



Instituto Federal de Educação Ciências e Tecnologia da Paraíba
Campus Campina Grande
Coordenação do Curso Superior de Bacharelado em
Engenharia de Computação

QuestõesPro: Uma plataforma de elaboração e correção de avaliações

Jonas Henriques Nascimento
Jorge Victor Leite Damasceno

Campina Grande
2023



Instituto Federal de Educação Ciências e Tecnologia da Paraíba
Campus Campina Grande
Coordenação do Curso Superior de Bacharelado em
Engenharia de Computação

Jonas Henriques Nascimento
Jorge Victor Leite Damasceno

QuestõesPro: Uma plataforma de elaboração e correção de avaliações

Trabalho de conclusão de curso apresentado à Coordenação do Curso Superior de Bacharelado em Engenharia de Computação do IFPB - Campus Campina Grande, como requisito parcial para a conclusão do curso de Bacharelado em Engenharia de Computação.

Orientador: Dr. Elmano Ramalho Cavalcanti.

Campina Grande
2023

N244q Nascimento, Jonas Henriques.

Questõespro: uma plataforma de elaboração e correção de avaliações / Jonas Henriques Nascimento, Jorge Victor Leite Damasceno. - Campina Grande, 2023.
21 f. : il.

Trabalho de Conclusão de Curso (Curso Superior de Bacharelado em Engenharia de Computação) - Instituto Federal da Paraíba, 2023.

Orientador: Prof. Dr. Elmano Ramalho Cavalcanti.

1.Plataforma de aplicação 2. Questõespro 3.
Desenvolvimento de sistemas. I. Damasceno, Jorge Victor Leite
II.Cavalcanti, Elmano Ramalho. III.Título.

CDU 004

Jonas Henriques Nascimento
Jorge Victor Leite Damasceno

QuestõesPro: Uma plataforma de elaboração e correção de avaliações

Trabalho de conclusão de curso apresentado à Coordenação do Curso Superior de Bacharelado em Engenharia de Computação do IFPB - Campus Campina Grande, como requisito parcial para a conclusão do curso de Bacharelado em Engenharia de Computação.

Orientador: Dr. Elmano Ramalho Cavalcanti.

Elmano Ramalho Cavalcanti
Orientador

Avaliador 1
Membro da banca

Avaliador 2
Membro da banca

Campina Grande
2023

QuestõesPro: Uma Plataforma de Elaboração e Correção de Avaliações

Jonas Henriques Nascimento, Jorge Victor Leite Damasceno

Curso de Bacharelado em Engenharia de Computação – Instituto Federal da Paraíba (IFPB) Campus Campina Grande - PB

{jonas.nascimento, jorge.victor}@academico.ifpb.edu.br

Abstract. *Student assessment, often done through multiple-choice tests, is a complex and exhausting activity, which requires countless hours of dedication. QuestõesPro is a web application platform developed to minimize the time spent by teachers on assessments and their applications in the classroom. In order to provide more functionalities and improve the user experience, this work aims to continue the system development process. With emphasis on the functionalities of importing questions from an assessment into a text document and exporting the assessment to an editable document. Despite the difficulties throughout development, the planned functionalities were implemented following the requirements thought at the beginning of the project.*

Resumo. *A avaliação dos alunos, muitas vezes feita por meio de provas de múltipla escolha, é uma atividade complexa e desgastante, que exige inúmeras horas de dedicação. O QuestõesPro é uma plataforma de aplicação web desenvolvida para minimizar o tempo gasto pelo professor nas avaliações e suas aplicações em sala de aula. Com o intuito de prover mais funcionalidades e melhorar a experiência com o usuário, esse trabalho tem como objetivo dar continuidade ao processo de desenvolvimento do sistema. Com ênfase para as funcionalidades de importação de questões de uma avaliação em um documento de texto (e.g. docx) e exportação da avaliação para documento editável. Apesar das dificuldades ao longo do desenvolvimento, as funcionalidades planejadas foram implementadas seguindo os requisitos pensados no início do projeto.*

1. Introdução

Uma pesquisa realizada entre julho e dezembro de 2022 pelo instituto Inteligência em Pesquisa e Consultoria Estratégica (IPEC) ouviu mais de seis mil professores de escolas públicas municipais e estaduais de todo o país, apontando que mais de 70% dos professores brasileiros estão estressados pela sobrecarga de trabalho (AGÊNCIA BRASIL, 2023). Entre as tarefas que exigem tempo e esforço consideráveis dos docentes estão a elaboração e correção de atividades de avaliação de aprendizagem, como listas de exercícios e provas. Para isso, alguns professores costumam selecionar e/ou adaptar questões de provas de concursos, vestibulares e afins. Pesquisas sobre técnicas de aprendizagem têm mostrado que responder questões é mais eficaz no tocante à retenção do conteúdo do que reler o assunto fazendo anotações/marcações (BJORK; DUNLOSKY; KORNELL, 2013).

Quanto à forma de aplicação de uma prova, ela pode ser impressa ou virtual. O uso de avaliações virtuais cresceu bastante durante a pandemia (ITANI et al., 2022), quando professores de todo o mundo passaram a criar suas provas em plataformas como o Moodle e o Google Forms. Tal prática perdura no mundo pós-pandemia e é apoiada

por diversas pesquisas que demonstram a importância do uso de questões objetivas na aprendizagem de novos conteúdos (KORNELL; BJORK; GARCIA, 2011).

A criação de sistemas de desenvolvimento de software voltados para auxiliar os professores na realização de provas online e na correção automática é um marco importante no campo da educação. Essas ferramentas oferecem uma série de benefícios, tanto para os educadores quanto para os alunos. Elas permitem que os professores economizem tempo na criação e correção de avaliações, reduzindo a carga de trabalho administrativo e, ao mesmo tempo, oferecem aos alunos uma experiência mais flexível e personalizada. Além disso, a correção automática garante uma avaliação imparcial e consistente, eliminando o potencial viés humano na atribuição de notas.

