



UNIVERSIDADE FEDERAL DO PARÁ
CAMPUS UNIVERSITÁRIO DO TOCANTINS/CAMETÁ
FACULDADE DE SISTEMAS DE INFORMAÇÃO

ILANNILDO VIANA DA CRUZ

DESENVOLVIMENTO DA PLATAFORMA WEB MONITORA TOCANTINS

Cametá-PA
2024

ILANNILDO VIANA DA CRUZ

DESENVOLVIMENTO DA PLATAFORMA WEB MONITORA TOCANTINS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Sistemas de Informação, Faculdade de Sistemas de Informação, Campus Universitário do Tocantins/Cametá, Universidade Federal do Pará, como requisito parcial à obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Ulisses Weyl da Cunha Costa.

**Cametá-PA
2024**

Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD
Sistema de Bibliotecas da Universidade Federal do Pará
Gerada automaticamente pelo módulo Ficat, mediante os dados fornecidos pelo(a) autor(a)

C955d Cruz, Ilannildo Viana da.
DESENVOLVIMENTO DA PLATAFORMA WEB
MONITORA TOCANTINS / Ilannildo Viana da Cruz. — 2024.
XLIV, 44 f. : il. color.

Orientador(a): Prof. Dr. Ulisses Weyl da Cunha Costa
Trabalho de Conclusão (Graduação) - Universidade Federal do
Pará, Campus Universitário de Cametá, Curso de Sistemas de
Informação, Cametá, 2024.

1. Covid19. 2. IoT. 3. Monitoramento. I. Título.

CDD 003

ILANNILDO VIANA DA CRUZ

DESENVOLVIMENTO DA PLATAFORMA WEB MONITORA TOCANTINS

Data da Defesa: Cametá, PA, 28 de Novembro de 2024.

Membros da Banca Examinadora:

Professor e orientador Ulisses Weyl da Cunha Costa, Dr. (Presidente)

Universidade Federal do Pará

Prof. Carlos dos Santos Portela, Dr. (Membro interno)

Universidade Federal do Pará

Prof. Dalmi Gama dos Santos, Dr. (Membro externo)

Universidade Federal do Pará

AGRADECIMENTOS

A conclusão deste trabalho representa não apenas o fim de uma jornada acadêmica, mas também o reflexo de todo o apoio e incentivo que recebi ao longo dessa trajetória.

Agradeço primeiramente a Deus, por me dar forças e sabedoria nos momentos mais desafiadores.

Ao meu orientador Ulisses Weyl da Cunha Costa, agradeço pela paciência, pelos ensinamentos e pelas valiosas orientações ao longo desse processo. Sua expertise e disposição em me guiar tornaram este trabalho possível e mais enriquecedor.

À Faculdade de Sistemas de Informação (FASI), sou grato por todo o conhecimento adquirido durante esses anos. Por me proporcionarem um ambiente de aprendizado e crescimento, que foi essencial para a formação deste trabalho.

À minha família, o meu mais sincero agradecimento por todo amor e suporte incondicional. Aos meus pais, por sempre acreditarem em mim e por me ensinarem o valor do esforço e da dedicação. Suas palavras de incentivo e compreensão foram essenciais em todos os momentos.

À minha esposa Laiane Melo Viana, meu porto seguro, que esteve ao meu lado em cada momento dessa caminhada. Obrigado por me apoiar, entender minhas ausências e por ser minha maior motivação.

Por fim, agradeço a todos que, de alguma forma, fizeram parte dessa jornada. Este trabalho é, também, fruto da colaboração e do carinho de cada um de vocês.

Muito obrigado!

“Em programação, assim como na vida, o código só se aperfeiçoa com persistência. Cada erro é uma oportunidade de aprender, e cada tentativa é um passo mais próximo da solução.”

RESUMO

Este projeto foi desenvolvido com o objetivo de monitorar a COVID-19 e a imunização na região do Baixo-Tocantins, utilizando tecnologias de Internet das Coisas (IoT) e dados do censo. Através de um aplicativo, informações sobre pessoas sintomáticas e imunização são coletadas, permitindo mapear os casos de COVID-19 e identificar indivíduos com o ciclo vacinal incompleto. A plataforma web correspondente exibe esses dados em gráficos e tabelas, além de fornecer relatórios estratégicos para apoiar a tomada de decisões. O sistema é de grande relevância para gestores de saúde, auxiliando na orientação de ações preventivas e no acompanhamento da imunização. A arquitetura da solução foi desenvolvida em ReactJS e NodeJS, seguindo o modelo incremental, e foi testada e implantada em um servidor web, garantindo uma visualização eficiente dos dados e acessibilidade remota para os usuários.

Palavras-chave: Covid19, IoT, Monitoramento.

ABSTRACT

This project was developed to monitor COVID-19 and vaccination in the Baixo-Tocantins region, using Internet of Things (IoT) technologies and census data. Through an application, information on symptomatic individuals and immunization is collected, allowing the mapping of COVID-19 cases and identifying individuals with incomplete vaccination schedules. The corresponding web platform displays this data in charts and tables and provides strategic reports to support decision-making. The system is highly relevant to health managers, aiding in preventive actions and monitoring immunization efforts. The solution's architecture was developed using ReactJS and NodeJS, following an incremental model, and has been tested and deployed on a web server, ensuring efficient data visualization and remote accessibility for users.

Keywords: Covid19, IoT, Monitoring.

LISTA DE FIGURAS

Figura 1 – Modelo Incremental.	16
Figura 2 – O processo de ciência de dados.....	19
Figura 3 – Exemplo de gráfico de barras.....	21
Figura 4 – Exemplo de gráfico de pizza.	21
Figura 5 – Componentes ReactJS	24
Figura 6 – Exemplo de arquivo JSX.....	25
Figura 7 – Exemplo de arquivo JSX.....	26
Figura 8 – Logo do Monitora Tocantins.....	34
Figura 9 – Diagrama de entidade e relacionamento	35
Figura 10 – Tela de Login	37
Figura 11 – Tela de Cadastro.....	38
Figura 12 – Tela de Recuperar senha	38
Figura 13 – Tela de Recuperar senha	39
Figura 14 – Menu lateral esquerdo	40
Figura 15 – Tela de dashboard	41
Figura 16 – Tela de cadastro e listagem de usuários	41
Figura 17 – Tela de questionários.....	42

LISTA DE QUADROS

Quadro 1 – Requisitos Funcionais.....	31
Quadro 2 – Requisitos Não-Funcionais.....	33

LISTA DE SIGLAS

TCC	Trabalho de Conclusão de Curso
UFPA	Universidade Federal do Pará
JSX	JavaScript XML
XML	Extensible Markup Language
IOT	Internet Of Things
UI	User Interface
UX	User Experience
RF	Requisitos Funcionais
RNF	Requisitos Não-Funcionais
AED	Análise Exploratória de Dados
AM	Aprendizado de Máquina
KDD	Knowledge Discovery in Databases
SPA	Single Page Application
DOM	Document Object Model
HTML	HyperText Markup Language
REST	Representational State Transfer
API	Application Programming Interface
URI	Uniform Resource Identifier
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
IBGE	Instituto Brasileiro de Geografia e Estatística
CPF	Cadastro de Pessoa Física
SQL	Structured Query Language
DB	Database

SUMÁRIO

1	INTRODUÇÃO	13
1.1	JUSTIFICATIVA	13
1.2	OBJETIVOS	14
1.2.1	Geral.....	14
1.2.2	Específicos	14
1.3	METODOLOGIA.....	15
1.4	ESTRUTURA.....	16
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	CIÊNCIA DE DADOS.....	18
2.1.1	Pré-processamento de dados	20
2.1.2	Visualização de dados	20
2.1.3	Dashboard	22
2.2	FERRAMENTAS DE DESENVOLVIMENTO	22
2.2.1	ReactJS.....	22
2.2.2	JSX.....	24
2.2.3	ES2017 (JavaScript).....	25
2.2.4	Rest/RestFul	26
2.2.5	NodeJS	27
2.2.6	MUI.....	27
2.3	TRABALHOS RELACIONADOS	28
3	METODOLOGIA	30
3.1	PLANEJAMENTO, PROJETO E IMPLEMENTAÇÃO	30
3.2	BASE DE DADOS MONITORA TOCANTINS.....	30
3.2.1	Características dos Dados	30
3.3	DESENVOLVIMENTO DO SISTEMA.....	31
3.4	REQUISITOS DO SISTEMA	31
3.4.1	Requisitos Funcionais	31
3.4.2	Requisitos Não-Funcionais	32
3.5	PROJETO DE INTERFACE.....	33
3.5.1	Logo	34
3.5.2	Ícones	34
3.5.3	Cores	34
3.6	PROJETO DE BANCO DE DADOS.....	34
4	RESULTADOS OBTIDOS.....	37
4.1	TELA DE LOGIN	37
4.2	TELA DE CADASTRO	37
4.3	TELA DE RECUPERAR SENHA.....	38
4.4	TELA DE RECUPERAR SENHA.....	39
4.5	MENU LATERAL ESQUERDO.....	39
4.6	DASHBOARD	40
4.7	TELAS DE CADASTRO E LISTAGEM DE USUÁRIOS	41
4.8	TELAS DE FORMULÁRIOS RESPONDIDOS	42
5	CONCLUSÃO	43
6	REFERÊNCIAS	44

1 INTRODUÇÃO

A rápida expansão da internet e o avanço tecnológico têm transformado profundamente o cenário do desenvolvimento de *software* e da tecnologia da informação. Com o crescente uso da internet, serviços, produtos e informações estão cada vez mais disponíveis *online*, proporcionando aos usuários uma gama ampla de opções.

Além disso, o surgimento de novas tecnologias com paradigmas inovadores tem impulsionado melhorias significativas na experiência do usuário nos sites. No entanto, essa evolução constante também traz desafios para os desenvolvedores, que precisam lidar com a complexidade dessas novas tecnologias e garantir a entrega de software de alta qualidade.

