
Curso de Ciência da Computação
Universidade Estadual de Mato Grosso do Sul

**APLICATIVO RESPONSIVO PARA ACOMPANHAMENTO DE PACIENTES DA
ESTRATÉGIA SAÚDE DA FAMÍLIA**

João Vitor de Oliveira

Dr^a. Glauca Gabriel Sass (Orientadora)

Dourados - MS
2022

APLICATIVO RESPONSIVO PARA ACOMPANHAMENTO DE PACIENTES DA
ESTRATÉGIA SAÚDE DA FAMÍLIA

João Vitor de Oliveira

Este exemplar corresponde à redação final da monografia da disciplina Projeto Final de Curso, devidamente corrigida e defendida por João Vitor de Oliveira e aprovada pela Banca Examinadora, como parte dos requisitos para a obtenção do título de Bacharel em Ciência da Computação.

Dourados, 08 de novembro de 2022.

Profa. Dra. Glaucia Gabriel Sass (Orientadora)

047a Oliveira, João Vitor de

Aplicativo responsivo para acompanhamento de pacientes da estratégia saúde da família / João Vitor de Oliveira. – Dourados, MS: UEMS, 2022.

71 p.

Monografia (Graduação) – Ciência da Computação – Universidade Estadual de Mato Grosso do Sul, 2022.

Orientadora: Prof. Dr^a Glaucia Gabriel Sass

1. Sistema de Informação 2. Aplicativo web responsivo 3. Estratégia Saúde da Família (ESF) I. Sass, Glaucia Gabriel II. Título

CDD 23.ed. - 005

**APLICATIVO RESPONSIVO PARA ACOMPANHAMENTO DE
PACIENTES DA ESTRATÉGIA SAÚDE DA FAMÍLIA**

João Vitor de Oliveira

Novembro de 2022

Banca Examinadora:

Profª. Dra. Glaucia Gabriel Sass (Orientadora)
Área de Computação - UEMS

Prof. Dr. Diogo Fernando Trevisan
Área de Computação - UEMS

Profª. Dra. Mercedes Rocío Gonzales Márquez
Área de Computação - UEMS

Dedico este trabalho à minha mãe Arlinda dos Santos Oliveira, ao meu pai Luciano Gomes de Oliveira, ao meu irmão Luiz Henrique de Oliveira e à minha namorada Aline Niz Tomaz que estiveram do meu lado me apoiando em cada passo que percorri até aqui. Além disso, este trabalho é dedicado a toda a minha família e amigos.

AGRADECIMENTOS

Primeiramente gostaria de agradecer a todos que estiveram me apoiando durante todo o decorrer deste estudo. Meus sinceros agradecimentos à minha orientadora e professora, Glaucia Gabriel Sass, pela oportunidade, por estar presente e por me acompanhar durante todo o desenvolvimento deste trabalho; agradeço também por comemorar comigo cada descoberta e progresso do trabalho.

A todos os meus amigos que estavam torcendo por mim, que ficaram encantados quando mostrei o trabalho pronto. Aos meus professores, que estiveram me proporcionando conhecimentos que não imaginava ter a uns anos atrás, enquanto sonhava em cursar Ciência da Computação. Gostaria de agradecer em especial a professora Adriana Betânia de Paula Molgora por me orientar em projeto de iniciação científica e por me dar a oportunidade de passar meus conhecimentos para outros alunos através de monitoria; ao professor André Chastel Lima, por me orientar em um projeto de extensão, onde eu pude passar meus conhecimentos em informática para pessoas da terceira idade.

A minha família, que não há palavras para descrever o quão são importantes para mim, são a base que eu preciso para caminhar e conquistar meus sonhos. A minha mãe Arlinda dos Santos Oliveira e ao meu pai Luciano Gomes de Oliveira que fazem ligações todos os dias para me apoiar e me dar forças mesmo estando longe, ao meu irmão Luiz Henrique de Oliveira e a minha namorada Aline Niz Tomaz a quem eu sou grato por me acompanhar em tudo o que faço e por ouvir meus desabafos quando eu mais preciso.

Agradeço a Deus por me dar forças para não desistir no meio do caminho, pois só Ele e eu sabemos o quanto foi difícil essa trajetória.

Muito obrigado a todos!

RESUMO

O uso da tecnologia na área da saúde tem ganhado cada vez mais espaço, a fim de auxiliar nos processos de atendimento, acompanhamento, diagnósticos, prescrição de medicamentos, entre outros meios de ajudar no cuidado à saúde das pessoas. Sistemas de informações voltados para a área da saúde têm como objetivo promover um serviço à saúde com mais eficiência e efetividade. Com base na necessidade de facilitar esse tipo de serviço, o presente trabalho apresenta a construção de um aplicativo responsivo com o intuito de facilitar os profissionais de saúde no acompanhamento de pacientes da Estratégia Saúde da Família, visto que os métodos de acompanhamento utilizados demandam esforço e tempo gasto no preenchimento de informações referentes aos pacientes. Deste modo, o sistema construído irá armazenar informações de pacientes como por exemplo, as condições de saúde; a microárea e localização dos mesmos em um mapa digital; além de possibilitar a geração de relatórios.

Palavras-chave: Sistema de Informação em Saúde, Estratégia Saúde da Família, Aplicativo *web* responsivo.

SUMÁRIO

1. INTRODUÇÃO	17
1.1. Justificativa	18
2. REFERENCIAL TEÓRICO	21
2.1. Estratégia Saúde da Família	21
2.2. Sistemas de Informação em Saúde	22
3. METODOLOGIA	25
4. DESENVOLVIMENTO	29
4.1. Levantamento de requisitos	29
4.1.1. Requisitos funcionais	29
4.1.2. Requisitos não funcionais	31
4.2. Análise de requisitos	31
4.2.1. Modelo de casos de uso	31
4.2.1.1. Diagrama de casos de uso	32
4.2.1.2. Descrição dos casos de uso	33
4.3. Implementação do aplicativo	40
5. CONCLUSÃO	49
REFERÊNCIAS BIBLIOGRÁFICAS	51
APÊNDICE A - PÁGINAS DA APLICAÇÃO FRONT-END	53
APÊNDICE B - COMO EXECUTAR A APLICAÇÃO	67
APÊNDICE C - CRIANDO UM USUÁRIO ADMINISTRADOR	69

LISTA DE SIGLAS

ACS	Agente Comunitário de Saúde
APS	Atenção Primária à Saúde
CPF	Cadastro de Pessoa Física
CSS	<i>Cascading Style Sheets</i>
ESF	Estratégia Saúde da Família
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
JSON	<i>JavaScript Object Notation</i>
JWT	<i>JSON Web Token</i>
PEC	Prontuário Eletrônico do Cidadão
PNIS	Política Nacional de Informação e Informática
SIS	Sistema de Informação em Saúde
e-SUS	Sistema Único de Saúde eletrônico
SUS	Sistema Único de Saúde
UBS	Unidade Básica de Saúde
UML	<i>Unified Modeling Language</i>

LISTA DE FIGURAS

Figura 1 - Mapa da ESF produzido pela enfermeira.....	19
Figura 2 - Legenda do mapa da ESF.....	19
Figura 3 - Alfinetes no mapa da ESF.....	20
Figura 4 - Diagrama de casos de uso para o aplicativo ESF.....	32
Figura 5 - Esquema de funcionamento da aplicação.....	40
Figura 6 - Código do acesso ao sistema do lado do back-end.....	42
Figura 7 - Formulário de acesso ao sistema.....	42
Figura 8 - Página inicial do sistema (Painel) na visão do Administrador ESF.....	43
Figura 9 - Página inicial do sistema (Painel) na visão do Agente de saúde.....	44
Figura 10 - Formulário de busca de pacientes sem os resultados de busca.....	45
Figura 11 - Busca de pacientes depois de realizar uma busca.....	46
Figura 12 - Exemplo de PDF Gerado.....	46
Figura 13 - Página de acesso ao sistema	53
Figura 14 - Página de criação de um novo agente de saúde.....	54
Figura 15 - Página de listagem de agentes de saúde.....	54
Figura 16 - Página de edição de um agente de saúde.....	55
Figura 17 - Página de criação de uma nova condição de saúde.....	55
Figura 18 - Página de listagem de condições de saúde.....	56
Figura 19 - Página de edição de uma condição de saúde.....	56
Figura 20 - Página de criação de microáreas.....	57
Figura 21 - Página de listagem de microáreas.....	57
Figura 22 - Página de edição de uma microárea.....	58
Figura 23 - Página de busca de pacientes com a geração de relatório PDF.....	58
Figura 24 - Página inicial (Painel) do Administrador ESF.....	59
Figura 25 - Página inicial (Painel) do Agente de Saúde.....	59
Figura 26 - Página de adicionar um novo paciente.....	60
Figura 27 - Página de listagem de pacientes.....	61
Figura 28 - Página de editar informações de um paciente.....	62
Figura 29 - Página de acesso ao sistema na visualização móvel.....	63
Figura 30 - Página inicial do aplicativo na visualização móvel.....	64
Figura 31 - Menu de navegação na versão móvel.....	65

Figura 32 - Código de criação de usuário Administrador sem dados.....	69
Figura 33 - Código de criação de usuário Administrador com dados.....	70

1. INTRODUÇÃO

Na sociedade atual em que vivemos, um mundo onde a tecnologia está em constante evolução, há o desejo, e muitas vezes, a necessidade de se usar tecnologias para facilitar e ganhar tempo em diversas áreas. A área da saúde segue utilizando essas tecnologias constantemente. Levando em conta essa necessidade, este estudo mostra o desenvolvimento de um aplicativo responsivo para ajudar e facilitar o acompanhamento de pacientes da Estratégia Saúde da Família (ESF) da Unidade Básica de Saúde (UBS) Altos do Indaiá, localizada no município de Dourados - MS.

No presente momento, os registros de atendimentos a pacientes da ESF são realizados utilizando métodos que demandam esforço, sobrecarga de trabalho e tempo gasto no preenchimento de informações referentes ao paciente. Todo o controle e organização dos dados são realizados utilizando-se planilhas e aplicativos do governo federal, o Sistema Único de Saúde eletrônico (e-SUS) direcionado para Atenção Primária à Saúde (APS) na modalidade de Prontuário Eletrônico do Cidadão (PEC), sendo que o mesmo não se assenta com a realidade prática no que diz respeito a assistência fornecida aos pacientes; e fichas de acompanhamento impressas. Além disso, os mapas que contém os pacientes e as áreas geográficas das microáreas que compõem a região da ESF são feitos de forma manual pela enfermeira responsável pela ESF. Logo, não tendo tudo armazenado e acessível em um só lugar, isto encaminha a ter um aplicativo que aborda essas informações de pacientes e que possa favorecer o acompanhamento dos mesmos.

Tendo como base essa situação apresentada acima, o objetivo geral deste estudo foi construir um aplicativo *web* responsivo para ajudar no acompanhamento de pacientes da ESF. Especificamente, fez-se o estudo das linguagens de programação e ferramentas de construção de sistemas *web*; realizou-se a coleta de dados a campo de início e depois identificou-se os casos de uso; além de desenvolver as sucessivas fases da criação do aplicativo.

A ideia por trás do aplicativo *web* ser responsivo é que o mesmo se adequa ao dispositivo e às resoluções de tela que estão executando. Isso significa que um usuário pode acessar o sistema utilizando, por exemplo, um computador *desktop* ou *notebook* ou até mesmo um celular, sem perder a qualidade e informações. Isso é necessário para que os agentes de saúde possam cadastrar os dados dos pacientes através de um dispositivo móvel.

Este estudo tem como hipótese primária, um *software* que auxilia no acompanhamento de pacientes, caso concebido, facilitará o processo de acompanhamento da população de uma determinada região. Como hipóteses secundárias do estudo, tem-se que o uso de determinadas ferramentas e linguagens de programação *web* favorece a construção do aplicativo de maneira mais rápida e padronizada. Assim como ter os tipos de dados e os casos de uso definidos simplifica o entendimento do que deve ser feito durante o desenvolvimento do projeto, facilitando o passo a passo do mesmo.

1.1. Justificativa

Considerando que atualmente o acompanhamento de pacientes da ESF funciona seguindo os seguintes passos: os agentes de saúde vão até as residências para fazer o acompanhamento dos pacientes e este é realizado em fichas impressas; depois os agentes de saúde passam os dados dos pacientes que estão nas fichas para uma planilha de Excel e para o aplicativo do governo federal, na modalidade PEC. Esta abordagem atual dificulta o acompanhamento de pacientes e também na elaboração de relatórios sobre os dados existentes.

Os relatórios são importantes também para fazer a remuneração dos agentes de saúde, uma vez que a remuneração é baseada na produtividade. Além disso, o mapa que contém as microáreas que compõem a região geográfica da ESF é feito de forma manual pela enfermeira responsável. No mapa produzido pela enfermeira é possível visualizar também alfinetes que representam os pacientes com suas condições de saúde da ESF. O mapa que é construído pela enfermeira pode ser visualizado na Figura 1.

Figura 1 - Mapa da ESF produzido pela enfermeira



Fonte: O autor.

A Figura 2 mostra de forma ampliada a legenda que representa o mapa da ESF, onde possui as áreas territoriais, as Agentes Comunitárias de Saúde (ACS), as microáreas e as condições de saúde.

