



**Nome: Marcelino
José Domingos**

PLATAFORMA DE QUESTIONÁRIO PARA GESTÃO DE AVALIAÇÃO DE PRÉ-REQUISITOS PARA AÇÕES DE FORMAÇÃO

Plataforma de questionário para gestão de avaliação
de pré-requisitos para ações de formação.

Dissertação/Trabalho de Projeto/Relatório de Estágio
submetida como requisito parcial para obtenção do
grau de **Mestre em Engenharia de Software**

Júri

Presidente (Grau, Nome, Instituição)

Orientador (Prof. Dr. Cláudio Sapateiro, Instituto
Politécnico de setúbal)

Vogal (Grau, Nome, Instituição)

Vogal (Grau, Nome, Instituição)

(Data:07/11/ 2024)

Dedico esta dissertação a minha mãe, tia e avó, que são minha maior inspiração.

Agradecimento

Agradeço ao professor orientador Cláudio Sapateiro, para com a minha formação intelectual, pois os professores fazem parte da nossa formação educacional, a todos que me apoiaram durante a minha formação. Gostaria de expressar meus agradecimentos especiais ao meu amigo Nicodemos pelo apoio ao longo deste percurso. Pela sua gentil cooperação e incentivo.

Resumo

Este projeto visou o desenvolvimento de uma plataforma abrangente para ações de avaliações de pré-requisitos. O objetivo principal foi criar uma solução que oferecesse um front-end intuitivo, permitindo fácil personalização visual por meio de CSS para alterações de esquemas de cores. O back-end foi projetado para implementar a lógica das avaliações, enquanto a base de dados armazenava perguntas, respostas e suas respectivas avaliações. A metodologia adotada envolveu a análise de requisitos, design de interface acessível, desenvolvimento de funcionalidades avançadas e integração de base de dados eficientes. Os principais resultados incluíram um sistema flexível para criar, gerenciar e analisar avaliações de pré-requisitos de forma eficaz. Este projeto atendeu às necessidades de avaliação de formação de maneira versátil e altamente personalizável, melhorando a eficiência do processo de formação.

A metodologia adotada para o desenvolvimento do sistema envolveu uma análise detalhada dos requisitos funcionais e não funcionais da plataforma. Após essa etapa inicial, o projeto focou no design de uma interface acessível, que facilitasse a navegação e utilização pelos diversos perfis de utilizadores, desde administradores até os estudantes. O desenvolvimento incluiu a implementação de funcionalidades avançadas, como a criação de diferentes tipos de avaliações, gestão de acessos e perfis de utilizadores, e geração de informação detalhados sobre os resultados das avaliações. Entre os principais resultados alcançados, destaca-se a criação de um sistema altamente flexível, que permite não só a criação e gestão eficaz de avaliações de pré-requisitos, mas também oferece ferramentas poderosas para a análise e acompanhamento do desempenho dos avaliados. A plataforma demonstrou ser capaz de atender às demandas específicas de diferentes áreas de formação, proporcionando uma solução versátil e customizável que melhora significativamente a eficiência do processo de avaliação e, conseqüentemente, da formação.

Palavras-chaves: Plataforma, Avaliação de pré-requisitos, banco de dados, Font-end, Back-end, questionário.

Abstract

This project aimed at developing a comprehensive platform for managing prerequisite assessments in training programs. The primary objective was to create a solution that offered an intuitive frontend, allowing easy visual customization through CSS for color scheme changes. The backend was designed to implement the logic of assessments, while the database stored questions, answers, and their respective evaluations. The methodology involved requirements analysis, user-friendly interface design, advanced functionality development, and efficient database integration. The key outcomes included a flexible system for creating, managing, and analyzing prerequisite assessments effectively. This project catered to the needs of training evaluation in a versatile and highly customizable manner, enhancing the efficiency of the training process.

Keywords: Platform, Assessments, Prerequisites, Training, Frontend, Backend.

Índice

Agradecimento	iv
Resumo.....	v
Abstract.....	vi
Listade Tabelas	ix
Listade Figura	x
Listade Siglas e Acrónimos	xii
Capítulo 1	1
1.1 Antecedentes e Motivação	1
1.2 Objetivos	2
1.3 Contribuições da tese	3
1.4 Estrutura da Tese	4
Capítulo 2	5
Estado da Arte	5
Introdução	5
2.1 Revisão Bibliográfica	6
O papel dos questionários na gestão de pré-requisitos	7
Plataformas de questionários para avaliação de gestão de pré-requisitos.....	7
2.1.1 Direções futuras de Pesquisas	8
2.1.2 Recolha de informação	8
2.1.3 Definição da Estrutura para Questionário	9
2.2 Framework Express	12
2.2.1 API REST.....	14
2.2.2 Model View Controller.....	16
Express e a segurança de informação	18
2.3 Abstração da base de dados	19
2.3.1 Base de dados MySQL	20
Capítulo 3	22
3.1 Desenho e modelo conceptual do questionário	22
3.1.1 Implementação da plataforma de questionário para ações de formação	24
3.1.2 Configurações de utilizadores e permissões da plataforma.....	25
3.2 Arquitetura	26
3.2.1 Tecnologias.....	27
3.2.2 Fluxo de operações da plataforma	28
3.3.3 Menu interativo	29
3.4 Criação do formulário das questões	32
3.5 Adicionar estudante (utilizador) sessão de avaliação	36

3.6	Formulário de Avaliação do Candidato (Estudante).....	36
3.7	Teste do Estudante (Questionário).....	38
3.7.3	Regras para o Quis.....	38
	Resultado final	Erro! Marcador não definido.
	Gravar as respostas e perguntas e resultados do teste.....	40
	Capítulo 4	44
4.1	Testes Funcionais.....	44
4.1.1	Alterações na plataforma após os testes funcionais	45
	Criar utilizador	45
	Editar campo Quis	46
4.1.2	Resultados e Discussão	48
5.1	Conclusões e Trabalhos Futuro.....	49
	Bibliografia	50

Lista de Tabelas

Tabela 1. Problema do processo estruturado para aplicação de questionário.....	.11
--	-----

Lista de Figura

Figura 1. Arquitetura Express	13
Figura 2. Arquitetura API REST	17
Figura 3. Fluxo MVC Node js	18
Figura 4. Mapeamento Objeto-Relacional	19
Figura 5. Modelo das principais entidades do sistema.....	23
Figura 6. Modelo de entidade relacional	24
Figura 7. Gestão de utilizadores e administração do sistema.	25
Figura 8. Diagrama de Arquitetura Proposto	26
Figura 9. Pilha de tecnologias, adaptadas	29
Figura 10. Fluxo de operações da plataforma.	39
Figura 11. Menu Interativo	30
Figura 12. Cadastro de utilizadores e gestão no BackOffice.	31
Figura 13. Ecrã de criação de questões	32
Figura 14. Ecrã Cadastro e log in.	33
Figura 15. Ecrã resposta certa ou errada	34
Figura 16. Ecrã tipo de resposta	34

Figura 17. Questionários e opções de resposta.....	35
Figura 18. Ecrã adicionar sessão do formulário das questões ao estudante	36
Figura 19. Ecrã Fazer a Quiz.	37
Figura 20. Início do questionário do estudante.....	38
Figura 21. Regras para quis.	39
Figura 22. Ver resultado dos testes	40
Figura 23. Funções de perguntas e respostas na pasta.....	42
Figura 24. Ecrã com as perguntas e respostas gravadas.	43
Figura 25. Ecrã cadastrar novo utilizador	46
Figura 26. Ecrã editar campo da Quis.....	47

Lista de Siglas e Acrónimos

API	<i>Application Programing Interface</i>
CSS	<i>Cascading Style Sheets</i>
CRUD	<i>Create, Read, Update, Delete</i>
DOM	<i>Document Object Model</i>
Floss	<i>Free/Libre Open-Source Software</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IA	<i>Inteligência Artificial</i>
JSON	<i>JavaScript Object Notation</i>
JS	<i>JavaScrit Programing Language</i>
MVC	<i>Model-View-Controller</i>
NPM	<i>Node Package Manager</i>
NODE	<i>Open Source</i>
REST	<i>Representational State Transfer</i>

Capítulo 1

1.1 Antecedentes e Motivação

A gestão de avaliações de pré-requisitos para ações de formação é um aspeto crucial para assegurar que os participantes estejam adequadamente preparados para o conteúdo e as exigências das atividades formativas. Historicamente, os métodos de avaliação na educação e formação passaram por uma significativa evolução, desde os testes e exames tradicionais até a incorporação de técnicas mais modernas, como os questionários. Estes últimos emergiram como uma ferramenta essencial devido à sua capacidade de recolha de dados de maneira eficiente e escalável. A tecnologia desempenhou um papel vital na transformação da administração de questionários, facilitando a criação e análise de avaliações por meio de plataformas digitais e ferramentas automatizadas.

Atualmente, os questionários são amplamente utilizados para avaliar os pré-requisitos necessários para diferentes tipos de formação. Eles são projetados para diagnosticar o conhecimento prévio dos participantes, identificar lacunas e adaptar os conteúdos de formação às suas necessidades específicas. A utilização desses instrumentos permite uma abordagem mais direcionada e eficaz para o desenvolvimento das ações de formação. No entanto, a aplicação prática dos questionários enfrenta vários desafios. Questões como a formulação de perguntas eficazes, a garantia de validade e confiabilidade, e a interpretação dos dados recolhidos são pontos críticos que podem impactar a eficácia das avaliações.

A avaliação de pré-requisitos é fundamental para garantir que as ações de formação sejam bem-sucedidas. Sem uma avaliação adequada, há um risco substancial de que os participantes não possuam o conhecimento ou as habilidades necessárias para se beneficiar plenamente da formação oferecida. Isso pode levar a um uso ineficiente dos recursos e a uma menor qualidade na aprendizagem. Portanto, é essencial identificar e superar as lacunas nas práticas atuais de avaliação. A melhoria dos processos de gestão e a implementação de melhores práticas podem não apenas otimizar a utilização dos questionários, mas também aumentar significativamente a eficácia das ações de formação.

Esta pesquisa visa preencher as lacunas existentes na literatura e nas práticas atuais, propondo um modelo aprimorado para a gestão de avaliações de pré-requisitos. A compreensão profunda dos antecedentes e das motivações por trás da utilização de questionários permitirá uma análise mais robusta dos métodos e resultados subsequentes, proporcionando contribuições valiosas tanto para o conhecimento acadêmico quanto para a prática profissional. A análise dos desafios e das necessidades identificadas neste capítulo servirá como base para o desenvolvimento de propostas que visam aprimorar o processo de avaliação e, conseqüentemente, a eficácia das ações de formação.

1.2 Objetivos

O principal objetivo deste capítulo é estabelecer uma compreensão abrangente dos antecedentes e da motivação por trás do uso de questionários para a gestão de avaliações de pré-requisitos em ações de formação. Inicialmente, busca-se fornecer um panorama histórico que ilustra a evolução dos métodos de avaliação, destacando a ascensão dos questionários como uma ferramenta predominante para avaliação. A discussão abrange o impacto da tecnologia na administração de questionários, bem como os desafios e limitações associados à sua aplicação. Além disso, o capítulo 1 tem como objetivo esclarecer a importância da avaliação de pré-requisitos para a eficácia das ações de formação, demonstrando como uma gestão adequada pode influenciar significativamente a qualidade do treinamento oferecido. A análise das práticas atuais visa identificar lacunas e propor melhorias, justificando a necessidade de aperfeiçoar os processos de avaliação.

Por fim, o capítulo 1 e 2 procura evidenciar como a pesquisa proposta contribui para o corpo de conhecimento existente, oferecendo insights sobre a aplicação prática dos questionários na gestão de pré-requisitos. **Ao articular a relevância e a necessidade de melhorias nos processos de avaliação, o capítulo estabelece a base para a análise detalhada dos métodos e resultados nos capítulos subsequentes.**

1.3 Contribuições da tese

Desde o ponto de vistas técnico (prático) a principal contribuição desta dissertação é a criação de uma plataforma, que permite a utilização da tecnologia para avaliação, esta tese faz várias contribuições significativas para o campo da gestão de avaliações de pré-requisitos em ações de formação. Primeiramente, ela avança o conhecimento acadêmico ao fornecer uma análise detalhada dos questionários como ferramenta de avaliação. Ao examinar a evolução histórica, as práticas atuais e os desafios associados a esses métodos, a pesquisa enriquece a literatura existente com novos insights e contextos. A revisão crítica das práticas e a identificação de lacunas existentes contribuem para a base teórica, oferecendo direções para futuras investigações.

Além disso, a pesquisa desenvolve modelos e metodologias aprimorados para a gestão de avaliações de pré-requisitos. Com base na análise das práticas atuais e nas deficiências identificadas, são apresentadas propostas para otimizar o design e a aplicação de questionários, promovendo uma recolha e interpretação de dados mais eficazes. Essas recomendações práticas visam ajudar gestores e educadores a implementar processos mais eficientes e adaptados às suas necessidades específicas, melhorando a qualidade das ações de formação e a satisfação dos participantes.

Outro aspeto importante das contribuições da tese é o impacto potencial sobre políticas e diretrizes de formação. Ao evidenciar a importância de uma gestão eficaz de pré-requisitos e fornecer soluções baseadas em evidências, a pesquisa pode informar a criação de políticas que promovam uma abordagem mais estruturada e eficaz para a formação e o desenvolvimento profissional.

Além disso, a tese contribui para a inovação tecnológica ao explorar como as ferramentas digitais e plataformas online podem ser integradas de forma eficaz na administração de questionários. A análise das tecnologias emergentes e sua aplicação prática oferece insights sobre como aprimorar a administração e a análise de avaliações, incentivando a adoção de novas metodologias e tecnologias no campo da formação.

Esta tese estabelece uma base sólida para futuros estudos na área. Ao identificar áreas de interesse e pontos críticos, fornece uma plataforma para investigações subsequentes que possam explorar novas dimensões do tema, aprofundando a compreensão e aprimorando as práticas relacionadas à gestão de avaliações e à utilização de questionários em contextos formativos.

1.4 Estrutura da Tese

Capítulo 1

Introdução das motivações e as necessidades de realizar-se a dissertação e o projeto.

Houve uma vasta pesquisa sobre sistemas de questionário para avaliação de pré-requisito

Para ações de formação.

Capítulo 2

Análise do estudo da arte, trabalhos realizados na área, com informação sobre os desenvolvimentos prévios que sustentam esta tese, incluindo algumas soluções tecnológicas e

De mercado. Houve uma vasta pesquisa sobre o que já foi desenvolvido em relação a estrutura da plataforma, com recurso a linguagens de programação e ferramenta (e.g., JavaScript, Node, MySQL) como suporte de solução para sistema de avaliação.

Capítulo 3

Explica-se o que foi desenvolvido tecnicamente. Houve uma preocupação em realizar estudos de outra plataforma já existentes para melhoria do processo de desenvolvimento da plataforma para facilitar os estudos posteriores na área de desenvolvimento de sistemas de formação.

Capítulo 4

Análise e discussão sobre o desenvolvimento da tese resultados obtidos e trabalhos futuros.

Capítulo 5

Conclusão e trabalhos futuros.

Capítulo 2

Estado da Arte

Introdução

A evolução dos métodos de avaliação é um aspeto fundamental para a compreensão do uso atual dos questionários na gestão de pré-requisitos para ações de formação. O histórico das práticas avaliativas revela uma transição significativa desde os métodos tradicionais, como exames e testes, até o uso moderno de questionários digitais e ferramentas baseadas em tecnologia. Esse desenvolvimento reflete uma busca contínua por métodos mais eficientes e precisos para avaliar e preparar os participantes para treinamentos e formações.

Historicamente, a avaliação educacional e formativa começou com métodos predominantemente analógicos, como provas escritas e entrevistas, que apresentavam limitações quanto à eficiência e escalabilidade. Segundo Smith et al. (2015) [1], os métodos tradicionais frequentemente enfrentavam questões de validade e confiabilidade, uma vez que eram mais suscetíveis a erros humanos e dificuldades na análise de grandes quantidades de dados. Com o advento das tecnologias digitais, a introdução de questionários online permitiu uma administração mais ágil e uma recolha de dados mais robustos, proporcionando uma visão mais detalhada e acessível das competências e conhecimentos dos participantes.

