

UNIVERSIDADE FEDERAL DE OURO PRETO  
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS  
DEPARTAMENTO DE COMPUTAÇÃO

DANIELLE EMELY DE SOUZA ALMEIDA

Orientador: Prof. Dr. Reinaldo Silva Fortes

Coorientador: Prof. M.Sc. Lucas Emiliano de Souza Moreira

**UMA APLICAÇÃO WEB VOLTADA PARA O  
TELEMONITORAMENTO DE CASOS DE COVID-19**

Ouro Preto, MG  
2023

UNIVERSIDADE FEDERAL DE OURO PRETO  
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS  
DEPARTAMENTO DE COMPUTAÇÃO

DANIELLE EMELY DE SOUZA ALMEIDA

**UMA APLICAÇÃO WEB VOLTADA PARA O TELEMONITORAMENTO DE CASOS  
DE COVID-19**

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

**Orientador:** Prof. Dr. Reinaldo Silva Fortes

**Coorientador:** Prof. M.Sc. Lucas Emiliano de Souza  
Moreira

Ouro Preto, MG  
2023



## FOLHA DE APROVAÇÃO

**Danielle Emely de Souza Almeida**

### **Uma aplicação web voltada para o telemonitoramento de Casos de COVID-19**

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Ciência da Computação

Aprovada em 27 de Março de 2023.

#### Membros da banca

Reinaldo Silva Fortes (Orientador) - Doutor - Universidade Federal de Ouro Preto  
Lucas Emiliano de Souza Moreira (Coorientador) - Mestre - Instituto Federal de Minas Gerais  
Vander Luis de Souza Freitas (Examinador) - Doutor - Universidade Federal de Ouro Preto  
Rafael Alves Bonfim de Queiroz (Examinador) - Doutor - Universidade Federal de Ouro Preto

Reinaldo Silva Fortes, Orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 27/03/2023.



Documento assinado eletronicamente por **Reinaldo Silva Fortes, PROFESSOR DE MAGISTERIO SUPERIOR**, em 28/03/2023, às 10:15, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site [http://sei.ufop.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **0493054** e o código CRC **245C9FD6**.

*Dedico este trabalho para toda minha família, amigos e professores que fizeram parte da minha jornada de graduação.*

# Agradecimentos

Gostaria de agradecer à minha família, que sempre me ofereceu suporte. Em especial, minha mãe Luciene Francisca e meu pai Dalmário Gomes, sem eles eu não teria chegado até aqui.

Sou grata também a todos os meus amigos que conquistei em Ouro Preto que sempre me apoiaram na graduação, onde incentivávamos mutualmente.

Por fim, gostaria de agradecer à UFOP e todos os professores que contribuíram para a minha formação. Em especial, gostaria de agradecer ao Reinaldo Silva Fortes, que esteve comigo no opCoders, atuando como meu orientador, repassando seus conhecimentos no projeto de extensão opCoders. O opCoders me proporcionou experiências enriquecedoras, tendo contato com os alunos de automação do Instituto Federal de Ouro Preto. Outro agradecimento especial, gostaria de dedicar ao professor Lucas Emiliano, que acreditou em mim para atuar no Projeto UFOP em Ação. Nesse projeto, aprendi muito com as suas orientações e onde nasceu o desenvolvimento deste trabalho.

*"Se eu vi mais longe, foi por estar sobre ombros de gigantes."* (NEWTON, 1676).

# Resumo

A pandemia da COVID-19 apresentou diversos desafios, que demandaram a criação de estratégias eficazes e rápidas para mitigar seus impactos. O monitoramento remoto se mostrou uma dessas estratégias eficazes, uma vez que os dados coletados são essenciais para o acompanhamento de cada caso e para uma análise epidemiológica adequada. A presente monografia tem como objetivo oferecer uma ferramenta automatizada que possibilite a visualização gráfica sintetizada dos dados de monitoramento, de modo que os profissionais possam realizar tomadas de decisão e pesquisas utilizando uma ferramenta gráfica intuitiva e eficiente. O processo de desenvolvimento foi realizado com base em uma aplicação web desenvolvida durante um projeto de iniciação científica intitulado “UFOP em Ação” da Universidade Federal de Ouro Preto. Assim, foi construído um Dashboard por meio do React, tendo como principal recurso a biblioteca Chart.js para geração de gráficos. Ainda como parte da metodologia de desenvolvimento, foram feitos testes de fluxo completo para analisar a eficiência e usabilidade da aplicação por meio do Selenium. Com a implementação da ferramenta, foi percebido que o React se apresenta como uma ferramenta eficiente para a construção de aplicações, graças ao seu vasto ecossistema de bibliotecas de desenvolvimento. Além disso, a solução apresentou um bom desempenho com a execução dos testes, demonstrando que o sistema apresenta gráficos de maneira rápida e com o fluxo de interações que traz um grau relevante de usabilidade, fornecendo maior comodidade e suporte aos profissionais da saúde.

**Palavras-chave:** Telemonitoramento. React. Dashboard. Telessaúde. Gráficos.

# Abstract

The COVID-19 pandemic presented several challenges, which required the creation of effective and quick strategies to mitigate its impacts. Remote monitoring proved to be one of these effective strategies, since the data collected is essential for the follow-up of each case and for an adequate epidemiological analysis. This monograph aims to offer an automated tool that allows the synthesized graphical visualization of monitoring data, so that professionals can carry out decision-making and research using an intuitive and efficient graphical tool. The development process was carried out based on a web application developed during an initiation project entitled “UFOP em Ação” at the Federal University of Ouro Preto. Thus, a Dashboard was built using React, having as its main resource the Chart.js library for generating graphs. Also as part of the development methodology, full flow tests were performed to analyze the efficiency and usability of the application through Selenium. With the implementation of the tool, it was realized that React presents itself as an efficient tool for building applications, thanks to its vast ecosystem of development libraries. In addition, the solution performed well when running the tests, demonstrating that the system presents graphics quickly and with the flow of interactions that brings a relevant degree of usability, providing greater convenience and support to health professionals.

**Keyword:** Telemonitoring, React, Dashboard. Telehealth. Charts.

# Lista de Ilustrações

Figura 2.1 – Hierarquia das ferramentas da Telessaúde. . . . .	6
Figura 2.2 – Exemplo de um <i>dashboard</i> . . . . .	8
Figura 2.3 – Popularidade das linguagens de programação no ano de 2021. . . . .	10
Figura 2.4 – Exemplo de um código JSX. . . . .	11
Figura 2.5 – Comparação entre Class Component e Functional Component. . . . .	13
Figura 2.6 – Exemplo de utilização de um componente pronto, disponível na biblioteca MUI. . . . .	14
Figura 2.7 – Tela principal do Selenium IDE . . . . .	16
Figura 2.8 – Insumos de medicamentos por região no gráfico de rosca e por estado no gráfico de barras. Gráficos gerados no trabalho de Rocha, Nascimento e Sousa (2021). . . . .	20
Figura 2.9 – Gráfico horizontal com informações dos insumos de respiradores em cada estado, construído no trabalho de Rocha, Nascimento e Sousa (2021). . . . .	20
Figura 2.10–Editor da ferramenta Bubble. . . . .	24
Figura 2.11–Tela de login pela plataforma Bubble. . . . .	24
Figura 2.12–Tela de um questionário pela plataforma Bubble. . . . .	25
Figura 2.13–Tabela com os bancos pela plataforma Bubble . . . . .	26
Figura 2.14–Protótipo dos componentes básicos para questionários. . . . .	26
Figura 2.15–Exemplo da tela de uma tabela de um conjunto de dados para monitoramento. . . . .	28
Figura 2.16–Exemplo da tela dos conjuntos de dados para acesso do usuário. . . . .	29
Figura 2.17–Exemplo da tela de um questionário da aplicação do projeto. . . . .	30
Figura 2.18–Exemplo do modal de cadastro de contatos próximos. . . . .	31
Figura 2.19–Exemplo do modal para preencher o motivo da recusa de entrevista. . . . .	32
Figura 3.1 – Estrutura de pastas do projeto . . . . .	34
Figura 3.2 – Exemplos de telas responsivas do <i>dashboard</i> . . . . .	36
Figura 3.3 – Protótipo do <i>dashboard</i> construído no Figma. . . . .	37
Figura 3.4 – Diagrama da lógica do componente genérico para os gráficos. . . . .	38
Figura 3.5 – Exemplos de gráficos do <i>dashboard</i> desenvolvido. . . . .	39
Figura 3.6 – Tela do <i>dashboard</i> desenvolvido. . . . .	40
Figura 3.7 – Exemplo de um seletor de tipo do gráfico. . . . .	41
Figura 3.8 – Exemplos da saída de execução de um caso de teste no Selenium. . . . .	44

# Lista de Tabelas

Tabela 3.1 – Exemplo dos passos e resultado esperado do Caso de Teste para o fluxo de Login. . . . .	42
Tabela 3.2 – Exemplo dos passos e resultado esperado do Caso de Teste para a funcionalidade do <i>accordion</i> . . . . .	42
Tabela A.1 – Caso de Teste C01 . . . . .	52
Tabela A.2 – Caso de Teste C02 . . . . .	53
Tabela A.3 – Caso de Teste C03 . . . . .	54
Tabela A.4 – Caso de Teste C04 . . . . .	55
Tabela A.5 – Caso de Teste C05 . . . . .	56
Tabela A.6 – Caso de Teste C06 . . . . .	57
Tabela A.7 – Caso de Teste C07 . . . . .	58

# Lista de Abreviaturas e Siglas

DECOM	Departamento de Computação
UFOP	Universidade Federal de Ouro Preto
COVID-19	Coronavirus Disease 2019
TIC	Tecnologias da Informação e da Comunicação
JS	JavaScript
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets ou Folhas de Estilo em Cascata
WHO	World Health Organization
OMS	Organização Mundial da Saúde
SARS	Severe Acute Respiratory Syndrome
SARS-Cov-2	Severe Acute Respiratory Syndrome Coronavirus 2
MERS	Middle East Respiratory Syndrome
EPI	Equipamento De proteção Individual
TICs	Tecnologias da Informação e Comunicação
HTTP	Hypertext Transfer Protocol
SQL	Structured Query Language
XML	eXtensible Markup Language
SGBD	Sistema de Gerenciamento de Banco de Dados
SCIH	Controle de Infecção Hospitalar
UFs	Unidades Federativas
MUI	Material-UI
JSX	JavaScriptXML
BaaS	Back-end as a Service
APS	Atenção Primária à Saúde
MDN	Mozilla Developer Network

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Justificativa	2
1.2	Objetivos	2
1.3	Organização do Trabalho	3
<b>2</b>	<b>Revisão Bibliográfica</b>	<b>4</b>
2.1	Fundamentação Teórica	4
2.1.1	A pandemia de COVID-19	4
2.1.2	Telemonitoramento, telessaúde e telemedicina	5
2.1.3	O papel do telemonitoramento na pandemia de COVID-19 e os desafios enfrentados por profissionais de saúde	7
2.1.4	A importância dos <i>dashboards</i> para gestão de saúde	7
2.1.5	Benefícios e tecnologias atuais de uma aplicação web	9
2.1.6	Uma visão geral das bibliotecas e frameworks JavaScript	10
2.1.6.1	A Biblioteca React	11
2.1.7	Bibliotecas para a criação de gráficos no JavaScript	12
2.1.8	<i>Back-end as a Service</i> e a ferramenta Firebase	14
2.1.9	Testes <i>End-To-End</i> e a ferramenta Selenium para ferramenta de automação	15
2.2	Trabalhos Relacionados	16
2.2.1	Incorporação do telemonitoramento da COVID-19 em um município do sul da Bahia: obstáculos e facilidades	17
2.2.2	Um <i>dashboard</i> para visualização de insumos relacionados à Covid-19	19
2.2.3	Teleatendimento como ferramenta de monitoramento de casos suspeitos e/ou confirmados de COVID-19	21
2.2.4	Projeto UFOP em Ação	22
2.2.4.1	Ferramenta Bubble e a construção a inicial do protótipo	23
2.2.4.2	Ferramentas de Web-App do Projeto UFOP em Ação	25
<b>3</b>	<b>Desenvolvimento</b>	<b>33</b>
3.1	Estruturação das pastas da aplicação	33
3.2	Implementação <i>front-end</i> e <i>back-end</i> do <i>dashboard</i>	35
3.3	Construção do <i>dashboard</i>	35
3.3.1	Implementação do <i>dashboard</i>	37
3.4	Execução de testes <i>End-To-End</i> na aplicação	39
3.4.1	Casos de teste como documentação para os testes <i>End-To-End</i>	39
3.4.2	Selenium como ferramenta para automação dos testes <i>End-To-End</i>	43
<b>4</b>	<b>Considerações Finais</b>	<b>46</b>
4.1	Conclusão	46

4.2 Trabalhos Futuros . . . . .	47
<b>Referências . . . . .</b>	<b>48</b>
<b>Apêndices</b>	<b>51</b>
<b>APÊNDICE A Casos de testes . . . . .</b>	<b>52</b>

# 1 Introdução

Durante os três últimos anos e no ano atual de 2023, o mundo enfrenta uma pandemia causada pela transmissão do novo coronavírus [SARS-CoV-2](#), ocasionando uma doença infecciosa conhecida como [COVID-19](#) ([CASCELLA MICHAEL RAJNIK, 2022](#)). Houve um impacto demográfico mundial, que resultou em mais de seis milhões de óbitos até o mês de março de 2023 ([WHO, 2022](#)).

Vários países tiveram que tomar medidas preventivas para reduzir a transmissão do vírus, como a obrigatoriedade no uso de máscaras faciais em locais fechados e movimentados, o incentivo à higienização frequente das mãos, distanciamento social, fechamento de escolas, restaurantes, bares e outros locais com eventos que possibilitam aglomeração.

Conforme a pandemia progrediu nos últimos anos, os progressos tecnológicos e as pesquisas possibilitaram a contenção do vírus por meio do desenvolvimento de vacinas. Esses progressos culminaram na redução de 63% do número total de mortes no primeiro ano de vacinação contra a COVID-19 ([WATSON et al., 2022](#)).

Com essa evolução positiva, as medidas previamente tomadas antes da vacinação passaram por uma flexibilização, trazendo medidas menos restritivas no uso de máscaras e no distanciamento social. Embora esse progresso venha acontecendo, atualmente há uma grande preocupação no avanço de variantes infecciosas e letais do SARS-CoV-2 ([VELLOSO, 2021](#)). Portanto, medidas preventivas e estratégias adotadas durante a pandemia se tornam ainda necessárias.

Uma das estratégias utilizadas pelos profissionais de saúde durante a pandemia consistiu no telemonitoramento, que se demonstrou como uma ferramenta eficaz para lidar com os desafios enfrentados pelos profissionais de saúde para obter acesso e monitoramento de casos confirmados para COVID-19, bem como para fornecer cuidados e orientações por parte das equipes de saúde para a população e atenção para casos com evolução no quadro clínico ([MIRANDA; FERREIRA; ANJOS, 2022](#)). Além disso, o processo de telemonitoramento traz a minimização da possibilidade de transmissão do vírus, por evitar a locomoção dos pacientes até o local de saúde, onde poderiam enfrentar locais movimentados durante a locomoção. Além disso, com o monitoramento remoto, os profissionais de saúde têm risco reduzido de contato físico com pacientes positivados. Tendo em vista a relevância da estratégia de telemonitoramento, foi realizada a construção de uma aplicação web que proporcione suporte para a equipe do projeto de iniciação científica intitulado [UFOP em Ação](#), detalhes do projeto estão descritos na Seção [2.2.4](#). Na ferramenta construída são oferecidos recursos para o acesso e monitoramento de pessoas com COVID-19 e pessoas que tiveram contato próximo a alguém positivado. Este trabalho de monografia aborda a apresentação de uma solução que foi desenvolvida após a identificação de problemas no processo de elaboração de elementos gráficos que compõem os dados do monitoramento realizado pela equipe

do projeto de iniciação científica UFOP em Ação. Os detalhes relativos à motivação, objetivos e desenvolvimento da solução são apresentados nas seções subseqüentes deste texto acadêmico.

## 1.1 Justificativa

O contato do profissional de saúde com pacientes diagnosticados com COVID-19 é de crucial importância para o acompanhamento da evolução do caso, além de ser fundamental para a condução de pesquisas sobre os efeitos da doença. Durante a pandemia, o contato remoto entre os profissionais da saúde e os pacientes se mostrou uma estratégia útil e relevante.

A obtenção de dados precisos e atualizados de cada caso confirmado é importante para a condução de pesquisas que visam entender os efeitos da COVID-19 em diferentes contextos. Uma ferramenta automatizada que possibilite a visualização gráfica e compacta dos dados gerados durante o processo de telemonitoramento pode ser útil na obtenção de diferentes percepções dos casos monitorados pelos profissionais da saúde durante a pandemia.

Nesse sentido, uma ferramenta automatizada que possibilite a visualização gráfica e resumida dos dados obtidos durante o processo de telemonitoramento pode fornecer suporte para a compreensão dos casos acompanhados pelos profissionais de saúde durante a pandemia. A análise desses dados pode contribuir para a tomada de decisões pelos profissionais da saúde, bem como para o desenvolvimento de pesquisas sobre os impactos da doença na sociedade.

## 1.2 Objetivos

O propósito geral desta monografia consiste em desenvolver uma ferramenta de suporte destinada a profissionais da área da saúde, com o intuito de permitir a visualização gráfica dos dados coletados durante o monitoramento. Essa ferramenta objetiva gerar esses gráficos de forma automatizada, possibilitando que os profissionais utilizem as informações visuais para embasar suas tomadas de decisão e pesquisas.

A fim de atender ao objetivo geral mencionado anteriormente, são estabelecidos os seguintes objetivos específicos:

- Utilizar tecnologias atuais de desenvolvimento web para obter uma interface amigável ao usuário final, de modo que o usuário não necessite de muito esforço para executar ações desejadas na aplicação final;
- Realizar testes na interface da aplicação para garantir a funcionalidade esperada de cada componente selecionado e construído;
- Realizar a segmentação dos dados obtidos através do monitoramento realizado pela equipe UFOP em Ação. Neste contexto, a segmentação consiste na divisão dos dados de forma a

isolar apenas as informações relevantes para a análise gráfica.

- Utilizar uma aplicação que empregue uma ferramenta segura e eficiente de manipulação e gerenciamento dos dados, para garantir que os dados sejam devidamente acessados quando requisitados;
- Empregar na aplicação gráficos com elementos visuais de fácil interpretação e que permita ao usuário interagir com cada um dos gráficos exibidos. Dessa forma, espera-se que a solução apresente uma variedade de tipos de gráficos com componentes visuais adequados para um *dashboard*<sup>1</sup>.

### 1.3 Organização do Trabalho

No Capítulo 1 são apresentados pontos importantes para que o leitor entenda a aplicabilidade da atuação da monografia, com a contextualização, a justificativa, os objetivos gerais e específicos do problema tratado. No Capítulo 2 está presente todo o embasamento bibliográfico, com trabalhos correlatos e fundamentação teórica. O Capítulo 3 apresenta as ferramentas utilizadas para a construção do *dashboard* proposto nesta monografia, além da estratégia utilizada para a adequação da plataforma entregue aos profissionais de saúde. Ainda no Capítulo 3 são passados resultados obtidos com o uso da metodologia tratada. E por fim, no Capítulo 4, se encontra a conclusão obtida com a realização desta monografia, tratando da hipótese levantada, e o alcance do objetivo geral deste trabalho, além da apresentação dos futuros trabalhos levantados.

---

<sup>1</sup> Um painel de controle, conhecido como *dashboard*, proporciona a visualização de um conjunto de informações por meio de uma apresentação visual composta por gráficos, tabelas, indicadores e outros elementos visuais.(TABLEAU, 2023)

## 2 Revisão Bibliográfica

Neste capítulo, são apresentadas duas seções. Na Seção 2.1, é apresentado o conteúdo teórico necessário para embasar os assuntos teóricos abordados nesta monografia. Na Seção 2.2, são especificados os trabalhos que possuem relação com o tema tratado nesta monografia e encontrados durante a revisão bibliográfica.

### 2.1 Fundamentação Teórica

Nesta seção, é apresentada uma revisão geral da literatura atual e das definições fundamentais para a compreensão do trabalho em questão.

Na Seção 2.1.1, o leitor será contextualizado ao problema da atual pandemia de COVID-19, com as principais estratégias de prevenção do avanço da pandemia. Em seguida, na Seção 2.1.2, são descritas as estratégias de telessaúde, visando esclarecer ao leitor a diferença e a relação entre a telessaúde, telemedicina e o telemonitoramento. Na Seção 2.1.3, é apresentado o papel do telemonitoramento ou monitoramento remoto no contexto da pandemia, bem como os desafios enfrentados pelos profissionais de saúde ao executar telemonitoramento. Uma importante fundamentação é trazida na Seção 2.1.4 a respeito da importância dos *dashboards* para a área da saúde, enquanto na Seção 2.1.5 são tratados os benefícios de uma aplicação web, para o leitor entender a relevância desse tipo de aplicação e se familiarize com alguns recursos disponíveis para o desenvolvimento web.

