

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA**  
**FACULDADE DE TECNOLOGIA DE BOTUCATU**  
**CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE**  
**SISTEMAS**

**ARIEL BARCELLOS DE OLIVEIRA MÜLLER**

**SOFTWARE PARA CRIAÇÃO E APLICAÇÃO DE TESTES EM FORMATO DE**  
**QUIZ**

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA  
FACULDADE DE TECNOLOGIA DE BOTUCATU  
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE  
SISTEMAS**

**ARIEL BARCELLOS DE OLIVEIRA MÜLLER**

**SOFTWARE PARA CRIAÇÃO E APLICAÇÃO DE TESTES EM FORMATO DE  
QUIZ**

Orientador: Prof. Me. Renato Luiz Gambarato

Trabalho de Conclusão de Curso apresentado à  
FATEC – Faculdade de Tecnologia de  
Botucatu para obtenção do título de Tecnólogo  
no Curso Superior de Análise e  
Desenvolvimento de Sistemas.

Botucatu-SP  
Junho – 2018

## SUMÁRIO

	Páginas
<b>1 INTRODUÇÃO.....</b>	<b>3</b>
<b>2 CARACTERIZAÇÃO DA TECNOLOGIA UTILIZADA.....</b>	<b>5</b>
2.1 Node.js.....	5
2.1.1 <i>Node Package Manage (NPM)</i> .....	5
2.2 Express.....	6
2.3 React.js.....	6
2.4 Redux.....	6
2.4.1 Redux-form.....	6
2.5 Passport.....	7
2.6 MongoDB.....	7
2.6.1 Mongoose.....	7
<b>3 FERRAMENTAS DE DESENVOLVIMENTO.....</b>	<b>8</b>
<b>4. DESCRIÇÃO DO SOFTWARE.....</b>	<b>10</b>
4.1 Tela de autenticação e inscrição.....	10
4.2 Barra de cabeçalho.....	11
4.3 Tela principal.....	11
4.4 Perfil do usuário.....	12
4.4.1 Editar perfil.....	13
4.5 Criar quizz.....	13
4.5.1 Criar questão.....	14
4.5.2 Criar alternativas.....	15
4.5.3 Escolha da alternativa correta.....	16
4.5.4 Quizz salvo com sucesso.....	16
4.6 Realizar Quizz.....	17
4.6.1 Tela de começo.....	17
4.6.2 Questão.....	18
4.6.3 Resposta correta.....	18
4.6.4 Resposta errada.....	19
4.6.5 Resultados do Quizz.....	20
<b>5 APLICABILIDADE.....</b>	<b>21</b>
<b>REFERÊNCIAS.....</b>	<b>22</b>

## 1. INTRODUÇÃO

O trabalho realizado consistiu em desenvolver uma aplicação com interface web que possibilite a criação e aplicação de testes em formatos de quizzes voltados para atividades pedagógicas na internet, em uma rede interna ou até localmente em uma estação de trabalho.

Para isso foi necessário que as ferramentas utilizadas no desenvolvimento permitissem escalabilidade e flexibilidade para que a aplicação não se limitasse ao que já está implementado, mas que também pudesse ser modificada com facilidade para atender a diferentes restrições.

Para isso, um fator muito importante na concepção deste projeto foi que a aplicação não deveria depender de recursos da internet para funcionar e nem de muitos recursos, para que pudesse ser implementado em lugares com diferentes níveis de infraestrutura.

Em relação aos testes, alguns aspectos foram muito importantes para justificar o modelo adotado. Baseado nas máquinas de ensinar de SKINNER (1958), os testes deveriam fornecer *feedback* imediatamente após cada resposta com os dados sobre acerto ou erro, além de uma breve explicação sobre o assunto questionado.

Hoje, muitas atividades são colocadas aos alunos em escolas e faculdades de uma forma um tanto quanto incoerente. O aluno realiza uma lista de exercícios para entregar ao professor, porém só tem o *feedback* sobre seus acertos e erros com o prazo de até semanas.

Há essa altura, muitas vezes o aluno já não sabe mais a razão que o levou a errar a questão. Além disso, muitas vezes as correções se limitam a dizer se a resposta do aluno está certa ou errada, o que é insuficiente para que o aluno realmente aprenda.

O criador do quiz, por outro lado, deve ter também um *feedback* sobre o rendimento dos alunos em seus testes, para que assim verifique como está o desempenho geral das pessoas que fizeram os testes.

Uma vez que a aplicação dá o *feedback* automaticamente ao aluno, o professor pode dedicar o tempo que tomaria para corrigir os trabalhos para avaliar o desempenho da turma e planejar suas próximas atividades.

