

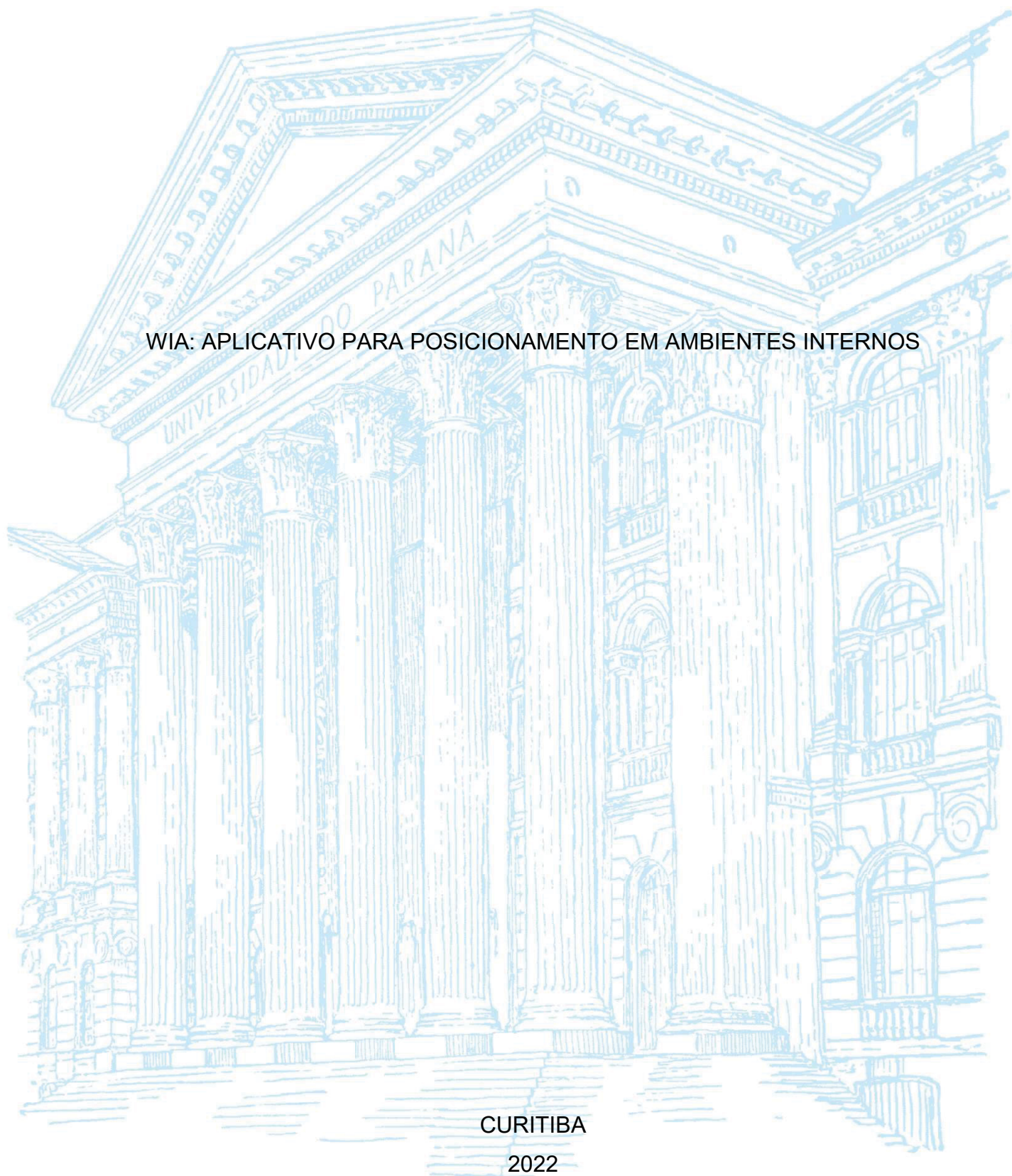
UNIVERSIDADE FEDERAL DO PARANÁ

KLEYTON LUCAS DE SOUZA

WIA: APLICATIVO PARA POSICIONAMENTO EM AMBIENTES INTERNOS

CURITIBA

2022



KLEYTON LUCAS DE SOUZA

WIA: APLICATIVO PARA POSICIONAMENTO EM AMBIENTES INTERNOS

Monografia apresentada ao curso de Pós-Graduação em Desenvolvimento Ágil de Software, Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Especialista em Desenvolvimento Ágil de Software.

Orientadora: Profa. Dra. Rafaela Mantovani Fontana

CURITIBA

2022



MINISTÉRIO DA EDUCAÇÃO  
SETOR DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA  
UNIVERSIDADE FEDERAL DO PARANÁ  
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO  
CURSO DE PÓS-GRADUAÇÃO DESENVOLVIMENTO ÁGIL  
DE SOFTWARE - 40001016375E1

## TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação DESENVOLVIMENTO ÁGIL DE SOFTWARE da Universidade Federal do Paraná foram convocados para realizar a arguição da Monografia de Especialização de **KLEYTON LUCAS DE SOUZA** intitulada: **WIA: APLICATIVO PARA POSICIONAMENTO EM AMBIENTES INTERNOS**, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua aprovação no rito de defesa. A outorga do título de especialista está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 09 de Dezembro de 2022.

RAFAELA MANTOVANI FONTANA  
Presidente da Banca Examinadora

JAIME WOJCIECHOWSKI  
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Para Marinah (Ninah), minha filha amada.

## **AGRADECIMENTOS**

Agradeço a Deus por me conceder minha filha, vê-la crescer me motiva e me orienta, ela sempre será o meu Norte, mesmo em dias dúbios.

Agradeço aos professores, Dra. Rafaela Mantovani Fontana por ter me incentivado a começar e ser capaz de finalizar, e ao Dr. Alexander Robert Kutzke, seu apoio e ajuda foram cruciais nos meus primeiros passos como desenvolvedor.

Agradeço também os professores, Dr. Razer Montaña e Dr. Jaime Wojciechowski, pelo conhecimento relevante transmitido em aulas planejadas e ministradas com maestria. Aulas agradáveis que facilitaram não só absorver como entender o conteúdo proposto.

Aos inúmeros, amigos e pessoas próximas, que ofereceram e forneceram ajuda sem restrições, e aos companheiros que ingressam na mesma jornada e desde então se apoiaram mutuamente.

Não se pode ensinar tudo a alguém, pode-se apenas ajudá-lo a encontrar  
por si mesmo o caminho.  
(Galileu Galilei)

## RESUMO

Na atualidade os celulares e dispositivos eletrônicos, principalmente através da comunicação com satélites, desempenham um papel fundamental em fornecer informações geográficas, como mapas, melhores caminhos ou até mesmo o histórico do fluxo de pessoas em certos locais. Ambientes que antes não dispunham de nenhuma informação geográfica pertinente, hoje a possuem ao toque de um celular. Contudo, essa inovação não está disponível em todos os locais, principalmente, locais fechados, devido a degradação que suas estruturas geram na comunicação com o satélite, diante disso não se pode adotar a mesma abordagem que locais abertos ou externos utilizam. O foco do projeto foi desenvolver uma solução que auxiliasse a mobilidade indoor no Prédio Histórico da Universidade Federal do Paraná, para tanto duas interfaces foram desenvolvidas, ambas utilizando o Flutter. Uma interface web que provê acesso ao mapa, destinos disponíveis e possíveis rotas. E o aplicativo, que além do mapa também fornece a localização via escaneamento de Qr Codes, e em sistemas Android o acompanhamento de rota, em que o aplicativo pode demonstrar graficamente no mapa o deslocamento do usuário utilizando os sensores do celular.

Palavras-chave: Mapa. Indoor. Mobilidade. Flutter. Acompanhamento de rota.

## **ABSTRACT**

Nowadays cell phones and electronic devices, mainly through communication with satellites, play a fundamental role in providing geographic information, such as maps, best paths or even the history of the flow of people in certain places. Environments that previously did not have any relevant geographic information, today have it at the touch of a cell phone. However, this innovation is not available in all places, mainly closed places, due to the degradation that their structures generate in the communication with the satellite, therefore it is not possible to adopt the same approach that open or external places use. The focus of the project was to develop a solution that would help indoor mobility in the Historic Building of the Federal University of Paraná, for which two interfaces were developed, both using Flutter. A web interface that provides access to the map, available destinations and possible routes. And the application, which in addition to the map also provides location via scanning of QR Codes, and on Android systems route tracking, in which the application can graphically demonstrate the user's displacement on the map using the cell phone sensors.

Keywords: Map. Indoor. Mobility. Flutter. Route Tracking.

## LISTA DE FIGURAS

Figura 1 - Poluição visual portas .....	19
Figura 2 - Poluição visual corredores .....	19
Figura 3 - Componentes da Tecnologia MEMS.....	24
Figura 4 - Giroscópio mecanismo.....	25
Figura 5 - Giroscópio atuando .....	25
Figura 6 - Mecanismos do acelerômetro .....	26
Figura 7 - Estrutura pasta mapa tiles .....	28
Figura 8 - Grafo não dirigido.....	30
Figura 9 - Grafo dirigido .....	30
Figura 10 - Grafo ponderado.....	31
Figura 11 - Estrutura grafo Prédio Histórico .....	31
Figura 12 - Configurar Google Maps .....	34
Figura 13 - Interior do Shopping Curitiba – Google Maps .....	35
Figura 14 - Adicionar planta indoor – Indoor GPS.....	36
Figura 15 - Adicionando pontos ao mapa – Indoor GPS .....	36
Figura 16 - Detectar movimentos – Indoor GPS.....	37
Figura 17 - Adicionar planta - HERE .....	38
Figura 18 - Mapeando planta - HERE .....	39
Figura 19 - Erro Detecção Wi-Fi - HERE.....	40
Figura 20 - Planejamento das Sprints .....	43
Figura 21 - Arquitetura WIA.....	52
Figura 22 - Tela inicial aplicativo .....	53
Figura 23 - Tela leitura QR Code .....	54
Figura 24 - Tela Mapa localização .....	55
Figura 25 - Tela mapa escolher destino .....	56
Figura 26 - Tela mapa rotas .....	57
Figura 27 - Tela mapa PDR.....	58
Figura 28 - Tela mapa zoom mínimo.....	59
Figura 29 - Tela mapa zoom máximo .....	60
Figura 30 - Tela mapa rotacionado .....	61
Figura 31 - Tela inicial interface web.....	62
Figura 32 - Tela do mapa web.....	63

Figura 33 - Tela mapa web zoom mínimo .....	64
Figura 34 - Tela mapa web zoom máximo .....	64
Figura 35 - Tela mapa escolher destino .....	65
Figura 36 - Tela mapa web rota .....	66
Figura 37 - Diagrama de caso de uso .....	74
Figura 38 - Diagrama de classes.....	80
Figura 39 - Diagrama de sequência Ver Localização.....	81
Figura 40 - Diagrama de sequência Visualizar Mapa.....	82
Figura 41 - Diagrama de sequência Escolher Destino .....	83
Figura 42 - Diagrama de sequência Acompanhar Rota .....	84

## LISTA DE QUADROS

Quadro 1 - Softwares semelhantes .....	33
--	----

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>16</b>
1.1 PROBLEMA .....	18
1.2 OBJETIVOS .....	19
1.2.1 Objetivo geral .....	19
1.2.2 Objetivos específicos.....	19
1.3 JUSTIFICATIVA .....	20
1.4 ESTRUTURA DO DOCUMENTO.....	21
<b>2 FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>22</b>
2.1 SISTEMA DE POSICIONAMENTO INTERNO.....	22
2.1.1 Sistema de Navegação Inercial.....	23
2.1.2 Sensores .....	24
2.1.3 Navegação Estimada .....	27
2.1.4 Mapas Locais .....	28
2.1.5 Caminho de Menor Custo.....	29
2.1.6 Perímetro Geográfico Virtual .....	32
2.2 SOFTWARES SEMELHANTES .....	33
<b>3 MATERIAIS E MÉTODOS</b> .....	<b>41</b>
3.1 PROCESSO DE ENGENHARIA DE SOFTWARE .....	41
3.2 MODELAGEM DE SOFTWARE.....	42
3.3 PLANO DE SPRINTS.....	43
3.3.1 SPRINT 1 .....	43
3.3.2 SPRINT 2 .....	44
3.3.3 SPRINT 3 .....	44
3.3.4 SPRINT 4 .....	44
3.3.5 SPRINT 5 .....	44
3.4 TECNOLOGIAS UTILIZADAS.....	44
3.4.1 ANDROID STUDIO .....	44
3.4.2 ASTAH COMMUNITY .....	45
3.4.3 DART.....	45
3.4.4 FLUTTER .....	46
3.4.5 GIT .....	46
3.4.6 GITHUB.....	46

3.4.7 GITHUB PAGES .....	47
3.4.8 GOOGLE MAPS.....	47
3.4.9 JAVA .....	47
3.4.10 MICROSOFT TO DO.....	47
3.4.11 MIRO .....	48
3.4.12 OPEN STREET MAPS .....	48
3.4.13 QGIS .....	48
3.4.14 PLUGINS.....	48
3.4.15 VISUAL STUDIO CODE.....	50
3.4.16 HARDWARE .....	50
<b>4 APRESENTAÇÃO DO SISTEMA.....</b>	<b>52</b>
4.1 ARQUITETURA DO SISTEMA.....	52
4.2 FUNCIONALIDADES .....	53
4.2.1 Aplicativo .....	53
4.2.2 Interface WEB .....	62
<b>5 CONSIDERAÇÕES FINAIS.....</b>	<b>67</b>
5.1 RECOMENDAÇÕES PARA TRABALHOS FUTUROS .....	67
<b>REFERÊNCIAS.....</b>	<b>69</b>
<b>APÊNDICE A – DIAGRAMA DE CASOS DE USO .....</b>	<b>74</b>
<b>APÊNDICE B – HISTÓRIAS DE USUÁRIO .....</b>	<b>75</b>
<b>APÊNDICE C – DIAGRAMA DE CLASSES.....</b>	<b>80</b>
<b>APÊNDICE D – DIAGRAMA DE SEQUÊNCIA .....</b>	<b>81</b>

## 1 INTRODUÇÃO

Ao longo da história o homem vem aperfeiçoando maneiras de obter sua localização espacial. No início, a observação do céu era a única disponível, até o surgimento da bússola no século 11. A partir desse momento, novas ferramentas, técnicas, métodos e dispositivos surgiram com esse objetivo (TOMTOM, 2020).

O *wayfinding* é um termo que foi utilizado pela primeira vez no livro “A imagem da Cidade” (LYNCH, 1960), em que o autor descreve sobre os elementos - visuais, táteis e auditivos - que podem ser auxiliares na orientação espacial dentro das cidades, fornecendo pistas sobre localizações e rotas às pessoas.

A técnica visa permitir que o indivíduo seja capaz de saber onde está e também para onde ir. A orientação espacial se dá pela interação com o ambiente e através da informação recebida. A “tomada de decisões” é desenvolvida, resultando na criação do mapa mental do ambiente (LYNCH, 1960; SCARIOT, 2013).

Durante a década de 70, o termo *wayfinding* tornou-se mais expansivo, agregando também o sentido de “orientação intuitiva”, como relata Ribeiro (2009, p. 28), o conceito agora é não só baseado na orientação espacial mas no entendimento que são necessários, para encontrar o caminho, a união dos processos de percepção, cognitivo e o de tomada de decisões.

O aperfeiçoamento do *wayfinding* veio com o surgimento do *design* de sinalização, sendo considerado como uma das disciplinas mais completas do *design*, segundo o livro “Design de Sinalização” (D’AGOSTINI, 2016). Com o objetivo de suprir as demandas de comunicação visual dos ambientes, o design de sinalização, ajusta-se ao distinto público que cada local possui. Para que os locais possam ser mais acessíveis e com informações pertinentes, o design da sinalização precisa ser abrangente e utilizar a comunicação como uma ferramenta multissensorial.

Para tanto, utiliza-se dados de pesquisas relativas ao ambiente e seus usuários, gerando uma análise com foco na melhoria - métodos, processos e componentes - do projeto de comunicação. É fundamental o acompanhamento da evolução de métodos de trabalho e de processos tecnológicos, e se necessário, aplicar ajustes na comunicação do ambiente com seus transeuntes (D’AGOSTINI, 2016).

Locais como, aeroportos, hospitais e shopping centers, dentre outros do seguimento - fechados, complexos e com várias pessoas, geralmente possuem um sistema de *wayfinding* eficaz, adaptado ao seu ambiente, público e cotidiano, pois, nesses ambientes geralmente de alto estresse o sistema contribui para a segurança, proteção e uma sensação de bem-estar (SEGD, 2016).

Atualmente o *wayfinding* encontra-se nos aplicativos de navegação, integrantes do cotidiano de muitas pessoas, provendo um eficiente método de *wayfinding* digital, principalmente em lugares desconhecidos. Entre os aplicativos com esse propósito, tem-se o *Google Maps* que possui mais de 10 bilhões de downloads (PLAY STORE, 2022).

A evolução tecnológica contribuiu preenchendo a lacuna da falta do *wayfinding* físico com a criação do digital, permitindo que as interações com o ambiente e as tomadas de decisões sejam realizadas através de dispositivos móveis.

Contudo alguns locais ainda não possuem nenhum tipo de *wayfinding*, seja ele físico ou digital. A navegação é geralmente disponibilizada via satélite, característica que a torna inviável em ambientes denominados *indoor*. Um ambiente *indoor* é uma referência a áreas fechadas ou cobertas, onde geralmente não se pode ver o céu, enquanto o *outdoor* é seu antagônico, representando os locais abertos.

Nota-se que em alguns ambientes, como o aeroporto do Galeão (PAIVA, 2016), já é adotada uma tratativa híbrida na implementação do *wayfinding*, onde os sistemas – digital e físico – se complementam, unindo atributos para entregar ao usuário uma ótima experiência no deslocamento interno.

Inferese, diante do exposto, que um sistema físico ou estático de *wayfinding* utilizando a própria estrutura do local, provê solidez e acessibilidade, pois entrega informações sobre mobilidade, sem interferência de outros fatores ou a necessidade de alguma ferramenta para tal. Enquanto o sistema digital entrega maior interação, flexibilidade e dinamismo nas informações sobre deslocamentos, pois não é estático, além de uma evidente evolução tecnológica atrelada aos locais que a utilizam.

## 1.1 PROBLEMA

O prédio histórico da UFPR é um ambiente *indoor*, composto por 6 andares, além de um subsolo, contendo um vasto número de salas, corredores e escadas, distribuídos em uma edificação com mais de 16 mil metros quadrados (G1, 2019).

Além de grande e complexa, essa infraestrutura é dinâmica, ou seja, as unidades que a compõem, ocasionalmente mudam suas localizações ou denominações. São efetuados por essas unidades, eventos e atendimentos à comunidade, ocasionais e diários.

Esses fatores somados ao grande público que realiza acessos muitas vezes inadiáveis e com horários definidos, tornam o prédio histórico um ambiente onde um sistema de *wayfinding* é essencial.

Ao se realizar o deslocamento interno, o usuário percebe que não existe nada relativo a *wayfinding* e muitos contam com as instruções dos funcionários da recepção, e de outras pessoas que lá estão ou com sorte informações sazonais aplicadas às paredes.

Como o próprio nome, o prédio histórico se trata de um patrimônio histórico, sua infraestrutura é tombada, culminando no repelimento de um mecanismo de *wayfinding* físico que possa degradar suas características arquitetônicas únicas.

Na tentativa de suprir a deficiência do *wayfinding*, o que se vê são cartazes com informações diversas, mas que no final acabam contribuindo com a poluição visual, pois se degradam rapidamente com o tempo e não têm um padrão de *wayfinding* definido, como pode-se visualizar nas FIGURA 1 e FIGURA 2.

Figura 2 - Poluição visual corredores



Figura 1 - Poluição visual portas



FONTE: O Autor (2022).

Notoriamente o processo que cria, e o serviço que entrega o *wayfinding* digital são complexos. Empresas que vendem esses serviços, importantíssimos para grandes construções como o prédio histórico da UFPR, geralmente não disponibilizam gratuitamente a tecnologia (RITCHER *et al.*, 2022).

O acesso a um local desconhecido e sem informações pertinentes sobre como se deslocar, entrega insegurança, frustração e uma péssima experiência ao visitante, mesmo ele entendendo a dinâmica do local em querer preservar sua história. A deficiência em prover uma orientação espacial adequada destaca-se, mais ainda, quando sem conhecimento prévio da infraestrutura, um indivíduo realiza uma visita ao local.

## 1.2 OBJETIVOS

### 1.2.1 Objetivo geral

Desenvolver um aplicativo que auxilie a localização e o deslocamento dentro do andar térreo do prédio histórico da UFPR.

