

UNIVERSIDADE DE TAUBATÉ

Vitor Corrêa Leite

William Isaque Rabelo Nunes

**MEDICAL UNIT FINDER: PROTÓTIPO DE APLICATIVO
PARA A LOCALIZAÇÃO E INDICAÇÃO DE UNIDADES
DE ATENDIMENTO MÉDICO**

Taubaté - SP

2021

Vitor Corrêa Leite
William Isaque Rabelo Nunes

**MEDICAL UNIT FINDER: PROTÓTIPO DE APLICATIVO
PARA A LOCALIZAÇÃO E INDICAÇÃO DE UNIDADES
DE ATENDIMENTO MÉDICO**

Trabalho de Graduação apresentado como requisito parcial para a conclusão do curso de Análise e desenvolvimento de Sistemas do Departamento de Informática da Universidade de Taubaté.

Orientador: Prof. Dawilmar Guimarães de Araújo

Taubaté – SP

2021

**Grupo Especial de Tratamento da Informação - GETI
Sistema Integrado de Bibliotecas – SIBi
Universidade de Taubaté - Unitau**

L533m Leite, Vitor Corrêa
Medical Unit Finder: protótipo de aplicativo para a localização e indicação de unidades de atendimento médico / Vitor Corrêa Leite, William Isaque Rabelo Nunes. -- 2021.
31 f. : il.

Monografia (graduação) – Universidade de Taubaté, Departamento de Informática, 2021.

Orientação: Prof. Me. Dawilmar Guimarães de Araújo, Departamento de Informática.

1. Medical Unit Finder. 2. Tecnologia de geolocalização. 3. Aplicativo Mobile. I. Nunes, William Isaque Rabelo. II. Universidade de Taubaté. Departamento de Informática. Graduação em Análise e Desenvolvimento de Sistemas. III. Título.

CDD – 005.26

VITOR CORRÊA LEITE
WILLIAM ISAQUE RABELO NUNES

**MEDICAL UNIT FINDER: PROTÓTIPO DE APLICATIVO PARA A
LOCALIZAÇÃO E INDICAÇÃO DE UNIDADES DE ATENDIMENTO
MÉDICO**

Trabalho de conclusão de curso apresentado
para obtenção do Certificado de Graduação
pelo Curso de Análise e desenvolvimento de
sistemas do Departamento de Informática da
Universidade de Taubaté

Data: 08 / 12 / 2021

Resultado: APROVADO

BANCA EXAMINADORA

Prof. Me. Dawilmar Guimarães Araújo



Universidade de Taubaté

Orientador

Prof. Dr. José Carlos Lombardi



Universidade de Taubaté

Convidado

Prof. Me. Antônio Esio M Salgado



Universidade de Taubaté

Convidado

Dedicamos este trabalho a todos que
contribuíram para a nossa formação acadêmica

AGRADECIMENTOS

Ao Prof. Me. Dawilmar Guimarães de Araújo, pelos conselhos com que orientou o nosso trabalho.

RESUMO

As tecnologias móveis, com serviços interconectados vêm se tornando cada vez mais presentes e necessárias na vida das pessoas e especial para necessidades de geolocalização. Este trabalho apresenta o *Medical Unit Finder*, um aplicativo de localização de unidades de atendimento médico para auxiliar, pessoas que necessitam saber a unidade mais próxima, com suas especialidades. Para tanto, definiram-se como metas para o desenvolvimento, as boas práticas da engenharia de *software*, iniciando pela análise das formas alternativas de busca por unidades médicas comumente realizadas pelas pessoas, dando um passo para uma proposta baseada na elicitação e especificação de requisitos do *Medical Unit Finder*. O projeto envolveu uma interface projetada de forma a estender e contribuir com as decisões e atitudes das pessoas no momento de crise e necessidade, informações já armazenadas na base de dados são acessadas inteligentemente e as melhores respostas no formato (mapa) são apresentadas.

Palavras-chaves: *Medical Unit Finder*; Tecnologias de geolocalização; Aplicativos *Mobile*.

ABSTRACT

Mobile technologies, with interconnected services, are becoming increasingly present and necessary in people's lives and for these purposes. This work presents the application that integrates these points, Medical Unit Finder an application for locating medical care units to help people who need to know the nearest unit, with their specialties. To this objective, promising steps for the development of good software engineering practices were defined, starting with the analysis of alternative ways of searching for medical units commonly performed by people, taking a step towards a proposal based on the elicitation and specification of requirements of the Medical Unit Finder. The project involved an interface designed to extend and contribute to people's decisions and attitudes in times of crisis and needs, information already stored in the database is accessed intelligently and the best responses in the format (map) are presented.