Somada à seção anterior, a Seção 2.1.6 traz uma visão geral sobre uma das linguagens mais utilizadas no processo de desenvolvimento web, e conta com a Subseção 2.1.6.1 que descreve a principal biblioteca JavaScript utilizada durante o desenvolvimento desta monografia, a biblioteca React. A Seção 2.1.8 trata sobre um tipo serviço muito utilizado para suporte no desenvolvimento *back-end* e sobre a ferramenta Firebase, que realiza o serviço discutido ainda na Seção 2.1.8. Por fim, na Seção 3.4 é apresentado informações sobre testes de fluxo completo e sobre o recurso Selenium.

#### 2.1.1 A pandemia de COVID-19

Os Coronavírus são uma família de vírus que podem causar doenças respiratórias em seres humanos, variando de casos leves a graves. Há três tipos de coronavírus conhecidos por causarem doenças graves em humanos: o SARS surgiu na China entre 2002 e 2003 causando uma epidemia de síndrome respiratória aguda grave; a síndrome respiratória do Oriente Médio, MERS, na Península Arábica em 2012; o SARS-CoV-2, que gerou a atual pandemia causada pela COVID-19, identificada pela primeira vez em Wuhan na China.

Uma semana após o alerta dos casos de pneumonia na cidade de Wuhan, em dezembro de 2019, a [OMS](#) confirmou a circulação do novo coronavírus como o causador da COVID-19. A organização declarou que o caso deveria ser tratado com emergência, devido ao grande surto de propagação da doença ([OPAS, 2022](#)).

A propagação intensa da doença é devida à sua fácil forma de transmissão, que ocorre principalmente pelo contato com gotículas de saliva expelidas por pessoas infectadas no ambiente. O contágio também é possível pelo toque em superfícies contaminadas ([VELLOSO, 2021](#)).

Diante da situação emergente de saúde, foram tomadas medidas não farmacológicas para a mitigação da fácil propagação do novo vírus, como o incentivo da higienização das mãos, o uso de máscaras em espaços compartilhados, distanciamento social e o impedimento de aglomerações. O atual novo coronavírus trouxe mudanças drásticas demograficamente, refletindo um alto número de óbitos até o mês de março de 2023.

Durante a pandemia, observaram-se altos picos de novos casos de COVID-19, que trouxe grandes desafios aos profissionais de saúde. Vários hospitais tiveram seus leitos ocupados por pessoas contaminadas, trazendo uma maior exposição dos profissionais da saúde ao vírus. Portanto, as medidas preventivas e uso de [EPI](#) adequados foram, e ainda são, indispensáveis nesse tipo de cenário enfrentado nos hospitais.

Os cuidados com os pacientes no ambiente hospitalar foram redobrados com o grande número de hospitalizações. Surgiu então a necessidade de ferramentas eficientes para apoiar os agentes da saúde, como o uso de Tecnologias da Informação e Comunicação ([TICs](#)), onde o telemonitoramento, a telessaúde e a telemedicina se mostraram como estratégias eficazes.

### **2.1.2 Telemonitoramento, telessaúde e telemedicina**

A inserção de inovação e tecnologia em contexto pandêmico, como o da COVID-19, mostrou-se relevante graças aos benefícios e facilidades que essa inserção proporciona. A telemedicina, a telessaúde e o telemonitoramento fazem parte de estratégias que envolvem a tecnologia. Essas estratégias são usadas por profissionais da saúde, sujeitos a diferentes treinamentos conforme a estratégia utilizada e com elas, é possível ter acesso aos usuários, de modo que seja possível oferecer cuidados e acompanhamento de diferentes casos, para tomar decisões associadas a cada um deles ([MIRANDA; FERREIRA; ANJOS, 2022](#)). Ainda assim, é importante destacar que o telemonitoramento, a telessaúde e a telemedicina não são sinônimos e possuem significados diferentes, com profundidades distintas nas aplicações tratadas em cada estratégia.

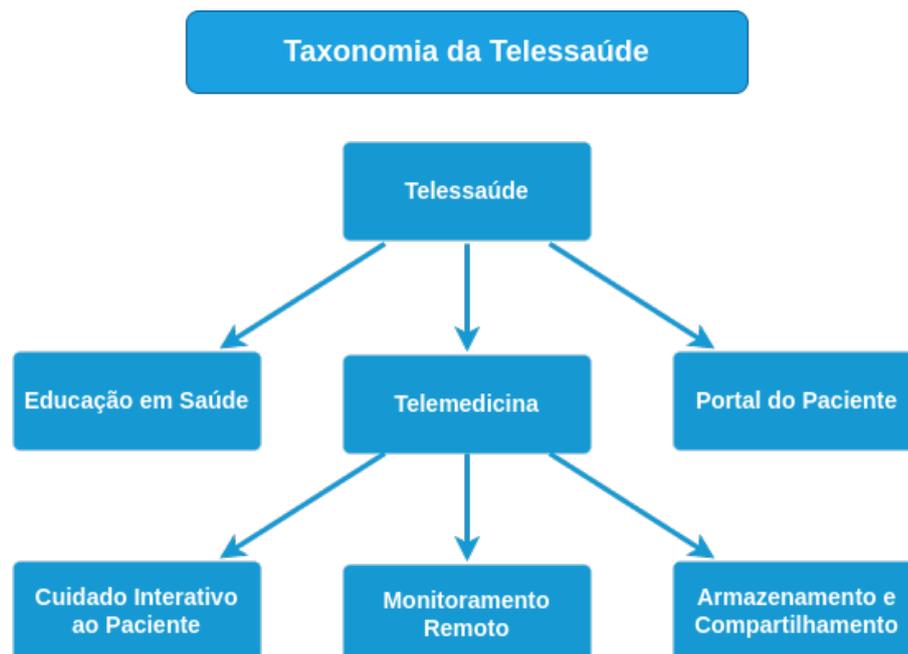
A telemedicina, praticada por profissionais de saúde, permite a interação com o paciente por meio de ferramentas de comunicação a distância, de modo que sejam feitas consultas remotas. Nessas consultas é possível serem feitas análises, diagnósticos e tratamento do paciente consultado. A telemedicina foi implantada na década de 70, onde foi vista a possibilidade de se realizar contato entre o médico e o paciente, sem a necessidade de locomoção. Essa possibilidade foi

pensada em um projeto no Hospital Geral de Boston, Massachusetts, que envolveu a combinação das áreas de Telecomunicação, Ciência da Computação e Saúde (URTIGA; LOUZADA; COSTA, 2004). O primeiro registro do uso da telemedicina foi durante a idade média, no período de pragas, onde um agente passava orientações e cuidados de um médico para a população (KIELING et al., 2021). Ao longo dos anos, os avanços tecnológicos nas ferramentas de comunicação remota têm trazido facilidades e qualidade no processo de atendimento, fazendo com que a interação entre o paciente e o médico contemple uma consulta de qualidade.

O telemonitoramento, ou monitoramento remoto, é feito por profissionais da saúde, não necessariamente por médicos formados, e essa estratégia faz parte de uma das modalidades de estratégias adotadas pela telemedicina (MULLER, 2022). Os pacientes que passam pelo telemonitoramento recebem monitoramento periódico da saúde, com a prestação de cuidados e orientações específicas para o caso acompanhado. Os dados gerados por um monitoramento são passados analisados por profissionais de saúde, e de acordo com esses dados é possível direcionar cada caso monitorado. Estratégias que utilizam TICs nos recursos usados para a saúde, entram como estratégias adotadas na Telessaúde, portanto, abrangem as metodologias tratadas na telemedicina e no telemonitoramento.

A telessaúde é composta por tecnologias que possibilitam serviços de saúde a distância, fazendo com que essa modalidade fique no topo. Existem diferentes tipos de tecnologias que entram como um tipo de telessaúde, que estão ilustrados na Figura 2.1, onde é possível visualizar a relação entre telemonitoramento e a telemedicina na hierarquia dos tipos de tecnologias da telessaúde (MULLER, 2022).

Figura 2.1 – Hierarquia das ferramentas da Telessaúde.



Fonte: Adaptado de Muller (2022).

### 2.1.3 O papel do telemonitoramento na pandemia de COVID-19 e os desafios enfrentados por profissionais de saúde

A grande crise de saúde causada pela pandemia de COVID-19 trouxe a necessidade de implantação de medidas para conter a transmissão do coronavírus e para garantir os cuidados de saúde.

O telemonitoramento se mostrou uma estratégia de relevância durante a pandemia. A adoção dessa estratégia o monitoramento remoto de pacientes para análise da evolução do caso e repasse dos cuidados necessários segundo o caso tratado. O colapso na saúde e nos ambientes hospitalares trouxe grandes desafios, principalmente para as equipes da saúde que estiveram na linha de frente da pandemia. Logo, a utilização do telemonitoramento por atuantes e estudantes da área da saúde se tornou uma realidade durante a pandemia de COVID-19 para a redução da demanda de atendimentos nos locais de saúde (BATISTA et al., 2021).

A implantação dos instrumentos de telessaúde trouxe diversos benefícios no contexto pandêmico, como o da COVID-19, mas também trouxe desafios, por se tratar de uma implantação de forma rápida (CAETANO et al., 2020). As principais dificuldades apresentadas no relato de profissionais de saúde que realizaram o telemonitoramento envolve fatores de adaptação e dificuldade do processo de implantação de uma modalidade nova no processo de monitoramento das pessoas atendidas por esses profissionais. A falta de infraestrutura foi um dos maiores desafios apontados na literatura ao incluir ferramentas de telessaúde, e não necessariamente se refere à falta de recursos financeiros, mas investimentos em treinamentos, em dispositivos e em internet de qualidade para o uso adequado das TICs no processo de telemonitoramento.

Nos relatos encontrados na literatura, os profissionais de saúde afirmaram ter problemas em se comunicar com o paciente, pois muitas das vezes a internet de ambos caía no processo de telemonitoramento, principalmente quando um ou todos os participantes do telemonitoramento se encontravam em um ambiente rural. Além disso, o investimento na qualidade de treinamento para os profissionais trouxe dificuldades na execução das tarefas de telemonitoramento, trazendo dificuldades no atendimento devido à baixa capacitação (MIRANDA; FERREIRA; ANJOS, 2022). O uso de plataformas que auxiliam no monitoramento remoto trouxe muitas facilidades e praticidades para os profissionais de saúde, mas a dificuldade no manuseio dessas plataformas também foi abordada. Portanto, é necessária a usabilidade, que consiste na facilidade do usuário em executar seus objetivos por meio de um sistema que possui passos para conclusão dos objetivos, sendo esses passos facilmente memorizados e satisfatórios (LIMA et al., 2021).

### 2.1.4 A importância dos *dashboards* para gestão de saúde

O dia a dia de um profissional da área da saúde exige esforço e dedicação para executar tarefas complexas e difíceis (FERNANDES et al., 2020). Durante a pandemia, houve um aumento significativos no número de casos a serem tratados, com o volume de novos pacientes com

COVID-19, além da necessidade de autoproteção. Portanto, durante o longo período de combate ao novo coronavírus, esses profissionais tiveram seus dias sobrecarregados com medidas de cuidado tanto para a própria saúde quanto para a dos pacientes atendidos.

A análise para compreender os diferentes cenários dos possíveis casos em um contexto pandêmico, a gestão dos cuidados e a pesquisa na saúde são pontos críticos que não podem ser deixados de lado e que são melhor executados com o auxílio de ferramentas de suporte para os profissionais da saúde. Nesse sentido, a *dashboard* surge como uma excelente ferramenta auxiliar, permitindo a investigação de diversas informações por meio de uma visualização em diferentes tipos de gráficos (FERNANDES et al., 2020).

Um *dashboard*, ou um painel de bordo, traz informações condensadas de um conjunto de dados extraídos de uma base. Nele, é possível ter um norte sobre as decisões a serem tomadas ao examinar dados que são apresentados de forma clara e objetiva, usando parâmetros e métricas (INTELIPOST, 2023). A Figura 2.2 ilustra um exemplo de *dashboard*, que possibilita a visualização geral de dados por meio de gráficos.

Figura 2.2 – Exemplo de um *dashboard*.



Fonte: Imagem retirada de Hoinaski (2023).

Durante a pandemia, foram construídos diferentes tipos de *dashboards* e distribuídos ao redor do mundo. Eles auxiliam no entendimento dos cenários correntes da pandemia de COVID-19.

No Brasil, houve instituições que desenvolveram *dashboards* para uma visão estadual e geral da pandemia, de forma que instituições de saúde, de ensino e empresas acompanhassem o estado de saúde da população com uma visão macro e micro da situação atual, possibilitando tomadas de decisões em cima dos levantamentos gerados pelos *dashboards* disponibilizados (FERNANDES et al., 2020)

Para um grupo de profissionais da saúde que realiza o monitoramento de casos de COVID-19, um *dashboard* pode ser uma ferramenta eficiente de apoio para análise dos dados coletados pela equipe, auxiliando na tomada de decisões e na realização de pesquisas com base nas informações apresentadas.

### 2.1.5 Benefícios e tecnologias atuais de uma aplicação web

Um sistema web fornece a possibilidade da integração de uma ampla gama de ferramentas introduzidas para o desenvolvimento web — em constante evolução — trazendo flexibilidade e aprimoramentos no sistema desenvolvido. Portanto, novas soluções podem ser adicionadas aos sites de tal forma que traga maior facilidade de uso e melhores desempenhos na aplicação.

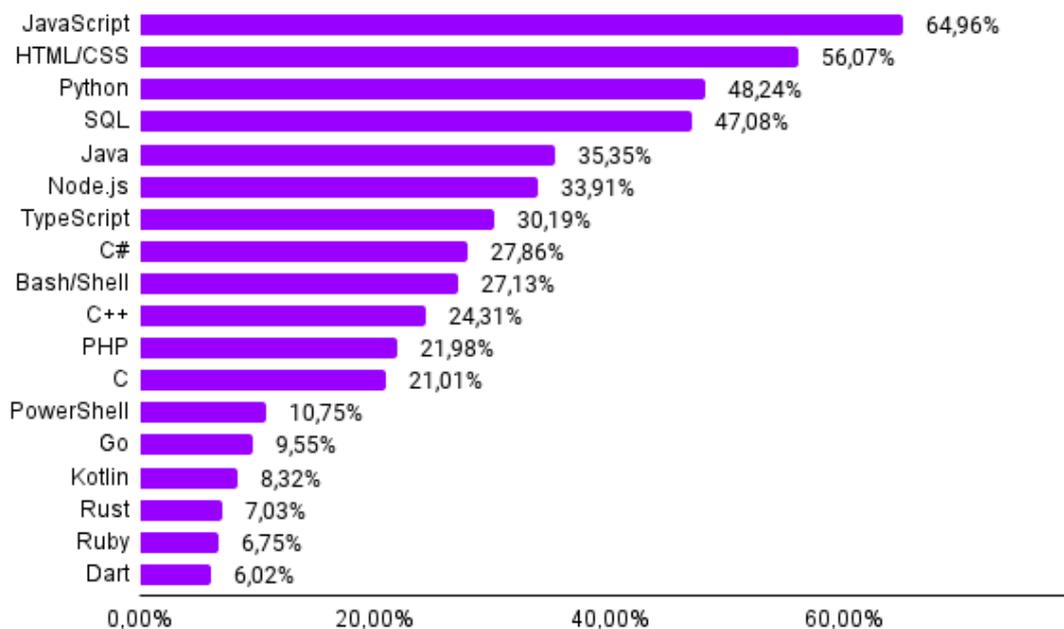
Para acessar uma aplicação web, basta ter um dispositivo com acesso à internet e ter um navegador que suporte a aplicação, sem dependência da qualidade do dispositivo e do sistema operacional. Além disso, o design responsivo de aplicações web possibilita maior facilidade no uso em dispositivos móveis.

Existem várias linguagens para a programação web, cuja escolha da linguagem varia para cada tipo de solução tratada. Dentre as linguagens, temos JavaScript, Python e PHP como as mais populares. Devido à constante evolução das ferramentas utilizadas em cada uma das linguagens de programação, ao longo dos anos é vista uma mudança na escolha de cada linguagem ou até na possibilidade de integração de diferentes linguagens para um serviço web. A Figura 2.3 ilustra, em porcentagem, a popularidade de cada linguagem voltada para o desenvolvimento web no ano de 2021. Os dados apresentados são resultados de uma pesquisa realizada pela plataforma StackOverflow, uma das plataformas para fóruns de dúvidas mais utilizada entre os desenvolvedores.

Segundo uma pesquisa realizada pelo StackOverflow, a linguagem JavaScript teve seu nono ano consecutivo como a linguagem mais usada em 2021, tendo seu papel principal no desenvolvimento web. Essa linguagem é normalmente utilizada com a linguagem de estilo *Cascading Style Sheet*, conhecida como CSS e a linguagem de marcação de hipertexto, conhecida como HTML. Em uma aplicação web, o HTML é utilizado para a estruturação dos elementos na tela, o CSS é responsável pela estética dos elementos na tela e o JavaScript traz dinamicidade na tela com movimentos e ações.

Embora seja possível criar estilos e interações visuais apenas com JavaScript, CSS e HTML, o crescente uso de aplicações web tem levado ao desenvolvimento e aprimoramento de

Figura 2.3 – Popularidade das linguagens de programação no ano de 2021.



Fonte: Adaptado de [stackoverflow \(2021\)](#).

diversas bibliotecas e *frameworks* em JavaScript. A escolha da biblioteca ou framework a ser utilizado varia conforme as necessidades específicas de cada aplicação.([STACKOVERFLOW, 2021](#)).

A escolha da biblioteca ou do *framework* deve considerar a relevância dos dados a serem tratados, tanto gerencialmente quanto visualmente. Além disso, uma aplicação web que trata de distintos contextos pode exigir a utilização de componentes visuais diversos para uma boa compreensão por parte do usuário, destacando a necessidade de uma tecnologia com uma ampla possibilidade de integração de novos componentes, reutilização de código e facilidade na manutenção. Para um site que aborda telessaúde, é fundamental ter uma navegação fácil, alta acessibilidade, baixa taxa de rejeição e segurança nos dados ([SYNTACTICS, 2022](#)).

### 2.1.6 Uma visão geral das bibliotecas e frameworks JavaScript

Frameworks e bibliotecas possuem códigos escritos por outros desenvolvedores, e esses códigos podem ser reutilizados e tratados em diferentes tipos de soluções, permitindo que funções e conjuntos de códigos sejam usados conforme a necessidade do desenvolvedor ([GEEKSFOR-GEEKS, 2022](#)). No entanto, as bibliotecas e *frameworks* têm abordagens diferentes entre si. As bibliotecas oferecem funções e classes que podem ser usadas e combinadas em qualquer parte do código, enquanto um *framework* permite que realize tarefas, de modo que o desenvolvedor forneça propriedades na estrutura do código de acordo com suas necessidades. Isso faz com que o *framework* cuide do fluxo de execução com base nas propriedades passadas pelo desenvolvedor

([FREECODECAMP, 2022](#)).

A linguagem de programação JavaScript é interpretada e orientada a objeto, conhecida como uma linguagem *scripting* dinâmica para a web e possui grandes potencialidades que levou a sua popularização para o manejo de comportamento de aplicações web ([MDN, 2022](#)).

### 2.1.6.1 A Biblioteca React

Uma das bibliotecas JavaScript mais utilizadas entre os desenvolvedores para o desenvolvimento front-end é a biblioteca React, por possibilitar a criação de interfaces de usuário divididas em partes por meio da manipulação de componentes que podem ser reutilizados ([GEEKSFORGEEKS, 2022](#)).

Os componentes agem como funções JavaScript, recebendo dados de entrada para renderizar na tela o componente React correspondente. O componente pode ser um botão, uma tabela, um menu de navegação, ou outros elementos que podem ser visualizados na tela. Por meio da biblioteca React são renderizados, de maneira eficiente, apenas aqueles componentes que sofreram alguma mudança nos dados, sendo assim, o desenvolvedor deve associar eventos condizentes às mudanças de estado para cada um dos componentes em tela ([REACT, 2022](#)).

Na Figura 2.4 é possível visualizar uma estrutura chamada **JSX**. Segundo a documentação oficial do React, o JSX é uma extensão de sintaxe para JavaScript que possibilita a escrita de códigos que se assemelham a marcações **HTML** ou **XML**. Embora opcional, essa sintaxe é comumente utilizada nas aplicações que utilizam React. Ao utilizar JSX, os comandos JavaScript podem ser escritos em chaves, como mostrado no exemplo da Figura 2.4

Ademais, a documentação ressalta que o uso do JSX facilita a escrita e a manutenção de códigos em React, permitindo que os desenvolvedores expressem visualmente a estrutura de seus componentes e tornando o código mais legível e compreensível.

Figura 2.4 – Exemplo de um código JSX.

```
function App() {  
  const text = 'Hello World';  
  
  return (  
    <div className="App">  
      <p> {text} </p>  
    </div>  
  );  
}
```

**Fonte:** Elaborado pela autora.

Por meio da sintaxe JSX podemos criar com React dois tipos de componentes utilizando-se classes ou funções JavaScript. Esses componentes são renderizados utilizando o método `render()`, o qual recebe o componente que será renderizado por parâmetro. Os componentes criados por meio de classe, recebem o nome de *Class Component* e aqueles construídos por meio de função, são identificados pelos *Functional Components*. A Figura 2.5 ilustra duas formas de criação de um componente. Na Figura 2.6(a) é exemplificada a criação do componente por meio de classe, já na figura 2.6(b), é exemplificado a criação por meio de função JavaScript.

