



**UFOP**

**Universidade Federal de Ouro Preto  
Instituto de Ciências Exatas e Aplicadas  
Departamento de Computação e Sistemas**

# **UM SISTEMA PARA MARCAÇÃO E AGENDAMENTO DE CONSULTAS COM ESPECIALISTAS NA ÁREA DA SAÚDE MUNICIPAL**

**Manoel Farias Paixão Júnior**

## **TRABALHO DE CONCLUSÃO DE CURSO**

**ORIENTAÇÃO:  
George Henrique Godim da Fonseca**

**Outubro, 2022  
João Monlevade–MG**

**Manoel Farias Paixão Júnior**

**UM SISTEMA PARA MARCAÇÃO E  
AGENDAMENTO DE CONSULTAS COM  
ESPECIALISTAS NA ÁREA DA SAÚDE  
MUNICIPAL**

Orientador: George Henrique Godim da Fonseca

Monografia apresentada ao curso de Sistemas de Informação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina “Trabalho de Conclusão de Curso II”.

**Universidade Federal de Ouro Preto**

**João Monlevade**

**Outubro de 2022**



## FOLHA DE APROVAÇÃO

**Manoel Farias Paixão Júnior**

### **Um sistema para marcação e agendamento de consultas com especialistas na área da saúde municipal**

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

Aprovada em 4 de novembro de 2022

#### Membros da banca

Doutor - George Henrique Godim da Fonseca - Orientador(a) Universidade Federal de Ouro Preto  
Doutor - Carlos Henrique Gomes Ferreira - Universidade Federal de Ouro Preto  
Doutor - Rafael Frederico Alexandre - Universidade Federal de Ouro Preto

George Henrique Godim da Fonseca, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 08/03/2023



Documento assinado eletronicamente por **George Henrique Godim da Fonseca, PROFESSOR DE MAGISTERIO SUPERIOR**, em 08/03/2023, às 17:18, 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 **0486714** e o código CRC **13424D5D**.

*Este trabalho é dedicado aos meus pais e ao meu avô, pelo apoio incondicional e pela total confiança em mim e nas minhas escolhas.*

# Agradecimentos

Agradeço primeiramente à Deus, por todas as minhas conquistas.

À minha mãe Eulália de Souza Vitor Paixão, por sempre acreditar e mim.

Ao meu pai, Manoel Farias Paixão e meu avô, Valdivino Vitor, pelo apoio em todas as decisões, mesmo em momentos difíceis.

Agradeço também ao meu orientar, Dr. George Henrique Godim da Fonseca, pelo excelente trabalho e direcionamento durante o projeto.

Agradecimento especial ao meu grande amigo, José Geraldo Duarte Fagundes, que me ajudou muito, e foi essencial para o desenvolvimento e conclusão do projeto.

À Joice Camille, que muito me ajudou e nunca deixou de acreditar em mim.

E finalmente agradeço a todos os professores do curso, obrigado por participar e contribuir no meu crescimento pessoal e acadêmico. profissional.

*“Viva como se fosse morrer amanhã. Aprenda como se fosse viver para sempre.”*

*- Mahatma Gandhi.*

# Resumo

A gestão municipal é dividida em vários setores para melhor organização e distribuição de recurso provenientes do Governo Federal. Um desses setores é o da saúde que de longe é a área mais importante e a que demanda mais atenção dentre todos os outros setores das prefeituras municipais. O trabalho em questão tem como objetivo desenvolver uma aplicação web, onde será possível fazer as marcações e agendamentos de consultas, cadastro de pacientes e de médicos em uma única plataforma web. O agendamento de consultas online é um dos grandes avanços tecnológicos que a telemedicina possibilitou para pacientes e profissionais de saúde, garantindo uma maior celeridade. Visando trazer essa praticidade, o *MedPoint* foi desenvolvido fazendo uso de ferramentas modernas de desenvolvimento web. Foi utilizada a linguagem de programação *Javascript* juntamente com o *React*, que é uma das suas bibliotecas mais utilizadas atualmente para desenvolvimento de interfaces web. O *Firebase* também tem um papel muito importante na aplicação, pois é através dele que é possível fazer a autenticação dos usuários e onde se localiza o banco de dados da plataforma. O *MedPoint* possui duas áreas logadas, uma para os usuários comuns, ou seja, os pacientes que tem o simples objetivo de marcar as suas consultas e a área do usuário administrador que diferente do comum, tem mais privilégios dentro da plataforma, podendo fazer cadastro de postos de saúde e de médicos. É possível ver o resultado da interface, juntamente com suas funcionalidades no endereço eletrônico <[www.medpointnovaera.com.br](http://www.medpointnovaera.com.br)>.

**Palavras-chaves:** *MedPoint*. Marcação de consultas. Sistema Web.

# Abstract

Municipal management is divided into several sectors for better organization and distribution of funds from the Federal Government. One of these sectors is health, which by far is the the most important area and the one that demands the most attention among all the other sectors of the municipal prefectures.

The work in question aims to develop a web application, where possible to make appointments and appointments, register patients and doctors on a single web platform. Online appointment scheduling is one of the great technological advances that telemedicine has made possible for patients and professionals health, ensuring greater speed.

Aiming to bring this practicality, MedPoint was developed using tools modern web development. Javascript programming language was used along with React, which is one of its most used libraries today for development of web interfaces. Firebase also plays a very important role in the application, because it is through it that it is possible to authenticate users and where to locates the platform database.

MedPoint has two logged areas, one for common users, that is, patients which has the simple objective of marking your queries and the admin user area that different from the common one, has more privileges within the platform, being able to register health posts and doctors. You can see the result of the interface, along with its functionalities at the electronic address <[www.medpointnovaera.com.br](http://www.medpointnovaera.com.br)>.

**Key-words:** *MedPoint*. Appointment booking; web system.



# Lista de ilustrações

Figura 1 – Desenvolvimento do protótipo . . . . .	18
Figura 2 – Lista de médicos do sistema <i>Doctor's Office</i> . . . . .	21
Figura 3 – Formulário para confirmação de consulta <i>Doctor's Office</i> . . . . .	21
Figura 4 – Painel de controle do administrador <i>Doctor's Office</i> . . . . .	22
Figura 5 – Calendário administrativo <i>MedPonit</i> . . . . .	23
Figura 6 – Calendário da plataforma . . . . .	24
Figura 7 – Calendário dos horários . . . . .	24
Figura 8 – Formulário de envio . . . . .	25
Figura 9 – Modal de solicitações . . . . .	26
Figura 10 – Painel do usuário Firebase . . . . .	28
Figura 11 – Painel Authentication Firebase . . . . .	29
Figura 12 – Editor Visual Studio Code . . . . .	30
Figura 13 – Diagrama de casos de uso . . . . .	31
Figura 14 – Diagrama de sequencia . . . . .	32
Figura 15 – Coleções no banco de dados Firestore . . . . .	33
Figura 16 – Coleções no banco de dados Firestore . . . . .	33
Figura 17 – Tela de login <i>MedPoint</i> . . . . .	35
Figura 18 – Tela de <i>login</i> com Facebook . . . . .	36
Figura 19 – Tela de cadastro ( <i>signup</i> ) . . . . .	36
Figura 20 – Tela de recuperação de senha . . . . .	37
Figura 21 – Tela do e-mail de recuperação de senha . . . . .	37
Figura 22 – Tela do calendário de solicitações . . . . .	38
Figura 23 – Tela do formulário de solicitação . . . . .	38
Figura 24 – Tela do calendário de solicitações . . . . .	39
Figura 25 – Tela de exclusão de solicitação . . . . .	40
Figura 26 – Tela do calendário de solicitações . . . . .	40
Figura 27 – Tela do calendário de solicitações . . . . .	41
Figura 28 – Tela do calendário de solicitações . . . . .	42
Figura 29 – Calendário modo administrador . . . . .	43
Figura 30 – Modal de solicitações do calendário . . . . .	44
Figura 31 – Modal de solicitações da barra lateral . . . . .	44
Figura 32 – Solicitação de consulta . . . . .	45
Figura 33 – Cores dos avatares . . . . .	46
Figura 34 – Tela de médicos . . . . .	46
Figura 35 – Tela de adicionar médico . . . . .	47

Figura 36 – Tela de postos de saúde . . . . .	47
Figura 37 – Tela de adicionar posto . . . . .	48
Figura 38 – Tela do calendário filtrado . . . . .	49
Figura 39 – Tela de login <i>mobile</i> . . . . .	50
Figura 40 – Tela do calendário <i>mobile</i> . . . . .	51

# Lista de abreviaturas e siglas

API            Application Programming Interface

VSCODE       Visual Studio Code

UI             User Interface

UX             User Experience

UFOP          Universidade Federal de Ouro Preto

TFD           Tratamento Fora do Domicilio

SI             Sistemas de Informação

SUS            Sistema Único de Saúde

ADMIN        Administrador

PSR            Posto de Saúde Regional

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
1.1	O problema de pesquisa	13
1.2	Objetivos	14
1.2.1	Objetivos específicos	14
1.3	Justificativa	14
1.4	Estrutura do trabalho	15
<b>2</b>	<b>CONCEITOS</b>	<b>16</b>
2.1	Sistema de informação	16
2.2	Padrões arquiteturais	16
2.3	Processo de software	17
2.3.1	Métodos ágeis	17
2.3.2	Engenharia de requisitos	17
2.3.3	Prototipação	18
2.3.4	Projeto e implementação de software	18
2.3.5	Teste de software	19
2.3.6	Testes automatizados de <i>User Interface (UI)</i>	19
2.4	Considerações finais	19
<b>3</b>	<b>TRABALHOS RELACIONADOS</b>	<b>20</b>
3.1	<i>Doctor's Office</i>	20
3.2	WE Engenharia	23
<b>4</b>	<b>METODOLOGIA</b>	<b>26</b>
4.1	Visão geral do sistema	26
4.2	Ferramentas e tecnologias utilizadas	27
4.2.1	<i>Typescript</i>	27
4.2.2	<i>React.js</i>	27
4.2.3	<i>Material-UI</i>	27
4.2.4	<i>Firebase</i>	28
4.2.5	<i>Github</i>	29
4.2.6	<i>Vercel</i>	29
4.2.7	<i>Visual Studio Code</i>	30
4.3	Detalhes do sistema	30
4.3.1	Diagrama de casos de uso	31
4.3.2	Diagrama de sequencia	31

<b>4.4</b>	<b>Banco de dados Firestore</b> . . . . .	<b>32</b>
4.4.1	Requisitos funcionais . . . . .	34
4.4.2	Requisitos não-funcionais . . . . .	34
<b>5</b>	<b>RESULTADOS</b> . . . . .	<b>35</b>
<b>5.1</b>	<i>Login</i> . . . . .	<b>35</b>
<b>5.2</b>	<b>Sistema modo usuário</b> . . . . .	<b>38</b>
<b>5.3</b>	<b>Sistema modo administrador</b> . . . . .	<b>42</b>
<b>6</b>	<b>CONCLUSÃO</b> . . . . .	<b>52</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>53</b>

# 1 Introdução

O *MedPoint* é um sistema de marcações de exames e consultas, com o intuito de celeridade e mais presteza no âmbito da saúde regional. A plataforma oferece uma maneira de controle de agenda médico-paciente de forma prática, usual e uma desobstrução no acúmulo e na demora para marcação das consultas. Criou-se uma avaliação visando compreender a aplicabilidade de padrões do software em funcionamento para desktop de marcação de consultas e exames na região de Nova Era, como estudo de caso, com o objetivo de compreensão de sua usabilidade, e novas ideias de como os padrões de projeto e padrões arquiteturais podem auxiliar no desenvolvimento de uma aplicação web.

## 1.1 O problema de pesquisa

O setor da saúde encontra-se abarcado com imensa demanda de pessoas que fazem contato com o setor da saúde de suas cidades, acarretando algumas precariedades e falta de agilidade em certos processos dentro da área. Mais especificamente, na marcação de exames e agendamento de consultas com especialistas e profissionais da área médica. Utilizando como exemplo a cidade de Nova Era, o processo em questão ainda é feito de maneira manual através de cadernos de anotações, onde é possível notar grande óbice na finalização de uma etapa muito importante para o atendimento ao público. A forma com a qual é feita a organização dos pacientes e identificar de maneira precisa quais médicos estão disponíveis nas suas respectivas datas de atendimento resulta em atraso nas marcações. É importante frisar que devemos levar em consideração os possíveis erros humanos ao fazer os agendamentos de maneira manual devido à falta de atenção ou distração no momento da marcação, afinal, o setor de marcação de consultas chamado de tratamento fora do domicílio (TFD), que segundo o (PORTALNOVAERA, 2014) é o atendimento de saúde a ser prestado pela Secretaria Municipal de Saúde aos usuários do Sistema único de Saúde (SUS), é um local onde se tem um grande fluxo de pessoas a procura de atendimento para solução de problemas de saúde. Devido a essa grande demanda, o TFD recebe algumas reclamações na demora de atendimento, pois o processo em questão se mantém de forma arcaica por muitos anos, tal cenário é propício para uma solução com a qual possa agilizar esse atendimento, ajudando assim os funcionários do setor e todos os cidadãos com os quais fazem uso do sistema de saúde municipal.

## 1.2 Objetivos

O presente trabalho consiste em modelar e desenvolver um sistema web que forneça as ferramentas necessárias para a utilização de maneira rápida e pratica, onde será possível realizar agendamentos de consultas com especialistas, cadastro de pacientes e de médicos em uma única plataforma on-line.

### 1.2.1 Objetivos específicos

O trabalho possui os seguintes objetivos específicos:

- Compreender as necessidades de informação do setor de saúde e de seus pacientes;
- Diminuir a quantidade de erros durante o processo de agendamento;
- Diminuir o tempo gasto durante o processo de marcação de consultas;
- Agilizar e facilitar a marcação das consultas;
- Permitir aos pacientes a realização de solicitações de consultas sem precisar ir até o posto de saúde;
- Permitir ao usuário administrador do sistema, aceitar ou recusar as solicitações recebidas de maneira prática e rápida.

## 1.3 Justificativa

Diante das dificuldades enfrentadas pelos pacientes e até mesmo os profissionais da área de saúde da prefeitura municipal de João Monlevade e Nova Era, notou-se vasta necessidade de uma ferramenta que otimizasse o tempo para realização das atividades relacionadas a marcação das consultas, além de automatizar e dar suporte às diversas tarefas referentes ao desempenho do TFD. O *MedPoint* surgiu, perante uma ideia do desenvolvimento de uma aplicação para gerenciamento, cadastro de pacientes e controles de agendas médicas, que após a sua conclusão, pode ser implantada para solucionar as demandas já apresentadas.

## 1.4 Estrutura do trabalho

O trabalho está organizado em 5 capítulos. O Capítulo 1 apresenta a introdução do trabalho, assim como os objetivos que levaram a promoção deste. O Capítulo 2 apresenta as definições básicas para o desenvolvimento do projeto e destaca algumas aplicações que apresentam funcionalidades semelhantes às do sistema proposto. O Capítulo 3 apresenta uma revisão da literatura, contendo uma discussão sobre outros aplicativos relacionados ao *MedPoint*. O Capítulo 4 descreve a metodologia, as ferramentas, modelagem do sistema, requisitos e diagramas, o protótipo da aplicação, escolhas tecnológicas, linguagens dentre outros aspectos utilizadas para o desenvolvimento do sistema. No Capítulo 5 são expostas análises e resultados da aplicação desenvolvida. Por fim, o Capítulo 6 é composto pelas considerações finais.



## 2 Conceitos básicos e revisão bibliográfica

Este capítulo apresenta os conceitos básicos e essenciais para a compreensão do contexto do presente trabalho e uma revisão da literatura acerca de conceitos, tecnologia e informações relacionadas ao desenvolvimento deste projeto

### 2.1 Sistema de informação

De acordo com LAUDON (2011), para entender totalmente os sistemas de informação, é necessário conhecer as suas dimensões mais amplas, a organização, a humana, a tecnológica e também seu poder de fornecer soluções para os desafios e problemas no âmbito empresarial. Laudon ainda defende que um sistema de Informação (SI) pode ser tecnicamente definido como um conjunto de componentes inter-relacionados que coletam, processam, armazenam e distribuem informações com o objetivo de dar apoio a uma organização nas atividades de tomada de decisões, coordenação e controle. Cabe a esses sistemas também auxiliar os colaboradores na visualização de assuntos complexos, criação de novos produtos e análise de problemas. Laudon estrutura a hierarquia da administração, definindo sua composição em níveis, cada nível tem diferentes necessidades de informações, de acordo com suas responsabilidades sendo os principais clientes dos sistemas de informação.

### 2.2 Padrões arquiteturais

Parafraseando BUSCHMANN e MEUNIER (1996), padrões arquiteturais representam os padrões de alto nível no sistema de projetos. Assim, padrões arquiteturais expressam e especificam a organização estrutural de uma aplicação, desta forma, toda atividade subsequente de desenvolvimento é dirigida por esta estrutura. Com isso, propriedades específicas globais no sistema são alcançadas, como exemplo a adaptabilidade da interface do usuário, tudo isso graças à ajuda de cada padrão arquitetural.

Dentre os padrões arquiteturais mais comuns o *model-view-controller* (modelo-visão-controle), separa o projeto do software em três camadas independentes: o modelo (manipulação da lógica de dados), a visão (a interface do usuário) e o controlador (fluxo de aplicação). O modelo *client-server* (cliente-servidor), o processamento da informação se divide em módulos e processos distintos, este tipo de arquitetura de software é bastante usado em aplicativos de bancos e e-mail. Já o modelo *layers* (camadas), tem os módulos e componentes do software organizados em camadas de funcionalidades, que podem

ser desconstruídas em diferentes serviços. Este padrão é mais usado em programas de e-commerce.