### 1.2.2 Objetivos específicos

Os objetivos específicos do trabalho são:

- a. Mostrar a localização atual do usuário sobre o mapa do local.
- b. Listar os destinos disponíveis do mapa.
- c. Criar uma rota entre a localização atual e um destino.
- d. Acompanhar o deslocamento do usuário.
- e. Funcionar de modo autônomo, sem sinal de Internet ou GPS (*Global Position System*).

### 1.3 JUSTIFICATIVA

A opção de criar um aplicativo para auxiliar a mobilidade dentro de um ambiente *indoor*, como o do prédio histórico, pode diminuir a poluição visual que as informações fixadas nas paredes causam. Por meio do aplicativo preserva-se o visual do local, e conseqüentemente a história atrelada a ele. Permitindo ainda o envio e a gerência de informações pertinentes de uma maneira eficaz e padronizada.

Segundo o estudo GENTE GLOBO (2021), os espaços públicos, como o prédio histórico da UFPR, são os locais onde as classes A, B e C com acesso à internet mais relatam frequentar.

Portanto uma grande parte dos frequentadores do campus Santos Andrade já possui um dispositivo móvel e acesso à internet, que seria usado apenas para baixar o aplicativo. Viabiliza-se assim, em lugar onde o wayfinding físico é evitado ou aplicado de maneira controversa, um sistema digital de *wayfinding* para a maioria dos usuários.

O aplicativo proposto adota o *wayfinding* digital, assim como um possível registro dos dados geoespaciais dos usuários, dois atributos que geralmente não são disponibilizados gratuitamente pelos aplicativos existentes.

O acesso aos dados geoespaciais podem conceder inúmeros benefícios, como no caso do prédio histórico: portaria digital, ao se registrar a entrada, saída e deslocamento internos; além de informações relevantes ao papel – turista, aluno, visitante ou calouro – do usuário, que podem ser fornecidas de forma personalizada e localizada, conforme o deslocamento ocorre.

Há de se lembrar sobre a pandemia de COVID-19 que afligiu todo o mundo. Uma ferramenta assim disponível na época, facilitaria e muito o distanciamento social, tanto quanto facilitaria o rastreamento de casos suspeitos.

Não menos importante, o aplicativo também agrega valor à instituição, transmitindo inovação e respeito ao usuário, refletido na preocupação com a experiência positiva ao se visitar a UFPR.

#### 1.4 ESTRUTURA DO DOCUMENTO

Este documento está estruturado da seguinte forma: o Capítulo 2 narra a fundamentação teórica sobre o sistema *indoor positioning system* e suas características mais relevantes perante o sistema proposto.

No Capítulo 3 os materiais e métodos, usados no desenvolvimento, são descritos. Detalhes como, linguagem de programação, ferramentas e técnicas são encontrados aqui. O Capítulo 4 tem a finalidade de demonstrar as telas do sistema e suas funcionalidades. E por fim o Capítulo 5, contém as considerações finais e propostas de trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo visa apresentar os elementos, do *IPS (Indoor Positioning System – Sistema de Posicionamento Interno)*, utilizados no projeto e demonstrar sistemas semelhantes ao WIA. As escolhas dos elementos na concepção do WIA, foram tomadas visando-se três objetivos: rapidez, baixo custo e autonomia.

Nesse sentido, a abordagem do aplicativo ao *IPS* se formou, levando-se em conta criar um sistema sem custos adicionais com equipamentos adjacentes ou infraestrutura do local e de rápida implementação. O aplicativo deve ser funcional mesmo sem o uso de sinais externos, ou seja, mesmo *off-line* ou no modo avião, o sistema não vai mudar seu comportamento.

### 2.1 SISTEMA DE POSICIONAMENTO INTERNO

Um sistema de posicionamento interno, ou da sigla em inglês *IPS – Indoor Position System* -, é pouco conhecido. Geralmente o sistema mais conhecido de localização, é o *GPS (Global Position System – Sistema de Posicionamento Global)*.

O *GPS* é geralmente o sistema de localização via satélite mais difundido. Entretanto, existem outros sistemas, como o *GLONASS (Global Navigation Satellite System – Sistema de Navegação Global por Satélite)*, ambos utilizam satélites para geolocalização em espaços abertos, ou onde o satélite possa enviar e receber dados (LAGO *et al.*, 2002).

Um *IPS (Indoor Position System – Sistema de posicionamento interno)* é comumente utilizado onde sistemas de navegação via satélite não têm precisão ou simplesmente não funcionam. Isso geralmente ocorre em locais fechados, devido a interferência da infraestrutura na comunicação com o satélite (CHELLY; SAMAMA, 2009).

Uma característica intrínseca a ambientes fechados, é possuir muitos obstáculos fixos e móveis, como pessoas, elevadores, mesas e escadas que causam reflexões e absorção do sinal. Foram gerados muitos esforços nos últimos anos para encontrar novas soluções para *IPS*, dado que pessoas passam cada vez mais tempo em ambientes *indoor* (SILVA, *et al.*, 2019).

O *IPS* pode ser composto por inúmeras técnicas, ferramentas, métodos ou dispositivos, o que denota ao sistema uma alta complexidade e inúmeras opções de implementação. Usualmente, a precisão da localização, e muitas vezes o preço da implementação, é relativo à quantidade de elementos que atuam em rede trocando informações sobre localizações (CHELLY; SAMAMA, 2009).

Um sistema de posicionamento interno pode ter muitas aplicações, como por exemplo: orientação de bombeiros, monitoramento de pessoas em liberdade condicional, orientação de pessoas em *shoppings*, rastreamento pessoal ou de ativos, orientação de pessoas com problemas de mobilidade ou não, e primeiros socorros para equipes de resgate ou emergências (SANTOS, 2018).

Uma série de estudos têm classificado e comparado as tecnologias de posicionamento internas em três principais categorias: propagação da onda, baseada em imagens e navegação inercial (SANTOS, 2018). O presente projeto utiliza somente o sistema de navegação inercial na detecção dos deslocamentos. Essa abordagem atribui autonomia, baixo custo e relativa velocidade na implementação, porém sacrifica-se a precisão.

### 2.1.1 Sistema de Navegação Inercial

O trabalho fará uso do conceito de sistema de navegação inercial (*INS - Inertial Navigation System*), afim de determinar o deslocamento do dispositivo. Um sistema de navegação inercial é denominado assim por utilizar sensores inerciais. A fim de melhorar a precisão pode-se utilizar outros sensores que estejam disponíveis no dispositivo, como bússola e altímetro por exemplo.

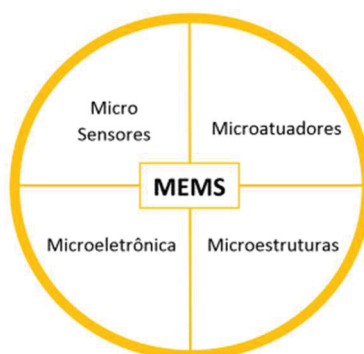
Segundo Christ e Wernli (2014), *INS* são sistemas que podem calcular posições, usando um ponto de referência ou coordenadas absolutas. Utilizando-se sensores inerciais, pode-se obter aceleração, velocidade, posição e ângulo do objeto em que os sensores atuam. Da união dos dados fornecidos pelos sensores inerciais, obtém-se um sistema de Unidade de Medida Inercial (*IMU – Inertial Measurement Unit*), muito utilizado para estabilização, determinação de posicionamento e controle de voo (EMBARCADOS, 2015).

### 2.1.2 Sensores

Basicamente, os sensores inerciais podem perceber as forças que atuam sobre um determinado objeto, mensurando em quais direções ou eixos ele foi movido. Esses sensores são do tipo MEMS (*Micro Electro Mechanical Systems* – Sistemas Micro Eletromecânicos), uma classe de dispositivo que combina eletrônica e mecânica em escala micrométrica (EMBARCADOS, 2015).

Saffo (2022), classifica a tecnologia MEMS como a essência da computação analógica. São sensores capazes de sentir e reagir a estímulos externos naturais onde uma combinação de conceitos multidisciplinares atuam para produzir sistemas integrados que responderão por funções de sensoriamento, controle e atuação. A Figura 3 demonstra a arquitetura de um componente da tecnologia MEMS.

Figura 3 - Componentes da Tecnologia MEMS



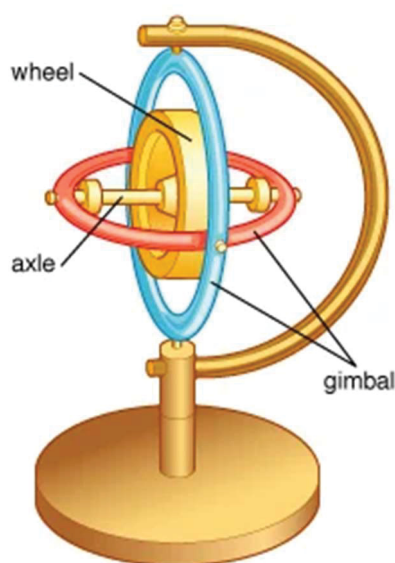
Fonte: <https://www.embarcados.com.br/mems/>

Entre os sensores do tipo MEMS, além dos inerciais - giroscópio e acelerômetro - explicados a seguir, o magnetômetro também será utilizado. Ressalta-se que muitos celulares atualmente possuem o altímetro, também chamado de barômetro. Este é outro sensor auxiliar importante na navegação inercial, usado para determinar qual andar ou piso o dispositivo se encontra. No entanto, como este projeto foca somente em um andar do prédio histórico, não é necessário o uso do barômetro com a função de altímetro.

## Giroscópio

Esse dispositivo, ou sensor, pode calcular a velocidade angular em torno de um eixo referência. Ou seja, pode-se medir a velocidade de rotação em um centro de rotação (ANAC, 2022). A FIGURA 4 apresenta um conceito de giroscópio.

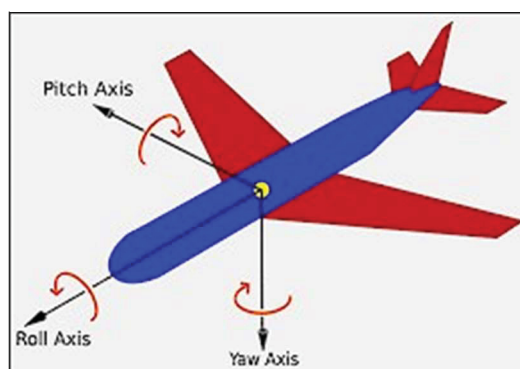
Figura 4 - Giroscópio mecanismo



Fonte: <https://www.britannica.com/technology/gyroscope/images-videos#/media/1/250498/144456>

O sistema de Unidade de Referência Inercial (*IRU - Inertial Reference Unit*) é composto por três eixos (EMBARCADOS, 2015). Sendo eles, o eixo transversal ou *Pitch*, o longitudinal ou *Roll*, e o vertical ou *Yaw*, como é demonstrado no avião da FIGURA 5.

Figura 5 - Giroscópio atuando



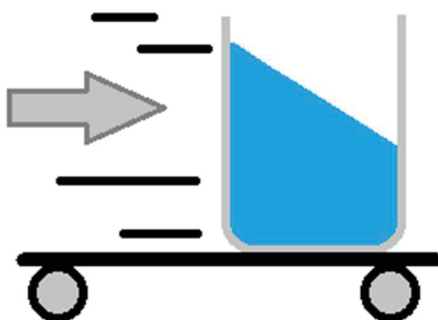
Fonte: [https://pt.wikipedia.org/wiki/Ficheiro:Yaw\\_Axis\\_Corrected.svg](https://pt.wikipedia.org/wiki/Ficheiro:Yaw_Axis_Corrected.svg)

## Acelerômetro

Como o nome sugere, pode detectar a aceleração do celular (ANAC, 2022). Empregado em muitos contextos, como em um *Hard Disk*, é usado para detectar a brusca aceleração que ocorre antes da queda do dispositivo. Assim, pode travar as cabeças de gravação, antes que a colisão ocorra e, possivelmente, a perda de dados. Esse exemplo foi usado, comercialmente, pela primeira vez no lançamento de um notebook pela IBM em 2003 (LIAO; ZHAO, 2005).

Os acelerômetros podem detectar e mensurar a aceleração<sup>1</sup> por caminhos diferentes, desde o mais complexo como por piezeletricidade<sup>2</sup>. Como de maneira mais simples e ilustrado na Figura 6. Utilizando-se um recipiente com água e realizando-se um movimento, percebe-se que haverá interferência no ângulo do nível da água. A partir desse ângulo calcula-se a aceleração exercida sobre o objeto.

Figura 6 - Mecanismos do acelerômetro



FONTE: O Autor (2022).

## Magnetômetro

Esse sensor no celular realiza o mesmo trabalho da bússola, que mede a intensidade, direção e sentido dos campos magnéticos próximos, com o objetivo de entregar a direção dos pontos cardeais (INPE, 2022).

Originalmente, o *INS* pode ser formado sem a bússola, mas o incremento desse sensor, quando disponível, pode acrescentar mais precisão ao sistema.

<sup>1</sup> A aceleração mede a mudança da velocidade em relação ao tempo. Portanto, pode-se afirmar que aceleração é a taxa de variação temporal da velocidade de um móvel (BRASIL ESCOLA, 2022) .

<sup>2</sup> Conjunto de fenômenos elétricos produzidos por pressões ou deformações exercidas sobre certos corpos (PRIBERAM, 2022).

### 2.1.3 Navegação Estimada

Essa técnica, conhecida como *Dead Reckoning (DR)* ou Navegação estimada, era comumente utilizada pelos marinheiros antes do desenvolvimento do *GPS* (EDELKAMP; SCHRÖDL, 2012), ela visa determinar a localização atual a partir da última localização conhecida (RANDELL, *et al.*, 2003).

Aliada a outras técnicas, o *dead reckoning* é utilizado nos mais avançados sistemas de localização *indoor*, como demonstram os estudos realizados por Lymberopoulos et al. (2015).

Já a técnica derivada, denominada *PDR (Pedestrian Dead Reckoning – Navegação estimada do pedestre)*, tem o mesmo objetivo da *DR*, porém foca no deslocamento a partir da ação de caminhar. Ela detecta, ao realizar a leitura e interpretação dos sensores inerciais, o início e o fim do ato de caminhar, estimando a distância e a direção que os passos foram efetuados. O *PDR* tem como principal fundamento a *IMU* (SANTOS, 2018).

A leitura dos sensores – magnetômetro, giroscópio e acelerômetro – emprega uma tática chamada de fusão de sensores, objetivando diminuir erros de leitura, em que os dados de um sensor podem calibrar os dados de outro quando confrontados. Esses dados vão integrar equações matemáticas que aplicam conceitos de álgebra linear, assim como filtros para suavizar ruídos, novamente, nas leituras emitidas pelos sensores. Todo esse processo visa melhorar a estimativa de direção, o comprimento, início e fim de cada sequência de passos executada (COUSSEAU, 2016).

Apesar dos métodos para melhorar a precisão acima descritos, um sistema unicamente utilizando *PDR*, em relação a precisão fica aquém do esperado. Esse problema é causado principalmente por ruídos e erros acumulados nas leituras, que são afetadas pela infraestrutura característica que os locais fechados possuem, sobretudo em sensores de baixa qualidade (SANTOS, 2018).

Sistemas de localização *indoor* modernos conciliam alguma outra técnica de posicionamento ao *PDR*, como por exemplo os *landmarks* utilizados pelo *UnLoc* (COUSSEAU, 2016).

O Sistema *UnLoc* (WANG et al., 2012), *Unsupervised Localization – Localização não supervisionada*, propõe a utilização de *landmarks* em locais onde é possível detectar padrões de sinais. Um elevador ou outra estrutura similar gera uma

assinatura distinta nos sinais captados pelos sensores, inferindo-se que o usuário está localizado próximo a essa *landmark*.

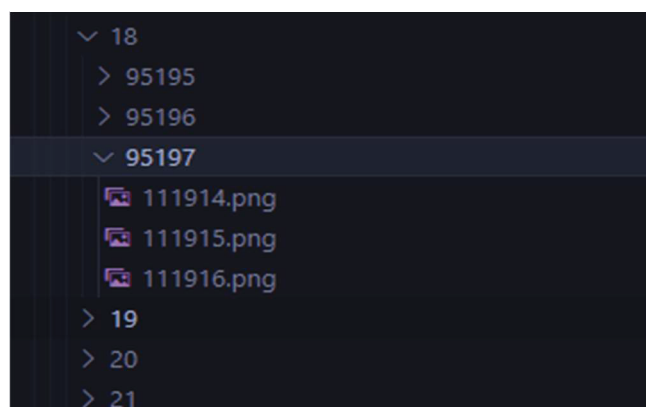
Integrando-se as duas técnicas, obtém-se um sistema híbrido mais preciso e robusto, onde os landmarks darão suporte as calibrações e eventuais erros do *PDR*. Neste projeto, utilizou-se o *PDR*, sem outras técnicas, devido ao tempo e a complexidade que elas demandam.

#### 2.1.4 Mapas Locais

Os Mapas locais são armazenados em blocos, e conhecidos como *Graphical Map Tiles* ou Blocos de mapas gráficos. Os blocos são chamados de *tile*, e cada um representa uma área quadrada do mapa, onde geralmente o padrão é de 256 x 256 *pixels*. Esses *tiles* são armazenados numa sequência estruturada, que permite a montagem dos blocos de forma dinâmica posteriormente.

O esquema dos mapas locais consiste em uma hierarquia de pastas começando pelo *zoom*, quanto maior o *zoom* mais *tiles* a pasta possui. Os *tiles* que formam o mapa relativo ao *zoom* atual, são ordenados numa sequência coordenada de coluna x linha. A FIGURA 7 demonstra essa estrutura de pastas onde se tem a opção dos seguintes *zooms*: 18, 19, 20 e 21.

Figura 7 - Estrutura pasta mapa tiles



FONTE: O Autor (2022).

O primeiro passo para se criar um mapa local, é realizar o processo que vai gerar uma *raster image*, que é o georreferenciamento da imagem do mapa desejado. Basicamente as *raster images* são matrizes de pixel cujos valores representam os locais, através de coordenadas, do espaço interno que essa imagem representa. Ao

se realizar o georreferenciamento da imagem de uma planta baixa, por exemplo, obtém-se uma *raster image* (SITUM, 2022).

A partir da *raster image*, o segundo passo é definir qual o método será empregado para manipular o arquivo georreferenciado criado. Define-se então as configurações para salvamento. O mapa pode ser salvo em certas extensões ou diretamente extraído. A extração irá gerar a estrutura de arquivos descrita no primeiro e segundo parágrafos, geralmente transcrita da seguinte forma “.../{z}/{x}/{y}.png”.

Paralelamente ao *tile map*, existe o padrão *vector tiles*, que também armazena um mapa, mas na forma de dados geométricos. Essa solução apresenta muito mais vantagens sobre o *tile map* (RITCHER *et al.*, 2022). A técnica não utiliza imagens, apenas texto para renderizar os mapas. Contudo, ainda é uma solução recente, e incompatível com outros elementos que integram o projeto.

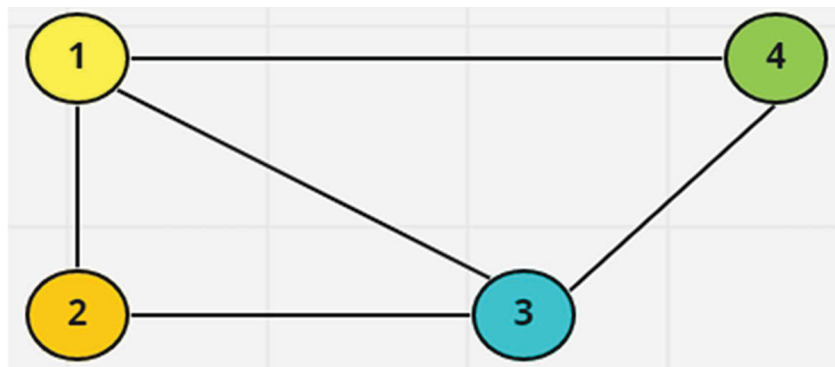
### 2.1.5 Caminho de Menor Custo

O Algoritmo de *Dijkstra* atendeu ao propósito de calcular e encontrar o caminho de menor custo. Esse cálculo envolve conceitos de estruturas de dados usadas para representar conexões entre pares de elementos, conhecida como grafos e apresentada a seguir, assim como posteriormente o Algoritmo de *Dijkstra* (FREECODECAMP, 2022).

- Grafos

Na estrutura do grafo não dirigido, apresentada na FIGURA 8, os círculos coloridos de um a quatro, representam os nós, ou vértices. As conexões entre os nós são chamadas de arestas.

