

UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO
INSTITUTO MULTIDISCIPLINAR

HUGO VIDAL ALVES REZENDE
MAXWEL BATALHA DA SILVA LOPES

**SimuLab: Sistema Web para Geração
de Simulados do ENEM**

Prof. Filipe Braidão do Carmo, D.Sc.
Orientador

Nova Iguaçu, Julho de 2025

SimuLab: Sistema Web para Geração de Simulados do ENEM

Hugo Vidal Alves Rezende
Maxwel Batalha da Silva Lopes

Projeto Final de Curso submetido ao Departamento de Ciência da Computação do Instituto Multidisciplinar da Universidade Federal Rural do Rio de Janeiro como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Apresentado por:

Hugo Vidal Alves Rezende

Maxwel Batalha da Silva Lopes

Aprovado por:

Prof. Filipe Braidão do Carmo, D.Sc.

Prof.^a Natalia Chaves Lessa , D.Sc.

Prof.^a Juliana Mendes Nascente e Silva Zamith , D.Sc.

NOVA IGUAÇU, RJ - BRASIL

Julho de 2025



DOCUMENTOS COMPROBATÓRIOS Nº 15015/2025 - CoordCGCC (12.28.01.00.00.98)

(Nº do Protocolo: NÃO PROTOCOLADO)

(Assinado digitalmente em 07/07/2025 19:29)

FILIFE BRAIDA DO CARMO
PROFESSOR DO MAGISTERIO SUPERIOR
DeptCC/IM (12.28.01.00.00.83)
Matrícula: ###295#4

(Assinado digitalmente em 09/07/2025 14:47)

JULIANA MENDES NASCENTE E SILVA ZAMITH
COORDENADOR CURS/POS-GRADUACAO - TITULAR
CoordCGCC (12.28.01.00.00.98)
Matrícula: ###731#0

(Assinado digitalmente em 08/07/2025 10:48)

NATALIA CHAVES LESSA
PROFESSOR DO MAGISTERIO SUPERIOR
DeptCC/IM (12.28.01.00.00.83)
Matrícula: ###435#4

(Assinado digitalmente em 08/07/2025 10:38)

HUGO VIDAL ALVES REZENDE
DISCENTE
Matrícula: 2021#####0

(Assinado digitalmente em 07/07/2025 22:51)

MAXWEL BATALHA DA SILVA LOPES
DISCENTE
Matrícula: 2021#####1

Visualize o documento original em <https://sipac.ufrrj.br/documentos/> informando seu número: **15015**, ano: **2025**, tipo: **DOCUMENTOS COMPROBATÓRIOS**, data de emissão: **07/07/2025** e o código de verificação: **f5de71f76a**

Agradecimentos

Hugo Vidal Alves Rezende

Primeiro, quero agradecer à minha família, que sempre acreditou em mim. Em especial, ao meu pai, Helio Alves, que nunca deixou de me incentivar a estudar. Sempre esteve ali, me apoiando de verdade, mesmo quando as coisas pareciam difíceis.

Aos meus amigos da faculdade — Luiz Filipe, Josue Bueno e Juliana Nogueira — obrigado por dividirem comigo tantas experiências, trabalhos, risadas e momentos de pressão também. Vocês fizeram a caminhada ser mais leve.

E um agradecimento especial ao Maxwell Batalha, que esteve comigo desde o primeiro dia da graduação. Compartilhamos muitas disciplinas, trabalhos e desafios ao longo do curso. Um grande parceiro nessa caminhada e um amigo que levo comigo.

Maxwel Batalha da Silva Lopes

Agradeço, antes de tudo, aos meus pais, Márcio Natividade e Micheli Batalha, com todo o meu amor e gratidão. Foram vocês que, desde cedo, me ensinaram o valor do estudo e da dedicação, mostrando que o conhecimento tem o poder de transformar vidas. Sei que muitos dos seus próprios sonhos ficaram em segundo plano para que eu pudesse realizar os meus, e, mesmo assim, vocês nunca hesitaram em me apoiar. À minha mãe, em especial, que abriu mão do sonho de cursar uma faculdade para trabalhar e garantir que eu e meus irmãos tivéssemos as oportunidades que ela não teve. Cada passo que dou carrega a força do amor e do sacrifício de vocês.

Aos meus avós, Jones Felisberto e Marily Batalha, minha eterna admiração e carinho. Trabalharam desde muito jovens e viam a universidade como um sonho distante. Ainda assim, nunca deixaram de acreditar em mim. Essa conquista também é de vocês, que, com tanto esforço e esperança, plantaram as bases que hoje me sustentam. Em especial, à minha avó, que sonhava em me ver formado desde os tempos de preparação para o vestibular. Tenho certeza de que, lá do céu, ela celebra comigo essa vitória.

Aos amigos que a vida me deu, meu mais sincero agradecimento. Obrigado por enxergarem em mim aquilo que, muitas vezes, eu mesmo não conseguia ver. Por me incentivarem e por estarem ao meu lado nas fases mais difíceis. Em especial, à Gabrielle, amiga desde os tempos do ensino fundamental, que sempre esteve presente, torcendo por mim a cada conquista e decisão.

Aos amigos que encontrei na universidade, meu profundo carinho. Juliana, Josué, Luiz e Hugo, nossa jornada foi marcada por desafios, aprendizados e, acima de tudo, por cumplicidade. Desde os primeiros dias de aula até os momentos finais, estivemos lado a lado: dividindo angústias, celebrando pequenas vitórias e sustentando uns aos outros quando tudo parecia desmoronar. Em meio a tantas incertezas, encontrei em vocês abrigo, força e amizade verdadeira. Cada conversa, cada riso, cada silêncio compartilhado tem um lugar especial nesta conquista. Levo vocês comigo, sempre.

RESUMO

SimuLab: Sistema Web para Geração de Simulados do ENEM

Hugo Vidal Alves Rezende e Maxwell Batalha da Silva Lopes

Julho/2025

Orientador: Filipe Braida do Carmo, D.Sc.

O ENEM configura-se como o principal mecanismo de ingresso ao ensino superior no Brasil, ao mesmo tempo em que evidencia profundas desigualdades regionais e socioeconômicas que impactam diretamente o desempenho dos candidatos. Diante desse cenário, este trabalho apresenta o SimuLab, uma plataforma web concebida para oferecer uma preparação mais acessível, personalizada e eficaz. O sistema possibilita a criação autônoma de simulados, com base em filtros, como área do conhecimento, disciplina, tópico e ano, e inclui a elaboração de redações avaliadas automaticamente segundo os cinco critérios estabelecidos pelo INEP. A proposta enfatiza a centralidade do estudante no processo de aprendizagem, proporcionando um ambiente que replica com fidelidade as condições da prova oficial e se adapta às necessidades individuais de cada usuário. Complementarmente, o SimuLab incorpora um painel de desempenho interativo, que apresenta a evolução do estudante por meio de visualizações gráficas detalhadas, permitindo a identificação de lacunas de conhecimento e o planejamento estratégico dos estudos. Dessa forma, o sistema busca não apenas aprimorar o desempenho de estudo dos usuários, mas também ampliar as oportunidades de acesso ao ensino superior de maneira mais justa e inclusiva.

ABSTRACT

SimuLab: Sistema Web para Geração de Simulados do ENEM

Hugo Vidal Alves Rezende and Maxwell Batalha da Silva Lopes

Julho/2025

Advisor: Filipe Braida do Carmo, D.Sc.

The ENEM serves as the primary mechanism for admission to higher education in Brazil, while simultaneously highlighting profound regional and socioeconomic inequalities that directly impact candidate performance. Given this scenario, this paper introduces SimuLab, a web platform designed to offer more accessible, personalized, and effective preparation. The system enables the autonomous creation of mock exams based on filters, such as area of knowledge, discipline, topic, and year, and includes the drafting of essays automatically evaluated according to the five criteria established by the INEP. The proposal emphasizes the student's centrality in the learning process, providing an environment that faithfully replicates the conditions of the official exam and adapts to each user's individual needs. Additionally, SimuLab incorporates an interactive performance dashboard, which displays the student's evolution through detailed graphical visualizations, allowing for the identification of knowledge gaps and strategic study planning. Thus, the system seeks not only to enhance users' study performance but also to expand opportunities for access to higher education in a fairer and more inclusive manner.

Lista de Figuras

2.1	Exemplo de Documento <i>HyperText Markup Language</i> (HTML) . . .	9
2.2	Exemplo de Documento HTML estilizado com <i>Cascading Style Sheets</i> (CSS)	9
3.1	Telas da aplicação proposta por (SILVA, 2023), com foco na personalização de simulados e análise de desempenho dos usuários.	16
3.2	Tela de feedback da plataforma Descomplica, indicando acertos, erros e áreas de conhecimento que exigem reforço.	17
3.3	Diagrama do Modelo Entidade-Relacionamento proposto.	39
4.1	Estrutura de rotas da aplicação baseada em páginas	46
4.2	Estrutura em camadas adotada no sistema	47
4.3	Tela de cadastro de usuário	50
4.4	Tela de login do sistema	50
4.5	Página inicial	51
4.6	Tela de criação de simulado	52
4.7	Tela de criação de simulado	53
4.8	Tela de criação de simulado	54
4.9	Tela de redação	55

4.10	Tela de resultado da redação	56
4.11	Tela Meus simulados	57
4.12	Painel com estatísticas gerais de desempenho do usuário	58
4.13	Gráfico de barras com desempenho por simulado	58
4.14	Gráfico com as notas obtidas nas redações	59
4.15	Gráfico radar com a afinidade por disciplina	60
4.16	Categorias com menor média de acertos	61
4.17	Recomendações de estudo com base no desempenho	61
4.18	Histórico completo de simulados realizados pelo usuário	62

Lista de Tabelas

3.1	Requisitos Funcionais	26
3.2	Regras de Negócio	27

Lista de Abreviaturas e Siglas

INEP	Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira
ENEM	Exame Nacional do Ensino Médio
SISU	Sistema de Seleção Unificada
PROUNI	Programa Universidade para Todos
FIES	Fundo de Financiamento Estudantil
UFRJ	Universidade Federal do Rio de Janeiro
TRI	Teoria de Resposta ao Item
UC	Caso de Uso
MER	Modelo de Entidade-Relacionamento
ARPA	<i>Advanced Research Projects Agency</i>
ARPANET	<i>Advanced Research Projects Agency Network</i>
IP	<i>Internet Protocol</i>
TCP	<i>Transmission Control Protocol</i>
IMP	<i>Interface Message Processors</i>
UCLA	Universidade da Califórnia em Los Angeles
SRI	<i>Stanford Research Institute</i>
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i>
WWW	<i>World Wide Web</i>
HTTP	<i>Hypertext Transfer Protocol</i>
DNS	Sistema de Nomes de Domínio

CERN	Conselho Europeu para Pesquisa Nuclear
HTML	<i>HyperText Markup Language</i>
TLD	<i>Top Level Domain</i>
URL	<i>Uniform Resource Locator</i>
CSS	<i>Cascading Style Sheets</i>
DOM	<i>Document Object Model</i>
API	<i>Application Programming Interface</i>
JSON	<i>JavaScript Object Notation</i>
LLM	<i>Large Language Model</i>
ORM	<i>Object-Relational Mapping</i>

Sumário

Agradecimentos	i
Resumo	iii
Abstract	iv
Lista de Figuras	v
Lista de Tabelas	vii
Lista de Abreviaturas e Siglas	viii
1 Introdução	1
2 Fundamentação	3
2.1 Internet	3
2.1.1 Origem	3
2.1.2 Arquitetura da Internet	5
2.1.2.1 Modelo Cliente-Servidor	5
2.1.2.2 Protocolos fundamentais	6
2.1.3 <i>World Wide Web</i>	7

2.2	Desenvolvimento Web	7
2.2.1	HTML e CSS	8
2.2.2	JavaScript	10
2.2.3	React	10
2.2.4	APIs e Comunicação com Serviços Externos	11
2.3	Modelos de Linguagem de Grande Porte	11
3	Proposta	13
3.1	Motivação	13
3.2	Trabalhos Relacionados	15
3.3	Proposta de Sistema	18
3.3.1	Funcionalidades	19
3.3.1.1	Simulados personalizados	20
3.3.1.2	Redação	22
3.3.1.3	Painel de Desempenho	24
3.3.2	Requisitos Funcionais	25
3.3.3	Regras de Negócios	27
3.3.4	Casos de Uso	29
3.3.5	Modelo de Dados	36
4	SimuLab	40
4.1	Tecnologias Utilizadas	40
4.2	Implementação do Sistema	42
4.2.1	Coleta e Processamento das Questões	42

4.2.2	Next.js	45
4.2.3	Arquitetura em Camadas	46
4.2.3.1	Camada de Apresentação	47
4.2.3.2	Camada de Lógica	48
4.2.3.3	Camada de Acesso a Dados	48
4.2.4	Sistema de Autenticação	49
4.2.5	Simulados	51
4.2.6	Tela de Desempenho	57
5	Conclusão	63
5.1	Considerações finais	63
5.2	Limitações e trabalhos futuros	64
	Referências	67

Capítulo 1

Introdução

Criado em 1998 pelo Ministério da Educação, o Exame Nacional do Ensino Médio (ENEM) surgiu com a finalidade de avaliar o desempenho dos estudantes ao término da educação básica. A partir de 2009, passou também a ser utilizado como critério de seleção para o ingresso em instituições de ensino superior públicas e privadas, consolidando-se como uma das principais formas de acesso à universidade no Brasil. Atualmente, o exame reúne anualmente milhões de inscritos e exerce papel central na definição de políticas públicas voltadas à educação. (INEP, 2025)

Sua estrutura é composta por quatro provas objetivas, totalizando cento e oitenta questões de múltipla escolha, além de uma redação dissertativo-argumentativa. As provas são aplicadas em dois dias, sendo que o primeiro inclui as áreas de Linguagens, Códigos e suas Tecnologias, Ciências Humanas e suas Tecnologias, além da redação; e o segundo dia aborda Ciências da Natureza e suas Tecnologias e Matemática e suas Tecnologias. O tempo disponível para realização das provas é de cinco horas e trinta minutos no primeiro dia e cinco horas no segundo, o que impõe aos participantes um grande desafio de gerenciamento de tempo e resistência física e mental.

O formato do ENEM, aliado à sua abordagem interdisciplinar e contextualizada, torna o exame um desafio complexo que vai muito além da simples memorização de conteúdos. Os participantes são continuamente exigidos a interpretar textos extensos e variados, analisar gráficos, tabelas e situações-problema, além de articular

conhecimentos de diferentes áreas para resolver questões que simulam contextos do cotidiano. Soma-se a isso a necessidade de produzir uma redação dissertativo-argumentativa que atenda aos critérios rigorosos estabelecidos pelo Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP), exigindo clareza, coerência e estruturação adequada da argumentação. Outro aspecto que intensifica a complexidade da prova é o gerenciamento do tempo: o número elevado de questões, combinado com prazos restritos em cada dia de aplicação, obriga o candidato a adotar uma estratégia eficiente de resolução, priorizando o equilíbrio entre velocidade e qualidade nas respostas.

Apesar da relevância do exame, muitos estudantes enfrentam dificuldades durante o processo de preparação, especialmente aqueles em situação de vulnerabilidade socioeconômica. Entre os principais desafios estão a limitação no acesso a materiais didáticos atualizados, a ausência de orientação pedagógica sistemática e a dificuldade de vivenciar, de forma prática, o ritmo e as condições reais da prova oficial. Essa combinação de fatores pode resultar em ansiedade, insegurança e baixa eficiência nos estudos.

Diante desse cenário, este trabalho apresenta o desenvolvimento do SimuLab, uma plataforma web de simulados adaptativos direcionada à preparação para o ENEM. A solução busca oferecer recursos que simulem de maneira fiel a experiência do exame, além de ferramentas que auxiliem na personalização dos estudos e no acompanhamento do desempenho ao longo do tempo.

Este trabalho está estruturado em cinco capítulos, que visam apresentar o desenvolvimento do projeto. O Capítulo 2, intitulado Fundamentação, apresenta os conceitos teóricos e técnicos que sustentam o projeto. O Capítulo 3 detalha a proposta do sistema, discutindo a motivação do projeto e os trabalhos relacionados existentes. O Capítulo 4 aborda as ferramentas utilizadas no desenvolvimento e descreve o processo de implementação prática da aplicação. Por fim, o Capítulo 5 apresenta as Conclusões, com as considerações finais sobre o projeto, suas limitações e as possibilidades para trabalhos futuros.

Capítulo 2

Fundamentação

2.1 Internet

Nesta seção, abordamos as origens históricas e os fundamentos técnicos que possibilitaram a criação da Internet e sustentam as aplicações modernas. Começamos definindo o que é a Internet e acompanhamos a evolução de computadores isolados para um sistema global de redes interconectadas.

