

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

ANDERSON SOUSA DUFECH CASTILHOS

**Ambiente para a Realização de
Questionários Dinâmicos Baseados na TRI**

Monografia apresentada como requisito parcial
para a obtenção do grau de Bacharel em
Engenharia da Computação

Orientador: Prof. Dr. Leandro Krug Wives
Coorientador: Msc. Oscar Ortegon Romero

Porto Alegre
2022

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos André Bulhões Mendes

Vice-Reitora: Prof^a. Patricia Pranke

Pró-Reitora de Graduação: Prof. Cíntia Boll

Diretora do Instituto de Informática: Prof^a. Carla Maria Dal Sasso Freitas

Coordenador do Curso de Engenharia de Computação: Prof. Walter Fetter Lages

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Agradeço a minha mãe Cláudia e meu irmão Lorenzo por me apoiarem e conviverem comigo durante todo o período da minha graduação. Agradeço ao meu pai Edison e ao meu avô Cláudio que, de onde estiverem, sei que estão zelando por mim e felizes de saber que estou me formando. Agradeço à minha vóia Suzanira, que me aguentou por todo esse tempo e não me deixou desistir da faculdade. Agradeço à minha namorada Júlia, que esteve comigo durante todo o período da graduação e sempre me apoiou, nos momentos bons e nos ruins.

Também agradeço aos meus orientadores Leandro e Oscar, e ao meu colega Arthur, que ajudaram bastante no desenvolvimento do trabalho.

Sem vocês, a realização deste trabalho não seria possível, muito obrigado!

RESUMO

Com a constante evolução tecnológica, percebemos que a tecnologia se tornou cada vez mais essencial para o ensino. Entretanto, ainda há necessidade de melhorias e evoluções, principalmente, na avaliação de estudantes. A TRI permite uma maior precisão na avaliação de estudantes, todavia não se encontrou algum produto que possibilite a geração e realização de questionários dinâmicos baseados na TRI de maneira responsiva e que permita o uso de *smartphones* pelos estudantes. O objetivo deste trabalho foi criar um ambiente que supre essa carência. Então, foi criada uma aplicação *front-end* que consome uma *API REST* já existente, de modo a permitir que professores gerenciem questionários, e que alunos possam responder seus respectivos questionários. Para validar o ambiente desenvolvido, foi utilizado um questionário de *feedback* com questões sobre usabilidade. Suas respostas indicam que o ambiente desenvolvido apresentou uma boa usabilidade e cumpriu o objetivo proposto. Entretanto, ainda há oportunidade de melhorar o ambiente, principalmente para deixá-lo mais amigável para os professores.

Palavras-chave: Questionários dinâmicos. TRI. Aplicação. API REST.

Environment for execution of dynamic questionnaires based on TRI

ABSTRACT

With the constant technological evolution, we realize that technology has become increasingly essential for teaching. However, there is still a need for improvement and evolution, mainly in the evaluation of students. IRT allows for greater precision in the assessment of students, however, no product was found that allows the generation and execution of dynamic questionnaires based on TRI in a responsive way and that allows the use of smartphones by students. The objective of this work was to create an environment that fills this need. So, a front-end application was created that consumes an existing REST API, in order to allow teachers to manage questionnaires, and students to be able to answer their respective questionnaires. To validate the developed environment, a feedback questionnaire with questions about usability was used. Their answers indicate that the developed environment presented good usability and fulfilled the proposed objective. However, there is still opportunity to improve the environment, mainly to make it more teacher-friendly.

Keywords: Dynamic Questionnaires, TRI, Application, REST API.

LISTA DE FIGURAS

Figura 4.1 Quadro <i>Kanban</i> no <i>Trello</i>	21
Figura 4.2 Exemplo de questão de <i>feedback</i>	22
Figura 5.1 Arquitetura geral do ambiente	24
Figura 5.2 Diagrama de modelos e relacionamentos	25
Figura 5.3 Organização das pastas do ambiente <i>front-end</i>	29
Figura 5.4 Casos de uso do ambiente <i>front-end</i>	30
Figura 6.1 Página de login	32
Figura 6.2 Página de cadastro	33
Figura 6.3 Listagem de categorias	34
Figura 6.4 Criação de categorias	34
Figura 6.5 Edição de categorias	35
Figura 6.6 Modal de confirmação de exclusão.....	35
Figura 6.7 Criação de aluno	36
Figura 6.8 Listagem de alunos	36
Figura 6.9 Edição de alunos	37
Figura 6.10 Criação de questão	38
Figura 6.11 Listagem de questões	39
Figura 6.12 Edição de questão	39
Figura 6.13 Criação de questionários.....	40
Figura 6.14 Listagem de questionários	40
Figura 6.15 Edição de questionário.....	40
Figura 6.16 Página para iniciar questionário	41
Figura 6.17 Resolução do questionário.....	42
Figura 7.1 Respostas da questão 4 do Questionário SUS	44
Figura 7.2 Respostas da questão 7 do Questionário SUS	45
Figura 7.3 Respostas da questão 9 do Questionário SUS	45
Figura 7.4 Respostas da questão 10 do Questionário SUS	46
Figura 7.5 Respostas da questão 1 do Questionário SUS	47
Figura 7.6 Respostas da última questão aberta	47

LISTA DE ABREVIATURAS E SIGLAS

EAD	Ensino a Distância
VLE	<i>Virtual Learning Environments</i>
REST	<i>Representational State Transfer</i>
CAT	<i>Computerized Adaptive Testing</i>
TCT	Teoria Clássica dos Testes
TRI	Teoria de Resposta ao Item
SUS	<i>System Usability Scale</i>
SPA	<i>Single Page Application</i>
UFRGS	Universidade Federal do Rio Grande do Sul
LGPD	Lei Geral de Proteção de Dados Pessoais

SUMÁRIO

1 INTRODUÇÃO	9
2 FUNDAMENTAÇÃO TEÓRICA	11
2.1 Teoria Clássica dos Testes	11
2.2 Teoria de Resposta ao Item	11
2.2.1 Habilidade do aluno	12
2.2.2 Modelos unidimensionais	13
2.2.3 Modelos para itens dicotômicos.....	13
2.2.4 Modelo logístico de três parâmetros	13
2.3 Ensino Híbrido	14
2.4 Testes Adaptativos Computadorizados	15
2.5 Desvio e Erro Padrão	15
2.6 Serviços REST	16
3 TRABALHOS RELACIONADOS	17
3.1 Análise de questionários para ambientes virtuais de aprendizagem baseado na TRI	17
3.2 Ambiente para Geração de Questionários Dinâmicos Baseados em TRI	18
3.3 AvalWeb - Sistema Interativo para gerência de questões e aplicação de avaliações na Web	18
3.4 Moodle UCAT	19
3.4.1 Resumo do Capítulo.....	19
4 METODOLOGIA	21
4.1 Questionário de Validação	22
5 AMBIENTE DESENVOLVIDO	24
5.1 Serviço REST	24
5.2 Ambiente front-end	26
6 GUIA DE USO	31
6.1 Guia para professores	31
6.1.1 Criar uma conta.....	31
6.1.2 Gerenciar categorias.....	31
6.1.3 Gerenciar alunos	32
6.1.4 Gerenciar questões	34
6.1.5 Gerenciar questionários	35
6.2 Guia para alunos	37
6.2.1 Acessar sua conta	37
6.2.2 Iniciar questionário	37
6.2.3 Responder questionário.....	38
7 AVALIAÇÃO DO AMBIENTE	43
7.1 Avaliação do Questionário SUS	43
7.2 Avaliação das questões abertas	44
8 CONCLUSÕES	48
REFERÊNCIAS	49
APÊNDICE A — QUESTIONÁRIO DE VALIDAÇÃO	51
APÊNDICE B — GUIA DE INSTALAÇÃO	56
B.1 Instalação do Ambiente front-end	56
B.2 Instalação da Aplicação REST	56

1 INTRODUÇÃO

Com a constante evolução tecnológica e as recentes e bruscas alterações no nosso cotidiano causadas pela pandemia, percebemos que a tecnologia se tornou cada vez mais essencial para o ensino. Entretanto, ainda há necessidade de melhorias e evoluções, principalmente, na avaliação de estudantes.

A Teoria Clássica dos Testes, que utiliza o resultado bruto total de um questionário para sua avaliação (WU; TAM; JEN, 2016), ainda é muito presente no ambiente estudantil. Todavia, por mais que métodos avaliativos baseados nessa teoria possam ser adequados para análises mais simples, ela deixa a desejar quando análises mais complexas são necessárias. Para isso, a Teoria de Resposta ao Item, ou TRI, é mais apropriada (PASQUALI; PRIMI, 2003).

A Teoria de Resposta ao Item permite uma maior precisão na avaliação de estudantes, pois utiliza parâmetros como a dificuldade da questão, a aptidão do aluno em determinada competência, e a aleatoriedade da resposta para determinar seu score final. A aplicação desses parâmetros, muito mais profundos do que apenas a quantidade total de acertos do questionário, propicia uma melhor comparação da proficiência dos estudantes nos determinados assuntos e possibilita a comparação do desempenho entre diferentes turmas, o que explica adoção deste método no Exame Nacional do Ensino Médio¹. No entanto, a utilização da TRI exige um maior amparo tecnológico computacional, pois seu processo requer análises e cálculos matemáticos complexos (ANDRADE; TAVARES; VALLE, 2000).

As medidas de distanciamento social causadas pela pandemia fizeram com que o ensino híbrido, que combina atividades presenciais com atividades à distância, ganhasse muito mais relevância em comparação ao presencial, que é restrito a atividades que exigem o comparecimento do estudante ao local de estudo. Mais do que nunca a tecnologia se faz presente para possibilitar a disseminação do conhecimento. Todavia, mesmo com a grande popularização dos *smartphones* e seu uso muito mais frequente do que o de computadores de mesa (MEIRELLES, 2021), grande parte das avaliações remotas não possibilitam que sua realização seja feita de maneira simples e clara através do telefone celular, o que seria inclusivo e satisfatório para os estudantes que assim desejassem.

Portanto, o objetivo deste trabalho é criar um ambiente online responsivo, que possibilite o uso de *smartphones* pelos alunos, para permitir a realização de questionários

¹<http://portal.mec.gov.br/ultimas-noticias/389-ensino-medio-2092297298/16042-onu-diz-que-teoria-de-resposta-ao-item-garante-isonomia-das-provas>

dinâmicos baseados na TRI. Este ambiente irá consumir e fazer adaptações necessárias em um serviço *REST*, criado por (ADOLFO, 2021) em seu Trabalho de Conclusão de curso. Esse serviço *REST* disponibiliza uma *API* que permite a gerência de diversos aspectos necessários, como alunos e questões, para uma correta administração de questionários dinâmicos avaliados com TRI. Dessa forma, deseja-se tornar o processo avaliativo mais eficiente para o professor e amigável para o estudante.

Este trabalho se insere num contexto voltado à educação, onde existem outros trabalhos que envolvem a metodologia TRI e testes adaptativos computadorizados. Este trabalho está organizado de forma que o capítulo seguinte apresenta uma Fundamentação Teórica, onde conceitos importantes relacionados ao trabalho são explicados. Em seguida, apresenta-se um capítulo de Trabalhos Relacionados, onde trabalhos semelhantes já existentes são apontados e descritos. Após, há um capítulo de Metodologia, que detalha o processo desenvolvimento do software vislumbrado, bem como seus requisitos. Na sequência, apresenta-se um capítulo de Ambiente Desenvolvido, que mostra detalhes técnicos da arquitetura e do desenvolvimento do ambiente. Depois, temos o capítulo Guia de Uso, que contém informações para orientar os usuários da aplicação. Em seguida, observa-se o capítulo Avaliação do Ambiente, que mostra o *feedback* dos usuários da aplicação. Por último, temos o capítulo de Conclusão, onde apresenta-se um resumo dos achados importantes deste trabalho, bem como sugestões para trabalhos futuros ou melhorias no trabalho atual.

2 FUNDAMENTAÇÃO TEÓRICA

Diversos conceitos da área educacional e da computação foram utilizados neste trabalho, e esses são descritos neste capítulo.

2.1 Teoria Clássica dos Testes

A TCT, ou Teoria Clássica dos Testes, representa uma análise de resultados de testes que se baseia no seu resultado bruto total. Essa teoria fornece uma série de estatísticas, dentre elas a confiabilidade do teste, a estimativa da dificuldade de um item específico e o cálculo de erro. Essa teoria é mais antiga que a Teoria de Resposta ao Item (TRI) e também bem menos complexa, entretanto isso não implica que a Teoria Clássica dos Testes esteja defasada. A TCT e a TRI podem até ser utilizadas concorrentemente para propósitos diferentes, visto que elas de normalmente proveem resultados complementares (PASQUALI; PRIMI, 2003).

Assim sendo, a compreensão dessas duas teorias é muito importante para que se defina qual abordagem é a mais adequada, dependendo do cenário e das métricas que se pretende utilizar.

2.2 Teoria de Resposta ao Item

A TRI (Teoria de Resposta ao Item) representa um conjunto de modelos matemáticos que contribuem nas conclusões referentes às habilidades ou competências avaliadas em um questionário. Esses modelos tornam possível a estimativa das referentes habilidades através da correlação entre as questões apresentadas ao aluno e sua pontuação obtida, tornando o questionário adaptado para cada estudante.

