

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

MATHEUS STEFANELLO LUZ

**Compass: Aplicação para Visualização de
Pontos de Interesse em Realidade
Aumentada**

Monografia apresentada como requisito parcial
para a obtenção do grau de Bacharel em
Engenharia da Computação

Orientador: Prof. Dr. Claudio Fernando Resin
Geyer

Porto Alegre
2020

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Rui Vicente Oppermann

Vice-Reitora: Prof^a. Jane Fraga Tutikian

Pró-Reitor de Graduação: Prof. Vladimir Pinheiro do Nascimento

Diretora do Instituto de Informática: Prof^a. Carla Maria Dal Sasso Freitas

Coordenador do Curso de Engenharia de Computação: Prof. André Inácio Reis

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

“All we have to decide is what to do with the time that is given to us.”

— GANDALF

AGRADECIMENTOS

Gostaria de dedicar este trabalho aos meus pais, Edimar e Martha, pois sem eles e sem o seu apoio incondicional, nada disso seria possível. Também gostaria de dedicá-lo ao meu irmão, Nicholas, que me incentivou por toda minha jornada na faculdade.

Agradeço especialmente à minha namorada, Natália, por me apoiar, ajudar, incentivar e guiar durante todo esse tempo.

Agradeço também ao meu orientador, Professor Cláudio Geyer, pelos conselhos e ajuda durante a realização deste trabalho.

Por fim, agradeço aos meus amigos e colegas de aula e trabalho, os quais foram fundamentais para mim durante todo esse período.

RESUMO

A aplicação Compass foi desenvolvida a fim de oferecer uma alternativa à visualização padrão de mapas e estabelecimentos que é adotada hoje em dia, difundida amplamente por outras aplicações. Utilizando-se de conceitos modernos de realidade aumentada, detecção de movimentos e predição de rotas, a aplicação almeja criar uma nova experiência de uso para os usuários, onde os mesmos poderão estar imersos com o ambiente a sua volta, através de um maneira fácil e interativa. A aplicação procura entregar ao usuário informações relevantes sobre o que está a sua volta, bem como possíveis pontos de interesse que o ajudem na tomada de decisões sobre lugares e estabelecimentos a serem visitados e conhecidos. Criada para dispositivos iOS, a aplicação utiliza padrões e conceitos de engenharia de *software* na sua arquitetura, além de utilizar *frameworks* e APIs disponíveis no sistema operacional e em sua SDK. Após implementação e testes com a aplicação, é possível se afirmar de que a precisão dela é satisfatória, além de entregar o que se propôs inicialmente. Contudo, também é válido afirmar que existem pontos de melhoria, tanto de funcionalidade quanto de *design*, a fim de que a aplicação possa se tornar cada vez mais útil para o usuário.

Palavras-chave: Realidade aumentada. localização. smartphone. movimentação.

Compass: An Application for Point of Interest Visualization Using Augmented Reality

ABSTRACT

The Compass application was created in order to offer an alternative to the way people look at maps nowadays, that is widely spread by other applications. Using modern concepts of augmented reality, movement detection and route prediction, this application aims to create a new usage experience to the users, where they can be immersed into the environment around them, through an easy and interactive way. The application also aims to deliver to the user relevant information about what is around them and possible points of interest for them, which can help making decisions about places and establishments to be visited. Created for iOS devices, the application uses software engineering patterns and definitions, while using APIs and frameworks that are available on the operational system and its SDK. After finishing the application implementation and its testing, it is possible to say that its precision is satisfying, and it does deliver what it had proposed. However, it is also possible to say that there are some improvement points, both functionality-wise and design-wise, in order to make the application more useful to the user.

Keywords: augmented reality, localization, smartphones, movement.

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
AR	Augmented Reality
CPU	Central Process Unit
GPS	Global Positioning System
HMD	Head-Mounted Display
HWD	Head-Worn Display
IDE	Integrated Development Environment
LCD	Liquid-Crystal Display
MVC	Model-View-Controller
MVP	Model-View-Presenter
MVVM	Model-View-ViewModel
OST	Optical-See-Through
SDK	Software Development Kit
SO	Sistema Operacional
VR	Virtual Reality
VST	Video-See-Through