Nesse sentido, surgiram os *frameworks* para o desenvolvimento *web*. De acordo com Lima e Lezana (2004, p. 178), um *framework* desempenha o papel de simplificar temas complexos, permitindo que sejam estudados e analisados de maneira mais acessível. Além disso, sua utilização é frequente para compartilhar ideias e descobertas com a comunidade, impulsionar o desenvolvimento de ferramentas, técnicas e procedimentos, entre outros benefícios. Essa abordagem é um conceito genérico acerca de um *framework*, mas trazendo para a área da computação, segundo o site GoDaddy (empresa de provedores e hospedagem *web*), um *framework* é uma estrutura que combina diferentes componentes genéricos, com a finalidade de simplificar, organizar e agilizar o processo de desenvolvimento de aplicações *web*.

Nesse contexto, é crucial que os profissionais da área estejam atualizados, sejam ágeis na aprendizagem e estejam dispostos a enfrentar os desafios que surgem nesse ambiente em constante mudança.

1.1 JUSTIFICATIVA

A universidade Federal do Pará (UFPA) Campus do Tocantins/Cametá, possui um projeto chamado Aplicando IoT e censo para o monitoramento de COVID-19 e imunização na região do Baixo-Tocantins, é um projeto que visa ao desenvolvimento de uma nova versão do aplicativo monitora Cametá, que objetiva mapear os casos de covid-19 e o avanço da imunização da população, através da captação de pessoas sintomáticas, as quais fornecerão as informações necessárias ao monitoramento do número de casos de ocorrência nos diferentes locais informando localização, grupo que interage e lugares que frequentam. No entanto, o projeto possui um grande gargalo quanto a análise e amostragem dos dados, bem como o controle dos usuários e formulários capitados.

Conforme mencionado por Santos (2013), a internet abriga o mais extenso acervo de informações globalmente disponível. As pessoas continuamente interagem, compartilhando um considerável volume de dados, os quais podem perder sua relevância em meio à profusão informacional. Nesse contexto, as organizações enfrentam como principal desafio a avaliação e interpretação dos dados cruciais para embasar suas decisões estratégicas. De acordo com Rezende et al. (2003), a "devido à incapacidade do ser humano de interpretar tamanha quantidade de dados, muita informação e conhecimento, possivelmente úteis, podem estar sendo desperdiçados, ficando ocultos dentro das bases de dados espalhadas pelo mundo".

Tendo como principal objetivo melhorar a visualização e interpretação das informações dos dados captados pelo projeto, foi idealizado uma plataforma *web* denominada Monitora Tocantins, que permitirá a visualização dos dados em forma de gráficos e tabelas, além de poder gerenciar os usuários e formulários. A análise dos procedimentos de obtenção de informações, o desenvolvimento de métricas e indicadores nas plataformas de mídia social, bem como a compreensão dos métodos de avaliação de resultados, são aspectos cruciais para a gestão eficaz de marca, produtos e interação com o público-alvo (PALOMINO; ANDRADE, 2013).

1.2 OBJETIVOS

1.2.1 Geral

Desenvolver o *Front-End* de uma plataforma *web* para exibir os dados em forma de gráficos e tabelas resultantes do cruzamento de dados do Projeto Aplicando IoT e censo para o monitoramento de COVID-19 e imunização na região do Baixo-Tocantins (Monitora Tocantins).

1.2.2 Específicos

- Realizar um estudo bibliográfico para identificar os requisitos de uma boa implementação de uma plataforma *web*.
- Estudar sobre o uso de *frameworks* e bibliotecas para a criação de páginas dinâmicas.
- Estudar e analisar os dados captados pelo aplicativo Monitora Tocantins
- Detalhar as tecnologias e ferramentas utilizadas e suas contribuições para na implementação deste projeto.

- Realizar testes das funcionalidades e implantar o sistema em um servidor *web*.

1.3 METODOLOGIA

Este trabalho iniciou a partir do levantamento dos Requisitos, pois, envolve a identificação e a documentação que o sistema deve atender, bem como as características de desempenho, segurança e usabilidade.

Em seguida, foi necessário a análise do Projeto de Interface do Aplicativo Monitora Tocantins para examinar e compreender os elementos e fluxos da plataforma, com o objetivo de criar uma interface de usuário (UI) que siga os padrões estabelecidos no aplicativo Monitora Tocantins, garantindo uma experiência consistente e intuitiva para os usuários.

Em paralelo ao processo de análise do Projeto de Interface, iniciou-se uma análise e estudo sobre o desenvolvimento *web* baseado em um *framework*, com o objetivo de pesquisar e avaliar um *framework* adequado para implementação de páginas dinâmicas, ou seja, que possuem os dados atualizados sem a necessidade de recarregar a página a todo momento para obter os dados, e claro, levando em consideração fatores como eficiência, escalabilidade e facilidade de manutenção. Ainda, buscou-se o estudo a cerca da implementação utilizando padrão de projeto baseado na arquitetura *Clean Code*, que envolve a compreensão dos princípios e práticas de programação que visam criar um código limpo, modular e de fácil compreensão, facilitando a manutenção e evolução do sistema.

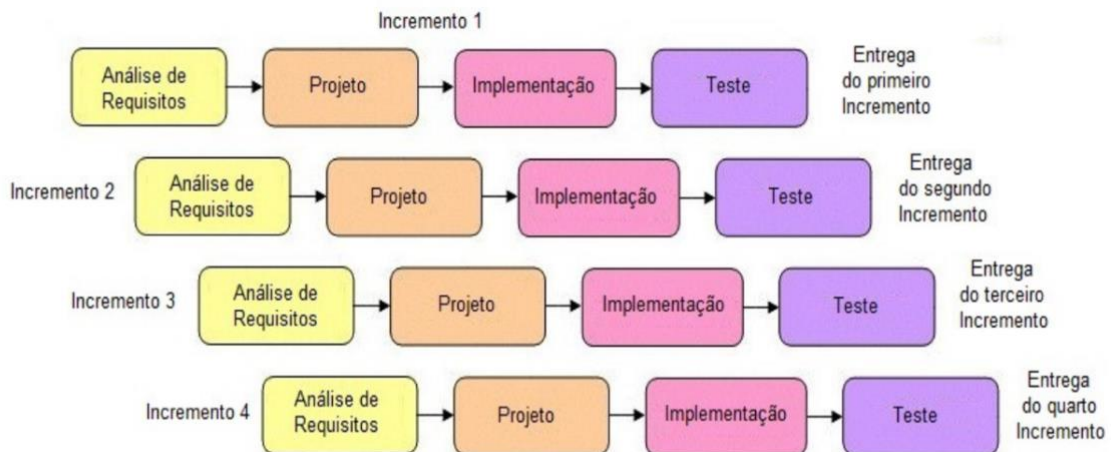
A seguir, foi dado início ao desenvolvimento e elaboração da interface do usuário (UI), pois é ela que envolve a criação dos elementos visuais e interativos da plataforma, com foco na usabilidade, estética e coerência visual, de modo a proporcionar uma experiência agradável e intuitiva aos usuários.

Por último, se deu o desenvolvimento e codificação da plataforma *web* utilizando a biblioteca ReactJs, pois ela traz vantagens, como a criação de componentes reutilizáveis e a capacidade de atualizar a interface do usuário de forma eficiente.

O desenvolvimento seguiu o modelo incremental, que combina características do modelo sequencial, ou modelo cascata, com a abordagem iterativa da prototipagem. De acordo com Pressman (2005), esse modelo pressupõe que o *software* pode ser continuamente expandido e aprimorado, sendo que cada incremento, individualmente ou em conjunto, é desenvolvido seguindo as etapas do modelo cascata. Assim, o desenvolvimento foi dividido em análise de requisitos, projeto de interface, implementação e teste, permitindo o progresso gradual e a incorporação de novas funcionalidades ao longo do processo.

O primeiro incremento, conhecido como “núcleo do produto” (PRESSMAN, 2006), desempenha um papel crucial, uma vez que abrange a implementação dos requisitos fundamentais necessários para o funcionamento do sistema e para atender, de forma mínima, às necessidades do cliente.

Figura 1 – Modelo Incremental.



Fonte: Pressman (2002, p.33).

Após esse processo, foram realizados os testes e a plataforma foi disponibilizada em um servidor *web*, permitindo que os usuários acessem e utilizem a plataforma remotamente.

1.4 ESTRUTURA

Este trabalho segue uma estrutura de 5 (cinco) capítulos, sendo que neste capítulo introdutório foram abordados a contextualização, justificativa, objetivos gerais e específicos que foram estabelecidos, além da metodologia empregada para o seu desenvolvimento.

O segundo capítulo descreve a fundamentação teórica apresenta o projeto Monitora Tocantins, no qual é feita uma conceituação sobre os objetivos do projeto. Na sequência, serão abordados os requisitos do projeto, ou seja, os requisitos funcionais (RF) e não funcionais (RNF), além do projeto de interface e o projeto de banco de dados utilizado no aplicativo Monitora Tocantins. Por fim, é demonstrado como se deu o processo de desenvolvimento da plataforma *web*.

O capítulo 3 mostra as ferramentas utilizadas durante o processo de desenvolvimento do *software*. Em seguida, no capítulo 4, é apresentado o sistema *web* Monitora Tocantins, mostrando os módulos e funcionalidades da aplicação.

O capítulo 5 é referente as considerações finais, apresentando um breve resumo do trabalho e enfatizando quais foram os resultados obtidos, as dificuldades enfrentadas para a execução e as lições aprendidas.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, serão abordados os conceitos pertinentes à Ciência de dados, análise de dados e a etapa de pré-processamento de dados. Além disso, irá ser abordado o conceito de visualização de dados com suas diferentes técnicas e o conceito e uso de *Dashboards*. Também, as ferramentas utilizadas ao longo do desenvolvimento do sistema *web*. Por último, a análise de trabalhos relacionados será apresentada e discutida.

