

Desenvolvimento de Gerador de Questionários - QuestioData

João Vitor Rodrigues Pessoa

2023

Resumo

Este trabalho de conclusão de curso apresenta um estudo sobre o desenvolvimento e uso de questionários online como ferramenta de coleta de dados. São discutidos os passos fundamentais para a construção de uma ferramenta geradora de questionários online. Além disso, é apresentado a construção de uma ferramenta utilizando tecnologias como Javascript, React e Node.js, com o objetivo de facilitar a criação de questionários online. A ferramenta desenvolvida permite que o usuário crie questionários com diferentes tipos de perguntas e opções de resposta.

Palavras chave: questionários online, coleta de dados, pesquisa, Javascript, React, Node.js.

Abstract

This undergraduate thesis presents a study on the development and use of online questionnaires as a data collection tool. The fundamental steps for building an online questionnaire generation tool are discussed. Additionally, the construction of a tool using technologies such as Javascript, React, and Node.js is presented, aiming to facilitate the creation of online questionnaires. The developed tool enables users to create customized questionnaires with various types of questions and response options.

Keywords: online questionnaires, data collection, research, Javascript, React, Node.js.

1 Introdução

Os questionários são uma das principais ferramentas de coleta de dados utilizadas em pesquisas e estudos de diversas áreas. Eles permitem coletar informações de forma padronizada, o que facilita a análise e interpretação dos dados obtidos. Porém, a criação de questionários pode ser um processo trabalhoso e demorado.

Com o objetivo de facilitar esse processo, surgiram os geradores de questionários. Essas ferramentas automatizam a criação de questionários, permitindo que o usuário selecione as perguntas desejadas e defina as opções de resposta. Alguns geradores também oferecem recursos como a aleatorização das perguntas e a geração de gráficos e relatórios.

Existem diversas opções de geradores de questionários disponíveis atualmente, tanto gratuitas quanto pagas. No entanto, muitas delas apresentam limitações em termos de recursos e personalização para planos gratuitos, o que pode limitar a sua utilidade em determinados contextos.

Diante desse cenário, justifica-se entender e avaliar os passos para o desenvolvimento de uma ferramenta geradora de questionários, visando explorar essas possíveis limitações na busca de desenvolver uma ferramenta que possa se adaptar (conforme futuras melhorias) às necessidades específicas de cada pesquisa ou estudo.

Neste trabalho de conclusão de curso, é apresentado o processo de desenvolvimento de uma ferramenta geradora de questionários, destacando suas funcionalidades, características e possíveis melhorias em relação às opções disponíveis atualmente. São utilizadas técnicas de desenvolvimento de software e metodologias ágeis para garantir a qualidade e eficiência do projeto.

2 Análise das principais ferramentas de questionários

A coleta de dados por meio de questionários online se tornou uma prática comum em diversas áreas de estudo e em empresas que buscam obter feedback de seus clientes. Existem diversas ferramentas de questionários disponíveis no mercado, mas aqui são analisados os planos **gratuitos** das ferramentas: Google Forms, SurveyMonkey e Typeform.

2.1 Google Forms

O Google Forms¹ é uma ferramenta de questionários gratuita desenvolvida pelo Google. O Forms é fácil de usar e oferece diversos recursos de personalização, como a inserção de imagens e vídeos nas perguntas, além de diferentes tipos de perguntas, como múltipla escolha, seleção de imagem, entre outros.

Destaca-se do gerador de questionários Google Forms os seguintes pontos fortes:

- Gratuito;
- Fácil de usar;
- Diversos recursos de personalização;
- Integração com outras ferramentas do Google, como o Google Sheets.

Destaca-se do gerador de questionários Google Forms os seguintes pontos fracos:

- Limitação na quantidade de respostas recebidas por questionário;
- Personalização limitada em comparação com outras ferramentas.

A figura 1 mostra a página de edição do questionário no Google Forms, exibindo alguns exemplos de tipos de campos possíveis para essa ferramenta:

2.2 SurveyMonkey

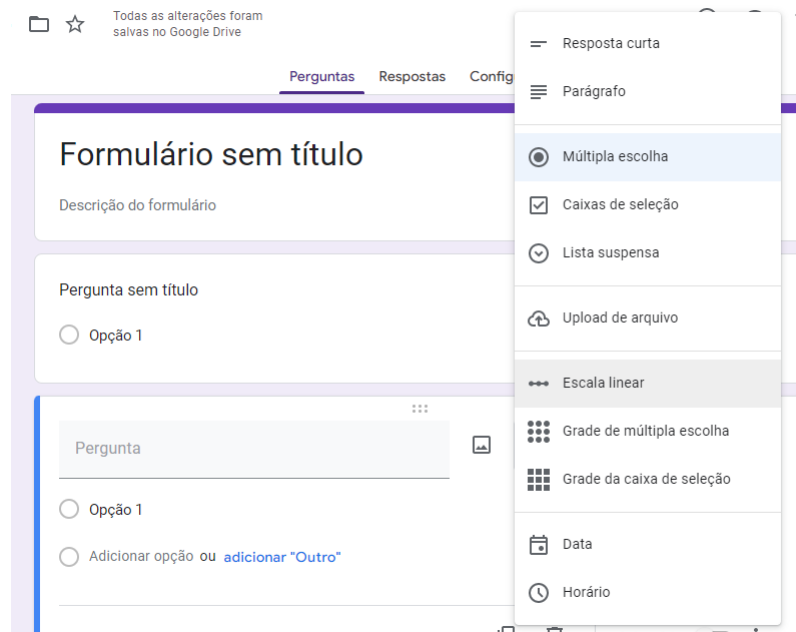
O SurveyMonkey² é uma ferramenta de questionários paga que oferece diversas opções de personalização e recursos avançados de análise de dados. É uma ferramenta amplamente utilizada em empresas e em pesquisas acadêmicas.

