



UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS I
CENTRO DE CIÊNCIA E TECNOLOGIA
CURSO DE CIENCIA DA COMPUTAÇÃO

DANIELE VENTURA BATISTA

DESENVOLVIMENTO FRONT-END DE APLICATIVO PARA BUSCA DE
SERVIÇOS DE SAÚDE

CAMPINA GRANDE
2022

DANIELE VENTURA BATISTA

DESENVOLVIMENTO FRONT-END DE APLICATIVO PARA BUSCA DE
SERVIÇOS DE SAÚDE

Trabalho de Conclusão de Curso
apresentado ao curso de Ciência da
Computação da Universidade Estadual da
Paraíba, como requisito parcial à
obtenção do título de Graduado em
Ciência da Computação

Área de concentração: Usabilidade e
Fatores Humanos

Orientador: Prof. Dr. Daniel Scherer

CAMPINA GRANDE
2022

DANIELE VENTURA BATISTA

**DESENVOLVIMENTO FRONT-END DE APLICATIVO PARA
BUSCA DE SERVIÇOS DE SAÚDE**

Trabalho de Conclusão de Curso de Graduação em Ciência da Computação da Universidade Estadual da Paraíba, como requisito à obtenção do título de Bacharel em Ciência da Computação.

Aprovada em 06 de Setembro de 2022.



Prof. Dr. Daniel Scherer (DC - UEPB)
Orientador(a)



Profa. Dra. Kézia de Vasconcelos Oliveira Dantas (DC - UEPB)
Examinador(a)



Profa. Dra. Sabrina de Figueirêdo Souto (DC - UEPB)
Examinador(a)

É expressamente proibido a comercialização deste documento, tanto na forma impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que na reprodução figure a identificação do autor, título, instituição e ano do trabalho.

B333d Batista, Daniele Ventura.

Desenvolvimento front-end de aplicativo para busca de serviços de saúde [manuscrito] / Daniele Ventura Batista. - 2022.

80 p. : il. colorido.

Digitado.

Trabalho de Conclusão de Curso (Graduação em Computação) - Universidade Estadual da Paraíba, Centro de Ciências e Tecnologia, 2022.

"Orientação : Prof. Dr. Daniel Scherer, Departamento de Computação - CCT."

1. Mobile health. 2. Aplicativos móveis. 3. Unidades de Saúde. I. Título

21. ed. CDD 005.3

AGRADECIMENTOS

Inicialmente agradeço a Deus por sempre me mostrar em tempos difíceis algo bom para acreditar e seguir em frente.

Agradeço a meu pai, Francisco de Assis Batista, pelo exemplo de superação, por sempre comemorar minhas conquistas e ser a primeira pessoa a me fazer acreditar que sou capaz, a minha mãe, Maria do Socorro Lopes Ventura Batista, que me proporcionou as melhores lembranças da minha vida e pela mulher e mãe incrível que ela foi, às minhas irmãs, Kaline Ventura e Michele Ventura por fazerem meus dias mais felizes e sempre me apoiarem.

Agradeço ao professor Daniel Scherer pelas oportunidades que me proporcionou durante o curso, pela orientação e paciência no desenvolvimento deste trabalho.

Com carinho agradeço aos meus amigos do curso, Carlos Eduardo e João Vitor, que juntos, compartilhamos as dificuldades e os desafios para concluir essa trajetória.

RESUMO

A superlotação nos hospitais é um drama no Brasil, ao procurar atendimento, brasileiros de todas as regiões se deparam com filas, maior tempo de espera por atendimento e falta de estrutura para os cuidados em saúde. A procura por serviços de saúde envolve fatores que, dependendo de como estão ordenados, definirão a escolha pelo usuário. Assim, alguns fatores são a gravidade ou a urgência do problema/necessidade, a tecnologia disponível, as condições de acesso (distância da moradia, transporte, o tempo e o custo de deslocamento), a agilidade no atendimento. O *mobile health* é a união entre saúde e dispositivos móveis, tendo como exemplo mais comum os aplicativos móveis. Considerando esse contexto e as possibilidades que as tecnologias oferecem, como os aplicativos móveis, neste trabalho é apresentado o desenvolvimento front-end de uma aplicação para Android e iOS, com foco em busca de unidades de saúde. Foi definido as tecnologias Angular e Ionic para o desenvolvimento do aplicativo. Este trabalho apresenta os requisitos e a documentação do aplicativo de busca de serviços de saúde, bem como as tecnologias que serão utilizadas.

Palavras-Chave: Mobile health; Aplicativos móveis; Unidades de Saúde.

ABSTRACT

Overcrowding in hospitals is a drama in Brazil, when looking for medical care, Brazilians from all regions are faced with queues, longer waiting times for medical care, and a lack of structure for health care. The search for health services involves factors that, depending on how they are ordered, will define the choice made by the user. Thus, some factors are the seriousness or urgency of the problem/need, the technology available, the conditions of access (distance from the house, transportation, time, and cost of travel), the agility in service. Mobile health is the union between health and mobile devices, with mobile applications as the most common example. Considering this context and the possibilities that technologies offer, such as mobile applications, this work presents the front-end development of an application for Android and iOS, focusing on the search for health units. Angular and Ionic technologies were defined for application development. This work presents the requirements and documentation of the health services search application, as well as the technologies that will be used.

Keywords: Mobile health; Mobile Applications; Health Units.

LISTA DE FIGURAS

Figura 1 – Desenvolvimento.....	17
Figura 2 - Sistemas operacionais mais utilizados no Brasil.....	20
Figura 3 - Telas do aplicativo Conecte SUS Cidadão.....	25
Figura 4 - Telas do aplicativo e-saúdeSP.....	26
Figura 5 - Telas do aplicativo Dr.consulta.....	27
Figura 6 - Telas do aplicativo Sabin.....	28
Figura 7 - Telas do aplicativo Conecta Recife App.....	29
Figura 8 - Telas do aplicativo App SAL.....	30
Figura 9 - Telas do aplicativo Unimed Cliente.....	31
Figura 10 - Versão Android.....	35
Figura 11 - Versão iOS.....	36
Figura 12 - Diagrama de casos de uso.....	37
Figura 13 - Diagrama de casos de uso.....	37
Figura 14 - Diagrama de atividades para buscar unidades de saúde.....	38
Figura 15 - Diagrama de atividades para responder questionário de saúde.....	39
Figura 16 - Histórias de usuário – Login.....	40
Figura 17 - Histórias de usuário – Responder Questionário de Saúde.....	41
Figura 18 - Histórias de usuário – Pesquisar Unidades de Saúde.....	41
Figura 19 - Histórias de usuário – Pesquisar Unidades de Saúde.....	42
Figura 20 - Histórias de usuário – Telefonar para Unidade de Saúde.....	42
Figura 21 - Histórias de usuário – Solicitar Motorista de Aplicativo.....	43
Figura 22 - Histórias de usuário – Visualizar Mapa.....	43
Figura 23 - Tela de Login.....	44
Figura 24 - Tela de Unidades de Saúde.....	45
Figura 25 - Tela de permissão.....	46
Figura 26 – Mapa Fullscreen.....	47
Figura 27 – Traçar rota.....	48
Figura 28 – Tela Buscar Unidade de Saúde.....	49
Figura 29 – Tela Buscar Unidade de Saúde.....	50
Figura 30 – Tela Buscar Unidade de Saúde.....	51
Figura 31 – Tela Especialidades Médicas.....	52

Figura 32 – Tela Especialidades Médicas.....	53
Figura 33 – Tela Login do sistema web.....	55
Figura 34 – Listagem das Unidades de Saúde Cadastradas - Sistema web.....	55
Figura 35 – Cadastrar Unidade de Saúde – Sistema web.....	57
Figura 36 – Adicionar especialidade – Sistema web.....	58
Figura 37 – Adicionar horário de funcionamento – Sistema web.....	59
Figura 38 – Visualização da Unidade de Saúde – Sistema web.....	60
Figura 39 – Editar Unidade de Saúde – Sistema web.....	61
Figura 40 – Mapa Unidades de Saúde – Sistema web.....	62
Figura 41 – Diagrama de sequência - Buscar unidades de saúde.....	68

LISTA DE QUADROS

Quadro 1 - Comparativo dos trabalhos relacionados.....	32
Quadro 2 - Comparação dos frameworks React, Vue e Angular.....	34

LISTA DE LISTAGEM DE CÓDIGOS

Listagem 1 – Uso do plugin geolocation.....	63
Listagem 2 – Uso do plugin Call Number.....	64
Listagem 3 – Uso do plugin Google Maps.....	64
Listagem 4 – Uso da Directions API.....	65
Listagem 5 – Uso da API Places Autocomplete Service.....	66
Listagem 6 – Uso da API Geocoding Service.....	66
Listagem 7 – Uso do Deep Link Uber.....	67
Listagem 8 – JSON.....	68
Listagem 9 – Uso do @ngui/map.....	69
Listagem 10 – JSON.....	69

LISTA DE ABREVIATURAS E SIGLAS

UPA	Unidades de Pronto Atendimento
SPA	Single-Page Applications
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
SDK	Software Development Kit
UX	User Experience
SUS	Sistema Único de Saúde
APP	Aplicativo
CNES	Cadastro Nacional de Estabelecimentos de Saúde

SUMÁRIO

1	Introdução	13
1.1	Definição do problema.....	15
1.2	Objetivos	16
1.2.1	Objetivos específicos.....	16
2	Metodologia	17
3	Revisão da literatura	19
3.1	Definições e Conceitos.....	19
3.1.1	Aplicativos móveis.....	19
3.1.2	Sistema Operacional Android.....	19
3.1.3	Sistema Operacional iOS	21
3.1.4	Front-end.....	21
3.1.5	Framework	22
3.1.6	HTML e CSS	24
3.1.7	Typescript.....	24
3.2	Trabalhos relacionados	24
3.2.1	Conecte SUS Cidadão	24
3.2.2	E-saúdeSP – Plataforma da Saúde Paulistana	26
3.2.3	Dr.consulta	27
3.2.4	Sabin Medicina Diagnóstica	28
3.2.5	Conecta Recife App.....	29
3.2.6	App SAU – Aplicativo da Saúde de Uruguaiana.....	30
3.2.7	Unimed Cliente.....	31
3.3	Comparativo dos trabalhos relacionados	32
4	Resultados	33

4.1	Descrição do sistema	33
4.2	Definição dos Frameworks Web e Mobile	33
4.3	Diagramas de casos de uso	36
4.4	Diagramas de atividade.....	38
4.5	Histórias de usuário.....	40
4.5.1	Histórias de usuário – Login	40
4.5.2	Histórias de usuário – Questionário de Saúde	41
4.5.3	Histórias de usuário – Pesquisar Unidades de Saúde	41
4.5.4	História de usuário – Telefonar para Unidade de Saúde.....	42
4.5.5	Histórias de usuário – Solicitar Motorista de Aplicativo	43
4.5.6	Histórias de usuário – Visualizar Mapa	43
4.6	Apresentação da aplicação Mobile e Web	44
4.6.1	Aplicação Mobile	44
4.6.2	Aplicação Web	54
4.7	Implementação.....	63
5	Considerações finais.....	71
5.1	Trabalhos futuros	72
	Referências bibliográficas.....	73
	APÊNDICE A – HISTÓRIAS DE USUÁRIO.....	75
	APÊNDICE B – INTERFACES GRÁFICAS DO SISTEMA WEB.....	78

1 INTRODUÇÃO

A Tecnologia Móvel pode ser definida como a forma de acessar a internet e outros recursos computacionais por meio de dispositivos móveis, tais como, celulares, iPhone, iPad, notebooks, smartpads, dentre outros (ALCANTARA; VIEIRA, 2011). O surgimento da tecnologia móvel e seus avanços tornaram processos complexos mais simples e flexíveis. Os smartphones tornaram-se mais acessíveis e modernos ao longo do tempo e assim a tecnologia se torna cada vez mais presente no cotidiano do brasileiro, tanto na esfera pessoal, educacional e profissional.

Segundo Meirelles (2021), o Brasil em 2020 tinha 440 milhões de dispositivos digitais em uso, ou seja, mais de 2 dispositivos digitais por habitante. Os smartphones avançaram em termos de capacidade de processamento, acesso a dados via internet, e diversos sensores de posicionamento que podem ajudar no monitoramento do usuário, por exemplo, durante a ida ao hospital e atualmente os smartphones dominam vários usos, como nos bancos, saúde, compras e mídias sociais, diversão.

Além do entretenimento, a tecnologia móvel também pode trazer benefícios principalmente em áreas críticas como a saúde, sendo uma alternativa para melhorar o acesso da população à saúde, favorecendo o trabalho do profissional ou mesmo, privilegiando o autocuidado e qualidade de vida, e essa é uma das possibilidades por meio de aplicativos mobile health.

A *mobile health* (mHealth) é definida como práticas médicas e de saúde pública apoiadas por aparatos portáteis, como celulares, aparelhos de monitoramento dos pacientes, assistentes pessoais digitais e outros aparelhos sem fio. É possível utilizar os recursos desses dispositivos para criar, armazenar, recuperar e transmitir informações entre usuários, profissionais e pacientes. A mHealth conta principalmente com a utilidade do telefone celular de serviços de voz, serviços de mensagens curtas (SMS) e serviços de mensagens multimídia (MMS), mas também inclui aplicativos mais complexos, como sistemas de localização, tecnologia

Bluetooth e telecomunicações móveis de terceira e quarta geração (sistemas 3G e 4G) (WA et al., 2020).

A superlotação de unidades de saúde e as filas para o tratamento hospitalar são consideradas um problema mundial, ao procurar atendimento, brasileiros de todas as regiões se deparam com filas, espera de horas e falta de estrutura para os cuidados em saúde.

Há alguns fatores que podem ser as causas da superlotação nos hospitais, levando ao atraso no diagnóstico e aumentando os riscos para pacientes, acompanhantes e funcionários. Esses fatores, de acordo com Morsch (2019) podem ser falta de disponibilidade de leitos, falta de pessoal qualificado, aumento do tempo de permanência no Serviço de Emergência Hospitalar, entre outros.

Com a dificuldade na marcação de consultas ou sem conhecimento das unidades de saúde especializadas, muitos cidadãos acabam se dirigindo aos prontos-socorros de grandes hospitais, pois sabem que ali haverá médicos, clínicos, cirurgiões e especialistas de plantão, onde ficam expostos a agentes de contaminação e precisam aguardar por horas para passar pela triagem. Apenas em casos de emergências e urgências os pacientes devem procurar serviços de atendimento imediato, como prontos-socorros. Esse é um dos fatores que provocam o aumento no tempo de espera pelo primeiro atendimento, além de elevar o risco de agravamento às condições de saúde dos pacientes.

De acordo com De (2017), a população prefere procurar unidades de urgência e emergência, pronto-socorros e unidades de pronto atendimento (UPAs) ou até mesmo hospitais, seja por acreditarem que é mais rápido o acesso ao tratamento para seus problemas de saúde, levando a necessidades não atendidas e o próprio esgotamento do sistema neste nível de atendimento, insatisfação dos usuários e desmotivação dos profissionais.

Segundo Uzelli (2013), a busca direta pelo nível terciário (procedimentos clínicos e/ou cirúrgico de média e grande complexidade de todas as especialidades) acontece por vários motivos: dificuldade no acesso aos níveis primários (consultas médicas clínicas básicas e de baixa complexidade), serviço ineficaz, qualidade ruim do atendimento e cultura da população em ser atendida em hospitais.

Com essa preferência da população em buscar atendimentos em unidades de emergência e UPAs, se inicia a superlotação desnecessária, onde problemas muitos simples e de baixa complexidade poderiam ser atendidos em unidades básicas ou unidades especializadas.

A busca por atendimentos não urgentes em serviços que deveriam ser dedicados aos atendimentos de casos mais graves é enfrentada tanto em serviços públicos de saúde e hospitais privados, que por vezes provocam dificuldades de gerenciamento dos serviços, insatisfação de seus pacientes, familiares, médicos e parceiros ou seguradoras de planos de saúde (DE, 2017).

Um problema enfrentado pela população também é à distância da sua moradia até as unidades de saúde. No estudo “avaliação das dificuldades enfrentadas pelo paciente para realização de uma consulta médica de nível terciário” feito por Uzelli (2013), mostra que a distância do local de atendimento se torna um dos fatores que dificultam o acesso da população, principalmente a de renda mais baixa. O principal meio de transporte mais utilizado pela população de casa para as unidades de saúde é o público, com duração de uma a duas horas, sendo esse motivo uma das principais barreiras para o atendimento.

Diante da necessidade de melhorar a superlotação das unidades de saúde e o desenvolvimento crescente de aplicativos que propõem soluções rápidas e práticas para a saúde móvel, este trabalho tem como objetivo o desenvolvimento front-end de um aplicativo de direcionamento do paciente ao tipo adequado de assistência necessária.

1.1 Definição do problema

No Brasil, existe uma cultura de recorrer ao pronto-socorro para solucionar problemas em geral. Idas desnecessárias a emergência é comum, assim que um sintoma é identificado, o cidadão recorre à emergência ou pronto-socorro para ser atendido ou pedir orientação. Esse comportamento gera um aumento nas filas e um custo que poderia ser evitado.

Se expor a um ambiente assim quando o cidadão está com uma doença simples, como resfriado, gripe comum, dor nas costas, de cabeça ou garganta, tende a fazer mais mal para a saúde do que bem.

Mais da metade dos problemas que os pacientes levam ao pronto-socorro não precisam ser atendidos em um hospital (SAÚDE, 2021). Muitos casos de saúde vistos nos prontos-socorros são problemas comuns, facilmente diagnosticáveis e tratáveis por um custo muito menor por um clínico geral ou um especialista.

Além do custo financeiro de uma visita ao médico, há também o tempo gasto em deslocamentos às unidades de saúde, que podem ser longas viagens. A população às vezes não tem conhecimento dos serviços de saúde perto da sua residência, além de não conhecer as especialidades médicas que pode estar precisando e assim continua a cultura de recorrer às emergências.

1.2 Objetivos

O objetivo geral do presente trabalho consiste em desenvolver o front-end de uma aplicação para os sistemas operacionais Android e iOS, com foco em auxiliar o cidadão na busca por unidades de saúde próximas a sua localização.

O aplicativo móvel tem o objetivo de direcionar o paciente a uma unidade de saúde próxima a sua localização que atenda a sua necessidade atual, com intuito de deixar a cultura de preferência de prontos-socorros e emergências, evitando assim a superlotação e incentivando a busca às unidades e/ou especialidades mais adequada para o seu correto tratamento.

1.2.1 Objetivos específicos

Os objetivos específicos do presente trabalho são:

- Aplicar os conhecimentos adquiridos ao longo do curso.
- Desenvolvimento do Front-end de uma aplicação *mobile*, para os sistemas operacionais Android e iOS.
- Permitir o cidadão buscar unidades de saúde próximas da sua localização.
- Desenvolver um questionário de saúde para auxiliar na identificação do serviço de saúde mais adequado para o usuário.

2 METODOLOGIA

Neste capítulo serão apresentados os métodos adotados para o desenvolvimento do trabalho, apresentando a sequência das principais atividades realizadas. A Figura 1 apresenta a cronologia dos processos para implementação da aplicação. Primeiramente foram definidas as tecnologias que iram ser utilizadas no desenvolvimento do aplicativo. Em seguida, o processo de levantamento dos requisitos e a modelagem dos diagramas, para melhor visualização do sistema. Por fim, o desenvolvimento do aplicativo.

Figura 1 – Desenvolvimento



Fonte: Próprio autor

a) Definição dos Frameworks Web e Mobile

Inicialmente foi necessário fazer um levantamento das tecnologias Front-end, com o objetivo de escolher a tecnologia adequada para o desenvolvimento da aplicação. Em uma breve pesquisa sobre os frameworks web mais usados em 2021, de acordo com Stack Overflow 2021 (OVERFLOW, 2021), os 5 principais foram React.js, jQuery, Express, Angular, Vue.js. Entre esses 5 frameworks web mais utilizados em 2021, decidimos focar em buscar mais informações entre os frameworks Angular, React e Vue, para afunilar a busca e conseguir escolher entre um deles para ser utilizado na implementação do aplicativo.