A Teoria de Resposta ao Item possui modelos que retratam a relação entre a habilidade de um aluno e a sua probabilidade de acertar a resposta referente ao item. Consequentemente, é possível realizar a avaliação de cada estudante de maneira independente, de forma que se apresente, para cada aluno, itens compatíveis com sua habilidade (SANTOS; GUEDES, 2005).

O fato da Teoria Clássica dos Testes associar as interpretações e análises ao questionário como um todo, tornando a comparação entre estudantes que não realizaram o

mesmo questionário impossibilitada, representa sua maior diferença com a TRI (ANDRADE; TAVARES; VALLE, 2000). Por sua vez, a TRI associa as interpretações e análises, focando em cada item respondido, e não apenas no escore bruto do questionário, o que torna possível tanto a comparação entre diferentes turmas submetidas a questionários que tenham alguns itens em comum, quanto a avaliação de estudantes de uma mesma turma que tenham realizado testes totalmente diferentes.

A relação entre a probabilidade de um estudante acertar uma questão e suas habilidades é expressa nos modelos matemáticos da TRI de maneira que a habilidade do aluno e a probabilidade do acerto do item sejam diretamente proporcionais, ou seja, uma maior habilidade do aluno implica em uma maior chance de acerto do respectivo item. de acordo com Andrade, Tavares e Valle (2000), os modelos propostos fundamentam-se basicamente em três elementos: a natureza do item, o número de populações envolvidas e a quantidade de habilidades que estão sendo analisada.

Sobre a natureza dos itens, segundo Bragion (2010), pode-se classificá-los como dicotômicos, polinômicos ou contínuos. São ditos dicotômicos quando eles são avaliados de maneira dual, ou seja, correto ou incorreto. São polinômicos quando para cada categoria de resposta incumbe-se uma probabilidade, e são classificados como contínuos quando, em questões abertas, a resposta é avaliada através de valores numéricos pertencentes a um intervalo de números reais.

Sobre a quantidade de habilidades, de acordo com Andrade, Tavares e Valle (2000), um modelo pode ser unidimensional quando avalia apenas uma habilidade, ou multidimensional quando analisa duas ou mais habilidades.

No escopo deste trabalho, foram utilizados itens dicotomizados, com apenas uma população envolvida e modelos unidimensionais.

2.2.1 Habilidade do aluno

Também conhecida como traço latente, no contexto da TRI, a habilidade do aluno refere-se à medida psicológica que está sendo avaliada, e seu uso provém o cálculo da chance que um estudante com determinada habilidade tem de responder corretamente a um item de determinada dificuldade (CONDE, 2007).

Dessa forma, a probabilidade do aluno responder uma questão de forma correta é calculada de acordo com sua habilidade e a dificuldade da questão. Além disso, devido à possibilidade da habilidade do indivíduo de se relacionar diretamente com a dificuldade

de um item, pode-se representar ambas variáveis em uma escala equivalente de valores.

2.2.2 Modelos unidimensionais

Modelos unidimensionais são aqueles que avaliam apenas uma habilidade (ANDRADE; TAVARES; VALLE, 2000). Eles podem, por sua vez, abranger uma ou várias amostras de indivíduos pertencentes a uma mesma população, onde cada amostra representa um grupo distinto.

Em vista disso, segundo Andrade, Tavares e Valle (2000), a definição de grupo se relaciona intimamente com o procedimento de amostragem, que se responsabiliza pela determinação da quantidade de populações que estarão envolvidas.

2.2.3 Modelos para itens dicotômicos

Na análise de itens dicotômicos, de acordo com Andrade, Tavares e Valle (2000), os tipos de modelos mais importantes são os modelos logísticos, que podem ser de três tipos diferentes, considerando a natureza da quantidade de parâmetros analisados: a dificuldade do item, a discriminação do item e a chance de um aluno acertar a resposta de um item que possui uma dificuldade maior do que o valor de sua habilidade.

De forma geral, os modelos para itens dicotômicos podem ser uma boa ferramenta para a análise tanto de questões (i.e., itens) abertas, por exemplo questões discursivas avaliadas de maneira dicotomizada, quanto de questões de múltipla escolha dicotomizadas (ANDRADE; TAVARES; VALLE, 2000). Para o escopo deste trabalho, serão tratadas apenas questões de múltipla escolha ou de verdadeiro ou falso, ou seja, apenas itens fechados.

2.2.4 Modelo logístico de três parâmetros

O modelo logístico de *três parâmetros*¹. Esse é um dos modelos mais importantes propostos pela TRI e fundamenta-se na premissa de que a relação entre a habilidade do indivíduo e sua probabilidade de responder corretamente um determinado item não é linear, e que uma maior habilidade de um indivíduo implica em uma maior probabilidade

¹ Seus três parâmetros, apontados na equação, são a_i , b_i e c_i

de acerto. De acordo com Andrade, Tavares e Valle (2000), a equação matemática que define esse modelo é:

$$P(U_{ij} = 1|\theta_j) = c_i + (1 - c_i) \frac{1}{1 + \exp^{(-Da_i(\theta_j - b_i))}} \quad (2.1)$$

com $i = 1, 2, 3, \dots, I$, e $j = 1, 2, 3, \dots, n$.

Na qual:

- θ_j indica a habilidade do indivíduo j .
- U_{ij} é uma variável dicotômica que vale 1 quando o indivíduo j acertou o item i , no caso de uma resposta incorreta, seu valor passa a ser 0.
- $P(U_{ij} = 1|\theta_j)$ representa a probabilidade de um indivíduo j com habilidade θ_j responder de forma correta o item i .
- b_i indica a dificuldade do item i , mensurado na mesma escala da habilidade.
- a_i indica o parâmetro de discriminação do item.
- c_i também conhecida como acerto casual, indica a probabilidade de um indivíduo com habilidade baixa responder de maneira correta o item i .
- D é uma constante e indica o fator de escala, normalmente seu valor é 1.

Neste trabalho, o modelo logístico de três parâmetros é o modelo utilizado, visto que o serviço *REST* criado no trabalho de Adolfo (2021) o utiliza como base para a TRI, o qual é consumido pelo ambiente *front-end* aqui desenvolvido.

2.3 Ensino Híbrido

Uma das crescentes modalidades de ensino a distância, o *Blended learning*, ou Ensino Híbrido, tem se mostrado cada vez mais importante. Sua principal característica consiste em combinar atividades presenciais com atividades educacionais a distância, possibilitadas através das tecnologias digitais de informação e comunicação (VALENTE, 2014).

Essa modalidade de ensino pode ser categorizada em quatro modelos distintos: flex, *blended* misturado, virtual enriquecido e rodízio. No modelo flex, a base do processo educacional é o conteúdo presente no ambiente online, e a parte flexível representa o tipo de abordagem feita nos momentos presenciais. No modelo *blended*, misturado, o

estudante pode decidir por complementar as disciplinas presenciais por meio da realização de uma ou mais disciplinas totalmente remotas. O modelo virtual enriquecido possui foco nas atividades a distância, mas oferece a possibilidade da realização de algumas atividades presenciais. Já o modelo rodízio proporciona ao estudante a oportunidade de intercalar entre as diferentes modalidades de aprendizagem (STAKER; HORN, 2012).

Portanto, percebe-se que o Ensino Híbrido possui diversos métodos ou estratégias distintas para associar as atividades presenciais com atividades remotas, e o que todas elas têm em comum é a necessidade do uso da tecnologia, principalmente para possibilitar a correta e eficiente execução das atividades educacionais não presenciais.

2.4 Testes Adaptativos Computadorizados

Também conhecidos como *Computerized Adaptive Testing* (CAT), os Testes Adaptativos Computadorizados, de acordo com Ortegon (2019), representam questionários que possam ser acessados em ambientes computacionais, nos quais seus itens são dinâmicos, ou seja, se adaptam de maneira personalizada dependendo do aluno que os está respondendo. Eles possibilitam, após a realização completa do teste, a análise da habilidade do estudante.

Dessa forma, ainda segundo Ortegon (2019), um CAT pode ser gerado por meio da metodologia TRI para possibilitar a análise de um conjunto de questões, de forma a auxiliar na escolha de qual conjunto de questões é o mais adequado para a avaliação de uma turma, por exemplo.

Além disso, de acordo com Cardoso (2001), um CAT também pode ser utilizado para permitir a criação de um sistema que vise auxiliar os professores na criação das avaliações, selecionando o grau de dificuldade dos itens que farão parte da avaliação. Estratégias interativas também podem ser utilizadas, de modo a variar o nível de dificuldade das questões conforme com suas respostas anteriores.

2.5 Desvio e Erro Padrão

Segundo Lunet, Severo e Barros (2006), o desvio padrão aponta de maneira quantitativa o quão distanciado o valor analisado está da média. Ou seja, ele nos mostra a dispersão do valor quanto à média, de forma que sua magnitude não depende do tamanho

da amostra, e sim do quão dispersa as observações estão da média.

Já o erro padrão da média, de acordo com Lunet, Severo e Barros (2006), pode ser calculado pela magnitude do desvio padrão da amostra dividido pela raiz quadrada da quantidade de observações da amostra. Seu valor representa uma estimativa do desvio padrão da distribuição das médias das amostras que possuam o mesmo tamanho e pertencentes à mesma população.

Neste trabalho, o erro padrão é utilizado como critério de parada para um questionário em andamento. Um questionário dinâmico baseado na TRI, conforme implementado no serviço *REST* do trabalho de Adolfo (2021), encerra-se quando as questões acabam ou quando o erro padrão do aluno fica menor que 0,315, que foi o valor decidido e implementado no serviço *REST* do trabalho de Adolfo (2021).

2.6 Serviços *REST*

Representational State Transfer, ou *REST*, significa Transferência de Estado Representacional, e representa uma arquitetura de software. Essa arquitetura engloba diversos outros estilos arquiteturais, e possui como objetivo principal promover o correto e eficiente compartilhamento de informações através de suas propriedades importantíssimas: simplicidade, usabilidade, extensibilidade e escalabilidade. Os serviços *REST* buscam potencializar a escalabilidade e independência dos módulos da implementação, ao mesmo tempo que visam reduzir a latência da comunicação na rede, ao se utilizarem das propriedades citadas anteriormente (Li; Chou, 2011).

Em vista disso, um serviço *REST* se define por representar um método eficiente, rápido e seguro que permite trocas de informações entre sistemas (Li; Chou, 2011).

Para o escopo deste trabalho, a Teoria de Resposta ao Item se mostrou ser mais apropriada, pois permite realizar uma avaliação individualizada, comparando a competência do estudante com o nível de dificuldade do respectivo item.

3 TRABALHOS RELACIONADOS

Encontra-se, na literatura, diferentes trabalhos que abordam sistemas dinâmicos de gerência de questionários, assim como metodologias de análises de questões com TRI. Alguns dos mais relevantes para este trabalho estão elencados a seguir.

3.1 Análise de questionários para ambientes virtuais de aprendizagem baseado na TRI

No trabalho de Ortegon e Wives (2019), a partir da análise histórica de questionários foi proposto um modelo para geração e planejamento de CATs para *Virtual Learning Environments* (VLE), ou ambientes virtuais de aprendizagem, baseados na metodologia TRI. Esse modelo possibilitou a classificações das questões pertencentes aos questionários analisados, considerando quais itens são mais adequados ao ensino de determinado assunto.

Dois experimentos foram realizados com o objetivo de validar a metodologia utilizada por Ortegon e Wives (2019). O primeiro considerou questionários respondidos em três semestres distintos em um curso oferecido pelo Instituto de Informática na UFRGS. Nele foram avaliados questionários sobre dois tópicos diferentes, um sobre *Hashing* e outro sobre Complexidade de Algoritmos. O segundo experimento considerou questionários respondidos em quatro semestres distintos no curso de Engenharia Elétrica da UFRGS. Foram considerados doze questionários obtidos (Ortegon; Wives, 2019).

Para a análise foram realizados os seguintes passos (Ortegon; Wives, 2019):

- Tratamento e limpeza dos dados, retirando tentativas múltiplas e incompletas.
- Aplicação de três modelos TRI (Rasch, logístico de 2 parâmetros e logístico de três parâmetros).
- Seleção e classificação das questões.

Após a efetuação das análises, foi concluído que, se as suposições de unidimensionalidade e independência local estiverem presentes, todas as questões analisadas são necessárias. Entretanto, as suposições podem ser mais ou menos relevantes que outras dependendo do assunto a ser ensinado (Ortegon; Wives, 2019).

3.2 Ambiente para Geração de Questionários Dinâmicos Baseados em TRI

No trabalho de Adolfo (2021) foi fornecida uma ferramenta que permite a geração de questionários adaptativos baseados na TRI. Essa ferramenta possui integração com a plataforma *Moodle*, através de um serviço *REST*. O objetivo da criação dessa ferramenta foi, principalmente, diminuir a complexidade envolvida no processo de avaliações com a metodologia TRI, que exige modelos matemáticos complexos para se efetuar a correta extração do resultado. Dessa forma, a ferramenta visou também permitir aos professores o aprimoramento dos seus questionários, através da diminuição do erro médio e adaptando as questões ao conhecimento e habilidade de seus alunos (ADOLFO, 2021).

Esse serviço *REST* é de extrema importância para a elaboração deste trabalho, visto que, após a realização de alguns ajustes necessários, ele possibilita que o ambiente *front-end* aqui desenvolvido requisite as informações e operações necessárias para a correta administração e realização dos questionários, tanto por parte dos professores quanto por parte dos alunos.