A Internet pode ser definida como uma vasta rede de redes — uma infraestrutura global que conecta dispositivos diversos, como *smartphones*, servidores, computadores pessoais e sensores, permitindo que todos se comuniquem de forma padronizada. Essa comunicação é viabilizada por um conjunto de protocolos que garante o envio eficiente e confiável das informações entre os pontos conectados. (CERF; KAHN, 1974)

2.1.1 Origem

Nas décadas de 1950 e 1960, as instituições utilizavam *mainframes* — computadores de grande porte e alto custo — acessados por meio de terminais locais. A troca de dados entre sistemas ocorria principalmente por fitas magnéticas e conexões telefônicas comutadas, métodos lentos, pouco confiáveis e de baixa escalabilidade (TANENBAUM; WETHERALL, 2011).

Em 1960, Licklider (1960) apresentou o conceito de uma “Rede Galáctica”, isto é, uma rede global que permitiria o acesso remoto a programas e dados a partir de qualquer lugar. Essa visão foi aprofundada em memorandos internos da *Advanced Research Projects Agency* (ARPA) em 1963, que delinearam as bases para interconectar diferentes centros de pesquisa (LICKLIDER, 1963).

Paralelamente, surgiram avanços teóricos que fundamentariam essa rede global. Em 1961, Leonard Kleinrock aplicou a teoria das filas à transmissão de dados em unidades discretas — os pacotes — e demonstrou que a comutação de pacotes superava em eficiência o modelo de circuitos comutados, no qual o emissor e o receptor mantêm uma conexão contínua (KLEINROCK, 1961). Essa abordagem tornava a comunicação não só mais rápida, mas também mais resiliente a falhas.

Já em 1965, Lawrence Roberts e Thomas Marill realizaram o primeiro experimento de comunicação remota entre dois computadores por meio de linhas telefônicas comutadas. Embora ainda dependesse de circuitos tradicionais, esse teste comprovou a viabilidade prática de interligar máquinas em localidades distintas, apontando para arquiteturas de rede mais sofisticadas (ROBERTS, 1966).

Em 1967, Roberts apresentou à ARPA um projeto detalhado para a criação de uma rede baseada na comutação de pacotes, com o objetivo de conectar diferentes instituições de pesquisa. Essa proposta resultaria na concepção da *Advanced Research Projects Agency Network* (ARPANET), cuja arquitetura previa o envio de dados em pequenos pacotes roteados dinamicamente pela rede, ao invés de exigir uma conexão dedicada ponto a ponto (ABBATE, 2000).

A ARPANET entrou em operação em outubro de 1969, com a instalação dos primeiros *Interface Message Processors* (IMP) na Universidade da Califórnia em Los Angeles (UCLA) e no *Stanford Research Institute* (SRI). Esses dispositivos, responsáveis por receber e encaminhar pacotes de dados, atuaram como os primeiros roteadores da rede. Nos meses seguintes, novos nós foram adicionados, e a ARPANET rapidamente se consolidou como o laboratório vivo para o desenvolvimento de protocolos de comunicação (SOCIETY, 2017).

Reconhecida como o embrião da Internet, a ARPANET foi o primeiro sistema funcional de comutação de pacotes, estabelecendo tanto os fundamentos técnicos quanto os princípios conceituais das redes distribuídas modernas. Embora seu uso inicial estivesse restrito a finalidades acadêmicas e militares, sua arquitetura descentralizada e flexível mostrou-se essencial para o crescimento posterior da rede em escala global (ABBATE, 2000).

O verdadeiro marco fundacional da Internet como a conhecemos ocorreu em 1^o de janeiro de 1983, quando a ARPANET adotou oficialmente o conjunto de protocolos TCP/IP (CERF; KAHN, 1974). A adoção desses protocolos padronizou a comunicação entre redes heterogêneas, permitindo a interconexão universal de sistemas e inaugurando uma nova era para as comunicações digitais.

2.1.2 Arquitetura da Internet

A arquitetura da Internet fundamenta-se primordialmente no paradigma de comunicação cliente-servidor, no qual um cliente — geralmente um navegador ou aplicação móvel — emite requisições a um servidor, que por sua vez disponibiliza recursos e serviços sobre a infraestrutura da rede. Esse modelo estabelece uma clara divisão de responsabilidades, conferindo ao servidor a função de hospedar e gerenciar recursos, ao passo que ao cliente compete a iniciação dos fluxos de comunicação. (KUROSE; ROSS, 2017)

2.1.2.1 Modelo Cliente-Servidor

No modelo cliente-servidor, o servidor permanece ativo aguardando conexões em portas bem definidas. Quando um cliente deseja um recurso, como uma página, ele estabelece uma conexão e envia uma requisição. O servidor processa essa requisição, acessa seus recursos internos, como bancos de dados, arquivos e lógica de aplicação, e devolve uma resposta ao cliente.

Essa abordagem promove uma separação clara entre as funções de fornecimento e consumo de serviços, favorecendo a escalabilidade e a centralização do controle. No

entanto, também implica desafios, como a necessidade de manter alta disponibilidade nos servidores e gerenciar simultaneamente múltiplas conexões.

2.1.2.2 Protocolos fundamentais

A pilha de protocolos *Transmission Control Protocol/Internet Protocol* (TCP/IP), segundo Kurose e Ross (2017), adota uma organização em camadas que encapsulam funções específicas e promovem a interoperabilidade entre clientes e servidores em ambientes heterogêneos. Essa proposta remonta ao trabalho seminal de Cerf e Kahn (1974), no qual o TCP/IP foi concebido sobre três pilares:

- **Camada de Rede (IP):** Cada dispositivo na Internet possui um identificador único, o endereço *Internet Protocol* (IP). No padrão IPv4, esse identificador consiste em quatro octetos decimais (*e.g.*, 192.168.0.1) que orientam o roteamento de pacotes por entre redes distintas (POSTEL, 1981).
- **Camada de Transporte (TCP):** Assentado sobre o IP, o *Transmission Control Protocol* (TCP) provê um serviço de entrega confiável fim a fim, empregando numeração sequencial de segmentos, confirmações e retransmissões automáticas quando necessário.
- **Camada de Aplicação:** Esta camada engloba protocolos que fornecem diretamente serviços ao usuário final:
 - Sistema de Nomes de Domínio (DNS): traduz nomes simbólicos (*e.g.*, exemplo.com) em endereços IP por meio de uma hierarquia de servidores raiz, *Top Level Domain* (TLD) e autoritativos (MOCKAPETRIS, 1987).
 - *Hypertext Transfer Protocol* (HTTP): segue o modelo requisição–resposta para transferência de recursos, com métodos como GET, POST e DELETE e códigos de status (200, 404, 500 etc.) que indicam o resultado das operações.

2.1.3 *World Wide Web*

Na virada da década de 1990, embora a infraestrutura da Internet já se apoiasse em protocolos consolidados como o TCP/IP, ainda faltava uma interface que tornasse a navegação e o compartilhamento de informações verdadeiramente acessíveis aos usuários.

Foi nesse contexto que, Berners-Lee (1991), no Conselho Europeu para Pesquisa Nuclear (CERN), propôs a criação da *World Wide Web* (WWW), uma camada de aplicação baseada em hipertexto, pensada para interligar documentos por meio de uma interface simples e interativa, comparável a um “livro digital”.

Para concretizar essa visão, foram definidos três padrões fundamentais que operam sobre o TCP/IP:

- **HTML:** linguagem de marcação que estrutura texto, imagens e *links* em um documento, permitindo a criação de páginas ricas e navegáveis;
- **HTTP:** protocolo de aplicação utilizado para a transferência de documentos entre cliente e servidor.
- ***Uniform Resource Locator* (URL):** esquema de endereçamento que atribui a cada recurso na Web um identificador único, *e.g.*, `http://exemplo.com/pagina.html`. Cada URL é composta por três partes principais: o protocolo, *e.g.*, `http://`, o nome do *host* ou domínio, *e.g.*, `exemplo.com`, e o caminho para o recurso, *e.g.*, `/pagina.html`.

Quando o usuário clica em um âncora, o navegador solicita o recurso via HTTP, recupera o documento e o renderiza, fechando o ciclo que integra DNS, TCP/IP e padrões de aplicação (KUROSE; ROSS, 2017).

2.2 Desenvolvimento Web

Com base nos padrões fundamentais da WWW a Web evoluiu de um sistema dedicado à exibição de documentos estáticos para uma plataforma dinâmica e

interativa. Segundo Berners-Lee (1991), a Web foi inicialmente concebida como um espaço de informação, mas revelou seu verdadeiro potencial ao se transformar em um ambiente para aplicações.

Essa evolução tecnológica possibilitou a transição de páginas estáticas para ambientes cada vez mais dinâmicos e interativos, permitindo o surgimento de interfaces cada vez mais sofisticadas. Nesta seção, apresentaremos os conceitos básicos para o funcionamento de uma página web.

2.2.1 HTML e CSS

O HTML é a linguagem fundamental para a estruturação de conteúdo na web, sendo especificada pelo W3C, principal órgão responsável pela padronização da Web (CONSORTIUM, 2014).

Sua sintaxe é baseada em elementos chamados *tags*, que são instruções delimitadas por colchetes angulares, *e.g.*, `<p>` paragrafo `</p>`, e servem para indicar a função de cada parte do conteúdo. As *tags* geralmente vêm em pares, uma de abertura e outra de fechamento, e podem conter atributos que adicionam informações adicionais sobre aquele elemento.

Por exemplo, a *tag* `<h1>` define um título de nível um, enquanto a `<p>` define um parágrafo. Já a *tag* `` é usada para inserir imagens e possui atributos como `src`, que especifica o caminho da imagem, e `alt`, que fornece um texto alternativo para a imagem.

As *tags* HTML podem ser agrupadas em categorias, tais como:

- **Estruturais:** `<html>`, `<head>`, `<body>`
- **De texto:** `<h1>`-`<h6>`, `<p>`, ``
- **De mídia:** ``, `<video>`
- **Navegação:** `<a>`, `<nav>`
- **Formulários:** `<form>`, `<input>`, `<button>`

A Figura 2.1 mostra um exemplo de documento HTML simples e sua renderização

em um navegador, evidenciando como as *tags* definem a estrutura e a apresentação inicial do conteúdo.



Figura 2.1: Exemplo de Documento HTML

O CSS, por sua vez, complementa o HTML ao definir a aparência e o estilo dos elementos estruturados. Com o CSS, é possível controlar diversos aspectos visuais, como cores, tamanhos, margens, espaçamentos, posicionamentos e fontes. Ele pode ser aplicado de três formas: diretamente embutido em uma *tag* HTML, dentro de uma *tag* `<style>` no cabeçalho da página, ou por meio de arquivos externos referenciados pelo HTML. Na Figura 2.2, é possível observar a estilização do documento utilizando o CSS diretamente embutido em uma *tag* HTML.



Figura 2.2: Exemplo de Documento HTML estilizado com CSS

2.2.2 JavaScript

O *JavaScript*, introduzido em 1995, consolidou-se como a linguagem de *script* padrão para o desenvolvimento de aplicações Web, compondo, ao lado do HTML e do CSS, a base tecnológica da Web moderna (FLANAGAN, 2020). Enquanto o HTML estrutura semanticamente os documentos e o CSS define sua aparência visual, o *JavaScript* adiciona interatividade e comportamentos dinâmicos, permitindo a criação de aplicações reativas e responsivas.

Essa capacidade de manipulação dinâmica é viabilizada pelo *Document Object Model* (DOM), uma interface orientada a objetos que representa a estrutura hierárquica do documento HTML como um conjunto de nós acessíveis via programação. Por meio da *tag* `<script>`, o *JavaScript* integra-se diretamente ao documento, possibilitando a leitura, modificação e remoção de elementos, bem como a alteração de estilos definidos em CSS, em tempo de execução.

2.2.3 React

O React é uma biblioteca *JavaScript* de código aberto desenvolvida pelo Facebook para a construção de interfaces de usuário, especialmente para aplicações Web de página única. Sua principal proposta é facilitar a criação de interfaces reativas, ou seja, que atualizam dinamicamente os elementos da página conforme os dados mudam, sem a necessidade de recarregamento completo do documento HTML. (Meta Platforms, Inc., 2025)

Um dos conceitos centrais do *React* é o uso de componentes, que são blocos reutilizáveis e independentes de código responsáveis por representar partes da interface. Cada componente pode possuir seu próprio estado interno e propriedades recebidas de componentes pai (*props*), permitindo uma estrutura modular e de fácil manutenção. Além disso, quando ocorrem alterações no estado da aplicação, o *React* compara o novo Virtual DOM com o anterior e realiza apenas as atualizações mínimas necessárias no DOM real, tornando o processo mais eficiente.

2.2.4 APIs e Comunicação com Serviços Externos

Um *Application Programming Interface* (API) é um conjunto de regras que permite a comunicação entre diferentes sistemas, abstraindo a complexidade interna e expondo apenas funcionalidades específicas por meio de chamadas bem definidas. No contexto do desenvolvimento Web, as APIs são frequentemente disponibilizadas por serviços remotos que operam sob o protocolo HTTP, permitindo que aplicações cliente realizem requisições e obtenham respostas em formatos estruturados, como *JavaScript Object Notation* (JSON)¹.

No ecossistema Web moderno, as APIs tornaram-se indispensáveis para integrar funcionalidades externas a aplicações sem a necessidade de desenvolver soluções do zero. Por exemplo, uma loja virtual pode consumir a API de um serviço de pagamentos para processar transações, ou utilizar a API do Google Maps para exibir localizações. Essa modularidade não apenas acelera o desenvolvimento, mas também garante atualizações automáticas nos serviços consumidos, desde que a interface da API (seus *endpoints* e formatos de dados) seja mantida estável.

2.3 Modelos de Linguagem de Grande Porte

Segundo Chomsky (1965), a linguagem é uma instância da capacidade humana inata de adquirir e usar um sistema de comunicação simbólica, estruturado por regras gramaticais internalizadas, que nos permite expressar pensamentos, ideias e emoções. Para o autor, essa capacidade é exclusiva da espécie humana e não resulta apenas da experiência ou do ambiente, mas de uma estrutura mental específica, a qual denomina de faculdade da linguagem.

Inspirados por essa habilidade humana, Modelos de Linguagem de Grande Porte ou *Large Language Model* (LLM) são sistemas computacionais desenvolvidos para simular, em alguma medida, o uso da linguagem natural. Esses modelos são treinados com grandes volumes de texto disponíveis publicamente — como livros, artigos e

¹ *JavaScript Object Notation* é um formato leve de troca de dados baseado em texto, estruturado em pares chave-valor e amplamente utilizado em APIs por sua legibilidade e facilidade de análise.

páginas da Internet — e, com base nesses dados, aprendem a reconhecer padrões e regularidades na forma como as palavras e frases são utilizadas (ZHAO; AL., 2023).

O princípio central de funcionamento dos LLMs é a previsão de palavras: dado um trecho de texto, o modelo tenta adivinhar qual palavra ou elemento textual vem a seguir. Para isso, ele ajusta internamente bilhões de parâmetros durante um processo de aprendizado automatizado, conhecido como pré-treinamento. Esse processo não exige intervenção humana direta, pois o próprio modelo aprende a partir das relações entre palavras e frases nos textos que analisa.

Na prática, o texto é dividido em unidades chamadas *tokens*, que podem corresponder a palavras inteiras, partes de palavras ou até mesmo caracteres isolados, dependendo da estratégia de tokenização adotada. Cada *token* recebe uma representação numérica que permite ao modelo avaliar probabilisticamente suas relações contextuais e prever o próximo *token* na sequência (ZHAO; AL., 2023).

Ao final dessa etapa de pré-treinamento, o LLM desenvolve uma representação estatística da linguagem, o que lhe permite realizar uma variedade de tarefas com relativa competência, mesmo sem ter sido treinado especificamente para cada uma delas.

Embora não compreendam linguagem da mesma forma que os humanos, os LLMs conseguem produzir textos coerentes e contextualizados, responder perguntas, traduzir entre idiomas, resumir conteúdos e até simular conversas. Para tarefas mais específicas ou para melhorar sua confiabilidade, os modelos podem passar por ajustes adicionais, como o *fine-tuning* — um pequeno retreinamento com exemplos selecionados — ou o uso de comandos textuais bem elaborados (*prompting*), que orientam a geração de respostas.

Em resumo, os LLMs representam uma forma avançada de processamento da linguagem natural, baseada em aprendizado de máquina em larga escala. Seu desenvolvimento reflete um esforço de aproximar a capacidade computacional do modo como os humanos utilizam a linguagem, ainda que por meios fundamentalmente distintos (ZHAO; AL., 2023).

Capítulo 3

Proposta

Este capítulo apresenta a proposta de desenvolvimento de uma plataforma digital voltada à preparação para o ENEM. A plataforma foi idealizada com o intuito de fornecer ferramentas que auxiliem os estudantes na organização dos estudos, na identificação de dificuldades específicas e na adaptação ao formato da prova. Além disso, este capítulo contextualiza a relevância do ENEM no cenário educacional brasileiro e os desafios enfrentados pelos candidatos.