Keywords: Medical Unit Finder; Geolocation technologies; Mobile Apps.

LISTA DE FIGURAS

Figura 1 – Tecnologias padrões da <i>web</i> .	5
Figura 2 - Tela resultado da execução de uma API.	7
Figura 3 -Estrutura do projeto.	12
Figura 4 - Tela inicial	13
Figura 5 - Tela de quilometragem.	13
Figura 6 - Mapa para localizar unidades	14
Figura 7 - Informações detalhadas da unidade médica.	14
Figura 8 - API dando erro.	15
Figura 9 - Horário de funcionamento e imagem.	16

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
APP	<i>Application</i>
CSS	<i>Cascading Style Sheets</i>
GPS	<i>global positioning system</i>
GSM	<i>Global System for Mobile</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IOS	<i>iPhone OS</i>
JS	<i>JavaScript</i>
MAC OS	<i>Macintosh Operating System</i>
MIT	<i>Massachusetts Institute of Technology</i>
SUS	<i>Sistema Único de Saúde</i>
UX	<i>User Experience</i>
WI-FI	<i>Wireless Fidelity</i>
WWW	<i>World Wide Web</i>

SUMÁRIO

1.	INTRODUÇÃO	1
1.1.	MOTIVAÇÃO	1
1.1.1.	OBJETIVO	2
2.	PESQUISA DE TECNOLOGIAS	3
2.1	TECNOLOGIAS DE GEOLOCALIZAÇÃO.....	3
2.2	APLICATIVOS MOBILE.....	3
2.2.1	TECNOLOGIAS BASES PARA O APP	4
2.2.2	VANTAGENS DO JAVASCRIPT.....	5
2.2.3	API.....	6
2.3.4	API GOOGLE MAPS.....	7
2.3.4	GOOGLE PLACES API.....	7
2.3.4.1	PLACES SEARCH	8
2.3.4.2	PLACES DETAILS.....	8
2.3.4.3	PLACES PHOTOS.....	8
2.3.4	REACT NATIVE.....	8
2.3.5.1	VANTAGENS DO REACT NATIVE	9
2.3.6	REACT NATIVE EXPO.....	10
2.3.7	REACT NATIVE EXPO PERMISSIONS.....	10
2.3.8	REACT NATIVE EXPO LOCATION	10
2.3.9	REACT NATIVE MAPS.....	10
3.	MEDICAL FINDER UNID: DESENVOLVIMENTO	11
3.1	IDEIAS INICIAIS	11
3.2	ARQUITETURA DO PROTÓTIPO.....	11
3.3	RESULTADOS	13
3.4	SOLUCIONANDO AS INCONSISTÊNCIAS	14
3.4.1	1 ° VERSÃO API	14
3.4.2.	API NOVA	15
4.	CONSIDERAÇÕES FINAIS	17
4.1	TRABALHOS FUTUROS.....	17
5.	REFERÊNCIAS BIBLIOGRÁFICAS	18

1. INTRODUÇÃO

A busca por uma unidade de atendimento médico disponível e pré-preparada, em um momento de emergência ou não, requer das pessoas algumas decisões, como informações prévias, localização mais próxima de especialidade ou não, assertividade da escolha. E para realizar estas buscas, diferentes estratégias são observadas.

Hoje em dia, diferentes tecnologias, em especial as *mobiles*, interconectam serviços, e vem se tornando cada vez mais presentes e diversificadas. São amplas as possibilidades de serviços oferecidos pelos aplicativos móveis, desde pedir uma comida ou um táxi. Não raro de encontrar aplicativos localizadores de pontos comerciais, escolas, parques etc. Alguns exemplos de aplicativos são o 99 (transporte individual), iFood (entrega de comida), Uber (aplicativo de transporte).

O que não é diferente para hospitais e unidade de atendimento médico em geral, onde um aplicativo (App) pode fazer a diferença entre gastar tempo ao buscar pela informação desejada para chegar até a unidade mais próxima, e ter um App para o auxiliar com o menor tempo decorrido para chegar ao destino. Com esse ponto destacado se torna fundamental a agilidade na busca.

1.1. MOTIVAÇÃO

Em um estudo empírico realizado pelos autores deste artigo, viu-se oportuno apresentar na íntegra alguns pontos chaves importantes para um aplicativo de localização especializada de unidade de atendimento médico, com um conceito chave, que a partir de algumas informações responda de forma intuitiva e inteligente na escolha da localização.

Utilizando os conhecimentos obtidos em prática de engenharia de *software* e desenvolvimento de soluções móveis centrou-se esforços em criar um *design* simples e intuitivo, focando muito no fácil uso para todas as pessoas, pois o aplicativo em casos sérios precisa ser usado de uma maneira rápida e sem complicações.