Destaca-se do gerador de questionários SurveyMonkey os seguintes pontos fortes:

- Diversas opções de personalização;
- Recursos avançados de análise de dados;
- Integração com outras ferramentas, como o Salesforce;

¹<https://www.google.com/intl/pt-BR/forms/about/>

²<https://pt.surveymonkey.com>



Fonte: <https://www.google.com/intl/pt-BR/forms>

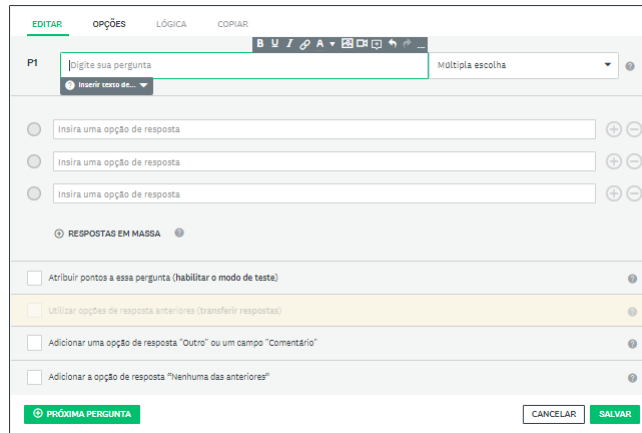
Figure 1: Layout Google Forms

- Suporte ao cliente 24 horas por dia.

Destaca-se do gerador de questionários SurveyMonkey os seguintes pontos fracos:

- Plano gratuito limitado;
- Interface não tão intuitiva;
- Limitação na quantidade de perguntas e respostas por questionário.

A figura 2 mostra a página de edição do questionário no SurveyMonkey, é possível notar a diferença de layout em relação à ferramenta mostrada anteriormente:



Fonte: <https://pt.surveymonkey.com>.

Figure 2: Layout SurveyMonkey

2.3 Typeform

O Typeform³ é uma ferramenta de questionários que oferece uma interface interativa e moderna, com diversos recursos de personalização e análise de dados. É amplamente utilizado em empresas de tecnologia e startups.

Destaca-se do gerador de questionários Typeform os seguintes pontos fortes:

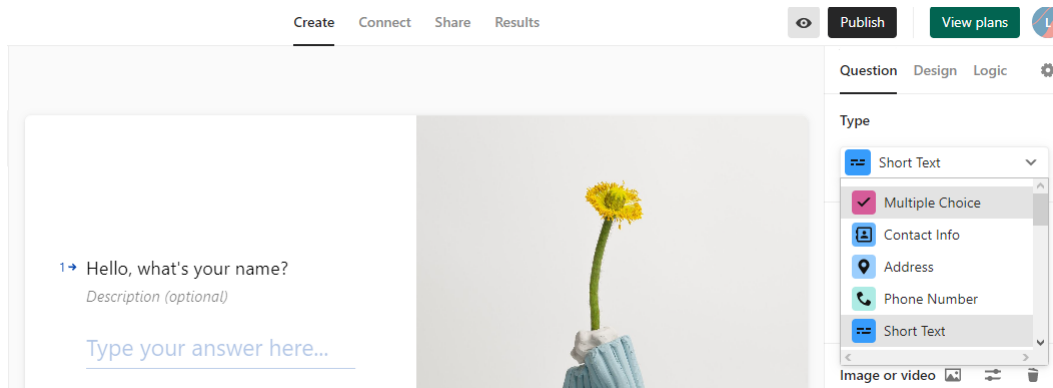
- Interface interativa e moderna;
- Diversas opções de personalização;
- Recursos avançados de análise de dados;
- Possibilidade de integração com outras ferramentas.

Destaca-se do gerador de questionários Typeform os seguintes pontos fracos:

- Recursos avançados apenas nos planos pagos;
- Preço mais elevado que outras ferramentas.

A figura 3 mostra a página de edição do questionário no Typeform, enfatizando sua aparência moderna e simplista:

³<https://www.typeform.com>



Fonte: <https://www.typeform.com>.

Figure 3: Layout Typeform

A tabela abaixo busca mostrar algumas das funcionalidades existentes nos geradores de questionários analisados anteriormente com algumas funcionalidades que o QuestioData (versão inicial) tem como objetivo.