Nas últimas décadas, vários sistemas que visam facilitar a geração de provas foram desenvolvidos e disponibilizados para professores e instituições de ensino no Brasil, a exemplo do SuperProfessor¹, ElaboreProvas², GradePen³ e ProvaFácil⁴. Suporte a correção automática de provas *online* e impressas, importação de questões e vasto banco de questões públicas são algumas das características almejadas pelos usuários.

Este artigo abordará o desenvolvimento de uma aplicação *web*, chamada QuestõesPro, que busca integrar as principais ferramentas para criação e correção de avaliações, oferecendo também uma forma de avaliação de desempenho do aluno. A plataforma tem o intuito de auxiliar e reduzir consideravelmente o tempo do professor na execução e elaboração de uma prova, tanto de forma *online* como presencial. Em resumo investigaremos as principais funcionalidades que essa ferramenta pode oferecer, fornecendo uma visão abrangente sobre o desenvolvimento de software destinado a auxiliar professores na realização de provas e na correção automática, destacando as tecnologias utilizadas e os desafios de implementação.

O restante desse artigo encontra-se organizado como segue: A descrição dos materiais e métodos utilizados no desenvolvimento do sistema web QuestõesPro é apresentada na Seção 2; As novas funcionalidades implementadas são descritas na Seção 3; Os resultados obtidos ao final do desenvolvimento do projeto na Seção 4; Por fim, a Seção 5 conclui o trabalho, apontando as limitações do projeto e os trabalhos futuros.

2. Fundamentação teórica

A avaliação do aluno no ambiente escolar ajuda o seu educador a medir o quanto o aluno está progredindo, isto é, o quanto o aluno está entendendo o assunto estudado em sala de aula. Através da avaliação, o professor pode entender as dificuldades do aluno e suas necessidades, como também, mudar a sua didática de ensino para melhorar a aprendizagem da sua turma como um todo, e assim, é por meio da avaliação que o professor recebe o *feedback* de cada indivíduo de sua sala de aula.

Para avaliar sua turma, o professor pode optar por escolher provas dissertativas ou objetivas, debates, trabalhos em grupo, seminários, entre outros. O método de

¹ <https://www.superprofessor.com.br>

² <https://www.elaboreprovas.com.br>

³ <https://www.gradepen.com>

⁴ <https://www.provafacilnaweb.com.br>

avaliação mais difundido em sala de aula, e também entre concursos e vestibulares é a prova objetiva, que é formada primordialmente por questões de múltipla escolha (QME).

2.1. Embasamento científico

As QME são largamente utilizadas como instrumento avaliativo. Concebidas no início do século 20⁵ (BUTLER, 2018), fazem hoje parte da vida não apenas de estudantes e professores, mas da maioria das pessoas, seja enquanto profissionais, consumidores e até eleitores. Afinal, tais questões estão também presentes em processos seletivos e pesquisas de satisfação. Na área da educação, a maioria dos exames nacionais de avaliação, a exemplo do ENEM, ENADE e concursos públicos, são baseados primordialmente em questões de múltipla-escolha (QME).

Conceitualmente, uma QME consiste de três partes: (1) um texto-base (opcional); (2) um enunciado e (3) um conjunto de alternativas, das quais apenas uma é a correta, enquanto as outras são chamadas de distratores (KEHOE, 1994). A quantidade de distratores costuma ser cinco ou quatro, sendo essa definição um tema de estudo de muitas pesquisas (VYAS; SUPE, 2008).

Outro ponto vital diz respeito a como elaborar e validar as alternativas de uma QME. Haladyna (2004) aborda essa problemática, discutindo e apresentando as diretrizes necessárias na criação de uma boa questão. Questões bem elaboradas devem abranger diferentes níveis de complexidade, evitar ambiguidades e refletir os principais conceitos do assunto. Além disso, o foco não deve ser apenas na memorização de respostas, mas na compreensão profunda dos princípios subjacentes.

É importante destacar que o ato de resolver questões de múltipla escolha contribui substancialmente para auxiliar no processo de compreensão e retenção de novos conceitos, já que auxilia na recuperação de um assunto visto anteriormente guardado na memória, além de que o aluno também pode ser induzido, através da associação e lógica, à resposta correta, o que promove o aprendizado do assunto através da avaliação (LITTLE, BJORK, 2015; BUTLER, 2018).

Mais recentemente, com o advento da IA generativa, plataformas como o ChatGPT, Bard e Bing estão sendo utilizadas para criar QME com qualidade aceitável, embora ainda possuam, na maioria dos casos, enunciados de pouca complexidade (NASUTION, 2023; AGARWAL, PRIYANKA, AYAN, 2023).

2.2. Trabalhos relacionados

Questões de múltipla escolha estão presentes em avaliações na maioria dos ambientes acadêmicos, além de concursos e vestibulares de todo o Brasil, o que desencadeia diversas ferramentas online que auxiliam o professor na elaboração e correção desse tipo de avaliação. Dos sistemas existentes para tal fim, o pioneiro deles é a plataforma SuperProfessor, que foi fundada em 1993 e rapidamente se popularizou, e hoje conta com um banco que possui mais de 180 mil questões. Além do SuperProfessor, existem várias outras plataformas, como o ElaboreProvas, GradePen e ProvaFácil, e suas funcionalidades podem ser vistas na Tabela 1.

⁵ <https://veritasjournal.org/2019/04/30/the-history-of-the-multiple-choice-question/>

Tabela 1. Plataformas de avaliação online e suas funcionalidades.

Funcionalidade/ Plataforma	Super Professor	ProvaFácil	GradePen	ElaboreProvas	QuestõesPro
Gratuito			x ¹		x
Elabora avaliações	x	x	x	x	x
Corrige		x	x		x
Banco de questões	x	x	x	x	x
Questões de múltipla escolha	x	x	x	x	x
Usuário pode criar questões	x	x	x	x	x
Embaralha questões	x	x			x
Gera Estatísticas sobre as questões	x		x	x	
Gera arquivo editável da avaliação	x			x	x
Exporta DOCX / PDF	x	x	x	x	x
Importa DOCX		x			x
Filtragem avançada das questões	x	x		x	x

1: Gratuito só para elaboração de provas. A correção é uma funcionalidade paga.