Na realização desta monografia, destaca-se as informações chave para a escolha da localização onde a distância entre o usuário e a unidade é um critério chave a se levar em conta:

- Uma interface de forma a estender e contribuir com as decisões e atitudes das pessoas no momento de crise e necessidade;
- Informações já armazenadas na base de dados, acessíveis intuitiva e inteligentemente e as melhores respostas no formato (mapa),

1.1.1. OBJETIVO

Este trabalho apresenta o *Medical Unit Finder*, um aplicativo de localização de unidades de atendimento médico para auxiliar pessoas que necessitam saber a unidade mais próxima, com suas especialidades.

2. PESQUISA DE TECNOLOGIAS

Para embasar a criação do protótipo viu-se na decisão de utilização de várias tecnologias e o suficiente entendimento destas, é apresentado nesta seção.

2.1 TECNOLOGIAS DE GEOLOCALIZAÇÃO

Em uma definição básica, a geolocalização é um recurso tecnológico que faz o rastreamento de um dispositivo por meio de uma conexão remota. Essa conectividade varia entre três métodos: GPS (Sistema de Posicionamento Geográfico), GSM (Sistema Global para Comunicações Móveis) e *wireless* (via Wi-Fi, por exemplo). (Cronapp, [2020])

Suas características principais são um aparelho receptor para a troca de informações entre torres ou satélites.

Alguns exemplos que utilizam a tecnologia de geolocalização são os radares que ficam em aeroportos para monitorar o tráfego aéreo.

2.2 APLICATIVOS MOBILE

Aplicativos mobile são uma classe de *softwares* desenvolvidos para serem instalados em dispositivos móveis que normalmente são baixados em uma loja *on-line*, como a Google play ou App store.

Suas características principais são a interface intuitiva, fácil de aprender e navegar para que o usuário tenha uma experiência adequada e como ele pode entreter ou facilitar a vida do usuário.

São inúmeras as razões e fatores que contribuíram para o crescimento do uso de tecnologias de aplicativos móveis, segundo Solvus (2021) destacam-se algumas estatísticas e dados do mercado de aplicativos:

1 - Os tablets ainda são um sucesso: 166,7 milhões de usuários em 2016, em 2020 espera-se que esse número cresça para 185 milhões.

2- Cerca de 7,3 bilhões de contratos com operadoras de celulares no mundo inteiro. E mais: até final de 2021, espera-se que esse número atinja 9,1 bilhões.

3 - A geração dos "*millenials*" já é a maior parte da população, 81% deles usam os celulares para fazer compras.

4 - Os mais velhos também usam: a faixa de idade dos 46 aos 54 anos é a que mais cresce o número de usuários de *smartphones*.

5 - A maior preocupação dos usuários é a segurança das transações (30% deles se preocupam com isso).

6 - Segundo um estudo neurocientífico, o nível de *stress* de um atraso, ou demora, no celular é semelhante ao de assistir a um filme de terror.

7 - Em 2015, o comércio de celulares movimentou 115 bilhões de dólares no mundo. Estima-se neste ano, que esse número deverá chegar a 142 bilhões. Isso é um crescimento de 23% em apenas um ano (maior do que o aumento do crescimento de 15% do ano anterior).

8 - Uma maior banda larga não resolve todos os problemas, especialmente em dispositivos móveis.

2.2.1 TECNOLOGIAS BASES PARA O APP

JavaScript é uma linguagem de programação que permite implementar itens complexos em páginas *web*, toda vez que uma página da *web* faz mais do que simplesmente mostrar a informação estática, mostrando conteúdo que se atualiza em um intervalo de tempo, mapas interativos ou gráficos 2D/3D animados etc, pode apostar que o JavaScript provavelmente está envolvido. É a terceira camada das tecnologias padrões da *web*, sendo elas HTML e CSS. (MDN Web Docs, 2021)

JavaScript é uma linguagem de programação que permite criar conteúdo que se atualiza dinamicamente, controlar multimídias, imagens animadas, eventos diversos.

É vasto o que se pode efetuar com algumas linhas de código JavaScript (MDN Web Docs, 2021).

HTML é a linguagem de marcação usada para estruturar e dar significado para o conteúdo *web*. Por exemplo, definindo parágrafos, cabeçalhos, tabelas de conteúdo ou inserindo imagens e vídeos na página (MDN Web Docs, 2021).

CSS é uma linguagem de regras de estilo para conteúdo HTML. Por exemplo, definindo cores de fundo e fontes, e posicionando o conteúdo em múltiplas colunas. A figura 1 mostra o *slogan* de identidade de cada tecnologia (MDN Web Docs, 2021).