Table 1: Comparativo de Geradores de Questionários

Funcionalidades	Google Forms	SurveyMonkey	Typeform	QuestioData
Criação de questionários	Sim	Sim	Sim	Sim
Tipos de perguntas suportados	Diversos	Diversos	Diversos	Básicos
Design personalizável	Sim	Sim	Sim	Não
Lógica condicional	Sim	Sim	Sim	Não
Coleta de respostas anônimas	Sim	Sim	Sim	Sim
Compartilhamento de questionários	Sim	Sim	Sim	Sim
Preço	Gratuito e pago	Gratuito e pago	Gratuito e pago	Gratuito

Fonte: Desenvolvido pelo Autor.

3 Tecnologias Utilizadas

Para o desenvolvimento do sistema gerador de questionários, são utilizadas diversas tecnologias que se complementam para oferecer uma solução eficiente e escalável.

HTML⁴, sigla para HyperText Markup Language (Linguagem de Marcação de Hipertexto), é uma linguagem de marcação utilizada na construção de

⁴<https://www.w3schools.com/html/>

páginas web. Desenvolvido por Tim Berners-Lee no início da década de 1990, o HTML é a base fundamental para a criação e estruturação do conteúdo de um website.

No desenvolvimento web, o HTML desempenha um papel crucial na apresentação e organização do conteúdo. Ele define a estrutura lógica de uma página, permitindo a criação de parágrafos, títulos, listas, tabelas, links e outros elementos que compõem a interface de um site.

A sintaxe do HTML é baseada em tags, que são elementos delimitados por colchetes angulares ("`<`" e "`>`") e que envolvem o conteúdo a ser exibido. Cada tag tem uma função específica e determina como o conteúdo dentro dela deve ser interpretado pelo navegador(SILVA, 2008).

CSS⁵ sigla para Cascading Style Sheets (Folhas de Estilo em Cascata), é uma linguagem de estilo utilizada no desenvolvimento web para controlar a apresentação visual e o layout dos elementos em uma página HTML. O CSS permite definir cores, fontes, tamanhos, posicionamento, espaçamento e outros aspectos visuais dos elementos de uma página.

Ao separar o conteúdo (HTML) da apresentação (CSS), o CSS proporciona uma maior flexibilidade e controle sobre o design de um site. Ele permite criar estilos consistentes e padronizados em todo o site, facilitando a manutenção e a atualização do layout.

CSS funciona através de regras, que são aplicadas aos elementos HTML usando seletores. Cada regra consiste em um seletor e um bloco de declarações. O seletor define quais elementos serão afetados pela regra e as declarações especificam como esses elementos devem ser estilizados. Por exemplo, a regra "`p color: blue;`" define que todos os parágrafos (`<p>`) devem ter a cor azul.

Existem diversas formas de aplicar estilos CSS a uma página HTML. O CSS pode ser inserido diretamente no documento HTML usando a tag "`<style>`", incorporado externamente através de um arquivo CSS separado ou até mesmo aplicado inline diretamente nos elementos HTML usando o atributo "`style`". A abordagem mais recomendada é utilizar um arquivo CSS externo, pois isso permite a separação completa do estilo da estrutura do documento, facilitando a manutenção e reutilização de estilos em várias páginas.

Além dos estilos básicos, o CSS também oferece recursos avançados, como seletores de classe e ID, pseudo-classes, animações e transições, entre outros. Isso permite criar efeitos visuais sofisticados, como menus suspensos,

⁵<https://developer.mozilla.org/pt-BR/docs/Web/CSS>

transições suaves, imagens de fundo, layouts responsivos entre outras funcionalidades.

O Bulma⁶ é um framework CSS responsivo⁷ de código aberto que fornece um conjunto de estilos predefinidos e componentes reutilizáveis para o desenvolvimento web. Ele é construído com base no conceito de design responsivo, o que significa que os sites criados com o Bulma se adaptam automaticamente a diferentes tamanhos de tela, proporcionando uma experiência consistente em dispositivos móveis, tablets e desktops.

O Bulma utiliza uma grade flexível como base para o layout, o que facilita a criação de designs responsivos e organizados. Ele oferece uma ampla gama de classes CSS prontas para uso que podem ser aplicadas a elementos HTML para definir estilos e estruturas. Além disso, o Bulma fornece componentes reutilizáveis, como botões, formulários, barras de navegação, cartões, modais, entre outros, que podem ser facilmente personalizados e implementados em um site (SHENOY, 2019).

A linguagem de programação escolhida foi o Javascript⁸, que é uma das linguagens mais utilizadas no desenvolvimento web (FLANAGAN, 2004) e oferece diversas bibliotecas e frameworks para o desenvolvimento de aplicações web. O uso de Javascript também permite que a aplicação seja executada em diversos navegadores sem a necessidade de plugins ou software adicional.

O banco de dados utilizado é o MySQL⁹, segundo o ranking do site DB-Engines¹⁰, o MySQL é um dos bancos de dados mais populares e amplamente utilizados. O MySQL é um banco de dados relacional, que oferece uma excelente performance, segurança e escalabilidade. Além disso, o MySQL é compatível com diversas tecnologias e oferece diversas opções de configuração e personalização. (MILANI, 2007)

Rank Maio 2023	Database	Modelo
1	Oracle	Relacional
2	MySQL	Relacional
3	Microsoft SQL Server	Relacional

Desenvolvido pelo Autor.