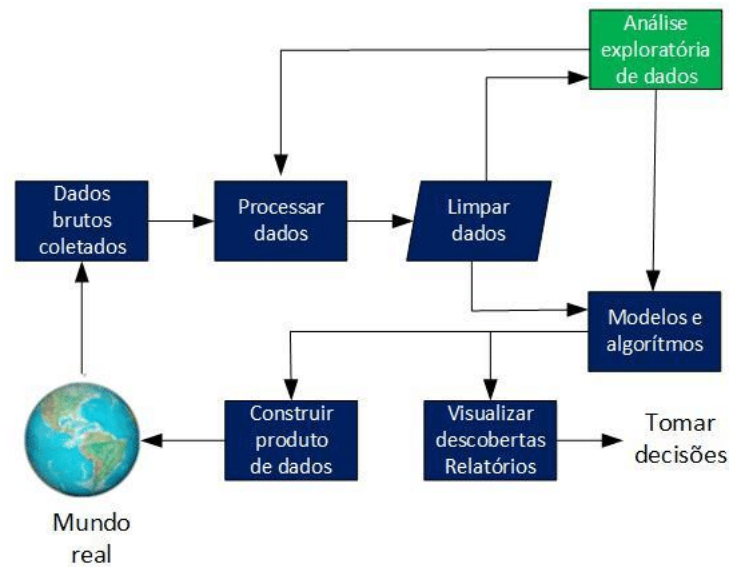
2.1 CIÊNCIA DE DADOS

Muitas pessoas costumam entender de forma errônea a ciência de dados apenas como o processo de análise de informações, onde os dados são examinados por meio de estatísticas, *machine learning* ou uma simples filtragem. Contudo, a ciência de dados vai além disso, para Silva (2022) a ciência de dados é mais complexa do que aparenta, pois ela utiliza técnicas semelhantes às da ciência da computação, abrangendo desde a obtenção dos dados, armazenamento, gerenciamento, visualização, compartilhamento, privacidade, modelagem, análise, e segurança dos dados, até a integração de todos esses serviços. Sendo assim, a ciência de dados engloba todo o ciclo de vida dos dados e é formada por uma variedade de disciplinas, modelos, tecnologias, processos e procedimentos relacionados ao manejo e análise de dados.

O modelo básico deste processo de ciência de dados utilizado neste artigo será o modelo definido em Schutt e O'Neil (2014) e representado na Figura 2. Em resumo, esse modelo se baseia nas seguintes etapas:

1. Análises estatísticas;
2. Mudança dos dados para outro formato (transformação) para ser usado ou processado: (análise, raspagem e formatação de dados)
3. Visualização (gráficos, sumários, ferramentas, etc.).

Figura 2 – O processo de ciência de dados.



Fonte: Schutt e O'Neil (2014).

Inicialmente, os dados são coletados do mundo real, em sua forma bruta, e passam por um processo inicial de limpeza e transformação, também conhecido como "data munging". Uma vez que os dados estão preparados, técnicas de Análise Exploratória de Dados (AED) são aplicadas para uma análise mais aprofundada. Em seguida, algoritmos de aprendizado de máquina entram em cena para extrair insights dos dados. Por fim, as informações são representadas visualmente utilizando diversas técnicas e ferramentas de visualização de dados. Este processo segue uma sequência lógica, onde cada etapa prepara os dados para a próxima, culminando na obtenção de insights valiosos a partir dos dados disponíveis.

De acordo com Amaral (2016), o ato de interpretar dados envolve a construção de uma narrativa. Nesse contexto, a visualização de dados desempenha um papel crucial, pois é responsável por transmitir essa narrativa de maneira clara e compreensível. Grus (2016) destaca dois propósitos essenciais para o uso da visualização: a exploração e a comunicação dos dados. A utilização de gráficos torna mais simples a identificação rápida de padrões e tendências nos dados analisados. De acordo Silva (2022) para uma visualização eficiente, é essencial selecionar a forma gráfica mais apropriada, organizar tabelas de maneira clara e optar pelo tipo de gráfico mais adequado a cada tipo de variável.

2.1.1 Pré-processamento de dados

Batista (2023) defende que a qualidade dos dados é uma das principais preocupações em AM (Aprendizado de Máquina) e KDD (Descoberta de Conhecimento em Bases de Dados). Esse fato ocorre, pois, a maioria dos métodos utilizados nessa área, tais como os algoritmos de aprendizagem profissional. Dessa forma, a qualidade do conhecimento extraído é amplamente determinada pela qualidade dos dados fornecidos como entrada.

Portanto, a limpeza, normalização e a preparação dos dados são etapas fundamentais em todo processo de mineração de dados e aprendizado de máquina, para garantir que os modelos possam realmente agregar valor e tomar decisões baseadas em fatos confiáveis e relevantes (BATISTA, 2023).

2.1.2 Visualização de dados

A visualização de dados, quando bem definida, atua como uma ferramenta que converte dados em informações, permitindo que o indivíduo, por meio de suas habilidades de percepção e cognição, desenvolva conhecimento a partir dessas informações (SILVA, 2022).

Para Linz (2022) a visualização consiste em representar dados de forma visível com o auxílio de computadores e softwares especializados para criar essas representações. O objetivo das visualizações é aumentar a carga cognitiva, tornando mais fácil o processo de construção de conhecimento.

Para uma visualização eficiente dos dados, é fundamental selecionar a representação gráfica adequada, organizar as tabelas de forma clara e escolher o gráfico mais apropriado para cada tipo de variável (SILVA, 2022).

O gráfico de barras é uma das formas mais populares de exibir visualmente dados quantitativos (SILVA, 2022). Nesse tipo de visualização, um eixo mostra as categorias comparadas, enquanto o outro apresenta os valores medidos (OLIVEIRA, 2022). De acordo com Filho (s.d.) as barras desse gráfico podem ser horizontais ou verticais, facilitando a comparação dos dados, que podem ser numéricos ou qualitativos.

Figura 3 – Exemplo de gráfico de barras.



Fonte: Autor.

Por outro lado, o **gráfico de pizza** é uma representação circular na qual as categorias são mostradas como fatias de pizza, sendo o tamanho de cada fatia proporcional à sua porcentagem ou valor em relação ao todo. É mais adequado para representar proporções de um conjunto de dados, mostrando como cada categoria contribui para o total.

Figura 4 – Exemplo de gráfico de pizza.



Fonte: Autor.

2.1.3 Dashboard

Os dashboards são representações visuais de informações, onde os dados são apresentados de forma visual, combinando texto e gráficos, com ênfase nos elementos gráficos (OLIVEIRA, 2022). Assim, a apresentação gráfica, quando manipulada adequadamente, pode comunicar de maneira mais eficiente e clara para quem está visualizando essas informações (CRUZ; JÚNIOR, 2022). Conforme Paranhos (2021), um dashboard exhibe as informações mais cruciais necessárias para alcançar um ou mais objetivos. Essas informações são consolidadas e organizadas em uma única tela para permitir um monitoramento rápido.

Os dashboards podem ser aplicados em diversos domínios de dados e para diferentes propósitos (OLIVEIRA, 2022). A utilização de métodos de exploração de dados, como dashboards para o monitoramento da pandemia de coronavírus, tem sido abordada por vários autores, devido à sua versatilidade, rapidez na análise e facilidade de compreensão tanto para tomadores de decisão quanto para o público em geral (OLIVEIRA, 2022).

2.2 FERRAMENTAS DE DESENVOLVIMENTO

2.2.1 ReactJS

Os Nos últimos anos, a utilização de *frameworks* e bibliotecas no desenvolvimento de software tem se tornado uma prática comum e altamente vantajosa, especialmente no contexto do desenvolvimento de plataformas *web*. Essas ferramentas oferecem um conjunto de classes, bibliotecas e padrões de desenvolvimento que simplificam e aceleram o processo de criação de sistemas, ao mesmo tempo em que promovem a organização e a manutenibilidade do código.

O ReactJS é uma biblioteca Javascript popular para a construção de interfaces do usuário (UI) interativas e dinâmicas, especialmente eficaz na construção de *Single-Page Applications* (SPA), um tipo de aplicativo web que oferece uma experiência de usuário contínua, sem a necessidade de recarregar a página inteira durante a navegação. Seu modelo de programação baseado em componentes e sua abordagem declarativa o tornaram uma escolha comum para o desenvolvimento de aplicações modernas e com muitas interações, como aplicações de análise de dados. A utilização dessa ferramenta traz diversas vantagens para o desenvolvimento de uma aplicação, como:

- **Componentização:** Promove o desenvolvimento baseado em componentes, o que permite dividir a interface do usuário em partes reutilizáveis e

independentes. Isso facilita a manutenção do código e a escalabilidade da aplicação.

- **Gerenciamento de estado simplificado:** SPAs frequentemente lidam com uma grande quantidade de estado do aplicativo. O React simplifica o gerenciamento desse estado por meio de bibliotecas como Redux ou o próprio contexto do React, tornando mais fácil manter a consistência dos dados em toda a aplicação.
- **Reatividade:** Facilita a criação de interfaces de usuário reativas, onde as alterações nos dados são automaticamente refletidas na interface do usuário sem a necessidade de manipulação manual do DOM.
- **Renderização do lado do cliente:** A renderização no lado do cliente é uma característica fundamental das SPAs, permitindo que o conteúdo seja atualizado dinamicamente sem a necessidade de recarregar a página. O React, com sua abordagem eficiente de virtual DOM, é altamente adequado para essa tarefa, proporcionando uma experiência de usuário rápida e fluida.

No entanto, é importante entender quais as desvantagens do uso dessas ferramentas no desenvolvimento e até que ponto elas são úteis para o projeto. Ao optar por utilizar *frameworks* ou bibliotecas para o desenvolvimento de uma plataforma *web* a equipe de desenvolvimento tem que entender que a Curva de aprendizado para iniciantes podem ser desafiadoras, principalmente para entender o conceito de Componentização e o uso do JSX (JavaScript XML), resultando em uma curva de aprendizado íngreme e muitas vezes frustrante. Além disso, outro ponto que pode ser problema é a Arquitetura aplicada ou Padrão de projeto. Embora o ReactJS forneça uma flexibilidade muito grande na organização de projetos, ele não prescreve uma arquitetura específica, o que pode ocasionar muitas inconsistências e a necessidade de implementar soluções adicionais para problemas como gerenciamento de estados globais.