Figura 1 – Tecnologias padrões da web



Fonte: Pereira, 2018.

2.2.2 VANTAGENS DO JAVASCRIPT

Os benefícios e pontos que sustentaram a decisão de uso desta tecnologia:

- Alta compatibilidade com plataformas, sistemas e navegadores *web*. (LONGEN, 2019)
- É mais leve e rápida que outras linguagens de programação. (LONGEN, 2019)
- Faz com que as páginas na internet sejam mais dinâmicas e interativas, características essenciais do UX. (LONGEN, 2019)
- Os navegadores interpretam a linguagem por conta própria, tirando a necessidade de usar um compilador. (LONGEN, 2019)
- Erros de programação são mais fáceis de encontrar e de corrigir. (LONGEN, 2019)

- Entre as mais populares, é a linguagem de programação mais fácil de aprender. (LONGEN, 2019)
- Executa comportamentos específicos em uma página, como cliques e efeitos personalizados. (LONGEN, 2019)

O que é ainda mais empolgante é a funcionalidade construída no topo do núcleo da linguagem JavaScript. As APIs (*Application Programming Interfaces* - Interface de Programação de Aplicativos) proveem capacidades extras para usar no código JavaScript (MDN Web Docs, 2021).

2.2.3 API

APIs são conjuntos prontos de blocos de código que permitem que um desenvolvedor implemente programas que seriam difíceis ou impossíveis de implementar. Eles fazem o mesmo para a programação que os *kits* de móveis prontos para a construção de casas - é muito mais fácil pegar os painéis prontos e aparafusá-los para formar uma estante de livros do que para elaborar o *design*, sair e encontrar a madeira, cortar todos os painéis no tamanho e formatos certos, encontrar os parafusos de tamanho correto e depois montá-los para formar uma estante de livros (MDN Web Docs, 2021).

Elas geralmente se formam em duas categorias:

- APIs de navegadores que já vem implementadas no navegador, e são capazes de expor dados do ambiente do computador, ou fazer coisas complexas e úteis. (MDN Web Docs, 2021).
- APIs de terceiros que não estão implementados no navegador automaticamente, e geralmente tem que pegar seu código e informações em algum lugar da *Web* (MDN Web Docs, 2021).

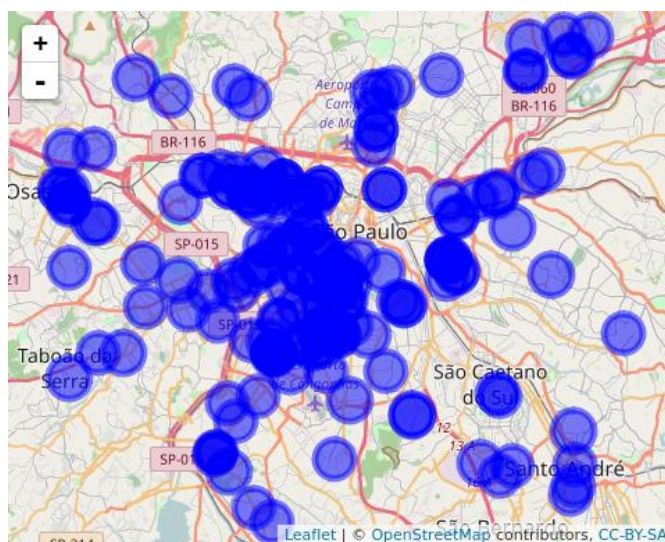
E a decisão de uso de uma API levou os autores a utilizar a API do Google Maps que permite inserir mapas customizados no seu *site* e outras diversas funcionalidades e criar uma API que corrige um problema na API pública do governo.

2.3.4 API GOOGLE MAPS

É um serviço público e gratuito que qualquer pessoa pode usar em seus *sites* e aplicações, ele possui várias API's que podem ser incorporadas ao *site*/aplicação dependendo de cada caso. (DOUGLAS, [2013])

Ela permite usar ou desenhar mapas na aplicação além de fornecer a opção de colocarmos um marcador, dar um *zoom* para ajudar na percepção e inserir pontos no mapa, dando também a possibilidade de personalização para cada um desses aspectos citados. A figura 2 demonstra uma imagem do que a API pode devolver, como exemplo os pontos azuis que são hospitais.

Figura 2 – Tela resultado da execução de uma API



Fonte: Julio, Trecenti 2017.

2.3.4 GOOGLE PLACES API

Consiste em um serviço oferecido pelo Google com o objetivo de retornar informações de lugares, estabelecimentos por meio de requisições HTTP.