3.1 Motivação

Inicialmente, o ENEM tinha o propósito de avaliar o desempenho dos estudantes ao término da educação básica. No entanto, em 2009, o exame passou por uma reformulação significativa, consolidando-se como o principal critério de acesso ao ensino superior no Brasil. Essa transformação representou um marco na educação nacional, unificando os processos seletivos de diversas instituições públicas e privadas e substituindo os vestibulares tradicionais, que variavam em formato e conteúdo. (INEP, 2025)

Antes dessa reformulação, os candidatos precisavam se preparar para múltiplos vestibulares, cada um com exigências e formatos distintos. Esse cenário impunha desafios como altos custos com inscrições, materiais de estudo e deslocamentos, além

da necessidade de adaptação a diferentes estilos de prova. Para estudantes de baixa renda ou oriundos da rede pública, essas dificuldades eram ainda mais acentuadas, reduzindo suas oportunidades de acesso ao ensino superior (SOBRINHO, 2004).

A implementação do Sistema de Seleção Unificada (SISU) em 2010 consolidou o ENEM como a principal porta de entrada para universidades públicas em todo o país. Além disso, o exame tornou-se um requisito essencial para programas como o Programa Universidade para Todos (PROUNI) e o Fundo de Financiamento Estudantil (FIES), ampliando as possibilidades de inclusão e contribuindo para a democratização do ensino superior. (INEP, 2025)

Entretanto, embora o ENEM tenha unificado os processos seletivos, as condições de preparação dos candidatos continuam profundamente desiguais. Conforme evidenciado por Andrade e Tavares (2021), estudantes de baixa renda apresentam resultados sistematicamente inferiores no exame, revelando uma correlação direta entre nível socioeconômico e desempenho.

O acesso a recursos educacionais digitais influencia diretamente o desempenho dos estudantes. Pesquisas da Universidade Federal do Rio de Janeiro (UFRJ) mostram que as desigualdades no exame aumentaram entre 2013 e 2021, especialmente durante a pandemia (BARTHOLO et al., 2023). Embora o modelo unificado tenha superado a fragmentação dos vestibulares, ele ainda reproduz desigualdades socioeconômicas que limitam oportunidades educacionais.

Nesse cenário de desigualdades persistentes, a tecnologia educacional desponta como uma aliada promissora na tentativa de mitigar essas disparidades. No entanto, muitas das plataformas digitais atualmente disponíveis ainda carecem de recursos suficientemente adaptativos, acessíveis e personalizados. Frequentemente, falta-lhes *feedback* eficaz e uma usabilidade compatível com o perfil dos estudantes que mais necessitam de suporte.

Entre os recursos com potencial transformador, destaca-se o uso de simulados, cuja eficácia na aprendizagem é amplamente respaldada por estudos em psicologia cognitiva (ROEDIGER; BUTLER, 2011). A prática por meio de simulados pode

consolidar o conhecimento, reduzir a ansiedade e aumentar a autoconfiança dos alunos — fatores especialmente relevantes para estudantes em contextos de vulnerabilidade educacional.

Diante desse panorama, este trabalho propõe o desenvolvimento de uma ferramenta digital voltada à criação de simulados personalizados para o ENEM, com geração automática de provas, *feedback* imediato e acompanhamento de desempenho. A proposta busca unir tecnologia, personalização e acessibilidade para oferecer uma experiência de estudo mais eficaz e inclusiva.

Ao aliar esses elementos, espera-se contribuir de forma significativa para o fortalecimento da preparação dos estudantes, especialmente daqueles em situação de maior vulnerabilidade.

3.2 Trabalhos Relacionados

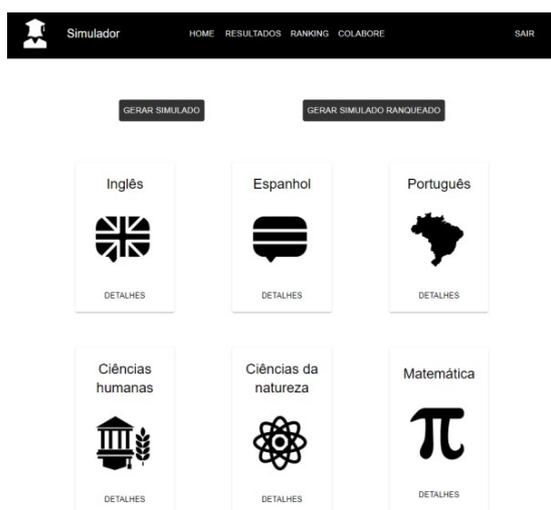
Nesta seção, analisam-se projetos que disponibilizam simulados voltados ao ENEM, com ênfase em suas funcionalidades, abordagens metodológicas e contribuições para a preparação dos estudantes.

Um exemplo de proposta com objetivos semelhantes é a monografia “Desenvolvimento de uma Aplicação Web para Simulados do ENEM” (SILVA, 2023), que descreve o processo de criação de uma plataforma web para aplicação de simulados baseados em edições anteriores do exame.

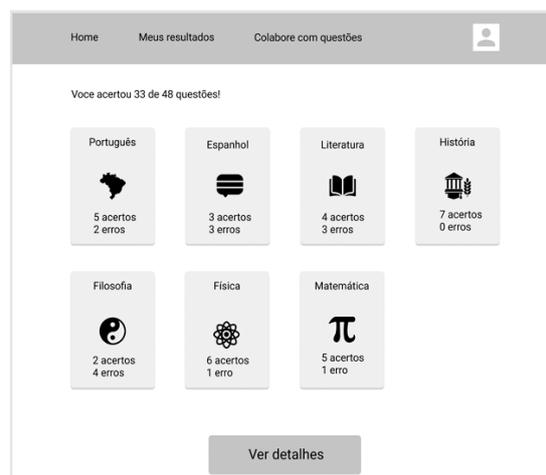
A aplicação permite a geração de simulados personalizados, nos quais o usuário define a quantidade de questões desejadas por área de conhecimento. A seleção das questões é realizada de forma aleatória pela aplicação, com base em um banco de dados composto por edições anteriores do ENEM. Há dois tipos de simulado disponíveis: o simulado comum, voltado exclusivamente para estudo, e o simulado ranqueado, que registra o desempenho dos usuários em um *ranking* geral. Essa abordagem busca incentivar a prática contínua por meio de uma dinâmica competitiva. A Figura 3.1a ilustra a tela inicial da plataforma desenvolvida.

A metodologia adotada no desenvolvimento envolveu a análise de tecnologias atuais, a implementação de uma interface intuitiva e a aplicação de boas práticas de Engenharia de Software. Foi construída uma base de dados com questões de vestibulares anteriores, integrada a um microserviço com o objetivo de facilitar futuras expansões e permitir a integração com outras soluções educacionais.

Como contribuição, o projeto propõe uma solução acessível e interativa, que visa otimizar o processo de estudo dos candidatos, permitindo o acompanhamento do progresso individual e a familiarização com o formato das provas. A proposta busca oferecer um ambiente de aprendizagem dinâmico, alinhado às demandas dos estudantes que almejam um melhor desempenho em exames seletivos. A Figura 3.1b apresenta a tela de *feedback* oferecida aos usuários após a realização dos simulados.



(a) Tela inicial da aplicação, com opções para configurar simulados personalizados com base nas áreas de conhecimento.



(b) Tela de feedback apresentada ao final do simulado, com informações sobre o desempenho individual e incentivo à prática contínua.

Figura 3.1: Telas da aplicação proposta por (SILVA, 2023), com foco na personalização de simulados e análise de desempenho dos usuários.

Embora a plataforma cumpra seu objetivo de fornecer simulados baseados em edições anteriores do ENEM, o *feedback* apresentado aos usuários após a realização das atividades é limitado. As informações disponibilizadas não exploram recursos visuais ou análises mais detalhadas por área de conhecimento, o que dificultaria a compreensão do desempenho individual de forma mais clara e estratégica. Essa limitação reduz o potencial pedagógico da ferramenta, pois impede que o estudante

identifique com precisão seus pontos fortes e fracos, comprometendo um estudo mais direcionado.

Outro projeto relevante para a preparação dos estudantes para o ENEM é o Simulado do Descomplica (DESCOMPLICA, 2025). Essa plataforma online oferece simulados completos, permitindo que os usuários testem seus conhecimentos em condições semelhantes às do exame oficial.

A plataforma se destaca por disponibilizar provas personalizadas e adaptativas, possibilitando que os estudantes escolham áreas do conhecimento específicas para treinar. Além disso, o sistema fornece um *feedback* detalhado sobre o desempenho do usuário, apontando pontos fortes e áreas que precisam de mais atenção, conforme ilustrado na Figura 3.2.

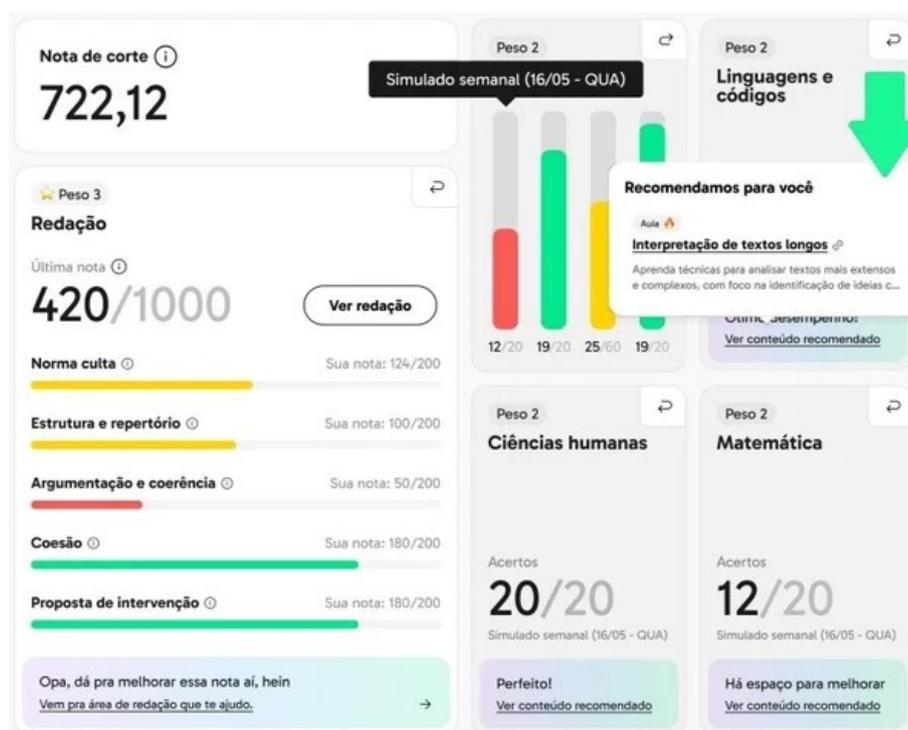


Figura 3.2: Tela de feedback da plataforma Descomplica, indicando acertos, erros e áreas de conhecimento que exigem reforço.

Um dos diferenciais do Simulado do Descomplica é sua interface amigável e responsiva, permitindo que os estudantes realizem os testes com facilidade, seja em um computador ou em dispositivos móveis. Além disso, a plataforma adota um sistema inteligente de correção automática baseado na Teoria de Resposta ao Item

Teoria de Resposta ao Item (TRI)¹, a mesma utilizada no ENEM.

As questões presentes nos simulados são, em sua maioria, retiradas de edições anteriores do ENEM, abrangendo provas desde 2009 até as mais recentes. Isso proporciona aos estudantes uma familiarização com o estilo e o nível de dificuldade das questões que podem encontrar no exame oficial. Além disso, os simulados são organizados por áreas do conhecimento, permitindo que os usuários foquem em disciplinas específicas conforme suas necessidades.

Adicionalmente, o Descomplica oferece suporte a videoaulas explicativas e materiais complementares, enriquecendo a experiência de aprendizagem. O foco na análise individualizada de desempenho possibilita que os estudantes identifiquem lacunas em seu aprendizado e elaborem estratégias mais eficazes para aprimorar seus resultados no exame.

Embora o Descomplica seja uma plataforma amplamente conhecida para preparação ao ENEM, ela apresenta algumas limitações que podem impactar a experiência dos estudantes. Um dos principais pontos é o custo de acesso, que pode ser um obstáculo para alunos com menos recursos financeiros. Além disso, apesar de oferecer filtragem das questões por área de conhecimento, a plataforma não permite uma personalização mais detalhada, como a seleção por temas específicos ou tipos variados de simulado, o que poderia tornar o estudo mais direcionado e eficiente. Essas limitações podem reduzir a flexibilidade da preparação, dificultando a adaptação do material às necessidades individuais dos usuários.

3.3 Proposta de Sistema

Diante das persistentes disparidades educacionais previamente evidenciadas, este trabalho propõe o desenvolvimento do SimuLab, uma plataforma web de apoio à preparação para o ENEM, estruturada em três módulos principais: Simulados Personalizados, Módulo de Redação e Painel Analítico de Desempenho.

¹TRI é um modelo estatístico utilizado para estimar a proficiência dos candidatos, considerando não apenas a quantidade de acertos, mas também a dificuldade das questões e a coerência das respostas.

A concepção do SimuLab fundamenta-se na necessidade de mitigar as disparidades educacionais que afetam estudantes em situação de vulnerabilidade, que muitas vezes enfrentam limitações no acesso a recursos educacionais e carecem de orientação sistemática ao longo do processo de preparação para o ENEM.

Conforme discutido anteriormente, a aprendizagem ativa por meio de simulados tem potencial para fortalecer a retenção de conteúdos e reduzir a ansiedade pré-prova. Nesse contexto, a proposta busca oferecer um ambiente que favoreça o estudo autônomo, contínuo e orientado por dados. Por meio da aplicação de simulados personalizados e da disponibilização de *feedbacks* analíticos, espera-se contribuir de maneira efetiva para o processo de aprendizagem dos estudantes, promovendo tanto a consolidação de conteúdos quanto o desenvolvimento de estratégias mais eficientes de preparação.

Nas subseções seguintes, serão detalhadas as funcionalidades de cada módulo — elementos centrais para o atendimento dos objetivos propostos — evidenciando seus propósitos e principais recursos. Em seguida, apresentam-se os requisitos funcionais, que definem as ações esperadas do sistema, e as regras de negócio, que estabelecem as diretrizes operacionais. Posteriormente, expõem-se os casos de uso, ilustrando as interações previstas entre os usuários e a plataforma. Por fim, detalha-se o modelo de dados, que descreve a estrutura lógica e técnica subjacente ao SimuLab.

3.3.1 Funcionalidades

O SimuLab organiza-se em três funcionalidades principais: simulados personalizados, módulo de redação e painel analítico de desempenho. Esses componentes são projetados para oferecer um suporte direto e orientado à preparação para o ENEM, com foco na personalização das atividades de estudo e no acompanhamento do desempenho.

3.3.1.1 Simulados personalizados

Os simulados personalizados configuram-se como o núcleo central de interação entre o estudante e a plataforma, representando a principal ferramenta para a construção de uma trajetória de estudos autônoma. Por meio deles, o estudante responde às questões e recebe *feedbacks*, favorecendo o diagnóstico contínuo de desempenho. Cada resposta registrada é utilizada para alimentar o painel de desempenho geral, oferecendo uma visão analítica e temporal da evolução individual ao longo do processo de preparação.

A plataforma proporciona ampla autonomia ao estudante na definição dos parâmetros de criação de seus simulados, alinhando a experiência de uso às suas necessidades de estudos. A personalização pode ocorrer em três níveis hierárquicos: Área de Conhecimento, Matéria ou Tópico — com a possibilidade de selecionar a quantidade de questões, variando de zero a cento e oitenta, conforme a necessidade do estudante. Embora o ENEM oficialmente categorize as questões apenas por Áreas de Conhecimento, a inclusão das camadas de Matéria e Tópico no sistema permite uma filtragem mais detalhada, facilitando o foco nas principais dificuldades identificadas pelo estudante.

Cada Área de Conhecimento agrupa diversas Matérias, as quais, por sua vez, podem ser subdivididas em Tópicos específicos. Ao selecionar múltiplos elementos, o sistema distribui automaticamente o total de questões de forma equilibrada entre os itens escolhidos. Por exemplo, ao solicitar dez questões abrangendo dois Tópicos, o sistema alocará cinco questões para cada um, assegurando proporcionalidade.

Além dos filtros de conteúdo, a plataforma oferece um recurso de *status* de questão: o estudante pode optar por responder apenas questões inéditas — aquelas que nunca foram respondidas — ou revisar exclusivamente questões que foram respondidas incorretamente no último contato. Esse filtro possibilita maior foco nas principais dificuldades, reforçando o aprendizado nas áreas onde o estudante apresenta lacunas.

A plataforma oferece dois tipos de filtros temporais. No filtro por edição, o estudante pode responder questões de um ano específico do ENEM sem limite de

tempo, permitindo uma revisão detalhada daquela prova. Já no filtro por dia de prova, o tempo simula o cronograma real do exame: o primeiro dia, que inclui Linguagens, Códigos, Tecnologias, Ciências Humanas e Redação, tem duração máxima de cinco horas e trinta minutos; o segundo dia, com Matemática e Ciências da Natureza, dura até cinco horas. Nesse formato, o cronômetro encerra automaticamente o simulado ao final do tempo previsto, aproximando a experiência das condições reais de prova. Nos demais formatos, o cronômetro serve apenas como referência para ajudar o estudante a controlar o tempo durante o estudo.

