

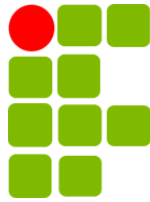
Instituto Federal de Educação, Ciência e Tecnologia da
Paraíba
Campus Campina Grande
Coordenação do Curso Superior de Engenharia de
Computação

Questfy: Um Software para Automatizar a Criação de Questões de Múltipla Escolha Utilizando LLM

GABRIEL DA SILVA NASCIMENTO
JHONATAN GUILHERME OLIVEIRA PEREIRA

Orientador: Igor Barbosa da Costa, D.Sc.

Campina Grande, Setembro de 2024
© Gabriel da Silva Nascimento
© Jhonatan Guilherme Oliveira Pereira



Instituto Federal de Educação, Ciência e Tecnologia da
Paraíba
Campus Campina Grande
Coordenação do Curso Superior de Engenharia de
Computação

Questfy: Um Software para Automatizar a Criação de Questões de Múltipla Escolha Utilizando LLM

GABRIEL DA SILVA NASCIMENTO
JHONATAN GUILHERME OLIVEIRA PEREIRA

Trabalho de Conclusão de Curso
apresentado ao Curso Engenharia de
Computação, do Instituto Federal da
Paraíba – Campus Campina Grande,
em cumprimento às exigências parciais
para a obtenção do título de Bacharel
em Engenharia de Computação.

Orientador: Igor Barbosa da Costa, D.Sc.

Campina Grande, Setembro de 2024

N244u Nascimento, Gabriel da Silva

UQuestfy: Um software para automatizar a criação de questões de múltipla escolha utilizando LLM / Gabriel da Silva Nascimento, Jhonatan Guilherme Oliveira Pereira. - Campina Grande, 2024.
30 f. : il.

Trabalho de Conclusão de Curso (Curso Superior de Engenharia de Computação) - Instituto Federal da Paraíba, 2024.

Orientador: Prof. D.Sc. Igor Barbosa da Costa

1. Engenharia de computação 2. Tecnologia educacional - automação de avaliação 3. Aprendizagem adaptativa 4. Modelos de linguagem de grande escala - Large Language Model (LLM), I. Pereira, Jhonatan Guilherme Oliveira II. Costa, Igor Barbosa da III. Título.

CDU 004:37.02

Questfy: Um Software para Automatizar a Criação de Questões de Múltipla Escolha Utilizando LLM

Trabalho de Conclusão de Curso apresentado ao Curso Engenharia de Computação, do Instituto Federal da Paraíba – Campus Campina Grande, em cumprimento às exigências parciais para a obtenção do título de Bacharel em Engenharia de Computação.

Aprovada em 18/09/2024

Igor Barbosa da Costa, D.Sc.
Orientador

Professor
Danyllo Wagner Albuquerque, D.Sc.

Professor
Elmano Ramalho Cavalcanti, D.Sc.

Campina Grande, Paraíba, Brasil
Setembro/2024

“Aprender uma lição sem enfrentar dificuldades não tem o mesmo valor, pois não se pode alcançar algo sem fazer sacrifícios. No entanto, ao superar os desafios e alcançar o objetivo desejado, desenvolve-se um coração forte como aço.”

Edward Elric

À Deus. À meus pais, familiares e amigos, por todo apoio e carinho!

Agradecimentos

Por Gabriel: Ao meu avô, Josete, que sempre apoiou minha decisão de priorizar os estudos, incentivando-me a continuar me aprimorando e adquirindo mais conhecimento, mesmo sem ter tido a oportunidade de uma alfabetização completa. À minha tia, Jucileide, por sempre apoiar minhas decisões corretas, orientar-me em meus equívocos e proporcionar apoio nos momentos mais difíceis.

Aos meus amigos e companheiros ao longo desta jornada, que proporcionaram não apenas momentos de diversão, mas também de apoio e compartilhamento. Em especial, gostaria de citar Lilian, Eduarda, Eduardo, Jhonatan e Lucas; guardarei nossas lembranças com muito carinho.

Aos professores que me transmitiram o conhecimento necessário para chegar até aqui, não apenas durante as aulas, mas também nos momentos de dúvida fora delas. Em especial, agradeço ao Fagner, Paulo e Danyllo; gostaria de ter tido mais tempo para conversar sobre diversos assuntos.

A todos mencionados, meu mais sincero agradecimento.

Por Jhonatan: À minha mãe, Alcione, meu suporte diário, que me apoiou em todos os momentos da minha vida e sempre me orientou a seguir o caminho correto, moldando a pessoa que sou hoje. Ao meu pai, Nevalto, exemplo de resiliência, que me fez acreditar no meu futuro e ser persistente na busca pelos meus objetivos. À minha namorada, Mayara, que me deu muito apoio durante a graduação, acreditou em mim e me fez acreditar em mim mesmo quando eu não conseguia. Amo vocês.

Aos meus amigos de infância e aos que conheci durante a graduação, que me ajudaram nos momentos em que precisei e compartilharam boas risadas em meio ao estresse da rotina. Em especial, ao TDT, meu querido grupo de amigos, especialmente Pedro, Thiago, Henrique, Symon e Rafael. E, claro, aos meus grandes amigos Gabriel e Lucas, que me acompanharam desde o primeiro dia até agora. Fico feliz por estar aqui hoje com vocês.

Aos professores que me desafiaram e compartilharam seus conhecimentos valiosos, que foram essenciais para minha formação como Engenheiro de Computação. Preciso citar Victor, Paulo, Jerônimo e Henrique; agradeço por serem referências.

A todos os mencionados, minha profunda gratidão por me guiarem até aqui.

Por ambos: A Igor, pelo apoio e paciência durante o desenvolvimento deste TCC. É uma honra contar com seu conhecimento, somos gratos.

Sumário

1	Introdução	13
2	Fundamentação Teórica	14
2.1	Tecnologia na Educação	14
2.2	Uso de Questões de Múltipla Escolha na Avaliação Educacional .	15
2.3	Inteligência Artificial na Educação	15
2.4	<i>Prompt Engineering</i>	16
2.5	Conclusão	16
3	Metodologia	17
3.1	Requisitos do Sistema	17
3.1.1	Requisitos Funcionais	17
3.1.2	Requisitos Não Funcionais	17
3.2	Tecnologias Utilizadas	18
3.2.1	Frontend	18
3.2.2	Backend	19
3.2.3	Modelo de LLM	20
3.3	Etapas do Desenvolvimento	21
3.3.1	Metodologia de Desenvolvimento	22
3.3.2	Desenvolvimento	22
3.4	Testes e Validações	24
4	Resultados	24
4.1	Criação e Edição de Questões	24
4.2	Gerenciamento de Questões	26
4.3	Exportação de Questões em PDF	27
5	Conclusão	28
5.1	Limitações e Trabalhos Futuros	29
5.2	Lições Aprendidas	29

Lista de Abreviaturas

API	<i>Application Programming Interface</i>
IA	Inteligência Artificial
LLM	<i>Large Language Model</i>
NLP	<i>Natural Language Processing</i>
PDF	<i>Portable Document Format</i>
QME	Questão de Múltipla Escolha

Lista de Figuras

1	Diagrama do Frontend.	19
2	Diagrama do Backend.	20
3	Cronograma de Desenvolvimento.	23
4	Tela de Definição dos Parâmetros para Geração das Questões.	25
5	Tela de Edição Após a Geração das Questões.	25
6	Tela de Edição ao Deletar uma Questão.	26
7	Barra Lateral de Exibição de Questões: Filtro de Pesquisa.	26
8	Tela de Edição de Questões.	27
9	Tela de Seleção de Questões para Exportação.	27
10	PDF Gerado na Exportação das Questões.	28

Questfy: Um Software para Automatizar a Criação de Questões de Múltipla Escolha Utilizando LLM

Nascimento S. Gabriel¹, Pereira O. G. Jhonatan¹, Barbosa C. Igor¹

¹Instituto Federal de Ciência e Tecnologia– IFPB - Campus Campina Grande (IFPB)
CEP 58432-300– Campina Grande - PB– Brazil

{nascimeno.gabriel, jhonatan.oliveira, igor.costa}@academico.ifpb.edu.br

Resumo. Este trabalho explora o uso da automação na criação de questões de múltipla escolha (QMEs) para avaliações educacionais, destacando o ganho de tempo proporcionado aos educadores. Fundamentado nas teorias de aprendizagem adaptativa e nos princípios da tecnologia educacional, o trabalho aborda os desafios enfrentados na elaboração de questões de alta qualidade e na personalização de avaliações, atividades que demandam tempo significativo. Para enfrentar esses desafios, foi desenvolvido o Questfy, um software projetado para automatizar a geração e gestão de QMEs. Utilizando a metodologia ágil Scrum e incorporando um modelo de Large Language Model (LLM), o Questfy possibilita a criação rápida e personalizada de avaliações, otimizando o tempo dos educadores para outras atividades pedagógicas. O sistema inclui uma interface intuitiva que permite a definição de parâmetros e edição das questões geradas, enquanto o backend, desenvolvido em Python com FastAPI e MongoDB, garante a robustez na gestão e armazenamento de dados. Os resultados indicam que a ferramenta é eficaz na automatização da criação de avaliações, ao mesmo tempo que sublinha a importância da revisão manual por educadores para assegurar a qualidade das questões. Futuras melhorias incluem a expansão para outros formatos de avaliação, a personalização avançada das avaliações e a integração com dispositivos móveis para leitura automatizada de gabaritos.