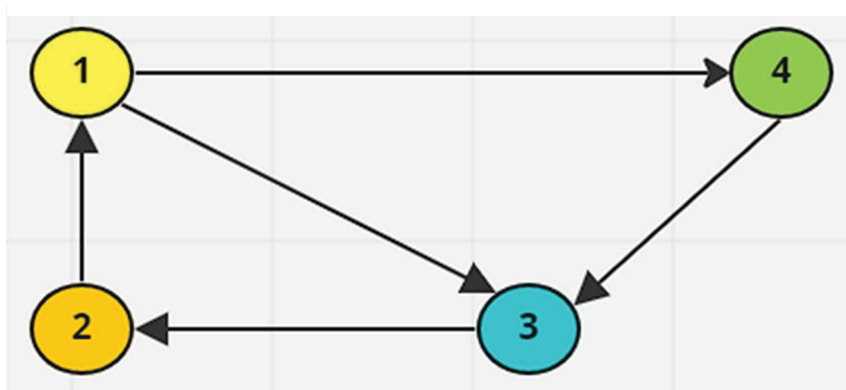
Figura 8 - Grafo não dirigido



FONTE: O Autor (2022).

Os grafos não dirigidos possuem arestas de duplo sentido, ou seja, a aresta liga os pontos em ambas as direções. Já o grafo dirigido tem suas arestas em forma de seta, essa seta indica que o caminho só pode ser percorrido na direção da seta, a FIGURA 9 representa um grafo dirigido (DIAS, 2022).

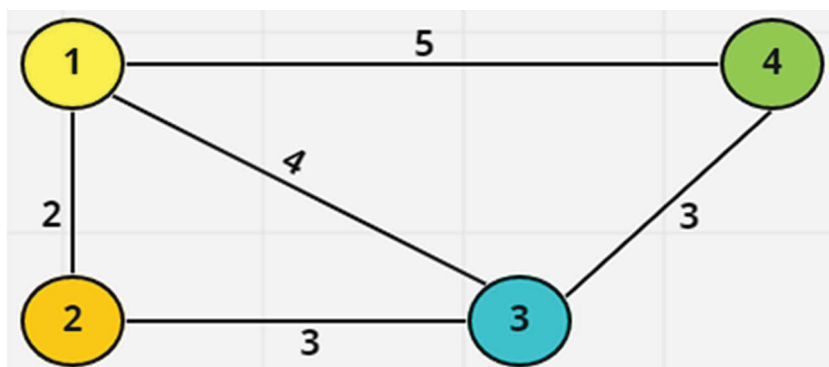
Figura 9 - Grafo dirigido



FONTE: O Autor (2022).

O grafo ponderado possui em suas arestas uma propriedade chamada peso ou custo. Essa propriedade atribui a aresta um número, que geralmente é, referente ao tempo de transposição da aresta de um ponto ao outro, quanto maior o número, maior é o tempo, a FIGURA 10 ilustra essa estrutura (DIAS, 2022).

Figura 10 - Grafo ponderado

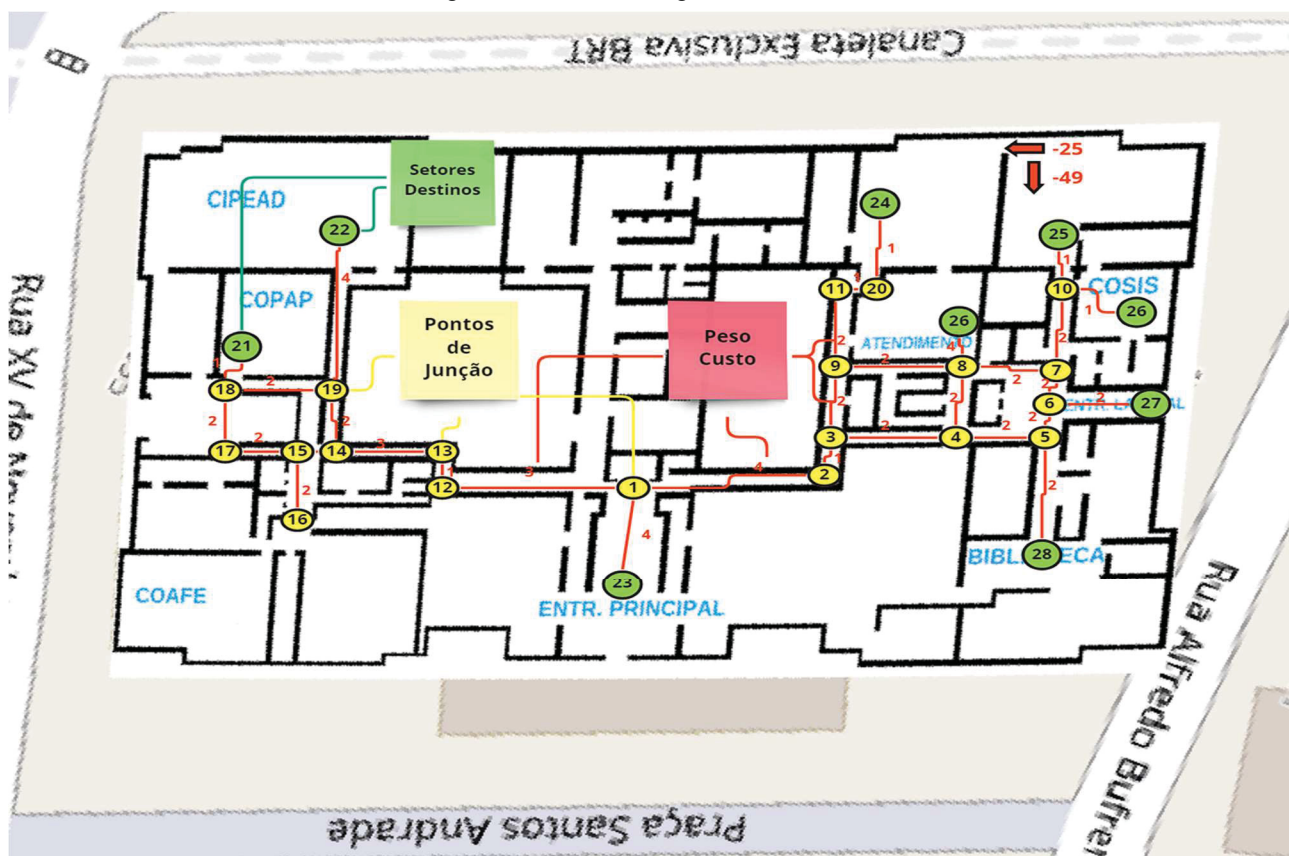


FONTE: O Autor (2022).

Uma observação importante sobre o peso das arestas, é que em um ambiente real, ele nem sempre é associado à distância, mas sim ao tempo ou outra característica importante que modele a conexão da aresta (DIAS, 2022).

A estrutura do grafo ponderado foi criada levando-se em conta a estrutura do prédio histórico, onde os nós podem representar setores ou pontos de intersecção, a FIGURA 11 pode demonstrar o esboço da estrutura do grafo criado.

Figura 11 - Estrutura grafo Prédio Histórico



FONTE: O Autor (2022).

- Algoritmo de *Dijkstra*

O algoritmo para encontrar o caminho de menor custo entre nós de um grafo foi criado e publicado pelo Dr. *Edsger W. Dijkstra*, um cientista da computação e engenheiro de softwares holandês, em 1959 em um artigo de três páginas intitulado “*A note on two problems in connexion with graphs*” (Uma nota sobre dois problemas em conexão com grafos) (DIAS, 2022). O algoritmo pode encontrar o caminho de menor custo entre um nó de origem e todos os outros nós do grafo e, por isso, é muito utilizado em dispositivos de *GPS* (DIAS, 2022).

Para isso, a partir da escolha do nó de origem o algoritmo realiza o seguinte processo para determinar o caminho de menor custo:

- Reconhece e atualiza, caso encontre um caminho menor, as distâncias de cada nó até o nó de origem;
- Ao encontrar o caminho de menor custo entre o nó de origem e outro nó, esse nó é marcado como visitado e adicionado ao caminho;
- O processo finaliza quando todos os nós do grafo forem adicionados ao caminho do nó de origem. Assim, o nó de origem possui um caminho de menor custo a todos os outros nós do grafo.

O algoritmo utiliza grafos ponderados e soma as arestas que são necessárias na conexão dos nós, obtendo assim o custo ou peso dos caminhos, e conseqüentemente o caminho de menor custo (DIAS, 2022).

#### 2.1.6 Perímetro Geográfico Virtual

O perímetro geográfico virtual (*geofence*) é criado em torno da área física do local, possibilitando interagir de forma personalizada ao se adentrar nessa área, visualizar a área que o local possui, acessos e saídas, mensurar o fluxo do local, bem como o tempo de permanência (ARCGIS, 2022).

Usualmente utiliza-se um polígono para demarcar áreas ou setores de interesse, chamados de *Point of Interest (POI)*. O polígono é criado a partir das coordenadas que cercam os limites da área. Essas áreas podem representar vários elementos, como salas, escadas, corredores ou até mesmo a edificação toda.

Diante da dinâmica dos mapas internos, perímetros também são estabelecidos ao se criar um mapa interno, definindo-se assim qual a sua abrangência

pois diferente de um mapa externo (*outdoor*), que geralmente representa grandes locais, ele tem tamanho reduzido à infraestrutura que representa.

## 2.2 SOFTWARES SEMELHANTES

Um software de navegação *indoor* precisa passar por um processo adaptativo, que molde o software aos atributos distintos que cada local possui. Além da configuração prévia, conhecimento em algumas áreas é mandatório, por esse motivo, não existem muitas opções disponíveis e sem custos.

A grande maioria dos aplicativos é monetizada de alguma forma ou liberado com restrições. Com o objetivo de analisar as opções disponíveis, foram escolhidos três *softwares*: *Google Maps*, *Indoor GPS* e o *HERE*. Todos têm como finalidade auxiliar na navegação *indoor*.

Um comparativo entre os *softwares* é demonstrado no QUADRO 1, elencando opções relevantes para navegação *indoor*.

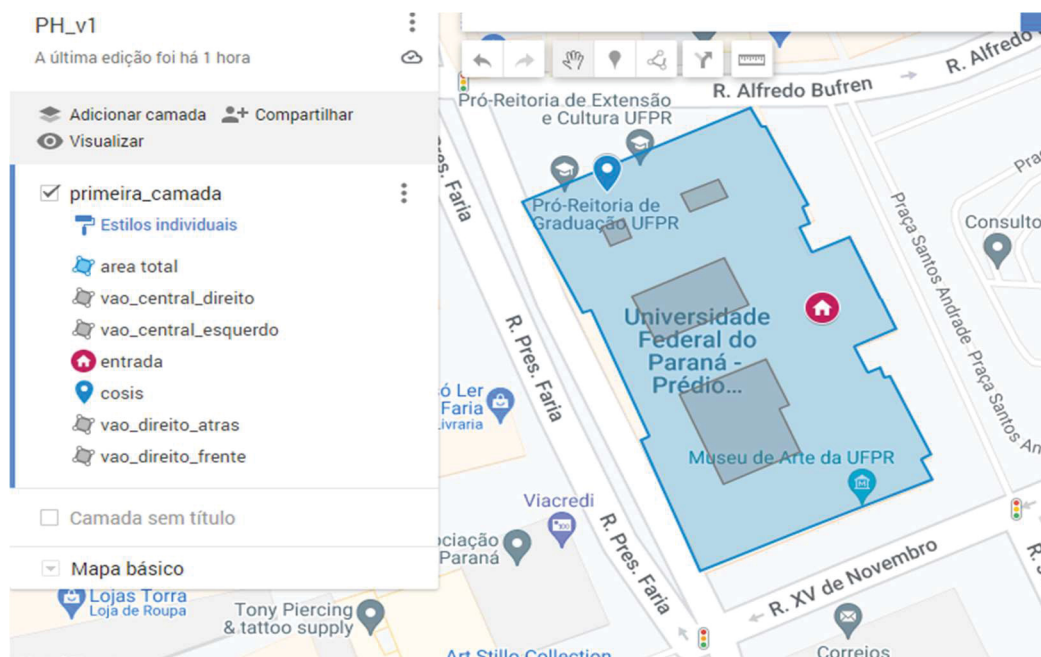
Quadro 1 - Softwares semelhantes

	WIA	Maps	HERE	GPS INDOOR
<i>Indoor</i>	✓	✓	✓	✓
<b>Autônomo</b>	✓	✗	✗	✗
<b>Totalmente gratuito</b>	✓	✗	✗	✗
<b>Pronto para uso</b>	✓	✗	✗	✗
<b>Aceita Andares</b>	✓	✓	✓	✗
<b>Cria Rotas</b>	✓	✓	✓	✗

FONTE: O Autor (2022).

O *Google Maps* já disponibiliza, além do deslocamento *outdoor*, também agora o *indoor*. A opção de navegação indoor pode ser utilizada quando a planta e/ou detalhes do local foram previamente configurados e armazenados no Google, como demonstra a FIGURA 12.

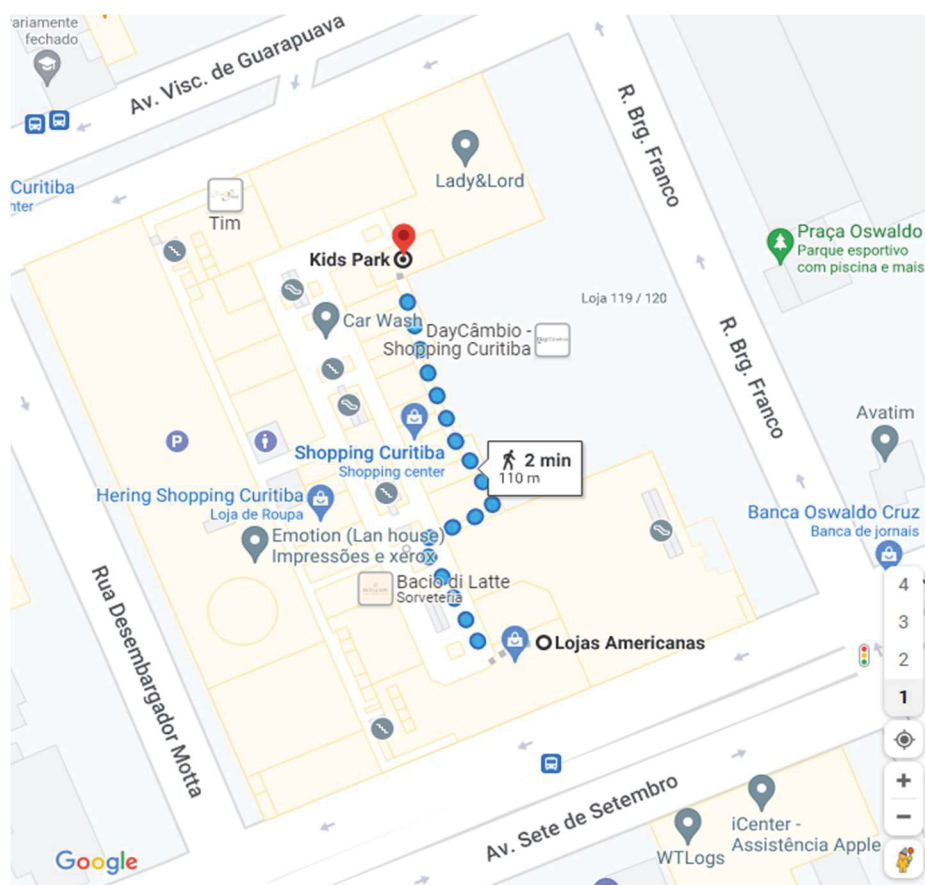
Figura 12 - Configurar Google Maps



FONTE: O Autor (2022).

A planta fica associada à sua conta do *Google*, porém é pública, pode ser exportada em vários formatos e até compartilhada. São necessárias várias configurações adjacentes para a planta atender a requisitos mínimos de usabilidade. Antes de ser disponibilizada, ela precisa ser aprovada pelo *Google*. Alguns ambientes já efetuaram essa configuração. Ao se aplicar um *zoom* sobre um local que já tenha sua infraestrutura configurada no *Google Maps*, os detalhes e ferramentas para se iniciar a navegação *indoor* aparecerão, conforme mostra a FIGURA 13, que mostra o interior de um shopping center.

Figura 13 - Interior do Shopping Curitiba – Google Maps



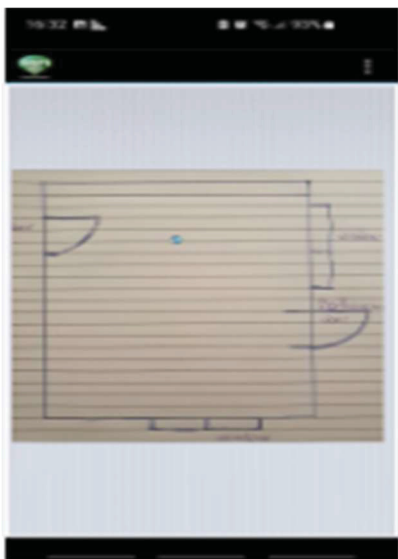
FONTE: O Autor (2022).

Uma característica que eventualmente destaca-se no modo *indoor*, diferentemente do modo *outdoor*, é a opção de seleção do piso ou andar, obviamente quando a estrutura possui esse atributo. Os dois modos, *outdoor* ou *indoor*, funcionam na mesma plataforma, o que define o modo atuante é o local do mapa em que a interação ocorre.

Para utilizar o *Google Maps* é necessário ter acesso a *internet*, assim como uma conta no *Google*, tanto para uso do aplicativo, quanto para criar uma solução utilizando as diversas ferramentas fornecidas. Mas para tal, um sistema de bilhetagem (cobrança por uso), será aplicado. O sistema é baseado na quantidade de usuários que irão utilizar o aplicativo e conseqüentemente os serviços de localização (GOOGLE MAPS BILLING, 2022).

O aplicativo *Indoor GPS*, disponibiliza uma maneira menos complicada na tratativa de inserção da planta do local, apesar de estar disponível apenas para *Android* como demonstra a FIGURA 14.

Figura 14 - Adicionar planta indoor – Indoor GPS



FONTE: O Autor (2022).

Ao adicionar a planta ou desenho relativo, insere-se em seguida um ponto inicial que deve ser referente a sua localização real perante a imagem inserida. Após isso, deve-se andar pelo local e ir adicionando outros pontos. O aplicativo irá traçar uma rota entre o ponto inicial, os pontos subsequentes e o ponto final, estes pontos irão determinar a escala entre o tamanho da imagem adicionada e do mundo real. A FIGURA 15 ilustra a adição dos pontos.

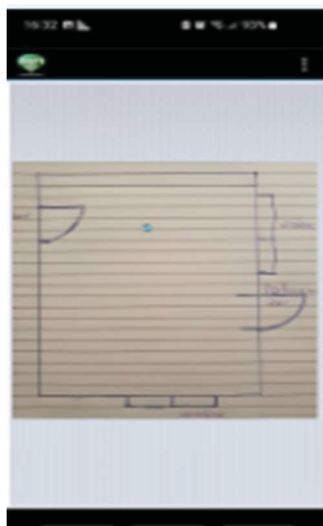
Figura 15 - Adicionando pontos ao mapa – Indoor GPS



FONTE: O Autor (2022).

Ao finalizar a adição dos pontos e geração do mapa, sua localização será destacada nele e, ao se movimentar, o ponto mostrado deve movimentar-se similarmente. Os testes realizados com esse aplicativo não foram assertivos, a leitura do aplicativo sobre a movimentação não foi efetuada, ao findar o processo o ponto não se movimentou, conforme demonstra a FIGURA 16.

Figura 16 - Detectar movimentos – Indoor GPS



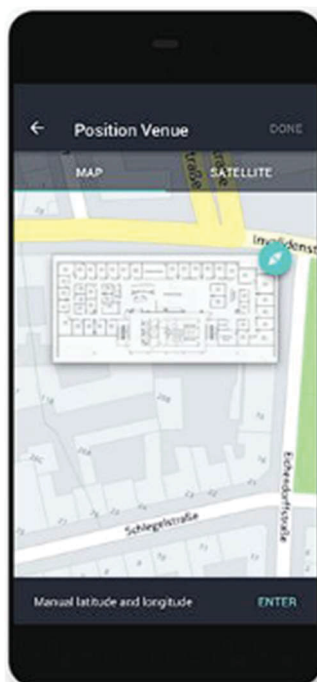
FONTE: O Autor (2022).

O *Indoor GPS*, também apresenta outra restrição, no primeiro processo de adicionar a planta e mapeá-la, o mapa gerado não é salvo, podendo somente ser utilizado enquanto o aplicativo está ativo. Para poder salvá-lo ou exportá-lo, deve-se adquirir com custos, a versão “pro” do aplicativo. Também não é possível adicionar pisos/andares ao mapa, assim como criar rotas.