3.3 AvalWeb - Sistema Interativo para gerência de questões e aplicação de avaliações na Web

No trabalho de Cardoso (2001) foi realizado um estudo dos processos de avaliação de alunos em um ambiente online. Baseado nesse estudo, foi apresentado um sistema que permite a gerência de questões e aplicação de avaliações. Esse sistema pretende auxiliar os professores na criação das avaliações, além de selecionar o grau de dificuldade dos itens que irão fazer parte da avaliação. A utilização de estratégias interativas voltadas para aplicação das avaliações também foi presente nesse trabalho, como por exemplo, o uso da variação dos níveis de dificuldade das questões, submetidas aos alunos, de acordo com suas respostas anteriores (CARDOSO, 2001).

Foi proposto também, integrando esse sistema de gerência de questões, um módulo de autoavaliações. Esse módulo fornece uma resposta imediata para o aluno, após ele responder um item, sobre qual é a resposta correta para a questão e qual a explicação que a justifica. Dessa forma, esse sistema pôde ensinar ao aluno e facilitar seu entendimento sobre determinado assunto ao invés de simplesmente determinar a correção da questão e sua nota (CARDOSO, 2001).

Por fim, um protótipo foi desenvolvido e demonstrou a viabilidade das ideias pre-

sentes em seu proposto modelo de avaliação (CARDOSO, 2001).

3.4 Moodle UCAT

O Moodle UCAT é um *plugin* desenvolvido para a plataforma Moodle, criado por (KIMURA; NAGAOKA, 2012). Esse *plugin* contribui para a criação de CATs baseado no modelo Rasch, o que possibilita que testes adaptativos na plataforma Moodle sejam criados. Dessa forma, é possível adaptar a dificuldade das questões para a habilidade dos alunos e balancear automaticamente o modelo à medida que os questionários são respondidos.

Nesse trabalho são destacado alguns atributos importantes para a correta implementação de CATs. A necessidade de uma grande quantidade de itens, principalmente nas fases de pré-teste e análise dos itens, é um desses atributos, ou seja, um expressivo banco de questões implica em uma melhor capacidade de processamento de questões para o modelo proposto pelo UCAT. Essa quantidade de itens será relevante, principalmente, nas fases de pré-teste e análise dos itens (KIMURA; NAGAOKA, 2012).

Segundo o relato dos autores, o Moodle UCAT consegue resolver os maiores problemas para geração de CAT, que são o pré-teste e a criação de novos itens no banco de questões, bem como a recalibração do banco após essa adição. Esses problemas são resolvidos por meio da metodologia TRI para predizer e recalibrar a dificuldade dos itens (KIMURA; NAGAOKA, 2012).

O *plugin* UCAT possibilita a implementação de questionários adaptativos em cursos no Moodle, por parte do respectivo professor. Entretanto, por depender de toda a infraestrutura do Moodle para processar as questões, não é possível reaproveitar o UCAT em outros VLEs. De qualquer forma, esse trabalho é muito importante para a geração de CATs baseado em TRI (KIMURA; NAGAOKA, 2012).

3.4.1 Resumo do Capítulo

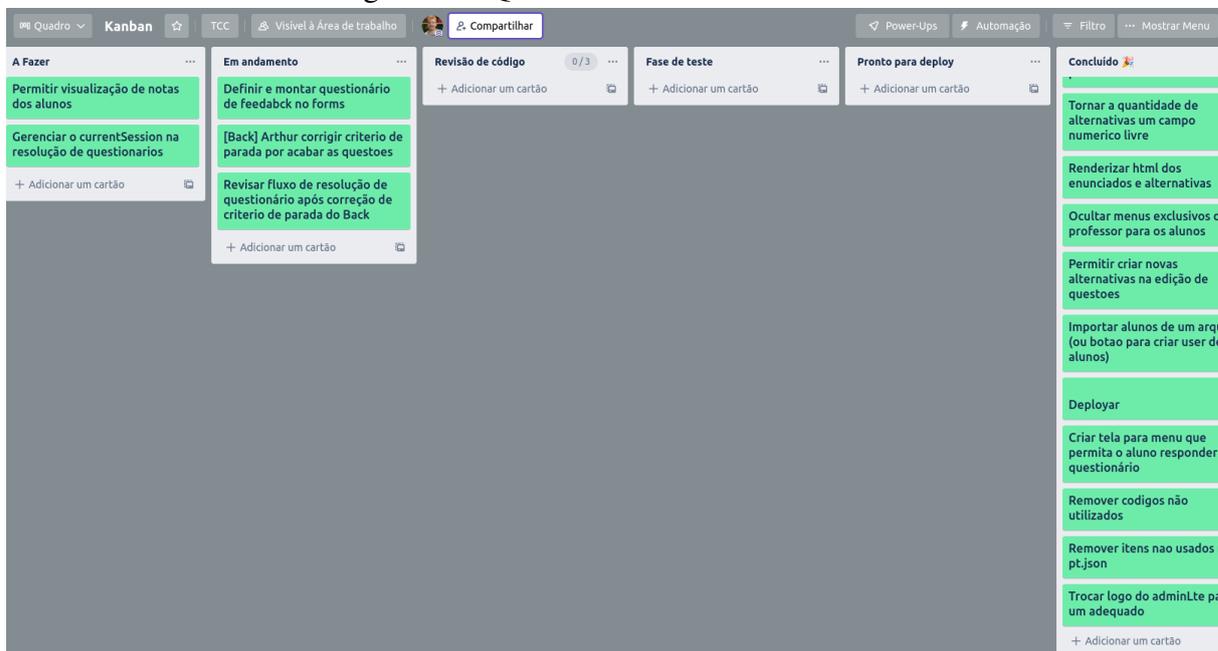
Os trabalhos aqui relatados contém diversos aspectos relacionados a este trabalho. Por exemplo, o modelo para geração e planejamento de CATs para VLEs que é abordado no trabalho de Ortegon e Wives (2019), o sistema que permite a gerência de questões e aplicação de avaliações do trabalho de Cardoso (2001) e o serviço *REST* que permite

a geração de questionários adaptativos baseados na TRI, do trabalho de Adolfo (2021), que é consumido diretamente pelo ambiente *front-end* aqui proposto. Entretanto, nenhum desses trabalhos permite que um estudante responda a um questionário dinâmico baseado na TRI através de seu *smartphone*, caso assim o deseje.

4 METODOLOGIA

O processo de desenvolvimento deste trabalho se deu através do uso da metodologia ágil *ScrumBut*¹. Nesse método as tarefas pendentes são descritas, organizadas e elencadas pelo sistema de controle e gestão de fluxo de produção *Kanban*, sem se utilizar de *sprints*, clássicas da metodologia *Scrum*². Para colocar essa metodologia em prática, foi utilizada a ferramenta *Trello*, onde foi possível facilmente montar um quadro *Kanban* com as colunas: a fazer, em andamento, revisão de código, fase de teste e concluído, conforme ilustra a Figura 4.1.

Figura 4.1: Quadro *Kanban* no *Trello*



Fonte: O autor

Para o ambiente *front-end*, tivemos como requisitos funcionais:

1. Login de usuários.
2. Cadastro de usuários.
3. Alteração de senha de usuários.
4. Cadastro, leitura, alteração e exclusão de categorias.
5. Cadastro, leitura, alteração e exclusão de alunos.
6. Cadastro, leitura, alteração e exclusão de questões.
7. Cadastro, leitura, alteração e exclusão de questionários.

¹<<https://www.scrum.org/resources/what-scrumbut>>

²<<https://www.scrum.org/resources/what-is-scrum>>

8. Realização de questionários, por parte dos alunos.

E como requisitos não funcionais:

1. Interface amigável para os usuários.
2. Interface responsiva, para que alunos possam usar o celular para responder aos questionários.

Para a validação deste trabalho, que visa principalmente garantir uma melhor experiência durante a realização de questionários dinâmicos baseados na TRI com o uso de *smartphones*, foi feita uma pesquisa com voluntários, utilizando um questionário de validação. A pesquisa, o questionário e seu resultado são detalhados na seção 4.1 e no capítulo 7.

Após a coleta do *feedback* dos voluntários através do questionário de validação, os resultados obtidos serão então analisados e avaliados para o embasamento da conclusão deste trabalho.

4.1 Questionário de Validação

Como forma para validar o objetivo do trabalho, foi utilizado o *System Usability Scale*³, ou SUS, com os alunos e professores voluntários que utilizaram a aplicação. Esse questionário, de maneira rápida, simples e confiável, cobre uma impressão abrangente da experiência do usuário sobre a usabilidade do sistema, de acordo com Brooke (1996). Ele é composto por dez perguntas, na qual as respostas variam de 1 a 5, sendo 1 discordância total e 5 concordância total, conforme ilustra a figura 4.2.

Figura 4.2: Exemplo de questão de *feedback*

Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente. *

Marcar apenas uma oval.

1	2	3	4	5	
Discordo completamente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo completamente

Fonte: O autor

De acordo com Brooke (1996), as respostas são analisadas a partir de sua conver-

³<<https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>>

são para valores numéricos entre 0 e 4, e então esses valores são somados e multiplicados por 2.5, o que resulta em um valor final entre 0 e 100. A média desse valor final, segundo Brooke (1996), é 68, então resultados superiores a 68 são definidos como acima da média, e resultados abaixo de 68 são definidos como abaixo da média.

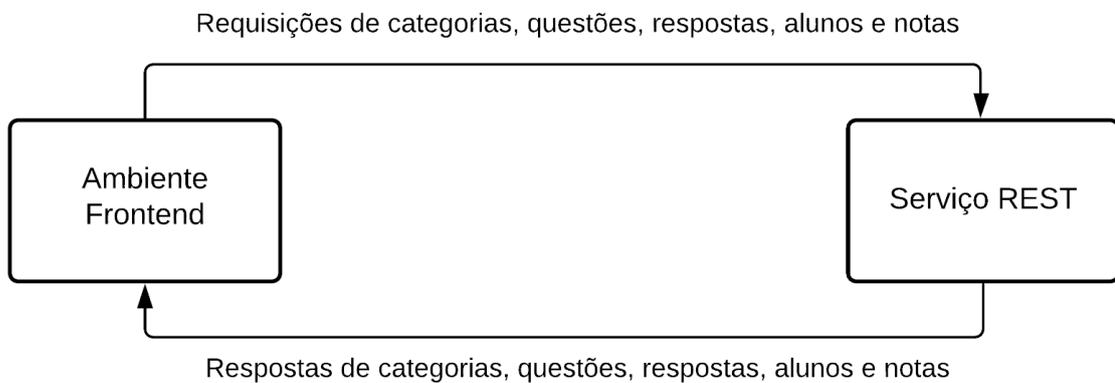
Além das dez perguntas do questionário SUS, foram adicionadas três perguntas abertas, para que o voluntário explique um pouco sobre sua experiência com a aplicação e aponte possíveis sugestões de melhoria. Além disso, foi adicionado uma pergunta objetiva, apenas para que o voluntário indique se utilizou a aplicação como aluno ou como professor.

Segundo Nielsen e Landauer (1993), cinco respostas são o bastante para encontrar 85% dos problemas de usabilidade, então aplicamos o questionário com pelo menos cinco voluntários. No total, o questionário ficou composto por 14 perguntas e pode ser conferido integralmente no apêndice A.

5 AMBIENTE DESENVOLVIDO

O ambiente desenvolvido neste trabalho se fundamentou em dois ambientes principais: o serviço *REST*, desenvolvido no trabalho "Ambiente para Geração de Questionários Dinâmicos Baseados em TRI", detalhado na seção 3.2, com algumas adaptações, e um ambiente *front-end* responsivo que consome o serviço *REST*. A arquitetura geral do ambiente é representada pela Figura 5.1, e constitui uma arquitetura cliente-servidor, onde o serviço *REST* atua como servidor, e o ambiente *front-end* como cliente.

Figura 5.1: Arquitetura geral do ambiente



Fonte: O autor

5.1 Serviço *REST*

O serviço *REST* abriga modelos responsáveis por persistir diversos dados necessários para o funcionamento correto da aplicação. Dentre eles, temos as tabelas:

- User, que abriga dados dos usuários da aplicação, sejam professores ou alunos.
- Student, que abriga dados específicos dos alunos.
- StudentGrade, que salva detalhes sobre as notas dos alunos.
- StudentAbilityLog, que guarda informações sobre as habilidades dos estudantes.
- Question, que abriga informações sobre as questões.
- QuestionAbilityLog, que guarda informações sobre a dificuldade das questões.
- Answer, que representa alternativas das questões.

ou incorreta. Para permitir que alunos também assumissem o papel de usuários do serviço, foi necessário adicionar, na tabela *User*, a coluna `student_id`, para que se possa consultar os dados do questionário do aluno quando ele acessa a aplicação. Além disso, foram feitas alterações pontuais e pequenas correções de *bugs* para permitir o correto consumo dos dados e serviços por parte do ambiente *front-end*.

5.2 Ambiente *front-end*

No ambiente *front-end*, foram utilizadas as linguagens *HTML*, *CSS* e *Typescript* com o *framework* *Vue.js*², juntamente com o *template* *AdminLTE 3*³ para a criação do ambiente *front-end* responsivo que consome a *API* e permite a realização de questionários dinâmicos baseados na Teoria da Resposta ao Item. A escolha das tecnologias citadas anteriormente se justifica pela facilidade que elas proporcionam de desenvolver e gerenciar aplicações *front-end* responsivas, e a escolha do *template* *AdminLTE 3* facilitou muito a responsividade. Saliento que o objetivo deste ambiente é permitir o uso de propósito geral através da *web*, tanto por alunos quanto por professores, mas com o foco em responsividade para que os alunos possam utilizar o celular se desejarem. Entretanto o uso da aplicação em computadores de mesa também é válido.

