

Sistema Web para Mentoria Online

Vinícius Andrade Gonzato, Alex Machado Borges, George Rodrigo Souza Gonçalves

¹Instituto Federal Farroupilha – Campus Frederico Westphalen
Bacharelado em Ciência da Computação
Frederico Westphalen – RS – Brazil

vinicius.2022014759@aluno.iffar.edu.br

alex.borges@iffarroupilha.edu.br

george.goncalves@iffar.edu.br

Resumo. *A crescente demanda por preparação para concursos evidencia a fragmentação das soluções existentes, que exigem múltiplas plataformas para mentorias, correção de redações e pagamentos. Este trabalho apresenta um sistema web que centraliza o agendamento de sessões e o envio de redações para correção por mentores, ambos com pagamento automatizado via PIX (Mercado Pago). Desenvolvido com Next.js, TypeScript, Prisma e PostgreSQL, o sistema oferece interfaces responsivas e seguras. Os resultados demonstram estabilidade, facilidade de uso e integração completa dos fluxos de mentoria, correção de redações e pagamentos, ampliando as possibilidades de acompanhamento individualizado para estudantes.*

Abstract. *The growing demand for exam preparation highlights the fragmentation of existing solutions, which often require multiple platforms for mentoring, essay correction, and payments. This work presents a web system that centralizes session scheduling and essay submission for mentor correction, both with automated PIX payment integration via Mercado Pago. Developed using Next.js, TypeScript, Prisma, and PostgreSQL, the platform provides secure and responsive interfaces. Results indicate stability, ease of use, and full integration of mentoring, essay correction, and payment workflows, expanding individualized support options for students.*

1. Introdução

A educação tem sido significativamente transformada pelo avanço das tecnologias digitais, especialmente no campo do ensino a distância e da mentoria online, que têm criado novas possibilidades para a preparação de candidatos a concursos públicos. No Brasil, esse setor apresenta relevância econômica expressiva: segundo o portal Revide, o mercado de concursos públicos deve movimentar cerca de cinco bilhões de reais em 2025, atraindo aproximadamente nove milhões de candidatos [Revide 2025]. Essa ampla procura por aprovação impulsiona a demanda por orientação especializada e personalizada, como *feedback* aprofundado em redações e acompanhamento com mentores.

Apesar da existência de diversas plataformas digitais no mercado, como Online [Gran Cursos Online 2025], Qconcursos [Qconcursos 2025] e Estratégia Concursos

[Estratégia Concursos 2025], que oferecem desde cursos preparatórios (conjuntos de materiais didáticos padronizados e pré-gravados) até mentorias individuais (acompanhamento personalizado e direto com um profissional qualificado). Muitas dessas soluções focam em modelos de conteúdo massificado ou não conseguem integrar, de forma eficiente e unificada, todas as funcionalidades necessárias para um acompanhamento contínuo e estruturado. Por exemplo, notou-se que mesmo plataformas que oferecem mentoria, como o Grancursos Online [Gran Cursos Online 2025], não garantem uma funcionalidade completa e unificada para o envio e a correção paga de redações. Essa falha se manifesta na falta de integração do sistema de correção de redações pagas diretamente com o mesmo fluxo de agendamento de sessões que atende às necessidades de acompanhamento especializado dos estudantes. Consequentemente, a falta de organização nos estudos e a dificuldade em conseguir acompanhamento individualizado e avaliação minuciosa, em particular na área de redação, tornam-se obstáculos comuns e amplos para muitos estudantes, além dos problemas de integração de serviços.

Nesse cenário, este trabalho apresenta o desenvolvimento e implementação de um Sistema Web para Mentoria Online, direcionado a estudantes de concursos públicos. O projeto busca oferecer uma solução tecnológica robusta que supra as necessidades do estudante e, simultaneamente, atenda a uma demanda de mercado. O desenvolvimento deste sistema foi motivado pela busca em responder à seguinte questão de pesquisa, que guia todo o trabalho:

Como uma plataforma web moderna pode, simultaneamente, aprimorar a preparação de estudantes para concursos públicos e proporcionar uma fonte de renda viável para profissionais da educação através de uma solução tecnológica robusta e escalável?

O objetivo geral deste trabalho foi, portanto, desenvolver e implementar um sistema de mentorias online que otimize a preparação de estudantes e ofereça um modelo de negócio sustentável para mentores. Para isso, a plataforma foi concebida para fornecer interfaces distintas para estudantes e mentores, com fluxo completo de agendamento e serviços de correção de redação com monetização integrada, demonstrando a aplicação prática de uma arquitetura moderna para resolver problemas educacionais e econômicos.

2. Fundamentação Teórica e Análise de Soluções Atuais

A presente seção estabelece a base teórica e o contexto mercadológico que justificam o desenvolvimento do Sistema Web para Mentoria Online. Para tal, é realizada uma revisão sobre o cenário da educação digital no Brasil, a importância do acompanhamento individualizado na preparação para concursos e uma análise crítica das soluções já existentes.

2.1. O Cenário da Educação Digital e a Preparação para Concursos no Brasil

A transformação digital impulsionou o segmento de Ensino à Distância (EAD), tornando-se um vetor essencial na preparação para concursos públicos. No Brasil, este é um mercado robusto e em expansão. As estimativas indicam que, em 2025, o setor deve movimentar cerca de cinco bilhões de reais e atrair aproximadamente nove milhões de candidatos, enfatizando-se o potencial de usuários que demandam de recursos especializados de preparação para concursos [Revide 2025].

Neste cenário de intensa concorrência, o acesso a materiais de estudo não é mais o único diferencial; a chave para a aprovação reside na capacidade do candidato de organizar seu estudo e obter orientação individualizada. A massa de dados e informações disponíveis exige que o estudante adote estratégias de acompanhamento que permitam identificar pontos fracos, refinar a argumentação e aprimorar a performance de forma minuciosa.

2.2. A Organização do Estudo e a Crítica da Redação

Um dos maiores desafios enfrentados pelos candidatos é a dificuldade em manter a organização dos estudos de forma autônoma e obter *feedback* de qualidade, especialmente em áreas subjetivas e classificatórias como a redação [Instituto Consciência GO 2025]. A literatura aponta que a falta de organização das ideias e a dificuldade em conseguir avaliação minuciosa são obstáculos comuns para esses estudantes de concursos públicos [Instituto Adriana Figueiredo 2025].

A prova discursiva de redação exige prática contínua e um retorno detalhado. No entanto, muitas soluções de EAD falham ao fornecer um ciclo de *feedback* eficiente, onde o estudante possa enviar seu texto, receber uma correção minuciosa e ter um canal direto com o mentor, tudo isso dentro de um fluxo de serviço bem definido e rastreável. A ausência de ferramentas que unifiquem agendamento, envio de material e *feedback*, contribui para a limitação da performance e gera frustração no processo de preparação.