Figura 2 - Legenda do mapa da ESF



Fonte: O autor.

Figura 3 - Alfinetes no mapa da ESF

Fonte: O autor.

Na Figura 3 é apresentado os alfinetes que representam os pacientes com suas condições de saúde, isto é, de acordo com a legenda. Por exemplo, o alfinete vermelho representa uma paciente gestante.

Por estas razões que o presente estudo é importante para com os envolvidos na ESF, o sistema além de armazenar os dados dos pacientes e das microáreas participantes, irá fornecer a possibilidade ao administrador geral de criar relatórios dos dados. O aplicativo ainda manterá o mapa com os pacientes e a região geográfica da ESF. Dessa forma, com o aplicativo será possível ter um controle moderno sobre os agentes de saúde e pacientes da ESF.

2. REFERENCIAL TEÓRICO

2.1. Estratégia Saúde da Família

A ESF surgiu como um modelo que propõe uma forma de reorganização da atenção básica no Brasil seguindo os princípios do Sistema Único de Saúde (SUS). Dentro deste modelo, é definido um conjunto de ações direcionadas à saúde tanto no campo individual quanto no coletivo, visando a promoção e proteção à saúde, produção de diagnósticos, tratamento e o cuidado da saúde. Essas ações são promovidas em áreas geográficas onde pode-se definir um diagnóstico da situação de saúde da população que reside neste território e, é realizada por equipes multidisciplinares, sendo a primeira fonte de acesso desta população ao sistema de saúde (MACINKO; MENDONÇA, 2018).

Em uma ESF, como já mencionado, atuam equipes multidisciplinares realizando ações em prol da saúde da população. Conforme Brasil (2012), essas equipes são compostas no mínimo por um médico generalista, ou médico de família, um enfermeiro, um auxiliar de enfermagem e pelas ACS. Além disso, um cirurgião-dentista e um auxiliar técnico em saúde bucal podem ajudar a compor a equipe. Cada ESF deve dar suporte a no máximo 4000 (quatro mil) pessoas.

Considerando as atividades desempenhadas pela equipe ESF, a ação de visita domiciliar realizada pelos ACS se destaca entre as demais por “[...] permitir aos enfermeiros e ACS conhecerem o contexto social e identificarem as necessidades de saúde das famílias assistidas pela equipe, permitindo uma maior aproximação com os determinantes do processo saúde-doença” (KEBIAN; ACIOLI, 2014).

A visita domiciliar dos ACS às famílias, ocorre em uma determinada base geográfica, conhecida como microárea. A tarefa desses agentes dentro de uma microárea, é cadastrar todas as pessoas e assim, manter os cadastros atualizados. Dessa forma, cada ACS tem responsabilidade sob uma microárea e que o número de pessoas não ultrapasse 750 pessoas (BRASIL, 2012).

Como o ACS é o primeiro da equipe ESF a ter o contato com a população, é ele quem identifica as condições de saúde, que pode ser uma situação que requer uma atenção mais especializada, como, de um enfermeiro ou médico, por exemplo, a realização de um curativo ou estabelecer um diagnóstico (no caso do médico) (KEBIAN; ACIOLI, 2014). Ter um melhor reconhecimento de problemas de saúde mais cedo, faz com que se tenha um diagnóstico mais preciso e também ajuda na prevenção de alguns tipos de doenças.

2.2. Sistemas de Informação em Saúde

Previamente, antes de entender o que é um Sistema de Informação em Saúde (SIS), é necessário saber o que é um sistema de informação. Sistemas de informação, de forma geral, são sistemas que buscam coletar, organizar, guardar, analisar e fornecer informações para uma determinada área, sem ter a necessidade de usar meios tecnológicos. É certo que há muitos sistemas de informação que dispõem de tecnologia da computação para realizar as mesmas ações que são realizadas em um sistema de informação tradicional (CAVALCANTE; SILVA; FERREIRA, 2011).

Os SIS são sistemas de informação aplicados à área da saúde. Eles são definidos como instrumentos que coletam dados, fazem o processamento desses dados, armazenam e distribuem as informações que irão ajudar no processo de prover um serviço de saúde à população com mais efetividade e eficiência, além de ter importância no processo de tomada de decisão e distribuir as tarefas das organizações de saúde (CAVALCANTE; SILVA; FERREIRA, 2011; MARTIN, 2010).

Assim, de acordo com Cavalcante, Silva e Ferreira (2011), “o principal objetivo de um SIS é apoiar e provocar mudanças na organização para melhorar o funcionamento dos processos de trabalho e o cuidado da saúde”. Esses sistemas voltados para o setor da saúde são inseridos, como por exemplo, em centros de atenção básica à saúde, em clínicas médicas, assim como, em prontuários eletrônicos.

Nesse sentido, a Política Nacional de Informação e Informática em Saúde (PNIIS), busca promover a utilização da tecnologia da informação para melhorar a organização dos processos de trabalho em saúde, de forma que haja melhoria na saúde da população. Assim, uns dos princípios da PNIIS a serem levados em conta é que a informação na área da saúde é

destinada ao gestor de saúde, ao trabalhador e aos cidadãos e que também seja capaz de gerar conhecimento. Deste modo, o uso da informática em saúde deve apoiar os profissionais dessa área, facilitando e organizando os registros que são realizados regularmente. Assim, a informação produzida ajudará a identificar os problemas de saúde individuais e coletivos da população (BRASIL, 2016).

3. METODOLOGIA

O principal método desta pesquisa foi o de desenvolvimento tecnológico, ou seja, o *design science research* onde o objetivo da pesquisa é gerar um produto. Este método de pesquisa é baseado em que um dado problema deve-se propor uma solução satisfatória que leve o problema para um estado melhor ou desejável. Além disso, este método é uma forma de diminuir a distância entre a teoria e a prática no que diz respeito a pesquisa (DRESCH; LACERDA; JUNIOR, 2015).

Além do método de desenvolvimento tecnológico, este estudo utiliza-se a pesquisa bibliográfica, onde procura-se explicar um problema tendo como base as referências teóricas publicadas em documentos e o pesquisador assume um papel crítico à frente dos documentos livros e artigos científicos, buscando especificar com nitidez o conteúdo. Além da modalidade de pesquisa bibliográfica, o estudo também faz uso da pesquisa descritiva ou de campo, procurando dados no ambiente relacionado ao tema, com a finalidade de registrar, analisar e interpretar os fatos, por meio de entrevistas, fazendo coleta de informações a respeito dos métodos utilizados.

Pela natureza deste estudo é empregado o método de abordagem hipotético-dedutivo que segue algumas etapas, no qual primeiramente dado um problema, busca-se encontrar uma solução provisória (teoria e tentativa), esta solução provisória faz crítica a pesquisa piloto de forma que elimine erros, depois este processo se renova gerando novos problemas.

Foi construído um aplicativo responsivo que facilita o acompanhamento de pacientes dentro de uma ESF, que está relacionado a UBS. Nessa estratégia de saúde, há uma população que está ligada de alguma forma ao aplicativo, que é composta pela enfermeira responsável pela ESF, as Agentes Comunitárias de Saúde que são compostas por 6 mulheres, e os pacientes que serão inseridos no sistema.

Para construir o aplicativo, foi necessário seguir algumas etapas da Engenharia de Software, assim como da Análise e Projeto de Sistemas. Etapas essas que fazem parte do processo de desenvolvimento de software, segundo Bezerra (2015), podem ser divididas em 6 (seis) atividades típicas, que são: levantamento de requisitos, análise, projeto (desenho),

implementação, testes e implantação. Seguindo ainda, uma metodologia de desenvolvimento de software é composta por todas as atividades para determinar, desenvolver, testar e conservar o software. Neste processo foi preciso definir quais, quando e como as atividades a serem executadas; controlar o progresso da construção; fazer a padronização dessa metodologia em uma forma de organização.

As atividades escolhidas para acompanhar o desenvolvimento do aplicativo são listadas e descritas a seguir. Dentre as atividades definidas por Bezerra (2015), utilizou-se as seguintes: o levantamento de requisitos, a análise e a implementação. A atividade de levantamento de requisitos é a etapa onde define-se e compreende-se o problema a ser resolvido pelo aplicativo. É nessa fase em que os usuários e desenvolvedores se entendem e juntos definem os requisitos do sistema. A Análise dos requisitos é a fase onde se estuda o produto do levantamento de requisitos. Nesta etapa o sistema é separado em componentes menores e busca-se entender como o sistema funciona, em outras palavras é a atividade na qual o objetivo é realizar uma análise aprofundada dos requisitos do sistema. A fase de implementação é onde o sistema é de fato criado com a codificação utilizando uma ou mais linguagens de programação (BEZERRA, 2015).

Na fase de implementação do aplicativo, as tecnologias usadas são relacionadas ao desenvolvimento de aplicações *web* (programas que podem ser acessados pelos navegadores através da internet). Assim, nesta fase, foi feito o estudo de linguagens como *HyperText Markup Language* (HTML), no português, linguagem de marcação de texto que é utilizada para a construção da estrutura do página *web* (MOZILLA, 2021a); *Cascading Style Sheets* (CSS) ou folhas de estilo em cascata, que é aplicada ao HTML para definir estilos e aparência à página *web* (MOZILLA, 2021b); por fim, tem-se a linguagem de programação *Javascript*, usada para dar funcionalidade e definir qual comportamento a aplicação deverá ter, além disso, pode ser encontrada em ambientes sem o navegador, como é no caso do *Node.js* (MOZILLA, 2021c). Essas linguagens são a base para se utilizar frameworks e bibliotecas de desenvolvimento *web*.

Ainda sobre a implementação, o sistema é composto por duas aplicações: uma representa o *front-end* e a outra o *back-end* e ambas se comunicam para que o aplicativo funcione de maneira correta. Podemos denominar o *front-end* como sendo o cliente e o *back-end* como o servidor. O front-end foi desenvolvido utilizando a biblioteca *JavaScript*

React, para o desenvolvimento da interface do aplicativo. Já *back-end*, foi desenvolvido utilizando o *Node.js* com o *framework Express*. Para guardar os dados da aplicação, foi escolhido o banco de dados PostgreSQL. Tanto para a construção do *back-end* quanto para o front-end a linguagem predominante usada foi o *Typescript*, que por sua vez é um superconjunto da linguagem *Javascript*. O principal objetivo do *Typescript* é ser um verificador de tipo estático, isto é, pode-se determinar erros de tipos de valores enquanto está escrevendo o código (MICROSOFT, 2022).

Podemos classificar o *front-end* como sendo o responsável pela interface gráfica do sistema, a parte em que o usuário tem acesso e pode interagir com ela. Já o *back-end* é o responsável por fornecer informações para o front-end, é ele quem salva as informações geradas no banco de dados.

Segundo Bezerra (2015), um componente humano importante para o processo de desenvolvimento é o especialista do negócio. Esse componente é apenas um indivíduo ou um grupo, que detém um certo grau de conhecimento da área em que o sistema será inserido. O especialista do negócio também é conhecido como cliente. Ele é quem vai utilizar o sistema ou que encomendou o sistema.

Ao término das atividades que fazem parte do desenvolvimento de *software*, a aplicação estará completa em sua primeira versão.

4. DESENVOLVIMENTO

Nesse ponto do trabalho, será abordado de fato o desenvolvimento do estudo, ou seja, o passo a passo para a criação do aplicativo. No decorrer das seguintes seções deste capítulo é apresentado o detalhamento de cada atividade realizada que faz parte do processo de construção do *software*. Neste sentido, a seção 4.1, apresenta a atividade de levantamento de requisitos; a seção 4.2, trata-se da análise de requisitos e por fim a seção 4.3 traz a implementação do aplicativo.

4.1. Levantamento de requisitos

Essa seção traz o levantamento de requisitos do aplicativo onde se produz o documento de requisitos. Este documento contém toda a informação acerca do domínio do sistema e engloba no mínimo duas principais divisões: a primeira é denominada como requisitos funcionais e a segunda como os requisitos não funcionais. Este documento deve ser escrito de maneira que os envolvidos no processo de desenvolvimento do *software* possam entender, e por essa razão não se pode utilizar conceitos técnicos que serão usados para construir o sistema. Um ponto desejável é que este documento esteja ordenado pelo grau de prioridade de um requisito, ou seja, o que deve ser feito primeiro e o que é considerado mais importante pelo usuário (BEZERRA, 2015).

4.1.1. Requisitos funcionais

Embasado nos requisitos funcionais do usuário, este tópico descreve de forma mais detalhada como os requisitos serão desenvolvidos no sistema. Portanto, os requisitos deste sistema são descritos seguindo o padrão em que “RF” é a abreviatura para requisitos funcionais, os dois dígitos formam o número daquele requisito, ele é seguido por um título, uma pequena descrição e por fim os usuários que irão usá-lo. Dessa forma os requisitos funcionais são:

[RF01] Acesso ao sistema

O sistema deverá ter acesso controlado, ou seja, apenas pessoas autorizadas podem ter acesso à aplicação. Para isso, o acesso só será permitido aos usuários que possuem um identificador, sendo o Cadastro de Pessoa Física (CPF) e uma senha.

Usuários: Administrador da ESF e Agente de Saúde.