Após levantar todos os requisitos e restrições do projeto e pesquisar sobre ferramentas em potencial para o desenvolvimento da aplicação, decidiu-se por utilizar a linguagem de programação JavaScript através do interpretador Node.js.

Havia sido escolhida a ferramenta Angular 5 para implementar a interface do usuário, porém, como a ferramenta utiliza linguagem TypeScript, foi decidido por utilizar a ferramenta React.js que utiliza apenas JavaScript para implementação, uma vez que o servidor já usará a linguagem.

## 2. CARACTERIZAÇÃO DA TECNOLOGIA UTILIZADA

### 2.1 Node.js

Node.js é um interpretador de códigos em linguagem JavaScript, baseado no interpretador Chrome V8 JavaScript Engine, que trabalha de forma assíncrona e dirigida a eventos. Em um servidor web Node, várias conexões podem ser trabalhadas concorrentemente e para cada conexão uma função de resposta será chamada. Quando não há trabalho a ser feito o servidor Node entra em estado de espera (Node.js Foundation, 2018).

O Node.js, segundo a Node.js Foundation (2018), é uma ferramenta leve, eficiente e ideal para desenvolver sistemas escaláveis. A ferramenta conta ainda com o maior ecossistema de bibliotecas de código aberto do mundo, o *Node Package Manager* (NPM).

#### 2.1.1 *Node Package Manager* (NPM)

NPM é um gerenciador de pacotes JavaScript e contém o maior registro de softwares do mundo, um grande banco de dados de software JavaScript. Com aproximadamente 3 bilhões de downloads por semana e mais de 600 mil pacotes, o NPM consiste em 3 componentes: O site, a interface de linha de comando e os registros (Npm, 2018).

O NPM foi utilizado para incorporar *frameworks* e bibliotecas de códigos ao software desenvolvido neste trabalho. Sendo que os principais são abordados nos próximos tópicos: Express, React.js, Redux, Redux-form e Passport.

## 2.2 Express

EXPRESS (2018) é um *framework* JavaScript para Node.js com propósito de ser rápido e minimalista. Consiste em uma aplicação web muito flexível que provê uma série de recursos robustos para aplicações web e mobile.

Com uma quantidade enorme de métodos HTTP, Express ajuda o desenvolvedor a implementar rotas e tratar os pedidos HTTP. Com Express se torna fácil trabalhar com o conteúdo dos pedidos, abstraindo parâmetros, corpo e cabeçalho em métodos do *framework* (Express, 2018).

## 2.3 React.js

Segundo FACEBOOK (2018), React.js é uma biblioteca JavaScript para construção de interfaces. Baseado em componentes, ele permite criar componentes encapsulados e manipular através de estados a composição de uma interface de usuário complexa.

Por esse motivo, o React.js torna muito mais simples a implementação de uma aplicação com conceito de *Single Page App*, onde as diversas telas da aplicação estão em uma única rota do navegador.

## 2.4 Redux

Segundo REDUX (2018), Redux é uma ferramenta que fornece contêiner de estados para aplicações JavaScript. Ele ajuda a escrever aplicações que se comportem consistentemente em diferentes ambientes (cliente, servidor ou nativo).

Ele utiliza o conceito de que estados de um componente só podem ser alterados através de uma ação, assegurando assim que nem as *views* e as funções de resposta do servidor poderão escrever diretamente o estado da aplicação. Por exemplo, se eu tenho os dados de usuário armazenados através do Redux, a aplicação somente poderá alterar esses dados através da chamada de uma ação já descrita no código (Redux, 2018).

### 2.4.1 Redux-form

Segundo RAMUSSEN (2018), Redux-form é uma biblioteca JavaScript para Redux que abstrai a criação a manipulação de formulários dentro de uma aplicação React. Seu funcionamento consiste em utilizar criadores de ações para guardar as informações de um formulário no contêiner Redux.

## 2.5 Passport

Segundo HANSON (2018), Passport é uma ferramenta para lidar com autenticações dentro de uma aplicação Node. Ele é extremamente flexível e modular e pode ser usado em qualquer aplicação web baseada em Express.

A biblioteca do Passport contém mais de 500 estratégias de autenticação, desde local (usuário e senha) até utilizando provedor OAuth como Google, Facebook, Twitter, entre outras (Hanson, 2018).

## 2.6 MongoDB

De acordo com MongoDB (2018), MongoDB é um sistema de gerenciamento de banco de dados não relacional de código aberto. Foi desenhado pensando em facilidade no desenvolvimento e escalabilidade.

Um registro em MongoDB é um documento e sua estrutura de dados é composta por pares de campo e valor, similares ao objetos JSON (MongoDB, 2018).