2.3. Análise Crítica das Soluções Atuais

Visando identificar o panorama de plataformas de auxílio para concurseiros, investigou-se as ferramentas existentes que auxiliam esses usuários (estudantes de concursos públicos) na gestão de seus estudos, de modo a identificar as lacunas existentes de forma a propor um novo sistema concorrente a estes e que, ao mesmo tempo, melhore a experiência do estudante. As soluções atuais podem ser categorizadas em dois grupos principais, com focos de negócio distintos: plataformas de conteúdo massificado e plataformas de mentoria e acompanhamento de estudos.

2.3.1. Plataformas de Conteúdo Massificado

Plataformas como a QConcursos [Qconcursos 2025] e a Estratégia Concursos [Estratégia Concursos 2025] concentram seus esforços na oferta de vastas bibliotecas de questões, simulados e aulas gravadas. Seu modelo de negócio é baseado principalmente na venda de assinaturas para acesso ao acervo de conteúdo. O ponto forte dessas plataformas reside na relação custo-benefício para a aquisição de conteúdo objetivo, como as milhares de questões e o suporte teórico abrangente. Embora sejam eficazes na entrega tanto de material teórico e como de exercícios práticos com questões objetivas, o serviço de mentoria e acompanhamento individualizado é secundário. O modelo de escala inviabiliza a customização profunda. O foco na escala de conteúdo, por vezes, limita a profundidade e a frequência do *feedback* individual, que se torna genérico ou delegado a ferramentas automatizadas. O aluno paga por um acervo e não por um serviço de acompanhamento especializado.

2.3.2. Plataformas de Mentoria e Acompanhamento de Estudos

Outras empresas de serviços de *coaching* individual, como a Grancursos Online [Gran Cursos Online 2025], oferecem serviços de orientação personalizada e mentorias focadas em planos de estudo e acompanhamento de carreira. A principal vantagem desse modelo está na alta personalização e na adaptação das estratégias de estudo às necessidades e ao ritmo do aluno, sendo especialmente relevante para o planejamento estratégico de longo prazo e para a correção detalhada de redações.

Apesar desses benefícios, a rigidez do modelo de negócio constitui sua principal limitação. Em geral, essas plataformas vinculam o acesso à mentoria individual à contratação de planos recorrentes, frequentemente com custo elevado e duração mínima predefinida, como pacotes trimestrais ou semestrais. Esse formato restringe a flexibilidade do usuário e dificulta a contratação de serviços pontuais.

Além disso, em algumas soluções, como a plataforma Organiza Concursos [Organiza Concursos 2025], o acesso a serviços de assessoramento ou plantão de dúvidas ocorre por meio de processos manuais, como o envio de formulários ou contato via canais externos, como o WhatsApp. Nesses casos, não há um fluxo de pagamento direto e automatizado para a contratação imediata de sessões ou serviços específicos, o que compromete a agilidade e a escalabilidade da oferta.

Essa combinação de planos rígidos e processos manuais impede a contratação isolada de serviços pontuais, como sessões avulsas de mentoria ou a correção de uma única redação. Como consequência, o estudante é forçado a manter um vínculo contínuo e, muitas vezes, mais dispendioso do que sua real necessidade.

Do ponto de vista do profissional, a viabilidade da oferta de serviços personalizados, como mentorias individuais e correção de redações, está diretamente relacionada à eficiência dos mecanismos de monetização disponíveis. A ausência de um fluxo transacional automatizado, observada em soluções baseadas em contato manual ou planos prolongados, representa um obstáculo à sustentabilidade desse modelo de prestação de serviços.

Nesse contexto, a possibilidade de contratação avulsa com pagamento imediato configura-se como um elemento central deste trabalho. Tal abordagem permite que o mentor organize, amplie e profissionalize sua atuação, garantindo maior previsibilidade financeira e redução de processos operacionais manuais.

Dessa forma, o sistema proposto neste TCC posiciona-se na intersecção de duas demandas complementares. De um lado, atende à busca do estudante por flexibilidade e qualidade no *feedback* de serviços especializados, como a correção de redações avulsas. De outro, oferece ao mentor uma ferramenta que automatiza os processos de agendamento, correção de redações e recebimento, contribuindo para a escalabilidade e a viabilidade financeira de suas atividades.

Diferentemente dessas abordagens, a proposta deste trabalho visa oferecer um sistema mais flexível, no qual o usuário pode contratar apenas o serviço desejado — seja o envio de uma redação ou uma sessão avulsa de mentoria — sem a necessidade de assinatura ou vínculo contínuo com um mentor.

3. Sistema Web para Mentoria Online: Proposta e Requisitos

Esta seção detalha a Proposta do Sistema Web para Mentoria Online e apresenta a modelagem necessária para sua implementação, incluindo os requisitos funcionais e não funcionais e os diagramas de Linguagem de Modelagem Unificada (do inglês *Unified Modeling Language* - UML) que guiam o desenvolvimento.

3.1. Requisitos do Sistema

A definição dos requisitos foi baseada na análise de mercado e nas lacunas identificadas nas soluções existentes, priorizando a flexibilidade transacional e o acompanhamento especializado.

3.1.1. Requisitos Funcionais (RFs)

Os requisitos funcionais definem as funcionalidades que o sistema deve executar para atender aos usuários (Estudante e Mentor):

- **RF01 - Autenticação de Usuários:** O sistema deve permitir o gerenciamento do acesso dos usuários por meio de:
 - **RF01.1 - Login de Usuário:** O sistema deve permitir que usuários realizem login utilizando credenciais previamente cadastradas.
 - **RF01.2 - Cadastro de Estudante:** O sistema deve permitir o cadastro de usuários do tipo *Estudante*.
 - **RF01.3 - Cadastro de Mentor:** O sistema deve permitir o cadastro de usuários do tipo *Mentor*.
 - **RF01.4 - Logout de Usuário:** O sistema deve permitir que usuários autenticados encerrem sua sessão de forma segura.
- **RF02 - Gestão de Perfil do Mentor:** O mentor deve conseguir cadastrar e atualizar sua biografia.
- **RF03 - Gestão de Disponibilidade:** O mentor deve conseguir definir seus horários disponíveis para agendamento, com controle de fuso horário.
- **RF04 - Visualização de Mentores:** O estudante deve conseguir visualizar uma lista de mentores disponíveis.
- **RF05 - Agendamento de Sessão:** O estudante deve conseguir agendar um horário disponível com o mentor de sua escolha.
- **RF06 - Integração de Pagamento PIX:** O sistema deve integrar-se com um *gateway* de pagamento para processar o pagamento do agendamento via PIX, gerando um *QR Code* e código *Copia e Cola*.
- **RF07 - Confirmação Automática de Pagamento:** O sistema deve receber a notificação de pagamento (via *webhook*) e atualizar automaticamente o status do serviço para "Confirmado".
- **RF08 - Envio de Redações:** O estudante deve conseguir enviar redações por meio do sistema para correção pelo mentor.
- **RF09 - Correção de Redações e Feedback:** O mentor deve conseguir corrigir as redações enviadas e fornecer feedback ao estudante.
- **RF10 - Gestão de Agendamentos (Estudante):** O estudante deve conseguir visualizar seus agendamentos.
- **RF11 - Gestão de Agendamentos (Mentor):** O mentor deve conseguir visualizar sua agenda de sessões confirmadas.

