de acordo com suas necessidades, permitindo que eles se concentrem em conteúdos em que precisam de mais aprimoramento e desenvolvimento. Os simulados gerados também podem ser utilizados para avaliar o desempenho dos usuários, ajudando-os a ter uma ideia

de como estão se preparando para os vestibulares. A plataforma possui um sistema de ranqueamento baseado em pontos, que permite aos usuários compararem seu desempenho com outros candidatos que usam a plataforma. Este sistema de ranqueamento será baseado nos resultados dos simulados resolvidos pelos usuários.

1.2 Objetivos

A plataforma permitirá a geração de simulados personalizados, de acordo com as necessidades de cada estudante, e possibilitará a comparação dos resultados obtidos com outros usuários, através de um *ranking* baseado em pontos. Além disso, o trabalho inclui a criação de um banco de questões de vestibulares passados, principalmente o ENEM, que serão integradas a um microsserviço e disponibilizadas para futuros trabalhos de outros desenvolvedores.

No desenvolvimento deste trabalho, foram avaliadas diversas tecnologias, ferramentas, plataformas e arquiteturas para alcançar os objetivos propostos. Para a criação da plataforma web para gerar simulados e *ranking* de desempenho, assim como o banco de exercícios de vestibulares, foi necessário o estudo cuidadoso dessas tecnologias. Os processos e as ferramentas utilizadas serão descritos e apresentados em tópicos subsequentes, incluindo conceitos importantes sobre o desenvolvimento web, suas características e práticas comuns.

1.2.1 Objetivo geral

Este trabalho tem como objetivo principal a criação de uma plataforma web para auxiliar estudantes que irão prestar o vestibular, oferecendo a possibilidade de gerar simulados personalizados, assim como o acesso a um banco de exercícios com questões de vestibulares passados. Além disso, a plataforma oferecerá aos usuários a possibilidade de comparar seus resultados com outros usuários através de um sistema de *ranking*. O desenvolvimento da plataforma foi realizado através da análise e estudo de tecnologias, ferramentas, plataformas, arquiteturas e maneiras de implementação, visando garantir a eficiência e a qualidade do sistema.

1.2.2 Objetivos específicos

O presente trabalho tem como objetivos específicos:

- Modelar e implementar uma aplicação Web para auxiliar estudantes que irão prestar o vestibular.
- Criar um banco de dados de questões a partir da importação de questões do ENEM.

- Validar aplicação proposta por meio de testes e com os usuários da aplicação.

1.3 Organização do trabalho

Este restante deste trabalho está organizado da seguinte maneira. No Capítulo 2 é apresentada revisão sobre os assuntos que serão tratados, bem como os principais conceitos, teorias e métodos relacionados. Além disso, é importante explicar a metodologia que será utilizada para alcançar os objetivos específicos do trabalho.

O Capítulo 3 apresenta uma revisão dos trabalhos similares ao escopo do presente trabalho. É importante destacar as contribuições e diferenças em relação aos resultados encontrados, bem como apresentar as lacunas que serão preenchidas com a realização deste trabalho.

No Capítulo 4 são descritos o desenvolvimento e os resultados obtidos a partir da metodologia adotada. No Capítulo 5 é apresentada a conclusão do trabalho, destacando os principais resultados encontrados e sua importância. Além disso, é importante apresentar as contribuições do trabalho em relação ao assunto tratado, bem como sugestões para trabalhos futuros.

2 Fundamentação Teórico-Metodológica

No presente capítulo serão apresentados conceitos relacionados ao trabalho, tal como os conceitos, tecnologias, ferramentas e métodos utilizados para a realização do mesmo. A Seção 2.1 apresentará o vestibular e o ingresso no ensino superior. A Seção 2.2 irá descrever a gamificação. A Seção 2.3 tratará dos sistemas web, abrangendo suas definições e conceitos. A Seção 2.4 irá abordar o desenvolvimento web. A Seção 2.5 abordará as bibliotecas e a Seção 2.6 irá apresentar as ferramentas e tecnologias utilizadas.

2.1 Vestibular e ingresso no ensino superior

De acordo com Alves (2021), o vestibular é um processo de seleção que consiste em uma prova de redação e um caderno de questões, abrangendo assuntos estudados durante o ensino médio. A classificação para ingressar na faculdade é realizada com base na pontuação obtida pelo candidato e na disponibilidade de vagas por curso, além de considerar os critérios de aprovação estabelecidos. Este é o método mais comum de ingresso no ensino superior no Brasil, sendo utilizado tanto por instituições públicas quanto privadas.

Segundo a literatura consultada, uma universidade pública é uma instituição de ensino pertencente ao poder público, seja ao nível federal, estadual ou municipal. Essas universidades oferecem ensino gratuito, com o financiamento proveniente da arrecadação de impostos. Por outro lado, as universidades privadas são instituições financeiramente autônomas, que cobram uma taxa anual do aluno para oferecer os seus serviços de ensino. No entanto, existem programas como o Programa Universidade Para Todos (PROUNI) e o Fundo de Financiamento ao Estudante do Ensino Superior (FIES), criados pelo governo federal, que buscam ampliar o acesso ao ensino superior e torná-lo mais acessível, seja através da redução ou isenção das taxas cobradas.

- Prouni: O Programa Universidade Para Todos ¹ é uma iniciativa do Governo Federal destinada a garantir o acesso à educação superior a todos os brasileiros. O PROUNI oferece bolsas de estudo integrais e parciais, que cobrem 50% do valor das mensalidades dos cursos de graduação e de formação específica, em instituições de ensino superior privadas. O programa é direcionado a estudantes sem diploma de nível superior.
- Fies: O Fundo de Financiamento Estudantil ² é um programa do Ministério da Educação que visa financiar a graduação em instituições de educação superior privadas

¹ Disponível em: <https://accessunico.mec.gov.br/prouni>

² Disponível em: <https://sisfiesportal.mec.gov.br/>

para estudantes matriculados em cursos superiores não gratuitos. O programa está destinado aos estudantes que passaram por avaliação positiva nos processos conduzidos pelo Ministério da Educação e desejam continuar seus estudos em cursos superiores.

O ensino em universidades públicas no Brasil é gratuito e reconhecido por sua qualidade elevada. Como resultado, há uma demanda significativa por vagas em tais instituições, levando a uma alta concorrência entre os candidatos. Para alcançar melhores resultados, é necessário que os candidatos se preparem de maneira adequada, por meio de diferentes técnicas e estratégias. Os três maiores vestibulares do país, medidos pelo número de inscrições anuais, são o ENEM, FUVEST e UNICAMP, na respectiva ordem, do maior para o menor.

2.1.1 ENEM

Para [Caetano \(2021\)](#), o Exame Nacional do Ensino Médio ([ENEM](#)) é a maior prova do Brasil, aplicada anualmente para mais de 5 milhões de estudantes do ensino médio. Seu objetivo é avaliar o desempenho escolar dos estudantes ao final da educação básica. Em 2009, o ENEM aperfeiçoou sua metodologia e passou a ser utilizado como mecanismo de acesso à educação superior.

O ENEM é composto por questões objetivas e uma redação. As questões objetivas cobrem um amplo espectro de assuntos e são avaliadas de acordo com uma escala de 0 a 1000 pontos. A redação é avaliada por meio de critérios como coesão, coerência, domínio da norma padrão da língua portuguesa, entre outros.

Desde 2020, o participante pode escolher entre fazer o exame impresso ou o ENEM Digital, com provas aplicadas em computadores, em locais de prova definidos pelo INEP. Além de avaliar o desempenho escolar dos estudantes, o ENEM também serve como mecanismo de acesso à educação superior, pois os resultados podem ser utilizados como base para a obtenção de bolsas de estudo e ingresso em universidades públicas e privadas.

Além disso, o ENEM também é utilizado como base para o cálculo do Índice de Desenvolvimento da Educação Básica ([IDEB](#)), que é uma avaliação do desempenho do ensino básico no Brasil. O resultado do ENEM é disponibilizado pelo Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira ([INEP](#)) após a conclusão da prova, e os estudantes podem consultar suas notas e desempenho em comparação a outros candidatos.

2.1.2 Fuvest

O Exame de Admissão da Fundação Universitária para o Vestibular (**Fuvest**)³ é uma prova de seleção aplicada anualmente com o objetivo de avaliar o desempenho escolar dos estudantes e selecionar candidatos para ingressar na Universidade de São Paulo (USP), uma das mais renomadas instituições de ensino superior do Brasil. A Fuvest foi fundada em 1977 e desde então tem sido responsável pela realização do exame de admissão para a USP.

A primeira fase é realizada em dois dias consecutivos e cobra conhecimentos de diversas disciplinas, como português, matemática, história, geografia, biologia, física, química, entre outras. Além disso, há também questões de atualidades e redação. Nesta fase, os estudantes precisam alcançar uma nota mínima para avançar para a segunda fase.

A segunda fase é composta por questões dissertativas e tem como objetivo avaliar a capacidade dos candidatos de argumentação e análise crítica. Além disso, a Fuvest também pode incluir provas práticas, como provas de habilidades específicas para algumas áreas, como música, artes e teatro.

Após a realização das duas fases, os candidatos são classificados de acordo com suas notas e o número de vagas disponíveis. A lista dos aprovados é divulgada em um período determinado pela Fuvest e os aprovados precisam confirmar sua matrícula nas datas estabelecidas.

Em resumo, o vestibular da Fuvest é uma avaliação rigorosa que avalia tanto conhecimentos técnicos quanto habilidades de análise e argumentação. É importante que os candidatos estejam preparados para enfrentar essa prova e alcançarem a classificação desejada.

2.1.3 UNICAMP

O processo seletivo da Universidade Estadual de Campinas (**UNICAMP**)⁴ é o processo de seleção para ingresso em cursos de graduação oferecidos pela instituição. É realizado anualmente e é composto por duas fases.

A Primeira Fase consiste em uma prova objetiva com questões de múltipla escolha de conhecimentos gerais, tais como: matemática, português, história, geografia, biologia, física, química, entre outras. Além disso, os candidatos também precisam escrever uma redação sobre um tema determinado pela instituição.

A Segunda Fase é composta por provas específicas para cada curso, que avaliam o conhecimento e habilidades necessárias para o desempenho do futuro estudante. Essas

³ Disponível em: <https://accessunico.mec.gov.br/prouni>

⁴ Disponível em: <https://www.unicamp.br/unicamp/>

provas podem ser compostas por questões teóricas, questões dissertativas, entrevistas, análise de projetos e outros tipos de avaliação.

A classificação dos candidatos é feita a partir da soma dos pontos obtidos nas duas fases do vestibular, além de levar em consideração o desempenho do estudante durante o ensino médio. A lista dos aprovados é divulgada pela UNICAMP após o processo de seleção.

Em geral, o vestibular da UNICAMP é conhecido por ser um dos mais exigentes do país, requerendo uma preparação rigorosa dos candidatos para obter boas notas.

2.2 Gameficação

Para [Fadel et al. \(2014\)](#), a gamificação é o uso de técnicas de jogo em contextos não lúdicos, como por exemplo em um ambiente de trabalho ou de ensino, a fim de estimular comportamentos positivos e motivar os participantes a alcançarem metas ou objetivos.

No escopo deste trabalho, a gamificação pode ser aplicada de diversas maneiras, por exemplo, criando desafios que incentivem os usuários a se engajarem mais na aprendizagem, aprimorar suas habilidades e atingir suas metas. Além disso, a gamificação pode ser usada para tornar o processo de aprendizagem mais lúdico e interativo, aumentando a motivação dos usuários e tornando-os mais propensos a reterem e aplicarem o conhecimento adquirido.

A gamificação pode ser realizada através de vários mecanismos, como por exemplo: pontuação, recompensas, progresso visual, desafios, *rankings* (tabelas de classificação), entre outros. Para que a gamificação seja eficaz, é importante considerar aspectos como a jornada do usuário, o design do jogo, a motivação dos jogadores e a implementação de mecanismos de feedback.

2.3 Definições e conceitos da web

O presente trabalho utiliza-se de vários tipos de tecnologias web para a sua construção, o que torna fundamental a apresentação dos conceitos mais relevantes destas tecnologias. Nesta seção, serão apresentados aspectos relevantes da web, incluindo a sua evolução, protocolos e tecnologias utilizadas na criação de sistemas e aplicações. Também será discutido o papel das tecnologias web na melhoria da implementação deste trabalho. É importante destacar que as informações apresentadas nesta seção serão de suma importância para contextualizar e entender a proposta deste trabalho.

2.3.1 Web

De acordo com [Carvalho \(2006\)](#), a Web é um sistema de hipertexto que permite aos usuários acrescentar informações e acessá-las por meio de computadores conectados à internet. Para tal, é necessário uma conexão com a internet e um navegador para visualizar os conteúdos disponíveis. A Web é frequentemente descrita como tendo passado por três eras, e atualmente estar entrando em uma quarta era, que podem ser definidas da seguinte forma:

- **Web 1.0:** Esta era da web foi marcada pelo surgimento de páginas estáticas, que não eram interativas e não permitiam a atualização ou edição pelo usuário. A principal função da web nesta época era a de compartilhar informações através de páginas simples e estáticas.
- **Web 2.0:** Nesta era, a web evoluiu para permitir a interatividade e a colaboração entre os usuários. A web 2.0 permitiu a criação de blogs, wikis, redes sociais e outras plataformas que possibilitam aos usuários criarem e compartilharem conteúdos.
- **Web 3.0:** A era da web 3.0 é marcada pelo surgimento de tecnologias que possibilitam a inteligência artificial, a análise de dados e o acesso à informação de forma mais eficiente e personalizada. A web 3.0 é vista como a era da "Semântica da Web", na qual as informações estão organizadas de forma a torná-las mais acessíveis e relevantes para os usuários.
- **Web 4.0:** É uma era hipotética da web que ainda não foi completamente definida ou implementada, mas é amplamente discutida como sendo a próxima evolução da web. É considerada como uma era onde a inteligência artificial, a Internet das Coisas e outras tecnologias avançadas se juntam para fornecer experiências de usuário ainda mais integradas e personalizadas.

2.3.2 Website

De acordo com [Gonçalves \(2022\)](#), um website é caracterizado como uma coleção de páginas da web, que são organizadas e hospedadas em um servidor na rede de computadores. Essas páginas são acessíveis ao público através da internet, bastando digitar o endereço do website na barra de pesquisa de um navegador de internet para acessar as informações e conteúdos oferecidos.

2.3.3 Sistemas web

De acordo com [Santos \(2018\)](#), um sistema web é diferenciado de um website convencional, pois ele oferece uma experiência personalizada ao usuário. Enquanto um

website apresenta conteúdo estático, restringindo as interações do usuário, um sistema web é uma aplicação que permite uma ampla gama de ações, tais como a visualização, edição e exclusão de informações vindas de fontes variadas, tais como bancos de dados e servidores. Portanto, um sistema web pode ser entendido como uma aplicação ou programa de computador, em vez de uma simples página estática.

Esses sistemas apresentam praticidade, pois o usuário não precisa instalar nenhum *software* localmente em sua máquina e pode acessá-los remotamente através da internet. Além disso, a natureza dinâmica dos sistemas web permite uma ampla flexibilidade, permitindo a customização da experiência do usuário de acordo com suas necessidades.

2.4 Desenvolvimento web

O desenvolvimento web é uma área da tecnologia da informação que se dedica ao desenvolvimento de aplicações e sistemas web. É composto por uma combinação de linguagens de programação, ferramentas e tecnologias que são usadas para construir sites e aplicativos para a web.

O desenvolvimento web é uma área em constante evolução e novas tecnologias e ferramentas são lançadas com frequência. Algumas dessas tecnologias foram utilizadas neste trabalho, portanto, serão apresentadas logo abaixo:

2.4.1 HTML

O *HyperText Markup Language (HTML)*⁵ é uma linguagem de marcação de hipertexto. Esta tecnologia é o "bloco de construção" mais básico utilizado na web. O HTML é o responsável por definir as estruturas das páginas que serão apresentadas, tais como seus textos, imagens, *hyperlinks*, vídeos e outros.