[RF02] Manter microáreas

Deve ser possível fazer o gerenciamento de microáreas. Para isso, precisa-se armazenar o número da microárea e a geolocalização.

Usuário: Administrador da ESF.

[RF03] Manter condições de saúde

Deve ser possível gerenciar as condições de saúde. Para esse registro precisa-se da descrição e da sigla referente a condição de saúde.

Usuário: Administrador da ESF.

[RF04] Manter agentes de saúde

O sistema deve permitir que o administrador da ESF faça o cadastro dos agentes de saúde. Para o registro desses agentes de saúde são necessários os seguintes dados: nome, CPF e senha.

Usuário: Administrador da ESF.

[RF05] Manter pacientes

O sistema deverá permitir o gerenciamento de pacientes da ESF. Para isso serão necessários os seguintes dados: nome completo, CPF, sexo (masculino e feminino), data de nascimento, observação, endereço (rua, número, bairro e complemento), o ponto geográfico (coordenadas) da residência do paciente, a microárea a qual o ele pertence e as condições de saúde do mesmo.

Usuários: Agente de Saúde e Administrador da ESF.

[RF06] Gerar relatórios de pacientes

O Administrador da ESF deve ter a possibilidade de ver todos os dados dos pacientes, assim como, gerar relatórios personalizados a partir dos dados cadastrados no sistema.

Usuário: Administrador da ESF.

4.1.2. Requisitos não funcionais

Este tópico traz os requisitos não funcionais, que correspondem às qualidades que o sistema possa possuir e que estão ligadas às funcionalidades do mesmo. Da mesma forma que os RFs, aqui, os requisitos não funcionais recebem a abreviatura “RNF”. Portanto, os requisitos não funcionais são:

[RNF01] Restrições ao acesso

Somente usuários autorizados poderão ter acesso ao sistema.

[RNF02] Compatibilidade

O sistema deve ser acessado tanto em computadores quanto em celulares “smartphones”, ou seja, deverá ser responsivo.

[RNF03] Software

O sistema será desenvolvido utilizando a linguagem de programação *JavaScript*, tanto no *front-end* quanto no *back-end*.

4.2. Análise de requisitos

Esta seção apresenta a atividade de análise de requisitos do sistema, onde os modelos do sistema são construídos, como uma forma de entender mais a fundo o sistema. De maneira semelhante à atividade de levantamento de requisitos, a análise também não dá importância ao meio tecnológico, e sim, uma forma de validar e verificar se o sistema satisfaz as expectativas do usuário, como base nos modelos construídos (BEZERRA, 2015). Para realizar a análise foram utilizadas ferramentas *Unified Modeling Language* (UML) para a construção do diagrama de casos de usos, com o objetivo de criar um modelo para o sistema.

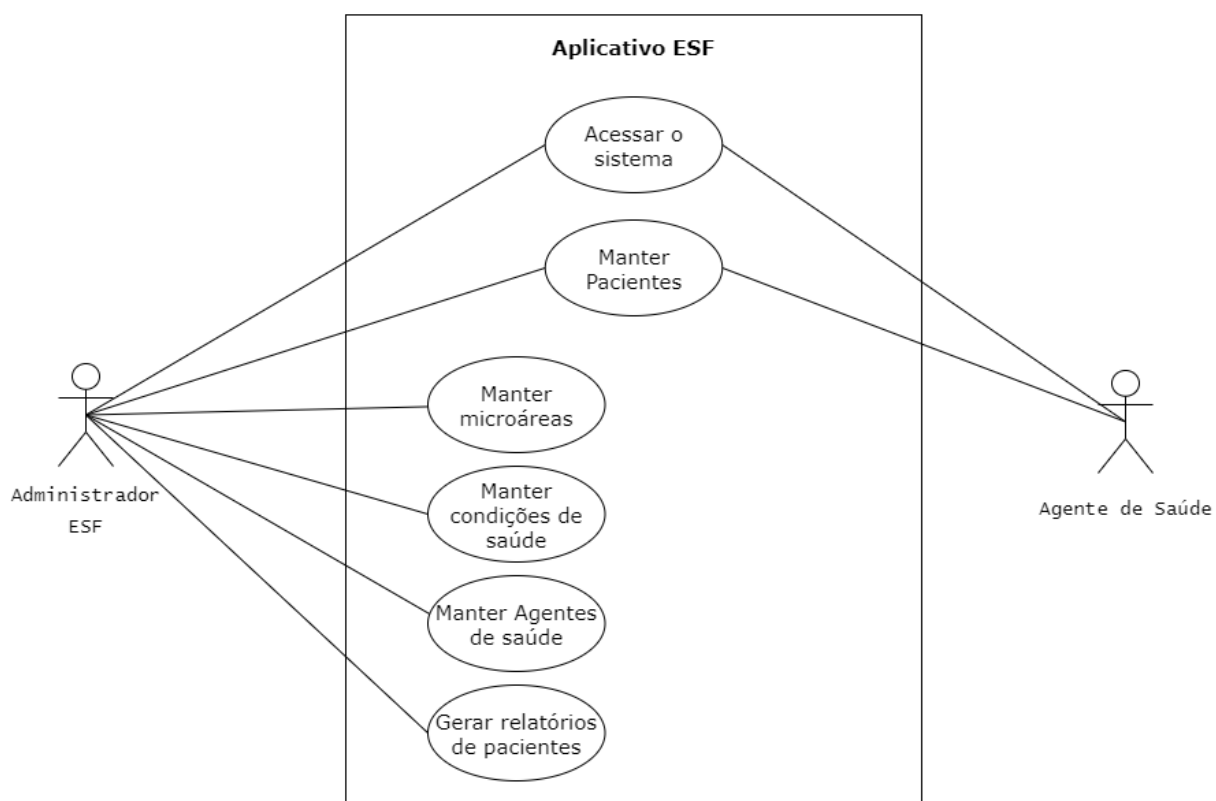
4.2.1. Modelo de casos de uso

O modelo de casos de uso é, em suma, um incremento dos requisitos funcionais do sistema, que faz uso de uma representação gráfica de fácil entendimento, envolvendo elementos como atores, casos de uso e relacionamentos. Para a elaboração desse modelo, utiliza-se o diagrama de casos de uso que, por sua vez, é uma ferramenta da UML.

4.2.1.1. Diagrama de casos de uso

Através de um diagrama de casos de uso da UML, pode-se ver que para o aplicativo de acompanhamento de pacientes da ESF tem-se dois atores, os quais são identificados como o Administrador ESF e o Agente de Saúde, além dos seis casos de uso, assim como está ilustrado na Figura 4.

Figura 4 - Diagrama de casos de uso para o aplicativo ESF



Fonte: O autor.

4.2.1.2. Descrição dos casos de uso

Na descrição dos casos de uso, o padrão adotado segue a seguinte estrutura: há um nome seguido pelo seu identificador como por exemplo: CSU01; e depois é informado o requisito funcional relacionado ao caso de uso. A parte que descreve o caso de uso utiliza os seguintes itens de descrição: sumário, ator primário ou atores, fluxo principal e fluxo alternativo. Deste modo, a seguir tem-se os seis casos de uso que o sistema possui.

Acessar o sistema (CSU01) - RF1 - RNF1

Sumário: O Administrador ESF e o Agente de Saúde podem acessar o sistema.

Atores: Administrador ESF e Agente de Saúde.

Fluxo Principal:

1. Os atores, preenchem os campos do formulário em branco na tela de acesso (*login*) ao sistema. Os campos são o CPF e senha.
2. O sistema verifica a validade dos dados, ou seja, se o usuário está cadastrado ou ativo no sistema. Se a validade dos dados estiver correta, o acesso ao sistema foi realizado com sucesso; caso contrário o sistema emite um erro com uma mensagem padronizada.
3. Com o acesso ao sistema realizado com sucesso. Cada ator será direcionado a uma tela de controle (Painel) respectiva a sua autorização (participação) que se tem no sistema.
4. O caso de uso termina.

Fluxo Alternativo (2): CPF ou Senha não são válidos.

- a) O ator informou um CPF ou senha que não estão cadastrados ou ativos no sistema.
- b) O ator recebe uma mensagem de erro do sistema e não pode acessar o sistema.

Fluxo Alternativo (2): Campos em branco.

- a) O ator não preencheu os campos do formulário corretamente.
- b) O ator recebe uma mensagem de erro do sistema pedindo para que o mesmo preencha corretamente os campos.

Manter microáreas (CSU02) - RF2

Sumário: O Administrador da ESF realiza o cadastro (inclusão, alteração e consulta) dos dados sobre as microáreas.

Ator Primário: Administrador ESF.

Fluxo Principal:

1. O Administrador da ESF requisita a manutenção das microáreas.
2. O sistema apresenta as operações que podem ser realizadas: a inclusão de uma nova microárea, a alteração dos dados de uma microárea e a consulta de microáreas.
3. O Administrador indica a opção a realizar ou finaliza o caso de uso.
4. O Administrador da ESF seleciona a operação desejada: Inclusão, Alteração ou Consulta.
5. Após a operação ter sido realizada, o caso de uso termina.

Fluxo Alternativo (4): Inclusão

- a) O Administrador requisita a inclusão de uma nova microárea.
- b) O sistema apresenta um formulário em branco para que os dados da microárea sejam preenchidos, os dados são: número da microárea e sua geolocalização.
- c) O Administrador preenche os campos do formulário.
- d) O sistema verifica a validade dos dados. Se os dados forem válidos, a microárea é incluída no sistema; caso contrário, o sistema envia um erro solicitando que os dados não válidos sejam corrigidos e reenviados.

Fluxo Alternativo (4): Alteração

- a) O Administrador da ESF solicita a alteração dos dados sobre uma microárea.
- b) O sistema traz um formulário com os dados já cadastrados da microárea a ser alterada.
- c) O Administrador altera as informações sobre uma microárea e requisita a sua atualização ou opta por finalizar o caso de uso.
- d) O sistema verifica a validade dos dados. Se os dados forem válidos, a microárea é alterada no sistema; caso contrário, o sistema envia um erro solicitando que os dados não válidos sejam corrigidos e reenviados.

Fluxo Alternativo (4) Consulta

- a) O Administrador da ESF solicita a realização de uma consulta sobre as microáreas cadastradas.
- b) O sistema apresenta uma lista contendo as microáreas.

Manter condições de saúde (CSU03) - RF3

Sumário: O Administrador da ESF realiza o cadastro (inclusão, remoção, alteração e consulta) dos dados sobre as condições de saúde.

Ator Primário: Administrador ESF.

Fluxo Principal:

1. O Administrador da ESF requisita a manutenção das condições de saúde.
2. O sistema apresenta as operações que podem ser realizadas: a inclusão de uma nova condição de saúde, a alteração dos dados de uma condição de saúde, a exclusão de uma condição de saúde e a consulta de condições de saúde.
3. O Administrador indica a opção a realizar ou opta por finalizar o caso de uso.
4. O Administrador da ESF seleciona a operação desejada: Inclusão, Remoção, Alteração ou Consulta.
5. Após a operação ter sido realizada, o caso de uso termina.

Fluxo Alternativo (4): Inclusão

- a) O Administrador requisita a inclusão de uma nova condição de saúde.
- b) O sistema apresenta um formulário em branco para que os dados da condição de saúde sejam preenchidos, os dados são: descrição e a sigla referente a condição de saúde.
- c) O Administrador preenche os campos do formulário.
- d) O sistema verifica a validade dos dados. Se os dados forem válidos, a condição de saúde é incluída no sistema com status “ativa”; caso contrário, o sistema envia um erro solicitando que os dados não válidos sejam corrigidos e reenviados.

Fluxo Alternativo (4): Remoção

- a) O Administrador solicita a remoção de uma determinada condição de saúde.
- b) O sistema atualiza o status da condição de saúde como sendo “inativa”.

Fluxo Alternativo (4): Alteração

- a) O Administrador da ESF solicita a alteração dos dados sobre uma condição de saúde.
- b) O sistema traz um formulário com os dados já cadastrados da condição de saúde a ser alterada.
- c) O Administrador altera as informações sobre uma condição de saúde e requisita a sua atualização ou opta por finalizar o caso de uso.
- d) O sistema verifica a validade dos dados. Se os dados forem válidos, a condição de saúde é alterada no sistema; caso contrário, o sistema envia um erro solicitando que os dados não válidos sejam corrigidos e reenviados.

Fluxo Alternativo (4) Consulta

- a) O Administrador da ESF solicita a realização de uma consulta sobre a lista de condições de saúde cadastradas.
- b) O sistema apresenta uma lista contendo as condições de saúde.

Manter Agentes de Saúde (CSU04) - RF5

Sumário: O Administrador da ESF realiza o cadastro (inclusão, remoção, alteração e consulta) de agentes de saúde.

Ator Primário: Administrador ESF.

Fluxo Principal:

1. O Administrador da ESF requisita a manutenção de agentes de saúde.
2. O sistema apresenta as operações que podem ser realizadas: a inclusão de um novo agente de saúde, a alteração dos dados de um agente de saúde, a remoção de um agente de saúde e a consulta de agentes de saúde.
3. O Administrador indica a opção a realizar ou finaliza o caso de uso.