### 2.6.1 Mongoose

Mongoose é um pacote para aplicações Node.js que fornece uma solução direta e baseada em esquemas para modelar os dados da aplicação. Inclui recursos integrados para fazer *casting* de tipos, validações e construção de *queries* (Mongoose, 2018).



### 3. FERRAMENTAS DE DESENVOLVIMENTO

O ambiente de desenvolvimento da aplicação consistiu em:

- Laptop com processador Intel Core i3-5005 CPU @ 2.00GHz \* 2, 3.8 Gb de memória RAM e disco rígido de 476.6 Gb;
- Sistema operacional Linux Mint 18.3 Cinnamon 64-bit;
- Editor de texto para códigos Microsoft Visual Studio Code;
- Editor de texto para códigos Sublime Text;
- Navegador Google Chrome;
- Navegador Mozilla Firefox;
- Emulador de terminal Terminator.

A metodologia utilizada para desenvolvimento do aplicativo consistiu dividir a aplicação por requisitos e em seguida implementar cada requisito em ciclos. Ao final de cada ciclo, uma nova funcionalidade estaria pronta para uso.

Os requisitos definidos para guiar a implementação foram: inscrição do usuário, autenticação de usuário, visualização de usuário, alteração de usuário, criação de novo quiz, visualização de quiz e resumo de quiz criados e feitos.

Em cada ciclo de requisito a ser implementado, aproveitando a característica flexível e escalável das tecnologias e ferramentas escolhidas, adotou-se o seguinte plano para implementação:

1. Criação de modelo do objeto para o banco de dados;
2. Implementação das rotas de back-end necessárias;
3. Implementação da comunicação client-server;
4. Implementação dos componentes de front-end;

5. Validação de dados;
6. Organização da estrutura HTML e de estilos CSS.

Ao final de cada requisito implementado, testes integrando as funcionalidades eram feitos com um banco de dados vazio. Esta prática foi adotada para corresponder a escalabilidade que permite a aplicação, uma vez que, em tempo de desenvolvimento, a cada fase era implementado aos modelos para banco de dados apenas os campos necessários até aquela funcionalidade.

Pela utilização dessa prática foi possível perceber o quão mais fácil fica o desenvolvimento de uma aplicação a partir do zero sem ter toda a modelagem de banco de dados feita.

## 4. DESCRIÇÃO DO SOFTWARE

### 4.1 Tela de autenticação e inscrição

*Figura 1: Tela de autenticação e inscrição da aplicação.*

The image shows the 'OpenQuiz' application interface. At the top, there is a green header with the text 'OpenQuiz'. Below the header, there are two main panels. The left panel is titled 'Já tem conta? Faça seu login:' and contains two input fields: 'E-mail' and 'Senha'. Below these fields is a green button labeled 'ENTRAR'. The right panel is titled 'Ainda não tem conta? Faça seu registro:' and contains three input fields: 'Primeiro Nome', 'Sobrenome', and 'E-mail'. Below these fields is another 'Senha' field. At the bottom of this panel is a green button labeled 'REGISTRAR'.

Acessando o aplicativo pela primeira vez ou sem ter se autenticado previamente, o usuário irá acessar a tela de autenticação e inscrição (Figura 1).

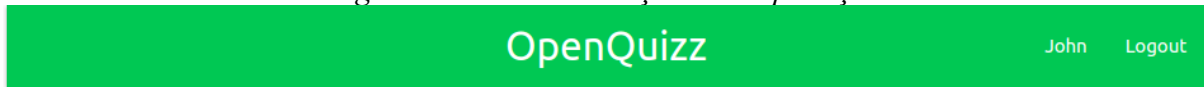
O usuário já cadastrado no sistema deverá preencher o e-mail e senha que se inscrevem para acessar o aplicativo.

O usuário ainda não cadastrado deverá preencher os dados solicitados no formulário de registro para criar uma conta.

Após a autenticação ou registro do usuário, ele será encaminhado para a página principal da aplicação.

## 4.2 Barra de cabeçalho

*Figura 2: Menu de cabeçalho da aplicação*



A barra de cabeçalho é o menu presente em toda a aplicação, enquanto a sessão do usuário estiver aberto (Figura 2).