A marcação HTML inclui tags, como `<head>`, `<title>`, `<body>`, `<header>`, `<footer>`, `<article>`, `<section>`, `<p>`, `<div>`, ``, ``, `<aside>`, `<audio>`, `<canvas>`, `<datalist>`, `<details>`, `<embed>`, `<nav>`, `<output>`, `<progress>`, `<video>`, ``, ``, `` e a sua disposição de maneira ordenada serve para fazer a marcação da página.

Segue um exemplo de código HTML.

⁵ Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTML>

Figura 1 – Exemplo de código HTML

```
<!DOCTYPE html> Untitled-1
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Exemplo de pagina HTML</title>
5   </head>
6   <body>
7     <h1>Meu primeiro cabecalho</h1>
8     <p>Este e um paragrafo simples.</p>
9     <ul>
10      <li>Item 1</li>
11      <li>Item 2</li>
12      <li>Item 3</li>
13    </ul>
14  </body>
15 </html>
16
```

Fonte: Elaborada pelo autor

2.4.2 CSS

O *Cascading Style Sheets* (CSS)⁶ é uma linguagem de estilo utilizada para descrever a apresentação de um documento escrito em HTML. Ele desempenha um papel importante na construção de páginas da web, pois permite ao desenvolvedor controlar aspectos visuais, como cor de fundo, fonte, espaçamento entre parágrafos e outros elementos. Além disso, o CSS permite a definição de regras de apresentação, como a quantidade de elementos que serão exibidos por linha na página.

Segue um exemplo de código CSS.

⁶ Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTML>

Figura 2 – Exemplo de código CSS

```
# body { Untitled-1 ●
1  body {
2    font-family: Arial, sans-serif;
3    font-size: 16px;
4    color: #333;
5    background-color: #fff;
6  }
7
8  h1 {
9    font-size: 32px;
10   color: #4CAF50;
11   text-align: center;
12   margin-top: 50px;
13 }
14
15 p {
16   line-height: 1.5;
17   margin-bottom: 20px;
18 }
```

Fonte: Elaborada pelo autor.

2.4.3 JavaScript

O JavaScript⁷ é uma linguagem de programação interpretada e multi-paradigma que é amplamente utilizada no contexto web. É uma das linguagens mais importantes e reconhecidas nos nossos navegadores, e existem também outras formas de utilizá-la fora do browser, como, por exemplo, com o Node.js.

Como linguagem multi-paradigma, o JavaScript suporta programação orientada a eventos, programação funcional e programação imperativa. Isso significa que os programadores têm a flexibilidade de escolher a abordagem que melhor se adapta às suas necessidades e ao problema que desejam resolver.

No contexto web, o JavaScript pode ser usado para uma ampla gama de tarefas, como realizar chamadas para servidores de bancos de dados, armazenamento de dados em memória, controle da exibição de elementos em páginas web, criação de linhas em tabelas de forma dinâmica, criação de interfaces de usuário, desenvolvimento de servidores, APIs e microsserviços.

Com a capacidade de ser utilizado para realizar tarefas dinâmicas, o JavaScript é uma linguagem crucial para o desenvolvimento de aplicações web eficientes e interativas.

A figura a seguir mostra um exemplo de um código JavaScript:

⁷ Disponível em: <https://www.javascript.com/>

Figura 3 – Exemplo de código JavaScript

```
JS var numeroAleatorio = Math.floor(Math.random() * 10) + 1;
1  var numeroAleatorio = Math.floor(Math.random() * 10) + 1;
2  var tentativas = 3;
3
4  while (tentativas > 0) {
5      var palpite = prompt("Tente adivinhar o número entre 1 e 10. Você tem " + tentativas + " tentativa(s) restante(s).");
6      palpite = parseInt(palpite);
7
8      if (palpite === numeroAleatorio) {
9          alert("Parabéns! Você acertou o número.");
10         break;
11     } else {
12         tentativas--;
13         if (tentativas === 0) {
14             alert("Você perdeu. O número era " + numeroAleatorio + ".");
15         } else {
16             alert("Tente novamente.");
17         }
18     }
19 }
```

Fonte: Elaborada pelo autor.

2.4.4 HTTP

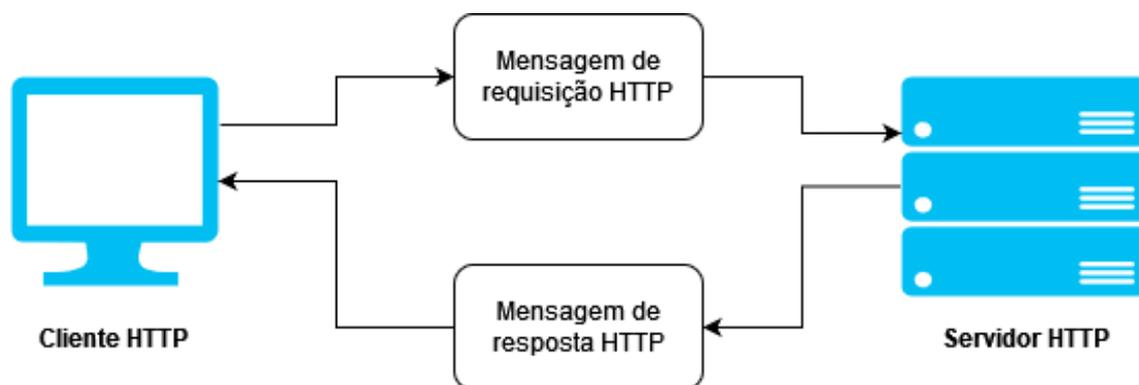
O *HyperText Transfer Protocol* ([HTTP](#)) é um protocolo de comunicação utilizado na internet, que permite a transferência de dados em uma rede IP. De acordo com [Gaspar \(2019\)](#), este protocolo possui características específicas, como a falta de estado, o que significa que ele não mantém informações sobre o histórico das transações realizadas.

O processo de comunicação é iniciado pelo cliente, que envia uma mensagem, também conhecida como requisição, solicitando um recurso ou enviando um comando para o servidor. A requisição é composta por diversos elementos, incluindo a URL do recurso desejado, o método HTTP utilizado e os cabeçalhos HTTP.

O servidor, por sua vez, responde à requisição enviando uma mensagem, que é recebida pelo cliente que a iniciou. Essa mensagem pode conter, por exemplo, os dados solicitados, informações de erro ou aviso de que o recurso não foi encontrado. A resposta do servidor também é composta por diversos elementos, incluindo o status HTTP, os cabeçalhos HTTP e o corpo da resposta.

O exemplo a seguir ilustra o funcionamento básico do HTTP:

Figura 4 – Representação de requisição e resposta do protocolo HTTP



Fonte: Elaborada pelo autor.

2.4.4.1 Cabeçalho da requisição

O cabeçalho de uma requisição é um elemento importante que permite enviar informações adicionais e complementares à própria requisição. Segundo a literatura especializada, existem quatro tipos distintos de cabeçalhos, são eles:

- **Request headers:** Contêm mais informações sobre o recurso a ser buscado ou sobre o cliente que está solicitando o recurso.
- **Response headers:** Contêm informações adicionais sobre a resposta, como sua localização ou sobre o servidor que a fornece.
- **Representation headers:** Contêm informações sobre o corpo do recurso.
- **Payload headers:** Contêm informações sobre dados de *payload*, incluindo o comprimento do conteúdo e a codificação usada para envio.

2.4.4.2 Corpo da Requisição

O corpo da requisição é uma parte fundamental do protocolo HTTP, pois é através dele que informações são enviadas ou recebidas pelo servidor. Este conjunto de dados é responsável por transmitir informações específicas, que podem ser relacionadas a cadastros de usuários em sistemas ou a informações de produtos em lojas virtuais, por exemplo.

Neste contexto, o corpo da mensagem pode ser transmitido de diversas formas, mas para fins deste trabalho, será considerado o formato *JavaScript Object Notation (JSON)* (JavaScript Object Notation). O JSON é uma estrutura de dados utilizada para representar objetos e seus respectivos dados, conforme ilustrado no exemplo:

Figura 5 – Estrutura de um objeto/arquivo JSON

```
1  {
2    "pessoa": {
3      "nome": "Maria",
4      "idade": 30,
5      "endereco": {
6        "rua": "Rua A",
7        "numero": 123,
8        "bairro": "Centro",
9        "cidade": "Belo Horizonte",
10       "estado": "MG"
11     }
12   },
13   "produtos": [
14     {
15       "nome": "Sapato",
16       "preco": 129.90,
17       "disponivel": false
18     }
19   ]
20 }
```

Fonte: Elaborada pelo autor.

2.4.4.3 Métodos da Requisição

O protocolo HTTP define ações que serão executadas pela requisição, através de métodos. Existem nove métodos definidos no protocolo, mas os mais comumente utilizados são:

- **GET**: Este método solicita a representação de um recurso específico;
- **POST**: Este método é utilizado para enviar informações para uma ação específica, como o cadastro de dados de um usuário;
- **PUT**: Este método substitui todas as representações atuais do recurso de destino com os dados enviados na requisição;
- **PATCH**: Este método aplica modificações parciais em um recurso específico;
- **DELETE**: Este método remove um recurso específico.

Cabe destacar que o protocolo HTTP é agnóstico quanto à função acionada pelo servidor, o que significa que uma chamada de método *DELETE* pode ser utilizada para inserir um usuário, por exemplo. No entanto, é uma prática inadequada e deve ser evitada.

2.4.4.4 Códigos de resposta

Os códigos de resposta do protocolo HTTP são códigos enviados pelo servidor com a resposta da requisição, e são usados para que o usuário possa avaliar se a requisição foi bem sucedida ou não, e para entender a natureza do erro, caso ocorra.

Alguns dos códigos de resposta mais comuns incluem:

- **200 OK**: A requisição foi bem sucedida;
- **201 Created**: O recurso foi criado com sucesso;
- **204 No Content**: A requisição foi bem sucedida, mas não há representação a ser retornada;
- **400 Bad Request**: O servidor não consegue entender a requisição devido a sintaxe inválida;
- **401 Unauthorized**: A requisição requer autenticação do usuário;
- **403 Forbidden**: O servidor entende a requisição, mas se recusa a autorizar;
- **404 Not Found**: O recurso não foi encontrado;
- **500 Internal Server Error**: Ocorreu um erro no servidor.

2.4.5 Bancos de dados

O conceito de banco de dados remonta a tempos antigos e pode ser definido como uma coleção de dados inter-relacionados que permitem a recuperação de informações, segundo [Cianconi \(1987\)](#). Esses dados são geralmente gerenciados por um Sistema Gerenciador de Bancos de Dados (SGBD) (ou do inglês *Data Base Management System (DBMS)*), que fornece acesso, segurança e garantia de integridade dos dados.

Os bancos de dados podem ser classificados em dois tipos principais: bancos de dados relacionais e bancos de dados não relacionais. O banco de dados relacional é baseado em tabelas, onde os dados são armazenados em relações entre tabelas. Já o banco de dados não relacional é baseado em documentos, chaves e valores ou gráficos, dependendo do tipo escolhido.

No escopo deste trabalho, foi escolhido um banco de dados não relacional devido a sua performance elevada e ao fato de que não é exigida uma relação tão estreita entre os dados.

2.4.5.1 Bancos não relacionais (NoSQL)

No contexto deste trabalho, foi utilizado o MongoDB⁸, um banco de dados não relacional, por sua performance e flexibilidade na estruturação dos dados. Em vez de tabelas relacionadas, o MongoDB usa *collections* para armazenar informações. Cada *collection* representa, em certa medida, uma tabela no modelo relacional.

Bancos de dados não relacionais são adequados para ambientes que requerem velocidade e baixa latência, especialmente com grande volume de dados e estrutura flexível. No MongoDB, uma *collection* pode conter dados de diferentes tipos e estruturas sem seguir as normas do modelo relacional.

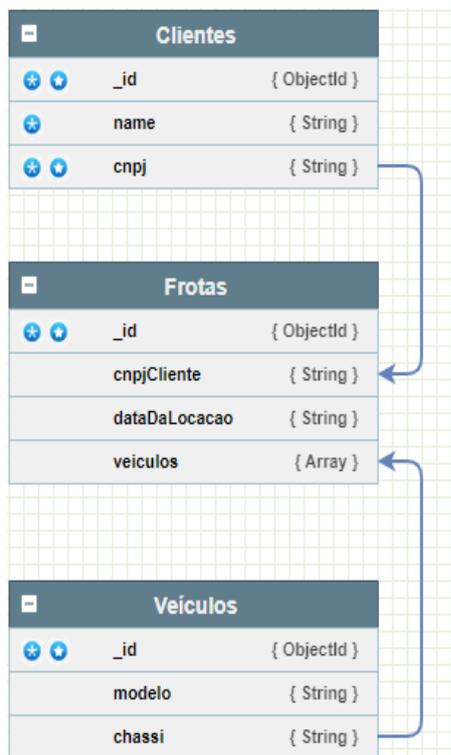
Com o uso de referências, é possível estabelecer uma ligação entre *collections*, permitindo acesso a informações relacionadas de forma eficiente. Este banco de dados foi escolhido para armazenar questões e dados do usuário, com a vantagem de oferecer performance e escalabilidade.

No esquema, *collections* podem se relacionar por meio de referências. Por exemplo, é possível inserir códigos de chassis de veículos em uma lista de veículos na *collection* de Frotas, e o CNPJ do cliente que contratou aquela frota também na *collection* de Frotas. No entanto, é importante ter cautela para evitar inconsistências nos dados.

A figura a seguir mostra um esquema relacional:

⁸ Disponível em: <https://www.mongodb.com/>

Figura 6 – Esquema não relacional simples de uma locadora de veículos



Fonte: Elaborada pelo autor.

2.4.6 API

Uma *Application Programming Interface* (API) é um conjunto de rotinas e padrões para acesso a aplicações ou bancos de dados. Ela permite que programas acessem recursos de outras aplicações, realizando tarefas como consultar, adicionar ou atualizar informações, sem a necessidade de conhecer a estrutura interna da aplicação.

Neste trabalho, as APIs se aplicam como uma forma de permitir que aplicações externas acessem os dados e as funcionalidades disponíveis. Ela é fundamental para a implementação de uma arquitetura de microserviços, permitindo a integração de diferentes aplicações e serviços de forma simples e eficiente.

A abordagem *Representational State Transfer* (REST) foi escolhida para este trabalho devido a sua simplicidade, escalabilidade e capacidade de integração com diferentes tecnologias. O modelo REST permite que aplicações cliente e servidor se comuniquem de forma simples e eficiente, usando HTTP para transferir dados. Em REST, os recursos são identificados por *Uniform Resource Identifiers* (URIs) e acessados através de métodos HTTP, como *GET*, *POST*, *PUT* e *DELETE*.

Existem dois tipos de APIs que seguem o estilo REST: APIs RESTful e APIs

RESTless. A diferença entre ambas está no nível de maturidade e aderência à arquitetura REST.

Quando uma API segue todas as definições estabelecidas pelo modelo REST ela é chamada de API RESTful. Para o escopo deste trabalho, não haveria necessidade de seguir todos os critérios de aceitação de uma API RESTful, porém, grande parte das restrições foram cumpridas.