## 2.3 Processo de software

Conforme [SOMMERVILLE \(2011\)](#), o processo de software é uma sequência de atividades que produz um produto de software. Desse modo, para todos os processos de software existem quatro atividades básicas, que são:

- Especificação de software: atividade onde é realizada a definição do software a ser produzido junto aos clientes.
- Desenvolvimento de software: execução de implementação do projeto do software.
- Validação de software: operação de verificação do software com intuito de testificar os desejos do cliente.
- Evolução de software: ação em que é realizada modificações devido a mudanças de requisitos por parte do cliente ou mercado.

### 2.3.1 Métodos ágeis

De acordo com [SOMMERVILLE \(2011\)](#), os métodos ágeis baseiam-se em uma abordagem incremental para a especificação, o desenvolvimento e a entrega do software. Eles são mais adequados ao desenvolvimento de aplicativos nos quais os requisitos de sistema mudam rapidamente durante o processo de desenvolvimento. Este método destinam-se a entregar o software rapidamente aos clientes, em funcionamento, e estes podem, em seguida, propor alterações e novos requisitos a serem incluídos nas iterações posteriores do sistema. O método também têm como objetivo reduzir a burocracia do processo, evitando qualquer trabalho de valor duvidoso de longo prazo e qualquer documentação que provavelmente nunca será usada.

### 2.3.2 Engenharia de requisitos

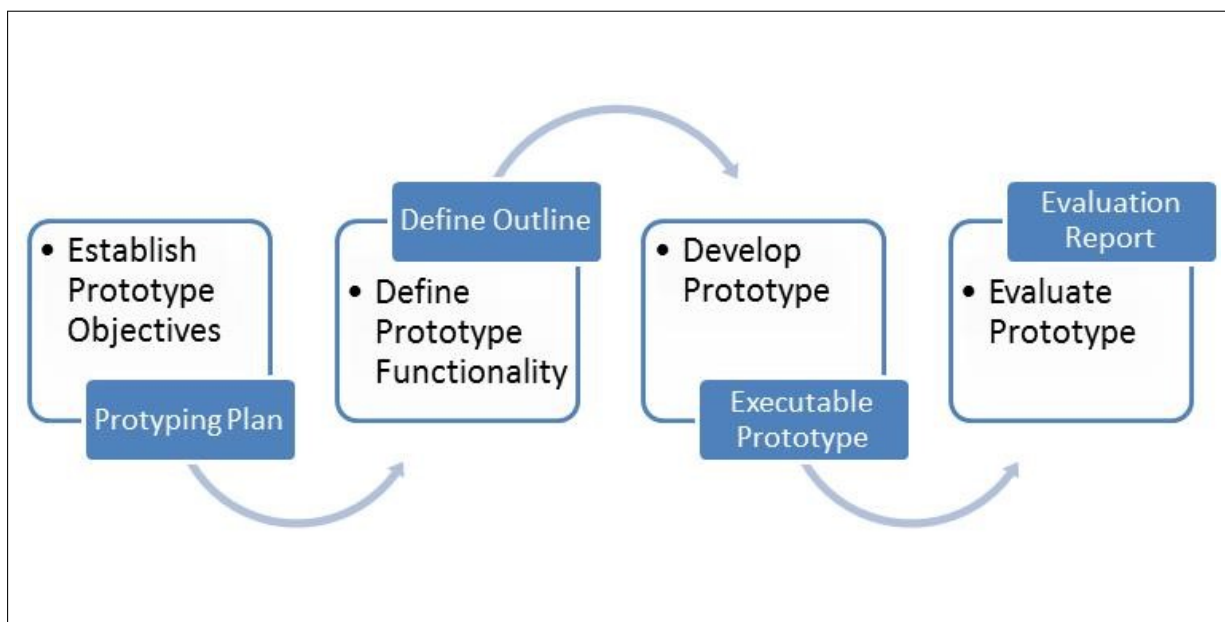
De acordo com [SOMMERVILLE \(2011\)](#), a engenharia de requisitos ou especificação de software é a ocupação que tem como escopo compreender e delimitar os serviços requisitados do sistema juntamente com a identificação de limitações relacionada à uma funcionalidade ou desenvolvimento do sistema. Por isso, a engenharia de requisitos fornece instrumentos apropriados para o entendimento daquilo que o cliente requisitou, a partir da análise das necessidades, avaliação da viabilidade, negociação de uma solução razoável, especificação da solução sem ambiguidades, validação da especificação e gerenciamento

das necessidades ao passo que são transformadas em um sistema (THAYER; BAILIN; DORFMAN, 1997) e (PRESSMAN, 2019).

### 2.3.3 Prototipação

Um protótipo é uma versão inicial de um software utilizado para difundir conceitos e testar opções de projeto discrepantes, pode ser considerado a primeira versão de um produto. Ademais, possibilita o descobrimento acentuado em razão do problema em conjuntura às suas possíveis soluções. Não apenas, um protótipo de software pode ser utilizado em um processo de desenvolvimento de software a fim de ajudar a antecipar as mudanças de requisitos, tal como a utilização no processo de engenharia de requisitos para ajudar na elicitacão e validacão dos requisitos. Ademais, os protótipos possibilitam que usuários vejam como o sistema dá suporte a seu trabalho, além de permitir obter novas ideias para requisitos e revelar erros e omissões nos requisitos propostos (SOMMERVILLE, 2011). Como exemplo visual, a Figura 01 apresenta um modelo de processo de desenvolvimento de protótipo, em questão é possível observar que é necessário inicialmente, estabelecer os objetivos do protótipo, em seguida definir as suas funcionalidades, para então prosseguir com seu desenvolvimento e avaliacaão.

Figura 1 – Desenvolvimento do protótipo



Fonte: SOMMERVILLE(2011)

### 2.3.4 Projeto e implementacão de software

Um projeto de software é uma descriçã do arranjo a ser implementado o qual abrange os modelos e estruturas de dados e as interfaces entre os componentes do sistema. (SOMMERVILLE, 2011), Desta maneira, as atividades no processo de projeto variam

em relação ao tipo de sistema desenvolvido, podem ser modelados e documentados. Já a implementação é o meio de construção de um sistema a partir de uma especificação. O projeto de software tem início logo após a análise e modelagem dos requisitos de software, sendo a última ação a que antecipa a etapa de construção no qual é realizada a geração do código e testes (PRESSMAN, 2019). É por ele que a qualidade é assimilada na engenharia de software, deste modo, o projeto fornece representações de software que permitem avaliações em termos de qualidade. Sem um projeto, o desenvolvimento de software pode gerar um sistema instável, difícil de ser testado, cuja a qualidade não pode ser avaliada.

### 2.3.5 Teste de software

Conforme SOMMERVILLE (2011), o teste de software tem como objetivo mostrar que um programa se comporta da maneira que foi proposto e para descobrir defeitos antes do uso. Para ser breve, o processo de teste possui dois objetivos diversos: O primeiro objetivo leva a testes de validação, onde o testador espera que a partir do uso de um conjunto de casos de teste que refletem a maneira esperada de uso do sistema, este execute corretamente; no que lhe concerne, o segundo objetivo estimula aos testes de defeitos, nestes casos os testes são projetados para revelar defeitos.

### 2.3.6 Testes automatizados de *User Interface* (UI)

Os testes de *User Interface* têm por objetivo atestar que os requisitos funcionais e recursos de um aplicativo estejam funcionando corretamente. Estes testes podem ser realizados utilizando-se de um usuário para testar todos os casos manualmente, certificando que as UI estão funcionando como esperado. Todavia, o teste realizado manualmente retardaria a conclusão, e o usuário estaria exposto a falhas humanas, como a possibilidade de não perceber um erro durante a execução de um teste. Por conseguinte, os testes automatizados são uma grande aposta para solucionar tais problemas, essa abordagem garante que os testes sejam mais confiáveis e eficientes (MAINIKAR, 2017). A afirmação de Maikar pode ser confirmada, tendo em vista a utilização dos *unit testing* (testes unitários), ferramenta bastante utilizada em projetos de médio e grande porte. Testes esses, que podem ser implementados tanto para verificação de funções, quanto de interfaces de componentes.

## 2.4 Considerações finais

Neste capítulo foram apresentados conceitos relacionados ao desenvolvimento deste trabalho, como a fundamentação da plataforma web utilizada no estudo, tecnologias relacionadas ao desenvolvimento, o processo de desenvolvimento de software, apresentação dos padrões arquiteturais correlatos, dentre outros conceitos.

## 3 Trabalhos relacionados

Este capítulo apresenta dois sistemas com finalidades similares ao *MedPoint*, são eles o sistema *Doctor's Office* e o sistema de marcação da WE Engenharia. O capítulo tem o intuito de mostrar as diferenças de cada sistema em questão. Nesse capítulo será possível observar que apesar de serem utilizados para marcação de consultas, veremos que cada sistema possui suas particularidades.

### 3.1 *Doctor's Office*

O *Doctor's Office* é um dos principais programas médicos existentes no mercado, sendo responsável por inúmeros casos de sucesso no uso do computador para auxiliar o trabalho médico e administrativo. Atualmente o sistema possui mais de 20.000 licenças, entre milhares de médicos individuais, clínicas de todos os portes e vários hospitais com centenas de médicos e usuários utilizando (PES, 2022).

O *Doctor's Office* é um programa de computador destinado a informatizar consultórios, clínicas médicas e hospitais, sendo construído visando atingir dois objetivos principais: ser de fácil utilização e possuir recursos que permitam a informatização completa do ambiente médico.

As aplicações do sistema no ambiente médico são inúmeras. Vão desde manutenção de um banco de dados com as fichas dos pacientes, passando pelo controle da agenda do médico, o monitoramento das despesas e receitas do consultório ou o arquivamento de fichas bibliográficas, indo até a manipulação de imagens (PES, 2022).

O sistema possui diversas funcionalidades, porém daremos atenção à marcação de consultas, modalidade com a qual podemos comparar com o *MedPoint*. No *Doctor's Office* a marcação se dá início com uma lista de médicos cadastrados, onde o paciente deve procurar o médico desejado através dos filtros iniciais, conforme mostrado na [Figura 02](#).

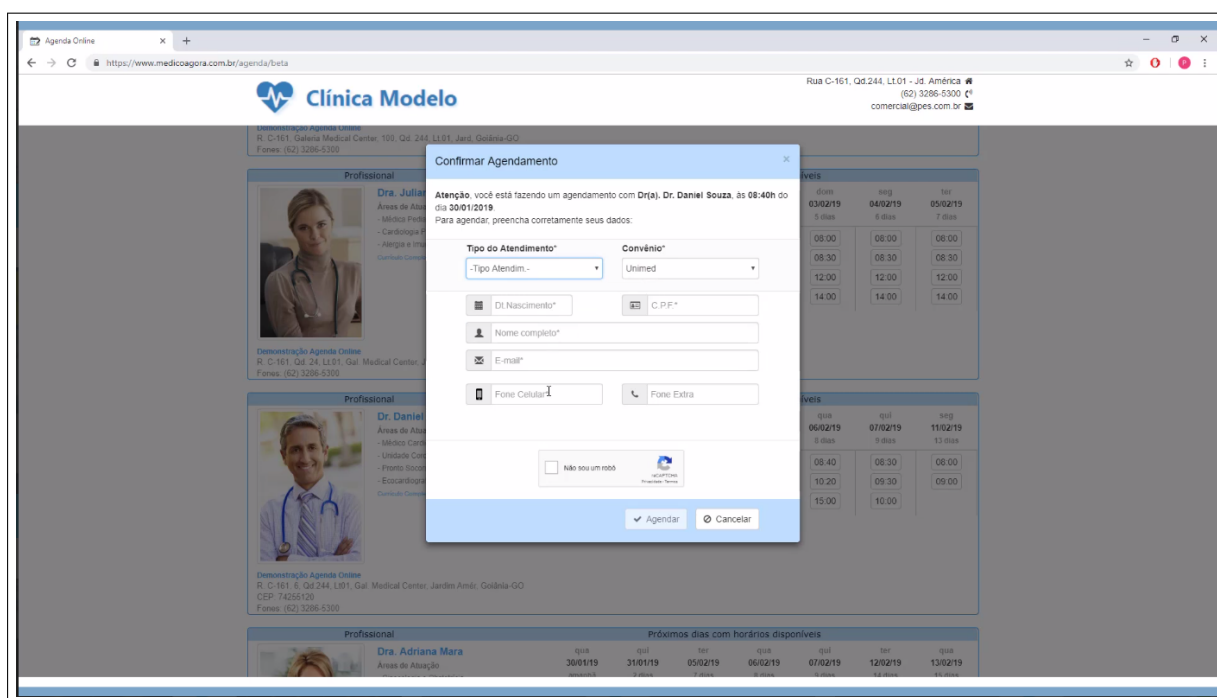
Figura 2 – Lista de médicos do sistema *Doctor's Office*



Fonte: site oficial PES.

Após selecionar o medico desejado, o paciente então informa seus dados e algumas informações necessários que serão enviadas para o administrador da continuidade na marcação, conforme apresentado na Figura 03.

Figura 3 – Formulário para confirmação de consulta *Doctor's Office*

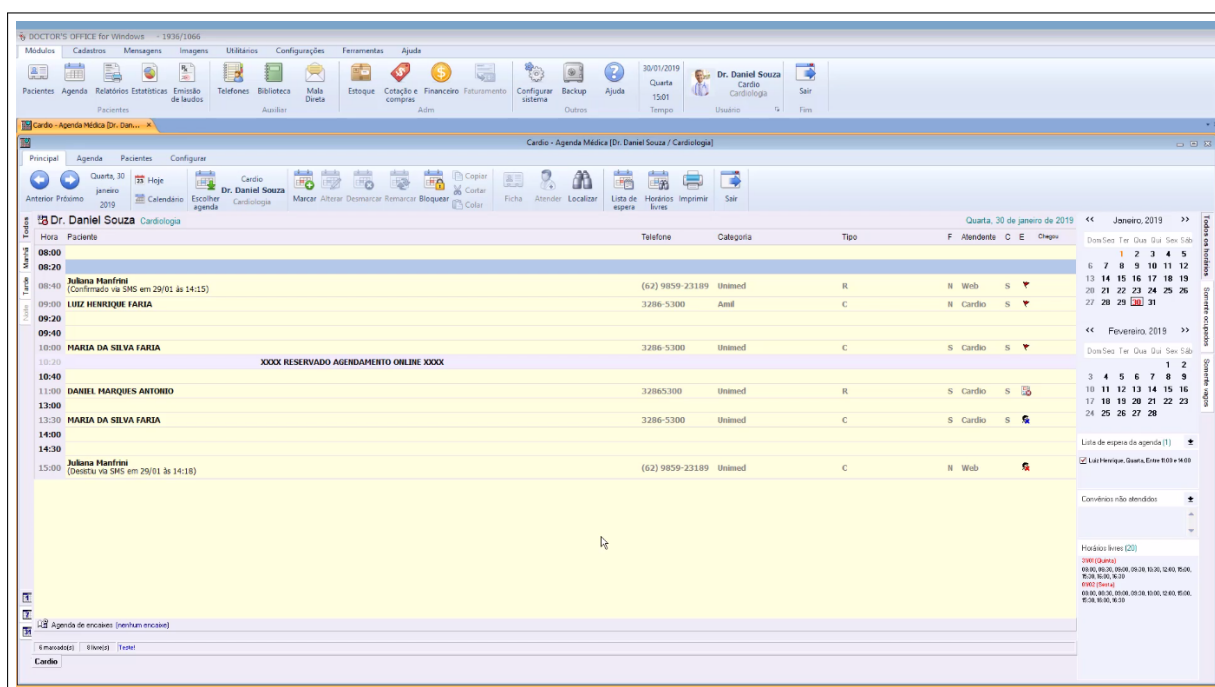


Fonte: site oficial PES.

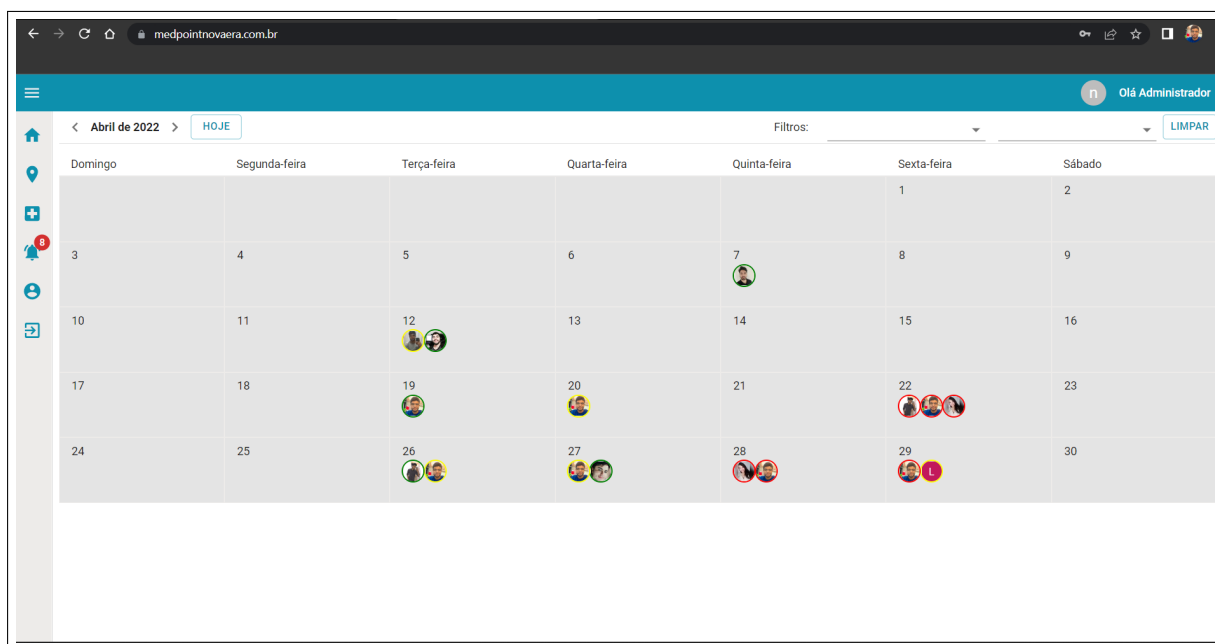
Diferente do *Doctor's Office* no sistema *MedPoint*, é necessário fazer o login para tentar marcar a consulta e também como tem seu foco na rede de saúde municipal das cidades do interior, tem como início de marcação das consultas o dia desejado pelo paciente, e somente em seguida o mesmo selecionará o médico disponível para aquele respectivo dia.