Figura 5 – Componentes ReactJS

```

52  };
53  };
54  };
55  const handleOpenUserMenu = (event: React.MouseEvent<HTMLElement>) => {
56    setAnchorElUser(event.currentTarget);
57  };
58  const handleCloseNavMenu = () => {
59    setAnchorElNav(null);
60  };
61  const handleCloseUserMenu = () => {
62    setAnchorElUser(null);
63  };
64  return (
65    <AppBar
66      position="sticky"
67      elevation={0}
68      sx={{
69        backgroundColor: theme.palette.background.default,
70      }}
71    >
72    <Container maxWidth="lg">
73      <Toolbar disableGutters>
74        <Link href="/" sx={{ display: { xs: "none", md: "flex" }, mr: 1 }}>
75          <img src={Logo} alt="" width="45" />
76        </Link>
77        <Box sx={{ flexGrow: 1, display: { xs: "flex", md: "none" } }}>
78          <IconButton
79            size="large"
80            aria-label="account of current user"
81            aria-controls="menu-appbar"
82            aria-haspopup="true"
83            onClick={handleOpenNavMenu}
84            color="secondary"
85          >
86            <MenuIcon color="action" />
87          </IconButton>
88          <Menu
89            id="menu-appbar"
90            anchorEl={anchorElNav}
91            anchorOrigin={{
92              vertical: "bottom",
93              horizontal: "left",
94            }}
95            keepMounted
96            transformOrigin={{
97              x: 0,
98              y: 0,
99            }}
100          />

```

Fonte: Autor.

2.2.2 JSX

O JSX é uma extensão de sintaxe JavaScript amplamente adotada pelo ReactJS para facilitar a criação de interfaces de usuário dinâmicas. Ao combinar a familiaridade do JavaScript com a expressividade do HTML, o JSX oferece uma abordagem intuitiva e eficiente para desenvolver componentes reutilizáveis e altamente interativos.

Figura 6 – Exemplo de arquivo JSX

```

src > pages > Login > TS index.tsx > Login
96
97
98   <Grid container component="main" sx={{ minHeight: "100vh" }}>
99     <Grid
100       item
101       xs={false}
102       sm={5}
103       md={7}
104       sx={{
105         backgroundColor: theme.palette.primary.main,
106         alignItems: "center",
107         justifyContent: "center",
108         display: "flex",
109       }}
110     >
111       <CustomBox
112         display="flex"
113         flexDirection="column"
114         justifyContent="center"
115         alignItems="center"
116       >
117         <ImgCarousel src={ImgBackground} alt="" />
118         <Typography variant="h4" fontWeight="bold" color="Background" mt={10}>
119           Fique em casa
120         </Typography>
121         <Typography variant="body1" color="Background" mt={2}>
122           Evite lugares públicos e aglomerações.
123         </Typography>
124       </CustomBox>
125     </Grid>
126     <Grid
127       item
128       xs={12}
129       sm={7}
130       md={5}
131       sx={{
132         // backgroundColor: "red",

```

Fonte: Autor.

2.2.3 ES2017 (JavaScript)

ES2017, também conhecido como ECMAScript 2017, é uma versão do padrão ECMAScript que trouxe algumas adições importantes para a linguagem JavaScript. Embora não tenha sido uma atualização tão abrangente quanto o ES6 (ECMAScript 2015), o ES2017 introduziu novos recursos e melhorias que ajudaram a tornar o JavaScript mais expressivo e eficiente.

Uma das adições mais significativas do ES2017 foi a introdução de duas novas palavras-chave: `async` e `await`. Essas palavras-chave permitem uma maneira mais limpa e síncrona de trabalhar com código assíncrono, tornando a escrita e a leitura de código assíncrono mais simples e intuitiva.

Com a introdução do `async/await`, os desenvolvedores podem escrever código assíncrono de forma muito semelhante ao código síncrono, o que torna mais fácil entender o fluxo de controle.

Figura 7 – Exemplo de arquivo Javascript

```
1
2 // Exemplo de código assíncrono com async/await
3 async function fetchData() {
4   try {
5     const response = await fetch('https://api.example.com/data');
6     const data = await response.json();
7     console.log(data);
8   } catch (error) {
9     console.error('Erro ao buscar os dados:', error);
10  }
11 }
12
13 fetchData();
```

Fonte: Autor.

2.2.4 Rest/RestFul

Os princípios Rest (Representational State Transfer) desempenham um papel fundamental na arquitetura de muitas aplicações web modernas, promovendo uma abordagem baseada em recursos e *endpoints* para a comunicação entre front-end e back-end. Ao adotar uma estrutura simples e padronizada, o Rest permite a criação de APIs flexíveis, escaláveis e fáceis de entender.

O REST define um conjunto de diretrizes para projetar serviços web que interagem de forma eficiente com o protocolo HTTP, ainda, ele especifica como os métodos HTTP (GET, POST, PUT, DELETE, etc.) devem ser usados para acessar e manipular recursos em uma aplicação web. Cada recurso na aplicação deve ser acessado através de uma URI (Uniform Resource Identifier) e seguir as regras estabelecidas pelo padrão REST. Em sua forma mais pura, uma implementação de serviço web REST segue quatro princípios básicos:

- **Uso explícito de métodos HTTP:** Cada método HTTP tem um propósito específico, como recuperar dados (GET), criar recursos (POST), atualizar recursos existentes (PUT) ou remover recursos (DELETE)
- **Stateless (Sem estado):** O servidor não mantém o estado da sessão do cliente entre requisições. Cada requisição feita pelo cliente contém todas as informações necessárias para o servidor entender e processar a solicitação, o que simplifica a escalabilidade e a confiabilidade do sistema.

- **Exposição de URIs estruturadas:** As URIs dos recursos devem seguir uma estrutura semelhante à um diretório de arquivos, tornando a organização e a navegação dos recursos da aplicação mais intuitivas e compreensíveis.
- **Transferência de dados via JSON ou XML:** Os dados transferidos entre o cliente e o servidor devem ser formatados usando JSON (JavaScript Object Notation) ou XML (Extensible Markup Language). JSON é amplamente preferido devido à sua simplicidade, legibilidade e ampla adoção em várias plataformas e tecnologias.

Esses princípios permitem que as aplicações REST forneçam uma interface de comunicação consistente, flexível e eficiente, adequada para uma variedade de clientes, como aplicações web e móveis (Android ou iOS), por exemplo.

2.2.5 NodeJS

O NodeJS desempenha um papel fundamental no desenvolvimento full-stack, proporcionando um ambiente de execução JavaScript do lado do servidor altamente eficiente e escalável. Ao integrar o NodeJS em conjunto com o ReactJS no front-end, os desenvolvedores podem criar aplicativos web full-stack coesos e consistentes. Ainda, permite a construção de aplicativos web completos, desde o back-end até o front-end, oferecendo uma experiência de desenvolvimento unificada e simplificada.

Ainda, ele oferece uma série de vantagens significativas para o desenvolvimento de aplicações web, como escalabilidade, desenvolvimento rápido e um vasto ecossistema de pacotes. No entanto, é importante considerar suas limitações e desafios, como desempenho single-thread e curva de aprendizado, ao decidir utilizá-lo em um projeto específico.

2.2.6 MUI

O O MUI (Material-UI) é uma biblioteca de componentes para ReactJS que implementa o Material Design, o sistema de design desenvolvido pelo Google. Ele oferece uma vasta gama de componentes prontos para uso, como botões, campos de texto, barras de navegação e muito mais, todos seguindo as diretrizes de design do Material Design. Ao utilizar o Design System em um projeto, deve-se entender suas vantagens e suas desvantagens.

Como vantagens, pode-se destacar as seguintes características:

- **Facilidade de Uso:** O MUI fornece uma ampla variedade de componentes prontos para uso, facilitando a criação de interfaces de usuário bonitas e consistentes.

- **Consistência Visual:** Todos os componentes do MUI seguem as diretrizes de design do Material Design, garantindo uma experiência visualmente coesa para os usuários.
- **Personalização Flexível:** Embora o MUI forneça estilos padrão de acordo com o Material Design, ele também oferece opções de personalização flexíveis para atender às necessidades específicas de design de cada projeto.
- **Comunidade Ativa:** O MUI possui uma comunidade ativa de desenvolvedores e mantenedores, o que significa que há suporte contínuo, atualizações regulares e uma ampla gama de recursos adicionais disponíveis.

No entanto, como desvantagens do MUI, temos as seguintes:

- **Peso Adicional:** A inclusão de uma biblioteca de componentes como o MUI pode adicionar peso extra ao aplicativo, aumentando o tempo de carregamento, especialmente se não forem utilizados muitos componentes da biblioteca.
- **Curva de Aprendizado:** Embora o MUI seja relativamente fácil de usar, pode haver uma curva de aprendizado para entender todas as opções de personalização e recursos avançados da biblioteca.
- **Personalização Limitada:** Embora o MUI ofereça opções de personalização, algumas personalizações avançadas podem ser mais difíceis de alcançar, especialmente se desviarem significativamente do Material Design.
- **Dependência de Terceiros:** Ao usar o MUI, os desenvolvedores estão adicionando uma dependência de terceiros ao projeto, o que pode introduzir desafios de manutenção e compatibilidade futuros.

2.3 TRABALHOS RELACIONADOS

No projeto realizado por Cruz e Júnior (2022) foi desenvolvida uma plataforma SMC (Sistema de Monitoramento da COVID-19) para otimizar a gestão da crise sanitária provocada pela pandemia, por meio da coleta, análise e visualização eficiente de dados relacionados a casos de COVID-19. Assim, busca-se aprimorar a resposta das unidades de saúde e minimizar o impacto da pandemia na sociedade.

Oliveira (2022) apresentou um projeto cujo objetivo foi criar um sistema de monitoramento da pandemia de Covid-19 que utilize software livre, integrando informações fornecidas pelo Painel da Covid-19 e dados do IBGE, ou seja, um dashboard que apresente

gráficos e outros elementos visuais, incluindo indicadores gerados a partir da integração com os dados do IBGE.