O React é uma biblioteca de código aberto, criada em 2011 pelo Facebook. Devido à potencialidade dessa biblioteca, seu ecossistema cresceu muito ao longo dos anos. Por isso, atualmente existem diversas bibliotecas de componentes React disponíveis para serem usadas. A biblioteca MUI é uma delas, criada pelo Google, que possui componentes baseados no *Material Design*. Esses componentes são simples e podem ser personalizados conforme as necessidades do desenvolvedor (RETOOL, 2022). Além da biblioteca MUI, outras também são amplamente utilizadas, como as bibliotecas *Ant Design* e *Bootstrap*. Cada uma possui um conjunto de componentes prontos para serem utilizados em uma aplicação, trazendo maior agilidade no processo de criação de uma interface amigável e moderna. Na Figura 2.6, é ilustrado um componente disponível pela biblioteca MUI, com um exemplo de uso. Esse componente é muito utilizado em interações do tipo *on-off*. O comportamento definido para os estados “ligado” e “desligado” são definidos por parte do desenvolvedor que utiliza o componente.

### 2.1.7 Bibliotecas para a criação de gráficos no JavaScript

Como citado anteriormente, o JavaScript é uma das linguagens mais populares entre os desenvolvedores e, graças a isso, existem diversas bibliotecas disponíveis que podem ser usadas com essa linguagem. Referente à criação de gráficos, existem várias bibliotecas, cada uma com suas próprias vantagens e desvantagens. A escolha da biblioteca varia segundo as necessidades de cada desenvolvedor. Entre as bibliotecas mais populares, podemos destacar:

- Chart.js: biblioteca de código aberto, onde é possível ter gráficos com animações e responsivos. A biblioteca disponibiliza oito tipos de gráficos (CHART.JS, 2023);
- Google Charts: uma biblioteca gratuita para a criação de gráficos, bastante flexível, e oferece bastante opções de gráficos, incluindo mapas, gráficos e diagramas (CHARTS, 2023);
- D3.js: permite a visualização de dados de uma maneira gráfica e interativa. Os dados passados para funções dessa biblioteca são dinamicamente associados em representação gráfica (D3.JS, 2023);
- Plotly.js: uma biblioteca de alto nível para a construção de gráficos, com certa de quarenta tipos de gráficos. Segundo a documentação da biblioteca, foi criada utilizando as bibliotecas

Figura 2.5 – Comparação entre Class Component e Functional Component.

```
1 import React, { Component } from "react";
2
3 class ClassComponent extends Component {
4   constructor(){
5     super();
6     this.state={
7       count :0
8     };
9     this.increase=this.increase.bind(this);
10  }
11
12  increase(){
13    this.setState({count : this.state.count +1});
14  }
15
16  render(){
17    return (
18      <div style={{margin:'50px'}}>
19        <h1>Hello</h1>
20        <h2> {this.state.count}</h2>
21        <button onClick={this.increase}> Add</button>
22
23      </div>
24    )
25  }
26 }
27
28 export default ClassComponent;
```

(a) Exemplo da criação de um *Class Component*.

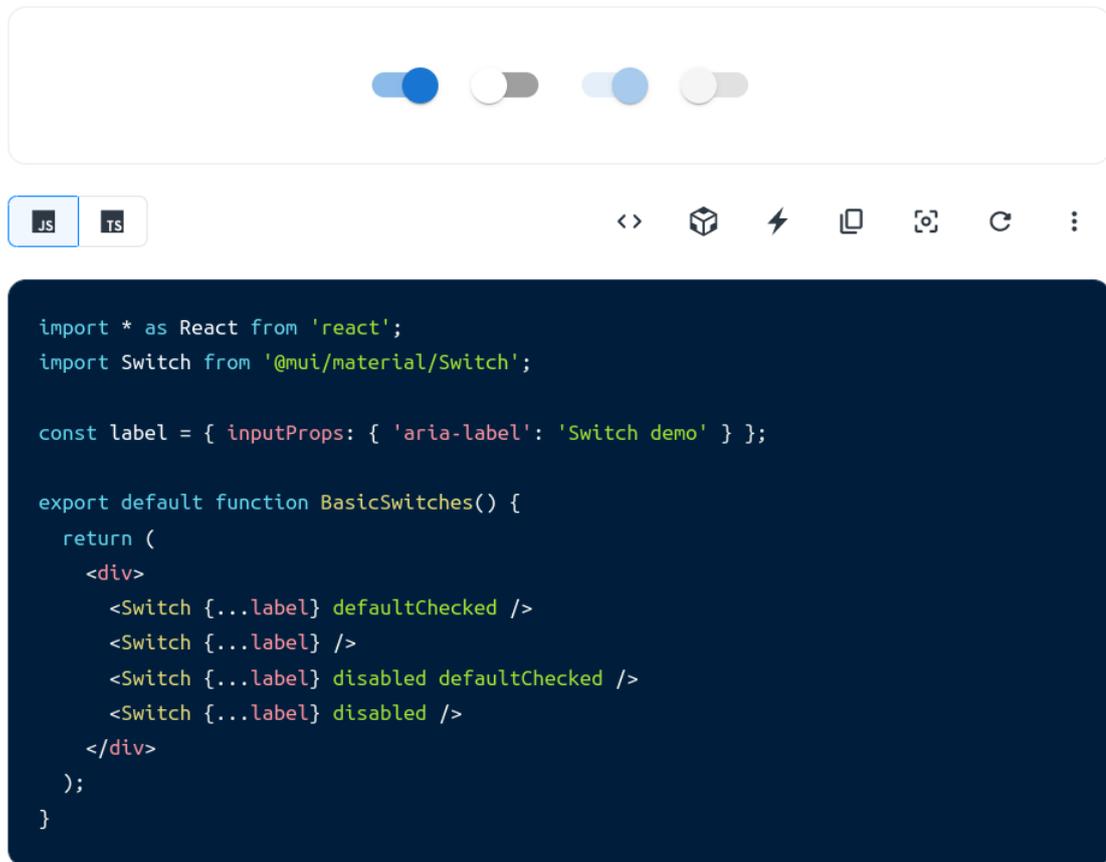
```
1 import React, { useState } from "react";
2
3 function FunctionalComponent(){
4   const [count, setCount] = useState(0);
5
6   const increase = () => {
7     setCount(count+1);
8   }
9
10  return (
11    <div style={{margin:'50px'}}>
12      <h1>Hello </h1>
13      <h2>{count}</h2>
14      <button onClick={increase}>Add</button>
15    </div>
16  )
17 }
18
19 export default FunctionalComponent;
```

(b) Exemplo da criação de um *Functional Component*.

**Fonte:** Captura de tela de um código adaptado de [GeeksForGeeks \(2023\)](#).

Figura 2.6 – Exemplo de utilização de um componente pronto, disponível na biblioteca MUI.

## Interruptores básicos



Fonte: Captura de tela da documentação de [MUI \(2023\)](#).

D3.js e Stack.gl, trazendo gráficos 3D, estatísticos e mapas SVG ([PLOTLY, 2023](#)).

### 2.1.8 *Back-end as a Service* e a ferramenta Firebase

Processo de segurança, autenticação, e segurança de dados, assim como outros processos *back-end* podem ser tarefas difíceis de serem executadas a depender da complexidade da aplicação, podendo gerar custos em termos de tempo e desenvolvimento.

O **BaaS** é um modelo de computação em nuvem que traz serviços para as tarefas *back-end*. Por meio de um Baas, é possível realizar o armazenamento de dados, autenticação, integrações, API, entre outros. Com isso, o desenvolvedor pode se concentrar no desenvolvimento *front-end*, integrando seus dados com um serviço BaaS sem se preocupar com tarefas complexas da infraestrutura dos seus dados.

O Firebase <sup>1</sup> é um modelo de computação em nuvem que oferece BaaS. Um dos princi-

<sup>1</sup> Segundo a documentação oficial do Firebase, essa ferramenta oferece suporte para o desenvolvimento de *software* e possui uma variedade de recursos que permitem a criação de sistemas de alta qualidade ([FIREBASE, 2023](#)). O Firebase é considerado um **BaaS**, que disponibiliza uma infraestrutura para o funcionamento interno de um

país serviços pelo Firebase são: banco de dados em tempo real; autenticação, com suporte de autenticação, dados via e-mail e senha, do número de telefone, e de redes sociais; hospedagem; funções de gerenciamento dos dados. O Firebase não é totalmente gratuito e contém planos para atender diferentes níveis de aplicação. Nesta monografia, foi utilizado o banco de dados real do Firebase, onde são utilizadas funções oferecidas pelo serviço, que realizam tarefas de autenticação, inclusão, exclusão, atualização e inserção de dados em tempo real.

### 2.1.9 Testes *End-To-End* e a ferramenta Selenium para ferramenta de automação

Uma aplicação web é formada por um conjunto de componentes visuais que podem ser interativos. Por meio desses elementos, o usuário pode realizar diferentes ações para obter um fluxo associado ao objetivo do usuário, respeitando os requerimentos necessários para completar o seu fluxo na aplicação. Os testes *End-To-End* são compostos por passos para testar se um fluxo corresponde ao esperado, onde cada passo sai de um estado ao outro com um resultado esperado do passo anterior (BOSE, 2023).

Os testes *End-To-End* feitos manualmente podem demandar tempo e podem causar falhas humanas, portanto, existem ferramentas para sua automação.

O Selenium<sup>2</sup> é uma das ferramentas mais populares para a execução de testes *End-To-End*, sendo uma biblioteca de código aberto poderosa, que pode ser integrada a diferentes linguagens de programação, como Java, Perl, JavaScript, PHP, Python, C#, entre outras. Com o Selenium é possível fazer diferentes testes, como: teste de desempenho da aplicação observando como os processos estão sendo executados, com análise de tempo de carregamento e resposta das interações possíveis em cada passo do teste; teste de interface, para validar o funcionamento dos componentes visuais em tela da aplicação para o usuário; testes de regressão, que permite verificar se uma modificação não afetou outras partes da solução testada.

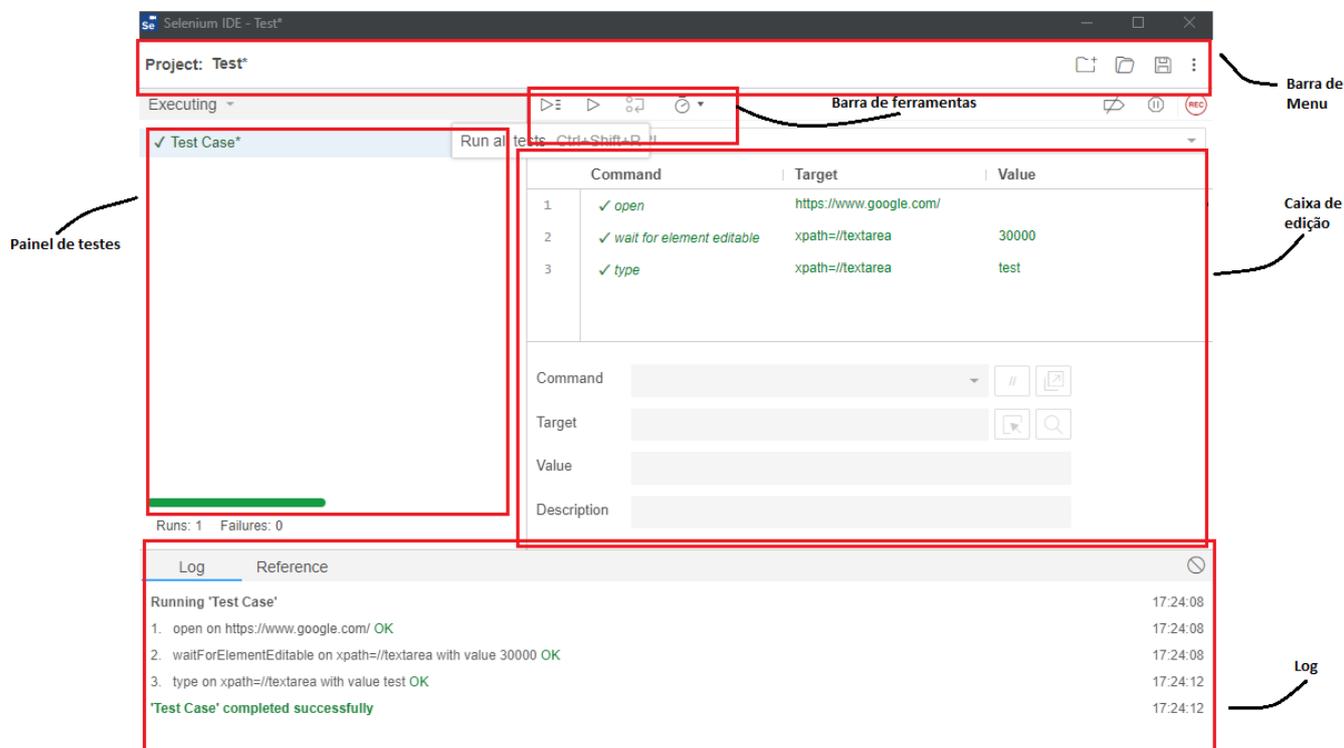
O Selenium oferece uma ferramenta chamada Selenium IDE, que possibilita a automação de testes de forma rápida, eficiente e com menos esforço. Com esta ferramenta, é possível criar testes sem a necessidade de conhecimento em programação, permitindo a gravação, reprodução e personalização de casos de testes que seguem uma sequência de passos baseados em suas respectivas propriedades. Na Figura 2.7 é possível observar a interface do Selenium IDE, a ferramenta utilizada neste trabalho.

---

*software* (REMESSA ONLINE, 2023).

<sup>2</sup> O Selenium é uma ferramenta de código aberto que permite a criação de testes automatizados em cima de aplicações web. Segundo a documentação do Selenium, essa ferramenta é uma suíte composta por vários recursos para automatizar interações com o usuário em um navegador (SELENIUM, 2023).

Figura 2.7 – Tela principal do Selenium IDE



Fonte: Adaptado de [Testbytes \(2023\)](#).

## 2.2 Trabalhos Relacionados

Nesta seção, é feita uma descrição dos trabalhos relacionados à monografia atual, com uma discussão a respeito do trabalho relacionado selecionado. O trabalho descrito na Seção 2.2.1 trata de um estudo sobre o grau de adaptação do telemonitoramento da COVID-19 em um município do sul da Bahia. Esse grau é analisado por meio do relato de experiência de profissionais da saúde que realizaram o telemonitoramento nessa região. Já a Seção 2.2.2 apresenta um trabalho correlato sobre o desenvolvimento *dashboard* usado para análise de insumos para a pandemia de COVID-19. Na Seção 2.2.3, é descrito um trabalho correlato sobre um relato de experiência do telemonitoramento realizado no estado do Paraná. Na Seção 2.2.4, é descrito o projeto de iniciação científica UFOP em Ação, incluindo o que foi realizado durante essa iniciação. Vale ressaltar que a presente monografia trata de uma solução para o problema de geração e análise gráfica de dados obtidos durante o telemonitoramento, utilizando uma aplicação web desenvolvida na iniciação. Na Subseção 2.2.4.1, é descrita uma ferramenta utilizada durante o projeto de iniciação que serviu para a criação de um protótipo no Figma, uma ferramenta explicada ainda na Seção 2.2.4.2. Por fim, na Seção 2.2.4.2, discute-se a implementação web realizada durante o projeto.

### 2.2.1 Incorporação do telemonitoramento da COVID-19 em um município do sul da Bahia: obstáculos e facilidades

O estudo de [Miranda, Ferreira e Anjos \(2022\)](#) procura descrever as dificuldades e facilidades da inclusão do telemonitoramento no município de Una, localizado no sul da Bahia, durante o período da pandemia de COVID-19.

No cenário de telemonitoramento, a equipe de Atenção Primária à Saúde (APS) realizou o acompanhamento por meio de ligações telefônicas, mensagens via *WhatsApp* e visitas domiciliares, dependendo da disponibilidade dos usuários para a utilização dessas ferramentas de monitoramento remoto.

O monitoramento era feito dia após dia, com o intuito de acompanhar o estado de saúde dos pacientes monitorados e fornecer os cuidados necessários, conferindo orientações específicas para cada caso em tratamento.

O pesquisa em [Miranda, Ferreira e Anjos \(2022\)](#) teve seu foco na análise da experiência do monitoramento feito por profissionais da saúde, a fim de tratar perspectivas positivas e negativas do uso dessa estratégia pela equipe da APS, além de avaliar o nível de incorporação das TICs no monitoramento de pacientes com COVID-19 no período de 2020 a 2021.

Para analisar o nível de incorporação do telemonitoramento, foi utilizada a técnica de análise temática proposta por [Bardin \(2016\)](#)<sup>3</sup>. Essa técnica foi aplicada aos dados coletados a partir das entrevistas realizadas com a equipe da APS, passando pelas etapas de ordenação, classificação e análise de dados definidas por [Bardin \(2016\)](#). Os dados foram adquiridos por meio de questionários e as respostas foram classificadas segundo os temas abordados. Essa classificação foi conduzida considerando a organização municipal, vigilância em saúde, assistência ao paciente e educação em saúde durante o processo de análise ([MIRANDA; FERREIRA; ANJOS, 2022](#)).

Os resultados analisados pelos autores em [Miranda, Ferreira e Anjos \(2022\)](#) constataram grau de incorporação do telemonitoramento de 69,20% de adequação. As facilidades e dificuldades na utilização de TICs durante o telemonitoramento foram extraídas por meio da análise do conteúdo das entrevistas, considerando cada tipo de classificação. Durante essas entrevistas, houve relatos passados pelos profissionais da saúde destacando as experiências segundo as classificações levantadas. Abaixo seguem relatos retirados da análise qualitativa, associados a cada classificação conduzida na pesquisa por [Miranda, Ferreira e Anjos \(2022\)](#). Cada relato é identificado por um número que corresponde à entrevista realizada. Ao final de cada relato, é indicada a identificação da entrevista a que ele pertence.

No quesito organizacional municipal foi passado o seguinte relato:

<sup>3</sup> Laurence Bardin é um autor que desenvolveu uma técnica chamada Análise de Conteúdo, usada para analisar pesquisas qualitativas. O autor divide a técnica em várias etapas para analisar conteúdos como documentos, entrevistas, artigos e pesquisas textuais.

*“Se tivéssemos uma qualidade melhor de internet e um maior avanço no uso dessas tecnologias, teríamos um ganho muito grande. [...] Tivemos várias situações onde a tecnologia nos ajudou no diagnóstico de pacientes mais graves. Nosso contato era basicamente por telefone. Assim, foi importante tanto para nós, profissionais de saúde, quanto para a própria família, que não podia estar com o paciente.” (E02)*

A respeito da vigilância epidemiológica, foi relatada uma dificuldade relacionada à conciliação entre o atendimento presencial e o atendimento remoto. O relato a seguir aborda essa questão:

*“No começo da pandemia tinha um fluxo muito grande de pacientes, já cheguei a monitorar 40 pacientes. Havia um fluxo grande de pacientes durante o dia na unidade de saúde e eu não conseguia fazer o monitoramento mesmo com o apoio da equipe. Em alguns momentos precisei ligar à noite para os pacientes.” (E05).*

O uso das TICs foi identificado como um facilitador, trazendo maior praticidade no processo de monitoramento e tomadas de decisões. Além disso, o telemonitoramento feito pela equipe de saúde se indicou ser uma ferramenta que otimiza o uso do tempo, ajudando no trabalho dos profissionais da equipe. Os seguintes relatos ressaltam os pontos positivos das TICs e do telemonitoramento:

*“Achei que o uso do telefone trouxe uma facilidade muito grande. Imagine como seria ter que sair para fazer busca ativa de vários casos suspeitos, ir de casa em casa, com toda a sobrecarga que os enfermeiros tinham. A equipe de saúde já estava exausta.” (E04)*

*“Sem TICs seria muito mais complexo, porque não podíamos estar adentrando as casas. Foi a forma que utilizamos para manter o acompanhamento desses pacientes. Tivemos vários casos, principalmente de pacientes de zona rural, que a tecnologia foi essencial para garantir que o serviço chegasse até o paciente.” (E07)*

*“O telemonitoramento ajudou bastante, facilitou também o nosso serviço porque enquanto fazíamos esse contato tinham pacientes em sala de espera, também aguardando atendimento. Então, conseguimos investigar o paciente que estava no domicílio sem deixar de atender os que estavam aqui esperando. Assim o tempo foi otimizado, além de promover à equipe o acesso seguro a esses pacientes.” (E11)*

Um dos pontos negativos levantados pelos profissionais de saúde foi o processo de capacitação que receberam para o telemonitoramento. O seguinte relato trata sobre a deficiência no processo na Educação em Saúde:

*“A capacitação profissional realizada foi muito superficial. Acho que todo mundo acabou aprendendo naturalmente. A força da pandemia foi fazendo com que nos esforçássemos mais. Houveram algumas explicações, mas foram superficiais também porque não se sabia muito”.* (E02)

Os resultados da pesquisa em [Miranda, Ferreira e Anjos \(2022\)](#) possibilitam uma visão geral de diferentes ângulos observados na experiência de profissionais da saúde no processo de aplicação do telemonitoramento. É relevante salientar que essa abordagem traz consigo desafios e vantagens que estão diretamente relacionados à infraestrutura utilizada.