Dando continuidade na marcação no sistema *Doctor's Office*, o próximo passo é o recebimento dessas informações pelo administrador do sistema, onde pode visualizar, alterar ou excluir essa consulta. É possível observar na [Figura 04](#), que o painel do sistema *Doctor's Office* possui muitas informações na tela, deixando assim o painel um pouco poluído, diferente do painel da *MedPoint* que possui um painel administrador bastante limpo, facilitando assim a usabilidade pelo usuário, conforme podemos ver na [Figura 05](#).

Figura 4 – Painel de controle do administrador *Doctor's Office*



Fonte: site oficial PES.

Figura 5 – Calendário administrativo *MedPonit*

Fonte: site MedPonit.

Assim que é recebida uma nova solicitação de consulta, o *MedPoint* informa através de uma notificação no painel lateral do sistema, que existem consultas que necessitam de respostas do administrador, fazendo com que nenhuma solicitação de marcação de consulta possa passar despercebida e evitando assim que o paciente fique sem resposta para sua solicitação.

Devido ao foco nas marcações, pode-se concluir que o *MedPoint* tenha uma maior facilidade de uso comparado ao sistema *Doctor's Office*, que tem integrado ao módulo de marcações, muitas outras funcionalidades, fazendo com que o sistema seja mais difícil de utilizar, necessitando assim de um treinamento específico para o total domínio das ferramentas pelos usuários.

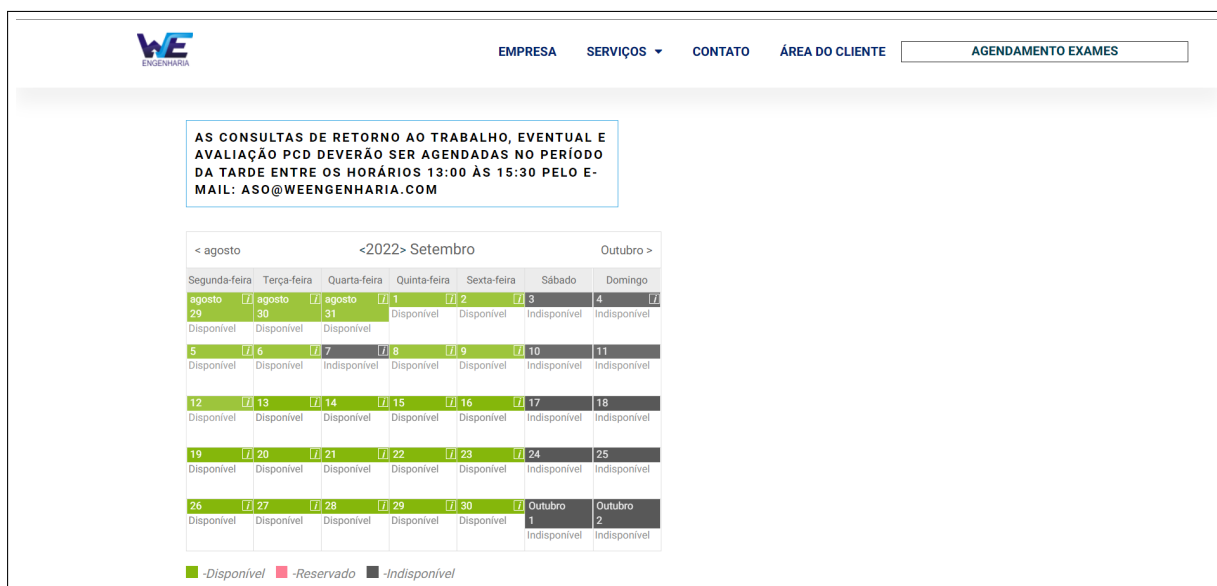
## 3.2 WE Engenharia

A WE Engenharia atua em consultoria nas áreas de meio ambiente, segurança do trabalho e saúde ocupacional, a empresa está há 19 anos no mercado nacional. Integra tecnologia, profissionalismo e ampla experiência de profissionais com sólida capacitação técnica para prestar um serviço diferenciado (WEENGENHARIA, 2022).

Como já mencionado anteriormente, um dos seguimentos com o qual a empresa atua, é na área da saúde, prestando serviços de exames ocupacionais para outras empresas. Se tratando das marcações dos exames, a organização faz uso de uma sistema próprio para este fim, como apresentado na [Figura 06](#).



Figura 6 – Calendário da plataforma



Fonte: site WE Engenharia.

A plataforma da WE Engenharia, assim como o *MedPoint*, também faz uso de um calendário para mostrar os dias disponíveis de atendimento, a serem selecionados pelo usuário. Assim que essa ação acontece é exibido um quadro, onde o usuário seleciona o horário disponível para o agendamento, conforme mostrado na [Figura 07](#).

Figura 7 – Calendário dos horários



Fonte: site WE Engenharia.

Após a seleção do dia e do horário pelo usuário, é exibido ao lado do calendário um pequeno formulário, para ser preenchido com os dados essenciais do paciente, para só então a solicitação ser enviada, conforme mostrado na [Figura 08](#). Diferente da plataforma da WE Engenharia, o *MedPoint* não necessita do preenchimento de formulário para o envio da solicitação de consulta, devido ao fato, do usuário ter acesso ao calendário de marcação, somente perante login de acesso, com isso, o sistema já possui os dados necessários para o envio da solicitação, não precisando assim de um novo preenchimento de formulário.

Figura 8 – Formulário de envio

**Preencha o formulário para concluir a reserva!**

**Encontro:** Data Setembro 14, 2022

**Hora:** 13:15

Nome Completo\*

Função\*

Telefone\*

Email da empresa\*

Data de Nascimento\*

Fonte: site WE Engenharia.

O sistema da WE Engenharia e o *MedPoint*, possuem características muito parecidas e processos semelhantes, porém quando se trata de praticidade, o *MedPoint* sai na frente, devido a facilidade de fazer login na plataforma com o uso da conta do *Facebook* ou *Gmail* do usuário, economizando assim mais tempo, já que não é necessário preencher nenhum formulário.

## 4 Metodologia

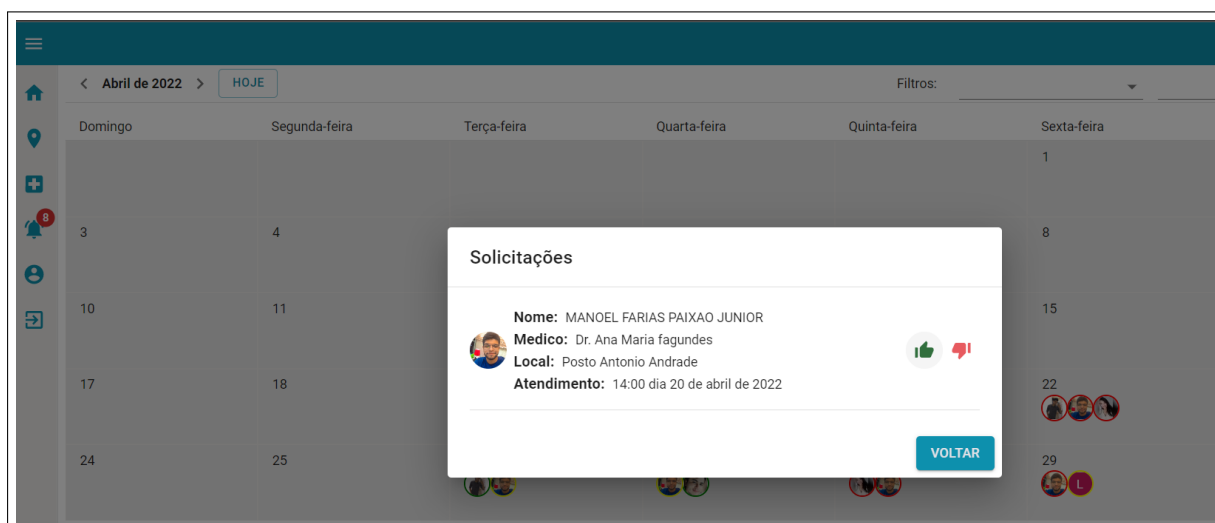
Este capítulo descreve a metodologia e as ferramentas, requisitos e propriedades utilizadas para a realização deste trabalho. Para alcançar todos os objetivos e gerar um resultado satisfatório, a metodologia seguiu as seguintes etapas:

- Levantamento dos requisitos;
- Análise de softwares relacionados;
- Desenvolvimento do modelo de dados;
- Desenvolvimento do software.

### 4.1 Visão geral do sistema

O *MedPoint* tem a finalidade de facilitar a marcação de consultas com especialistas de maneira rápida e prática, sem a necessidade do paciente ter que ir até o posto de saúde para efetuar essa marcação, além de agilizar o processo de marcação, o sistema também trás mais segurança aos dados dos pacientes e histórico das consultas, uma vez que o mesmo se encontram em um banco de dados na nuvens. Através do sistema é possível fazer o cadastro de médicos e postos de saúde, analisar as solicitações feitas pelos usuários, tendo a possibilidade de aceitar ou não essa solicitação, conforme mostrado na [Figura 09](#).

Figura 9 – Modal de solicitações



Fonte: site MedPoint

## 4.2 Ferramentas e tecnologias utilizadas

Essa seção caracteriza as principais linguagens de programação, ferramentas e tecnologias utilizadas no desenvolvimento do trabalho. As linguagens de programação utilizadas foram o *Typescript* e *React*

### 4.2.1 *Typescript*

É importante lembrar que o *JavaScript* é uma linguagem de programação interpretada estruturada, de script em alto nível com tipagem dinâmica fraca e multiparadigma. O *TypeScript* mantém uma relação incomum com o *JavaScript*. O *TypeScript* oferece todos os recursos do *JavaScript* e uma camada adicional sobre eles: o sistema de tipos do *TypeScript*. Por exemplo, o *JavaScript* fornece primitivos de linguagem com tipo *string*(texto) ou tipo *number*(número), mas não verifica se você os atribuiu consistentemente, já o *TypeScript* faz.

Isso significa que seu código *JavaScript* de trabalho existente também é código *TypeScript*. O principal benefício do *TypeScript* é que ele pode destacar comportamentos inesperados em seu código, diminuindo a chance de bugs ([TYPESCRIPT, 2022](#)).

### 4.2.2 *React.js*

O *React* é Uma biblioteca *JavaScript* para criar interfaces de usuário. O *React* faz com que a criação de *UIs*(Interfaces de usuário) interativas seja uma tarefa fácil. Crie views simples para cada estado na sua aplicação, e o *React* irá atualizar e renderizar de forma eficiente apenas os componentes necessários na medida em que os dados mudam. Views declarativas fazem com que seu código seja mais previsível e simples de depurar, ([REACTJS, 2022](#))

### 4.2.3 *Material-UI*

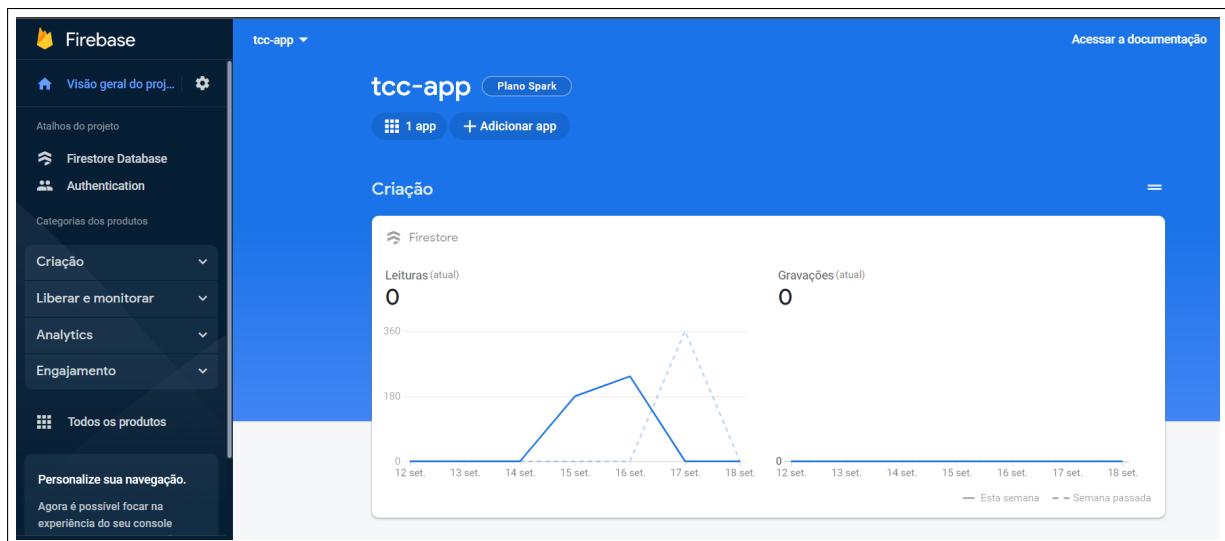
Material UI é uma biblioteca de componentes *React* de código aberto que implementa o Material Design do *Google*. Ele inclui uma coleção abrangente de componentes pré-construídos que estão prontos para uso na produção imediatamente. A interface do usuário do material é bonita por design e apresenta um conjunto de opções de personalização que facilitam a implementação de seu próprio sistema de design personalizado em cima de nossos componentes, ([MATERIAL-UI, 2022](#)).

#### 4.2.4 Firebase

O Firebase é uma plataforma de desenvolvimento de aplicativos que ajuda você a criar e desenvolver aplicativos e jogos que os usuários adoram. Apoiado pelo Google e confiável por milhões de empresas em todo o mundo, (FIREBASE, 2022a)

A plataforma é muito completa e intuitiva, possui varias ferramentas para auxilio no desenvolvimento de aplicações *web* e *mobile*. A mesma é composta por um painel lateral com suas ferramentas disponíveis para os desenvolvedores, e uma *dashboard* central, onde é informado ao usuário algumas informações importantes sobre sua aplicação, como podemos observar na Figura 10, é informado a quantidade de leitura e gravações no Firestore, que é um das principais ferramentas do Firebase.

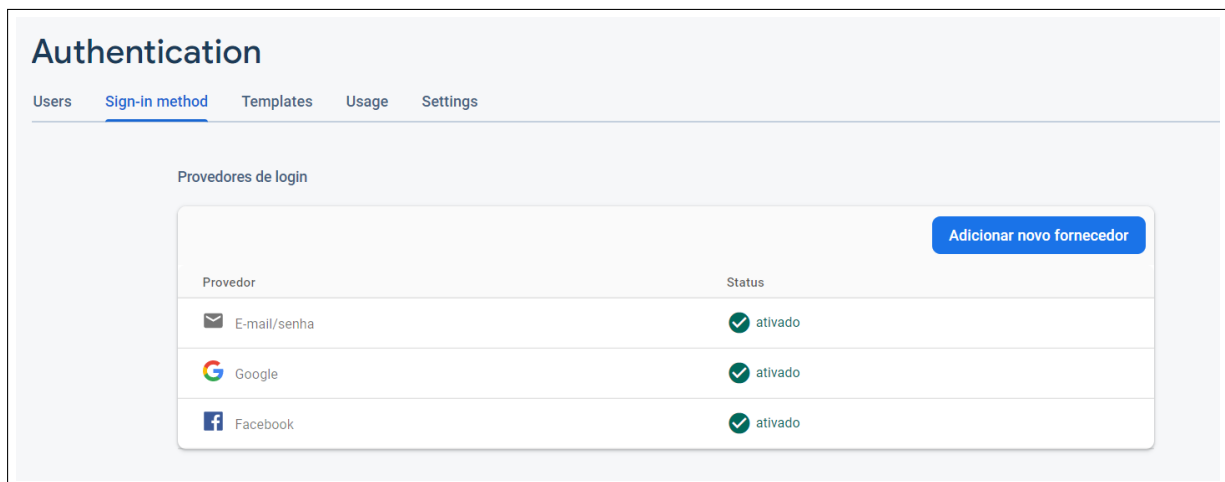
Figura 10 – Painel do usuário Firebase



Fonte: site Firebase

Sendo um elemento chave no sistema, o *MedPoint* faz uso de duas ferramentas do Firebase, são eles o *Authentication* e o *Firestore* já citado anteriormente. Cada ferramenta tem a sua funcionalidade particular, toda parte de segurança de acesso e *login* do sistema, foi feito através do *Authentication*, tanto os acessos via redes sociais, como o *Facebook* e *Google*, quanto os acessos por usuário e senha, como podemos ver na Figura 11.

Figura 11 – Painel Authentication Firebase



Fonte: site Firebase

O Firestore por sua vez, também possui participação muito relevante na aplicação, pois se trata do banco de dados do *MedPoint*, a ferramenta consegue salvar, alterar e excluir, as informações do seu banco NoSQL de maneira rápida e pratica, a ferramenta também disponibiliza uma biblioteca do próprio Firebase, para ser utilização durante o desenvolvimento das aplicações, com o intuito de facilitar a interação do *Front end* do sistema com seu banco de dados Firestore.

#### 4.2.5 Github

Para controle de versionamento, foi utilizado o *GitHub*. Amplamente utilizado por desenvolvedores em todo o mundo, o *GitHub* cria repositórios com códigos fonte onde pessoas conseguem acessar, colaborar, criar chamados e tirar dúvidas ([GITHUB, 2022](#)).