⁶<https://bulma.io/documentation/>

⁷pode ser utilizado em diferentes dispositivos

⁸<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>

⁹<https://www.mysql.com>

¹⁰<https://db-engines.com/en/ranking>

O conjunto de tecnologias escolhidas para o desenvolvimento do sistema gerador de questionários objetiva a criação de uma aplicação moderna, eficiente e escalável, que atenda às necessidades dos usuários e ofereça uma boa experiência de uso.

4 Bibliotecas, Frameworks e Ferramentas de Suporte

Parte importante no desenvolvimento do QuestioData além dos *frameworks*, são as bibliotecas utilizadas que possibilitam um melhor funcionamento da aplicação.

4.1 React

O React¹¹ foi escolhido como o framework de front-end para a construção do sistema gerador de questionários. O React é uma biblioteca Javascript desenvolvida pelo Facebook (FEDOSEJEV, 2015), que permite a criação de interfaces de usuário de forma declarativa, componentizada e eficiente.¹² Além disso, o React é compatível com outras tecnologias, o que o torna uma opção para o desenvolvimento de aplicações complexas.

4.2 Node.js

O Node.js¹³ foi utilizado como a plataforma de backend para a aplicação. O Node.js é uma plataforma de desenvolvimento de aplicações em Javascript, que permite a execução de código no lado do servidor. O uso de Node.js torna possível a criação de aplicações escaláveis e de alta performance (PEREIRA, 2014).

4.3 Express.js

O Express.js¹⁴ é um framework web rápido, flexível e minimalista para Node.js, que permite construir aplicativos web robustos e escaláveis. Ele fornece uma

¹¹<https://react.dev>

¹²<https://blog.xpeducacao.com.br/react-javascript/>

¹³<https://nodejs.org/en/about>

¹⁴<https://expressjs.com>

camada de abstração sobre o Node.js, simplificando o desenvolvimento de aplicativos web ao oferecer uma ampla gama de recursos e utilitários(PETERS, 2017).

O Express.js é conhecido por sua simplicidade e facilidade de uso. Ele possui uma API minimalista que permite criar facilmente rotas, lidar com solicitações HTTP, gerenciar sessões, configurar *middlewares*, entre outras funcionalidades. Com o Express.js, os desenvolvedores podem criar aplicativos web completos com poucas linhas de código.

Uma das principais características do Express.js é seu sistema de roteamento flexível. Ele permite definir rotas para diferentes URLs e métodos HTTP, como GET, POST, PUT e DELETE, e associar manipuladores de rota para lidar com essas solicitações. Além disso, o Express.js suporta a criação de rotas parametrizadas, o que permite capturar valores dinâmicos nas URLs e usá-los no processamento da solicitação.

4.4 Nodemon

Fazendo uso do Node.js, é importante ter uma experiência de desenvolvimento fluida e eficiente. Uma biblioteca que pode aprimorar esse processo é o Nodemon¹⁵. O Nodemon é uma ferramenta que monitora as alterações em seus arquivos de origem do Node.js e reinicia automaticamente o servidor sempre que uma modificação é detectada. Isso elimina a necessidade de reiniciar manualmente o servidor a cada alteração no código, economizando tempo e esforço dos desenvolvedores.

A utilização do Nodemon é simples. Basta instalá-lo globalmente ou localmente no projeto, dependendo de suas preferências. Em seguida, executa-se o aplicativo Node.js com o comando "nodemon" em vez de "node". O Nodemon iniciará o servidor e monitorará os arquivos especificados, reiniciando-o automaticamente sempre que houver alterações.

4.5 Sequelize

A biblioteca Sequelize¹⁶ é uma poderosa ORM (Object-Relational Mapping) para Node.js, que oferece uma camada de abstração para a interação com bancos de dados relacionais(BARSOTI; GIBERTONI, 2020). Ela permite

¹⁵<https://www.npmjs.com/package/nodemon>

¹⁶<https://sequelize.org>

```
TERMINAL  CONSOLE DE DEPURACÃO  PROBLEMAS  SAÍDA
o vitor@Vr:~/TCC/backend$ nodemon index.js
[nodemon] 2.0.16
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index.js`
```

Desenvolvido pelo autor

Figure 4: Exemplo de uso do nodemon

aos desenvolvedores escrever consultas SQL de forma mais intuitiva e simplificada, utilizando a sintaxe do JavaScript.

Com o Sequelize, é possível criar, consultar, atualizar e excluir registros em um banco de dados relacional sem a necessidade de escrever consultas SQL manualmente. A biblioteca suporta vários sistemas de gerenciamento de banco de dados, tornando-a altamente flexível e adequada para uma ampla gama de projetos.

```
backend > config > JS database.js > [?] default
1  import { Sequelize } from "sequelize";
2
3  const db = new Sequelize('tcc_db', 'root', 'senha123', {
4    host: "localhost",
5    dialect: "mysql"
6  });
7
8  export default db;
```

Desenvolvido pelo autor

Figure 5: Exemplo de uso do sequelize

4.6 React Router DOM

O React Router DOM¹⁷ é uma biblioteca do React.js. Ela permite a criação de rotas dinâmicas e a navegação entre diferentes componentes de uma aplicação. Com o React Router DOM, é possível construir aplicações de página única (*Single-Page Applications - SPAs*) que oferecem uma experiência de usuário fluida, sem a necessidade de recarregar a página inteira a cada transição.