4. O Administrador da ESF seleciona a operação desejada: Inclusão, Remoção, Alteração ou Consulta.
5. Após a operação ter sido realizada, o caso de uso termina.

Fluxo Alternativo (4): Inclusão

- a) O Administrador requisita a inclusão de um novo agente de saúde.
- b) O sistema apresenta um formulário em branco para que os dados sobre um agente de saúde sejam preenchidos, os dados do formulário são: nome, CPF e senha.
- c) O Administrador preenche os campos do formulário.
- d) O sistema verifica a validade dos dados. Se os dados forem válidos, ou seja, se o agente a ser incluído ainda não consta no sistema, então o agente é incluído no sistema com sucesso; caso contrário, o sistema envia um erro solicitando que os dados não válidos sejam corrigidos e a verificação é repetida.

Fluxo Alternativo (4): Remoção

- a) O Administrador solicita a remoção de um agente de saúde.
- b) O sistema atualiza o status do agente de saúde como sendo “inativo”. A partir desse momento o agente não poderá mais acessar o sistema.

Fluxo Alternativo (4): Alteração

- a) O Administrador da ESF solicita a alteração dos dados sobre um agente de saúde.
- b) O sistema traz um formulário com os dados já cadastrados do agente de saúde a ser alterado.
- c) O Administrador altera as informações e requisita a atualização ou opta por finalizar o caso de uso.
- d) O sistema verifica a validade dos dados. Se os dados forem válidos, ou seja, se as informações modificadas do agente não afetam o funcionamento do sistema, então o agente é alterado no sistema; caso contrário, o sistema envia um erro solicitando que os dados não válidos sejam corrigidos e a verificação é repetida.

Fluxo Alternativo (4) Consulta

- a) O Administrador da ESF solicita a realização de uma consulta sobre a lista de agentes de saúde cadastrados no sistema.
- b) O sistema apresenta uma lista com os agentes de saúde.

Manter pacientes (CSU5) - RF6

Sumário: O Administrador ESF e o Agente de Saúde realizam o cadastro (inclusão, alteração e consulta) de pacientes.

Atores: Administrador ESF e Agente de Saúde.

Fluxo Principal:

1. O ator requisita a manutenção de pacientes.
2. O sistema apresenta as operações que podem ser realizadas: a inclusão de um novo paciente, a alteração dos dados de um paciente e a consulta de pacientes.
3. O ator indica a opção a realizar ou finaliza o caso de uso.
4. O ator seleciona a operação desejada: Inclusão, Remoção, Alteração ou Consulta.
5. Após a operação ter sido realizada, o caso de uso termina.

Fluxo Alternativo (4): Inclusão

- a) O ator requisita ao sistema a inclusão de um novo paciente.
- b) O sistema apresenta um formulário em branco para que os dados sobre um paciente sejam preenchidos, os dados do formulário são: nome; CPF; sexo (masculino ou feminino); data de nascimento; observação sobre o paciente; as condições de saúde que o paciente apresenta; o endereço completo com logradouro, número, bairro e complemento; a microárea em que ele se situa o ponto geográfico da residência.
- c) O ator preenche os campos do formulário.
- d) O sistema verifica a validade dos dados. Se os dados forem válidos, o paciente é incluído no sistema; caso contrário, o sistema envia um erro solicitando que os dados não válidos sejam corrigidos e a verificação é repetida.

Fluxo Alternativo (3): Alteração

- a) O ator solicita a alteração dos dados sobre um paciente.
- b) O sistema traz um formulário com os dados já cadastrados do paciente a ser alterado.
- c) O ator altera as informações e requisita a atualização ou opta por finalizar o caso de uso.
- d) O sistema verifica a validade dos dados. Se os dados forem válidos, o paciente é atualizado no sistema; caso contrário, o sistema envia um erro solicitando que os dados não válidos sejam corrigidos e a verificação é repetida.

Fluxo Alternativo (3) Consulta

- a) O ator solicita a realização de uma consulta sobre a lista de pacientes cadastrados.
- b) O sistema apresenta uma lista contendo os pacientes.

Gerar relatórios de pacientes (CSU06) - RF4

Sumário: O Administrador da ESF gera relatórios personalizados de pacientes.

Ator Primário: Administrador ESF.

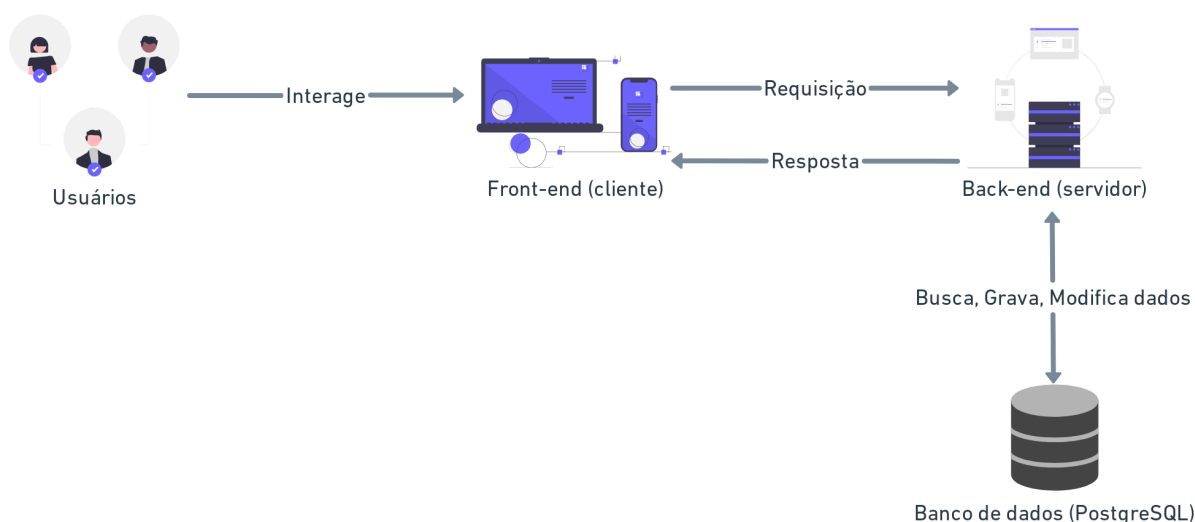
Fluxo Principal:

1. O Administrador da ESF solicita ao sistema um relatório de pacientes.
2. O sistema traz opções sobre os pacientes para o ator selecionar.
3. O Administrador seleciona as opções que o convém ou opta por finalizar o caso de uso.
4. O sistema gera o relatório e coloca à disposição do Administrador da ESF.
5. O caso de uso termina.

4.3. Implementação do aplicativo

Essa seção trata da codificação do aplicativo *web* responsivo. Portanto, temos o desenvolvimento do *front-end*, ou seja, a interface gráfica do aplicativo. Essa interface foi construída utilizando o *React* com algumas bibliotecas de apoio para trabalhar com mapas, validações de formulários, gerenciamento de rotas, buscar informações do *back-end*, renderizar mensagens instantâneas, entre outras. Em conjunto com o *front-end*, o autor deste trabalho realizou a construção do *back-end* que, neste caso, atende as requisições (mensagens/solicitações enviadas a partir do cliente) como por exemplo, autenticar o acesso de um usuário, realizar cadastros, fazer buscas e retornar os resultados, entre outras solicitações. A Figura 5 apresenta um esquema de como as partes da aplicação funcionam quando ocorre uma requisição.

Figura 5 - Esquema de funcionamento da aplicação



Fonte: O autor

A fim de exemplificar melhor a comunicação entre as partes da aplicação, veremos como um usuário acessa o sistema. Dessa maneira, temos a primeira página *web* da aplicação que corresponde ao primeiro requisito funcional, o de “Acesso ao sistema”, onde os atores entram o CPF e senha para acessar o sistema, sendo chamado de *login*. Para realizar o acesso à aplicação, após o usuário preencher sem erros o formulário, uma chamada ao *back-end* ocorre e as informações preenchidas pelo usuário são enviadas através de uma requisição *Hypertext Transfer Protocol* (HTTP) usando o verbo POST.

Os verbos ou métodos de requisição HTTP, são usados para dar semântica às ações a serem executadas para um certo recurso. No caso do método POST, ele é utilizado quando há a necessidade de enviar/salvar uma informação a um recurso, onde pode haver mudanças no recurso. Sendo mais prático, esse método é usado geralmente quando se quer criar, modificar uma entidade ou trazer segurança quando está realizando *login* na aplicação (MOZILLA, 2022).

Seguindo com o exemplo do *login*, depois que os dados do usuário chegam ao *back-end*, ocorrem alguns passos. Seguindo o código da Figura 6, na linha 3 temos o primeiro passo da rota de fazer *login* que é pegar os dados do corpo da requisição que vem do *front-end*; nas linha 5 a 9, há uma busca no banco de dados para retornar o usuário conforme o CPF; depois é feita uma verificação para ver se o usuário está cadastrado na aplicação. Se o mesmo existe o próximo passo é executado, caso contrário é retornado um erro ao *front-end*. Esse passo é descrito nas linhas 12 a 14; nas linhas 16 a 18 é feita uma verificação para certificar que o usuário esteja ativo na aplicação.

Seguindo com os passos, temos a verificação de senha, ou seja, a senha que é recebida pelo corpo da requisição é comparada com a que está criptografada e gravada no banco de dados, nesta verificação é utilizada a biblioteca *bcrypt* que, auxilia na criptografia e verificação de senhas, vemos isso nas linhas 20 a 24. Caso a senha esteja correta, os últimos dois passos são executados. Dessa forma, as linhas 32 a 36 correspondem a geração de um *token* de autenticação com o padrão *JSON Web Token* (JWT). Após a geração, o último passo é executado, onde o servidor retorna ao cliente o *token* gerado e um objeto *Javascript* contendo algumas informações sobre o usuário que está acessando o sistema. A Figura 6, apresenta todo o código dos passos detalhados e na Figura 7, tem-se o formulário de *login* da aplicação no lado do cliente.

O padrão JWT define uma forma compacta para a transmissão segura de dados entre o cliente e servidor utilizando um tipo *JavaScript Object Notation* (JSON) e na aplicação é usado principalmente para manter o usuário logado por um determinado tempo no sistema (24 horas) e também é utilizado para controlar a autorização/participação de cada usuário no front-end (JWT, 2022). A biblioteca que foi utilizada para a geração e verificação de *tokens* foi a *jsonwebtoken*.

Figura 6 - Código do acesso ao sistema do lado do *back-end*

```
1 // Rota de login da aplicação
2 UsersRouter.post('/login', async (request: Request, response: Response): Promise<Response> => {
3   const { cpf, password } = request.body;
4
5   const user = await prisma.user.findUnique({
6     where: {
7       cpf,
8     }
9   });
10
11 // se o usuário não foi encontrado
12 if (!user) {
13   return new AppError(400, "CPF ou senha estão incorretos!").execute(response);
14 }
15
16 if (user.deletedAt) {
17   return new AppError(400, "Usuário desativado!").execute(response);
18 }
19
20 const checkPassword = verifyPassword(password, user.password);
21 // verifica se a senha está correta
22 if (!checkPassword) {
23   return new AppError(400, "CPF ou senha estão incorretos!").execute(response);
24 }
25
26 const userInfo = {
27   id: user.id,
28   role: user.role,
29   name: user.name
30 };
31
32 // Cria token de autenticação
33 const token = jwt.sign({}, process.env.JWT_SECRET, {
34   subject: user.id,
35   expiresIn: "1d",
36 });
37
38 return response.status(200).json({
39   user: userInfo,
40   token
41 });
42 });
```

Fonte: O autor.

Figura 7 - Formulário de acesso ao sistema

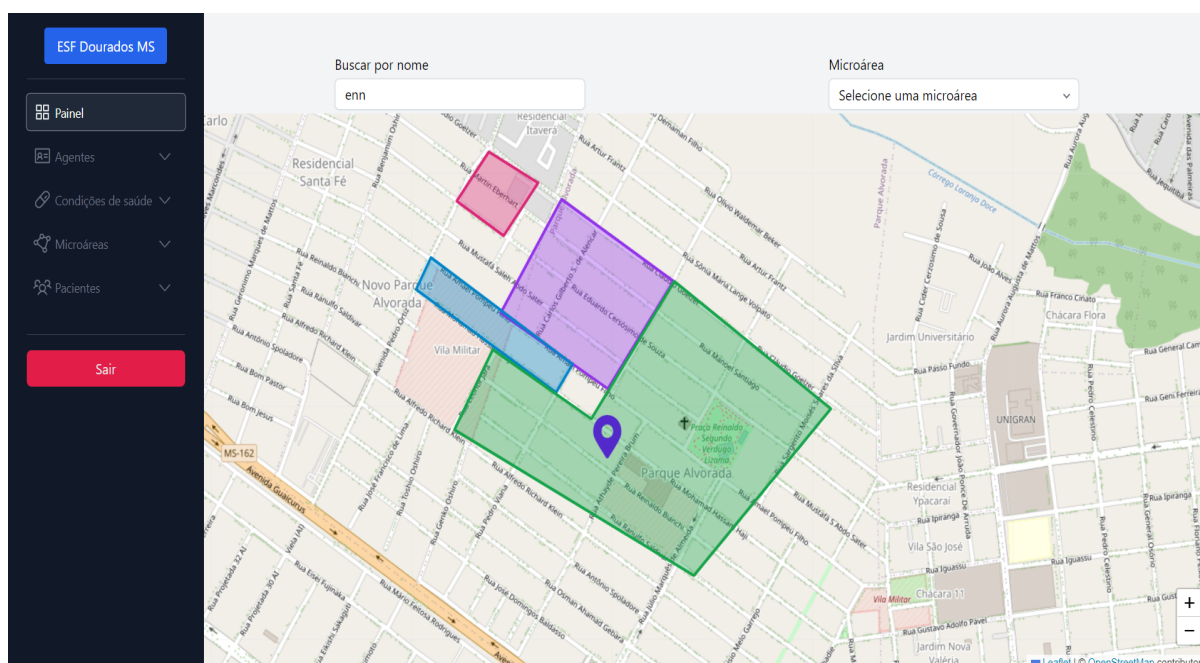
Formulário de login com o título "Faça seu login". O formulário contém dois campos de entrada: "CPF" com o valor "999.999.999-99" e "Senha" com o placeholder "Digite sua senha". Abaixo dos campos há um botão azul com o texto "Entrar".