Após definir o framework web, o próximo passo foi verificar qual framework para desenvolvimento mobile seria interessante utilizar.

b) Levantamento de requisitos

O levantamento de requisitos iniciou com estudo e observações das necessidades principais dos usuários alvo do aplicativo, percebendo quais funcionalidades seriam úteis em um aplicativo mHealth, para que os usuários não

tivessem dificuldades em utilizar a aplicação. Pesquisa acerca de aplicativos com finalidades semelhantes, focando sempre nas aplicações que possuíam localização de unidades de saúde. Esses dados geraram uma base de conhecimento necessária para a proposta do aplicativo, a partir do qual foi definido o escopo do sistema, bem como os requisitos que deveriam estar presentes nele.

c) Modelagem

Na fase de modelagem ocorreu a criação dos casos de uso, a elaboração dos diagramas de atividade, diagramas de sequência e criação das histórias do usuário.

d) Implementação

Na fase de implementação, foi desenvolvido o front-end do aplicativo. O desenvolvimento foi baseado a partir dos requisitos e da modelagem concebidos na etapa 2 e 3, respectivamente.

A fase de implementação foi dividida em duas etapas. A primeira etapa foi focada no desenvolvimento das telas do aplicativo versão *mobile*, para busca das unidades de saúde próximas ao usuário, focando sempre em um paciente que estivesse mal de saúde e necessitasse utilizar a aplicação, essas interfaces têm o objetivo de serem de fácil utilização.

A segunda etapa foi focada na implementação do sistema web, para cadastro das unidades de saúde que compõe a aplicação *mobile*.

3 REVISÃO DA LITERATURA

Neste capítulo é abordada toda a revisão da literatura do trabalho, ou seja, todo o fundamento que foi tomado como base para a realização desse projeto. Ele está dividido em três seções principais: Definições e Conceitos (seção 3.1), Trabalhos Relacionados (seção 3.2) e Comparativo dos trabalhos relacionados (seção 3.3).

3.1 Definições e Conceitos

Esta seção apresenta as tecnologias que serão utilizadas para o desenvolvimento do aplicativo, assim como conceitos que são fundamentais para a compreensão de alguns termos neste trabalho.

3.1.1 Aplicativos móveis

Aplicativo móvel é um software desenvolvido para ser instalado em dispositivos móveis, como celulares e tablets, ou no computador e em smartTVs. Uma parte dos aplicativos disponíveis são gratuitos, enquanto outros são pagos. Estes aplicativos são pré-instalados ou vêm direto da fábrica, baixados pelos clientes de várias plataformas de distribuição de software móvel.

Os aplicativos são desenvolvidos com o propósito de facilitar a vida das pessoas oferecendo um serviço simples que pode ser feito com apenas alguns cliques. Eles desempenham diversas funções: mensageiros online, streaming, gerenciadores, editores de fotos e vídeo, gerenciamento de dados, realização de compras, acesso a redes sociais, dentre outros.

3.1.2 Sistema Operacional Android

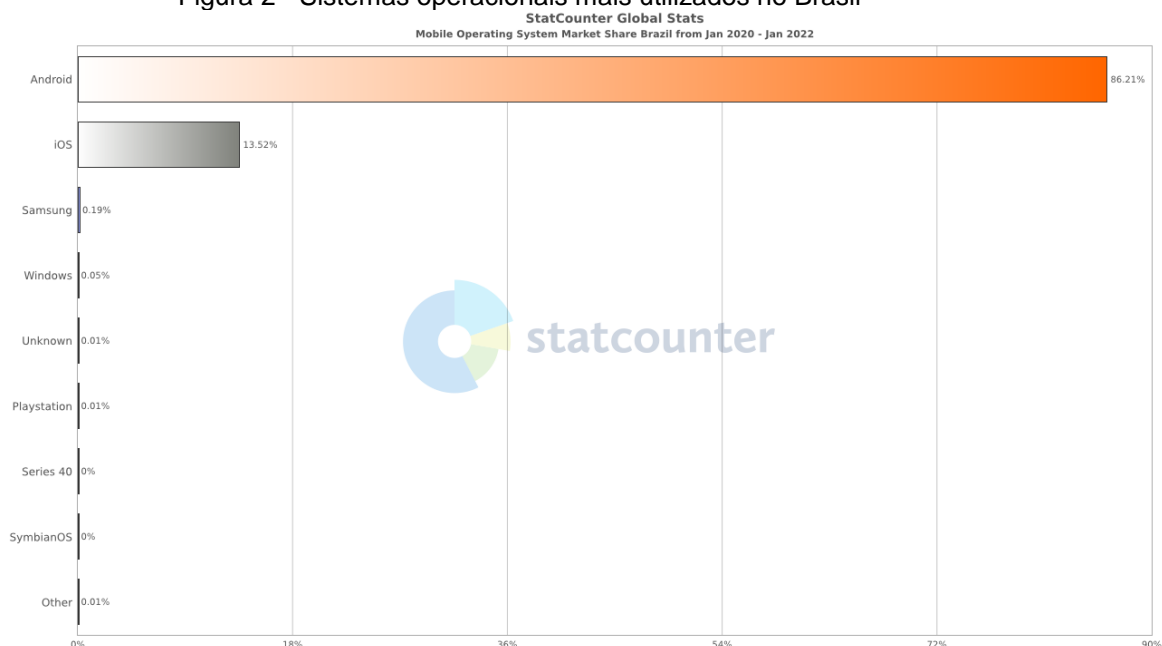
O Android é uma plataforma aberta voltada para dispositivos móveis desenvolvida pela Google e atualmente é mantida pela Open Handset Alliance (OHA). Todas as aplicações desenvolvidas para essa plataforma foram criadas com a linguagem Java, o que facilita muitos programadores com conhecimentos em Java (ou de outras linguagens próximas de Java como C++ e C#) a desenvolver aplicações para o Android.

Atualmente, os dispositivos Android incluem smartphones, tablets, e-readers, robôs, motores a jato, satélites da NASA, consoles de jogos, geladeiras, televisões, câmeras, equipamentos voltados à saúde, relógios inteligentes (smartwatches), sistemas automotivos de “infotainment” de bordo (para controlar rádio, GPS, ligações telefônicas, termostato, etc.) e muitos outros (MARCUS; OLIVEIRA, 2019).

O sistema operacional é de código-fonte aberto e gratuito. Isso permite ver o código-fonte do Android e como seus recursos são implementados (DEITEL; DEITEL; DEITEL, 2015).

O Android funciona basicamente como qualquer outro sistema operacional para dispositivos móveis, oferecendo funções de gerenciar os processos dos aplicativos e manipular os recursos de hardware presentes no dispositivo. Um dos pontos interessantes é que o Android permite que o usuário associe sua conta do Google ao dispositivo e possa ter acesso aos seus dados nos aplicativos padrões do sistema. Ele conta com o Google Play Store (GOOGLE PLAY, 2022), que é uma loja virtual onde o usuário pode baixar e instalar outros aplicativos desenvolvidos por terceiros. De acordo com Statcounter (2022), o Android é o sistema operacional mais utilizado atualmente no Brasil em aparelhos móveis, como mostrado na Figura 2, e iOS o segundo mais utilizado.

Figura 2 - Sistemas operacionais mais utilizados no Brasil



Fonte: Statcounter

3.1.3 Sistema Operacional iOS

O sistema operacional iOS foi lançado originalmente em 2007 com o primeiro iPhone, por Steve Jobs . O iOS é a plataforma utilizada em produtos Apple como o iPhone, iPad e iPod. O idioma principal oferecido pela Apple para o iOS desenvolvimento é Objective-C.

Ele dispõe de recursos para atender às necessidades específicas de um dispositivo móvel da Apple, considerando a entrada e a saída de dados armazenados.

O iOS é um software e já vem da fábrica com aplicativos nativos, como aplicativo de Música, Podcasts, TV, Mapas, Notas, FaceTime, iBook, iTunes, Contatos, Livros, Safari, assistente de voz “Siri” e serviço de nuvem “iCloud”. O usuário tem a opção de baixar novos aplicativos para o sistema através da App Store (STORE, 2022), uma loja virtual de aplicativos exclusivos da Apple, que contém tanto aplicativos padrões quanto aplicativos desenvolvidos por terceiros.

3.1.4 Front-end

Front-end é o responsável por toda a estrutura, design, conteúdo, comportamento, desempenho e capacidade de resposta de um site ou aplicação, ou seja, tudo o que é apresentado aos usuários para interação. O front-end consiste em tudo que realiza uma interação direta com o usuário, toda a programação da interface gráfica de um sistema.

O front-end serve para que os usuários possam interagir com uma aplicação, em outras palavras, para construir a interface de utilização de um site, aplicativo ou software. O desenvolvimento front-end é interpretado pelos navegadores ou sistemas operacionais e os ajuda a transmitir aqueles elementos a quem deseja interagir com eles.

Há diversas linguagens que atuam com formatos visuais, sendo as mais conhecidas HTML, CSS e JavaScript, que possuem diferentes funcionalidades com o mesmo objetivo quando mencionamos o front-end.

3.1.5 Framework

De acordo com Neves e Junior (2020), um framework é uma aplicação que oferece um conjunto de ferramentas que torna o desenvolvimento de aplicações mais produtivo e flexível.

O objetivo de um framework, segundo Araújo (2015), é aumentar a produtividade e minimizar os esforços gerados no desenvolvimento. Isso possibilita que os desenvolvedores se preocupem com outras questões mais importantes.

Frameworks conseguem resolver problemas recorrentes com uma abordagem genérica. Com isso, o desenvolvedor não precisa ficar reescrevendo softwares, podendo focar seus esforços em resolver os problemas em si. Eles também ajudam no desenvolvimento rápido e seguro de aplicações. Suas funcionalidades permitem atender qualquer necessidade do profissional, desde que o framework selecionado seja direcionado a função desejada.

Os frameworks de desenvolvimento *mobile* multiplataforma são populares por permitirem o desenvolvimento híbrido, possibilitando um único código ser compilado em vários sistemas operacionais móveis, como Android, Windows Phone, iOS e BlackBerry OS, sem a necessidade de reescrevê-lo, tornando estas aplicações mais práticas e econômicas para o mercado.

3.1.5.1 Framework Angular

Angular (ANGULAR, 2022), é um framework front-end criado por Miško Hevery e Adam Abron em 2009 e é um framework JavaScript open source (código aberto), *client-side* (do lado do cliente) que promove uma alta produtividade na experiência do desenvolvimento Web. A primeira versão foi lançada em 2010 com o nome AngularJS, mas foi completamente reescrita e em 2016 passou a ser chamada Angular 2.

É utilizado para a criação de *Single-Page Applications* (SPA), ou seja, uma única página web, com o objetivo de fornecer a experiência ao usuário parecida com a de um aplicativo de desktop, onde o código é carregado na página única, de forma dinâmica. Angular vem com um conjunto de bibliotecas poderosas que se pode importar, possibilitando construir aplicações com uma qualidade e produtividade surpreendente.

A arquitetura do Angular é baseada em componentes, que funcionam como blocos de construção adicionados conforme a necessidade de utilização. Isso torna o desenvolvimento da aplicação mais flexível, pois facilita a reutilização de códigos sem a necessidade de escrevê-los novamente.

3.1.5.2 Framework Ionic

O framework Ionic é um SDK criado por Max Lynch, Ben Sperry e Adam Bradley, da Drifty Co. em 2013. A versão original foi lançada em 2013 e construída sobre AngularJS e Apache Cordova.

O Ionic é um kit de ferramentas de interface do usuário para criar aplicativos móveis e de desktop de alta qualidade e desempenho usando tecnologias da Web, como HTML, CSS e JavaScript, com integrações para estruturas populares como Angular, React e Vue (IONIC, 2022a).

Ele se concentra na interação de UX e interface do usuário do front-end de um aplicativo - controles de interface do usuário, interações, gestos, animações. É fácil de aprender e se integra a outras bibliotecas ou estruturas, como Angular, React ou Vue. Alternativamente, ele pode ser usado de forma independente sem qualquer estrutura de front-end usando um script simples.

Ionic é um framework para desenvolvimento híbrido, tendo como principal vantagem o código único para várias plataformas, como Android, iOS e Windows Phone, garantindo maior facilidade no desenvolvimento, custos mais baixos e tornando a manutenção mais simples para todas as plataformas.

3.1.5.3 Apache Cordova

Cordova é uma estrutura de desenvolvimento móvel de código aberto. Ele permite usar tecnologias padrão da Web, como HTML5, CSS3 e JavaScript para desenvolvimento multiplataforma, evitando a linguagem de desenvolvimento nativa de cada plataforma móvel (CORDOVA, 2022).

Como o Ionic, é um framework desenvolvido para a criação de aplicativos móveis híbridos. O Ionic mescla o funcionamento do Apache Cordova e do Angular para facilitar o desenvolvimento, de modo que é possível utilizar todas as ferramentas do Angular para desenvolver o aplicativo, sendo o Apache Cordova

responsável pela geração do aplicativo e também por funcionalidades específicas de dispositivos móveis, como o acesso a câmera, bluetooth, geocalização, etc.

3.1.6 HTML e CSS

HTML significa HiperText Markup Language, ou seja, Linguagem de Marcação de Hipertexto. O HTML é o componente básico da web, ele permite inserir o conteúdo e estabelecer a estrutura básica de um website. Portanto, ele serve para dar significado e organizar as informações de uma página na web. Sem isso, o navegador não saberia exibir textos como elementos ou carregar imagens e outros conteúdos.

CSS é a sigla para Cascading Style Sheets, ou seja, Folhas de Estilo em Cascatas. É uma maneira de dar estilo ao código criado por linguagens como HTML, XML ou XHTML, por exemplo. De forma prática, ela funciona como uma camada de personalização ao conteúdo visível.

O CSS é um código em que você pode fazer alterações rápidas de layout, como definição de cores e fontes, por exemplo. Essa camada proporciona não apenas a facilidade de personalização, como também ajuda a diminuir a repetição de conteúdo na estrutura do código.

3.1.7 Typescript

Typescript é uma linguagem open-source desenvolvida pela Microsoft. É um superconjunto de JavaScript e foi criada com a finalidade de facilitar o desenvolvimento de grandes aplicações. Seu código é transpilado para JavaScript. O código TypeScript é convertido em JavaScript, que é executado em qualquer lugar que o JavaScript seja executado : em um navegador, em Node.js ou Deno e em seus aplicativos (TYPESCRIPT, 2022).

3.2 Trabalhos relacionados

Existem diversos aplicativos móveis que possui funcionalidade de buscar por localizações de unidades de saúde, como:

3.2.1 Conecte SUS Cidadão

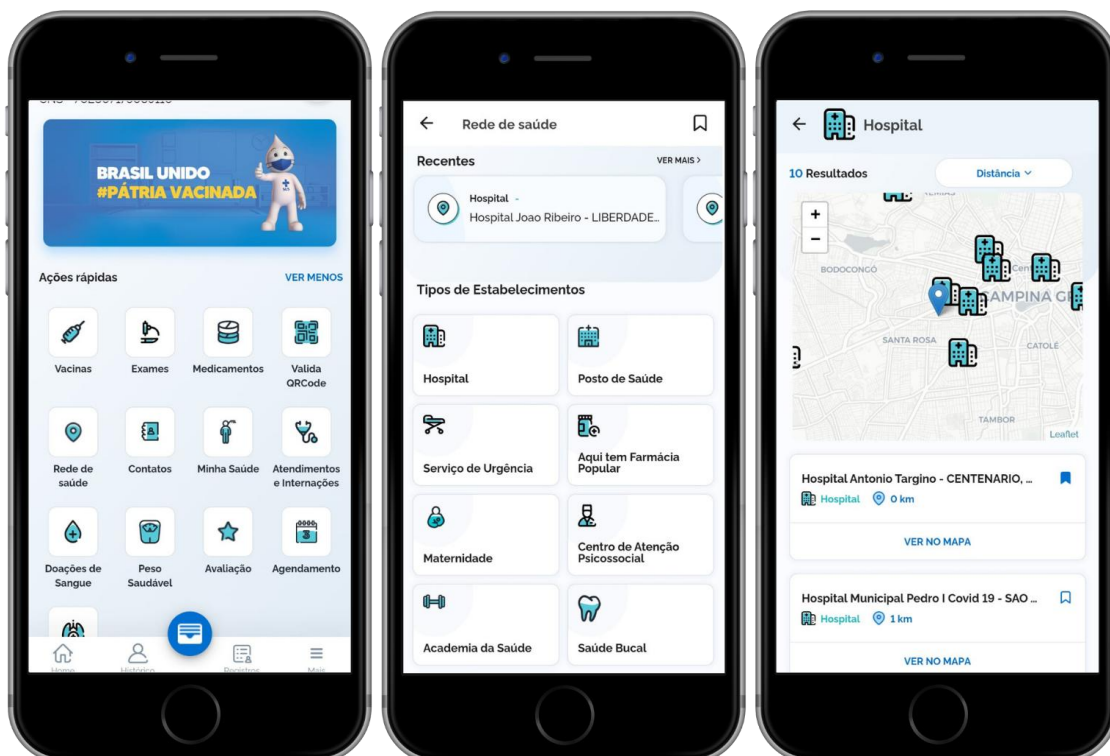
O Conecte SUS Cidadão (CONNECTE SUS, 2021) é o aplicativo oficial do Ministério da Saúde, a porta de acesso aos serviços do Sistema Único de Saúde

(SUS) de forma digital, permitindo ao cidadão acompanhar na palma de sua mão o seu histórico clínico. O aplicativo está disponível para dispositivos Android e iOS, quanto para versão web.

O aplicativo Conecte SUS mostra as informações gerais do cidadão, como carteira de vacinação digital, certificado nacional de vacinação COVID-19, resultados de exames laboratoriais de COVID-19, medicamentos dispensados pelo programa "Farmácia Popular", além dos registros de doações de sangue e acompanhamento da posição na fila de transplantes. A área Serviços de Saúde permite ao cidadão identificar estabelecimentos de saúde próximos à sua localização ou pelo tipo, além de permitir adicionar aos favoritos os mais relevantes.

A página Atendimentos apresenta a lista de procedimentos realizados pelo cidadão nos estabelecimentos de saúde, como, vacinas, exames, medicamentos e atendimentos nos estabelecimentos da atenção básica. O aplicativo permite que sejam avaliados ou, ainda, que o Ministério da Saúde seja informado quando o atendimento não for realizado.

Figura 3 - Telas do aplicativo Conecte SUS Cidadão



Fonte: Conecte SUS Cidadão

3.2.2 E-saúdeSP – Plataforma da Saúde Paulistana

A Plataforma de Saúde Paulistana (E-SAUDESP, 2022) é uma criação da Secretaria Municipal de Saúde de São Paulo para facilitar o acesso e o registro das informações de saúde do cidadão paulistano. Concebida para integrar e disponibilizar todas as informações de saúde envolvidas no atendimento SUS do âmbito municipal de São Paulo.

O App possibilita que o cidadão registre diariamente seus dados de saúde, permitindo o monitoramento de informações de importância, como aferições de pressão e glicemia capilar, registro de peso, alergias e medicações de uso contínuo.

Além disso, o e-saúdeSP reúne grande quantidade de informações úteis da saúde, como orientações oficiais da Secretaria Municipal de Saúde, armazenamento do histórico clínico do paciente, e georreferenciamento para localização de unidades de saúde. O e-saúdeSP está disponível para dispositivos Android e iOS e em versão web.