Todas as plataformas citadas possuem alguma funcionalidade em comum, como o banco de questões públicas, a possibilidade de que o usuário elabore avaliações, crie questões, e exporte suas avaliações em formato de documento de texto (e.g. docx). A GradePen se destaca por ser parcialmente gratuita; apenas a correção de questões requer pagamento. E possui um banco de questões construído somente pelos próprios usuários, além de não possuir uma filtragem avançada, dificultando a consulta no próprio banco. As demais plataformas possuem planos para professores e instituições baseados na compra de créditos que podem ser gastos de acordo com a quantidade de questões selecionadas.

Além do SuperProfessor, outra plataforma também está no mercado há mais de 10 anos, que é a ProvaFácil, que possui uma maior quantidade e variedade de serviços além da elaboração de avaliações. O sistema possui planos para universidades, escolas, professores e alunos, onde ajuda a gerenciar calendários de provas, organização de alunos, gestão de avaliação, e acesso a dados estratégicos sobre o desempenho da instituição como um todo. O ProvaFácil possui um banco de questões onde as questões só são acessíveis pelas instituições que as criaram, então pode ser uma funcionalidade relativamente limitada, dependendo da quantidade de questões no banco.

A plataforma ElaboreProvas possui um banco com 226 mil questões, provenientes majoritariamente de exames vestibulares, classificadas em disciplinas, assuntos e tópicos. Adicionalmente, apresenta estatísticas globais sobre as disciplinas e

seus assuntos e possui um filtro avançado de questões em que o usuário pode escolher filtrar questões por região, instituição, ano, nível e tipo.

2.3. Embasamento tecnológico

As tecnologias escolhidas para o projeto em questão, são ferramentas muito utilizadas pelas empresas nos dias de hoje, nos quais são os *frameworks* React.js (front-end) e Java Spring Boot (*back-end*). A escolha dessas tecnologias se deu pelo motivo de sua flexibilidade e praticidade juntamente com a sua alta usabilidade no mercado de trabalho.

2.3.1. React.js

React é uma biblioteca JavaScript para construir interfaces de usuário (BODUCH; ADAM; DERKS, 2020). Ele é popular por ser fácil de usar, altamente flexível e escalável, e é usado por muitas empresas de tecnologia, incluindo o Facebook, Instagram e Airbnb. É uma biblioteca de JavaScript usada para criar interfaces de usuário interativas e dinâmicas em aplicativos da web. Permitindo o desenvolvimento de componentes reutilizáveis que são atualizados automaticamente conforme os seus dados mudem, oferecendo um desempenho eficiente e uma experiência de usuário responsiva, sendo conhecido por sua flexibilidade e escalabilidade.

Através do React, é possível desenvolver uma SPA (*single page application*), uma aplicação de página única onde o projeto possui apenas um documento, e seus componentes são atualizados dinamicamente seguindo uma lógica pré-definida. Ao carregar uma página SPA, todas as requisições que serão necessárias são feitas no início, mantendo a fluidez da navegação feita pelo usuário. A linguagem nativa usada pela biblioteca é a Javascript, mas também aceita a linguagem Typescript, que é uma linguagem de programação com base em Javascript que possui tipagem estática e interfaces, o que contribui para a detecção e prevenção de erros durante as etapas de desenvolvimento.

2.3.2. Spring Boot

O Spring Boot é uma ferramenta que nasceu a partir do Spring, um framework desenvolvido para o desenvolvimento de aplicações Java, onde o desenvolvimento da aplicação ocorre com mais agilidade e uma mínima (ou possivelmente nenhuma) perda de tempo com a configuração do projeto (WALLS, 2015). Ele fornece configurações de segurança padrão e inclui servidores Web integrados. Além disso, o Spring Boot oferece recursos prontos para uso em produção, oferecendo suporte em ambientes de construção comuns e interage facilmente com outros elementos do próprio ecossistema. Devido à sua simplicidade de configuração e eficácia, é frequentemente utilizado em projetos *web*, *microserviços* e sistemas de negócios.

2.3.3. API RESTful e JSON

Uma API expõe um conjunto de dados e funcionalidades para facilitar interações entre programas de computador e permitir a troca de informação entre eles (MASSE; MARK, 2011). O termo REST(*Representational State Transfer*) ou RESTful, se refere a um conjunto de restrições sobre como a API deve funcionar. Quando uma API RESTful troca recursos com outra aplicação, ela utiliza o protocolo HTTP, que envia os

dados em diversos formatos, como HTML, PHP, JSON, entre outros. O JSON (JavaScript Object Notation) é uma linguagem em formato de texto utilizado para armazenar dados, e é conhecida por possibilitar uma comunicação e troca de dados entre diferentes linguagens de maneira simples e prática.

2.3.4 Apache POI e Docx4J

Para manipular documentos em formato de documento de texto em formato docx, são necessárias bibliotecas externas às nativas da linguagem Java. Documentos em formato docx têm, em sua base, uma estrutura baseada em XML chamada WordprocessingML, onde todos os elementos do documento estão organizados. Para manipular a linguagem de marcação WordprocessingML existem bibliotecas como Apache POI⁶ e Docx4J⁷.

A biblioteca Apache POI tem como função a manipulação de documentos do pacote de recursos Microsoft Office (e.g. Word, Excel, PowerPoint) e é uma biblioteca de alto nível de controle que fornece uma interface para auxiliar no desenvolvimento de aplicações que necessitam dessa manipulação. Por ser uma biblioteca atualizada constantemente, a Apache POI, que é mantida pela Apache Software Foundation, pode ser facilmente utilizada, e permite a manipulação fácil de documentos Microsoft Office em poucas linhas de código.

Por outro lado, a Docx4J é uma biblioteca de código aberto mantido pela Plutext, onde o usuário pode manipular o documento Word em baixo nível, ou seja, a sua desvantagem é que o usuário precisa escrever muito mais linhas de código para fazer tarefas simples, porém, a biblioteca possui uma maior flexibilidade, comparando com Apache POI, onde possui menos limitações, além de possuir métodos de conversão de blocos HTML para parágrafos dentro do documento.