3.1.2. Requisitos Não Funcionais (RNFs)

Os requisitos não funcionais definem critérios de qualidade, restrições tecnológicas e aspectos de usabilidade:

- **RNF01 - Usabilidade e Responsividade:** A interface do sistema deve ser intuitiva, moderna e totalmente responsiva, garantindo usabilidade em dispositivos móveis e *desktop* (*Web Responsive*).
- **RNF02 - Segurança (Autenticação):** A autenticação deve ser segura, utilizando *tokens* e criptografia de senhas.
- **RNF03 - Segurança (Transacional):** As comunicações com o *gateway* de pagamento devem ser realizadas via HTTPS e o *webhook* deve ser validado para garantir a integridade dos dados transacionais.
- **RNF04 - Performance:** A aplicação deve utilizar otimizações de *caching* e renderização de servidor.

3.2. Modelagem do Sistema

A modelagem é apresentada utilizando a UML, detalhando a estrutura estática e o comportamento dinâmico do sistema.

3.2.1. Diagrama de Casos de Uso

O Diagrama de Casos de Uso, representado na Figura 1, descreve de forma estrutural como os diferentes atores interagem com o Sistema de Mentoria Online e quais funcionalidades essenciais são oferecidas pela aplicação. Ele é organizado em cinco pacotes principais (Autenticação, Gestão de Estudante, Gestão de Mentor, Gestão de Redações e Processamento de Pagamento), que agrupam casos de uso relacionados entre si e facilitam a visualização dos módulos do sistema.

Na Figura 1, aparece o ator "Estudante", que representa o usuário que busca mentoria e acompanhamento individualizado. Esse ator possui acesso a três grupos de funcionalidades: (1) na área de Autenticação, ele pode realizar login e cadastro; (2) na Gestão de Estudante, pode visualizar os mentores cadastrados, agendar sessões com o profissional escolhido, consultar seus próprios agendamentos — verificando sessões futuras ou já concluídas — e enviar redações para correção por mentores, mediante confirmação do pagamento; (3) por fim, o estudante interage de forma indireta com o módulo de Processamento de Pagamento, necessário para a contratação tanto de sessões de mentoria quanto do serviço de correção de redações.

Na Figura 1, também temos o ator "Mentor", que representa o profissional responsável por oferecer as sessões de orientação e avaliação. Assim como o estudante, ele interage com a área de Autenticação, mas possui um conjunto adicional de operações vinculadas à Gestão de Mentor e à Gestão de Redações. Entre essas funcionalidades, o diagrama evidencia que o mentor pode cadastrar seu perfil, atualizar sua biografia, definir horários disponíveis para atendimento, visualizar sua própria agenda de sessões agendadas pelos estudantes e corrigir redações enviadas, fornecendo feedback individualizado aos estudantes após a liberação do serviço pelo sistema.

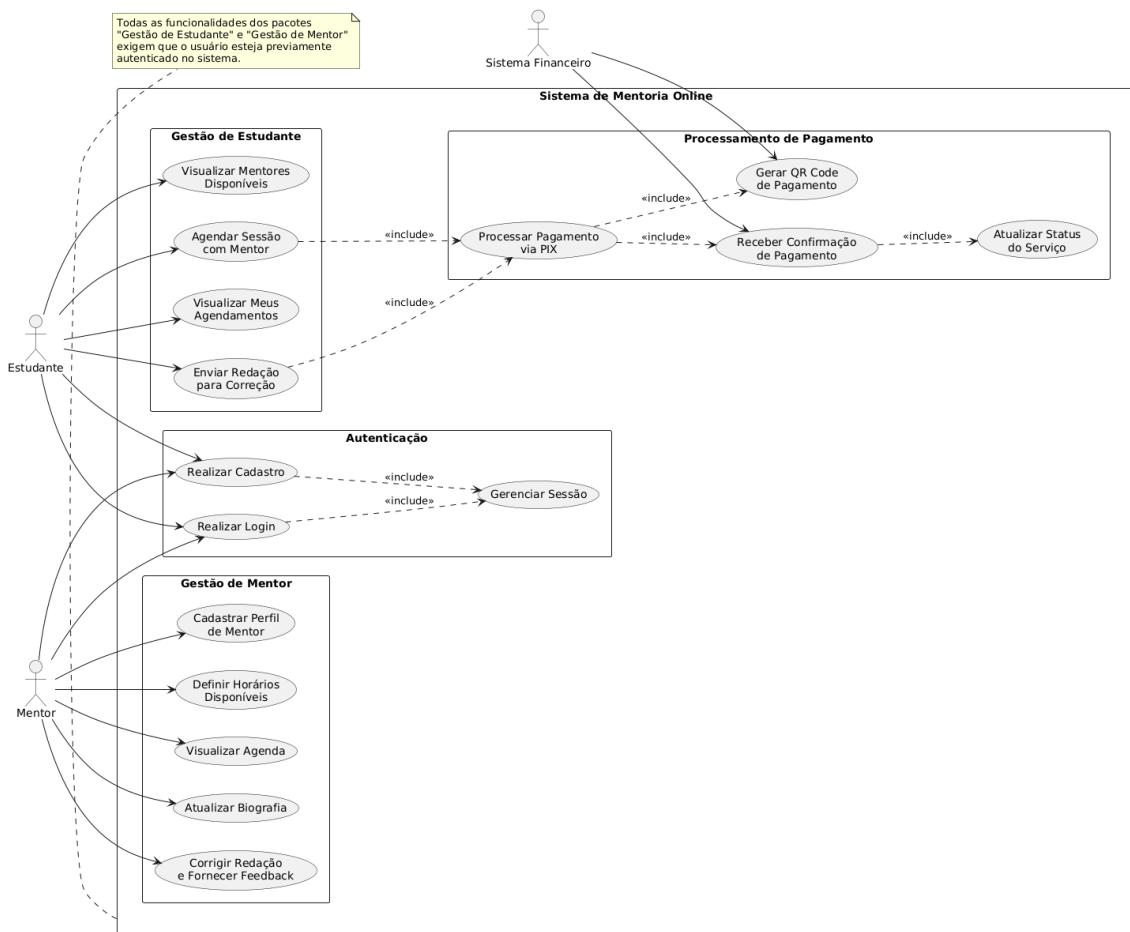


Figura 1. Diagrama de Casos de Uso do Sistema Web para Mentoria Online.

O terceiro ator presente na Figura 1 é o "Sistema Financeiro", que aparece separado dos demais atores por representar um serviço externo responsável pelo processamento dos pagamentos realizados via PIX. Esse ator interage exclusivamente com os casos de uso relacionados ao fluxo financeiro, incluindo a geração do QR Code de pagamento, o recebimento da confirmação financeira e, a partir dessa confirmação, a atualização automática do status do serviço contratado, seja ele uma sessão de mentoria ou a correção de uma redação. A presença desse ator deixa claro que o sistema depende de um processo assíncrono — o envio de uma notificação (*webhook*) pelo serviço financeiro — para confirmar transações concluídas com sucesso.