LISTA DE FIGURAS

Figura 2.1 <i>Head-Mounted Display</i>	16
Figura 2.2 <i>Virtual-Reality Continuum</i>	17
Figura 2.3 <i>Optical-See-Through</i>	17
Figura 2.4 <i>Video-See-Through</i>	18
Figura 2.5 Realidade aumentada aplicada na educação	19
Figura 2.6 Realidade aumentada aplicada na indústria de jogos	20
Figura 5.1 Diagrama da arquitetura MVC	29
Figura 5.2 Diagrama da arquitetura MVP.....	29
Figura 5.3 Diagrama da arquitetura MVVM	30
Figura 5.4 Diagrama da arquitetura MVVM utilizando Coordinators.....	31
Figura 6.1 Estrutura do servidor	33
Figura 6.2 Estrutura da aplicação.....	38
Figura 6.3 Tela inicial da aplicação.....	39
Figura 6.4 Experiência AR mostrando (a) a ESEFID e (b) a PUCRS	41
Figura 6.5 Medições de distância entre ponto de referência e (a) a ESEFID e (b) a PUCRS	42
Figura 6.6 Tela de detalhe de um estabelecimento	44
Figura 7.1 Precisão de medições e acurácia de marcações	45
Figura 7.2 Nível de correção da tela de detalhe	46
Figura 7.3 Nível de satisfação com a experiência de uso.....	46
Figura 7.4 Utilidade da aplicação na vida cotidiana	47

LISTA DE TABELAS

Tabela 3.1	Histórico de versões do iOS	21
Tabela 3.2	<i>Frameworks</i> presentes no conjunto <i>App Frameworks</i>	22
Tabela 3.3	<i>Frameworks</i> presentes no conjunto <i>Graphics and Games</i>	23
Tabela 3.4	<i>Frameworks</i> presentes no conjunto <i>App Services</i>	23
Tabela 4.1	Resumo dos frameworks utilizados	27
Tabela 6.1	Níveis de precisão na medição da posição	36
Tabela 6.2	Resultados da chamada de rede para coordenada geográfica	40

SUMÁRIO

1 INTRODUÇÃO	12
1.1 Motivação	13
1.2 Objetivo	13
1.3 Estrutura do Texto	14
2 REALIDADE MISTA	15
2.1 Origem	15
2.2 Realidade Virtual	15
2.2.1 <i>Head-Mounted Display</i>	16
2.3 Realidade Aumentada	17
2.3.1 Trabalhos Relacionados	18
3 TECNOLOGIA UTILIZADA	21
3.1 Origem	21
3.2 SDK para iOS	22
3.2.1 <i>App Frameworks</i>	22
3.2.2 <i>Graphics and Games</i>	22
3.2.3 <i>App Services</i>	22
4 FRAMEWORKS UTILIZADOS NA APLICAÇÃO	24
4.1 Visão Geral	24
4.2 <i>CoreLocation</i>	24
4.3 <i>CoreMotion</i>	24
4.4 <i>ARKit</i>	25
4.4.1 <i>ARSession</i>	25
4.4.2 <i>ARFrame</i>	25
4.4.3 <i>ARAnchor</i>	26
4.4.4 <i>ARWorldTrackingConfiguration</i>	26
4.5 <i>SpriteKit</i>	26
4.6 <i>Back-end</i>	26
5 APLICAÇÃO PARA DISPOSITIVOS MÓVEIS	28
5.1 Visão Geral	28
5.2 Padrões Arquiteturais	28
5.2.1 MVC	28
5.2.2 MVP	29
5.2.3 MVVM	30
5.3 Estrutura	31
5.3.1 Fluxo de navegação	31
6 SOLUÇÃO PROPOSTA	32
6.1 Visão geral	32
6.2 <i>Back-end</i>	33
6.2.1 API do Google Places	34
6.2.2 API do Facebook Places	34
6.3 Aplicativo iOS	34
6.3.1 Posição	34
6.3.2 Movimentação	35
6.3.2.1 Acelerômetro	36
6.3.3 Localização	36
6.3.4 Privacidade	37
6.3.5 Consumo de bateria	37
6.3.6 Estrutura da aplicação	37

6.3.7 Seleção do tipo de estabelecimento	38
6.3.8 Experiência AR.....	39
6.3.9 Detalhe do estabelecimento	43
7 RESULTADOS E DISCUSSÕES.....	45
8 CONCLUSÃO	48
REFERÊNCIAS	49
APÊNDICE A — QUESTIONÁRIO DIRECIONADO AOS USUÁRIOS DA APLICAÇÃO IOS.....	52
APÊNDICE B — TRABALHO DE GRADUAÇÃO I	54