As solicitações de resultados disponíveis para a utilização da API, consideradas neste protótipo são descritas:

2.3.4.1 PLACES SEARCH

Integra as solicitações de lugares do Google *Places*, a API tem como intuito retornar uma lista de locais com base na localização do usuário ou em uma palavra-chave de pesquisa.

2.3.4.2 PLACES DETAILS

Retorna informações mais detalhadas sobre um lugar específico, incluindo comentários do usuário, número de telefone e horário de funcionamento do estabelecimento.

2.3.4.3 PLACES PHOTOS

Fornece o acesso à base de dados do Google com milhões de fotos relacionadas a lugares.

Ele permite utilizarmos fotos tiradas por clientes ou donos do estabelecimento e com isso pode-se obter uma grande variedade de fotos de todos os estabelecimentos que o google permite.

2.3.4 REACT NATIVE

Criado pelo Facebook em 2015 sobre a licença MIT, o *React Native* é um *Framework* para desenvolvimento de aplicativos móveis multiplataforma. (PAULA, [2020])

Um *Framework* é um facilitador no desenvolvimento de diversas aplicações e, sem dúvida, sua utilização poupa tempo e custos para quem utiliza, pois de forma mais básica, é um conjunto de bibliotecas utilizadas para criar uma base, onde as aplicações são construídas, um otimizador de recursos. (PAULA, [2020])

Baseado no React, *framework* JS para desenvolvimento *web*, o *React Native* possibilita a criação de aplicações móvel multiplataforma (*Android* e *iOS*) utilizando apenas Javascript. Porém, diferente de outros *frameworks* com esta mesma finalidade (Cordova, por exemplo), todo o código desenvolvido com o *React Native* é convertido para linguagem nativa do sistema operacional, o que torna o app muito mais fluido. (PAULA, [2020])

2.3.5.1 VANTAGENS DO REACT NATIVE

O *React Native* possui diversas características marcantes e fundamentais para sua ampla adoção, como:

- Acessar a interface e os recursos nativos do *Android* e *IOS* utilizando JavaScript; (PAULA, [2020])
- O código produzido é semelhante ao *React* para *Web*; (PAULA, [2020])
- Possui a base de conhecimento compartilhada entre o desenvolvimento *mobile* e *front-end*; (PAULA, [2020])
- Todo código desenvolvido é convertido para a linguagem nativa do sistema operacional; (PAULA, [2020])
- Com o *React Native* consegue-se desenvolver aplicações para *Android* e *IOS* utilizando um código único; (PAULA, [2020])
- Por ser multiplataforma, pode-se desenvolver aplicações com *React Native* utilizando qualquer sistema operacional (*Windows*, *macOS* ou *Linux*). (PAULA, [2020])

2.3.6 REACT NATIVE EXPO

Conjunto de ferramentas criado em torno do *React Native*, fornece um conjunto de ferramentas que simplificam o desenvolvimento e o teste do aplicativo *React Native* e fornece os componentes da interface do usuário e serviços que geralmente estão disponíveis em componentes *React Native* nativos de terceiros.

2.3.7 REACT NATIVE EXPO PERMISSIONS

Permite solicitar várias permissões para acessar os sensores do dispositivo, dados pessoais etc.

Existem vários tipos de permissões, serão citadas algumas como notificação, câmera e contatos, porém o que será utilizado será explicado na próxima seção.

2.3.8 REACT NATIVE EXPO LOCATION

Com a permissão obtida da expo *permissions*, a expo *location* permite a leitura de informações de geolocalização do dispositivo. Para conseguir a localização atual do usuário com sua permissão.

2.3.9 REACT NATIVE MAPS

O *react-native-maps* é um sistema de componentes para mapas que vem com o código nativo da plataforma que precisa ser compilado junto com o *React Native*.

Foi utilizado no código desse protótipo para a renderização de um mapa. No caso está sendo utilizada a API do Google *Maps* e com isso torna-se possível a utilização no aplicativo.

3. MEDICAL FINDER UNID: DESENVOLVIMENTO

Processo de desenvolvimento do protótipo ideias e decisões.

3.1 Ideias Iniciais

As pessoas que usam aplicativos querem ter a facilidade de receber informações de forma rápida e simples. Com isso em mente iniciou-se uma discussão sobre como seria a ‘interface’ do aplicativo, e que precisaria de 3 coisas essenciais para a facilidade do usuário.

1. Mapa: Como forma de ajudar o usuário a localizar as informações desejadas, no momento em que conceder a permissão, um mapa irá aparecer na tela do aparelho mostrando ao usuário informações próximas ao seu redor.