Para a arquitetura do ambiente *front-end*, foram definidos alguns padrões, com o objetivo de tornar o processo de trabalho mais ágil e organizado, facilitado o desenvolvimento e futuras manutenções ou evoluções. Como padrão para organização de pastas, foram feitas algumas divisões para tornar o projeto mais organizado, conforme ilustra a Figura 5.3 e o repositório⁴ do projeto. Na pasta raiz temos diversos arquivos de configuração, como o `package.json`, que é um arquivo utilizado para gerenciar as dependências do projeto. A pasta `src`, que abriga os diversos arquivos fonte para o projeto, foi separada em subdivisões para facilitar a organização, onde temos as pastas:

- `app`, que abriga a definição e exportação da principal classe da aplicação.
- `assets`, que possui arquivos anexos, como a imagem do logo da aplicação.
- `components`, que abriga a definição de componentes básicos, como botões, inputs e checkboxes.
- `interfaces`, que abriga as interfaces, como a de autenticação.

²<<https://vuejs.org/>>

³<<https://adminlte.io/themes/v3/>>

⁴<<https://github.com/castilhos90124/questionnaire-generator-frontend>>

- `modules`, que abriga os componentes de autenticação e os componentes da estrutura básica da SPA⁵, como o cabeçalho, rodapé e menu lateral.
- `pages`, que engloba os componentes referentes às páginas da aplicação⁶.
- `router`, que possui o mapeamento para as rotas do sistema.
- `services`, que abriga diversos métodos para comunicação com o serviço *REST*.
- `store`, que possui métodos relacionados a utilização do *local storage*⁷.
- `translation`, que possui os arquivos de tradução de chaves, atualmente apenas em português.
- `utils`, que abriga métodos genéricos utilizados em mais de um componente.

Para atender aos requisitos descritos anteriormente, foi necessário criar ou reutilizar diversos componentes. Em termos de código, um componente nesta aplicação se constitui geralmente de quatro arquivos distintos:

1. um arquivo `.html`, que possui os elementos html pertencentes ao componente, bem como possíveis instâncias de outros componentes.
2. um arquivo `.scss`, que abriga os estilos específicos do componente.
3. um arquivo `.ts`, que possui a definição da classe, bem como os métodos e atributos correspondentes para garantir o correto funcionamento do componente.
4. um arquivo `.vue`, que unifica os arquivos anteriores e garante que possam ser invocados corretamente pelo framework.

Dentre os principais componentes utilizados neste ambiente, temos:

- `Button`, componente de botão padrão da aplicação, instanciado em diversos outros componentes.
- `Checkbox`, componente de *checkbox* padrão da aplicação, instanciado no componente `Register`.
- `Dropdown`, componente de *dropdown*, utilizado nos menus da aplicação.
- `Input`, componente de *input* padrão da aplicação, instanciado em diversos outros componentes.
- `MenuItem`, componente que representa um item do menu lateral, instanciado no componente `MenuSidebar`.

⁵Acrônimo para *Single Page Application*

⁶Páginas de gerência de categorias, alunos, questões, questionários e alteração de senha

⁷Armazenamento local de dados na máquina do cliente

- `Footer`, componente que representa o rodapé da página.
- `Header`, componente que representa o cabeçalho da página.
- `MenuSidebar`, componente que representa o menu lateral da aplicação.
- `Main`, componente que representa a página principal da aplicação, instancia os menus, o rodapé e o cabeçalho.
- `ManageQuestion`, representa a página de gerência de questões.
- `ManageCategory`, representa a página de gerência de categorias.
- `ManageStudent`, representa a página de gerência de alunos.
- `ManageQuestionnaire`, representa a página de gerência de questionários.
- `Questionnaire`, representa a página de resolução de questionários.
- `ChangePassword`, representa a página de alteração de senha.
- `Login`, representa a página de *login*.
- `Register`, representa a página de cadastro de professores.

Para a hospedagem do ambiente *front-end*, foi utilizada a plataforma Heroku⁸. Ela representa uma plataforma *cloud*, com suporte a diversas tecnologias, que permite que aplicações, como a desenvolvida neste trabalho, executem em *containers* virtuais em um ambiente confiável. Ela foi escolhida pois possibilita a hospedagem da aplicação de maneira rápida e gratuita.

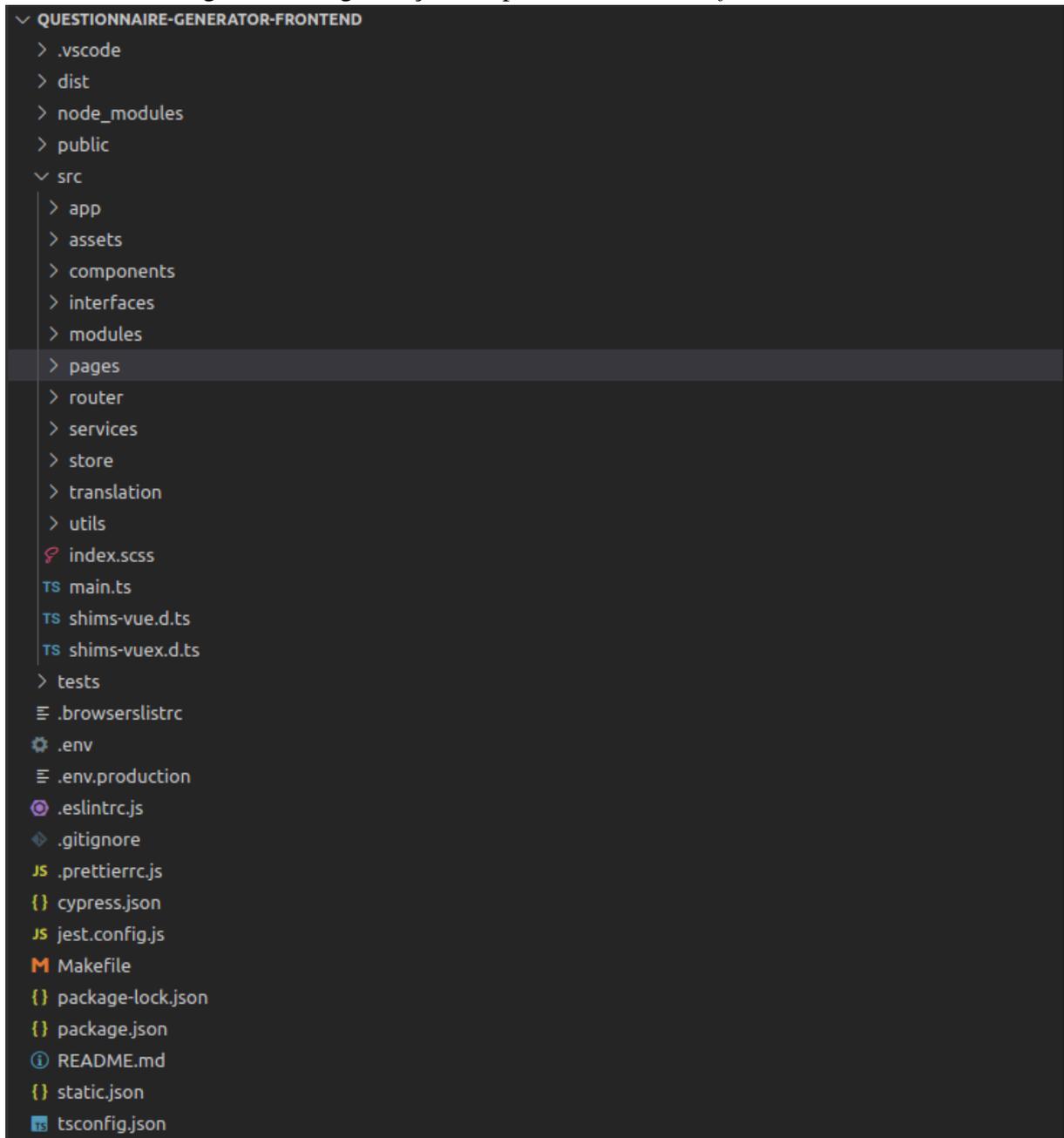
Por meio desse ambiente, o professor pode cadastrar, visualizar, editar e deletar categorias, questões e informações sobre alunos, além de permitir a geração de questionários adaptativos. Já os alunos tem acesso apenas ao seu questionário atual, caso exista, e podem responder este questionário criado pelo professor. Tanto professor quanto alunos podem alterar sua senha cadastrada. A Figura 5.4 ilustra esses casos de uso descritos.

Os dois ambientes principais utilizam a tecnologia *Git*⁹ para controle de gerenciamento de código, e o produto *GitHub*¹⁰ para a hospedagem do desenvolvimento do software com suporte a essa tecnologia. A escolha desse produto se justifica pois está entre os melhores do mercado e possui uma comunidade altamente ativa.

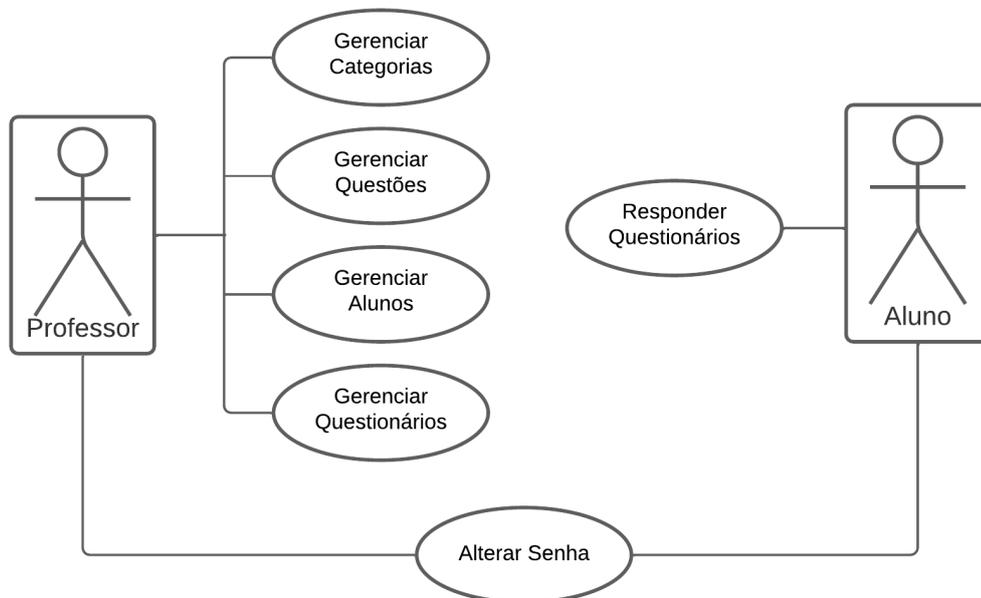
⁸[<https://www.heroku.com/>](https://www.heroku.com/)

⁹[<https://git-scm.com/>](https://git-scm.com/)

¹⁰[<https://github.com/>](https://github.com/)

Figura 5.3: Organização das pastas do ambiente *front-end*

Fonte: O autor

Figura 5.4: Casos de uso do ambiente *front-end*

Fonte: O autor

6 GUIA DE USO

Neste capítulo está documentado um tutorial para o uso da aplicação tanto por parte dos professores, quanto por parte dos alunos.

6.1 Guia para professores

Diferentemente dos alunos, os professores possuem acesso a funcionalidades diferenciadas, que os permitem gerenciar os diversos aspectos necessários para uma correta e eficiente administração de questionários.

6.1.1 Criar uma conta

Para criar uma conta, o professor deve acessar a aplicação¹ e clicar no link descrito como 'Sou professor', ilustrado na figura 6.1 aonde será redirecionado para a tela de cadastro. Então o professor deve preencher as informações necessárias na tela de cadastro e clicar no botão 'Cadastrar', conforme mostra a figura 6.2.

Feito o cadastro, o professor será redirecionado para a tela de login, onde pode inserir suas credenciais para logar na aplicação.

6.1.2 Gerenciar categorias

Para gerenciar as categorias, primeiro o professor deve clicar na primeira opção do menu lateral 'Categorias', para acessar a página de gerência de categorias.