1 INTRODUÇÃO

Realidade Aumentada é uma tecnologia que oferece um novo paradigma para interagir com um computador. Ela é tratada como uma variação de VR (*Virtual Reality*), que é definida como uma tecnologia de imersão onde, uma vez que o usuário esteja inserido no mundo virtual, não consegue mais ver o mundo real a sua volta. A Realidade Aumentada, por sua vez, almeja o contrário disso, adicionando e suplementando o mundo real, ao invés de mudá-lo completamente. O seu principal objetivo é aprimorar a percepção e interação do usuário através da adição de modelos 3D em um ambiente real, a fim de misturar objetos reais com objetos virtuais (AZUMA, 1997).

A tecnologia de realidade aumentada, apesar de poder ser implementada de várias maneiras diferentes, apresenta três principais componentes em comum: dados geoespaciais para o objeto virtual, uma superfície para projetar a imagem virtual e um *hardware* com poder de processamento para os gráficos, animações e manipulações de imagens, como, por exemplo, um computador com monitor (CARMIGNIANI et al., 2010).

Além dos três componentes, sistemas de realidade aumentada precisam de uma câmera capaz de capturar e processar o movimento do usuário e transferir esses dados capturados para os objetos virtuais, e, por fim, precisam de um *display* para mostrar tanto os objetos reais quanto os virtuais, todos já processados. Existem dois tipos de tecnologia de *display*: VST e OST (BOTELLA et al., 2005). A primeira funciona através da captura de objetos reais por uma câmera, onde os mesmos são mesclados com objetos virtuais e projetados em um monitor ou tela. A segunda tecnologia funciona através da adição de objetos virtuais em superfícies transparentes, como, por exemplo, vidros, os quais são sobrepostos em objetos reais, e é através deles que o usuário enxerga a realidade aumentada. A principal diferença entre as duas tecnologias é a latência, pois a primeira possui uma camada extra de processamento de objetos reais que a segunda não tem (CIPRESSO et al., 2018).

Atualmente, os computadores estão cada vez menores e mais poderosos. Os *smartphones* já são vistos como a nova geração de computadores e uma grande parte da população mundial faz uso deles diariamente (SWANSON; TAYLOR, 2011). Além do grande poder de processamento, os *smartphones* também podem ser considerados um tipo de tecnologia VST quando o assunto é realidade aumentada. Na rotina de uso dos usuários, é muito comum a utilização de aplicativos de mapas para locomoção e aplicativos de busca de estabelecimentos para lazer (MERRY; BETTINGER, 2019). Com base nisso,

a aplicação aqui descrita, chamada de Compass busca mudar a forma com que o usuário realiza tais buscas e interage com os aplicativos, valendo-se dos conceitos da realidade aumentada previamente descritos.

1.1 Motivação

O estudo e uso dessa tecnologia de realidade aumentada não é novo. Por outro lado, o número de aplicações comuns e populares se valendo dela ainda é baixo e apenas recentemente recebeu um aumento considerável no seu uso, o que se deve ao rápido avanço das tecnologias de *smartphones* (SHATTE; HOLDSWORTH; LEE, 2014). Existe uma vasta gama de áreas onde a realidade aumentada poderia ser utilizada para melhorar a experiência que o público tem em situações cotidianas. Uma dessas situações é a de viagens para lugares novos e desconhecidos. Mapas comuns podem auxiliar a encontrar estabelecimentos que possam ser do interesse do usuário e, além disso, ainda existem aplicações para sugestões de lugares a serem visitados. Porém, mesmo com todo o avanço na tecnologia dos *smartphones* e de realidade aumentada, não se encontrou na literatura uma solução que busque inserir o usuário na própria cidade de uma maneira diferente, se valendo de informações valiosas como sua posição, suas preferências e pontos de interesse da própria cidade.

Uma das propostas do Compass é mesclar os dois mundos: o mundo real, em que o usuário está visitando e não possui pleno conhecimento de rotas, ruas e lugares, e o mundo virtual, que possui dados como fotos, revisões, avaliações, recomendações, horários de funcionamento e planejamento de rotas. Uma vez que a aplicação é facilmente executada em dispositivos móveis, não é requerido nenhum equipamento adicional, apenas o próprio aparelho pessoal.