Palavras-chave: *Transformação Digital, Educação, Geração Automática de Questões, LLM, Software Educacional*

Abstract. This study explores the use of automation in the creation of multiple-choice questions (MCQs) for educational assessments, highlighting the time savings provided to educators. Grounded in adaptive learning theories and educational technology principles, the work addresses the challenges faced in creating high-quality questions and customizing assessments, activities that require significant time. To tackle these challenges, the software Questfy was developed, designed to automate the generation and management of MCQs. Utilizing the agile Scrum methodology and incorporating a Large Language Model (LLM), Questfy enables the rapid and personalized creation of assessments, optimizing educators' time for other pedagogical activities. The system includes an intuitive interface that allows the definition of parameters and the editing of generated questions, while the backend, developed in Python with FastAPI and MongoDB, ensures robustness in data management and storage. The results indicate that the tool is effective in automating assessment creation, while also

highlighting the importance of manual review by educators to ensure the quality of the questions. Future improvements include the expansion to other assessment formats, advanced assessment customization, and integration with mobile devices for automated answer sheet scanning.

Keywords: *Digital Transformation, Education, Automated Question Generation, LLM, Educational Software*

1. Introdução

A transformação digital tem impactado profundamente a sociedade, alterando a forma como interagimos, aprendemos e ensinamos. No contexto educacional, a incorporação de tecnologias como a comunicação instantânea e ambientes de realidade virtual não apenas redefine os paradigmas pedagógicos, mas também cria oportunidades para métodos de ensino mais eficazes e personalizados. Esse movimento foi significativamente acelerado pela pandemia de COVID-19, que forçou uma rápida adoção de plataformas educacionais como *Google Classroom* e *Google Meet*, eliminando barreiras impostas pela distância física e assegurando a continuidade da educação.

Entretanto, essa rápida integração tecnológica também trouxe à tona desafios significativos para os educadores. A necessidade de adaptar abordagens pedagógicas para um ambiente virtual, especialmente para apoiar alunos com deficiência, é exacerbada pela falta de treinamento e recursos adequados [Ciobanu et al. 2023]. Além disso, muitos professores enfrentam limitações em suas competências digitais, o que evidencia uma lacuna que precisa ser preenchida para garantir que a tecnologia seja uma aliada eficaz no processo de ensino [Vásquez-Pajuelo et al. 2024]. Mesmo em ambientes urbanos não modernizados, onde o acesso à tecnologia é limitado, há uma correlação positiva entre o uso dessas ferramentas e o desenvolvimento de competências digitais [Vásquez-Pajuelo et al. 2024]. Portanto, a capacitação dos educadores em tecnologias da informação é essencial para assegurar uma transição bem-sucedida para o ensino digital.

Dentro desse novo cenário educacional, o papel do educador não se limita mais à transmissão de conhecimento; ele agora inclui a criação e a adaptação de materiais de avaliação que possam acompanhar a dinâmica do aprendizado digital. A elaboração de questões de múltipla escolha (QMEs) de alta qualidade, por exemplo, tornou-se um desafio ainda maior, exigindo não apenas conhecimento aprofundado do conteúdo, mas também criatividade e habilidades técnicas. Além disso, a necessidade de atualizar constantemente essas questões para manter sua relevância, especialmente em um ambiente onde as informações se tornam obsoletas rapidamente, adiciona uma camada extra de complexidade ao trabalho dos educadores.

A avaliação dos alunos, uma das etapas mais críticas do processo educativo, é frequentemente realizada por meio de questionários. Embora sejam amplamente utilizados, a criação de QMEs eficazes demanda tempo e esforço consideráveis, sendo um processo que muitas vezes sobrecarrega os educadores [Huang et al. 2020]. As QMEs são também alvo de críticas por estarem, muitas vezes, excessivamente focadas na recordação de detalhes, em vez de avaliarem a compreensão conceitual dos alunos [Biggs et al. 2022]. Além disso, o processo de substituição e atualização dessas questões, necessário para garantir sua validade, aumenta ainda mais a carga de trabalho dos professores [Kurdi et al. 2020].

Para enfrentar esses desafios, a inteligência artificial (IA) surge como uma solução

promissora. A IA oferece o potencial de transformar a criação de materiais didáticos e avaliações, automatizando processos repetitivos e permitindo uma personalização maior. Isso não só alivia a carga de trabalho dos educadores, como também melhora a qualidade das interações pedagógicas.

Nesse contexto, propomos o desenvolvimento de um software, o *Questfy*, que gerencia e automatiza a geração de questões de múltipla escolha, utilizando um modelo de LLM (Large Language Model). O software facilita a criação de avaliações ao permitir a geração de questões com base em temas e níveis de dificuldade definidos pelo educador, com possibilidade de edição. As questões são organizadas por tema e podem ser reutilizadas e exportadas em formato PDF, economizando tempo e esforço dos educadores.

O principal objetivo é otimizar o processo de criação de avaliações, liberando os educadores para focarem em atividades pedagógicas mais significativas, enquanto o sistema cuida das tarefas repetitivas. Além disso, o software oferece a possibilidade de criar exercícios personalizados para os alunos, fortalecendo o papel da tecnologia como um aliado no ensino.

2. Fundamentação Teórica

Esta Seção examina a interseção entre tecnologia, métodos de avaliação e inteligência artificial na educação, destacando as oportunidades e desafios que emergem dessa integração. A fundamentação teórica que se segue pretende fornecer uma visão abrangente das transformações educacionais impulsionadas pela tecnologia, bem como das implicações dessas mudanças para a prática pedagógica, preparando o terreno para a solução proposta neste trabalho.

2.1. Tecnologia na Educação

O avanço tecnológico aplicado à educação tem possibilitado a personalização do ensino e o desenvolvimento de práticas pedagógicas mais eficazes. Josué et al. [Josué et al. 2023] demonstram que a tecnologia na educação promove o aprimoramento da didática e a criação de ferramentas que diversificam e ampliam as possibilidades de ensino. Nadeem et al. [Nadeem et al. 2024] reforçam que a tecnologia é essencial para a educação moderna, pois permite a implementação de novas ferramentas em sala de aula e facilita a conexão e o compartilhamento de conhecimentos entre educadores de diferentes localidades.

Jumman [Sani et al. 2024] argumentam que a tecnologia desempenha um papel crucial no engajamento e aumento da motivação dos alunos, permitindo um ensino mais personalizado e diversificado. Além disso, a tecnologia motiva os educadores a apresentarem o conteúdo de forma dinâmica e entusiasmada, promovendo uma maior comunicação e colaboração entre todos os envolvidos. Joshi [Joshi 2023] destaca a importância da diversificação tecnológica no ambiente educacional, onde a combinação de hardware e software gera mídias inovadoras que auxiliam na diversificação do ensino.

Historicamente, a introdução de tecnologias na educação começou com o uso de projetores e computadores para aprimorar as experiências de aprendizado [Kudryashov et al. 2024]. A combinação do ensino presencial com o ensino remoto (online) gerou o conceito de “ensino combinado”, que aproveita as ferramentas digitais para melhorar a experiência de aprendizado e desenvolver competências exigidas nas últimas

décadas [Dancsa et al. 2023]. A pandemia de COVID-19 atuou como um catalisador para o ensino remoto, destacando a importância das ferramentas digitais e inserindo uma diversidade de novas tecnologias para ensino e avaliação remotos [Dancsa et al. 2023]. Essas transformações são fundamentais para entender o contexto em que a solução proposta foi desenvolvida, uma vez que o software visa atender às novas demandas educacionais emergentes desse cenário.

2.2. Uso de Questões de Múltipla Escolha na Avaliação Educacional

A avaliação educacional desempenha um papel significativo no aprimoramento do aprendizado dos alunos, servindo como uma abordagem sistemática para monitorar o progresso e o desenvolvimento dos estudantes. Ela também é essencial para manter a qualidade da educação nas instituições [Md Din et al. 2023]. Entre os métodos de avaliação, as questões de múltipla escolha (QME) são amplamente utilizadas devido à sua capacidade de permitir uma pontuação rápida e precisa, além de cobrir um amplo conteúdo instrucional em um curto espaço de tempo [Brown and Abdunabi 2017].