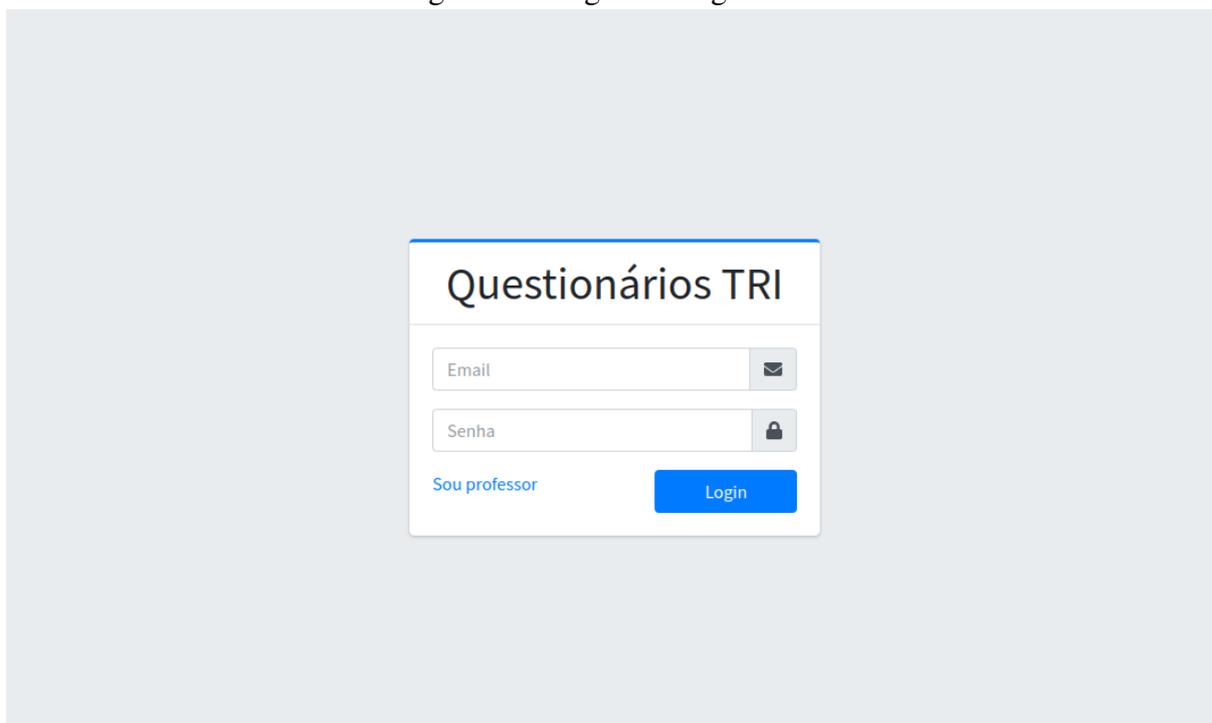
Então, para criar uma categoria, o professor deve clicar no botão verde com símbolo de adição presente no canto superior direito da tabela, conforme mostra a figura 6.3. Após isso, um modal será aberto na tela e permitirá que o professor preencha os dados, como nos ilustra a figura 6.4.

Após preencher os campos desejados, o professor deve clicar no botão 'Criar' para finalizar a criação de uma categoria. Esse processo pode ser repetido inúmeras vezes, até que o professor possua a quantidade desejada de categorias.

Feito isso, é possível visualizar na tabela as categorias criadas, onde também ve-

¹<http://questionariostri.herokuapp.com/>

Figura 6.1: Página de login



Fonte: <<http://questionariostri.herokuapp.com/login>>

mos que, para cada categoria presente na tabela, existe um botão de edição e um de exclusão no final da linha, como podemos observar na figura 6.3.

Para editar uma categoria existente, basta clicar no botão de 'lápiz', presente na figura 6.3, referente a categoria que se deseja editar. Então será aberto um modal, com os dados da categoria preenchidos, ilustrado na figura 6.5 onde o professor pode editar os valores que julgar necessário e salvar essas alterações clicando no botão 'Editar'.

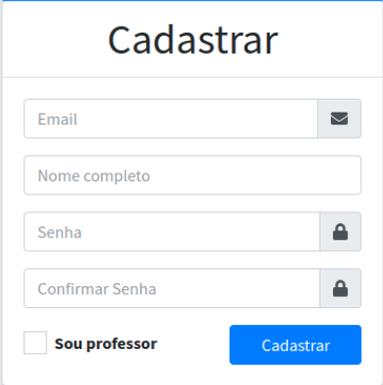
Para excluir uma categoria, o professor deve clicar no botão de 'lixeira', presente na figura 6.3, referente a categoria que se deseja deletar. Então será aberto um modal de confirmação, como mostra a figura 6.6, onde o professor pode confirmar a exclusão do item clicando em 'Excluir', ou pode cancelar a deleção clicando em 'Cancelar' ou fechando o modal no 'X' em seu canto superior direito.

6.1.3 Gerenciar alunos

De análoga às categorias, para gerenciar alunos, o professor deve primeiramente clicar na segunda opção do menu lateral 'Alunos', para acessar a página de gerência de alunos.

Então, para criar um aluno, o professor deve clicar no botão verde com símbolo

Figura 6.2: Página de cadastro



A imagem mostra uma interface de usuário para o cadastro de um usuário. O formulário é intitulado "Cadastrar" e contém os seguintes campos e elementos:

- Um campo de texto para "Email" com um ícone de envelope à direita.
- Um campo de texto para "Nome completo".
- Um campo de texto para "Senha" com um ícone de cadeado à direita.
- Um campo de texto para "Confirmar Senha" com um ícone de cadeado à direita.
- Um checkbox rotulado "Sou professor".
- Um botão azul rotulado "Cadastrar".

Fonte: <<http://questionariostri.herokuapp.com/register>>

de adição presente no canto superior direito da tabela. Após isso, um modal será aberto na tela e permitirá que o professor preencha os dados, como nos ilustra a figura 6.7.

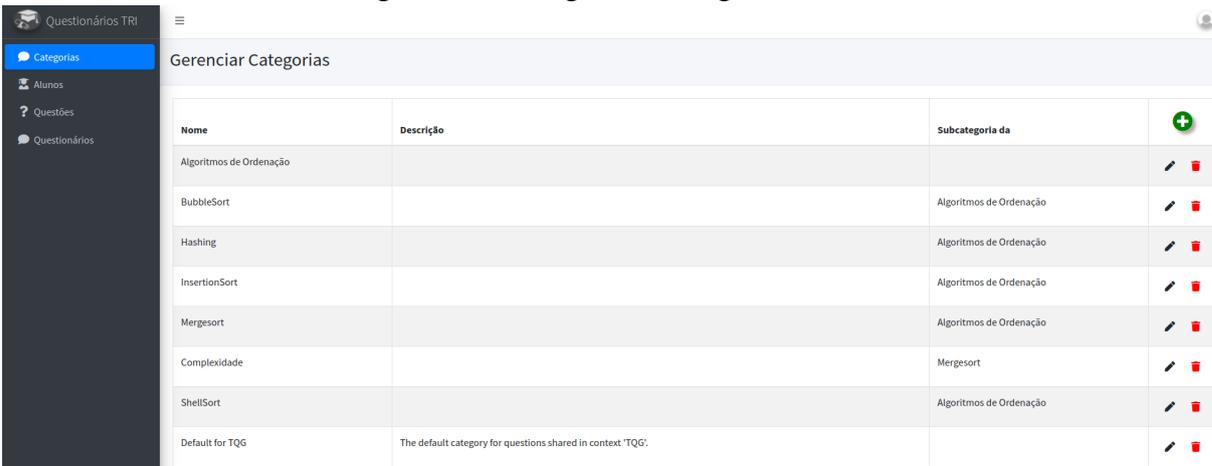
Após preencher os campos desejados, o professor deve clicar no botão 'Criar' para finalizar a inclusão de um aluno. Esse processo pode ser repetido inúmeras vezes, até que o professor inclua todos os alunos necessários.

Feito isso, é possível visualizar na tabela os alunos criados, onde também vemos que, assim como nas categorias, existe um botão de edição e um de exclusão para cada linha da tabela, como podemos observar na figura 6.8.

Para editar um aluno existente, basta clicar no botão de 'lápiz' referente ao aluno que se deseja editar. Então será aberto um modal, com os dados do aluno preenchidos, ilustrado na figura 6.9 onde o professor pode editar os valores que julgar necessário e salvar essas alterações clicando no botão 'Editar'.

Para excluir um aluno, o professor deve clicar no botão de 'lixeira' referente ao aluno que se deseja deletar. Então será aberto um modal de confirmação, semelhante ao da figura 6.6, onde o professor pode confirmar a exclusão do item clicando em 'Excluir', ou pode cancelar a deleção clicando em 'Cancelar' ou fechando o modal no 'X' em seu canto superior direito.

Figura 6.3: Listagem de categorias



The screenshot shows a web interface for managing categories. On the left is a dark sidebar with navigation options: 'Questionários TRI', 'Categorias' (highlighted), 'Alunos', 'Questões', and 'Questionários'. The main area is titled 'Gerenciar Categorias' and contains a table with the following data:

Nome	Descrição	Subcategoria da	
Algoritmos de Ordenação			 
BubbleSort		Algoritmos de Ordenação	 
Hashing		Algoritmos de Ordenação	 
InsertionSort		Algoritmos de Ordenação	 
Mergesort		Algoritmos de Ordenação	 
Complexidade		Mergesort	 
ShellSort		Algoritmos de Ordenação	 
Default for TQG	The default category for questions shared in context 'TQG'.		 

Fonte: <<http://questionariostri.herokuapp.com/manage-category>>

Figura 6.4: Criação de categorias



The screenshot shows a modal window titled 'Criar Categoria' with a close button in the top right corner. The form contains three input fields:

- Nome da Categoria***: A text input field.
- Descrição**: A text input field with a small icon in the bottom right corner.
- Subcategoria da**: A dropdown menu.

At the bottom right of the modal are two buttons: 'Cancelar' (grey) and 'Criar' (blue). Below the modal, a small text label reads: 'The default category for questions shared in context 'TQG'.'

Fonte: <<http://questionariostri.herokuapp.com/manage-category>>

6.1.4 Gerenciar questões

Para gerenciar as questões, inicialmente o professor deve clicar na terceira opção do menu lateral 'Questões', para acessar a página correspondente.

Então, para criar uma nova questão, o professor deve clicar no botão verde com símbolo de adição presente no canto superior direito da tabela. Após isso, um modal será aberto na tela e permitirá que o professor preencha os dados, como nos ilustra a figura 6.10.

Após preencher os campos desejados, o professor deve clicar no botão 'Criar' para finalizar a inclusão da questão. Esse processo pode ser repetido inúmeras vezes, até que o professor inclua todas as questões necessárias.

Feito isso, é possível visualizar na tabela a listagem de questões, como podemos

Figura 6.5: Edição de categorias

The default category for questions shared in context 'TQG'.

Fonte: <<http://questionariostri.herokuapp.com/manage-category>>

Figura 6.6: Modal de confirmação de exclusão

Fonte: <<http://questionariostri.herokuapp.com/manage-category>>

observar na figura 6.11.

Para editar uma questão existente, basta clicar no botão de 'lápiz' referente à questão que se deseja editar. Então será aberto um modal, com os dados da questão preenchidas, ilustrado na figura 6.12 onde o professor pode editar os valores que julgar necessário e salvar essas alterações clicando no botão 'Editar'.

Para excluir uma questão, o professor deve clicar no botão de 'lixeira' referente à questão que se deseja deletar. Então será aberto um modal de confirmação, semelhante ao da figura 6.6, onde o professor decide se irá concluir sua ação.

6.1.5 Gerenciar questionários

Para gerenciar questionários, inicialmente o professor deve clicar na quarta opção do menu lateral 'Questionários', para acessar a página correspondente.

Então, para iniciar um ou mais questionários, o professor deve clicar no botão verde com símbolo de adição presente no canto superior direito da tabela. Após isso, um

Figura 6.7: Criação de aluno

Modal de criação de aluno com o seguinte layout:

- Título: Criar Aluno
- Campos de entrada:
 - Nome*
 - Sobrenome*
 - Email* (com ícone de envelope)
- Botões: Cancelar (cinza) e Criar (azul)

Fonte: <<http://questionariostri.herokuapp.com/manage-student>>

Figura 6.8: Listagem de alunos

Tela de listagem de alunos com o seguinte layout:

- Menu lateral: Questionários TRI, Categorias, Alunos (selecionado), Questões, Questionários
- Título: Gerenciar Alunos
- Tabela de alunos:

Nome	Email	Habilidade	
Anderson Sousa	asdasdas@asdas.com	Média	 
Anderson Sousa Dufech	sdasasda@asdas.com	Alta	 
nome1_nome2	nome1nome2@ashdas.com	Alta	 
Aluno teste	alunoteste@email.com	Média	 
aluno wifes	wivesaluno@email.com	Média	 
aluno nao registrado	naoregistrado@email.com	Média	 
aluninho editerson da silva edited	aluninho@email.com	Alta	 

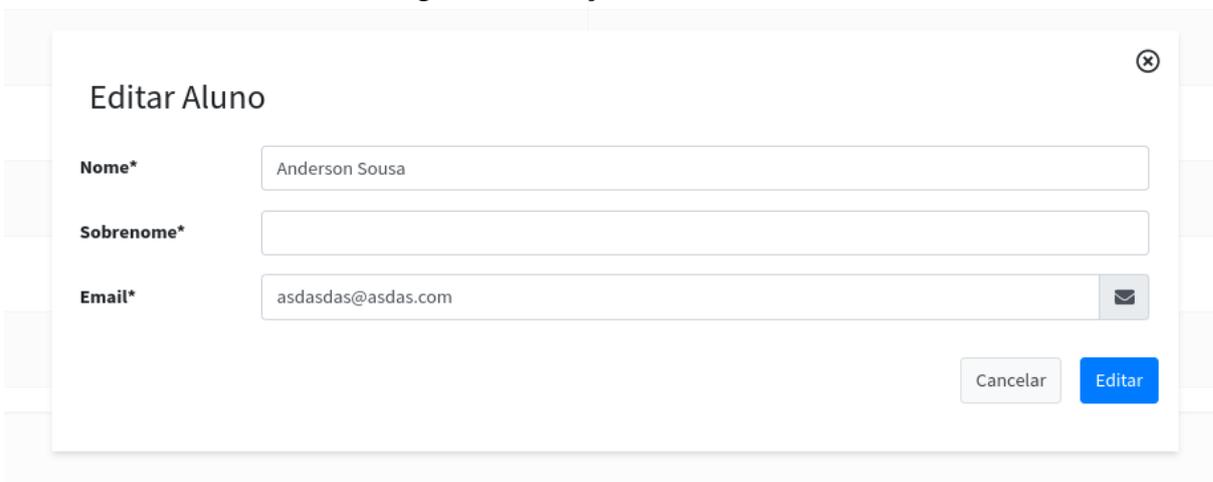
Fonte: <<http://questionariostri.herokuapp.com/manage-student>>

modal será aberto na tela e permitirá que o professor preencha os dados, como nos ilustra a figura 6.13.