Os recursos utilizados, como telefones e ferramentas de mensagem instantânea, trouxeram facilidades no contato com os pacientes. No entanto, é importante que tanto o profissional quanto o paciente tenham acesso aos instrumentos necessários para o monitoramento remoto. Além disso, é possível concluir que uma capacitação adequada da equipe é essencial para garantir que o profissional realize o acompanhamento sem enfrentar muitas complicações.

A modalidade de comunicação utilizada para interagir com os pacientes monitorados no estudo correlato é semelhante àquela usada pelos usuários da aplicação web abordada nesta monografia, em que são realizadas ligações telefônicas com os indivíduos monitorados. No entanto, existem diferenças nas ferramentas empregadas para registrar e acompanhar os dados monitorados. Nesta monografia, todo o ciclo de monitoramento é realizado por meio de uma aplicação web, o que contribui para tornar o processo mais prático e eficiente.

### **2.2.2 Um *dashboard* para visualização de insumos relacionados à Covid-19**

O trabalho em [Rocha, Nascimento e Sousa \(2021\)](#) trata da construção de um *dashboard* utilizando a base de dados fornecida pelo Governo Federal sobre os insumos para o combate à COVID-19. A finalidade dessa construção está no fornecimento de uma ferramenta que traga informações sintetizadas sobre os insumos disponíveis diante de um cenário pandêmico, tanto as instituições como a população em geral na tomada de decisões.

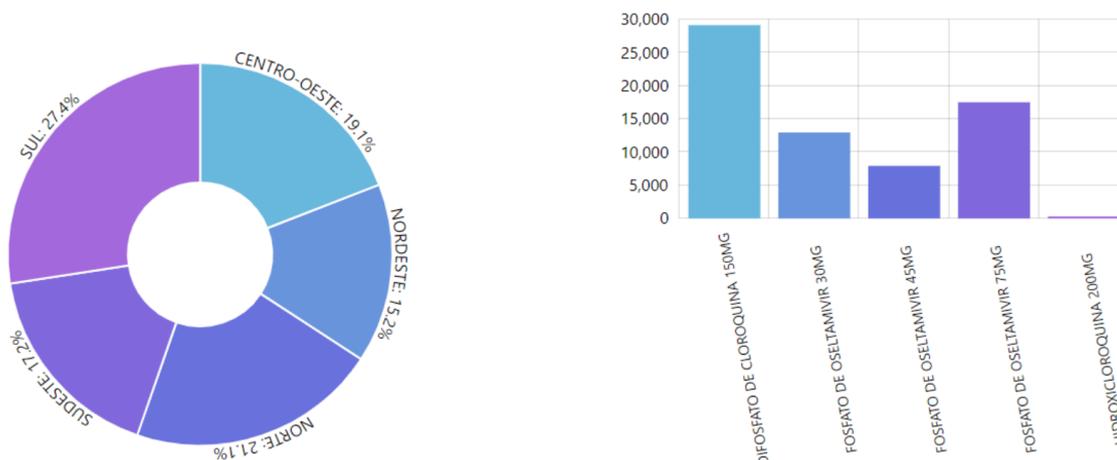
A distribuição de equipamentos para o combate da pandemia, como máscaras, álcool, luvas, proteção facial, aventais, medicamentos e equipamentos de saúde em geral, é de extrema importância na luta contra a COVID-19. O trabalho relata a importância da visibilidade dos insumos distribuídos pelo Governo Federal devido à importância do monitoramento de ações e estratégias feitas pelo governo para os municípios brasileiros.

Com isso, os autores de [Rocha, Nascimento e Sousa \(2021\)](#) propuseram a construção de um *dashboard* utilizando tecnologias web e banco de dados para análise e visualização dos dados de fornecimento de insumos médicos às Unidades Federativas (UFs) brasileiras.

O *dashboard* construído é composto por diversos tipos de gráficos que fornecem informações variadas sobre a disponibilidade de insumos nos estados brasileiros. Além disso, há um

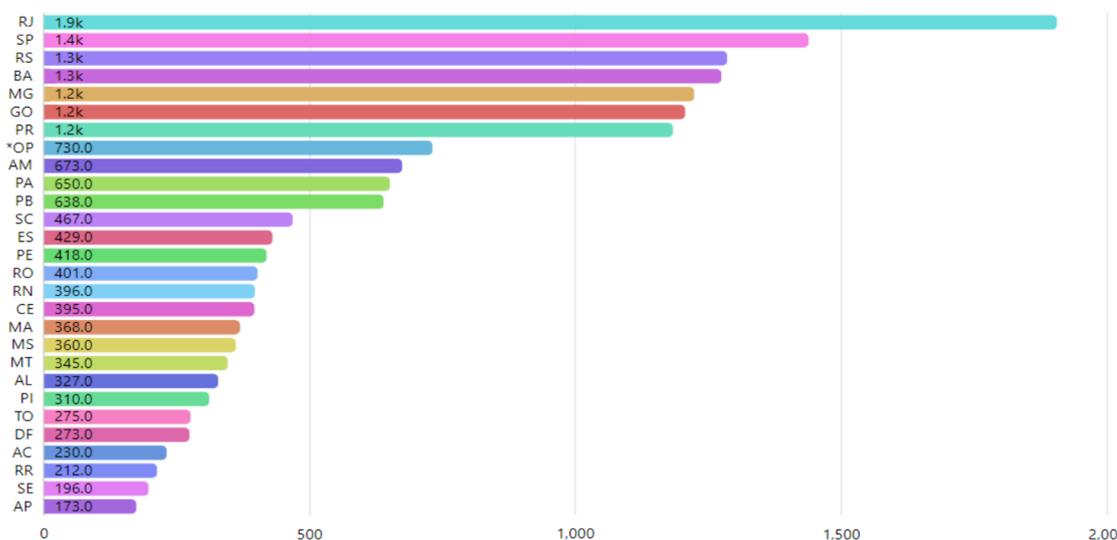
gráfico específico que apresenta dados sobre a quantidade de respiradores por país. As Figuras 2.8 e 2.9 mostram exemplos de gráficos gerados pela *dashboard* do trabalho em Rocha, Nascimento e Sousa (2021).

Figura 2.8 – Insumos de medicamentos por região no gráfico de rosca e por estado no gráfico de barras. Gráficos gerados no trabalho de Rocha, Nascimento e Sousa (2021).



Fonte: Captura de tela da imagem de Rocha, Nascimento e Sousa (2021).

Figura 2.9 – Gráfico horizontal com informações dos insumos de respiradores em cada estado, construído no trabalho de Rocha, Nascimento e Sousa (2021).



Fonte: Captura de tela da imagem de Rocha, Nascimento e Sousa (2021).

O trabalho destaca as tecnologias usadas para o desenvolvimento do *dashboard* que veio por meio de uma solução web. A estrutura básica do sistema foi implantada utilizando JS, CSS e HTML. Outras tecnologias também foram utilizadas para o bom funcionamento e interação com o *dashboard*. Foram utilizados:

- *Node JS*: uma tecnologia para execução de comandos JS;

- *Express*: uma biblioteca para gerenciar requisições [HTTP](#);
- *Handlebars*: um *template engine* que permite a execução de comandos personalizados em HTML, tais como estruturas condicionais e de repetição, diretamente a partir do arquivo HTML;
- *Body-parser*: um *middleware* para gerenciamento de formulários usando [XML](#) e JSON por meio de requisições HTTP;
- *Amcharts*, *Google Charts* e *Leaflet*: bibliotecas JS para criação de diferentes tipos de gráficos.

Para o gerenciamento dos dados tratados no website foi utilizado um Sistema de Gerenciamento de Banco de Dados ([SGBD](#)), o Mysql, sido utilizado por trazer alto desempenho, facilidade de uso e segurança. A manipulação dos dados foi feita com a versão relacional do Mysql, na linguagem Structured Query ([SQL](#)).

O *dashboard* implementado em [Rocha, Nascimento e Sousa \(2021\)](#) possui uma abordagem semelhante à proposta neste trabalho. Sendo um *dashboard* implementado por meio de uma solução web e com a tratativa de dados relacionados ao contexto da pandemia de COVID-19. Dentre as ferramentas utilizadas, o trabalho traz um diferencial principal nas bibliotecas JS para a geração de gráficos, nas tecnologias para realizar requisições HTTP e no gerenciamento de dados. Toda a estrutura do *dashboard* se mostrou com soluções básicas de desenvolvimento web, por meio da utilização de bibliotecas JS e gerenciamento de dados por meio de um banco relacional. Além disso, o trabalho reforça a relevância da linguagem JavaScript, que possibilita a integração de boas bibliotecas para a geração de diferentes tipos de gráficos que processam bem em uma página web.

### **2.2.3 Teleatendimento como ferramenta de monitoramento de casos suspeitos e/ou confirmados de COVID-19**

Em [Athayde et al. \(2020\)](#), os autores apresentam uma abordagem sobre o telemonitoramento realizado na cidade de Curitiba - PR. Durante o processo de telemonitoramento de casos suspeitos e/ou confirmados de COVID-19, a equipe de gestão utilizou diferentes meios de comunicação para se comunicar com os pacientes monitorados. Com o objetivo de garantir uma comunicação eficiente e ágil, foram utilizados um aplicativo próprio, o Chatboxt, além do WhatsApp e chamadas telefônicas

A equipe composta por multiprofissionais realizou um curso de dezoito horas disponibilizado pelo Ministério da Saúde, chamado Manejo da COVID 19 na Atenção Primária à Saúde.

No fluxo de monitoramento, foram usadas diferentes técnicas para cada forma de comunicação com o paciente monitorado. Em destaque, no serviço de atendimento via telefone, é feita uma busca de casos suspeitos notificados pela secretaria de saúde do Paraná, assim como o acompanhamento de casos confirmados passados pelo Serviço de Controle de Infecção Hospitalar (SCIH). No primeiro contato com o caso confirmado, é realizada uma ligação informando sobre o resultado do teste e fornecendo orientações sobre o envio de SMS com perguntas para acompanhar o estado de saúde do paciente. As respostas dessas perguntas são registradas e monitoradas por meio de um *dashboard*, onde é analisada a necessidade de contato telefônico da enfermeira. Caso não haja necessidade, uma pessoa enfermeira faz uma ligação telefônica entre dez a quatorze dias após o registro das respostas.

Com a utilização do aplicativo Chatboxt, os autores em [Athayde et al. \(2020\)](#) descrevem que esse sistema realiza diversas perguntas que são usadas para avaliar o estado de saúde do paciente. Por exemplo, são feitas perguntas sobre a presença de febre, dores de cabeça e no corpo, dificuldade para respirar, outros sintomas e aspectos que possam indicar uma possível infecção pelo vírus. Com base nas respostas a essas perguntas, são fornecidas orientações e determinado o grau de gravidade do caso, para que sejam feitos os encaminhamentos necessários para atendimento no Porto Atendimento.

Por fim, no aplicativo próprio “Cuidar+”, são fornecidas perguntas para classificar pacientes nas cores verde, vermelho e amarelo. Os pacientes vermelhos ou amarelos passam por um telemonitoramento via SMS.

É possível identificar no trabalho, uma abordagem mista no monitoramento remoto. São feitas diferentes tratativas com base na ferramenta utilizada. Com isso, é possível analisar qual tratativa pode ser mais adequada nos contextos de cada caso acompanhado. Uma similaridade apresentada no trabalho descrito em [Athayde et al. \(2020\)](#) está no uso de um *dashboard*, uma ferramenta de visualização de dados de interesse, para tomar decisões em cima desses dados apresentados.

#### **2.2.4 Projeto UFOP em Ação**

A atual monografia trata de uma extensão do trabalho desenvolvido no projeto de iniciação científica intitulado “Projeto UFOP em Ação”. A equipe do projeto é formada por alunos da graduação e da pós-graduação. Grande parte dos participantes é da área da Farmácia da Universidade Federal de Ouro Preto (UFOP). Durante os anos de 2020 e 2021, a equipe realizou o telemonitoramento de pacientes diagnosticados com COVID-19 e pacientes suspeitos de terem a doença. Os casos notificados foram levantados pelo TeleSUS e pela Secretaria Municipal de Saúde (SMS). Além disso, o projeto também realiza medidas preventivas de transmissão da COVID-19, como a distribuição de máscaras e atividades para o repasse de conscientização para a população ([ESCOLA DE FARMÁCIA DA UFOP, 2020](#)).

O projeto trouxe inovação ao realizar a implantação de uma aplicação web que serviu de suporte para maior eficiência dos dados colhidos durante as ligações no processo de telemonitoramento ([ESCOLA DE FARMÁCIA DA UFOP, 2020](#)). A aplicação web desenvolvida no projeto foi realizada utilizando uma plataforma no-code, chamada Bubble. A utilização dessa ferramenta se deu pela necessidade de soluções rápidas para o suporte no telemonitoramento de casos de COVID-19, onde o cenário pandêmico em questão tinha uma grande massa de novos casos por dia.

Na segunda parte do projeto, foi percebida a necessidade de uma aplicação web utilizando ferramentas de desenvolvimento atuais e com uma boa atuação, e que não seja tão limitante como a plataforma Bubble. Com isso, foi desenvolvida uma nova aplicação web com as funcionalidades semelhantes àquelas utilizadas na primeira solução e com maiores recursos para o telemonitoramento. Essa nova aplicação foi feita utilizando React, e demais bibliotecas JavaScript. A interface passou por alterações e o fluxo seguiu semelhante ao fluxo da solução feita com o Bubble, com a possibilidade de responder questionários, registrar o monitoramento feito utilizando um tipo de questionário, que varia para o caso tratado, realizar consultas dos casos confirmados e casos suspeitos, realizar downloads de planilhas em tempo real e upload de planilhas com dados de casos confirmados, assim como o acesso de todos para cada caso por meio de uma tabela interativa que possibilita realizar diferentes tipos de ações. Os tipos de questionários e as tabelas interativas, serão detalhados nas seções a seguir.

Neste trabalho, é desenvolvida uma extensão da solução da aplicação web, com foco na geração de um *dashboard* baseado nas informações de monitoramento fornecidas pelo projeto. A estrutura dos dados gerados e o componente para a geração dos gráficos no site são os principais elementos de desenvolvimento. Todo o processo de estruturação dos dados e a ferramenta utilizada para a criação dos gráficos serão detalhados no próximo capítulo.

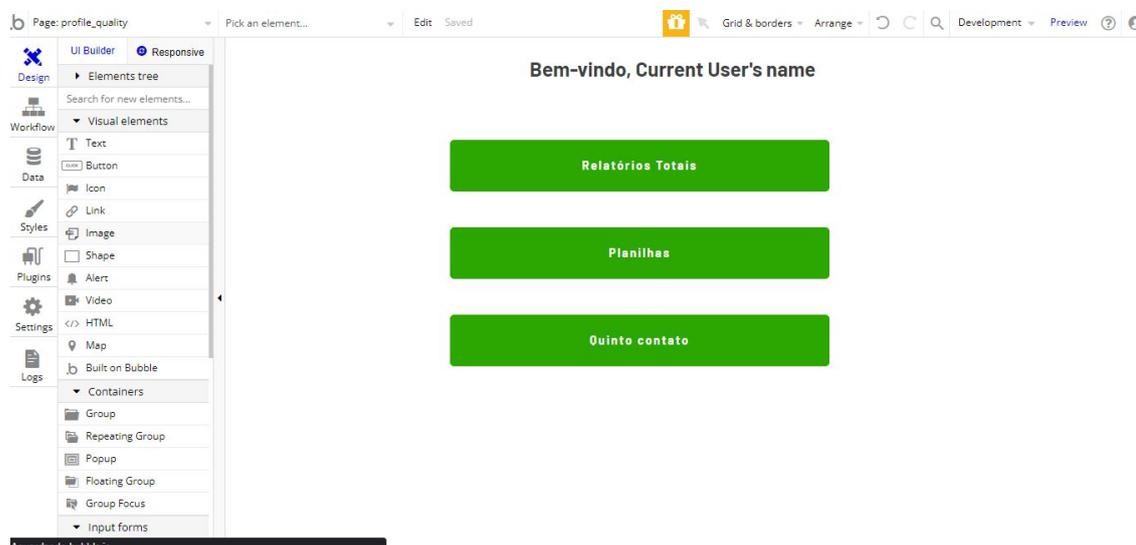
#### 2.2.4.1 Ferramenta Bubble e a construção a inicial do protótipo

Inicialmente, o design do website do projeto foi gerado com base no site já existente, que foi construído utilizando o Bubble. O Bubble é uma ferramenta que permite a criação de sites usando um editor de arrastar e soltar, no qual os elementos são posicionados no editor e posteriormente processados pelo Bubble para serem exibidos no site final. Esses elementos podem ser botões, campos de textos, tabelas, entre outros componentes. É possível associar uma interação para os elementos interativos da tela e assim, construir telas interativas para o site montado. Além disso, a ferramenta traz *plugins* que aumentam os recursos utilizados e melhoram a adaptação com a solução implementada no Bubble ([BUBBLE, 2022](#)).

A Figura 2.10 ilustra um exemplo da interface do editor do Bubble.io. Na barra lateral direita, é possível acessar as diversas ferramentas disponíveis para o editor do Bubble, enquanto na barra mais à direita encontram-se os elementos de configuração do projeto construído para a execução do site. No centro da tela, é apresentado um editor com os componentes que compõem

uma determinada tela a ser renderizada no site resultante do projeto de criação.

Figura 2.10 – Editor da ferramenta Bubble.



**Fonte:** Captura de tela da aplicação web desenvolvida no Bubble como parte do Projeto UFOP em Ação.

Abaixo seguem imagens das principais telas construídas pela plataforma Bubble. Na Figura 2.11 é mostrada a tela de login e na Figura 2.12 é mostrada a tela com um questionário a ser preenchido por um profissional de saúde durante um telemonitoramento remoto. A Figura 2.13 mostra a tela onde se encontram os dados no banco de dados para cada tipo de caso dos monitorados e pendentes para monitorados, onde, os dados na imagem estão censuradas por possuírem informações sensíveis.

Figura 2.11 – Tela de login pela plataforma Bubble.



**Fonte:** Captura de tela da aplicação web desenvolvida no Bubble como parte do Projeto UFOP em Ação.

Figura 2.12 – Tela de um questionário pela plataforma Bubble.

**Recusar Entrevista** **Registrar Obito**

### Terceiro Questionário

Bom dia (ou boa tarde)! Meu nome é Danielle, sou entrevistador(a) da Universidade Federal de Ouro Preto e estou realizando, em colaboração com a Secretaria Municipal de Saúde de \_\_\_\_\_, o rastreamento e monitoramento para COVID-19. Sua participação é muito importante. Podemos conversar? Esta conversa terá duração de aproximadamente 10 minutos. Gostaríamos de lhe fazer algumas perguntas. As informações dadas pelo(a) Sr.(a) não serão divulgadas, manteremos sigilo de todas as informações prestadas aqui, tudo bem? Vamos começar?

#### Bloco 1 – Informações básicas do(a) entrevistado(a)

Vou começar fazendo umas perguntas básicas sobre o(a) Sr.(a)

Nome  Telefone

Nome da mãe

Endereço

Rua/Av  n°  Complemento  Bairro

Sexo  Feminino  Masculino

Data de nascimento

Sua cor ou raça é:

Qual o seu estado civil?

CPF

Qual é o seu peso atual? (Em kg)

Qual é a sua altura? (Em cm)

É profissional da saúde? Não  Sim

É profissional de segurança? Não  Sim

Você foi vacinado? Não  Sim

Há pessoas acima de 60 anos que residem com você? Não  Sim

#### Bloco 2 - Informações sobre a saúde do(a) entrevistado(a)

Agora eu gostaria de saber a respeito da sua saúde e dos familiares que moram com você!

Quando você obteve a confirmação do diagnóstico para COVID-19?

Como você obteve a confirmação:

**Fonte:** Captura de tela da aplicação web desenvolvida no Bubble como parte do Projeto UFOP em Ação.

O design da aplicação web feito no Projeto em Ação com o Bubble foi reformulado com a ajuda do Figma, uma ferramenta colaborativa. Conhecida por sua facilidade de uso, flexibilidade e recursos de colaboração e amplamente usada por profissionais de design (FIGMA, 2022).