¹⁷<https://reactrouter.com/en/6.12.1/start/tutorial>

Uma das principais vantagens do React Router DOM é a sua simplicidade de uso. Com apenas algumas linhas de código, é possível definir rotas e vinculá-las a componentes específicos. Essas rotas podem ser configuradas para corresponder a determinados URLs e acionar a renderização do componente correspondente. Além disso, é possível passar parâmetros e até mesmo acessar informações da URL para personalizar o comportamento dos componentes.

Outro benefício importante do React Router DOM é a sua capacidade de lidar com a navegação do histórico do navegador. Ele mantém um histórico de navegação, permitindo que os usuários utilizem os botões "Voltar" e "Avançar" do navegador para navegar entre as diferentes rotas da aplicação.

Além disso, o React Router DOM oferece recursos avançados, como redirecionamentos, proteção de rotas autenticadas e até mesmo a possibilidade de renderizar rotas aninhadas. Esses recursos são especialmente úteis em aplicações maiores e mais complexas, onde é necessário controlar o acesso a determinadas rotas com base na autenticação do usuário ou na lógica de negócios da aplicação.(GANATRA, 2018)

5 Desenvolvimento

O desenvolvimento de um gerador de questionários é um projeto que envolve diversas ferramentas e tecnologias para que seja efetivo e eficiente. A seguir explicado o uso de cada uma das seguintes ferramentas: HTML, CSS, Javascript, mysql, Node.js, express.js, React e Bulma.

A primeira etapa para o desenvolvimento do gerador de formulários foi a criação da interface gráfica, que foi desenvolvida com a utilização de HTML e CSS. O HTML¹⁸ é a linguagem responsável pela estruturação do conteúdo do site e o CSS¹⁹ é a linguagem responsável pela estilização e formatação da página. Essas duas ferramentas são essenciais para a criação de um layout atraente e funcional para os usuários.

A próxima etapa é a implementação da lógica do sistema, que foi feita por meio de Javascript. Esta linguagem é utilizada tanto para validação dos dados inseridos pelos usuários, quanto para a interatividade da página, como a adição dinâmica de campos de formulário, por exemplo.

¹⁸<https://developer.mozilla.org/pt-BR/docs/Web/HTML>

¹⁹<https://developer.mozilla.org/pt-BR/docs/Web/CSS>

Para armazenar as informações dos formulários criados pelos usuários, foi utilizado o MySQL, um sistema de gerenciamento de banco de dados relacional, que permite o armazenamento e a manipulação dos dados de forma organizada e eficiente.

Para a criação da aplicação, foi utilizado o Node.js, um ambiente de execução de código Javascript do lado do servidor. O Node.js permitiu que a aplicação fosse executada diretamente no servidor, permitindo uma maior escalabilidade e flexibilidade.

Para a criação das rotas da aplicação, foi utilizado o Express.js²⁰, um framework para NodeJs que facilita a criação de aplicações web, definindo rotas, middlewares e integração com o banco de dados.

Para a criação da interface gráfica da aplicação, foi utilizado o React, uma biblioteca JavaScript para a criação de interfaces de usuário interativas. O React permitiu a criação de componentes reutilizáveis, o que tornou o desenvolvimento da interface gráfica mais rápido e eficiente.

Por fim, foi utilizado o Bulma²¹, um framework CSS que permitiu a criação de interfaces responsivas e personalizáveis, facilitando a criação de uma interface atraente e amigável para o usuário.

Em resumo, para o desenvolvimento de um gerador de formulários, foi utilizado diversas ferramentas e tecnologias, como HTML, CSS, Javascript, MySQL, Node.js, Express.js, React e Bulma. Essas ferramentas foram essenciais para a criação de uma aplicação web efetiva e eficiente, desde a interface gráfica até a manipulação de dados no banco de dados.

5.1 Modelagem do Banco de Dados

A modelagem de banco de dados é essencial para garantir a estrutura correta e as relações entre os dados envolvidos.

5.1.1 Tabelas

A modelagem inclui as seguintes tabelas:

- **usuario:** Esta tabela armazena informações sobre os usuários que podem utilizar o gerador de questionários. Ela possui um campo "id"

²⁰<https://expressjs.com/pt-br/>

²¹<https://bulma.io>

como chave primária e auto incrementável, além das colunas "nome" e "email" para armazenar o nome e o e-mail do usuário, respectivamente.

- **form:** A tabela "form" é responsável por armazenar os formulários de questionários. Ela possui um campo "id" como chave primária e auto incrementável, e também possui as colunas "nome" e "descricao" para armazenar o nome e a descrição do formulário.
- **usuario_form:** Essa tabela estabelece a relação entre os usuários e os formulários. Ela possui as colunas "usuarioid" e "formularioid" como chaves estrangeiras referenciando os campos "id" das tabelas "usuario" e "form", respectivamente. Além disso, ela possui a coluna "respondido" que indica se o usuário respondeu ou não o formulário.
- **pergunta:** A tabela "pergunta" armazena as perguntas que compõem os questionários. Ela possui um campo "id" como chave primária e auto incrementável, além da coluna "formularioid" como chave estrangeira referenciando o campo "id" da tabela "form", e a coluna "pergunta" para armazenar o texto da pergunta.
- **resposta:** Essa tabela é responsável por armazenar as respostas dos usuários. Ela possui as colunas "usuarioid", "formularioid" e "perguntaid" como chaves estrangeiras referenciando os campos correspondentes nas tabelas "usuario_form" e "pergunta". Além disso, ela possui a coluna "resposta" para armazenar o texto da resposta dada pelo usuário.