No entanto, a criação de QMEs eficazes exige tempo e esforço consideráveis, muitas vezes sobrecarregando os educadores [Huang et al. 2020]. A necessidade de substituição e atualização constante dessas questões, para garantir sua validade, aumenta ainda mais a carga de trabalho dos professores [Kurdi et al. 2020]. Esses desafios justificam a busca por soluções que automatizem a criação e gestão de QMEs, como o software desenvolvido neste trabalho, que visa aliviar a carga dos educadores e melhorar a qualidade das avaliações.

2.3. Inteligência Artificial na Educação

A integração da inteligência artificial (IA) na educação apresenta um imenso potencial para revolucionar o aprendizado tradicional, oferecendo experiências personalizadas que atendem às necessidades individuais dos alunos. As tecnologias de IA, como sistemas inteligentes de tutoria, adaptam o conteúdo educacional a cada aluno, aumentando o engajamento e o sucesso acadêmico [Harry 2023]. Alkan [Alkan 2024] conclui que a IA tem um potencial significativo para aprimorar as experiências educacionais, ajudando a identificar os pontos fortes e fracos dos alunos, fornecer respostas automáticas e facilitar experiências de aprendizado personalizadas.

O aprendizado profundo (deep learning) e o processamento de linguagem natural (PNL) são tecnologias centrais para essa transformação educacional, pois aprimoram as capacidades das aplicações de IA. O aprendizado profundo, com sua capacidade de modelar padrões complexos por meio de redes neurais, possibilita o desenvolvimento de sistemas inteligentes de tutoria que se adaptam a estilos de aprendizagem individuais, oferecendo experiências educacionais personalizadas. O PNL, por sua vez, permite que máquinas entendam e interajam com a linguagem humana, essencial para criar ferramentas educacionais baseadas em IA, como assistentes virtuais e sistemas de avaliação automatizados [Alkan 2024, Lievertz 2019, Lima et al. 2024, Abdulmunem 2023]. Essas ferramentas não apenas ajudam a reduzir os encargos administrativos dos educadores, mas também fornecem feedback e apoio imediatos aos alunos, promovendo um ambiente de aprendizado mais interativo e envolvente.

Embora a IA ofereça benefícios substanciais, ela também enfrenta desafios e limitações. Um dos principais desafios é a incapacidade dos sistemas de IA de repli-

car completamente o papel complexo e multifacetado dos educadores humanos. Embora a IA possa auxiliar em várias tarefas, ela não substitui o toque humano necessário para um ensino e aprendizagem eficazes [Alkan 2024, Sadiku et al. 2021, Lievertz 2019]. Reconhecer essas limitações é crucial para o desenvolvimento de ferramentas educacionais equilibradas, como o software proposto neste trabalho, que busca integrar a inovação tecnológica com a prática pedagógica tradicional.

2.4. *Prompt Engineering*

O *prompt engineering* é o processo de projetar e refinar solicitações de entrada para obter respostas desejadas de modelos de IA baseados em NLP. O domínio desse processo é crucial para maximizar o potencial de modelos como o GPT em diversas aplicações e setores. Esse processo auxilia na obtenção de melhores resultados, orientando a IA a produzir respostas relevantes e precisas [Ekin 2023].

O *prompt engineering* eficaz inclui técnicas como:

- **Clareza:** As solicitações devem ser claras e concisas. A ambiguidade pode levar a respostas inesperadas ou irrelevantes, por isso é essencial articular a solicitação com precisão.
- **Restrições explícitas:** Definir diretrizes específicas no *prompt* pode ajudar a restringir as respostas da IA, incluindo o formato, tamanho ou estilo da saída desejada.
- **Experimentação:** A experimentação permite que os usuários descubram quais solicitações produzem os melhores resultados e ajuda a refinar a abordagem ao longo do tempo.
- **Refinamento iterativo:** Melhorar continuamente as solicitações com base nos resultados obtidos é vital. Esse processo iterativo ajuda a aprimorar a eficácia dos *prompt* utilizados.

Portanto, com a aplicação cuidadosa de técnicas como a clareza nas solicitações, a definição de restrições explícitas e o refinamento iterativo, obtêm-se respostas mais precisas e alinhadas aos objetivos propostos. À medida que a experimentação com *prompt* continua a evoluir, novas estratégias e melhores resultados podem ser alcançados, aprimorando ainda mais o potencial dos modelos de NLP [Ekin 2023] [White et al. 2023].

2.5. Conclusão

Conclui-se que a tecnologia, aliada à inteligência artificial, tem desempenhado um papel transformador na educação, proporcionando novos métodos de ensino e avaliação que atendem de maneira mais eficaz às necessidades individuais dos alunos. Essas inovações não apenas tornam o ensino mais acessível, mas também permitem uma personalização que era inimaginável há algumas décadas. No entanto, apesar dos avanços, desafios como a dependência excessiva de algoritmos e a necessidade de preservar a interação humana na educação permanecem. Superar essas limitações exigirá uma abordagem equilibrada, que valorize tanto as inovações tecnológicas quanto a prática pedagógica tradicional. Essa integração harmoniosa é fundamental para que as tecnologias emergentes possam realmente cumprir seu potencial transformador no campo educacional, formando a base sobre a qual o presente trabalho foi desenvolvido.

3. Metodologia

O método deste trabalho combinou práticas ágeis com tecnologias específicas para frontend, backend e modelos de LLM, visando o desenvolvimento de um software robusto, escalável e adaptável. A escolha pela metodologia ágil foi motivada pela necessidade de iterar rapidamente e adaptar-se às mudanças nas demandas, garantindo uma entrega contínua de valor ao longo do desenvolvimento. Para construir a solução proposta, foram utilizadas as seguintes tecnologias: React para o frontend, Python com FastAPI para o backend e MongoDB para o gerenciamento de dados. O modelo GPT da OpenAI foi empregado para a geração automatizada de questões, utilizando prompts otimizados para garantir precisão e relevância nas respostas. O gerenciamento do projeto foi realizado na plataforma GitHub, que facilitou a colaboração e o versionamento contínuo do código. Além disso, testes unitários e de integração foram aplicados para assegurar a qualidade e a funcionalidade do sistema em todas as suas camadas.

3.1. Requisitos do Sistema

Esta seção descreve os requisitos funcionais e não funcionais definidos para o desenvolvimento do sistema de geração automatizada de questões de múltipla escolha.

3.1.1. Requisitos Funcionais

- **RF01:** O sistema deve permitir o login do usuário para garantir sessões seguras.
- **RF02:** O usuário deve ser capaz de gerar questões de múltipla escolha, fornecendo tema, nível de dificuldade e quantidade.
- **RF03:** O sistema deve permitir que os usuários editem o enunciado, alternativas ou excluam questões geradas.
- **RF04:** As questões e suas edições devem ser armazenadas em um banco de dados, permitindo acesso futuro e personalização.
- **RF05:** O usuário deve poder selecionar questões previamente geradas para compor uma avaliação, que será exportada em formato PDF.
- **RF06:** O sistema deve gerar o PDF e incluir um campo de gabarito.
- **RF07:** O backend deve validar se o tema das questões já foi gerado anteriormente e, se necessário, acionar o modelo de LLM para gerar novas questões.
- **RF08:** O sistema deve utilizar o modelo de LLM para gerar questões a partir dos dados fornecidos pelos usuários.

3.1.2. Requisitos Não Funcionais

- **RNF01:** O sistema deve ser eficiente na geração e renderização das questões e avaliações, utilizando o Virtual DOM do React para otimizar a performance no frontend.
- **RNF02:** O backend deve ser escalável para lidar com múltiplos usuários simultâneos e requisições de geração de questões, mantendo tempos de resposta adequados.
- **RNF03:** As sessões dos usuários devem ser protegidas por autenticação, garantindo que apenas o usuário tenha acesso às questões editadas.

- **RNF04:** O código do frontend deve ser modular, permitindo a reutilização de componentes React, enquanto o backend deve seguir boas práticas para facilitar a manutenção e extensões futuras.
- **RNF05:** O sistema deve estar disponível e funcionar de maneira confiável, minimizando tempo de inatividade.
- **RNF06:** A geração de PDFs deve ser compatível com diferentes sistemas operacionais e dispositivos.
- **RNF07:** A comunicação com a API do GPT deve ser robusta e otimizada para evitar latência nas respostas geradas.

3.2. Tecnologias Utilizadas

No desenvolvimento do software, a integração entre frontend, backend e banco de dados foi fundamental para criar uma aplicação robusta, escalável e interativa. Cada tecnologia desempenhou um papel crucial para garantir não apenas a funcionalidade do sistema, mas também a manutenção e a escalabilidade da solução.

3.2.1. Frontend

O frontend, responsável pela interface do usuário, foi desenvolvido utilizando o *framework* React, conhecido por facilitar a construção de interfaces dinâmicas e eficientes. A escolha por React deveu-se à sua capacidade de criar componentes reutilizáveis que gerenciam seu próprio estado, permitindo a modificação individual de um componente sem a necessidade de atualizar toda a página. Uma das principais características do React é o uso do Virtual DOM, que otimiza a atualização e renderização das interfaces, melhorando significativamente o desempenho da aplicação.