Após preencher as informações necessárias, o professor deve clicar no botão 'Criar' para finalizar a criação dos questionários. Feito isso, é possível visualizar na tabela a listagem de questionários, como podemos observar na figura 6.14. Com a criação de um questionário, automaticamente é criada uma conta para cada estudante selecionado, para que eles possam acessar e responder o questionário. Por padrão, a credencial de login de cada aluno é seu email cadastrado pelo professor, e a senha é os caracteres que antecedem o '@' do seu email. Tanto alunos quanto professores podem alterar suas senhas se assim desejarem.

Para editar a categoria de um questionário ainda não iniciado, basta clicar no botão de 'lápiz' referente à questão que se deseja editar. Então será aberto um modal, com os dados do questionário preenchidos, ilustrado na figura 6.15 onde o professor pode editar

Figura 6.9: Edição de alunos

A imagem mostra um modal de edição de aluno com o título "Editar Aluno". O modal contém três campos de texto obrigatórios: "Nome*" com o valor "Anderson Sousa", "Sobrenome*" que está vazio, e "Email*" com o valor "asdasdas@asdas.com". No canto inferior direito do modal, há dois botões: "Cancelar" (cinza) e "Editar" (azul).

Editar Aluno

Nome* Anderson Sousa

Sobrenome*

Email* asdasdas@asdas.com

Cancelar Editar

Fonte: <<http://questionariosri.herokuapp.com/manage-student>>

a categoria do questionário e salvar a alteração clicando no botão 'Editar'. Questionários já iniciados não podem ser editados.

Para excluir um questionário, o professor deve clicar no botão de 'lixeira' referente ao questionário que se deseja deletar. Então será aberto um modal de confirmação, semelhante ao da figura 6.6, onde o professor decide se irá concluir sua ação.

6.2 Guia para alunos

6.2.1 Acessar sua conta

Para responder a um questionário já criado pelo professor, o aluno deve acessar a aplicação ², preenchendo seu email no campo correspondente e sua senha padrão, que corresponde a tudo que vem antes do '@' em seu email.

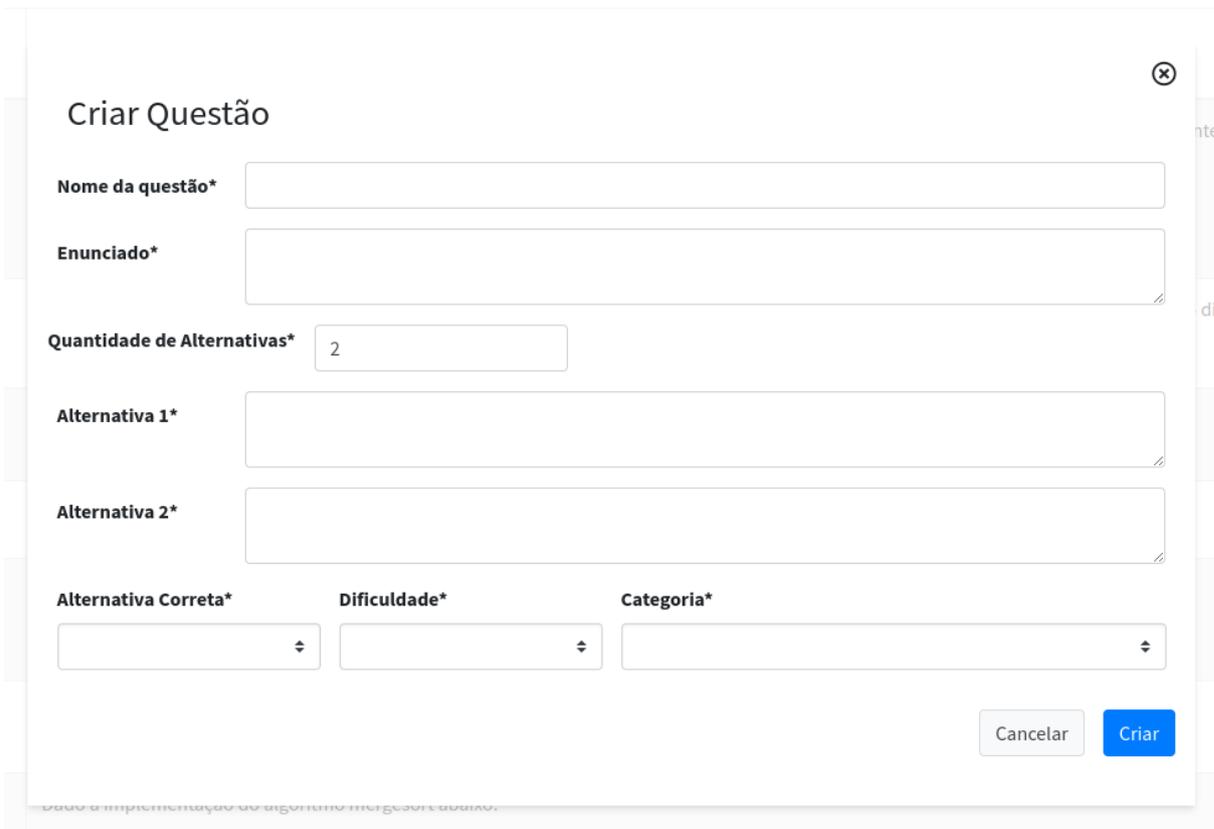
Feito o login, o aluno pode alterar sua senha clicando no botão com símbolo de 'avatar' no canto superior direito para abrir o menu do usuário, e logo em seguida clicar no botão 'Alterar Senha', que o redirecionará para a página de alteração de senha.

6.2.2 Iniciar questionário

Para iniciar a resolução do questionário, o aluno deve clicar na opção 'Questionários' no menu, onde basta clicar no botão indicado para o início da resolução, conforme

²<<http://questionariosri.herokuapp.com/>>

Figura 6.10: Criação de questão



O formulário 'Criar Questão' apresenta os seguintes campos:

- Nome da questão***: Campo de texto para o título da pergunta.
- Enunciado***: Área de texto grande para o conteúdo da pergunta.
- Quantidade de Alternativas***: Campo de entrada numérica com o valor '2' preenchido.
- Alternativa 1***: Campo de texto para a primeira opção de resposta.
- Alternativa 2***: Campo de texto para a segunda opção de resposta.
- Alternativa Correta***: Menu suspenso para selecionar a resposta correta.
- Dificuldade***: Menu suspenso para definir o nível de dificuldade.
- Categoria***: Menu suspenso para classificar a questão.

Botões de ação: 'Cancelar' (cinza) e 'Criar' (azul).

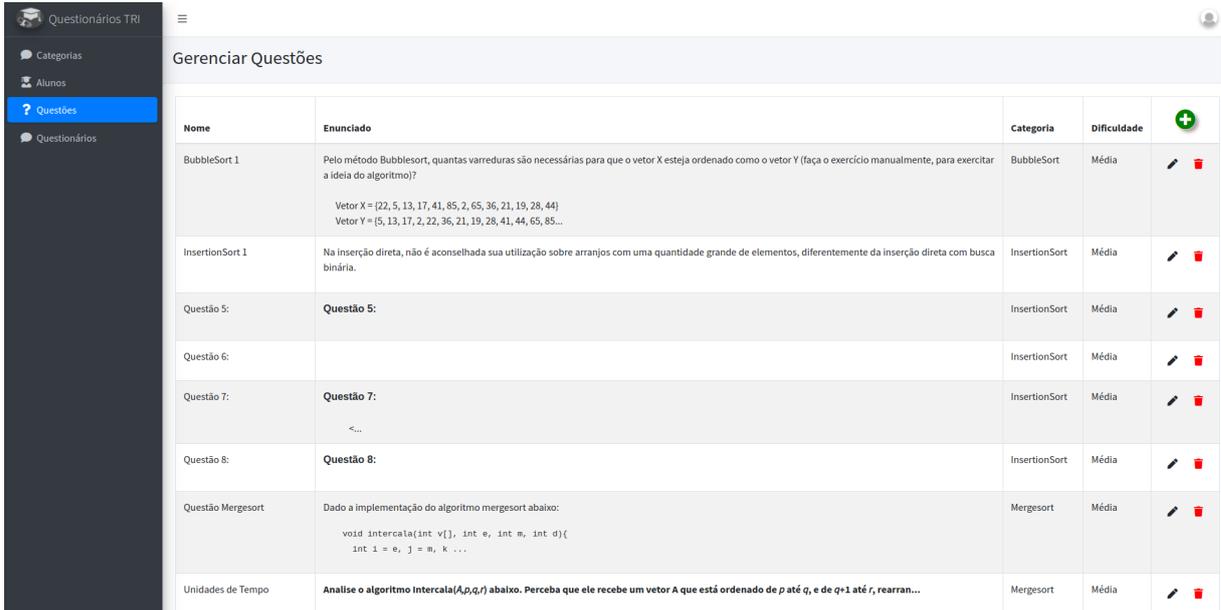
Fonte: <<http://questionariostri.herokuapp.com/manage-question>>

mostra a figura 6.16.

6.2.3 Responder questionário

Conforme a Figura 6.17 nos mostra, para responder ao questionário basta o aluno ler o enunciado apresentado, selecionar a alternativa que julga correta, e clicar em 'Responder'. Esse processo se repete até o fim do questionário.

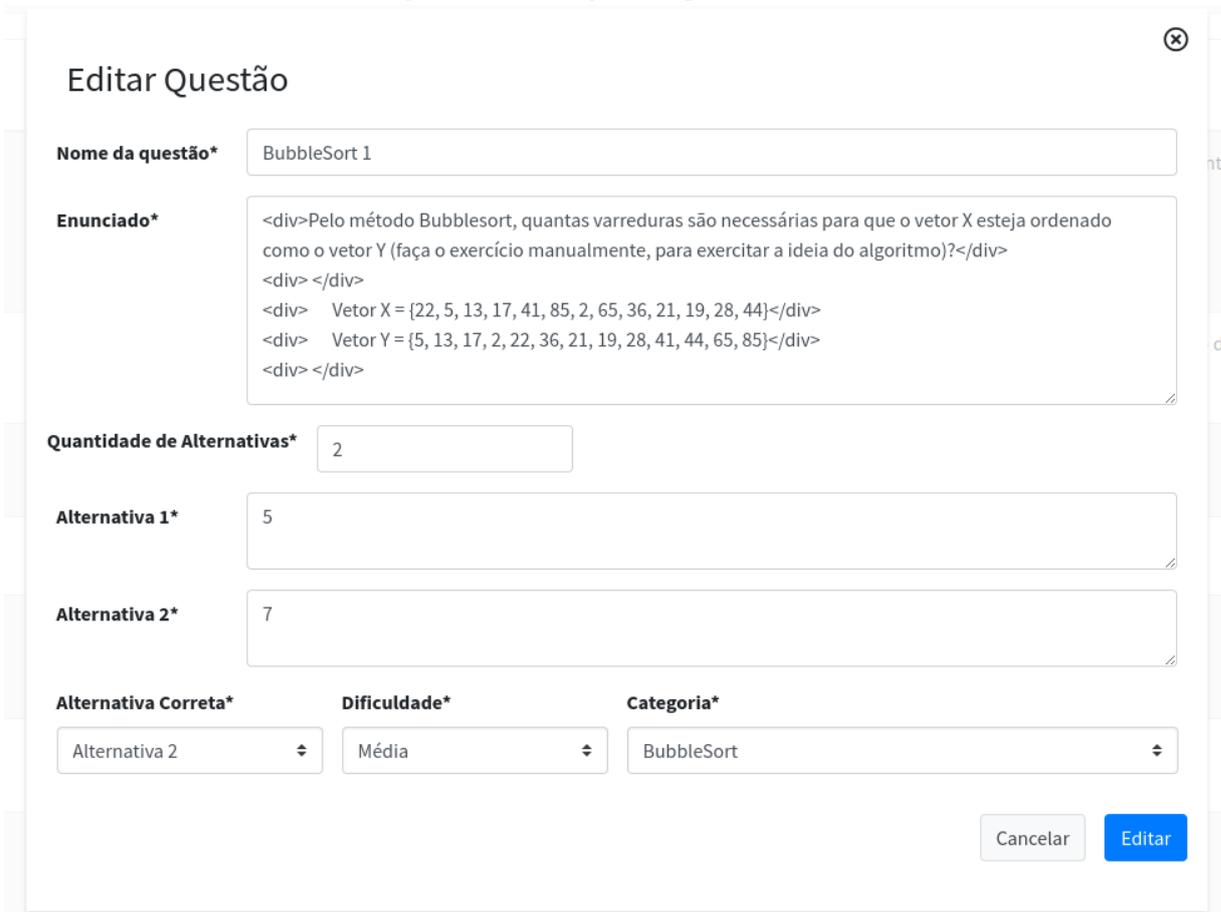
Figura 6.11: Listagem de questões



Nome	Enunciado	Categoria	Dificuldade	
BubbleSort 1	Pelo método Bubblesort, quantas varreduras são necessárias para que o vetor X esteja ordenado como o vetor Y (faça o exercício manualmente, para exercitar a ideia do algoritmo)? Vetor X = {22, 5, 13, 17, 41, 85, 2, 65, 36, 21, 19, 28, 44} Vetor Y = {5, 13, 17, 2, 22, 36, 21, 19, 28, 41, 44, 65, 85...	BubbleSort	Média	 
InsertionSort 1	Na inserção direta, não é aconselhada sua utilização sobre arranjos com uma quantidade grande de elementos, diferentemente da inserção direta com busca binária.	InsertionSort	Média	 
Questão 5:	Questão 5:	InsertionSort	Média	 
Questão 6:		InsertionSort	Média	 
Questão 7:	Questão 7: <...>	InsertionSort	Média	 
Questão 8:	Questão 8:	InsertionSort	Média	 
Questão Mergesort	Dado a implementação do algoritmo mergesort abaixo: <pre>void intercala(int v[], int e, int m, int d){ int i = e, j = m, k ...</pre>	Mergesort	Média	 
Unidades de Tempo	Análise o algoritmo intercala(A,p,q,r) abaixo. Perceba que ele recebe um vetor A que está ordenado de p até q, e de q+1 até r, rearran...	Mergesort	Média	 