Dentro do pacote de Processamento de Pagamento, o diagrama evidencia uma sequência de dependências entre os casos de uso. Tanto o ato de agendar uma sessão quanto o envio de uma redação incluem a necessidade de processar o pagamento; a operação de processamento, por sua vez, inclui a geração do QR Code e aguarda a confirmação do pagamento para, somente após esse retorno, acionar a atualização do status do serviço. Essa estrutura demonstra que o sistema não apenas registra solicitações de mentoria ou correção de redações, mas também integra o fluxo financeiro de forma segura e automatizada.

Por fim, o diagrama evidencia o componente interno *Gerenciar Sessão*, associ-

ado aos processos de autenticação do sistema. Além disso, conforme indicado por nota explicativa no diagrama, todas as funcionalidades pertencentes à Gestão de Estudante e à Gestão de Mentor pressupõem que o usuário esteja autenticado, ou seja, possua uma sessão válida ativa. Essa modelagem reforça o controle de acesso às funcionalidades do sistema sem sobrecarregar visualmente o diagrama com múltiplas relações de inclusão.

Dessa forma, o Diagrama de Casos de Uso não apenas identifica quem interage com o sistema, mas também organiza logicamente as operações de cada ator, mostrando como as funcionalidades de mentoria, envio e correção de redações e processamento de pagamentos se relacionam entre si dentro da aplicação.

3.2.2. Diagrama Entidade-Relacionamento (DER)

O Diagrama Entidade-Relacionamento (DER), apresentado na Figura 2, representa a estrutura lógica dos dados do Sistema de Mentoria Online e descreve como as entidades se relacionam entre si. O DER enfatiza a organização das informações em tabelas, suas chaves primárias e estrangeiras, bem como as cardinalidades que definem as regras do modelo de dados adotado pela aplicação.

A entidade *User* constitui o núcleo do sistema, armazenando informações essenciais de autenticação e identificação dos usuários. A partir dela, estabelece-se um relacionamento opcional com a entidade *Mentor*, indicando que um usuário pode ou não possuir um perfil de mentor. Esse modelo permite que o mesmo usuário atue exclusivamente como estudante ou acumule também o papel de mentor na plataforma. O mentor está associado às matérias em que atua, bem como aos horários disponibilizados para atendimento, registrados na entidade *Schedule*.

O processo de mentoria síncrona é representado pela entidade *Appointment*, que centraliza múltiplas relações: cada agendamento envolve um estudante (usuário), um mentor, uma matéria específica e um horário previamente definido. Essa entidade permite compreender como as sessões de mentoria são estruturadas e vinculadas aos demais componentes do sistema.

Além das sessões de mentoria, o sistema também contempla o envio e a correção de redações, representados pela entidade *Essay*. Cada redação está associada a um estudante, responsável pelo envio do texto, e a um mentor, encarregado da correção e do fornecimento de feedback. A entidade armazena informações como o conteúdo da redação, seu status no fluxo de correção, nota atribuída e comentários do mentor, possibilitando o acompanhamento completo do processo avaliativo.

O fluxo financeiro do sistema é modelado pela entidade *Payment*, que representa as transações realizadas via PIX por meio de um serviço externo de pagamento. Diferentemente de modelos restritos a um único tipo de serviço, a entidade *Payment* pode estar associada tanto a um agendamento de mentoria (*Appointment*) quanto a uma redação (*Essay*), permitindo reutilização do mesmo mecanismo de pagamento para diferentes funcionalidades do sistema. Essa abordagem reduz redundâncias e torna o modelo mais flexível e escalável.

As cardinalidades presentes no DER evidenciam as principais regras de negócio da aplicação: um usuário pode ou não possuir perfil de mentor; um mentor pode cadastrar

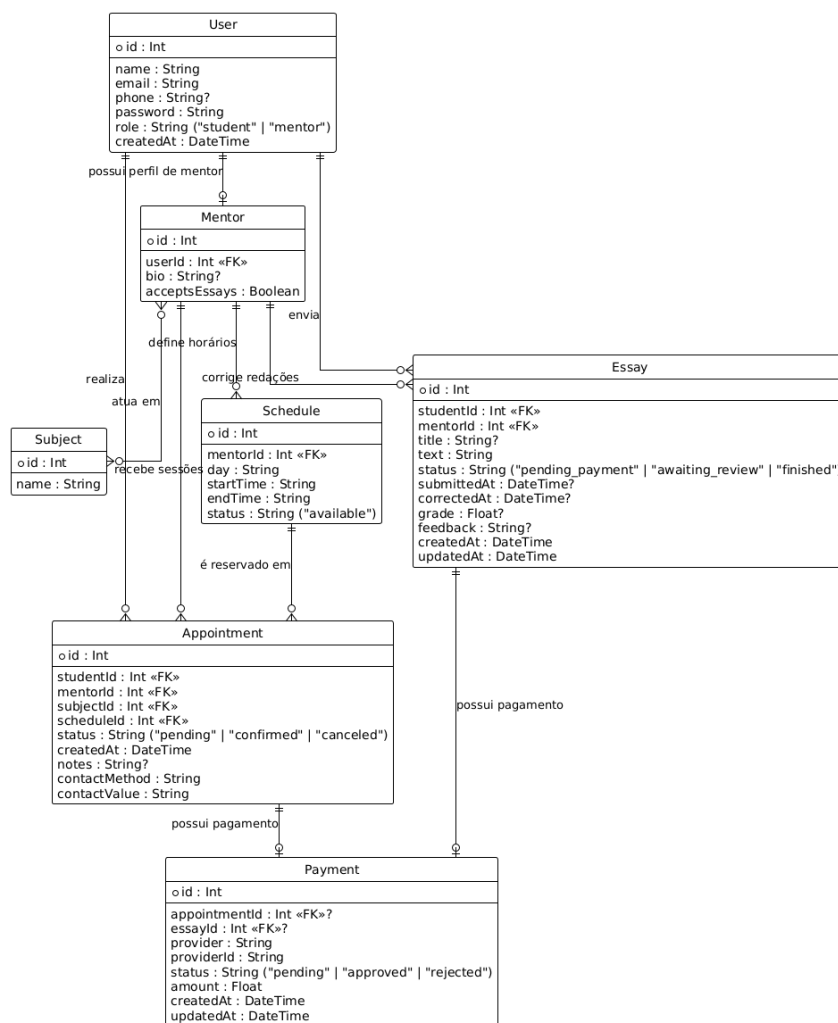


Figura 2. Diagrama de Entidade-Relacionamento.

diversos horários, receber múltiplos agendamentos e corrigir várias redações; um estudante pode realizar vários agendamentos e enviar múltiplas redações ao longo do tempo; cada agendamento está vinculado a um único estudante, um único mentor, uma matéria e um horário; e a relação entre mentores e matérias é do tipo muitos para muitos.