2.3.5 JUnit + MockMVC + Cypress

As ferramentas JUnit, MockMVC e Cypress, são frameworks de testes amplamente utilizadas durante o desenvolvimento e teste de um sistema, onde cada um desempenha um papel importante para garantir que as funcionalidades do projeto operem de acordo com o planejado.

JUnit é um framework de teste bastante usado em linguagens de programação java, permitindo criar e executar testes unitários (GARCIA; BONI, 2017). Ele proporciona um ambiente estruturado que auxilia na automatização de testes para confirmar se componentes específicos de um programa, principalmente classes ou métodos, operam de acordo com o planejado. O JUnit é uma ferramenta fundamental no desenvolvimento de software orientado por testes.

O MockMVC é uma parte do framework Spring Test que é amplamente utilizado para testar controladores em aplicações baseadas no Spring Framework, particularmente em aplicações web (GUTIERREZ, 2016). Ele oferece a capacidade de simular solicitações HTTP e interagir com os controladores de forma isolada, sem a necessidade de um servidor web real.

⁶ <https://www.poi.apache.org/>

⁷ <https://www.docx4java.org/>

Cypress é uma ferramenta de teste de front-end para aplicativos da web, esta poderosa ferramenta busca permitir de maneira fácil, rápida e confiável o desenvolvimento e execução de testes automatizados de qualquer aplicação web que possa ser acessada via navegador. Ele executa os testes em tempo real enquanto a aplicação, permitindo que os desenvolvedores visualizem as ações do teste e os resultados instantaneamente.

2.3.6 PostgreSQL e H2 Database

O PostgreSQL é um sistema de gerenciamento de banco de dados relacional (SGBDR), e é considerado o sistema de banco de dados de código aberto mais avançado do mundo (Milani, 2008). O sistema possui todas as funcionalidades que um sistema comercial (e.g. DB2 e Oracle) possui e pode ser largamente utilizado em um ambiente de produção.

O H2 Database também é um SGBDR, e é conhecido por ser leve, rápido e fácil de incorporar em aplicativos Java. A escolha desse sistema foi feita por ser portátil e adequado para uso não escalável durante a fase de desenvolvimento de aplicações Java, suportando operação em memória, o que significa que os dados podem ser armazenados na memória do sistema.

3. QuestõesPro

O produto mínimo viável do QuestõesPro foi inicialmente desenvolvido por uma equipe de estudantes de graduação da Universidade Federal da Paraíba (UFCG), e suas funcionalidades foram entregues implementadas e 80% testadas. O modelo de desenvolvimento utilizado durante o projeto foi inspirado no Scrum, com elementos de Kanban. Em suma: foi construído um backlog priorizado do produto; e o desenvolvimento das funcionalidades foi separado em sprints, cada sprint com seu backlog priorizado; e ocorreram reuniões tanto no início (*sprint planning*) quanto no fim de cada sprint (*sprint retrospective*).

A equipe anterior escolheu a estrutura de organização Kanban, que permite organização de atividades (*to do*, *in progress* e *done*) em quadros de tarefas, e foi utilizado o GitHub Projects como interface visual. O projeto já possuía algumas funcionalidades fundamentais, como o banco de questões com pesquisa por filtragem (Figura 1), turmas virtuais, criação, agendamento (Figura 2) e aplicação de avaliações (Figura 3).

Banco de Questões ?

[+ Adicionar](#) [Importar](#)

Procure por palavra-chave

Disciplinas 1 selecionados Assuntos 3 selecionados Bancas 3 selecionados

Dificuldades Níveis Anos

Status

Filtrar por: Assuntos 1 Arquitetura TCP/IP 21.10 DHCP (Dynamic Host Configuration Protocol) 39 Endereçamento IP

Disciplinas Redes de Computadores Bancas CETRO UFCG QP-User

Figura 1.Tela de consulta ao Banco de Questões, com destaque para os filtros e os botões de adicionar questão e importar prova (.docx).

O banco de questões possui uma filtragem por disciplina, assunto, banca e ano. E, posteriormente, nesse trabalho, foram adicionados nível e dificuldade. Além de mais opções de filtros, foi adicionado um botão que leva o usuário para a página de importação de questões. Um exemplo de tela do banco de questões pode ser vista na Figura 1.

Agendamento	Prova	Turma	Início do prazo	Fim do prazo	Duração	Status	Ações
Lista 1 - B...	Avaliação	Algoritmos	16/11/2023...	16/11/202...	60 min	ABERTO	  
Avaliação ...	Avaliação	Algoritmos	16/11/2023...	16/11/202...	60 min	ABERTO	  
Avaliação ...	Avaliação	Algoritmos	16/11/2023...	16/11/202...	60 min	ABERTO	  
Linhas por página: 10 1-3 de 3 < >							

Figura 2.Tabela de agendamentos de avaliações

A Figura 2 mostra a tabela de agendamentos, todos os modelos de provas que forem agendados serão mostrados nessa tela, pelo qual o professor pode visualizar os prazos, duração, o status e pré-visualizar o teste da maneira visualizada pelo aluno (Figura 3).

02:59:44 2/10 1/10

PRAZO RESPONDIDAS PULADAS

1 2 3 4 5 ... 10

A respeito de sistema integrado de gestão empresarial (ERP, do inglês, *enterprise resource planning*), é correto afirmar que

- ☐ a) aumenta a agilidade de programadores dos times de desenvolvimento diretamente.
- ☐ b) é de simples implementação em empresas já consolidadas.
- ☐ c) oferece aos gestores uma visão estratégica e detalhada de tudo o que acontece na operação.
- ☐ d) dificulta uma gestão integrada.
- ☐ e) a utilização dele por si só torna uma empresa verdadeiramente integrada.

← ANTERIOR PRÓXIMA → SUBMITER RESPOSTAS

Figura 3. Tela de resolução de prova.