1.2 Objetivo

A criação da aplicação tem como objetivo principal proporcionar uma alternativa para a maneira atual de localização em ambientes não conhecidos. Através de marcações flutuantes ao redor do usuário com os pontos relevantes da cidade, o Compass busca a imersão do usuário no contexto a sua volta, levando o mesmo a realizar a opção por estabelecimentos baseado em mais do que proximidade, considerando lugares frequen-

temente visitados, lugares preferidos e de interesse do próprio usuário, além de fotos, vídeos e outros atributos que são relevantes na tomada de decisão. Com base em todas as informações disponíveis, tanto do usuário quanto dos pontos de interesse, a aplicação busca mostrar a cidade por outros olhos, através da tela de um *smartphone*. Busca-se também criar uma aplicação que utilize conceitos modernos de arquitetura e engenharia de software, focadas em aplicações para dispositivos móveis.

1.3 Estrutura do Texto

O texto está organizado da seguinte forma: o capítulo 2 apresenta estudos e definições sobre a tecnologia de realidade mista (virtual e aumentada). O capítulo 3 apresenta o sistema operacional iOS. O capítulo 4 apresenta os *frameworks* utilizados. O capítulo 5 apresenta definições sobre arquitetura e engenharia de *software* focados em aplicações *mobile*. O capítulo 6 apresenta a solução proposta. O capítulo 7 mostra os resultados e avaliações. O capítulo 8 apresenta a conclusão e possível trabalho futuro.

2 REALIDADE MISTA

Neste capítulo, a origem dos termos realidade virtual e aumentada será abordada, além de definir como cada um funciona e quando se aplica. Também se mencionará aplicações que utilizam tais tecnologias, e, por fim, serão expostos alguns dos trabalhos relacionados na área.

2.1 Origem

A tecnologia de realidade aumentada tem como suas raízes o campo de pesquisa de interfaces em computação (MEKNI; LEMIEUX, 2014), sendo considerada uma tecnologia proveniente dos estudos de realidade virtual, e, portanto, mais recente que esta.

2.2 Realidade Virtual

O conceito de realidade virtual foi definido pela primeira vez em meados de 1960, por Ivan Sutherland. Para ele, a realidade virtual, comumente chamada de VR, é uma janela na qual o usuário percebe o mundo virtual como se ele parecesse real e no qual ele pudesse agir de maneira realista (SUTHERLAND, 2001).

Para Michael Gigante, VR é a ilusão de participar em um ambiente sintético, ao invés de apenas observá-lo. VR implica em um visor estereoscópio 3D, que rastreia os movimentos da cabeça de um usuário, um rastreador para mãos e corpo e, por fim, um dispositivo de som binaural. Para ele, VR é uma experiência imersiva e multissensorial (GIGANTE, 1993).

Já para Fuchs e Bishop, VR é definido por gráficos em tempo real com modelos 3D, combinados com uma tecnologia de visor que proporciona imersão no mundo modelado e manipulação direta sob o mesmo (FUCHS; BISHOP, 1992).

Apesar de distintas, as três definições possuem três elementos em comum de sistemas de VR: imersão, sensação de presença no ambiente criado e possibilidade de interagir com o mesmo.

Existem três tipos básicos de tecnologia de VR:

- *Sistemas não-imersivos* são o mais simples tipo de aplicação VR pois utiliza monitores para reproduzir imagens do mundo real;

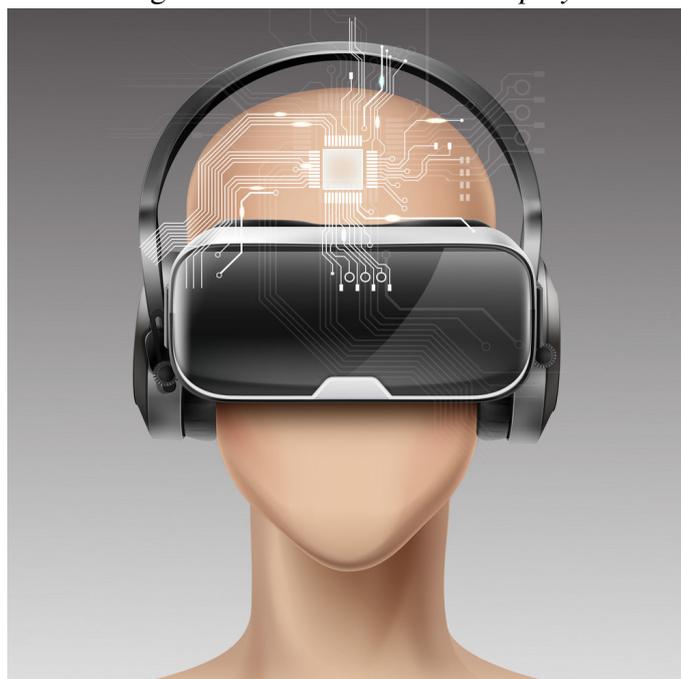
- *Sistemas semi-imersivos* proporcionam uma imagem estéreo de uma cena tri-dimensional, vista num monitor utilizando uma projeção em perspectiva à posição da cabeça do usuário;
- *Sistemas imersivos* proporcionam uma experiência simulada completa, fazendo uso de dispositivos com sensores, como, por exemplo, o HMD (*Head-Mounted Display*), que melhora a visão estereoscópica do ambiente, através do movimento da cabeça do usuário, e a sensação de imersão, através de áudio e *feedback* háptico.