O aplicativo *HERE Indoor Radio Mapper*, é uma outra opção para navegação *indoor*. Contudo, é necessário criar uma conta para realizar acesso ao aplicativo, a versão oferecida é limitada, e para se ter acesso completo é necessário realizar pagamentos.

A forma disponibilizada pelo aplicativo para adicionar uma planta e suas devidas configurações, foram as mais intuitivas e fáceis de se executar. O aplicativo oferece a opção de enviar sua planta, via arquivo ou disparo da câmera, e sobrepô-la a um mapa gerado via satélite, conforme exemplifica a FIGURA 17.

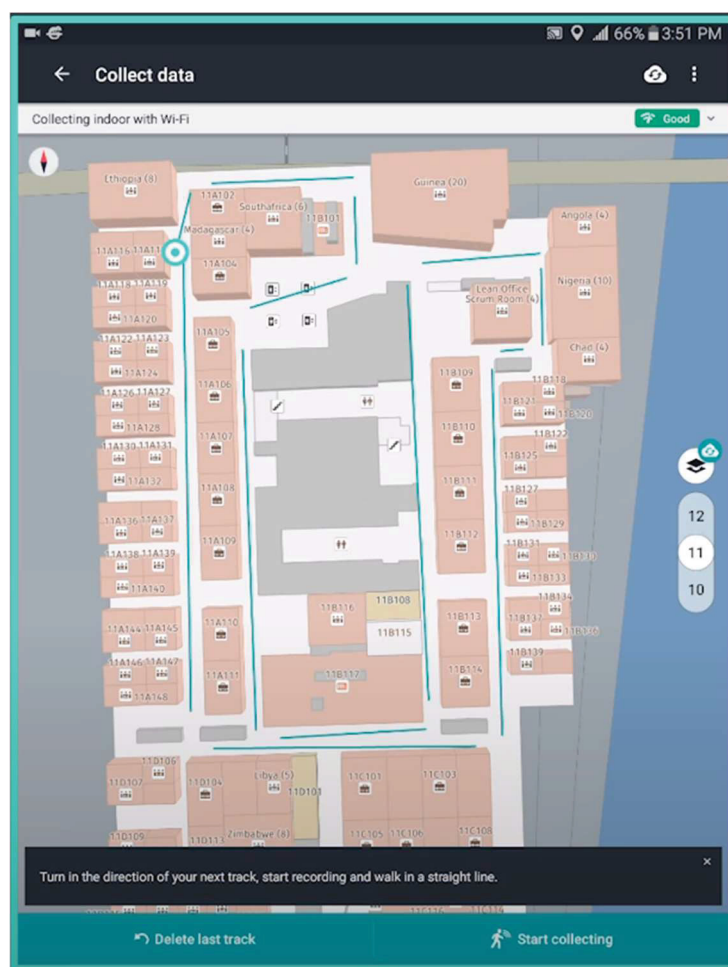
Figura 17 - Adicionar planta - HERE



FONTE: <https://pbs.twimg.com/media/DnXgeKCWsAAQG38>

Esse aplicativo pode trabalhar com vários elementos que compõem o sistema *IPS*. Primariamente trabalha com o escaneamento da intensidade do sinal de *Wi-Fi* e *Bluetooth* que o local possui, e como suplemento também utiliza os sensores do celular para aprimorar a localização. Ao finalizar a inclusão da planta, deve-se mapeá-la, inserindo sua localização inicial, e na sequência efetua-se deslocamentos. Esse processo é demonstrado na FIGURA 18.

Figura 18 - Mapeando planta - HERE



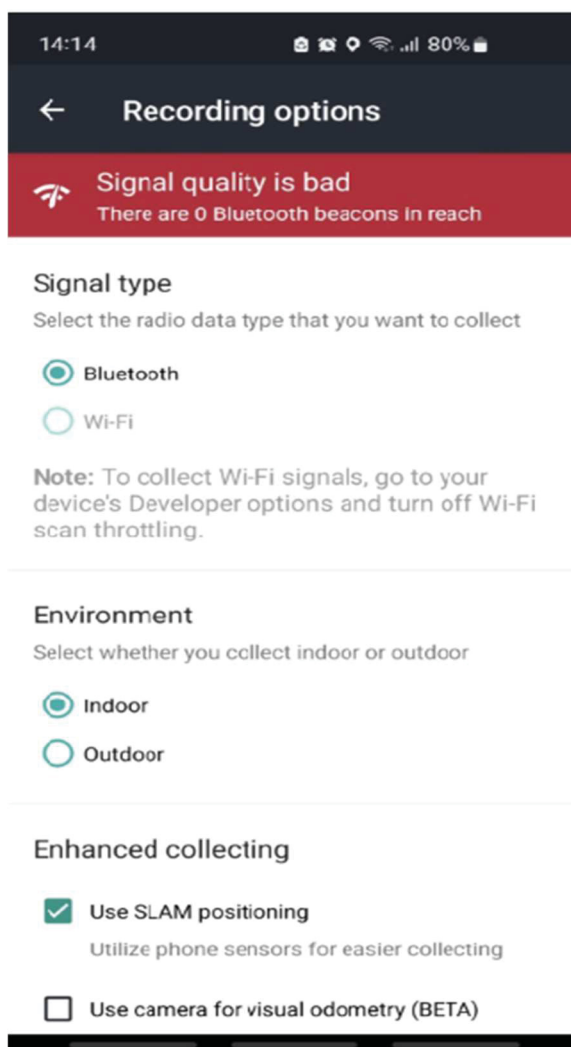
FONTE: <https://youtu.be/Ysw63DPCYQM>

Entretanto, nos testes realizados em locais pequenos verificou-se que a sobreposição de uma planta pequena sobre o mapa via satélite é de difícil ajuste, pois o *zoom* aplicado a imagem do satélite não atinge o tamanho da planta do local. Assim, o aplicativo tem dificuldades em realizar a transição de imagens referentes a locais pequenos, devido à desproporção entre as imagens de satélite e local.

A detecção de movimentos do dispositivo foi bem-sucedida. Ao se movimentar, o ponto na planta referente ao dispositivo também se movimenta, mas precariamente devido a desproporção da planta em relação ao mapa de fundo.

Em contrapartida, não foi possível ativar a detecção do *Wi-Fi*. O aplicativo requer uma permissão específica para o *Wi-Fi*, liberada através de configurações avançadas do dispositivo para um acesso menos restritivo do *Wi-Fi*, conforme mostra a FIGURA 19.

Figura 19 - Erro Detecção Wi-Fi - HERE



FONTE: O Autor (2022).

Da mesma forma que com o aplicativo *Google Maps*, pode-se disponibilizar a planta pelo aplicativo, assim como acessar outras plantas já finalizadas que outros usuários compartilharam, porém, a planta não é pública. O *HERE* também disponibiliza a adição de pisos/andares as plantas.

Esta seção apresentou a fundamentação teórica, que foi o alicerce no desenvolvimento do projeto. A seção seguinte visa mostrar os materiais e métodos que viabilizaram a criação do aplicativo.

### 3 MATERIAIS E MÉTODOS

Sistemas de *softwares* são abstratos e intocáveis, não há impedimentos pela lei da física, processos de manufatura ou propriedades dos materiais. Um ativo intangível, de certo modo, pode tornar a engenharia de *softwares* simplificada inicialmente em projetos pequenos, de estudo ou testes.

Contudo, os sistemas de *softwares* podem se tornar difíceis de entender, caros para alterar e complexos rapidamente. Sendo assim, todos os *softwares* profissionais precisam de engenharia de software, embora não das mesmas técnicas (SOMMERVILLE, 2011).

#### 3.1 PROCESSO DE ENGENHARIA DE SOFTWARE

O processo de *software* é a denominação de uma sequência de atividades relacionadas entre si, que viabilizam a produção de um *software*. A entrega rápida no desenvolvimento de *softwares* é um requisito crítico atualmente, e processos de *softwares*, que pretendem especificar todos os requisitos antes de iniciar essa jornada, não estão adaptados ao desenvolvimento rápido de *software* (SOMMERVILLE, 2011).

A ideia de processos capazes de lidar com mudanças nos requisitos decolou com o desenvolvimento da noção de abordagens ágeis, como Metodologia de Desenvolvimento de Sistemas Dinâmicos (DSDM, *dynamic systems development method*) (STAPLETON, 1997), *Scrum* (SCHWABER e BEEDLE, 2001) e *Extreme Programming* (BECK, 1999).

Nos métodos ágeis, o desenvolvimento se dá por incrementos, em que cada novo incremento é relativo a uma nova funcionalidade do *software*, adiando assim comprometer com requisitos futuros. Compartilhando princípios ágeis, cada método propõe seus próprios processos para atingir esse objetivo (SOMMERVILLE, 2011).

Apresentando foco no gerenciamento iterativo, ao invés de abordagens técnicas, o *Scrum* foi a metodologia ágil utilizada para fornecer um framework de gerenciamento de projetos (SOMMERVILLE, 2011, SCHWABER, 2004; SCHWABER e BEEDLE, 2001). A definição de Scrum, segundo Schwaber e Sutherland (2013, p. 03), é “um framework dentro do qual pessoas podem tratar e resolver problemas

complexos e adaptativos, enquanto produtiva e criativamente entregam produtos com o mais alto valor possível”.

O Scrum é dividido em três fases: planejamento geral, desenvolvimento – ciclos de entrega – e finalização. A primeira fase define os objetivos gerais e arquitetura de *software* do projeto; em seguida a fase de desenvolvimento, com ciclos de entrega de funcionalidades em forma de incremento para o sistema, descritos como *Sprints*; a terceira e uma última fase representa as ações de encerramento do *software*, entrega a documentação, manuais do usuário, e avalia as lições aprendidas (SOMMERVILLE, 2011).

### 3.2 MODELAGEM DE SOFTWARE

A UML – *Unified Modeling Language* ou Linguagem de Modelagem unificada foi utilizada para modelar e documentar o desenvolvimento do projeto, de propósito geral ela pode ser aplicada a todos os domínios de aplicação. É definida como uma linguagem de modelagem e não de programação, cujo o objetivo é auxiliar nas definições das características do sistema, seus requisitos, comportamento, estrutura lógica e a dinâmica de processos (GUEDES, 2018).

O diagrama de caso de uso, encontrado no Apêndice A, é o diagrama mais geral e informal da *UML*. Fornece uma visão de como será a interação entre o usuário e o sistema, utilizado normalmente nas fases de levantamento e análise de requisitos. Também é consultado durante o processo de modelagem, além de servir de base para outros diagramas (GUEDES, 2018).

O Apêndice C abriga o diagrama de classes. Trata-se de uma estrutura lógica estática das classes do sistema com seus atributos, métodos e a relação entre as classes existentes (GUEDES, 2018).

Demonstrado no Apêndice D, o diagrama de sequência visa descrever na linha do tempo as interações com o sistema. As mensagens geradas nesse processo representam a comunicação entre as suas entidades (ARAUJO, 2019).

Por fim, as histórias de usuário constam no Apêndice B. As histórias expressam os requisitos, partindo-se das informações adquiridas estima-se o esforço e recursos necessários para a implementação da história de usuário. Deve-se atribuir prioridades ao cenário de cada história. A importância da funcionalidade para o

sistema é que vai definir, majoritariamente, a ordem de implementação (SOMMERVILLE, 2021).

A seguir é apresentada a metodologia Scrum adaptada para o projeto, com descrição do ciclo de *sprints* que atuaram na entrega dos incrementos de software.

### 3.3 PLANO DE SPRINTS

Uma *sprint* é um tempo fixo de no máximo um mês, que ao final gera uma versão incremental utilizável do produto. Uma nova *sprint* se inicia, após a conclusão da anterior.

Uma lista de tarefas definidas na reunião de planejamento da *sprint*, definem o escopo da *sprint*. Reuniões diárias auxiliam o desenvolvimento das atividades, enquanto, revisão e retrospectiva são as ações que marcam o encerramento da *sprint* (SCHWABER e SUTHERLAND, 2013).

As Sprints do trabalho foram planejadas conforme FIGURA 20.

Figura 20 - Planejamento das Sprints

Iteração/Sprint 1	Iteração/Sprint 2	Iteração/Sprint 3	Iteração/Sprint 4	Iteração/Sprint 5
<b>Tema:</b> Criar o mapa local	<b>Tema:</b> Escolher um destino	<b>Tema:</b> Traçar a rota até o destino	<b>Tema:</b> Acompanhar o deslocamento do usuário	<b>Tema:</b> Criar o módulo web
<b>Data Início:</b> 13/06/2022	<b>Data Início:</b> 18/07/2022	<b>Data Início:</b> 22/08/2022	<b>Data Início:</b> 26/09/2022	<b>Data Início:</b> 31/10/2022
<b>Data Fim:</b> 15/07/2022	<b>Data Fim:</b> 19/08/2022	<b>Data Fim:</b> 23/09/2022	<b>Data Fim:</b> 28/10/2022	<b>Data Fim:</b> 30/11/2022

FONTE: O Autor (2022).

#### 3.3.1 SPRINT 1

A primeira *sprint* foi criada objetivando a entrega do MVP (*Minimum Viable Product*), que consiste na primeira interação do usuário com o aplicativo, onde ele vai efetuar a leitura do Qr Code pelo aplicativo, e receberá como resposta, um mapa do local com sua localização em destaque.

As duas principais atividades da *sprint* foram criar um mapa local e um leitor de Qr Code, em que a criação do mapa ocupou quase toda *sprint*.

### 3.3.2 SPRINT 2

No Incremento do produto, que a segunda sprint entrega, a funcionalidade de escolher um destino foi adicionada ao sistema, renderizando uma lista suspensa dos destinos acessíveis dentro do predio histórico, possibilitando ao usuário selecionar um deles.

Essa tarefa foi executada a partir do mapeamento geográfico dos setores ou destinos que o mapa local, criado na primeira *sprint*, oferece na lista suspensa.

### 3.3.3 SPRINT 3

O caminho entre a localização atual do usuário e o destino selecionado, na sprint 2, será evidenciado no mapa. Essa sprint foi concebida através do algoritmo de *dijkstra*, e mapeamento dos destinos utilizando estrutura de grafos.

### 3.3.4 SPRINT 4

Na iteração dessa sprint, foi disponibilizado o acompanhamento de rota. Para tanto um algoritmo em Java disponibilizado no GitHub foi utilizado em conjunto com o Flutter.

### 3.3.5 SPRINT 5

Durante essa sprint, que fecha o planejamento, foi desenvolvido o incremento de um modulo web. Disponível para acesso e consulta ao mapa, suas rotas e destinos via interface web.

## 3.4 TECNOLOGIAS UTILIZADAS

Ferramentas e tecnologias utilizadas para a análise e desenvolvimento do aplicativo, serão apresentadas nesse capítulo.

### 3.4.1 ANDROID STUDIO

O Android Studio fornece além da *IDE (Integrated Development Environment)*, ferramentas, recursos e algumas facilidades para efetuar as configurações necessárias para o desenvolvimento. A abordagem de instalação do Android Studio e utilização do VS Code como editor de código são comumente vistas, e foram efetuadas nesse projeto.

Ao que tange a parte de desenvolvimento do plugin, utilizou-se exclusivamente o Android Studio. Foi necessário criar um novo projeto Android utilizando a linguagem Java e após incorporá-lo ao projeto principal como *plugin* (ANDROID STUDIO, 2022).

### 3.4.2 ASTAH COMMUNITY

Utilizado na produção dos diagramas, o Astah é uma ferramenta conhecida por muitos desenvolvedores, possui uma versão gratuita chamada *community* (ASTAH, 2022).

### 3.4.3 DART

O Dart foi a linguagem padrão que o Google escolheu ao se criar o Flutter, e que ele mesmo criou em 2011, inicialmente revelado na conferência GOTO em Aarhus, Dinamarca, contudo sua versão inicial saiu no final de 2013 (ZAMMETTI, 2020).

A linguagem possui as seguintes características: totalmente orientado a objetos, garbage-collected (gerencia alocação/desalocação de memória), estilo baseado em C, sistema de tipagem sólido (Flexibilidade na tipagem), compilação antecipada e fornece isolamento de contextos, além de outras características não descritas aqui que o Dart oferece (ZAMMETTI, 2020).

Segundo a pesquisa de Papadopoulou (2019) sobre as linguagens mais importantes para os desenvolvedores em 2019, em termos de interesses por elas, duas linguagens se destacaram: Dart e Python, o Dart em primeiro lugar. Embora uma linguagem de uso geral, seu crescimento e popularidade, vem em grande parte graças ao Flutter (DART, 2022).

### 3.4.4 FLUTTER

No livro Flutter na prática, Zammetti (2020, p. 16) faz sua primeira menção ao Flutter comparando-o a um unicórnio, quando relata que construir aplicativos é uma tarefa complexa. Basicamente deve-se escolher entre performance ou multiplataforma - Android e iOS - para desenvolver um aplicativo móvel: “Devo apresentá-lo então ao unicórnio que é essa realidade fascinante: o Flutter!”.

O autor faz tal analogia devido a experiência em ver outros *frameworks* que se propuseram a fazer o que o Flutter faz, mas não passavam de lendas (unicórnios). No entanto agora, a lenda se tornou realidade, um framework realmente multiplataforma e com recursos nativos é realidade.

O Flutter utiliza a linguagem Dart, ambos desenvolvidos pelo Google. Ele possui a propriedade de se escrever o código, para várias plataformas, uma única vez, obtendo desempenho e recursos nativos. O *framework* disponibiliza através da comunidade do Flutter, alguns *plugins* que fornecem ferramentas e bibliotecas para o desenvolvimento (FLUTTER, 2022).

Utilizando o conceito de *Widgets*, o Flutter permite criar componentes personalizáveis e com maior flexibilidade na construção de *UIs* (*User Interface* – Interface do usuário), ele próprio renderiza seus componentes, podendo renderizar *UIs* de aplicativos a 120 quadros por segundo. Oferece ainda *Widgets* de design específico – Android e iOS – separados em dois conjuntos respectivamente denominados *Material Design* e *Cupertino* (ZAMMETTI, 2020).

### 3.4.5 GIT

O GIT é um sistema de controle de versão de código, pode ser utilizado para registrar o histórico de versão qualquer tipo de arquivo. Sua maior utilização é no versionamento de código, atributo a qual o WIA fez uso. É um projeto de código aberto e com manutenção ativa, desenvolvido em 2005 (GIT, 2022).

### 3.4.6 GITHUB

O GitHub é uma plataforma baseada em nuvem com o objetivo de gerenciar o código do sistema de versionamento git, pode prover ferramentas para monitorar, testar e implantar código, e utilizada no código do projeto (GITHUB, 2022).

#### 3.4.7 GITHUB PAGES

O GitHub Pages é um serviço de hospedagem de site estático, a ferramenta permite hospedar no domínio github.io do GitHub, possibilitando também criar um domínio personalizado, esse serviço hospeda o módulo web do projeto (GITHUB PAGES, 2022).

#### 3.4.8 GOOGLE MAPS

O Google Maps foi utilizado como auxiliar para conferir coordenadas, localizações, realizar alguns testes de conceito e até mesmo como referência (GOOGLE MAPS, 2022).

#### 3.4.9 JAVA

A linguagem foi utilizada para criar o modulo de PDR, linguagem aceita nos dispositivos Android. Java é uma linguagem de programação orientada a objetos, fortemente tipada, em que o código precisa ser compilado, e executado em sua máquina virtual (JVM), característica que lhe fornece a independência de plataforma (JAVA, 2022).

#### 3.4.10 MICROSOFT TO DO

O aplicativo *To Do* foi utilizado para gerenciar tarefas pendentes, organizar a lista de tarefas a fazer, bem como realizar acesso on-line a elas, além de ser integrado ao e-mail outlook e ao sistema operacional Windows 10, utilizado no desenvolvimento.

Ferramentas integradas, quando apropriadas, como o caso do *To Do*, são ótimas opções, pois evitam, poluir o sistema com muitos *softwares*, realizar configurações iniciais muitas vezes custosas em tempo e entendimento (TO DO, 2022).

### 3.4.11 MIRO

O Miro é uma plataforma de lousa interativa digital, fornecendo ferramentas para a construção de mapas mentais, diagramas, quadros com notas e muito mais. Conta com plano gratuito, além suportar trabalhos colaborativos. Foi utilizado para criar os grafos apresentados no projeto e outras demandas gráficas que tarefas de criação que projetos utilizam (MIRO, 2022).

### 3.4.12 OPEN STREET MAPS

O OpenStreetMaps é um projeto que entrega vários serviços e ferramentas gratuitas, direcionadas aos mapas e mantido pela comunidade. Neste projeto, foi utilizado como plano de fundo, ou seja, a planta baixa do prédio histórico foi georreferenciada utilizando-se o serviço de mapas que a plataforma oferece gratuitamente (OSM, 2022).