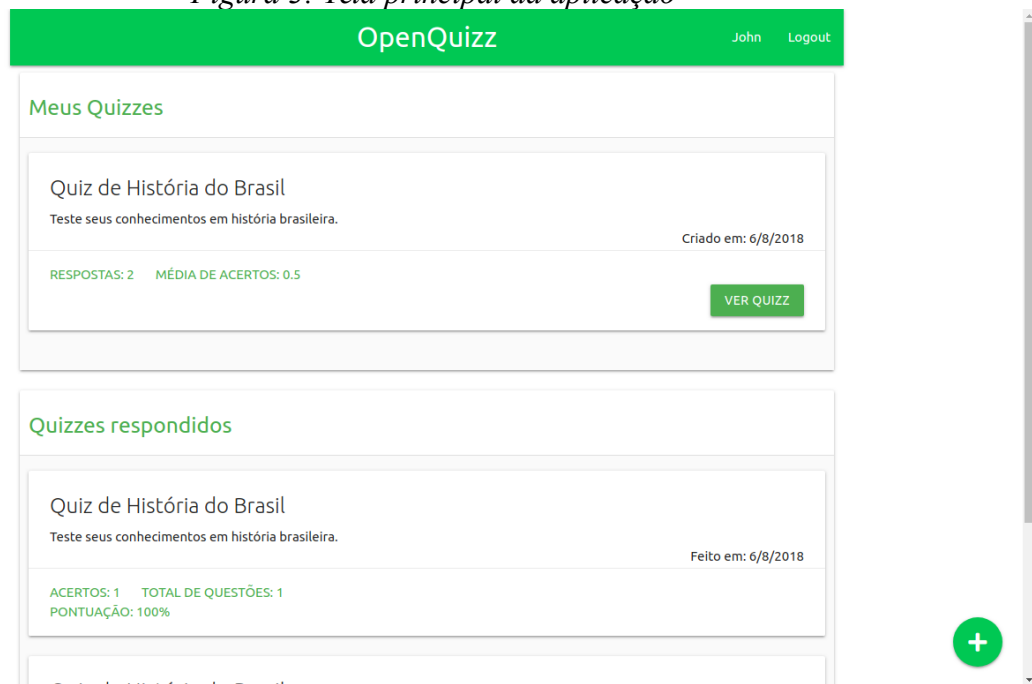
Clicando sobre o nome do usuário, será exibida a tela de Perfil do usuário (Figura 4).

Clicando sobre o link OpenQuizz, será exibida a tela principal (Figura 3).

Clicando sobre o link Logout, o usuário será desconectado do aplicativo e retornará à tela de autenticação e inscrição (Figura 1).

## 4.3 Tela principal

*Figura 3: Tela principal da aplicação*



A tela principal mostra um resumo das atividades do usuário através das seções de “Meus Quizzes” e “Quizzes respondidos”. Também na tela principal, está o botão para funcionalidade de criar um novo Quizz (Figura 3).

A seção “Meus Quizzes” mostra os Quizzes criados pelo usuário, bem como o número de respostas que o Quizz já obteve e a média de acertos.

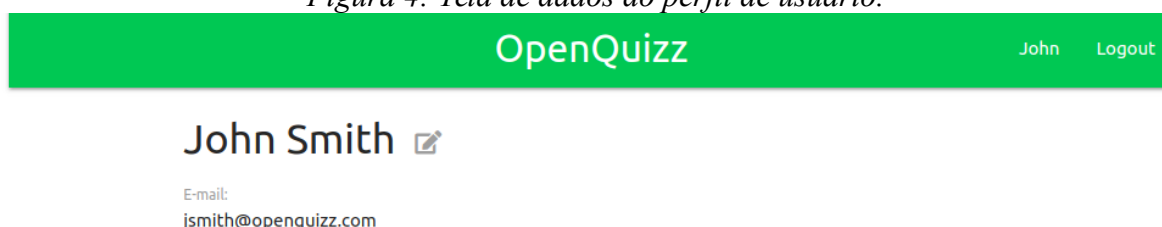
O botão “Ver Quizz” dentro de cada Quizz redireciona o usuário para o link de divulgação do Quizz e tela de começo do Quizz (Figura 11).

A seção de “Quizzes respondidos” mostra informações sobre os Quizzes já respondidos pelo usuário, incluindo o total de acertos, total de questões do Quizz e porcentagem de acertos.

O botão com o símbolo de adição, localizado no canto inferior direito da tela, permite ao usuário criar um novo Quizz .