Fonte: <<http://questionariostri.herokuapp.com/manage-question>>

Figura 6.12: Edição de questão



Editar Questão

Nome da questão*

Enunciado*

Quantidade de Alternativas*

Alternativa 1*

Alternativa 2*

Alternativa Correta* **Dificuldade*** **Categoria***

Fonte: <<http://questionariostri.herokuapp.com/manage-question>>

Figura 6.13: Criação de questionários

Criar Questionário

Alunos*

Categoria*

Mínimo de Questões* 15

Fonte: <<http://questionariostri.herokuapp.com/manage-questionnaire>>

Figura 6.14: Listagem de questionários

Gerenciar Questionários

Aluno	Categoria	Questões	Status	
sdasada@asdas.com	Algoritmos	15	Em progresso	
aluninho@email.com	Algoritmos	15	Em progresso	
nome1nome2@ashdas.com	Algoritmos	15	Em progresso	
alunoteste@email.com	reais	15	Em progresso	
naoregistrado@email.com	Algoritmos	15	Criado	

Fonte: <<http://questionariostri.herokuapp.com/manage-questionnaire>>

Figura 6.15: Edição de questionário

Editar Questionário

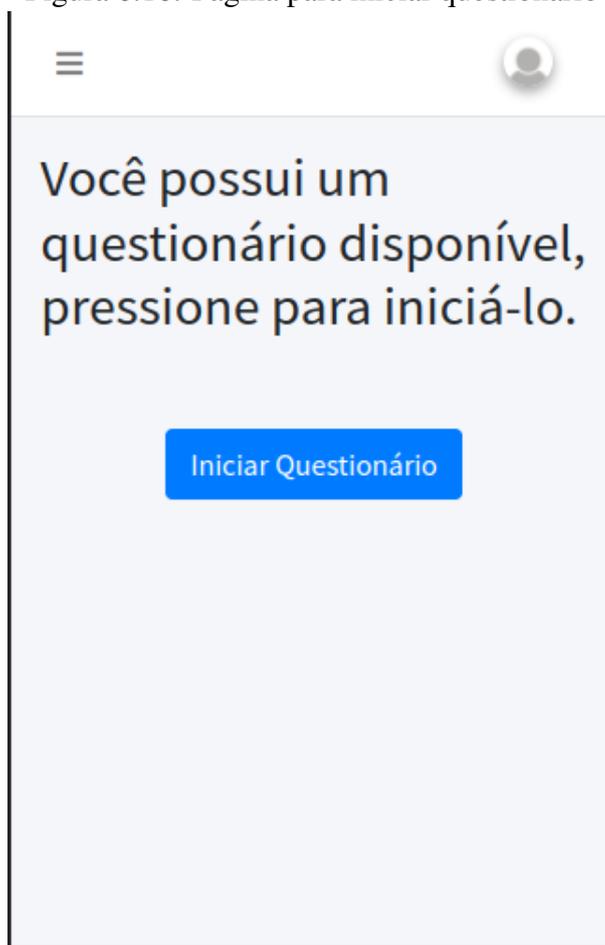
Alunos*

Categoria*

Mínimo de Questões* 15

Fonte: <<http://questionariostri.herokuapp.com/manage-questionnaire>>

Figura 6.16: Página para iniciar questionário



Fonte: <<http://questionariosri.herokuapp.com/questionnaire>>

Figura 6.17: Resolução do questionário

Responda a seguinte questão:

Pelo método Bubblesort, quantas varreduras são necessárias para que o vetor X esteja ordenado como o vetor Y (faça o exercício manualmente, para exercitar a ideia do algoritmo)?

Vetor X = {22, 5, 13, 17, 41, 85, 2, 65, 36, 21, 19, 28, 44}

Vetor Y = {5, 13, 17, 2, 22, 36, 21, 19, 28, 41, 44, 65, 85}

5

7

Responder

Fonte: <<http://questionariostri.herokuapp.com/questionnaire>>

7 AVALIAÇÃO DO AMBIENTE

Para a avaliação do ambiente desenvolvido, foi aplicado o questionário de validação, que pode ser verificado no apêndice A, para 6 voluntários, sendo 5 alunos e 1 professor. O professor que participou da avaliação foi o cliente especialista (orientador). O fato de apenas um professor ter participado da avaliação é um aspecto limitante deste trabalho, o ideal seria termos mais professores voluntários.

Conforme detalhado na seção 4.1, o questionário escolhido foi o SUS¹, com o objetivo de avaliar a experiência do usuário sobre a usabilidade do ambiente. Além disso, conforme ilustra as 3 últimas questões presentes no apêndice A, também foram feitas perguntas abertas com o objetivo de coletar informações sobre possíveis dificuldades ao se utilizar a aplicação, sugestões de melhorias e possíveis vantagens ou benefícios que os participantes acreditam a aplicação possui.

7.1 Avaliação do Questionário SUS

Analisando as respostas coletadas, podemos verificar que 4 questões se destacaram positivamente e com resultados equivalentes, como ilustra as figuras 7.1, 7.2, 7.3 e 7.4. Essas respostas indicam, de modo geral, uma facilidade no uso e um sentimento de confiança por parte dos voluntários, que não precisaram de ajudas técnicas para utilizar o sistema, que acreditam que as pessoas rapidamente aprenderão como usar o sistema e que não precisaram aprender coisas novas para utilizar a aplicação.

Como destaque negativo, temos a resposta indicada na figura 7.5. Nela, percebemos que poucos voluntários gostariam de utilizar a aplicação com frequência. Na seção 7.2 analisamos as respostas abertas dos participantes, o que pode nos dar sugestões de como melhorar o ambiente para que alunos e professores passem a querer utilizá-lo com mais frequência.

De acordo com Brooke (1996), para calcular corretamente o resultado de um Questionário SUS, devemos:

1. Nas questões pares, atribuir 5 menos o valor da resposta
2. Nas questões ímpares, atribuir o valor da resposta subtraído de 1
3. Somar as respostas de cada voluntário e multiplicar por 2.5

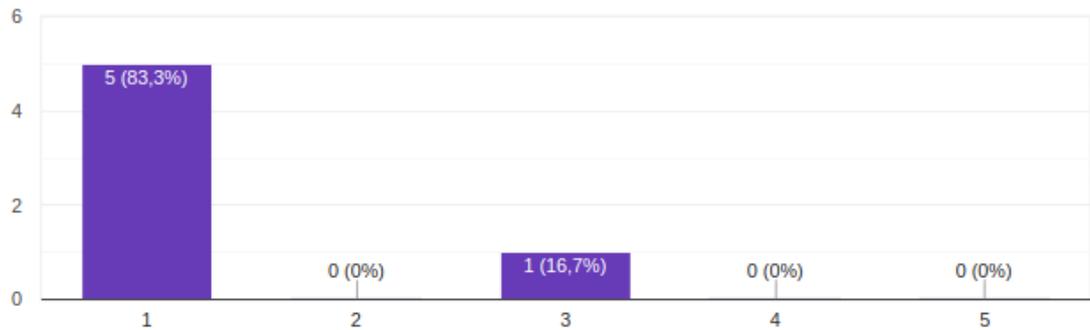
¹<<https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>>

Figura 7.1: Respostas da questão 4 do Questionário SUS

Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.



6 respostas



Fonte: O autor

4. O valor final deve estar em uma escala entre 0 e 100

Aplicando esses passos para as respostas de cada voluntário, temos:

- Voluntário 1 (professor): 77,5
- Voluntário 2: 95
- Voluntário 3: 50
- Voluntário 4: 97,5
- Voluntário 5: 95
- Voluntário 6: 90
- Média: 84,1

Podemos observar que a média final das respostas ficou bem acima de 68, que é a média indicada pela literatura Brooke (1996), o que indica que, de modo geral, os usuários participantes do teste tiveram uma boa experiência de usabilidade.

7.2 Avaliação das questões abertas

Analisando as respostas das últimas questões do questionário, podemos ter uma orientação para futuras melhorias que podem ser feitas no ambiente.

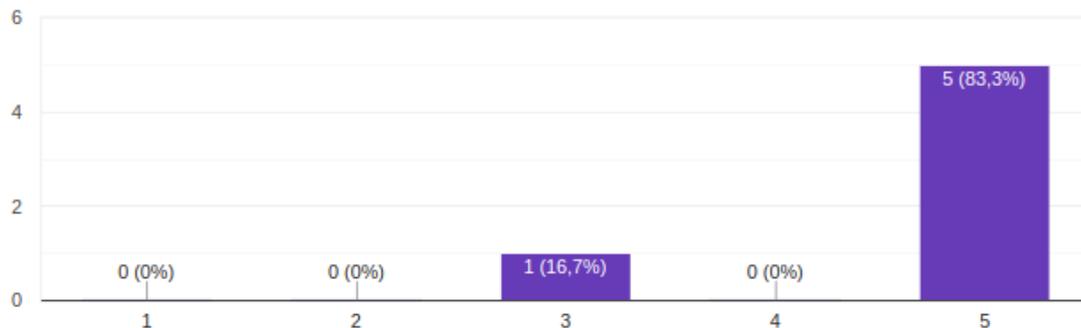
A primeira questão pergunta se o participante teve dificuldades ao utilizar a aplicação. A maioria dos estudantes voluntários não teve nenhuma dificuldade, entretanto o

Figura 7.2: Respostas da questão 7 do Questionário SUS

Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.



6 respostas



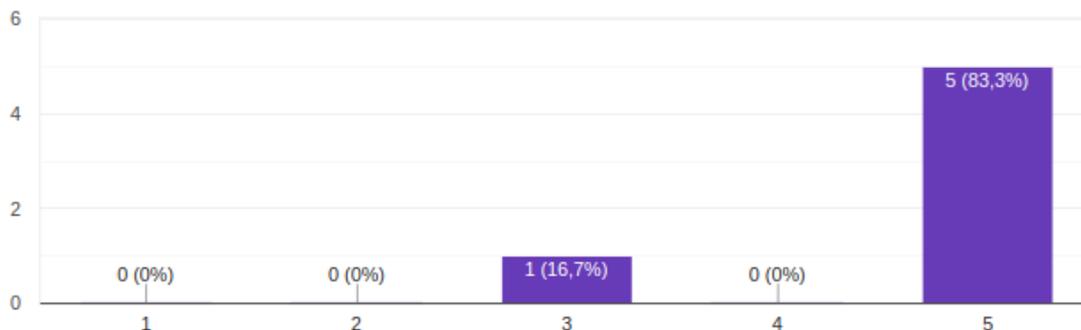
Fonte: O autor

Figura 7.3: Respostas da questão 9 do Questionário SUS

Eu me senti confiante ao usar o sistema.



6 respostas



Fonte: O autor

professor voluntário apontou diversas dificuldades, dentre elas:

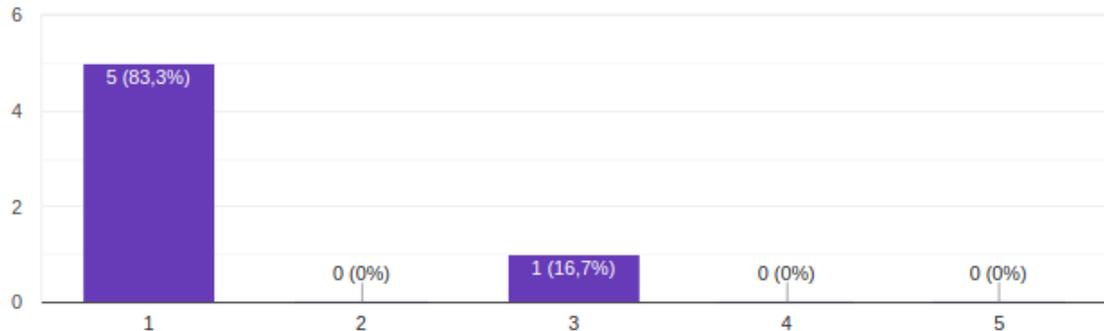
- O fato das questões serem estritamente de múltipla escolha e com apenas uma alternativa correta torna difícil a configuração de questões de V ou F.
- Não ter a funcionalidade de reordenar, filtrar ou buscar por questões específicas no módulo de gerenciamento de questões, nem como selecionar várias para realizar uma única ação conjunta.
- Não ter a funcionalidade de pré-visualizar uma questão ou um questionário.
- Não ter a funcionalidade de cadastrar alunos a partir de uma lista ou arquivo.
- Dúvidas sobre o cadastro de alunos e de professores.