Dessa forma, o DER fornece uma visão clara e consistente da estrutura relacional do banco de dados, permitindo compreender como o sistema organiza suas informações, integra os fluxos de mentoria, correção de redações e pagamentos, e sustenta o funcionamento global da plataforma.

3.2.3. Diagramas de Sequência

Os diagramas de sequência a seguir detalham dois fluxos principais do sistema — o de Agendamento de Sessão com Pagamento PIX e o de Envio de Redação para Correção, ambos envolvendo um mecanismo de pagamento online. As figuras evidenciam a ordem temporal das interações entre os principais participantes: Estudante, Interface Web (UI), API, Banco de Dados (Prisma DB) e o serviço de pagamentos Mercado Pago. O fluxo é

modelado considerando todas as etapas importantes, desde a solicitação dos serviços até a confirmação automática do pagamento, incluindo o tratamento assíncrono por *webhook*.

A Figura 3 apresenta o fluxo para agendamento de sessões com mentores. O processo inicia quando o estudante solicita um agendamento através da interface web (UI), que envia os dados do pedido para a API do sistema. A API registra o agendamento no banco de dados (DB/Prisma) com status "pendente" e então solicita ao serviço Mercado Pago a criação de uma cobrança PIX. O Mercado Pago retorna o QR Code de pagamento, que é exibido ao estudante na interface web. Após a realização do pagamento, o Mercado Pago notifica o backend do sistema através de um *webhook*. Recebida e validada a notificação, a API atualiza automaticamente o status do agendamento para "pago" no banco de dados, e a confirmação é sinalizada ao estudante na interface.

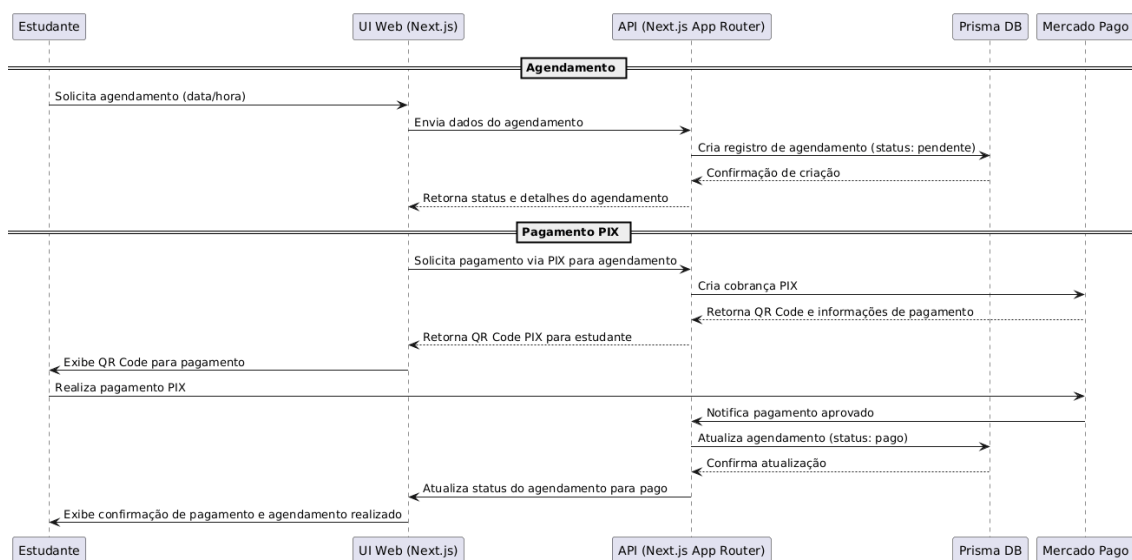


Figura 3. Diagrama de Sequência do fluxo de Agendamento de Sessão e Pagamento PIX.

A Figura 4 ilustra o fluxo referente ao envio de uma redação para correção. O estudante acessa a página de envio, seleciona o mentor e preenche o formulário com o título e o texto da redação. Ao enviar, a interface web encaminha os dados para a API, que registra a redação no banco de dados com status "pendente de pagamento" e, na sequência, solicita ao Mercado Pago a geração de um QR Code PIX para o serviço de correção. O QR Code retornado é apresentado ao estudante para que efetue o pagamento. Após este, o Mercado Pago envia uma notificação (*webhook*) à API, que então atualiza o status da redação para "paga", liberando-a para correção pelo mentor. A confirmação do sucesso é comunicada ao estudante na própria interface.

Em ambos os fluxos, destaca-se a utilização da automatização via *webhook* — que elimina a necessidade de confirmação manual do pagamento, promovendo rapidez, segurança e acessibilidade para todos os usuários. Essas modelagens auxiliam diretamente no entendimento do comportamento dinâmico do sistema e favorecem tanto o desenvolvimento quanto o uso inclusivo da plataforma.

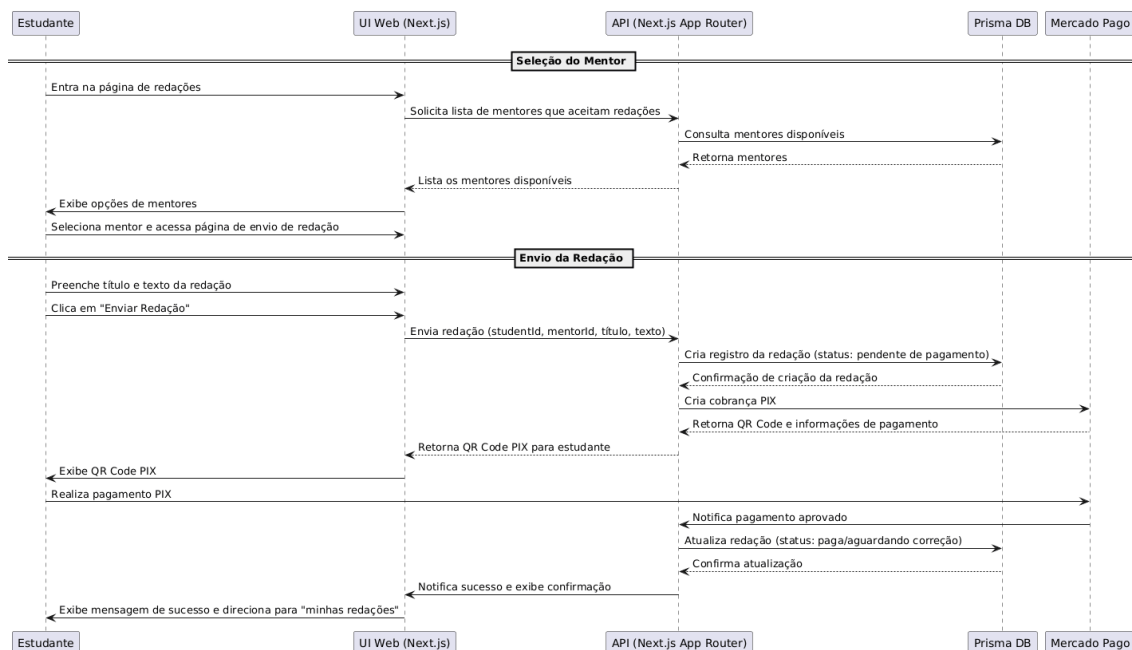


Figura 4. Diagrama de Sequência do fluxo de Envio de Redação para Correção, com Pagamento PIX integrado.