Sistemas com alto grau de imersão, ao contrário dos outros dois sistemas, permitem que a interação e ações feitas sejam percebidas como ações reais ou confundidas com. Nesses sistemas, o uso de HMDs é essencial, e foi a partir deles que a tecnologia de realidade aumentada começou a ser pensada (CIPRESSO et al., 2018).

2.2.1 *Head-Mounted Display*

Os HMD são dispositivos de visualização usados na cabeça ou como parte de um capacete. Eles podem ser divididos em duas categorias: *monocular HMD*, o qual possui apenas um visor óptico, e *binocular HMD*, o qual possui dois visores, um para cada olho (SUTHERLAND, 2001). O HMD possui diversas aplicações, sendo as mais conhecidas para aviação, engenharia, medicina e indústria de jogos.

Figura 2.1: *Head-Mounted Display*



Fonte: VectorStock, 2019

2.3 Realidade Aumentada

O conceito de realidade aumentada foi definido décadas depois do VR, em 1994. Milgram e Kishino conceitualizaram o *Virtual-Reality Continuum*, o qual leva em consideração quatro sistemas: o ambiente real, a realidade aumentada (AR), a virtualidade aumentada e o ambiente virtual, também chamado de realidade virtual (VR). Para eles, AR pode ser definida como uma nova tecnologia, na qual objetos virtuais são adicionados ao mundo real e em tempo real durante a experiência do usuário (MILGRAM; KISHINO, 1994).



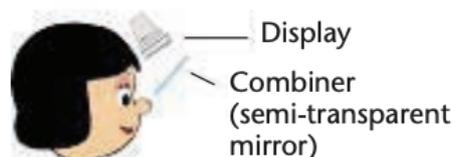
Fonte: Azuma et al. (2001)

Para Azuma et al., o termo AR é usado para descrever sistemas que possuem as seguintes características: 1) combinam o real com o virtual num ambiente real; 2) possuem interação em tempo real; 3) registram em três dimensões, onde o alinhamento preciso de objetos reais e virtuais é imprescindível (Azuma et al., 2001).

Ainda por Azuma et al., existem dois tipos básicos de tecnologia AR:

- *Head-Worn Displays (HWD)* são semelhantes aos HMDs da tecnologia VR. O dispositivo é colocado na frente dos olhos do usuário, e pode funcionar de duas maneiras distintas:
 - *OST*, que é constituído por um visor semitransparente e um projetor, o qual adiciona objetos virtuais no visor, sem obstruir a visão do usuário, criando, assim, a experiência de AR através de projeções em um plano entre o usuário e o mundo real.

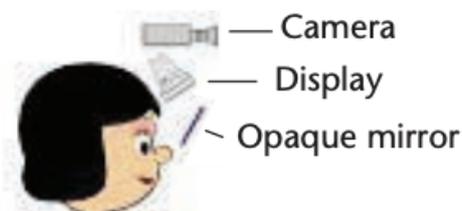
Figura 2.3: *Optical-See-Through*



Fonte: Azuma et al. (2001)

- *VST*, que é constituído por um visor opaco, um projetor e uma câmara. Ele funciona a partir da captação do mundo real através da câmara, somado com a adição de objetos virtuais e, por fim, projeção no visor opaco.