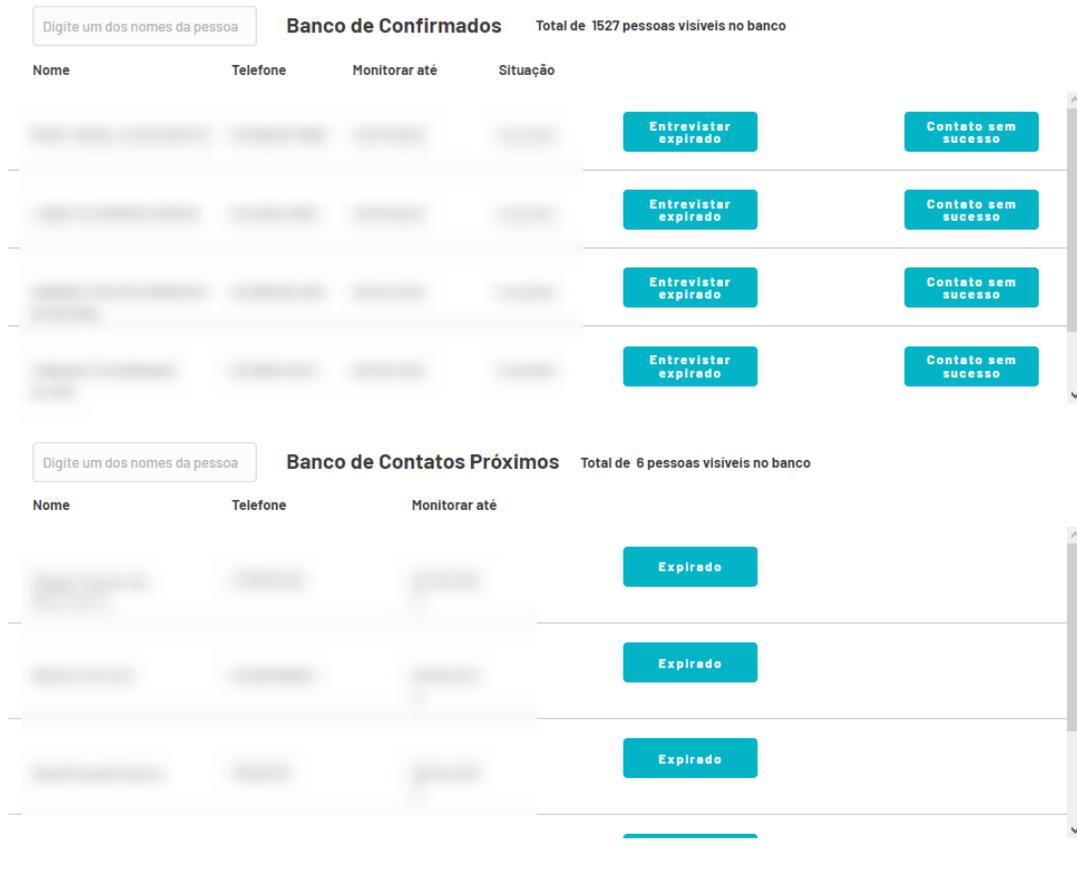
O protótipo criado no Figma foi elaborado com base em alterações para melhorar a usabilidade da aplicação, visando proporcionar ao usuário maior facilidade no uso e compreensão dos elementos presentes na tela.

Os componentes inicialmente prototipados foram referentes aos componentes básicos de um formulário. Esses componentes são compostos por botões, caixa de seleção, campo de texto que serão exibidos na tela dos questionários, como mostrado na Figura 2.14. O Figma oferece uma comunidade onde é possível disponibilizar projetos criados nele, que podem servir de *template* para outros usuários do sistema. Os elementos de questionário usados no protótipo foram retirados de um dos **templates** disponíveis no Figma.

#### 2.2.4.2 Ferramentas de Web-App do Projeto UFOP em Ação

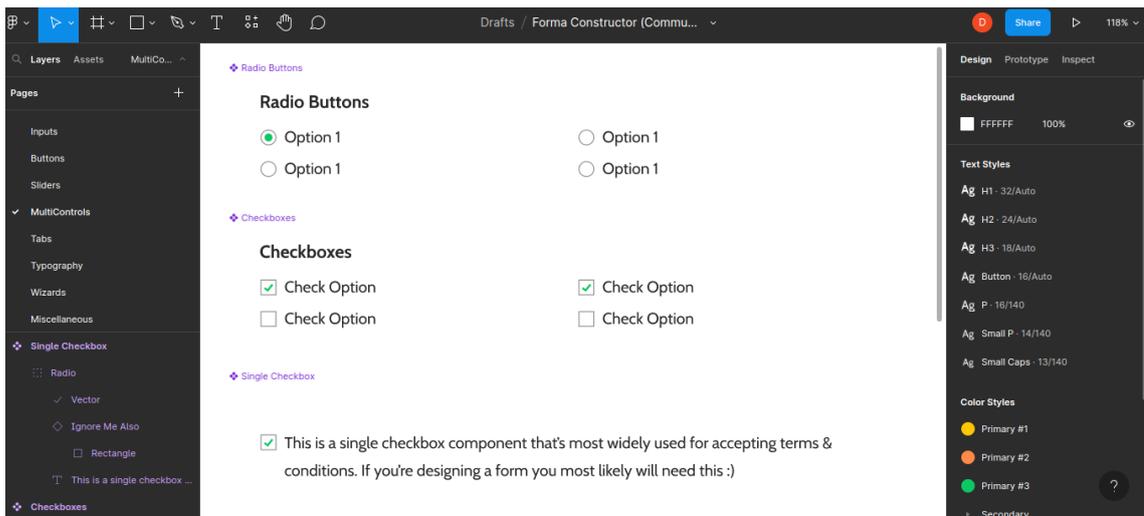
Para a construção do Web-App do Projeto, foram estabelecidos quais ferramentas seriam responsáveis para a construção das partes *front-end* e *back-end*. As ferramentas utilizadas para a

Figura 2.13 – Tabela com os bancos pela plataforma Bubble



Fonte: Captura de tela da aplicação web desenvolvida no Bubble como parte do Projeto UFOP em Ação.

Figura 2.14 – Protótipo dos componentes básicos para questionários.



Fonte: Captura de tela do protótipo no Figma desenvolvido como parte do Projeto UFOP em Ação.

implementação *front-end* foram:

- Biblioteca MUI;
- *Material-Table*;
- *Styled-components* <sup>4</sup>;
- React e funções primitivas.

O *back-end* desenvolvido na aplicação web do projeto foi construído utilizando o Firebase, referenciado na Seção 2.1.8. Para a aplicação, foi utilizado um banco de dados em tempo real, disponível em nuvem. Conforme a documentação do Firebase descreve, existem diferentes formas de conectar sua aplicação ao banco para diferentes linguagens utilizando um documento de configuração para a inicialização de uma aplicação Firebase. Feita a integração com o Firebase, é possível utilizar os serviços para gerenciamento, autenticação, manipulação de dados e entre outros. O tipo do banco em tempo real é do tipo NoSQL, o que significa que ele não usa tabelas com linhas e colunas, como em bancos de dados relacionais tradicionais.

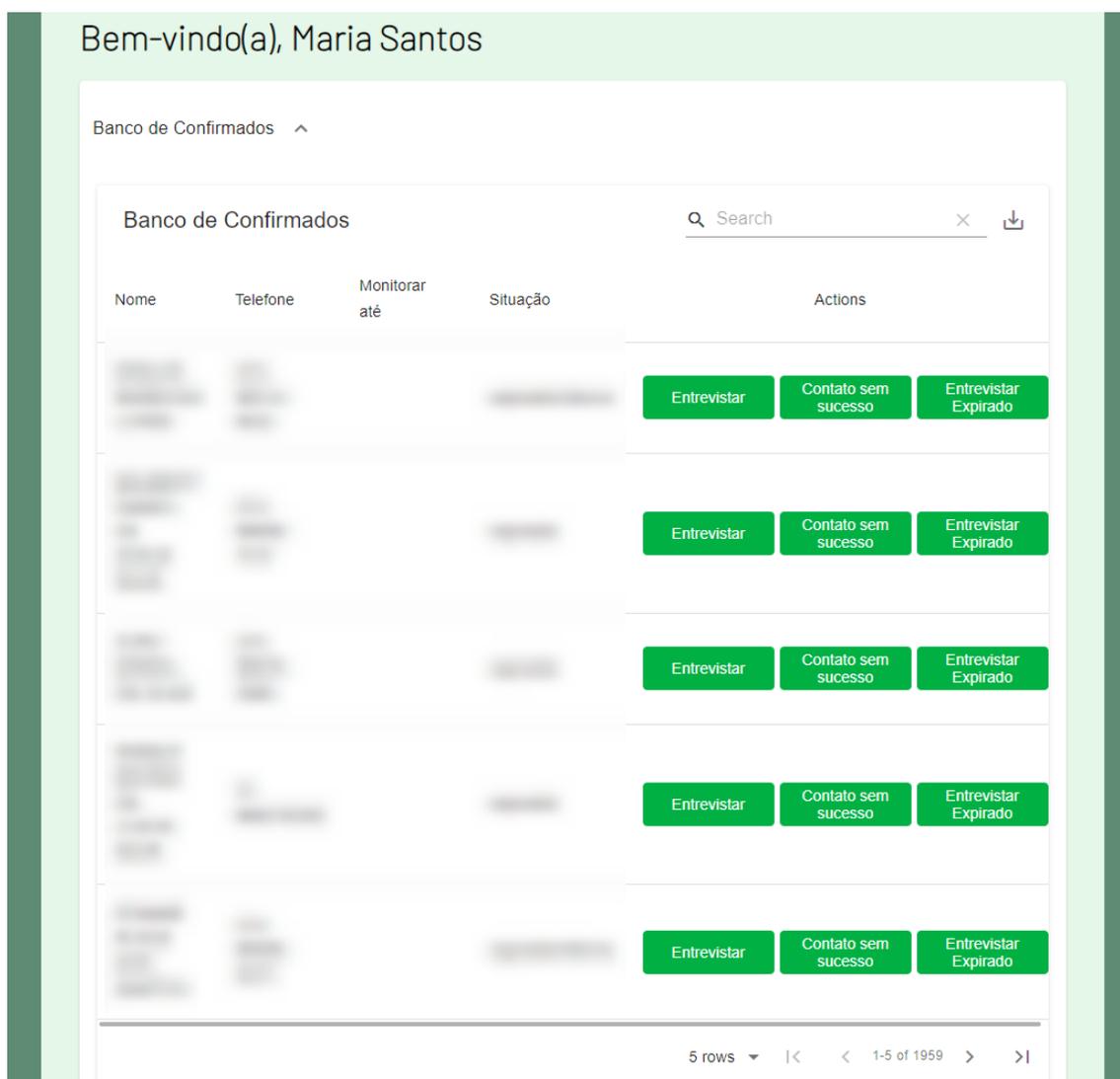
A Figura 2.17 mostra um exemplo da tela construída para os questionários a serem respondidos por um membro da equipe do projeto, com base nas respostas fornecidas pelo monitorado durante uma ligação telefônica. As Figuras 2.15 e 2.16 mostram uma tela em diferentes estados. A primeira imagem ilustra a tela com um **accordion** aberto, onde estão os dados de casos confirmados registrados no sistema. Na segunda imagem, os *accordions* estão fechados, sendo que os dois primeiros contêm uma tabela com o registro dos casos confirmados e dos contatos próximos, e os dois últimos exibem em tabela aqueles que estão sob monitoramento pelo usuário.

Os questionários para primeiro contato com um entrevistado, da aplicação do projeto, permitem que sejam feitos cadastrados de contatos próximos de um monitorado. A Figura 2.18 mostra a tela de um cadastro de um contato próximo, feita por meio de um elemento flutuante na tela, chamado modal. Existe também a possibilidade de determinar um desfecho ou recusar a entrevista, em caso da recusa do entrevistado para realizar o monitoramento, sido feito, também, por um modal, exemplificado na Figura 2.19.

---

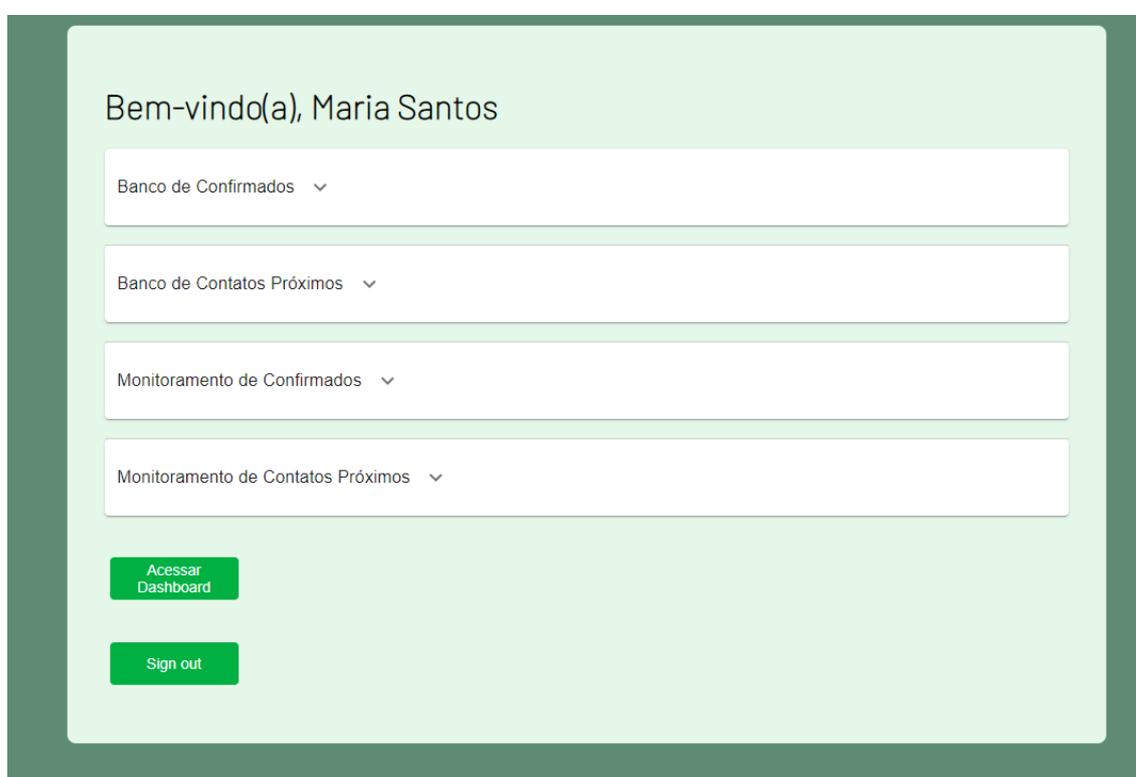
<sup>4</sup> O *styled-components* é uma biblioteca para React e React Native que possibilita a estilização de componentes React usando uma sintaxe semelhante ao CSS. O uso dessa biblioteca oferece maior flexibilidade, facilidade na manutenção e controle sobre os estilos aplicados aos componentes. (COMPONENTS, 2023)

Figura 2.15 – Exemplo da tela de uma tabela de um conjunto de dados para monitoramento.



**Fonte:** Captura de tela da aplicação web desenvolvida utilizando React como parte do Projeto UFOP em Ação.

Figura 2.16 – Exemplo da tela dos conjuntos de dados para acesso do usuário.



**Fonte:** Captura de tela da aplicação web desenvolvida utilizando React como parte do Projeto UFOP em Ação.

Figura 2.17 – Exemplo da tela de um questionário da aplicação do projeto.

The image shows a screenshot of a web questionnaire titled "Questionário 3". At the top right, it indicates "Tempo de início: 16:25:44". The main text of the questionnaire reads: "Bom dia (ou boa tarde)! Meu nome é Maria Santos, sou entrevistador(a) da Universidade Federal de Ouro Preto e estou realizando, em colaboração com a Secretaria Municipal de Saúde de \_\_\_\_\_, o rastreamento e monitoramento para COVID-19. Sua participação é muito importante. Podemos conversar? Esta conversa terá duração de aproximadamente 10 minutos. Gostaríamos de lhe fazer algumas perguntas. As informações dadas pelo(a) Sr.(a) não serão divulgadas, manteremos sigilo de todas as informações prestadas aqui, tudo bem? Vamos começar?". Below the text are two green buttons: "Recusar Entrevista" and "Registrar Óbito".

The questionnaire is divided into three stages, with the current stage being "Etapa 1 de 3: Informações Básicas". The form includes the following fields and options:

- Nome:
- Telefone:
- Nome da Mãe:
- Endereço:
  - Rua/Av:
  - nº:
  - Complemento:
  - Bairro:
- Sexo:
  - Feminino
  - Masculino
- Data de nascimento:
- Sua cor ou raça é:
- Qual o seu estado civil?:
- CPF:
- Qual o seu peso atual?:
- Qual a sua altura?:
- É profissional da saúde?:
  - Sim
  - Não
- É profissional da segurança?:
  - Sim
  - Não
- Você foi vacinado?:
  - Sim
  - Não
- Há pessoas acima de 60 anos que reside com você?:
  - Sim
  - Não

At the bottom of the form is a green button labeled "Próxima Etapa".

**Fonte:** Captura de tela da aplicação web desenvolvida utilizando React como parte do Projeto UFOP em Ação.

Figura 2.18 – Exemplo do modal de cadastro de contatos próximos.

Etapa 3 de 3

←

Informações Sobre os Contatos Próximos

Cadastrar Contato

Enviar entrevista

**Cadastrar Contato**

Nome

Telefone

Relação com o caso

- Domiciliar
- Familiar (extradomiciliar)
- Laboral
- Escolar
- Evento social
- Outro

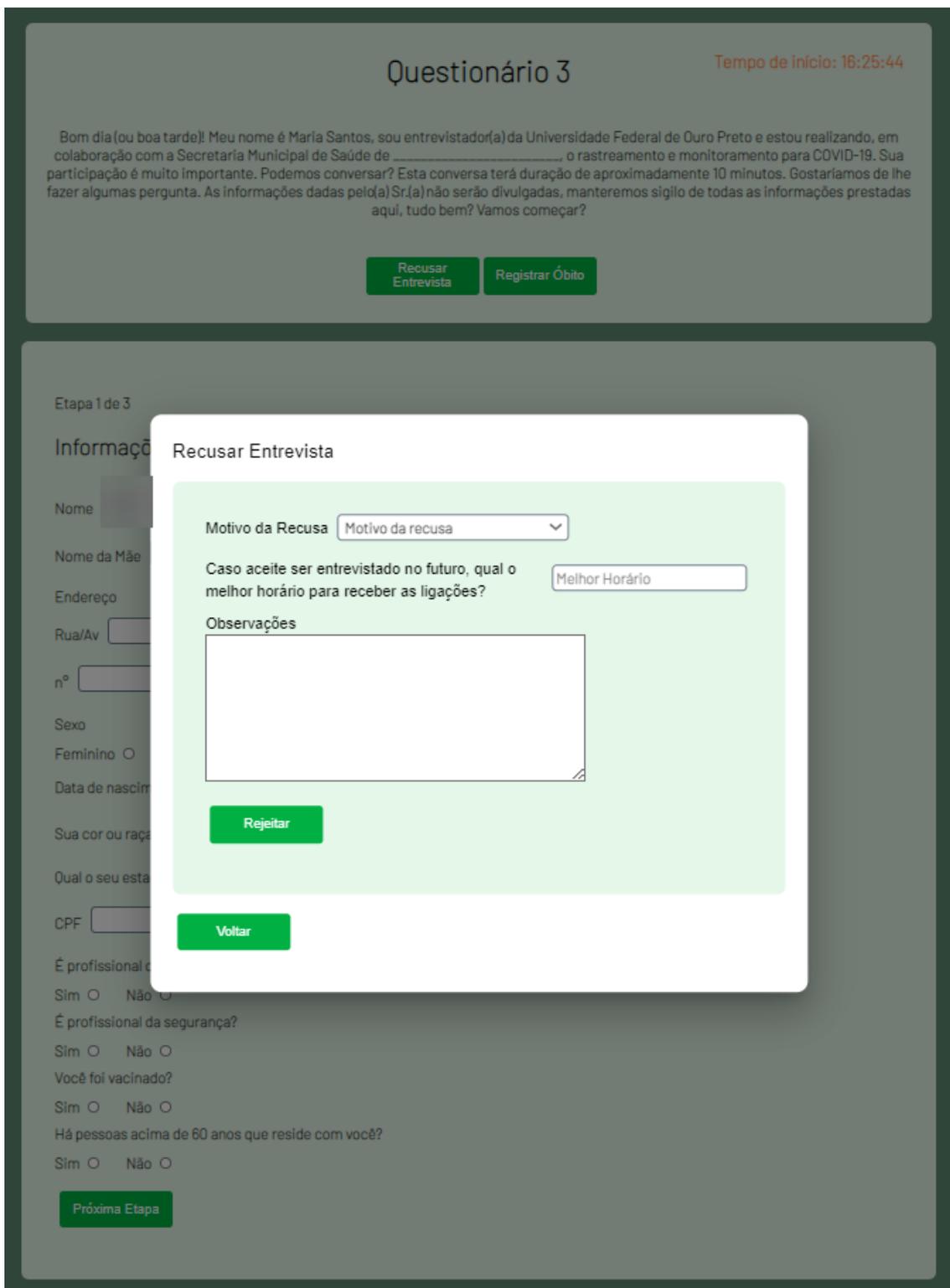
Data do último contato

Enviar

Voltar

**Fonte:** Captura de tela da aplicação web desenvolvida utilizando React como parte do Projeto UFOP em Ação.

Figura 2.19 – Exemplo do modal para preencher o motivo da recusa de entrevista.