4. Desenvolvimento e Resultados

Esta seção apresenta os principais conceitos, ferramentas e tecnologias relacionadas ao desenvolvimento web que embasam a criação da aplicação sugerida, além de ressaltar pesquisas e autores que respaldam as decisões técnicas tomadas no projeto.

4.1. Full-Stack com Next.js

O desenvolvimento da aplicação foi realizado utilizando o `Next.js`, uma estrutura `React` para a construção de aplicativos da Web full-stack. Foram usados `React Components` para criar interfaces de usuário e o `Next.js` para recursos e otimizações adicionais, como roteamento automático, renderização do lado do servidor e geração de sites estáticos. A ferramenta configurou automaticamente ferramentas de nível inferior, como empacotadores e compiladores, permitindo foco na criação do produto e envio rápido. Segundo Neves (2024), o `Next.js` melhorou significativamente a velocidade de carregamento ao preparar partes do site antecipadamente no servidor, além de facilitar a criação e organização de páginas em projetos de maior escala [Neves 2024].

O `App Router` do `Next.js` oferece suporte a novos recursos do `React`, como componentes do servidor, garantindo aplicativos interativos, dinâmicos e rápidos. Essa abordagem full-stack elimina a necessidade de frameworks separados para back-end e front-end, simplificando o desenvolvimento e a manutenção [Next.js 2025].

4.2. Front-End

O front-end foi construído com o `Next.js`, utilizando o `App Router` para criar interfaces responsivas e acessíveis. A estilização foi realizada com o `Tailwind CSS`, permitindo composições rápidas de estilos diretamente no markup `HTML`, o que otimizou a produtividade e a escalabilidade do design [Tailwind Labs 2025]. Além disso, foram utilizados

componentes do *shadcn/ui* [Shadcn 2023], uma biblioteca baseada em Tailwind que fornece elementos reutilizáveis e acessíveis. Esses componentes foram essenciais para a construção de elementos da interface, como os cartões de mentoria (cards), o carrossel de navegação e o conjunto de abas presentes na barra lateral esquerda, garantindo consistência visual e maior velocidade de desenvolvimento.

A Figura 5 mostra a tela inicial da plataforma desenvolvida. No topo, há três cartões grandes representando categorias de mentoria: Delegacia de Polícia, OAB e TCE. À direita, há um botão vermelho “Começar agora” dentro do módulo de Correção de Redação. Abaixo, são exibidos calendários com datas disponíveis para agendamento de consultas com mentores, cada dia apresentado como um pequeno cartão retangular. À esquerda, existe uma barra lateral vertical com itens de navegação, incluindo “Início”, “Meus Agendamentos” e “Minhas Redações”.

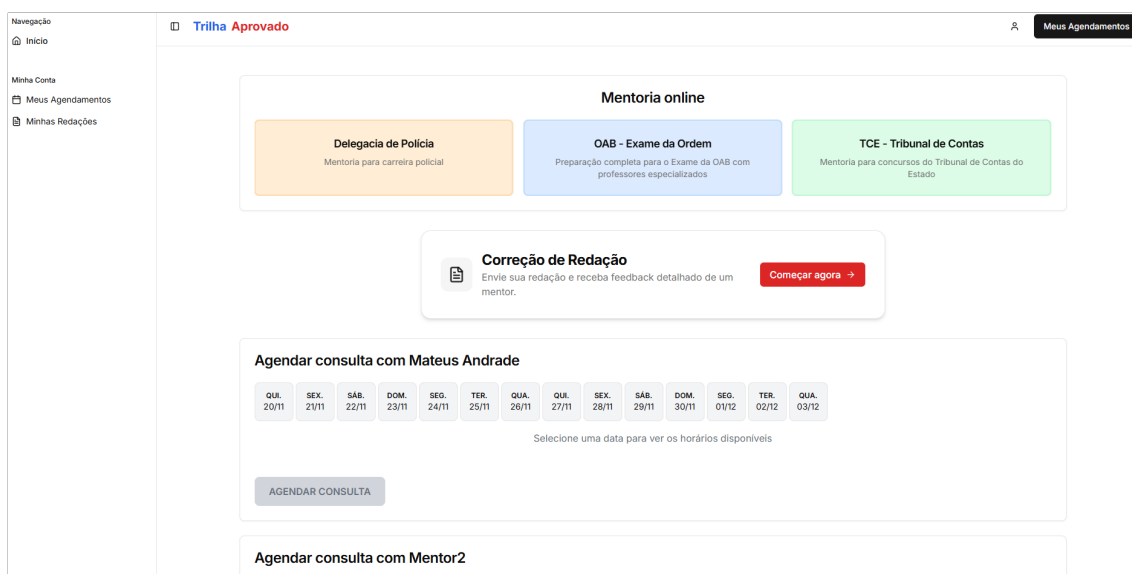


Figura 5. Tela inicial da aplicação, exibindo as opções de mentoria, o módulo de correção de redações e o sistema de agendamentos.

A interface apresentada na Figura 6 corresponde ao módulo de escrita de redações da plataforma. Na parte superior da tela, há um campo destinado ao título do texto, seguido por uma barra de ferramentas que permite ao usuário ajustar o tamanho da fonte, aumentar ou diminuir sua escala e aplicar indentação automática no início dos parágrafos. Abaixo desses controles, encontra-se a área principal de escrita, estruturada como uma folha pautada digital, com linhas horizontais e uma margem vertical à esquerda, simulando um caderno escolar. No centro dessa área, um texto guia indica ao usuário onde iniciar a redação. Essa organização visual oferece um ambiente de escrita limpo e intuitivo e acessível.

A interface exibida na Figura 7 corresponde ao Painel do Mentor, destinado à gestão de horários e informações cadastrais do profissional responsável pelas mentorias. No topo da tela, há três campos preenchíveis: nome do mentor, número de telefone e uma breve descrição. Abaixo desses campos, um botão azul permite atualizar o perfil. Logo em seguida, encontra-se o módulo de seleção de horários, no qual é apresentada uma faixa horizontal de datas referentes aos próximos 30 dias; cada data aparece dentro de

Escrever Redação

Desenvolva seu texto utilizando até 30 linhas

Título da redação

Fonte: A- 18 A+ Indentar Use Enter para criar parágrafos. Use Indentar ou Tab para começar parágrafo.

Comece a escrever sua redação aqui...