### 3.4.13 PLUGINS

A linguagem utilizada no desenvolvimento do aplicativo foi o Dart, com o framework *Flutter*. Um *plugin* no *Flutter*, segundo a definição do Google, é um *software* que adiciona recursos ao seu aplicativo GOOGLE DEVELOPERS CODELABS (2021).

Os *Plugins* são uma parte importante do Flutter, fornecendo várias capacidades e recursos, e foram utilizados no WIA para renderizar e gerenciar o mapa e seus atributos, ler o *QR Code*, efetuar o acompanhamento de rota e até atualizar o ícone do iniciador do aplicativo WIA. Plugins oficialmente fornecidos pelo Flutter estão disponíveis em <https://pub.dev/>.

Ressalta-se que nem todos os plugins estão disponíveis para as 6 plataformas do Flutter, sendo elas denominadas aqui: Linux, Windows, iOS, MacOS, Android e WEB. Os seguintes *plugins* foram utilizados:

PLUGIN\_PDR

Na biblioteca de plugins não foi encontrado nenhum *plugin* que provesse suporte ao PDR, foi necessário criar um plugin próprio, chamado de `plugin_pdr`. Assim utilizando código em Java disponibilizado no GitHub, o `plugin_pdr` foi criado, o código usado está sob licença *Creative Commons* e os créditos ao criador estão mantidos (CHENHONGRUIXUAN, 2021).

Basicamente, no contexto do projeto, a técnica realiza a leitura dos sensores inerciais, e outros sensores que possam agregar e estejam acessíveis no celular. E quando o usuário começa a caminhar, ela irá mensurar a direção e a distância do deslocamento, a partir de um ponto inicial.

Nos dados brutos gerados pelos sensores, devem ser utilizadas equações, então, a fim de diminuir o ruído. Outras equações, que adotam técnicas de interpretação de passos, são aplicadas a esses dados visando detectar o início e fim do ato de caminhar, quando ele ocorrer, além da direção e a velocidade em que ocorreu (COUSSEAU, 2016).

Para se chegar nesse objetivo, um algoritmo de *PDR* se faz necessário, e para implementá-lo um plugin foi criado (DOCS FLUTTER, 2022).

Ao se criar um plugin, se deve especificar quais plataformas o plugin dará suporte, e ao se especificar a plataforma Android ou iOS, pode-se também especificar qual será a linguagem utilizada, para Android as opções são Java e Kotlin, enquanto iOS possui o Swift e Objective-C.

Dentre os códigos que foram encontrados no GitHub, somente 3 estavam em Java, uma das linguagens aceitas no plugin, e tinham algum reconhecimento da comunidade do GitHub, como estrelas ou seguidores. Por esse motivo, a única plataforma suportada pelo plugin é o Android.

## FLUTTER\_BARCODE\_SCANNER

Adiciona o recurso de leitura de *QR Code* e *Barcodes* (código de barras) através da câmera, tanto para Android como para iOS (FLUTTER\_BARCODE\_SCANNER, 2022).

## FLUTTER\_MAP

O plugin `flutter_map` fornece recursos para renderizar mapas locais ou fornecidos pela rede e seus atributos, como linhas e polígonos. Renderiza os elementos em forma de camadas, que podem ser sobrepostas.

Fornece a possibilidade de habilitar certos gestos, na tela do mapa, que podem chamar comando de zoom, adicionar ou mudar, uma marcação ao local, gerenciando os ícones que delimitaram espaços e localizações no mapa (FLUTTER\_MAP, 2022).

## DIJKSTRA

Implementa o algoritmo de Dijkstra, para solucionar o problema do caminho mais curto num grafo. Utilizado para criar uma rota entre destinos utilizando o menor caminho entre eles (DIJKSTRA, 2022).

### 3.4.14 QGIS

O QGIS é um sistema de Informações Geográficas livre e de código aberto. Foi utilizado para realizar o georreferenciamento da planta baixa, gerar e criar o padrão do mapa local do projeto (QGIS, 2022).

### 3.4.15 VISUAL STUDIO CODE

A *IDE* – Integrated Development Environment – VS Code da Microsoft foi utilizada na maior parte do desenvolvimento do aplicativo, ela foi lançada em 2015 destinada ao desenvolvimento *web*, é leve e multiplataforma além de atender a uma gama enorme de projetos, uma característica muito importante, gerando muito material de aprendizagem e suporte sobre a ferramenta, além de totalmente gratuita (VS CODE, 2022) .

### 3.4.16 HARDWARE

O Hardware utilizado é composto por um computador de arquitetura x86, contendo os seguintes componentes:

- OS: Windows 10 Pro - 21H2.

- CPU: AMD - Phenom II x4 965 BE.
- RAM: 12GB DDR3 800MHz in dual-channel 2 x 4GB + 2 x 2GB.
- System Storage: SSD Samsung 840 EVO 120GB.
- Data Storage: 4TB Hard Disk SATA2 in 2 x 2TB.
- VGA: Radeon RX 550 4GB GDDR5 128bits.

Devido ao tempo de vida do processador, mais de 11 anos, ele não possui as instruções necessárias ao emulador de dispositivos, que cria um celular virtual para desenvolvimento e testes. Além do emulador não poder virtualizar os sensores necessários para os testes, como acelerômetro, giroscópio e outros.

Levando em conta essas intemperes, foi necessário utilizar um dispositivo físico no desenvolvimento, detalhado a seguir:

- Model: Samsung Galaxy s10e – SM-G970F.
- SoC: Samsung Exynos 9820 – 2.7GHz Octa-Core.
- Display: 5.8 Dynamic AMOLED – 1080 x 2280 – 60Hz.
- GPU: ARM – Mali-G76 MP12.
- RAM: 6GB LPDDR4X 2133MHz in dual-channel 2 x 3GB.
- Internal Storage: 128GB MicroSD – UFS 2.1.
- External Storage: 128GB MicroSD Samsung Evo plus class 10.
- OS: Android 12 – Snow Cone.

O capítulo explanou sobre a metodologia de desenvolvimento de *software* utilizada no aplicativo. Já o próximo capítulo apresentará as telas do WIA e suas respectivas funcionalidades.

## 4 APRESENTAÇÃO DO SISTEMA

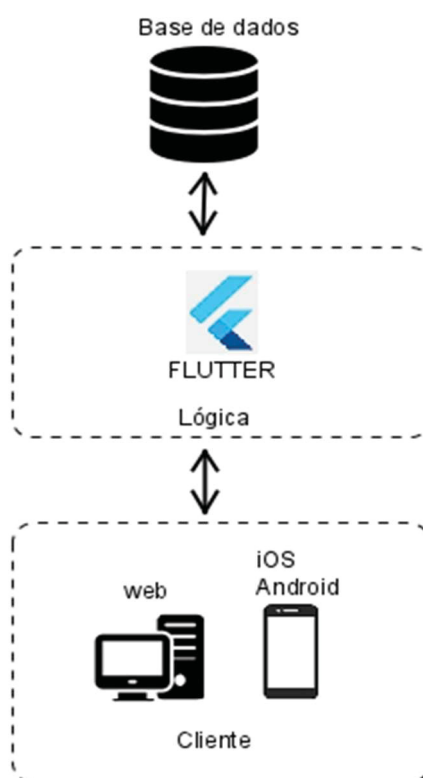
O Sistema possui uma interface web para eventuais consultas ao mapa, seus destinos e rotas. A interface web está disponível no *GitHub Pages*, e pode ser acessada através do *link* <https://kleytonsouza.github.io/wia/#/>.

Utilizando a interface *mobile*, iOS ou Android, o sistema pode obter a localização dos dispositivos através da leitura de QR Code, e disponibiliza o módulo *PDR* em celulares Android.

### 4.1 ARQUITETURA DO SISTEMA

O WIA está disponível para *Android* e *iOS*, assim como para plataforma web, porém seu módulo de *PDR*, somente para *Android*. A primeira camada suporta essas três interfaces do cliente. A segunda camada é de responsabilidade do *Flutter*, que se encarrega de todo o restante, utilizando o Dart, ele provê a lógica do sistema e o acesso aos dados. A FIGURA 21 ilustra a arquitetura.

Figura 21 - Arquitetura WIA



FONTE: O Autor (2022).

## 4.2 FUNCIONALIDADES

Por meio da interface *web*, o usuário pode realizar uma consulta ao mapa, possíveis destinos e rotas. Quanto utiliza o aplicativo, há também a funcionalidade de acompanhamento de rotas, localização dos dispositivos e melhor manejo do mapa utilizando a tela do *mobile*. Entende-se que mesmo limitado, dar suporte à interface *web* é primordial.

### 4.2.1 Aplicativo

Foi o objetivo do presente trabalho, onde a função conceito do *PDR* pode ser aplicada e demonstrada, fornecendo ainda a localização atual com a leitura de *QR Code*, e através de gestos na tela, uma manipulação mais dinâmica do mapa. A tela inicial do aplicativo pode ser visualizada na FIGURA 22.

Figura 22 - Tela inicial aplicativo



FONTE: O Autor (2022).

Ao se tocar no botão “Ler QR Code”, a tela referente a leitura será invocada, e pode ser notada na FIGURA 23.

Figura 23 - Tela leitura QR Code



FONTE: O Autor (2022).

A câmera do celular será invocada, e deve ser direcionada ao QR Code, a leitura é realizada quase que instantaneamente. Há três opções disponíveis nesta tela, e serão descritas conforme seus ícones.

Da esquerda para a direita, representando o ícone da câmera e duas setas desenhando no interior em forma de um círculo, esse botão quando acionado muda a câmera entre frontal e traseira. O segundo ícone, um raio cortado, liga ou desliga a luz – flash da camera, e finalmente o ultimo “Cancel”, aborta a operação, fecha a câmera e retorna a tela inicial.

Assim que o código é interpretado, a próxima tela com o mapa do prédio histórico e a localização do código são demonstrados, utilizando-se um ícone amarelo na localização do código, bem como pintando a área pertencente a ele de azul claro, como a FIGURA 24 abaixo ilustra.

Figura 24 - Tela Mapa localização



FONTE: O Autor (2022).

Na tela do mapa nota-se que o título em destaque na primeira linha informa o nome do setor ao qual o código pertence. Já o mapa pode ser manipulado de três formas: duas utilizando-se os dedos em formato de pinça, onde o movimento na tela de abrir ou fechar a pinça, será interpretado como diminuir ou aumentar o *zoom* da imagem; o de girar a pinça analogicamente girará o mapa; e o simples gesto de deslizar o dedo sobre a tela irá mover a imagem para o lado que ação foi efetuada.

Para escolher um destino, e visualizar uma rota até ele, deve-se tocar na lista suspensa denominada “Selecione um Destino” e dentre a lista de destinos mostrados, escolher um. Atualmente são disponibilizados destinos do andar térreo do prédio histórico da UFPR. Essa opção pode ser denotada na FIGURA 25.

Figura 25 - Tela mapa escolher destino



FONTE: O Autor (2022).

Assim que a escolha do destino é realizada, nota-se que uma rota entre o destino escolhido e a localização atual, obtida através da leitura do QR Code, é renderizada em cor azul escuro no mapa, conforme mostra a FIGURA 26.

Figura 26 - Tela mapa rotas

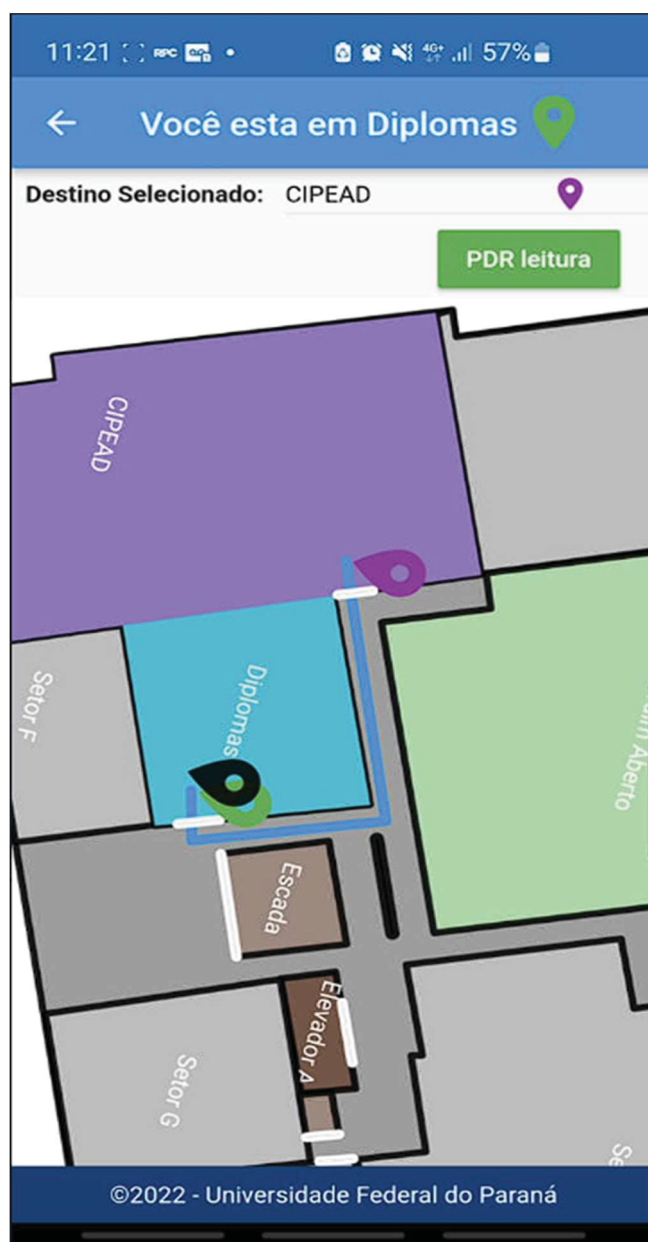


FONTE: O Autor (2022).

Quando a rota é obtida, e outros detalhes tornam-se perceptíveis: área do destino selecionado é destacada em roxo, o ícone da localização atual muda da cor amarela para a cor verde, e um ícone representando o destino é destacado também na cor roxa.

É possível ainda, no Android, utilizar o acompanhamento de rota. Basta tocar no botão com os dizeres “PDR Leitura”, e um terceiro ícone de cor preta será mostrado, na mesma localização inicial e irá se deslocar conforme o usuário se desloca pelo prédio. A FIGURA 27 pode demonstrar isso.

Figura 27 - Tela mapa PDR



FONTE: O Autor (2022).

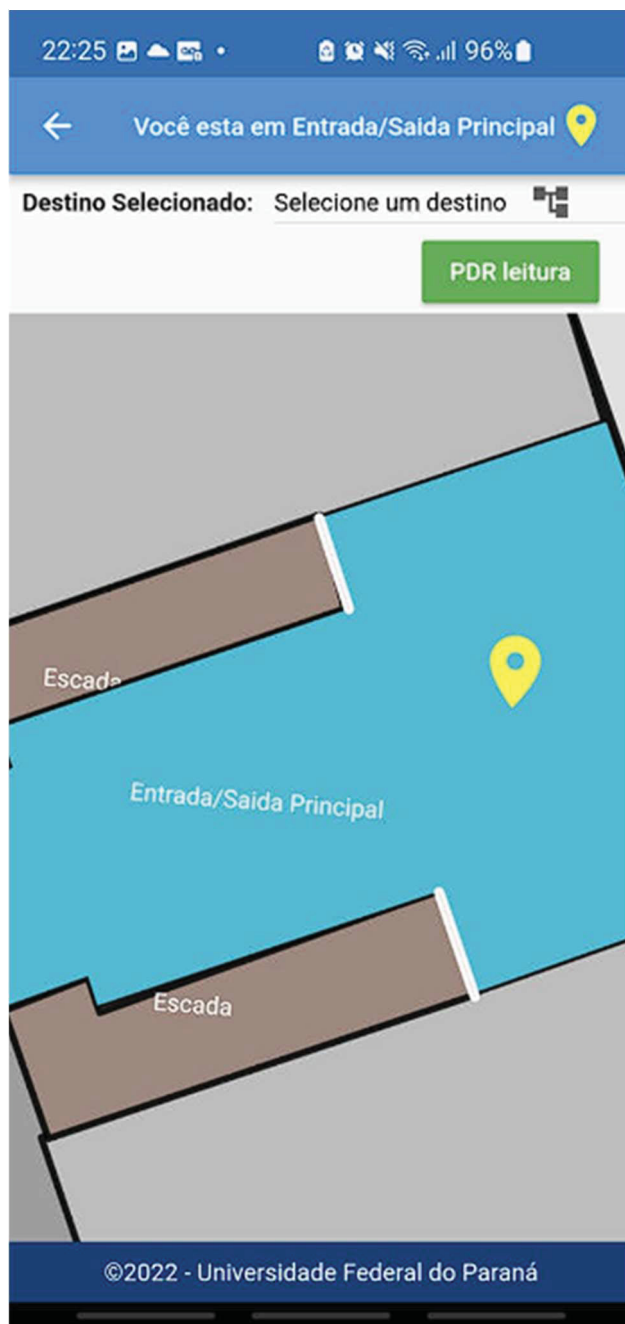
O mapa local do prédio tem seus limites demarcados, com foco na interpretação e orientação do usuário, perante a estrutura do local. Isso se aplica também a ampliação ou *zoom* do mapa, com limite mínimo de 18 e máximo de 22, começando em 19, as Figuras 28 e 29 demonstram respectivamente o menor e o maior nível de *zoom*.

Figura 28 - Tela mapa zoom mínimo



FONTE: O Autor (2022).

Figura 29 - Tela mapa zoom máximo



FONTE: O Autor (2022).

O aplicativo também permite rotacionar ou girar o mapa. Inicialmente, ele é renderizado com o Norte representando a parte superior da imagem e assim respectivamente, direita Leste, inferior Sul e direita Oeste. Para fornecer uma melhor usabilidade e facilidade de interpretação da localização e do deslocamento, realizar um gesto de girar com dois dedos na tela do celular, conseqüentemente irá girar a imagem no mesmo sentido do gesto. A FIGURA 30 demonstra isso em um deslocamento de rota sendo realizado com o *PDR* ativado.

Figura 30 - Tela mapa rotacionado

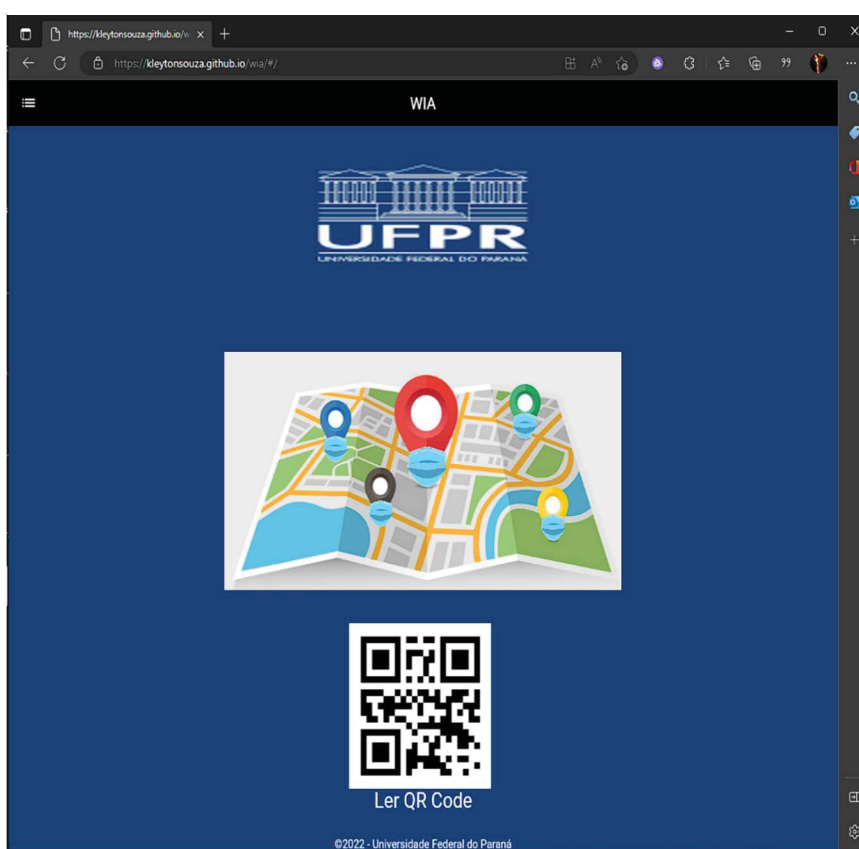


FONTE: O Autor (2022).