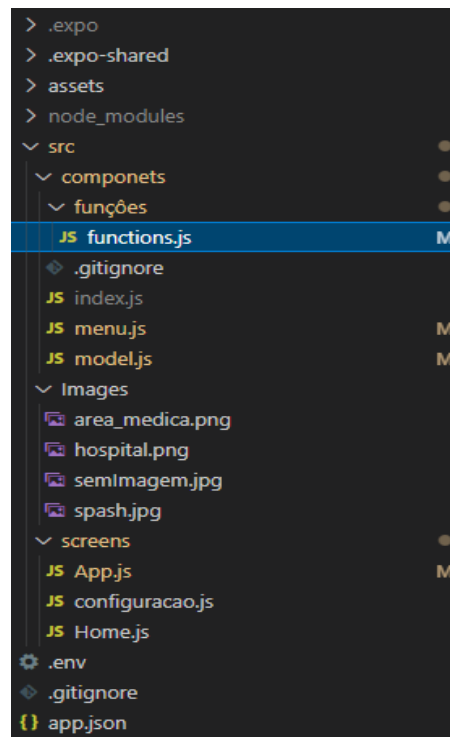
2. Barra de pesquisa: Junto com o mapa uma barra de pesquisa que o usuário pode digitar um endereço manualmente, e o aplicativo mostrara as informações com esses requisitos mais próximos no endereço colocado. O aplicativo é pensado servir para o uso em casos de emergências, mas também, que possa buscar e localizar clínicas especializadas, que o usuário possa necessitar traçando a rota mais próxima da sua localização atual.

3. Detalhamento de rota: Quando o usuário encontrar a informação desejada, poderá tocar no ícone que aparecer no mapa para receber informações sobre o cálculo de distância e de tempo, o nome e bairro da localidade de destino e também mostrará as informações da unidade selecionada.

3.2 Arquitetura do protótipo.

O protótipo está sendo dividido em pasta, onde cada uma delas tem sua própria função na organização do projeto. A figura 3 mostra a estrutura do projeto:

Figura 3 – Estrutura do projeto



Fonte: Os Autores

Dentro da pasta componentes contém o *menu* onde é feita a barra de rolagem para avançar as unidades no mapa e também contém o *model* que traz as informações detalhadas da unidade.

A pasta funções com métodos para o funcionamento do protótipo como a função da localização atual do usuário e requisições para a API.

Há também a pasta imagens, que contém as imagens do aplicativo como o ícone da aplicação e a imagem da tela inicial quando o usuário abre o protótipo.

E a pasta *screens*, onde foi feita a tela do projeto. Por exemplo, a tela *home*, que é a tela inicial do protótipo, onde tem os botões para iniciar o protótipo ou ir para as configurações.

3.3 RESULTADOS

Os resultados obtidos após a conclusão do protótipo são os seguintes: com o aplicativo aberto mostra duas opções iniciar e configurações. A figura 4 mostra a tela inicial do protótipo.

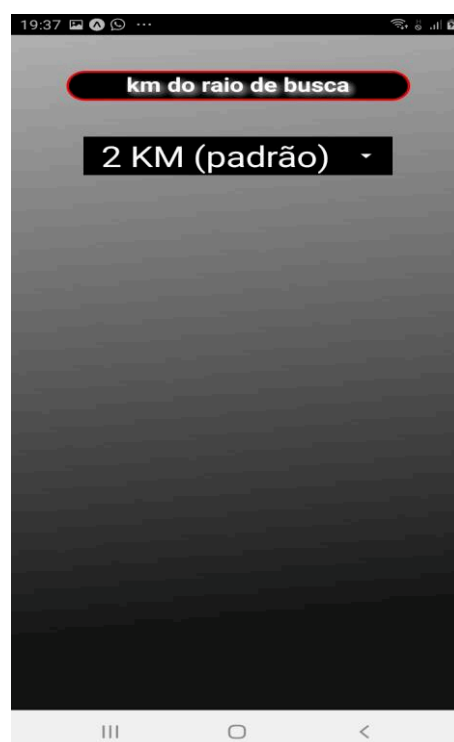
Quando o usuário acessar o botão de configuração, ele será levado para uma tela, onde definirá a quilometragem para o alcance do raio de busca. A figura 5 mostra a tela de quilometragem.