Durante a realização do simulado, o sistema disponibiliza uma barra lateral de navegação que concentra informações essenciais para o acompanhamento do progresso da prova. Nela, o estudante pode visualizar rapidamente o *status* de cada questão: respondida, não respondida. Além disso, o cronômetro oficial do simulado, quando aplicável, também é exibido nesse espaço, permitindo o controle do tempo restante de forma contínua e acessível.

Essa funcionalidade foi concebida para simular, no ambiente virtual, o papel do cartão-resposta presente na aplicação oficial do ENEM, no qual o candidato tem uma visão geral das questões já respondidas. No contexto digital, a ausência dessa barra obrigaria o estudante a percorrer individualmente todas as questões para verificar seu avanço, prejudicando a fluidez da experiência.

Após a conclusão do simulado, essa mesma barra mantém sua relevância ao permitir a revisão do desempenho, destacando graficamente as questões corretas em verde, incorretas em vermelho e não respondidas em branco, facilitando uma análise visual rápida e intuitiva dos resultados.

Na correção pós-simulado, as alternativas das questões são sinalizadas de forma visual para facilitar a compreensão dos resultados. Respostas corretas aparecem com destaque em verde, indicando que o estudante escolheu a alternativa certa. Respostas incorretas são marcadas em vermelho, e a alternativa correta é exibida em verde, permitindo ao estudante identificar o erro e a resposta certa. Já nas questões não respondidas, nenhuma alternativa é marcada como escolhida, mas a correta aparece com uma borda verde, indicando qual seria a resposta esperada. Essa visualização

facilita a análise dos erros e acertos, tornando o processo de revisão mais eficiente e contribuindo para a aprendizagem contínua.

Para garantir transparência em todo o ciclo de uso, cada simulado inicia com o *status* Pendente, permanece assim até que o estudante finalize manualmente ou até o término automático do tempo, quando então recebe o *status* Concluído. Nos simulados que incluem redação, após o envio do texto ele passa a constar com o *status* Corrigindo Redação enquanto aguarda avaliação. Essa gestão assegura que o estudante possa acompanhar, em cada etapa, se o simulado está aguardando início, finalizado ou em fase de correção.

Além disso, a plataforma disponibiliza uma página de listagem de simulados em que o estudante visualiza todos os simulados realizados, incluindo o tipo de simulado, os filtros aplicados, o *status* e a quantidade de questões selecionadas. Nos simulados com *status* Concluído, são exibidos também o número de acertos e erros. Essa visão facilita o gerenciamento das atividades de estudo e o gerenciamento dos simulados por parte dos estudantes.

3.3.1.2 Redação

A redação se destaca pelo peso significativo na composição da nota final do ENEM e pela necessidade de um controle rigoroso do tempo durante a escrita. Na plataforma, ela se apresenta como um tipo específico de simulado, com características e tratamentos próprios, mas totalmente integrado à lógica geral do sistema. Assim como os demais simulados, a redação possui *status* associados ao seu ciclo de uso (como Pendente, Corrigindo Redação e Concluído), pode ser listada na página geral de simulados do estudante, e segue o mesmo fluxo de criação, realização e posterior análise de desempenho.

Para atender às exigências do exame, a plataforma oferece dois formatos de prática: simulados exclusivos de redação e redações integradas ao simulado do primeiro dia de prova. Nos simulados exclusivos, o estudante pode escolher entre temas oficiais do INEP ou temas inéditos gerados por inteligência artificial. Essa funcionalidade busca contornar a limitação imposta pelo número restrito de temas divulgados oficialmente,

oferecendo uma variedade maior de propostas para prática contínua. Já no simulado do primeiro dia, um tema inédito é disponibilizado automaticamente — podendo ser tanto um tema oficial quanto um tema gerado por inteligência artificial — com o objetivo de reproduzir integralmente as condições reais do exame.

Para reproduzir fielmente as condições do ENEM e evitar distorções da digitação direta — como a perda da noção de espaço disponível, o ritmo acelerado que não reflete a escrita à mão e o descuido com altura e espaçamento de linhas — a plataforma orienta o estudante a produzir sua redação em papel antes de iniciar o simulado. A escrita manual, além de exigir disciplina no gerenciamento do tempo, é naturalmente mais lenta e limitada pelas sete a trinta linhas estipuladas pelo INEP. Esse exercício prepara o aluno para sentir o mesmo controle de tempo e espaço que terá na prova oficial.

Concluído o texto manuscrito, o estudante transcreve integralmente o manuscrito para o editor da plataforma, preservando o número de linhas e o espaçamento. O sistema registra o tempo desde a leitura do tema até o estudante finalizar a redação ou atingir o tempo limite, desconsiderando exclusivamente o período de digitação.

A correção da redação é realizada por meio de inteligência artificial, que segue os cinco critérios de avaliação estabelecidos pelo INEP, atribuindo até duzentos pontos para cada um deles e aplicando penalidades de vinte pontos conforme as orientações oficiais. Além da pontuação, o sistema fornece justificativas detalhadas para cada critério avaliado. Esse processo busca simular com fidelidade as condições de correção da prova real, oferecendo ao estudante um retorno preciso e fundamentado sobre seu desempenho.

Com o objetivo de promover um acompanhamento mais eficaz do desempenho dos estudantes e incentivar a produção textual de forma planejada e reflexiva, a plataforma estabelece o limite de uma redação por semana por usuário. Essa restrição busca assegurar um intervalo adequado entre as submissões, permitindo que o estudante tenha tempo suficiente para assimilar o *feedback* recebido e aprimorar suas habilidades de escrita antes de realizar uma nova tentativa. Além disso, a definição desse intervalo visa preservar a estabilidade e o bom funcionamento do

serviço, beneficiando o conjunto dos usuários da plataforma.

3.3.1.3 *Painel de Desempenho*

Para estudantes que não contam com apoio pedagógico ou orientação, a capacidade de monitorar e interpretar o próprio desempenho torna-se essencial. Com o objetivo de atender a essa necessidade, a plataforma oferece um painel de desempenho que centraliza e organiza todas as informações relacionadas ao histórico de resolução de questões e simulados.

Esse painel funciona como uma ferramenta analítica abrangente, permitindo que o estudante acompanhe sua evolução de forma contínua, baseada em dados objetivos. A interface é dividida em seções que exploram o desempenho sob diferentes perspectivas, visando oferecer um diagnóstico preciso sobre o nível de preparo em relação aos conteúdos exigidos pelo ENEM.

Na seção de visão geral, o estudante tem acesso a indicadores que sintetizam sua trajetória na plataforma, como o número total de simulados realizados, a quantidade de questões respondidas, a média de acertos e a taxa geral de desempenho. Esses dados fornecem um panorama claro e objetivo sobre o esforço empreendido e os resultados alcançados até o momento. Complementando essa visão, um gráfico de barras ilustra, para cada simulado, o número de acertos e erros, facilitando a identificação de tendências de progresso ou regressão. Além disso, um gráfico de pizza destaca as categorias de conteúdo com melhor desempenho, oferecendo um primeiro diagnóstico das áreas de maior domínio do estudante.

A seção Desempenho foca na comparação entre o rendimento individual e a média geral dos usuários da plataforma. Esse recurso é particularmente útil para estudantes autônomos, que muitas vezes não possuem um referencial externo para avaliar seu nível de preparação. A interface apresenta comparativos percentuais entre a média do usuário, a média da comunidade e a diferença entre elas. Um gráfico de dispersão, intitulado Afinidade por Disciplina, permite visualizar quais áreas do conhecimento apresentam melhor desempenho médio, facilitando a identificação de possíveis afinidades ou dificuldades específicas.

Na seção Categorias, a análise do desempenho é aprofundada por tópicos. Dois gráficos de pizza evidenciam, respectivamente, as categorias com maior e menor taxa de acerto, possibilitando uma compreensão mais segmentada do desempenho. Essa abordagem permite ao estudante estruturar seu plano de estudos de maneira mais estratégica, priorizando conteúdos que requerem maior atenção. Com base nessas informações, a plataforma ainda gera recomendações de estudo personalizadas, funcionando como uma forma de orientação pedagógica automatizada.

Por fim, a seção Histórico apresenta, em ordem cronológica, todas as tentativas de simulados realizadas pelo usuário. A tabela exibe informações como data de realização, *status* de conclusão, número de acertos, erros e a taxa de acerto correspondente. Essa organização facilita o retorno a tentativas anteriores, promovendo uma análise detalhada da evolução ao longo do tempo.

De modo geral, o painel de desempenho tem como principal objetivo não apenas fornecer estatísticas, mas transformar o histórico de interação do estudante com a plataforma em uma fonte contínua de *feedback*. Essa abordagem visa compensar a ausência de acompanhamento pedagógico direto, promovendo a autonomia do estudante para diagnosticar dificuldades, reconhecer avanços e planejar seus próximos passos de forma mais eficaz.

3.3.2 Requisitos Funcionais

Os requisitos funcionais descrevem as funcionalidades que o sistema deve oferecer para atender às necessidades dos usuários. Eles especificam o que o sistema deve fazer, detalhando as interações possíveis e os comportamentos esperados. Esses requisitos estão ilustrados na Tabela 3.1, que apresenta as funcionalidades essenciais a serem implementadas.

Além de servir como base para o desenvolvimento, garantindo que todas as funcionalidades essenciais sejam implementadas corretamente, esses requisitos são importantes para orientar testes, validações e futuras melhorias no sistema.

Tabela 3.1: Requisitos Funcionais

ID	Requisito	Descrição
RF-01	Cadastrar usuário	Permitir que o usuário cadastre uma conta com nome completo, e-mail e senha.
RF-02	Autenticar usuário	Permitir que o usuário acesse o sistema por meio de e-mail e senha.
RF-03	Encerrar sessão	Permitir que o usuário encerre sua sessão de forma segura.
RF-04	Criar simulado	Permitir que o usuário crie e realize simulados configurando filtros por área, matéria e tópico.
RF-05	Alterar resposta	Permitir que o usuário altere suas respostas enquanto o simulado estiver em andamento.
RF-06	Visualizar resultados do simulado	Permitir que o usuário visualize os acertos, erros e gabarito.
RF-07	Criar redação	Permitir que o usuário escreva uma redação com tema e cronômetro ativado.
RF-08	Corrigir redação	Corrigir as redações que foram finalizadas.
RF-09	Exibir painel de desempenho	Exibir ao usuário um resumo com gráficos de evolução e taxa de acerto.
RF-10	Listar histórico de simulados	Listar simulados finalizados com data, tipo e resultados.
RF-11	Listar histórico de redações	Listar redações enviadas e mostrar evolução das notas por competência.

3.3.3 Regras de Negócios

As regras de negócios estabelecem os parâmetros e os processos que regem o funcionamento do sistema, definindo diretrizes precisas para o tratamento dos dados e a execução das funcionalidades. Essas regras determinam a personalização dos simulados conforme os parâmetros definidos pelo usuário, os critérios de avaliação da redação e a apresentação das métricas de desempenho.

Essas diretrizes estão ilustradas na Tabela 3.2, que descreve com mais detalhes os processos envolvidos. Ao padronizar esses processos, as regras de negócios asseguram a coerência, a eficiência e a integridade do sistema, facilitando sua manutenção e evolução, além de garantir que o produto final esteja alinhado aos objetivos e às necessidades dos usuários.

Tabela 3.2: Regras de Negócio

ID	Regra de Negócio	Descrição	Requisitos Associados
RN-01	Formato de e-mail válido	O endereço de e-mail informado deve obedecer ao padrão de formatação válido.	RF-01
RN-02	Unicidade do e-mail	Cada e-mail cadastrado deve ser único no sistema; não é permitida a duplicidade.	RF-01
RN-03	Complexidade mínima da senha	A senha deve conter no mínimo oito caracteres, incluindo letras maiúsculas, minúsculas, números e símbolos.	RF-01
RN-04	Distribuição proporcional de questões	As questões devem ser distribuídas de forma equilibrada entre os tópicos selecionados pelo usuário.	RF-04

Tabela 3.2 – continuação da página anterior

ID	Regra de Negócio	Descrição	Requisitos Associados
RN-05	Filtros de seleção de questões	Devem ser aplicados filtros por status (inéditas/últimos erros), ano e quantidade de questões.	RF-04
RN-06	Inclusão completa de edição do ENEM	Todas as questões da edição selecionada devem ser incluídas no simulado, sem aplicação de filtros adicionais.	RF-04
RN-07	Modo por dia de prova	O modo de dia de prova deve simular tempo e estrutura oficial: Dia 1 (cinco horas e trinta minutos) e Dia 2 (cinco horas), com cronômetro automático.	RF-04
RN-08	Critério de finalização	Finalizar simulado ao esgotar o tempo ou por ação explícita do usuário	RF-04
RN-09	Correção visual de questões	Destacar acertos e erros após finalização, mostrando gabarito.	RF-04
RN-10	Temas de redação oficiais e inéditos	Devem ser disponibilizados temas oficiais e inéditos, incluindo propostas geradas por inteligência artificial.	RF-07

Tabela 3.2 – continuação da página anterior

ID	Regra de Negócio	Descrição	Requisitos Associados
RN-11	Registro do tempo de escrita	O sistema deve registrar apenas o tempo efetivo dedicado à elaboração do texto.	RF-07
RN-12	Frequência de correção de redação	Cada usuário poderá ter apenas uma redação corrigida por inteligência artificial a cada 7 dias.	RF-08
RN-13	Comparativo de desempenho	O sistema deve exibir o desempenho percentual do usuário em comparação com a média dos demais participantes.	RF-09
RN-14	Análise por tópico	O sistema deve identificar os tópicos com melhor e pior desempenho e sugerir conteúdos para estudo direcionado.	RF-09
RN-15	Registro de simulados finalizados	Cada simulado finalizado deve ser registrado automaticamente com data e resultados obtidos.	RF-10, RF-11

3.3.4 Casos de Uso

Um Caso de Uso (UC) descreve um cenário no qual um ator interage com o sistema para alcançar um objetivo específico. Cada UC contém uma sequência de passos que detalham o fluxo principal da interação, além de possíveis variações ou exceções. A seguir, são apresentados os casos de uso mais relevantes do sistema.

UC-01: Cadastrar usuário

Ator Principal: Usuário

Descrição: Permitir que um usuário crie uma conta fornecendo nome, e-mail e senha válidos.

Pré-condição: O usuário não possuir conta no sistema.

Pós-condição: Conta criada no sistema.

Fluxo Principal:

1. O usuário acessa a página de cadastro.
2. O sistema solicita dados.
3. O usuário preenche nome, e-mail e senha.
4. O usuário envia o formulário.
5. O sistema coleta e valida os dados.
6. O sistema exibe mensagem de sucesso e redireciona o visitante para a tela de *login*.

Fluxo Alternativo (5): E-mail já cadastrado ou inválido.

- a) O sistema detecta que o e-mail informado já está cadastrado ou possui formato inválido, exibe uma mensagem de erro e retorna ao passo 2 do fluxo Principal.

Fluxo Alternativo (5): Senha não atende aos critérios mínimos.

- a) O sistema detecta que a senha não atende aos critérios mínimos, exibe mensagem de erro apropriada e retorna ao passo 2 do fluxo Principal.

UC-02: Autenticar Usuário

Ator Principal: Usuário

Descrição: Permitir que o usuário acesse sua conta informando e-mail e senha válidos.

Pré-condição: Usuário já possui conta ativa.

Pós-condição: Usuário autenticado e direcionado ao painel principal.

Fluxo Principal:

1. O usuário acessa a página de *login*.
2. O sistema solicita dados.
3. O usuário preenche e-mail e senha.
4. O usuário envia credenciais.
5. O sistema verifica existência da conta e validade da senha.
6. O sistema gera sessão autenticada e redireciona ao painel principal.

Fluxo Alternativo (5): Credenciais inválidas ou conta inexistente.

- a) O sistema não reconhece as credenciais fornecidas, exibe uma mensagem de erro e retorna ao passo 2 do Fluxo Principal.

UC-03: Encerrar Sessão

Ator Principal: Usuário

Descrição: Permitir que o usuário encerre sua sessão de forma segura.

Pré-condição: Usuário deve estar autenticado.

Pós-condição: Sessão do usuário invalidada.

Fluxo Principal:

1. O usuário seleciona “Sair” no menu de perfil.
2. O sistema invalida a sessão.

3. O sistema redireciona para a tela de *login*.

UC-04: Criar Simulado

Ator Principal: Usuário

Descrição: Permitir que o usuário crie um simulado aplicando filtros de área, matéria, tópico, status e ano, definindo modo de aplicação.

Pré-condição: Usuário autenticado.

Pós-condição: Simulado configurado e iniciado.

Fluxo Principal:

1. O usuário acessa “Criar Simulado”.
2. o sistema solicita dados.
3. O usuário define os filtros para criação do simulado.
4. O usuário e confirma.
5. O sistema processa filtros selecionados.
6. O sistema seleciona questões.
7. O sistema cria o simulado.
8. O sistema direciona usuário para o simulado.
9. O sistema inicia cronômetro, se aplicável.

Fluxo Alternativo (5): Filtros inválidos.

- a) O sistema informa que não é possível criar simulado com os filtros selecionados e retorna para o passo 2 do fluxo principal.