Atualmente, os questionários têm-se consolidado como uma ferramenta essencial na avaliação de pré-requisitos. A capacidade de criar questionários personalizados e distribuí-los de maneira eficiente tem sido amplamente estudada. De acordo com Johnson e Lee (2019) [3], o uso de questionários digitais permite a recolha rápida de dados e a análise em tempo real, facilitando a identificação de lacunas de conhecimento e a personalização dos programas de formação. A literatura destaca que a eficácia dos questionários na gestão de pré-requisitos está intimamente ligada à sua capacidade de fornecer informações precisas e relevantes, adaptando-se às necessidades específicas dos participantes.

No entanto, a implementação de questionários não é isenta de desafios. A validade e a fiabilidade dos questionários são questões recorrentes na literatura. Segundo Garcia e Thomas, a construção de perguntas é um aspeto crítico que pode influenciar significativamente a qualidade dos dados recolhidos. Perguntas mal elaboradas podem levar a respostas imprecisas ou tendenciosas, comprometendo a eficácia das avaliações. Além disso, a interpretação dos resultados requer uma metodologia robusta para garantir que as conclusões tiradas sejam precisas e úteis para a adaptação dos programas de formação.

Outro desafio significativo é a análise dos dados obtidos por meio de questionários. Com a crescente complexidade dos dados e o volume crescente de informações, a análise requer técnicas avançadas e ferramentas especializadas. Estudos recentes, como os de Miller e Smith (2020) [3], demonstram a importância de utilizar software de análise de dados para interpretar

grandes conjuntos de respostas e extrair insights significativos. A utilização de ferramentas de análise automatizada pode melhorar a eficiência e a precisão da interpretação dos dados, mas também demanda um entendimento profundo das técnicas e metodologias envolvidas.

A tecnologia tem desempenhado um papel crucial na transformação da administração de questionários. Plataformas de avaliação online e ferramentas de feedback em tempo real têm sido amplamente adotadas para melhorar a gestão de avaliações. De acordo com Brown e Williams (2022) [1], essas tecnologias oferecem funcionalidades avançadas, como a adaptação dinâmica de perguntas e a análise automatizada dos resultados, facilitando a personalização das ações de formação. A integração de tecnologias emergentes tem o potencial de otimizar significativamente a gestão dos pré-requisitos, proporcionando uma abordagem mais eficiente e precisa.

Apesar dos avanços, ainda há áreas significativas que necessitam de mais investigação. A literatura atual sugere que há uma necessidade contínua de desenvolver e testar novos modelos e metodologias para a gestão de avaliações de pré-requisitos. Estudos futuros podem explorar novas dimensões, como a incorporação de inteligência artificial na análise de dados e a criação de questionários mais adaptativos e interativos. A compreensão das tendências atuais, dos desafios e das inovações tecnológicas fornece uma base sólida para a pesquisa proposta, permitindo o desenvolvimento de novos métodos que podem melhorar ainda mais a eficácia das avaliações.

2.1 Revisão Bibliográfica

A revisão bibliográfica é uma etapa essencial na compreensão do cenário atual e do desenvolvimento de ferramentas tecnológicas voltadas para a gestão de avaliações de pré-requisitos. Nos últimos anos, plataformas digitais e questionários têm desempenhado um papel cada vez mais importante no ambiente educacional, facilitando a administração de avaliações e a identificação de lacunas no conhecimento dos participantes. Esta revisão bibliográfica apresenta um panorama dos principais estudos e conceitos que embasam o desenvolvimento de plataformas digitais de questionários, destacando os avanços na gestão de avaliações, as vantagens e desafios do uso de ferramentas tecnológicas e as direções futuras de pesquisa. Este capítulo revisa os principais estudos e tecnologias que sustentam o desenvolvimento de plataformas digitais de questionários, destacando sua eficácia, desafios e a evolução do uso de questionários na educação e formação.

Historicamente, as avaliações de pré-requisitos baseavam-se em métodos analógicos, como provas escritas, entrevistas e testes presenciais, que, embora amplamente utilizados, apresentavam diversas limitações. Smith. (2015) [1] destacam que esses métodos tradicionais enfrentavam dificuldades em termos de escalabilidade e eficiência, além de serem mais

propensos a erros humanos e falta de padronização nos critérios de correção. A transição para métodos digitais começou a ganhar força com o desenvolvimento de ferramentas online, permitindo que a coleta e análise de dados fossem realizadas de maneira mais rápida e precisa.

Com o avanço da tecnologia, os questionários digitais emergiram como uma solução eficiente, facilitando a administração de avaliações em larga escala e oferecendo uma visão mais detalhada sobre o nível de conhecimento dos participantes. Johnson e Lee (2019) [2] apontam que a principal vantagem dos questionários digitais é a possibilidade de personalização, permitindo que as questões sejam adaptadas de acordo com as necessidades específicas do público-alvo.

O papel dos questionários na gestão de pré-requisitos

Os questionários digitais tornaram-se uma ferramenta essencial para a gestão de pré-requisitos em ações de formação, devido à sua capacidade de recolha dados em tempo real e fornecer insights imediatos sobre o nível de preparação dos participantes. Estudos como o de Garcia e Thomas (2021) [3] reforçam a importância da fórmula adequada das perguntas, uma vez que questionários mal elaborados podem levar a respostas imprecisas ou enviesadas, comprometendo a validade dos resultados.

Plataformas de questionários para avaliação de gestão de pré-requisitos

Diversas plataformas digitais têm sido desenvolvidas com o intuito de facilitar a criação, distribuição e análise de questionários para a gestão de pré-requisitos. Entre as mais utilizadas estão o Google Forms, SurveyMonkey e plataformas de LMS (Learning Management System), como Moodle e Canvas. Essas ferramentas oferecem funcionalidades avançadas que permitem, por exemplo, a adaptação dinâmica das perguntas com base nas respostas anteriores dos participantes, conforme discutido por Brown e Williams (2022) [5]. Essas plataformas também são altamente personalizáveis, permitindo que os instrutores ajustem as questões de acordo com o conteúdo específico do curso ou treinamento, além de possibilitar a análise imediata dos resultados. A integração dessas ferramentas com sistemas de feedback em tempo real melhora a capacidade de adaptação dos programas de formação, ajustando-se às necessidades de cada participante.

Apesar das vantagens significativas proporcionadas pelos questionários digitais, a literatura aponta para uma série de desafios que ainda precisam ser superados. Garcia e Thomas (2021) [3] discutem a dificuldade de garantir a validade e a confiabilidade dos questionários, especialmente em contextos onde a participação dos usuários pode ser influenciada por fatores externos, como o ambiente virtual ou a clareza das instruções.

2.1.1 Direções futuras de Pesquisas

A revisão da literatura indica que há um espaço considerável para o desenvolvimento de novos modelos e metodologias para a gestão de pré-requisitos em ações de formação. Estudos futuros devem focar em melhorar a eficácia dos questionários digitais, especialmente no que diz respeito à sua adaptação e personalização. A integração de IA e outras tecnologias emergentes pode abrir novas possibilidades para a criação de questionários mais dinâmicos e responsivos às necessidades dos participantes.

2.1.2 Recolha de informação

A recolha de informação é uma etapa crucial no desenvolvimento da plataforma de questionários para gestão de avaliações de pré-requisitos para ações de formação. Esta fase envolve a definição das tecnologias a serem utilizadas, a compreensão das fontes de dados, a metodologia para recolha de informações, o design da base de dados, o processo de desenvolvimento, e os métodos de validação e verificação da informação. Primeiramente, as tecnologias escolhidas para o desenvolvimento da plataforma foram JavaScript, Express e MySQL Workbench. JavaScript foi selecionado como a linguagem de programação principal para o desenvolvimento da interface da plataforma devido à sua flexibilidade e ampla adoção no desenvolvimento web. Node.js foi escolhido para o backend por suas vantagens em termos de escalabilidade e eficiência na gestão de requisições assíncronas. MySQL Workbench foi utilizado para a gestão de base de dados, sendo uma ferramenta robusta para modelagem, administração e design de base de dados.

Robert M, Groves, consideram que a metodologia de recolha de informação envolveu a definição de requisitos funcionais e não funcionais da plataforma. Os requisitos funcionais foram estabelecidos através de pesquisas, para entender as funcionalidades necessárias da plataforma. Os requisitos não funcionais, como segurança, desempenho e usabilidade, foram definidos considerando as melhores práticas e necessidades específicas do projeto.

O design da base de dados foi uma parte fundamental do projeto. Foram criados esquemas e tabelas detalhadas para suportar as funcionalidades da plataforma, incluindo tabelas de utilizadores, questionários, respostas e avaliações. Diagramas ER (Entidade-Relacionamento) gerados no MySQL Workbench foram utilizados para ilustrar e planejar a estrutura de base de dados, garantindo que todas as relações entre as tabelas fossem corretamente estabelecidas.

O processo de desenvolvimento envolveu tanto o front-end quanto o back-end da plataforma. No desenvolvimento do frontend, JavaScript foi utilizado para construir a interface do

usuário, possivelmente utilizando frameworks ou bibliotecas adicionais, como Angular, para melhorar a eficiência e a experiência do usuário. No desenvolvimento do backend, o Express foi utilizado para configurar o servidor, criar endpoints e implementar a lógica de negócio necessária. A integração com o banco de dados foi feita através da configuração de conexões e a realização de operações CRUD (Create, Read, Update, Delete), garantindo que o Express pudesse se comunicar de forma eficaz com o MySQL.

Os métodos de validação e verificação foram cruciais para garantir a integridade dos dados e a funcionalidade da plataforma. Foram realizados testes unitários e testes de integração para validar as operações de dados e garantir que todos os componentes da plataforma funcionassem conforme esperado. Estes testes ajudaram a identificar e corrigir possíveis erros e assegurar que a plataforma estivesse pronta para uso.

O Express foi escolhido como a tecnologia para o desenvolvimento do backend da aplicação. Destaque que o Express é uma framework para Node.js que proporciona uma estrutura robusta e flexível para construir APIs e servidores web. Explique que o Express facilita o roteamento, o gerenciamento de requisições e respostas, e a integração com base de dados através de middleware.

Em conclusão, a recolha de informação para o desenvolvimento da plataforma envolveu uma combinação de tecnologias bem escolhidas, uma metodologia estruturada para a coleta de dados, um design detalhado da base de dados, um processo de desenvolvimento robusto e métodos rigorosos de validação e verificação. Esses elementos garantiram que a plataforma pudesse atender de forma eficaz às necessidades de gestão de avaliações de pré-requisitos para ações de formação.

2.1.3 Definição da Estrutura para Questionário

A falta de um processo estruturado para a aplicação de questionários na gestão de avaliações de pré-requisitos para ações de formação representa um problema significativo em muitas organizações. A eficácia dos programas de formação depende diretamente da capacidade de identificar com precisão as necessidades formativas dos participantes. No entanto, sem um processo estruturado, essa identificação muitas vezes é inadequada, resultando em diversos desafios e consequências negativas.

Um dos principais problemas é que os questionários de avaliação de pré-requisitos, quando não são estruturados, tendem a ser elaborados de maneira inconsistente e sem uma metodologia clara. Isso pode levar à criação de questionários que não cobrem todos os aspetos relevantes dos pré-requisitos, contêm perguntas mal formuladas ou ambíguas e não são aplicados de

maneira sistemática a todos os participantes. Como resultado, os dados recolhidos podem ser imprecisos ou irrelevantes, comprometendo a qualidade das informações necessárias para o planeamento das formações.

Thomas Gilbert, disse que causas dessa falta de estrutura são variadas. Muitas vezes, os responsáveis pela criação dos questionários não possuem o treinamento adequado ou a experiência necessária em metodologias de avaliação. Além disso, a limitação de recursos, tanto em termos de tempo quanto de ferramentas tecnológicas, pode impedir a implementação de um processo estruturado. Em algumas organizações, a pressão para iniciar programas de formação rapidamente leva à negligência na fase de avaliação de pré-requisitos, e a ausência de padrões organizacionais para a criação e aplicação de questionários resulta em abordagens inconsistentes. As consequências dessa falta de estrutura são numerosas e impactam diretamente a eficácia dos programas de formação. Sem uma avaliação precisa dos pré-requisitos, as formações podem não abordar as áreas mais críticas de desenvolvimento dos participantes, resultando em programas que não atendem às reais necessidades dos envolvidos. Isso não só compromete a qualidade das formações, mas também leva ao desperdício de recursos, pois formações mal direcionadas resultam em baixo retorno sobre o investimento de tempo e dinheiro.

Problema da falta de um processo estruturado para a aplicação de questionários na gestão de avaliações de pré-requisitos para ações de formação.

A insatisfação dos participantes é uma consequência direta de programas de formação que não estão alinhados com suas necessidades e expectativas. Quando os conteúdos dos cursos não são relevantes ou o nível do curso não é adequado, os participantes ficam frustrados e desmotivados, o que pode levar a uma menor participação e engajamento nas atividades de formação. Isso, por sua vez, impacta negativamente a qualificação profissional dos funcionários e, conseqüentemente, o desempenho organizacional como um todo.

Exemplos práticos ilustram esses problemas. Em uma grande empresa de tecnologia, a ausência de um processo estruturado na aplicação de questionários levou a uma insatisfação de 40% dos participantes com os programas de formação, resultando em um turnover mais alto e custos adicionais para a organização. De acordo com uma pesquisa da Associação Brasileira de Treinamento e Desenvolvimento (ABTD) 35% das empresas relatam dificuldades significativas na identificação precisa das necessidades de formação devido a questionários inadequados. Além disso, um relatório da McKinsey & Company estima que organizações que não investem em processos estruturados para avaliações de pré-requisitos podem enfrentar uma redução de até 20% na eficácia de seus programas de formação, resultando em perda de competitividade e aumento dos custos operacionais.

Fonte: <https://conecta.abtd.com.br/pesquisa-panorama-2023-2024/>, <https://abtd.com.br/>, https://conecta.abtd.com.br/Pesquisa_Panorama_Obrigado

Para mitigar esses problemas, é essencial desenvolver um processo estruturado para a criação e aplicação de questionários de avaliação de pré-requisitos. Esse processo deve incluir o desenvolvimento de padrões e melhores práticas, com diretrizes claras para a criação de questionários baseadas em evidências científicas. A capacitação dos profissionais envolvidos na elaboração e aplicação dos questionários é fundamental, garantindo que eles tenham formação adequada para realizar essa tarefa de maneira eficaz. A utilização de ferramentas tecnológicas pode facilitar a recolha e análise de dados, tornando o processo mais eficiente e preciso. Além disso, é importante implementar um sistema de monitoramento e avaliação contínua para garantir a eficácia do processo e fazer ajustes conforme necessário.

Em conclusão, a falta de um processo estruturado para a aplicação de questionários de avaliação de pré-requisitos é um problema significativo que impacta diretamente a eficácia das ações de formação nas organizações. Ao abordar esse problema de maneira sistemática e implementar soluções adequadas, as organizações podem melhorar significativamente a qualidade de suas formações, aumentar a satisfação dos participantes e obter melhores resultados no desempenho organizacional. A implementação de um processo estruturado é essencial para garantir que os programas de formação realmente atendam às necessidades dos participantes e contribuam para o desenvolvimento contínuo e a competitividade das organizações.

Tabela 1. Problema do Processo estruturado para aplicação de questionário

Problema da falta de um processo estruturado para a aplicação de questionários na gestão de avaliações de pré-requisitos para ações de formação	
Aspetto	Descrição
Problema	Falta de um processo estruturado para a aplicação de questionários na gestão de avaliações de pré-requisitos para ações de formação
Descrição-do Problema	Questionários elaborados de forma inconsistente, sem metodologia clara, resultando em dados imprecisos ou irrelevantes.
Causas	Falta de conhecimento e experiência dos responsáveis Ausência de padrões organizacionais
Consequências	Identificação inadequada das necessidades formativas Baixa qualidade das formações
Exemplos/Dados	Pesquisa da ABTD: 35% das empresas enfrentam dificuldades devido a questionários inadequados Redução de até 20% na eficácia dos programas de formação
Relevância-da Solução	Desenvolver um processo estruturado que inclua padrões e melhores práticas.
Resultados Esperados	Melhoria na identificação das necessidades formativas Maior alinhamento dos cursos

2.2 Framework Express

O Express é um framework para Node.js que facilita a criação de aplicações web e APIs. Ele é conhecido por sua simplicidade e flexibilidade, tornando o desenvolvimento de backend mais eficiente. Com o Express, é possível criar rotas para gerenciar diferentes endpoints da aplicação, como a criação, leitura, atualização e exclusão de questionários e avaliações. Ele usa middleware para processar requisições e respostas, permitindo a implementação de funcionalidades como autenticação e logging de forma modular.