O diagrama na Figura 1 ilustra o funcionamento principal do frontend. Ao acessar a aplicação, o usuário é inicialmente direcionado para a página de login, garantindo a segurança e a individualidade das sessões de cada usuário. Após o login, o sistema oferece duas principais funcionalidades: geração de questões e geração de avaliações.

Na geração de questões, o usuário fornece informações como o tema, nível de dificuldade e quantidade de questões desejadas. A partir desses dados, o frontend faz uma requisição ao backend, que utiliza o modelo de LLM para gerar as questões. As questões geradas são então exibidas em uma interface de edição, onde o usuário pode ajustar o cabeçalho, editar as alternativas ou excluir questões. Após as edições, as questões são salvas ou deletadas conforme a escolha do usuário.

Na geração de avaliações, o usuário seleciona questões previamente geradas para compor uma avaliação que pode ser exportada em formato PDF. O PDF gerado inclui um cabeçalho personalizável e um campo de gabarito para que os alunos preencham suas respostas.

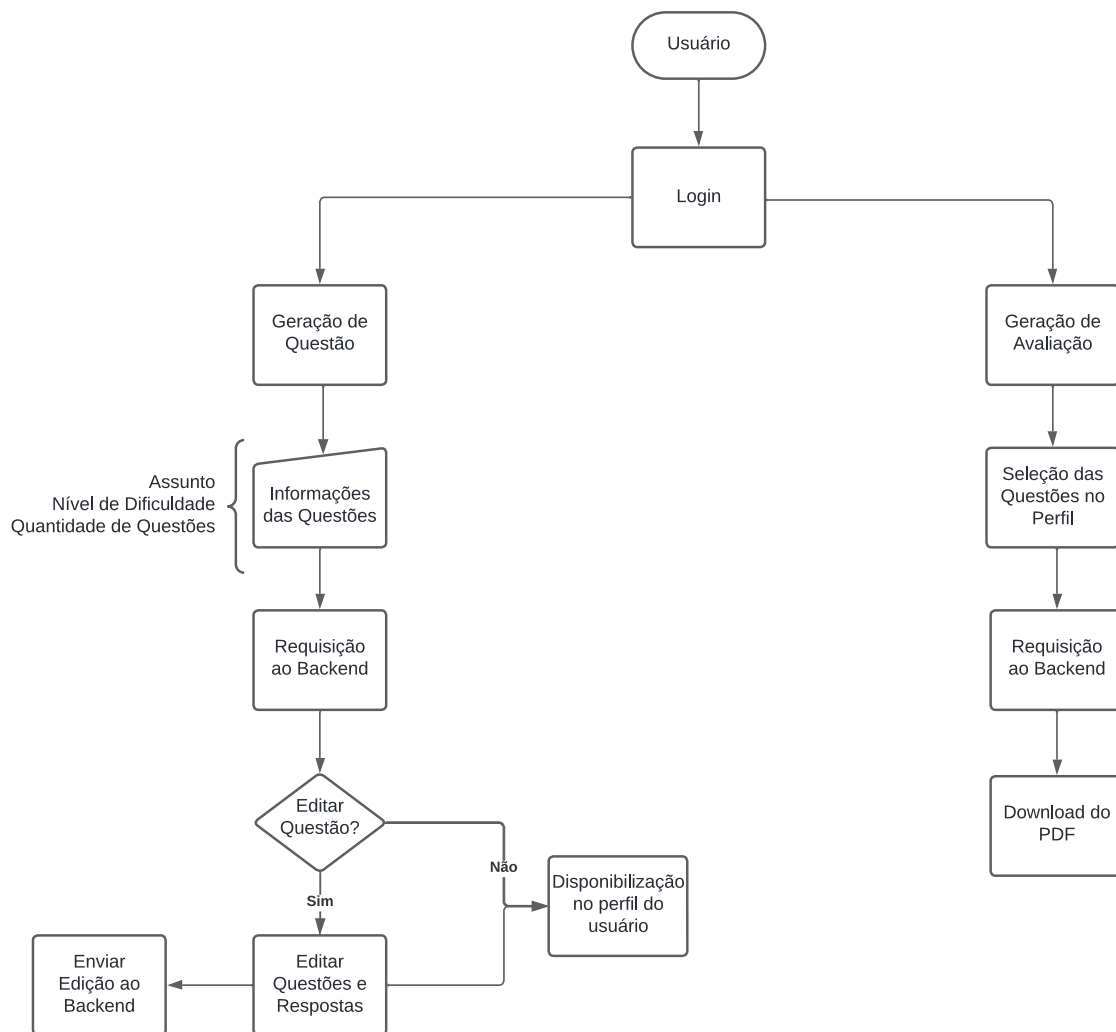


Figura 1. Diagrama do Frontend.

3.2.2. Backend

O backend, responsável pelo gerenciamento principal do sistema, foi desenvolvido utilizando Python em conjunto com a *framework* FastAPI. Python foi escolhido por sua versatilidade, simplicidade e grande variedade de bibliotecas, enquanto o FastAPI foi selecionado por permitir a criação de APIs robustas e seguras com facilidade. O MongoDB foi utilizado como sistema gerencial de banco de dados devido à sua estrutura não relacional, que facilitou flexibilidade no armazenamento dos dados, assim, permitindo que as questões e suas alternativas possam ser salvas em diferentes formatos.

A Figura 2 apresenta o diagrama de funcionamento do backend. O backend gerencia todas as requisições feitas pelo frontend, processando os dados e integrando-os ao banco de dados para garantir o funcionamento completo do sistema.

Na geração de questões, o backend valida se o assunto inserido já foi gerado anteriormente, e verifica quantas questões ainda precisam ser geradas, assim permitindo uma dinâmica maior para cada usuário e evitando a utilização do modelo em toda solicitação. Se existir e não tiver sido criada pelo mesmo usuário, as questões é retornada; caso

contrário, o modelo de LLM é acionado para gerar novas questões. As edições feitas pelo usuário são armazenadas em um banco de dados individual, garantindo que apenas o usuário tenha acesso a essas versões editadas.

Para a geração de avaliações, o backend compila as questões selecionadas e cria um arquivo PDF utilizando LaTeX. Este arquivo inclui um cabeçalho, um campo de gabarito e as questões selecionadas, que são então retornadas ao frontend para download pelo usuário.

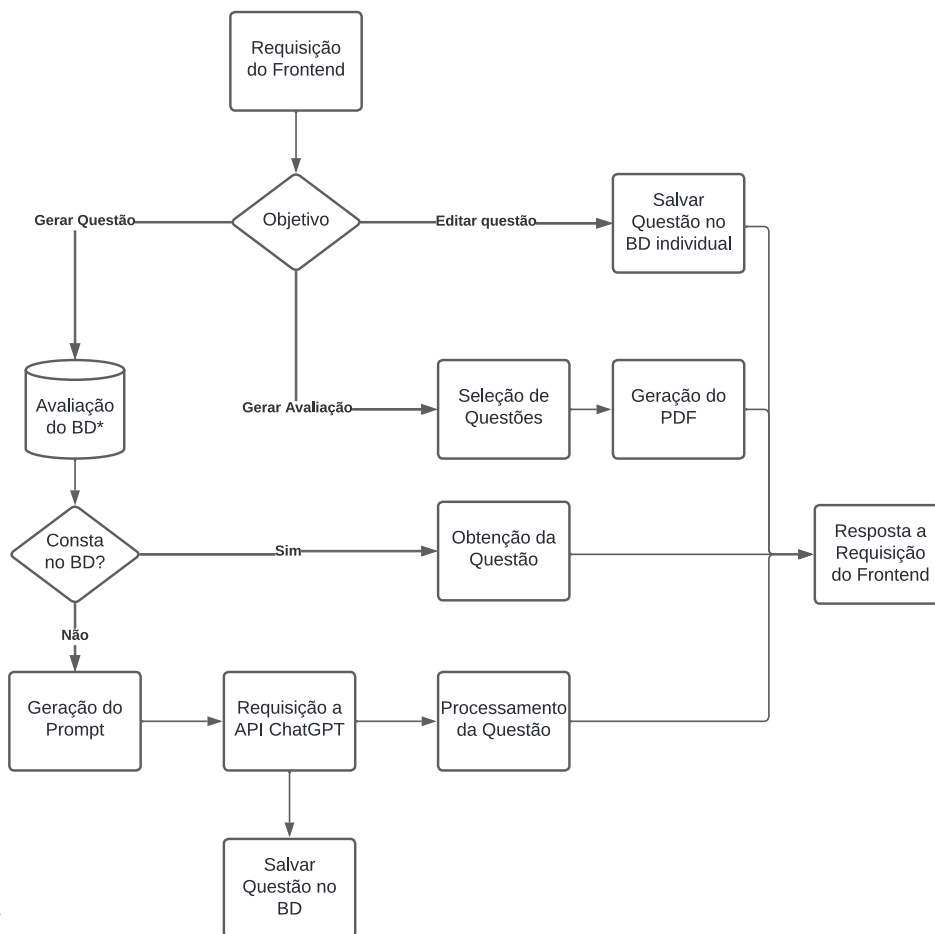


Figura 2. Diagrama do Backend.