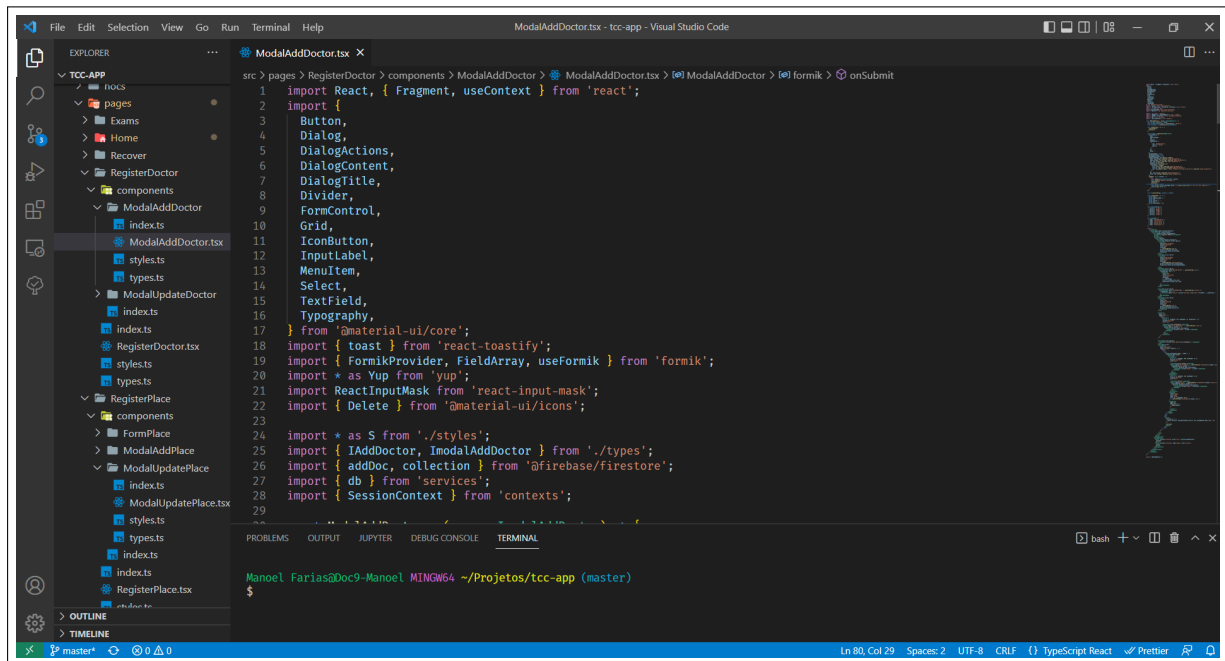
#### 4.2.6 Vercel

Vercel é uma plataforma para *frameworks* de *front-end* e sites estáticos , construída para se integrar ao seu conteúdo *headless*, comércio ou banco de dados. O Vercel fornece uma experiência de desenvolvedor sem atritos para cuidar das coisas difíceis: implantar instantaneamente, dimensionar automaticamente e fornecer conteúdo personalizado em todo o mundo. A ferramenta também permitimos que as equipes de *front-end* desenvolvam, visualizem e enviem experiências de usuário agradáveis, onde o desempenho é o padrão ([VERCEL, 2022](#)).

### 4.2.7 Visual Studio Code

O Visual Studio Code é um editor de código-fonte leve, mas poderoso, que é executado em sua área de trabalho e está disponível para Windows, macOS e Linux. Ele vem com suporte integrado para JavaScript, TypeScript e Node.js e possui um rico ecossistema de extensões para outras linguagens e ambientes de execução (como C++, Java, Python, PHP, Go, .NET), (VSCODE, 2022).

Figura 12 – Editor Visual Studio Code



Fonte: disponibilizado pelo aluno

## 4.3 Detalhes do sistema

Nessa sessão serão apresentados alguns detalhes do sistema, como os diagramas de caso de uso e de sequencia, juntamente com os requisitos funcionais, não-funcionais e o banco de dados NoSQL do Firebase utilizado na aplicação.

### 4.3.1 Diagrama de casos de uso

A Figura 13 apresenta o diagrama de casos de Uso, que foi desenvolvido com o suporte da ferramenta Astah. Existem dois atores na *MedPoint* e eles foram definidos como Usuário e Administrador sendo apenas o Administrador capaz de ter acesso as funcionalidades gerais do sistema.

- Usuário: Ator responsável por fazer *login* ou cadastrar-se na aplicação, fazer a solicitação de consulta, verificar seu histórico de solicitações e alterar suas solicitações em aberto.
- Administrador: Ator responsável pelo gerenciamento do sistema, cadastrar, alterar e excluir médicos e postos de saúde, aceitar ou negar as solicitações de consultas enviadas pelo usuário.

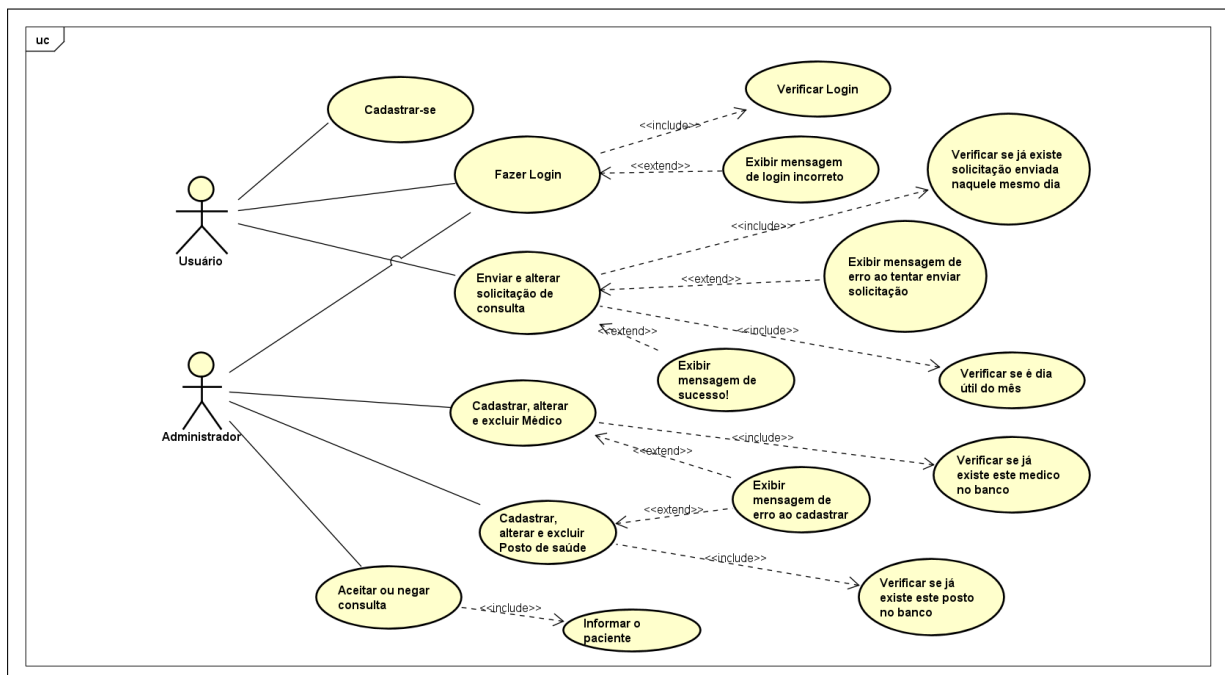


Figura 13 – Diagrama de casos de uso

### 4.3.2 Diagrama de sequencia

A Figura 14 apresenta o diagrama de sequencia, que assim como o diagrama anterior também foi desenvolvido com o suporte da ferramenta Astah. Neste diagrama, é exibida a maneira com a qual o usuário interage com o sistema e suas funcionalidades.

O usuário tentar fazer login no sistema, caso o mesmo não possua, ele necessitará fazer seu cadastro, somente após a confirmação cadastral o usuário poderá acessar o sistema e prosseguir com a solicitação de consulta.



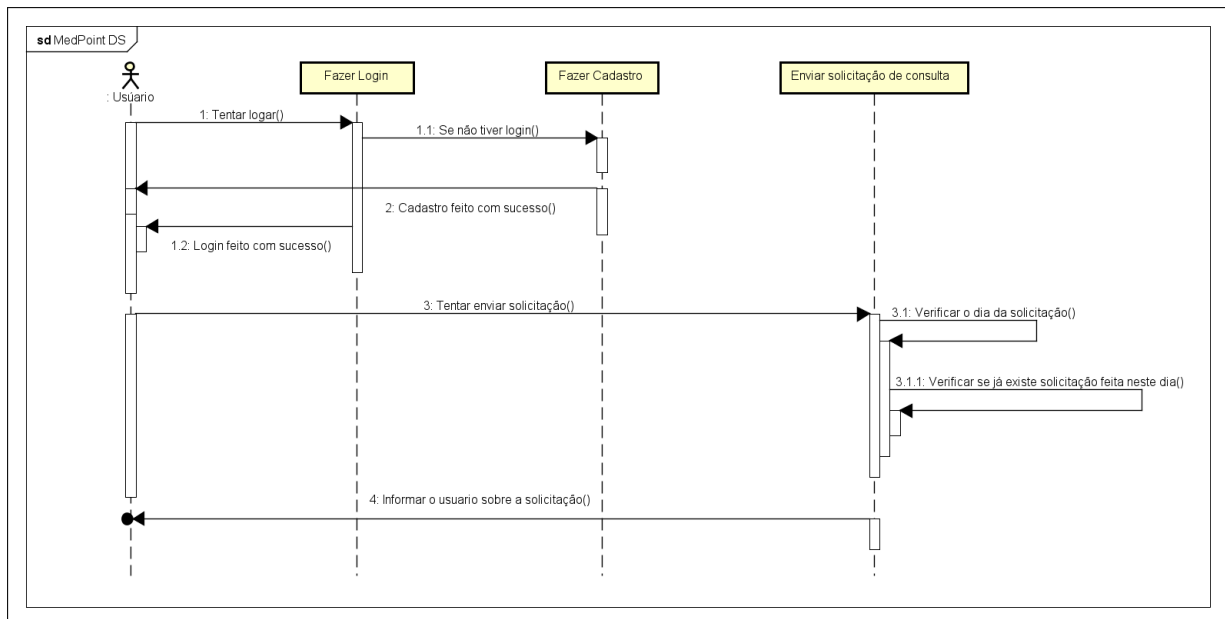


Figura 14 – Diagrama de sequencia

## 4.4 Banco de dados Firestore

O Cloud Firestore é um banco de dados flexível e escalonável para desenvolvimento focado em dispositivos móveis, Web e servidores pelo Firebase e do Google Cloud. Como o Firebase Realtime Database, ele mantém seus dados em sincronia em aplicativos cliente usando listeners em tempo real. Além disso, oferece suporte off-line para dispositivos móveis e Web para que você possa criar aplicativos responsivos que funcionem independentemente da latência da rede ou da conectividade com a Internet (FIREBASE, 2022b).

Tendo em vista a maneira flexível com a qual os bancos de dados NoSQL pode proporcionar, o sistema foi projetado em cima dessa facilidade, e todas as chamadas feitas ao banco de dados são executadas direto no *front-end* e executadas pelas próprias funções disponibilizadas pelo Firebase. O banco de dados Firestore é dividido em coleções, e essas coleções tem seus documentos e campos próprios, ou seja, cada coleção é como se fosse uma tabela com seus determinados campos onde serão armazenadas as informações, conforme mostrado no exemplo da Figura 15

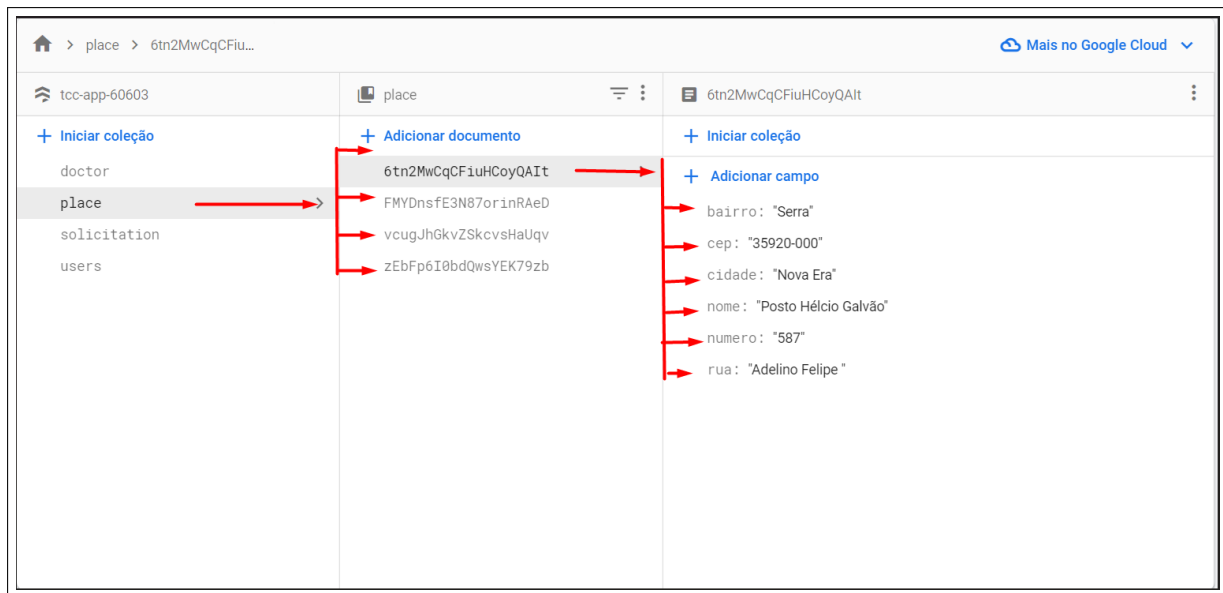


Figura 15 – Coleções no banco de dados Firestore

O *MedPoint* faz uso de apenas 4 coleções para armazenar suas informações. A hierarquia do Firestone acontece da esquerda para a direita, então cada coleção possui seu documento com seu código de identificação, formado automaticamente e de maneira aleatória. Seguindo a ordem, é possível observar que cada documento possui seus campos, onde são guardados as informações do sistema. No Firestone é possível fazer cadastro de *Array*, com suas determinadas informações, caso aja necessidade de implementação, como exemplo do *MedPoint*, que adiciona ao campo atendimento um *array*, com as informações do determinado dia do atendimento, conforme mostrado na [Figura 16](#) .

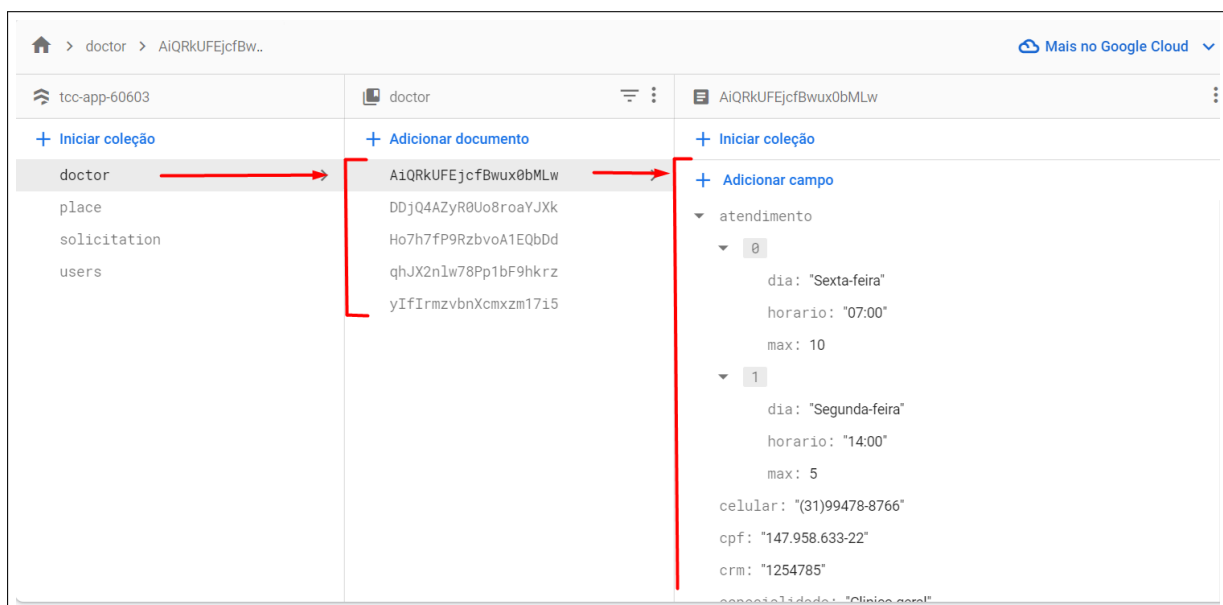


Figura 16 – Coleções no banco de dados Firestore

#### 4.4.1 Requisitos funcionais

Os requisitos funcionais do sistema são:

- permitir o cadastro dos usuários;
- permitir o cadastro, alteração, exclusão consultas tanto dos médicos quanto de postos de saúde;
- permitir o envio de solicitações de consultas pelo usuário;
- permitir a aprovação ou não da solicitação perante o usuário administrador;
- disponibilizar a possibilidade de cancelamento da solicitação enviada pelo próprio usuário;
- disponibilizar a possibilidade de fazer login através da conta do Google e do Facebook;
- possibilidade de filtrar as solicitações por medico e posto pelo usuário administrador;

#### 4.4.2 Requisitos não-funcionais

Os requisitos não-funcionais do sistema são:

- disponibilidade, o sistema deve ficar disponíveis para ser acessado de qualquer dispositivo desde que o usuário tenha acesso a internet;
- confiabilidade, o sistema deve mostrar apenas informações com as quais realmente foram feitas pelos usuários;
- desempenho, os sistema deve conseguir finalizar os processos solicitados pelos usuários de maneira rápida e fluida;
- segurança, o sistema deve conter maneiras de impedir usuário com más intenções tenham acesso.