No estudo desenvolvido por Branco et al. (2020), é apresentado um Painel COVID-19 para cidades do Norte Fluminense, que compila dados de casos confirmados, óbitos e indicadores epidemiológicos, além do índice de isolamento social. Seu objetivo é fornecer informações visualizadas para gestores, profissionais de saúde e a população, facilitando a tomada de decisões e promovendo a transparência. A metodologia inclui a coleta e processamento de dados, com visualizações interativas desenvolvidas em Python usando as bibliotecas Dash e Plotly.

Estes trabalhos destacam a importância de plataformas e sistemas de monitoramento para a gestão da crise sanitária provocada pela COVID-19. Essas iniciativas demonstram como a coleta, análise e visualização eficaz de dados podem aprimorar a resposta das unidades de saúde e fornecer informações cruciais para gestores e profissionais.

3 METODOLOGIA

3.1 PLANEJAMENTO, PROJETO E IMPLEMENTAÇÃO

O sistema desenvolvido neste trabalho é responsável pela exibição de um *Dashboard* com dados da Covid-19 e imunização na região do baixo Tocantins, que foram captados através do aplicativo Monitora Tocantins. Os códigos desenvolvidos estão armazenados de forma pública em um repositório no Github¹.

A aplicação web é responsável por disponibilizar um *Dashboard* com os dados da Covid-19. Para tornar essa aplicação disponível na web, foi utilizada a plataforma Heroku, que com sua licença gratuita foi o suficiente durante o início do desenvolvimento deste trabalho.

3.2 BASE DE DADOS MONITORA TOCANTINS

O aplicativo Monitora Tocantins teve como objetivo mapear os casos de covid-19 e o avanço da imunização da população, através da captação de dados das pessoas sintomáticas e informações a respeito do número de pessoas imunizadas. Os dados foram captados no período de 01 de agosto de 2023 até 16 de dezembro de 2023, através de um senso realizado na região do Baixo Tocantins, bem como regiões ribeirinhas, zonas rurais e zonas urbanas do município de Cametá – PA. Os dados foram armazenados em um banco de dados SQL (Structured Query Language) e a cada registro da Tabela ‘Forms Responses’ se refere à um formulário respondido no censo, esse registro possui referência do entrevistado e do entrevistador, ambos os registros salvos em uma tabela chamada ‘Users’.

3.2.1 Características dos Dados

Os dados da Covid-19 captados pelo aplicativo Monitora Tocantins se encontram em um arquivo no formato SQL e possuem 5 tabelas e podem ser visualizadas no Apêndice A e são descritas a seguir:

1. **Users:** É a tabela responsável por armazenar todos os usuários cadastrados pelo senso e contém dados como Nome, CPF e Data de nascimento, etc.
2. **Addresses:** É a tabela responsável por armazenar todos os endereços dos usuários.
3. **Exported Files:** É a tabela armazena a referência de arquivos e relatórios gerados pelo sistema.
4. **Forms Responses:** É a tabela responsável por armazenar as respostas dos formulários

5. **Forms Responses On Users:** Essa tabela guarda a referência do usuário entrevistado, o entrevistador, data da realização do formulário.

Neste trabalho, utilizou-se, principalmente, a tabela de “Forms Responses” (Respostas dos Formulários), por ser a tabela responsável por armazenar os dados referentes aos sintomas, comorbidades, localidade, situações da Covid-19. Além disso, utilizou-se o relacionamento com a tabela “Users” para a obtenção somente do gênero e data de nascimento do entrevistado.

3.3 DESENVOLVIMENTO DO SISTEMA

O desenvolvimento do sistema abordado neste trabalho depende de algumas tecnologias e para que se possa obter um bom software, deve-se compreender e entender bem o funcionamento de cada uma delas. O conhecimento em ReactJs é de suma importância para a construção das telas do sistema (Front-end) e o conhecimento em NodeJs para a criação do *web service* (Serviço RESTFul para servir os dados vindos do banco de dados).

Além da escolha das tecnologias para o desenvolvimento, foi imprescindível passar pelas etapas iniciais para o desenvolvimento de um bom sistema, que foram descritas no capítulo 1.3.

3.4 REQUISITOS DO SISTEMA

Segundo Pressman (2002), os requisitos permitem obter informações que resultam na especificação das características operacionais da aplicação.

3.4.1 Requisitos Funcionais

Os requisitos funcionais referem-se aos requisitos que estão relacionados com a maneira com que o sistema deve operar, onde se especificam as entradas e saídas do sistema e o relacionamento comportamental entre elas, assim como a iteração com o usuário.

Desta forma, os requisitos identificados para o primeiro ciclo desse projeto estão no Quadro 1.

Quadro 1 – Requisitos Funcionais.

ID	NOME	DESCRIÇÃO
----	------	-----------

RF000	Perfil de usuário	O sistema deve permitir o cadastro de usuários do tipo comum, administrador, recenseador, profissional da saúde.
RF001	Administrador	O sistema deve permitir que o administrador visualize e edite o tipo de outros usuários
RF002	Efetuar login	O sistema deve permitir que os usuários acessem o sistema com as seguintes informações: e-mail e senha.
RF003	Gerar Relatório	O sistema deve permitir que o usuário administrador emita relatórios gerais de acordo com as atividades realizadas.
RF004	Recuperar senha	O sistema deve permitir que o usuário recupere a senha de acesso ao sistema.
RF005	Efetuar logout	O sistema deve permitir que os usuários encerrem o seu acesso ao sistema.
RF006	Visualizar estatísticas dos formulários	O sistema deve permitir que o usuário visualize as estatísticas dos formulários, bem como: número de formulários respondidos, em quais locais visitados (cidades, distritos, comunidades, ilhas, vilas, bairros, ruas etc.).

Fonte: Autor.

3.4.2 Requisitos Não-Funcionais

Os requisitos não-funcionais são aqueles que não estão especificamente relacionados com a funcionalidade do sistema. Eles impõem restrições no produto a ser desenvolvido e/ou no processo de desenvolvimento do sistema como também especificam restrições externas as

quais o produto precisa atender. Neste sentido, referem-se a questões como: segurança, confiabilidade, usabilidade, performance, entre outros.

Desta forma, os requisitos não-funcionais identificados para o primeiro ciclo do projeto estão descritos no Quadro 2:

Quadro 2 – Requisitos Não-Funcionais.

ID	NOME	DESCRIÇÃO
RNF 01	Sistema Web	o sistema deve ser desenvolvido para ser utilizado via web
RNF 02	Manutenção	Quando o sistema estiver fora do ar devido a algum problema de interno o sistema deve ter uma tela mostrando ao usuário uma mensagem que o site está em manutenção
RNF 03	Controle de versionamento	O software Github será utilizado para fazer o controle de versionamento do código de cada desenvolvedor.
RNF 04	Validação de usuário	Toda vez que um usuário for realizar o seu cadastro o sistema deve verificar se ele já possui cadastro no sistema, com isso o sistema irá mostrar uma mensagem informando que este usuário já possui cadastro no sistema.
RNF 05	Criptografia	O sistema deve criptografar a senha dos usuários (tokens/hash).

Fonte: Autor.

3.5 PROJETO DE INTERFACE

O documento de projeto de interface estabelece os elementos visuais e a interação do sistema com os usuários, iniciando pela definição da identidade visual. A identidade visual engloba os componentes visuais que identificam uma instituição, marca, produto ou serviço, tais como logotipo, símbolos visuais, paleta de cores, tipografias e layouts. Nesta seção, será apresentada a identidade visual do sistema *web* Monitora Tocantins.

3.5.1 Logo

Figura 8 – Logo do Monitora Tocantins



Fonte: Autor.

Esta é a logo do sistema Monitora Tocantins. A característica principal é a uma imagem em “M”, que representa um gráfico de linhas, remetendo as características do projeto, ou seja, análise de dados e é a inicial do nome do projeto. Quanto ao texto, foi utilizada duas cores (azul e violeta).

3.5.2 Ícones

Os ícones utilizados buscam auxiliar a descrição de itens do produto, servindo como guias aos usuários, facilitando na sua utilização. Para esse projeto, optou-se por utilizar a biblioteca Phosphor¹. O Phosphor é uma família de ícones flexíveis e gratuitos para interfaces, diagramas, apresentações.