Fluxo Alternativo (6): Número de questões insuficiente.

- a) O sistema informa que não há questões suficientes para criar o simulado com as características desejadas e retorna para o passo 2 do fluxo principal.

UC-05: Visualizar Resultados do Simulado

Ator Principal: Usuário

Descrição: Permitir que o usuário visualize acertos, erros, gabarito e tempo gasto após finalizar o simulado.

Pré-condição: Simulado finalizado.

Pós-condição: Usuário visualiza relatório completo do simulado.

Fluxo Principal:

1. O usuário acessa a página de simulados.
2. O usuário seleciona o simulado.
3. O sistema apresenta acertos, erros, gabarito destacado e tempo total.

Fluxos Alternativo (2): Simulado ainda em andamento

1. O sistema identifica que o simulado ainda não foi finalizado e redireciona o usuário para continuar sua realização.

UC-06: Criar Redação

Ator Principal: Usuário

Descrição: Permitir que o usuário elabore uma redação com tema oficial ou inédito gerado por inteligência artificial, com registro do tempo de escrita.

Pré-condição: Usuário autenticado.

Pós-condição: Redação salva e pronta para correção; tempo registrado.

Fluxo Principal:

1. O usuário acessa a página “Criar Simulado”.
2. O usuário seleciona um simulado do tipo redação.
3. O usuário escolhe se deseja um tema oficial ou gerado por inteligência artificial.
4. O usuário confirma a escolha.
5. O sistema recupera ou gera o tema conforme selecionado.
6. O sistema redireciona o usuário para a área de escrita da redação.
7. O sistema exibe as instruções, o tema e os textos motivadores, iniciando a contagem do tempo de escrita.
8. O usuário escreve a redação.
9. O usuário finaliza a escrita.
10. O sistema redireciona o usuário para a tela de submissão e registra o tempo total.
11. O usuário digita a versão final da redação e a envia.
12. O sistema salva o texto para correção.

Fluxo Alternativo (5): Tema indisponível ou erro ao gerar tema

- a) O sistema não consegue recuperar o tema selecionado ou gerar um novo, exibe uma mensagem de erro e retorna ao passo 2 do fluxo principal.

UC-07: Corrigir Redação

Ator Principal: Sistema

Descrição: Permitir que o sistema realize a correção automática de redações submetidas por usuários, avaliando por competência e gerando nota e *feedback*.

Pré-condição: Redação submetida para correção.

Pós-condição: Redação corrigida, com nota final e *feedback* detalhado por competência apresentados ao usuário.

Fluxo Principal:

1. O sistema realiza uma análise preliminar para verificar se a redação é passível de anulação.
2. O sistema avalia a redação com base nos critérios por competência.
3. O sistema calcula a nota final e gera o *feedback* por competência.
4. O sistema registra a correção.

Fluxo Alternativo (2): Sistema de correção indisponível

- a) O sistema não consegue corrigir redação e o caso de uso é encerrado.

Fluxo Alternativo (3): Sistema de correção indisponível

- a) O sistema não consegue corrigir redação e o caso de uso é encerrado.

UC-08: Exibir Painel de Desempenho

Ator Principal: Usuário

Descrição: Apresentar ao usuário gráficos de evolução, taxa de acertos e comparativos com a média geral.

Pré-condição: O usuário possui histórico de simulados ou redações.

Pós-condição: O usuário visualiza indicadores de desempenho.

Fluxo Principal:

1. O usuário acessa o menu “Painel de Desempenho”.
2. O sistema agrega os dados de simulados e redações do usuário.

3. O sistema gera e exibe gráficos de evolução, taxa de acertos e comparativos com a média geral.

Fluxo Alternativo (2): Usuário não possui histórico de atividades.

- a) O sistema informa que ainda não há dados disponíveis e recomenda que o usuário realize um simulado ou escreva uma redação.

3.3.5 Modelo de Dados

Esta seção apresenta a modelagem de dados proposta para o sistema, elaborada a partir dos requisitos funcionais e regras de negócio discutidos anteriormente. A Figura 3.3 ilustra o Modelo de Entidade-Relacionamento (MER), evidenciando as entidades, atributos e seus relacionamentos

No centro do modelo está a entidade Usuário (*User*), responsável por armazenar as informações básicas de cada estudante e por manter o vínculo com os simulados realizados e as redações avaliadas. Por exemplo, o atributo *lastEssayDate* registra a data da última redação corrigida, permitindo impor regras de limitação de envio sem consultar o histórico completo de cada usuário — o que contribui para a otimização do tempo de resposta do sistema.

Os simulados são representados como entidades independentes (*Simulated*), incluindo dados como tipo, período de realização, status e métricas de desempenho. A entidade está ligada ao usuário e pode conter uma redação associada a redação (*Essay*), quando aplicável. Além disso, cada simulado armazena o tempo total de realização, a pontuação final da redação e outras características baseadas nos parâmetros de criação do simulado.

A relação entre simulados e questões é intermediada pela entidade *Simulated-Question*, permitindo armazenar informações específicas da interação do usuário com cada item. São registrados se o usuário já acessou a questão, se acertou ou errou, qual alternativa marcou, além das datas de resposta e da última visualização. Essa

granularidade não só possibilita o rastreamento detalhado do progresso do estudante, mas também viabiliza a criação de simulados adaptativos com base no histórico de interação: questões nunca vistas ou respondidas incorretamente em contatos anteriores podem ser priorizadas em novas versões do simulado.

As questões (*Question*) foram modeladas com foco em flexibilidade e aderência às necessidades pedagógicas da plataforma. Cada registro contempla atributos como o ano de aplicação, um contexto textual opcional para enriquecer a compreensão do enunciado e uma introdução específica para as alternativas, quando aplicável.

Essa lógica de associação foi expandida para estruturar uma organização pedagógica robusta, baseada na separação entre disciplinas, matérias e categorias. Essa modelagem não apenas permite reorganizações curriculares de forma ágil, como também otimiza a criação de simulados por meio da aplicação de filtros mais precisos e combináveis, sem necessidade de reprocessar os registros de questões. Em especial, a entidade *QuestionCategory* viabiliza a atribuição de múltiplas categorias a uma mesma questão, ampliando a flexibilidade para análises pedagógicas e extração de estatísticas por diferentes recortes temáticos.

Ao optar por armazenar essas classificações como entidades independentes, a modelagem proporciona maior escalabilidade e facilidade de manutenção, viabilizando a inclusão de novas categorias ou ajustes na organização curricular sem a necessidade de reformulações complexas no banco de dados. Além disso, essa abordagem melhora a eficiência das consultas, garantindo que o sistema possa operar com bom desempenho mesmo diante de um volume crescente de questões e critérios de seleção.

As alternativas (*Alternative*) foram separadas em uma entidade própria, o que confere maior controle sobre o número e o conteúdo das opções de cada questão. Cada alternativa armazena a letra identificadora, o texto descritivo e um indicador correção. Essa modelagem garante que mudanças no formato de avaliação — como aumento ou redução no número de opções ou novos estilos de perguntas do ENEM — sejam facilmente suportadas sem ajustes complexos na estrutura principal.

No módulo de redação (*Essay*), cada tema é tratado como entidade autônoma,

contendo ano, textos motivadores e um indicador que sinaliza se o tema é proveniente do INEP. Esses temas podem ser reutilizados em diversos simulados, e suas correções são armazenadas separadamente por meio da entidade *SimulatedEssayScore*. Esta registra, para cada critério de avaliação, a pontuação obtida e eventual justificativa. Além disso, a estrutura independente de critérios permite ajustes futuros ou inclusão de novos parâmetros de correção sem impacto na estrutura principal.

Como um todo, a modelagem adotada permite ao sistema atender de forma eficiente às necessidades de armazenamento, recuperação e análise de dados. A separação entre as entidades e a estrutura relacional favorece não apenas a integridade das informações, mas também a construção de funcionalidades como a página de desempenho, filtros e os *feedbacks*.

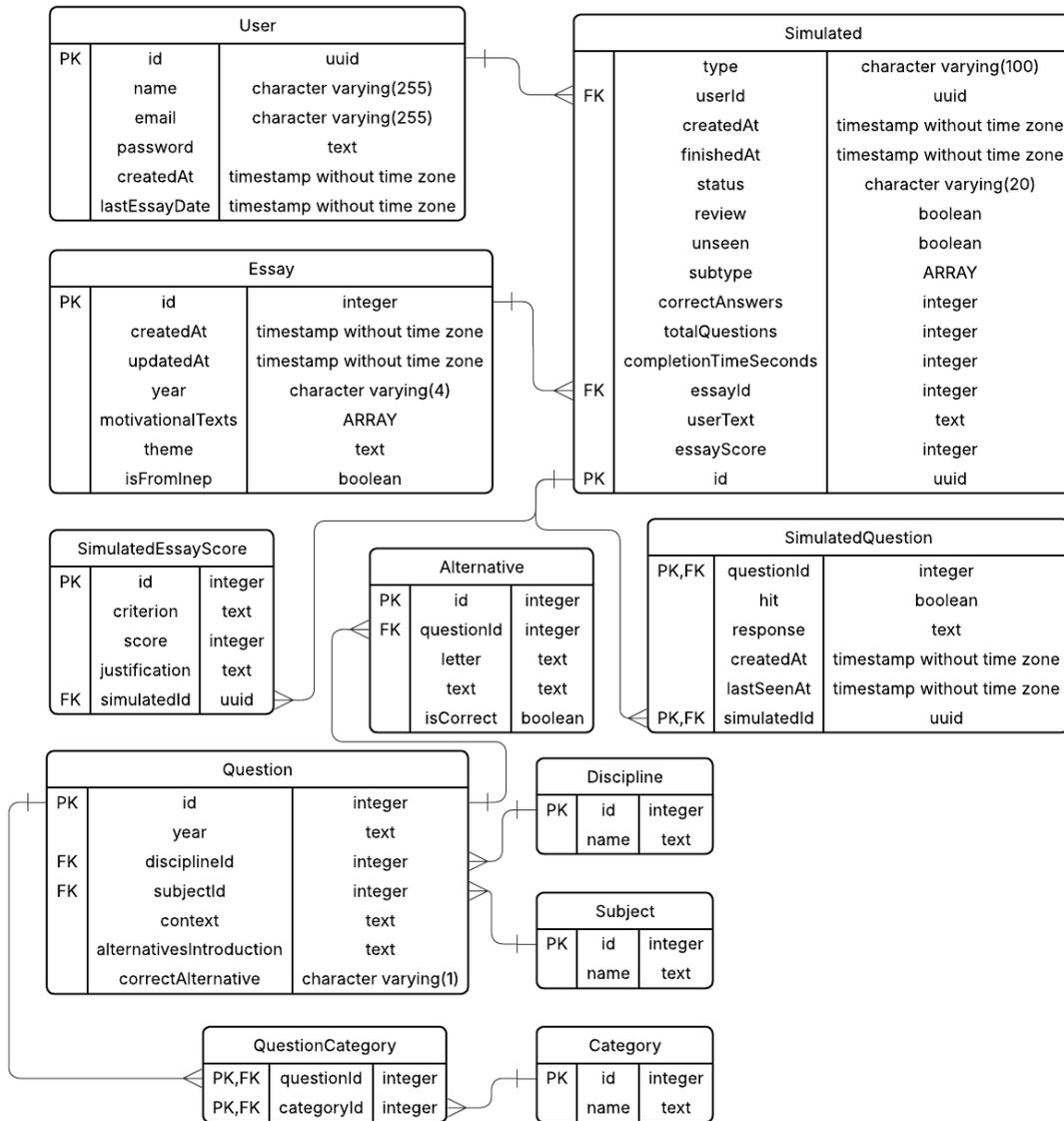


Figura 3.3: Diagrama do Modelo Entidade-Relacionamento proposto.

Capítulo 4

SimuLab

Este capítulo descreve o processo de desenvolvimento da aplicação proposta, abordando desde a obtenção e organização dos dados até a estruturação da arquitetura do sistema e as ferramentas utilizadas em sua construção. O objetivo é detalhar as decisões técnicas e metodológicas tomadas ao longo do projeto, de modo a demonstrar como os recursos foram integrados para alcançar os objetivos propostos.

4.1 Tecnologias Utilizadas

O SimuLab configura-se como uma plataforma de estudo cuja proposta central é oferecer uma experiência personalizada, eficiente e escalável, contemplando desde a aplicação de simulados dinâmicos até a correção automática de redações. Para a consecução desses objetivos, adotou-se uma pilha tecnológica que equilibra robustez, flexibilidade e agilidade no desenvolvimento.

O núcleo da implementação foi desenvolvido em *TypeScript*¹, escolha que se justifica pela introdução de tipagem estática ao *JavaScript*, elemento crucial para a detecção precoce de inconsistências durante a fase de desenvolvimento e para a manutenção da integridade do código à medida que funcionalidades complexas - como filtros dinâmicos e armazenamento em *cache* do progresso dos usuários - são

¹<https://www.typescriptlang.org/>

incorporadas.

Recorreu-se ao *framework* Next.js², que atua como o motor principal para construir toda a aplicação do SimuLab. Ele permite que as páginas sejam pré-renderizadas, ou seja, já vêm prontas para o navegador, o que garante tempos de carregamento reduzidos e desempenho consistente, mesmo com muitos usuários e diferentes tipos de consultas. Com o Next.js, o SimuLab consegue entregar tanto a parte visível para o usuário quanto processar as informações no servidor de forma integrada.

A parte visual do SimuLab foi estruturada com *React*³ para a composição de interfaces modulares e reativas, favorecendo a reutilização de componentes. A estilização fundamenta-se em *Tailwind CSS*⁴, por viabilizar rapidamente *layouts* responsivos e de fácil customização, ao passo que a biblioteca *shadcn/ui*⁵ complementa o arsenal de componentes com soluções padronizadas e acessíveis.

Para gerenciar e armazenar os dados, empregou-se o *Object-Relational Mapping* (ORM) *Prisma*⁶ em conjunto com o banco de dados relacional *PostgreSQL*, combinação que facilita o mapeamento de um modelo de dados abrangente - incluindo simulados, questões e redações - ao mesmo tempo em que garante tipagem forte, segurança na persistência e suporte a consultas complexas, requisitos fundamentais para o tratamento de grandes volumes e variedades de informações.

Adicionalmente, o SimuLab integra modelos avançados de inteligência artificial - como GPT-4⁷ e DeepSeek⁸ - por meio de APIs, de modo a automatizar a correção das redações e promover a categorização semântica das questões. Essa abordagem reduz substancialmente a intervenção manual, eleva a precisão das avaliações e enriquece a experiência do usuário. A seleção desses modelos fundamentou-se em sua reconhecida competência no processamento de linguagem natural e em sua comprovada capacidade de escalar conforme as demandas do sistema.

²<<https://nextjs.org/>>

³<<https://reactjs.org/>>

⁴<<https://tailwindcss.com/>>

⁵<<https://shadcn.com/ui/>>

⁶<<https://www.prisma.io/>>

⁷<<https://openai.com/gpt-4>>

⁸<<https://www.deepseek.ai/>>

4.2 Implementação do Sistema

Nesta seção, apresentamos uma visão detalhada do desenvolvimento do sistema, com foco nas funcionalidades principais e na interação do usuário. Serão discutidos os aspectos do processo de implementação, incluindo decisões técnicas e a forma como essas escolhas atendem aos objetivos do sistema. Cada funcionalidade será descrita em seu contexto, acompanhada de exemplos de fluxo de usuário.

4.2.1 Coleta e Processamento das Questões

A primeira etapa da implementação consistiu na obtenção de questões do ENEM. Embora a base oficial seja disponibilizada pelo INEP no formato PDF, optou-se por utilizar um repositório público no GitHub (YUNGER7, 2024) para agilizar o desenvolvimento. Este repositório contém os dados já estruturados em arquivos JSON, organizados por ano, abrangendo as edições de 2009 a 2023.

Cada arquivo anual continha as 180 questões aplicadas naquela edição do exame, totalizando 2700 questões disponíveis para análise. Essa estrutura pronta em JSON eliminou a necessidade de conversão manual dos dados e facilitou diretamente o uso de *scripts* automatizados para leitura e processamento.

O processo de categorização semântica foi implementado para complementar os dados disponibilizados pelo INEP, que contém apenas a área de conhecimento de cada questão. Para enriquecer essas informações, desenvolveu-se um *script* em *JavaScript* capaz de atribuir automaticamente duas propriedades adicionais a cada item: matéria e tópico específico.

A estrutura de referência utilizada para a classificação das questões foi baseada na Matriz de Referência do ENEM, que estabelece a relação entre cada área do conhecimento e suas respectivas matérias. Por exemplo, a área de Ciências da Natureza está associada a disciplinas como Química, Biologia e Física. A partir dessas matérias, foram definidos manualmente os tópicos mais recorrentes, com base na análise de documentos curriculares e em edições anteriores do exame.

A automatização do processo de categorização utilizou a API da OpenAI com o modelo `gpt-3.5-turbo`, um LLM especializado em tarefas de linguagem natural. O *script* percorria cada questão no arquivo JSON, construía um *prompt* contextualizado com o enunciado e as alternativas, e submetia à API para identificação da matéria e tópico correspondentes.