5.1.2 Relações

As tabelas estão relacionadas da seguinte forma:

- A tabela "usuario" tem uma relação um-para-muitos com a tabela "usuario_form", já que um usuário pode ter vários formulários associados a ele.
- A tabela "form" também possui uma relação um-para-muitos com a tabela "usuario_form", pois um formulário pode estar associado a vários usuários.
- A tabela "form" tem uma relação um-para-muitos com a tabela "pergunta", uma vez que um formulário pode conter várias perguntas.

- A tabela "pergunta" possui uma relação um-para-muitos com a tabela "resposta", pois uma pergunta pode ter várias respostas.
- A tabela "usuario_form" possui chaves estrangeiras que referenciam as tabelas "usuario" e "form", estabelecendo a relação entre usuários e formulários.
- A tabela "resposta" tem chaves estrangeiras que referenciam as tabelas "usuario_form" e "pergunta", estabelecendo a relação entre respostas, usuários e perguntas.

Essa modelagem de banco de dados permite armazenar de forma organizada e relacionada todas as informações necessárias para criar e gerenciar questionários, incluindo usuários, formulários, perguntas e respostas.

```

1  -- Criação da tabela "usuario"
2  • CREATE TABLE usuario (
3      id INT AUTO_INCREMENT PRIMARY KEY,
4      nome VARCHAR(255),
5      email VARCHAR(255)
6  );
7
8  -- Criação da tabela "form"
9  • CREATE TABLE form (
10     id INT AUTO_INCREMENT PRIMARY KEY,
11     nome VARCHAR(255),
12     descricao VARCHAR(255)
13 );
14

```

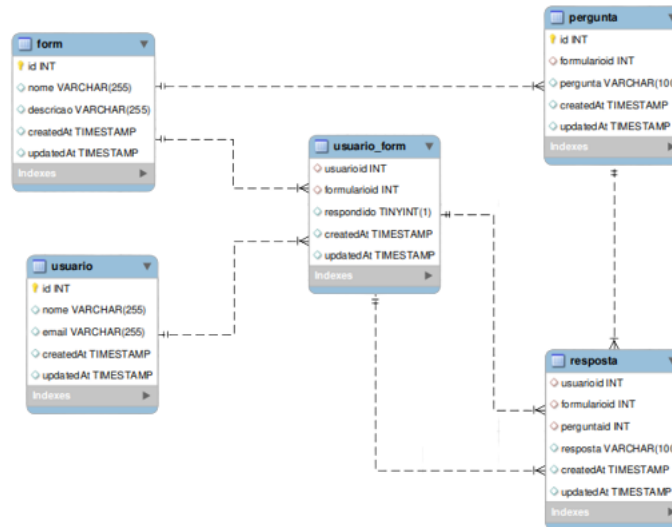
Fonte: Elaborada pelo Autor.

Figure 6: Script de construção das tabelas.

5.2 Criação das Rotas

A criação de rotas é uma parte essencial para definir o comportamento e a lógica da aplicação. Para o gerador de formulários, foi definidas rotas que lidem com as diferentes etapas do processo de criação e manipulação de formulários, essas etapas vão desde a criação do formulário até a resposta final do usuário que é armazenada no banco de dados.

A figura 8 ilustra os caminhos de rota para alguns dos componentes criados para o QuestioData.



Fonte: Elaborada pelo Autor.

Figure 7: Diagrama de Relacionamento do Banco.

```
function App() {
  return (
    <Router>
      <div className="container">
        <div className="columns">
          <div className="column is-half is-offset-one-quarter">
            <Routes>
              <Route exact path="/" element={<CampoList/>}>
            </Route>
              <Route path="/add" element={<AddCampo/>}>
            </Route>
              <Route path="/edit/:id" element={<EditCampo/>}>
            </Route>
              <Route path="/form" element={<VerificaCampo/>}>
            </Route>
            </Routes>
          </div>
        </div>
      </div>
    </Router>
  );
}
export default App;
```