Figura 4 - Telas do aplicativo e-saúdeSP



Fonte: e-saúdeSP

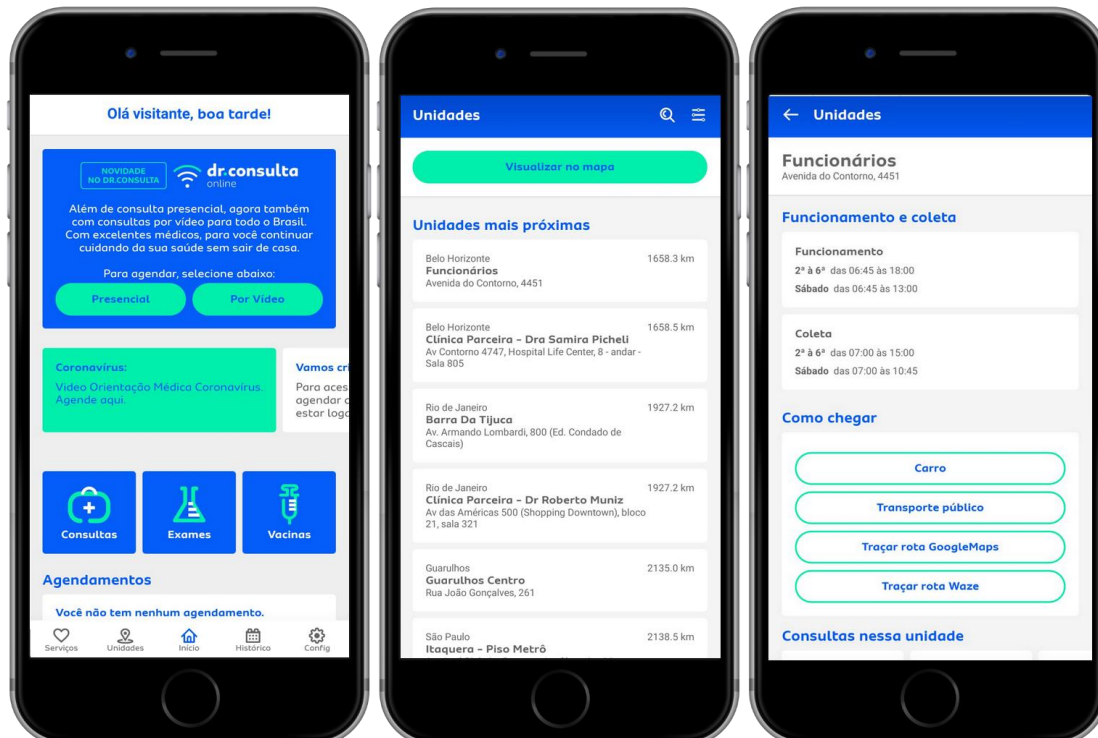
3.2.3 Dr.consulta

O dr.consulta (DR.CONSULTA, 2021) é o aplicativo de uma rede de centros médicos que permite a marcação de exames e consultas. O usuário tem acesso a maiores informações sobre os profissionais, como número do CRM, formação e avaliações de outros pacientes. Caso tenha alguma urgência em agendar alguma consulta, é possível fazê-la no mesmo dia.

O usuário pode agendar consultas presenciais e online, agendar cirurgias, agendar exames, conferir seus próximos agendamentos, acessar o histórico de visitas, resultados de exames e encaminhamentos de notas fiscais e recibos pelo celular, georreferenciamento para localização de unidades de saúde, clicando nelas aparece às informações de cada uma, desde horário de funcionamento, direcionamento de como chegar até ela e serviços disponíveis.

O dr.consulta está disponível para dispositivos Android e iOS e em versão web.

Figura 5 - Telas do aplicativo Dr.consulta



Fonte: Dr.consulta

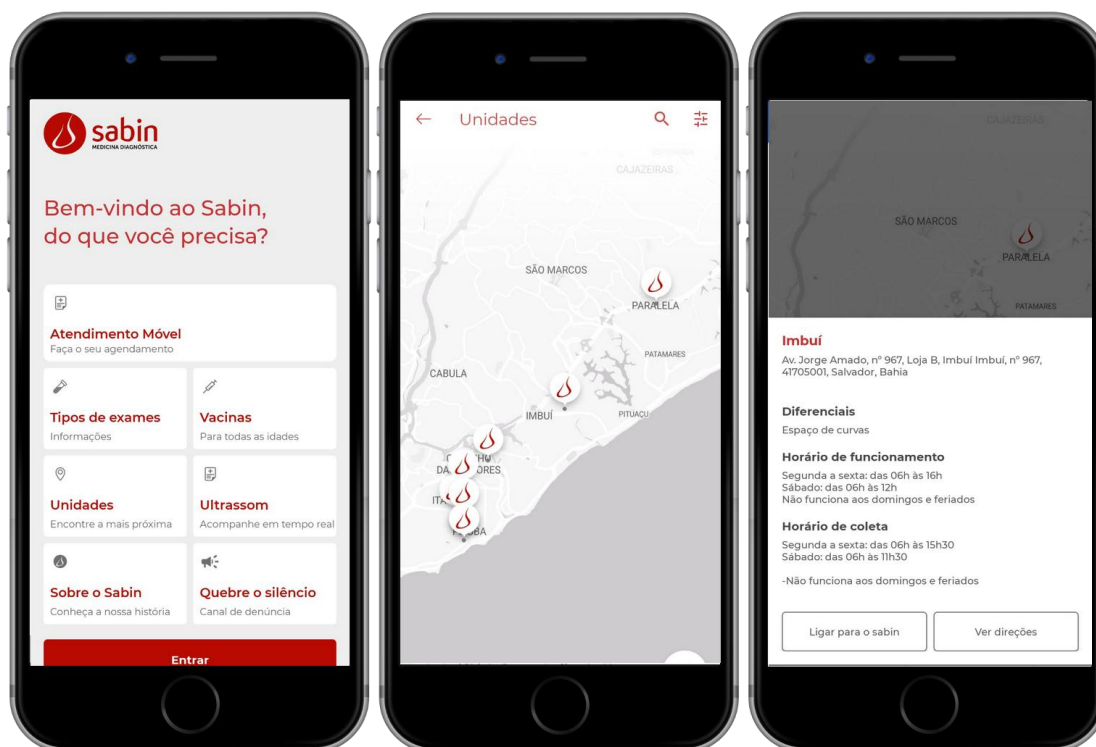
3.2.4 Sabin Medicina Diagnóstica

O Sabin Medicina Diagnóstica (SABIN, 2021) é um aplicativo criado pelo Laboratório Sabin, o objetivo da solução é proporcionar mais praticidade e mobilidade para médicos e pacientes, que podem acessar o conteúdo dos exames, economizando tempo.

Cada perfil do aplicativo dispõe de funcionalidades distintas. O cliente tem acesso aos laudos e históricos de seus resultados, informações sobre os exames, localizador de unidades integrado com o Google Maps, notícias de saúde e calendários de vacinas e de datas sazonais. Já o médico, pode acompanhar os resultados de exames de seus pacientes e saber as últimas novidades e informações sobre exames, possui acesso à pesquisa da Classificação Internacional de Doenças (CID) e aos estudos científicos.

O Sabin está disponível para dispositivos Android e iOS e em versão web.

Figura 6 - Telas do aplicativo Sabin



Fonte: Sabin

3.2.5 Conecta Recife App

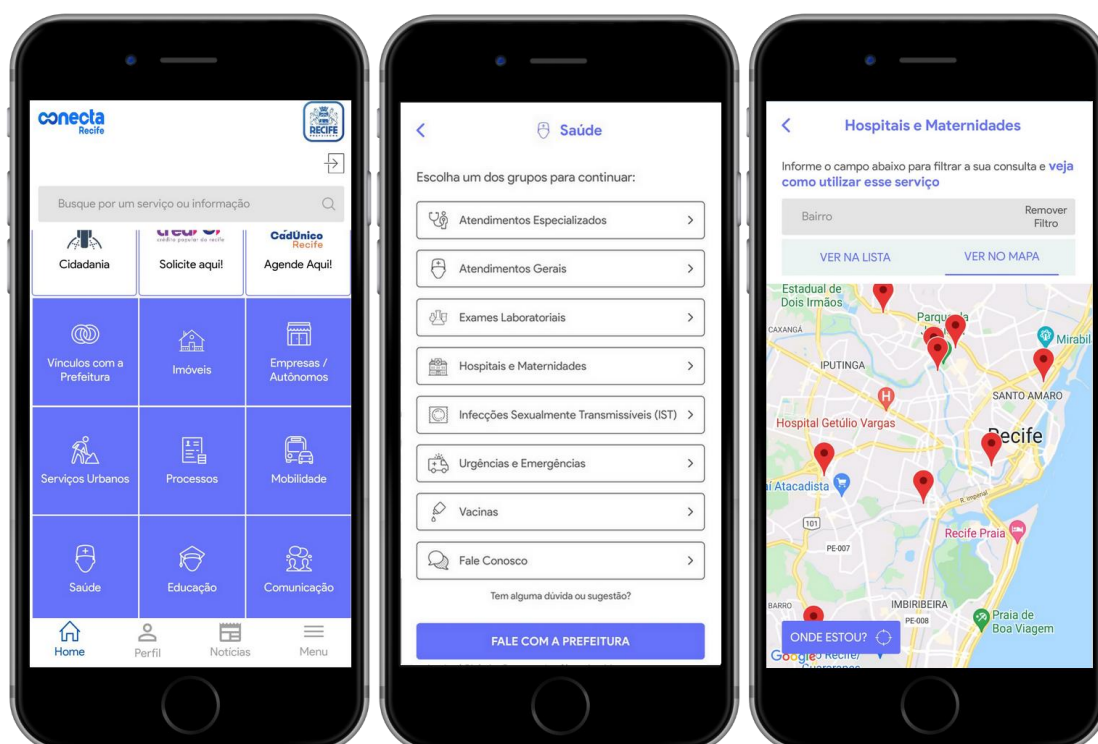
O Conecta Recife App (CONNECTA RECIFE, 2021) é um aplicativo que permite o cidadão ter acesso a mais de 120 serviços e informações da administração da cidade, de forma simples e rápida.

Através dele o cidadão pode solicitar e consultar vários serviços da Prefeitura do Recife, a exemplo das Informações Tributárias, Processos Administrativos, Educação, Saúde, Trânsito e Ciclovias, entre outros. Também será possível solicitar atendimentos públicos municipais, como geração de documentos, execução de serviços públicos como capinação.

A seção saúde possibilita o agendamento da vacina contra COVID-19, localização de serviços de saúde, como hospitais, maternidades, urgências, emergências, entre outros.

O Conecta Recife App está disponível para dispositivos Android e iOS.

Figura 7 - Telas do aplicativo Conecta Recife App



Fonte: Conecta Recife App

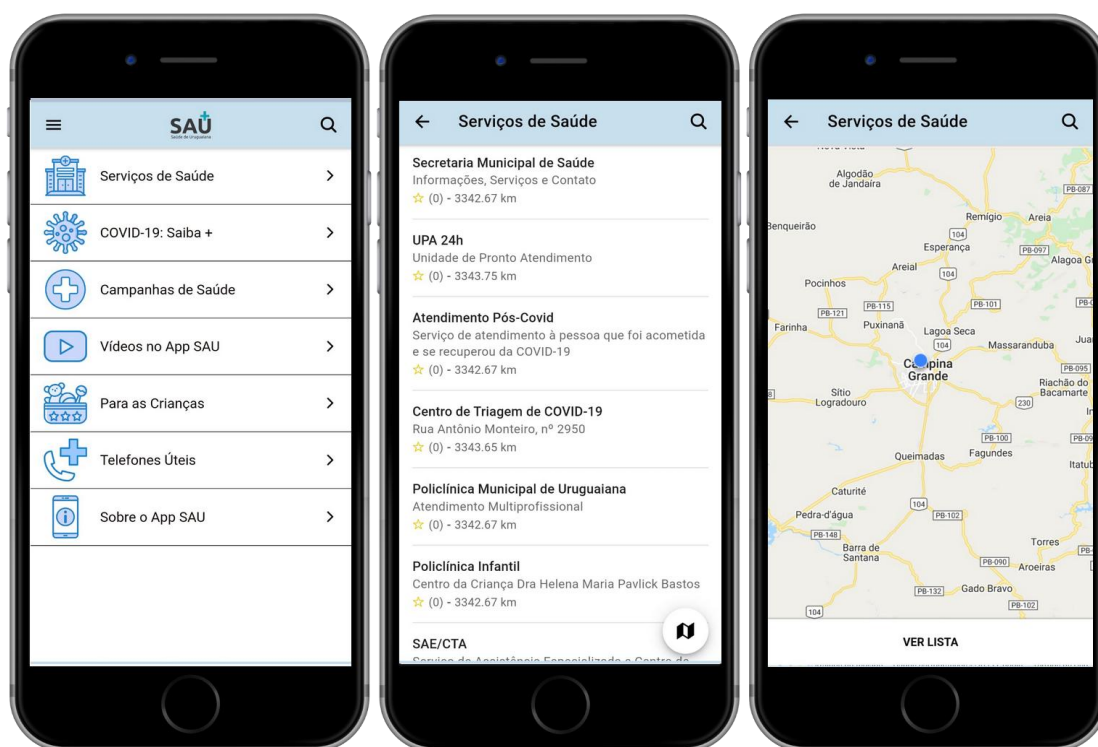
3.2.6 App SAU – Aplicativo da Saúde de Uruguaiiana

O App SAU (APP SAU, 2020), aplicativo desenvolvido por docentes da Unipampa e um acadêmico de Medicina da universidade, auxilia a população com a divulgação de endereços, serviços e contatos das unidades de Uruguaiiana. Também informa sobre protocolos sanitários de segurança da pandemia.

As informações no App SAU contemplam aspectos como endereços, horários de funcionamento, contatos, o que cada serviço oferece, além de campanhas de saúde, telefones úteis (como o do Serviço de Atendimento Móvel de Urgência – SAMU) e dicas sobre o novo coronavírus (o que é, como se prevenir, quais são os sintomas; como ocorre a transmissão). Também é possível visualizar no mapa de Uruguaiiana onde cada um desses serviços se encontra, bem como as campanhas de saúde.

O App SAU está disponível para dispositivos Android e em versão web.

Figura 8 - Telas do aplicativo App SAU



Fonte: App SAU

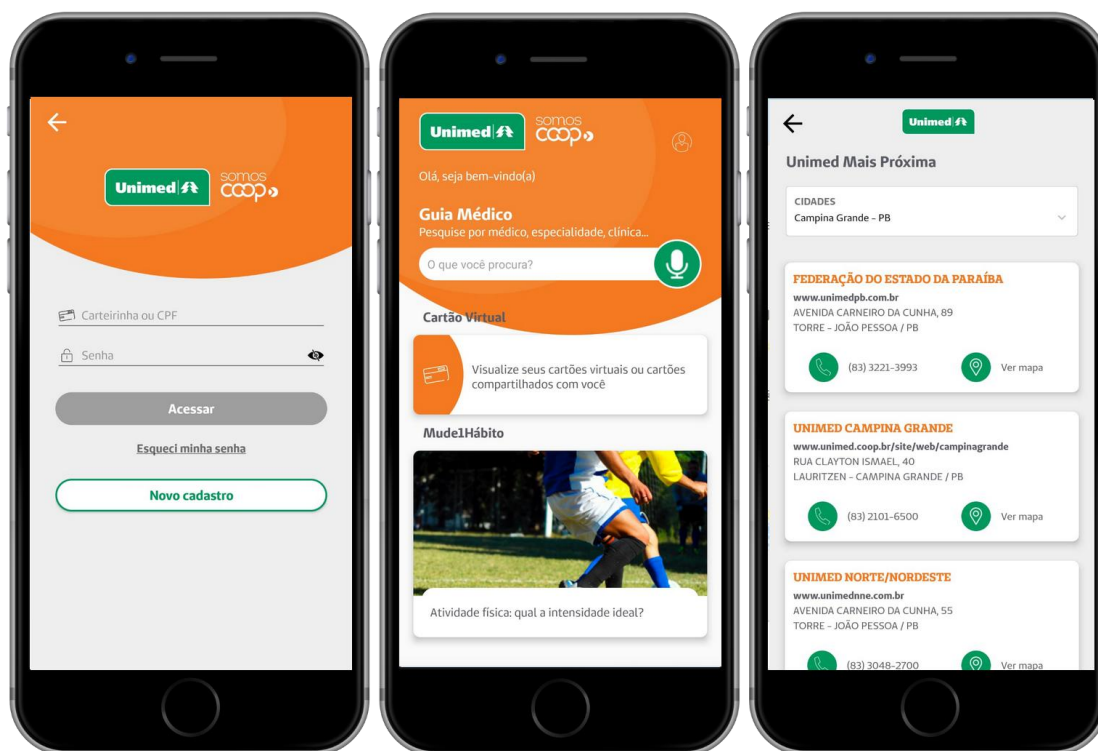
3.2.7 Unimed Cliente

O Unimed Cliente (UNIMED CLIENTE, 2021) é um aplicativo que possibilita o acesso ao Guia Médico Nacional Unimed com a praticidade do comando de voz, acesso a carteirinha virtual Unimed e conteúdos de saúde e qualidade de vida.

Algumas funcionalidades do app são acesso a carteirinha virtual, guia médico, que possibilita o usuário encontrar o médico mais próximo na especialidade que precisa, agendamentos de consultas, acompanhamento de todos os serviços, consultas e exames utilizados através do Plano.

O Unimed Cliente está disponível para dispositivos Android e iOS.

Figura 9 - Telas do aplicativo Unimed Cliente



Fonte: Unimed Cliente

3.3 Comparativo dos trabalhos relacionados

Esta seção apresenta a comparação das funcionalidades principais de todos os trabalhos relacionados (Quadro 1).

Quadro 1 - Comparativo dos trabalhos relacionados

Funcionalidades/ Aplicativo	Conecte SUS	E- saúdeSP	Dr.co nsulta	Sabin	Conecta Recife	App SAU	Unimed Cliente
Cartão de Vacinas	Sim	Sim	Não	Não	Sim	Não	Não
Localização de Serviços de Saúde	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Agendamento de Exames	Não	Não	Sim	Sim	Não	Não	Sim
Resultados de Exames	Não	Sim	Sim	Sim	Não	Não	Sim
Agendar Vacinas	Não	Sim	Sim	Não	Sim	Não	Não
Teleconsultas	Não	Sim	Sim	Não	Não	Não	Não
Agendamento de Consulta Presencial	Não	Não	Sim	Não	Não	Não	Sim

Fonte: Próprio autor

4 RESULTADOS

Este capítulo apresenta os resultados deste trabalho que tratou da modelagem de um aplicativo móvel para busca de unidades de saúde a partir da localização e especialidade médica necessária para o usuário.

4.1 Descrição do sistema




O aplicativo visa facilitar a busca de unidades de saúde de acordo com as necessidades do usuário. Para usar o aplicativo e utilizar as funcionalidades realizar buscas, telefonar para unidade de saúde, traçar rota e solicitar um motorista de aplicativo, não é necessário que o usuário realize cadastro prévio. Porém, se a aplicação for de um cliente privado, como Unimed, Funasa, ou um cliente público, como SUS, entre outros, o usuário deverá fazer um cadastro e login na aplicação, onde essas informações de cadastro devem ser fornecidas pelo cliente. Para que seja possível o usuário responder o questionário com perguntas sobre sintomas que ele pode estar sentindo, com finalidade de conseguir indicá-lo uma unidade de saúde ou especialidade para ser atendido/tratado, deve ser feito o cadastro e login na aplicação, as perguntas do questionário deverão ser fornecidas pelo cliente.

O aplicativo também conta com uma versão web para que seja possível o cadastro das unidades de saúde disponíveis no aplicativo, assim como as informações do questionário de saúde.

4.2 Definição dos Frameworks Web e Mobile

Como citado anteriormente na metodologia, afunilamos a busca dos frameworks web para React, Vue e Angular. Em seguida, fizemos um levantamento sobre os frameworks web escolhidos, foi gerado um quadro de comparação entre eles, essa comparação pode ser encontrada no Quadro 2. A partir desse quadro, conseguimos definir qual framework web utilizar no desenvolvimento da aplicação.

Quadro 2 - Comparação dos frameworks React, Vue e Angular.