#### 4.2.2 Interface WEB

O primeiro ponto pode ser observado na funcionalidade de leitura de *QR Code*, que na interface web não está habilitada. Ao se clicar no botão referente a leitura do código, será diretamente apresentada a tela seguinte. Graficamente as telas iniciais são iguais, *web* ou *mobile*, a diferença ocorre na parte lógica e na renderização. A tela inicial utilizando a interface WEB no browser Edge é demonstrada na FIGURA 31.

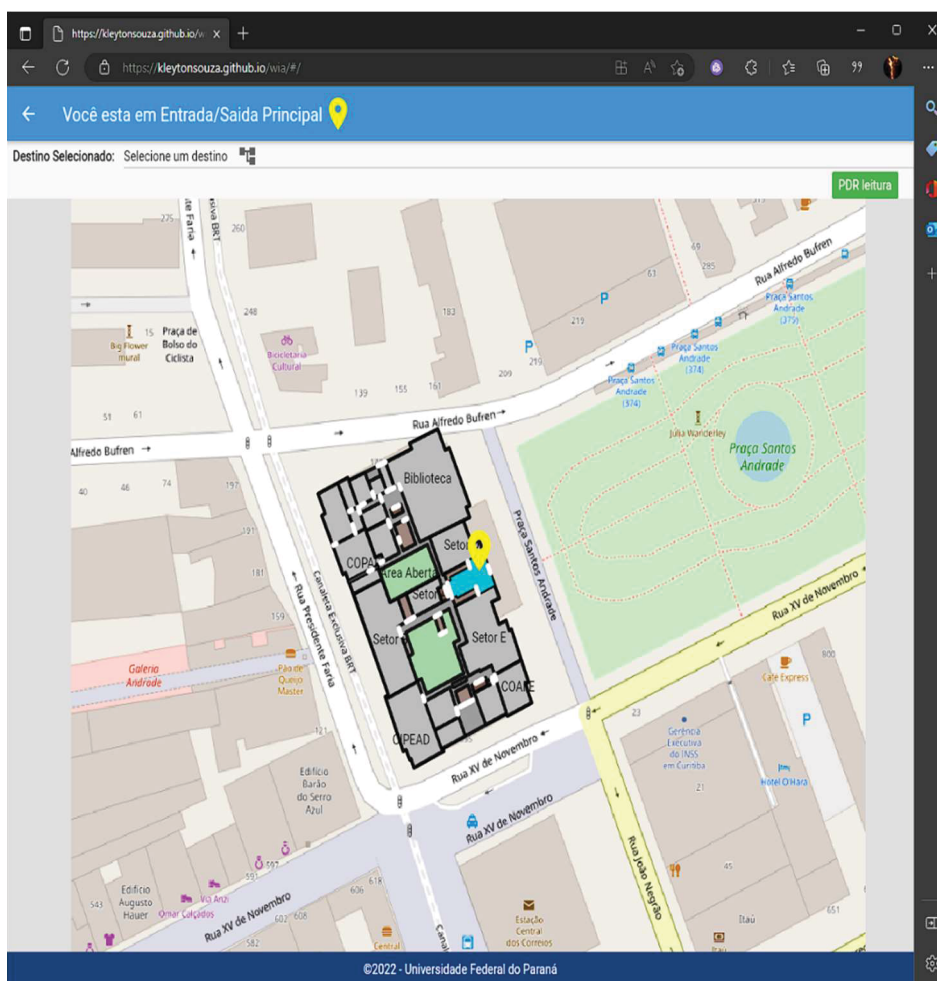
Figura 31 - Tela inicial interface web



FONTE: O Autor (2022).

A tela apresentada, ao se clicar no botão denominado “Ler QR Code”, será diretamente a tela do mapa do prédio histórico da UFPR. Partindo do ponto padrão inicial, a Entrada principal localizada em frente à praça Santos Andrade. A tela do mapa na interface web, pode ser percebida na FIGURA 32.

Figura 32 - Tela do mapa web



FONTE: O Autor (2022).

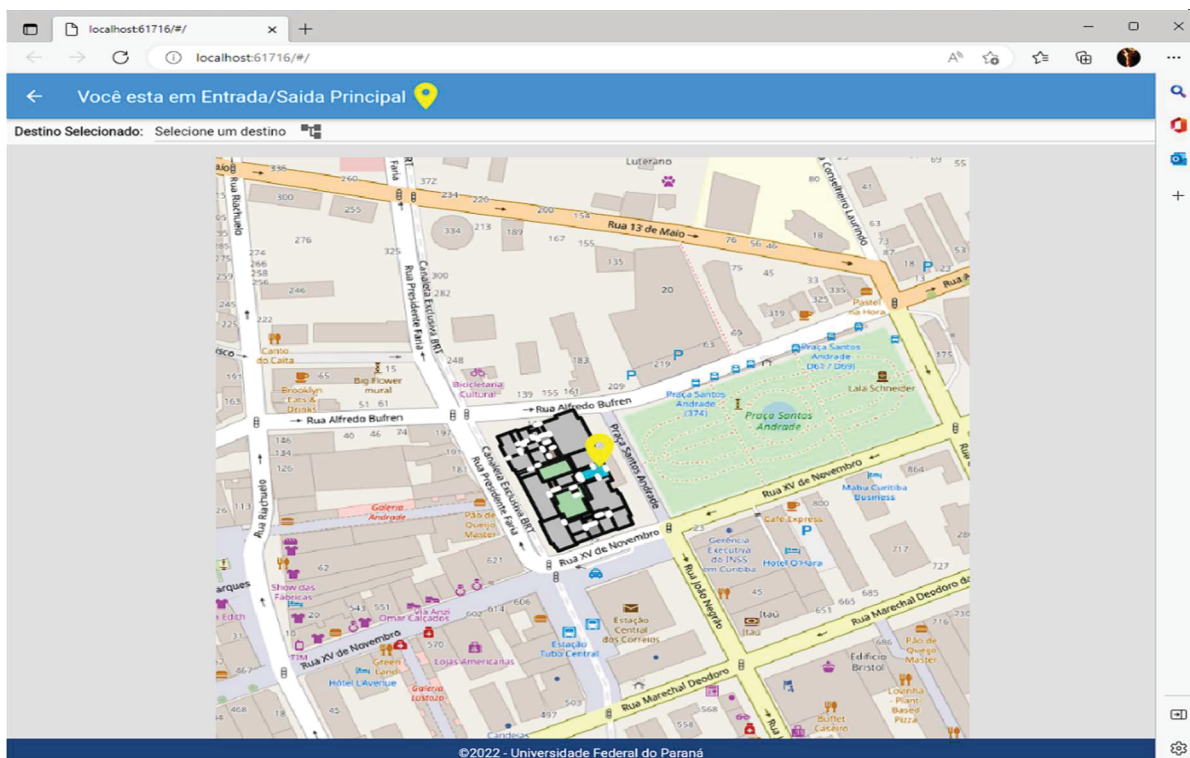
Nota-se que o botão de cor verde com o escrito “PDR Leitura” que deveria ser renderizado somente no modo *mobile*, está presente no modo *web*. É uma característica do GitHub Pages. No servidor local o botão não é renderizado, mas mesmo presente, sua função está inapta na interface *web*.

A ausência do *PDR* e do leitor de *QR Code* na interface *web* atribui-se pela falta dos sensores e de uma câmera nessa plataforma, essenciais para o desempenho dessas ações. Observa-se também que o acesso via browser é possível a dispositivos não móveis, como computadores *Desktop*, que usualmente não possuem sensores inerciais e mesmo dotados de câmera, a leitura deve ser realizada no local em que o *QR Code* se encontra.

Outra característica notada na interface *mobile* e que não está disponível na *web*, é a rotação do mapa, pelo menos não de maneira nativa. No entanto a funcionalidade de ampliar o mapa está disponível, onde o nível começa em 18 e vai

até 22, abrindo o mapa por padrão em 19. A FIGURA 33 demonstra o menor nível de zoom possível (18), enquanto a FIGURA 34 o maior (22).

Figura 33 - Tela mapa web zoom mínimo



FONTE: O Autor (2022).

Figura 34 - Tela mapa web zoom máximo

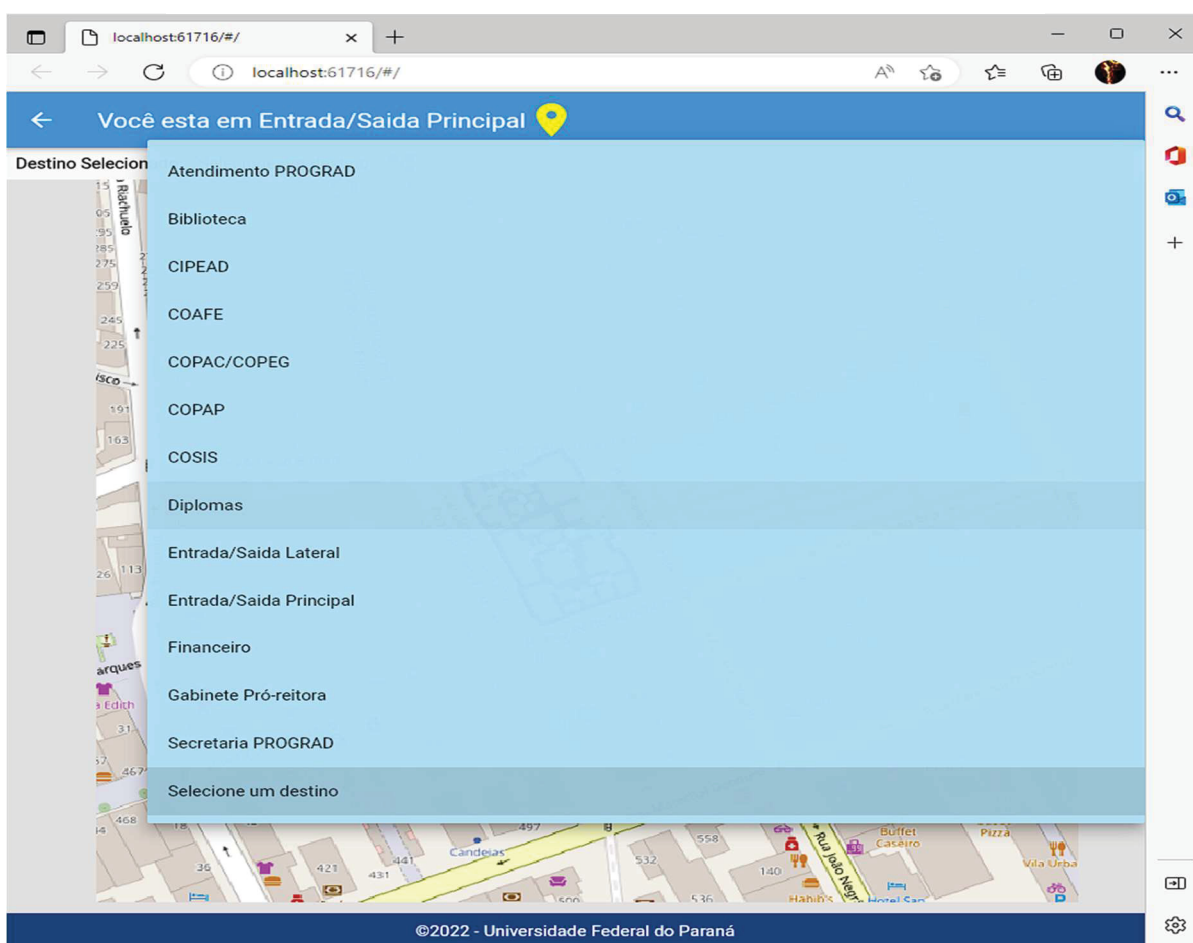


FONTE: O Autor (2022).

Comparando-se ambas as imagens, nota-se que detalhes distintos são expostos conforme o nível de *zoom* é aplicado, como outras estruturas próximas, ruas e edificações que compõem o entorno do campus, assim como detalhes da infraestrutura interna do prédio histórico.

Para se encontrar um destino, dentre os disponíveis e acessíveis, foi disponibilizado uma lista. Na tela do mapa, basta clicar na lista suspensa, denominada “Selecione um destino”, e o aplicativo irá mostrar uma lista de destinos, permitindo a escolha de um, como pode-se ver na FIGURA 35.

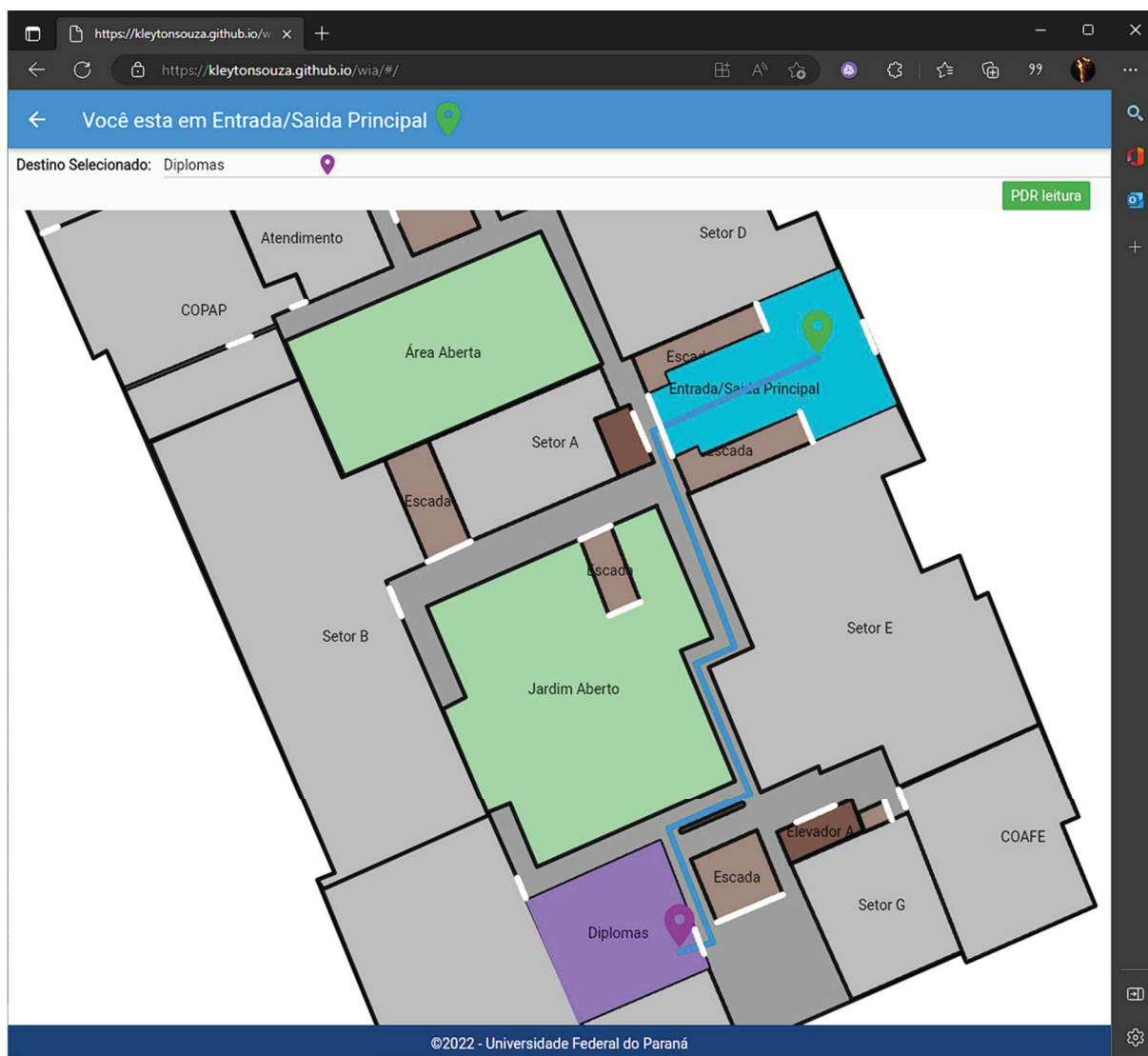
Figura 35 - Tela mapa escolher destino



FONTE: O Autor (2022).

Ao se escolher um destino uma rota será traçada até ele, e sua área será destacada na cor roxa. Pode-se conferir essa ação na FIGURA 36.

Figura 36 - Tela mapa web rota



FONTE: O Autor (2022).

Neste capítulo o sistema construído foi exposto, suas funcionalidades detalhadas através das telas apresentadas, tanto do aplicativo quanto da interface web, a fim de ilustrar sua utilização. A seguir serão descritas as considerações finais e recomendações para trabalhos futuros.

## 5 CONSIDERAÇÕES FINAIS

Esse trabalho apresentou o sistema de posicionamento interno WIA, que tem como objetivo auxiliar na mobilidade *indoor* do prédio histórico da UFPR. Em sua interface web, fornece consulta ao mapa, destino e rotas. Pelo seu aplicativo, além destas funcionalidades, permite também realizar o acompanhamento de rotas e localização do dispositivo de forma autônoma.

A concepção do sistema se deu utilizando metodologias ágeis de desenvolvimento de *software*. Utilizando-se essas premissas, a primeira entrega, na primeira *sprint*, resultou no *MVP* do produto. As outras quatro *sprints* seguintes foram o complemento do produto. Foi possível entregar funcionalidades de maneira flexível e programada, finalizando a entrega de sistema completo ao findar de todas as sprints.

O WIA foi desenvolvido em Flutter, mas grande parte desprendida em seu desenvolvimento foi em estudos de áreas diversas, como estudo geográfico e manipulação de ferramentas técnicas especializadas (QGIS), para entender e assim desenvolver, armazenar e gerenciar o próprio mapa interno.

Entende-se que o sistema resultante deste trabalho fornece de forma satisfatória o auxílio à mobilidade interna no prédio histórico da UFPR. A solução final ainda requer aprimoramentos no *PDR*, que apresenta pouca precisão, mas é primordial pois é a base do sistema proposto no que tange o acompanhamento ao deslocamento do dispositivo.

Ao se criar um sistema baseado no *IPS*, como é o WIA, os primeiros passos necessários já foram realizados aqui, ou seja, primariamente é imprescindível ter o mapa do local, seus setores e rotas mapeadas, além do *PDR*. Com essa base sólida, é possível evoluir a solução e obter-se um sistema híbrido de alta precisão, como recomenda-se a seguir.

### 5.1 RECOMENDAÇÕES PARA TRABALHOS FUTUROS

Durante o desenvolvimento do projeto, percebeu-se algumas necessidades relevantes, que não foram implementadas devido à falta de tempo hábil. Contudo, a base está completa e apta a receber novos componentes ou elementos que poderão agregar ao sistema, e suprir as necessidades percebidas, listadas a seguir:

- Receber e analisar o sinal WI-FI interno, e atuar em conjunto com o *PDR*, provendo mais precisão ao sistema;
- Analisar o ambiente e criar as *landmarks* e adicionar ao sistema;
- Salvar e Gerenciar os dados geoespaciais obtidos. Permitindo assim personalizar e gerenciar mensagens, acessos, fluxos e itinerários;
- Estender o processo aplicado ao primeiro andar para os demais andares;
- Adicionar novos módulos que suportem dispositivos com conexões wireless, como: *Bluetooth*, *RFID* (Radio Frequency Identification, significa identificação por radiofrequência) e outras sem fio. Podendo assim iniciar um processo de *IoT*<sup>3</sup> (Internet das Coisas – *Internet of Things*);
- Adicionar um módulo de login, e assim identificar os usuários.

---

<sup>3</sup> descreve a rede de objetos físicos incorporados a sensores, software e outras tecnologias com o objetivo de conectar e trocar dados com outros dispositivos e sistemas pela internet (ORACLE, 2022)

## REFERÊNCIAS

AGÊNCIA NACIONAL DE AVIAÇÃO CIVIL (ANAC). ANACpédia Acelerômetro. Disponível em: [https://www2.anac.gov.br/anacpedia/por\\_ing/tr131.htm](https://www2.anac.gov.br/anacpedia/por_ing/tr131.htm). Acesso em: 23 mai. 2022.

AGÊNCIA NACIONAL DE AVIAÇÃO CIVIL (ANAC). ANACpédia Giroscópio. Disponível em: [https://www2.anac.gov.br/anacpedia/por\\_ing/tr1554.htm](https://www2.anac.gov.br/anacpedia/por_ing/tr1554.htm). Acesso em: 23 mai. 2022.

ANDROID STUDIO. Disponível em: <https://developer.android.com/studio>. Acesso em: 23 nov. 2022.

ARAÚJO, F. J. D. **Verificação de Refinamento em Diagramas de Sequência com Estrutura de Controle**. Trabalho de Graduação (Bacharelado em Ciências da Computação), Universidade Federal de Pernambuco, Recife, 2019. Disponível em: [https://repository.ufrpe.br/bitstream/123456789/1858/1/tcc\\_danieljosefreiredearaujo.pdf](https://repository.ufrpe.br/bitstream/123456789/1858/1/tcc_danieljosefreiredearaujo.pdf). Acesso em: 26 nov. 2022.