#### 4.4 Perfil do usuário

*Figura 4: Tela de dados do perfil de usuário.*

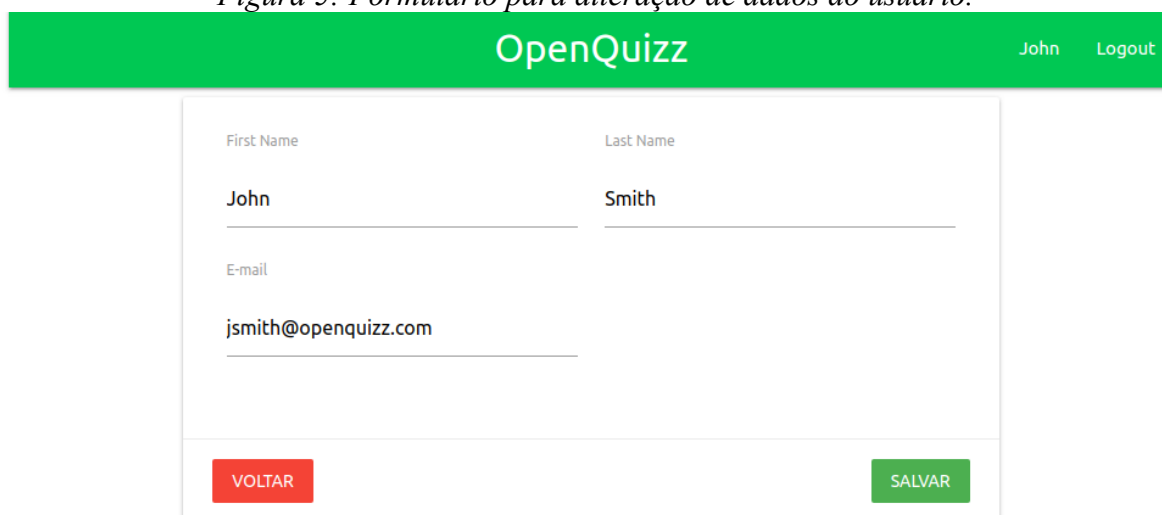


A tela de perfil do usuário mostra os dados de primeiro nome, sobrenome e e-mail do usuário (Figura 4).

O ícone ao lado direito do nome do usuário exibe o formulário para alteração dos dados de cadastro do usuário.

#### 4.4.1 Editar perfil

*Figura 5: Formulário para alteração de dados do usuário.*

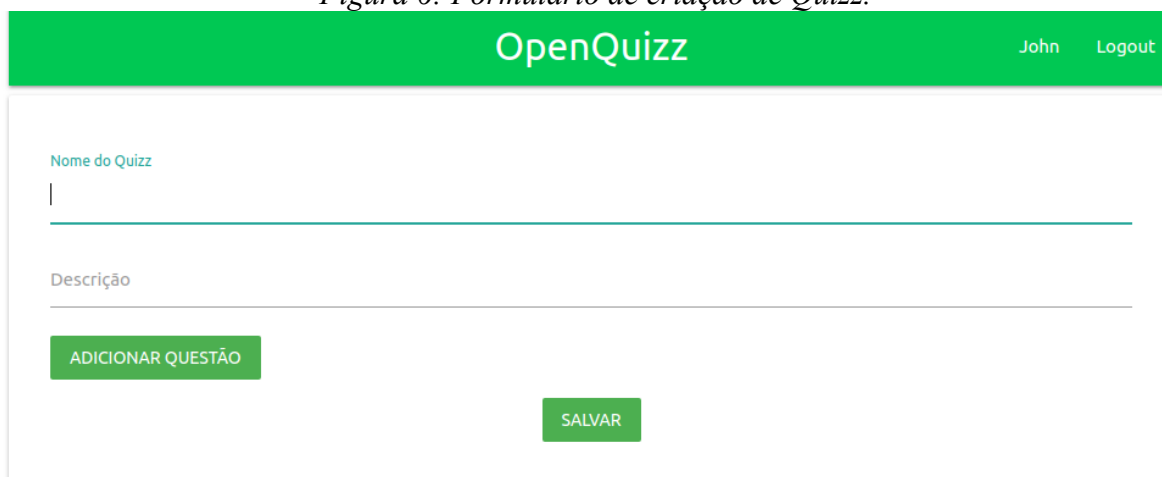


The screenshot shows the OpenQuizz user profile editing form. At the top, there is a green header with the text "OpenQuizz" and "John Logout" on the right. Below the header, the form contains three input fields: "First Name" with the value "John", "Last Name" with the value "Smith", and "E-mail" with the value "jsmith@openquizz.com". At the bottom of the form, there are two buttons: a red "VOLTAR" button on the left and a green "SALVAR" button on the right.

O formulário para editar perfil permite ao usuário alterar os dados de primeiro nome, sobrenome e e-mail (Figura 5).

#### 4.5 Criar quizz

*Figura 6: Formulário de criação de Quizz.*



The screenshot shows the OpenQuizz quiz creation form. At the top, there is a green header with the text "OpenQuizz" and "John Logout" on the right. Below the header, the form contains two input fields: "Nome do Quizz" and "Descrição". Below the "Nome do Quizz" field, there is a green button labeled "ADICIONAR QUESTÃO". At the bottom of the form, there is a green button labeled "SALVAR".

Através de botão para criar novo Quizz, o usuário tem acesso ao formulário para preencher o nome do Quizz, e a descrição do Quizz (Figura 6).