Figura 6. Tela do módulo de escrita de redações, exibindo campo para título, barra de ferramentas e área pautada destinada à produção textual.

um cartão retangular com o dia da semana e o respectivo dia do mês. Ao selecionar uma data, são exibidos os horários disponíveis, organizados em cartões verdes distribuídos em linhas, representando opções como 08:00, 09:00, 10:00, entre outros. Mais abaixo, há uma área denominada “Horários selecionados”, que lista todos os horários previamente escolhidos pelo mentor, cada um dentro de um cartão azul claro com um botão de remover ao lado. No final do painel, há um botão para confirmar os horários cadastrados. À esquerda da página, encontra-se uma barra lateral com opções de navegação, divididas entre seções de conta do usuário e funcionalidades específicas do mentor, incluindo Dashboard, Agendamentos e Redações.

4.3. Programação Orientada a Objetos (POO)

A Programação Orientada a Objetos (POO) constituiu uma abordagem fundamental para a organização do código e a reutilização de componentes no desenvolvimento realizado. Ela possibilitou que sistemas fossem construídos com base em abstrações como classes e objetos, facilitando a manutenção, escalabilidade e modularização. Essa metodologia atuou de forma eficaz em conjunto com frameworks modernos como `Next.js` e `NestJS`, especialmente quando usada com `Prisma` ORM, pois promoveu uma arquitetura mais limpa e reutilizável. Em projetos com `NestJS`, por exemplo, foi comum encapsular regras de negócio em classes e serviços, alinhando-se perfeitamente aos princípios da POO. No `Next.js`, a divisão de responsabilidades e a utilização de componentes foram incre-

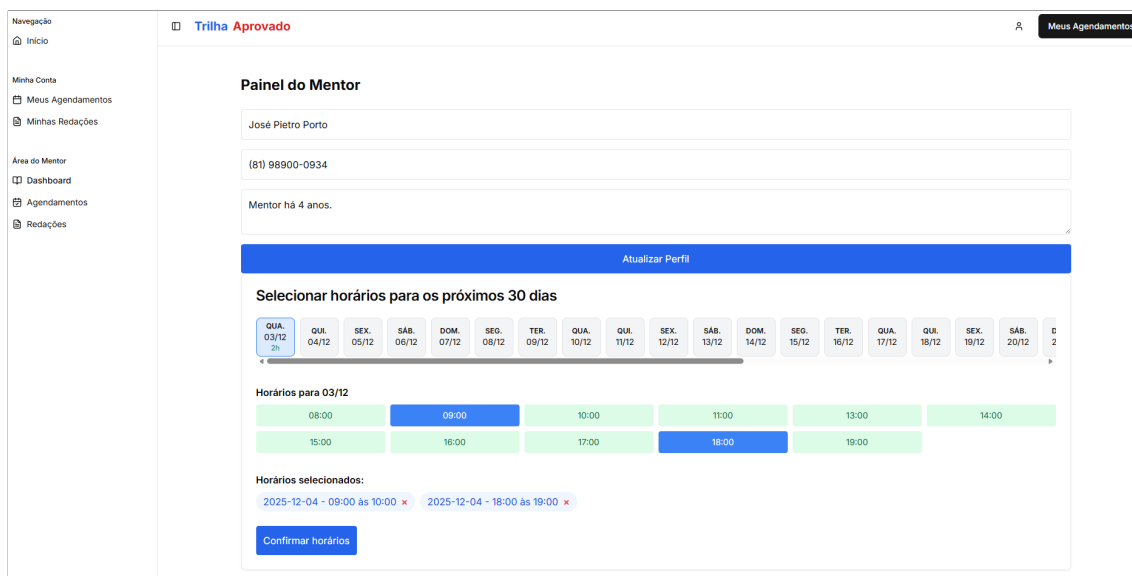


Figura 7. Painel do Mentor, com campos de informações pessoais, seleção de horários disponíveis e ferramentas de gerenciamento de agenda.

mentadas com conceitos de orientação a objetos na camada de serviços.

Com a crescente complexidade dos sistemas, a reutilização de código tornou-se um requisito essencial para evitar aumento exponencial no tempo de desenvolvimento. A Programação Orientada a Objetos atende a essa demanda ao permitir que partes do código sejam independentes e reutilizáveis, contribuindo significativamente para a produtividade e manutenção dos projetos [Henrique 2014].

4.4. Testes Unitários

A qualidade do código foi assegurada por meio de testes automatizados utilizando o Vitest, que possibilitou a verificação do funcionamento de pequenas partes do sistema de forma isolada, auxiliando na prevenção de falhas inesperadas e proporcionando maior segurança no desempenho da aplicação. Os testes unitários favoreceram a detecção precoce de erros lógicos, contribuindo para a economia de tempo durante o desenvolvimento e servindo como documentação prática do comportamento esperado do código [Next.js 2025].

4.5. Banco de Dados

Foi utilizado o Prisma Postgres, um banco de dados sem servidor amigável ao desenvolvedor, configurado e integrado ao projeto. A solução proporcionou consultas rápidas e escalabilidade adequada para o ambiente de produção. O Prisma Postgres integrou-se ao fluxo de desenvolvimento com um cliente de banco de dados tipado e práticas de migração que facilitaram a evolução do esquema [Vercel 2025a].

4.6. TypeScript

O TypeScript foi utilizado como linguagem principal, trazendo tipagem estática para o código. Sua tipagem auxiliou na prevenção de erros ao longo do desenvolvimento e elevou a qualidade geral do código, tornando-o mais previsível e seguro. Esse controle foi vantajoso tanto para aplicações front-end quanto para back-end, facilitando

refatorações e permitindo que problemas fossem detectados ainda na fase de compilação [Casa do Desenvolvedor 2023].

4.7. Deploy e Hosting

Para o deploy e hospedagem da aplicação, foi utilizada a plataforma *Vercel*, reconhecida por sua integração nativa com frameworks como *Next.js* e por oferecer uma infraestrutura otimizada para aplicações web modernas. A *Vercel* destacou-se por proporcionar deploy automático a partir de repositórios *Git*, ambientes de visualização para testar alterações antes da produção, e uma rede de distribuição de conteúdo (CDN) global que garantiu alta performance e disponibilidade.

A plataforma oferece recursos avançados como reversão instantânea de implantações para recuperação rápida de incidentes de produção, pacote de observabilidade para monitoramento de desempenho e depuração de fluxos de trabalho e aplicativos de IA, e otimizações automáticas, como regeneração estática incremental e otimização de imagens, que contribuem significativamente para a experiência do usuário final. Além disso, a *Vercel* facilita a implementação de práticas de desenvolvimento modernas, incluindo integração contínua e deploy contínuo (CI/CD), permitindo que a equipe de desenvolvimento mantenha um fluxo de trabalho ágil e confiável [Vercel 2025b].