2.4.7 Microsserviços

Segundo a definição disponibilizada pela [AWS \(2022\)](#), um microsserviço pode ser descrito como:

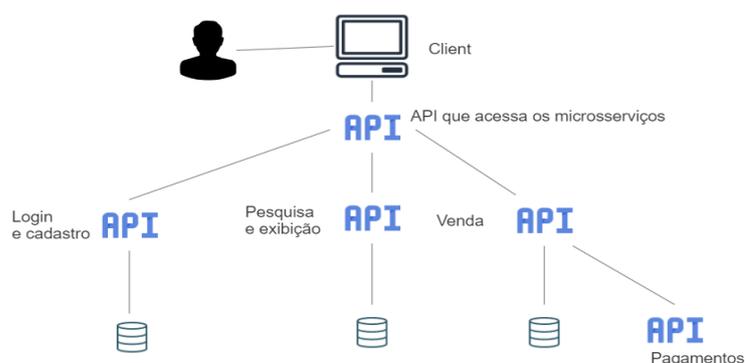
Uma abordagem arquitetônica e organizacional do desenvolvimento de software na qual o software consiste em pequenos serviços independentes que se comunicam usando APIs bem definidas. Esses serviços pertencem a pequenas equipes autossuficientes.

Esse tipo de abordagem arquitetural é altamente escalável e eficiente, pois, cada serviço da aplicação existe de forma individual, onde, normalmente, há uma API principal responsável por orquestrar todos estes serviços.

Um bom exemplo, seria o de uma loja online de varejo, onde existe um microsserviço responsável pelo login e cadastro dos usuários, outro para a exibição e pesquisa de produtos, um que realiza a venda dos produtos e outro que se integra com este último para ser o responsável pelas transações bancárias envolvidas.

Segue um exemplo de como seria a arquitetura do produto descrito acima:

Figura 7 – Esquema arquitetural de microsserviços



Fonte: Elaborada pelo autor.

2.5 Bibliotecas

Para entendermos os conceitos que serão apresentados na seção seguinte, primeiro, temos que entender o que é uma biblioteca. Segundo [Silva \(2020\)](#) uma biblioteca, no contexto da computação, é uma coleção de subprogramas que utilizamos no desenvolvimento de um determinado software.

Durante o desenvolvimento de uma funcionalidade, é bem provável que a linguagem escolhida não seja capaz, em sua forma pura, de executar determinada ação. Um exemplo, num contexto de desenvolvimento JavaScript, a linguagem não possui nenhum recurso pronto para calcular a diferença de dias entre duas datas, portanto, será necessário implementar essa funcionalidade.

Imagine que no contexto citado acima, serão necessárias mais algumas outras implementações para realizar operações com datas, que não foram disponibilizadas pela linguagem. Ao criar todas essas funcionalidades, seria altamente viável agrupá-las numa coleção de funcionalidades, assim, durante o desenvolvimento de outra aplicação, essas implementações poderiam ser reaproveitadas, poupando muito tempo de desenvolvimento na resolução de problemas que já foram resolvidos. Este é o conceito por trás das bibliotecas, sua utilização aumenta muito o desempenho na hora do desenvolvimento, poupando muito tempo de trabalho e deixando o código mais limpo, mais legível e menor.

2.6 Ferramentas e tecnologias utilizadas

2.6.1 TypeScript

A Seção 2.4.3 foi dedicada à definição do JavaScript e à apresentação das suas aplicações principais. No entanto, em 2012, a Microsoft deu início ao desenvolvimento do TypeScript⁹.

O TypeScript é uma linguagem de programação de código aberto, fortemente tipada, que proporciona inúmeras funcionalidades e possibilidades adicionais em relação ao JavaScript, tornando-se uma ferramenta de desenvolvimento muito mais robusta e eficiente..

Com sua forte tipagem, o TypeScript permite aos desenvolvedores escrever códigos de forma mais clara e organizada, além de permitir a detecção de erros em tempo de compilação, o que é uma grande vantagem em comparação com o JavaScript. Além disso, o TypeScript suporta a programação orientada a objetos, o que é uma característica fundamental em muitos projetos de desenvolvimento de software.

⁹ Disponível em: <https://www.typescriptlang.org/>

Na Tabela 1, estão algumas das principais diferenças entre o JavaScript e o TypeScript:

Tabela 1 – Tabela comparativa entre linguagens.

JavaScript	TypeScript
Desenvolvida para criar conteúdo dinâmico para páginas web.	Criada para desenvolver projetos de alta complexidade e alta escalabilidade.
É necessário executar o código para verificar possíveis erros.	É compilada, possibilitando a identificação de erros, sugerindo correções antes da execução do código e gerando economia de tempo.
É compreendida pelo navegador.	É convertida para JavaScript para que o navegador entenda.
Tipagem fraca e sem a opção de tipagem estática.	Fortemente tipada, suporta tanto tipagem estática quanto tipagem dinâmica.
Bibliotecas JavaScript funcionam por padrão.	Suporta todas as bibliotecas e qualquer tipo de código nativo JavaScript. Desde a sua criação, várias bibliotecas foram adaptadas para suportarem a tipagem do TypeScript.
Não suporta interfaces, tipos genéricos, interfaces ou enumeradores.	Suporta interfaces, tipos genéricos, interfaces e enumeradores.
As funções não podem ter parâmetros opcionais.	As funções podem ter parâmetros opcionais.
Comunidade de desenvolvimento gigantesca.	Comunidade de desenvolvimento em crescimento muito expressivo.

Fonte: Elaborada pelo autor.

Embora a tipagem forte do TypeScript possa exigir um pouco mais de esforço inicial para o aprendizado, os ganhos em termos de velocidade na detecção de erros, legibilidade e compreensão do código, e diversas outras vantagens, tornam o TypeScript uma escolha mais viável e adequada para o escopo deste trabalho.

2.6.2 React

O React¹⁰ é uma biblioteca de criação de interfaces de usuário que permite compor elementos complexos a partir de pequenos e isolados códigos chamados componentes. O projeto foi iniciado pelo Facebook para a criação de visualizações baseadas em componentes, visando facilitar a sincronização de atividades simultâneas na rede social. Os componentes permitem que a tela seja decomposta em vários blocos separados, onde cada um pode ter estados, propriedades, comportamentos e usos diferentes, tornando o código mais curto,

¹⁰ Disponível em: <https://reactjs.org/>

organizado e performático. A versão 16.8 da biblioteca introduziu os *hooks*, que permitem que os componentes sejam escritos como funções e acrescentam novas funcionalidades à biblioteca.

A escolha do React para este trabalho se deu por sua flexibilidade e escalabilidade, juntamente com a capacidade de compor interfaces complexas a partir de componentes simples, tornando o desenvolvimento mais eficiente e fácil de manter. Além disso, o React é uma tecnologia amplamente utilizada e com uma comunidade ativa de desenvolvedores, o que garante uma grande quantidade de recursos e soluções disponíveis. Para o escopo deste trabalho, o React foi integrado a uma outra biblioteca para a criação das interfaces de usuário, o MUI.

2.6.2.1 MUI

A MUI¹¹, ou Material-UI, é uma biblioteca de componentes de interface de usuário que segue o conjunto de padrões e práticas do *design system*¹² Material Design da Google. Esta biblioteca permite aos desenvolvedores implementar aparências visuais consistentes e coerentes com o design system Material Design, sem precisar desenvolver componentes personalizados do zero. O uso da MUI garante uma experiência de usuário uniforme e de alta qualidade para os usuários finais, além de ajudar a agilizar o processo de desenvolvimento, uma vez que muitos componentes já estão prontos para uso.

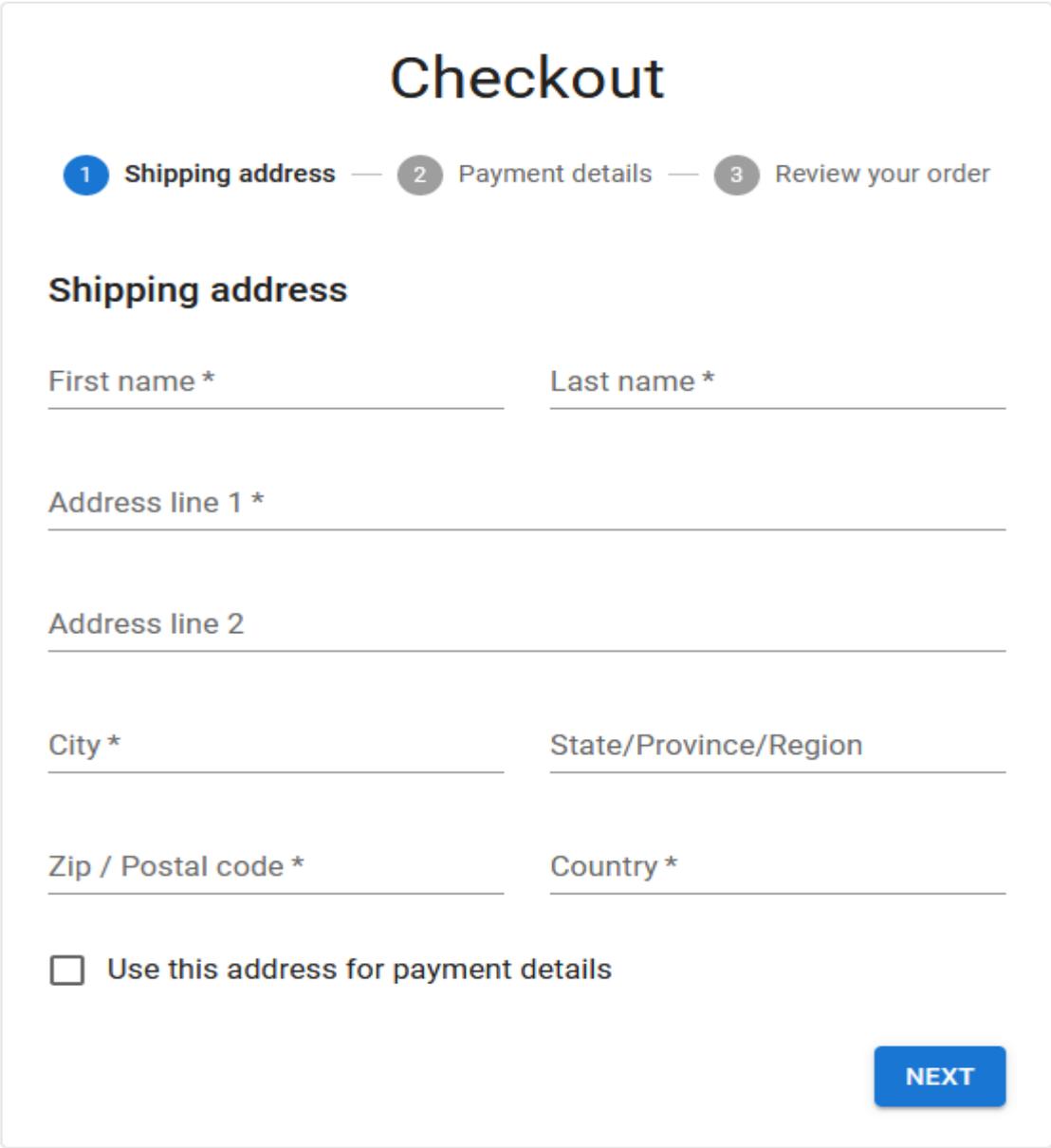
Para este projeto, a escolha da biblioteca MUI se deu por sua capacidade em proporcionar uma interface visualmente atraente e padronizada, ajustando-se aos padrões de design de sistemas comumente encontrados na web. Além disso, a MUI é uma ferramenta extremamente completa, com uma ampla variedade de implementações internas, tornando o processo de desenvolvimento mais ágil e dinâmico.

A seguir, um exemplo de uma tela construída usando o Material Design:

¹¹ Disponível em: <https://mui.com/>

¹² Um conjunto de padrões interconectados e práticas compartilhadas organizadas de forma coerente

Figura 8 – Exemplo de tela contruida com Material Design



The image shows a checkout form titled "Checkout" with a progress indicator at the top showing three steps: "1 Shipping address" (active), "2 Payment details", and "3 Review your order". The form is divided into sections for "Shipping address" and "Payment details". The "Shipping address" section includes fields for "First name *", "Last name *", "Address line 1 *", "Address line 2", "City *", "State/Province/Region", "Zip / Postal code *", and "Country *". There is a checkbox labeled "Use this address for payment details". A blue "NEXT" button is located at the bottom right of the form.

Fonte: Elaborada pelo autor.

2.6.3 Node.js

O Node.js¹³ é uma plataforma de desenvolvimento altamente versátil e poderosa que permite a execução de códigos escritos em JavaScript e TypeScript fora do ambiente do navegador. Isso é possível graças ao uso do Chrome's V8 JavaScript Engine, que não apenas interpreta, mas também compila e executa o código. Com o Node.js, é possível criar

¹³ Disponível em: <https://nodejs.org/en/>

aplicações robustas e escaláveis, que podem ser executadas em uma ampla variedade de dispositivos e sistemas operacionais. Além disso, o Node.js possui uma ampla comunidade de desenvolvedores e uma ampla gama de bibliotecas e pacotes para ajudar na criação de aplicações avançadas.

O Node.js é projetado para funcionar de forma assíncrona, o que significa que ele foi desenvolvido para não bloquear o desempenho da aplicação durante a execução. Para fazer isso, o Node.js delega processos demorados para segundo plano, mantendo a fluidez da aplicação. Além disso, o Node.js é *single-thread*, o que significa que todos os processos do servidor são executados em uma única *thread*, ao invés de serem executados em várias *threads* separadas. Essa arquitetura é benéfica por simplificar a administração do código e torná-lo mais fácil de gerenciar, além de ser mais eficiente em termos de recursos do sistema.

Por ser multi-plataforma, multi-paradigma, ser bastante eficiente e escalável, o Node.js acaba sendo uma das primeiras opções quando se fala de desenvolvimento web. Para a elaboração deste trabalho, ele foi usado em conjunto com o *framework*¹⁴ Express.js.

2.6.3.1 Express.js

O Express.js¹⁵ é um framework para Node.js que possui vários recursos para a criação de servidores HTTP. Ele fornece uma série de recursos e ferramentas para ajudar na criação de aplicações web de forma rápida e eficiente e é amplamente utilizado devido à sua facilidade de uso e flexibilidade. É uma escolha popular entre desenvolvedores de aplicações web devido a sua simplicidade e capacidade de integração com outras bibliotecas e tecnologias. As APIs implementadas neste trabalho foram contruídas utilizando Express.js.

2.6.4 MongoDB

O MongoDB¹⁶ é uma plataforma de banco de dados não relacional de código aberto que é altamente eficiente e escalável. Ele permite o armazenamento de grandes quantidades de informações de forma não estruturada, as quais são organizadas em *collections*. Cada documento dentro de uma *collection* é único e identificado por um ObjectId, que é um campo identificador exclusivo.

A escolha do MongoDB para este trabalho se deve à sua eficiência e capacidade de lidar com quantidades significativas de dados, além da flexibilidade oferecida pelo armazenamento não estruturado de informações, o que é muito útil, visto que as estruturas de dados e o escopo do trabalho poderiam se alterar.

¹⁴ Definição: Uma ferramenta que possui várias estruturas e implementações prontas para serem usadas.