Ethan Brown disse num dos seus livro que a integração com bancos de dados é simplificada, especialmente quando se utiliza um ORM ou ODM, o que facilita a modelagem e manipulação de dados. O Express também oferece suporte para a criação de APIs RESTful, o que é essencial para a gestão de questionários e avaliações. Além disso, a configuração de segurança e autenticação pode ser gerida através de middleware, garantindo que a aplicação esteja protegida contra acessos não autorizados.

Com sua arquitetura modular e suporte para escalabilidade, o Express é uma escolha robusta para construir um backend eficiente para a plataforma de questionários, oferecendo uma base sólida para a implementação de lógica complexa e para o crescimento futuro da aplicação

Express

O Express é um framework simples para Node.js projetado para simplificar a criação de servidores web e APIs. Ele fornece uma base sólida e flexível para desenvolver o backend de aplicações, oferecendo uma série de características e benefícios que são particularmente úteis para o desenvolvimento de plataformas como a sua, que gerencia questionários e avaliações. O Express não impõe uma estrutura rígida de projeto, o que permite uma alta flexibilidade na organização do código e na implementação de funcionalidades. Isso é ideal para sua plataforma, pois permite personalizar o backend de acordo com as necessidades específicas de gestão de questionários e avaliações. A simplicidade do Express permite uma configuração rápida e eficiente, facilitando o início do desenvolvimento.

Uma das principais características do Express é seu sistema de middleware. Middleware são funções que interceptam as requisições e respostas HTTP, permitindo executar código, modificar requisições e respostas, e terminar o ciclo de solicitação-resposta. Isso é útil para adicionar funcionalidades como:

Autenticação: Usando middleware, você pode proteger endpoints da API, garantindo que apenas usuários autenticados possam acessar certas funcionalidades.

Validação de Dados: Middleware de validação de dados pode ser usado para validar e sanitizar os dados recebidos nas requisições antes que eles sejam processados.

Logging: Middleware pode registrar informações sobre requisições e respostas, ajudando na depuração e monitoramento da aplicação.

Roteamento: O Express facilita a criação e gerenciamento de rotas. Cada rota pode corresponder a um endpoint específico da API que lida com operações como a criação, leitura, atualização e exclusão de questionários e avaliações. A definição clara e organizada de rotas ajuda a manter o código do backend limpo e estruturado, e permite uma fácil manutenção e expansão da aplicação.

Integração com Bancos de Dados: O Express se integra bem com ORMs (Object-Relational Mappers) e ODMs (Object-Document Mappers), como Sequelize e Mongoose. Essas ferramentas permitem interagir com a base de dados de maneira eficiente, modelando os dados de questionários, perguntas e avaliações. Isso simplifica operações CRUD (Create, Read, Update, Delete), permitindo que você se concentre na lógica de negócio sem se preocupar com detalhes baixos de manipulação de dados

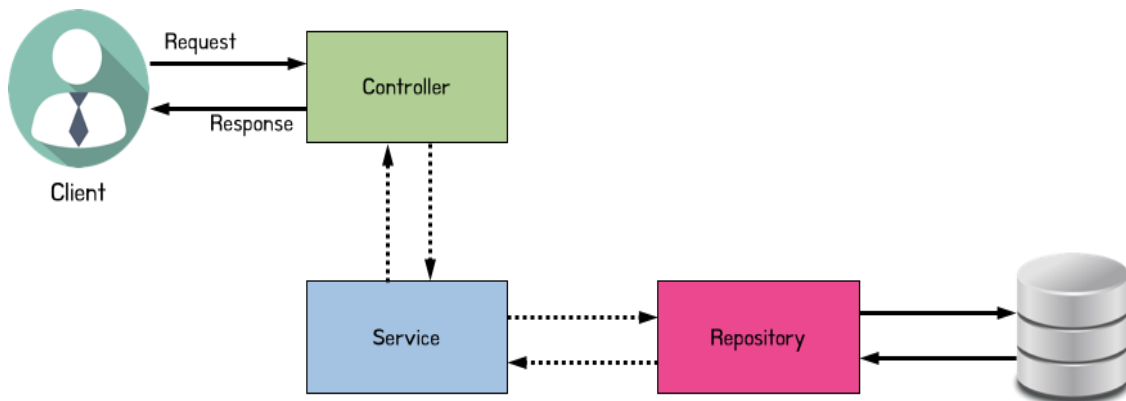


Figura 1. Arquitetura Express (<https://kgk1599.tistory.com/193>)

2.2.1 API REST

Um bom design de API é importante em uma arquitetura de micros serviços, porque toda a troca de dados entre serviços acontece por meio de mensagens ou chamadas de API. As APIs devem ser eficientes para evitar a criação de E/S tagarelas. Como os serviços são projetados por equipas que trabalham de forma independente, as APIs devem ter semântica e esquemas de controlo de versão bem definidos, para que as atualizações não interrompam outros serviços.

API REST, que significa Representational State Transfer Application Programming Interface, é um estilo arquitetónico amplamente utilizado para criar serviços web que interagem com sistemas distribuídos. Baseada em conceitos do protocolo HTTP, uma API REST usa métodos HTTP para realizar operações sobre recursos, que são representações de dados acessíveis via URLs.

Os principais métodos HTTP utilizados em API REST são GET, POST, PUT e DELETE. O método GET é usado para solicitar dados de um recurso específico, enquanto o POST é utilizado para criar novos recursos. O PUT serve para atualizar recursos existentes, substituindo a representação atual do recurso, e o DELETE é usado para remover um recurso.

Na arquitetura REST, recursos são identificados por URL, e as interações com esses recursos são realizadas através dos métodos HTTP mencionados. Cada recurso pode ter múltiplas representações, como JSON ou XML, que definem como os dados são transmitidos entre o cliente e o servidor. JSON é o formato mais comum devido à sua simplicidade e fácil integração com JavaScript.

De acordo com Martin Fowler, uma característica fundamental das API's REST é que elas são stateless, ou seja, cada requisição do cliente ao servidor deve conter toda a informação necessária para que o servidor entenda e processe a requisição. Isso significa que o servidor não armazena estado sobre as requisições do cliente entre diferentes chamadas, o que melhora a escalabilidade e a independência das interações.

Além disso, a API REST devem seguir os princípios de recursos e representações. Recursos são abstrações de dados, como usuários ou produtos, e são manipulados através de URL. As representações desses recursos, como JSON ou XML, são formatadas e interpretadas conforme solicitado. Isso permite que diferentes clientes, independentemente de suas tecnologias ou plataformas, interajam com a API de maneira consistente.

A API REST também utilizam códigos de status HTTP para indicar o resultado das

operações. Por exemplo, um código de status 200 indica sucesso, 201 é usado para indicar que um recurso foi criado com sucesso, 204 é retornado quando uma requisição foi bem-sucedida, mas não há conteúdo a ser retornado, enquanto 400 e 404 indicam erros de cliente e recursos não encontrados, respectivamente.

Outro conceito importante em API REST é a utilização de hipermídia como um motor de estado de aplicativo (HATEOAS). Isso significa que uma API REST deve fornecer links para outras ações e recursos relacionados no contexto da resposta, permitindo que os clientes descubram dinamicamente as funcionalidades da API e naveguem por ela sem a necessidade de conhecimento prévio sobre suas URL.

A segurança da API REST pode ser gerida através de autenticação e autorização. Métodos comuns incluem autenticação baseada em tokens, como OAuth ou JWT, e autenticação básica com nome de usuário e senha. Além disso, práticas como o uso de HTTPS para criptografar dados em trânsito e a implementação de rate limiting para prevenir abusos são fundamentais para proteger as API.

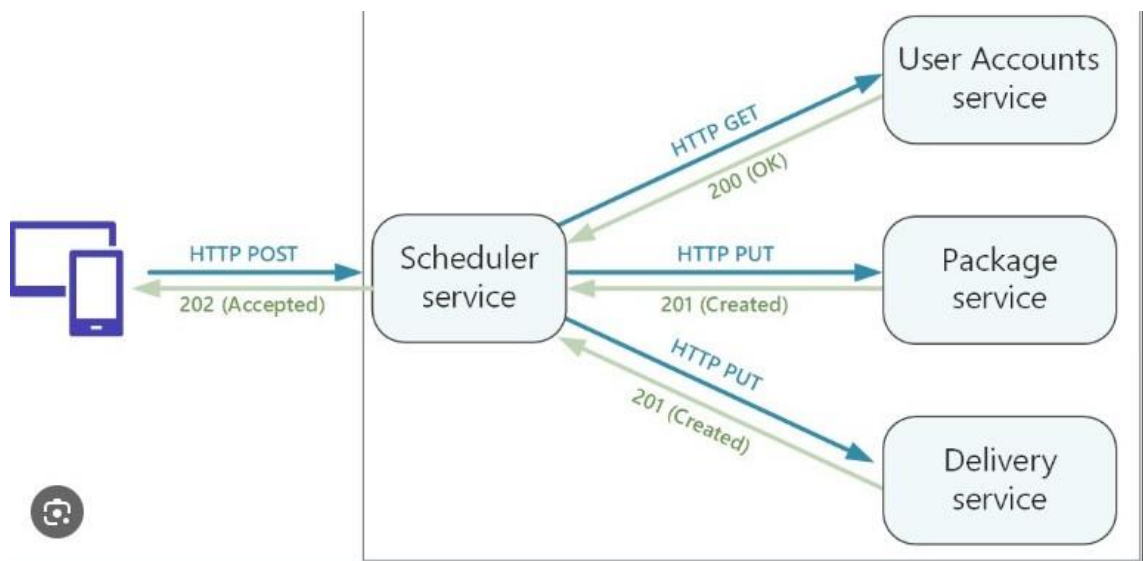


Figura 2 Arquitetura API REST - [\(Conceção da API - Azure Architecture Center | Microsoft Learn\)](#)

2.2.2 Model View Controller

Express não é apenas um framework, é um ambiente de execução JavaScript multiplataforma especialmente construído para lidar com o servidor. Além disso, o padrão MVC (Model-View-Controller) em Node.js permite que você inclua JavaScript tanto no lado do cliente quanto no lado do servidor.

Utilizando o Express, você tem a oportunidade de usar JavaScript em todos os três componentes do MVC. No MVC, o modelo opera no seu servidor e a visualização no navegador; no entanto, ambos são escritos em JavaScript.

Express permite a execução de JavaScript fora do navegador, tornando-o uma ferramenta poderosa para desenvolvimento de servidores. Este ambiente de execução é conhecido pela sua eficiência e escalabilidade, especialmente em aplicações que exigem operações de I/O intensivas, como servidores web. Uma das características mais notáveis do Express é seu modelo de concorrência baseado em eventos, que permite a gestão de múltiplas conexões ao mesmo tempo. Diferente dos servidores tradicionais que usam threads para cada conexão, Express utiliza uma única thread não bloqueante, o que resulta em uma melhor performance em aplicações que necessitam de alta simultaneidade.

Kyle Simpson, afirmou padrão MVC ajuda a organizar o código de maneira modular, separando a lógica de negócios (modelo), a interface do usuário (visualização) e o controle do fluxo de aplicação (controlador). Com JavaScript no lado do servidor e do cliente, o código pode ser reutilizado, aumentando a eficiência e reduzindo o esforço de desenvolvimento. A separação das preocupações facilita a manutenção e atualização do código, já que alterações em uma parte do sistema não afetam diretamente as outras. Com Node.js e MVC, é possível desenvolver aplicações de maneira ágil e rápida, aproveitando as capacidades do JavaScript e a estrutura organizada do MVC.

Imagine que você está desenvolvendo uma aplicação web para gerenciamento de tarefas. Com Node.js e MVC, você pode definir a estrutura dos dados das tarefas no servidor utilizando JavaScript e um banco de dados SQL como MongoDB. Pode criar a interface do usuário em JavaScript, utilizando frameworks como React ou Angular, que renderizam a visualização no navegador. E pode implementar a lógica de negócios para adicionar, remover e atualizar tarefas, gerenciando as requisições do usuário e comunicando-se com o modelo.

Node.js possui um ecossistema rico de bibliotecas e frameworks, como Express.js para o desenvolvimento de servidores web, Socket.io para comunicação em tempo real, e Sequelize para ORM (Object-Relational Mapping). Este ecossistema robusto acelera o desenvolvimento e oferece soluções prontas para muitos desafios comuns. A comunidade ativa de desenvolvedores Node.js contribui para um crescimento contínuo e melhoria do ambiente. Existem inúmeros tutoriais, fóruns e repositórios de código aberto disponíveis, que ajudam novos desenvolvedores a aprender e avançar rapidamente.

Em resumo, Express, com seu ambiente de execução eficiente e compatibilidade com o padrão MVC, oferece uma plataforma poderosa para o desenvolvimento de aplicações modernas, escaláveis e de alto desempenho. A utilização de JavaScript em todo o stack de desenvolvimento simplifica o processo e abre inúmeras possibilidades para inovações e melhorias contínuas.

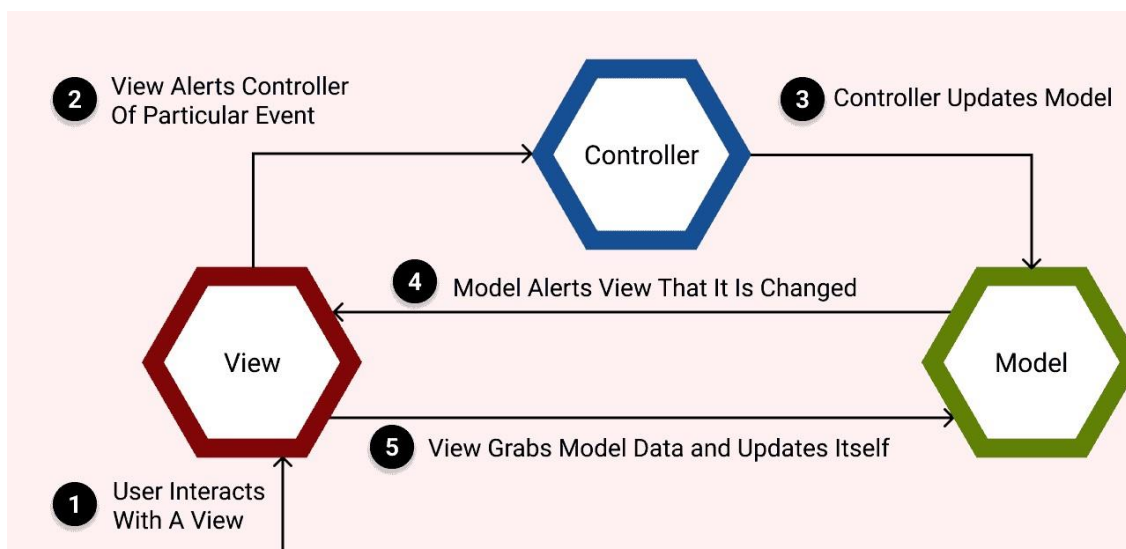


Figura 3. Fluxo MVC Node js. Extraído de (<https://www.esparkinfo.com/blog/build-node-js-mvc-application.html>)

Express e a segurança de informação

A segurança em aplicações construídas com o framework Express é fundamental para proteger dados e garantir a integridade do sistema. O Express é flexível e permite a integração de várias práticas e ferramentas de segurança para proteger a aplicação contra diferentes tipos de ataques e vulnerabilidades. Ethan Brown e David Herron chegaram a conclusão para proteger contra ataques como Cross-Site Scripting (XSS), é crucial sanitizar e filtrar as entradas de usuário. Utilizar bibliotecas como xss-clean ajuda a prevenir que scripts maliciosos sejam injetados e executados no navegador. Outro risco é o Cross-Site Request Forgery (CSRF), que pode ser mitigado com o middleware csrf, que adiciona um token único a cada requisição para verificar a legitimidade da origem da solicitação. O Express oferece várias ferramentas para implementar segurança na aplicação. Usando pacotes como helmet para proteger headers HTTP e express-rate-limit para limitar o número de requisições, você pode fortalecer a segurança da aplicação. A autenticação pode ser implementada com middleware que verifica tokens ou sessões, garantindo que apenas usuários autorizados possam acessar certas funcionalidades da plataforma.