O botão para adicionar questão exibe um novo componente com os campos para acrescentar uma questão ao Quizz.


O botão Salvar salva e finaliza o processo de criação de Quizz, mostrando a tela com link para divulgação do mesmo (Figura 10).

#### 4.5.1 Criar questão

*Figura 7: Formulário de criação de questão.*

Nome do Quiz  
Quiz de História do Brasil

Descrição  
Teste seus conhecimentos em história brasileira.

**Questão #1** 

Enunciado  
|

Explicação

Adicionar alternativa...

**Resposta Correta**

ADICIONAR QUESTÃO


Ao clicar no botão Adicionar Questão, o usuário terá acesso ao formulário para criar uma nova questão (Figura 7).

O usuário deverá preencher os campos Enunciado e o campo Explicação da questão.

Para excluir uma questão, o usuário deve clicar no ícone de lixeira no canto superior direito do componente Questão.

#### 4.5.2 Criar alternativas

Figura 8: Formulário de criação de alternativas.



The image shows a web interface for creating a question. At the top, it says "Questão #1" with a red trash icon. Below that, the question text is "Em que ano aconteceu a proclamação da independência do Brasil?". The explanation is "Em 1822, Dom Pedro I proclamou o Brasil como um país independente.". There are three alternatives listed: "Alternativa #1" with the value "1808", "Alternativa #2" with "1822", and "Alternativa #3" with "1889". Each alternative has a trash icon to its right. At the bottom, there is a button labeled "Adicionar alternativa..." with a text input field below it.

Ao clicar no campo “Adicionar alternativa...” será criada uma alternativa para a questão atual (Figura 8).

O usuário poderá adicionar quantas alternativas desejar para cada questão.

Para excluir uma alternativa, o usuário deverá clicar no ícone de lixeira no canto superior esquerdo da alternativa que deseja apagar.



#### 4.5.3 Escolha da alternativa correta

Figura 9: Formulário de seleção de alternativa correta da questão.

O formulário apresenta quatro alternativas de resposta, cada uma em um campo de texto com um ícone de lixeira para exclusão:

- Alternativa #2: 1822
- Alternativa #3: 1889
- Alternativa #4: 1824

Abaixo das alternativas, há um campo para adicionar uma nova alternativa, rotulado "Adicionar alternativa...".

Na seção "Resposta Correta", há uma caixa de seleção com as seguintes opções:

- Alternativa #1
- Alternativa #2 (selecionada)
- Alternativa #3
- Alternativa #4

Depois de adicionadas todas as alternativas, o usuário deverá utilizar a caixa de seleção para indicar qual alternativa é a alternativa correta da questão (Figura 9).

#### 4.5.4 Quizz salvo com sucesso

Figura 10: Tela de confirmação de criação de Quizz.

A tela de confirmação de criação de Quizz apresenta o seguinte conteúdo:

- Logo "OpenQuizz" e o nome de usuário "John" com o link "Logout".
- Texto principal: "Quizz salvo com sucesso!".
- Subtítulo: "Quiz de História do Brasil".
- Texto de instrução: "Utilize o link abaixo para compartilhar seu quizz".
- Link de compartilhamento: <http://localhost:3000/takequizz/5b1aa93ddb9df2baee4ff09>

Ao salvar a Questão, será exibido ao usuário a informação de que o Quizz foi salvo com sucesso (Figura 10).

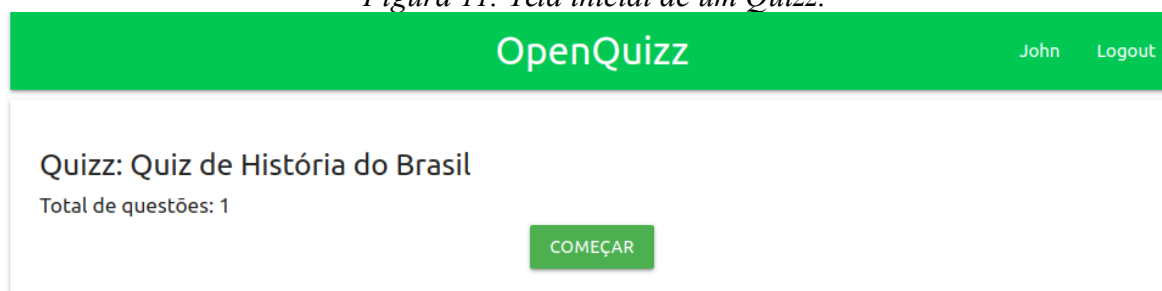
Também nessa tela, o usuário terá acesso ao link para divulgação do Quizz. Este link estará disponível posteriormente na tela principal, na lista de Quizzes criados pelo usuário (Figura 3).