¹⁵ Disponível em: <https://expressjs.com/>

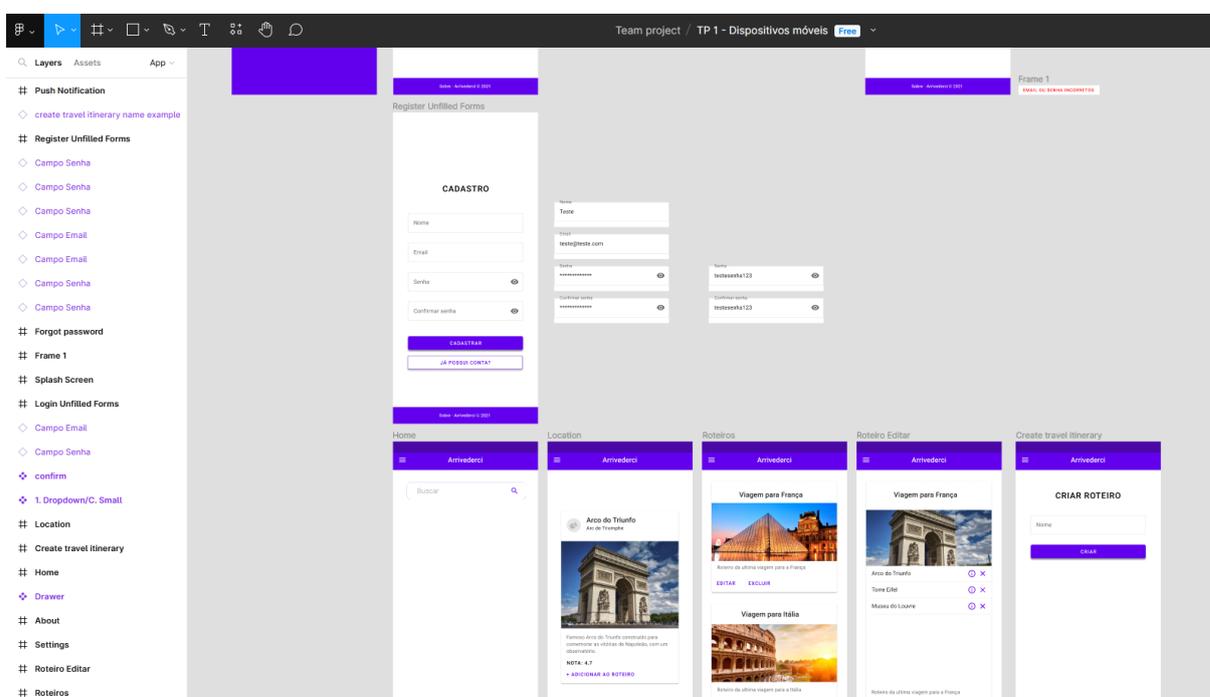
¹⁶ Disponível em: <https://www.mongodb.com/home>

2.6.5 Figma

O Figma¹⁷ é uma ferramenta de prototipação de interfaces altamente eficiente, amplamente utilizada no setor de *design* para planejamento e criação de *layouts* de aplicações. Ele se destaca como a ferramenta líder de seu segmento no mercado de trabalho devido a sua capacidade de permitir a colaboração simultânea de vários usuários em um único projeto. Além disso, o Figma oferece uma ampla variedade de recursos e ferramentas avançadas que tornam o processo de criação de protótipos mais fácil e eficiente, o que o torna uma escolha ideal para o escopo deste trabalho.

Nesta ferramenta o usuário também pode definir os fluxos da aplicação, fazendo com o os testes do protótipo sejam mais eficientes e assertivos.

Figura 9 – Exemplo de projeto construído no Figma



Fonte: Retirado de Figma (2023).

2.6.6 Git e GitHub

O Git¹⁸ é uma ferramenta de gerenciamento de versão de código que foi projetada pelo desenvolvedor finlandês, Linus Torvalds, também conhecido como o criador do sistema operacional Linux. Ele é amplamente utilizado no desenvolvimento de *software* e permite que o histórico de edições dos arquivos do projeto seja armazenado de forma eficiente, garantindo que nada seja perdido durante o desenvolvimento. Além disso, o Git controla

¹⁷ Disponível em: <https://www.figma.com/>

¹⁸ Disponível em: <https://git-scm.com/>

as versões das atualizações de um projeto, permitindo que desenvolvedores colaborem e garantindo a integridade do código.

Já o GitHub¹⁹ é uma plataforma de hospedagem de código online que utiliza as funcionalidades do Git como base. Ele permite que desenvolvedores contribuam com projetos públicos e privados, além de proporcionar uma ampla comunidade de desenvolvedores com quem é possível colaborar e aprender. No escopo deste trabalho, o GitHub é escolhido como uma das ferramentas para o controle de versão do código fonte dos projetos, pois oferece uma solução integrada e de fácil acesso para o armazenamento e compartilhamento do código.

2.6.7 Docker

O Docker²⁰ é uma tecnologia de virtualização de ambientes em *containers* que permite a criação de imagens contendo todas as dependências necessárias para a execução de um projeto. Essas imagens são executadas de forma isolada, garantindo a reprodução do mesmo ambiente em diferentes máquinas. No âmbito deste trabalho, o Docker foi escolhido para a execução de uma instância do MongoDB, permitindo o armazenamento local dos dados da aplicação, garantindo a compatibilidade e portabilidade do projeto. Além disso, a utilização do Docker facilita a manutenção e atualização do ambiente de desenvolvimento.

2.7 Considerações

Ao longo deste capítulo, foram apresentados conceitos importantes relacionados ao trabalho e ao ensino superior, como o processo de vestibular e ingresso na faculdade, a gamificação, os sistemas web, o desenvolvimento web, as bibliotecas e as ferramentas e tecnologias utilizadas. Ao descrever os conceitos e ferramentas utilizadas, é mais fácil entender o porquê do uso das mesmas tanto para o processo de desenvolvimento quanto para o resultado final da aplicação, assim como se relacionam com as motivações para a realização do projeto.

No próximo capítulo, serão considerados os trabalhos correlatos à este, apresentando as principais semelhanças e diferenças entre os sistemas apresentados e o sistema implementado para o escopo deste trabalho.

¹⁹ Disponível em: <https://github.com/>

²⁰ Disponível em: <https://www.docker.com/>

3 Trabalhos correlatos

Antes da implementação do trabalho, foi realizada uma pesquisa para analisar as plataformas similares existentes no mercado. Após a análise, foi identificado que todas elas cumpriam seu propósito de forma satisfatória, porém apresentavam diferenças que serão detalhadas nas seções abaixo.

3.1 Simulador do Brasil Escola

O simulador do Brasil Escola se destaca por contar com uma grande quantidade de questões de edições passadas do ENEM. Entretanto, a plataforma não oferece uma forma clara e direta de acessar estas questões.

A plataforma cumpre sua função de gerar simulados de maneira eficiente, mas apresenta algumas limitações quanto ao número de questões disponíveis. É possível gerar simulados apenas com 180 questões ou com um número específico de questões correspondente a um determinado conteúdo do ENEM, como, por exemplo, 45 questões de ciências da natureza, 45 questões de matemática, entre outros. Além disso, a plataforma não oferece nenhum tipo de gamificação para tornar a experiência de estudo mais atrativa e motivadora para o usuário.

Segue o exemplo de uma imagem da interface inicial do Simulador do Brasil Escola:

Figura 10 – Simulador do Brasil Escola

Prontos

 **Ciências da Natureza**

45 Questões de Biologia, Física e Química.

[Iniciar](#)

 **Matemática**

45 Questões de Matemática e suas Tecnologias.

[Iniciar](#)

 **Linguagens e Códigos**

45 Questões de Língua Portuguesa, Literatura, Inglês ou Espanhol, Artes e Educação Física.

Espanhol Inglês

[Iniciar](#)

 **Ciências Humanas**

45 Questões de História, Geografia, Filosofia e Sociologia.

[Iniciar](#)

1 **1º Dia do ENEM**

90 Questões de Linguagens e Códigos e Ciências Humanas.

Espanhol Inglês

[Iniciar](#)

2 **2º Dia do ENEM**

90 Questões de Matemática e Ciências da Natureza.

[Iniciar](#)

 **ENEM Completo**

180 Questões de todas as áreas de conhecimento do ENEM.

Espanhol Inglês

[Iniciar](#)

Crie o seu

Ciências da Natureza:

Matemática:

Linguagens e Códigos:

Ciências Humanas:

Linguagens e Códigos (Espanhol):

Linguagens e Códigos (Inglês):

Faltam: **180**Total:

Os simulados gerados poderão conter até **180** questões.

[Limpar](#) [Iniciar](#)

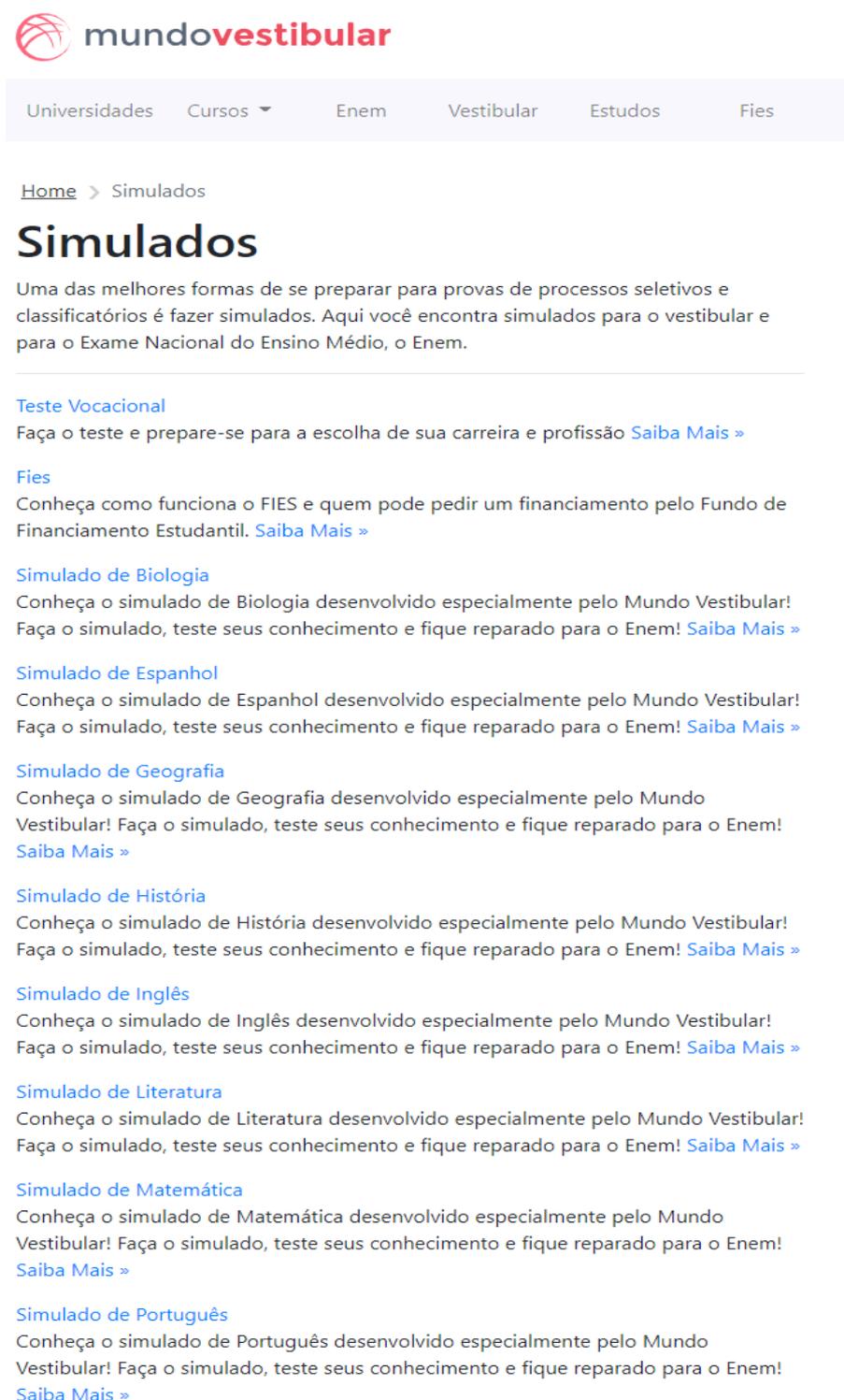
Fonte: Retirado de [Brasil Escola \(2023\)](#).

3.2 Simulador Mundo Vestibular

O simulador do Mundo Vestibular apresenta uma interface simples, mas comum. Neste simulador, o usuário pode escolher uma disciplina e gerar um simulado aleatório com um número fixo de questões. Esse número varia de acordo com o conteúdo escolhido. A

plataforma não oferece uma forma para que desenvolvedores acessem as questões disponíveis e também não conta com nenhum tipo de gamificação.

Figura 11 – Simulador do Mundo Vestibular



 **mundovestibular**

Universidades Cursos ▾ Enem Vestibular Estudos Fies

Home > Simulados

Simulados

Uma das melhores formas de se preparar para provas de processos seletivos e classificatórios é fazer simulados. Aqui você encontra simulados para o vestibular e para o Exame Nacional do Ensino Médio, o Enem.

[Teste Vocacional](#)
Faça o teste e prepare-se para a escolha de sua carreira e profissão [Saiba Mais »](#)

[Fies](#)
Conheça como funciona o FIES e quem pode pedir um financiamento pelo Fundo de Financiamento Estudantil. [Saiba Mais »](#)

[Simulado de Biologia](#)
Conheça o simulado de Biologia desenvolvido especialmente pelo Mundo Vestibular! Faça o simulado, teste seus conhecimentos e fique preparado para o Enem! [Saiba Mais »](#)

[Simulado de Espanhol](#)
Conheça o simulado de Espanhol desenvolvido especialmente pelo Mundo Vestibular! Faça o simulado, teste seus conhecimentos e fique preparado para o Enem! [Saiba Mais »](#)

[Simulado de Geografia](#)
Conheça o simulado de Geografia desenvolvido especialmente pelo Mundo Vestibular! Faça o simulado, teste seus conhecimentos e fique preparado para o Enem! [Saiba Mais »](#)

[Simulado de História](#)
Conheça o simulado de História desenvolvido especialmente pelo Mundo Vestibular! Faça o simulado, teste seus conhecimentos e fique preparado para o Enem! [Saiba Mais »](#)

[Simulado de Inglês](#)
Conheça o simulado de Inglês desenvolvido especialmente pelo Mundo Vestibular! Faça o simulado, teste seus conhecimentos e fique preparado para o Enem! [Saiba Mais »](#)

[Simulado de Literatura](#)
Conheça o simulado de Literatura desenvolvido especialmente pelo Mundo Vestibular! Faça o simulado, teste seus conhecimentos e fique preparado para o Enem! [Saiba Mais »](#)

[Simulado de Matemática](#)
Conheça o simulado de Matemática desenvolvido especialmente pelo Mundo Vestibular! Faça o simulado, teste seus conhecimentos e fique preparado para o Enem! [Saiba Mais »](#)

[Simulado de Português](#)
Conheça o simulado de Português desenvolvido especialmente pelo Mundo Vestibular! Faça o simulado, teste seus conhecimentos e fique preparado para o Enem! [Saiba Mais »](#)

Fonte: Retirado de [Mundo Vestibular \(2023\)](#).

3.3 Simulador Beduka

O simulador da Beduka possui comportamento semelhante ao simulador do Brasil Escola descrito na Seção 3.1, com a diferença de ser obrigatório responder 90 questões ao invés de 180.

Figura 12 – Simulador Beduka

The screenshot displays the Beduka simulation interface. At the top, the Beduka logo is visible with the tagline 'BUSCADOR DE FACULDADES'. Navigation buttons for 'João' and 'Sair' are present. The main heading is 'SIMULADO DO ENEM'. A progress bar shows 45 questions, with the first question selected. A timer indicates 00:00:54. A 'CONCLUIR SIMULADO' button is located at the bottom right. The question text is as follows:

CIÊNCIAS DA NATUREZA E SUAS TECNOLOGIAS

Questão 01
Enem, 2018

Corredores ecológicos visam mitigar os efeitos da fragmentação dos ecossistemas promovendo a ligação entre diferentes áreas, com o objetivo de proporcionar o deslocamento de animais, a dispersão de sementes e o aumento da cobertura vegetal. São instituídos com base em informações como estudos sobre o deslocamento de espécies, sua área de vida (área necessária para o suprimento de suas necessidades vitais e reprodutivas) e a distribuição de suas populações.

Disponível em: www.mma.gov.br. Acesso em: 30 nov. 2017 (adaptado).