A proteção contra SQL Injection é vital, especialmente ao manipular dados no banco de dados. Usar ORMs como Sequelize pode ajudar, pois eles protegem contra esse tipo de ataque automaticamente. Também é importante validar e sanitizar todos os dados recebidos antes de processá-los para garantir que não contenham comandos maliciosos.

Outra consideração importante é a proteção de senhas. Utilizar bcrypt para hash e comparação de senhas é uma prática recomendada. Nunca armazene senhas em texto claro; sempre utilize técnicas de hashing seguras para proteger as credenciais dos usuários. Para proteger a aplicação contra ataques de força bruta, a implementação de rate limiting é uma boa prática. O pacote express-rate-limit pode limitar o número de requisições que um usuário pode fazer em um determinado período, o que ajuda a prevenir abusos e ataques de força bruta.

A comunicação segura entre o cliente e o servidor deve ser garantida através do uso de HTTPS. Configurar o servidor para aceitar apenas conexões HTTPS ajuda a proteger os dados contra intercetações. Além disso, configurar o header HTTP Strict Transport Security (HSTS) força os navegadores a se conectarem apenas através de HTTPS.

2.3 Abstração da base de dados

Na programação orientada a objetos, os aplicativos são contruídos usando objetos. Estes objetos refletem objetos da vida real e eles contem dados. O modelo relacional que é amplamente utilizado para armazenamento de dados em aplicativos, utiliza tabelas para armazenar dados e linguagens de manipulação de dados para interagir com esses dados. No desenvolvimento de sistemas de gestão de avaliações de pré-requisitos para ações de formação, a abstração da base de dados desempenha um papel fundamental na organização, manipulação e acesso eficiente aos dados. Neste projeto, a combinação de MySQL, Node.js e Sequelize foi escolhida para implementar uma camada de abstração robusta e escalável, garantindo que a manipulação de dados seja realizada de forma segura e eficiente.

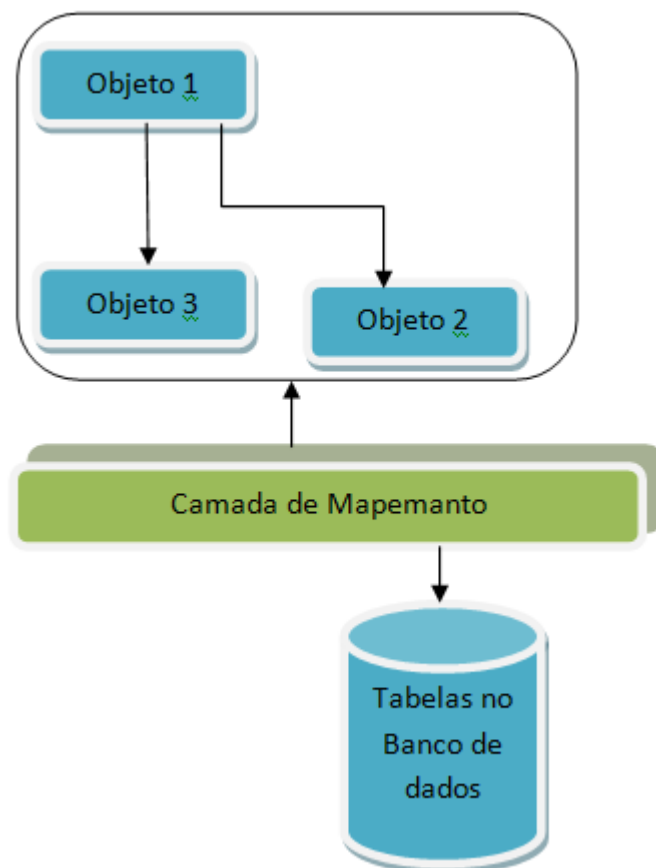


Figura 4 Mapeamento Objeto-Relacional

A estrutura da base de dados foi projetada para suportar a complexidade inerente ao processo de gestão de avaliações, incluindo a definição de pré-requisitos, questionários, respostas e a relação entre participantes e suas avaliações. Para facilitar a interação com a base de dados, foi utilizada a biblioteca Sequelize, um ORM (Object-Relational Mapping) para Node.js, que permite mapear tabelas do banco de dados para modelos JavaScript. Essa abstração simplifica o código e torna as operações de CRUD (Create, Read, Update, Delete) mais intuitivas.

2.3.1 Base de dados MySQL

O MySQL continua a evoluir e a se adaptar as novas necessidades tecnológicas. A Oracle lança regularmente novas versões do MySQL, incorporando melhorias de desempenho, segurança e novas funcionalidades para atender as demandas dos desenvolvedores e empresas. O MySQL teve sua origem na Suécia, em 1994, com a equipe de desenvolvedores Michael Widenius, David Axmark e Allan Larsson. Desde sua primeira versão em 1995, o MySQL se destacou por ser um banco de dados de código aberto com alto desempenho e custo-benefício, tornando-se uma solução popular em todo o mundo. A aquisição pela Sun Microsystems e, posteriormente, pela Oracle, ajudou a garantir sua continuidade e desenvolvimento, enquanto o surgimento de forks como o MariaDB mostra o dinamismo da comunidade de código aberto em torno do MySQL.

Ao longo dos anos esta base de dados conquistou reputação por sua arquitetura comprovada, confiabilidade, integridade de dados. MySQL é um dos sistemas de gerenciamento de banco de dados relacionais mais populares e amplamente utilizados no mundo. Sua popularidade se deve, em grande parte, à sua confiabilidade e robustez. Com mais de duas décadas de desenvolvimento e aprimoramento, o MySQL tem um histórico comprovado de estabilidade e desempenho, o que o torna uma escolha segura para armazenar e gerenciar dados críticos em aplicações. O MySQL possui uma excelente compatibilidade com Node.js através de bibliotecas e módulos, como MySQL e sequelize. Essas bibliotecas permitem uma integração fluida entre base de dados e o ambiente de execução Node.js, oferecendo suporte para operações de CRUD (Create, Read, Update, Delete) e outras funcionalidades essenciais.

Neste contexto da plataforma, SQL é essencial para definir e organizar a estrutura dos dados. Usamos SQL para criar tabelas que armazenam informações sobre questionários, perguntas, respostas e avaliações. Essa organização facilita o armazenamento sistemático e a recuperação dos dados necessários para o funcionamento da plataforma. A capacidade de realizar consultas complexas é uma das principais vantagens do SQL. Você precisará extrair informações específicas dos dados armazenados para gerar relatórios e análises sobre o

desempenho dos usuários e a eficácia dos questionários. SQL permite que você execute consultas sofisticadas para agregar, filtrar e ordenar dados conforme necessário para essas análises, proporcionando insights valiosos sobre o cumprimento dos pré-requisitos e a efetividade das ações de formação.

Além disso, SQL oferece mecanismos para garantir a integridade dos dados através de restrições como chaves primárias e estrangeiras, o que é vital para assegurar que as relações entre os dados sejam mantidas corretamente e que as entradas sejam válidas. Isso garante a precisão e confiabilidade dos questionários e avaliações na plataforma. A manutenção e atualização dos dados também são facilitadas por SQL. Com SQL, você pode facilmente adicionar novos questionários, atualizar informações existentes e remover dados desatualizados, garantindo que a plataforma permaneça atualizada e relevante. SQL também contribui para a escalabilidade e performance da plataforma. Ele permite a otimização de consultas e a criação de índices para melhorar a eficiência do banco de dados. Isso é crucial para garantir que a plataforma possa lidar com grandes volumes de dados e um número crescente de usuários sem comprometer a velocidade e a eficácia das operações. A segurança dos dados é outro aspecto importante que SQL aborda. Você pode definir permissões e roles para controlar o acesso aos dados, garantindo que apenas usuários autorizados possam acessar ou modificar informações sensíveis. Isso ajuda a proteger as informações dos usuários e a manter a integridade dos dados na plataforma. SQL é amplamente suportado por diversas tecnologias e ferramentas, facilitando a integração com o backend da sua plataforma. Muitas ferramentas e bibliotecas de backend, como ORMs (Object-Relational Mappers), utilizam SQL para interagir com base de dados relacionais, tornando a integração com seu sistema mais fluido e eficiente.

SQL permite a realização de backups e a recuperação de dados, garantindo que as informações armazenadas possam ser preservadas e restauradas em caso de falhas ou perda de dados. Esta funcionalidade é crucial para garantir a continuidade dos serviços e a proteção contra a perda de dados importante.

Capítulo 3

Processo de questionários para gestão de avaliações de pré-requisitos para ações de formação.

3.1 Desenho e modelo conceptual do questionário

Propõe-se um processo de questionário que adota uma abordagem orientada a perguntas e respostas para registar os dados que integram a plataforma de gestão de questionários, em conformidade com as regras definidas para a sua construção. Os dados dos utilizadores, bem como as perguntas e respostas, serão armazenados num modelo de dados relacional (estruturado), utilizando o sistema de gestão de bases de dados MySQL.

Para solucionar o problema do formulário das questões, respostas e resultados e outros campos, foi desenhada uma plataforma suportadas nas entidades (definidas) apresentadas na figura 6, onde deveram ser interpretadas e implementadas no Framework com as seguintes descrições:

A entidade quis permite a criação de nova quis, que é constituída por dois campos a serem preenchido, Esses campos são: nome, tópico. Na entidade categoria ajuda a categorizar a quis sessão, nesta entidade tem um campo nome. Entidade Quis Secção tem relação com as entidades categoria e quis, e ajuda a categorizar quis sessão, onde cada formulário de question tem uma sessão para diferenciar das outas sessões. Os formulários de questões atribuída cada utilizador terá datas deferentes para que os dados pareçam como historial para diferenciar com os questionários criados num outro dia.

Quis question é a entidade que permite criação de perguntas onde tem três campos deferentes, que são preenchidos a pergunta e valores das perguntas e a sessão em que este formulário de pergunta fara parte, depois de cridas as perguntas o formulário da quis question permite adicionar as respetivas respostas. Neste mesmo formulário tem quatro campos que são: Tipo de respostas: que permite o administrador do sistema escolher o tipo de resposta; se é texto ou imagem, e se a opção da resposta é certa ou errada. Os formulários serão preenchidos repetidamente para diferentes utilizadores, o formulário de pergunta tem um botão que permite o administrador remover qualquer pergunta ou resposta. O utilizador terá um questionário de múltiplas escolhas onde vai escolher apenas a respostas certas. Quando o administrador for preencher o formulário de questões e respostas para um utilizador terá que preencher sessão para determinar para qual sessão o questionário faz parte assim como categoria.

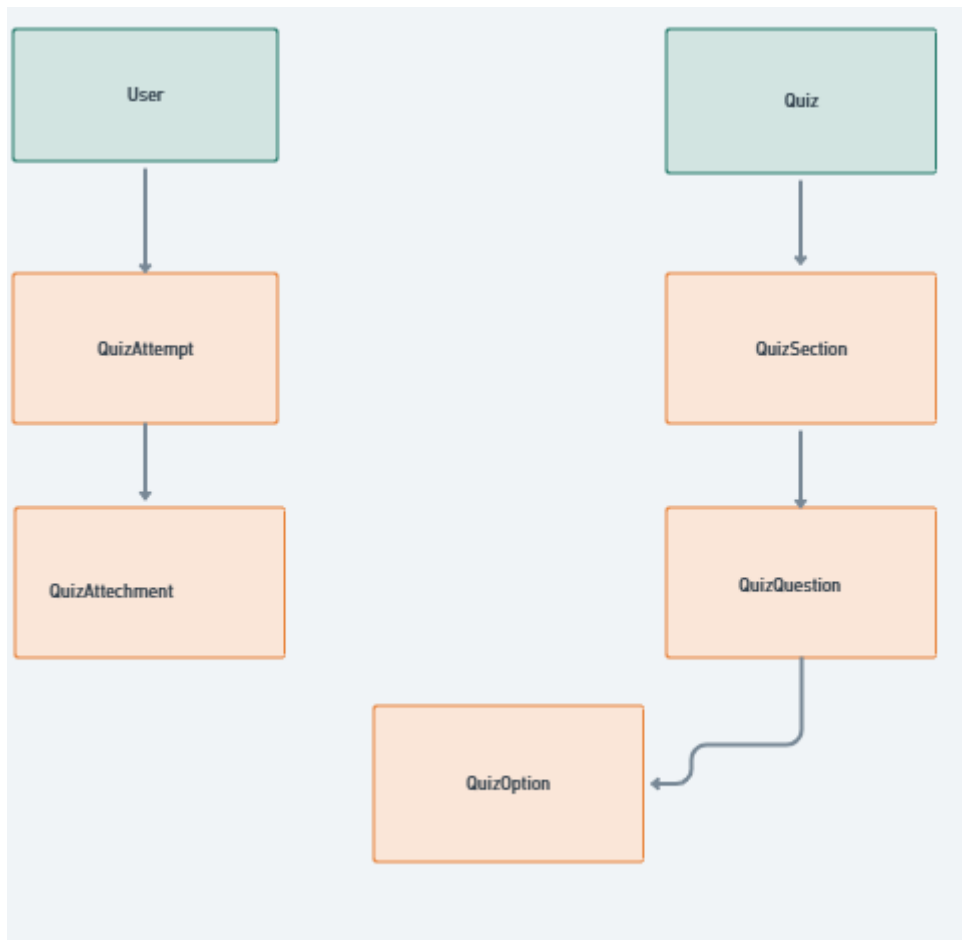


Figura 5 Modelo das principais entidades do sistema.

Os modelos de entidades desenvolvidos neste trabalho, como são apresentados na figura 6, foram implementados, respeitando os relacionamentos entre os modelos previamente, (conceitualmente), definidos. Na figura 6, se podem observar as tabelas criadas com o recurso do framework Node js Express.

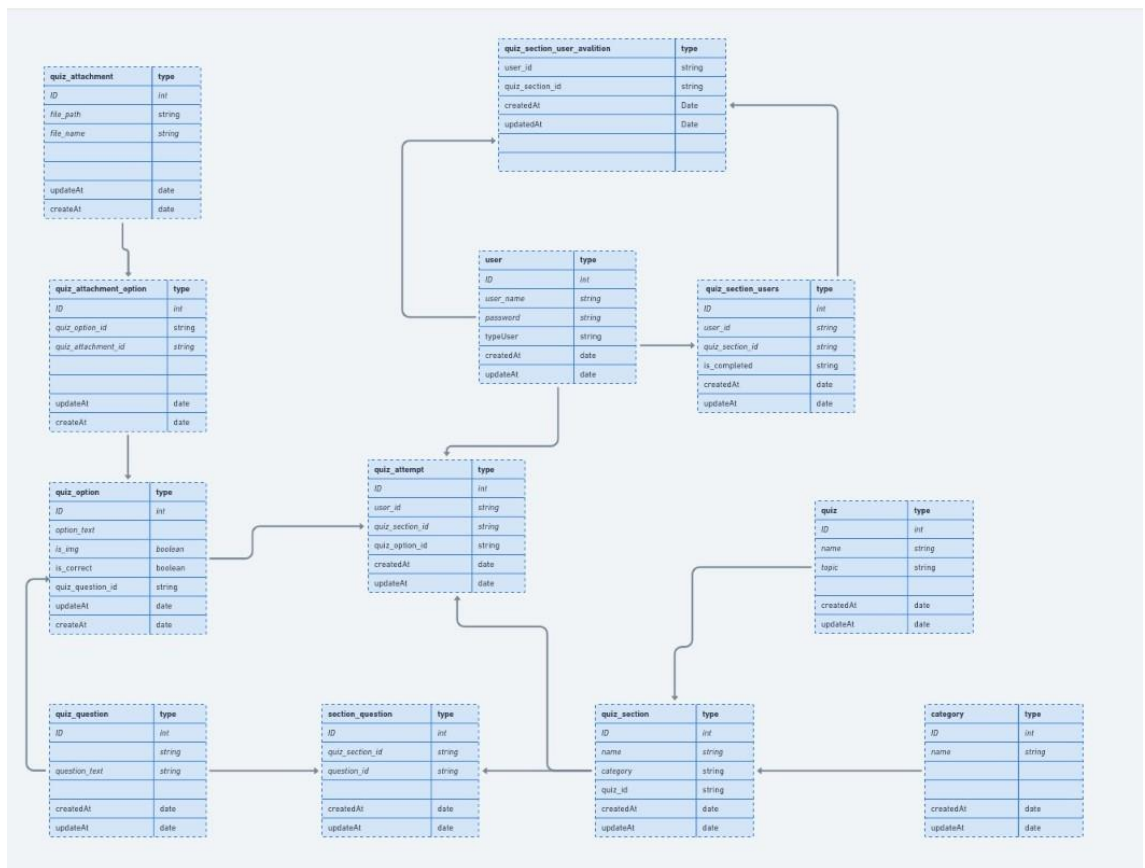


Figura 6 Modelo de entidade relacional

3.1.1 Implementação da plataforma de questionário para ações de formação

Para a implementação da plataforma de questionário, dada a versatilidade e complexidade dos formulários serem criados se decidiu desenvolver um modelo / estrutura cliente – servidor, onde o site contendo as tecnologias definidas para a parte servidor – back-end, para a parte do cliente - front-end. Na figura 9, se descrevem as tecnologias que foram selecionadas, como base para implementar a plataforma de questionário.