Figura 2.4: *Video-See-Through*



Fonte: Azuma et al. (2001)

No sistema HWD, a tecnologia mais utilizada é a de OST, como podemos ver em produtos já criados, como, por exemplo, o Google Glass.

- *Handheld displays* funcionam na sua grande maioria com a tecnologia VST. No início, funcionavam a partir de uma tela LCD com uma câmara acoplada, onde a imagem da câmara era projetada juntamente com os objetos virtuais, se comportando como uma espécie de janela para o mundo real, através da qual é possível observar a realidade aumentada.

O aumento no poder de processamento e redução do tamanho dos computadores, juntamente com o crescente número de pessoas utilizando dispositivos móveis, levou empresas fabricantes de *smartphones* e seus respectivos *softwares* a criarem *frameworks* que possibilitam o uso de celulares como dispositivos de realidade aumentada, o que os torna parte dessa categoria de tecnologia.

Com o crescente interesse das empresas fabricantes de celulares em tecnologias AR, o desenvolvimento e distribuição de aplicações AR ficou muito mais acessível. O equipamento necessário para a criação de aplicações não é mais um obstáculo como antigamente, então, com isso, pesquisadores começaram a propor soluções nas mais diversas áreas de conhecimento (SWANSON; TAYLOR, 2011).

2.3.1 Trabalhos Relacionados

A realidade aumentada aplicada a dispositivos móveis possibilita que a tecnologia seja utilizada fora de uma área pré-determinada com equipamentos já montados (ALEM; HUANG, 2011). Idealmente, isso possibilita a adição de camadas de informação a qual-

quer ambiente, em qualquer lugar (Sá; CHURCHILL, 2012). No capítulo 1, mencionou-se uma parte da vasta gama de possíveis aplicações utilizando a tecnologia de realidade aumentada. Pesquisadores das mais diversas áreas propuseram soluções que almejam mudar a maneira como enxergamos determinados assuntos, como, por exemplo, educação, medicina, turismo e até mesmo jogos eletrônicos.

As propostas na área de educação são inúmeras. Como descreve Yilmaz (2018), a tecnologia de realidade aumentada aplicada na educação pode enriquecer a experiência de aprendizado, pois ela possibilita o aumento na interação de quem está aprendendo com o conteúdo a ser aprendido, além de tornar o processo de aprendizado mais atrativo, efetivo, significativo e motivante. O nível das aplicações pode variar desde o ensino fundamental até o ensino superior, como pode ser visto, por exemplo, na proposta de um projeto para a disciplina de Desenho Técnico para os cursos de engenharia (OLIVEIRA; SILVA, 2019), onde um objeto a ser estudado é projetado virtualmente para que os alunos possam compreendê-lo de maneira mais assertiva.