Nessa estratégia, a recuperação da biodiversidade é efetiva porque

- A** propicia o fluxo gênico.
- B** intensifica o manejo de espécies.
- C** amplia o processo de ocupação humana.
- D** aumenta o número de indivíduos nas populações.
- E** favorece a formação de ilhas de proteção integral.

SELECIONE UMA RESPOSTA:

A B C D E

voltar pular próxima

Fonte: Retirado de Beduka (2023).

3.4 Considerações

Com base nas análises dos trabalhos correlatos, é possível concluir que existem plataformas que cumprem bem o seu papel de oferecer simulados para o ENEM ou para outros vestibulares. No entanto, ainda há espaço para melhorias.

É importante destacar que a plataforma construída neste trabalho procura oferecer diferenciais em relação aos trabalhos correlatos, como a inclusão de gamificação, possibilidade de acesso ao acervo de questões pelos desenvolvedores e também flexibilidade no que se diz respeito ao número de questões por conteúdo buscando atender às necessidades dos usuários de forma mais completa e satisfatória. Nos capítulos posteriores serão apresentadas as características e os diferenciais da plataforma desenvolvida neste trabalho.

A análise dos trabalhos correlatos permite ainda uma reflexão sobre a importância de se oferecer soluções que possam atender às demandas do público de forma eficiente e eficaz. É necessário que sejam constantemente procuradas formas de inovar e melhorar as plataformas já existentes, buscando sempre atender às expectativas dos usuários e superá-las.

4 Desenvolvimento e Resultados

O desenvolvimento deste trabalho foi realizado em sete etapas principais, contemplando várias das práticas comumente apontadas nos processos de engenharia de *software*. Foram elas:

1. Levantamento de requisitos inicial: Antes de iniciar o desenvolvimento propriamente dito, é importante realizar um levantamento de requisitos para entender quais são as necessidades do projeto e definir quais são as melhores ferramentas para utilizar na implementação. De acordo com [Sommerville \(2011\)](#), os requisitos são elementos importantes para garantir o sucesso do projeto. Nesta etapa, foram escolhidas as ferramentas necessárias para o desenvolvimento, descritas em textos anteriores e também foi decidido como tudo seria contruído.
2. Prototipação com Figma: A seguir, foi feita a prototipação de um wireframe no Figma, que serviu como base para a criação da interface de usuário, para ilustrar uma ideia de como seriam os posicionamentos de componentes e fluxos.
3. Criação do banco de questões via *script*: Para que a API que foi implementada na etapa à seguir possua dados de consumo, foi criada uma forma automática de popular o banco de dados da aplicação com questões de vestibulares de edições anteriores.
4. Implementação da API de questões de vestibulares passados: Em seguida, foi feita a implementação da API de questões de vestibulares antigos. Essa API foi criada para dar acesso questões antigas de vestibulares, que serão usadas na aplicação principal e também disponibilizada para que outros desenvolvedores possam usar.
5. Construção da API principal da aplicação: Após a criação da API de questões de vestibulares antigos, foi feita a construção da API principal da aplicação. Essa API é a base da aplicação, e é responsável por lidar com as principais funcionalidades da mesma, como o armazenamento de dados, a busca de informações e a interação com o usuário.
6. Criação da interface de usuário: Após a etapa anterior, foi realizada a criação da interface de usuário. Essa interface é a parte visual da aplicação, e é responsável por apresentar as informações para o usuário e permitir que ele interaja com o sistema. A interface foi criada com base no *wireframe* prototipado no Figma.
7. Implementação de testes: Para garantir a qualidade da implementação final, foi criada uma cobertura de testes no código, garantindo assim a funcionalidade da aplicação.

O código fonte e documentação do projeto encontram-se disponíveis na plataforma de hospedagem GitHub, onde são mantidos e atualizados.

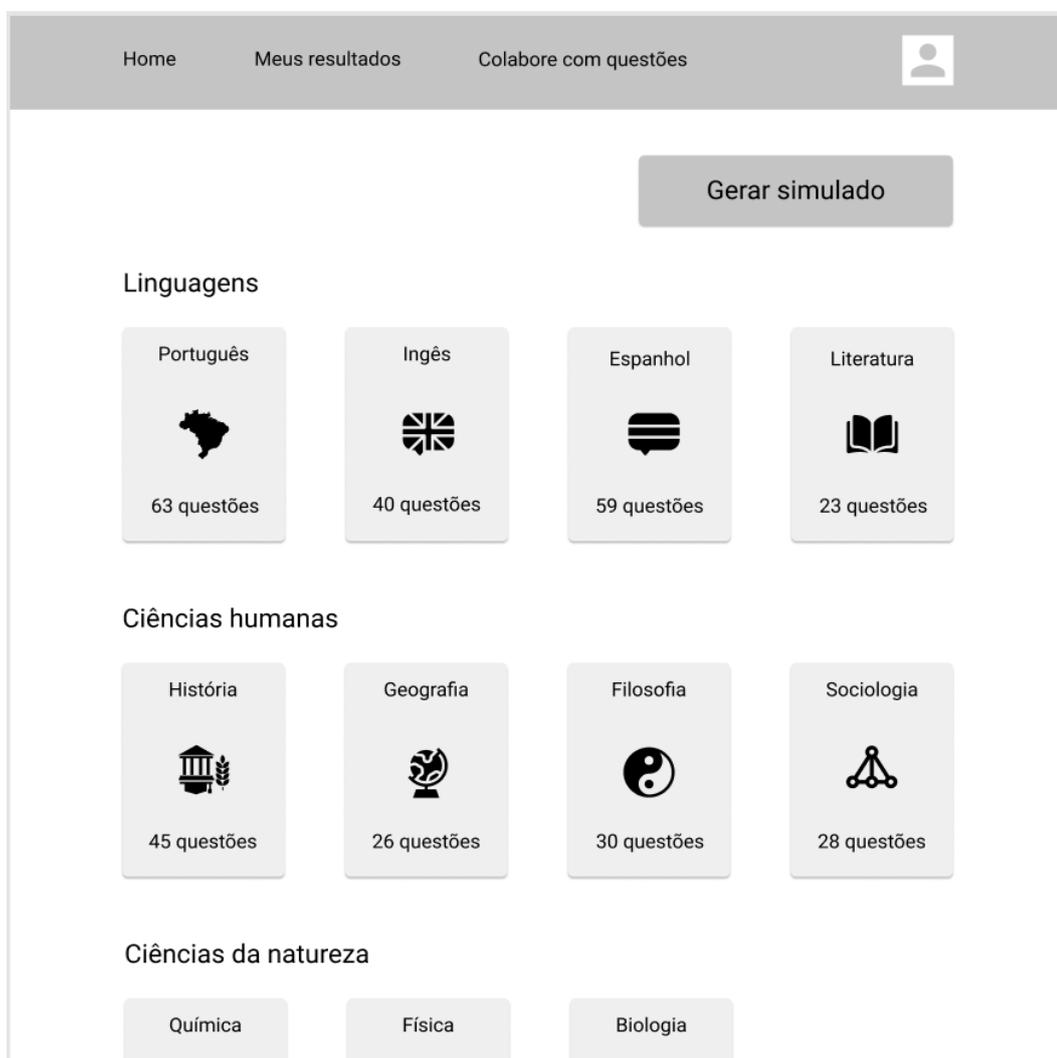
4.1 *Wireframe* da aplicação

Antes de começar qualquer implementação, foi criado um *wireframe* com algumas das principais funcionalidades da aplicação. Um *wireframe* é uma representação visual e estrutural de uma página ou interface de aplicativo. É uma representação simplificada e de baixa fidelidade e descreve como será o posicionamento de elementos, espaçamento, hierarquia, navegação e outros aspectos funcionais. Portanto, o protótipo do sistema que será representado abaixo difere visualmente do resultado final, que ainda será apresentado no trabalho.

Essa etapa de prototipagem é uma parte fundamental do processo de *design*, pois permite a experimentação e provas de conceito das funcionalidades e a estrutura de uma interface antes de investir tempo e recursos na construção de uma versão funcional. Além disso, o uso de *wireframes* pode ajudar a identificar problemas e desafios em potencial na usabilidade e interação com o usuário, permitindo que sejam resolvidos antes da fase de desenvolvimento.

A seguir serão listadas algumas das imagens do processo de prototipação do sistemas. Estas imagens foram construídas no Figma.

Figura 13 – Wireframe - Tela de inicial



Fonte: Elaborada pelo autor.

Na primeira versão da tela inicial da aplicação em questão, eram apresentadas algumas informações específicas de cada disciplina, como o seu nome, uma imagem representativa e a quantidade de questões disponíveis. Na versão definitiva da tela inicial, os conteúdos foram agrupados para condizer com a estrutura apresentada no Enem, de modo a proporcionar uma visão geral mais clara e organizada do sistema. Com essa mudança, foi possível apresentar informações mais relevantes e de forma mais clara, aumentando a usabilidade da aplicação e melhorando a experiência do usuário.

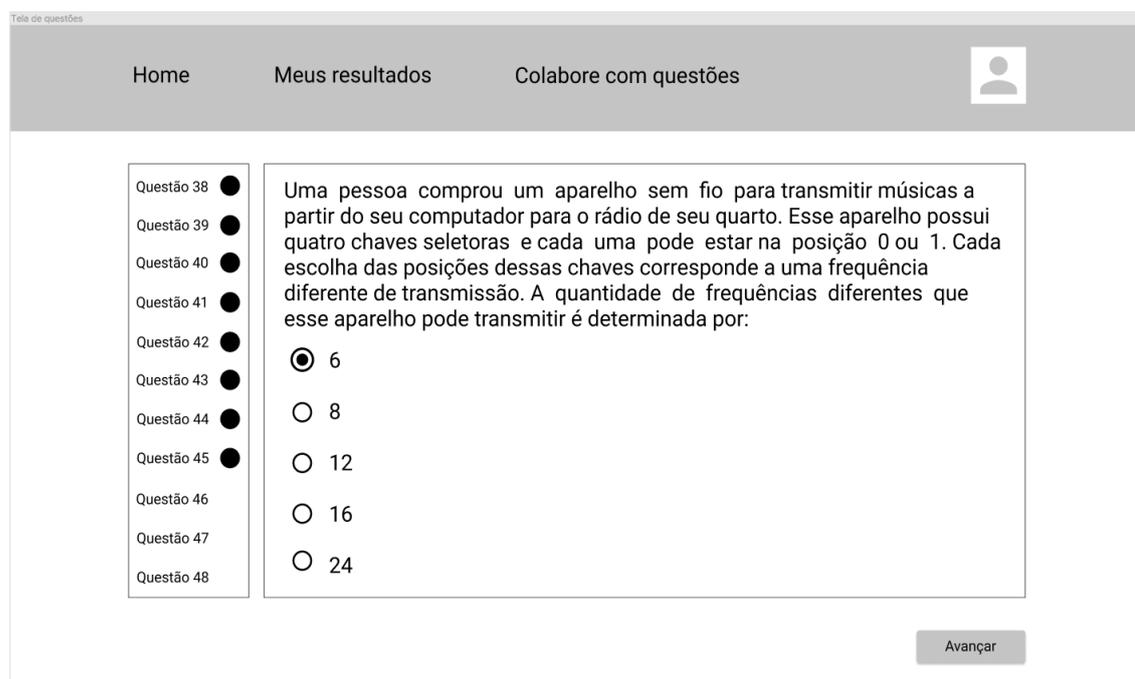
Figura 14 – Wireframe - Modal de geração de simulados



Fonte: Elaborada pelo autor.

O setor de número de questões na tela de montagem de simulados foi criado com o objetivo de oferecer uma forma intuitiva para o usuário escolher quantas questões ele deseja incluir em seu simulado. Essa funcionalidade é considerada importante, pois pode ajudar o usuário a personalizar o simulado de acordo com suas necessidades e preferências. A versão prototipada, se assemelha muito à versão final.

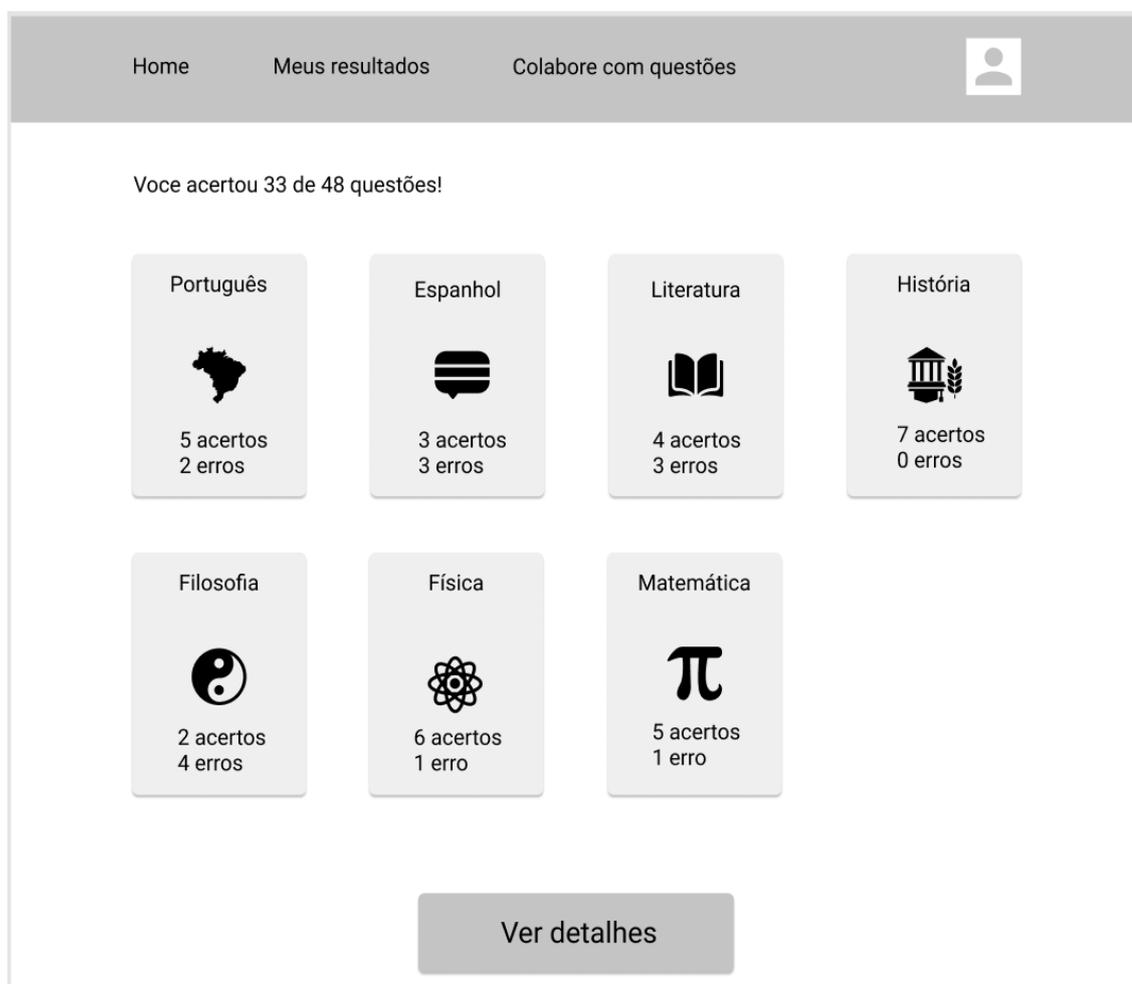
Figura 15 – Wireframe - Tela de resolução de simulados



Fonte: Elaborada pelo autor.

Na interface de resolução de questões, é apresentado um menu de navegação localizado à esquerda, que permite ao usuário percorrer as questões. A maior parte da tela é destinada à exibição do enunciado da questão e suas respectivas alternativas.