O preenchimento desses *prompts* foi realizado de forma programática. O sistema iterava sobre um conjunto de objetos JSON, cada um representando uma questão do ENEM. Em seguida, os campos do *template* do *prompt* eram dinamicamente substituídos com base nos valores das chaves do JSON correspondente.

O primeiro *prompt*, especificamente, foi projetado para identificar a matéria principal da questão. Para sua montagem, utilizavam-se informações como o enunciado da questão (`question.context`), as alternativas (`question.alternativeA`, `alternativeB`, etc.) e a disciplina associada à questão (`discipline`), além de uma lista de possíveis matérias (`possibleSubjects`) para guiar a resposta do modelo. A estrutura base desse *prompt* é apresentada a seguir:

```
Dada a seguinte questão: "${question.context} " " +
question.alternativesIntroduction", com as seguintes alternativas:
A) ${question.alternativeA}
B) ${question.alternativeB}
C) ${question.alternativeC}
D) ${question.alternativeD}
E) ${question.alternativeE}
Sabendo que a disciplina é ${discipline}, identifique o único assunto mais
relevante e diretamente relacionado ao tema principal da questão. Assuntos
secundários devem ser ignorados, e o foco deve estar no tema central
tratado pela questão.
Utilize apenas os assuntos da lista a seguir:
Lista de assuntos: ${possibleSubjects.join(", ")}
A resposta deve ser apenas um único assunto, como no exemplo: "assunto1".
```

A resposta esperada vinha formatada para conter unicamente a matéria identificada, sem informações adicionais, facilitando o processamento subsequente pelo sistema.

O segundo *prompt* foi desenvolvido para determinar os tópicos mais relevantes de uma questão, com base na matéria previamente identificada. Para sua construção, foram fornecidos os mesmos dados da questão, porém, em vez da disciplina, o *prompt* recebia a matéria previamente identificada (`subject`) e uma lista de tópicos possíveis (`possibleCategories`) específicos para aquela matéria. O objetivo era obter uma ou, no máximo, duas categorias que representassem o conceito central da questão.

Foram utilizados parâmetros específicos nas requisições à API para garantir consistência nos resultados: temperatura definida em 0.25 para reduzir aleatoriedade, limite de 500 `tokens` por requisição, e configuração para gerar apenas uma resposta por consulta. Essa abordagem permitiu alta eficiência no processamento sem intervenção manual.

Após a classificação, as informações semânticas retornadas pelo modelo foram validadas e incorporadas ao objeto original de cada questão no arquivo JSON. Esse enriquecimento de metadados permitiu implementar mecanismos avançados de filtragem e análise no sistema proposto, superando as limitações da base de dados original.

Um aspecto no processamento das questões do ENEM está na sua natureza textual e multimídia. Os enunciados e alternativas podem apresentar-se exclusivamente como texto, ou incluir imagens e formatações específicas. Para gerenciar essa diversidade, foi necessário implementar um método de tratamento de conteúdo.

As questões utilizavam marcadores no formato ``, onde `UUID` representa um identificador único para uma imagem. As imagens correspondentes a esses identificadores são previamente armazenadas em um serviço de armazenamento em nuvem. No momento da renderização da questão, o `UUID` contido nos marcadores é usado para consultar esse serviço e recuperar a imagem, que é inserida no ponto especificado no texto original.

Além do tratamento de imagens, o sistema interpreta outros marcadores textuais para formatação de conteúdo, como negrito (**texto**) e itálico (*texto* ou texto). Essa lógica opera para converter o texto bruto em um componente renderizado, integrando imagens e marcadores de formatação para exibição.

4.2.2 Next.js

Neste projeto, o Next.js foi adotado como plataforma principal para a aplicação Web, explorando recursos como os diretórios `app` e `pages`, os *hooks* para interações no lado do cliente, e o sistema de autenticação com NextAuth.js. Essa escolha permitiu acelerar o desenvolvimento e garantir uma arquitetura moderna, escalável e de alta performance.

O Next.js utiliza um sistema de roteamento automático baseado na estrutura de arquivos dentro das pastas *pages* ou `app`. Cada arquivo representa uma rota da aplicação, o que facilita a organização do código e a separação de responsabilidades. Neste projeto, essa abordagem foi usada para estruturar as páginas principais da aplicação, como o *dashboard*, o *login*, e os simulados, permitindo também aplicar controladores de forma clara e modular em cada rota.

A Figura 4.1 ilustra a estrutura de rotas do projeto. Cada subpasta da pasta `pages` representa uma rota acessível. Por exemplo, a pasta `createSimulated` corresponde à rota `/createSimulated`, responsável pela criação de novos simulados. A pasta `simulated/[id]` representa uma rota dinâmica que exibe o conteúdo de um simulado específico, identificado por um *id*. Já a pasta `simulationResult/[id]` trata da visualização dos resultados de simulados finalizados. As rotas `login`, `signup` e `notFound` correspondem às funcionalidades de autenticação e página de erro, respectivamente.

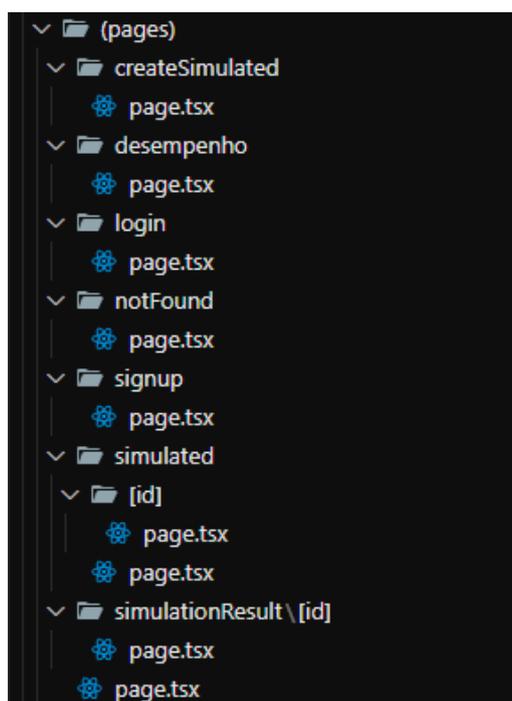


Figura 4.1: Estrutura de rotas da aplicação baseada em páginas

A autenticação foi implementada utilizando o NextAuth.js, uma biblioteca amplamente utilizada em aplicações desenvolvidas com Next.js, por sua integração nativa e facilidade de configuração. Neste projeto, foi adotado o provedor baseado em credenciais (*Credentials Provider*), que permite autenticar usuários por meio de e-mail e senha. Essa abordagem proporciona maior controle sobre a lógica de verificação, realizada por meio de um serviço interno da aplicação.

A persistência da autenticação foi configurada utilizando tokens JWT⁹, eliminando a necessidade de armazenamento de sessões no servidor e favorecendo a escalabilidade. Essa integração com o *NextAuth.js* contribui para uma autenticação segura, flexível e bem alinhada ao ecossistema do *Next.js*.

4.2.3 Arquitetura em Camadas

A arquitetura em camadas constitui um paradigma estrutural que particiona o sistema em níveis hierárquicos, cada qual com responsabilidades bem definidas.

⁹JSON Web *Token* é um padrão aberto que define um formato compacto e seguro para a transmissão de informações entre partes como um objeto JSON. É comumente usado para autenticação e troca de informações em aplicações web.

Essa abordagem promove modularidade, isolando aspectos de interface, regras de negócio e persistência de dados, e estabelece um fluxo de dependências unidirecional. No contexto deste projeto, tal organização foi adotada para garantir escalabilidade, facilidade de manutenção e adaptação a requisitos dinâmicos.

A Figura 4.2 ilustra essa composição em três camadas principais. Cada nível comunica-se apenas com o imediatamente inferior, assegurando baixo acoplamento e mitigando propagação de efeitos indesejados entre módulos.

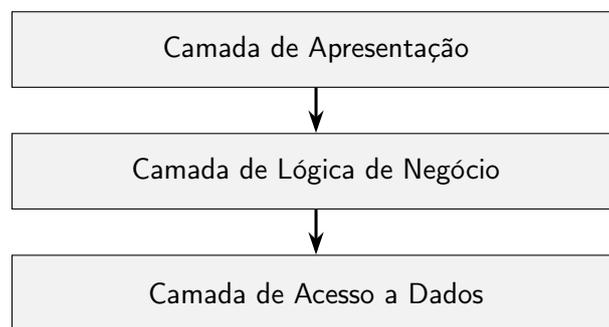


Figura 4.2: Estrutura em camadas adotada no sistema

4.2.3.1 Camada de Apresentação

A Camada de Apresentação é responsável pela interface com o usuário e pela composição das telas da aplicação. Implementada utilizando a biblioteca *React* em conjunto com o sistema de roteamento do *Next.js*, essa camada adota uma abordagem baseada em componentes funcionais, favorecendo reutilização, legibilidade e organização da interface em unidades coesas.

A estrutura dessa camada é dividida em dois níveis principais: os *components*, que são unidades visuais reutilizáveis, e os *containers*, responsáveis por orquestrar a lógica de obtenção e preparação dos dados necessários à renderização. Os *components* recebem *props* e se mantêm puros e focados na renderização, enquanto os *containers* interagem com os serviços da camada de lógica, recuperando informações do usuário, questões, resultados de simulados e outras entidades relevantes.

Além disso, são utilizados *hooks* personalizados (*custom hooks*) para encapsular comportamentos reutilizáveis, como controle de formulários, tratamento de estado

local. Esse padrão contribui para a separação de responsabilidades, tornando o código mais testável e de fácil manutenção.

Por não conter lógica de negócio nem acesso direto ao banco de dados, essa camada funciona como uma consumidora de serviços das camadas inferiores, mantendo um baixo acoplamento e promovendo a flexibilidade da aplicação frente a mudanças na lógica ou estrutura de dados. Essa separação clara entre interface e processamento permite evoluir a interface independentemente das demais partes do sistema.

4.2.3.2 Camada de Lógica

A Camada de Lógica constitui o núcleo funcional e intelectual da aplicação, centralizando as regras de domínio, os fluxos de processamento e a orquestração dos casos de uso. Sua principal responsabilidade é interpretar comandos provenientes da Camada de Apresentação, realizar validações, aplicar regras de negócio e coordenar a comunicação com os repositórios de dados da camada inferior.

No escopo deste projeto, essa camada é organizada em módulos denominados *services*, cada um encapsulando responsabilidades bem definidas, como criação de simulados, filtragem de questões, cálculo de notas, aplicação de critérios de correção, geração de temas de redação, e organização dos resultados. Esses *services* operam de forma isolada dos detalhes de infraestrutura, o que favorece a testabilidade e a reutilização de lógica em diferentes pontos da aplicação.

Além disso, ao concentrar as decisões de negócio nesta camada, o sistema mantém-se mais coeso, facilitando manutenções evolutivas, extensões e adequações a mudanças nos requisitos funcionais.

4.2.3.3 Camada de Acesso a Dados

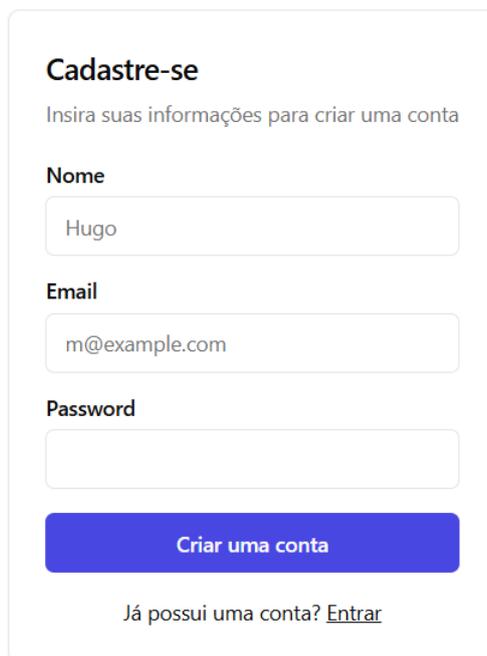
A Camada de Acesso a Dados é responsável por isolar a lógica de persistência e abstrair o mecanismo de comunicação com o banco de dados, de forma a preservar a integridade das demais camadas da aplicação. Neste projeto, essa função é desempenhada utilizando o Prisma em conjunto com o banco de dados *PostgreSQL*.

O Prisma atua como uma ponte entre a aplicação e o banco, oferecendo uma interface tipada e segura para execução de operações de leitura e escrita. Ele gera automaticamente clientes com base em um esquema definido em `schema.prisma`, onde são modeladas as entidades, seus relacionamentos e regras de validação. A partir desse esquema, são criadas migrações versionadas, que mantêm a consistência entre o modelo de dados da aplicação e a estrutura do banco.

As interações com o Prisma são organizadas em *repositories*, os quais encapsulam consultas e comandos específicos de cada entidade. Essa estrutura promove o princípio de separação de responsabilidades, permitindo que alterações na camada de persistência não impactem diretamente os serviços ou a camada de apresentação. Dessa forma, reforça-se o baixo acoplamento e a coesão modular do sistema.

4.2.4 Sistema de Autenticação

O processo de cadastro de usuários tem início na rota `/signup`, onde é exibido um formulário para preenchimento dos dados necessários. A Figura 4.3 ilustra o formulário de cadastro. Ao submeter o formulário, o *NextAuth.js* aciona uma *service* responsável por realizar as validações necessárias, como a verificação da complexidade da senha, a aplicação de criptografia (com *bcrypt*) e a verificação de unicidade do e-mail no banco de dados. Após a validação e o armazenamento seguro dos dados, o usuário é redirecionado automaticamente para a tela de login na rota `/login`, concluindo o fluxo de cadastro com sucesso.



Cadastre-se
Insira suas informações para criar uma conta

Nome

Email

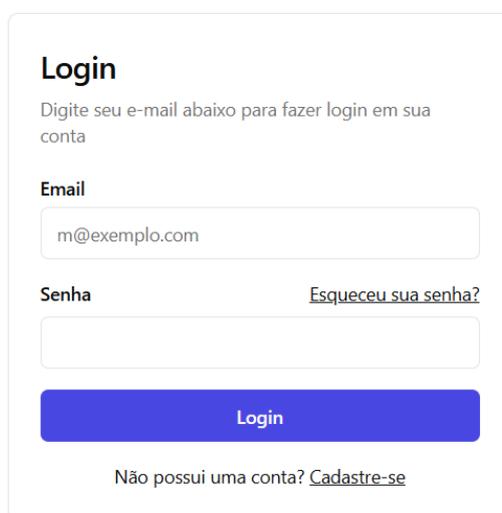
Password

Criar uma conta

Já possui uma conta? [Entrar](#)

Figura 4.3: Tela de cadastro de usuário

Após o preenchimento do formulário de *login*, ilustrado na Figura 4.4, o *NextAuth* encaminha os dados para a *service*, responsável por realizar as validações necessárias, como a verificação do e-mail e a comparação do *hash* da senha. Com as credenciais confirmadas, o usuário é autenticado e redirecionado automaticamente para a página inicial.



Login
Digite seu e-mail abaixo para fazer login em sua conta

Email

Senha [Esqueceu sua senha?](#)

Login

Não possui uma conta? [Cadastre-se](#)

Figura 4.4: Tela de login do sistema

Com o usuário autenticado, ele passa a ter acesso a rotas previamente restritas, podendo navegar pelas opções disponíveis no menu da página inicial. Entre elas, estão a visualização da lista de simulados realizados, a criação de novos simulados e o acesso à área de desempenho, conforme ilustrado na Figura 4.5.



Figura 4.5: Página inicial

4.2.5 Simulados

Ao acessar a página inicial, o usuário tem a opção de navegar até a página de criação de simulados, conforme ilustrado na Figura 4.6. Nessa interface, é possível configurar um novo simulado de acordo com as regras de negócio estabelecidas anteriormente na Tabela 3.2, onde são formalizados os critérios para composição e restrições dos simulados.

Criar Simulado
Configure as opções para o seu simulado personalizado

Filtro
Selecione o tipo de simulado

Área de estudo
Selecione a área de estudo

Quantidade de Questões
1

Opções adicionais:

- Questões Inéditas
- Revisar Questões que Errou

Criar Simulado

Figura 4.6: Tela de criação de simulado

Ao preencher as informações necessárias para a criação do simulado, é gerado um objeto contendo todas as combinações de parâmetros definidos, o qual é enviado para a *service* responsável por esse processo. Caso todos os critérios sejam atendidos, o usuário é redirecionado para a tela do simulado. Se houver alguma inconsistência nos dados, a *service* retorna uma mensagem de erro, exibida diretamente na interface de criação.

Na tela do simulado, a *service* envia ao *hook* uma lista com os identificadores das questões selecionadas. A partir disso, o sistema carrega dinamicamente as questões conforme são visualizadas pelo usuário, realizando a consulta ao banco de dados apenas no momento da visualização. Uma vez carregada, a questão é armazenada localmente, evitando novas requisições caso o usuário retorne a ela posteriormente.