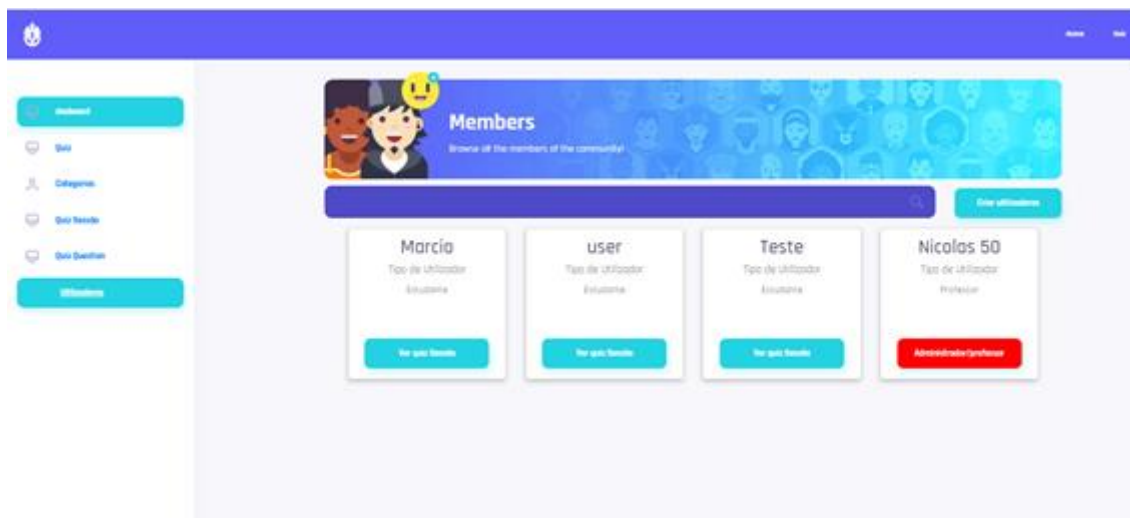


Figura 7 Gestão de utilizadores e administração do sistema.

3.1.2 Configurações de utilizadores e permissões da plataforma

Para garantir o controlo de utilizadores de acesso, foi necessário desenvolver um controle de autenticação, para garantir que houvesse dois tipos de perfis de usuários diferentes: administrador de sistema e um utilizador (perfil mais básico) onde o utilizador apenas é atribuído um questionário de perguntas com as suas respetivas opções de respostas onde ele apenas responde o questionário. Neste sentido foi utilizado o import {loginUserDTO} from”../dto/logoutUserDTO, que contem os modelos login e logout, conforme demonstrado no código abaixo:

```
Export class loginUseruseCase
Implements use<case<loginUserDTO,Promise<Result<userModel>>>
{
```

Node.js com Express é uma combinação poderosa para o desenvolvimento de back-offices, que são interfaces administrativas ou painéis de controlo usados por administradores ou outros usuários para gerenciar diversas operações, como controle de dados, usuários, permissões e funcionalidades específicas da aplicação. Essas interfaces permitem uma administração eficiente do sistema, facilitando o acesso a diferentes recursos de forma organizada e segura. No caso de sistemas educacionais, perfis como Professor (Admin) e estudante já são

amplamente conhecidos e implementados. O controlo desses perfis é gerido através do campo `type_user` na tabela `user` no banco de dados. Esse campo define o tipo de usuário, permitindo que o sistema diferencie as permissões e acessos para cada tipo de perfil, garantindo que administradores tenham controlo total do sistema enquanto os estudantes têm acesso limitado às funcionalidades pertinentes ao seu papel.

3.2 Arquitetura

O frontend é a interface do usuário (UI), onde os usuários interagem com a plataforma, respondem aos questionários e visualizam os resultados. Ele se comunica com o backend por meio de chamadas HTTP (API REST). Para o “back-end” foi desenvolvido um servidor Web para aplicação e executá-lo, e uma base de dados para armazenar as informações. Como o nome sugere, o back-end trabalha na parte de trás da aplicação, escrevendo programas que garantam que o servidor, a aplicação Web e a base de dados funcionam harmonicamente.

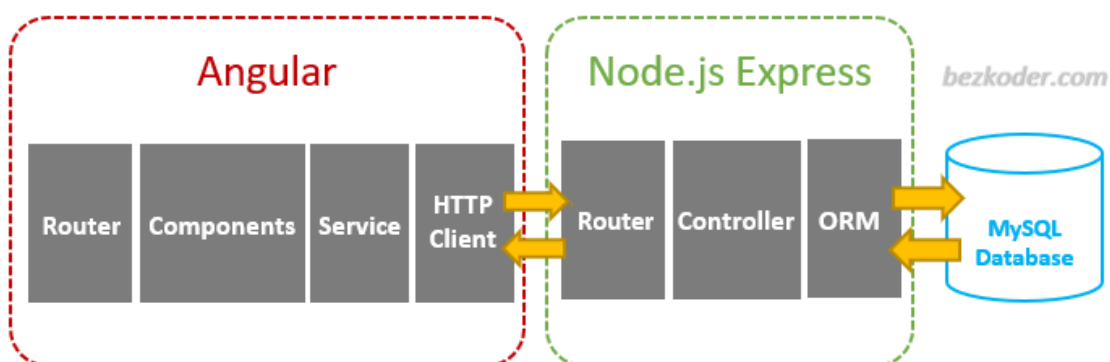


Figura 8 - Diagrama de Arquitetura Proposto

Apesar de ainda estarmos nesta fase inicial, objetivo destas experiências é o de conseguir simular a comunicação através da api, tentando sempre correlacionar com a realidade da funcionalidade da plataforma. O utilizador a cessar a plataforma por meio do frontend desenvolvido em Angular.

O frontend faz uma requisição HTTP para o backend (Node.js com Express), pedindo informações, como um questionário ou enviando respostas de avaliações. O backend processa essa requisição, faz validações (ex: validação de pré-requisitos) e consulta ou grava dados no banco de dados MySQL. O backend retorna os resultados para o frontend, que exibe essas informações ao utilizador.

3.2.1 Tecnologias

Existem diversos modos de comunicação a acontecer de forma sequencial, porém, o principal acaba por ser a comunicação entre o browser e o servidor NodeJS através da Api rest (http). Estes api rest são usados em diversos contextos, especialmente quando há a necessidade de manipular dados de forma escalável e eficiente. A sua comunicação ocorre sobre o protocolo HTTP e apresenta algumas características de requisições e respostas independentes, sem a necessidade de uma conexão persistente ou negociação contínua.

Uma das principais vantagens do modelo REST é a sua natureza stateless, o que significa que cada requisição do cliente ao servidor é tratada de forma isolada. Ou seja, o servidor não precisa manter o estado da conexão entre diferentes requisições, o que reduz a carga sobre os recursos do servidor e facilita a escalabilidade horizontal. Isso permite que sistemas distribuídos, como plataformas de grande porte, possam lidar com milhões de requisições simultâneas, uma vez que a escalabilidade se torna mais eficiente, com o balanceamento de carga entre diferentes servidores. Outro ponto relevante é que a API REST usa amplamente o formato JSON (JavaScript Object Notation) para a troca de dados. Este formato é leve e de fácil leitura tanto para humanos quanto para máquinas, o que o torna ideal para aplicações web modernas.

Ao enviar e receber dados em JSON, tanto o cliente (navegador) quanto o servidor conseguem processar as informações rapidamente, o que melhora a performance da aplicação. Além disso, a API REST permite o uso de métodos HTTP padronizados, como GET, POST, PUT, e DELETE, cada um com um propósito específico. Outro aspecto que fortalece o uso da API REST em sistemas modernos é a sua independência tecnológica. Como a API REST são baseadas em HTTP, elas podem ser consumidas por qualquer tipo de cliente, seja ele um navegador, um aplicativo móvel, ou mesmo outro serviço backend. Essa flexibilidade permite que diferentes tipos de dispositivos e aplicações possam interagir com o servidor Node.js de forma consistente e padronizada.

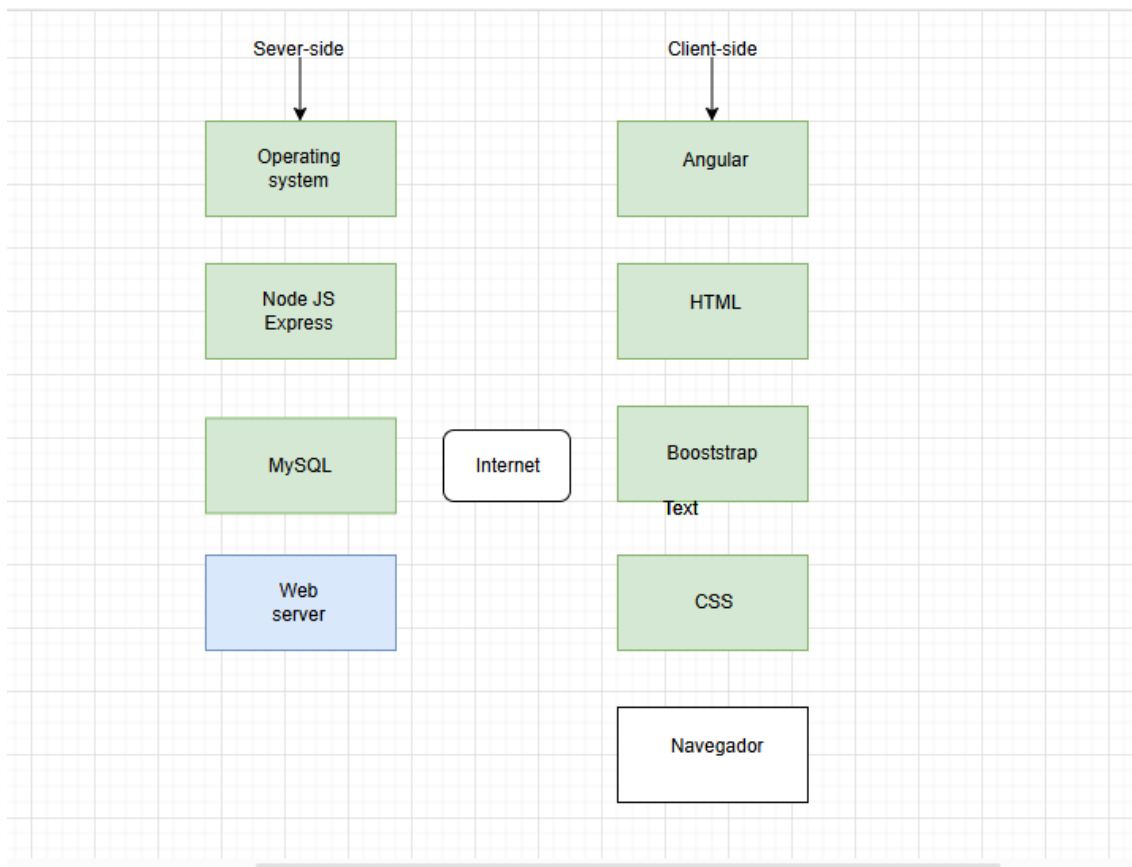


Figura 9 Pilha de tecnologias, adaptadas.

3.2.2 Fluxo de operações da plataforma

Para compreender melhor o funcionamento da plataforma de questionários, foi elaborado um diagrama de fluxo de atividades (Figura 9), que ilustra a fase inicial de cadastro. Nesse estágio, tanto o administrador quanto os usuários realizam suas primeiras ações dentro do sistema. O administrador é responsável por registrar novos usuários, criar e configurar os formulários de perguntas e respostas, e atribuir questionários de avaliação aos utilizadores com base nas áreas de formação designadas.

A Figura 9 destaca que, após o cadastro, o utilizador recebe um questionário composto por perguntas com várias opções de resposta. A sua tarefa consiste em selecionar a resposta correta para cada pergunta no formulário. Esse processo assegura que o utilizador esteja preparado para as formações propostas, facilitando a gestão de avaliações e garantindo que os pré-requisitos sejam cumpridos de forma eficiente.

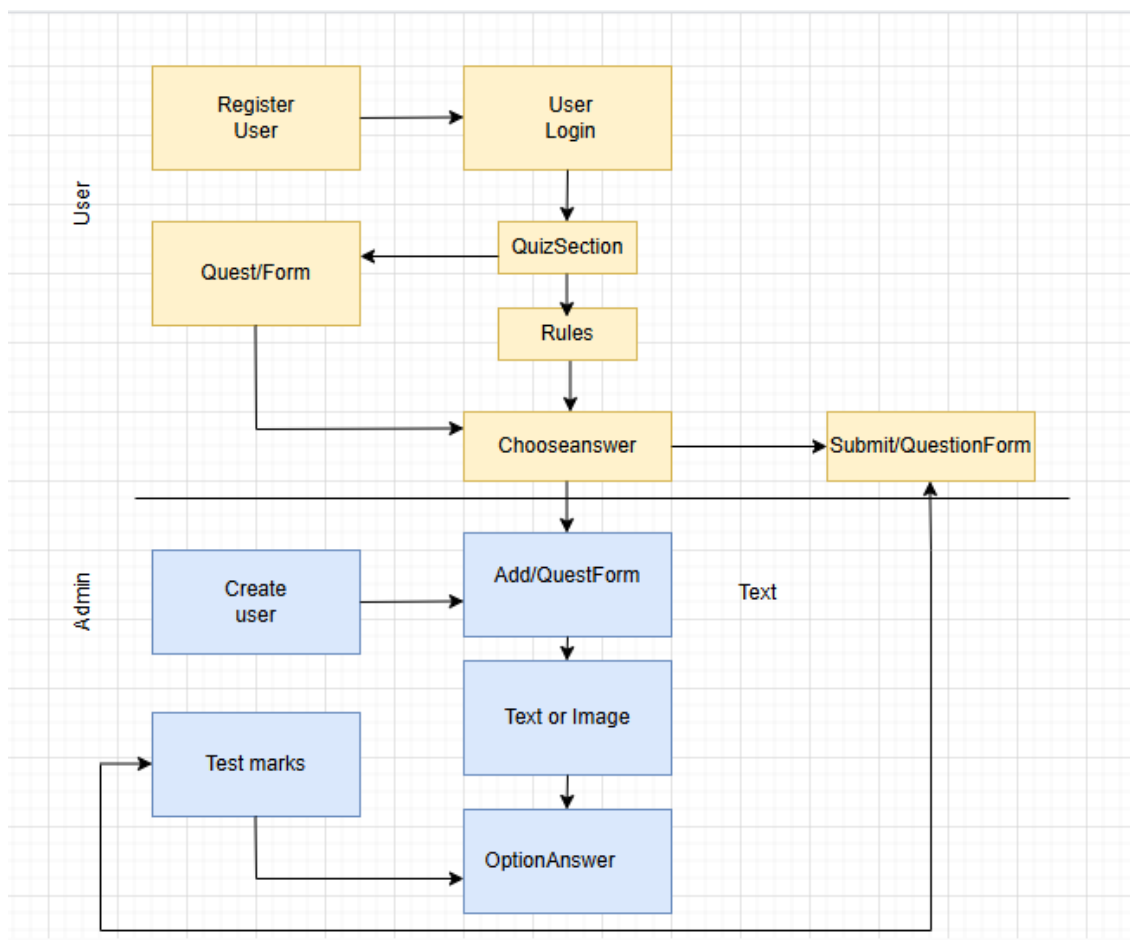


Figura 10. Fluxo de operações da plataforma.

3.3.3 Menu interativo

Para acedermos a todos os ecrãs de cadastro e formulários, foi necessário implementar o menu site, desenhado e feito a medida, com base na interação com os utilizadores.