Fonte: O autor.

Um ponto importante da aplicação são as microáreas e as localizações geográficas dos pacientes, dessa forma, quando o Agente de Saúde ou Administrador ESF acessam o sistema, ambos vêm um mapa com todas as microáreas. Essa parte corresponde à tela principal da aplicação e nesta página, é possível pesquisar um ou mais pacientes pela inicial do nome ou pela microárea. Pode-se notar que tirando a tela de *login* todas as outras páginas do sistema contém um menu de navegação. A Figura 8, traz essa página que chamaremos de Painel.

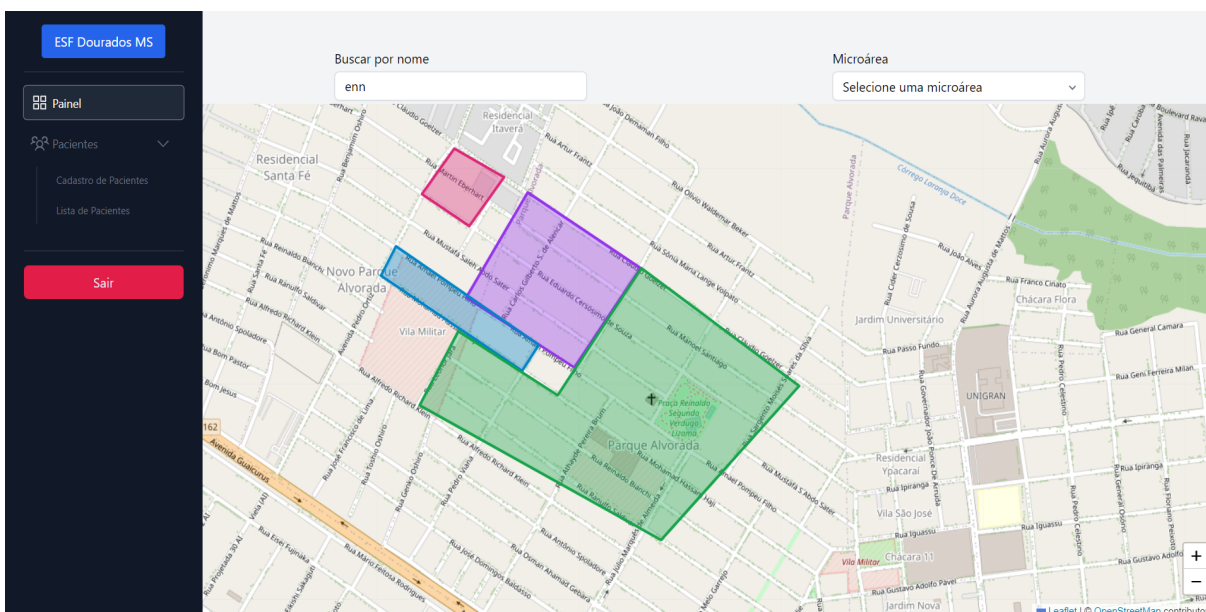
Em detalhes, os polígonos que estão plotados no mapa da Figura 8 representam as microáreas cadastradas e o ícone de localização correspondente ao paciente encontrado pela busca pela inicial de nome “enn”. As microáreas são criadas apenas pelo Administrador ESF, por conta da sua autorização no sistema e portanto apenas ele tem acesso ao item do menu “Microáreas” e aos seus subitens, isso ocorre também para outros itens de menus como o de “Agentes”, “Condições de saúde”.

Figura 8 - Página inicial do sistema (Painel) na visão do Administrador ESF



Fonte: O autor.

Figura 9 - Página inicial do sistema (Painel) na visão do Agente de saúde



Fonte: O autor.

A fim de comparação entre as visões dos usuários do sistema tem-se que, o menu de navegação que fica ao lado esquerdo da tela, para o agente de saúde é reduzido se comparado com o do administrador ESF. Outra comparação é na pesquisa, a inicial de nome “enn” é buscada na Figura 9, porém ele não traz nenhum ícone de localização, uma vez que aquele paciente encontrado na Figura 8 foi criado por outro agente e neste caso só quem realiza o cadastro e o administrador podem vê-lo.

Os mapas que aparecem nas páginas do front-end foram gerados utilizando as bibliotecas *Leaflet* e *React Leaflet*, ambas utilizando o *OpenStreetMap* para atribuição de mapas. O *Leaflet* é uma biblioteca *Javascript* para criação de mapas interativos e compatíveis com plataformas de desktop e dispositivos móveis (AGAFONKIN, 2022). Já o *React Leaflet* é também uma biblioteca *Javascript*, que fornece ligações entre o *React* e o *Leaflet*, ou seja, não substitui o *Leaflet*, na verdade ele pega algumas camadas do *Leaflet* e abstrai em componentes *React* (CAM et al., 2022).

Dentre todas as páginas codificadas do aplicativo, a de busca de pacientes foi a última a ser implementada. Essa página além de fazer uma busca personalizada, traz o último requisito funcional (RF06) “Gerar relatórios de pacientes”, onde apenas o administrador ESF tem acesso. Assim que o ator acessa a página, um formulário de busca é apresentado com

alguns campos opcionais. Assim que o ator clica no botão “Buscar”, uma requisição é feita ao *back-end* e a resposta é uma lista contendo pacientes que atendem os campos selecionados no formulário. A lista retornada é apresentada em uma tabela. A geração de relatório ocorre depois da busca, há um botão que fica abaixo da tabela como o título de “Gerar PDF”. Ao clicar neste botão uma nova guia do navegador é aberta contendo um arquivo PDF com o relatório. O relatório é, basicamente, uma lista de pacientes encontrados a partir da busca personalizada. Na Figura 10, pode-se ver o formulário da página de busca em branco e sem ter feito a busca, já na Figura 11 tem-se a página depois da busca.

Trabalhar com geração de PDF não é uma tarefa simples, precisa-se configurar todas as definições do documento a ser criado, para isso foi usada a biblioteca *PdfMake* que tem como objetivo fazer a geração de documentos PDF para uso tanto no lado do servidor quanto para o lado do cliente (PDFMAKE, 2022). Neste caso, o *PdfMake* foi utilizado no lado do cliente, ou seja, junto com o *React*.

Figura 10 - Formulário de busca de pacientes sem os resultados de busca

Busca de Pacientes

Personalize sua busca

Agentes
Selecione um agente

Microárea
Selecione uma microárea

Condição de Saúde
Selecione as condições de saúde

+ Busca avançada

Limpar Buscar

Resultado da busca

Nenhum paciente encontrado!

Gerar PDF

Fonte: O autor.

Figura 11 - Busca de pacientes depois de realizar uma busca

Personalize sua busca

Agentes

Microárea

Condição de Saúde

⊕ Busca avançada

Limpar
Buscar

Resultado da busca

CPF	NOME	MICROÁREA	IDADE	AÇÕES
231.231.313-11	Luis Henrique	4	18	Alterar
231.313.131-31	Ennery	4	24	Alterar

Gerar

Fonte: O autor.

Figura 12 - Exemplo de PDF Gerado

Relatório de Pacientes

CPF	Nome	Idade	Sexo	Condição de Saúde
231.231.313-11	Luis Henrique	18	MASCULINO	
231.313.131-31	Ennery	24	MASCULINO	DT1, HIP,

Fonte: O autor.

A Figura 12 apresenta uma parte de um PDF gerado a partir dos dados dos pacientes buscados, conforme apresentado na Figura 11. Para esse exemplo temos dois pacientes que atendem à condição de busca sobre a microárea 4.

Todos os dados apresentados nas figuras são fictícios, foram criados com objetivo de testar e demonstrar a execução do aplicativo. Porém toda a codificação foi desenvolvida no intuito de guardar dados reais e planejada para ser expandida.

5. CONCLUSÃO

A partir deste presente trabalho alguns resultados foram obtidos, entre eles está o aplicativo que busca auxiliar no acompanhamento de pacientes da ESF, o mesmo é responsivo aos diversos dispositivos, sejam eles móveis ou *desktop*. Além disso, o aplicativo possibilita cadastros e acessos necessários para que seja possível realizar o acompanhamento de pacientes em uma ESF de forma moderna. O objetivo só foi alcançado por causa de estudos referentes às tecnologias necessárias para a construção de sistemas *web*, e, além de tudo, estudos relacionados com o domínio e ecossistema do projeto, em outras palavras, respondendo algumas perguntas como por exemplo: “o que é uma ESF?”, “como funciona o acompanhamento de pacientes em uma ESF?”, “quem são os componentes humanos desse ambiente?”, “o que é necessário ter no aplicativo?”, entre outras.

Por conseguinte, com o decorrer do desenvolvimento deste trabalho, foi possível adquirir novos conhecimentos acerca do desenvolvimento de aplicativos *web* e a forma de estruturar um sistema para atender o problema proposto. Assim, o produto gerado a partir deste trabalho tem como objetivo mostrar que se pode utilizar a tecnologia na área da saúde, especificamente, aplicando ao acompanhamento de pacientes da ESF.

Durante as pesquisas e o andamento deste trabalho, foram surgindo questionamentos e curiosidades com o intuito de ampliar o sistema, esses constituíram em propósitos para serem aplicados em futuras pesquisas. Esses propósitos foram separados em alguns itens, como:

- Expandir o sistema para que o mesmo possa ter várias ESF cadastradas em uma mesma UBS;
- Fazer uma possível integração com o sistema do governo para que diminua o trabalho, ou seja, o aplicativo deste trabalho terá informações do sistema do governo e assim não precisará passar informações de um sistema para o outro;
- E sendo mais ousado, ampliar o sistema para comportar todas as UBS do município. Fazendo novas hierarquias e visões (permissões) de diferentes usuários.

REFERÊNCIAS BIBLIOGRÁFICAS

AGAFONKIN, Volodymyr. **Leaflet - Overview**. Leaflet, 2022. Disponível em: <https://leafletjs.com/index.html>. Acesso em: 23 set. 2022.

BEZERRA, Eduardo. **Princípios de análise e projeto de sistemas com UML**. 3 .ed. Rio de Janeiro: Elsevier, 2015.

BRASIL. Ministério da Saúde. **Política Nacional de Atenção Básica**. Brasília: Ministério da Saúde, 2012.

BRASIL. Ministério da Saúde. **Política Nacional de Informação e Informática em Saúde**. Brasília: Ministério da Saúde. 2016.

CAM, Paul Le. **React-Leaflet - Introduction**. React Leaflet, 2022. Disponível em: <https://react-leaflet.js.org/docs/start-introduction/>. Acesso em: 24 set. 2022.

CAVALCANTE, Ricardo Bezerra; SILVA, Poliana Cavalcante; FERREIRA, Marina Nagata. Sistemas de informação em saúde: possibilidades e desafios. **REUFISM - Revista de enfermagem da UFSM**. [S.I.], v. 1, n. 2, p. 290-299, mai./ago. 2011. Disponível em: <https://periodicos.ufsm.br/reufism/article/view/2580>. Acesso em: 10 jul. 2022.

DRESCH, Aline; LACERDA, Daniel Pacheco; JUNIOR, José Antonio Valle Antunes. **Design Science research: método de pesquisa para avanço da ciência e tecnologia**. Porto Alegre: Bookman, 2015.

JWT. **Introduction to JSON Web Tokens**. Auth0, 2022. Disponível em: <https://jwt.io/introduction>. Acesso em 23 set. 2022.

MACINKO, James; MENDONÇA, Claunara Schilling. **Estratégia saúde da família, um forte modelo de atenção primária à saúde que traz resultados**. Revista Saúde Debate, Rio de Janeiro, v. 42, p. 18-37, 2018. Disponível em: <https://www.scielo.br/j/sdeb/a/Kr7jdgRFHmdqnMcP3GG8JTB/?lang=pt&format=pdf>. Acesso em: 28 jun. 2022.

MARTIN, Heimar de Fátima. Sistemas de informação em saúde: considerações gerais. JHI - Journal of Health Informatics. [S.I], v. 2, n. 1, p-20-24, jan./mar. 2010. Disponível em: <https://jhi.sbis.org.br/index.php/jhi-sbis/article/view/4>. Acesso em: 11 jul. 2022.