	React 	Vue.js 	Angular 
Linguagem de programação	JavaScript, HTML e CSS.	JavaScript, HTML e CSS.	TypeScript, JavaScript, Dart, HTML e CSS.
Empresa	Controlado pelo Facebook	Completamente financiado e apoiado pela comunidade de código aberto	Controlado pelo Google
Curva de aprendizado	Média	Pequena	Alta
Sobrecarga	Ele permite selecionar exatamente o que o desenvolvedor precisa, mas também fornece a capacidade de usar um grande número de pacotes de terceiros. Assim, react não inclui um monte de sobrecarga.	Não inclui muita sobrecarga. Este FW inclui um monte de elementos que podem não ser necessários em uma página, no entanto, permite a ativação desses elementos necessários.	Angular vem com um monte de complementos. Embora a otimização possa cortar um monte de código desnecessário, a estrutura do fundo em que FW é "sobrecarregada com código".
Documentação	Documentação oficial pequena, mas contem varios exemplos e trechos de códigos.	Documentação relativamente pequena	Documentação extensa
Bibliotecas	React é uma estrutura para desenvolvimento de UI, portanto, aplicativos escritos com React precisam de bibliotecas adicionais para serem usadas.	Vue.js ainda não conta com a mesma quantidade de recursos (como plugins) que seus concorrentes diretos possuem – o Angular e o React.	Angular é uma estrutura completa para desenvolvimento de software, que geralmente não requer bibliotecas adicionais.
Mobile frameworks suportados	React native, Ionic.	Ionic Vue, Framework7 Vue, Vue Native, NativeScript Vue, Quasar Framework.	Ionic , NativeScript.

Fonte: Próprio autor

Após a comparação entre os frameworks, vimos algumas vantagens em utilizar o framework Angular para o desenvolvimento da aplicação. Essas vantagens foram:

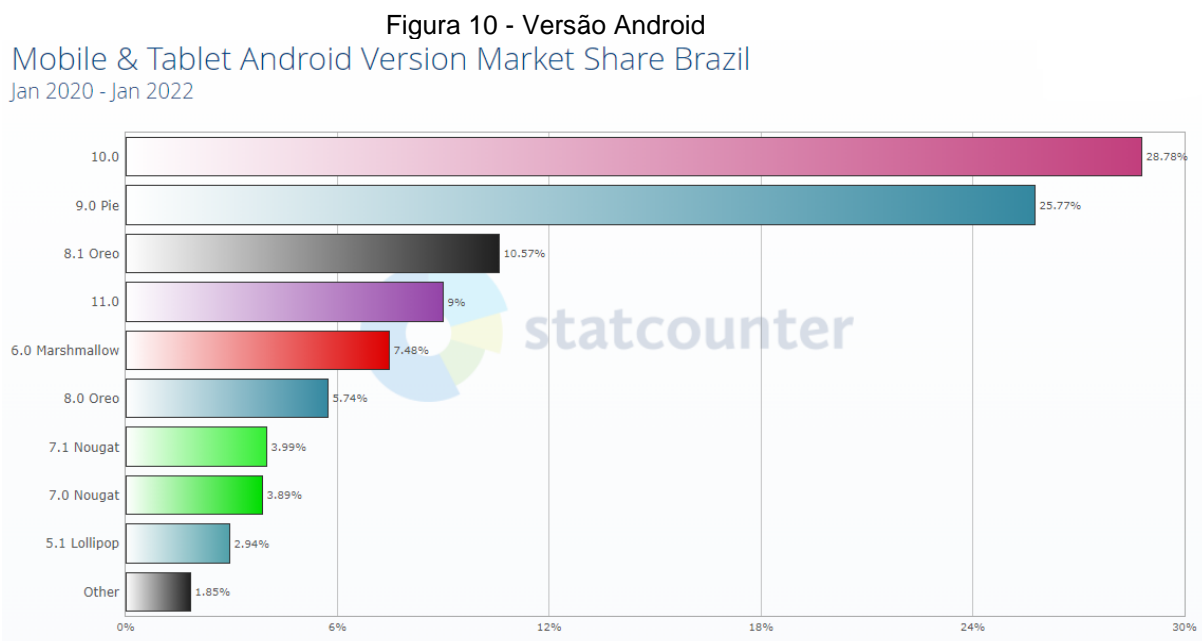
- Angular é uma estrutura completa, que geralmente não requer bibliotecas adicionais.
- O Angular possui um conjunto de ferramentas de material integrado e oferece uma variedade de componentes de design de material pré construídos. Existem vários botões, layouts. Por causa disso, a configuração da interface do usuário se torna mais simples e rápida.
- Possui uma extensa documentação e muitos tutoriais disponíveis.

Com a escolha do framework Angular, o próximo passo foi verificar qual framework para desenvolvimento mobile seria interessante utilizar com o Angular. Após levantamentos sobre os frameworks de desenvolvimento mobile que suportam

o Angular, decidimos utilizar o Ionic. Os motivos de escolha para a utilização do Ionic foram:

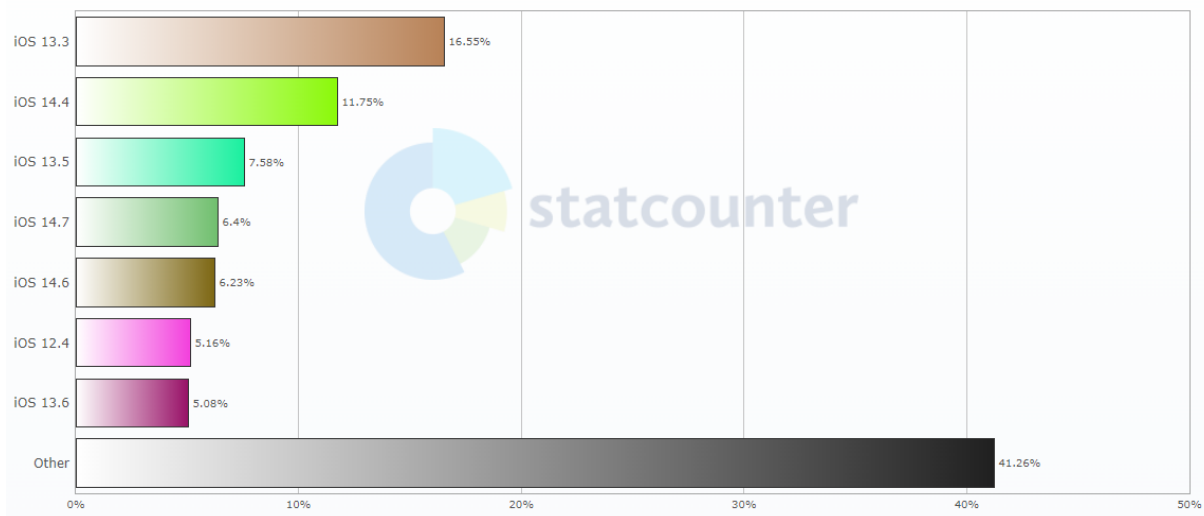
- O desenvolvimento de uma aplicação em Ionic é bem parecido com o de uma aplicação web.
- É multiplataforma, com Ionic podemos desenvolver um único código que seja executado em diferentes sistemas operacionais, como Android e iOS.
- A documentação do Ionic é consistente.
- Com o Ionic, é possível realizar testes no browser.

A aplicação será desenvolvida para os sistemas operacionais Android e iOS. De acordo com Statcounter (2022), o Android 10 é a versão que mais está sendo utilizada desde 2020 no Brasil, como pode ser visto na Figura 10, ainda segundo Statcounter (2022), as versões iOS menores que a versão 12.4 são as mais utilizadas no Brasil atualmente, como mostrado na Figura 11.



Fonte: Statcounter

Figura 11 - Versão iOS
Mobile & Tablet iOS Version Market Share Brazil
Jan 2020 - Jan 2022



Fonte: Statcounter

Com esses dados coletados, foi possível definir qual versão do Ionic será utilizada no desenvolvimento da aplicação. Será utilizado o Ionic v5, ele suporta versões do Android 5.0+ e iOS 11.0+ (IONIC, 2020).

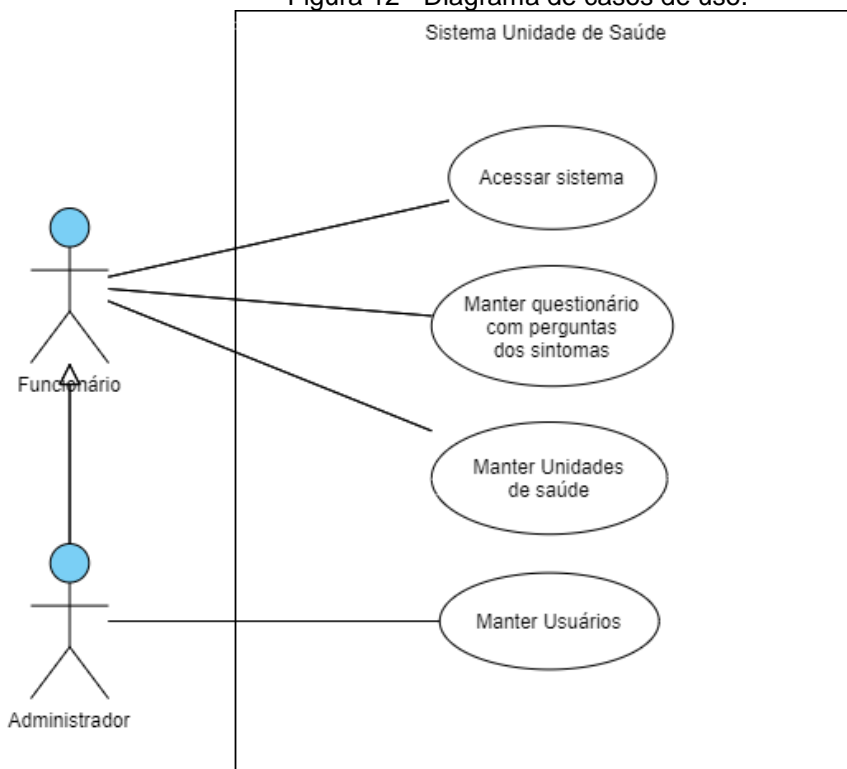
4.3 Diagramas de casos de uso

Após o levantamento dos requisitos, foram criados dois diagramas de casos de uso com o objetivo de modelar o comportamento do sistema, identificando as interações entre a aplicação e seus atores, conforme as Figura 12 e Figura 13.

Foram identificados três atores para interação com as funcionalidades do aplicativo:

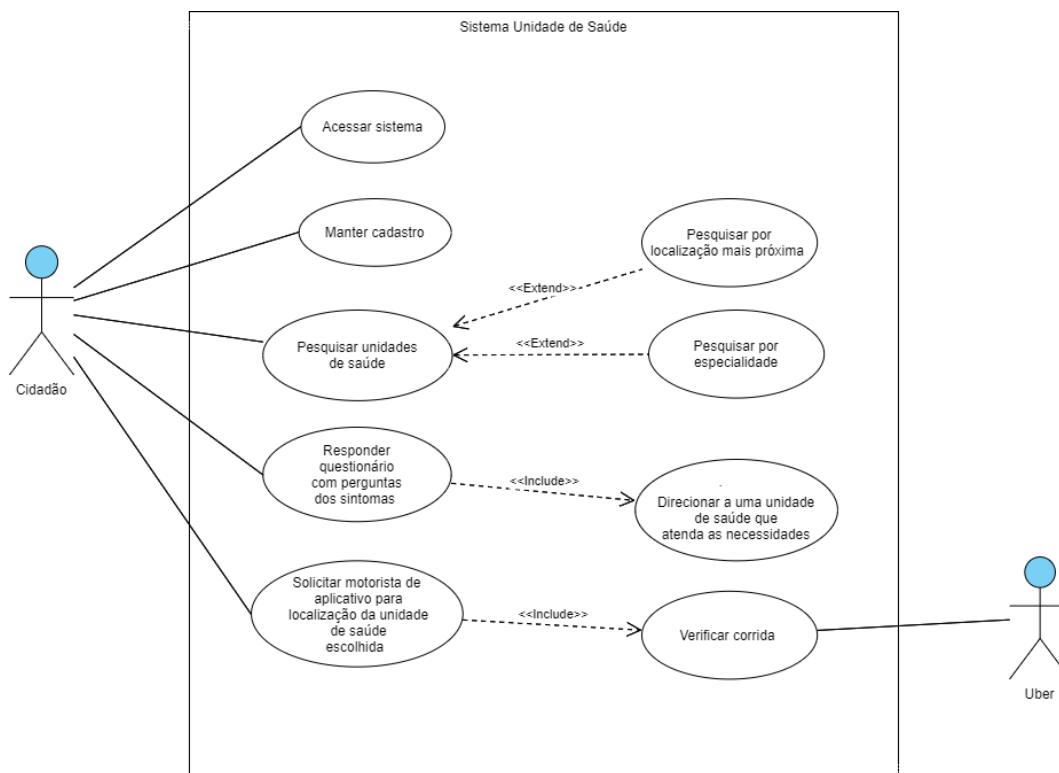
- Funcionário – terá como funcionalidades essenciais manter os serviços de saúde na aplicação, manter questionário com perguntas dos sintomas.
- Administrador – herda as funcionalidades do funcionário e, ainda, pode manter o cadastro dos usuários do sistema.
- Cidadão – terá como funcionalidades manter seu cadastro, pesquisar unidades de saúde, responder questionário sobre sua saúde e solicitar motorista de aplicativo.

Figura 12 - Diagrama de casos de uso.



Fonte: Próprio autor

Figura 13 - Diagrama de casos de uso



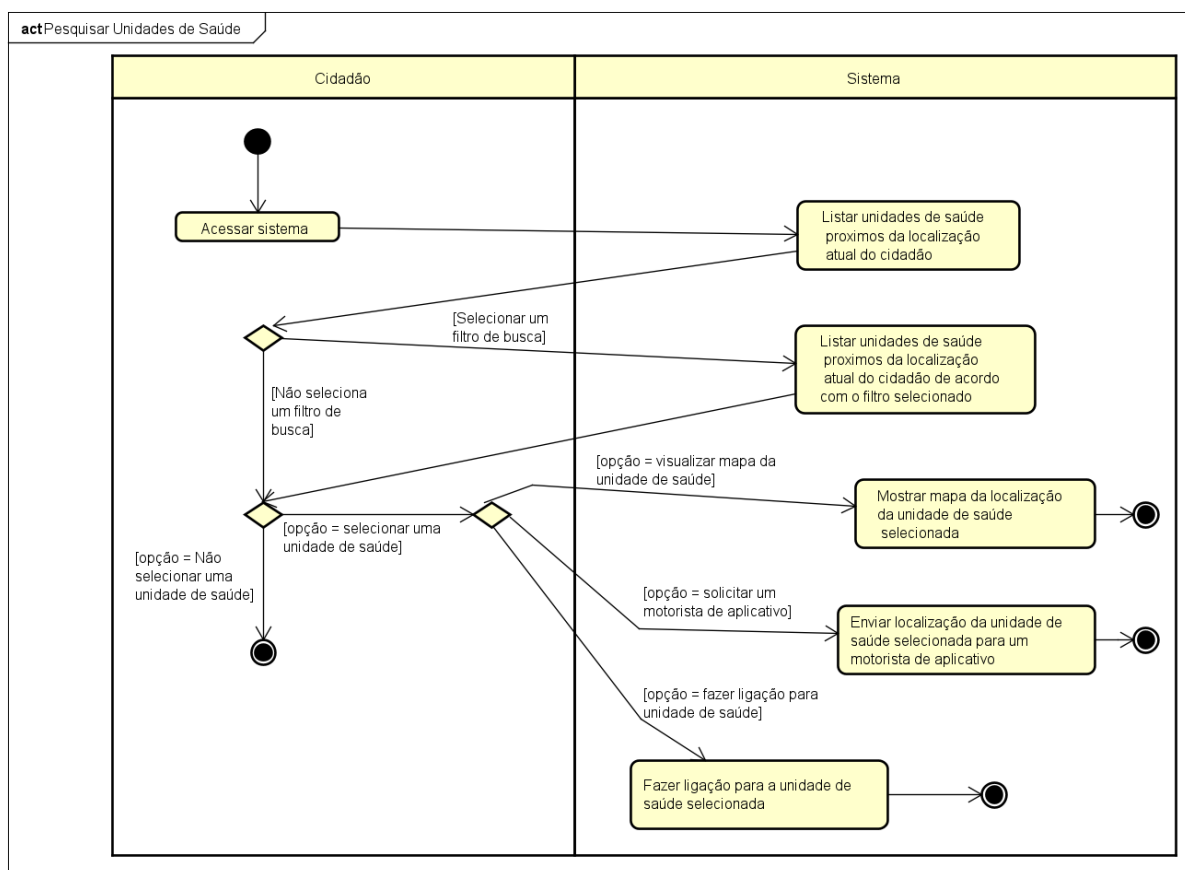
Fonte: Próprio autor

4.4 Diagramas de atividade

Os diagramas de atividades apresentados nas Figuras 14 e 15 representam o fluxo de interação do usuário com as principais funcionalidades da aplicação. Os diagramas de atividades a seguir, foram desenhados utilizando a ferramenta Astah UML.

Na Figura 14, o diagrama apresenta as principais ações realizadas quando o usuário quer localizar uma unidade de saúde, podendo indicar a especialidade médica como filtro de busca. Selecionados os serviços de saúde, o usuário pode visualizar informações deles, como telefone para contato, horário de funcionamento, distância, podendo visualizar no mapa onde se localiza o serviço e podendo solicitar um motorista de aplicativo para a rota desejada.

Figura 14 - Diagrama de atividades para buscar unidades de saúde

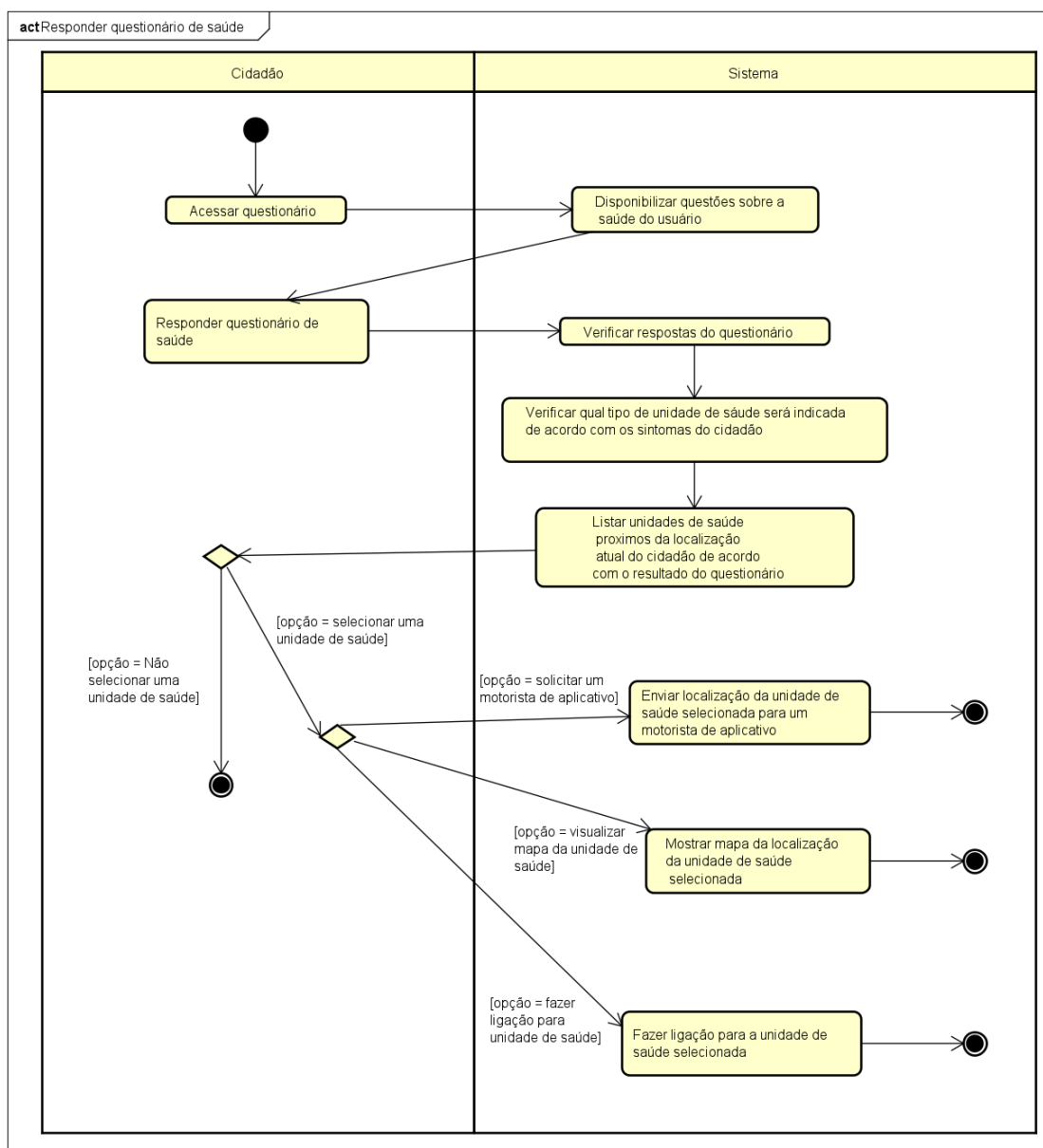


Fonte: Próprio autor

Na Figura 15, o diagrama apresenta as principais ações realizadas quando o usuário responde o questionário sobre sintomas que ele está sentindo, com o

objetivo do sistema conseguir indicá-lo uma unidade de saúde que atenda às suas necessidades. Depois de respondido o questionário, o sistema selecionará os serviços de saúde que possua a especialidade que o usuário necessita. O usuário poderá visualizar informações das unidades de saúde indicadas a ele, como telefone para contato, horário de funcionamento, distância, podendo visualizar no mapa onde se localiza o serviço e podendo solicitar um motorista de aplicativo para a rota desejada.