3.1. Mapeamento do sistema

O desenvolvimento de novas funcionalidades para o projeto ficou a cargo da atual equipe, e, para isso foi necessário a aprendizagem das tecnologias utilizadas e o mapeamento da escrita e do funcionamento de todo o código, tanto dos endpoints da API na camada de regras de negócio (ou *back-end*), quanto dos componentes e rotas da aplicação web na camada de apresentação (ou *front-end*).

3.1.1. Camada de apresentação

Notou-se que o desenvolvimento do *front-end* foi inicialmente feito com a linguagem Javascript, sendo depois substituída(ainda pela equipe inicial) pela linguagem Typescript. Também foram usadas bibliotecas externas para organização do layout da página (MaterialUI e Styled Components), internalização da interface (i18n), navegação por rotas e apresentação dos elementos da página (React Router na versão 6), além do *framework* Jest, que serviu para criar testes unitários.

3.1.2. Camada de regras de negócio

Para o desenvolvimento do *back-end*, foi utilizado o *framework* Java Spring Boot, versão 2.5, o SGBDR em memória H2 e o SGBDR PostgreSQL, as bibliotecas ModelMapper e Lombok, e o JWT (*json web token*) como camada de segurança na comunicação com a API. O Maven foi a ferramenta escolhida para automatizar a gerência de dependências do projeto.

3.1.3. Testes

Os testes unitários foram implementados no *back-end* utilizando a biblioteca JUnit e usando a ferramenta Mock MVC, contida no ecossistema Spring, que possibilita o teste e as chamadas diretamente à API. Para cada entidade, foi executado um conjunto

de testes, que consistiu em testar as rotinas básicas na arquitetura RestFul (e.g. *GET*, *POST*, *UPDATE*, *DELETE*), e os cenários de uso do sistema, como a autenticação do usuário, ou a execução de uma prova.

3.2. Metodologia de desenvolvimento e versionamento

Para o controle de versões do projeto, foi utilizado o Git e o GitHub. A primeira equipe que desenvolveu o sistema, em 2022, havia utilizado Scrum, em que para cada *sprint* existiu uma *branch* (ou ramificação da versão anterior do projeto), sendo mesclada com a *branch* principal no fim de cada *sprint*. Entretanto, com a equipe atual, composta apenas por dois membros, foi escolhido apenas o método de organização Kanban, ainda utilizando os quadros de tarefas do Github Projects. À medida em que as funcionalidades foram sendo concluídas, a versão referente a cada funcionalidade foi sendo mesclada à versão final do projeto.

3.3. Arquitetura do sistema

A arquitetura do sistema (Figura 1) é em formato API REST, onde a comunicação entre a aplicação web e o servidor é feita por meio de requisições HTTP. Deste modo, as regras de negócio do sistema são encapsuladas dentro de serviços, que são chamadas pelos *endpoints* da API.

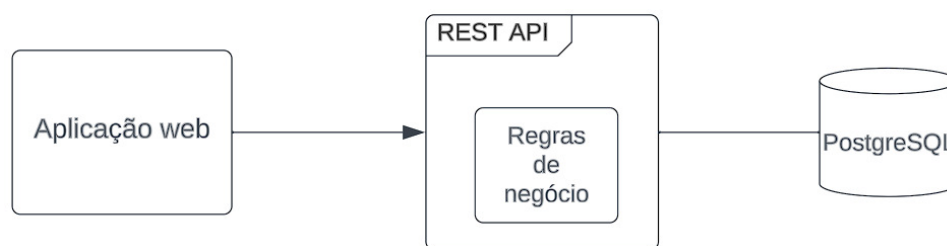


Figura 1. Diagrama de arquitetura do sistema

3.4. Persistência dos dados

Por meio da interface JPA (*Java Persistence API*) que é nativa da linguagem Java para persistência de dados, o sistema conecta-se ao PostgreSQL. Um fragmento do modelo de entidade e relacionamento (MER) foi criado (Figura 2) para descrever o relacionamento entre as principais entidades do projeto, onde professor e aluno são ambos usuários, autenticados por e-mail e senha.

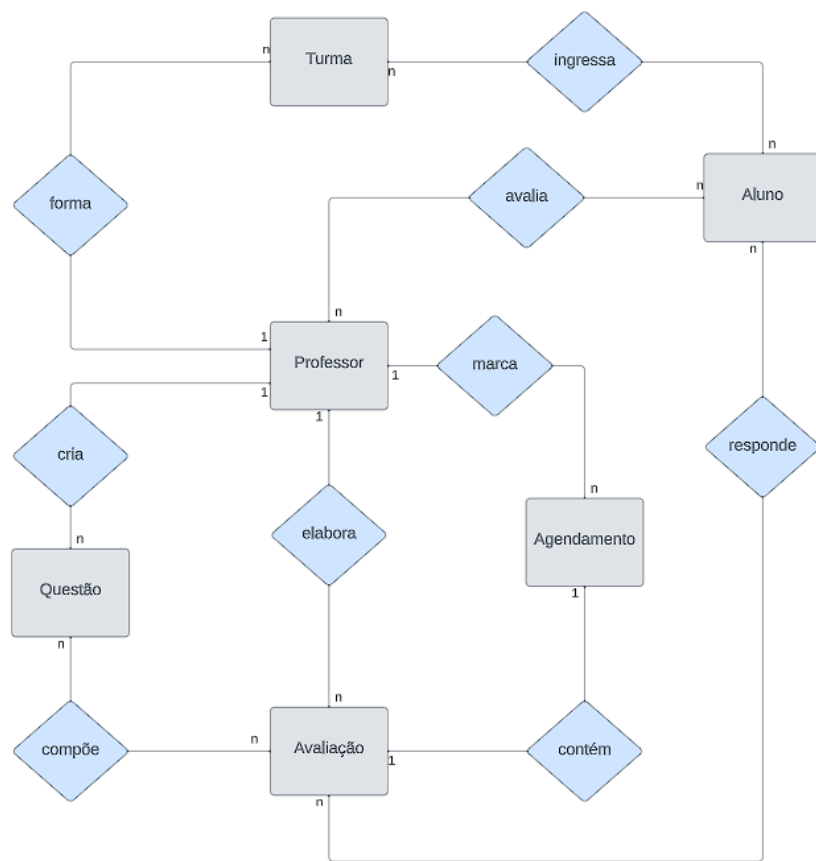


Figura 2. Modelo de Entidade e Relacionamento do banco de dados

O professor pode criar questões, e também pode elaborar modelos de avaliações a partir das questões criadas disponíveis no banco, além de poder agendar a aplicação das avaliações que pode ser de forma virtual ou física (por meio de prova impressa) para determinada turma. Por outro lado, o aluno que participa da turma pode responder a avaliação agendada previamente pelo professor, que o avalia de acordo com o seu desempenho.