4.8. Monetização de Aplicações Web

Um ponto central da arquitetura é a integração com o serviço de pagamentos. O *Mercado Pago* foi utilizado para o processamento de pagamentos via *PIX*, permitindo ao sistema gerar *QR Codes* e registrar transações de forma automatizada. A plataforma oferece uma *Checkout API* que possibilita a definição de uma *notification_url*, por meio da qual o sistema recebe notificações assíncronas via *webhooks* sempre que o status de uma transação é atualizado pelo provedor. Essa comunicação possibilita que, após a confirmação bancária do pagamento, o agendamento correspondente seja automaticamente atualizado para o status “confirmado”, concluindo o fluxo sem necessidade de intervenção manual [Mercado Pago 2025].

4.9. Controle de Versão com Git e GitHub

O desenvolvimento contou com a utilização do sistema de controle de versões *Git* [Git 2025], permitindo registrar, organizar e acompanhar cada modificação realizada no código-fonte. O uso do *Git* garantiu rastreabilidade, segurança e facilidade na reversão de alterações quando necessário.

Para hospedagem do repositório e colaboração no desenvolvimento, utilizou-se a plataforma *GitHub* [GitHub 2025]. A ferramenta forneceu recursos como gerenciamento de *branches*, histórico de *commits*, *issues* e integração com *pipelines* automatizadas, contribuindo para uma construção organizada e transparente do software. Além disso, o desenvolvimento foi centralizado no repositório oficial do projeto, disponível em: <https://github.com/Vini0100/trilha-aprovado>. Nesse repositório, foram organizados todos os artefatos do sistema, incluindo código-fonte, documentação, histórico de *commits*, *issues* e *branches* utilizados ao longo do processo. A disponibilização pública do repositório também possibilita transparência, reprodutibilidade e acompanhamento completo da evolução do software.

5. Conclusões

Este trabalho apresentou o desenvolvimento e a implementação de um Sistema Web para Mentoria Online destinado a estudantes de concursos públicos. A plataforma foi construída utilizando tecnologias modernas (Next.js 15, TypeScript, Prisma ORM e PostgreSQL) e integrou pagamentos via Mercado Pago (PIX), com atualização automática de status por webhook.

Os resultados obtidos demonstraram que os objetivos específicos foram alcançados: as funcionalidades previstas (autenticação, agendamento com mentores, integração de pagamentos e interfaces responsivas) sendo validadas por meio de testes unitários. A aplicação mostrou estabilidade e aderência aos requisitos definidos, e o fluxo de pagamento operou de ponta a ponta, exibindo confirmação ao usuário após a aprovação do PIX.

O contexto de crescimento do mercado de concursos no Brasil reforça a relevância da solução, e os achados indicam que uma plataforma web moderna pode, simultaneamente, apoiar a organização do estudo dos candidatos e viabilizar uma fonte de renda para mentores, por meio de uma solução tecnicamente robusta e escalável.

Como trabalhos futuros, recomenda-se: (i) conduzir avaliações com usuários finais em estudos controlados de usabilidade; (ii) ampliar funcionalidades, como notificações e lembretes, reagendamentos e relatórios de desempenho; (iii) realizar o *hardening* de produção, contemplando observabilidade completa, validação de assinatura de webhooks, políticas de segurança e conformidade com a Lei Geral de Proteção de Dados (LGPD); (iv) otimizar a escalabilidade do sistema e os custos operacionais; (v) expandir a cobertura de testes automatizados e os requisitos de acessibilidade; (vi) realizar testes de caixa branca mais abrangentes, incluindo critérios formais de cobertura, como cobertura de decisões, condições e caminhos independentes, visando uma análise aprofundada do comportamento interno do código; (vii) conduzir testes de caixa preta, avaliando o sistema com base em seus requisitos funcionais e não funcionais, sem considerar a estrutura interna, a fim de verificar a conformidade com os cenários de uso esperados; (viii) executar testes de integração e testes de sistema para validar a correta comunicação entre os módulos e o funcionamento da aplicação de forma integrada; e (ix) aplicar testes de regressão para garantir que evoluções e manutenções futuras não introduzam falhas em funcionalidades previamente implementadas.

Referências

- Casa do Desenvolvedor (2023). Typescript: descubra seus detalhes técnicos e o real motivo dele existir! <https://blog.casadodesenvolvedor.com.br/typescript/>. Acesso em: 30 abr. 2025.
- Estratégia Concursos (2025). Estratégia concursos. <https://www.estrategiacursos.com.br>. Acesso em: 10 nov. 2025.
- Git (2025). Git: Fast version control system. <https://git-scm.com/>. Acesso em: 25 nov. 2025.
- GitHub (2025). Github. <https://github.com/>. Acesso em: 25 nov. 2025.
- Gran Cursos Online (2025). Gran cursos online. <https://www.grancursosonline.com.br>. Acesso em: 10 nov. 2025.

- Henrique (2014). Os 4 pilares da programação orientada a objetos. <https://www.devmidia.com.br/os-4-pilares-da-programacao-orientada-a-objetos/9264>. Acesso em: 13 maio 2025.
- Instituto Adriana Figueiredo (2025). Bloqueio criativo: entenda a estrutura da redação para concurso e desbloqueie a criatividade. <https://professora.adrianafigueiredocursos.com.br/blog/bloqueio-criativo/>. Acesso em: 1 nov. 2025.
- Instituto Consciência GO (2025). Redação como fator decisivo em concursos públicos. <https://icg.edu.br/redacao-como-fator-decisivo-em-concursos-publicos/>. Acesso em: 8 nov. 2025.
- Mercado Pago (2025). Checkout transparente (checkout api): Visão geral. <https://www.mercadopago.com.br/developers/pt/docs/checkout-api/overview>. Acesso em: 26 out. 2025.
- Neves, V. (2024). Next.js: O que é e por que usar esse framework? Alura. <https://www.alura.com.br/artigos/next-js>. Acesso em: 5 maio 2025.
- Next.js (2025). Next.js documentation. <https://nextjs.org/docs>. Acesso em: 4 out. 2025.
- Organiza Concursos (2025). Organiza concursos: Plataforma de questões e organização de estudos. <https://organizaconcursos.com.br>. Acesso em: 10 nov. 2025.
- Qconcursos (2025). Qconcursos: Plataforma de questões e estudos. <https://www.qconcursos.com>. Acesso em: 10 nov. 2025.
- Revide (2025). Mercado de concursos públicos no brasil deve crescer e movimentar r\$ 5 bilhões em 2025. <https://www.revide.com.br/noticias/economia/mercado-de-concursos-publicos-no-brasil-deve-crescer-e-movimentar-r-5-bilhoes-em-2025/>. Acesso em: 28 maio 2025.
- Shadcn (2023). shadcn/ui: Reusable components built with tailwind css. <https://ui.shadcn.com>. Acesso em: 20 nov. 2025.
- Tailwind Labs (2025). Tailwind css. <https://tailwindcss.com/>. Acesso em: 10 maio 2025.
- Vercel (2025a). Prisma postgres. <https://vercel.com/marketplace/prisma>. Acesso em: 4 out. 2025.
- Vercel (2025b). Vercel: Develop. preview. ship. <https://vercel.com/>. Acesso em: 4 out. 2025.