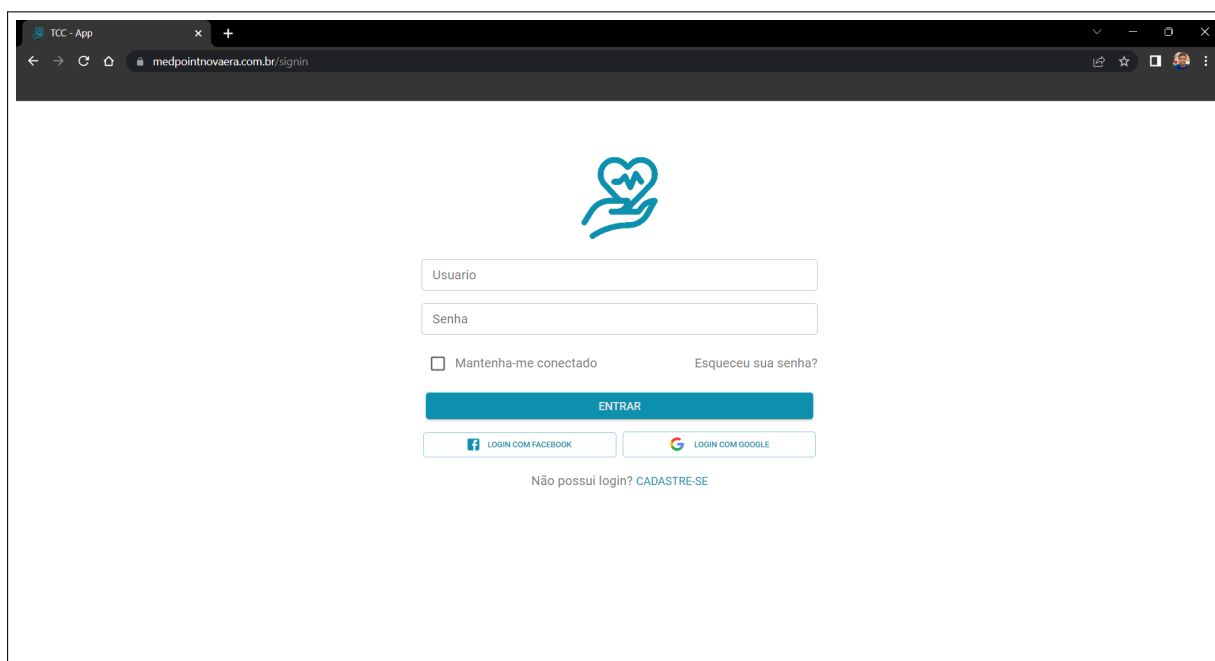
## 5 Resultados

Esse capítulo apresenta a versão atual desenvolvida do *software*. O *MedPoint* foi dividido em 3 módulos principais: *Login*, Sistema modo usuário e Sistema modo administrador. Cada módulo desenvolvido é apresentado nas sessões a seguir.

### 5.1 *Login*

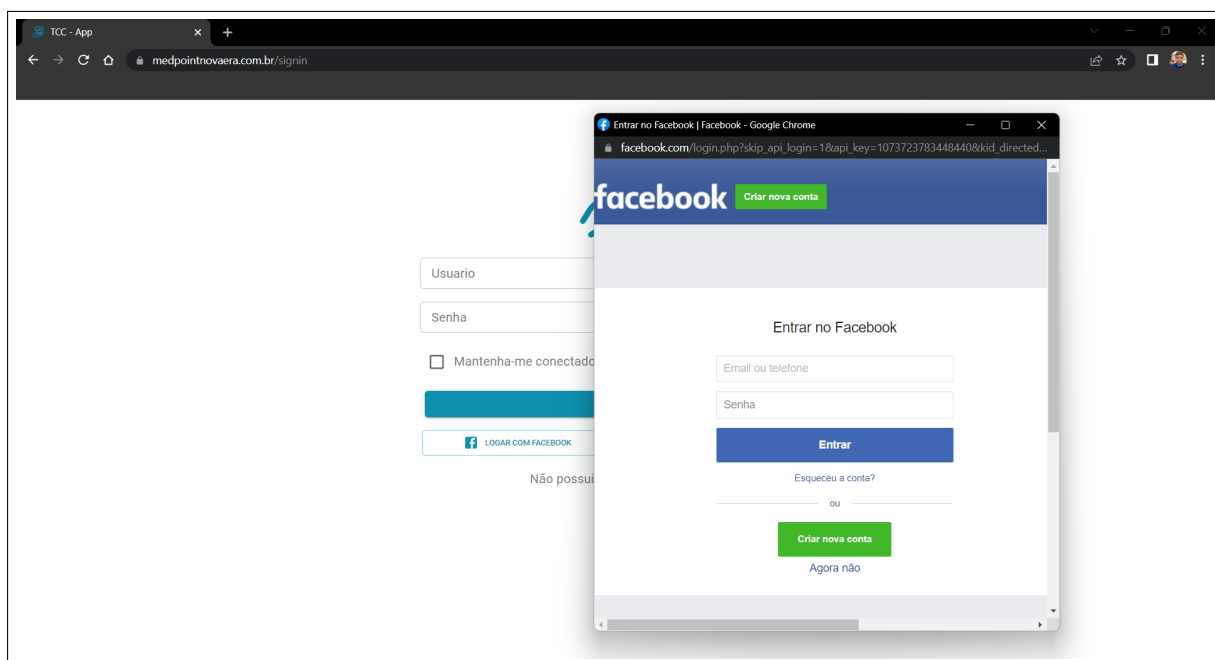
Visando a segurança dos usuários, é necessário fazer *login* para ter acesso as funcionalidades do sistema, conforme mostrado na [Figura 17](#).

Figura 17 – Tela de login *MedPoint*

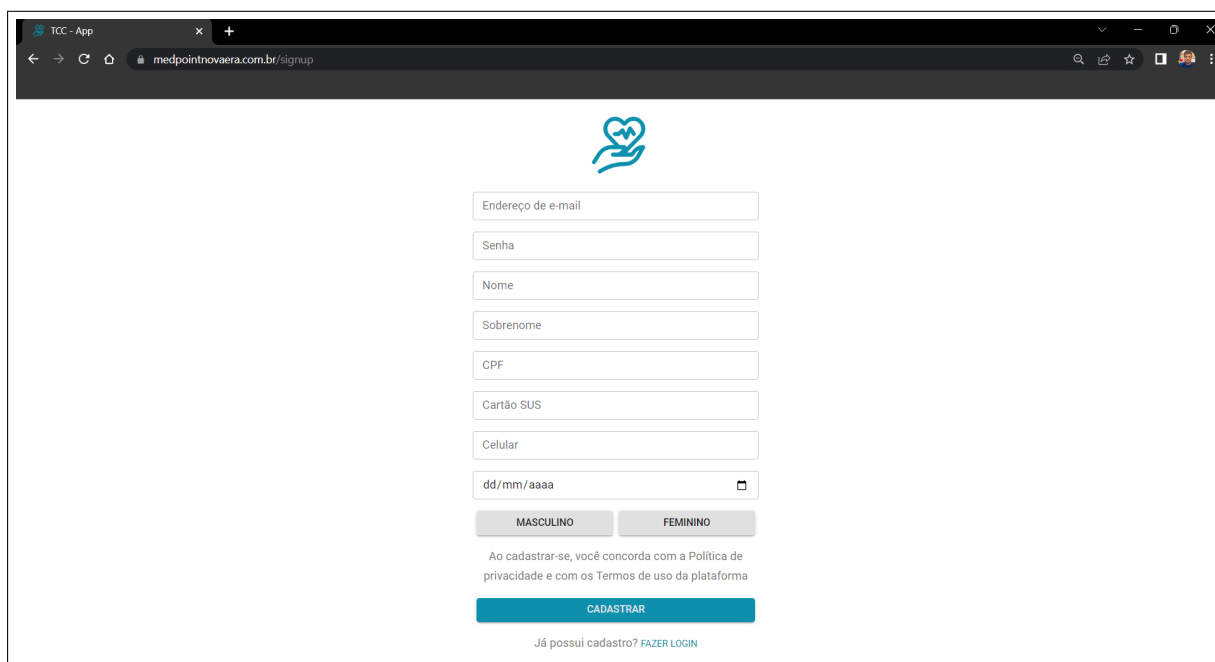


Fonte: Site *MedPoint*

O *MedPoint* disponibiliza três maneiras para se efetuar o *login*, a primeira pode ser feita de maneira rápida e pratica, fazendo uso do botão de *login* com o *Facebook*. Logo em seguida será mostrado uma tela, onde será necessário colocar as credenciais caso o mesmo não tenha feito *login* com a rede social anteriormente, como mostrado na [Figura 18](#). Seguindo a mesma logica, a segunda maneira de efetuar o *login* é fazendo uso do botão de *login* com o *Google*.

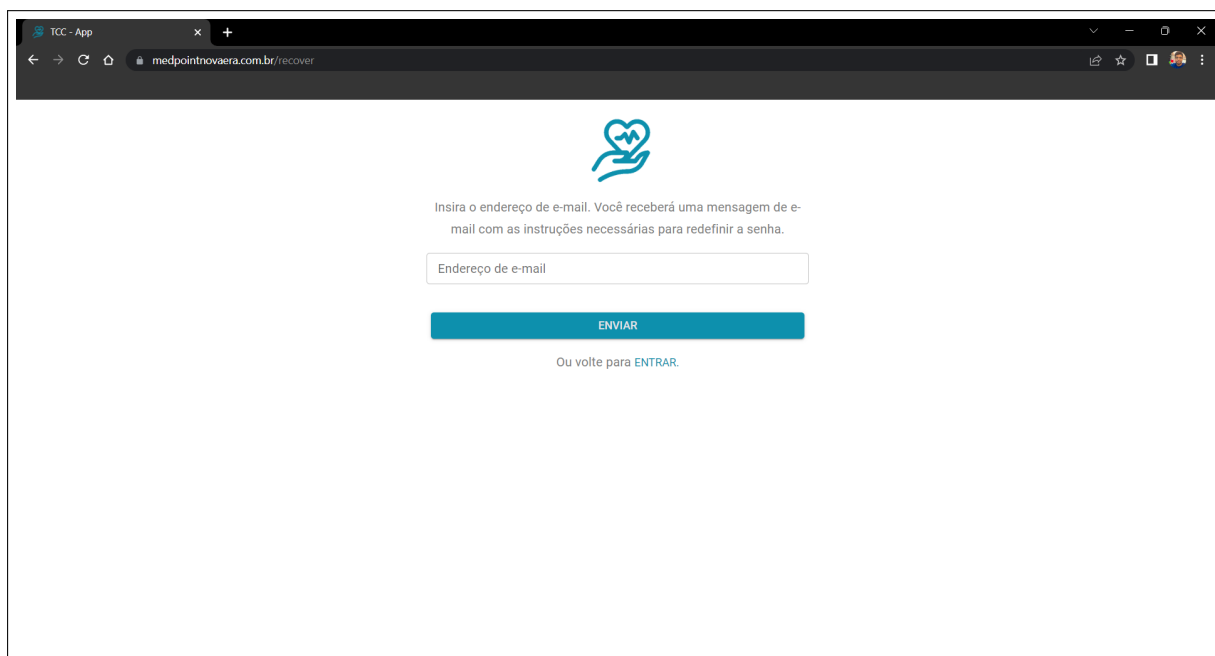
Figura 18 – Tela de *login* com FacebookFonte: Site *MedPoint*

A terceira maneira de acessar o sistema, é informando o usuário e senha cadastrados anteriormente. Caso o mesmo não possua essas credenciais, é necessário fazer o cadastro na página de *signup*, informando os dados necessários para ser efetuado a solicitação de consulta posteriormente, conforme mostrado na [Figura 19](#).

Figura 19 – Tela de cadastro (*signup*)Fonte: Site *MedPoint*

Tendo em vista a possibilidade da perda ou esquecimento da senha cadastrada para fazer *login* no sistema, o *MedPoint* disponibiliza o processo de recuperação de senha, onde o usuário informa o e-mail cadastrado anteriormente, conforme mostrado na [Figura 20](#).

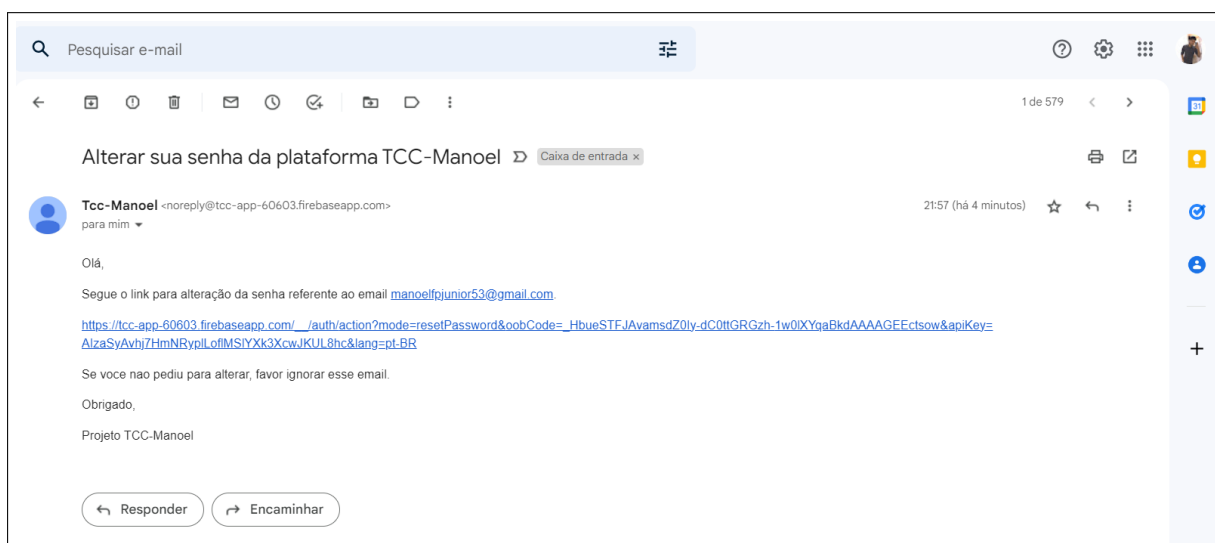
Figura 20 – Tela de recuperação de senha



Fonte: Site *MedPoint*

Em seguida é enviado um link para o e-mail informado, onde através desse link é possível fazer o cadastro de uma nova senha de acesso ao sistema, conforme a [Figura 21](#).

Figura 21 – Tela do e-mail de recuperação de senha

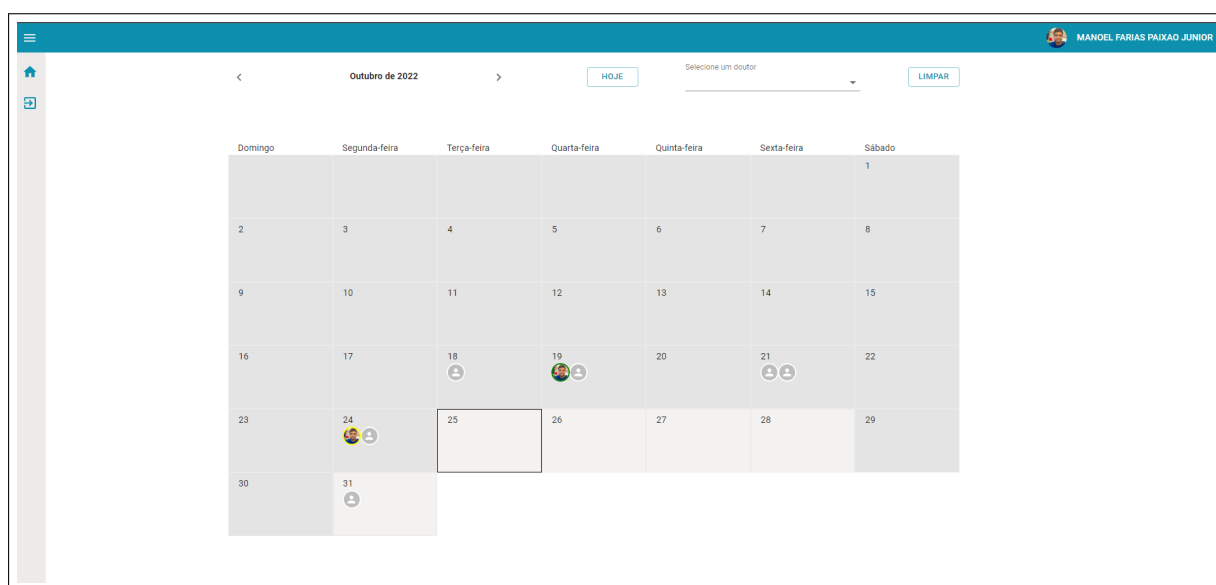


Fonte: Site *MedPoint*

## 5.2 Sistema modo usuário

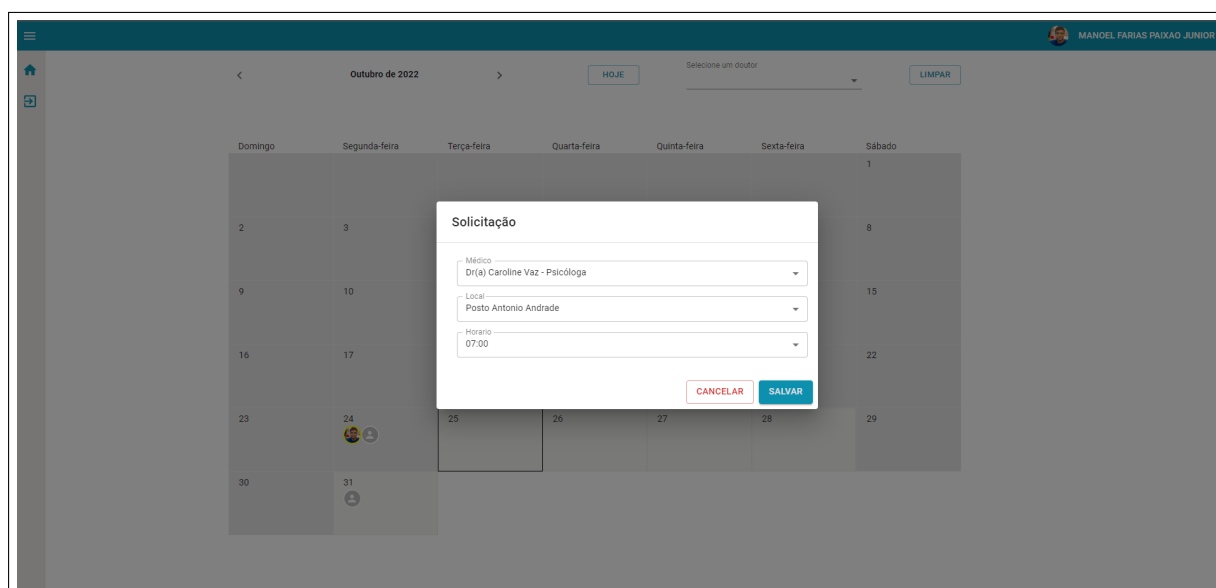
Após passar pela etapa de *login*, é possível ter acesso a parte de solicitações de consultas do sistema. A tela de solicitações vista no modo usuário é possível observar um calendário, onde será feito a escolha do dia desejado para o envio da solicitação de consulta, como mostrado na [Figura 22](#).