3.5. Funcionalidades

As funcionalidades (Tabela 2) iniciais do projeto foram definidas pelo *Product Owner*, um professor da área de computação do IFPB, em conjunto com a equipe anterior, tentando responder à pergunta sobre o problema que motiva o projeto e seu escopo. Com o problema definido, os principais objetivos do sistema real foram estabelecidos e priorizados. Definem-se em:

- Facilitar o processo de uso e descoberta de questões
- Gerenciar alunos por turma
- Aplicar e agendar provas que o professor tenha criado dentro da plataforma
- Se integrar com sistemas de correção de prova física automáticas

Tabela 2. Funcionalidades implementadas pela equipe anterior

#	Funcionalidades planejadas (MVP)	Status
1	Autenticação básica dos usuários professor e aluno.	Operacional
2	Professor deve poder criar turmas.	Operacional
3	Aluno deve poder ingressar a uma ou mais turmas por meio de um código alfanumérico.	Operacional
4	Professor deve poder criar sua própria questão	Operacional
5	Professor deve poder elaborar avaliações com questões próprias ou disponíveis no banco de questões.	Operacional
6	Professor deve poder agendar suas avaliações para suas turmas.	Operacional
7	Aluno deve poder responder a provas agendadas para as turmas que participa.	Operacional

Para o desenvolvimento da segunda etapa do projeto, foram pensadas adição de melhorias, juntamente com correção de bugs presentes nas funcionalidades que já existiam, para que agregasse valor e complexidade ao projeto atual, além de garantir a continuidade do desenvolvimento do projeto realizado pela equipe antecessora. Na tabela 3, são mostradas as funcionalidades previstas.

Tabela 3. Funcionalidades implementadas pela segunda equipe (autores deste artigo)

#	Funcionalidades planejadas	Status
1	Gerar provas físicas (com questões embaralhadas ou não).	Operacional
2	Pesquisar questões por grau de dificuldade e nível (fundamental, médio e superior).	Operacional
3	Importar questões do Word (.docx)	Operacional
4	O aluno deve poder sair da Turma	Operacional
5	Mostrar estatísticas de acertos por questão para o aluno	Operacional
6	Aluno deve poder visualizar a lista de turmas em que participa	Operacional
7	Mostrar desempenhos individuais dos alunos para o professor	Operacional
8	O aluno pode visualizar a prova após finalizá-la no site	Operacional

4. Resultados

As funcionalidades planejadas foram implementadas, além de duas outras que não estavam planejadas inicialmente, que foram o suporte a imagens nas questões e a autenticação via Google. Os detalhes sobre a implementação de todas essas funcionalidades são descritos a seguir.

4.1. Correção de bugs

Foi realizada a correção de alguns bugs, com destaque para uma falha no front-end em que o usuário (perfil aluno) estava impossibilitado de visualizar suas turmas e, conseqüentemente, suas notas, provas e correções (Figura 4).



Figura 4. Tela de turmas vazia.

Além desta correção foram realizadas correções de outros bugs, mostrados na tabela 4. Uma refatoração no código do front-end também se fez necessária, porém, por falta de tempo hábil, somente foram feitas modificações nos arquivos que eram necessários para a implementação de uma nova funcionalidade.

Tabela 4. Bugs corrigidos

#	Bugs corrigidos
1	O aluno não consegue visualizar as turmas em que participa.
2	O aluno não consegue visualizar suas respostas ao terminar uma prova.
3	O professor não consegue apagar agendamentos nem turmas.

4.2. Estatísticas sobre o estudante

As estatísticas sobre o desempenho dos estudantes são de suma importância para o acompanhamento feito pelo professor. Essa funcionalidade foi implementada tanto para o professor como para o aluno por meio da biblioteca Google Charts, a qual disponibiliza uma ampla variedade de gráficos que foram elaborados utilizando dados obtidos pela API no back-end.

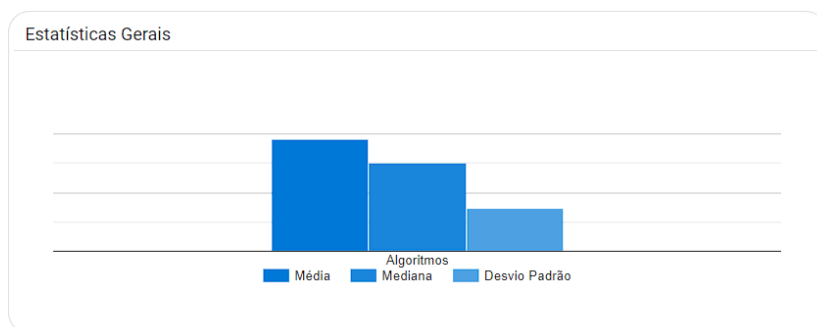


Figura 4. Gráfico de média, mediana e desvio padrão de um estudante.

Como observado no gráfico para cada avaliação em uma turma, o professor consegue visualizar a média, mediana e desvio padrão das notas dos estudantes. Também, é possível saber quantas e quais avaliações foram ou não entregues, além do tempo gasto para terminar cada avaliação, de acordo com o gráfico da figura 5.