#### 4.6 Realizar Quizz

Para realizar um Quizz, o usuário deve possuir o link que lhe dará acesso à página inicial para começar a responder.

##### 4.6.1 Tela de começo

*Figura 11: Tela inicial de um Quizz.*

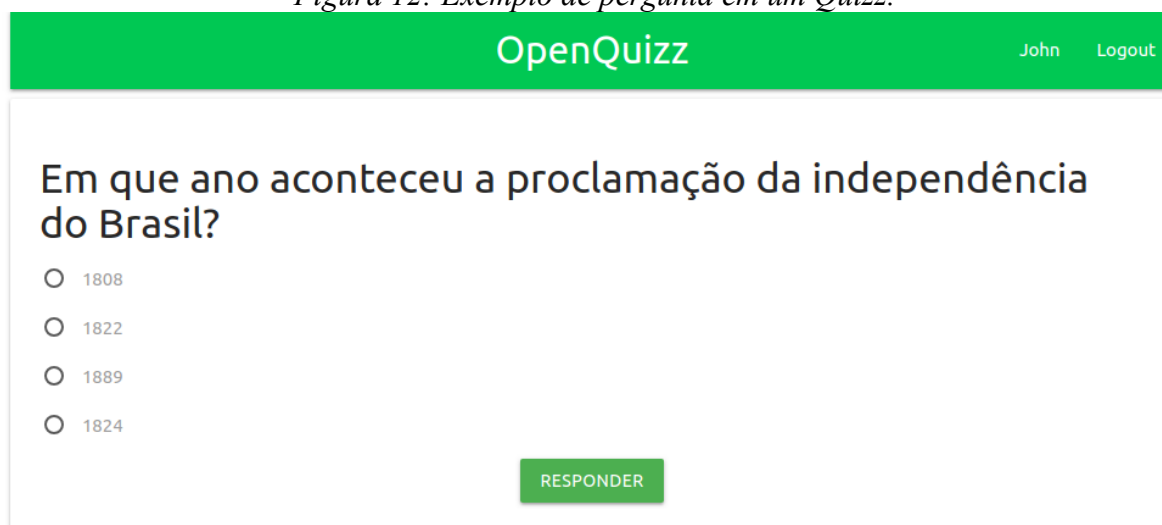


A tela inicial do Quizz contém o nome do Quizz que irá ser respondido, o total de questões e um botão para iniciar a realização do teste (Figura 11).

A qualquer momento que um usuário atualizar a página do navegador durante um Quizz, ele será redirecionado de volta a essa tela e o trabalho feito será perdido.

#### 4.6.2 Questão

Figura 12: Exemplo de pergunta em um Quiz.



The screenshot shows the OpenQuiz application interface. At the top, there is a green header with the text "OpenQuiz" and user information "John Logout". The main content area is white and contains the question: "Em que ano aconteceu a proclamação da independência do Brasil?". Below the question are four radio button options: "1808", "1822", "1889", and "1824". At the bottom center of the question area is a green button labeled "RESPONDER".

Ao começar um Quiz, o usuário terá acesso a uma questão do Quiz de cada vez. Cada questão conterá o enunciado, as alternativas para que o usuário escolha ao menos uma e o botão para responder à pergunta (Figura 12).

Caso a resposta esteja certa, será exibida a tela de resposta certa (Figura 13). Se a resposta estiver errada, será exibida a tela de resposta errada (Figura 14).

#### 4.6.3 Resposta correta

Figura 13: Tela de resposta correta.



The screenshot shows the OpenQuiz application interface displaying the correct answer. The header is green with "OpenQuiz" and "John Logout". The main content area has a light green background. It starts with a green checkmark icon and the text "Resposta certa!". Below this is the question: "Em que ano aconteceu a proclamação da independência do Brasil?". The answer is displayed as "Resposta: 1822". Below the answer is a white box containing the explanation: "Explicação: Em 1822, Dom Pedro I proclamou o Brasil como um país independente.". At the bottom center is a green button labeled "CONTINUAR".

Caso o usuário responda a pergunta corretamente, será exibida a tela contendo a informação “Resposta certa!”, seguida do enunciado da questão, a resposta e a explicação da questão atual (Figura 13).

O botão Continuar levará à próxima pergunta ou, no caso de não haver próxima pergunta, à tela de resultados do Quiz feito.