**Fonte:** Captura de tela da aplicação web desenvolvida utilizando React como parte do Projeto UFOP em Ação.

## 3 Desenvolvimento

Neste capítulo, são apresentados os processos de desenvolvimento do trabalho. A Seção 3.1 discorre sobre a estrutura de pastas da aplicação deste estudo, um aspecto crucial em aplicações que empregam a tecnologia React. Em seguida, na Seção 3.2, são apresentados os detalhes sobre como o desenvolvimento tanto do *front-end* quanto do *back-end* foi aproveitado do projeto UFOP em Ação e quais foram as adições trazidas para este trabalho. Os detalhes mais relevantes sobre a construção do *dashboard* são apresentados na Seção 3.3, incluindo informações sobre o protótipo desenvolvido e a implementação na Seção 3.3.1. Posteriormente, na Seção 3.4, são apresentadas informações sobre os testes desenvolvidos para a validação da implementação do *dashboard*. Por fim, na Seção 3.4.2, é descrito o desenvolvimento de automações dos testes elaborados por meio da ferramenta Selenium.

### 3.1 Estruturação das pastas da aplicação

A estruturação de pastas é um estágio importante na construção de uma aplicação web utilizando React, por ajudar a organizar e gerenciar melhor os arquivos do projeto. Não há uma única forma de organização das pastas para ser usada em todas as aplicações React, no entanto, há uma estrutura comumente usada nos projetos com o uso da biblioteca React. A organização das pastas do web-app deste trabalho segue a estrutura ilustrada na Figura 3.1, com as seguintes características:

- A pasta ‘node\_modules’ é gerada quando se instalam as dependências do projeto por meio do gerenciador de pacotes do Node JS;
- Na pasta ‘components’ estão os componentes para serem usados na interface, com uma pasta para cada tipo de componente, onde contém outros arquivos que variam entre dependências dos componentes e definições de subcomponentes;
- As páginas da aplicação estão contidas na pasta ‘pages’, onde possui uma pasta para cada página com seus arquivos de definição;
- Para gerenciamento de estados dos componentes e compartilhamento desses estados, é utilizada uma biblioteca conhecida para gerenciamento de estados, a biblioteca Redux. Ao utilizá-la, é possível definir um objeto global que contém as informações para renderizar um componente e atualizar suas informações conforme as alterações referentes a esse objeto acontecer, refletindo o resultado dessa atualização em toda aplicação. Portanto, na pasta ‘redux’ ocorre as definições dos objetos globais criados por meio da biblioteca redux;

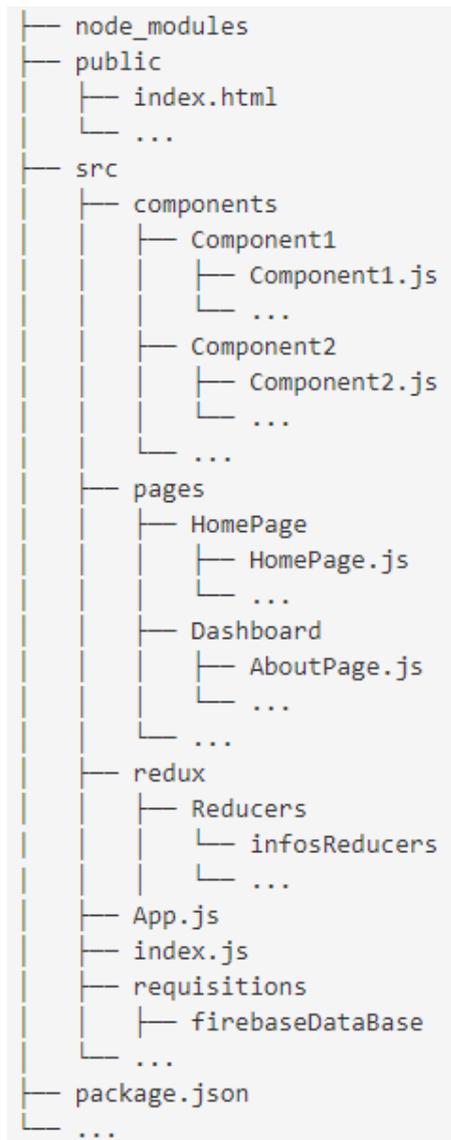


Figura 3.1 – Estrutura de pastas do projeto

- O arquivo `App.js` contém o componente raiz da aplicação, onde se encontram as rotas de toda a aplicação, esse arquivo é chamado no arquivo `index.html`, na pasta `public`, renderiza o componente `App` na página HTML, inicializando a aplicação;
- O arquivo `package.json` inclui detalhes sobre as dependências do projeto e os *scripts* para iniciar o servidor. As dependências trazem informações importantes para definir o que será necessário trazer para a aplicação no momento da instalação inicial via `npm`, assim como, a descrição do versionamento de cada dependência utilizada.

## 3.2 Implementação *front-end* e *back-end* do *dashboard*

As implementações de *front-end* e *back-end* que foram previamente desenvolvidas para o projeto UFOP em Ação, mencionado na Seção 2.2.4, foram aproveitadas na construção do *dashboard* neste estudo. Para realizar a solicitação dos dados tratados na nova tela do sistema, foram utilizadas funções específicas, de forma a não afetar a estrutura dos dados armazenados no Firebase, nem a maneira como esses dados são solicitados no sistema. No entanto, foram construídos novos elementos visuais descritos no decorrer desta Seção.

O React oferece uma grande vantagem ao permitir a integração de diferentes bibliotecas com componentes prontos, possibilitando o uso direto desses componentes ou sua adaptação para atender melhor às necessidades específicas. No caso da aplicação desenvolvida no projeto UFOP em Ação, e utilizada nesta monografia, alguns componentes foram criados manualmente. No entanto, é importante destacar que a maioria significativa dos componentes foi personalizada e modularizada a partir de componentes extraídos de bibliotecas. Muitos componentes presentes na aplicação são da biblioteca MUI que passaram por personalização, como os componentes *accordion* e seletores de opções, que estão presentes na implementação do *dashboard*. Além disso, foram criados outros componentes do zero, como os *cards* e botões.

A interface do *dashboard* apresenta gráficos criados por meio do *react-chartjs-2*, uma biblioteca do *Chart.js* que oferece componentes React para a criação de gráficos. Esses componentes têm estruturas de dados semelhantes às interpretadas pelo *Chart.js*, incluindo suas características visuais correspondentes.

As bibliotecas MUI e *Chart.js* fornecem elementos responsivos, ou seja, que se ajustam automaticamente ao tamanho da página da web. Ao redimensionar a página, os elementos se adaptam ao tamanho atual da tela, proporcionando uma visualização mais adequada para aquela tela específica. Portanto, ao utilizar os elementos responsivos dessas bibliotecas, foi possível criar um painel responsivo e adaptável tanto para o *dashboard* quanto para as demais telas. A Figura 3.2 ilustra exemplos de responsividade presente na tela do *dashboard*.

## 3.3 Construção do *dashboard*

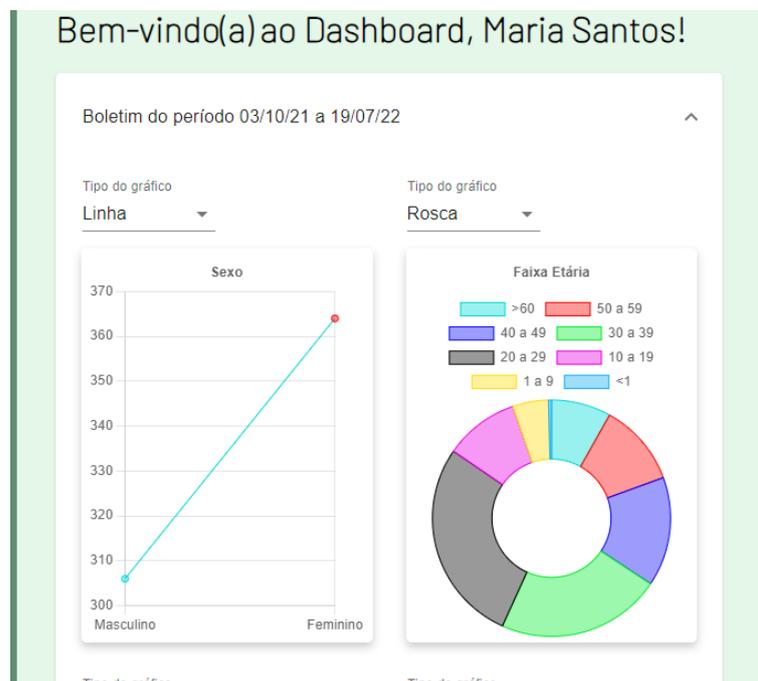
Para a construção do *dashboard* na aplicação, foi criado um protótipo interativo no Figma da tela a ser desenvolvida. Esse protótipo contém um fluxo de interações que permite visualizar os diferentes estados de tela que estarão visíveis ao usuário. Por meio dele, foi possível verificar a execução funcional do fluxo do *dashboard*, bem como os componentes reutilizáveis e seus respectivos designs. Além disso, o protótipo permitiu validar a solução de design e interação do *dashboard* em relação à solução prevista para este trabalho. Na Figura 3.3 é possível visualizar o protótipo do *dashboard* construído no Figma.

O protótipo incorpora gráficos, que foram extraídos de um modelo repleto de componentes

Figura 3.2 – Exemplos de telas responsivas do *dashboard*.



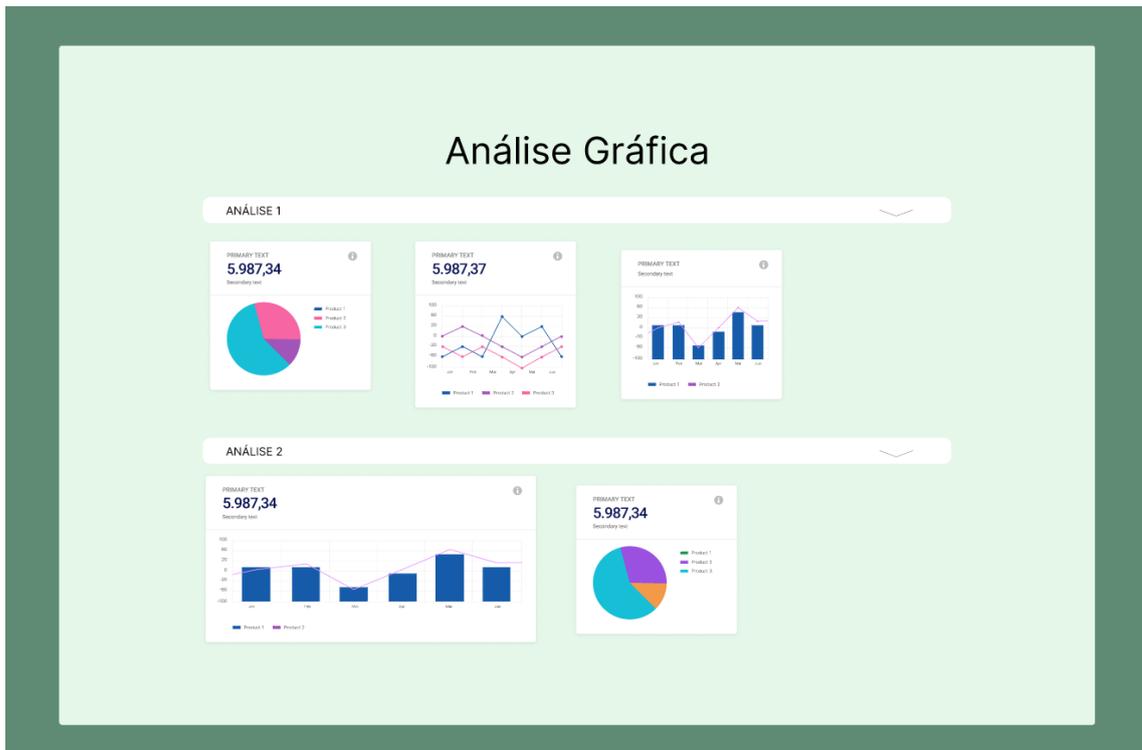
(a) Tela do *dashboard* reduzida e com os *accordions* fechados.



(b) Tela do *dashboard* reduzida e com os *accordions* abertos exibindo os gráficos.

**Fonte:** Capturas de tela da interface do *dashboard* construído como parte do desenvolvimento desta monografia.

gráficos disponíveis na comunidade do Figma para uso gratuito. Além dos componentes de gráfico, há um componente chamado *accordion* que está prototipado na tela, o qual permite que você manipule a visualização de conteúdos em tela. Um clique no *accordion* realiza a expansão e

Figura 3.3 – Protótipo do *dashboard* construído no Figma.

**Fonte:** Captura de tela do protótipo construído no Figma como parte do desenvolvimento desta monografia.

contração do conteúdo contido nele.

### 3.3.1 Implementação do *dashboard*

A construção do *dashboard* na aplicação se iniciou com a implementação de um componente para os gráficos. A ideia central foi criar um componente genérico para renderizar um gráfico na tela. Esse componente recebe, como propriedades, os dados para serem usados em um elemento de gráfico disponível na biblioteca Chart.js.

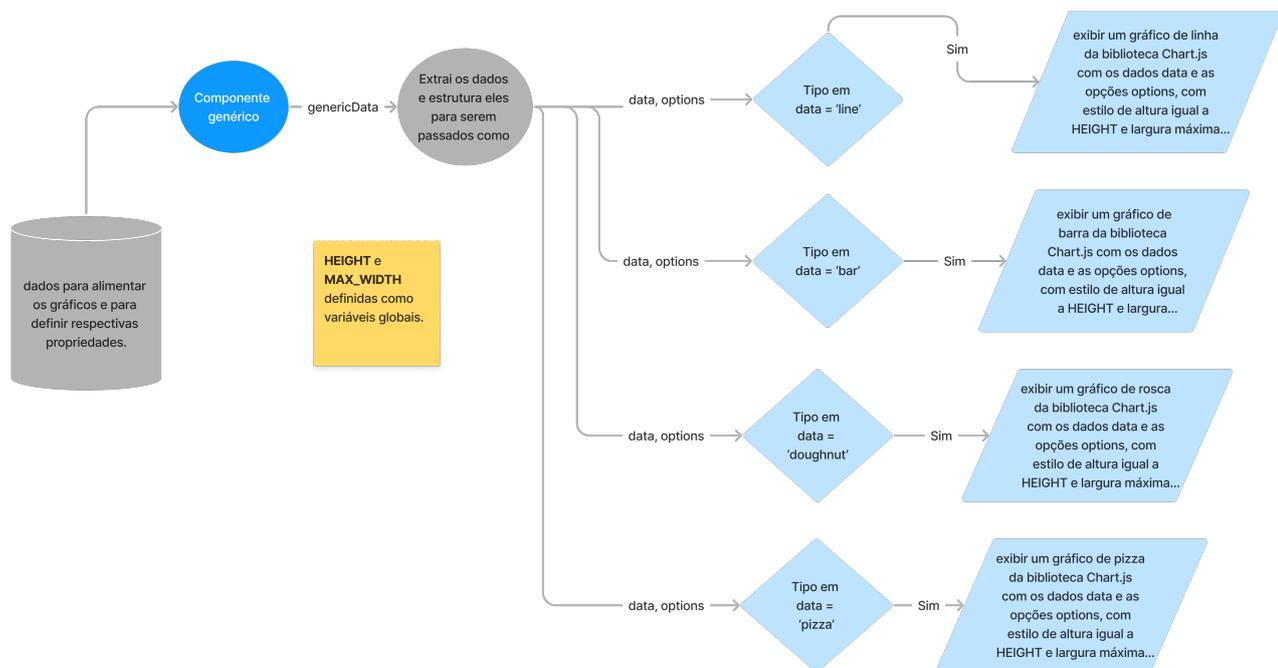
A documentação oficial da biblioteca Chart.js descreve a estrutura dos dados a serem interpretados pelo elemento de gráfico. As informações podem ser fornecidas em diferentes formatos, mas, por padrão, estão associadas ao tipo e à escala do gráfico (CHART.JS, 2023).

Existem três propriedades essenciais para a apresentação de dados a serem plotados. A primeira delas refere-se aos 'labels', que representam os rótulos associados aos pontos do gráfico e são obrigatórios em gráficos de linha e barra. A segunda propriedade diz respeito aos 'datasets', que fornecem informações sobre um conjunto de dados a ser plotado no gráfico. Segundo a documentação oficial, essas informações definem características do design dos elementos do gráfico, como cor da linha, cor de preenchimento de uma área e opções para personalização dos tamanhos dos elementos contidos em um 'dataset'. Por fim, os dados de interesse são passados na terceira propriedade, denominada 'data', definida em um objeto 'dataset'.

Além das três propriedades mencionadas anteriormente, existe uma propriedade poderosa para personalização de gráficos, denominada 'option'. Por meio dela, é possível sobrescrever as propriedades padrão dos gráficos e adaptar aspectos visuais e comportamentais, como personalização de títulos e subtítulos, configurações personalizadas para os eixos do gráfico, legendas, tooltips, animações e escalas de cores. Essa propriedade oferece um alto grau de flexibilidade para ajustar o gráfico conforme a utilização dele em uma aplicação.

A Figura 3.4 apresenta um diagrama que ilustra a lógica utilizada na construção de gráficos por meio de um componente genérico. A criação de um componente genérico possibilita uma modularização e personalização mais eficiente do código, onde os dados são manipulados em um único componente e reutilizados nos subcomponentes, para melhor organização e gerenciamento.

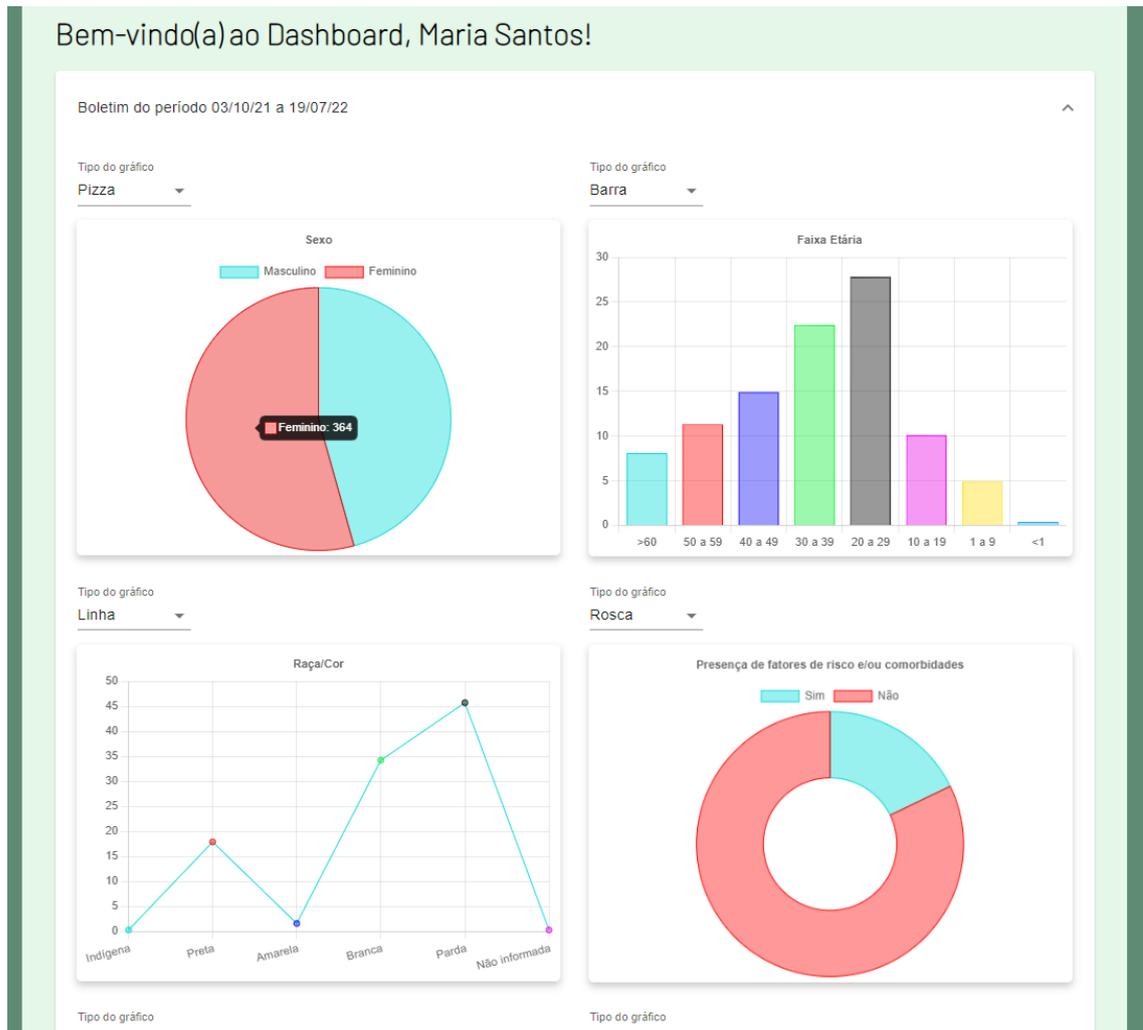
Figura 3.4 – Diagrama da lógica do componente genérico para os gráficos.



**Fonte:** Captura do diagrama elaborado pela autora.

Um *dashboard* é uma interface que apresenta informações por meio de elementos visuais, tais como gráficos, tabelas, mapas, entre outros. No contexto deste trabalho, o *dashboard* é composto exclusivamente por gráficos que representam informações de um boletim epidemiológico. Esse boletim é elaborado com base nos dados do telemonitoramento realizado pela equipe do projeto UFOP em Ação, bem como nas notificações de casos de COVID-19 provenientes do sistema de notificação e-SUS-Ve, os quais são repassadas à equipe do projeto.