- Administrador – Registar e faz parte da administração do site, o acesso BackOffice.
- Utilizador – parte da realização da avaliação que recebe.

Para criação do menu, foi necessário a criação de tabelas/ modelos na base de dados que utiliza o recurso a framework. Node js para a construção do servidor e express para gerenciamento de rotas e requisições. A criação de uma API Restful que manipula dados usado em um banco dedos como MySQL, Express é um framework para Node.js que facilita a construção de servidores web e API. Ele lida com requisições do cliente, acessa bancos de dados, processa lógica de negócio, e envia respostas de volta para o cliente Angular.

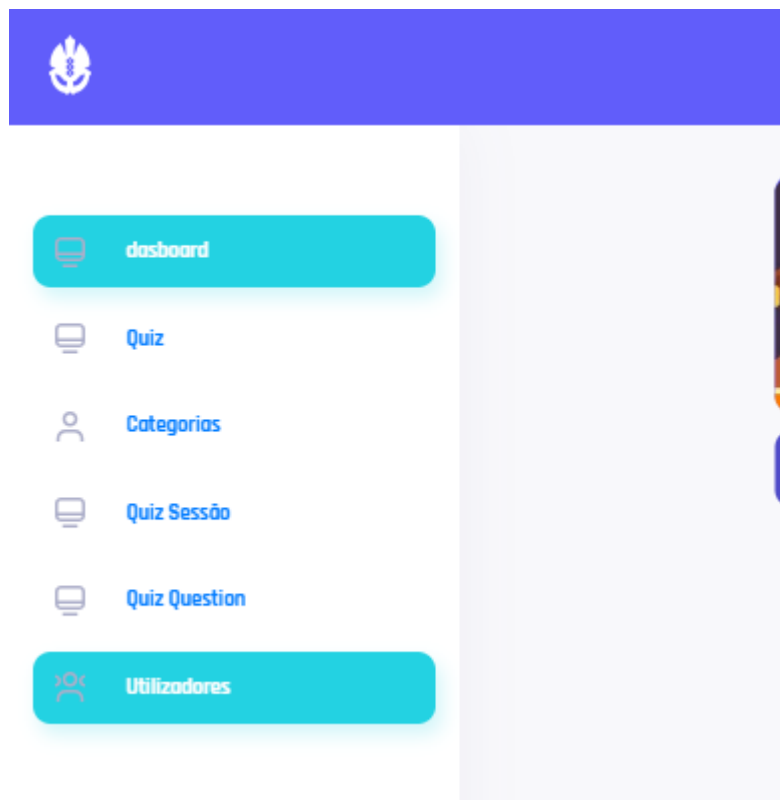


Figura 11- Menu Interativo

Para controlar o acesso aos menus da plataforma, foram criados dois grupos de utilizadores: `admin_member` e `user_member` (ver Figura 11).

O grupo `admin_member` possui privilégios para visualizar e interagir com todos os menus e submenus da plataforma. Já os membros do grupo `user_member` têm acesso restrito apenas à área do questionário, onde podem visualizar, responder e rever as perguntas e respostas que lhes foram atribuídas. Além disso, podem acessar o total de questionários respondidos e o número total de perguntas por meio de um dashboard disponível na área do administrador. Para gerenciar as permissões na plataforma, foi criada uma página de login que verifica as credenciais do utilizador e, com base no grupo a que pertence, atribui as permissões adequadas aos itens da plataforma.

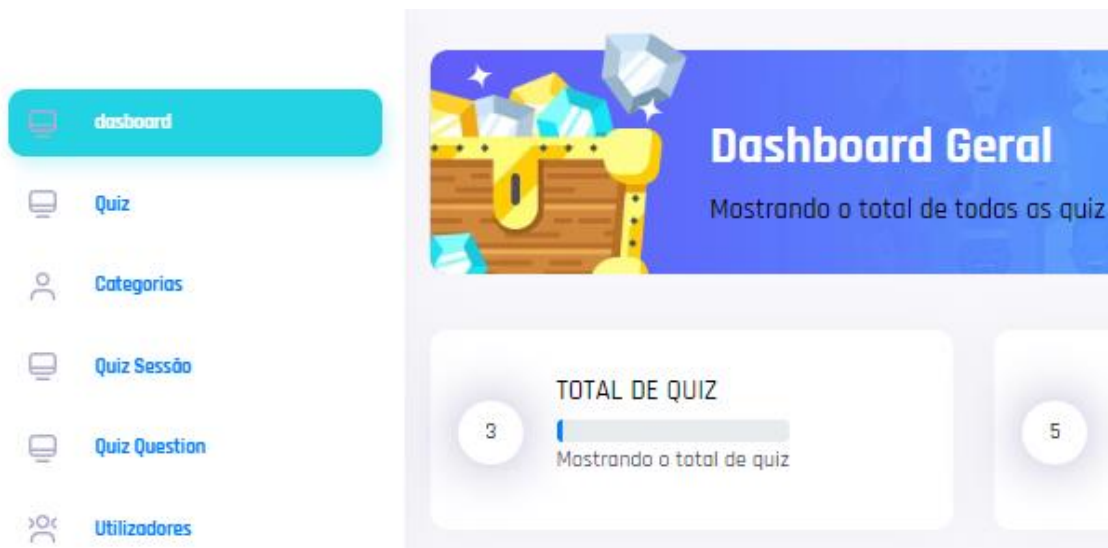


Figura 12. Cadastro de utilizadores e gestão no BackOffice.

3.4 Criação do formulário das questões

Para a criação do formulário de questões (Perguntas), foi necessário desenvolver uma interface específica para a criação de questionários, permitindo a inclusão das respetivas perguntas. Além disso, cada pergunta pode ter um valor atribuído e ser associada à sessão correspondente. Foram também adicionados botões funcionais, como "Adicionar Pergunta", "Criar Questionário" e "Fechar" (ver Figura 12), para facilitar a criação e gestão dos questionários.).

The screenshot shows a web interface for creating a quiz. At the top is a blue header. Below it is a white box with the title "Criar Quiz Perguntas". Inside this box, there are three input fields: "escreve a pergunta", "insere o value", and "Quiz Sessão" (a dropdown menu). To the right of these fields is a dark blue button labeled "Adicionar Perguntas". Below the input fields is a yellow status bar that says "Nenhum A pergunta Adicionada". At the bottom of the white box are two large buttons: a blue "Criar Quiz" button and a red "Fechar" button.

Figura 13. Ecrã de criação de questões.

Conforme mostrado na Figura 13, um dos primeiros ecrãs desenvolvidos foi o de cadastro de utilizadores. Neste ecrã, são exibidos os campos "Username" e "Password", "Cadastra-se" conforme ilustrado na referida figura.

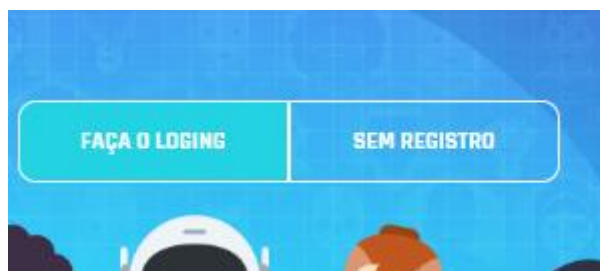
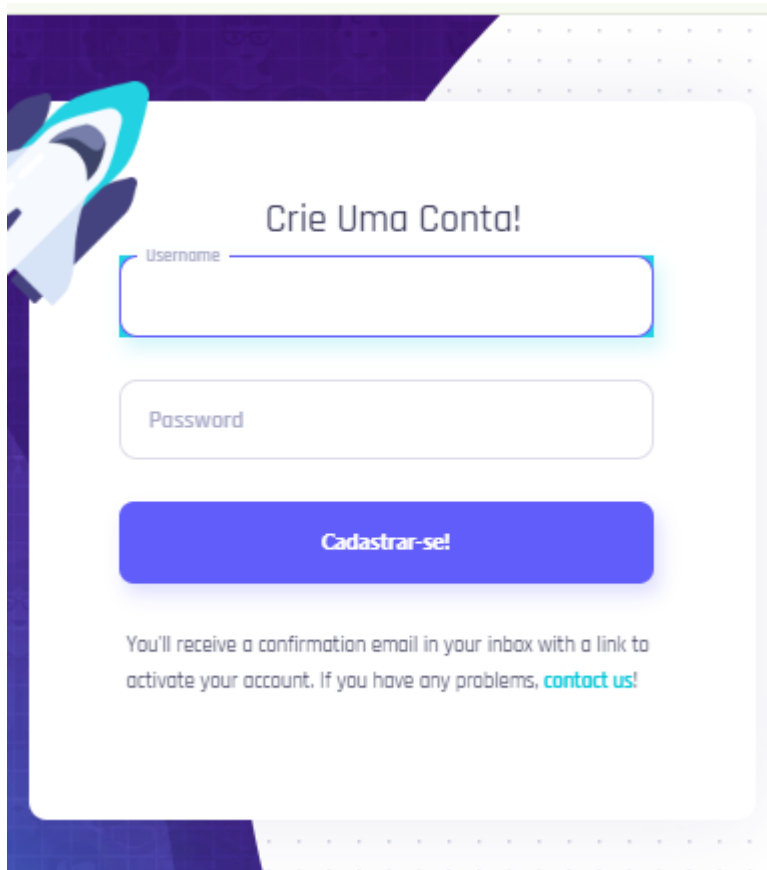


Figura 14 - Ecrã Cadastro e log in.

Para adicionar diferentes tipos de resposta, como texto ou imagem, foi implementada a funcionalidade que permite carregar imagens diretamente a partir do formulário, caso a opção de resposta seja uma imagem. Após o upload da imagem, é possível definir as opções de resposta, identificando quais são corretas e quais são incorretas. No caso de respostas em formato de texto, basta inserir a resposta diretamente no campo correspondente. Essa funcionalidade oferece flexibilidade ao configurar as respostas para cada pergunta do questionário. A Figura 16 ilustra esse processo, destacando as etapas de adição e configuração das opções de resposta.

Pergunta - qual é o nome do Marcelo de casa?

Tipo de resposta:

carregar a imagem:

Correcto: Correcto

Errado: Errado

#	Resposta	Correcto	Remove
---	----------	----------	--------

Figura 15 Ecrã resposta certa ou errada.

No ecrã de "opções de Resposta certa e errada", desenvolvido para permitir a adição de diferentes tipos de respostas em um formulário de cadastro, foi necessário disponibilizar algumas funcionalidades essenciais. Entre elas, destaca-se o botão "Adicionar Opções", que permite inserir novas alternativas de resposta. Além disso, foram incluídos os botões "Correto" e "Errado", que possibilitam a marcação das opções como válidas ou inválidas, respetivamente. Essas funcionalidades são fundamentais para definir e organizar as possíveis respostas dentro do formulário, garantindo que o utilizador possa personalizar os questionários de forma eficiente. A Figura 16 ilustra essa configuração e a disposição dos botões.

Pergunta - qual é o nome do Marcelo de casa?

Tipo de resposta:

carregar a imagem:

Correcto: Correcto

Errado: Errado

#	Resposta	Correcto	Remove
---	----------	----------	--------

Figura 16 Ecrã tipo de resposta

O ecrã mostrado na Figura 16, referente à funcionalidade "Adicionar Opções", foi desenvolvido para suportar a exibição de uma tabela em HTML. Essa tabela permite ao utilizador visualizar, de forma organizada e centralizada, todas as opções de respostas cadastradas para o questionário. Cada linha da tabela representa uma opção de resposta, exibindo-a diretamente

no ecrã, facilitando a sua revisão e gestão. Além disso, foi implementado o botão "Remover", que possibilita a exclusão de qualquer opção de resposta indesejada com um simples clique, tornando o processo de edição mais eficiente e intuitivo. A Figura 15 ilustra a disposição dessas funcionalidades, destacando a tabela de respostas e o botão de remoção.

No módulo de "Quiz Question", é possível adicionar várias opções de resposta para cada pergunta presente no formulário. Essas opções são configuradas de forma que possam ser atribuídas posteriormente a um utilizador específico durante a aplicação do questionário. Cada pergunta pode ter múltiplas respostas, e o sistema permite definir quais são corretas ou incorretas, garantindo flexibilidade na criação dos questionários. A interface, ilustrada na Figura 17, oferece uma visão clara das perguntas e respetivas respostas, facilitando a gestão e personalização de cada questionário de acordo com as necessidades do utilizador.

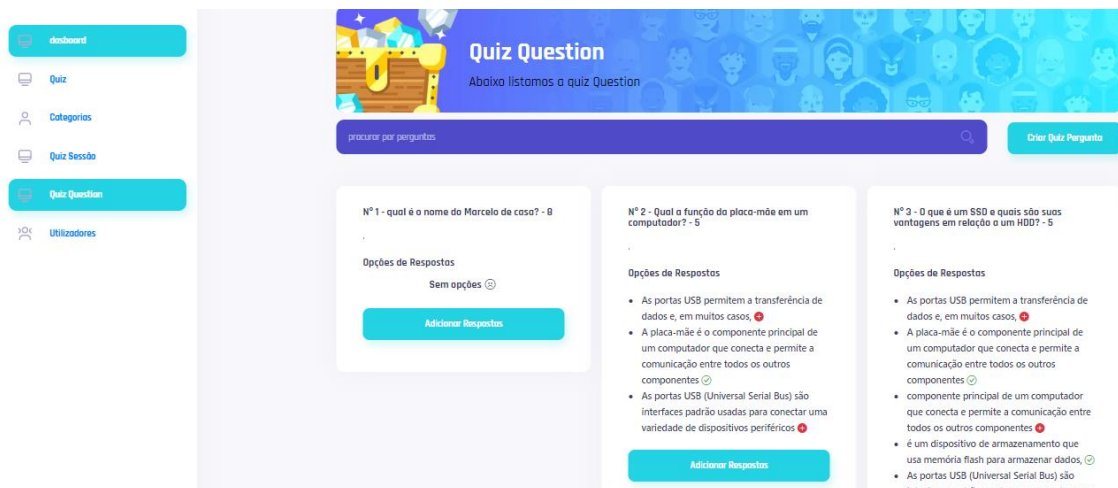


Figura 17 Questionários e opções de resposta.

3.5 Adicionar estudante (utilizador) sessão de avaliação

Para permitir a adição de candidatos a uma sessão de avaliação, foi necessário desenvolver um ecrã específico intitulado "Adicionar Candidato". Este ecrã facilita a inclusão de candidatos ao formulário que contém as questões de avaliação. A interface permite seleccionar um candidato previamente cadastrado e associá-lo a uma sessão de avaliação específica. Após a atribuição, o candidato pode cessar o questionário e responder às questões de avaliação designadas para essa sessão. A Figura 16 ilustra essa funcionalidade, mostrando o processo de vinculação do candidato ao formulário de avaliação e garantindo que ele tenha acesso adequado para realizar a avaliação.