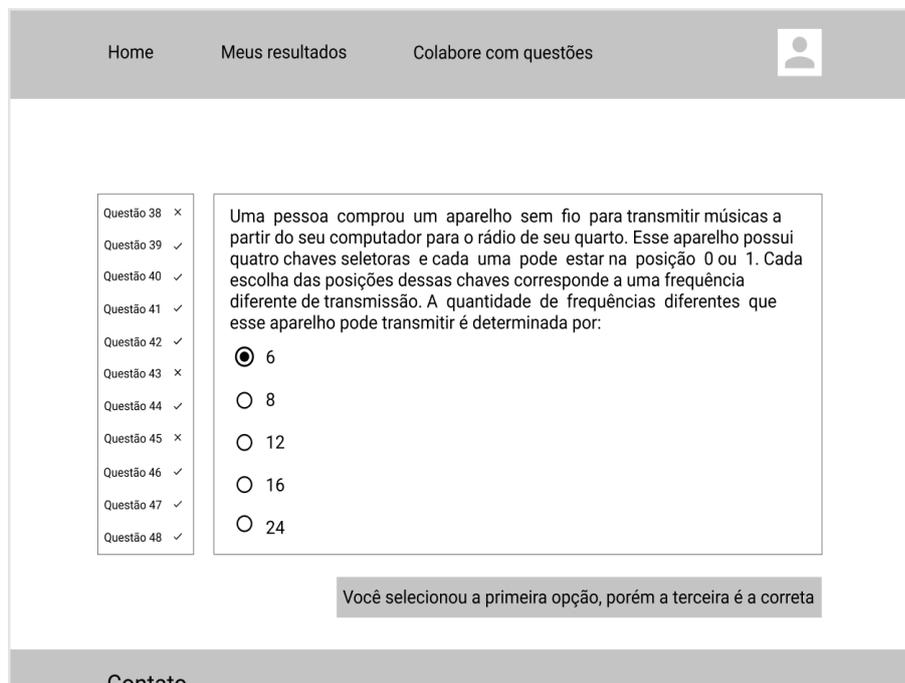
Figura 16 – Wireframe - Feedback



Fonte: Elaborada pelo autor.

Através da interface de feedback, é possível verificar o número de acertos e erros obtidos pelo usuário após a realização do simulado.

Figura 17 – Wireframe - Questão correta



Fonte: Elaborada pelo autor.

As imagens da versão final do trabalho foram geradas por meio da captura de telas da aplicação em funcionamento e serão disponibilizadas na Seção 4.5.

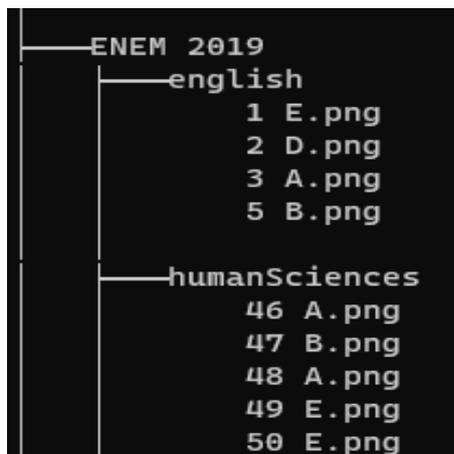
4.2 Script de carregamento de questões

O script de carregamento de questões é uma ferramenta que tem como objetivo popular, de forma eficiente, a collection de exercícios no banco de dados de uma aplicação. Antes de executar esse script, é necessário que as imagens correspondentes às questões estejam organizadas em uma estrutura de pastas bem definida. Essa estrutura de pastas é criada para facilitar o processo e garantir que as imagens estejam dispostas da forma correta.

Para ajudar nesse processo, foi desenvolvido um script auxiliar que gera as imagens referentes aos enunciados das questões de edições anteriores do ENEM, a partir do caderno azul da prova. Esse script cria as imagens e as insere na estrutura de pastas de forma automatizada, poupando o trabalho manual de organização das pastas.

A estrutura de pastas é definida de forma hierárquica, com pastas principais e subpastas. O exemplo a seguir demonstra a estrutura descrita:

Figura 18 – Estrutura de pastas para o script de carga



Fonte: Elaborada pelo autor.

As pastas principais são definidas de acordo com a matéria e o ano da prova, enquanto as subpastas são criadas para cada questão da prova. Essa estrutura permite que as imagens sejam facilmente localizadas e carregadas pelo script de carregamento de questões, tornando o processo de inclusão de novas questões no banco de dados mais rápido e eficiente.

4.3 API de questões e geração de simulados

Esta API é a responsável por prover acesso às *collections* de exercícios e de simulados com rotas para:

- Recuperar a quantidade de questões em cada área do conhecimento distinta.
- Recuperar uma questão pelo seu identificador.
- Recuperar um exercício. Pode ser realizada por meio de sua referência registrada no banco de dados da aplicação. Cada exercício possui uma referência única, que fornece informações sobre a prova de onde ele foi extraído, a edição da prova, a área do conhecimento a que pertence e o seu número de enunciado na prova original. .
- Inserir exercícios.
- Gerar um simulado com questões aleatórias com a quantidade de questões desejadas e separadas por área do conhecimento.
- Corrigir um simulado depois que o mesmo é enviado para a avaliação.

- Exibir os resultados dos simulados realizados.
- Revisar um simulado realizado anteriormente.
- Recuperar os resultados gerais de um usuário.

As rotas foram separadas em públicas e privadas com o objetivo de disponibilizar uma API de questões para usuários que desejam utilizá-las em seus próprios trabalhos. Para consultar as questões, os usuários devem informar a referência desejada, seguindo uma estrutura padrão que inclui o vestibular, a edição e o número da questão.

4.4 API principal

A API principal do projeto é responsável por gerenciar as principais funcionalidades da aplicação, tais como o login, o cadastro de usuários e a conexão com a API de geração de simulados.

O módulo de login é responsável por autenticar os usuários da aplicação. Quando um usuário entra com seu nome de usuário e senha, a API verifica se as informações estão corretas e, se estiverem, permite que o usuário acesse a aplicação.

Já o módulo de cadastro de usuários é responsável por registrar novos usuários na aplicação. Quando um usuário preenche o formulário de cadastro, a API valida as informações fornecidas e, se estiverem corretas, registra o novo usuário na base de dados.

Além disso, a API principal também faz a conexão com a API de geração de simulados, que é responsável por prover os exercícios para a aplicação. A API de geração de simulados é consultada sempre que o usuário solicita a geração de um novo simulado.

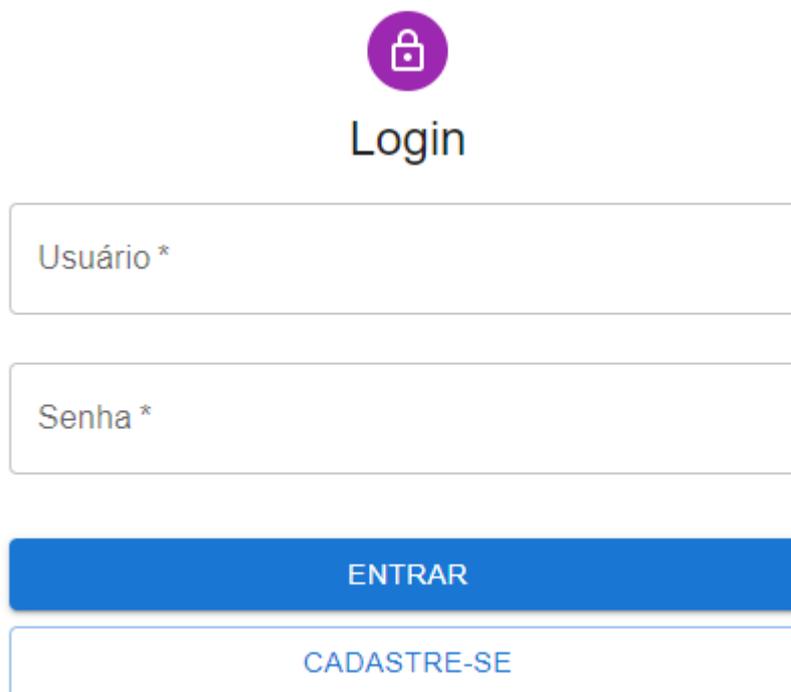
4.5 Interface de usuário e detalhamento dos requisitos

Como mencionado anteriormente, antes do início do desenvolvimento, foi criado um wireframe que demonstrava o funcionamento básico da aplicação. Nos tópicos subsequentes, as funcionalidades da versão final da aplicação serão descritas com mais detalhes, incluindo as interfaces finais em funcionamento. Além disso, os requisitos da aplicação serão apresentados de forma mais detalhada, relacionando-os a suas respectivas telas para uma melhor compreensão do funcionamento da aplicação.

4.5.1 Login do usuário

A tela de login possui uma interface simples e objetiva, composta por um formulário que possui duas entradas para usuário e senha, além de um botão para realizar o login e outro para direcionar o usuário à tela de cadastro.

Figura 19 – Tela de login da aplicação



A tela de login da aplicação apresenta o seguinte layout:

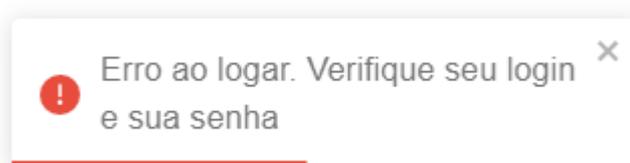
- Um ícone de cadeado dentro de um círculo roxo.
- O título "Login" em uma fonte preta.
- Dois campos de entrada de texto: "Usuário *" e "Senha *".
- Dois botões: "ENTRAR" (em um botão azul sólido) e "CADASTRE-SE" (em um botão branco com contorno azul).

Fonte: Elaborada pelo autor.

A tela de login apresenta uma interface simples e intuitiva, composta por um formulário que contém campos para inserção do nome de usuário e senha. Além disso, existem botões para realizar o acesso e direcionar o usuário para a tela de cadastro.

Em caso de sucesso na autenticação, o usuário é direcionado para a tela principal da aplicação. Por outro lado, se as credenciais estiverem incorretas, uma mensagem de erro é apresentada para informar o usuário.

Figura 20 – Erro ao fazer login



Fonte: Elaborada pelo autor.

Ao realizar o login com sucesso, é gerado um *JSON Web Token (JWT)* que contém

informações importantes para a aplicação, tais como nome do usuário, endereço de e-mail e nível de permissão na plataforma. Este *token* é armazenado no contexto interno da aplicação, gerido pelo React, e pode ser acessado por todos os componentes hierarquicamente internos a ele. A utilização do JWT é uma forma segura de armazenar dados internos, uma vez que, apesar de ser possível decodificar o seu conteúdo, não é possível alterá-lo sem uma chave de encriptação interna que é mantida em segredo.

4.5.2 Cadastro de usuário

A tela de cadastro de usuários apresenta um formulário que solicita informações necessárias para a utilização dos recursos da aplicação. O formulário contém campos para o nome, sobrenome, endereço de e-mail, senha, confirmação da senha, um botão para registro e outro para retornar à tela anterior.

Figura 21 – Tela de cadastro de usuários



A tela de cadastro de usuários apresenta o seguinte layout:

- Um ícone de perfil de usuário em um círculo roxo no topo.
- O título "Cadastre-se" em negrito.
- Dois campos de entrada para "Nome *" e "Sobrenome *".
- Um campo de entrada para "Usuário *".
- Um campo de entrada para "E-mail *".
- Um campo de entrada para "Senha *".
- Um campo de entrada para "Confirme a senha *".
- Um botão azul com o texto "REGISTRAR".
- Um botão branco com o texto "VOLTAR".

Fonte: Elaborada pelo autor.

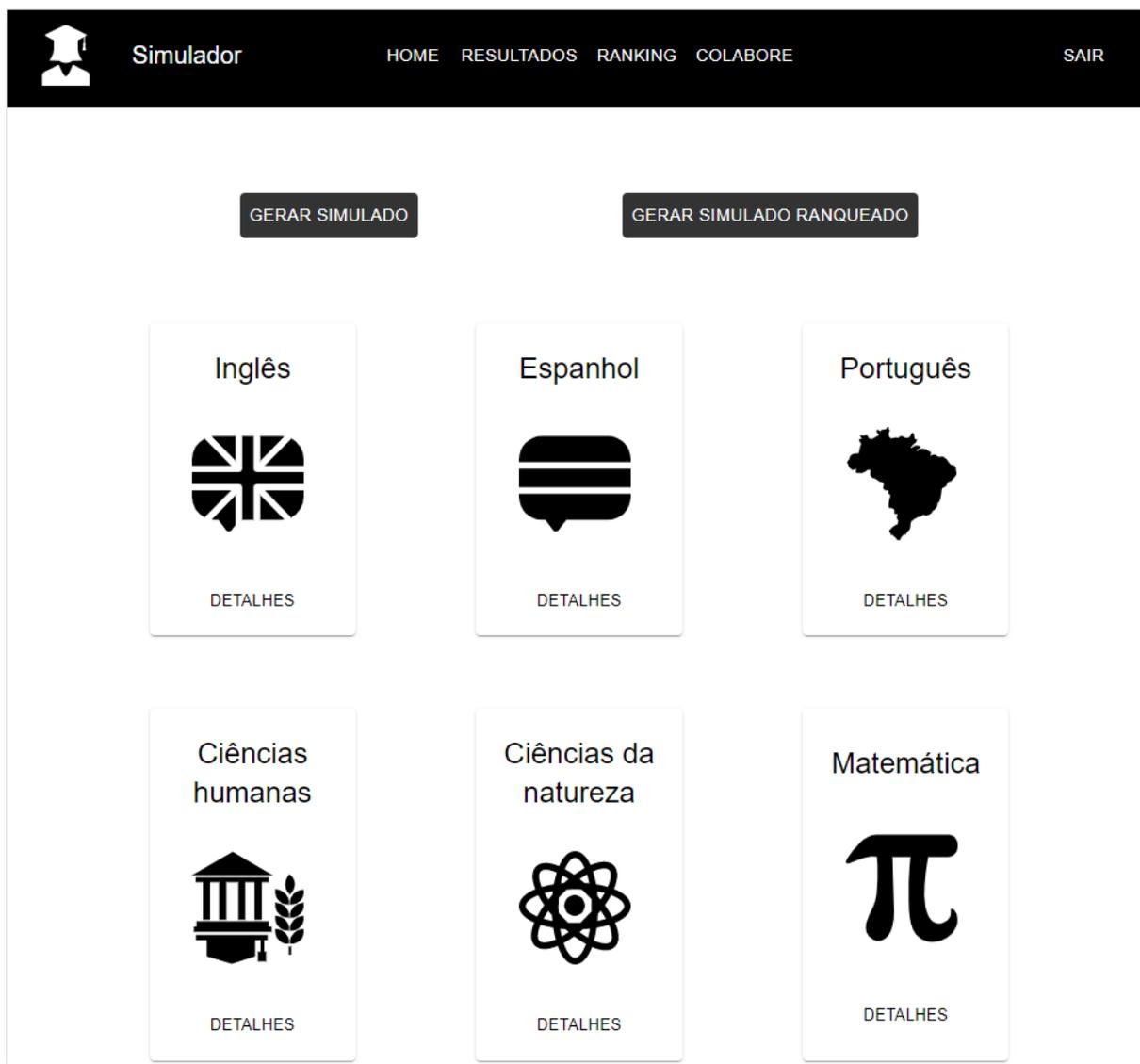
Ao finalizar o preenchimento de todos os campos necessários no formulário de cadastro, o usuário poderá completar o seu registro na aplicação. É importante destacar que os campos de senha e confirmação de senha devem ser preenchidos com as mesmas

informações e caso haja divergência entre estas informações, uma mensagem de alerta será exibida para o usuário. Para garantir a segurança dos dados, a senha inserida será criptografada antes de ser armazenada no banco de dados. Ao concluir o cadastro, o usuário receberá uma permissão padrão de usuário, limitando seu acesso apenas aos recursos disponíveis na aplicação. Finalmente, o usuário será direcionado para a tela de login.