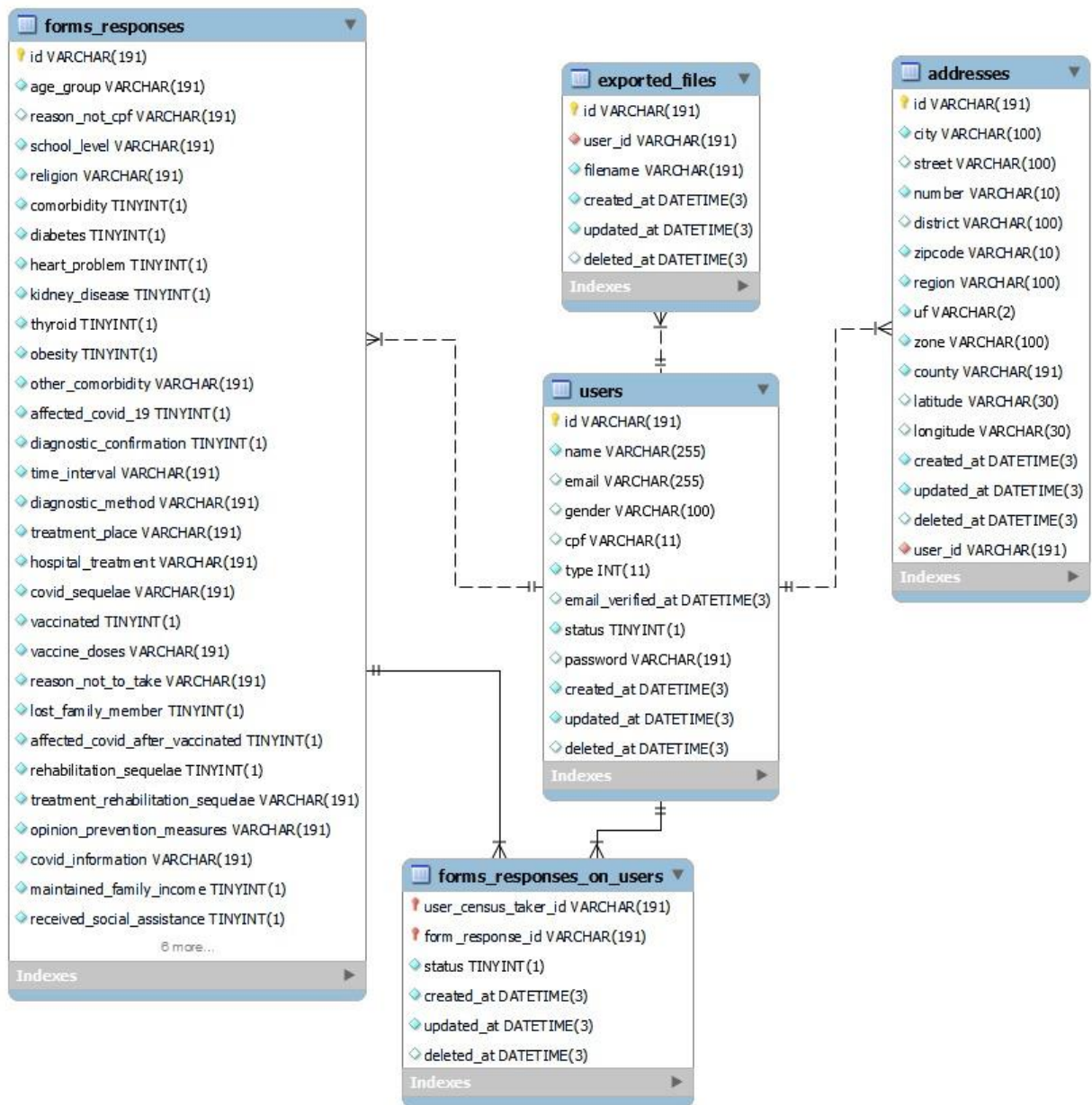
3.5.3 Cores

A cor predominante do sistema é o azul, que remete confiança, segurança e sofisticação e elegância para os usuários do aplicativo, na identidade visual do App Monitora Cametá, que tem o código em hexadecimal #0075FF. Abaixo temos a figura 2 que ilustra a paleta de cores que será utilizada no desenvolvimento da interface do aplicativo

3.6 PROJETO DE BANCO DE DADOS

Um Projeto de Banco de Dados (BD) é toda a parte de modelagem do banco do sistema para que esse seja confiável, eficiente e eficaz. O projeto é de suma importância para o desenvolvimento do sistema, pois, caso ele for mal ou bem construído, pode ocasionar o insucesso ou sucesso do BD e, assim, esse poderá ou não alcançar os objetivos requeridos pelo cliente do projeto.

Figura 9 – Diagrama de entidade e relacionamento



Fonte: Autor.

Este sistema de banco de dados foi projetado para gerenciar eficientemente dados dos formulários, usuários, endereço e arquivos necessários para processos específicos. Cada tabela é construída para capturar informações específicas, garantindo rastreabilidade e conformidade. A utilização de chaves primárias e relacionamentos entre tabelas permite a construção de uma estrutura robusta, na qual os dados podem ser consultados e atualizados conforme necessário, promovendo a integridade e a consistência das informações armazenadas.

- **forms_responses**: Esta tabela armazena as respostas submetidas pelos usuários para diferentes formulários. Cada resposta é identificada de forma única pelo campo id e está

associada a atributos detalhados que capturam as informações fornecidas durante a submissão, como `user_id` (referência do usuário que respondeu o formulário). Esta tabela também suporta o armazenamento de informações de auditoria, como `created_at` e `updated_at`, para rastrear alterações.

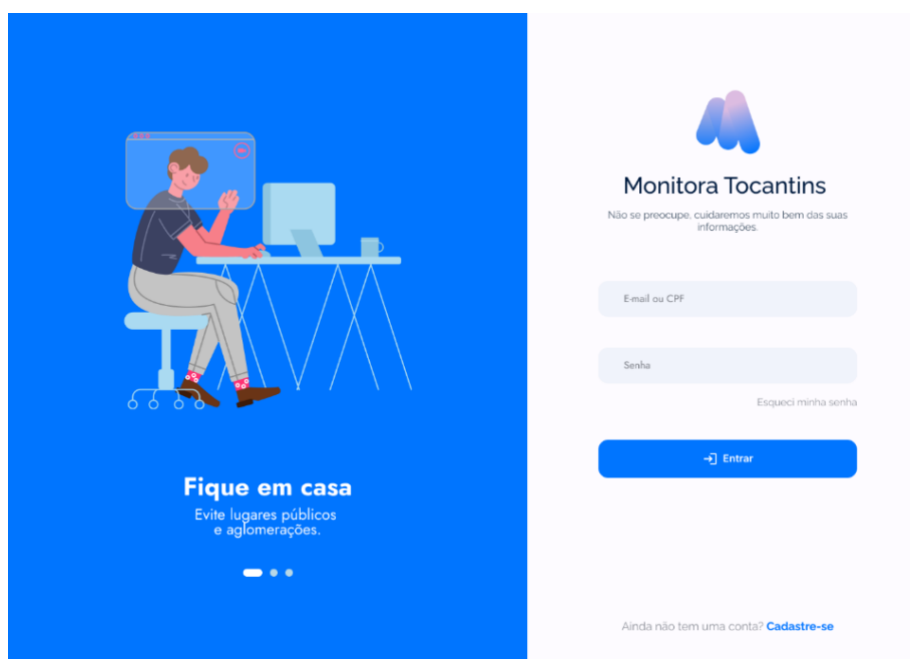
- **exported_files**: A tabela **exported_files** define os arquivos que são necessários ou esperados como parte de um processo ou etapa específica dentro do sistema. Cada arquivo exportado é identificado por um id único e inclui atributos como `filename` (nome do arquivo), e `user_id` (referência do usuário que exportou o arquivo). Os campos `created_at` e `updated_at` armazenam o histórico de criação e modificações para fins de auditoria.
- **addresses**: A tabela **addresses** contém informações de endereço associadas aos usuários do sistema. Cada endereço é identificado por um id único e inclui campos como `street` (rua), `city` (cidade), `state` (estado), e `zip_code` (código postal). A associação com a tabela **users** permite que o sistema vincule cada endereço a um usuário específico. A inclusão dos campos `created_at` e `updated_at` facilita o rastreamento do histórico de alterações no endereço.
- **users**: A tabela **users** armazena dados de identificação e autenticação dos usuários do sistema. Cada usuário é identificado por um id único e possui atributos como `name` (nome), `email` (endereço de e-mail para contato e login), `password` (senha criptografada), `phone` (telefone para contato), e `role` (função ou permissão do usuário). Os campos `created_at` e `updated_at` são usados para fins de auditoria. Esta tabela é central para a autenticação e autorização de usuários dentro do sistema, além de servir como base para relacionamentos com outras tabelas, como **forms_responses** e **addresses**.
- **forms_responses_on_users**: A tabela **forms_responses_on_users** atua como uma tabela associativa, representando um relacionamento muitos-para-muitos entre **forms_responses** e **users**. Cada registro nesta tabela indica que um determinado usuário respondeu a um formulário específico, através do campo `user_response_id` que vincula `user_id` (usuário) a `forms_response_id` (resposta de formulário). Os campos `created_at` e `updated_at` rastreiam o momento de criação e modificações, permitindo auditoria precisa do histórico de respostas de cada usuário. Esta estrutura permite a vinculação flexível de múltiplos usuários a múltiplos formulários, suportando cenários colaborativos e de auditoria detalhada.

4 RESULTADOS OBTIDOS

4.1 TELA DE LOGIN

A tela de login tem como objetivo de facilitar o login de usuários que estão retornando e registrar novos, contendo campos de inserção de “E-mail ou CPF” e “Senha”, botão de ação “entrar” para os usuários acessarem suas contas e links de auxílio, “esqueci minha senha” e para cadastros de novas contas “Cadastre-se”, oferecendo assim, opções para recuperação de acesso ou criação de contas.

Figura 10 – Tela de Login

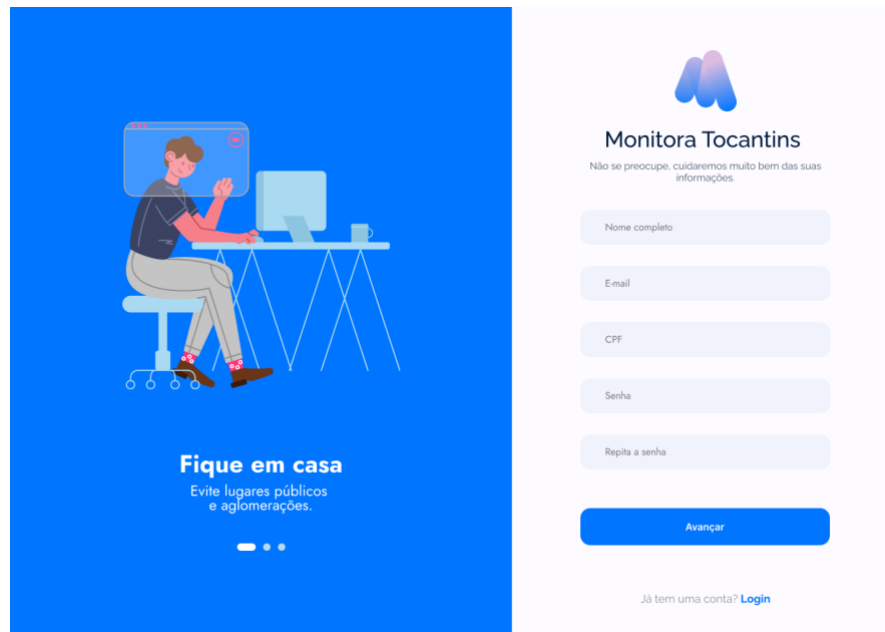


Fonte: Autor.

4.2 TELA DE CADASTRO

Ao acessar a opção “cadastre-se” o usuário será direcionado à tela de criação de conta, onde ele irá preencher informações essenciais, como nome completo, E-mail, CPF e senha, e poderá avançar para a tela seguinte.