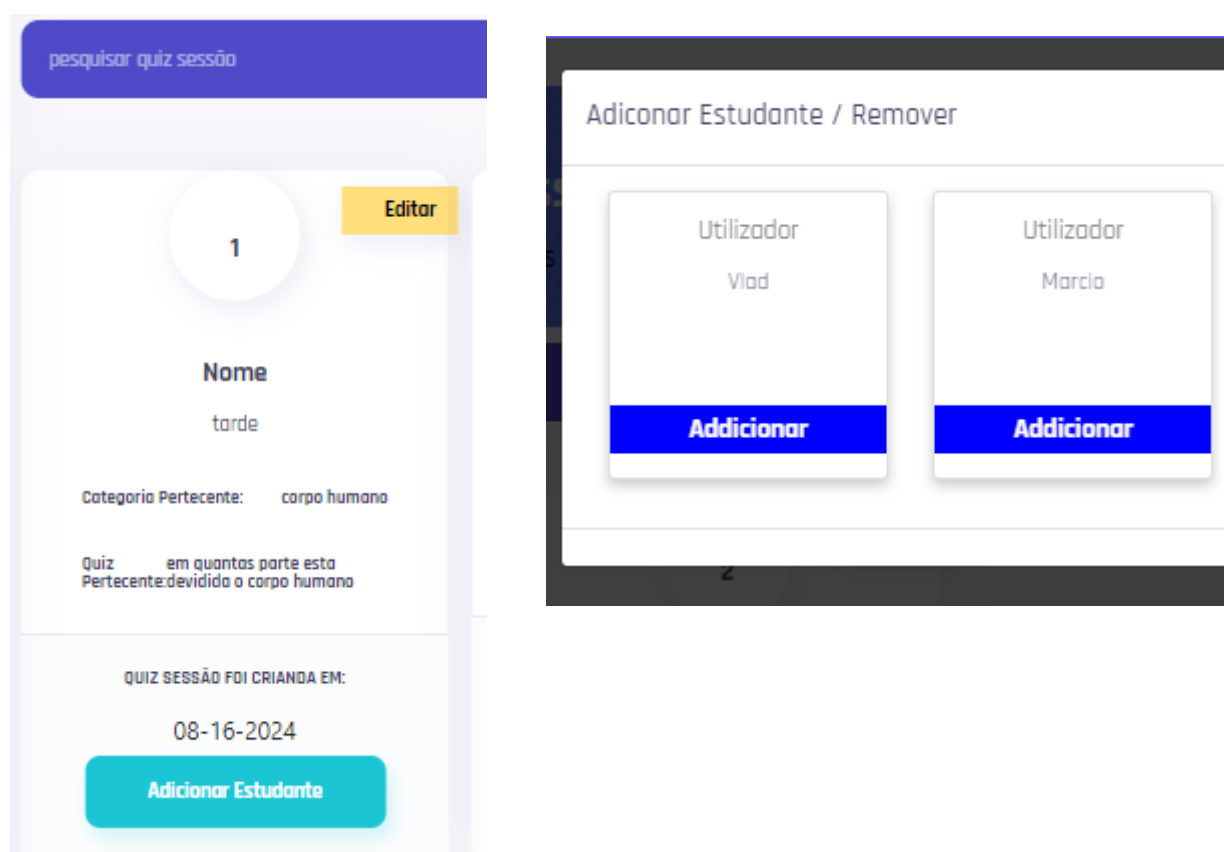


Figura 18. Ecrã adicionar sessão do formulário das questões ao estudante.

Ecrã Fazer a Quiz

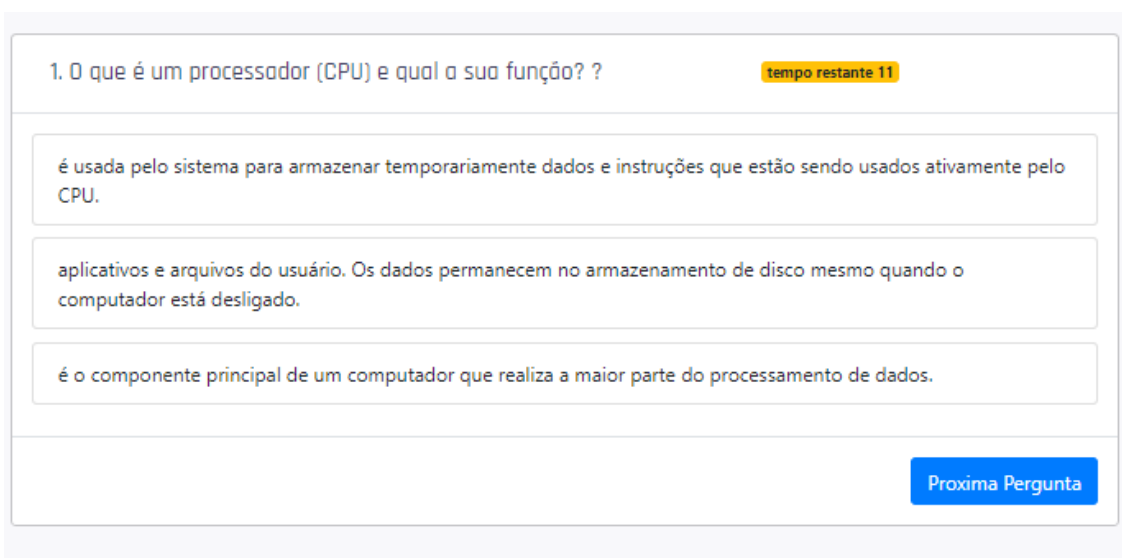
Para que o candidato (estudante) possa realizar o teste de questionário, foi criada uma view específica dentro da sessão de avaliação. Esta view permite que o candidato inicie o teste (quiz) que lhe foi atribuído, com todas as questões configuradas para a sua avaliação. Ao cessar essa interface, o candidato pode visualizar as instruções da avaliação, bem como o tempo disponível, se aplicável, e iniciar o questionário de forma direta. As respostas são registradas e processadas automaticamente pela plataforma, permitindo um acompanhamento em tempo real da performance do candidato. A Figura 18 ilustra essa funcionalidade, mostrando o fluxo de como o candidato interage com a interface para realizar a sua avaliação.



Figura 19 Ecrã Fazer a Quiz.

3.6 Teste do Estudante (Questionário)

Para permitir que o estudante realize o teste, foi desenvolvida uma view específica que exibe todas as perguntas do questionário em formato interativo. Nessa view, cada pergunta é apresentada com suas respectivas opções de resposta, permitindo que o estudante selecione aquela que considera correta. A interface foi projetada de forma a ser clara e intuitiva, facilitando a navegação entre as perguntas e a escolha das respostas. Além disso, a view oferece recursos adicionais, como feedback visual ao selecionar uma resposta e a possibilidade de revisão antes da submissão final. A Figura 19 ilustra essa funcionalidade, mostrando como as perguntas e opções de resposta são organizadas na interface, permitindo ao estudante completar o teste de forma eficiente e organizada.



1. O que é um processador (CPU) e qual a sua função? ? tempo restante 11

é usada pelo sistema para armazenar temporariamente dados e instruções que estão sendo usados ativamente pelo CPU.

aplicativos e arquivos do usuário. Os dados permanecem no armazenamento de disco mesmo quando o computador está desligado.

é o componente principal de um computador que realiza a maior parte do processamento de dados.

Proxima Pergunta

Figura 20 Início do questionário do estudante.

Regras para o quis

Dentro do formulário de teste do estudante, foi necessário desenvolver uma view específica que apresenta as regras a serem seguidas antes de iniciar o teste de avaliação. Esta view fornece informações essenciais, como o tempo limite para a conclusão do teste, a pontuação necessária para aprovação, e orientações sobre como responder às perguntas. Além disso, as regras incluem informações sobre a possibilidade de revisar as respostas e a importância de seguir o código de conduta durante a avaliação. A interface é clara e organizada, permitindo que o estudante compreenda facilmente as expectativas antes de começar o teste. A Figura 21 ilustra esta funcionalidade, destacando a apresentação das regras e sua relevância para garantir que o estudante esteja preparado para o teste.

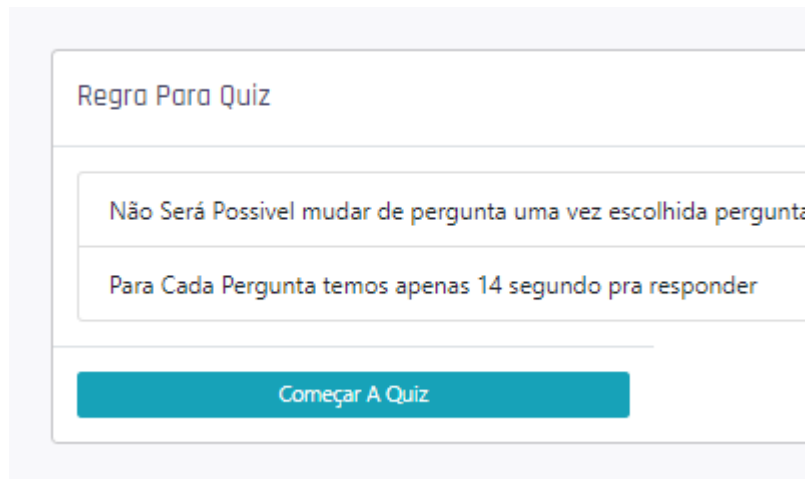


Figura 21 Ecrã regras da quis.

Resultado

Para a visualização dos resultados do teste de avaliação, foi desenvolvida uma tabela interativa utilizando uma combinação de JavaScript, HTML e CSS. Esta tabela apresenta de forma clara e organizada os resultados do questionário respondido pelo estudante. Um botão intitulado "Ver Resultado" foi implementado, permitindo que tanto o estudante quanto o administrador acessem facilmente as informações sobre o desempenho do candidato. Na tabela, são exibidos detalhes cruciais, incluindo o total de perguntas corretas, o total de perguntas incorretas e a pontuação final obtida pelo estudante. Além disso, a interface é projetada para ser intuitiva, com cores e ícones que ajudam a destacar os resultados, facilitando a compreensão do desempenho.

Essa funcionalidade não apenas fornece feedback imediato ao estudante, mas também permite ao administrador acompanhar o desempenho dos candidatos de forma eficaz. A Figura 22 ilustra esta interface, mostrando a disposição da tabela de resultados e os elementos visuais que contribuem para uma melhor experiência do utilizador.

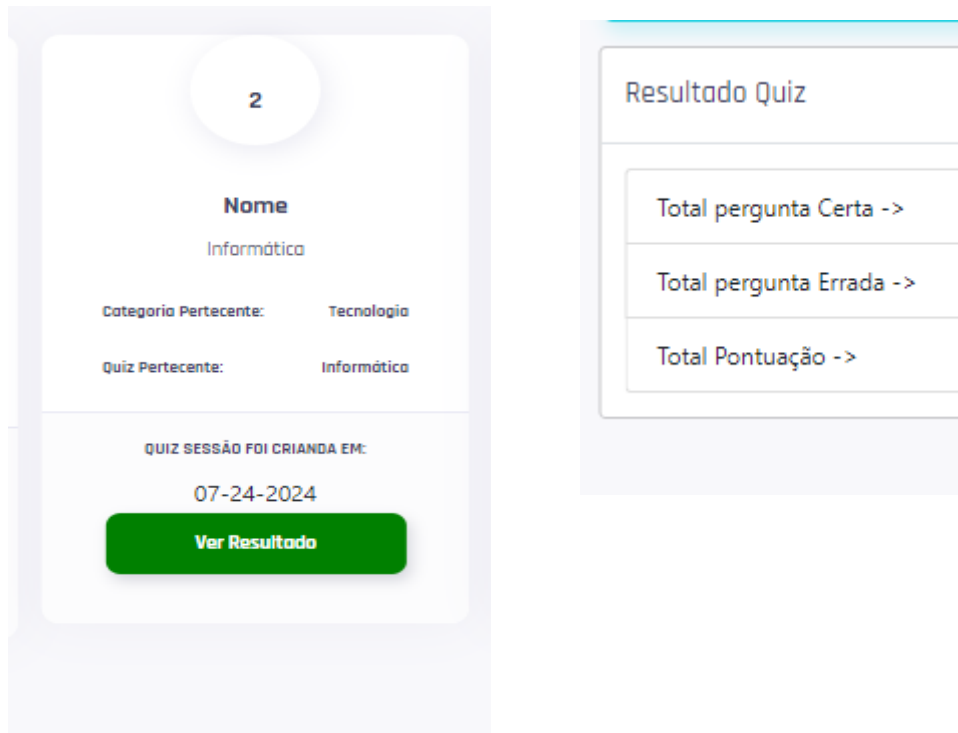


Figura 22 Ecrã consultar resultado final do teste

Gravar as respostas e perguntas e resultados do teste

Para gravar os dados das respostas e perguntas, foi criada uma solução que utiliza o recurso Angular juntamente com Express js.

Para gravar as respostas na base de dados, foi criada uma função Express:

```
import { db } from "../../utils/db.server";
import { saveQuizAttemptedInterface, saveQuizAttemptedInterfaceRepo } from
"../usecase/saveQuizAttemptedByUser/saveQuizAttemptedByUserUseCase";
```

```
export class QuizAttemptedRepository implements saveQuizAttemptedInterfaceRepo {

    public async listByUserAndQuizSection(data:Omit<saveQuizAttemptedInterface,
"option_id">): Promise<any> {

        try {

            const quiz_attempt = await db.quiz_attempt.findMany({
```



```

    }

    }

    async create(data: saveQuizAttemptedInterface): Promise<void> {

        try {
            await db.quiz_attempt.create({
                data: {
                    user_id: data.user_id,
                    quiz_option_id: data.option_id,
                    quiz_section_id: data.quiz_section_id
                }
            })
        } catch (error) {
            throw new Error("Falha ao cadastra.");
        }

    }

```

Podemos ver como esta estruturada as pastas onde se encontram as funções para guardar a resposta na figura 24



Figura 23 Funções de perguntas e respostas na pasta.

Na Figura 23, apresentamos a estrutura organizacional da pasta que contém as funções responsáveis por gerenciar o armazenamento das perguntas e respostas do sistema. Este ecrã ilustra como as opções de respostas são criadas e registadas no formulário, destacando a lógica subjacente ao processo de cadastro. É importante ressaltar que apenas os administradores têm acesso a este formulário, garantindo que o processo de inserção e edição das perguntas e respostas seja controlado e seguro. O design da interface é intuitivo, permitindo que o administrador insira novas perguntas e suas respetivas opções de forma eficiente, além de editar ou remover perguntas existentes conforme necessário.

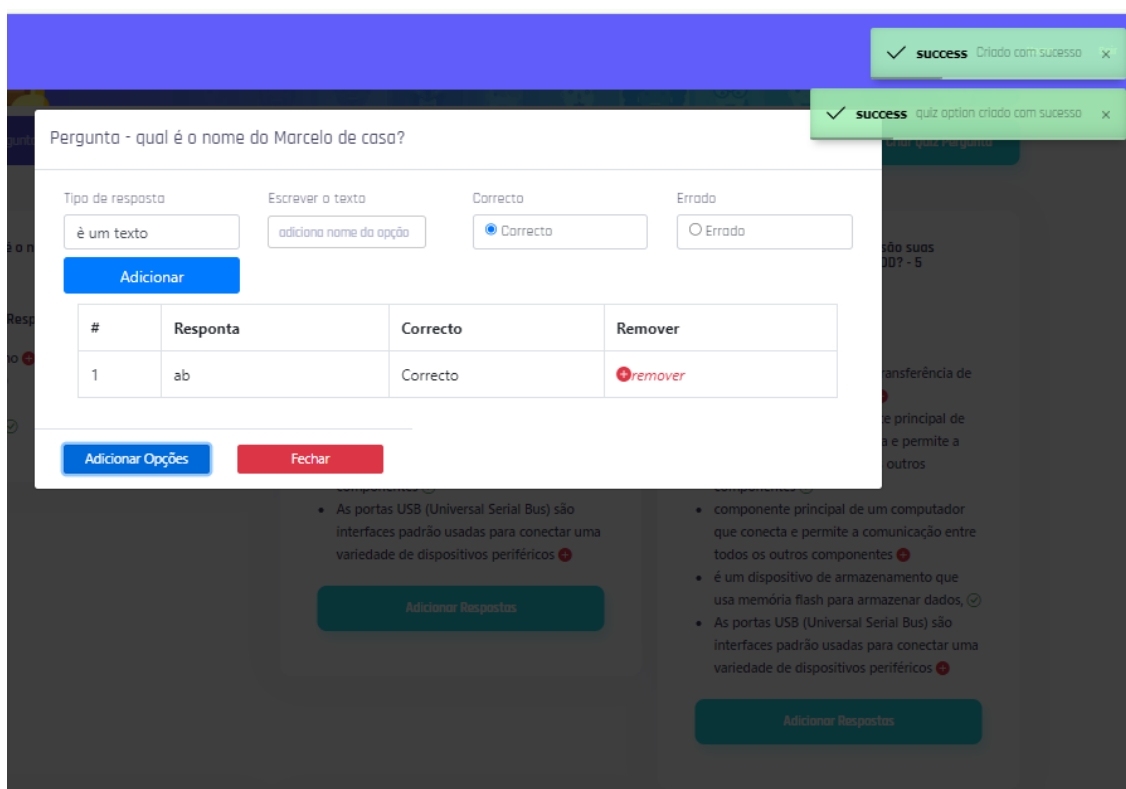


Figura 24 Ecrã com as perguntas e respostas gravadas.