3.2.3. Modelo de LLM

O modelo de *Large Language Models* (LLM) utilizado neste sistema foi o GPT da OpenAI devido a facilidade de utilização e reconhecimento no mercado, que emprega *deep learning* e processamento de linguagem natural (PLN) para compreender, processar e gerar texto em linguagem natural de forma coerente. A integração do GPT ao sistema foi realizada via API, permitindo que as requisições de texto fossem enviadas e processadas de maneira eficiente.

A construção de *prompts* específicos foi essencial para otimizar a precisão das respostas geradas pelo modelo de LLM. Um *prompt* bem estruturado faz a diferença na qua-

lidade das questões geradas, minimizando erros e inconsistências. O exemplo de *prompt* a seguir foi utilizado para a geração de questões de múltipla escolha, sendo adaptado conforme as informações fornecidas pelo usuário, como o tema e o nível de dificuldade.

No exemplo abaixo, os campos em caixa alta representam as informações fornecidas pelo usuário:

1. N: Número de questões.
2. TEMA: Assunto da geração de questões.
3. DIFICULDADE: Nível de dificuldade da questão.

```
Gere N questão(ões) de múltipla escolha sobre TEMA.
Utilize os itens em {EXPLICAÇÃO} e estructure a
resposta a partir da {ESTRUTURA} seguindo as {REGRAS}.

{EXPLICAÇÃO}
[Enunciado]: Objetivo da questão, detalhando informações
complementares e apresentando o problema de forma objetiva.
[Alternativas]: Possíveis respostas para o enunciado,
podendo conter detalhes do assunto objetivo ou informações
incorretas mas plausíveis (em caso de resposta errada).
[Respostas]: Lista com as alternativas corretas.

{ESTRUTURA}
[ {
  "enunciado": [Enunciado],
  "alternativa_1": [Alternativas],
  "alternativa_2": [Alternativas],
  "alternativa_3": [Alternativas],
  "alternativa_4": [Alternativas],
  "alternativa_5": [Alternativas],
  "respostas": [Respostas]
} ]

{REGRAS}
> O [Enunciado] deve ser objetivo.
> A complexidade da questão deve ser DIFICULDADE em uma
escala de 1 a 5, sendo 1 questões introdutórias e 5
questões avançadas.
> Cada questão deve haver 1 [Enunciado] e 5 [Alternativas].
> A resposta pode haver 1 ou mais alternativas corretas. Em
caso de impossibilidade na geração da questão, retorne a
{ESTRUTURA} preenchida de strings vazias.
> Siga a {ESTRUTURA} acima e substitua os elementos conforme
a inserção das questões utilizando os itens em {EXPLICAÇÃO}.
```

3.3. Etapas do Desenvolvimento

O desenvolvimento da aplicação *Questfy* foi estruturada em etapas bem definidas e interdependentes, assegurando um progresso contínuo e alinhado com os objetivos do projeto.

A implementação foi guiada por uma metodologia ágil, que permitiu uma abordagem iterativa e incremental. Cada fase do desenvolvimento foi cuidadosamente planejada e executada, desde a definição do escopo até a finalização dos testes, garantindo que todos os requisitos funcionais e de desempenho fossem plenamente atendidos. Esta subseção descreve detalhadamente o processo de desenvolvimento, que se desdobrou em cinco grandes etapas: planejamento, desenvolvimento do modelo de LLM, implementação do backend, construção do frontend e a fase de testes. A seguir, cada uma dessas etapas é apresentada, destacando as atividades realizadas, os desafios enfrentados e as soluções adotadas para assegurar a qualidade e a eficiência do sistema final.

3.3.1. Metodologia de Desenvolvimento

A metodologia ágil foi adotada neste projeto devido à sua capacidade de promover a adaptabilidade, a colaboração e o progresso iterativo no desenvolvimento de software. A abordagem ágil priorizou ciclos de desenvolvimento curtos e flexíveis, que permitiram ajustes rápidos e frequentes. Isso assegurou a entrega contínua de valor ao longo do ciclo de vida do projeto.

Para a gestão do projeto, foi utilizada a metodologia Scrum, que organizou a comunicação e promoveu a responsabilidade colaborativa entre os membros da equipe. O Scrum foi implementado com o auxílio da plataforma GitHub, que, com sua funcionalidade de *projects*, permitiu o gerenciamento eficiente do cronograma, das atividades e das prioridades, garantindo um fluxo contínuo e organizado no desenvolvimento.

3.3.2. Desenvolvimento

O desenvolvimento do sistema seguiu um processo iterativo, estruturado em cinco etapas principais: planejamento, desenvolvimento do modelo de LLM, backend, frontend e testes. Cada uma dessas etapas foi meticulosamente abordada para garantir que o sistema final atendesse aos requisitos funcionais e de desempenho estabelecidos. A Figura 3 apresenta um cronograma detalhado, que ilustra as etapas do projeto e os respectivos períodos de desenvolvimento.

O projeto inicia-se com a Definição de Escopo e Tecnologias na Sprint 1, cujo foco é estabelecer de forma clara o que será desenvolvido e quais tecnologias serão empregadas. Paralelamente, ainda na Sprint 1, ocorre a Definição da Metodologia, onde a equipe escolhe o modelo de trabalho a ser seguido ao longo do projeto, assegurando uma abordagem coesa e eficiente.

Na sequência, o desenvolvimento do Modelo de LLM começa com a Definição do Modelo de LLM a partir da Sprint 1, estendendo-se até a Sprint 5. Durante esse período, a equipe realiza estudos aprofundados para identificar as melhores práticas e configurações para o modelo a ser utilizado. A partir da Sprint 4 até a Sprint 6, os esforços se concentram em Estudos sobre Prompt Engineering, explorando técnicas que otimizam a interação com o modelo de LLM. Em paralelo, inicia-se o Desenvolvimento e Validação de Prompts, atividade que se estende da Sprint 4 a Sprint 9, com o objetivo de criar prompts eficazes e validar sua aplicabilidade. Na Sprint 6, tem início o Aperfeiçoamento de Prompts,

concluído na Sprint 8, refinando os prompts criados para obter melhores resultados. Da Sprint 7 a Sprint 11, a equipe trabalha na Aplicação da API da OpenAI, integrando o modelo de LLM ao projeto através da API. Por fim, nas Sprints 15 a 18, uma nova fase de Teste e Validação de Prompts é realizada, revisando os prompts desenvolvidos para assegurar sua eficácia.

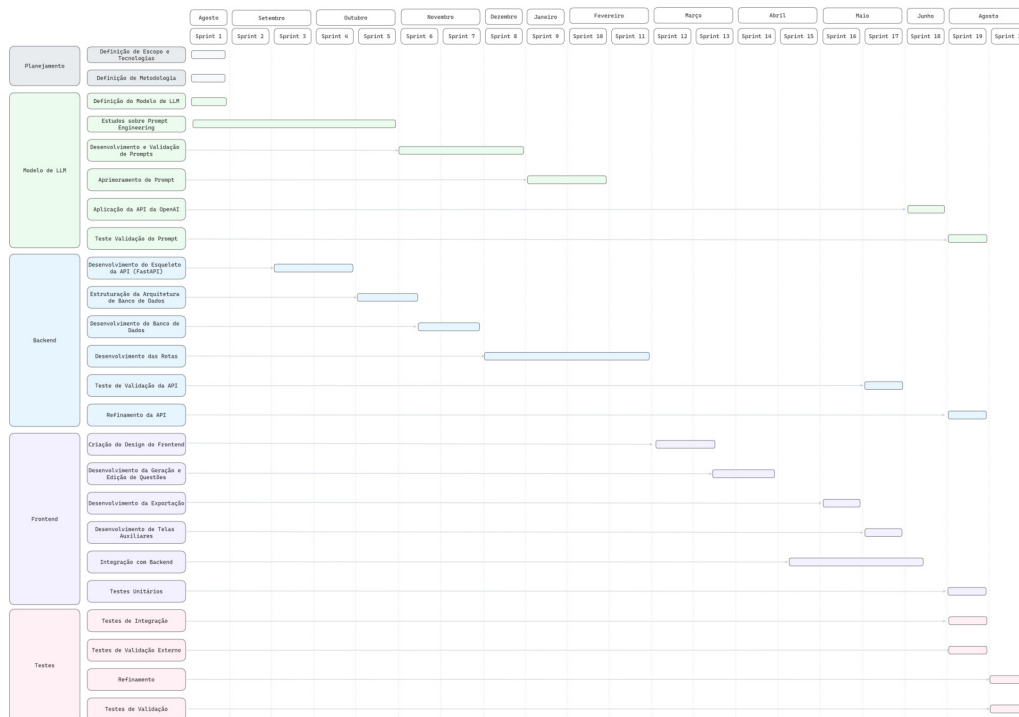


Figura 3. Cronograma de Desenvolvimento.