Figura 2.5: Realidade aumentada aplicada na educação

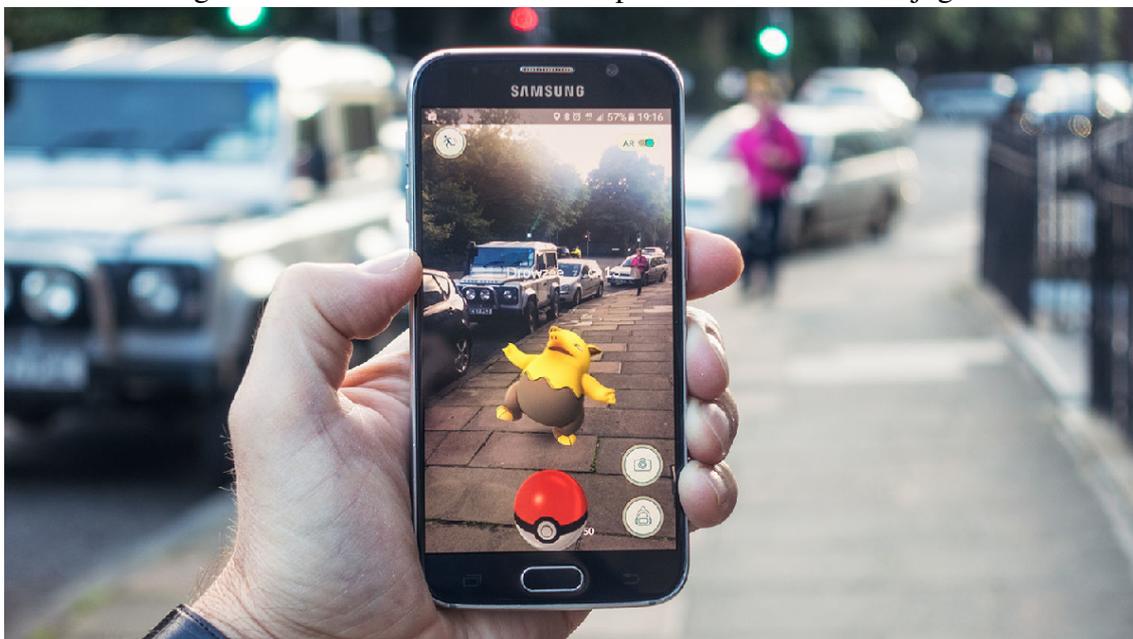


Fonte: Oliveira e Silva (2019)

Um jogo que se popularizou recentemente e utiliza recursos de realidade aumentada é o Pokémon GO (Niantic, 2019). Combinando técnicas de sobreposição do mundo real com o mundo virtual, juntamente com informações de localização do usuário, o jogo oferece uma jogabilidade e experiência que são totalmente diferentes dos jogos que não

fazem uso de tal tecnologia.

Figura 2.6: Realidade aumentada aplicada na indústria de jogos



Fonte: Getty Images, 2019

Na área médica, essas aplicações têm encontrado seu espaço no tratamento de problemas psicológicos como, por exemplo, na aplicação sugerida por (JUAN et al., 2005), em que usuários com aracnofobia podem ser expostos a modelos 3D de aranhas sem a necessidade da interação real com o animal, proporcionando, assim, meios de superar o medo. Em outra aplicação sugerida por (Escobedo et al., 2014), a tecnologia é empregada a fim de ajudar no tratamento e acompanhamento de crianças com autismo que possuem dificuldades para manter o foco e a atenção em determinados assuntos.

A área de turismo é a que mais se assemelha com a proposta desse trabalho. Existem algumas aplicações consolidadas que são responsáveis por mapear e informar sobre pontos turísticos ou históricos de uma determinada região. O aplicativo *Visit Orlando* (Orlando/Orange County Convention & Visitors Bureau, INC, 2019), por exemplo, mapeia os pontos de interesse dos parques de diversões mais famosos da cidade de Orlando utilizando a tecnologia de realidade aumentada, tornando mais fácil a navegação e visitação de seus clientes. Já o aplicativo *Florence Travel Guide* (Daniel Juarez Garcia, 2019) é responsável por mostrar ao usuário a rica história da cidade de Florença através de marcações flutuantes espalhadas por toda a cidade.

Assim como parte dos trabalhos relacionados, a proposta deste trabalho foi implementada para dispositivos móveis, especificamente para o sistema operacional iOS, utilizando tecnologias e *frameworks* próprios para a plataforma.

3 TECNOLOGIA UTILIZADA

Neste capítulo será abordada a origem do sistema operacional iOS, bem como os dispositivos que o utilizam hoje em dia. Além disso, serão apresentadas as ferramentas de desenvolvimento para o mesmo, das quais algumas foram utilizadas na construção da aplicação.

3.1 Origem

O iOS é um sistema operacional para dispositivos móveis criado e desenvolvido pela Apple Inc., cujo alvo é apenas o *hardware* proprietário da empresa. O sistema é hoje o segundo mais popular do mundo, atrás apenas do sistema operacional Android, e está disponível para iPhones, iPods e iPads, todos fabricados pela própria empresa.

Anunciado para o mundo em 2007, juntamente com o primeiro iPhone da empresa, o sistema se baseia numa interface de usuário com manipulação direta através de uma tela multi-toque. A primeira versão levou o nome de Iphone OS, e a partir da quarta versão, recebeu o nome de iOS, o qual é complementado com o número da versão, como pode ser visto na tabela 3.1.

Tabela 3.1: Histórico de versões do iOS

Nome do SO	Data de lançamento
Iphone OS 1	Junho, 2007
Iphone OS 2	Julho, 2008
Iphone OS 3	Junho, 2009
iOS 4	Junho, 2010
iOS 5	Outubro, 2011
iOS 6	Setembro, 2012
iOS 7	Setembro, 2013
iOS 8	Setembro, 2014
iOS 9	Setembro, 2015
iOS 10	Setembro, 2016
iOS 11	Setembro, 2017
iOS 12	Setembro, 2018
iOS 13	Setembro, 2019

Em conjunto com o lançamento da segunda versão do SO, foi lançada a SDK (*Software Development Kit*) para iOS, que possibilitou que desenvolvedores externos à Apple começassem a construir aplicativos para o novo sistema operacional da empresa. Aliada ao lançamento da AppStore, que pode ser definida como um serviço de distribuição

digital de aplicativos, pagos ou gratuitos, a SDK tornou a plataforma iOS a primeira escolha para novos desenvolvedores de aplicações *mobile*.