ARCGIS. **Get Started with geofences in ArcGIS Field Maps**. Blog. Josh Clifford. Disponível em: <https://www.esri.com/arcgis-blog/products/field-maps/field-mobility/get-started-with-geofences-in-arcgis-field-maps/>. Acesso em: 06 nov. 2022.

ASTAH COMMUNITY. Disponível em: <https://astah.net/>. Acesso em: 23 nov. 2022.

BECK, K. Embracing Change with Extreme Programming. **IEEE Computer**, v. 32, n. 10, oct. 1999, p. 70-78.

BRASIL ESCOLA. Aceleração. Disponível em: <https://brasilecola.uol.com.br/fisica/aceleracao.htm>. Acesso em: 27 nov. 2022.

CHELLY, M.; SAMAMA, N. New techniques for indoor positioning, combining deterministic and estimation methods. **ENC-GNSS 2009: European Navigation Conference** - Global Navigation Satellite Systems, may. 2009, Naples, Italy. pp.1 - 12.

CHRIST, R. D.; WERNLI, R. L. **The ROV Manual** A User Guide for Remotely Operated Vehicles. Butterworth-Heinemann, 2014.

COUSSEAU, V. M. R. **Um Estudo Sistemático Sobre Técnicas de Dead Reckoning para Localização Indoor**. Trabalho de Graduação (Bacharelado em Ciências da Computação), Universidade Federal de Pernambuco, Recife, 2016. Disponível em: <https://www.cin.ufpe.br/~tg/2016-2/vmrc-proposta.pdf>. Acesso em: 21 nov. 2022.

D'Agostini, D. **Design de sinalização**. São Paulo: Blucher, 2016.

DART. Disponível em: <https://dart.dev/>. Acesso em: 23 nov. 2022.

DEVMIO. **Results are in: Fully fledged Dart takes first in 2019 poll.** Eirini-Eleni Papadopoulou. Disponível em: <https://devm.io/programming/poll-results-dart-word-2019-154779>. Acesso em: 22 nov. 2022.

DIJKSTRA. Disponível em: <https://pub.dev/packages/dijkstra>. Acesso: em 23 nov. 2022.

DOCS FLUTTER. Developing plugin packages. Disponível em: <https://docs.flutter.dev/development/packages-and-plugins/developing-packages#plugin>. Acesso em: 22 nov. 2022.

DO LAGO, I. F.; FERREIRA, L. D. D.; KRUEGER, C. P. GPS E GLONASS: ASPECTOS TEÓRICOS E APLICAÇÕES PRÁTICAS. **Boletim de Ciências Geodésicas**, [S.l.], dec. 2002. ISSN 1982-2170. Disponível em: <https://revistas.ufpr.br/bcg/article/view/1419>. Acesso em: 01 nov. 2022.

EDELKAMP, S.; SCHRÖDL, S. **Heuristic Search: Theory and Applications.** São Francisco: Morgan Kaufmann, 2012.

EMBARCADOS. **Sensores Inerciais – Parte 2.** Henrique Torres. Disponível em: <https://www.embarcados.com.br/sensores-inerciais-parte-2>. Acesso em: 12 mai. 2022.

FLUTTER. Disponível em: <https://flutter.dev>. Acesso em: 23 nov. 2022.

FLUTTER\_BARCODE\_SCANNER. Disponível em [https://pub.dev/packages/flutter\\_barcode\\_scanner](https://pub.dev/packages/flutter_barcode_scanner). Acesso em: 23 nov. 2022.

FLUTTER\_MAP. Disponível em: [https://pub.dev/packages/flutter\\_map](https://pub.dev/packages/flutter_map). Acesso em: 23 nov. 2022.

FREECODECAMP. **Algoritmo de caminho de custo mínimo de Dijkstra** – uma introdução detalhada e visual. Cayo Dias. 27 mar. 2022. Disponível em: <https://www.freecodecamp.org/portuguese/news/algoritmo-de-caminho-de-custo-minimo-de-dijkstra-uma-introducao-detalhada-e-visual/>. Acesso em: 04 nov. 2022.

GENTE GLOBO. **VIVÊNCIA NOS ESPAÇOS URBANOS** - O comportamento dos brasileiros em relação à mobilidade. Disponível em: <https://gente.globo.com/vivencias-nos-espacos-urbanos/>. Acesso em: 07 abr. 2022.

GIT. Disponível em: <https://git-scm.com/>. Acesso em: 23 nov. 2022.

GITHUB. Disponível em: <https://github.com/>. Acesso em: 23 nov. 2022.

GITHUB. **PDR Algorithm.** Chenhongruixuan. Disponível em: <https://github.com/ChenHongruixuan/In-outdoorSeamlessPositioningNavigationSystem>. Acesso em: 23 nov. 2022.

GITHUB PAGES. Disponível em: <https://pages.github.com/>. Acesso em: 23 nov. 2022.

GOOGLE DEVELOPERS CODELABS. **How to write a Flutter plugin**. Site. Aaron Clarke. 06 out. 2021. Disponível em: <https://codelabs.developers.google.com/codelabs/write-flutter-plugin#0>. Acesso em: 10 mai. 2022.

GOOGLE MAPS. Disponível em: <https://google.com.br/maps> . Acesso em: 23 nov. 2022.

GOOGLE MAPS BILING. Disponível em: <https://developers.google.com/maps/documentation/android-sdk/usage-and-billing>. Acesso em: 23 nov. 2022.

GUEDES, G. T. **UML 2-Uma abordagem prática**. São Paulo: Novatec, 2018.

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS (INPE). MAGNETOMETER. Disponível em: <http://www2.inpe.br/climaespacial/portal/magnetometer-about/>. Acesso em: 23 mai. 2022.

JAVA. Disponível em: <https://www.java.com/pt-BR/>. Acesso em: 23 nov. 2022.

LIAO, W.; ZHAO, Y. Using Dual-Axis Accelerometers to Protect Hard Disk Drives. **Analog Dialog**, nov. 2005. Disponível em: <https://www.analog.com/en/analog-dialogue/articles/using-accelerometers-to-protect-hard-drives.html>. Acesso em: 23 mai. 2022.

LYMBEROPOULOS, D. et al. A realistic evaluation and comparison of indoor location technologies. **Proceedings Of The 14th International Conference On Information Processing In Sensor Networks - Ipsn '15**, [s.l.], p.178-189, 2015. Disponível em: <http://dx.doi.org/10.1145/2737095.2737726>. Acesso em: 25 nov. 2022.

LYNCH, K. **The Image of City**. The MIT Press, 1960.

MIRO. Disponível em: <https://miro.com/pt/>. Acesso em: 23 nov. 2022.

MOBILETIME. **Aeroporto do Galeão é o primeiro da América Latina a ter app com navegação indoor**. Fernando Paiva. São Paulo, 28 jun. 2016. Disponível em: <https://www.mobiletime.com.br/noticias/28/06/2016/aeroporto-do-galeao-e-o-primeiro-da-america-latina-a-ter-app-com-navegacao-indoor/>. Acesso em: 11 mai. 2022.

ORACLE. O que é IoT?. Disponível em: <https://www.oracle.com/br/internet-of-things/what-is-iot/>. Acesso em: 24 mai. 2022.

OSM. Disponível em: <https://www.openstreetmap.org/>. Acesso em: 23 nov. 2022.

Play Store. Google Maps. Disponível em:

<https://play.google.com/store/apps/details?id=com.google.android.apps.maps&hl=pt-BR&q=US>. Acesso em: 23 mai. 2022.

PORTAL DE NOTÍCIAS DA GLOBO (G1). Prédio da UFPR representa a História da Universidade mais antiga do país. Confea. Disponível em:

<https://g1.globo.com/pr/parana/especial-publicitario/confea/noticia/2019/10/21/predio-da-ufpr-representa-a-historia-da-universidade-mais-antiga-do-pais.ghtml>. Acesso em: 04 abr. 2022.

PRIBERAM. Piezeletricidade. Disponível em:

<https://dicionario.priberam.org/piezeletricidade>. Acesso em: 12 set. 2022.

QGIS. Disponível em: [https://qgis.org/pt\\_BR/site/](https://qgis.org/pt_BR/site/). Acesso em: 23 nov. 2022.

RANDELL, C.; DJIALLIS, C.; MULLER, H. Personal position measurement using dead reckoning, **Seventh IEEE International Symposium on Wearable Computers**, New York, pp. 166-173, 2003. Disponível em:

<https://ieeexplore.ieee.org/document/1241408>

RIBEIRO, L. G. Onde estou? Para onde vou? Ergonomia do ambiente construído: wayfinding e aeroportos. 2009. 250 f. Tese (Doutorado) - Curso de Pós-graduação em Design, Artes & Design, Puc-rio, Rio de Janeiro, 2009. Cap. 2. Disponível em:

<https://doi.org/10.17771/PUCRio.acad.32541>. Acesso em: 27 nov. 2022.

RICHTER, J.; THOMAS, R.; LANGE, D.; GRAICHEN, T.; HEINKEL, U. Development of a Platform-independent Renderer for the Rendering of OpenStreetMap Indoor Maps in Flutter. **17th International Conference on Computer Graphics Theory and Applications**. Online Streaming GRAPP 2022: 6th – 8th feb. 2022.

ROY, N.; WANG, H.; CHOUDHURY, R. R. I am a smartphone and i can tell my user's walking direction. **Proceedings Of The 12th Annual International Conference On Mobile Systems, Applications, And Services - Mobisys '14**, [s.l.], p.329-342, 2014. Association for Computing Machinery (ACM). Disponível em:

<http://dx.doi.org/10.1145/2594368.2594392>. Acesso em: 22 nov. 2022.

SAFFO. **Sensors: The Next Wave of Infotech Innovation**. Paul Saffo. Disponível em: <http://www.saffo.com/essays/sensors-the-next-wave-of-infotech-innovation/>. Acesso em: 12 mai. 2022.

SANTOS, M. A. Localização em Ambientes internos utilizando PDR e Wi-Fi. 2018. Dissertação (Mestrado). Universidade Federal do Amazonas, Manaus 2018.

Disponível em: <https://tede.ufam.edu.br/handle/tede/6517>. Acesso em 28 nov. 2022.

SCARIOT, C. A. Avaliação de sistemas de informação para wayfinding: um estudo comparativo entre academia e mercado em Curitiba. 2013. 179 f. Dissertação (Mestrado) - Curso de Design, Universidade Federal do Paraná, Curitiba, 2013.

Disponível em: <http://dSPACE.c3sl.ufpr.br:8080/dSPACE/handle/1884/30375>. Acesso em: 02 abr. 2022.

SCHWABER, K. **Agile Project Management with Scrum**. Seattle: Microsoft Press, 2004.

SCHWABER, K.; BEEDLE, M. **Agile Software Development with Scrum**. New Jersey: Pearson, 2001.

SCHWABER, K.; Sutherland, J. **Guia do Scrum**. Disponível em: <https://scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf>. Acesso em: 09 mai. 2022.

SEGD. Society for Experiential Graphic Designs. Disponível em: <https://segd.org/what-wayfinding>. Acesso em: 03 abr. 2022.

SITUM. Indoor Maps. Disponível em: <https://situm.com/en/blog-eng/indoor-positioning/indoor-maps/>. Acesso em: 25 out. 2022.

SILVA, G. M. M.; SOSPEDRA, J. T.; Huerta, J. A Meta-Review of Indoor Positioning Systems. MDPI – **Open Access Journals**. Disponível em: <https://www.mdpi.com/1424-8220/19/20/4507>. Acesso em: 28 out. 2022.

SOMMERVILLE, I. **Engenharia de Software**. São Paulo: Pearson, 2011.

STAPLETON, J. **DSDM Dynamic Systems Development Method**. Harlow: Addison-Wesley, 1997.

TO DO. Disponível em: <https://todo.microsoft.com/pt-br/>. Acesso em: 23 nov. 2022.

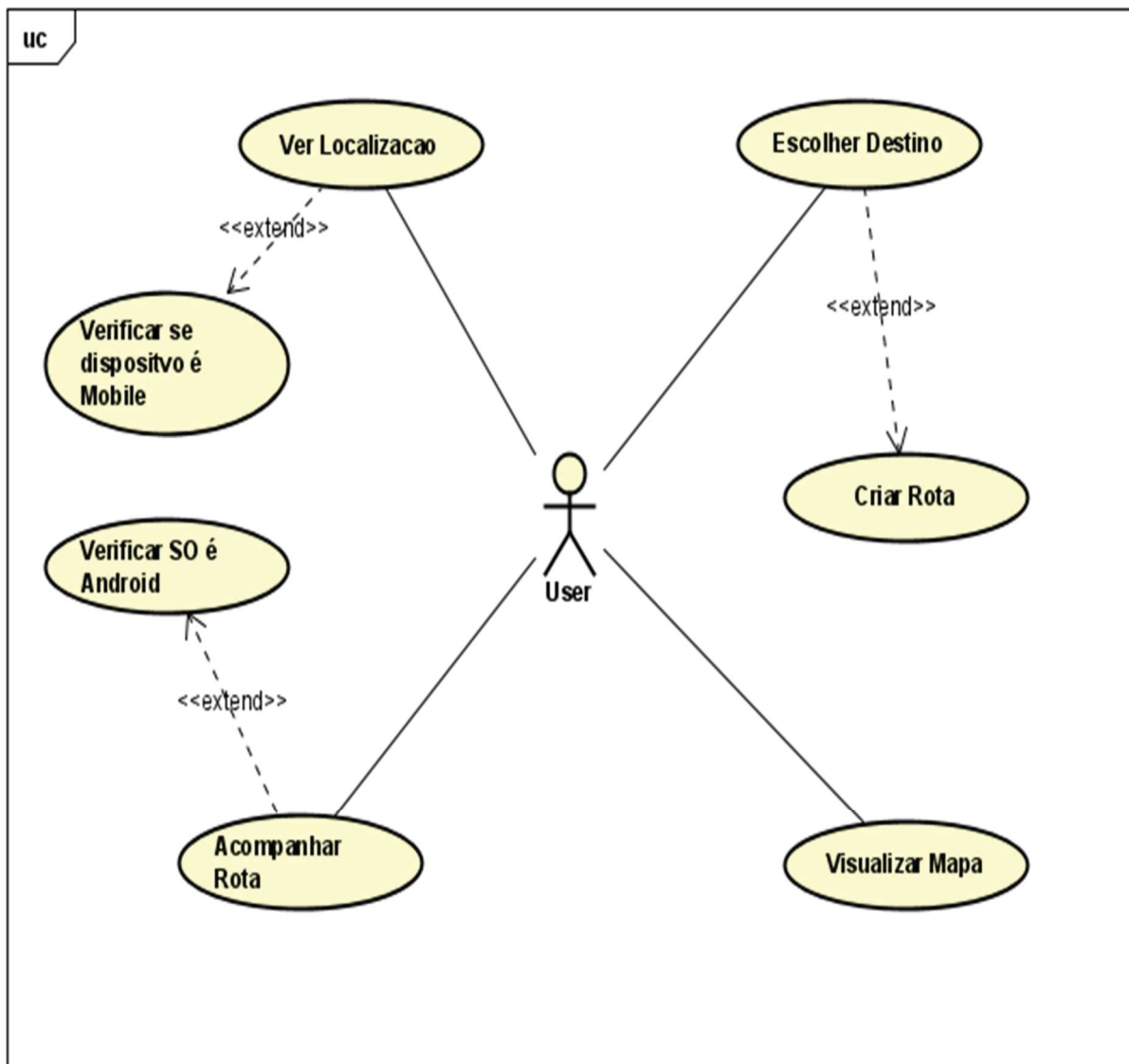
TOMTOM. **The History of navigation and what's next**. George de Boer Global Director Marketing Automotivo. 20 out. 2020. Disponível em: <https://www.tomtom.com/newsroom/explainers-and-insights/history-of-navigation/>. Acesso em: 24 mai. 2022.

VS CODE. Disponível em: <https://code.visualstudio.com/>. Acesso em: 23 nov. 2022.

ZAMMETTI, F. **Flutter na prática**. São Paulo: Novatec, 2020.

## APÊNDICE A – DIAGRAMA DE CASOS DE USO

Figura 37 - Diagrama de caso de uso



FONTE: O Autor (2022).

## APÊNDICE B – HISTÓRIAS DE USUÁRIO

Visualizar  
Mapa

**SENDO** um usuário que realizou a leitura do QR Code  
**OU** um usuário que realizou acesso pela interface web  
**QUERO** realizar o acesso ao mapa interno  
**PARA** poder consultar a estrutura interna  
**Tela:**



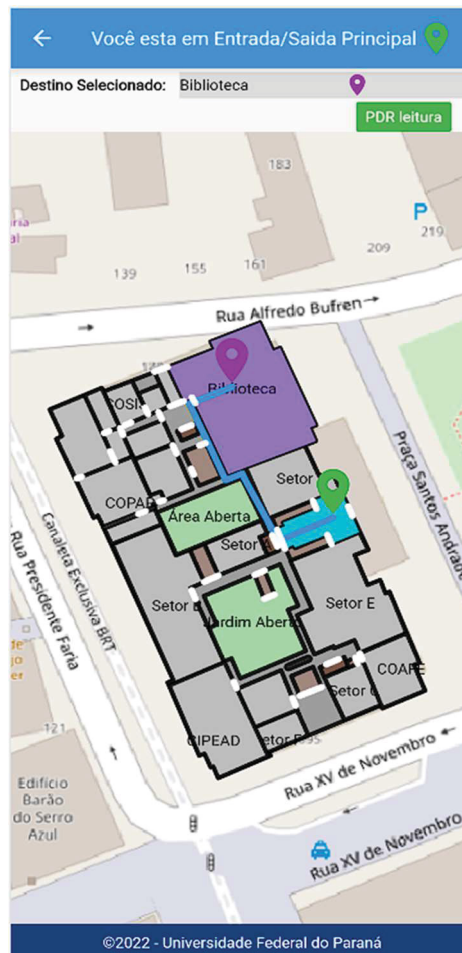
### CRITÉRIOS DE ACEITAÇÃO

- 1) Deve realizar o acesso web
- 2) Deve instalar o aplicativo no dispositivo
- 3) Deve, no modo, web clicar no botão “ler QR Code”
- 4) Deve, no aplicativo, realizar a leitura do QR Code.
- 5) Deve mostrar o mapa interno do prédio histórico da UFPR.
- 6) Deve mostrar a localização do QR Code, se no aplicativo
- 7) Deve mostrar a localização padrão como a entrada principal, se na interface web.
- 8) Não deve renderizar o botão “iniciar PDR”, na interface web.
- 9) Deve renderizar o botão “iniciar PDR”, no aplicativo.

Escolher  
Destino

**SEENDO** um usuário que realizou a leitura do QR Code  
**OU** um usuário que realizou acesso pela interface web  
**QUERO** escolher um destino  
**PARA** saber onde o destino se encontra no mapa

**Telas:**



### CRITÉRIOS DE ACEITAÇÃO

- 1) Deve mostrar uma lista dos destinos disponíveis
- 2) Deve destacar a área do destino no mapa quando selecionado
- 3) Deve destacar a área da localização atual.

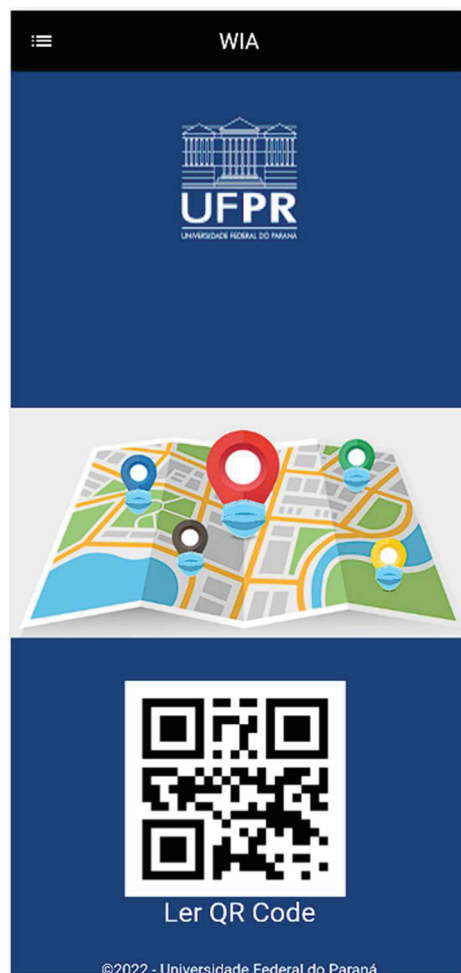
Ver  
localização

**SENDO** um usuário que realizou a leitura do QR Code

**QUERO** saber onde estou

**PARA** poder me localizar

**Telas:**



### CRITÉRIOS DE ACEITAÇÃO

- 1) Deve possuir um dispositivo para ler o QR Code
- 2) Deve ler o Qr Code
- 3) Deve enviar a localização do Qr Code para a tela do mapa.
- 4) Deve destacar a área referente a localização

Criar Rota

**SEDO** um usuário que realizou a leitura do QR Code  
**OU** um usuário que realizou acesso pela interface web  
**QUERO** ver a rota entre a minha localização e o destino selecionado

**PARA** poder escolher o melhor caminho

**E** não me perder

**Tela:**



### CRITÉRIOS DE ACEITAÇÃO

- 1) Deve selecionar um destino na lista de destinos
- 2) Deve mudar a cor da área do destino no mapa
- 3) Deve mudar a cor da área da localização atual.
- 4) Deve escolher o menor caminho entre a localização e o destino selecionado, destacando a rota a ser seguida no mapa.