O desenvolvimento do Backend inicia-se na Sprint 4 com a criação do Esqueleto da API utilizando FastAPI, estendendo-se até a Sprint 7. Simultaneamente, durante os Sprints 5 a 7, é realizada a Estruturação da Arquitetura do Banco de Dados, estabelecendo a base para o armazenamento e gerenciamento de dados. A partir da Sprint 6 até a Sprint 8, ocorre o Desenvolvimento do Banco de Dados, com a criação e otimização dos documentos no MongoDB e suas coleções. O Desenvolvimento das Rotas inicia-se na Sprint 8 e prossegue até a Sprint 12, com a implementação das rotas necessárias para a comunicação entre o frontend e o backend. Nas Sprints 9 a 14, são conduzidos Testes de Validação da API, garantindo que todas as funcionalidades da API operem corretamente. A fase de Refinamento da API ocorre nas Sprints 12 a 17, ajustando e otimizando a API com base nos feedbacks obtidos durante os testes.

O desenvolvimento do Frontend começa no Sprint 10 com a Criação do Design do Frontend, onde é definida a interface visual a ser utilizada no projeto, finalizando essa etapa na Sprint 12. Em paralelo, das Sprints 13 a 16, ocorre o Desenvolvimento da Geração e Edição de Questões, complementado pelo Desenvolvimento da Exportação no mesmo período, criando funcionalidades essenciais para o gerenciamento de questões no sistema. O Desenvolvimento de Telas Auxiliares é realizado das Sprints 13 a 17, criando componentes adicionais para complementar a interface do usuário. A partir da Sprint 16 até a Sprint 19, ocorre a Integração com Backend, garantindo uma comunicação

eficaz entre as partes. Nas Sprints 17 a 19, é realizado Testes Unitários, verificando o funcionamento individual de cada componente do frontend para assegurar sua correta operação.

Por fim, a fase de Testes intensifica-se nas Sprints 18 a 20 com Testes de Integração, validando o funcionamento conjunto de todos os componentes do sistema. Na Sprint 20, são realizados Testes de Validação Externa, avaliando a estrutura do sistema desenvolvido. Em paralelo, a Sprint 20 é também dedicado ao Refinamento, onde ajustes finais são feitos com base nos resultados dos testes anteriores. O projeto é concluído com os Testes de Validação, assegurando que todas as funcionalidades atendem aos requisitos definidos e que o sistema está pronto para entrega.

3.4. Testes e Validações

A realização de testes foi uma etapa crucial no desenvolvimento do software, com o objetivo de avaliar o desempenho do sistema e identificar erros tanto no desenvolvimento quanto na estrutura. No desenvolvimento do sistema, foram aplicados testes estruturais para validar o funcionamento do sistema, divididos em testes unitários e testes de integração.

Os testes unitários foram direcionados à avaliação de partes isoladas do código, verificando cada funcionalidade individualmente para assegurar seu correto funcionamento de forma independente do restante do sistema. Esses testes foram conduzidos manualmente pelos desenvolvedores após a conclusão de cada componente do sistema.

Os testes de integração, por outro lado, tiveram o objetivo de avaliar o sistema como um todo, após a integração das diversas partes. Esses testes verificaram o comportamento conjunto do sistema para garantir que ele se comportasse conforme o esperado. Os testes de integração foram realizados manualmente pelos desenvolvedores e por alguns usuários de teste, validando o funcionamento correto do sistema em sua totalidade. É importante citar que os usuários testes avaliaram o funcionamento do sistema e seu design, necessitando de uma etapa de teste mais robusta juntamente aos usuários finais (educadores).

4. Resultados

Nesta seção, detalhamos os resultados obtidos com o desenvolvimento do software *Questfy*, uma ferramenta projetada para automatizar a criação e permitir a edição, gerenciamento e exportação de questões de múltipla escolha. A seguir, descrevemos cada uma das principais funcionalidades do sistema, exemplificando seu uso e destacando os benefícios que elas oferecem aos usuários.

4.1. Criação e Edição de Questões

Uma das principais funcionalidades do *Questfy* é a criação automatizada de questões de múltipla escolha, facilitada pelo uso de um modelo de *Large Language Model* (LLM). Para ilustrar essa funcionalidade, consideramos o seguinte exemplo: **Quantidade de perguntas = 3, Dificuldade = Médio e Assunto = "Como calcular o desempenho de modelos de forecasting?"**.

A Figura 4 mostra a interface onde o usuário insere esses parâmetros. Uma vez que as informações são preenchidas, o usuário clica no botão *Gerar*, que aciona uma requisição ao backend, onde os dados são processados pelo modelo de LLM.

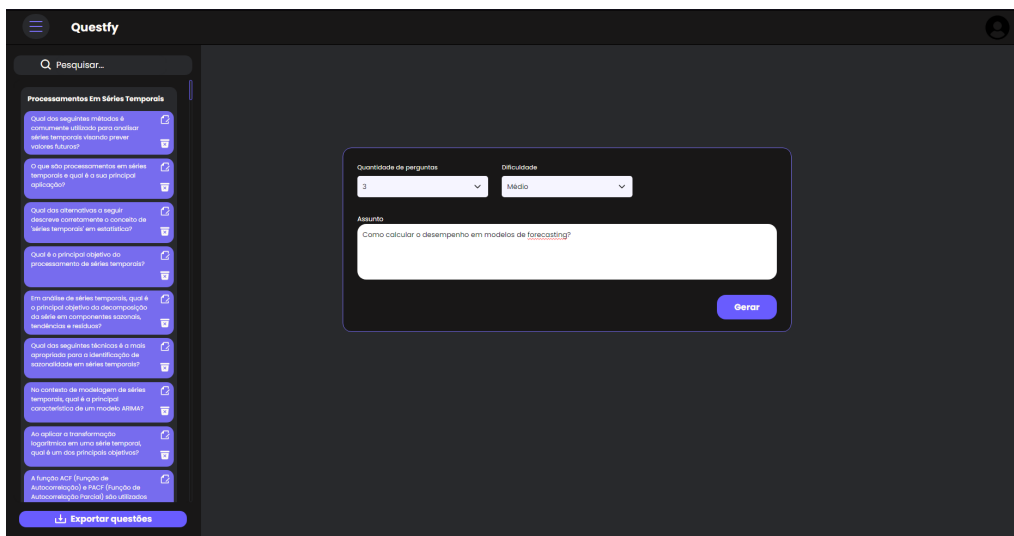


Figura 4. Tela de Definição dos Parâmetros para Geração das Questões.

Após o processamento, o sistema retorna um conjunto de questões geradas automaticamente, apresentadas ao usuário na tela de edição, conforme mostrado na Figura 5. Nesta interface, cada questão é representada por uma aba, contendo o enunciado e cinco alternativas de resposta. Ao lado de cada alternativa, um seletor de cores indica se a resposta é correta (verde) ou incorreta (vermelho).

Essa tela de edição permite ao usuário revisar as questões geradas e realizar modificações conforme necessário. As funcionalidades disponíveis incluem:

- **Editar:** O usuário pode ajustar o enunciado, modificar as alternativas e definir quais são as corretas.
- **Descartar:** Caso uma questão não atenda às expectativas, o usuário pode marcá-la para exclusão, mudando a cor da aba para vermelho. A questão pode ser restaurada se necessário.
- **Concluir:** Ao final da revisão, o usuário confirma as alterações, salvando as questões no sistema.

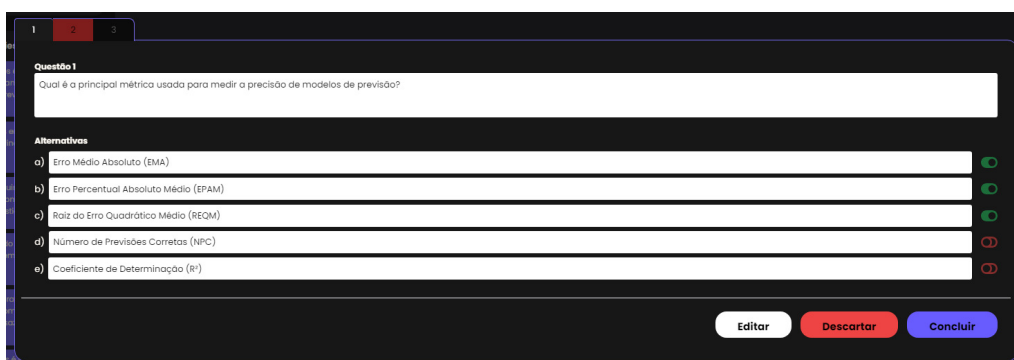


Figura 5. Tela de Edição Após a Geração das Questões.