Figura 22 – Tela do calendário de solicitações



Fonte: Site *MedPoint*

Figura 23 – Tela do formulário de solicitação

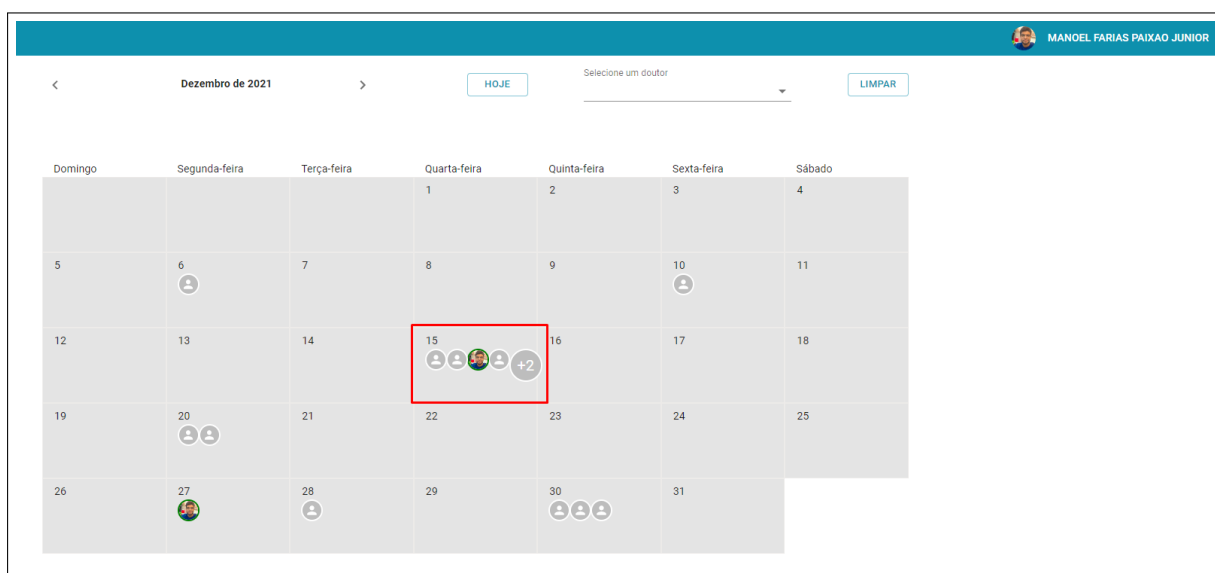


Fonte: Site *MedPoint*

Após o usuário clicar no dia desejado pra consulta, um pequeno formulário é exibido conforme mostrado na [Figura 23](#). A primeira escolha que deve ser feita pelo usuário é a do medico desejado, seguido do posto de saúde referente ao medico e o horário de atendimento do mesmo. O *MedPoint* segue o fluxo de marcação de consultas dos pequenos municípios, onde a consulta é marcada para o mesmo horário e o atendimento é feito conforme a ordem de chegada.

No calendário é possível observar, se já existem solicitações feitas por outros usuários do sistema daquele determinado dia, porem não é possível identifica-los, somente o administrador tem essa permissão. Essa informação ajuda o usuário a saber se o dia em questão já possui muitos atendimentos, conforme mostrado na [Figura 24](#).

Figura 24 – Tela do calendário de solicitações



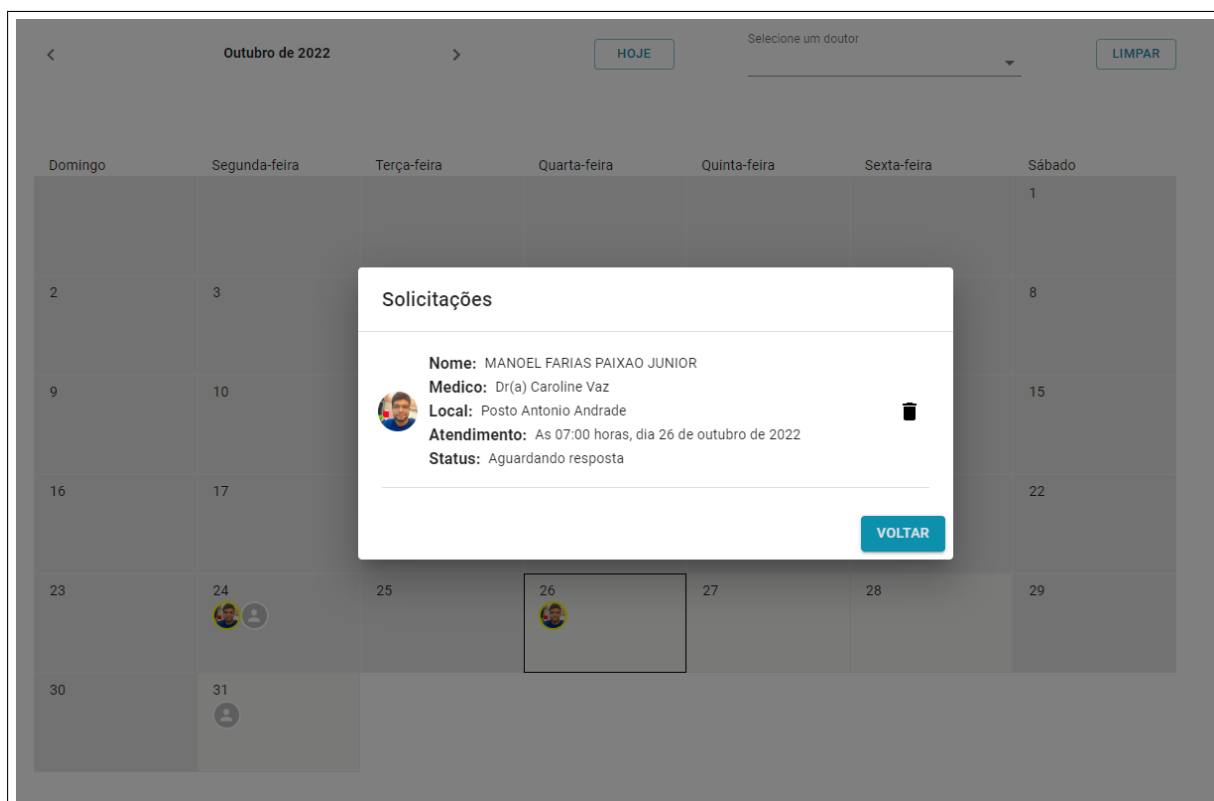
Fonte: Site *MedPoint*

Um detalhe importante, é que o usuário pode ver seu avatar no calendários nos dias referentes as solicitações de consultas já feitas por ele, podendo então usar essa informação como um pequeno histórico de registro de consultas já feitas. O usuário pode identificar se essas solicitações anteriores foram aceitas ou não pelo administrador do posto de saúde, pelas cores em volta do avatar. As solicitações com cores verdes significam que foram aceitas pelo administrador, as vermelhas mostram que as solicitações foram negadas e as amarelas significam pendencias, ou seja, o administrador não deu retorno daquela solicitação.

Caso o usuário queira mudar o dia da consulta escolhida anteriormente, é possível excluir a solicitação já enviada. Basta selecionar o dia desejado e clicar no ícone de *delete*, conforme mostrado na [Figura 25](#), logo em seguida é possível fazer uma nova solicitação naquele mesmo dia, ou em outro dia desejado.



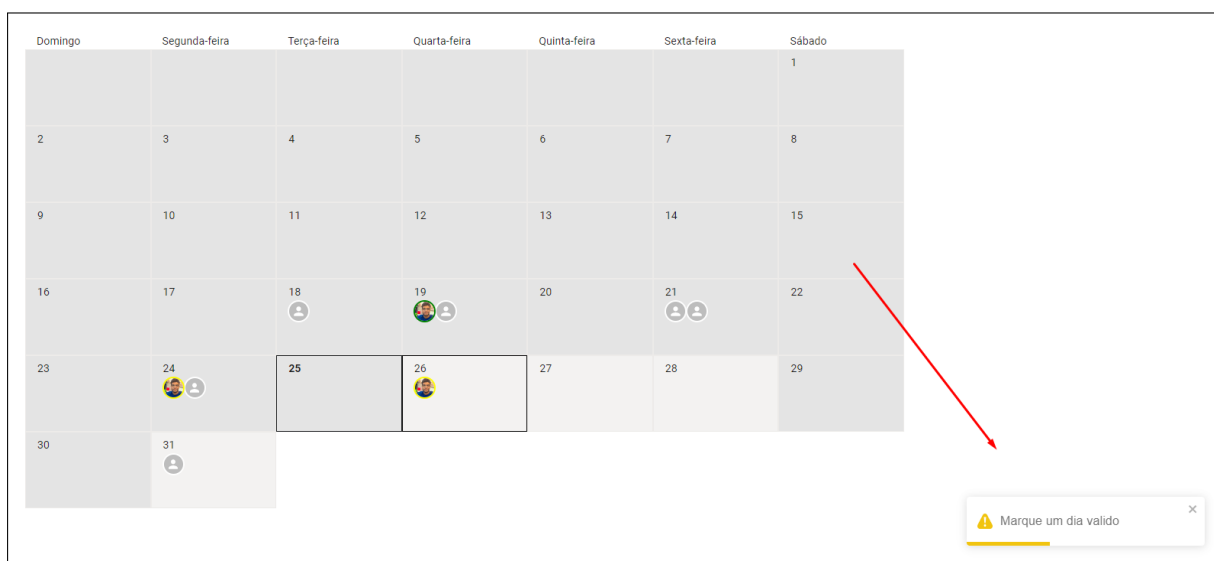
Figura 25 – Tela de exclusão de solicitação



Fonte: Site *MedPoint*

Não é permitido o envio de solicitações, feitas aos sábados, domingos e dias anteriores ao dia atual. Caso o usuário tente fazer uma solicitação nos dias inválidos, o *MedPoint* irá exibir uma mensagem informando o erro, conforme mostrado na [Figura 26](#).

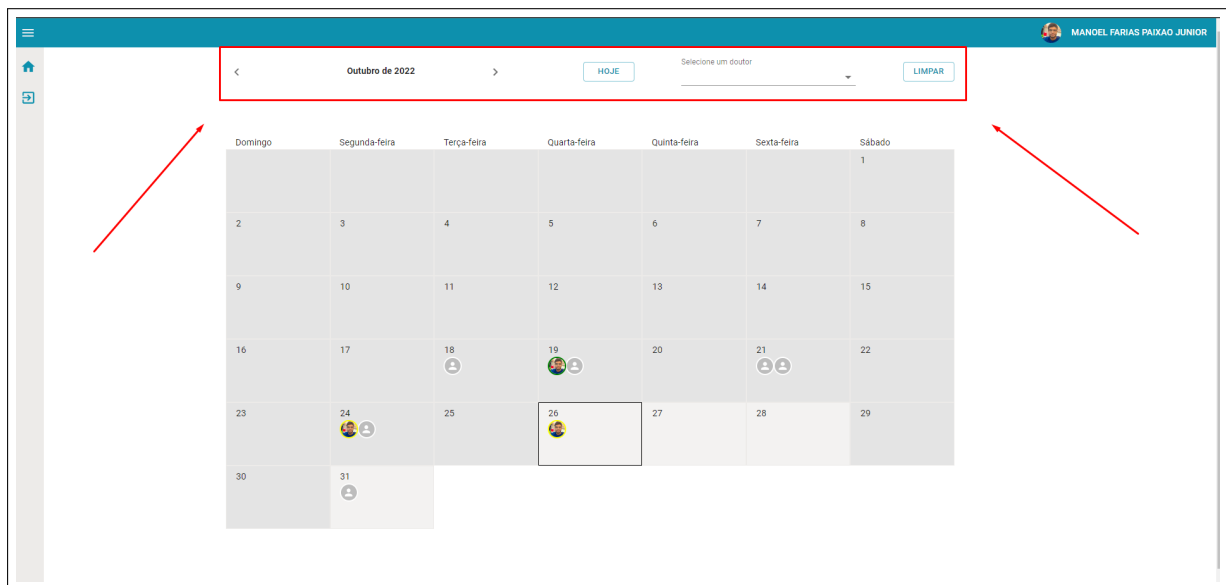
Figura 26 – Tela do calendário de solicitações



Fonte: Site *MedPoint*

Note que na [Figura 22](#) é possível observar uma barra de funcionalidades logo a cima do calendário de solicitações, sendo composta por : um navegador de datas onde é possível ter acesso aos meses anteriores ou aos próximos meses do calendário, e ao seu lado um filtro de médicos, seguido de um botão que limpa os dados selecionados no filtro, como evidenciado na [Figura 27](#).

Figura 27 – Tela do calendário de solicitações



Fonte: Site *MedPoint*

Com o intuito de ajudar o usuário a encontrar o melhor dia para enviar uma solicitação de um medico específico, basta o usuário selecione algum medico no filtro, automaticamente o sistema informará os dias de atendimento daquele medico selecionado, colorindo o calendário de verde nos dias de atendimento do medico, como mostrado na [Figura 28](#).

Figura 28 – Tela do calendário de solicitações

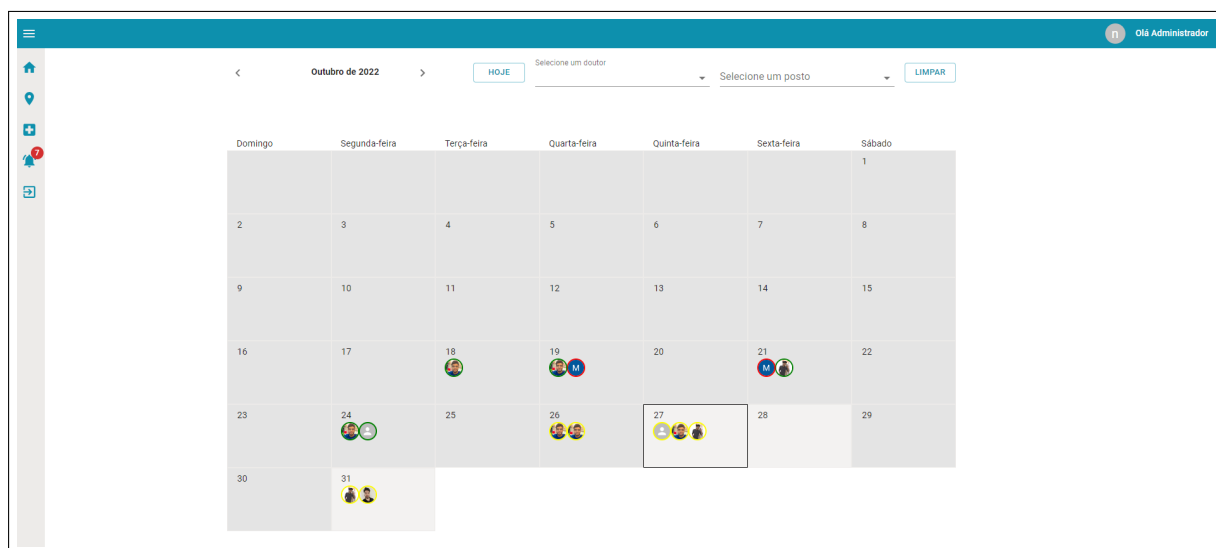
Domingo	Segunda-feira	Terça-feira	Quarta-feira	Quinta-feira	Sexta-feira	Sábado
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Fonte: Site *MedPoint*

### 5.3 Sistema modo administrador

Como mencionado anteriormente o *MedPoint* possui dois módulos distintos, um de usuário e outro de administrador. O sistema faz essa interpretação através de um campo *boolean* dentro do banco de dados, onde informa se o e-mail que fez *login* no sistema é usuário ou administrador. Assim que o usuário enviar a solicitação de consulta, as informações contidas na solicitação irão chegar no sistema, e serão exibidas no calendário com a visão de administrador, conforme mostrado na [Figura 29](#).