Figura 4 – Tela inicial



Fonte: Os Autores

Figura 5 – Tela de quilometragem

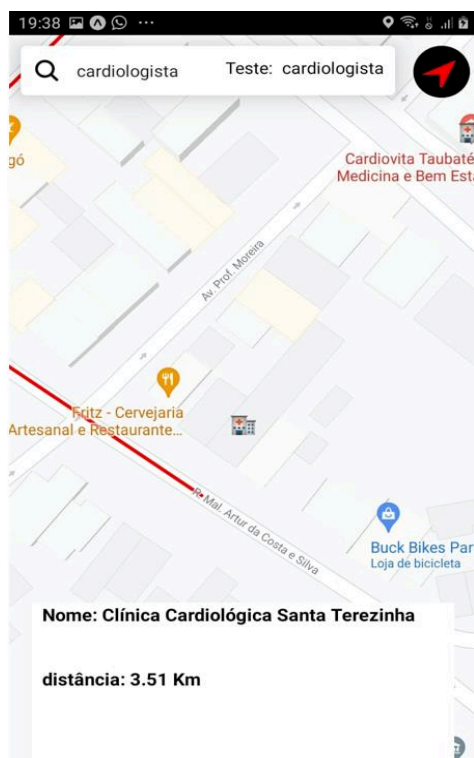


Fonte: Os Autores

Se o usuário acessar o botão início, ele será levado para o mapa onde ele poderá escrever uma especialidade e isso irá levar à busca pelas unidades no mapa. A figura 6 mostra o mapa que o usuário pode localizar as unidades.

Depois que o usuário achar a unidade que procura, poderá ver as informações com mais detalhes apertando o botão i permite o usuário visualizar informações detalhadas sobre a unidade médica. A figura 7 mostra as informações sobre a unidade.

Figura 6 – Mapa para localizar unidades



Fonte: Os Autores

Figura 7 – Informações detalhadas da unidade



Fonte: Os Autores

3.4 SOLUCIONANDO AS INCONSISTÊNCIAS

Soluções para alguns problemas durante o desenvolvimento que foram encontradas enquanto era feito o teste que será mostrado nas seções posteriores.

3.4.1 1º VERSÃO API

Quando iniciou-se a desenvolver o aplicativo, foram feitos vários testes utilizando como base a API do governo mobile-aceite.tcu.gov.br, onde foi possível localizar unidades médicas e filtrar informações uteis para o protótipo. Por exemplo, (Descrição da unidade, Vínculo com os SUS). Enquanto os testes ocorriam no protótipo utilizando essa API, um erro inesperado no servidor onde essa API está hospedada a tornando indisponível para uso na aplicação e por isso foi necessário substituir por uma nova versão. A figura 8 mostra a *site* onde a API não está funcionando mais.

Figura 8 – API DANDO ERRO

```
JBWEB000065: HTTP Status 500 - Request processing failed; nested exception is
javax.persistence.PersistenceException: org.hibernate.exception.GenericJDBCException: Could not open
connection

JBWEB000309: type JBWEB000066: Exception report
JBWEB000068: message Request processing failed; nested exception is javax.persistence.PersistenceException: org.hibernate.exception.GenericJDBCException: Could not open connection
JBWEB000069: description JBWEB000145: The server encountered an internal error that prevented it from fulfilling this request.
JBWEB000070: exception
org.springframework.web.util.NestedServletException: Request processing failed; nested exception is javax.persistence.PersistenceException: org.hibernate.exception.GenericJ
org.springframework.web.servlet.FrameworkServlet.processRequest(FrameworkServlet.java:979)
org.springframework.web.servlet.FrameworkServlet.doGet(FrameworkServlet.java:858)
javax.servlet.http.HttpServlet.service(HttpServlet.java:734)
org.springframework.web.servlet.FrameworkServlet.service(FrameworkServlet.java:843)
javax.servlet.http.HttpServlet.service(HttpServlet.java:847)
br.gov.tcu.arquitetura.servlet.filter.FiltroArquiteturalAbstrato.doFilter(FiltroArquiteturalAbstrato.java:65)
br.gov.tcu.arquitetura.servlet.filter.FiltroArquiteturalAbstrato.doFilter(FiltroArquiteturalAbstrato.java:65)
br.gov.tcu.arquitetura.util.jsf.ie.IEModoCompatibilidadeFilter.doFilterInternal(IEModoCompatibilidadeFilter.java:32)
org.springframework.web.filter.OncePerRequestFilter.doFilter(OncePerRequestFilter.java:187)
br.gov.tcu.arquitetura.servlet.filter.FiltroArquiteturalAbstrato.doFilter(FiltroArquiteturalAbstrato.java:65)
br.gov.tcu.arquitetura.servlet.filter.excecao.ExceptionHandlerFilter.doFilter(ExceptionHandlerFilter.java:59)
br.gov.tcu.arquitetura.servlet.filter.FiltroArquiteturalAbstrato.doFilter(FiltroArquiteturalAbstrato.java:65)
com.ocpssoft.pretty.PrettyFilter.doFilter(PrettyFilter.java:145)
br.gov.tcu.arquitetura.servlet.filter.FiltroArquiteturalAbstrato.doFilter(FiltroArquiteturalAbstrato.java:65)
com.ocpssoft.pretty.PrettyFilter.doFilter(PrettyFilter.java:145)
org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:330)
```

Fonte: Os Autores

Para ajustar as inconsistências para um bom funcionamento do protótipo, a solução encontrada foi utilizar a API google *geocoding*, que tem como função transformar endereços em coordenadas geográficas e também foram utilizadas requisições http em paralelo para otimizar a cada chamada na API google *geocoding*.