Figura 15 - Diagrama de atividades para responder questionário de saúde



Fonte: Próprio autor

4.5 Histórias de usuário

Durante a fase de modelagem, foram criadas as histórias de usuário, focando nas necessidades do usuário, pensando sempre em vários contextos e cenários, como a visão de um administrador, de um paciente e de um futuro cliente para a aplicação, como Unimed ou SUS.

Foram criadas 13 histórias de usuário, nesta seção são apresentadas 7 delas, as demais estão inseridas no apêndice A.

4.5.1 Histórias de usuário – Login

Figura 16 - Histórias de usuário – Login

Título: Fazer login
US: Como administrador, quero que o cidadão tenha a opção de fazer login no sistema se ele for um sistema comercial, para que consiga listar as unidades de saúde de acordo com o cliente.
Critérios de aceitação 1 - Quero ter a opção da senha ser visível no login. 2 - Ter a opção de manter conta conectada para não precisar estar inserindo os dados. 3 - Informações de login fornecidas pelo cliente. (como SUS, Unimed, entre outros).

Fonte: Próprio autor

4.5.2 Histórias de usuário – Questionário de Saúde

Figura 17 - Histórias de usuário – Responder Questionário de Saúde

Título: Responder questionário de saúde
<p>US:</p> <p>Como doente, estou sentindo dor facial e não sei para qual unidade de saúde ir, eu quero ter uma opção que possa me indicar uma unidade de saúde que atenda minhas necessidades e que seja perto da minha localização atual.</p>
<p>Critérios de aceitação</p> <p>1 - Se o sistema for comercial, as perguntas do questionário será fornecidas pelo cliente. (como SUS, Unimed, entre outros).</p>

Fonte: Próprio autor

4.5.3 Histórias de usuário – Pesquisar Unidades de Saúde

Figura 18 - Histórias de usuário – Pesquisar Unidades de Saúde

Título: Pesquisar unidades de saúde
<p>US:</p> <p>Como cliente Unimed, quero poder ver a lista de unidades de saúde Unimed perto da minha localização atual, para que eu possa chegar rápido e ser atendido por um profissional.</p>
<p>Critérios de aceitação</p> <p>1 - Poder pesquisar unidades de saúde mais próximas da minha localização.</p> <p>2 - Será necessário ter feito login no sistema.</p> <p>3 - Ter a opção de utilizar localização atual pelo GPS.</p> <p>4 - Informações de login fornecidas pela Unimed.</p>

Fonte: Próprio autor

Figura 19- Histórias de usuário – Pesquisar Unidades de Saúde

Título: Pesquisar unidades de saúde
US: Como doente, estou sentindo dores na garganta e ouvido, quero poder ver uma lista de unidades de saúde que possuem médicos otorrinolaringologistas perto da minha localização atual, para que eu possa chegar rápido e ser atendido por um profissional.
Critérios de aceitação 1 - Poder pesquisar unidades de saúde mais próxima da minha localização com especialidade que preciso. 2 - Não ser necessário ter feito login no sistema. 3 - Ter a opção de utilizar localização atual pelo GPS.

Fonte: Próprio autor

4.5.4 História de usuário – Telefonar para Unidade de Saúde

Figura 20 - Histórias de usuário – Telefonar para Unidade de Saúde

Título: Telefonar para a unidade de saúde
US: Como paciente, escolhi uma unidade de saúde para ser atendido, mas queria tirar algumas dúvidas sobre quais documentos preciso levar para ser atendido, quero poder ter a opção de contatos com a unidade de saúde, como número de telefone.
Critérios de aceitação 1 - Necessário ter localização atual informada no sistema.

Fonte: Próprio autor

4.5.5 Histórias de usuário – Solicitar Motorista de Aplicativo

Figura 21 - Histórias de usuário – Solicitar Motorista de Aplicativo

Título: Solicitar motorista de aplicativo
US: Como doente, quero poder pedir um Uber facilmente, para poder ir à localização da unidade de saúde que escolhi.
Critérios de aceitação 1 - Enviar localização da unidade de saúde selecionada para o aplicativo de transporte.

Fonte: Próprio autor

4.5.6 Histórias de usuário – Visualizar Mapa

Figura 22 - Histórias de usuário – Visualizar Mapa

Título: Visualizar mapa da unidade de saúde
US: Como doente, escolhi uma unidade de saúde para ser atendida, mas não sei o endereço dela, quero poder ver em um mapa onde a unidade de saúde se localiza na região.
Critérios de aceitação 1 - Necessário ter uma localização atual informada no sistema.

Fonte: Próprio autor

4.6 Apresentação da aplicação Mobile e Web

4.6.1 Aplicação Mobile

O aplicativo é composto das seguintes interfaces gráficas, sendo elas as funcionalidades principais da aplicação:

4.6.1.1 Login

Se a aplicação for de um cliente privado ou público, como Funasa ou SUS, entre outros, o usuário deverá possuir um cadastro para ter acesso ao aplicativo. Para utilizar a aplicação deverá realizar sua autenticação utilizando seu login e sua correspondente senha de acesso, sendo essas informações de cadastro/login fornecidas pelo cliente.

Figura 23 - Tela de Login

18:00

←

Bem-vindo(a)!

Login

Senha

ENTRAR

[Esqueci minha senha](#)

CADASTRE-SE

Fonte: Próprio autor

4.6.1.2 Unidades de Saúde

A tela principal da aplicação é a Unidades de Saúde, onde serão disponibilizadas todas as unidades de saúde próximas ao usuário.

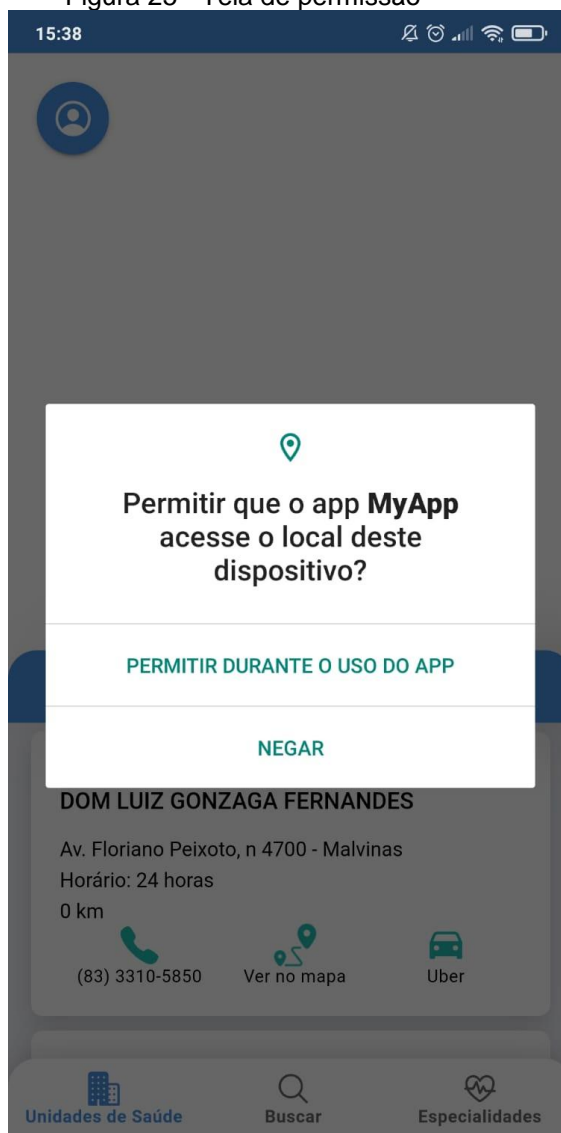
A interface possui um mapa para visualização dos serviços próximos. Para que o sistema consiga identificar a localização atual do dispositivo, é necessário que o usuário dê a permissão para a aplicação acessar essa informação no dispositivo (Figura 25). Após a permissão, o sistema terá acesso à latitude e longitude do usuário. Se for necessário, é possível o usuário arrastar seu marcador no mapa para melhorar a precisão da sua localização.

Figura 24 - Tela de Unidades de Saúde



Fonte: Próprio autor

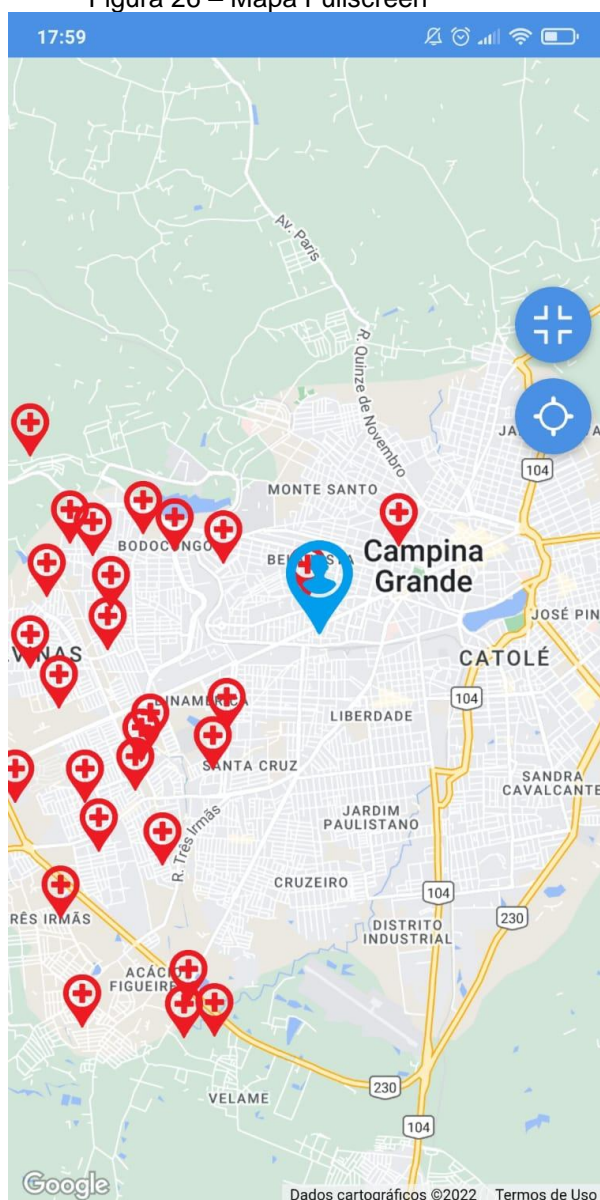
Figura 25 - Tela de permissão



Fonte: Próprio autor

O mapa disponível na aplicação possui dois controles, o controle de *fullscreen*, deixando o mapa ocupar toda a tela do dispositivo, como mostrado na Figura 26. Também possui o controle de centralizar o mapa na localização atual do usuário.

Figura 26 – Mapa Fullscreen



Fonte: Próprio autor

A aplicação disponibiliza uma lista dos serviços próximos ao usuário. Cada serviço tem as seguintes opções:

- Telefonar para a unidade de saúde.
- Traçar a rota da localização atual até o serviço desejado, como mostrado na figura 27.
- Solicitar um motorista do aplicativo Uber, ao selecionar essa opção, o usuário é redirecionado para a aplicação do Uber e o sistema envia as informações de corrida necessárias, local de origem e de destino.

Figura 27 – Traçar rota



Fonte: Próprio autor

4.6.1.3 Buscar Unidade de Saúde

A segunda tela da aplicação é a Buscar Serviço. Ela possui todas as funcionalidades da tela principal, Tela Unidades de Saúde (Figura 24). Sua diferença é que ela possui uma barra de busca. Visando melhorar a experiência do usuário, a aplicação oferece duas opções de buscas, a primeira é a opção de filtrar os serviços próximos por Urgências, Emergências e UBS, como mostrado na Figura 28.

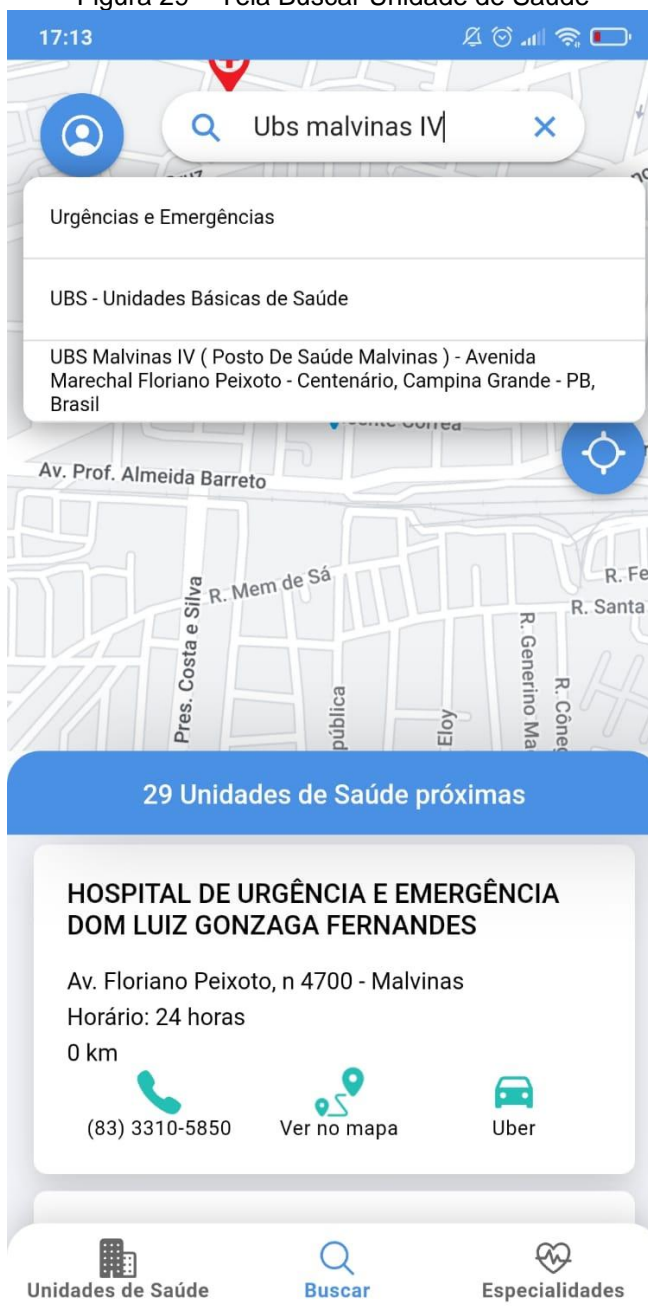
Figura 28 – Tela Buscar Unidade de Saúde



Fonte: Próprio autor

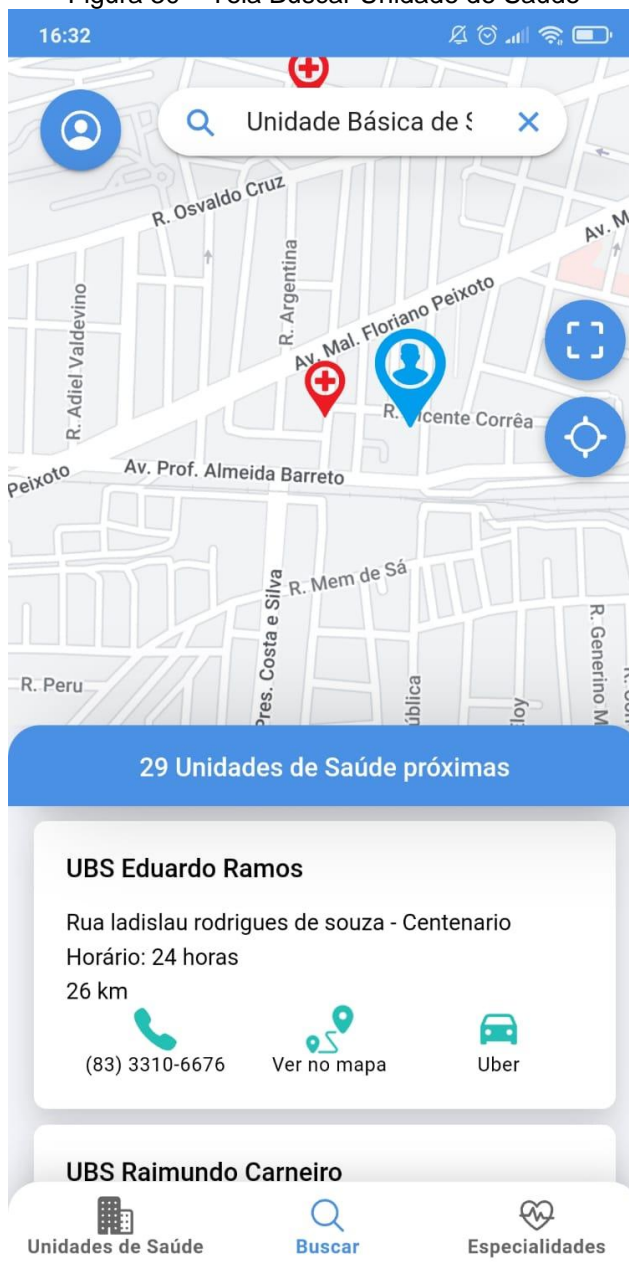
A segunda opção de busca é informar o nome de uma unidade de saúde específica, para localizá-la no mapa e ver suas informações no topo da listagem de serviços próximos, como mostrada na Figura 29 e Figura 30.