Para armazenar e manipular os dados das avaliações, utilizamos as funcionalidades avançadas do MySQL, como consultas complexas, joins e stored procedures, que permitiram organizar e recuperar os dados de maneira estruturada. Essa abordagem foi essencial para gerenciar os diferentes tipos de dados envolvidos nas avaliações, como as respostas dos usuários, os critérios de avaliação, e os resultados. O uso do MySQL, Angular, e Express proporcionou uma base sólida para o desenvolvimento de uma plataforma robusta e eficiente, capaz de gerenciar as complexidades inerentes à avaliação de pré-requisitos em ações de formação. A combinação dessas tecnologias resultou em um sistema coeso, fácil de manter e escalável, preparado para crescer e se adaptar a futuras necessidades.

Capítulo 4

Análise e discussão dos Resultados

Cenário Experimental da plataforma de questionário

4.1 Testes Funcionais

Com o objetivo de criar um cenário experimental de testes funcionais, o primeiro protótipo da plataforma de questionários foi instalado durante um período de 10 dias em dois computadores diferentes de estudantes do IPS. Um dos estudantes ficou responsável por alimentar, utilizar e avaliar a plataforma de forma experimental. Ao fim deste teste, foi produzido documentos com as propostas de alteração e correção. Os testes tiveram o acompanhamento e orientações aos utilizadores tendo foco na usabilidade de plataforma de forma mais eficiente.

A cada reunião o utilizador relatavam por videoconferência os problemas encontrados e possíveis melhorias. Por fim, era criado um documento com os itens a serem corrigidos melhorados. Como o resultado do teste o primeiro protótipo da plataforma resultou um documento contendo propostas para correção e melhorias, que se descrevem a seguir:

- 1- Alterar "editar" quis, para ter a possibilidade da editar os campos das quis.

É necessário alterar a funcionalidade de "editar quis" para incluir a possibilidade de edição dos diferentes campos e configurações das quizzes. Isso incluiria a possibilidade de modificar perguntas, respostas, tempo limite, instruções, formato das questões (múltipla escolha, verdadeiro ou falso, etc.), além de ajustes de layout ou personalização de design. Essa melhoria visa aumentar a flexibilidade dos usuários na adaptação das quizzes às suas necessidades específicas, garantindo uma experiência mais personalizada e adequada aos objetivos educacionais ou de avaliação. Essa nova abordagem não apenas simplifica a manutenção de quizzes, como também facilita a revisão e atualização de conteúdos de maneira eficiente.

- 2- Alterar e permitir que administrador / professor consegue criar novos utilizadores a partir da área do professor / administrador.

É essencial permitir que administradores e professores tenham a capacidade de criar novos utilizadores diretamente a partir da área administrativa ou da área de professor. Essa funcionalidade proporcionaria maior controle e agilidade no processo de gestão de utilizadores, permitindo que os responsáveis pelo sistema educacional (ou plataforma) possam criar contas de alunos, outros professores, assistentes ou colaboradores, sem a necessidade de recorrer a áreas externas ou procedimentos manuais extensos.

Realizar esses testes ajudará a assegurar que a plataforma funcione corretamente em todos os aspetos críticos, proporcionando uma experiência confiável e eficaz para os utilizadores finais.

4.1.1 Alterações na plataforma após os testes funcionais

Após a realização dos testes funcionais, conseguimos identificar os principais problemas técnicos que precisavam ser resolvidos antes da implementação final da plataforma. Esses testes revelaram não apenas falhas de desempenho, mas também áreas onde a usabilidade e a funcionalidade poderiam ser aprimoradas para oferecer uma melhor experiência ao utilizador.

Criar utilizador

O ecrã de criação de utilizador (cadastro) foi projetado para oferecer a opção de definir o tipo de utilizador a ser cadastrado: utilizador normal (estudante) ou administrador (professor). Essa alteração foi implementada para facilitar a diferença entre os tipos de usuários no sistema, permitindo uma gestão mais eficiente e adequada das permissões e funções.

Além disso, essa modificação foi realizada com base nos feedbacks recebido após os primeiros testes, onde foi identificado que a distinção entre os tipos de utilizadores era essencial para uma operação mais eficaz do sistema. A Figura 25 ilustra essa funcionalidade, mostrando como as opções estão organizadas na interface de cadastro, facilitando a seleção do tipo de utilizador de forma intuitiva.

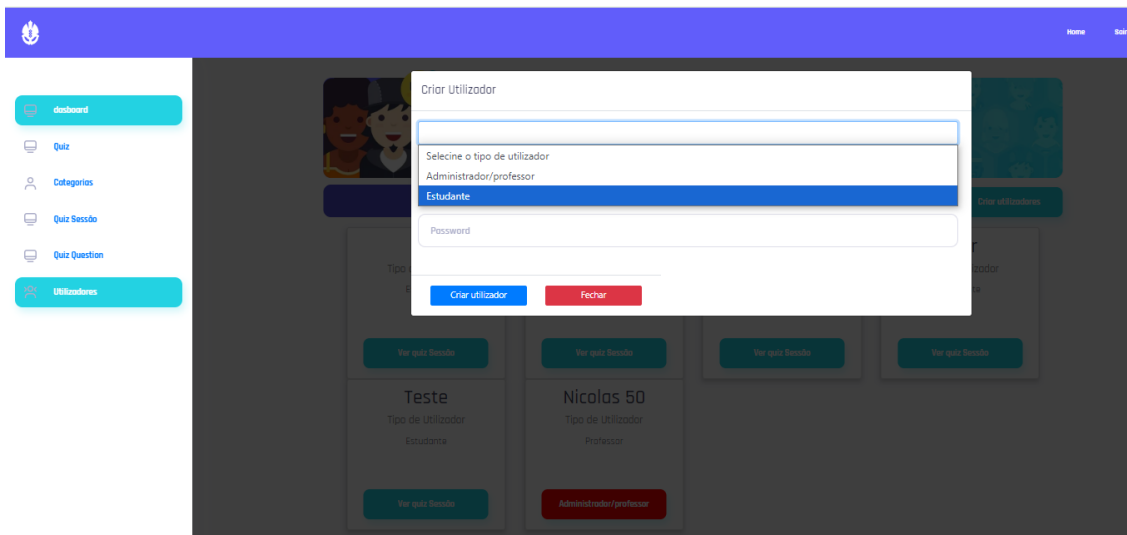


Figura 25 Ecrã cadastrar novo utilizador

Editar campo quis

O ecrã da quiz, em sua versão inicial, não permitia a edição de campos caso houvesse erros de digitação, como no nome do quiz, no tópico ou em outros campos relevantes. Essa limitação dificultava a correção de informações incorretas e gerava insatisfação entre os administradores.

Para resolver essa questão, foi implementada uma nova funcionalidade que permite a edição de qualquer campo diretamente a partir do ecrã do quiz. Para a cessar essa funcionalidade, o administrador deve navegar até o menu na área de administração, onde será exibida uma lista de todos os quizzes disponíveis. No canto superior direito de cada quiz, há um botão identificado como "Editar". Ao clicar nessa opção, o administrador é levado a um formulário onde pode modificar livremente qualquer informação, incluindo o nome do quiz, o tópico associado e outros campos relevantes. Essa atualização não apenas facilita a correção de erros, mas também melhora a eficiência na gestão dos quizzes, permitindo que os administradores mantenham informações precisas e atualizadas. Essa melhoria foi fundamental para otimizar a experiência do usuário e assegurar que a plataforma atenda melhor às necessidades dos administradores, garantindo um ambiente mais funcional e amigável.

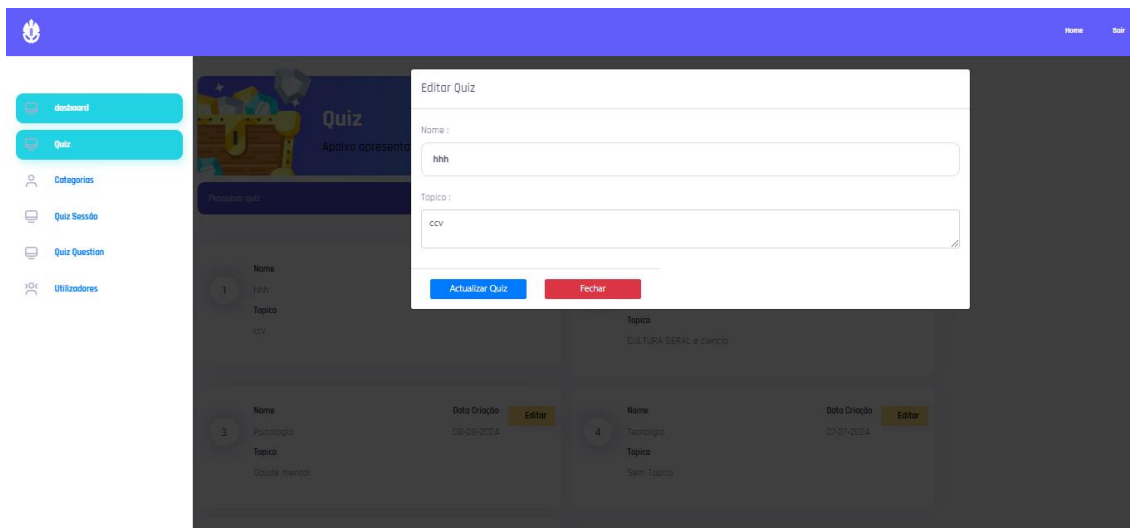


Figura 26 Ecrã editar campo da Quis

4.1.2 Resultados e Discussão

A plataforma de questionários para gestão de avaliações de pré-requisitos para ações de formação foi desenvolvida com sucesso, utilizando Node.js, Express, Angular e MySQL. A funcionalidade principal da plataforma foi implementada conforme o planejado, permitindo a criação, gerenciamento e resposta a questionários de forma eficiente. O desempenho do sistema foi satisfatório, com tempos de resposta rápidos e uma interface de usuário responsivo e intuitiva. A API, construída em Node.js e Express, lidou bem com múltiplas requisições simultâneas, garantindo uma experiência fluida para os usuários.

O banco de dados MySQL gerenciou os dados dos questionários, respostas e usuários de maneira eficaz, garantindo a integridade e segurança das informações. Os testes de usabilidade indicaram que os usuários acharam a plataforma fácil de usar, especialmente na criação e visualização de questionários. As medidas de segurança implementadas, como criptografia de senhas e proteção contra SQL Injection, mostraram-se adequadas para proteger os dados dos usuários.

No entanto, alguns desafios foram identificados. A escalabilidade do sistema, embora demonstrada em testes iniciais, pode exigir otimizações adicionais para suportar um grande número de usuários simultâneos. A segurança, apesar de eficaz até o momento, precisará ser continuamente atualizada para proteger contra novas ameaças. Além disso, a experiência do usuário, embora positiva, pode ser aprimorada com funcionalidades adicionais, como recomendações automáticas de cursos e feedback imediato após a conclusão dos questionários.

A escalabilidade da plataforma permite futuras expansões, como a integração com outras ferramentas de gestão de aprendizagem ou a incorporação de inteligência artificial para análise avançada de dados. Essas melhorias podem tornar a plataforma ainda mais útil para a gestão de pré-requisitos em ações de formação, aumentando seu impacto na educação e no desenvolvimento profissional.

Capítulo 5

5.1 Conclusões e Trabalhos Futuro

A plataforma de questionários para gestão de avaliações de pré-requisitos para ações de formação foi desenvolvida com sucesso, utilizando Node.js, Express, Angular e MySQL. A funcionalidade principal da plataforma foi implementada conforme o planejado, permitindo a criação, gerenciamento e resposta a questionários de forma eficiente. O desempenho do sistema foi satisfatório, com tempos de resposta rápidos e uma interface de usuário responsivo e intuitiva. A API, construída em Node.js e Express, lida bem com múltiplas requisições simultâneas, garantindo uma experiência fluida para os usuários.

O banco de dados MySQL gerenciou os dados dos questionários, respostas e usuários de maneira eficaz, garantindo a integridade e segurança das informações. Os testes de usabilidade indicaram que os usuários acharam a plataforma fácil de usar, especialmente na criação e visualização de questionários. As medidas de segurança implementadas, como criptografia de senhas e proteção contra SQL Injection, mostraram-se adequadas para proteger os dados dos usuários.

No entanto, alguns desafios foram identificados. A escalabilidade do sistema, embora demonstrada em testes iniciais, pode exigir otimizações adicionais para suportar um grande número de usuários simultâneos. A segurança, apesar de eficaz até o momento, precisará ser continuamente atualizada para proteger contra novas ameaças. Além disso, a experiência do usuário, embora positiva, pode ser aprimorada com funcionalidades adicionais, como recomendações automáticas de cursos e feedback imediato após a conclusão dos questionários. A escalabilidade da plataforma permite futuras expansões, como a integração com outras ferramentas de gestão de aprendizagem ou a incorporação de inteligência artificial para análise avançada de dados. Essas melhorias podem tornar a plataforma ainda mais útil para a gestão de pré-requisitos em ações de formação, aumentando seu impacto na educação e no desenvolvimento profissional.

Bibliografia

- [1] Johnston,. (1997). Weaving a syllaweb: Consideration before constructing an on-line syllabus. ERIC document ED 412 937.
<https://www.sciencedirect.com/science/article/pii/S875546150100069X#bBIB26>
- [2] Janet Salmons, <https://www.torrossa.com/it/resources/an/5282196>
- [3] Allen, M.; Bourhis, , E. Comparing student, Vol. 52, No. 3, Maio 2003. Satisfaction with distance education to traditional classrooms in higher education: A meta-analysis. Am. J. Dist. Educ. 2002
<https://www.mdpi.com/2071-1050/12/11/4629>
- [4] Adamson, C. (2012) Learning in a VUCA world, Online Educa Berlin News Portal, November 13.
<https://edisciplinas.usp.br/mod/book/tool/print/index.php?id=2981579>
- [5] Norman E. Gronlund– *Livro: Assessment of Student Achievement, Tenth Edition*
- [6] Renata Lúcia de Souza Gaúna Monteiro- 1º Ten (QOCON Ped) adjunta do Centro de Planejamento e Avaliação do Ensino (CEPLAE) da Escola Superior de Guerra.
<https://recite.unicarioca.edu.br/rccte/index.php/rccte/article/view/71>
- [7] Node js Framework, <https://www.simform.com/blog/best-nodejs-frameworks/>
- [8] Robert,Floyd,https://www.google.pt/books/edition/Survey_Methodology/ctow8zWdyFgC?hl=en&gbpv=1 Survey Methodology
- [9] Thomas F. Gilbert, <https://search.worldcat.org/pt/title/103769424>HumanCompetence Engineering Worthy Performance
- [10] Ethan Brown, , <https://cheatsheetseries.owasp.org/>
- [11] Fluxo MVC, <https://www.esparkinfo.com/blog/build-node-js-mvc-application.html>
- [12] API Rest, [Conceção da API - Azure Architecture Center | Microsoft Learn, https://martinfowler.com/](https://martinfowler.com/) Martin Fowler
- [13] Fluxo MVC, <https://www.esparkinfo.com/blog/build-node-js-mvc-application.html>
- [14] Model <https://www.educba.com/what-is-orm/>

- [15] Bootstrap <https://bootstrapmade.com/>

- [16] David Axmark, https://bookdown.org/labxss/coorte_adm2/trabalhando-com-o-sgbd-mysql.html, The Definitive Guide

- [17] Angular <https://angular.dev/tutorials/learn-angular>

- [18] JavaScript <https://www.w3schools.com/js/DEFAULT.asp>

- [19] Express, <https://www.devmedia.com.br/orm-object-relational-mapping-revista-easy-net-magazine-28/27158>

- [20] Arquitetura, <https://www.bezkoder.com/angular-17-node-js-express-mysql/>

- [21] REST API Architectural, <https://masteringbackend.com/hubs/backend-engineering/rest-api-architectural-constraints>

- [22] REST Overview, <https://www.javaguides.net/2018/06/overview-of-rest.html>

- [23] Restful API Stateless, <https://www.gyata.ai/rest-api/restful-api-stateless>

- [24] Robert M, <https://journals.sagepub.com/doi/abs/10.5005/jp-journals-10021-1104>

Anexo

Hardware e Software utilizado na implementação da plataforma
Código fonte disponível no GitHub, não público:

Hardware utilizado na instalação
Mínimo de 4,00 gb de memória RAM
System Type: 64 bit operating system, x64-based processor
Intel (R) core (TM) i5 – 6200U CPU @ 2.30GHz 2.40 GHz
Version: 22H2

Software Requirements

Node 18
Angular 13
Express 4.19.2
HTML 5
Css 3
Prisma