Conforme o protótipo apresentado na Seção 3.3.1, ocorreu a implementação dos elementos do *dashboard*, com elementos de gráficos e *accordion*. No entanto, a implementação teve mudança para a inclusão de novos elementos, como *cards* e um seletores. Os *cards* permitem acoplar os

Figura 3.5 – Exemplos de gráficos do *dashboard* desenvolvido.

**Fonte:** Captura de tela da interface do *dashboard* construído como parte do desenvolvimento desta monografia.

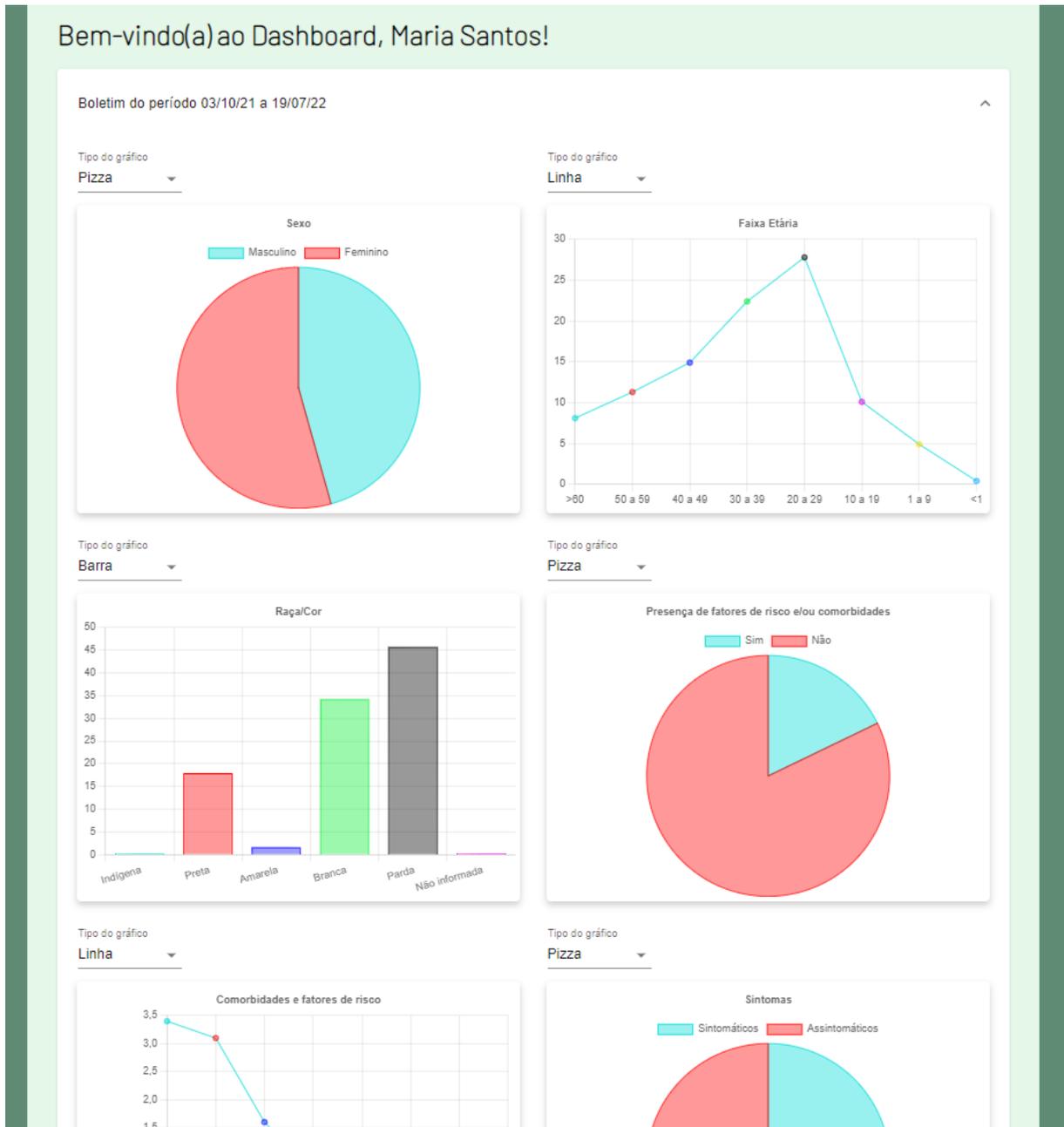
gráficos em “quadros” dentro do *accordion* e o seletores foram implementados para o usuário manipular o tipo de um gráfico visível na tela. As Figuras 3.5 e 3.6 ilustram exemplos dos gráficos implementados. Na Figura 3.7 é mostrado um exemplo do seletor de tipo de gráfico aberto, que oferece opções para a escolha do tipo de gráfico desejado.

## 3.4 Execução de testes *End-To-End* na aplicação

### 3.4.1 Casos de teste como documentação para os testes *End-To-End*

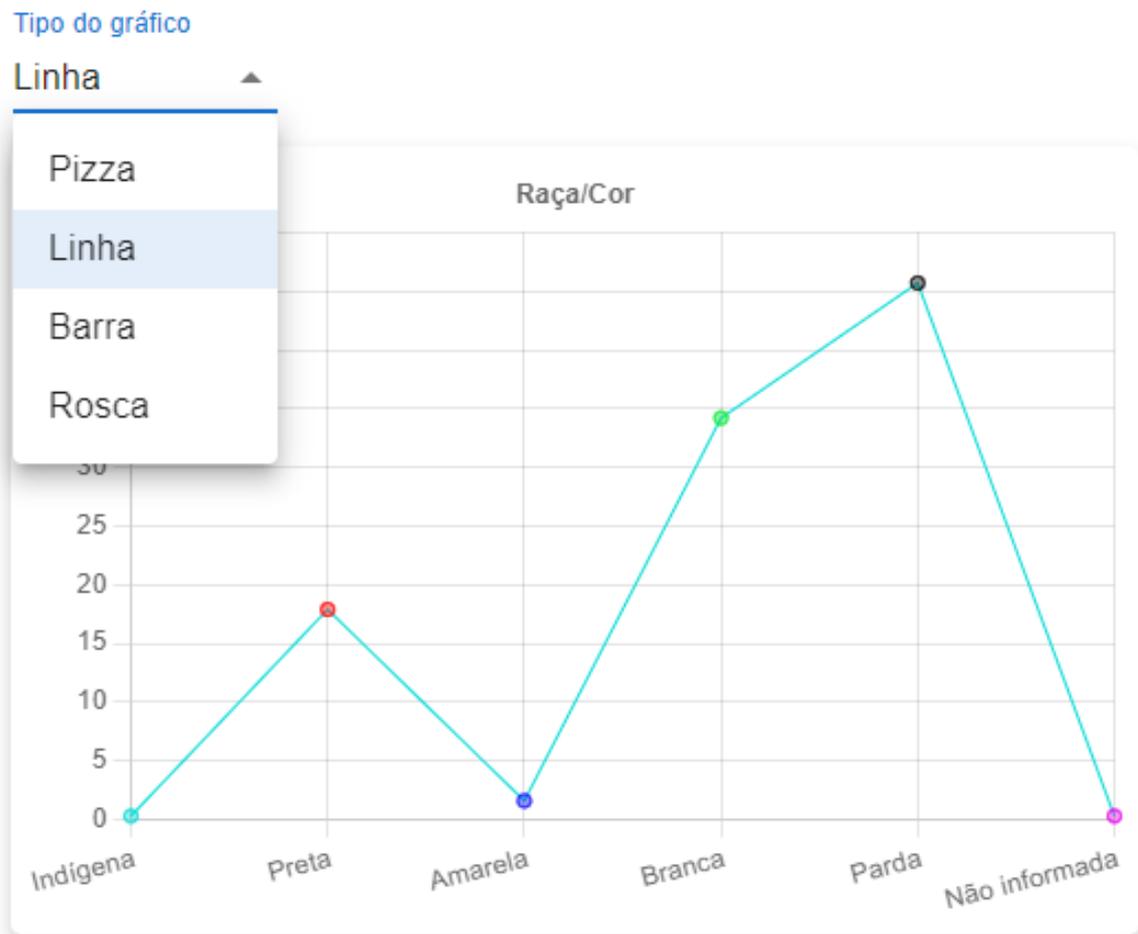
Os casos de testes são um tipo de documentação que possui informações minuciosas acerca dos cenários a serem testados em um software. Cada caso de teste é formado por um conjunto de etapas específicas que devem ser seguidas para comprovar a efetividade do software. Estas etapas englobam os seguintes pontos:

Figura 3.6 – Tela do *dashboard* desenvolvido.



**Fonte:** Captura de tela da interface do *dashboard* construído como parte do desenvolvimento desta monografia.

Figura 3.7 – Exemplo de um seletor de tipo do gráfico.



**Fonte:** Captura de tela da interface do *dashboard* construído como parte do desenvolvimento desta monografia.

1. Identificação (ID) e título;
2. Objetivo do teste;
3. Pré-condições;
4. Passos detalhados a serem seguidos;
5. Critérios de saída ou resultados esperados;
6. Pós-condições;
7. Observações adicionais (opcional).

As Tabelas 3.1 e 3.2 trazem exemplos de passos e resultados esperados de dois casos de testes criados para a execução dos passos em um teste *End-To-End*. O primeiro caso de teste é um caso básico de validação do processo de login no software e o segundo caso é um caso

Tabela 3.1 – Exemplo dos passos e resultado esperado do Caso de Teste para o fluxo de Login.

<b>C01 - Validar Login no web-app</b>
<p><b>Passos do caso de teste:</b></p> <ol style="list-style-type: none"> <li>1 - Abrir a página de login da web-app.</li> <li>2 - Inserir as credenciais válidas (nome de usuário e senha).</li> <li>3 - Clicar no botão de login.</li> <li>4 - Verificar se o usuário é redirecionado para a página inicial da web-app.</li> </ol>
<p><b>Resultado esperado:</b></p> <p>O usuário deve conseguir fazer login com sucesso no web-app e ser redirecionado para a página inicial.</p>

mais específico da aplicação para a validação do funcionamento da expansão e contração do *accordion*.

<b>C07 - Validar expansão do accordion</b>
<p><b>Passos do caso de teste:</b></p> <ol style="list-style-type: none"> <li>1 - Abrir a página de login da web-app;</li> <li>2 - Inserir as credenciais válidas (nome de usuário e senha);</li> <li>3 - Clicar no botão de login;</li> <li>4 - Verificar se o usuário é redirecionado para a página inicial da web-app;</li> <li>5 - Clicar no botão "Acessar Dashboard";</li> <li>6 - Clicar no primeiro accordion exibido;</li> <li>7 - Clicar no segundo accordion exibido;</li> <li>8 - Clicar no primeiro accordion exibido.</li> </ol>
<p><b>Resultado esperado:</b></p> <p>O usuário deve conseguir gerenciar a visualização em tela de um conjunto de gráficos de um boletim.</p>

Tabela 3.2 – Exemplo dos passos e resultado esperado do Caso de Teste para a funcionalidade do *accordion*.

Foram criados sete casos de teste visando abranger uma cobertura de validação das funcionalidades da aplicação. Dentre esses casos de teste, dois foram direcionados à validação de funcionalidades de login e cadastro, enquanto outros casos de teste se destinaram para a validação da visualização *dashboard*, a interação do tooltip em gráficos ao passar o mouse sob eles, o

vínculo de visibilidade de uma legenda do gráfico com o seu respectivo conjunto de dados ao clicar na legenda e a expansão do *accordion* no *dashboard*. Os casos de teste são detalhados no Apêndice A.

### 3.4.2 Selenium como ferramenta para automação dos testes *End-To-End*

Um das ferramentas mais utilizadas para construção de testes *End-To-End* é o Selenium. Como mencionado na Seção 2.1.9, a ferramenta utilizada para os testes deste trabalho foi o Selenium IDE, o qual executa testes de maneira rápida e eficiente.

Os casos *End-To-End* foram utilizados para especificar o fluxo de cada teste que seria automatizado. Dos sete casos de teste criados, cinco foram automatizados. Os dois casos de teste que não foram automatizados correspondem aos testes que validam a funcionalidade de *tooltips* e a visibilidade de um conjunto em um gráfico ao clicar na legenda correspondente.

Os *tooltips* são pequenas caixas de diálogo no gráfico que fornecem informações específicas sobre um conjunto de dados quando o usuário passa o cursor do mouse sobre o elemento visual desse conjunto. Como o gráfico é um elemento *canvas*<sup>1</sup> em HTML, não é possível acessar seus elementos filhos, incluindo o *tooltip*. Portanto, não foi possível automatizar esse teste.

Da mesma forma, a validação da funcionalidade de visibilidade dos conjuntos de dados ao clicar na legenda também não pôde ser automatizada devido ao mesmo problema com os *tooltips*. Os conjuntos de dados e a legenda são elementos filhos de um elemento *canvas*, o que dificulta o acesso para a automação.

O processo de automação foi uma tarefa relativamente simples, utilizando a interface do Selenium IDE. Para cada passo do teste, um comando foi inserido através da interface do Selenium. O Selenium fornece uma ampla variedade de comandos, incluindo cliques, validação de elementos em tela, inserção de texto, inserção de estruturas de repetição, inserção de *scripts* em JavaScript, entre outros. Cada comando possui propriedades que serão interpretadas de maneira distinta, no entanto, uma das principais propriedades é o ‘Target’, que irá pegar um elemento em tela conforme a especificação passada a ele, o campo de ‘Target’ é mostrado na Figura 3.7 na Seção 3.4. A especificação usada nos casos de teste para pegar um elemento em tela foi o **XPath**, uma linguagem de consulta para localizar e selecionar elementos em um documento XML. A consulta assume o documento XML como uma árvore, onde cada elemento representa um nó. A Figura 3.8 mostra os exemplos das saídas que o Selenium fornece após a execução de um caso de teste. A Figura 3.9(a) mostra a saída da execução do teste para validação de *login*, e Figura 3.9(b) para a expansão do *accordion*. Na saída de cada teste é ilustrado o XPath utilizado para pegar o elemento alvo na interface, referente a um determinado caso do teste.

<sup>1</sup> Segundo a documentação oficial do MDN, o elemento *canvas* é um elemento HTML que oferece a capacidade de renderizar conteúdos variados em um bloco, podendo ser um gráfico, uma imagem ou uma animação. Esses elementos são desenhados em tempo real em uma página web, permitindo uma visualização interativa e dinâmica (DOCS, 2023).

Figura 3.8 – Exemplos da saída de execução de um caso de teste no Selenium.

**Running 'C01 - Validar login no web-app'**

1. open on http://localhost:3000/ OK
2. waitForElementVisible on xpath=//input[contains(@class, 'MuiOutlinedInput') and contains(@name, 'email')] with value 30000 OK
3. type on xpath=//input[contains(@class, 'MuiOutlinedInput') and contains(@name, 'email')] with value user.teste@teste.com OK
4. type on xpath=//input[contains(@class, 'MuiOutlinedInput') and contains(@name, 'password')] with value Teste123 OK
5. click on xpath=//button[contains(text(), 'Logar')] OK
6. waitForElementVisible on xpath=//h1[contains(text(), 'Bem-vindo(a)')] with value 30000 OK
7. executeScript on return ['Banco de Confirmados', 'Banco de Contatos Próximos', 'Monitoramento de Confirmados', 'Monitoramento
8. forEach on dataList with value item OK
9. verifyElementPresent on xpath=//\*[contains(text(), '\${item}')] OK
10. end OK
11. click on xpath=//button[contains(text(), 'Sign out')] OK

**'C01 - Validar login no web-app' completed successfully**(a) Saída da execução do caso de teste para validar o *login***Running 'C07 - Validar expansão do accordion'**

1. open on http://localhost:3000/ OK
2. waitForElementVisible on xpath=//input[contains(@class, 'MuiOutlinedInput') and contains(@name, 'email']
3. type on xpath=//input[contains(@class, 'MuiOutlinedInput') and contains(@name, 'email')] with value user.t
4. type on xpath=//input[contains(@class, 'MuiOutlinedInput') and contains(@name, 'password')] with value T
5. click on xpath=//button[contains(text(), 'Logar')] OK
6. waitForElementVisible on xpath=//h1[contains(text(), 'Bem-vindo(a)')] with value 30000 OK
7. executeScript on return ['Banco de Confirmados', 'Banco de Contatos Próximos', 'Monitoramento de Confir
8. forEach on dataList with value item OK
9. verifyElementPresent on xpath=//\*[contains(text(), '\${item}')] OK
10. end OK
11. click on xpath=//button[contains(text(), 'Acessar Dashboard')] OK
12. waitForElementVisible on xpath=//h1[contains(text(), 'Bem-vindo(a) ao Dashboard')] with value 30000 OK
13. executeScript on return [1, 2] with value numbers OK
14. forEach on numbers with value num OK
- echo: 1
16. click on xpath=(//div[contains(@class, 'MuiAccordion')]/p[contains(text(), 'Boletim')])[\${num}] OK
- echo: 2
17. end OK
18. click on xpath=//button[contains(text(), 'Acessar Home')] OK
19. click on xpath=//button[contains(text(), 'Sign out')] OK

**'C07 - Validar expansão do accordion' completed successfully**(b) Saída da execução do caso de teste para validar a expansão do *accordion* no *dashboard***Fonte:** Capturas de tela das saídas geradas pelos testes executados no Selenium IDE.

Em geral, os testes de *End-To-End* foram realizados para verificar a consistência dos elementos desenvolvidos durante a implementação do painel, bem como analisar o desempenho da aplicação em relação ao tempo de requisição e carregamento de dados. Essa análise foi viabilizada por meio da saída da execução dos testes, que exibe o tempo de execução de cada etapa, permitindo assim verificar o intervalo de tempo entre uma etapa e outra.

Foram executados cinco testes no Selenium para verificar os fluxos principais para a validação da implementação. Os casos testados foram de *login*, cadastro, validação da funcionalidade dos *accordions* da tela *dashboard*, validação da visualização dos elementos do *dashboard* e validação da troca do tipo de gráfico. Todos os testes obtiveram êxito. Dois testes não foram executados, pois não foi possível pegar os elementos em tela por meio da linguagem de consulta utilizada, o XPath. Com isso, os dois testes restantes não foram automatizados e testados manualmente, consistindo na validação de elementos do Chart.js, que não puderam ser alcançados na hierarquia de nós do elemento. O gráfico um elemento canvas do HTML e que não possui nós internos para pegar legendas, áreas do gráfico, títulos e outros elementos do gráfico renderizado por meio de um elemento canvas.

## 4 Considerações Finais

Neste capítulo, é feita uma consideração final de todo o trabalho. Na Seção 4.1 é apresentada uma conclusão tratando do problema deste trabalho e como a solução até o momento atende aos objetivos traçados inicialmente. E na Seção 4.2, são listados e descritos, os trabalhos a serem alcançados, partindo do atual estado de desenvolvimento.

### 4.1 Conclusão

As aplicações web se mostraram poderosas ferramentas hoje em dia devido à facilidade de acesso a essas aplicações, bastando ter um dispositivo com navegador e acesso à internet. Além da possibilidade da criação de design moderno e responsivo e com diversas ferramentas no mercado para o suporte de aplicações com qualidade. As manutenções e mudanças são facilmente acessadas pelos usuários, já que aparecem automaticamente quando o site da aplicação é atualizado.

A pandemia de COVID-19 trouxe grandes desafios, com impactos demográficos. Os cuidados com a saúde tiveram que ser redobrados, trazendo a necessidade de medidas preventivas ao contágio com o novo coronavírus. Ferramentas da telessaúde se tornaram grandes aliadas na luta contra a COVID-19 para os profissionais de saúde. Uma das estratégias da telessaúde, que se tornou um destaque como uma solução rápida, foi o telemonitoramento ou monitoramento remoto. Essa estratégia deu suporte aos profissionais de saúde, que pode ser executada por pessoas capacitadas para realizar o monitoramento remoto, diminuindo a demanda em unidades de saúde.

Diante do problema apresentado, nesta monografia, é apresentada uma aplicação web, que objetiva entregar suporte no telemonitoramento na obtenção dos dados relativos aos monitoramentos realizados para fins de pesquisa por parte dos profissionais de saúde. Para alcançar essa meta, outras metas foram estabelecidas considerando que uma aplicação web com uma interface afetiva e de fácil uso é essencial para garantir que os profissionais da saúde que não possuem muito contato com a tecnologia, realizem o monitoramento sem muito esforço, trazendo maior praticidade, produtividade e ganho de tempo. Sendo assim, foram levantados elementos simples e modernos, que sejam de fácil interpretação, com uma interação simples e amigável para serem colocados na interface do usuário.

A implementação do *dashboard* neste estudo foi identificada como uma solução eficaz no suporte ao processo de análise de dados pelos profissionais da área da saúde. Anteriormente, a equipe responsável pelo telemonitoramento do Projeto UFOP em Ação precisava dedicar dias para a elaboração manual dos boletins epidemiológicos, utilizando informações coletadas por meio do monitoramento e das notificações do e-SUS-Ve, e produzindo uma variedade de gráficos. A solução proposta neste estudo permitiu que esses gráficos fossem gerados automaticamente

em poucos minutos, demonstrando, mais uma vez, a efetividade dessa ferramenta como suporte aos profissionais da área da saúde.