Figura 11 – Tela de Cadastro

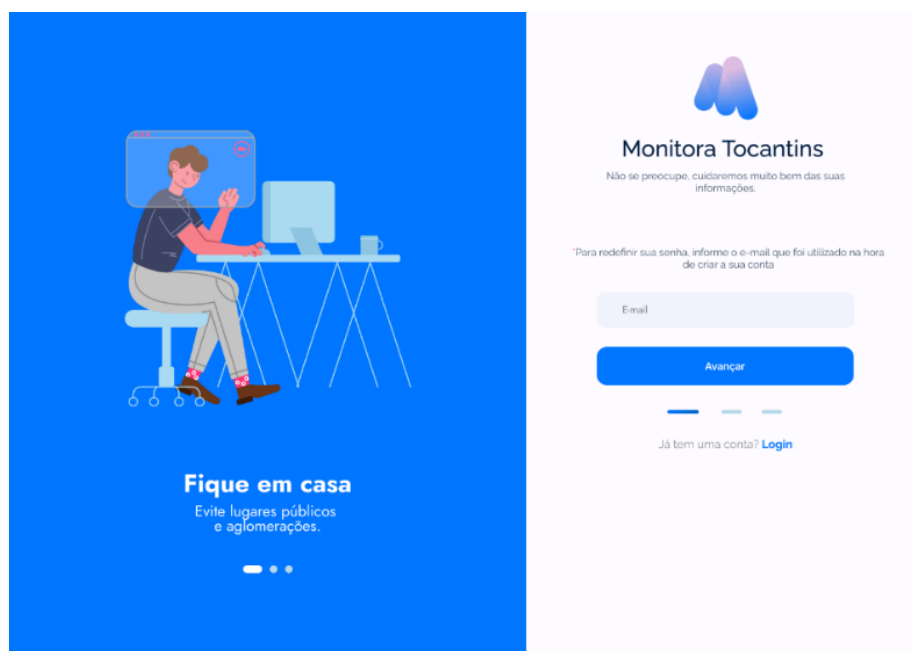


Fonte: Autor.

4.3 TELA DE RECUPERAR SENHA

Ao clicar no link de recuperação de senha, o usuário é direcionado para uma tela simples, onde é necessário apenas o seu E-mail para iniciar o processo. A interface inclui: Campo de E-mail, Botão Avançar e Link para Login, essa tela é verificada na figura 12.

Figura 12 – Tela de Recuperar senha

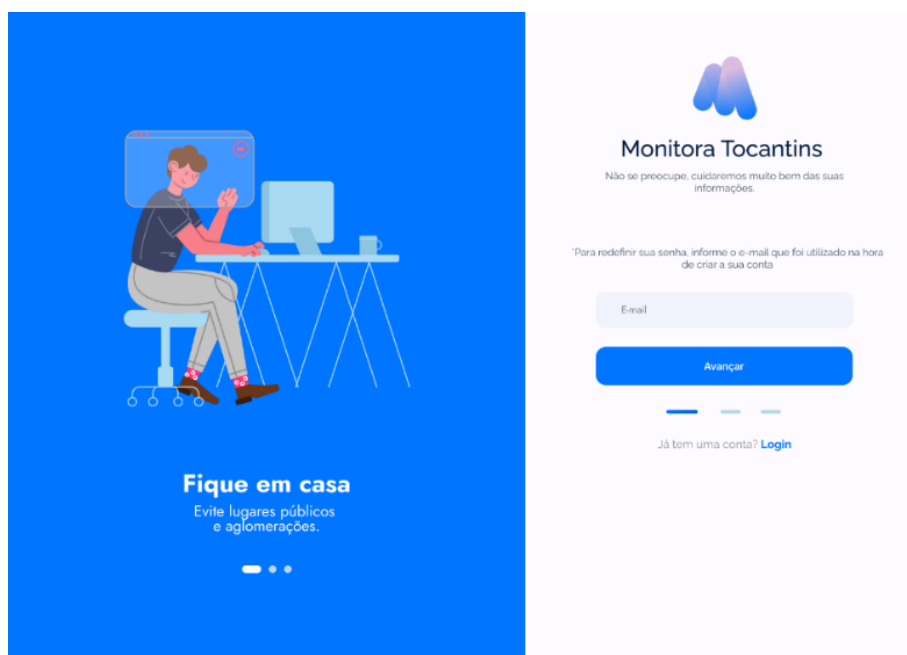


Fonte: Autor.

4.4 TELA DE RECUPERAR SENHA

Ao clicar no link de recuperação de senha, o usuário é direcionado para uma tela simples, onde é necessário apenas o seu E-mail para iniciar o processo. A interface inclui: Campo de E-mail, Botão Avançar e Link para Login, essa tela é verificada na figura 13.

Figura 13 – Tela de Recuperar senha

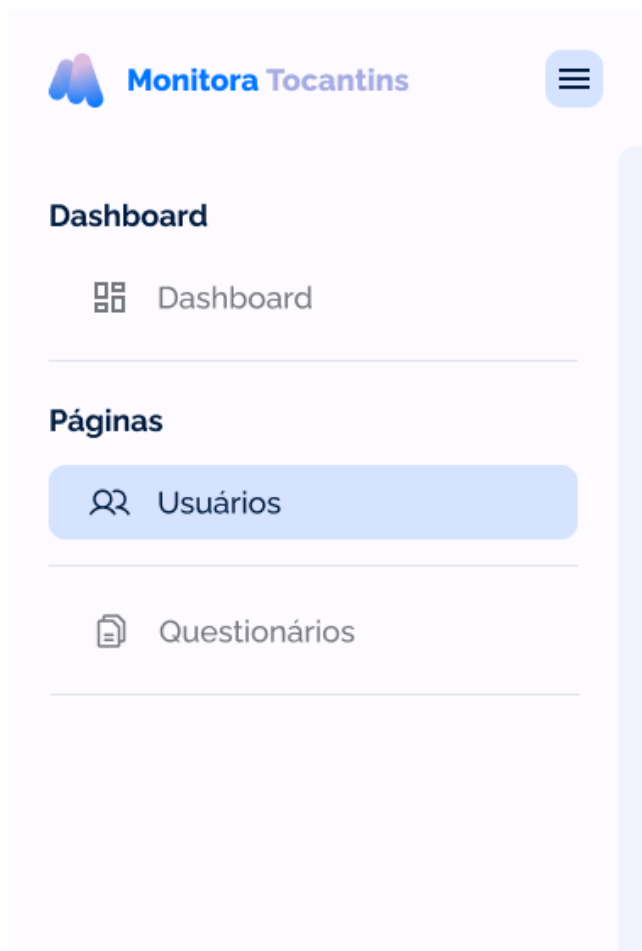


Fonte: Autor.

4.5 MENU LATERAL ESQUERDO

Pode-se observar na figura 14, o menu de navegação, que tem como objetivo facilitar a movimentação entre diferentes seções da plataforma, ele possui link para o painel de controle “Dashboard”, “Usuários” e “Questionários”.

Figura 14 – Menu lateral esquerdo

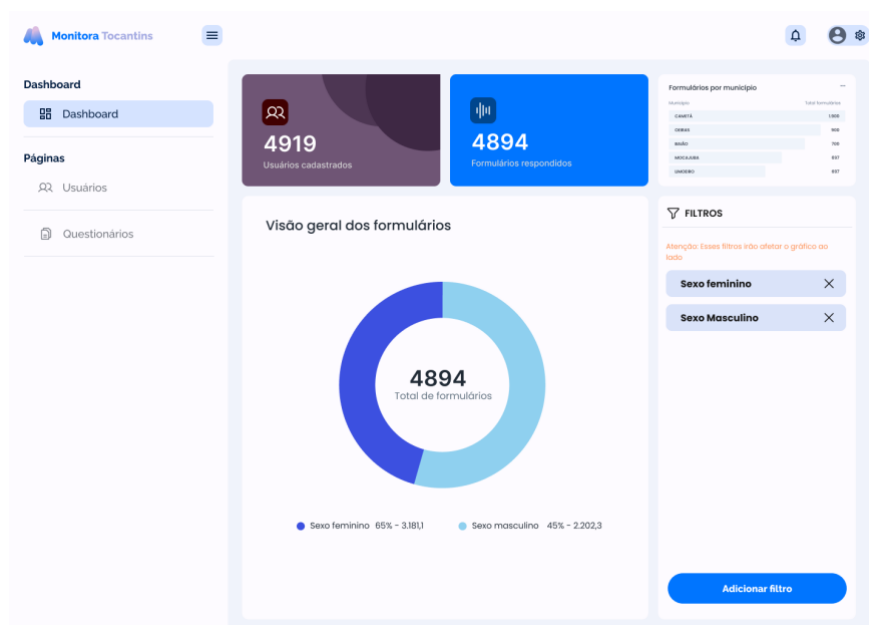


Fonte: Autor.

4.6 DASHBOARD

O dashboard, mostrado na figura 15, é o painel principal da plataforma, e tem como finalidade apresentar vários componentes projetados para fornecer uma visão geral e abrangente dos dados do usuário e da pesquisa, como número de pessoas cadastradas, número de formulários respondidos e é possível obter uma visão geral dos formulários e, ainda, a porcentagem de mulheres e homens que participaram da pesquisa, e entre outros através da função de filtragem, assim como verificar esses dados em um município específico.

Figura 15 – Tela de dashboard



Fonte: Autor.

4.7 TELAS DE CADASTRO E LISTAGEM DE USUÁRIOS

Essa tela, figura 16, tem a função de cadastro de novos usuários e também, mostrar os que já estão cadastrados na plataforma, contendo informações como “Nome”, “E-mail”, “Tipo”, “Status” e “Ações”.