Figura 7.4: Respostas da questão 10 do Questionário SUS

Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.



6 respostas



Fonte: O autor

Isso deixa claro que a aplicação tem uma boa margem para futuras melhorias, principalmente para melhorar a usabilidade dos professores.

A segunda questão aberta pediu sugestões de melhoria para a aplicação. Dentre as principais sugestões dos alunos voluntários, temos:

- Opção de voltar para a questão anterior.
- Possibilidade de acessar com o protocolo https.
- Visualizar o nome do questionário.

As sugestões do professor se resumem em corrigir ou implementar o que foi apontado por ele como dificuldades.

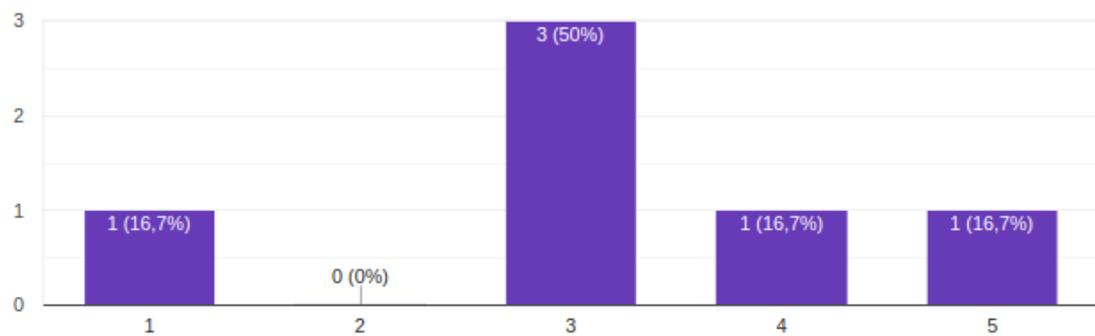
A última questão pergunta quais vantagens ou benefícios o voluntário vê na aplicação. Suas respostas, conforme ilustra a figura 7.6, mostram que a simplicidade e a facilidade no uso foram percebidas por alguns voluntários, além da vantagem de possibilitar o uso por dispositivos de diversos tamanhos.

Figura 7.5: Respostas da questão 1 do Questionário SUS

Eu acho que gostaria de usar esse sistema com frequência.

 Copiar

6 respostas



Fonte: O autor

Figura 7.6: Respostas da última questão aberta

Quais vantagens ou benefícios você acredita que a aplicação tem?

6 respostas

Criação de questionários dinâmicos, adaptação para conteúdos em dispositivos de diferentes tamanhos. Simplicidade de uso por parte dos estudantes.

Simplicidade e fluidez.

não sei

Auxilia na avaliação do aluno por parte do professor, usando uma metodologia de avaliação mais moderna.

Achei intuitivo, fácil e prático de utilizar. A visualização e a compreensão são melhores do que os questionários feitos diretamente no moodle, na minha opinião. O acesso também é mais direto do que avaliações realizadas dentro das turmas/cursos no moodle.

Achei bem simples, performática, direto ao ponto.

Fonte: O autor

8 CONCLUSÕES

Neste trabalho, foi proposto um ambiente online responsivo, que possibilite o uso de *smartphones* pelos alunos, para permitir a realização de questionários dinâmicos baseados na TRI. De maneira que torne o processo avaliativo mais eficiente para o professor, por permitir a utilização da TRI, e amigável para o estudante.

Através da análise da avaliação do ambiente, presente no capítulo 7, percebemos que este objetivo foi cumprido. Os professores podem gerenciar os diversos aspectos necessários para uma correta administração de questionários dinâmicos baseados na TRI, e os alunos voluntários, em sua maioria, acharam o sistema simples e intuitivo. Sua usabilidade se demonstrou bem positiva, com uma pontuação média de 84,1 no Questionário SUS, bem acima da média de 68 pontos, que é a média indicada em Brooke (1996).

Todavia, diversos aspectos podem ser melhorados, principalmente em funcionalidades específicas dos professores. Conforme apontado na seção 7.2, o professor voluntário sentiu dificuldades que certamente poderiam ser sentidas por outros professores que usassem a aplicação, como o fato das questões serem estritamente de múltipla escolha e com apenas uma alternativa correta, dificuldades na visualização de questões e questionários, além de sentir falta de diversas funcionalidades que facilitariam o gerenciamento das categorias, alunos, questões e questionários. Além disso, uma funcionalidade atrativa para alunos que não foi desenvolvida é permitir o retorno para a questão anterior sem que isso afete o sorteio dinâmico. Outra limitação foi não ter focado em análise de segurança e na LGPD, além de não ter sido feito um estudo de escalabilidade.

Para futuros trabalhos, sugere-se a elaboração de melhorias na interface do atual trabalho, e que sejam testadas com mais professores e alunos voluntários, para que as atuais dificuldades e limitações observadas possam ser mitigadas, e o ambiente possa ser mais amigável para os professores.

REFERÊNCIAS

ADOLFO, A. **Ambiente para Geração de Questionários Dinâmicos Baseados em Teoria da Resposta ao Item**. Thesis (PhD) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2021. Available from Internet: <<http://www.bibliotecadigital.ufrgs.br/da.php?nrb=001128286&loc=2021&l=304efd92bda5fe7e>>.

ANDRADE, D.; TAVARES, H.; VALLE, R. **Teoria da Resposta ao Item: Conceitos e Aplicações**. [S.l.]: SINAPE, 2000.

BRAGION, M. **Análise combinada de exames vestibulares da Universidade Federal de Lavras usando a teoria de resposta ao item**. Thesis (PhD) — Universidade Federal de Lavras, Lavras, 2010.

BROOKE, J. Sus: a “quick and dirty” usability. **Usability evaluation in industry**, v. 189, n. 3, 1996.

CARDOSO, R. F. Avalweb: sistema interativo para gerência de questões e aplicação de avaliações na web. 2001.

CONDE, J. L. F. Unidimensionalidade e a propriedade de invariância das estimativas da habilidade pela tri. **Avaliação Psicológica**, Ribeirão Preto, Brasil, v. 6, n. 2, p. 205–215, 2007.

KIMURA, A. O. T.; NAGAOKA, K. Moodle ucat: a computer-adaptive test module for moodle based on the rasch model. In: **Proceedings of the fifth International Conference on Measurement in Health, Education, Psychology and Marketing**. Perth, Australia: [s.n.], 2012. p. 23–25.

Li, L.; Chou, W. Design and describe rest api without violating rest: A petri net based approach. In: **2011 IEEE International Conference on Web Services**. [S.l.: s.n.], 2011. p. 508–515.

LUNET, N.; SEVERO, M.; BARROS, H. Desvio padrão ou erro padrão. **Arquivos de Medicina**, v. 20, 02 2006.

MEIRELLES, F. de S. **Panorama do Uso de TI no Brasil**. 32. ed. São Paulo, 2021. (Pesquisa Anual do Uso de TI). Available from Internet: <<https://eaesp.fgv.br/sites/eaesp.fgv.br/files/u68/fgvcia2021pesti-panorama.docx>>.

NIELSEN, J.; LANDAUER, T. K. A mathematical model of the finding of usability problems. In: **Proceedings of the INTERACT’93 and CHI’93 conference on Human factors in computing systems**. [S.l.: s.n.], 1993. p. 206–213.

Ortegon, G. M. O.; Wives, L. Analysis of questionnaires for virtual learning environments based on item response theory. In: **Anais do XXX Simpósio Brasileiro de Informática na Educação (SBIE 2019)**. [S.l.: s.n.], 2019.

ORTEGON, O. **ANALYSIS AND ADAPTATION OF QUESTIONNAIRES BASED ON ITEM RESPONSE THEORY**. Dissertation (Master) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 6 2019. Advisor: Prof. Dr. Leandro Krug Wives.

PASQUALI, L.; PRIMI, R. Fundamentos da teoria da resposta ao item -tri. **PePSIC**, v. 2, n. 2, p. 99–110, 2003.

SANTOS, F.; GUEDES, L. Testes adaptativos informatizados baseados em teoria de resposta ao item utilizados em ambientes virtuais de aprendizagem. **RENOTE - Revista Novas Tecnologias na Educação**, v. 3, n. 2, 2005.

STAKER, H.; HORN, M. B. Classifying k–12 blended learning. Innosight Institute, 2012.

VALENTE, J. A. Blended learning e as mudanças no ensino superior: a proposta da sala de aula invertida. **Educar em revista**, SciELO Brasil, p. 79–97, 2014.

WU, M.; TAM, H.; JEN, T. Classical test theory. In: **Educational Measurement for Applied Researchers**. Singapore: Springer, 2016. p. 73–90.

APÊNDICE A — QUESTIONÁRIO DE VALIDAÇÃO

Feedback - Questionários TRI

***Obrigatório**

1. Você utilizou a aplicação como professor ou como aluno? *

Marcar apenas uma oval.

Professor

Aluno

2. Eu acho que gostaria de usar esse sistema com frequência. *

Marcar apenas uma oval.

1 2 3 4 5

Discordo completamente Concordo completamente

3. Eu achei o sistema desnecessariamente complexo *

Marcar apenas uma oval.

1 2 3 4 5

Discordo completamente Concordo completamente

4. Eu achei o sistema fácil de usar *

Marcar apenas uma oval.

1 2 3 4 5

Discordo completamente Concordo completamente

5. Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo completamente	<input type="radio"/>	Concordo completamente				

6. Eu acho que as funções do sistema estão muito bem integradas *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo completamente	<input type="radio"/>	Concordo completamente				

7. Eu acho que o sistema apresenta muita inconsistência. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo completamente	<input type="radio"/>	Concordo completamente				

8. Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo completamente	<input type="radio"/>	Concordo completamente				

9. Eu achei o sistema atrapalhado de usar. *

Marcar apenas uma oval.

1 2 3 4 5

Discordo completamente Concordo completamente

10. Eu me senti confiante ao usar o sistema. *

Marcar apenas uma oval.

1 2 3 4 5

Discordo completamente Concordo completamente

11. Eu precisei aprender várias coisas novas antes de conseguir usar o sistema. *

Marcar apenas uma oval.

1 2 3 4 5

Discordo completamente Concordo completamente

12. Houveram dificuldades ao utilizar a aplicação? Se sim, quais? *

13. Você tem sugestões de melhoria para a aplicação? Quais? *

14. Quais vantagens ou benefícios você acredita que a aplicação tem? *

Este conteúdo não foi criado nem aprovado pelo Google.

Google Formulários

APÊNDICE B — GUIA DE INSTALAÇÃO

Aqui serão apresentados os passos para instalação e configuração do ambiente *front-end* e da aplicação REST em um ambiente computacional.

B.1 Instalação do Ambiente *front-end*

Os requisitos para a instalação do ambiente *front-end* são:

1. *npm* instalado.
2. *node* instalado.
3. Porta 8080 liberada, ou configurar outra porta para uso.

Garantindo que os requisitos estejam instalados, deve-se clonar o repositório da aplicação *front-end*¹, acessar a pasta raiz do projeto e executar o comando `npm run serve`.

Após a execução, a aplicação deve estar rodando com sucesso na máquina, e para usá-la basta acessar `http://localhost:8080/`. Entretanto, seu funcionamento só será correto após a instalação da aplicação REST.

B.2 Instalação da Aplicação REST

Os requisitos para a instalação da aplicação REST são:

1. *docker* instalado.
2. *docker-compose* instalado.
3. Porta 3000 liberada, ou configurar outra porta para uso.

Garantindo que o ambiente computacional possui os requisitos necessários, deve-se clonar o repositório da aplicação REST², acessar a pasta raiz do projeto e executar o comando `docker-compose up -d`. Este comando irá executar dois *containers docker*, um deles chamado `<nomedapastaraiz>_myapp_1`, contendo o ambiente virtual para a execução da aplicação REST, utilizando o framework Laravel, e outro chamado `<nomedapastaraiz>_mariadb_1`, contendo um banco de dados para persistência

¹<<https://github.com/castilhos90124/questionnaire-generator-frontend>>

²<<https://github.com/arthuradolfo/TRI-Questionnaire-Generator>>

dos dados da aplicação REST.

Após a execução com sucesso dos containers, devem ser executados os seguintes comandos:

- `docker exec <nomedapastaraiz>_myapp_1 php artisan passport:install:` esse comando irá criar os segredos necessários para a implementação da autenticação *OAuth2*.
- `docker exec <nomedapastaraiz>_myapp_1 sh setup_mirt.sh:` esse comando irá instalar as dependências necessárias para executar a biblioteca MIRT do R, utilizada para calcular o modelo TRI.

Executados estes passos, a aplicação estará rodando com sucesso no endereço e porta `http://localhost:3000`.