A navegação entre as questões pode ser feita por meio dos botões localizados

no rodapé da tela ou através de uma barra lateral acessível por um botão no canto superior direito. Essa barra, ilustrada na Figura 4.7, exibe todas as questões do simulado, indicando visualmente seu estado: questões ainda não respondidas aparecem em branco, enquanto as já respondidas são destacadas em azul, facilitando a orientação do usuário durante a realização do simulado.

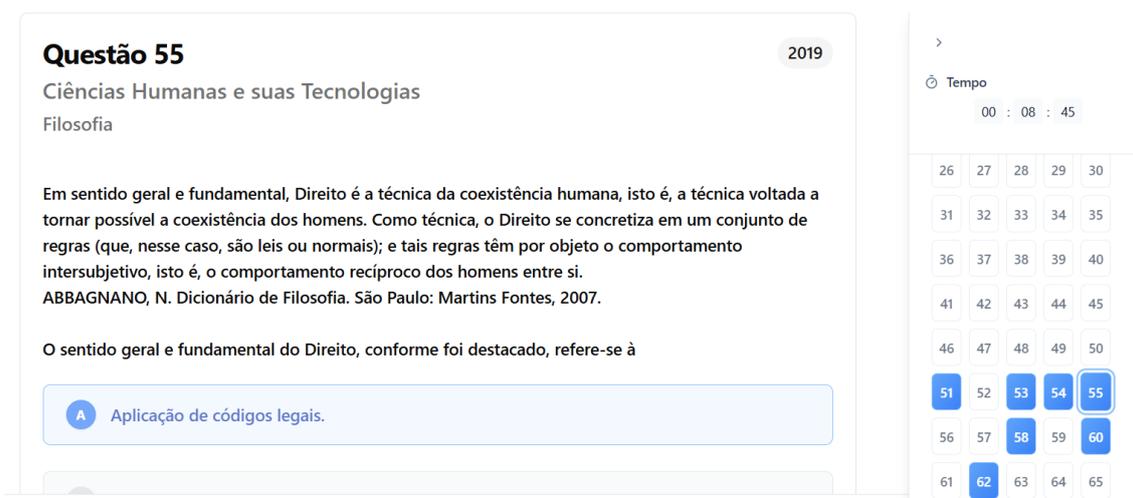


Figura 4.7: Tela de criação de simulado

Quando o usuário responde uma questão, a *service* responsável pelo simulado é acionada imediatamente para registrar essa resposta no banco de dados. Esse procedimento garante a persistência do progresso, mesmo em casos de desconexão ou fechamento inesperado do navegador. Enquanto o simulado estiver com o *status* “Fazendo”, o usuário pode reabrir a atividade a qualquer momento, responder novas questões ou modificar respostas quantas vezes desejar.

O tempo exibido na barra lateral é calculado com base na diferença entre a data de criação do simulado e o momento em que ele é aberto. Essa diferença fornece o tempo total, em segundos, desde a criação até a abertura, permitindo exibir uma contagem que representa há quanto tempo o simulado foi iniciado. Essa abordagem foi escolhida por sua simplicidade de implementação e por atender adequadamente aos objetivos de exibição do tempo de atividade do simulado.

Com o simulado em *status* de “Concluído”, o usuário pode visualizar as questões respondidas, conforme ilustrado na Figura 4.8, porém sem a possibilidade de alterar

suas respostas. As alternativas selecionadas corretamente são destacadas em verde. Caso a resposta esteja incorreta, a alternativa escolhida é exibida em vermelho, enquanto a correta é sinalizada em verde para que o usuário possa identificar o erro. Para as questões deixadas em branco, apenas a alternativa correta é indicada, destacada com uma borda verde.



Figura 4.8: Tela de criação de simulado

Além dos simulados objetivos, a plataforma também oferece dois tipos específicos voltados para a prática da redação: o simulado do tipo “Redação” e o simulado do tipo “ENEM”. O primeiro é focado exclusivamente na produção textual, enquanto o segundo simula a estrutura completa do exame. Em ambos os casos, o usuário pode escolher entre um tema oficial disponibilizado pelo INEP ou um tema gerado por inteligência artificial, ampliando as possibilidades de prática.

Ao optar por um tema gerado por inteligência artificial, a *service* realiza uma busca por temas previamente gerados que ainda não estejam associados ao usuário atual. Essa prática visa otimizar o uso de recursos, evitando chamadas desnecessárias ao modelo, uma vez que os temas não são exclusivos e podem ser reutilizados por diferentes usuários.

Caso não haja temas disponíveis, a *service* realiza uma requisição ao modelo *deepseek-chat*, utilizando temperatura um para promover maior variabilidade na geração. O *prompt* enviado solicita a criação de um tema de redação no estilo do

ENEM, exigindo que a resposta seja estruturada em um objeto JSON contendo dois campos: "theme", com o título da proposta, e "motivationalTexts", uma lista com até três textos motivadores relacionados ao tema.

Após a criação da redação, o usuário é redirecionado para o simulado correspondente, que contém a proposta gerada. Na nova tela, são exibidas inicialmente as instruções para a realização da redação, seguidas pela área com o tema e os textos motivadores, conforme ilustrado na Figura 4.9. O usuário permanece nessa interface durante o tempo definido pelas regras de negócio descritas na Tabela 3.2.



Figura 4.9: Tela de redação

Quando o usuário finaliza a redação, ele é redirecionado para um formulário de submissão de redação. Após o envio, a *service* responsável inicia automaticamente o processo de correção da redação, e o usuário é redirecionado para a tela de lista de simulados, onde aguarda a mudança de status do simulado de “Corrigindo” para “Concluído”.

O processo de correção é realizado por meio de seis requisições à API da DeepSeek. A primeira delas tem como objetivo verificar a anulabilidade da redação: o texto e o tema são enviados ao modelo, que deve analisar se a redação deve ser anulada de acordo com os critérios definidos pelo INEP. Caso o texto seja considerado anulável, o modelo retorna a justificativa correspondente e o processo é encerrado. Caso contrário, a correção prossegue com mais cinco requisições — uma para cada critério de avaliação do ENEM.

Inicialmente, foi testada a abordagem de correção por meio de uma única requisição abrangendo todos os critérios. No entanto, os resultados obtidos foram inconsistentes. A solução adotada foi realizar uma requisição separada para cada critério, o que, embora mais custoso, resultou em respostas mais coerentes e alinhadas com os padrões esperados. Cada requisição retorna uma nota de 0 a 200 e uma justificativa para a pontuação atribuída.

Ao final do processo, a *service* responsável compila os resultados, soma as notas dos critérios e atualiza o status do simulado para “Concluído”. A Figura 4.10 apresenta a tela de visualização dos resultados da redação.

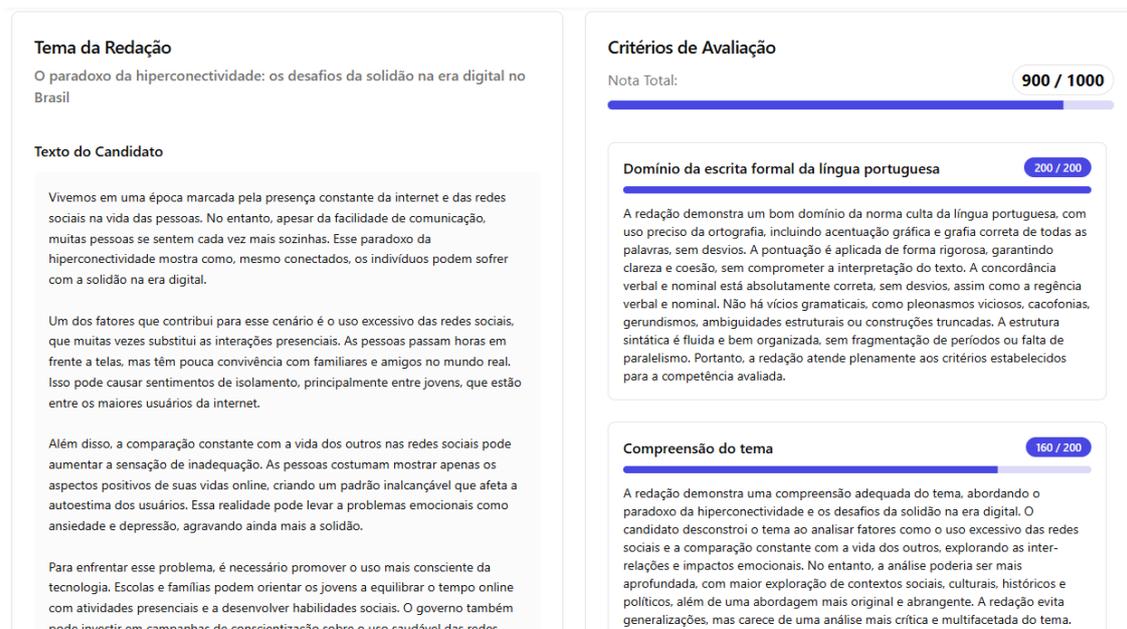


Figura 4.10: Tela de resultado da redação

Na tela “Meus Simulados”, são listados todos os simulados já realizados pelo

usuário, apresentando informações relevantes como o tipo e os subtipos selecionados durante a criação. Dependendo da natureza do simulado, também são exibidos parâmetros adicionais, como o tema da redação, a quantidade de questões acertadas, a nota obtida na correção da redação, entre outros. Além disso, a tela informa o status de cada simulado, indicando se ele está em andamento, em correção ou concluído. A Figura 4.11 ilustra a interface dessa tela.

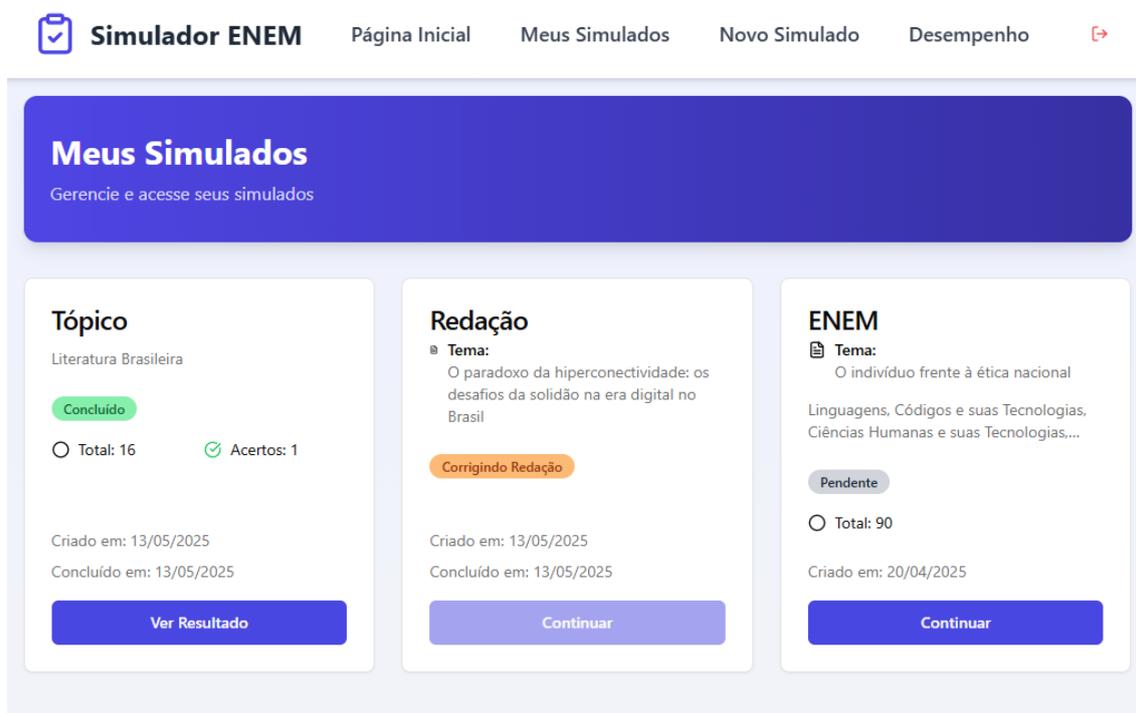


Figura 4.11: Tela Meus simulados

4.2.6 Tela de Desempenho

A Tela de Desempenho foi desenvolvida com o objetivo de oferecer ao usuário uma visão abrangente e detalhada sobre seu progresso acadêmico na plataforma. Essa interface, implementada com React.js e utilizando a biblioteca de visualização Recharts, apresenta uma série de métricas organizadas em gráficos e componentes visuais, facilitando a compreensão e o acompanhamento contínuo da evolução do estudante.

Ao acessar essa área, o usuário visualiza inicialmente um painel de resumo contendo estatísticas gerais, como o número total de simulados realizados, quantidade

de questões respondidas, média individual e taxa de acertos. Esses dados são exibidos de forma concisa e acompanhados por ícones e indicadores de variação percentual, permitindo a comparação entre o desempenho do usuário e a média geral da plataforma. A Figura 4.12 apresenta o painel principal da tela de desempenho com os cartões informativos.

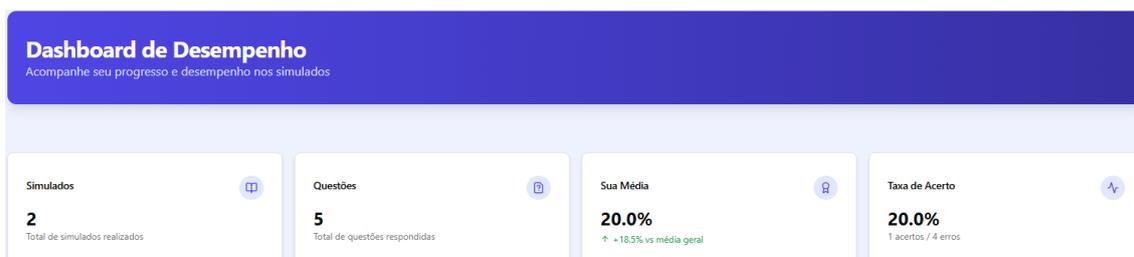


Figura 4.12: Painel com estatísticas gerais de desempenho do usuário

A interface é dividida em abas temáticas: Visão Geral, Desempenho, Categorias e Histórico. Na aba Visão Geral, são apresentados gráficos de barras que ilustram o desempenho por simulado, segmentando os acertos e erros de cada tentativa. A Figura 4.13 mostra a visualização dos acertos e erros dos simulados.

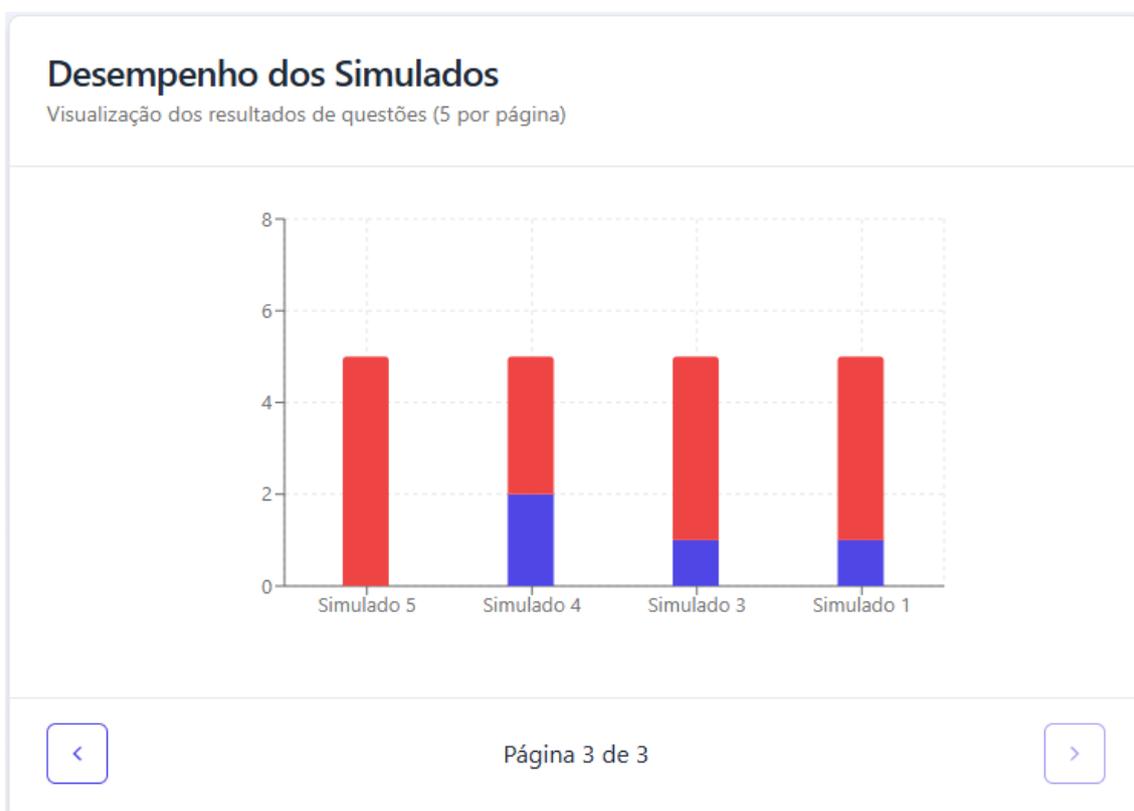


Figura 4.13: Gráfico de barras com desempenho por simulado

Para os simulados do tipo redação, as notas atribuídas são representadas por meio de um gráfico exclusivo, com eixo vertical ajustado para o intervalo de 0 a 1000 pontos, conforme os critérios do ENEM. Esse gráfico é ilustrado na Figura 4.14.