Acompanhar  
Rota

**SEDO** um usuário que realizou a leitura do QR Code

**QUERO** visualizar meu deslocamento

**PARA** ver onde estou a todo o momento

Tela:

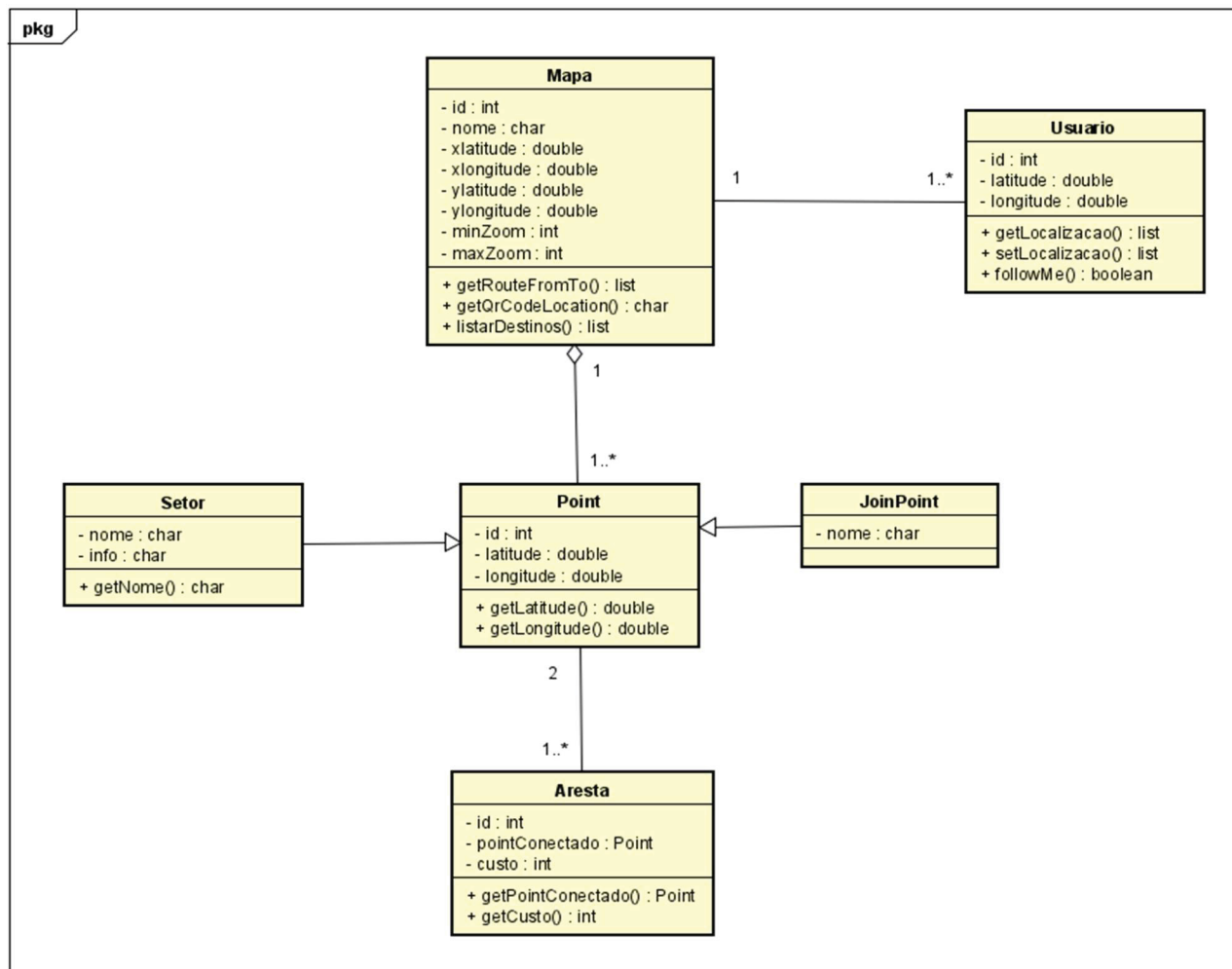


### CRITÉRIOS DE ACEITAÇÃO

- 1) Deve clicar no botão "iniciar PDR"
- 2) Deve mostrar um icone de cor preta.
- 3) Deve mostrar o deslocamento do icone análogo ao do usuário no mapa.

## APÊNDICE C – DIAGRAMA DE CLASSES

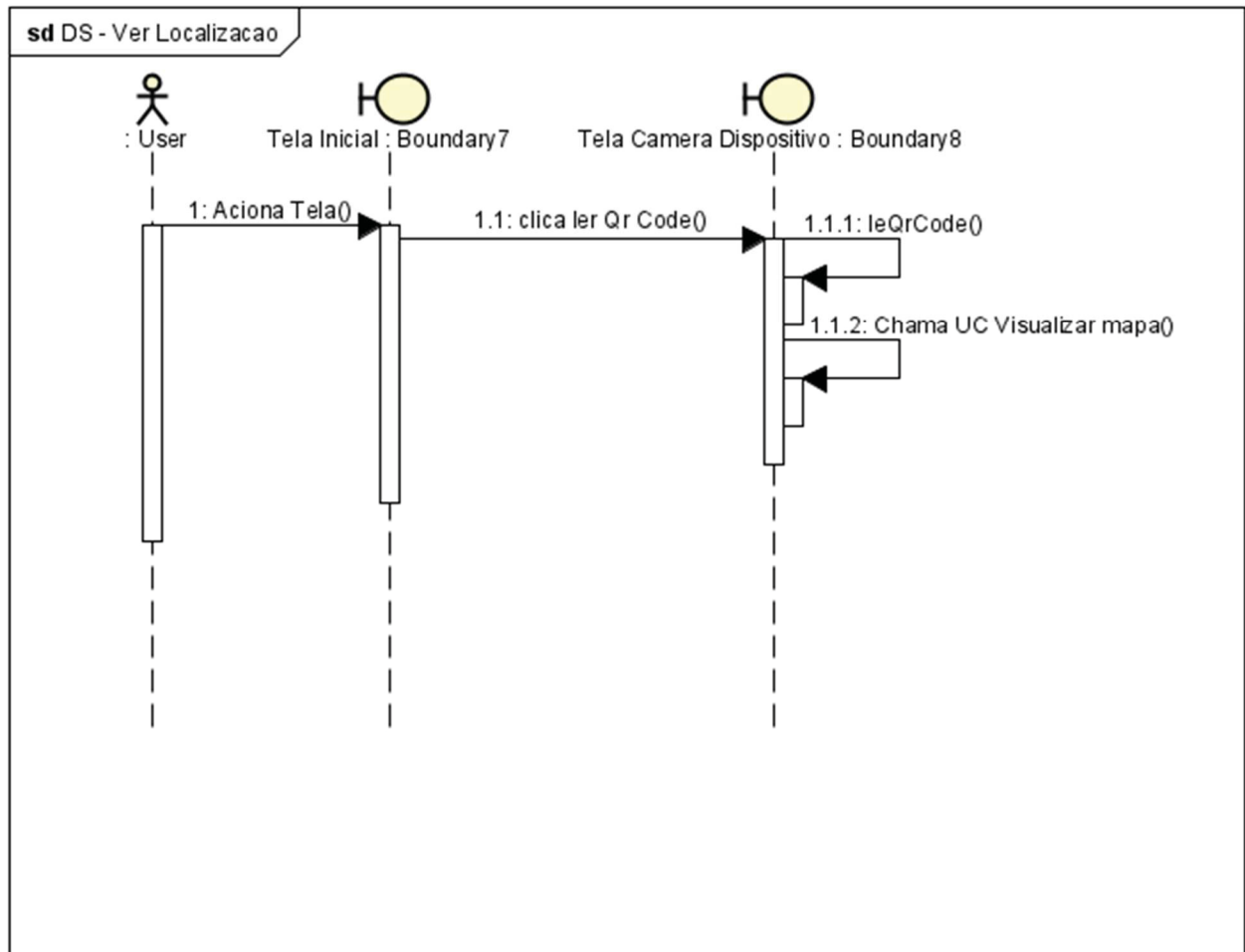
Figura 38 - Diagrama de classes



FONTE: O Autor (2022).

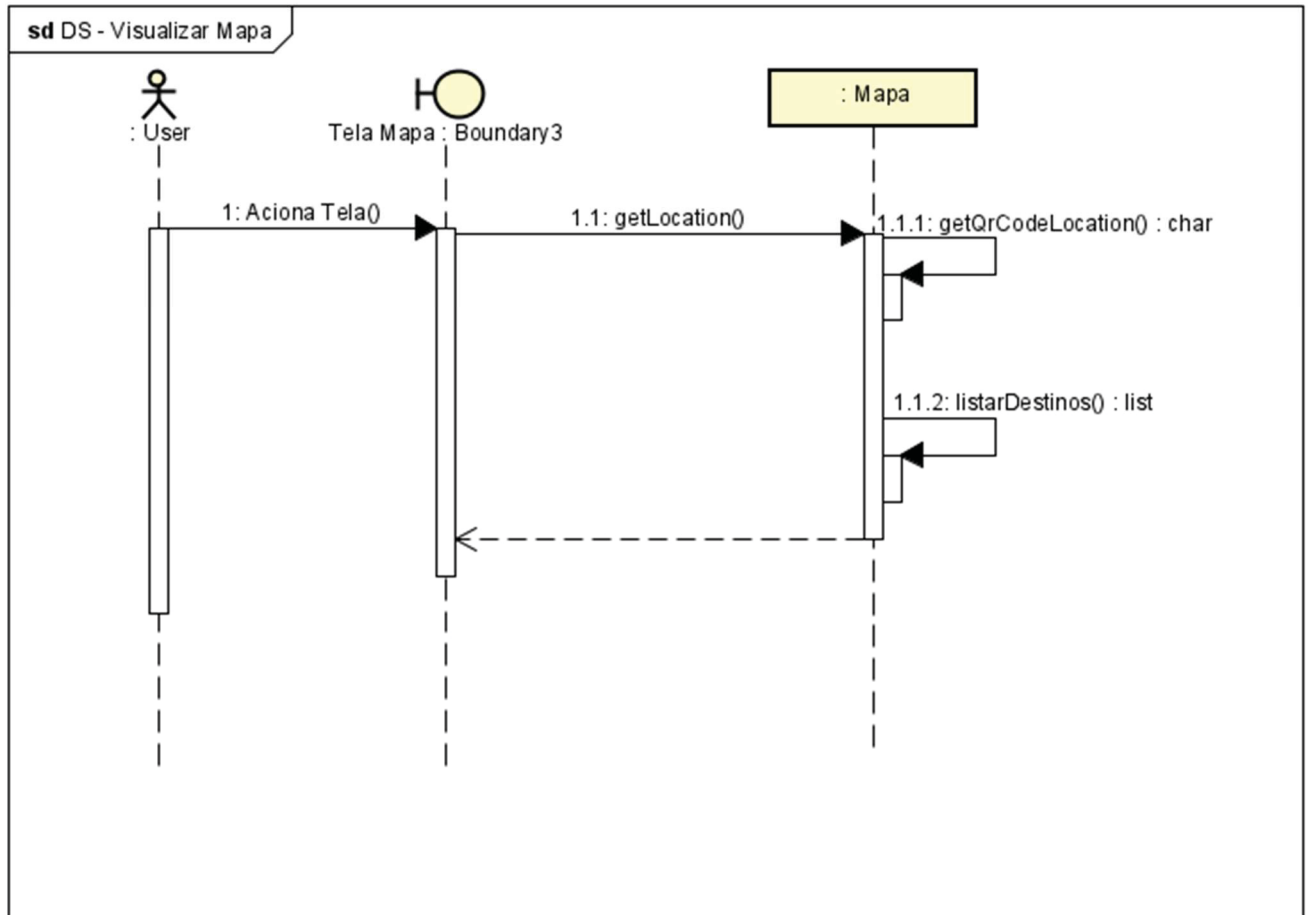
## APÊNDICE D – DIAGRAMA DE SEQUÊNCIA

Figura 39 - Diagrama de sequência Ver Localização



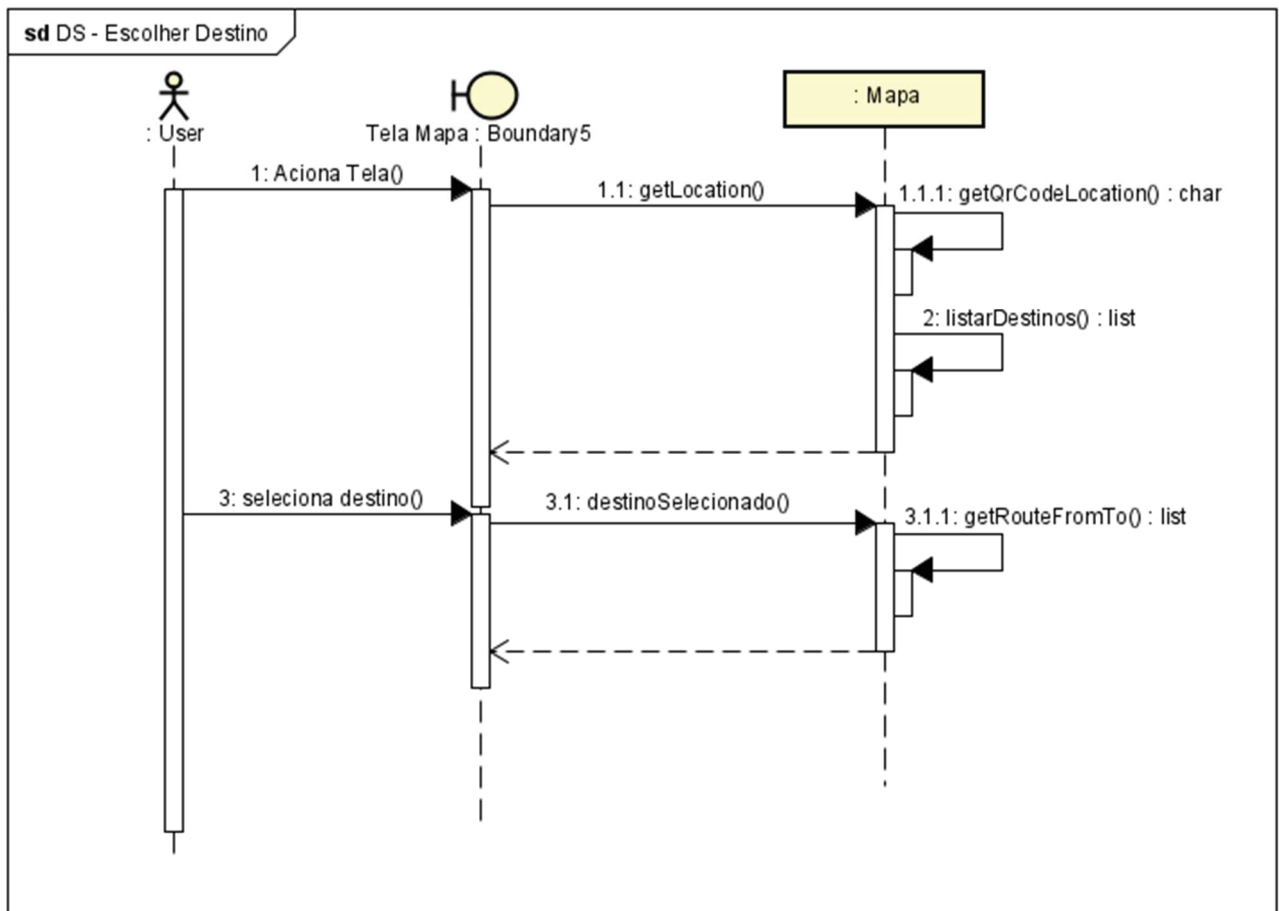
FONTE: O Autor (2022).

Figura 40 - Diagrama de sequência Visualizar Mapa



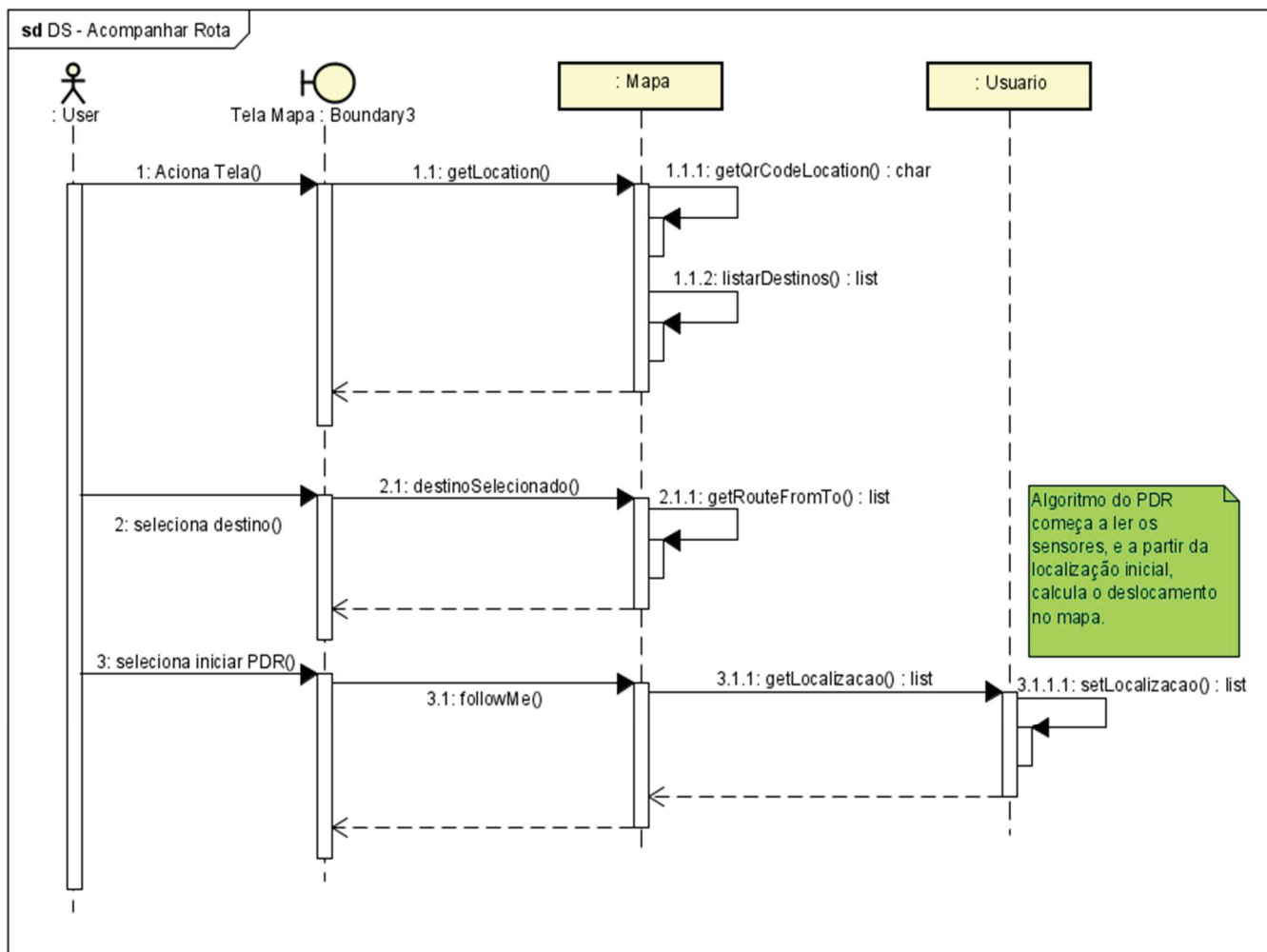
FONTE: O Autor (2022).

Figura 41 - Diagrama de sequência Escolher Destino



FONTE: O Autor (2022).

Figura 42 - Diagrama de sequência Acompanhar Rota



FONTE: O Autor (2022).