Figura 29 – Tela Buscar Unidade de Saúde



Fonte: Próprio autor

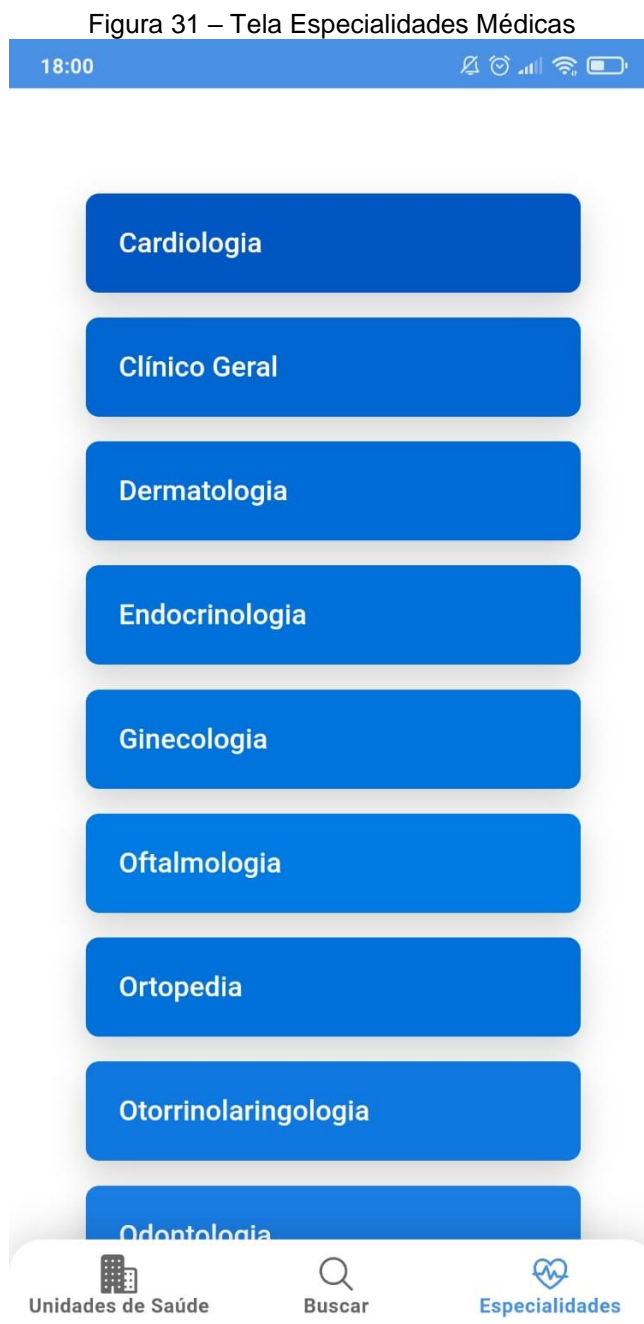
Figura 30 – Tela Buscar Unidade de Saúde



Fonte: Próprio autor

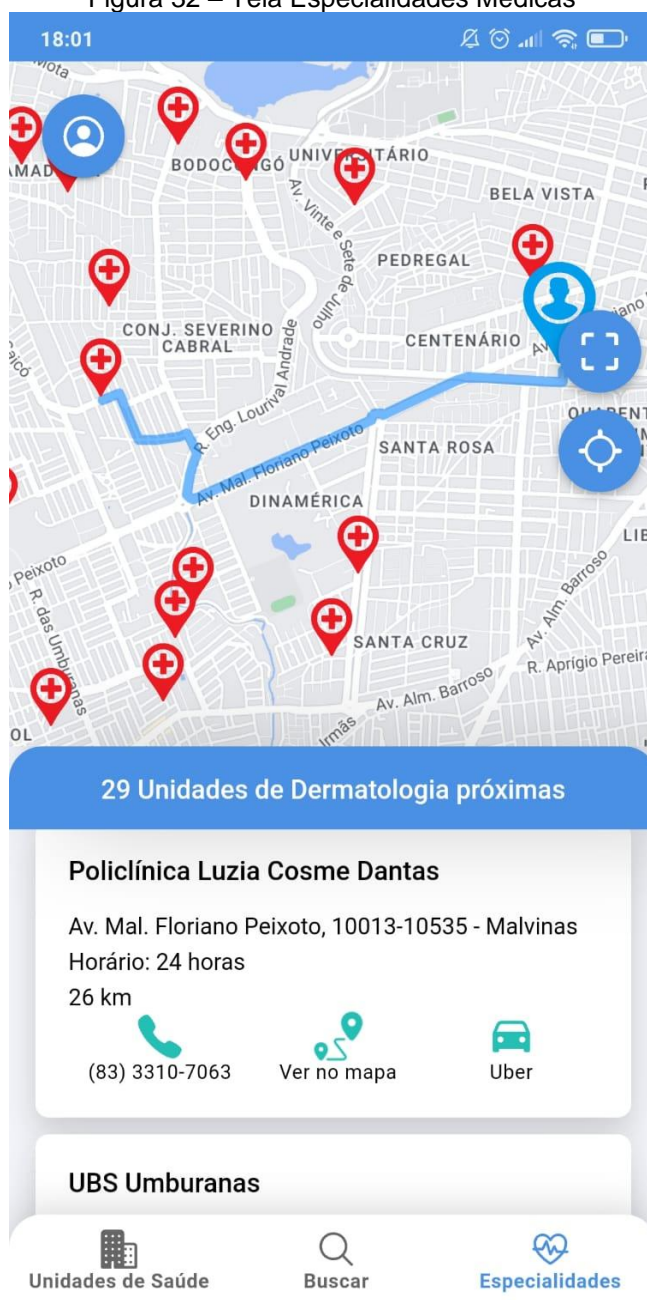
4.6.1.4 Especialidades Médicas

A terceira tela da aplicação é a Especialidades Médicas (Figura 31). A partir dela, o usuário poderá filtrar as unidades de saúde próximas de acordo com a especialidade médica escolhida.



Fonte: Próprio autor

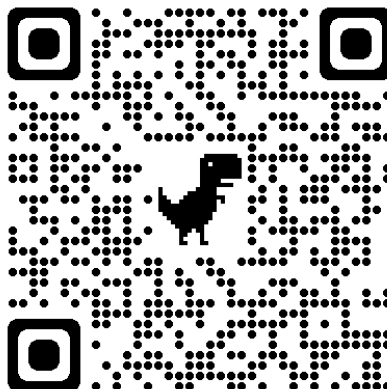
Figura 32 – Tela Especialidades Médicas



Fonte: Próprio autor

4.6.1.5 Código Fonte

O código fonte da aplicação Mobile foi hospedado no GitHub, podendo ser acessado através do link do repositório ou QR Code.



<https://github.com/Danieleventura/app-servico-saude-TCC>

4.6.2 Aplicação Web

Foram desenvolvidas 7 telas do sistema web, mas nesta seção iremos citar as telas das principais funcionalidades do sistema, as demais interfaces gráficas estarão disponíveis no apêndice B.

O sistema web é composto das seguintes principais interfaces gráficas:

4.6.2.1 Login

Como o sistema web será utilizado por administradores e funcionários, ele possui uma tela de login, sempre sendo necessária a autenticação do usuário para ter acesso às funcionalidades da aplicação.

Figura 33 – Tela Login - Sistema web



Fonte: Próprio autor

4.6.2.2 Unidades de Saúde

A principal funcionalidade do sistema web é o gerenciamento das unidades de saúde disponíveis no aplicativo móvel. O sistema possui uma tela de listagem dos serviços cadastrados na aplicação e que estarão disponíveis para os usuários na aplicação *mobile*, como mostrado na Figura 34.

Nessa interface é possível visualizar uma unidade de saúde, cadastrar novo serviço, assim como importar dados de unidades de saúde, exportar dados como XLS ou PDF, ordenar a lista dos serviços por id, tipo de estabelecimento e cidade.

Figura 34 – Listagem das Unidades de Saúde Cadastradas - Sistema web

ID	CNES	NOME	TIPO ESTABELECIMENTO	CIDADE	BAIRRO	TELEFONE
1	7113692	HOSPITAL MUNICIPAL DR SEVERINO BEZERRA DE CARVALHO	HOSPITAL ESPECIALIZADO	CAMPINA GRANDE-IBGE-250400	CENTENARIO	(83) 3341 2097
2	7249071	HOSPITAL DA VISAO DE CAMPINA GRANDE	HOSPITAL ESPECIALIZADO	CAMPINA GRANDE-IBGE-250400	CENTRO	(83) 3066-0606
3	2362287	INSTITUTO DE SAUDE ELPIDIO DE ALMEIDA	HOSPITAL ESPECIALIZADO	CAMPINA GRANDE-IBGE-250400	CENTRO	(83) 33106356
4	9360093	UBS MALVINAS IV	CENTRO DE SAUDE/UNIDADE BASICA	CAMPINA GRANDE-IBGE-250400	ALVARO GAUDENCIO	(83) 33107042
5	9352880	UBS EDUARDO RAMOS	CENTRO DE SAUDE/UNIDADE BASICA	CAMPINA GRANDE-IBGE-250400	CENTENARIO	(83) --
6	9352910	UBS LIBERDADE III	CENTRO DE SAUDE/UNIDADE BASICA	CAMPINA GRANDE-IBGE-250400	LIBERDADE	(83) --

Fonte: Próprio autor



4.6.2.3 Cadastrar Unidade de Saúde

A interface para cadastro de uma unidade de Saúde é simples e objetiva, como pode ser vista na Figura 35. O formulário de cadastro possui campos obrigatórios, caso o funcionário não os preencha, o cadastro não será finalizado.

Os campos obrigatórios são:

- Nome da unidade de saúde
- CNES, o código de CNES é a identificação nacional dos estabelecimentos de saúde, objetivando identificar univocamente uma instituição.
- Telefone
- Tipo de estabelecimento
- As especialidades do estabelecimento
- Horário de funcionamento
- Endereço, bairro e cidade
- Latitude e Longitude

Figura 35 – Cadastrar Unidade de Saúde – Sistema web

Home Oi! Fernanda Santos  2  Sair

Cadastrar Unidade de saúde

OS CAMPOS MARCADOS COM ASTERISCO (*) SÃO DE PREENCHIMENTO OBRIGATÓRIO.

NOME* CNES*

TELEFONE*

DEPENDÊNCIA GESTÃO

TIPO ESTABELECIMENTO*

ESPECIALIDADES DO ESTABELECIMENTO*

Especialidades médicas

[Adicionar especialidades](#)

HORÁRIO DE FUNCIONAMENTO*


DIA DA SEMANA	HORÁRIO
<input type="checkbox"/> SEGUNDA	--:-- <input type="text"/> ÀS --:-- <input type="text"/>
<input type="checkbox"/> TERÇA	--:-- <input type="text"/> ÀS --:-- <input type="text"/>
<input type="checkbox"/> QUARTA	--:-- <input type="text"/> ÀS --:-- <input type="text"/>
<input type="checkbox"/> QUINTA	--:-- <input type="text"/> ÀS --:-- <input type="text"/>
<input type="checkbox"/> SEXTA	--:-- <input type="text"/> ÀS --:-- <input type="text"/>
<input type="checkbox"/> SÁBADO	--:-- <input type="text"/> ÀS --:-- <input type="text"/>
<input type="checkbox"/> DOMINGO	--:-- <input type="text"/> ÀS --:-- <input type="text"/>

ENDEREÇO* BAIRRO*

CIDADE* CEP

LATITUDE* LONGITUDE*

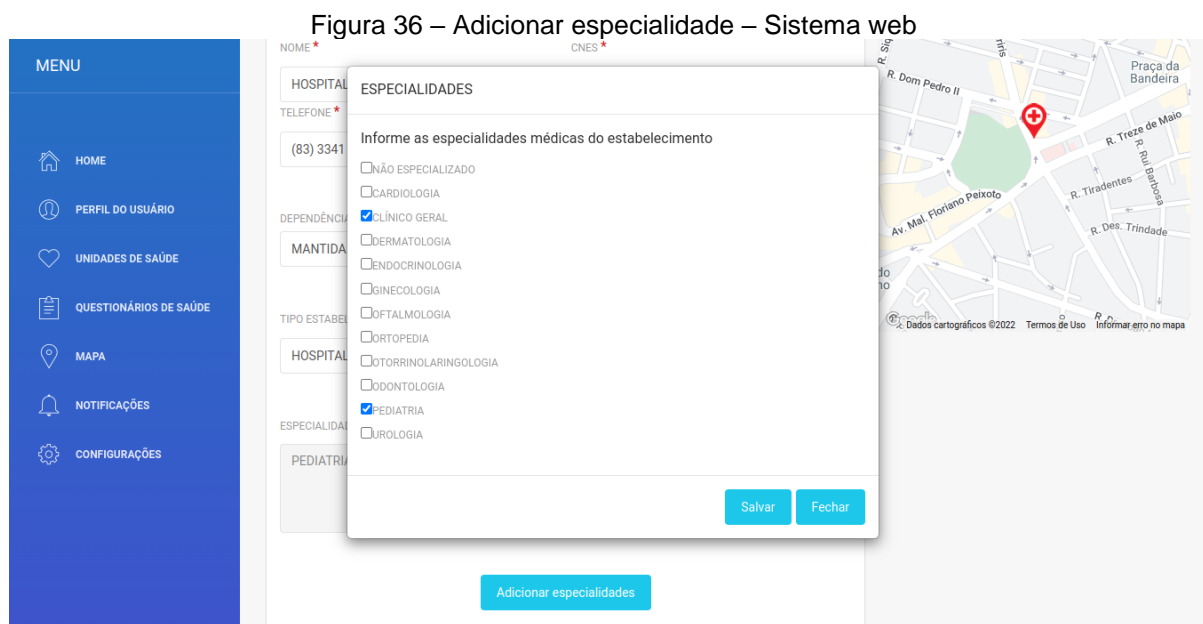
[Salvar Unidade de saúde](#)
[Cancelar Unidade de saúde](#)



Dados cartográficos ©2022 | Termos de Uso | Informar erro no mapa

Fonte: Próprio autor

É importante o usuário informar as especialidades médicas que o estabelecimento oferece. Ao clicar no botão “Adicionar especialidades”, é aberta uma janela que disponibiliza especialidades médicas para o funcionário marcar as que fazem parte do estabelecimento, como mostrado na Figura 36, caso o serviço não seja do tipo especializado, poderá ser marcado a opção “Não especializado”.



Fonte: Próprio autor

Outro campo importante é o horário de funcionamento do estabelecimento. No formulário é disponibilizado *checkbox* com os dias da semana, o funcionário deverá marcar os dias da semana que o estabelecimento estará aberto e informar o horário de abertura e fechamento da unidade de saúde, como mostrado na Figura 37.

Figura 37 – Adicionar horário de funcionamento – Sistema web

HORÁRIO DE FUNCIONAMENTO *

DIA DA SEMANA	HORÁRIO	Á S	HORÁRIO
<input checked="" type="checkbox"/> SEGUNDA	12:00	Á S	21:00
<input checked="" type="checkbox"/> TERÇA	07:00	Á S	19:00
<input type="checkbox"/> QUARTA	--:--	Á S	19 00
<input type="checkbox"/> QUINTA	--:--	Á S	20 01
<input type="checkbox"/> SEXTA	--:--	Á S	21 02
<input type="checkbox"/> SÁBADO	--:--	Á S	22 03
<input type="checkbox"/> DOMINGO	--:--	Á S	23 04
			00 05
			01 06

ENDEREÇO *



CIDADE * **CEP**

Fonte: Próprio autor

4.6.2.4 Visualizar Unidade de Saúde

Selecionando uma unidade de saúde, o usuário será direcionado para a tela de visualização dos dados do serviço, como mostrado na Figura 38. Se desejar, o usuário poderá editar a unidade de saúde selecionada (Figura 39).


Figura 38 – Visualização da Unidade de Saúde – Sistema web

Home Oi! Fernanda Santos   Sair

Unidade de saúde

NOME	CNES
HOSPITAL MUNICIPAL DR SEVERINO BEZERRA DE CARVALHO	7113692
TELEFONE	
(83) 3341 2097	
DEPENDÊNCIA	GESTÃO
MANTIDA	MUNICIPAL
TIPO ESTABELECIMENTO	
HOSPITAL ESPECIALIZADO	
ESPECIALIDADES DO ESTABELECIMENTO	
PEDIATRIA	
CLÍNICO GERAL	
HORÁRIO DE FUNCIONAMENTO	
Segunda-Feira: 07:00 às 17:00	
Terça-Feira: 07:00 às 17:00	
Quarta-Feira: 07:00 às 17:00	
Quinta-Feira: 07:00 às 17:00	
Sexta-Feira: 07:00 às 17:00	
ENDEREÇO	BAIRRO
FLORIANO PEIXOTO	CENTENARIO
CIDADE	CEP
CAMPINA GRANDE-IBGE-250400	-
LATITUDE	LONGITUDE
-7.221412	-35.888401


[Editar Unidade de saúde](#)



Dados cartográficos ©2022 Termos de Uso Informar erro no mapa

Fonte: Próprio autor

Figura 39 – Editar Unidade de Saúde – Sistema web

Home Oi! Fernanda Santos  Sair

Editar Unidade de saúde

OS CAMPOS MARCADOS COM ASTERISCO (*) SÃO DE PREENCHIMENTO OBRIGATÓRIO.

NOME * **CNES ***

TELEFONE *

DEPENDÊNCIA **GESTÃO**

TIPO ESTABELECIMENTO *

ESPECIALIDADES DO ESTABELECIMENTO *

[Adicionar especialidades](#)

HORÁRIO DE FUNCIONAMENTO *


DIA DA SEMANA	HORÁRIO	
<input checked="" type="checkbox"/> SEGUNDA	07:00	ÁS 17:00
<input checked="" type="checkbox"/> TERÇA	07:00	ÁS 17:00
<input checked="" type="checkbox"/> QUARTA	07:00	ÁS 17:00
<input checked="" type="checkbox"/> QUINTA	07:00	ÁS 17:00
<input checked="" type="checkbox"/> SEXTA	07:00	ÁS 17:00
<input type="checkbox"/> SÁBADO	--:--	ÁS --:--
<input type="checkbox"/> DOMINGO	--:--	ÁS --:--

ENDEREÇO * **BAIRRO ***

CIDADE * **CEP**

LATITUDE * **LONGITUDE ***

[Atualizar Unidade de saúde](#)
[Excluir Unidade de saúde](#)

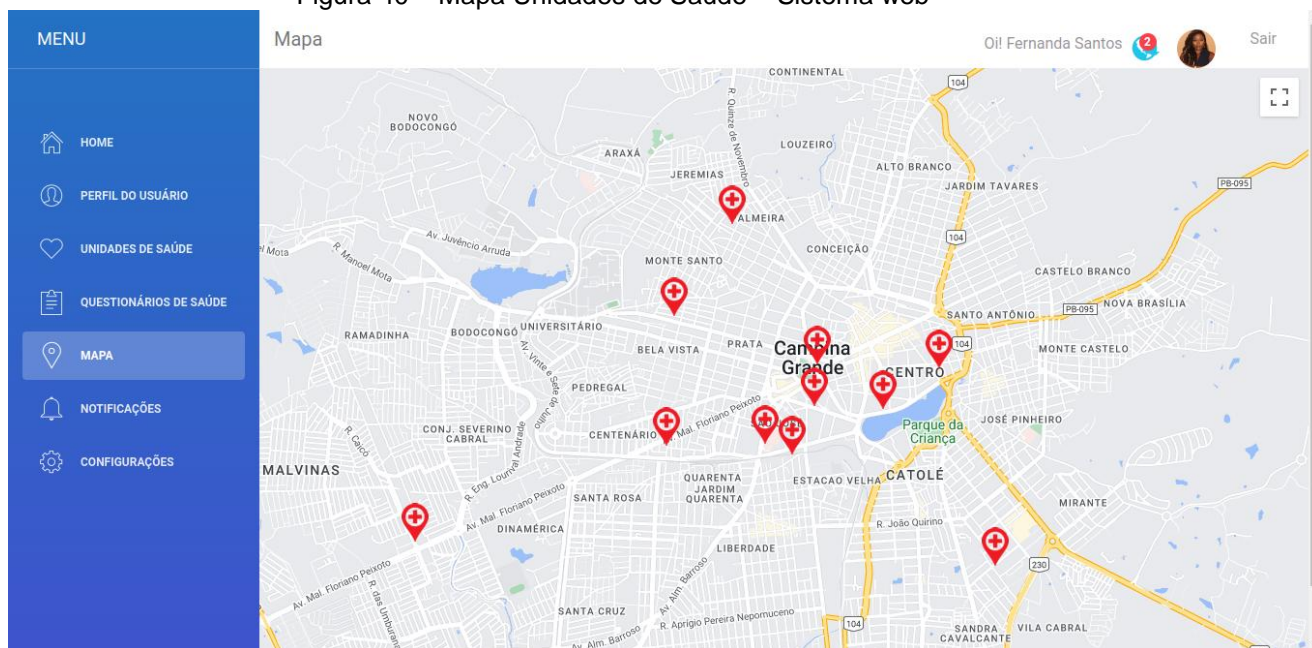


Fonte: Próprio autor

4.6.2.5 Mapa Unidades de Saúde

No sistema o usuário poderá visualizar em um mapa as unidades de saúde cadastradas, assim como o usuário da aplicação móvel.