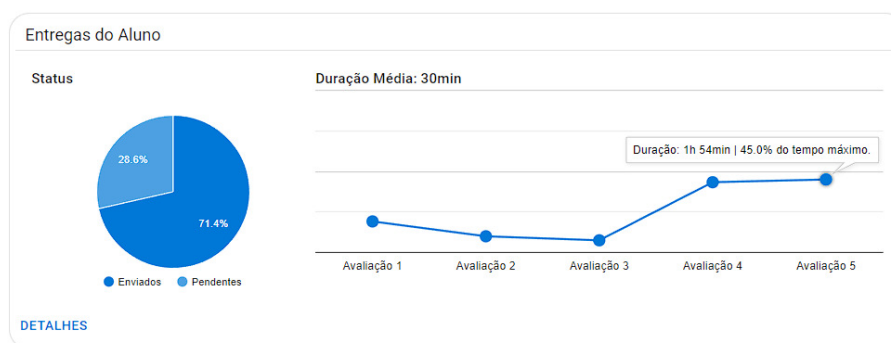


Figura 5. Gráfico de entregas do aluno e gráfico com tempo gasto para completar cada avaliação.

O aluno, por sua vez, para cada turma em que participa, também consegue visualizar sua própria média, mediana, e desvio padrão (em um gráfico como o da Figura 4 para cada turma), verificar as avaliações que foram entregues, e também pode acompanhar os prazos de entrega por meio de um gráfico de cronograma.

4.3. Exportação da prova para documento de texto editável.

O professor que realiza o agendamento de sua avaliação pode necessitar imprimir uma versão física dessa avaliação. Sendo assim, para que o usuário gere um documento de texto referente a uma avaliação, é necessário inicialmente selecionar um template, o qual detalha a estrutura do documento. Por exemplo, o usuário pode optar

por incluir um cabeçalho, o tipo e dimensão da fonte e a organização dos elementos do cabeçalho.

Nome	Valor
#1 Instituição	Instituto Federal da Paraíba
#2 Local	: Campina Grande - PB
#3 Professor	: Jonas H Nascimento
#4 Disciplina	: Teoria dos Grafos
#5 Aluno	:

Figura 6. Tela de informações sobre template a ser utilizado na geração do documento

Um elemento de cabeçalho possui nome, que é uma propriedade desse elemento, e o seu valor, ocupando uma linha no cabeçalho. Assim, o usuário pode escrever as informações que precisa no cabeçalho em um template, e sempre que precisar ele pode escolher esse template para servir de base para futuros documentos.

Quando o documento é gerado, cada elemento de cabeçalho ocupa uma linha no cabeçalho da folha (Figura 7), de maneira que os elementos que não tem valor permanecem como espaço em branco a ser preenchido manualmente pelo aluno ou professor durante a aplicação da avaliação.

Instituição: Instituto Federal da Paraíba
Local: Campina Grande - PB
Aluno:
Professor: Jonas Henriques Nascimento
Turma:
Nota:

Figura 7. Visualização do template na folha de documento em formato docx.

Após a seleção do *template*, o usuário já pode selecionar o agendamento ou o modelo de avaliação e executar o *download* do documento em formato “.docx” (Figura 8).

Selecione um agendamento ou um modelo de avaliação. ^

Selecione uma avaliação

Avaliação ▾

BAIXAR O DOCUMENTO

Figura 8. Tela de download de avaliação convertida em documento Word.

4.4. Importação de questões de um documento de texto

O processo de criação de questões recebeu mais uma funcionalidade, que é a de importar questões de uma prova em formato de documento de texto (e.g. docx). Para importar as questões, o professor pode selecionar um documento, executar uma breve configuração sobre a separação das questões no documento (Figura 9), e então, extrair as questões.

Configure a importação

Eu numero minhas questões no formato:

selecione

Número seguido por símb... ▾

Eu numero minhas alternativas no formato:

selecione

Letra seguida por símbolo ▾

As questões são separadas por :

selecione

Parágrafo ▾

As alternativas são separadas por:

selecione

Parágrafo ▾

IMPORTAR DOCUMENTO

Figura 9. Configuração de importação de questões

Após configurar como as questões dentro do documento estão separadas, o professor precisa, para cada questão extraída, selecionar seu assunto, disciplina, nível de dificuldade, e só depois que todas as questões estiverem configuradas, pode salvá-las.

4.5. Suporte a questões com imagens

Anteriormente, as questões com imagens ainda não haviam sido implementadas, gerando uma limitação no banco de questões, onde as imagens das questões simplesmente não apareciam. Então, a funcionalidade de suportar questões com imagens foi adicionada, tanto na aplicação web, quanto na geração de documentos de texto.

O tipo de teste de software que representa a figura abaixo é:

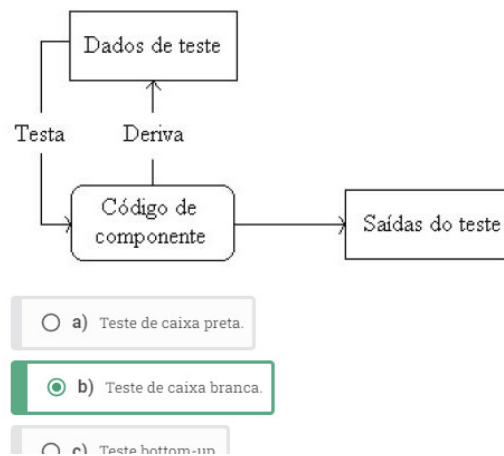


Figura 10. Questão com imagem no banco de questões

Enquanto implementado um servidor externo para o armazenamento das imagens, o próprio servidor da aplicação armazena localmente as imagens e as dispõe para o usuário autenticado, ou seja, para questões privadas, que foram criadas pelo professor, as imagens só serão mostradas para o autor das questões, isso impossibilita que outros usuários possam acessar imagens privadas.

4.6. Login via google e verificação de e-mail

O login via Google possibilita o usuário logar e se cadastrar de forma mais rápida e simples, permitindo que faça *login* na aplicação usando suas credenciais do Google ao invés de criar e gerenciar contas separadas para a aplicação, permitindo que os usuários acessem a plataforma usando suas contas do Google, o que é conveniente e seguro.