MICROSOFT. **TypeScript Documentation**. Typescript Lang, 2022. Disponível em: <https://www.typescriptlang.org/docs/>. Acesso em: 24 set. 2022.

MOZILLA. **HTML**: Linguagem de marcação de hipertexto. MOZILLA, 2021a. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTML>. Acesso em: 30 jun. 2022.

MOZILLA. **CSS**. MOZILLA, 2021b. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/CSS>. Acesso em: 30 jun. 2022.

MOZILLA. **JavaScript**. MOZILLA, 2021c. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>. Acesso em: 06 jul. 2022.

MOZILLA. **Métodos de requisição HTTP**. MOZILLA, 2022. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Methods>. Acesso em: 22 set. 2022.

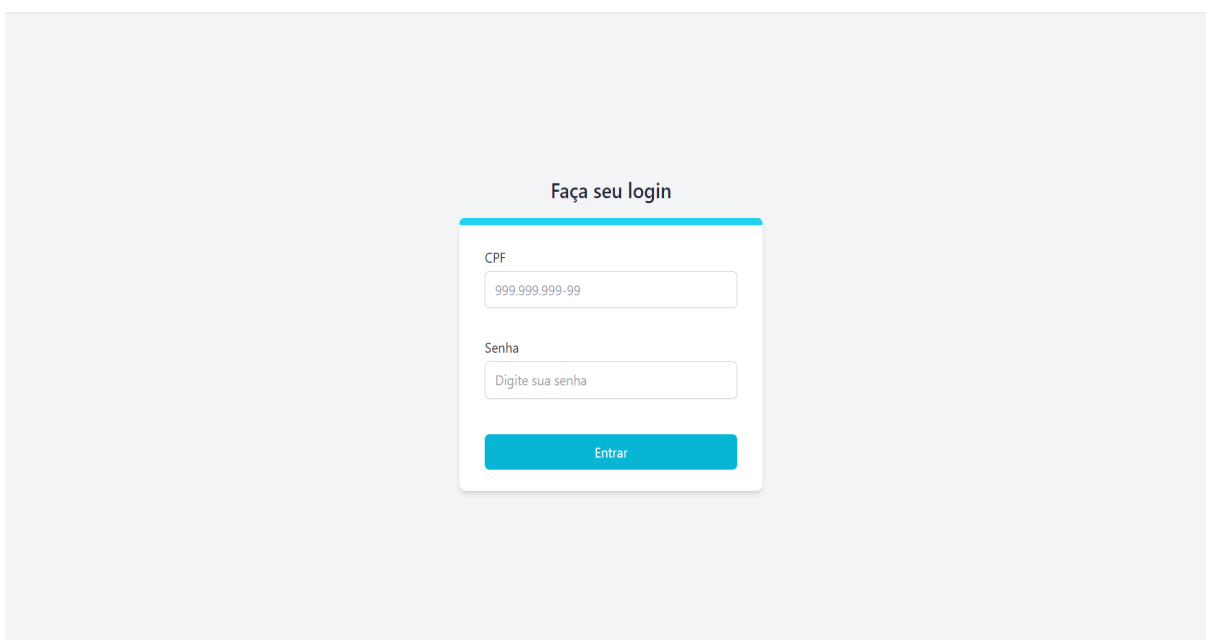
PDFMAKE. **Pdfmake**. Pdfmake, 2022. Disponível em: <https://pdfmake.github.io/docs/0.1/getting-started/>. Acesso em: 25 set. 2022.

APÊNDICE A - PÁGINAS DA APLICAÇÃO FRONT-END

A seguir são mostradas as páginas desenvolvidas da aplicação *front-end*, ou seja, a parte da interface gráfica do aplicativo *web*. Primeiramente, temos a página inicial para todos os usuários, ou seja a página que controla o acesso ao sistema, esta é comum para todos os usuários. Depois é mostrado as telas que pertencem à visão do administrador ESF e as páginas que são pertencentes aos ACS e ao mesmo tempo ao administrador, no sentido gráfico e não de funcionamento. Por fim são mostradas algumas páginas com o tamanho menor, ou seja, para celulares, onde podemos ver a responsividade do aplicativo. Um ponto importante é que os dados que estão nas figuras não são dados reais, são apenas para fins demonstrativos.

A.1 - Página de acesso ao sistema

Figura 13 - Página de acesso ao sistema



A imagem mostra a interface de login de um sistema web. No topo, o texto "Faça seu login" está centralizado. Abaixo dele, há um formulário com dois campos de entrada: "CPF" com o valor "999.999.999-99" e "Senha" com o placeholder "Digite sua senha". Um botão azul com o texto "Entrar" está posicionado abaixo dos campos.

Fonte: O autor

A.2 - Páginas do administrador ESF

Figura 14 - Página de criação de um novo agente de saúde

ESF Dourados MS

Painel

Agentes

Criar novo agente

Lista de agentes

Condições de saúde

Microáreas

Pacientes

Sair

Adicione um Agente

Nome Completo

CPF

Password

Cancelar Salvar

Fonte: O autor.

Figura 15 - Página de listagem de agentes de saúde

ESF Dourados MS

Painel

Agentes

Criar novo agente

Lista de agentes

Condições de saúde

Microáreas

Pacientes

Sair

Lista de Agentes

CPF	NOME	STATUS	AÇÕES
231.019.112-12	Alba	Ativo	Desativar Alterar
123.213.131-13	Maria	Ativo	Desativar Alterar

Fonte: O autor.

Figura 16 - Página de edição de um agente de saúde

A imagem mostra a interface de usuário para a edição de um agente de saúde. No topo, há um botão azul com o texto "ESF Dourados MS". Abaixo dele, um menu lateral escuro contém opções como "Painel", "Agentes" (selecionado), "Criar novo agente", "Lista de agentes", "Condições de saúde", "Microáreas" e "Pacientes", além de um botão vermelho "Sair". O conteúdo principal da página é branco e contém o título "Editando Alba". Abaixo do título, há um formulário com os seguintes campos: "Nome Completo" com o valor "Alba", "CPF" com o valor "231.019.112-12" e "Password" com caracteres ocultos por pontos. No final do formulário, há dois botões: "Cancelar" em vermelho e "Salvar" em azul.

Fonte: O autor.

Figura 17 - Página de criação de uma nova condição de saúde

A imagem mostra a interface de usuário para a criação de uma nova condição de saúde. No topo, há um botão azul com o texto "ESF Dourados MS". Abaixo dele, o mesmo menu lateral escuro é exibido, mas com "Condições de saúde" selecionado. O conteúdo principal da página é branco e contém o título "Criar condição de saúde". Abaixo do título, há um formulário com os seguintes campos: "Condição de saúde" com o placeholder "Sigla da condição de saúde" e "Descrição" com o placeholder "Digite a descrição para a condição de saúde". No final do formulário, há dois botões: "Cancelar" em vermelho e "Salvar" em azul.

Fonte: O autor.

Figura 18 - Página de listagem de condições de saúde

The screenshot shows a web application interface for managing health conditions. On the left is a dark sidebar with a menu containing 'Painel', 'Agentes', 'Condições de saúde' (selected), 'Criar condição de saúde', 'Microáreas', and 'Pacientes', along with a 'Sair' button. The main content area is titled 'Lista de condições de saúde' and displays a table with two rows of data.

SIGLA	DESCRIÇÃO	STATUS	AÇÕES
DTI	Diabetes Tipo 1	Ativo	<button>Desativar</button> <button>Alterar</button>
HIP	Hipertensão (pressão alta)	Ativo	<button>Desativar</button> <button>Alterar</button>

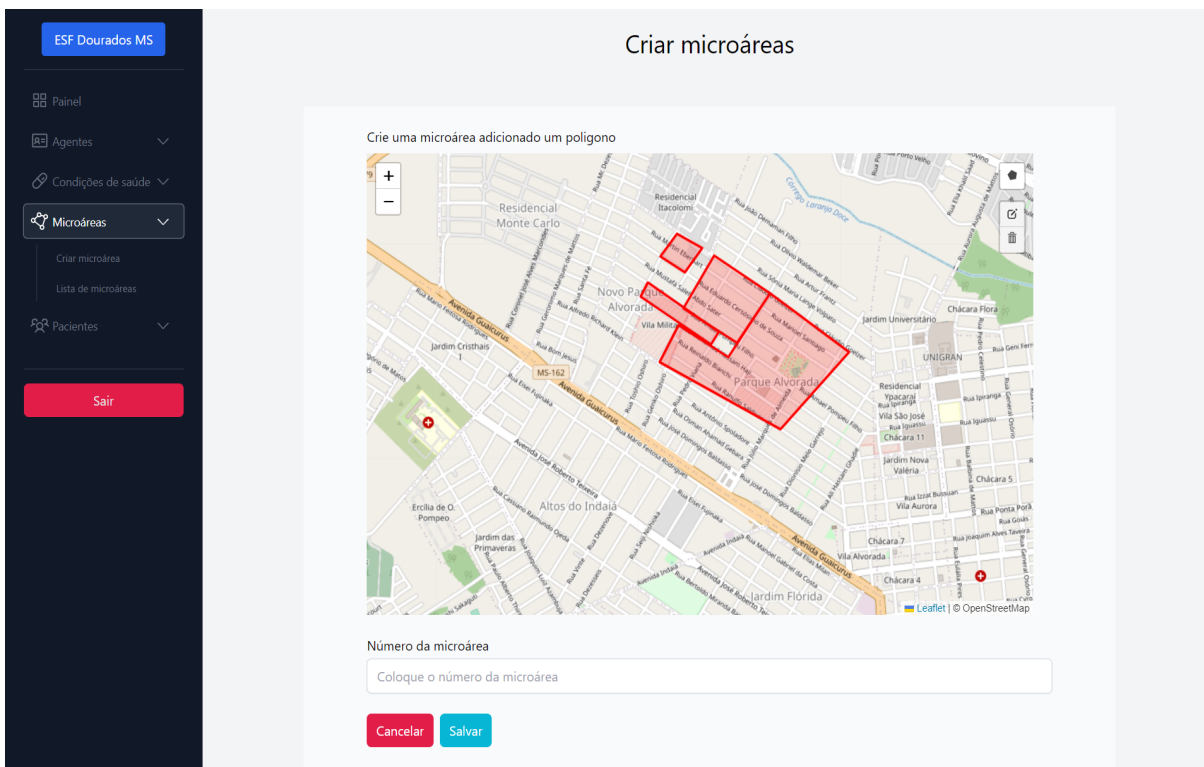
Fonte: O autor.

Figura 19 - Página de edição de uma condição de saúde

The screenshot shows the editing interface for a health condition. The sidebar is identical to the previous figure. The main content area is titled 'Editando condição de saúde: HIP' and contains a form with two input fields: 'Sigla da condição de saúde' (containing 'HIP') and 'Descrição' (containing 'Hipertensão (pressão alta)'). Below the form are 'Cancelar' and 'Salvar' buttons.

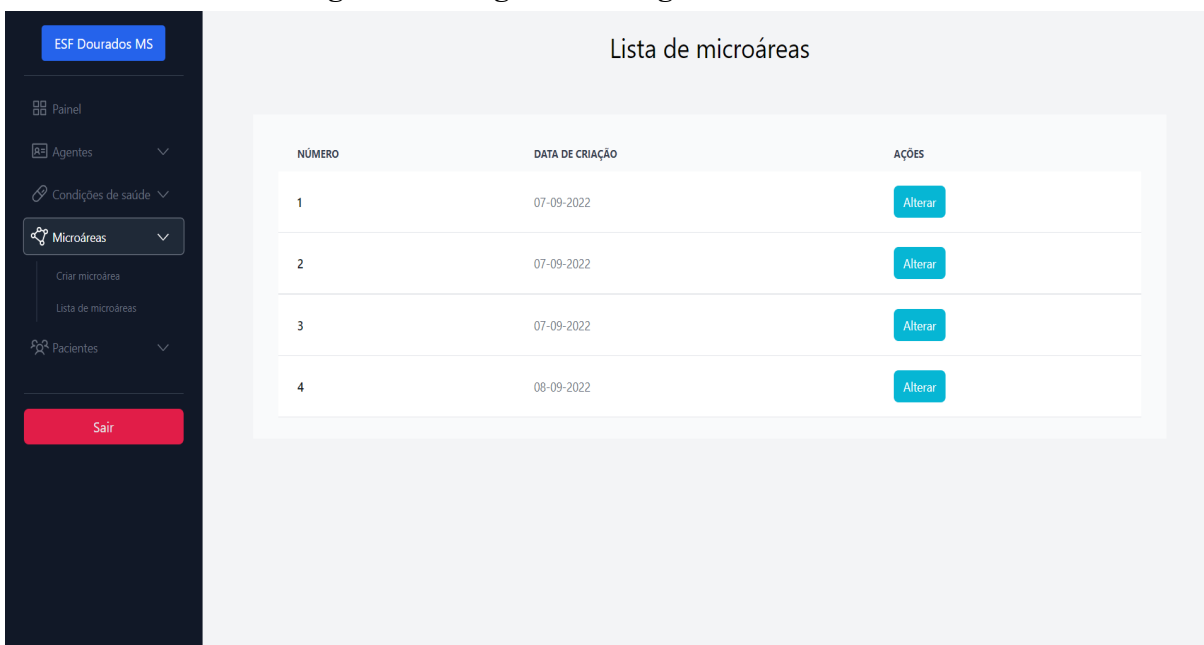
Fonte: O autor.