4.5.3 Tela inicial

Na tela inicial, o usuário terá acesso às principais funcionalidades da aplicação. Essa tela apresenta a barra de navegação, que pode ser acessada em todas as outras telas e oferece acesso às páginas de resultados, *ranking*, interface colaborativa (para usuários administradores) e ao botão de logout. Além disso, na tela inicial, o usuário pode gerar simulados comuns e ranqueados. Cada cartão de disciplina inclui um botão de detalhamento e, ao clicá-lo, será aberto um modal que explicará o conteúdo da disciplina em questão.

Figura 22 – Tela inicial da aplicação



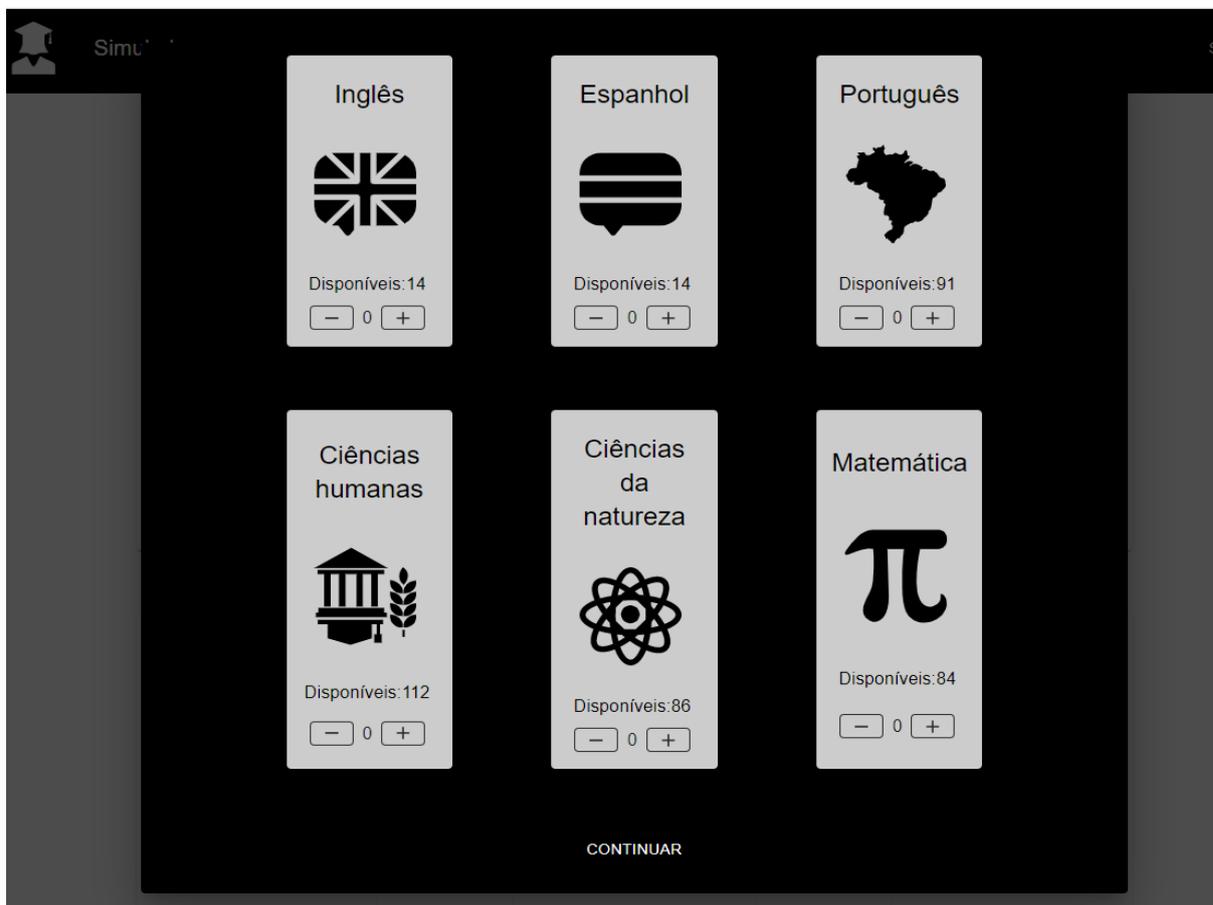
Fonte: Elaborada pelo autor.

4.5.4 Geração dos simulados

A funcionalidade de geração de simulados é apresentada através de um modal que é exibido ao usuário ao clicar em uma das opções de geração de simulados disponíveis na tela inicial da aplicação. Neste modal, o usuário é convidado a selecionar a quantidade de questões desejadas para cada área de conhecimento que deseja incluir no simulado. É importante destacar que não é possível selecionar uma quantidade de questões maior do que a quantidade disponível para cada área. Em caso de tentativa de seleção de uma quantidade superior à disponível, o sistema apresentará uma mensagem de erro. Ao clicar na opção de continuar, a API que se comunica com a *collection* de exercícios no banco de

dados selecionará questões aleatórias e diferentes correspondentes à quantidade escolhida em cada área de conhecimento.

Figura 23 – Modal de geração de simulados



Fonte: Elaborada pelo autor.

4.5.4.1 Simulados comuns

Ao clicar na opção "GERAR SIMULADO", o usuário será direcionado para a tela de resolução dos exercícios. É importante destacar que esse simulado é destinado apenas para fins de estudo e não será registrado no *ranking* de desempenho da plataforma.

4.5.4.2 Simulados ranqueados

A opção de simulados ranqueados é similar aos jogos online competitivos, já que o resultado é registrado no *ranking* geral. Nesse *ranking*, os usuários são classificados de acordo com o desempenho, levando em consideração a relação entre o número de questões corretas e o número de questões resolvidas. Aqueles que obtiveram um melhor desempenho são colocados no topo da lista, enquanto aqueles com desempenho inferior ficam nas posições inferiores.

A distinção entre os dois tipos de simulados é crucial para a implementação da aplicação, uma vez que ela atenderá ao requisito de gamificação estabelecido na análise de requisitos. Além disso, ela ajudará na retenção de usuários e evitará que aqueles que se sentem incomodados em utilizar o sistema, seja por questões pessoais, como exposição, ou simplesmente por não quererem ser comparados em termos de desempenho, possam aproveitar os serviços disponibilizados pela plataforma.

4.5.5 Resolução dos simulados

Na tela de resolução de simulados, o usuário terá acesso a questões sorteadas aleatoriamente e terá a possibilidade de as resolver. Cada questão apresentará cinco alternativas (A, B, C, D ou E), semelhantes ao formato do ENEM, e o usuário deverá escolher apenas uma delas como resposta.

Figura 24 – Tela de resolução do simulado



Fonte: Elaborada pelo autor.

Após a resolução de todas as questões um botão escrito SUBMETER QUESTIONÁRIO será apresentado na tela, para que o usuário possa clicar e finalizar seu

simulado.

Figura 25 – Finalização do simulado

The screenshot shows a web interface for a simulation. At the top, there is a navigation bar with a logo on the left, the word "Simulador", and menu items: HOME, RESULTADOS, RANKING, COLABORE, and SAIR. Below the navigation bar is a progress indicator showing a sequence of numbers from 1 to 14, with 14 highlighted. Below the progress indicator are five radio buttons labeled A, B, C, D, and E. A button labeled "SUBMETER QUESTIONÁRIO" is positioned below the radio buttons. The main content area displays a question titled "Questão 99 - Ciências da Natureza e suas Tecnologias". The question text is: "A resolução de um instrumento de medição é definida como a menor variação de uma grandeza que pode ser medida. O paquímetro é um instrumento de medição de grandezas de comprimento linear. Um fabricante de componentes para portões eletrônicos produz eixos com diâmetros diferentes e possui paquímetros com resoluções de 0,02 mm (paquímetro A) e 0,05 mm (paquímetro B). Um funcionário dessa empresa mediu um eixo com diâmetro nominal de 11,0 mm com uso do paquímetro A, e outro eixo com diâmetro nominal de 12,5 mm com o paquímetro B. Quais são as possíveis leituras obtidas com o uso dos paquímetros A e B, em milímetro, respectivamente?". Below the question text are five multiple-choice options labeled A through E: (A) 11,01 e 12,50; (B) 11,02 e 12,51; (C) 11,04 e 12,55; (D) 11,05 e 12,50; (E) 11,06 e 12,54.

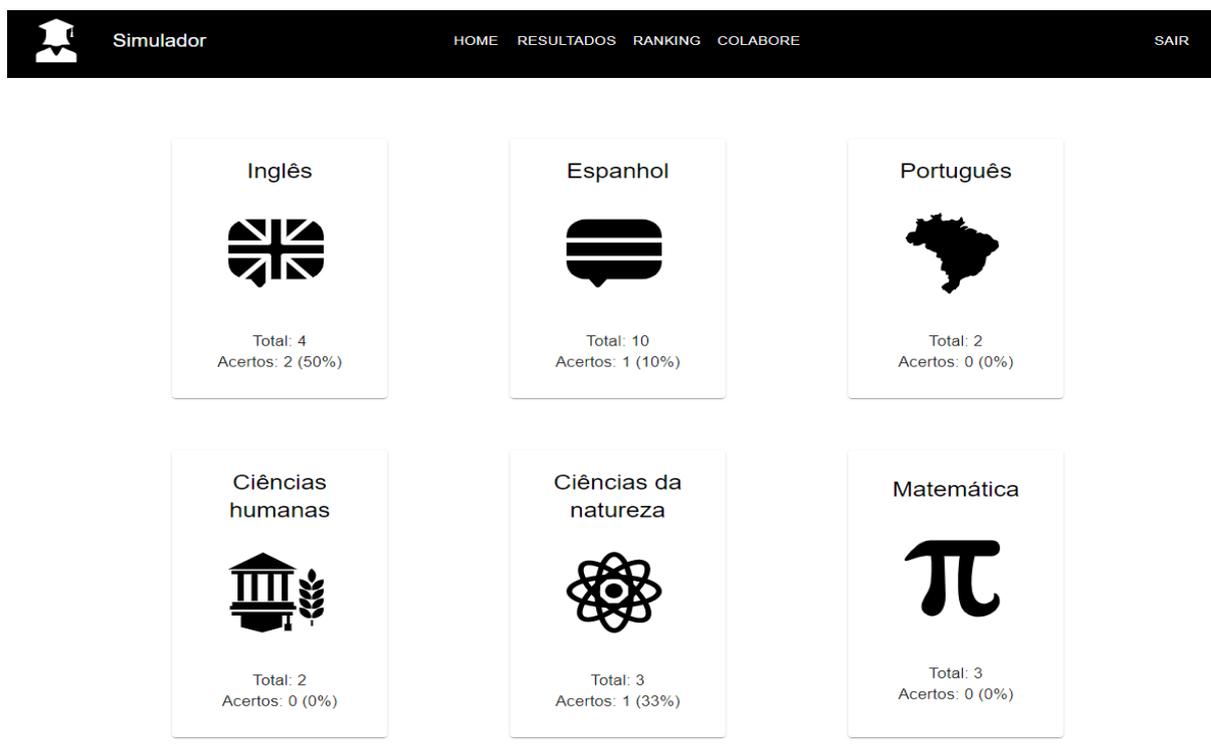
Fonte: Elaborada pelo autor.

4.5.6 Tela de resultados

A tela de resultados tem como objetivo apresentar ao usuário informações detalhadas sobre seu desempenho em simulados realizados anteriormente. Além disso, ela oferece a possibilidade de revisão. Nesta interface, são exibidos seis cartões e uma tabela para melhor visualização dos resultados.

Cada *card* apresenta a porcentagem de acerto geral do usuário em uma determinada disciplina, que é calculada com base na relação entre o número de questões corretas respondidas e o número total de questões naquela matéria. Esta porcentagem de acerto é apresentada de forma clara e objetiva na tela de resultados, possibilitando ao usuário uma avaliação detalhada de seu desempenho.

Figura 26 – Média geral dos resultados do usuário



Fonte: Elaborada pelo autor.

Figura 27 – Resultados das provas

Identificador	Data da realização	Quantidade de exercícios	Pontuação
639649d83ceb6ae5b75de41a	11/12/2022	4	1
63964a713ceb6ae5b75de48b	11/12/2022	5	1
63964a893ceb6ae5b75de4d1	11/12/2022	5	1
6397c5d66d42f2b680c6b80b	13/12/2022	2	1
6397c5ef6d42f2b680c6b84b	13/12/2022	2	0
6397cfe46d42f2b680c6b996	13/12/2022	2	0
6397d1496d42f2b680c6ba23	13/12/2022	2	0

Fonte: Elaborada pelo autor.

A tabela abaixo dos cartões de média geral do usuário apresenta o histórico de provas realizadas pelo usuário, incluindo as ranqueadas e não ranqueadas. Nesta tabela, o usuário pode verificar a sua taxa de acerto individual em cada simulado que foi gerado e submetido.

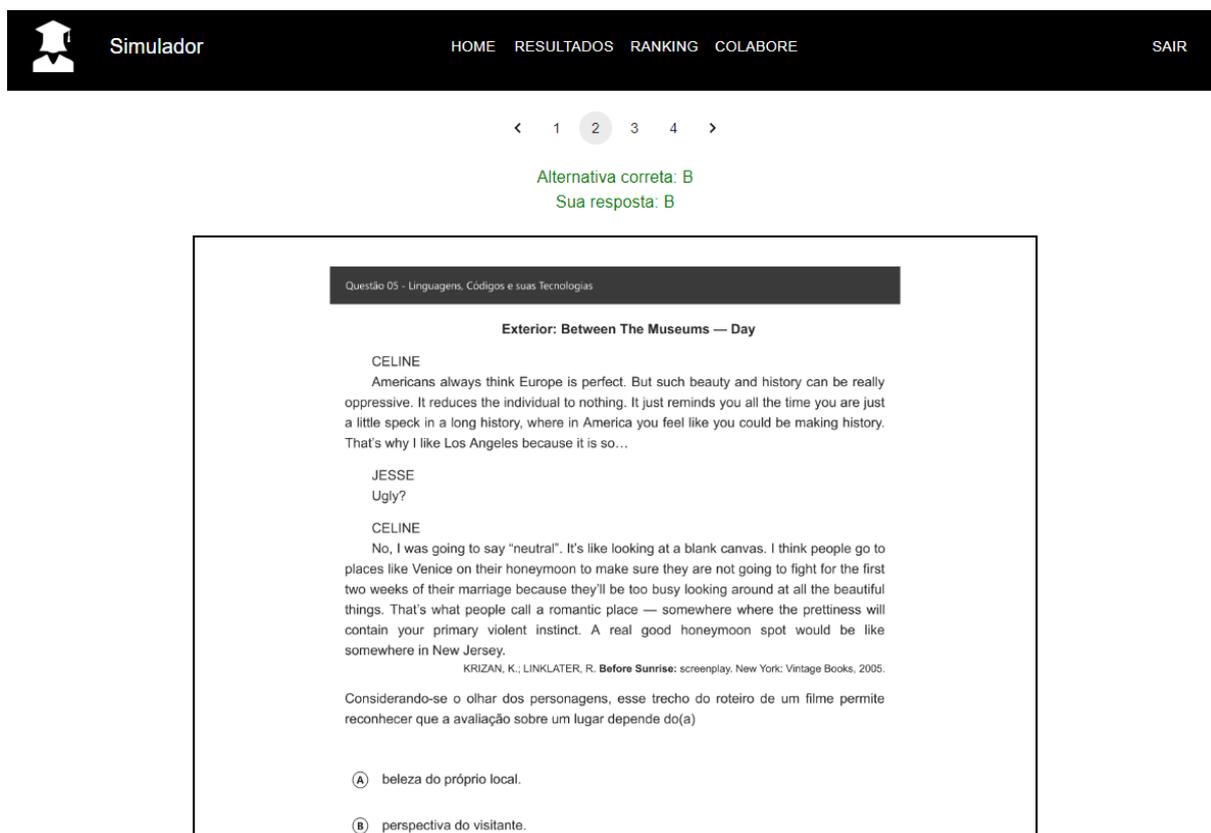
4.5.6.1 Detalhamento do simulado resolvido

A tabela, situada abaixo dos cartões de média geral do usuário, exibe o histórico de simulados realizados pelo usuário, tanto ranqueados quanto não ranqueados. Neste local, o usuário pode verificar sua taxa de acerto individual em cada simulado que foi gerado e submetido.