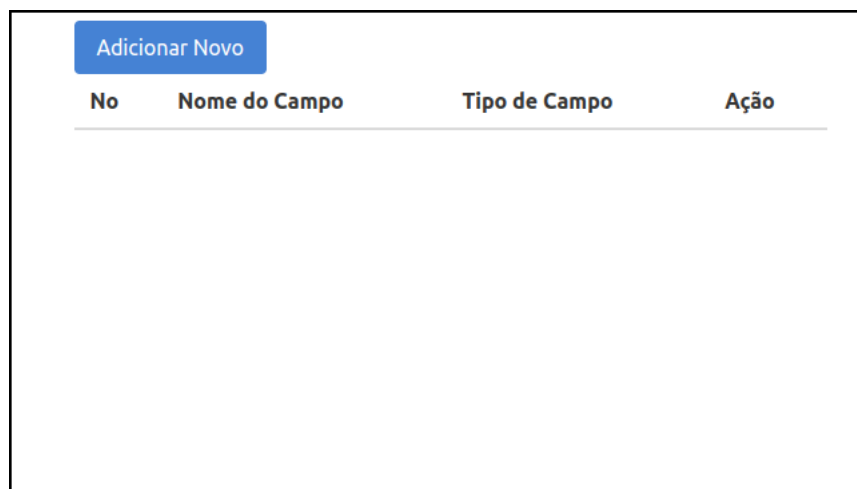
Fonte: Elaborada pelo Autor.

Figure 8: Script responsável pelas rotas da página.

5.3 Funcionamento

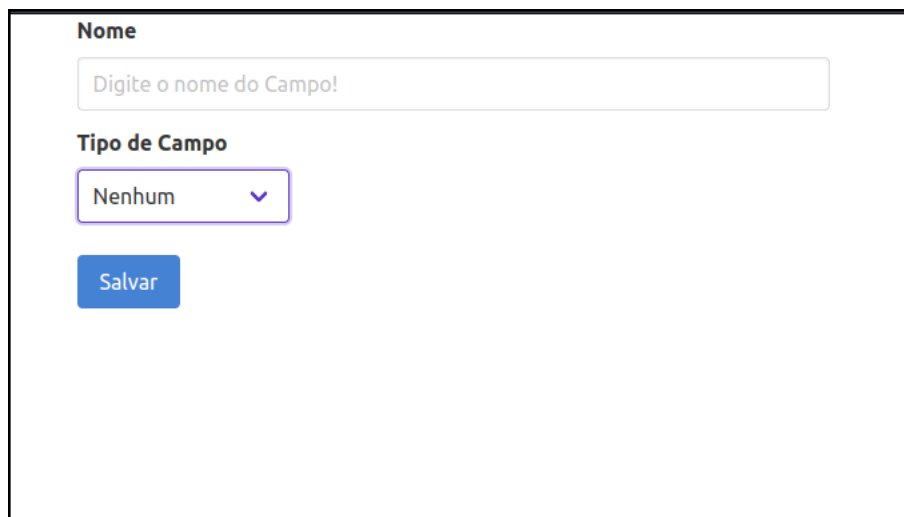
Na página inicial da aplicação é possível ver/gerenciar todos os campos do formulário nesta pagina é possível criar novos campos, editar os já existentes

ou excluí-los, conforme mostram as imagens abaixo:



Fonte: Elaborada pelo Autor.

Figure 9: Página Inicial.

A screenshot of a form for creating a new field. It features a text input field labeled "Nome" with the placeholder text "Digite o nome do Campo!". Below this is a dropdown menu labeled "Tipo de Campo" with the selected option "Nenhum" and a downward arrow. At the bottom of the form is a blue button labeled "Salvar".

Fonte: Elaborada pelo Autor.

Figure 10: Página de Criação dos Campos.

Após a criação de cada campo, a aplicação volta para a página de gerenciamento onde podemos ver, excluir ou editar os campos existentes.

Adicionar Novo

No	Nome do Campo	Tipo de Campo	Ação
1	Nome	Texto	<input type="button" value="Editar"/> <input type="button" value="Deletar"/>
2	Telefone	Texto	<input type="button" value="Editar"/> <input type="button" value="Deletar"/>
3	E-mail	Texto	<input type="button" value="Editar"/> <input type="button" value="Deletar"/>
4	Deseja receber atualizações via e-mail?	Checkbox	<input type="button" value="Editar"/> <input type="button" value="Deletar"/>
5	Sua Mensagem	Texto	<input type="button" value="Editar"/> <input type="button" value="Deletar"/>

Fonte: Elaborada pelo Autor.

Figure 11: Página Inicial com Campos.

Se houver necessidade de edição, o usuário poderá fazer através da seguinte tela que abrirá no botão de "Editar", será possível mudar o nome e o tipo do campo escolhido.

Nome

Deseja receber atualizações via e-mail?

Tipo de Campo

Checkbox ▼

Atualizar

Fonte: Elaborada pelo Autor.

Figure 12: Página de Edição de Campo.

No final do processo de criação dos campos, é possível ver como o questionário ficará antes de criá-lo.

The image shows a form for creating a questionnaire. It contains the following fields and labels:

- Nome**: Input field with placeholder text "Digite o Texto!"
- Telefone**: Input field with placeholder text "Digite o Texto!"
- E-mail**: Input field with placeholder text "Digite o Texto!"
- Deseja receber atualizações via e-mail?**: Input field with placeholder text "Digite o Texto!"
- Sua Mensagem**: Input field with placeholder text "Digite o Texto!"
- Criar Formulário**: A blue button at the bottom.

Fonte: Elaborada pelo Autor.

Figure 13: Exemplo de visualização de Questionário.

Após a confirmação de criação do questionário, é gerado um pop-up contendo o link para o questionário que poderá ser compartilhado com diversos usuários.