Figura 20 - Página de criação de microáreas



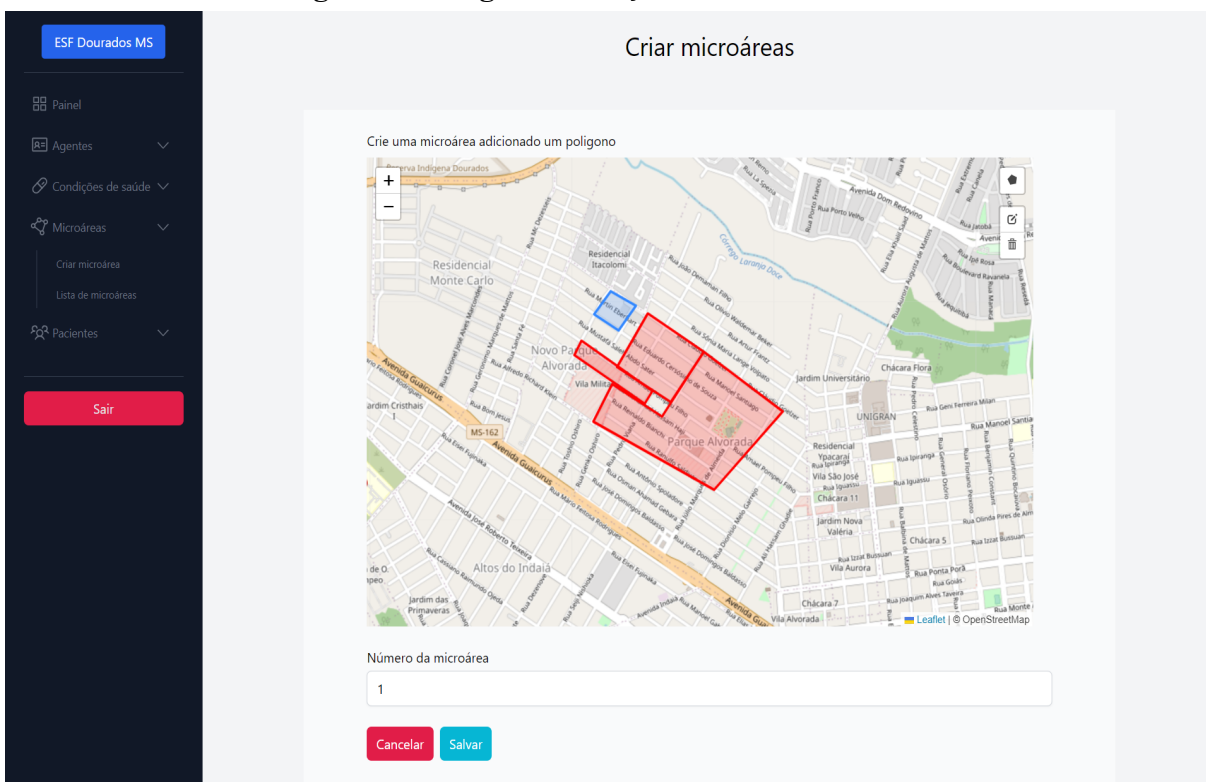
Fonte: O autor.

Figura 21 - Página de listagem de microáreas



Fonte: O autor.

Figura 22 - Página de edição de uma microárea



Fonte: O autor.

Figura 23 - Página de busca de pacientes com a geração de relatório PDF

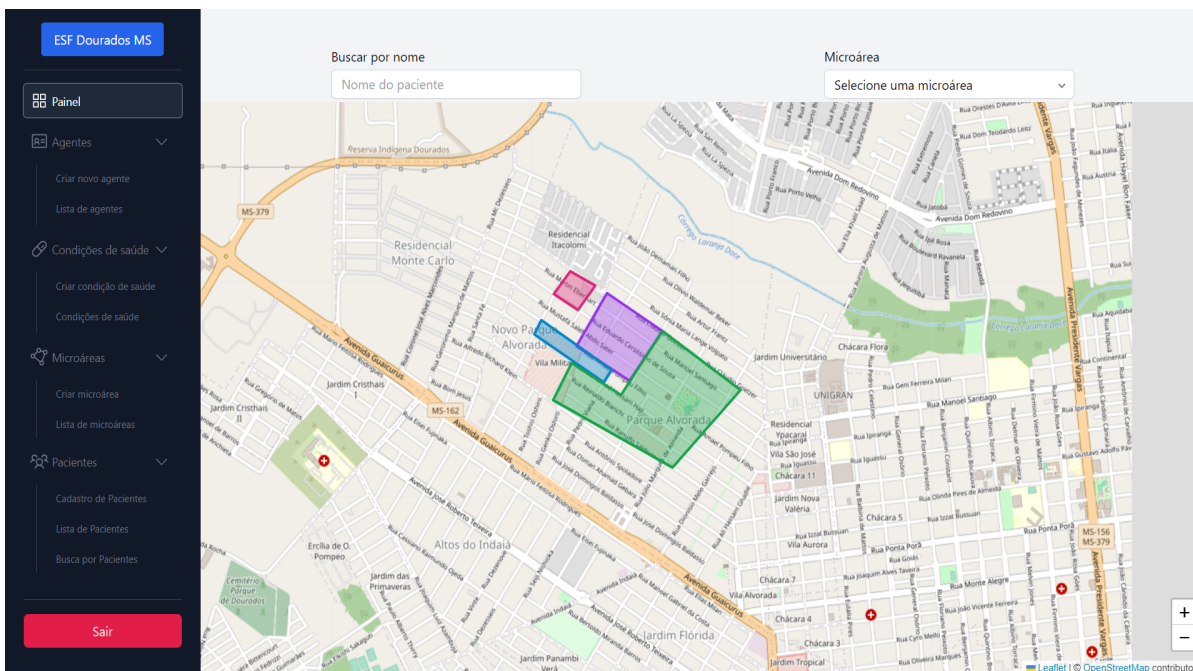


Fonte: O autor.

A.3 - Páginas do Agente de Saúde e do Administrador ESF

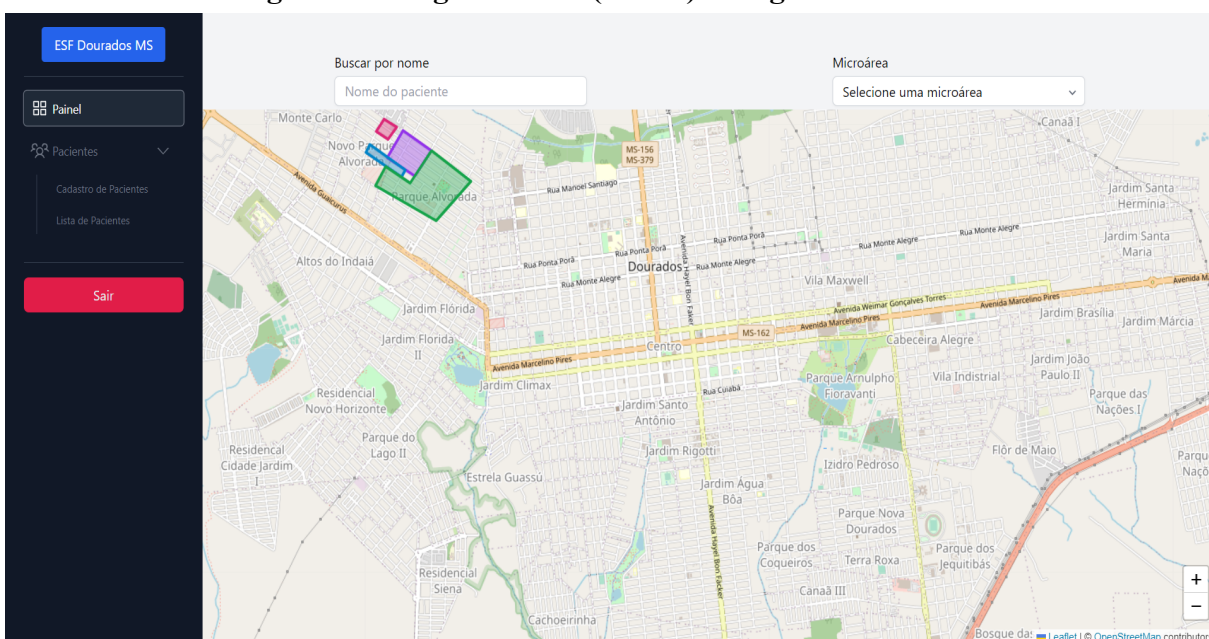
A principal diferença visual entre as páginas mostradas a seguir é os elementos que compõem o menu de navegação. Essa diferença pode ser vista nas Figuras 24 e 25. As demais telas desta seção são iguais visualmente, isto é, ignorando o menu de navegação.

Figura 24 - Página inicial (Painel) do Administrador ESF



Fonte: O autor.

Figura 25 - Página inicial (Painel) do Agente de Saúde



Fonte: O autor.

Figura 26 - Página de adicionar um novo paciente

ESF Dourados MS

Panel

Pacientes

Cadastro de Pacientes

Lista de Pacientes

Sair

Cadastro de Paciente

Informações Gerais

Nome Completo

CPF

Sexo

Data de Nascimento

Observação

Condição de Saúde

Endereço

Logradouro

Número

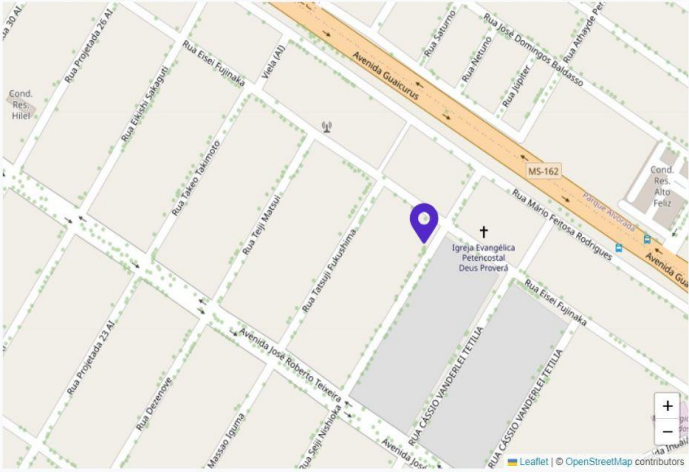
Bairro

Complemento

Geolocalização

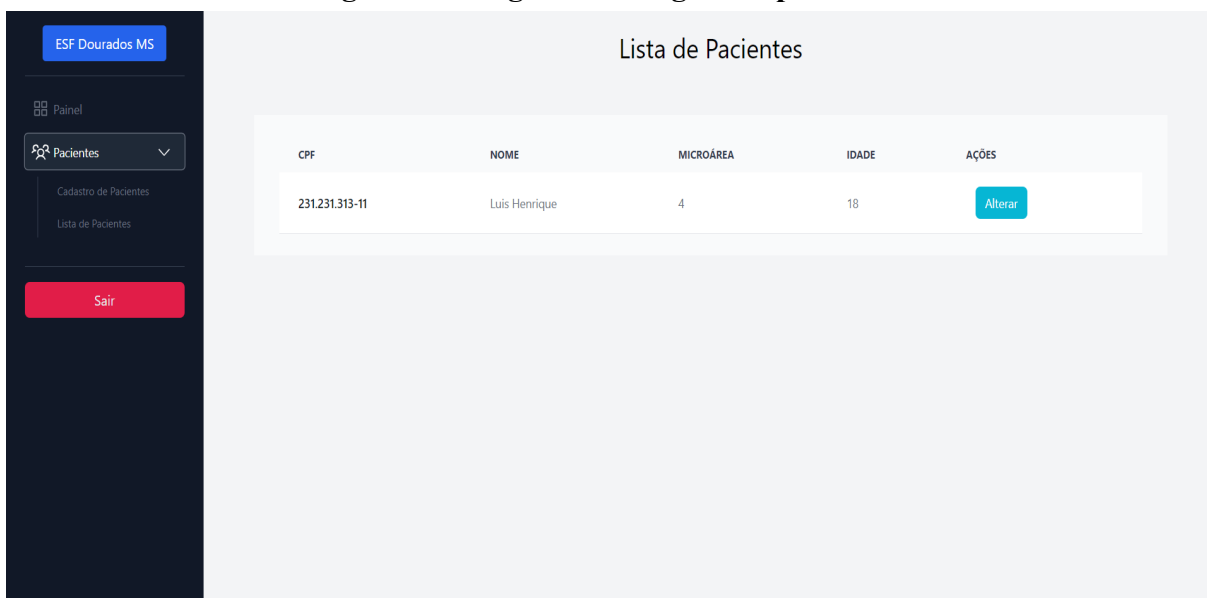
Microárea

Ajuste o marcador para ser mais precisa a localização



Cancelar Salvar

Fonte: O autor.

Figura 27 - Página de listagem de pacientes

ESF Dourados MS

Painel

Pacientes

Cadastro de Pacientes

Lista de Pacientes

Sair

Lista de Pacientes

CPF	NOME	MICROÁREA	IDADE	AÇÕES
231.231.313-11	Luis Henrique	4	18	Alterar

Fonte: O autor.