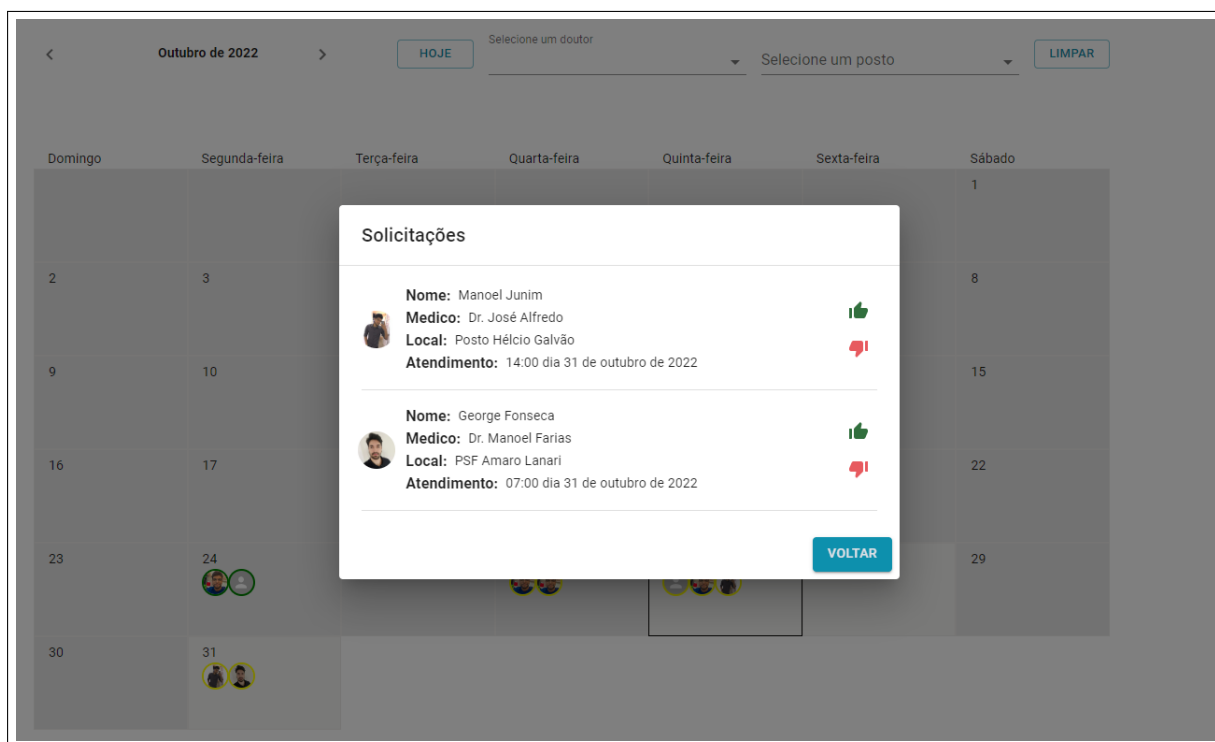
Figura 29 – Calendário modo administrador



Fonte: Site *MedPoint*

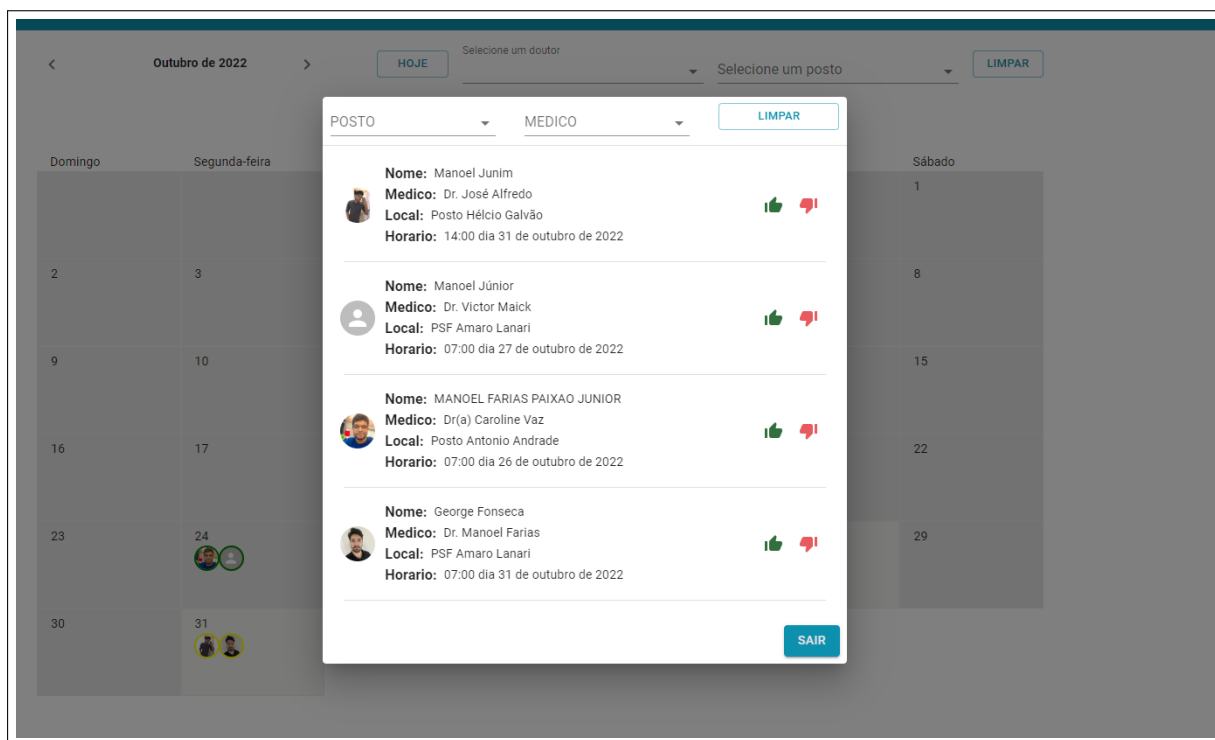
Diferente do calendário na visão de usuário, no modo administrador é possível ver todas as solicitações presentes no calendário sem nenhuma restrição. As solicitações podem ser visualizadas pelo administrador de duas maneiras diferentes clicando no dia desejado e visualizando as solicitações daquele dia, conforme a [Figura 30](#), ou clicando na barra lateral no ícone em formato de sino, onde será exibido todas as solicitações que ainda não foram respondidas, conforme a [Figura 31](#).

Figura 30 – Modal de solicitações do calendário



Fonte: Site *MedPoint*

Figura 31 – Modal de solicitações da barra lateral



Fonte: Site *MedPoint*

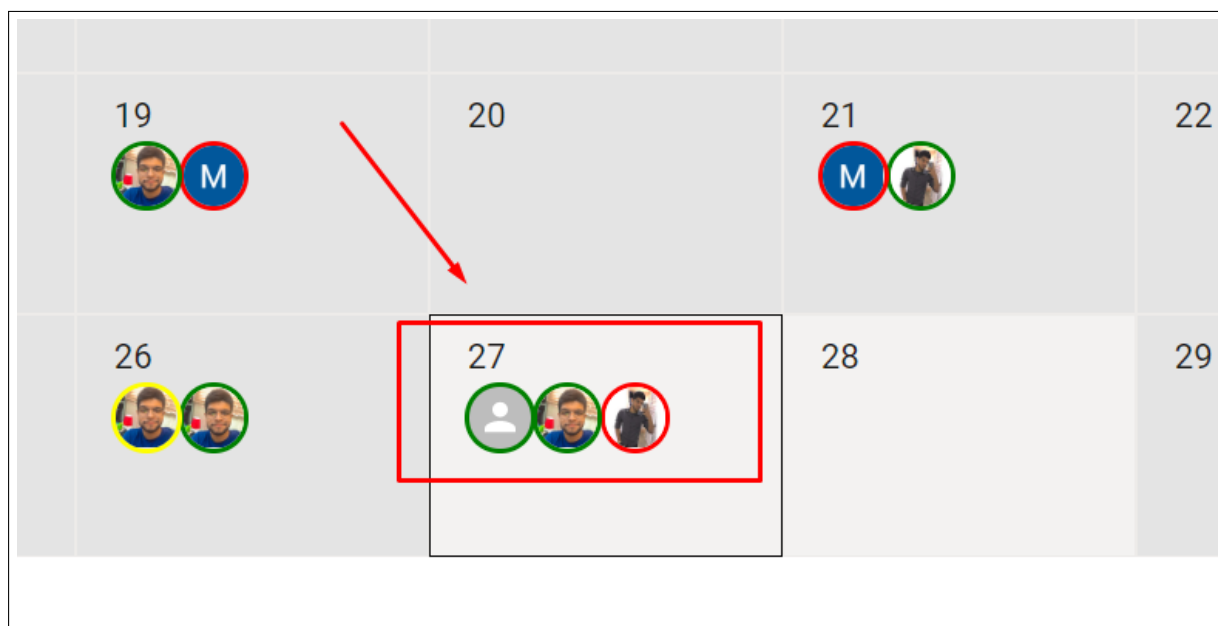
No modal de exibição de solicitações, é possível observar todas as informações necessárias para a confirmação da consulta. Caso não haja imprevistos, e o médico em questão confirmou a sua presença no dia da consulta, o administrador pode então, aceitar a solicitação enviada, clicando no ícone de positivo do lado direito do modal de solicitações, caso contrário o mesmo pode negar a solicitação clicando no ícone de negativo, conforme a [Figura 32](#). Seja qual a resposta, essa informação será enviada para o usuário, através de uma ligação do próprio administrador, como também o calendário de exibição do usuário irá mudar o status da sua solicitação, alterando para a cor determinada pela resposta. Caso positiva será verde, caso negativa será vermelha, conforme mostrado na [Figura 33](#).

Figura 32 – Solicitação de consulta



Fonte: Site *MedPoint*

Figura 33 – Cores dos avatares



Fonte: Site *MedPoint*

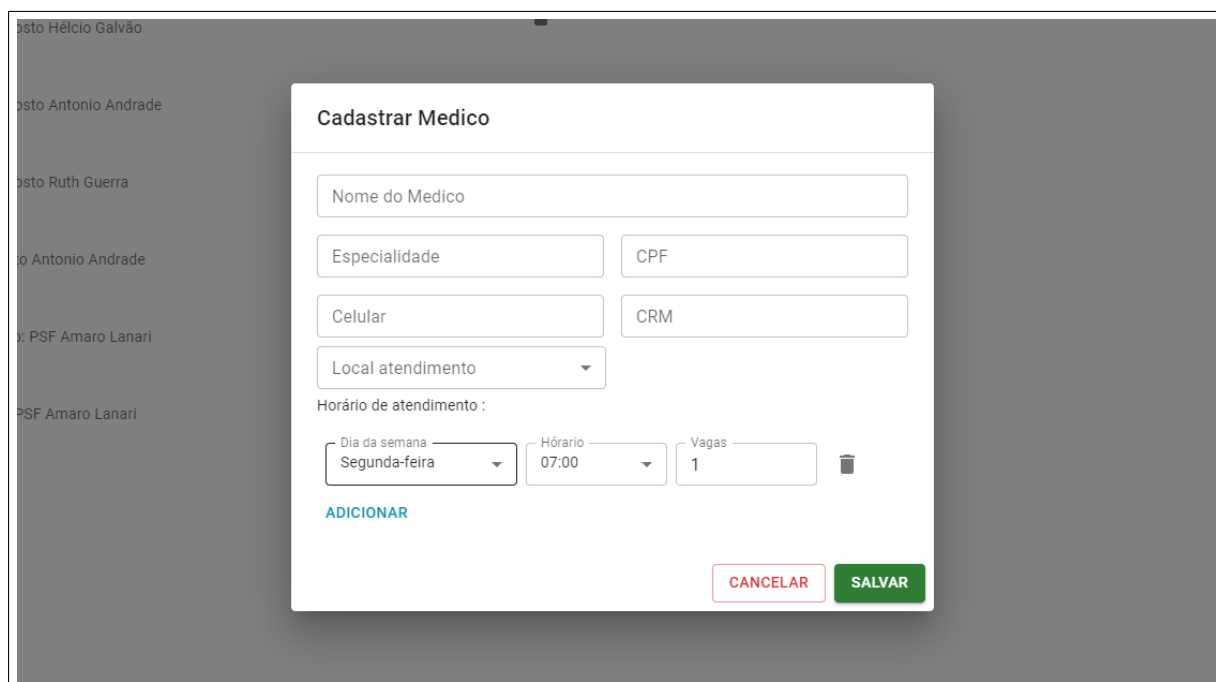
No modo administrador é possível ter acesso a todos os médicos cadastrados, tendo a permissão para fazer novos cadastros, alterar e excluir algum registro caso necessário, assim como mostrado na [Figura 34](#) e [Figura 35](#).

Figura 34 – Tela de médicos



Fonte: Site *MedPoint*

Figura 35 – Tela de adicionar médico



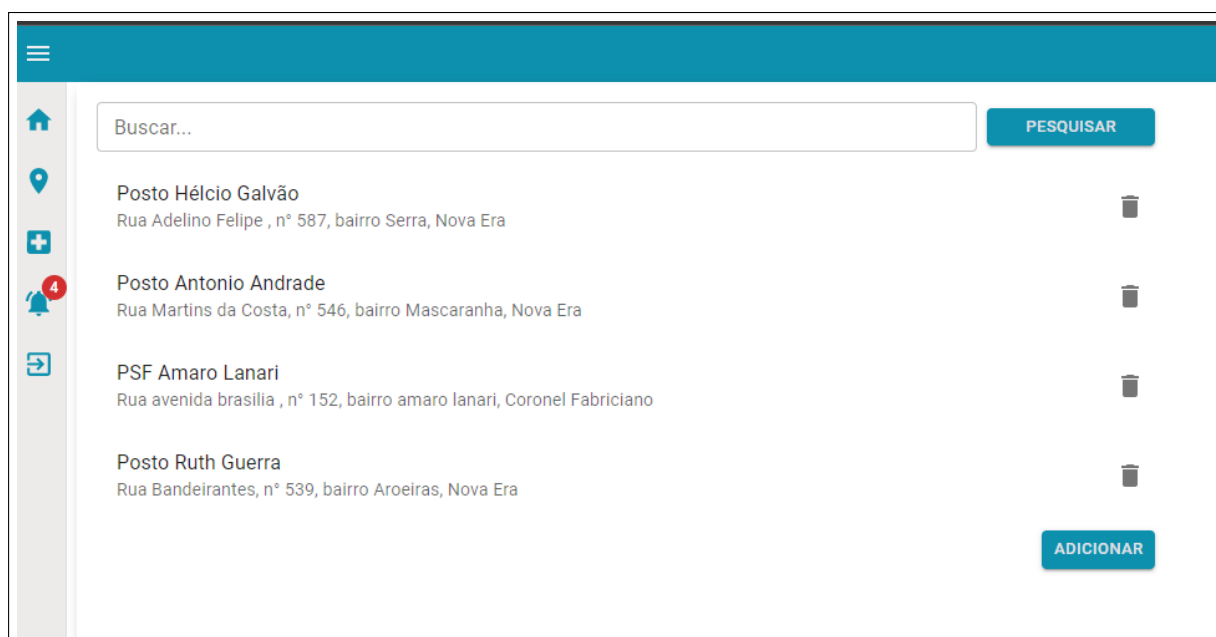
The screenshot shows a mobile application interface for adding a doctor. A modal window titled "Cadastrar Medico" is centered on a grey background. The form contains the following fields and controls:

- Nome do Medico: Text input field.
- Especialidade: Text input field.
- CPF: Text input field.
- Celular: Text input field.
- CRM: Text input field.
- Local atendimento: Dropdown menu.
- Horário de atendimento: Section header.
- Dia da semana: Dropdown menu (selected: Segunda-feira).
- Hórorio: Dropdown menu (selected: 07:00).
- Vagas: Text input field (value: 1) with a trash icon to its right.
- ADICIONAR: Blue button.
- CANCELAR: Red button.
- SALVAR: Green button.

Fonte: Site *MedPoint*

O mesmo procedimento pode ser feito com os postos, o administrador tem permissão de visualiza todos os registros, adicionar novos postos, alterar as informações e excluir caso necessário, assim como mostrado na [Figura 36](#) e [Figura 37](#).

Figura 36 – Tela de postos de saúde



The screenshot shows a mobile application interface for health posts. It features a teal header with a menu icon. Below the header is a search bar with the text "Buscar..." and a "PESQUISAR" button. A list of health posts is displayed, each with a location name and address, and a trash icon for deletion. The posts listed are:

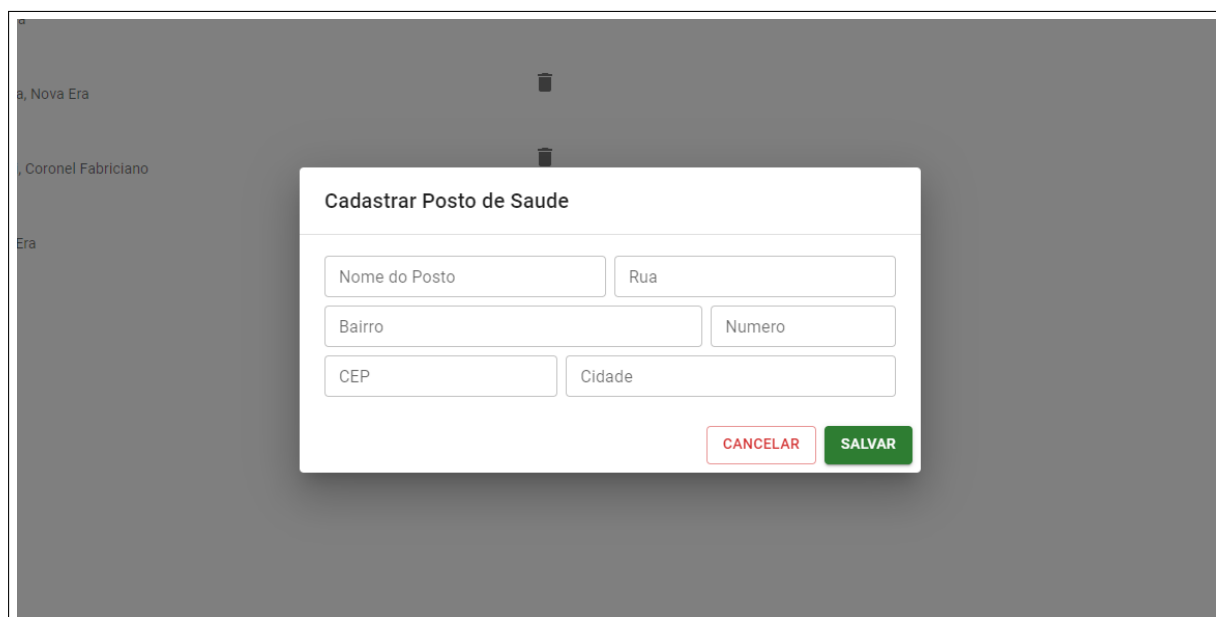
- Posto Hélcio Galvão  
Rua Adelino Felipe , n° 587, bairro Serra, Nova Era
- Posto Antonio Andrade  
Rua Martins da Costa, n° 546, bairro Mascaranha, Nova Era
- PSF Amaro Lanari  
Rua avenida brasilia , n° 152, bairro amaro lanari, Coronel Fabriciano
- Posto Ruth Guerra  
Rua Bandeirantes, n° 539, bairro Aroeiras, Nova Era

At the bottom right, there is an "ADICIONAR" button. A sidebar on the left contains icons for home, location, medical services, notifications (with a red '4' badge), and a share icon.