A imagem mostra a interface de login de uma aplicação web. No topo, o título 'Login' é exibido em uma fonte escura. Abaixo dele, uma mensagem de boas-vindas 'Seja bem-vindo(a)!' aparece em uma fonte menor. O formulário contém dois campos de entrada: 'Email' com o placeholder 'Digite seu email' e 'Senha' com o placeholder 'Digite sua senha' e um ícone de olho para alternar a visibilidade. Abaixo dos campos, há um botão preto com o texto 'ENTRAR' em branco. Segue-se a palavra 'OU' centralizada. Abaixo, há um botão branco com o ícone do Google e o texto 'ENTRAR COM GOOGLE'. No final, há um botão azul com o texto 'Ainda não possui uma conta ? Cadastre-se' em branco.

Figura 11. Tela de login.

Na página de *login* foi adicionado um botão onde é possível efetuar a autenticação com a conta Google do usuário (Figura 11), onde no final do processo será obtido alguns dados do usuário fornecidos pela API do Google por exemplo nome, e-mail e foto de perfil. Essa funcionalidade facilita o acesso do usuário ao sistema e viabiliza outra forma de login segura e eficiente.

5. Conclusão

Com base nos resultados apresentados, é possível concluir que as funcionalidades planejadas foram finalizadas para esse trabalho. Além disso, todas foram acrescentadas e validadas. A aplicação foi projetada e implementada com a finalidade de prover um maior controle do tempo ao professor, além de facilitar o processo de elaboração de avaliações.

Durante o desenvolvimento do sistema, houveram dificuldades no processo de implementação devido a íngreme curva de aprendizagem inicial. Tal fato deu-se dado o projeto existente possuir uma vasta gama de tecnologias, além de diversas bibliotecas de terceiros, que desencadeou em algumas limitações na adição de novas funcionalidades. Um dos principais empecilhos enfrentados, foi a falta de organização na estruturação do código, mais especificamente na parte do front-end, que contribui para um ritmo mais lento de desenvolvimento. Além disso, a escassez de bibliotecas gratuitas e atualizadas para a geração de documentos de texto usando a linguagem Java também deu-se como fato preponderante que atrasaria o processo de desenvolvimento.

Diversas novas funcionalidades foram planejadas para execução em ciclos futuros de desenvolvimento, como exemplo a ideia de acrescentar a funcionalidade de geração automática de questões, que foi despriorizada devido a sua alta demanda temporal para sua realização, e a utilização de uma inteligência artificial para classificação de nível das questões.

Também foi idealizado o desenvolvimento de um aplicativo móvel para o QuestõesPro, fundamentando-se na possibilidade de correção das avaliações físicas pelo celular através de uma foto do gabarito da prova e da identificação da mesma por um QR code.

O QuestõesPro visa apresentar uma interface simples, direta e amigável para com o usuário. Dessa forma, sua existência pode inspirar e auxiliar na criação de novas ferramentas ou funcionalidades no mercado que possam facilitar e melhorar o processo de ensino e avaliação por parte do educador.

Por fim, pode-se concluir que as funcionalidades desenvolvidas neste trabalho foram devidamente implementadas e estão em funcionamento, agregando valor ao projeto. As novas funções agora podem auxiliar o professor durante o processo de elaboração de avaliações, e possibilitar uma melhor otimização do tempo por parte do professor, durante a elaboração e execução de uma avaliação..

Referências

Agarwal, Mayank, Priyanka Sharma, and Ayan Goswami. "Analysing the applicability of ChatGPT, Bard, and Bing to generate reasoning-based multiple-choice questions in medical physiology." *Cureus* 15.6 (2023).

Boduch, Adam, and Roy Derks. "React and React Native: A complete hands-on guide to modern web and mobile development with React. js". Packt Publishing Ltd, 2020.

Brady, Anne-Marie. "Assessment of learning with multiple-choice questions." *Nurse Education in Practice* 5.4 (2005): 238-242.

Butler, Andrew C. "Multiple-choice testing in education: Are the best practices for assessment also good for learning?." *Journal of Applied Research in Memory and Cognition* 7.3 (2018): 323-331.

Garcia, Boni. "Mastering Software Testing with JUnit 5: Comprehensive guide to develop high quality Java applications". Packt Publishing Ltd, 2017.

Gutierrez, Felipe. "Testing with Spring Boot." *Pro Spring Boot* (2016): 107-120.

Haladyna, Thomas M. *Developing and validating multiple-choice test items*. Routledge, 2004.

Kehoe, Jerard. "Writing multiple-choice test items." *Practical Assessment, Research, and Evaluation* 4.1 (1994): 9.

Little, J. L.; Bjork, E. L. "Optimizing multiple-choice tests as tools for learning." *Memory & Cognition* 43 (2015): 14-26.

Masse, Mark. "REST API design rulebook: designing consistent RESTful web service interfaces". O'Reilly Media, Inc., 2011.

Milani, André. "PostgreSQL-Guia do Programador". Novatec Editora, 2008.

Nasution, N. E. A. "Using artificial intelligence to create biology multiple choice questions for higher education." *Agricultural and Environmental Education* 2.1 (2023).

Vyas, R. and Supe, A., 2008. Multiple choice questions: a literature review on the optimal number of options. *Natl Med J India*, 21(3), pp.130-3.

Walls, Craig. "Spring Boot in action". Simon and Schuster, 2015.

	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
	Campus Campina Grande - Código INEP: 25137409
	R. Tranquilino 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 TCC

Assunto:	Versão Final TCC
Assinado por:	Jorge Damasceno
Tipo do Documento:	Anexo
Situação:	Finalizado
Nível de Acesso:	Ostensivo (Público)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- Jorge Victor Leite Damasceno, ALUNO (201811250015) DE BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO - CAMPINA GRANDE, em 01/03/2025 14:45:25.

Este documento foi armazenado no SUAP em 01/03/2025. 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: 1408106
Código de Autenticação: 6cf48bcdfb