Figura 28 - Página de editar informações de um paciente

ESF Dourados MS

Panel

Pacientes

Cadastro de Pacientes

Lista de Pacientes

Sair

Editar Paciente - Luis Henrique

Informações Gerais

Nome Completo

CPF

Sexo Data de Nascimento

Masculino 14/02/2004

Observação

Condição de Saúde

Endereço

Logradouro Número

Rua Amael Pompeu Filho 122

Bairro

Complemento

Geolocalização

Microárea

Ajuste o marcador para ser mais precisa a localização

Leaflet | © OpenStreetMap contributors

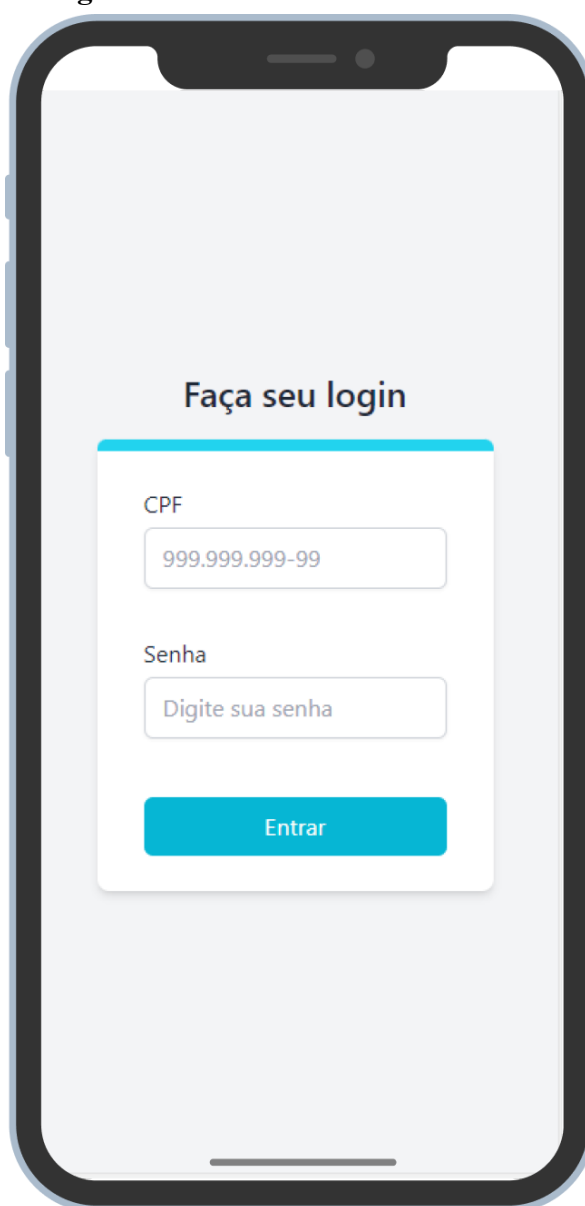
Cancelar
Salvar

Fonte: O autor.

A.4 - Páginas no responsivo

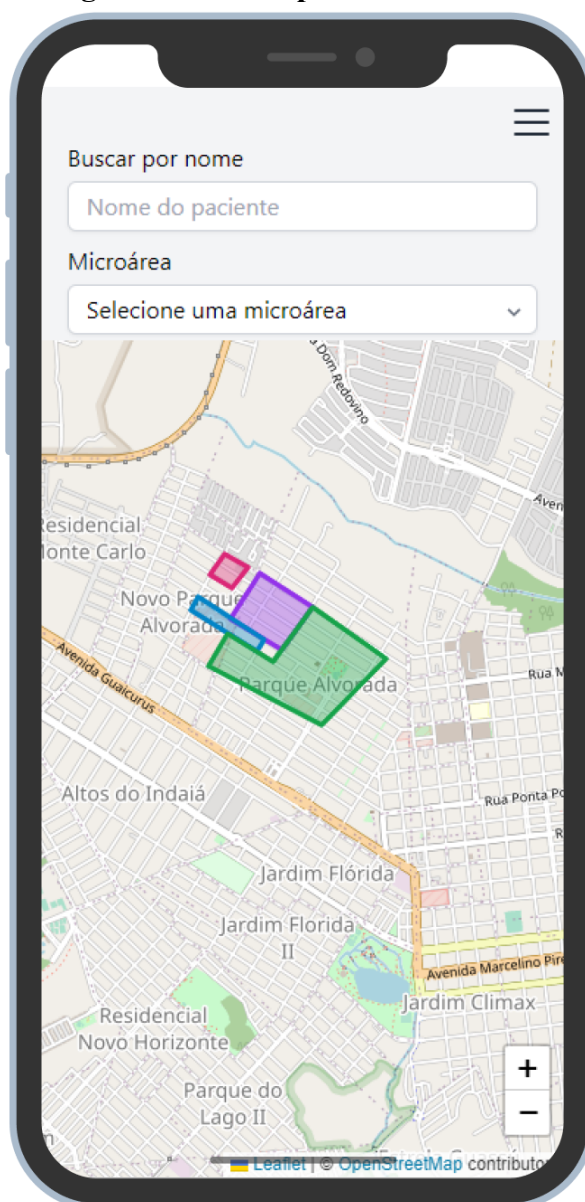
Neste tópico, podemos ver algumas das páginas do sistema em sua versão menor, o que é muito próximo da visão do aplicativo em um dispositivo móvel. As figuras a seguir mostram essa visão. Como podemos ver, a Figura 31 traz o menu de navegação para dispositivos móveis, assim pode-se navegar entre as telas da aplicação usando o mesmo. O botão com três traços no canto superior direito na Figura 30 serve para abrir o menu, o mesmo se apresenta na Figura 31 com a cor branca com a função de fechar o menu.

Figura 29 - Página de acesso ao sistema na visualização móvel



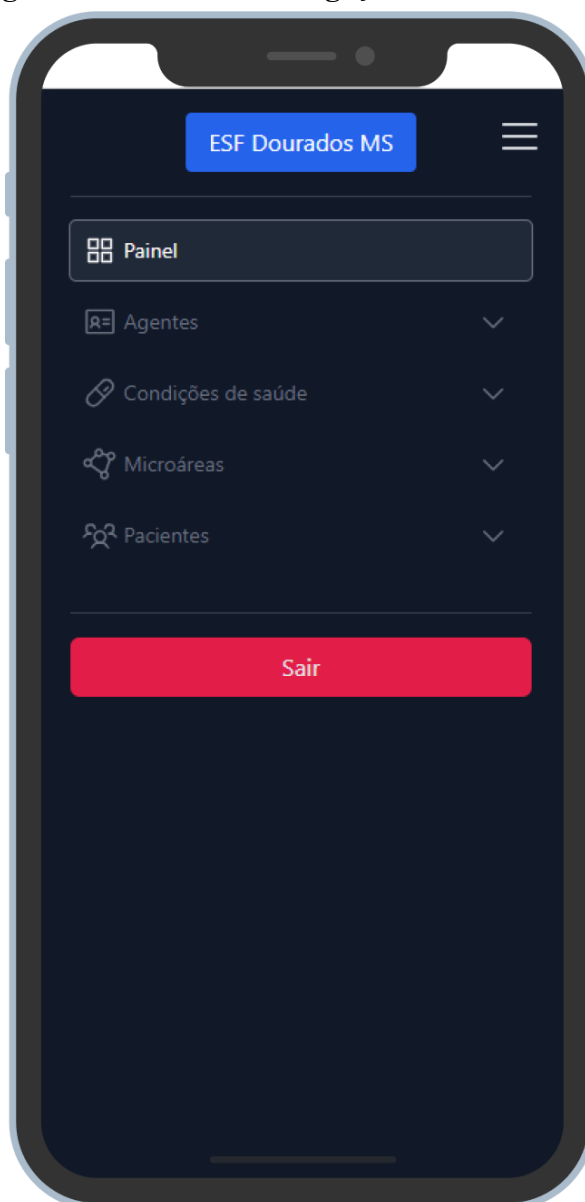
Fonte: O autor.

Figura 30 - Página inicial do aplicativo na visualização móvel



Fonte: O autor.

Figura 31 - Menu de navegação na versão móvel



Fonte: O autor.

APÊNDICE B - COMO EXECUTAR A APLICAÇÃO

Para o desenvolvimento e também para a execução local da aplicação é necessário de algumas ferramentas instaladas no sistema operacional. A principal ferramenta é o *Node.js*, ele é o que faz rodar a aplicação localmente, depois precisa-se do gerenciador de pacotes *Node.js* o *npm*, por fim é necessário ter o *Docker* e sua extensão *Docker Compose* para rodar o banco de dados.

Cada sistema operacional segue uma instalação, por isso é importante seguir a recomendação de instalação de cada ferramenta, os links oficiais para os sites destas ferramentas são descritos abaixo:

- *Node.js*: <https://nodejs.org/en/>
- *npm*: <https://www.npmjs.com/>
- *Docker*: <https://www.docker.com/>
- *Docker Compose*: <https://docs.docker.com/compose/install/>

Com as ferramentas já configuradas, basta seguir o passo a passo a seguir para executar a aplicação. Com o terminal aberto na pasta do projeto execute os seguintes comandos:

1. Abrir a pasta do servidor
`cd server`
2. Inicial o banco de dados com *Docker Compose*
`docker-compose up -d`
3. Instalar as dependências do servidor
`npm install`
4. Iniciar o servidor
`npm run dev`

A partir deste ponto o servidor (*back-end*) já estará rodando, próximo passo é executar a aplicação cliente (*front-end*), para isso é necessário abrir um novo terminal na pasta do projeto e executar os seguintes comandos:

1. Abrir a pasta *web*
`cd web`
2. Instalar as dependências do projeto *ReactJS*
`npm install`
3. Executar a aplicação *web*
`npm run dev`

Com ambas as aplicações rodando, é possível ver a mesma no navegador. Portanto o último passo é abrir o navegador e colocar a seguir *url*:

<http://localhost:5173/>

APÊNDICE C - CRIANDO UM USUÁRIO ADMINISTRADOR

Nesta seção é apresentado um tutorial sobre como criar um usuário administrador no sistema. Para criar esse tipo de usuário é necessário que a aplicação esteja rodando, sendo no mínimo o servidor e o banco de dados (Apêndice B).

Antes mesmo de executar o comando para criar o usuário, precisa-se configurar os dados. Portanto o primeiro passo a se fazer é abrir o arquivo `create_administrator.ts`, que está localizado na pasta `server/src/actors/`. Ao abrir o arquivo, é mostrado o código da Figura 32.

Figura 32 - Código de criação de usuário Administrador sem dados

```
1 import { PrismaClient } from '@prisma/client';
2 import { encryptedPassword } from '../utils/bcrypt';
3
4 // instância do cliente do prisma
5 const prisma = new PrismaClient();
6
7 async function main() {
8
9     // Preencher os dados do administrador
10    const data = {
11        name: "",
12        cpf: "",
13        password: "",
14    }
15
16    const password = encryptedPassword(data.password);
17
18    // criando administrador para o aplicativo
19    const newAdministrator = await prisma.user.create({
20        data: {
21            name: data.name,
22            cpf: data.cpf,
23            password,
24            role: 'ADMIN'
25        }
26    });
27
28    if (newAdministrator) {
29        console.log('[PRISMA] Administrador criado com sucesso!');
30        console.log(newAdministrator);
31    }
32 }
33
34 main()
35 .catch((e) => console.error(e))
36 .finally(async () => await prisma.$disconnect())
```

Fonte: O autor.

As linhas 11, 12 e 13 devem ser preenchidas com nome, CPF e senha respectivamente. Um exemplo é mostrado na Figura 33. É de extrema importância que os dados sejam preenchidos corretamente.

Figura 33 - Código de criação de usuário Administrador com dados

```
1 import { PrismaClient } from '@prisma/client';
2 import { encryptedPassword } from '../utils/bcrypt';
3
4 // instância do cliente do prisma
5 const prisma = new PrismaClient();
6
7 * async function main() {
8
9     // Preencher os dados do administrador
10 *   const data = {
11       name: "João Vitor de Olivera",
12       cpf: "38213411101", // Importante que seja exatamente 11 dígitos
13       password: "minhasenha123",
14     }
15
16     const password = encryptedPassword(data.password);
17
18     // criando administrador para o aplicativo
19 *   const newAdministrador = await prisma.user.create({
20 *     data: {
21       name: data.name,
22       cpf: data.cpf,
23       password,
24       role: 'ADMIN'
25     }
26   });
27
28 *   if (newAdministrador) {
29     console.log('[PRISMA] Administrador criado com sucesso!');
30     console.log(newAdministrador);
31   }
32 }
33
34 main()
35   .catch((e) => console.error(e))
36   .finally(async () => await prisma.$disconnect())
37
```

Fonte: O autor.

Com os dados já preenchidos, o próximo passo é abrir um terminal na pasta do projeto e executar os seguintes comandos:

1. Abrir a pasta do servidor
`cd server`
2. Executar o comando de criação de usuário administrador
`npm run create-adm`

Depois de executar esse último comando o usuário administrador está criado.