A figura 14 apresenta um pop-up gerado pelo sistema com uma tela e um botão de copiar link para acesso ao questionário gerado.

The image shows a dark-themed background with a light-colored pop-up window titled "Link do Questionário". The pop-up contains the following elements:

- Link:** A text input field containing the URL "https://www.exemplo.com/formulario/undefined".
- Copiar Link**: A button located at the bottom of the pop-up.

Fonte: Elaborada pelo Autor.

Figure 14: Tela de Compartilhamento de Link.

6 Restrições

O QuestioData por estar na versão inicial, ainda possui algumas restrições que podem ser atualizadas com versões futuras, algumas das restrições conhecidas são:

- **Validação de Campos:** O QuestioData possui restrições na validação dos campos, o que significa que nem todos os tipos de entradas de dados podem ser verificados ou validados adequadamente. Isso pode resultar em problemas como entradas inválidas ou inconsistentes nos questionários.
- **Variedade de Campos:** O sistema tem uma limitação em relação à variedade de campos que podem ser incluídos nos questionários. Isso significa que certos tipos de campos ou formatos podem não ser suportados ou aceitos pelo sistema, restringindo a flexibilidade na criação de questionários.
- **Personalização de Questionário com Imagens:** Atualmente, o QuestioData não oferece suporte total para personalização de questionários com imagens. Isso implica que recursos como a inclusão de imagens no conteúdo dos questionários podem estar ausentes ou ser limitados em sua funcionalidade.
- **Segurança dos Dados:** O QuestioData possui restrições em relação à segurança dos dados coletados. Embora sejam implementadas medidas básicas de segurança, pode haver lacunas em termos de proteção de dados sensíveis ou confidenciais. Isso pode representar riscos de privacidade ou vulnerabilidades em relação ao acesso ou uso indevido das informações coletadas.
- **Lei Geral de Proteção de Dados (LGPD):** O QuestioData não está totalmente em conformidade com a Lei Geral de Proteção de Dados (LGPD). A LGPD é uma legislação que define diretrizes e regras para o tratamento de dados pessoais no Brasil. Como o QuestioData ainda está em sua versão inicial, pode não cumprir totalmente todos os requisitos da LGPD, o que pode resultar em questões legais ou falta de conformidade com a legislação vigente.

7 Melhorias

Durante o desenvolvimento do QuestioData²², foram pensadas algumas melhorias (além do desenvolvimento de soluções para as restrições citadas anteriormente) para utilizar os dados obtidos com as respostas dos questionários e de alguma maneira oferecer algumas análises para empresas ou equipes como um diferencial da ferramenta, visando esse objetivo essas melhorias abaixo foram pensadas:

- Sistema de planos pagos para equipes e empresas;
- Personalização de questionários com identidade visual da empresa/cliente;
- Construção de questionários com base em templates predefinidos;
- Reaproveitamento de questionários criados anteriormente;
- Geração e envio por e-mail de arquivo .xlsx contendo respostas;
- Geração e envio de relatório PDF separado por tópicos de cada questionário gerado;
- Integração com ferramenta de automação para envio de relatório assim que o questionário gerado for respondido;
- Integração com Power BI²³ e geração de dashboard personalizado.

8 Conclusão

Este trabalho tem como objetivo apresentar o desenvolvimento de um gerador de questionários online, que pode servir como ferramenta de suporte para profissionais de diversas áreas. Por ser uma ferramenta online, o sistema poderá ser utilizado de qualquer lugar a qualquer momento, uma vez que o usuário possua acesso a internet, representando uma contribuição significativa para pesquisa e coleta de dados, podendo também ser aprimorada conforme melhorias descritas na seção 7 deste trabalho, trazendo novas *features* que as ferramentas concorrentes citadas anteriormente não possuem.

²²Disponível no Github em <https://github.com/VitorRodgs/TCC>

²³<https://learn.microsoft.com/pt-br/power-bi/fundamentals/power-bi-overview>

References

BARSONI, N.; GIBERTONI, D. Impacto que o sequelize traz para o desenvolvimento de uma api construída em node. js com express. js. *Revista Interface Tecnológica*, v. 17, n. 2, p. 231–243, 2020.

FEDOSEJEV, A. *React. js essentials*. [S.l.]: Packt Publishing Ltd, 2015.

FLANAGAN, D. *JavaScript: o guia definitivo*. [S.l.]: Bookman Editora, 2004.

GANATRA, S. *React Router Quick Start Guide: Routing in React Applications Made Easy*. [S.l.]: Packt Publishing Ltd, 2018.

MILANI, A. *MySQL-guia do programador*. [S.l.]: Novatec Editora, 2007.

PEREIRA, C. R. *Aplicações web real-time com Node. js*. [S.l.]: Editora Casa do Código, 2014.

PETERS, C. Building rich internet applications with node. js and express. js. *Rich Internet Applications w/HTML and Javascript*, p. 15, 2017.

SHENOY, A. *Learning Bulma: Understand How to Develop Responsive, Mobile-first Websites Using This Impressive, Modern Framework*. [S.l.]: Apress, 2019.

SILVA, M. S. *Criando sites com HTML: sites de alta qualidade com HTML e CSS*. [S.l.]: Novatec Editora, 2008.