3.4.2. API NOVA

Como não foi possível mais utilizar a 1 versão da API do governo decidiu-se utilizar uma alternativa, a API do google *maps* utilizando o serviço *places* e o serviço *places details*, que dão um detalhamento básico de funcionamento de cada unidade médica. Por isso o conceito com essa nova API continua vivo pois ela consegue ainda localizar as unidades médicas, apesar de ter algumas limitações nas descrições das unidades e na descrição das especialidades.

Foi necessário trocar alguns dados que não podiam mais ser acessados, como o convênio do SUS da unidade, detalhamento sobre procedimentos que eram oferecidos pela unidade.

Para substituir isso decidiu-se utilizar certas informações oferecidas pela API do google como indicação sobre o horário de funcionamento e uma imagem da unidade, se disponível. (A figura 9 mostra o horário de funcionamento e a imagem).

Figura 9 – Horário de funcionamento e imagem



Fonte: Os Autores

4. CONSIDERAÇÕES FINAIS

Este trabalho apresenta o Medical Unit Finder, um aplicativo de localização de unidades médicas, com fácil manuseio, favorável a usuários gerais em emergências e de uso geral.

Apesar das dificuldades durante o desenvolvimento como a API do governo, o que proporcionou uma oportunidade de aprendizado na estratégia de desenvolvimento e solução, foi muito gratificante ter passado várias horas do dia buscando e ampliando conhecimento para melhorar e ajustar erros que foram aparecendo durante o processo de construção do protótipo, e por causa disso foi proporcionado aos autores ótimas oportunidades no futuro.

4.1 TRABALHOS FUTUROS

Após a finalização do protótipo, foi concluído que seria possível atualizar e acrescentar uma versão, em que à barra de pesquisa seja alterada para que contenha especialidades médicas predefinidas, projetar um menu com mais detalhes como por exemplo: o vínculo com o SUS que foi perdido por causa da inconsistência na API do governo e incluir os profissionais atuantes na unidade médica.

5. REFERÊNCIAS BIBLIOGRÁFICAS

ANDRADE, Ana Paula. *O que é o React Native?* Treinaweb. 2020. Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-o-react-native>. Acesso em: 25 jun. 2021

BRANDÃO, Bruna. *O que é geolocalização: como essa tecnologia revolucionou o cotidiano de tantas pessoas e empresas?* MAPLINK. 2020. Disponível em: <https://maplink.global/blog/o-que-e-geolocalizacao/>. Acesso em: 25 jun. 2021

CRONAPP. *Geolocalização em aplicativos: o que é e como funciona?* 2020. Disponível em <https://bityli.com/RNWcXj>. Acesso em: 25 jun. 2021

DOUGLAS, Allan. *Introdução à Google Maps API*. DEVMEDIA. 2013. Disponível em: <https://www.devmedia.com.br/introducao-a-google-maps-api/26967>. Acesso em: 25 jun. 2021

LONGEN, Andrei Silveira. *O Que é JavaScript e Como Funciona*. Weblink, 2019. Disponível em: <https://www.weblink.com.br/blog/programacao/o-que-e-javascript/>. Acesso em: 25 jun. 2021.

MDN Web Docs, 2021. *O que é JavaScript?* Disponível em: https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First_steps/What_is_JavaScript. Acesso em: 25 jun. 2021.

PEREIRA, Fernanda. *HTML, CSS e Javascript – Entendendo melhor a base da programação Front-End*. Apexensino, 2018. Disponível em: <<https://apexensino.com.br/base-da-programacao-front-end/>>. Acesso em: 27 de jun. de 2021

SOLVUS®. *22 estatísticas do mercado de aplicativos que todos precisam saber*. Disponível em: <http://solvus.com.br/22-estatisticas-do-mercado-de-aplicativos-que-todos-precisam-saber/>. Acessado em: 20/08/2021.

TRECENTI, *Julio*. *Como utilizar apis do google maps para encontrar o que quiser!*. MUNDO API. 2017. Disponível em: <https://mundoapi.com.br/materias/google-maps-api-2/>. Acesso em: 25 jun. 2021