Ao clicar na linha correspondente ao simulado desejado, o usuário é direcionado para a tela de revisão desse simulado, na qual é possível verificar quais questões foram respondidas corretamente e quais não. Esta tela segue o mesmo padrão da tela de resolução de simulados, com uma barra de navegação entre as questões na parte superior e a visualização da questão na parte inferior.

Quando uma questão é apresentada como correta pelo usuário, é destacado um texto verde indicando a alternativa correta da questão e a alternativa selecionada pelo usuário. Quando a questão é apresentada como incorreta, é destacado um texto vermelho indicando a alternativa correta e a alternativa preenchida pelo usuário.

Figura 28 – Resultados das provas - Alternativa correta



The screenshot shows a simulation interface with a black header bar. On the left is a white icon of a graduation cap and the word "Simulador". In the center are the menu items "HOME", "RESULTADOS", "RANKING", and "COLABORE". On the right is "SAIR". Below the header is a navigation bar with arrows and numbers 1, 2, 3, 4, where 2 is highlighted. Below that, the text "Alternativa correta: B" and "Sua resposta: B" is displayed in green. The main content area is a white box with a black border. At the top of this box is a dark grey bar with the text "Questão 05 - Linguagens, Códigos e suas Tecnologias". Below this is the title "Exterior: Between The Museums — Day". The text of the question follows, including dialogue between Celine and Jesse, a reference to the film "Before Sunrise", and two multiple-choice options: (A) beleza do próprio local. and (B) perspectiva do visitante.

Questão 05 - Linguagens, Códigos e suas Tecnologias

Exterior: Between The Museums — Day

CELINE
Americans always think Europe is perfect. But such beauty and history can be really oppressive. It reduces the individual to nothing. It just reminds you all the time you are just a little speck in a long history, where in America you feel like you could be making history. That's why I like Los Angeles because it is so...

JESSE
Ugly?

CELINE
No, I was going to say "neutral". It's like looking at a blank canvas. I think people go to places like Venice on their honeymoon to make sure they are not going to fight for the first two weeks of their marriage because they'll be too busy looking around at all the beautiful things. That's what people call a romantic place — somewhere where the prettiness will contain your primary violent instinct. A real good honeymoon spot would be like somewhere in New Jersey.

KRIZAN, K.; LINKLATER, R. *Before Sunrise*: screenplay. New York: Vintage Books, 2005.

Considerando-se o olhar dos personagens, esse trecho do roteiro de um filme permite reconhecer que a avaliação sobre um lugar depende do(a)

(A) beleza do próprio local.

(B) perspectiva do visitante.

Fonte: Elaborada pelo autor.

Figura 29 – Resultados das provas - Alternativa incorreta

Simulador HOME RESULTADOS RANKING COLABORE SAIR

< 1 2 3 4 >

Alternativa correta: E
Sua resposta: A

Questão 05 - Linguagens, Códigos e suas Tecnologias

If Women Had Their Own Currency, Here's What It Would Be Worth
Charlotte Alter @charlottealter Maya Rhodan @m_rhodan July 31, 2014



After a little girl asked President Obama why there aren't any women on U.S. currency, he said that adding some female faces to our cash sounded like a "pretty good idea". Almost immediately, all of our fantasies came alive on the web. What would, let's say, Ruth Bader Ginsburg look like on a \$20 bill? Where would we spend our Beyoncé \$10 bill first? Will our grandmas give us a Susan B. Anthony \$5 bill on our birthdays and tell us not to spend it all at once?

But then we remembered: because of the wage gap, a dollar for a woman is not the same as a dollar for a man. Although the true extent of the gender pay gap is widely disputed even

Fonte: Elaborada pelo autor.

4.5.6.2 *Ranking* de usuários

A funcionalidade de *ranking* de usuários tem como objetivo classificar os usuários de acordo com seus desempenhos em simulados ranqueados. A classificação é realizada por meio de uma ordenação decrescente das taxas de acerto de cada usuário, calculadas através da relação entre o número de questões corretas e o número de questões respondidas em simulados ranqueados. Assim, os usuários com as maiores taxas de acerto estarão listados no topo do *ranking*, enquanto os de menor desempenho serão exibidos na parte inferior da tabela.

Figura 30 – *Ranking* de usuários

Posição	E-mail	Questões resolvidas	Questões corretas	Taxa de acerto (%)
1º	ana.gomes@gmail.com	18	11	61.11%
2º	alan.pinho@gmail.com	17	3	17.65%
3º	jvmr535@gmail.com	20	3	15.00%

Fonte: Elaborada pelo autor.

4.5.7 Inserção de questões feitas de maneira manual (Administrador)

Além do *script* de carga, que será apresentado posteriormente, os usuários administradores têm a possibilidade de popular o banco de dados de questões por meio da interface administrativa. Nesta interface, cada questão é inserida de forma individual, onde o administrador deve selecionar a imagem da questão, a área do conhecimento a qual ela se relaciona e a alternativa correta. Dessa forma, o usuário pode registrar a questão no banco de dados.

Figura 31 – *Ranking* de usuários

Fonte: Elaborada pelo autor.

4.6 Testes e resultados

Para avaliar a experiência dos usuários e a efetividade da plataforma desenvolvida, foi realizado um teste por meio de um formulário do *Google Forms*. O formulário continha perguntas relacionadas à experiência de uso da plataforma e suas funcionalidades.

O formulário foi respondido por dez usuários, que forneceram informações valiosas sobre a usabilidade da plataforma e suas funcionalidades. A maioria dos usuários classificou a plataforma como intuitiva e afirmou que a utilizaria durante seu período de estudos para vestibulares. Além disso, a maioria acredita que a plataforma pode ocasionar em uma melhora nos rendimentos em vestibulares e que a gamificação ajudou em sua retenção na plataforma. Abaixo, estarão os dados referentes às respostas dos usuários para cada pergunta, respectivamente:

1. Como você classificaria a usabilidade da plataforma?
 - Intuitiva: 90%
 - Regular: 10%
 - Tive algumas dúvidas durante a utilização: 0%
 - Conta-intuitiva: 0%
2. No seu período de estudos para vestibulares, usaria uma plataforma como esta?
 - Sim: 70%
 - Provavelmente sim: 30%
 - Não: 0%
 - Provavelmente não: 0%
 - Nunca prestei vestibular: 0%
3. Você acredita que plataformas como esta possam ocasionar em uma melhora nos rendimentos em vestibulares?
 - Sim: 80%
 - Provavelmente sim: 20%
 - Não: 0%
 - Provavelmente não: 0%
4. Você acredita que a gamificação (modo de simulados ranqueados) ajudou na sua retenção na plataforma?
 - Sim: 80%

- Um pouco: 20%
- Não: 0%

5. Quais outras funcionalidades você acha que agregariam à plataforma?

- Sistema de recompensas que aumenta a retenção na plataforma: 10%
- Multi funcionalidade e ambiente de colaboração: 10%
- Não respondeu: 80%

Com base nas respostas fornecidas pelos usuários, podemos concluir que a plataforma foi bem recebida pelos usuários e que suas funcionalidades são intuitivas e efetivas para o período de estudos para vestibulares. Além disso, a gamificação parece ser uma funcionalidade importante para retenção dos usuários na plataforma. As sugestões de funcionalidades adicionais serão levadas em consideração para futuras melhorias da plataforma.

4.7 Considerações

Este capítulo descreveu como foi o processo de implementação e os resultados finais do trabalho. Ele traz uma alternativa aprimorada de preparação para vestibulares, oferecendo acesso a questões de edições antigas por meio de uma API e uma interface de usuário intuitiva. A metodologia de desenvolvimento seguiu as práticas comuns na engenharia de software, incluindo levantamento de requisitos, prototipação, implementação da API, construção da interface de usuário e testes de qualidade. Além disso, a automatização do carregamento de questões por meio de um *script* garante uma gestão eficiente dos dados.

Com esta solução, os estudantes terão a oportunidade de visitar questões antigas e se preparar melhor para os vestibulares. A disponibilização da API também incentiva outros desenvolvedores a utilizarem e melhorarem a aplicação, tornando-a ainda mais útil para a comunidade.

5 Conclusão

A resolução de exercícios de vestibulares antigos é uma importante ferramenta para a preparação para as provas de vestibular. Esses exercícios oferecem uma ampla gama de questões que cobrem os conteúdos mais comuns dessas provas, permitindo que os estudantes se familiarizem com o tipo de questão que pode ser encontrada na prova. Além disso, a resolução de exercícios antigos também ajuda a desenvolver as habilidades de pensamento crítico e de resolução de problemas, que são importantes para o sucesso em vestibulares e em outras provas.

No trabalho, foi apresentada uma fundamentação teórico-metodológica sólida, que incluiu conceitos relevantes como vestibular e ingresso no ensino superior, gamificação, sistemas web, desenvolvimento web e bibliotecas, entre outros. Além disso, foram descritas as ferramentas e tecnologias utilizadas para o desenvolvimento do trabalho.

Os estudantes podem identificar os conteúdos que não dominam completamente e, assim, direcionar sua preparação para essas áreas. Isso é particularmente útil para os estudantes que estão se preparando para vestibulares com provas específicas, como a medicina ou engenharia, que cobram conhecimentos mais aprofundados em determinadas áreas.

Ademais, a realização destes simulados é importante para ajudar os estudantes a se adaptarem ao ritmo e à pressão das provas e a resolução de exercícios de vestibulares antigos é uma ferramenta valiosa para a preparação para as provas de vestibular.

O presente trabalho não somente propicia a experiência mencionada anteriormente, como também fornece uma ferramenta valiosa para outros programadores que buscam lidar com uma implementação semelhante. A disponibilização do banco de questões através de uma API, possibilita uma significativa economia de tempo na implementação, facilitando bastante o processo. Dessa forma, é possível obter uma solução mais eficiente e otimizada para o desenvolvimento de projetos similares.

5.1 Trabalhos Futuros

Algumas sugestões para trabalhos futuros incluem:

- Expandir a plataforma para outras áreas de estudos, além de vestibulares, como concursos públicos e exames de proficiência.
- Realizar estudos de longo prazo para avaliar o impacto da plataforma no desempenho dos estudantes em vestibulares e outras provas.

-
- Implementar um login social, permitindo que os usuários se registrem e façam login na plataforma de forma mais rápida e fácil, utilizando suas contas em redes sociais como Facebook, Google e Twitter.
 - Expandir o escopo do trabalho para incluir questões de outros tipos de exames, como exames de autoescola, e de outros concursos famosos, como Banco do Brasil, Caixa Econômica Federal e outros.
 - Implementação da opção de acesso anônimo na aplicação. Isto pode aumentar a flexibilidade da plataforma, tornando-a mais acessível e confortável para alguns usuários.
 - Mostrar os resultados gerais como gráficos circulares para melhorar a visualização.

Referências

- ALVES, J. *O que é vestibular? Entenda o conceito e como funciona*. 2021. Disponível em: <<https://www.educamaisbrasil.com.br/educacao/noticias/o-que-e-vestibular-entenda-o-conceito-e-como-funciona>>. Acesso em: 27 nov. 2022. Nenhuma citação no texto.
- AWS. *O que são microsserviços?* 2022. Disponível em: <<https://aws.amazon.com/pt/microservices>>. Acesso em: 27 nov. 2022. Nenhuma citação no texto.
- BEDUKA. *Simulado do ENEM*. 2023. <<https://beduka.com/simulado-enem>>. Acesso em: 17 mar. 2023. Nenhuma citação no texto.
- Brasil Escola. *Simulado*. 2023. <<https://vestibular.brasilecola.uol.com.br/enem/simulado>>. Acesso em: 17 mar. 2023. Nenhuma citação no texto.
- CAETANO Érica. *O que é o Enem?* 2021. Disponível em: <<https://vestibular.mundoeducacao.uol.com.br/enem/o-enem.htm>>. Acesso em: 27 nov. 2022. Nenhuma citação no texto.
- CARVALHO, G. P. de. Uma reflexão sobre a rede mundial de computadores. *Sociedade e Estado, Brasília*, v. 21, n. 2, p. 549–554, 2006. Nenhuma citação no texto.
- CIANCONI, R. B. Banco de dados de acesso público. *Ciência da Informação*, v. 16, n. 1, p. 53–59, 1987. Nenhuma citação no texto.
- FADEL, L. M. et al. *Gameificação na Educação*. [S.l.]: Pimenta Cultural, 2014. v. 1. Nenhuma citação no texto.
- FIGMA. *Dispositivos móveis*. 2023. <<https://www.figma.com>>. Acesso em: 17 mar. 2023. Nenhuma citação no texto.
- GASPAR, L. *Protocolo HTTP: entenda o que é e para que serve!* 2019. Disponível em: <<https://www.hostgator.com.br/blog/o-que-e-protocolo-http/>>. Acesso em: 27 nov. 2022. Nenhuma citação no texto.
- GONÇALVES, A. *O Que é Site, Que Tipos Existem e Como ter o seu Próprio*. 2022. Disponível em: <<https://www.hostinger.com.br/tutoriais/o-que-e-site>>. Acesso em: 27 nov. 2022. Nenhuma citação no texto.
- Mundo Vestibular. *Simulados*. 2023. <<https://www.mundovestibular.com.br/categoria/simulados>>. Acesso em: 17 mar. 2023. Nenhuma citação no texto.
- SANTOS, G. *Sistema Web e Site, quais as diferenças entre eles?* 2018. Disponível em: <<http://cpejr.com.br/site/diferenca-sistema-web-site/>>. Acesso em: 27 nov. 2022. Nenhuma citação no texto.
- SILVA, G. *O que é uma biblioteca?* 2020. Disponível em: <<https://coodesh.com/blog/dicionario/o-que-e-biblioteca/>>. Acesso em: 27 nov. 2022. Nenhuma citação no texto.
- SOMMERVILLE, I. *Engenharia de Software*. [S.l.]: Pearson Education do Brasil, 2011. v. 9. Nenhuma citação no texto.

Apêndices

APÊNDICE A – Formulário de experiência do usuário e feedbacks

Formulário sem título

Experiência de uso da plataforma de geração de simulados

*Obrigatório

1. Como você classificaria a usabilidade da plataforma? *

Marcar apenas uma oval.

- Intuitiva
- Regular
- Tive algumas dúvidas durante a utilização
- Conta-intuitiva

2. No seu período de estudos para vestibulares, usaria uma plataforma como esta? *

Marcar apenas uma oval.

- Sim
- Provavelmente sim
- Provavelmente não
- Não
- Nunca prestei vestibular

3. Você acredita que plataformas como esta possam ocasionar em uma melhora nos rendimentos em vestibulares? *

Marcar apenas uma oval.

- Sim
- Provavelmente sim
- Provavelmente não
- Não

4. Você acredita que a gamificação (modo de simulados ranqueados) ajudou na sua retenção na plataforma? *

Marcar apenas uma oval.

- Sim
- Um pouco
- Não

5. Quais outras funcionalidades você acha que agregariam à plataforma?

Este conteúdo não foi criado nem aprovado pelo Google.

Google Formulários