A Figura 6 ilustra o processo de exclusão de uma questão. Quando uma questão é marcada para exclusão, o botão “Descartar” é substituído por “Restaurar”, permitindo que o usuário reverta a exclusão antes de concluir a edição.

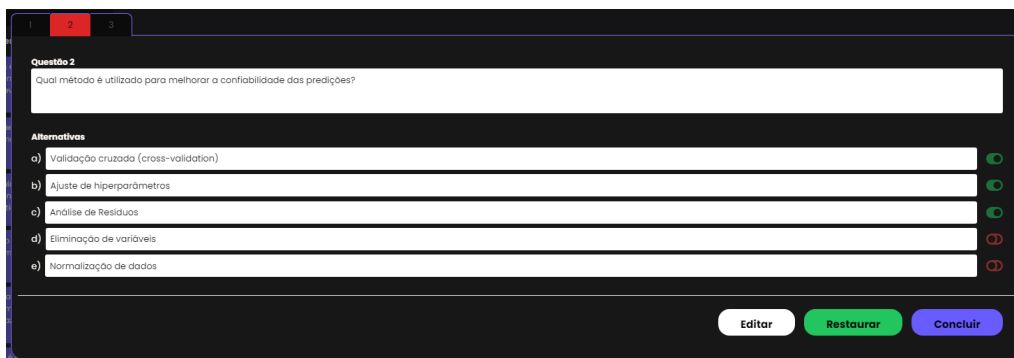


Figura 6. Tela de Edição ao Deletar uma Questão.

Essa funcionalidade de criação e edição é crucial para educadores que buscam personalizar as avaliações de acordo com o nível de dificuldade e o conteúdo desejado, oferecendo flexibilidade e melhor controle sobre as questões geradas.

4.2. Gerenciamento de Questões

O *Questfy* não apenas gera e edita questões, mas também permite o gerenciamento eficiente de um banco de questões. Após a criação e edição, todas as questões são armazenadas em uma base de dados organizada por temas, níveis de dificuldade e outras categorias definidas pelo usuário.

A Figura 7 mostra a barra lateral onde as questões armazenadas são exibidas. Essa interface oferece filtros de pesquisa que permitem ao usuário localizar rapidamente as questões desejadas, facilitando o gerenciamento de grandes volumes de conteúdo. As questões são organizadas por assunto e podem ser selecionadas para edição ou exclusão.

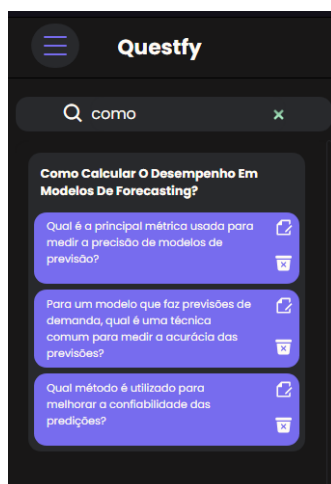


Figura 7. Barra Lateral de Exibição de Questões: Filtro de Pesquisa.

Ao lado de cada questão na lista, há dois botões principais:

- **Editar:** Permite ao usuário acessar a tela de edição para modificar a questão, conforme mostrado na Figura 8.
- **Excluir:** Remove a questão do banco de dados do usuário, garantindo que apenas as questões relevantes sejam mantidas.

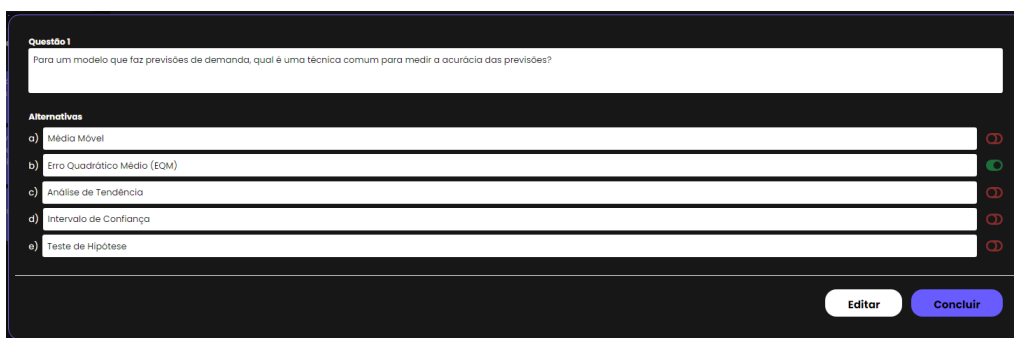


Figura 8. Tela de Edição de Questões.

Esse sistema de gerenciamento é essencial para educadores que precisam manter um banco de questões organizado e acessível, especialmente em contextos onde diferentes avaliações são criadas ao longo do tempo.

4.3. Exportação de Questões em PDF

A exportação das questões para um formato de fácil distribuição é uma das funcionalidades mais valiosas do *Questfy*. Após selecionar as questões desejadas, o usuário pode exportá-las em formato PDF, pronto para ser utilizado em provas e avaliações formais.

Na parte inferior da barra lateral, conforme mostrado na Figura 9, encontra-se o botão para exportação. Ao clicar nele, o usuário é direcionado para a tela de seleção de questões, onde pode escolher quais questões deseja incluir no PDF final.

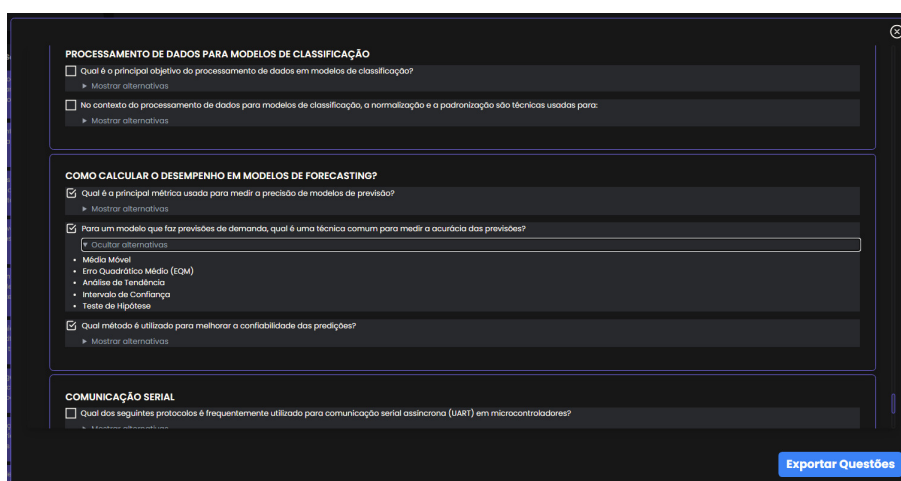


Figura 9. Tela de Seleção de Questões para Exportação.

Depois de selecionar as questões, o sistema gera um PDF, que inclui um cabeçalho personalizável com o nome do avaliador e do aluno, um gabarito e todas as questões selecionadas. A Figura 10 mostra um exemplo do PDF gerado.

Essa funcionalidade facilita a distribuição das avaliações, garantindo que o educador tenha uma ferramenta eficiente para gerar provas e outros materiais didáticos com rapidez e qualidade.

Professor:

Aluno(a): _____

Gabarito

1	a	b	c	d	e
2	a	b	c	d	e
3	a	b	c	d	e

- Qual é a principal métrica usada para medir a precisão de modelos de previsão?
 - Erro Médio Absoluto (EMA)
 - Erro Percentual Absoluto Médio (EPAM)
 - Raiz do Erro Quadrático Médio (REQM)
 - Número de Previsões Corretas (NPC)
 - Coefficiente de Determinação (R^2)
- Para um modelo que faz previsões de demanda, qual é uma técnica comum para medir a acurácia das previsões?
 - Média Móvel
 - Erro Quadrático Médio (EQM)
 - Análise de Tendência
 - Intervalo de Confiança
 - Teste de Hipótese
- Qual método é utilizado para melhorar a confiabilidade das predições?
 - Validação cruzada (cross-validation)
 - Ajuste de hiperparâmetros
 - Análise de Resíduos
 - Eliminação de variáveis
 - Normalização de dados

Figura 10. PDF Gerado na Exportação das Questões.

5. Conclusão

A transformação digital no setor educacional tem gerado novos desafios e oportunidades, especialmente no que diz respeito à adaptação de processos tradicionais ao ambiente online. Este trabalho contribui para o enfrentamento dessas questões com o desenvolvimento do *Questfy*, uma ferramenta inovadora voltada para a geração automatizada de questões de múltipla escolha (QMEs). A solução proposta atende à demanda por uma aplicação que simplifique a criação de avaliações, utilizando as questões geradas de forma eficiente e precisa, proporcionando maior agilidade e qualidade no contexto do ensino remoto.

A revisão da literatura destaca a necessidade de ferramentas que não apenas facilitam a criação de conteúdos, mas também oferecem flexibilidade e personalização. O *Questfy* preenche essa lacuna, combinando o poder dos *Large Language Models* (LLMs) com uma interface intuitiva e funcionalidades que permitem aos educadores adaptar as questões às suas necessidades pedagógicas específicas.