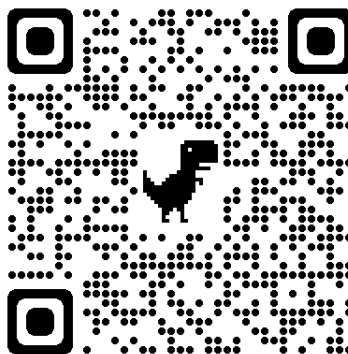
Figura 40 – Mapa Unidades de Saúde – Sistema web



Fonte: Próprio autor

4.6.2.6 Código Fonte

O código fonte da aplicação Web foi hospedado no GitHub, podendo ser acessado através do link do repositório ou QR Code.



<https://github.com/Danieleventura/crud-sistema-web-TCC>

4.7 Implementação

A implementação da aplicação *mobile* foi realizada com as tecnologias Ionic v5, Angular, HTML, CSS e Typescript.

Como o Ionic é um framework para desenvolvimento híbrido, para ter acesso a algumas funcionalidades nativas da plataforma Android ou IOS, como acesso a geolocalização, fazer chamadas telefônicas, entre outros, foi utilizado o Apache Cordova.

Existem muitos *plugins* do Cordova para o Ionic. Um *plugin* é um código complementar que fornece interface JavaScript para componentes nativos. Eles permitem que seu aplicativo use recursos de dispositivos nativos além do que está disponível para aplicativos da Web puros (CORDOVA, 2022).

Para a aplicação identificar a localização do usuário, foi utilizado o *plugin* Geolocation. Este *plugin* fornece informações sobre a localização do dispositivo, como latitude e longitude (IONIC, 2022b).

Quando a aplicação é iniciada no dispositivo do usuário, é solicitada através do *plugin* Geolocation a localização atual do dispositivo, na Listagem 1 apresenta o método que faz essa solicitação. Se a aplicação tiver sucesso em acessar a localização do dispositivo, então o sistema irá carregar o mapa com o marcador do usuário e solicitar as unidades de serviço próximas.

Listagem 1 – Uso do plugin geolocation

```
async initPage(){
  await this.geolocation.getCurrentPosition().then((result) => {
    this.myPosition = new google.maps.LatLng(result.coords.latitude,
result.coords.longitude);
    this.loadMap(result.coords.latitude, result.coords.longitude);
  }).catch((error) => {
    console.log('Error getting location', error);
  });
}
```

Fonte: Próprio autor

Foi utilizado o *plugin* Call Number para realizar chamadas para as unidades de saúde. Este *plugin* possibilita ligar para um número diretamente do aplicativo Cordova/Ionic. Na Listagem 2 mostra a implementação para realizar a chamada telefônica.

Listagem 2 – Uso do plugin Call Number

```
makeCall(numberPhone) {
  this.callNumber.callNumber(numberPhone, true).then(res =>
    console.log('Launched dialer!', res))
    .catch(err => console.log('Error launching dialer', err));
}
```

Fonte: Próprio autor

Foi utilizado o *plugin* Google Maps e Maps JavaScript API (GOOGLE, 2022a) na aplicação. Esta API permite adicionar mapas, marcadores, controles personalizados, possibilitando a modelagem do mapa de acordo com a necessidade da aplicação. Para utilizar essa API na aplicação, é necessário acessar o Google Cloud Platform Console e criar uma conta de cobrança, usada para rastrear os custos associados aos seus projetos, criar um projeto para gerenciar serviços, credenciais, cobrança, APIs. E por último habilitar a Maps JavaScript API, que fornece a chave de acesso aos serviços da API, que deve ser adicionada à aplicação para a configuração do mapa. A chave da API é usada para autenticar as solicitações associadas à aplicação para fins de uso e cobrança.

Conforme pode ser visto na Listagem 3, é possível verificar como é feita a utilização do *plugin* Google Maps por meio do método de `loadMap`, o método informa no mapa onde o usuário se encontra localizado.

Listagem 3 – Uso do plugin Google Maps

```
loadMap(lat, lng){
  const styles = {
    hide: [
      {
        featureType: 'transit',
        elementType: 'labels.icon',
        stylers: [{ visibility: 'off' }],
      },
      {
        featureType: 'poi',
        elementType: 'labels.icon',
        stylers: [{ visibility: 'off' }],},},};
  const latLng = new google.maps.LatLng(lat, lng);
  const mapOption = {
    center: latLng,
    zoom: 16,
    mapTypeId: google.maps.MapTypeId.ROADMAP,
    disableDefaultUI: true,
    keyboardShortcuts: false,
```

```

    fullscreenControl: false,
    styles: styles.hide
  };
  const element = document.getElementById('map1');
  this.map= new google.maps.Map(element, mapOption);
  var image = {
    url: 'assets/img/marker-user.png',
    scaledSize: new google.maps.Size(70, 70)
  };
  const marker = new google.maps.Marker({
    position: latLng,
    map: this.map,
    title: 'Minha localização',
    icon: image,
    draggable:true,
    animation: google.maps.Animation.DROP});}

```

Fonte: Próprio autor

Para o cálculo das rotas no mapa, foi utilizado a Directions API. A Directions API é um serviço da Web que usa uma solicitação HTTP para retornar rotas formatadas em JSON ou XML entre locais (GOOGLE, 2022b). Como mostrado na Listagem 4, para ser possível traçar uma rota no mapa é necessário informar como parâmetros as coordenadas geográficas de origem e destino.

Listagem 4 – Uso da Directions API

```

plotRouteMap(service) {
  const latLngH = new google.maps.LatLng(service.lat, service.lng) ;
  const rota = google.maps.DirectionsRequest = {
    origin: this.myPosition,
    destination: latLngH,
    unitSystem: google.maps.UnitSystem.METRIC,
    travelMode: google.maps.TravelMode.DRIVING
  };
  this.direction.route(rota, (result, status) => {
    if(status==='OK'){
      this.directionsRender.setMap(this.map);
      this.directionsRender.setDirections(result);
      this.directionsRender.setOptions({suppressMarkers: true});
      console.log(result);
    }
  });
}

```

Fonte: Próprio autor

As APIs Places Autocomplete Service e Geocoding Service foram utilizadas para a ação de buscar as unidades de saúde na barra de pesquisa. Places Autocomplete Service é um serviço da Web que retorna previsões de local em

resposta a uma solicitação HTTP (GOOGLE, 2022c). A solicitação especifica uma string de pesquisa textual, como o nome da unidade de saúde desejada. A aplicação, portanto, envia consultas à medida que o usuário digita, para fornecer previsões de local em tempo real, na Listagem 5 apresenta o uso da API.

Listagem 5 – Uso da API Places Autocomplete Service

```
searchAddress(event: any) {
  const search = event.target.value as string;
  this.autoComplete.getPlacePredictions({input: search},
(arrayLocais, status) =>{
  if(status==='OK' && this.addressId !== this.searchValue){
    this.ngZone.run(()=>{
      this.addressList = arrayLocais;
    });}
  });}
```

Fonte: Próprio autor

Com a API Geocoding Service é possível converter endereços (como "1600 Amphitheatre Parkway, Mountain View, CA") em coordenadas geográficas (como latitude 37.423021 e longitude -122.083739) (GOOGLE, 2022d). Utilizando a API foi possível identificar a coordenada geográfica da unidade de serviço que o usuário fez a busca utilizando a API Places Autocomplete Service, como mostrado na Listagem 6.

Listagem 6 – Uso da API Geocoding Service

```
getAddress(service) {
  this.addressList = [];
  this.filterList = [];
  new google.maps.Geocoder().geocode({address: service.description},
resultado =>{
  this.addressId = service.description;
  this.searchValue = service.description;
  this.map.setCenter(resultado[0].geometry.location);
  this.map.setZoom(16);
  this.loadHospital(resultado, service.description);
  if(this.directionsRender !== null){
    this.directionsRender.setMap(null); });});}
```

Fonte: Próprio autor

Foi utilizado o Deep Links da API da Uber para solicitar motorista pela aplicação. Deep Link fornece interoperabilidade de aplicativos entre um aplicativo nativo ou visualização da Web e o aplicativo nativo da Uber. Os links diretos são simplesmente URLs que fazem referência ao aplicativo Uber e suportam parâmetros

de consulta para afetar o comportamento do aplicativo Uber uma vez iniciado (UBER, 2022).

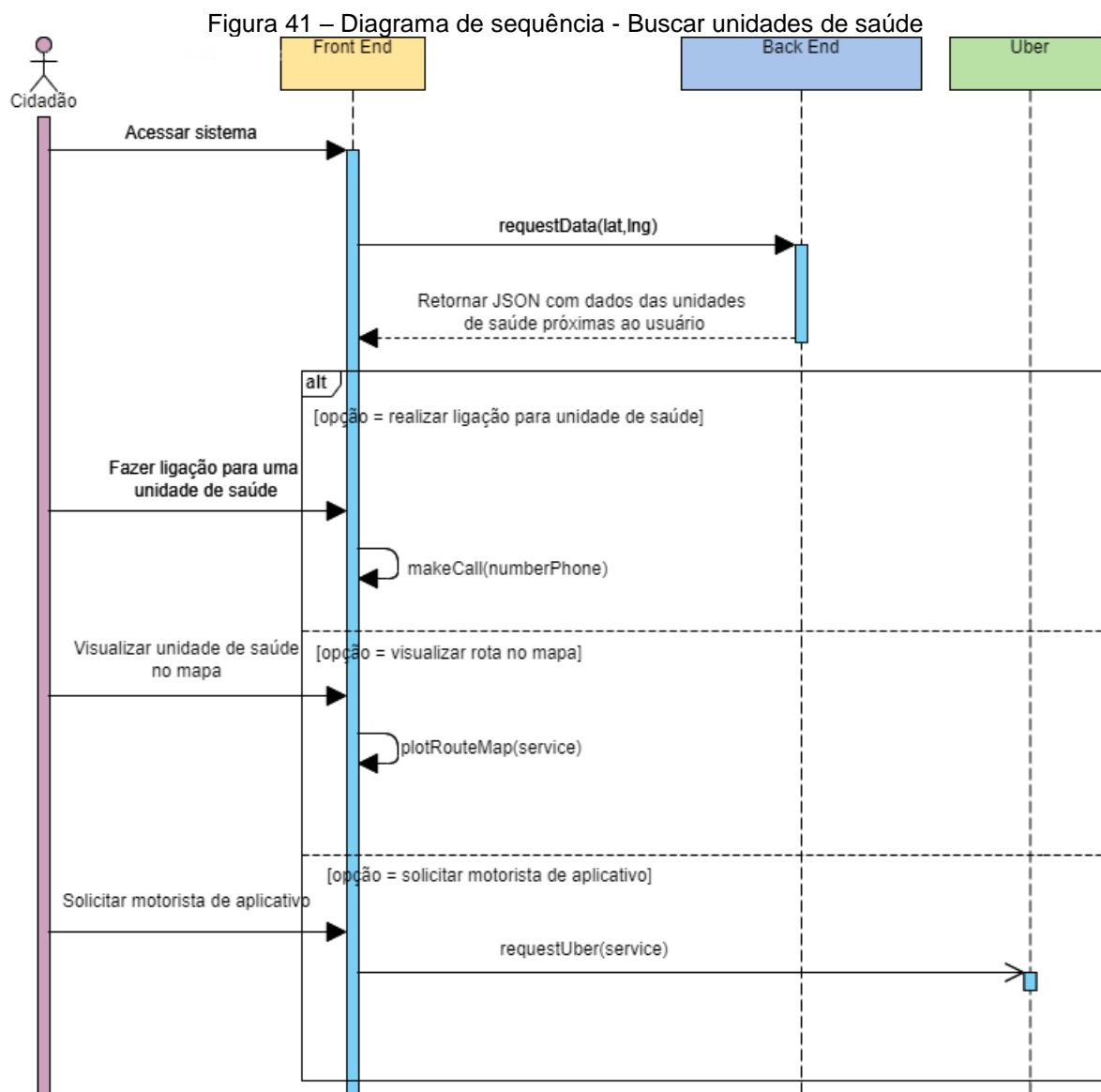
O parâmetro *pickup=my_location* informa a Uber a localização do dispositivo. O parâmetro *dropoff[latitude]* e *dropoff[longitude]* informa o local de destino da corrida, sendo essa a localização da unidade de saúde desejada. A Listagem 7 apresenta o uso da API.

Listagem 7 – Uso do Deep Link Uber

```
requestUber(service) { const url =  
'https://m.uber.com/ul/?action=setPickup&client_id=  
<CLIENT_ID>&pickup=my_location&dropoff[formatted_address]=' +  
service.name + '&dropoff[latitude]=' + service.lat +  
'&dropoff[longitude]=' + service.lng; window.open(url, '_system',  
'location=yes'); }
```

Fonte: Próprio autor

A Figura 41 apresenta o diagrama de sequência que representa o fluxo de interação do front-end com o back-end. Esse fluxo apresenta as principais ações realizadas quando a aplicação é iniciada. O front-end envia para o back-end a latitude e longitude do usuário, com isso ele deverá retornar um JSON com as informações dos serviços próximos.



Fonte: Próprio autor

O arquivo JSON retornado pelo back-end deverá conter todas as informações necessárias que o front-end precisa, seguindo a seguinte estrutura:

Listagem 8 – JSON

```

[ {
  "name": "HOSPITAL DE URGÊNCIA E EMERGÊNCIA DOM LUIZ GONZAGA FERNANDES",
  "address": "Av. Floriano Peixoto, n 4700 - Malvinas",
  "phone": "(83) 3310-5850",
  "time": "24 horas",
  "km": 0,
  "lat": -7.236958161492183,
  "lng": -35.929954448041556
} ]
  
```

Fonte: Próprio autor

A implementação da aplicação web foi realizada com as tecnologias Angular, HTML, CSS e Typescript.

No desenvolvimento da aplicação web foi utilizado o Bootstrap para facilitar a programação de alguns componentes, como o menu lateral do sistema. O Bootstrap (BOOTSTRAP, 2022) é uma ferramenta gratuita para desenvolvimento HTML, CSS e JS. Sendo um framework de desenvolvimento front-end gratuito e de código aberto para a criação de sites e aplicativos da web, ele facilita o desenvolvimento de sites e aplicativos responsivos e *mobile-first*.

Também foi utilizada a Maps JavaScript API no sistema web, junto com a API foi utilizado o componente @ngui/map. Com esse componente instalado é possível utilizar e estilizar o Google Maps na aplicação. Com o componente e a API, foi possível implementar o mapa para visualização das unidades de saúde cadastradas no sistema. A listagem 9 mostra a utilização do componente na aplicação.

Listagem 9 – Uso do @ngui/map

```
<ngui-map zoom="15" style="margin-top:60px;height: 100%; width: 100%;
position: absolute;top: 0px;
        left: 0px; background-color: rgb(229, 227, 223);"
[styles]="stylesHide.hide" center="{{dataJson[0].lat}},{{dataJson[0].lng}}"
[disableDefaultUI]="true"
        [keyboardShortcuts]="false" [fullscreenControl]="true">
    <marker *ngFor="let data of dataJson"
        (click)="infoWindow(data)" [position]="[data.lat,data.lng]"
[icon]="marker" ></marker> </ngui-map>
```

Fonte: Próprio au

A aplicação depende de dados vindos do back-end, como as informações das unidades de saúde cadastradas, então o arquivo JSON retornado pelo back-end deverá conter todas as informações necessárias que o front-end precisa, seguindo a seguinte estrutura:

Listagem 10 – JSON

```
[{
  "id": 1,
  "name": "HOSPITAL MUNICIPAL DR SEVERINO BEZERRA DE CARVALHO",
  "address": "FLORIANO PEIXOTO",
  "district": "CENTENARIO",
  "phone": "(83) 3341 2097",
  "time": ["Segunda-Feira: 07:00 às 17:00",
    "Terça-Feira: 07:00 às 17:00",
    "Quarta-Feira: 07:00 às 17:00",
    "Quinta-Feira: 07:00 às 17:00",
    "Sexta-Feira: 07:00 às 17:00"],
  "lat": -7.221412,
```

```
"lng": -35.888401,  
"city": "CAMPINA GRANDE-IBGE-250400",  
"type": "HOSPITAL ESPECIALIZADO",  
"management": "MUNICIPAL",  
"dependency": "MANTIDA",  
"CNES": 7113692,  
"speciality":["PEDIATRIA", "CLÍNICO GERAL"],  
"CEP": 58428-130  
}]
```

Fonte: Próprio autor

Na área da saúde a aplicação de cores precisa ser adequada para transmitir a sensação de maior bem estar para o paciente, família e profissionais (BOCCANERA; BOCCANERA; BARBOSA, 2006). A paleta de cores de uma aplicação pode influenciar na usabilidade do aplicativo e na experiência do usuário.

Sabendo que as cores são importantes na área da saúde, foram selecionadas as cores azul-claro, verde-claro e branco para uso da paleta de cores das aplicações desenvolvidas, *mobile* e *web*.

De acordo com Boccanera et al. (2006) a cor branca está associada, entre outros aspectos, à simplicidade, limpeza, paz, pureza, harmonia e estabilidade. O verde é uma cor fria, aliviando e acalmando tanto física quanto mentalmente, ele é considerado como uma cor tranqüilizante. A cor azul proporciona um ambiente calmo e organizado, ela reduz o stress e a tensão.

Com essas combinações de cores na aplicação, é esperado que o usuário tenha uma melhor experiência ao utilizá-la.

5 CONSIDERAÇÕES FINAIS

Este trabalho apresentou o desenvolvimento front-end e a documentação do aplicativo para busca de serviços de saúde, apresentando a sua justificativa, descrição geral, tecnologias utilizadas para a implementação e seus resultados.

O aplicativo tem como sua principal funcionalidade a busca pelas unidades de saúde na região de localização do usuário. Para que esses serviços sejam visualizados pelos usuários é necessário que eles sejam previamente cadastrados por meio do sistema web.

Para o desenvolvimento do trabalho, foi necessário estudos sobre: experiência de usuário para a criação da interface gráfica; integração com APIs externas, como Google Maps; utilização de plugins Cordova; desenvolvimento com as tecnologias Ionic, Angular e Bootstrap.

Durante a modelagem e implementação, foi importante se pôr no lugar do usuário final da aplicação, foi possível entender as necessidades do aplicativo, na visão de um usuário em um momento de saúde frágil, respeitando a experiência de usuário, como telas de fácil entendimento e objetivas. Focamos em alcançar o maior número de dispositivos Android e IOS.

Na fase de implementação, foi necessário definir a troca de informações que o front-end terá com o back-end. No momento que este trabalho estava sendo desenvolvido, em paralelo, outro aluno estava desenvolvendo o back-end da aplicação como seu trabalho de conclusão de curso. Então, foi importante neste trabalho definir a estrutura do arquivo JSON e as informações necessárias contidas nele, que serão retornadas pelo back-end, para que o front-end recebesse os dados certos das unidades de saúde.

Uma dificuldade encontrada na fase de implementação, foi o uso de componentes no framework Ionic. Em uma implementação teste da aplicação *mobile*, foi desenvolvido um componente para a funcionalidade mapa que utiliza a API Google Maps, não foi possível utilizar o mesmo componente em telas diferentes, sendo retornado alguns erros ao tentar utilizá-lo em mais de uma interface, ficando para estudos futuros o aprendizado da programação e utilização de componentes do Ionic.

Alguns requisitos importantes e desejados das aplicações acabaram não sendo desenvolvidos, como o questionário de saúde, que auxiliaria o usuário na busca de uma unidade de saúde que trate sua necessidade, pois não houve tempo hábil de serem analisados, planejados e executados da mesma maneira e com o mesmo empenho e dedicação que os outros requisitos foram. As funcionalidades não implementadas neste trabalho poderão ser desenvolvidas em trabalhos futuros, complementando e melhorando as aplicações.

Os resultados obtidos foram satisfatórios, visto que o aplicativo mostrou a possibilidade de implementação de um problema recorrente em todo o país, a cultura de ir ao pronto socorro, ajudando o usuário a conhecer e localizar as unidades de saúde próximas e suas especialidades. A aplicação não está presa a nenhuma plataforma móvel específica e nem a alguma região do país, podendo ser utilizada por pessoas que estejam precisando encontrar serviços de saúde na sua cidade ou região.