Fonte: Site *MedPoint*



Figura 37 – Tela de adicionar posto



Nome do Posto

Rua

Bairro

Numero

CEP

Cidade

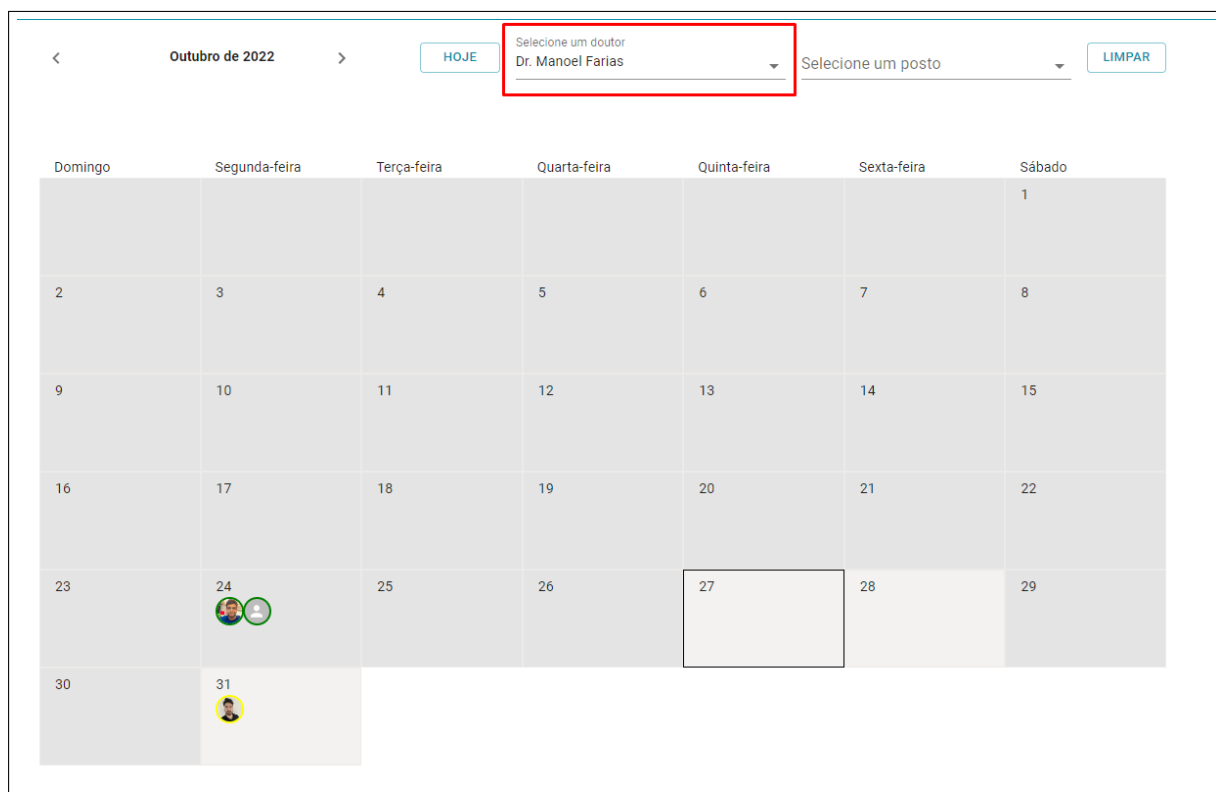
CANCELAR

SALVAR

Fonte: Site *MedPoint*

O sistema no modo administrador assim como no modo usuário, também possui acesso aos filtros na barra acima do calendário, porém com finalidades diferentes. O filtro no modo administrador tem como objetivo exibir apenas as solicitações daquele determinado medico e posto selecionado na pesquisa, facilitante assim a busca por solicitações específicas, assim como mostrada na [Figura 38](#).

Figura 38 – Tela do calendário filtrado



Fonte: Site *MedPoint*

por ultimo mas não menos importante, o *MedPoint* foi desenvolvido para ser responsivo em todas as telas do sistema, ou seja, os componentes se adequam para todos os tamanhos de dispositivos, sejam eles *desktop* ou *mobile*.

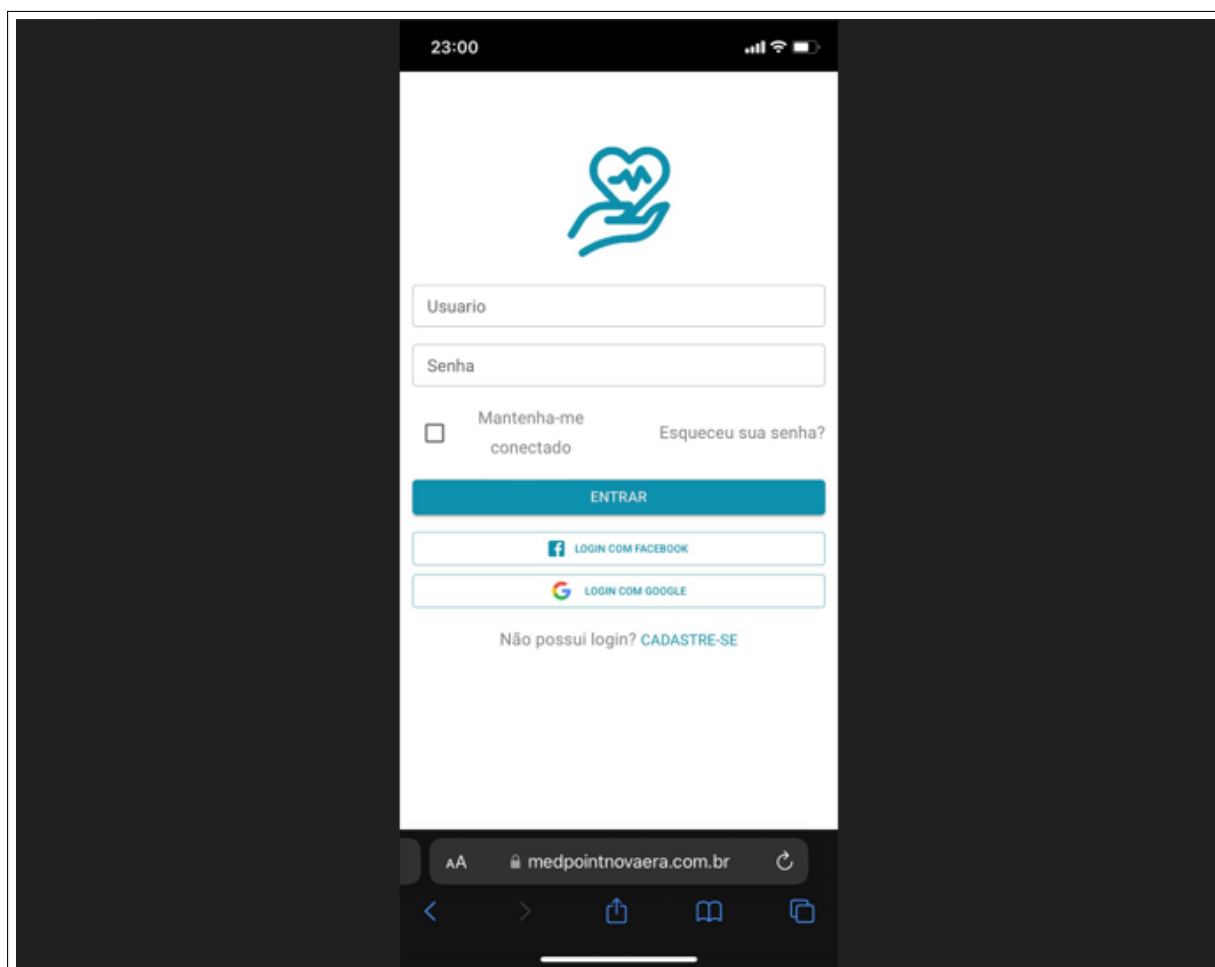
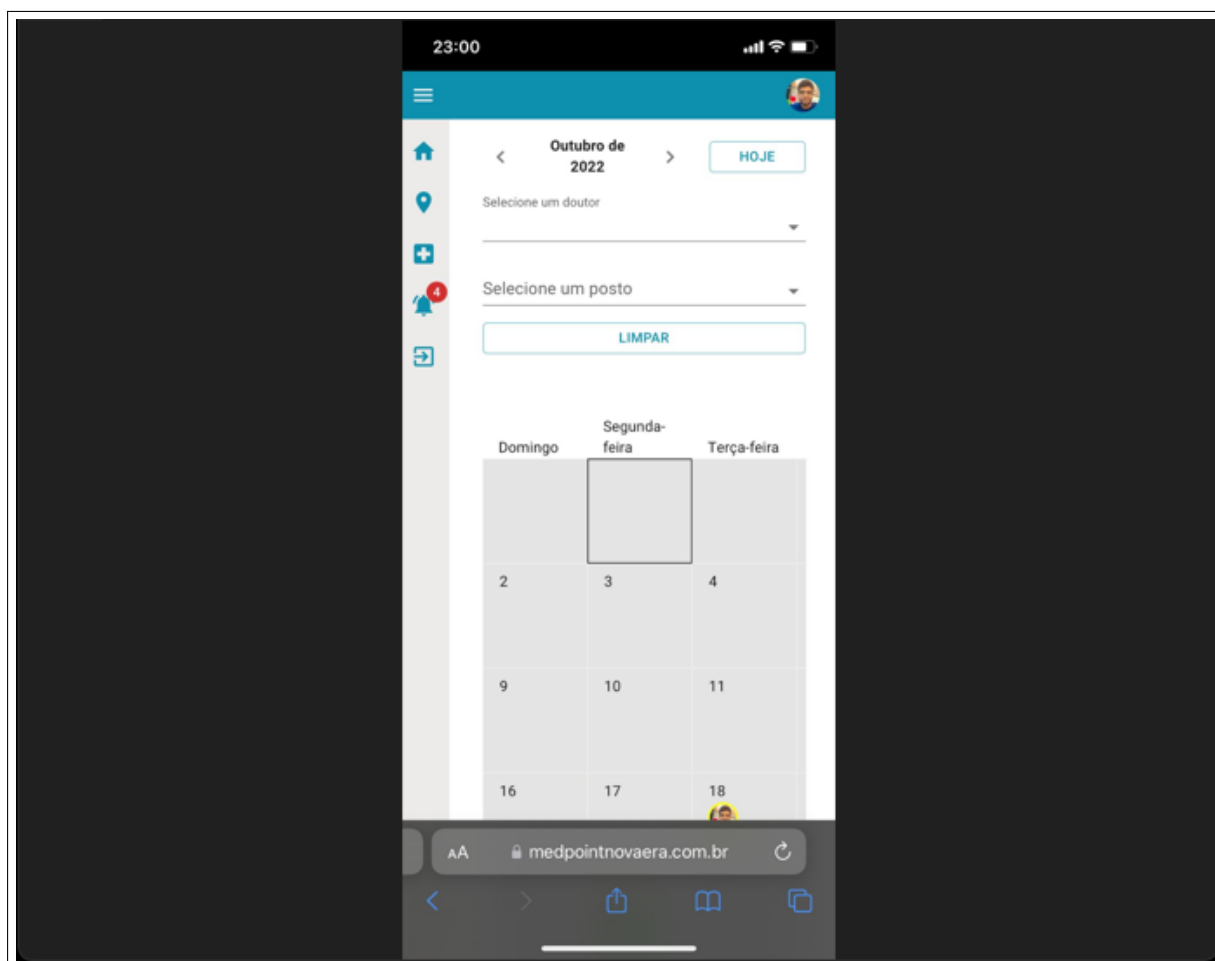
Figura 39 – Tela de login *mobile*Fonte: Site *MedPoint*

Figura 40 – Tela do calendário *mobile*

Fonte: Site *MedPoint*

## 6 Conclusão

Este trabalho abordou o desenvolvimento de uma plataforma *web* para gestão de marcações de consultas. Para a aplicação em questão utilizou-se diferentes ferramentas: o *React* que é uma biblioteca *Javascript* para a criação de **UI**. Foi utilizado também o *TypeScript* que é um superconjunto de *Javascript*, ou seja, um conjunto de ferramentas e formas mais eficientes de escrever código *Javascript*, adicionando recursos que não estão presentes de maneira nativa na linguagem. Outra ferramenta bastante utilizada no desenvolvimento do projeto foi o *Firebase*, dessa plataforma foi utilizado o banco de dados *Firestore* que é um banco do tipo *NoSQL*, outra ferramenta do *Firebase* utilizada foi o *Authentication*, com a funcionalidade de autenticar os logins de acesso ao sistema.

Certamente este trabalho contribui com os diversos trabalhos correlatos que investiga, desenvolve ou compara padrões aplicáveis no desenvolvimento de aplicações moveis, em contraste, neste foi possível aplicar padrões em uma aplicação relativamente maior que as aplicações de outros trabalhos. Não apenas, este trabalho descreveu a fundamentação teórica acerca do trabalho, o processo do levantamento de requisitos juntamente com os requisitos levantados, a modelagem do sistema, os protótipos desenvolvidos, o desenvolvimento da aplicação nos padrões arquiteturais e os testes da aplicação. Como apresentado, o desenvolvimento da aplicação tinha como objetivo dar suporte às diversas tarefas referentes ao gerenciamento das marcações nos **PSR**. Contudo, durante o desenvolvimento da aplicação, ocorreram entraves que acarretou o atraso na disponibilização da plataforma e consequentemente, o mesmo não pode ser testado da forma como pretendida. No entanto, foi possível durante o processo de desenvolvimento da aplicação identificar diferentes necessidades do **PSR**, além da aquisição de um grande aprendizado documentado neste trabalho, no que se diz respeito ao modo de funcionamento dos **PSR**.

A implantação do sistema desenvolvido resultará em uma escolha assertiva, haja vista a necessidade de suporte em diversas atividades de gerenciamento. Com a implantação do sistema, é presumível a diminuição nas filas e maior celeridade para se conseguir vagas, o estreitamento do relacionamento com os clientes e, por fim, uma boa gestão do conhecimento, uma vez que tais mudanças institucionais certamente irão ocorrer futuramente. Dito isto, é possível encontrar todo o projeto desenvolvido neste trabalho no [www.medpointnovaera.com.br](http://www.medpointnovaera.com.br).

# Referências

- BUSCHMANN; MEUNIER. *R. Pattern-oriented Software Architecture—A System of Patterns*. edição (16 agosto 1996). [S.l.]: Wiley, 1996. Volume 1. Citado na página 16.
- FIREBASE. *FIREBASE*. 2022. Disponível em: <<https://firebase.google.com/?hl=pt>>. Citado na página 28.
- FIREBASE. *FIREBASE*. 2022. Disponível em: <<https://firebase.google.com/docs/firestore>>. Citado na página 32.
- GITHUB. *GITHUB*. 2022. Disponível em: <<https://github.com/>>. Citado na página 29.
- LAUDON. *J. P. Sistemas de informação gerenciais*. Edição 17. [S.l.]: Bookman, 2011. Citado na página 16.
- MAINKAR. *P. Expert Android Programming: Master skills to build enterprise grade Android applications*. 1º edição. ed. [S.l.]: Packt Publishing, 2017. Citado na página 19.
- MATERIAL-UI. *MATERIAL-UI*. 2022. Disponível em: <<https://mui.com/pt/material-ui/getting-started/overview>>. Citado na página 27.
- PES. *PES*. 2022. Disponível em: <<https://www.pes.com.br/doctors-office-10>>. Citado na página 20.
- PORTALNOVAERA. *Decretos*. 2014. Disponível em: <[https://www.novaera.mg.gov.br/abrir\\_arquivo.aspx/Decretos\\_1843\\_2014?cdLocal=5&arquivo={B5CA51A0-CCDD-BECB-D115-812AEADDA843}.pd](https://www.novaera.mg.gov.br/abrir_arquivo.aspx/Decretos_1843_2014?cdLocal=5&arquivo={B5CA51A0-CCDD-BECB-D115-812AEADDA843}.pd)>. Citado na página 13.
- PRESSMAN. *R. S. Engenharia de Software*. Edição 7. [S.l.]: Amgh Editora, 2019. Citado 2 vezes nas páginas 18 e 19.
- REACTJS. *REACTJS*. 2022. Disponível em: <<https://pt-br.reactjs.org>>. Citado na página 27.
- SOMMERVILLE. *Engenharia de Software*. 9º edição. ed. [S.l.]: Pearson Universidades, 2011. Citado 3 vezes nas páginas 17, 18 e 19.
- THAYER; BAILIN; DORFMAN. *M. Software requirements engineerings*. 2º edição. ed. [S.l.]: Wiley-IEEE Computer Society PR, 1997. Citado na página 18.
- TYPESCRIPT. *TYPESCRIPT*. 2022. Disponível em: <<https://www.typescriptlang.org>>. Citado na página 27.
- VERCEL. *VERCEL*. 2022. Disponível em: <<https://vercel.com/docs>>. Citado na página 29.
- VSCODE. *VSCODE*. 2022. Disponível em: <<https://code.visualstudio.com/docs>>. Citado na página 30.

---

WEENGRANHARIA. *WEENGRANHARIA*. 2022. Disponível em: <<https://weengrnharia.com>>. Citado na página 23.