Adicionalmente, os testes automatizados End-To-End realizados no Selenium indicaram que a aplicação React do sistema apresentou desempenho satisfatório no carregamento de dados e componentes, especialmente no que se refere à carga de gráficos. Ademais, os testes foram executados seguindo as etapas descritas nos casos de teste documentados, cujos objetivos são garantir a funcionalidade da aplicação por meio de um fluxo desejado no sistema. Entretanto, alguns componentes não puderam ser capturados por meio do uso de XPath, resultando na impossibilidade de executar dois testes restantes por meio de automação. Como resultado, esses testes foram executados manualmente, evidenciando que, apesar de ser uma ferramenta simples e eficiente para a automação de testes, o Selenium IDE pode apresentar limitações que restringem sua abrangência de aplicação.

## 4.2 Trabalhos Futuros

A solução proposta aborda o telemonitoramento de casos de COVID-19 e traz um apoio ao profissional da saúde nessa tarefa, assim como no processo de pesquisas a serem feitas com os dados obtidos dos monitoramentos. No entanto, os dados gerados ainda não estão devidamente estruturados para a geração de análises gráficas, que servirão para a pesquisa dos profissionais. Portanto, parte desenvolvimento está incompleto para realizar a geração dos gráficos seguindo os dados gerados pelo telemonitoramento realizado pela atual equipe do projeto Ufop em Ação.

Existem funcionalidades que seriam úteis de serem implantadas no *dashboard*, mas que não puderam ser implementadas devido a restrições de tempo. Entre essas funcionalidades, destacam-se: adição e remoção de gráficos em um boletim epidemiológico, controle de visibilidade dos gráficos para que o usuário possa escolher quais gráficos são exibidos na tela e filtragem de informações do painel, permitindo a busca por gráficos de um contexto de informação específico. Portanto, como trabalhos futuros, é importante que os usuários tenham maior controle sobre as informações apresentadas no painel de controle, possibilitando que personalizem o seu painel de acordo com suas necessidades.

No mais, pretende-se expandir a solução proposta para atender outros tipos de doenças que podem ser monitoradas remotamente, além de trazer formas de interação com o monitorado por meio da aplicação web e não somente por ligação telefônica.

# Referências

- ATHAYDE, K. R. N.; LIMA, G. R.; GAIO, A. V.; CASTRO, T. S.; SOARES, J.; STREME, L.; OLIVEIRA, A. A.; FILHO, E. d. S. Telemonitoramento de enfermagem no manejo do covid-19: Relato de experiência de uma operadora de saúde no estado de parana. 2020.
- BARDIN, L. *Análise de conteúdo*. São Paulo: Edições 70, 2016.
- BATISTA, T. S. C.; REBOUÇAS, D. A. de A.; ALMEIDA, L. A. da H.; SANTANA, R. S.; JUNIOR, J. F.; SANTOS, G. B.; SILVA, É. R.; TOFANI, P. S. Telemonitoramento em usuários do sus por estudantes de fisioterapia e fisioterapeutas durante a pandemia da covid-19: relato de experiência. *Brazilian Journal of Health Review*, v. 4, n. 3, p. 11071–11082, 2021.
- BOSE, S. *What is End To End Testing?* 2023. Disponível em: <<https://www.browserstack.com/guide/end-to-end-testing>>. Acesso em: em março de 2023.
- BUBBLE. *Bubble*. 2022. Disponível em: <<https://www.bubble.io/>>. Acesso em: em agosto de 2022.
- CAETANO, R.; SILVA, A. B.; GUEDES, A. C. C. M.; PAIVA, C. C. N. d.; RIBEIRO, G. d. R.; SANTOS, D. L.; SILVA, R. M. d. Desafios e oportunidades para telessaúde em tempos da pandemia pela covid-19: uma reflexão sobre os espaços e iniciativas no contexto brasileiro. *Cadernos de saúde pública*, SciELO Brasil, v. 36, 2020.
- CASCELLA MICHAEL RAJNIK, A. A. S. C. D. R. D. N. M. *Features, Evaluation, and Treatment of Coronavirus (COVID-19)*. 2022. Disponível em: <<https://www.ncbi.nlm.nih.gov/books/NBK554776/>>. Acesso em: em agosto de 2022.
- CHART.JS. *Chart.js*. 2023. Disponível em: <<https://www.chartjs.org/docs/>>. Acesso em: em março de 2023.
- CHARTS, G. *Data-Driven Documents*. 2023. Disponível em: <<https://d3js.org>>. Acesso em: em março de 2023.
- COMPONENTS, S. *Documentation*. 2023. Disponível em: <<https://styled-components.com/docs>>. Acesso em: em março de 2023.
- D3.JS. *Interactive charts for browsers and mobile devices*. 2023. Disponível em: <<https://developers.google.com/chart>>. Acesso em: em março de 2023.
- DOCS, M. W. *Canvas*. 2023. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/canvas>>. Acesso em: em março de 2023.
- FERNANDES, A. M. da R.; HENRIQUE, A. S.; LIEBEL, G.; DAZZI, R. L. S.; MEZADRI, T. A relevância dos dashboards para a gestão da saúde na pandemia causada pelo covid-19. *Brazilian Journal of Development*, v. 6, n. 6, p. 39263–39274, 2020.
- FIGMA. *Figma connects everyone in the design process so teams can deliver better products, faster*. 2022. Disponível em: <<https://www.figma.com>>. Acesso em: em outubro de 2022.

- FIREBASE. *Firestore: descubra para que serve, como funciona e como usar*. 2023. Disponível em: <<https://www.remessaonline.com.br/blog/firebase-descubra-para-que-serve-como-funciona-e-como-usar/>>. Acesso em: em março de 2023.
- FREECODECAMP. *A diferença entre um framework e uma biblioteca*. 2022. Disponível em: <<https://www.freecodecamp.org/portuguese/news/a-diferenca-entre-um-framework-e-uma-biblioteca/>>. Acesso em setembro de 2022.
- GEEKSFORGEES. *Which Libraries and Frameworks available in JavaScript?* 2022. Disponível em: <<https://www.geeksforgeeks.org/which-libraries-and-frameworks-available-in-javascript/>>. Acesso em: em setembro de 2022.
- GEEKSFORGEES. *Differences between Functional Components and Class Components in React*. 2023. Disponível em: <<https://www.geeksforgeeks.org/differences-between-functional-components-and-class-components-in-react/>>. Acesso em: em março de 2023.
- HOINASKI, F. *KPI. OS MAIS IMPORTANTES PARA O CONTROLE ESTRATÉGICO DO SETOR DE COMPRA*. 2023. Disponível em: <<https://ibid.com.br/blog/kpi/>>. Acesso em: em março de 2023.
- INTELIPOST. *A importância do dashboard para e-commerce*. 2023. Disponível em: <<https://www.intelipost.com.br/blog/a-importancia-do-dashboard-para-e-commerce/>>. Acesso em: em janeiro de 2023.
- KIELING, D. L.; SILVA, D. L. da; WITT, F. de M.; MAGNAGNANO, O. A. A importância da telemedicina no contexto da pandemia de covid-19. *Fag Journal Of Health (FJH)*, v. 3, n. 1, p. 90–97, 2021.
- LIMA, L. D. G.; TOMASCHEWSKI-BARLEM, J. G.; PALOSKI, G. d. R.; BARLEM, E. L. D.; ROCHA, L. P.; CASTANHEIRA, J. S. Atuação de estudantes de enfermagem em um serviço de telemedicina durante a pandemia covid-19. *Revista Gaúcha de Enfermagem*, SciELO Brasil, v. 42, 2021.
- MDN. *About JavaScript*. 2022. Disponível em: <[https://developer.mozilla.org/en-US/docs/Web/JavaScript/About\\_JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript)>. Acesso em: setembro de 2022.
- MIRANDA, M. L.; FERREIRA, V. S. C.; ANJOS, S. D. S. dos. Incorporação do telemonitoramento da covid-19 em um município do sul da bahia: obstáculos e facilidades. *Research, Society and Development*, v. 11, n. 5, p. e53911528287–e53911528287, 2022.
- MUI. *Switch*. 2023. Disponível em: <<https://mui.com/material-ui/react-switch/>>. Acesso em: em março de 2023.
- MULLER, M. E. *The Difference Between Telehealth and Telemedicine, and Where Remote Patient Monitoring Fits In*. 2022. Disponível em: <<https://www.healthrecoveryolutions.com/blog/the-difference-between-telehealth-and-telemedicine>>. Acesso em: em agosto de 2022.
- NEWTON, I. *Se eu vi mais longe, foi por estar sobre ombros de gigantes*. 1676. Disponível em: <<https://www.pensador.com/frase/MTMwMjY/>>. Acesso em: outubro de 2022.
- OPAS. *Folha informativa sobre COVID-19*. 2022. Disponível em: <<https://www.paho.org/pt/covid19>>. Acesso em: em agosto de 2022.

- PLOTLY. *Plotly JavaScript Open Source Graphing Library*. 2023. Disponível em: <<https://plotly.com/javascript/>>. Acesso em: em março de 2023.
- REACT. *A JavaScript library for building user interfaces*. 2022. Disponível em: <<https://reactjs.org>>. Acesso em: agosto de 2022.
- RETOOL. *Best React component libraries*. 2022. Disponível em: <<https://retool.com/blog/react-component-libraries/>>. Acesso em: outubro de 2022.
- ROCHA, P. P. D. F.; NASCIMENTO, S. M.; SOUSA, R. R. de. Um dashboard para visualização de insumos relacionados à covid-19 a dashboard for viewing inputs related to covid-19. *Brazilian Journal of Development*, v. 7, n. 8, p. 84457–84482, 2021.
- SELENIUM, A. *Selenium*. 2023. Disponível em: <<https://www.selenium.dev/about/>>. Acesso em: em março de 2023.
- STACKOVERFLOW. *Developer Survey*. 2021. Disponível em: <<https://insights.stackoverflow.com/survey/2021#most-popular-technologies-language>>. Acesso em: em setembro de 2022.
- SYNTACTICS. *Telehealth Website Design: Best Principles to Consider*. 2022. Disponível em: <<https://www.syntacticsinc.com/news-articles-cat/telehealth-website-design-best-principles-consider/>>. Acesso em: em setembro de 2022.
- TABLEAU. *What is a dashboard? A complete overview*. 2023. Disponível em: <<https://www.tableau.com/learn/articles/dashboards/what-is>>. Acesso em: em março de 2023.
- TESTBYTES. *Selenium IDE Tutorial For Beginner*. 2023. Disponível em: <<https://www.testbytes.net/blog/selenium-ide/>>. Acesso em: em março de 2023.
- ESCOLA DE FARMÁCIA DA UFOP. *Equipe do projeto UFOP em Ação*. 2020. Disponível em: <<https://escoladefarmacia.ufop.br/news/ufop-em-aç~ao>>. Acesso em: em agosto de 2022.
- REMESSA ONLINE. *Firebase*. 2023. Disponível em: <<https://www.remissaonline.com.br/blog/firebase-descubra-para-que-serve-como-funciona-e-como-usar/>>. Acesso em: em março de 2023.
- URTIGA, K. S.; LOUZADA, L. A.; COSTA, C. L. B. Telemedicina: uma visão geral do estado da arte. In: *IX Congresso Brasileiro de Informática em Saúde*. [S.l.: s.n.], 2004.
- VELLOSO, C. S. A atuação da vacinação no contexto da pandemia da covid-19. 2021.
- WATSON, O. J.; BARNESLEY, G.; TOOR, J.; HOGAN, A. B.; WINSKILL, P.; GHANI, A. C. Global impact of the first year of covid-19 vaccination: a mathematical modelling study. *The Lancet Infectious Diseases*, Elsevier, 2022.
- WHO. *WHO Coronavirus (COVID-19) Dashboard*. 2022. Disponível em: <<https://covid19.who.int>>. Acesso em: em agosto de 2022.

# **Apêndices**

# APÊNDICE A – Casos de testes

<p><b>C01 - Validar Login no web-app</b></p> <p>Caso de teste para fazer login na web-app.</p>
<p><b>Passos do caso de teste:</b></p> <ol style="list-style-type: none"> <li>1. Abrir a página de login da web-app.</li> <li>2. Inserir as credenciais válidas (nome de usuário e senha).</li> <li>3. Clicar no botão de login.</li> <li>4. Verificar se o usuário é redirecionado para a página inicial da web-app.</li> </ol>
<p><b>Resultado esperado:</b></p> <p>O usuário deve conseguir fazer login com sucesso na web-app e ser redirecionado para a página inicial.</p>
<p><b>Dados de teste:</b></p> <ul style="list-style-type: none"> <li>• Nome de usuário e senha válidos.</li> </ul>
<p><b>Pré-condições:</b></p> <ul style="list-style-type: none"> <li>• A web-app está acessível e em execução;</li> <li>• O usuário tem credenciais de login válidas.</li> </ul>
<p><b>Pós-condições:</b></p> <ul style="list-style-type: none"> <li>• O usuário está logado e pode acessar os recursos da web-app.</li> </ul>
<p><b>Ambiente de teste:</b></p> <ul style="list-style-type: none"> <li>• Sistema operacional: Sistema operacional utilizado</li> <li>• Navegador: Navegador utilizado</li> <li>• Conexão com a internet: Tipo de conexão com a internet utilizada</li> </ul>

Tabela A.1 – Caso de Teste C01

## C02 - Validar cadastro no web-app

Caso de teste para fazer cadastro na web-app.

### Passos do caso de teste:

1. Abrir a página de login da web-app.
2. Clicar em “Criar uma conta”.
3. Inserir e-mail válido para cadastro no campo de e-mail.
4. Inserir o nome no campo de nome.
5. Inserir uma senha com no mínimo seis dígitos no campo “Senha”.
6. Inserir a senha digitada no passo 5 no campo “Repetir senha”.
7. Clicar no campo de seleção “Selecione uma função”.
8. Selecionar a função “Entrevistador”.
9. Clicar no botão “Cadastrar”.

### Resultado esperado:

O usuário deve conseguir fazer login com sucesso na web-app e ser redirecionado para a página inicial.

### Dados de teste:

- Nome de usuário e senha válidos.

### Pré-condições:

- A web-app está acessível e em execução.
- O usuário tem credenciais de cadastro válidas.

### Pós-condições:

- O usuário está logado e pode acessar os recursos da web-app.

### Ambiente de teste:

- Sistema operacional: Sistema operacional utilizado
- Navegador: Navegador utilizado
- Conexão com a internet: Tipo de conexão com a internet utilizada

Tabela A.2 – Caso de Teste C02

<h2>C03 - Validar visualização dos dados na dashboard</h2> <p>Caso de teste para testar a visualização dos dados da dashboard.</p>
<p><b>Passos do caso de teste:</b></p> <ol style="list-style-type: none"><li>1. Abrir a página de login da web-app.</li><li>2. Inserir as credenciais válidas (nome de usuário e senha).</li><li>3. Clicar no botão de login.</li><li>4. Verificar se o usuário é redirecionado para a página inicial da web-app.</li><li>5. Clicar no botão “Acessar Dashboard”.</li></ol>
<p><b>Resultado esperado:</b></p> <p>O usuário deve conseguir visualizar os gráficos referentes aos dados do boletim epidemiológico gerado por meio dos monitoramentos remotos.</p>
<p><b>Dados de teste:</b></p> <ul style="list-style-type: none"><li>• Nome de usuário e senha válidos.</li></ul>
<p><b>Pré-condições:</b></p> <ul style="list-style-type: none"><li>• A web-app está acessível e em execução;</li><li>• O usuário tem credenciais de login válidas.</li></ul>
<p><b>Pós-condições:</b></p> <ul style="list-style-type: none"><li>• O usuário pode visualizar e interagir com os gráficos exibidos em tela.</li></ul>
<p><b>Ambiente de teste:</b></p> <ul style="list-style-type: none"><li>• Sistema operacional: Sistema operacional utilizado</li><li>• Navegador: Navegador utilizado</li><li>• Conexão com a internet: Tipo de conexão com a internet utilizada</li></ul>

Tabela A.3 – Caso de Teste C03

## C04 - Validar troca de gráfico no dashboard

Caso de teste para testar a troca do tipo dos gráficos do dashboard.

### Passos do caso de teste:

1. Abrir a página de login da web-app.
2. Inserir as credenciais válidas (nome de usuário e senha).
3. Clicar no botão de login.
4. Verificar se o usuário é redirecionado para a página inicial da web-app.
5. Clicar no botão “Acessar Dashboard”.
6. Clicar na lista de seleção “Tipo do gráfico” de um gráfico exibido em tela.
7. Selecionar o tipo “pizza”.
8. Selecionar o tipo “rosca”.
9. Selecionar o tipo “barra”.
10. Selecionar o tipo “linha”.

### Resultado esperado:

O usuário deve conseguir visualizar a troca do tipo do gráfico selecionado na lista de seleção do gráfico escolhido para a troca.

### Dados de teste:

- Nome de usuário e senha válidos.

### Pré-condições:

- A web-app está acessível e em execução;
- O usuário tem credenciais de login válidas.

### Pós-condições:

- O usuário pode visualizar o gráfico com o tipo referente ao tipo selecionado para troca.

### Ambiente de teste:

- Sistema operacional: Sistema operacional utilizado
- Navegador: Navegador utilizado
- Conexão com a internet: Tipo de conexão com a internet utilizada

Tabela A.4 – Caso de Teste C04

## C05 - Validar interação do tooltip no gráfico no dashboard

Caso de teste para validar a funcionalidade do *tooltip* do gráfico.

### Passos do caso de teste:

1. Abrir a página de login da web-app.
2. Inserir as credenciais válidas (nome de usuário e senha).
3. Clicar no botão de login.
4. Verificar se o usuário é redirecionado para a página inicial da web-app.
5. Clicar no botão “Acessar Dashboard”.
6. Posicionar o mouse no gráfico em cima da representação de um conjunto de dados de um gráfico.

### Resultado esperado:

O usuário deve conseguir visualizar um *tooltip* indicando a cor e a quantidade do conjunto de dados abaixo do cursor do mouse.

### Dados de teste:

- Nome de usuário e senha válidos.

### Pré-condições:

- A web-app está acessível e em execução;
- O usuário tem credenciais de login válidas.

### Pós-condições:

- O usuário pode visualizar por meio do *tooltip* a informação representada na ilustração do gráfico.

### Ambiente de teste:

- Sistema operacional: Sistema operacional utilizado
- Navegador: Navegador utilizado
- Conexão com a internet: Tipo de conexão com a internet utilizada

Tabela A.5 – Caso de Teste C05

## **C06 - Validar interação do vínculo de visibilidade da legenda com o seu conjunto de dados (gráficos de pizza/rosca/barra).**

Caso de teste para testar a interação para gerenciamento da visibilidade do conjunto de dados com conjunto indicado na legenda.

### **Passos do caso de teste:**

1. Abrir a página de login da web-app.
2. Inserir as credenciais válidas (nome de usuário e senha).
3. Clicar no botão de login.
4. Verificar se o usuário é redirecionado para a página inicial da web-app.
5. Clicar na legenda de um gráfico de pizza/rosca/barra.

### **Resultado esperado:**

O usuário deve conseguir gerenciar a visualização de um conjunto de dados por meio da legenda que referencia esse conjunto no gráfico.

### **Dados de teste:**

- Nome de usuário e senha válidos.

### **Pré-condições:**

- A web-app está acessível e em execução;
- O usuário tem credenciais de login válidas.

### **Pós-condições:**

- O usuário pode visualizar o gráfico com ou sem a visibilidade do conjunto de dados da legenda clicada pelo usuário.

### **Ambiente de teste:**

- Sistema operacional: Sistema operacional utilizado
- Navegador: Navegador utilizado
- Conexão com a internet: Tipo de conexão com a internet utilizada

Tabela A.6 – Caso de Teste C06

## C07 - Validar expansão do *accordion*

Caso de teste para testar a execução do *accordion* para gerenciar a visibilidade dos gráficos de um boletim.

### Passos do caso de teste:

1. Abrir a página de login da web-app.
2. Inserir as credenciais válidas (nome de usuário e senha).
3. Clicar no botão de login.
4. Verificar se o usuário é redirecionado para a página inicial da web-app.
5. Clicar no botão “Acessar Dashboard.
6. Clicar no primeiro *accordion* exibido.
7. Clicar no segundo *accordion* exibido.
8. Clicar no primeiro *accordion* exibido.

### Resultado esperado:

O usuário deve conseguir fazer login com sucesso na web-app e ser redirecionado para a página inicial.

### Dados de teste:

- Nome de usuário e senha válidos.

### Pré-condições:

- A web-app está acessível e em execução;
- O usuário tem credenciais de login válidas.

### Pós-condições:

- O usuário deve conseguir gerenciar a visualização em tela de um conjunto de gráficos de um boletim.

### Ambiente de teste:

- Sistema operacional: Sistema operacional utilizado
- Navegador: Navegador utilizado
- Conexão com a internet: Tipo de conexão com a internet utilizada

Tabela A.7 – Caso de Teste C07