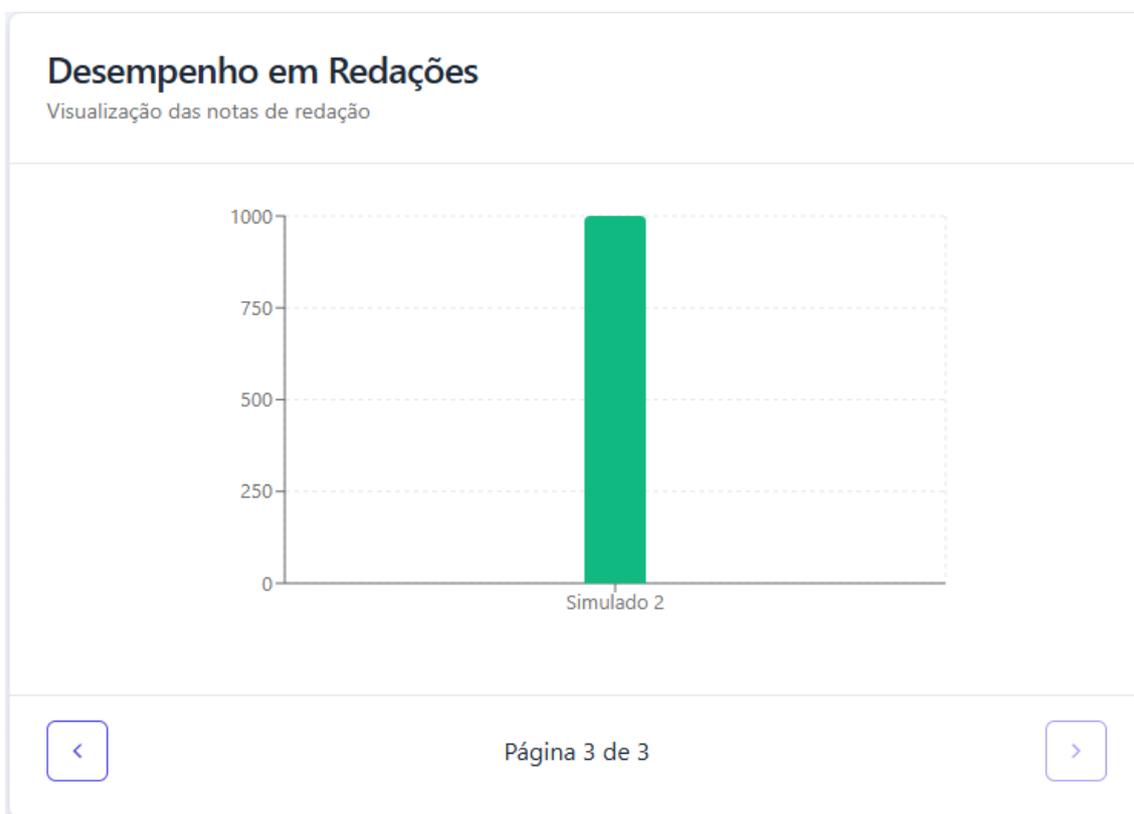


Figura 4.14: Gráfico com as notas obtidas nas redações

Outro recurso importante é a análise de afinidade por disciplina, exibida através de um gráfico radar que mostra as áreas de maior domínio do usuário com base na média de acertos por área do conhecimento. Essa informação é essencial para o autodiagnóstico e definição de estratégias de estudo. A Figura 4.15 demonstra essa visualização.



Figura 4.15: Gráfico radar com a afinidade por disciplina

Além disso, a aba também apresenta um gráfico de pizza com as cinco categorias com menor desempenho de acordo com a Figura 4.16, permitindo uma análise comparativa rápida. Essas visualizações são acompanhadas de uma legenda personalizada, que inclui percentual de acertos e quantidade de questões. Com base nessas informações, o sistema gera automaticamente recomendações de estudo de acordo com a Figura 4.17 na aba de Categorias, sugerindo que o usuário dedique mais atenção às áreas de menor rendimento.

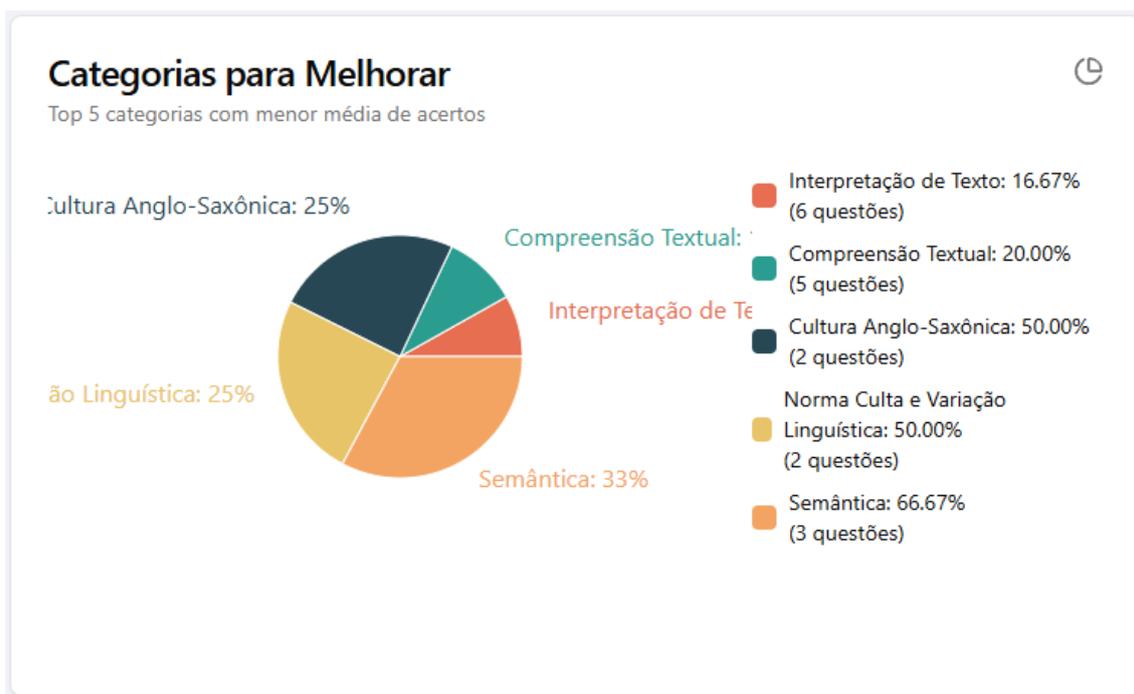


Figura 4.16: Categorias com menor média de acertos



Figura 4.17: Recomendações de estudo com base no desempenho

E por fim, é possível visualizar todos os simulados realizados, com informações detalhadas como data, status, quantidade de acertos e erros, taxa de acerto e nota da redação (quando aplicável). A navegação é assistida por ordenação cronológica, destacando os simulados mais recentes. A Figura 4.18 exibe a interface dessa aba.

Simulados Recentes
Últimos simulados realizados

Simulado	Data	Status	Acertos	Erros	Taxa
Simulado 1	15/05/2025	Concluído	1	4	20.0%
Simulado 2	11/05/2025	Concluído	Nota da Redação: 1000/1000		
Simulado 3	11/05/2025	Concluído	1	4	20.0%
Simulado 4	15/05/2025	Concluído	2	3	40.0%
Simulado 5	15/05/2025	Concluído	0	5	0.0%

[Ver todos os simulados](#)

Figura 4.18: Histórico completo de simulados realizados pelo usuário

Capítulo 5

Conclusão

Este capítulo apresenta as considerações finais acerca do desenvolvimento do SimuLab, destacando os principais resultados alcançados, como a geração de simulados personalizados, a correção automatizada de redações e o fornecimento de *feedback* detalhado ao usuário. Além disso, são discutidas as principais limitações identificadas durante o projeto e possíveis direções para aprimoramentos e expansões futuras da plataforma.

5.1 Considerações finais

O presente trabalho documentou o processo de concepção, desenvolvimento e implementação do SimuLab, uma plataforma digital voltada à preparação para o ENEM, cuja proposta central é integrar simulados personalizados, correção automatizada de redações e um *dashboard* analítico de desempenho.

Os resultados obtidos ao longo do projeto demonstram que a plataforma oferece aos estudantes uma experiência de estudo mais estratégica, autônoma e orientada por dados. Por meio das funcionalidades desenvolvidas, os usuários podem identificar suas principais dificuldades, receber *feedback* detalhado sobre seu desempenho e, a partir dessas informações, reestruturar seu plano de estudos de forma mais direcionada e eficiente.

A escolha por tecnologias modernas como *Next.js*, *React* e *Prisma*, combinada à integração com LLMs, como o GPT e o DeepSeek, proporcionou ao sistema escalabilidade e capacidade de processamento avançado. Essa infraestrutura tecnológica viabilizou funcionalidades como a categorização semântica de questões e a correção detalhada de redações, com geração automática de notas e justificativas individualizadas por critério avaliativo, em conformidade com os parâmetros estabelecidos pelo INEP.

5.2 Limitações e trabalhos futuros

Apesar dos avanços alcançados, o *SimuLab* ainda apresenta limitações que devem ser levadas em consideração nas próximas etapas de desenvolvimento. Uma das principais diz respeito à ausência da implementação da TRI para a classificação dos níveis de dificuldade das questões. A adoção da TRI exigiria um volume significativo de dados de interação dos usuários para permitir a calibragem estatística apropriada, o que, devido às restrições de escopo e tempo deste trabalho, não foi viável. A incorporação desse modelo representaria um salto qualitativo na precisão da categorização das questões, tornando o processo de seleção de itens mais adaptativo e sensível ao perfil de desempenho de cada estudante.

Outra limitação relevante está relacionada ao escopo atual do acervo de questões. No momento, o *SimuLab* contempla exclusivamente as provas oficiais do ENEM, abrangendo o período de 2009 a 2023. Embora esse conjunto represente uma base sólida e atualizada anualmente com as cento e oitenta questões liberadas pelo INEP, ele permanece limitado a itens já aplicados. Para ampliar a diversidade, atualidade e representatividade do banco de questões, propõe-se a futura implementação de um módulo de geração automática de itens. Tal módulo poderia empregar modelos de inteligência artificial treinados com base nas competências, habilidades e estilo linguístico característicos do ENEM, permitindo a criação de questões inéditas e personalizadas, o que enriqueceria substancialmente o repertório disponível na plataforma.

Além disso, há espaço para aprimoramentos significativos nas funcionalidades de *feedback* ao usuário. Atualmente, a plataforma oferece correção detalhada apenas para as redações, por meio de LLMs capazes de atribuir notas e fornecer justificativas individualizadas conforme os critérios do exame. No entanto, um avanço desejável seria estender esse nível de detalhamento às questões objetivas, fornecendo explicações contextualizadas sobre cada alternativa correta ou incorreta, além de orientações conceituais que auxiliem o estudante a compreender os erros cometidos e a consolidar o aprendizado.

A ausência de um sistema de recomendação também constitui uma limitação no que diz respeito à personalização da experiência de aprendizagem. A capacidade de analisar o histórico de desempenho dos usuários e sugerir automaticamente tópicos, questões ou trilhas de estudo com base em padrões recorrentes de acerto e erro pode ser explorada. A implementação de algoritmos de recomendação, baseados em técnicas de aprendizado de máquina, integrados ao perfil comportamental de cada estudante, representa um caminho promissor para tornar o processo de preparação mais direcionado e eficiente.

Outra vertente interessante para trabalhos futuros envolve o aprofundamento na análise do perfil dos estudantes. A coleta de informações adicionais, como o curso desejado ou a instituição de ensino superior almejada, poderia permitir a geração de simulados mais personalizados, alinhados às exigências específicas de determinadas carreiras ou áreas do conhecimento. Além disso, o desenvolvimento de ferramentas de *benchmarking* — como estatísticas comparativas entre usuários com metas similares, indicadores de desempenho por área, tempo médio de resolução e progressão ao longo do tempo — contribuiria para que os estudantes tenham parâmetros mais objetivos e contextualizados para medir seu progresso.

A implementação dessas melhorias consolidaria o SimuLab não apenas como uma plataforma de realização de simulados, mas como uma solução educacional completa, inteligente e inclusiva. Ao promover uma preparação mais eficaz, personalizada e adaptável às realidades individuais dos estudantes brasileiros, o sistema poderá contribuir de forma mais significativa para o fortalecimento do acesso ao ensino

superior no país.

Referências

ABBATE, J. *Inventing the Internet*. [S.l.]: MIT Press, 2000.

ANDRADE, C. M. L.; TAVARES, M. G. Nível socioeconômico e desempenho no enem: uma análise multinível. *Em Aberto*, 2021. Disponível em: <<https://emaberto.inep.gov.br/ojs3/index.php/emaberto/article/view/5002>>.

BARTHOLO, T. et al. *Oportunidades educacionais de estudantes concluintes do Ensino Médio: um estudo do Enem entre 2013 e 2021*. [S.l.], 2023. Disponível em: <<https://www.institutounibanco.org.br/conteudo/desigualdades-socioeconomicas-na-participacao-no-enem-dobraram-durante-pandemia/>>.

BERNERS-LEE, T. *World Wide Web: A Proposal*. 1991. CERN. Disponível em: <<http://info.cern.ch>>.

CERF, V. G.; KAHN, R. E. A protocol for packet network intercommunication. *IEEE Transactions on Communications*, v. 22, n. 5, p. 637–648, 1974. Disponível em: <<https://ieeexplore.ieee.org/document/1092259>>.

CHOMSKY, N. *Aspects of the Theory of Syntax*. Cambridge, MA: MIT Press, 1965.

CONSORTIUM, W. W. W. *HTML5: A vocabulary and associated APIs for HTML and XHTML*. [S.l.], 2014. W3C Recommendation. Disponível em: <<https://www.w3.org/TR/html5/>>.

DESCOMPLICA. *Simulado ENEM Descomplica*. 2025. Plataforma Online. Acesso em: 23 mar. 2025. Disponível em: <<https://simulado.descomplica.com.br>>.

FLANAGAN, D. *JavaScript: The Definitive Guide*. 7th. ed. [S.l.]: O'Reilly Media, 2020.

INEP. *Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira*. 2025. Acesso em: 23 mar. 2025. Disponível em: <<https://www.gov.br/inep>>.

KLEINROCK, L. *Information Flow in Large Communication Nets*. [S.l.], 1961. 1–35 p. Disponível em: <<https://www.lk.cs.ucla.edu/data/files/Kleinrock/Information%20Flow%20in%20Large%20Communication%20Nets.pdf>>.

KUROSE, J. F.; ROSS, K. W. *Computer Networking: A Top-Down Approach*. [S.l.]: Pearson, 2017.

- LICKLIDER, J. Man-computer symbiosis. *IRE Transactions on Human Factors in Electronics*, HFE-1, p. 4–11, 1960.
- LICKLIDER, J. *Members and Affiliates of the Intergalactic Computer Network*. 1963. Memorando da ARPA.
- Meta Platforms, Inc. *React – A JavaScript Library for Building User Interfaces*. 2025. <<https://reactjs.org/>>.
- MOCKAPETRIS, P. *Domain names - concepts and facilities*. RFC Editor, 1987. RFC 1034. Disponível em: <<https://www.rfc-editor.org/info/rfc1034>>.
- POSTEL, J. *Internet Protocol*. [S.l.], 1981. Disponível em: <<https://www.rfc-editor.org/rfc/rfc791>>.
- ROBERTS, L. G. *Toward a Cooperative Network of Time-Shared Computers*. [S.l.], 1966.
- ROEDIGER, H. L.; BUTLER, A. C. The critical role of retrieval practice in long-term retention. *Trends in Cognitive Sciences*, v. 15, n. 1, p. 20–27, 2011.
- SILVA, J. *Desenvolvimento de uma Aplicação Web para Simulados do ENEM*. 2023. Monografia - Universidade Federal de Ouro Preto. Disponível em: <https://www.monografias.ufop.br/bitstream/35400000/5525/6/MONOGRAFIA_DesenvolvimentoAplicaçãõWeb.pdf>.
- SOBRINHO, J. D. Avaliação ética e política em função da educação como direito público ou como mercadoria? *Educação & Sociedade*, 2004. Disponível em: <<https://www.scielo.br/j/es/a/mmDFy9Sk6vHzq7R4hJxWKNk/?lang=pt>>.
- SOCIETY, I. *A Brief History of the Internet*. 2017. Disponível em: <<https://www.internetsociety.org/internet/history-internet/brief-history-internet/>>.
- TANENBAUM, A. S.; WETHERALL, D. J. *Redes de Computadores*. 5. ed. [S.l.]: Pearson, 2011.
- YUNGER7. *enem-api*. 2024. <<https://github.com/yunger7/enem-api>>.
- ZHAO, W. X.; AL. et. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.