Os resultados demonstram que o *Questfy* é eficaz em automatizar a geração de

questões, ao mesmo tempo em que oferece a flexibilidade necessária para personalizar as avaliações. Essa combinação de automação e personalização melhora significativamente o processo de criação de questões, economizando tempo e esforço dos educadores e elevando a qualidade das avaliações.

5.1. Limitações e Trabalhos Futuros

Embora o sistema desenvolvido neste trabalho tenha se mostrado eficaz em sua proposta, ele apresenta algumas limitações que restringem sua aplicabilidade em sua forma atual. A geração de questões de múltipla escolha, por exemplo, exibe as alternativas corretas diretamente na interface, exigindo a avaliação e correção manual por parte do educador. A atividade de correção e atribuição de notas, no entanto, pode ser automatizada por meio de uma aplicação móvel, o que facilitaria significativamente o seu uso e aumentaria a eficiência do processo.

Além disso, é importante considerar a expansão do sistema para permitir a geração de outros formatos de avaliação, além das questões de múltipla escolha. Isso possibilitaria a criação de avaliações mais robustas e completas, atendendo a uma maior diversidade de necessidades pedagógicas. A personalização avançada das avaliações, permitindo ao usuário estruturar e customizar o conteúdo de maneira mais flexível, também é uma melhoria relevante que poderia ser incorporada.

Outro ponto de alta relevância é a realização de estudos de caso em larga escala para validar a aplicação em diferentes contextos educacionais. A partir desses estudos, seria possível ajustar o sistema com base no feedback dos usuários, corrigindo eventuais falhas e aprimorando a aplicação para torná-la mais adequada e confortável para o uso diário.

Com essas melhorias, o *Questfy* tem o potencial de se tornar uma plataforma essencial para a educação digital, facilitando a criação de avaliações de alta qualidade e adaptando-se continuamente às necessidades de uma educação em constante evolução.

5.2. Lições Aprendidas

O desenvolvimento da aplicação *Questfy* proporcionou uma série de aprendizados, desafios e oportunidades que contribuíram significativamente para o aprimoramento de nossas habilidades em cada etapa deste projeto.

Um dos aspectos que proporcionou maior aprendizado foi o *prompt engineering*, que demonstrou ser crucial para a obtenção de resultados mais completos e precisos em relação às informações desejadas. A correta estruturação do *prompt* não apenas organiza a resposta, mas também permite a personalização dos detalhes e a qualidade do conteúdo gerado. No entanto, é importante destacar que não existe um *prompt* universal; é necessário desenvolver um *prompt* personalizado que se adéque especificamente ao problema em questão.

A criação do design e o desenvolvimento do frontend foram as atividades que apresentaram maior grau de dificuldade, em parte devido à falta de familiaridade não apenas com as ferramentas e tecnologias utilizadas, mas também com o processo criativo e a validação funcional de uma interface que fosse intuitiva e fluida. Para superar esses desafios, foi necessário realizar estudos complementares, além de buscar avaliações de

terceiros acerca do design e fluidez do sistema, o que resultou em uma interface simples, mas funcional.


Após uma avaliação cuidadosa de todas as etapas, concluímos que uma modificação no planejamento e na estrutura de desenvolvimento poderia ter contribuído para um progresso mais fluido e para uma melhor gestão dos problemas enfrentados. A antecipação das etapas de design e do desenvolvimento inicial do frontend para o início do projeto teria possibilitado mais tempo para estudos e ajustes de funcionamento, ao mesmo tempo em que o desenvolvimento de certas partes do backend poderia ser postergado, já que a familiaridade dos autores com essa tecnologia reduziria o impacto no resultado final.

Além disso, a inclusão de um período de duas sprints dedicado exclusivamente ao planejamento poderia ter auxiliado na definição completa de todos os pontos a serem abordados na aplicação, detalhando a prioridade no desenvolvimento de cada item.

Referências

- Abdulmunem, R. A. (2023). Artificial intelligence in education. In *Advances in Mobile and Distance Learning*, pages 241–255. IGI Global.
- Alkan, A. (2024). Artificial intelligence: Its role and potential in education. *İnsan ve Toplum Bilim. Araştırmaları Derg.*, 13(1):483–497.
- Biggs, J., Tang, C., and Kennedy, G. (2022). *Teaching for quality learning at university 5e*. McGraw-hill education (UK).
- Brown, G. T. L. and Abdulnabi, H. H. A. (2017). Evaluating the quality of higher education instructor-constructed multiple-choice tests: Impact on student grades. *Front. Educ.*, 2.
- Ciobanu, N. R., Barth, K. M., and Florescu, M. C. (2023). Adapting the education and training system to technological evolution. *Educ. 21*, (26):28–33.
- Dancsa, D., Štempeľová, I., Takáč, O., and Annuš, N. (2023). Digital tools in education. *International Journal of Advanced Natural Sciences and Engineering Researches*, 7(4):289–294.
- Ekin, S. (2023). Prompt engineering for ChatGPT: A quick guide to techniques, tips, and best practices.
- Harry, A. (2023). Role of AI in education. *Injury*, 2(3):260–268.
- Huang, A., Hancock, D., Clemson, M., Yeo, G. C., Harney, D. J., Denny, P., and Denyer, G. (2020). Selecting student-authored questions for summative assessments. *bioRxiv*.
- Joshi, S. (2023). TECHNOLOGY IN EDUCATION. *VIDYA - A JOURNAL OF GUJARAT UNIVERSITY*, 2(2):3–5.
- Josué, A., Bedoya-Flores, M. C., Mosquera-Quiñonez, E. F., Mesías-Simisterra, Á. E., and Bautista-Sánchez, J. V. (2023). Educational platforms: Digital tools for the teaching-learning process in education. *Ibero-American Journal of Education & Society Research*, 3(1):259–263.
- Kudryashov, A. A., Gostev, D. V., and Satskaya, A. A. (2024). The development of digitalization in the field of education. *Industrial Economics*, (1):157–161.
- Kurdi, G., Leo, J., Parsia, B., Sattler, U., and Al-Emari, S. (2020). A systematic review of automatic question generation for educational purposes. *Int. J. Artif. Intell. Educ.*, 30(1):121–204.
- Lievertz, M. (2019). Artificial intelligence in education. In *Artificial Intelligence and Machine Learning for Business for Non-Engineers*, pages 125–140. CRC Press.
- Lima, G. P. T., Melo, M. C. d., Oliveira, E. S. d., Matos, M. D. D., Araújo, R. H. P. d., and Demuner, J. A. (2024). Artificial intelligence in education. In *Academic Education Navigating the Path of Knowledge*. Seven Editora.
- Md Din, A. G. B., bin Zabidin, M. A. R., Moawad Ali Tahawi, A. R., Md Din, O. B., and Aawi, F. (2023). Educational assessment: Its types and outcomes: A bibliometric analytical study. *Int. J. Acad. Res. Bus. Soc. Sci.*, 13(11).

- Nadeem, M., Rafiq, M. A., and Jameel, K. (2024). The role of educational technology in academia. In *Advances in Information Quality and Management*, pages 1–12. IGI Global.
- Sadiku, M. N. O., Ashaolu, T. J., Ajayi-Majebi, A., and Musa, S. M. (2021). Artificial intelligence in education. *International Journal Of Scientific Advances*, 2(1).
- Sani, J., Kamal, M. M., Biswas, T. K., Chowdhury, S., Roy, S., and Sarwar, S. (2024). Technologies used in education. *International Journal For Multidisciplinary Research*, 6(3).
- Vásquez-Pajuelo, L., Rodriguez-Barboza, J. R., Bartra-Rivero, K. R., Quintanilla-Alarcón, E. A., Vega-Jaime, W., and Chavarri-Joo, E. F. (2024). Digital challenges: The need to improve the use of information technologies in teaching. *Data and Meta-data*, 3:216.
- White, J., Hays, S., Fu, Q., Spencer-Smith, J., and Schmidt, D. C. (2023). Chatgpt prompt patterns for improving code quality, refactoring, requirements elicitation, and software design.

	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
	Campus Campina Grande - Código INEP: 25137409
	R. Tranquílino Coelho Lemos, 671, Dinamérica, CEP 58432-300, Campina Grande (PB)
	CNPJ: 10.783.898/0003-37 - Telefone: (83) 2102.6200

Documento Digitalizado Ostensivo (Público)

Versão Final do TCC

Assunto:	Versão Final do TCC
Assinado por:	Jhonatan Oliveira
Tipo do Documento:	Projeto
Situação:	Finalizado
Nível de Acesso:	Ostensivo (Público)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- **Jhonatan Guilherme Oliveira Pereira**, ALUNO (201921250012) DE BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO - CAMPINA GRANDE, em 07/10/2024 16:47:21.

Este documento foi armazenado no SUAP em 07/10/2024. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1269617

Código de Autenticação: 9ccae5ce1f