5.1 Trabalhos futuros

Como trabalhos futuros ficam todas as funcionalidades pensadas, porém não desenvolvidas e melhorias no código, com destaque para alguns que estão listados a seguir:

- Questionário de saúde: a implementação dessa funcionalidade agregaria muito a aplicação, já que auxiliaria o usuário a descobrir o serviço médico que ele necessita.
- Melhorias no design: O design da aplicação web possui melhorias que podem ser realizadas, para que tenha uma melhor usabilidade.
- API para cadastro: No sistema web seria interessante utilizar no formulário de cadastro uma API que ao informar o CEP do estabelecimento, retornasse as informações do endereço, como nome da rua, bairro e cidade.
- Melhorias no código: Há melhorias significativas no código da aplicação móvel. Durante a implementação não foi muito explorado a criação de componentes, sendo componente uma vantagem do framework Ionic. O código da aplicação pode ser reduzido utilizando componentes para algumas funcionalidades.

REFERÊNCIAS BIBLIOGRÁFICAS

ALCANTARA, C. A. A.; VIEIRA, A. L. N. TECNOLOGIA MÓVEL: UMA TENDÊNCIA, UMA REALIDADE. 2011.

ANGULAR. **Angular**. Disponível em: <<https://angular.io/>>. Acesso em: 27 mar. 2022.

APP SAU. **Aplicativo da Saúde de Uruguaiiana**. Disponível em: <https://pwa4.app.vc/app_da_saude_em_uruguaiiana/home>. Acesso em: 28 mar. 2022.

ARAUJO, R. C. Frameworks de Desenvolvimento Mobile Multiplataforma - Uma Análise a partir da ISO 9126. 2015.

BOCCANERA, N. B.; BOCCANERA, S. F. B.; BARBOSA, M. A. As cores no ambiente de terapia intensiva: percepções de pacientes e profissionais. **Revista da Escola de Enfermagem da USP**, v. 40, n. 3, p. 343–349, 2006.

BOOTSTRAP. **Bootstrap**. Disponível em: <<https://getbootstrap.com.br/>>. Acesso em: 26 jul. 2022.

CONECTA RECIFE. **Conecta Recife**. Disponível em: <<https://conectarecife.recife.pe.gov.br/>>. Acesso em: 28 mar. 2022.

CONECTE SUS. **Conecte SUS Cidadão**. Disponível em: <<https://www.gov.br/saude/pt-br/assuntos/conecte-sus/cidadao>>. Acesso em: 12 mar. 2022.

CORDOVA. **Apache Cordova**. Disponível em: <<https://cordova.apache.org/docs/en/3.1.0/guide/overview/>>. Acesso em: 23 jul. 2022.

DE, R. Teletriagem: Há Benefícios Para Os Sistemas De Saúde E Seus Usuários ? 2017.

DEITEL, P.; DEITEL, H.; DEITEL, A. **Android™ para Programadores Uma abordagem baseada em aplicativos**. [s.l: s.n.].

DR.CONSULTA. **Dr.consulta**. Disponível em: <<https://drconsulta.com/>>. Acesso em: 28 mar. 2022.

E-SAUDESP. **e-saudeSP – Plataforma da Saúde Paulistana**. Disponível em: <<https://e-saude.sp.prefeitura.sp.gov.br/public-login>>. Acesso em: 28 mar. 2022.

GOOGLE. **API JavaScript do Google Maps**. Disponível em: <<https://developers.google.com/maps/documentation/javascript/overview>>. Acesso em: 25 jul. 2022a.

GOOGLE. **Directions API overview**. Disponível em: <<https://developers.google.com/maps/documentation/directions/overview>>. Acesso em: 25 jul. 2022b.

GOOGLE. **Place Autocomplete**. Disponível em: <<https://developers.google.com/maps/documentation/places/web-service/autocomplete>>. Acesso em: 25 jul. 2022c.

GOOGLE. **Geocoding Service**. Disponível em: <<https://developers.google.com/maps/documentation/javascript/geocoding>>. Acesso em: 25 jul. 2022d.

GOOGLE PLAY. **Google Play**. Disponível em: <https://play.google.com/store/?utm_source=latam_Med&utm_medium=hasem&utm_content=Jul1520&utm_campaign=Evergreen&pcampaignid=MKT-FDR-latam-br-1002290-Med-hasem-py-Evergreen-Jul1520-Text_Search_BKWS-34087548205&gclid=CjwKCAjwloCSBhAeEiwA3hVo_eiRoL06X6bPR5bo>. Acesso

em: 27 mar. 2022.

IONIC. **Ionic Docs - Browser Support**. Disponível em: <<https://ionicframework.com/docs/reference/browser-support>>. Acesso em: 26 mar. 2022.

IONIC. **Introdução ao Ionic**. Disponível em: <<https://ionicframework.com/docs/>>. Acesso em: 27 mar. 2022a.

IONIC. **Ionic Docs**. Disponível em: <<https://ionicframework.com/docs/native/geolocation>>. Acesso em: 25 jul. 2022b.

MARCUS, M.; OLIVEIRA, R. D. E. **Introdução ao android**. v. 2, p. 1–71, 2019.

MEIRELLES, F. S. **Panorama do Uso de TI no Brasil**, 2021.

MORSCH, D. J. A. **CAUSAS, CONSEQUÊNCIAS E POSSÍVEIS SOLUÇÕES DA SUPERLOTAÇÃO NOS HOSPITAIS**. Disponível em: <<https://telemedicinamorsch.com.br/blog/superlotacao-nos-hospitais>>. Acesso em: 16 mar. 2022.

NEVES, J.; JUNIOR, V. M. **Uma Análise Comparativa Entre Flutter E React Native Como Frameworks Para Desenvolvimento Híbrido De Aplicativos Mobile ... Ciência da Computação-Tubarão**, 2020.

OVERFLOW, S. **Tecnologias mais populares: Frameworks Web**. Disponível em: <<https://insights.stackoverflow.com/survey/2021#section-most-popular-technologies-web-frameworks>>. Acesso em: 26 mar. 2022.

SABIN. **Sabin Medicina Diagnóstica**. Disponível em: <<https://www.sabin.com.br/>>. Acesso em: 28 mar. 2022.

SAÚDE, C. **O problema das idas desnecessárias à emergência**. Disponível em: <<https://www.conexasaude.com.br/blog/idas-desnecessarias-a-emergencia/>>. Acesso em: 30 mar. 2022.

STATCOUNTER. **Statcounter**. Disponível em: <<https://gs.statcounter.com/>>. Acesso em: 26 mar. 2022.

STORE, A. **App Store**. Disponível em: <<https://www.apple.com/br/app-store/>>. Acesso em: 27 mar. 2022.

TYPESCRIPT. **TypeScript**. Disponível em: <<https://www.typescriptlang.org/>>. Acesso em: 23 jul. 2022.

UBER. **Introduction to Deep Links**. Disponível em: <<https://developer.uber.com/docs/riders/ride-requests/tutorials/deep-links/introduction>>. Acesso em: 25 jul. 2022.

UNIMED CLIENTE. **App Unimed Cliente**. Disponível em: <<https://www.appunimed.com/beneficiario/beneficiario/index.html>>. Acesso em: 28 mar. 2022.

UZELLI, G. D. P. et al. **Avaliação das dificuldades enfrentadas pelo paciente para realização de uma consulta médica de nível terciário**. **Com. Ciência Saúde**, v. 23, n. 3, p. 207–14, 2013.

WA, O. et al. **evidence synthesis (Review)**. 2020.

APÊNDICE A – HISTÓRIAS DE USUÁRIO

Figura 42 - Histórias de usuário – Login

<p>Título: Fazer login</p>
<p>US:</p> <p>Como cliente Unimed, eu quero que o sistema tenha login com informações de entrada da Unimed.</p>
<p>Critérios de aceitação</p> <ol style="list-style-type: none"> 1 - Quero ter a opção da senha ser visível no login. 2 - Ter a opção de manter conta conectada para não precisar estar inserindo os dados. 3 - Informações de login fornecidas pela Unimed.

Fonte: Próprio autor

Figura 43 - Histórias de usuário – Pesquisar Unidades de Saúde

<p>Título: Pesquisar unidades de saúde</p>
<p>US:</p> <p>Como doente, quero poder ver uma lista de unidades de saúde perto da minha localização atual, para que eu possa chegar rápido e ser atendido por um profissional.</p>
<p>Critérios de aceitação</p> <ol style="list-style-type: none"> 1 - Poder pesquisar unidades de saúde mais próximas da minha localização. 2 - Não será necessário ter feito login no sistema. 3 - Ter a opção de utilizar localização atual pelo GPS.

Fonte: Próprio autor

Figura 44 - Histórias de usuário – Visualizar Mapa

Título: Visualizar mapa da unidade de saúde
US: Como doente, escolhi uma unidade de saúde para ser atendido, mas não entendi onde fica o endereço dela, quero poder ver em um mapa onde a unidade de saúde se localiza na região, que seja de fácil entendimento.
Critérios de aceitação 1 - Necessário ter uma localização atual informada no sistema.

Fonte: Próprio autor

Figura 45 - Histórias de usuário – Responder Questionário de Saúde

Título: Responder questionário de saúde
US: Como cliente Unimed, eu quero que o cidadão tenha opção de poder responder um questionário com perguntas sobre sintomas que ele está sentindo e depois ser redirecionado a uma unidade de saúde que atenda às suas necessidades perto da sua localização.
Critérios de aceitação 1 - Perguntas do questionário fornecidas pelo cliente Unimed.

Fonte: Próprio autor

Figura 46 - Histórias de usuário – Pesquisar Unidades de Saúde

Título: Pesquisar unidades de saúde
US: Como doente, estou sentindo alguns sintomas da Covid19, como febre e dificuldade para respirar, quero poder ver uma lista de hospitais de campanha ou unidades de saúde que atende minha necessidade perto da minha localização atual, para que eu possa chegar rápido e ser atendido por um profissional.
Critérios de aceitação 1 - Poder pesquisar unidades de saúde mais próxima da minha localização com especialidade que preciso. 2 - Não ser necessário ter feito login no sistema. 3 - Ter a opção de utilizar localização atual pelo GPS.

Fonte: Próprio autor

Figura 47 - Histórias de usuário – Solicitar Motorista de Aplicativo

Título: Solicitar motorista de aplicativo
US: Como doente, estou sentindo dores no olho e não consigo dirigir, preciso ir ao oftalmologista perto da minha casa, quero poder pedir um Uber facilmente, para poder ir para localização da unidade de saúde que escolhi.
Critérios de aceitação 1 - Enviar localização da unidade de saúde selecionada para o aplicativo de transporte.

Fonte: Próprio autor

APÊNDICE B – INTERFACES GRÁFICAS DO SISTEMA WEB

Tela Home – Sistema web

Home

Oi! Fernanda Santos Sair

MENU

- HOME
- PERFIL DO USUÁRIO
- UNIDADES DE SAÚDE
- QUESTIONÁRIOS DE SAÚDE
- MAPA
- NOTIFICAÇÕES
- CONFIGURAÇÕES

Tarefas
Funcionário

- Cadastrar as novas unidades de saúde
- Verificar com Marcela as novas informações
- Reunião com Marcos - 15h
- Retornar ligação para supervisor

Atualizado 3 minutos atrás

JULHO 2022

S	T	Q	Q	S	S	D
27	28	29	30	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	31		

Fonte: Próprio autor

Tela Perfil de Usuário – Sistema web

Perfil do Usuário

Oi! Fernanda Santos Sair

MENU

- HOME
- PERFIL DO USUÁRIO
- UNIDADES DE SAÚDE
- QUESTIONÁRIOS DE SAÚDE
- MAPA
- NOTIFICAÇÕES
- CONFIGURAÇÕES

Editar Perfil

EMPRESA (DESABILITADO) USERNAME EMAIL
 Serviços Saúde Company fernanda.saude Email

PRIMEIRO NOME SEGUNDO NOME
 Fernanda Santos

ENDEREÇO
 Avenida 24 de Agosto

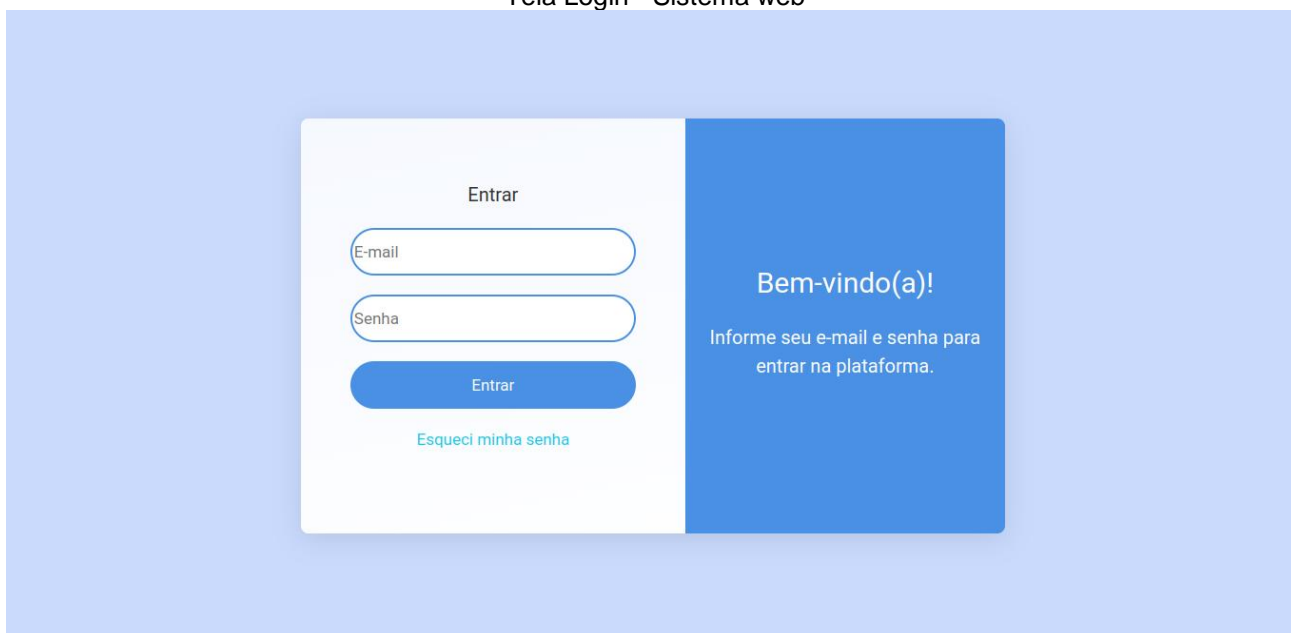
CIDADE PAÍS CÓDIGO POSTAL
 São Paulo Brasil

Atualizar Perfil

Fernanda Santos
fernanda.saude
Funcionário

Fonte: Próprio autor

Tela Login - Sistema web



Fonte: Próprio autor

Listagem das Unidades de Saúde Cadastradas - Sistema web

MENU

- HOME
- PERFIL DO USUÁRIO
- UNIDADES DE SAÚDE**
- QUESTIONÁRIOS DE SAÚDE
- MAPA
- NOTIFICAÇÕES
- CONFIGURAÇÕES

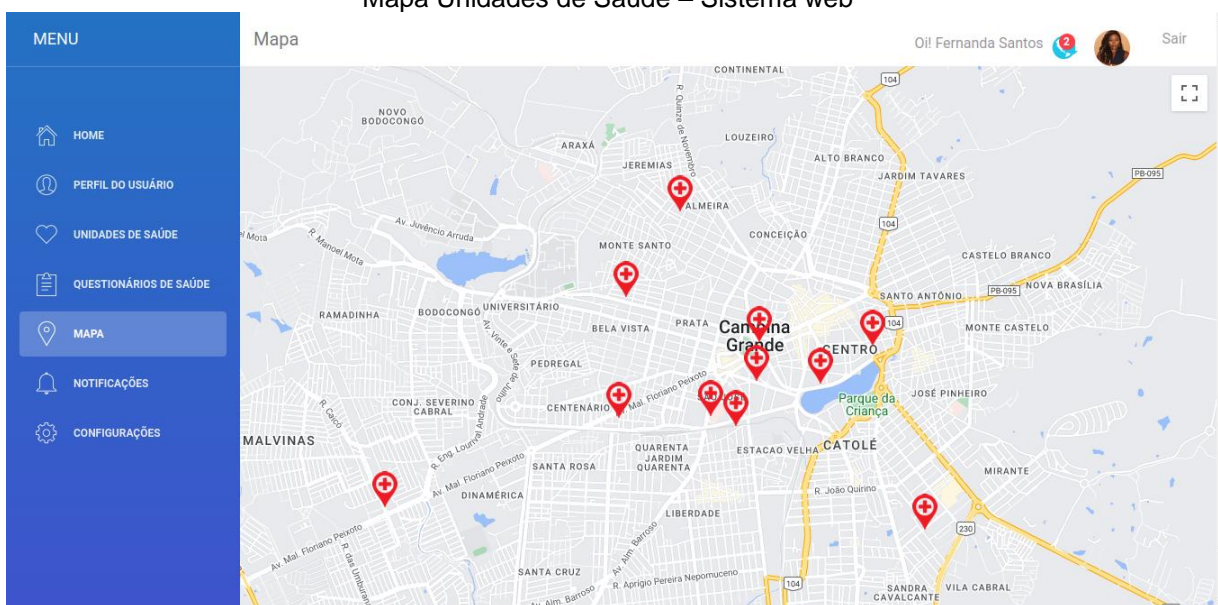
Unidades de saúde Oii Fernanda Santos Sair

11 Unidades de saúde cadastradas

ID	CNES	NOME	TIPO ESTABELECIMENTO	CIDADE	BAIRRO	TELEFONE
1	7113692	HOSPITAL MUNICIPAL DR SEVERINO BEZERRA DE CARVALHO	HOSPITAL ESPECIALIZADO	CAMPINA GRANDE-IBGE-250400	CENTENARIO	(83) 3341 2097
2	7249071	HOSPITAL DA VISAO DE CAMPINA GRANDE	HOSPITAL ESPECIALIZADO	CAMPINA GRANDE-IBGE-250400	CENTRO	(83) 3066-0606
3	2362287	INSTITUTO DE SAUDE ELPIDIO DE ALMEIDA	HOSPITAL ESPECIALIZADO	CAMPINA GRANDE-IBGE-250400	CENTRO	(83) 33106356
4	9360093	UBS MALVINAS IV	CENTRO DE SAUDE/UNIDADE BASICA	CAMPINA GRANDE-IBGE-250400	ALVARO GAUDENCIO	(83) 33107042
5	9352880	UBS EDUARDO RAMOS	CENTRO DE SAUDE/UNIDADE BASICA	CAMPINA GRANDE-IBGE-250400	CENTENARIO	(83) --
6	9352910	UBS LIBERDADE III	CENTRO DE SAUDE/UNIDADE BASICA	CAMPINA GRANDE-IBGE-250400	LIBERDADE	(83) --
		CENTRO DE SAUDE/UNIDADE	CAMPINA GRANDE			(83)

Fonte: Próprio autor

Mapa Unidades de Saúde – Sistema web



Fonte: Próprio autor

Visualização da Unidade de Saúde – Sistema web

Home Oi! Fernanda Santos 2 Sair

Unidade de saúde

NOME	CNES
HOSPITAL MUNICIPAL DR SEVERINO BEZERRA DE CARVALHO	7113692
TELEFONE	
(83) 3341 2097	
DEPENDÊNCIA	GESTÃO
MANTIDA	MUNICIPAL
TIPO ESTABELECIMENTO	
HOSPITAL ESPECIALIZADO	
ESPECIALIDADES DO ESTABELECIMENTO	
PEDIATRIA	
CLÍNICO GERAL	
HORÁRIO DE FUNCIONAMENTO	
Segunda-Feira: 07:00 às 17:00	
Terça-Feira: 07:00 às 17:00	
Quarta-Feira: 07:00 às 17:00	
Quinta-Feira: 07:00 às 17:00	
Sexta-Feira: 07:00 às 17:00	
ENDEREÇO	BAIRRO
FLORIANO PEIXOTO	CENTENARIO
CIDADE	CEP
CAMPINA GRANDE-IBGE-250400	-
LATITUDE	LONGITUDE
-7.221412	-35.888401

[Editar Unidade de saúde](#)

Fonte: Próprio autor