Figura 16 – Tela de cadastro e listagem de usuários

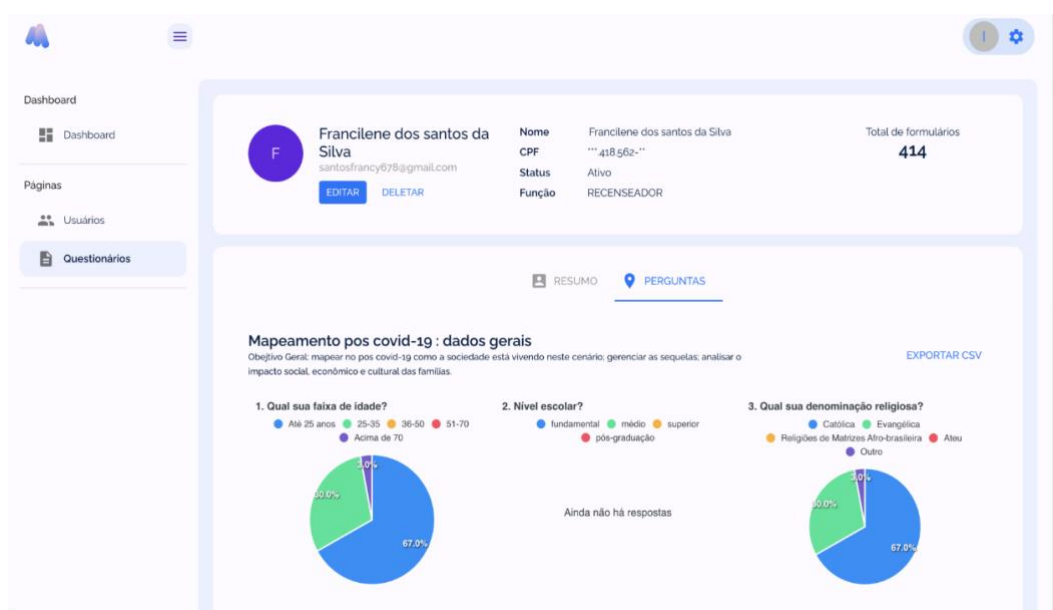
ID	Nome	E-mail	Tipo	Status	Ações
1	Fulano de tal	fulano@email.com	Comum	Ativo	EDITAR DELETAR
2	Fulano de tal	fulano@email.com	Recenseador	Inativo	EDITAR DELETAR
3	Fulano de tal	fulano@email.com	Recenseador	Ativo	EDITAR DELETAR
4	Fulano de tal	fulano@email.com	Recenseador	Ativo	EDITAR DELETAR
5	Fulano de tal	fulano@email.com	Comum	Ativo	EDITAR DELETAR
6	Fulano de tal	fulano@email.com	Comum	Ativo	EDITAR DELETAR
7	Fulano de tal	fulano@email.com	Admin	Ativo	EDITAR DELETAR
8	Fulano de tal	fulano@email.com	Comum	Ativo	EDITAR DELETAR
9	Fulano de tal	fulano@email.com	Comum	Ativo	EDITAR DELETAR
10	Fulano de tal	fulano@email.com	Comum	Ativo	EDITAR DELETAR

Fonte: Autor.

4.8 TELAS DE FORMULÁRIOS RESPONDIDOS

Na tela de formulários, observada na figura 17, é possível fazer um acompanhamento mais detalhado da quantidade de perguntas e respostas obtidas em um determinado local através dos graficos, também é possível verificar a quantidade de formulários aplicados pelo recenseador.

Figura 17 – Tela de questionários



Fonte: Autor.

5 CONCLUSÃO

O projeto Monitora Tocantins resultou em uma plataforma web robusta e eficiente, projetada para monitorar a COVID-19 e o processo de imunização na região do Baixo-Tocantins. A integração de tecnologias avançadas, como IoT e dados do censo, permitiu a coleta, análise e apresentação de informações críticas de forma dinâmica e acessível. Essas funcionalidades foram fundamentais para compreender a disseminação do vírus, o índice de adesão à vacinação e as lacunas no ciclo vacinal, fornecendo suporte estratégico para a tomada de decisões por parte dos gestores de saúde.

Além disso, a plataforma se destacou por promover a conscientização da população, facilitando o acesso a dados sobre o estágio de imunização e reforçando a importância de medidas preventivas contra a COVID-19. Os relatórios em tempo real e as visualizações detalhadas se consolidaram como ferramentas indispensáveis para otimizar a gestão da saúde pública, demonstrando a eficácia e relevância do sistema na luta contra a pandemia.

Como trabalhos futuros, propõe-se a expansão do sistema para monitorar outras doenças de relevância epidemiológica, como dengue e influenza, adaptando algoritmos e modelos de dados para atender a novas necessidades. Outra iniciativa será integrar a plataforma a sistemas nacionais de saúde, promovendo uma troca de informações mais ampla e colaborativa entre diferentes regiões. O uso de aprendizado de máquina também será explorado para prever surtos e padrões epidemiológicos, ampliando a capacidade preditiva do sistema. Por fim, melhorias na interface serão priorizadas, visando torná-la ainda mais acessível, com funcionalidades como notificações personalizadas e alertas preventivos, garantindo maior impacto social e alcance populacional.

Os resultados obtidos confirmam que o projeto se consolidou como uma solução tecnológica de grande valor social, não apenas no enfrentamento da COVID-19, mas também como um modelo replicável para outros contextos e crises sanitárias. Sua continuidade e evolução são essenciais para a gestão eficaz da saúde pública e para o enfrentamento de desafios futuros.

6 REFERÊNCIAS

AMARAL, F. **Introdução à Ciência de dados: mineração de Dados e Big Data**. 1. Ed. Ver. [S.L.]: Starlin Altas book, 2016.

BATISTA, G.E.A.P.A. **Pré-Processamento de Dados em Aprendizado de Máquina Supervisionado**. USP- São Carlos. Mar. 2003.

BRANCO, André et al. **Desenvolvimento e implementação do painel covid-19 para municípios da região norte fluminense**. 2020.

CRUZ, L.R.; JÚNIOR, R.S.M. **Dashboard para monitoramento de dados referentes à pandemia de COVID19 e Vacinação em lagarto**. Instituto Federal de Sergipe- Campus Lagarto. Lagarto- SE- Brasil. 2022.

_____. **Dashboard para monitoramento de dados referentes à pandemia de COVID19 e Vacinação em lagarto**. Instituto Federal de Sergipe- Campus Lagarto. Lagarto- SE- Brasil. 2022.

FILHO, J.M. **Tabelas e Gráficos**. Universidade Estadual Paulista. Acervo digital Unesp. Disponível em: <acervodigital.unesp.br/bitstream/123456789/41572/1/01d20t03.pdf>. Acesso em: 21 de out. 2024.

GoDaddy. **Saiba o que é framework em programação e por que esse recurso é tão eficiente**. Disponível em: <https://br.godaddy.com/blog/o-que-e-framework-em-programacao/#:~:text=Um%20framework%20%C3%A9%20uma%20arquitetura,o%20processo%20de%20desenvolvimento%20web>. Acesso em 08 de nov. de 2023:

GRUS, J. **Data Science do Zero**. Primeiras regras com o Phyton. Rio de Janeiro: alta book, 2016.

LIMA, E. P. de; LEZANA, Á. G. R.. **DESENVOLVENDO UM FRAMEWORK PARA ESTUDAR A AÇÃO ORGANIZACIONAL: DAS COMPETÊNCIAS AO MODELO ORGANIZACIONAL**. Gestão e Produção, Curitiba; Florianópolis, v.12, n.2, p.177-190, mai.-ago. 2005. Disponível em: <https://www.scielo.br/j/gp/a/7b3dyswQthxzYFwQxNL86nk/?format=pdf&lang=pt>. /Acesso em 08 de nov. 2023.

LINZ, J.E.S. **A aplicação de técnicas de visualização de dados na disseminação do conhecimento**. Trabalho de conclusão de curso. Universidade Estadual da Paraíba, Centro de Ciências e Tecnologia. 2022.

OLIVEIRA, D. de. **Desenvolvimento de um sistema para monitoramento da COVID19 no Espírito Santo utilizando software livre**. Instituto Federal do espírito Santo. 2022.

_____. **Desenvolvimento de um sistema para monitoramento da COVID19 no Espírito Santo utilizando software livre**. Instituto Federal do espírito Santo. 2022.

_____. **Desenvolvimento de um sistema para monitoramento da COVID19 no Espírito Santo utilizando software livre**. Instituto Federal do espírito Santo. 2022.

_____. **Engenharia de Software**. 5.ed. Rio de Janeiro: McGraw-Hill, 2002.

PALOMINO, P. T.; ANDRADE, L. A. de. **Perspectivas em redes sociais: a inteligência coletiva como ferramenta de análise de métricas e indicadores de desempenho**. Revista GEMInIS, v. 4, n. 2, p. 164–175, 2013.

PARANHOS, H.P. **Desenvolvimento De Um Dashboard Para Visualização Dos Dados Educacionais Dos Discentes Do Instituto De Ciências Exatas E Aplicadas Da UFOP.** [manuscrito] Hanna Paola Paranhos. 2021.

PRESSMAN, R.S.. **Engenharia de Software.** McGraw-Hill. 6 ed. 2006.

REZENDE, S. O. et al. **Mineração de dados. Sistemas inteligentes:** fundamentos e aplicações, v. 1, p. 307–335, 2003.

SANTOS, A. D. P. d. **Descobrimos eventos locais utilizando análise de séries temporais nos dados do twitter.** 2013.

SILVA, L.A.N. **Ciência de dados como método de transformação de dados em informação.** Instituto Federal de Educação, Ciência e Tecnologia do Sertão Pernambucano, Salgueiro, 2022.

SILVA, W. S. **Visualização de dados:** Análise dos procedimentos para proporcionar visualizações de dados para agências de fomento à ciência e tecnologia. Universidade Federal de Pernambuco. Recife/PE. 2022.

_____. **Visualização de dados:** Análise dos procedimentos para proporcionar visualizações de dados para agências de fomento à ciência e tecnologia. Universidade Federal de Pernambuco. Recife/PE. 2022.

_____. **Visualização de dados:** Análise dos procedimentos para proporcionar visualizações de dados para agências de fomento à ciência e tecnologia. Universidade Federal de Pernambuco. Recife/PE. 2022.