#### 4.6.4 Resposta errada

*Figura 14: Tela de resposta errada.*

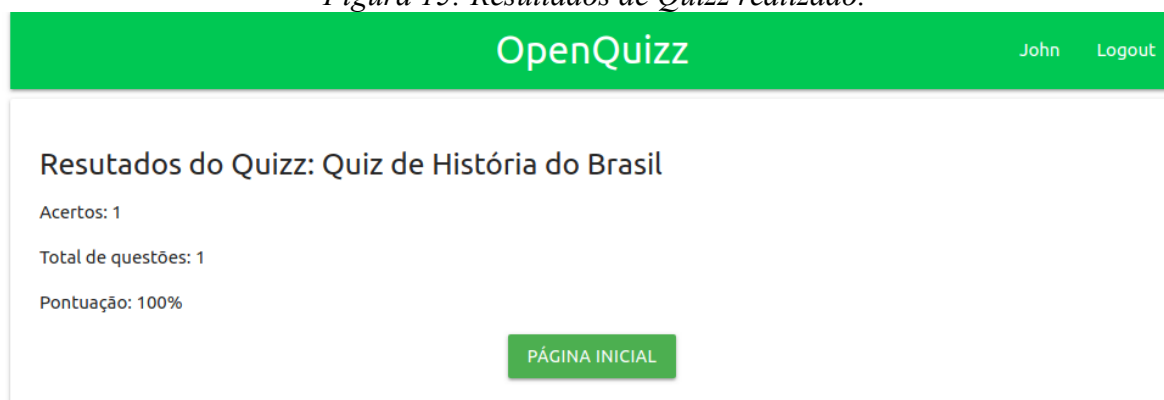


Caso o usuário não responda a pergunta corretamente, será exibida a tela contendo a informação “Resposta errada!”, seguida do enunciado da questão, a resposta correta, a resposta dada pelo usuário e a explicação da questão atual (Figura 14).

O botão Continuar levará à próxima pergunta ou, no caso de não haver próxima pergunta, à tela de resultados do Quiz feito.

#### 4.6.5 Resultados do Quiz

*Figura 15: Resultados de Quiz realizado.*



Após responder todas as questões, serão exibidos os resultados do Quiz realizado, contendo o número total de acertos, o total de questões respondidas e a pontuação em percentual do usuário (Figura 15).

Ao clicar no botão Página Inicial, o usuário será redirecionado à tela principal do aplicativo.

## 5. APLICABILIDADE

A aplicação foi desenvolvida a partir dos requisitos e restrições definidos com a existência de área para inscrição e autenticação de usuário. Uma vez cadastrado, o usuário poderá tanto criar novos testes como responder ao já criados, desde que tenha o link de acesso ao quiz.

A área de criação de quiz permite ao usuário flexibilidade na estruturação do teste, uma vez que o usuário pode criar quantas alternativas desejar para cada pergunta. Uma vez criado o teste, o usuário pode compartilhá-lo através de um link único para cada teste.

Ao responder um teste, o usuário tem o *feedback* a cada resposta dada sobre seu acerto ou erro e uma breve explicação sobre a razão da resposta certa.

O software foi desenvolvido para permitir a implementação e escalabilidade em diversos níveis de infraestrutura. Uma vez instaladas as dependências através do NPM, o software não necessita de internet para funcionar.

Utilizando o conceito *Single Page App* e com interface intuitiva e amigável o software pode ser aplicado em atividades pedagógicas das mais diversas áreas e integrado às mais variadas metodologias de ensino.

## REFERÊNCIAS

EXPRESS Project. Express documentation. Disponível em: <https://github.com/expressjs/express/> Acesso em: 6, Jun. 2018.

FACEBOOK Inc. React.js: Hello World. Disponível em: <https://reactjs.org/docs/hello-world.html> Acesso em: 6, Jun. 2018.

HANSON, J. Passport Documentation. Disponível em: <http://www.passportjs.org/docs/> Acesso em 6, Jun. 2018.

MONGODB Inc. MongoDB Manual. Disponível em: <https://docs.mongodb.com/manual/> Acesso em 6, Jun. 2018.

MONGOOSE. Mongoose documentation. Disponível em: <http://mongoosejs.com/docs/> Acesso em 6, Jun. 2018.

NODE.JS Foundation. About Node.js Foundation. Disponível em: <https://foundation.nodejs.org/about> Acesso em: 6, Jun. 2018.

NPM. The npm documentation. Disponível em: <https://docs.npmjs.com/> Acesso em: 6, Jun. 2018.

RASMUSSEN, E. Getting started with redux-form. Disponível em: <https://redux-form.com/7.3.0/docs/gettingstarted.md/> Acesso em 6, Jun. 2018.

REDUX. Redux.js: Read Me. Disponível em: <https://redux.js.org/> Acesso em 6, Jun. 2018.

SKINNER, B. F. Teaching Machines. **Science**, Washington, v. 128, n. 3330, p. 969-977, Out, 1958. Disponível em: <https://www.jstor.org/stable/1755240> Acesso em: 7, Jun. 2018.