3.2 SDK para iOS

A SDK lançada vem evoluindo ao longo dos anos e serve como base de todo o desenvolvimento. Em conjunto com o Xcode, que é a IDE proprietária da empresa, ela possibilita que desenvolvedores criem aplicações utilizando as linguagens Objective-C e Swift, provendo vários *frameworks* com propósitos diferentes.

3.2.1 App Frameworks

É um conjunto de *frameworks* básicos para o desenvolvimento, com os mais relevantes descritos na tabela 3.2.

Tabela 3.2: *Frameworks* presentes no conjunto *App Frameworks*

Nome do <i>framework</i>	Funcionalidade
Swift	Possibilita o uso da linguagem de código aberto Swift em aplicações iOS
Foundation	Permite acesso a tipos essenciais de dados, coleções e funções de sistema
UIKit	Fornecer elementos básicos e avançados para implementação de interface visual

3.2.2 Graphics and Games

É um conjunto de *frameworks* para manipulação e processamento de imagens e para desenvolvimento de jogos, além de possuir ferramentas de modelagem de objetos, com os mais relevantes descritos na tabela 3.3.

3.2.3 App Services

É um conjunto de *frameworks* para serviços em geral da aplicação, como, por exemplo, localização e movimentação, com os mais relevantes descritos na tabela 3.4.

Tabela 3.3: *Frameworks* presentes no conjunto *Graphics and Games*

Nome do <i>framework</i>	Funcionalidade
ARKit	Facilita a criação de aplicações em realidade aumentada através de processamento avançado de cenas
Core Animation	Responsável por tratar e executar animações com alta taxa de quadros por segundo e baixo uso de CPU
SceneKit	Auxilia na criação de jogos 3D, com simulação de física, colisão e efeitos de partículas
SpriteKit	Auxilia na criação de jogos 2D

Tabela 3.4: *Frameworks* presentes no conjunto *App Services*

Nome do <i>framework</i>	Funcionalidade
Core Data	Responsável por fazer persistência de dados nas aplicações
Core Location	Oferece um serviço para determinar posição geográfica, altitude e orientação, utilizando múltiplos sensores do dispositivo
Core Motion	Processa leituras do acelerômetro, giroscópio, pedômetro e eventos relacionados ao ambiente
MapKit	Usado para mostrar mapas e imagens de satélites na aplicação, além de mostrar marcações e informações nos mesmos

Neste capítulo, além de descrever a origem do sistema operacional iOS, alguns dos serviços e *frameworks* disponíveis na sua SDK foram apresentados. No capítulo 4 serão citados e explicados os *frameworks* que foram de fato utilizados na construção da aplicação.

4 FRAMEWORKS UTILIZADOS NA APLICAÇÃO

Neste capítulo, como já mencionado anteriormente, serão apresentados todos os *frameworks* utilizados na construção da aplicação iOS, e, por fim, será também apresentado o *framework* utilizado na construção da aplicação *back-end*, a qual será melhor detalhada no capítulo 6.

4.1 Visão Geral

Frameworks são, por definição, abstrações de *software* que provém funcionalidades genéricas e que podem ser seletivamente especializadas ou sobrescritas por desenvolvedores ou usuários. Eles assumem a forma de bibliotecas e possuem uma API bem definida que é utilizável pela aplicação sendo desenvolvida. Seu propósito é facilitar e simplificar o ambiente de desenvolvimento, possibilitando que o foco dos desenvolvedores seja usado para requisitos de projeto ao invés de tarefas mundanas e repetitivas (GAMMA, 1995). No projeto da aplicação, alguns *frameworks* já citados serão utilizados com diversos fins, como, por exemplo, rastreamento de locomoção e movimentação do dispositivo, localização do mesmo e até mesmo para a funcionalidade de realidade aumentada.

4.2 CoreLocation

O CoreLocation providencia funcionalidades para determinar a latitude, longitude, altitude e orientação de um dispositivo. O *framework* obtém dados utilizando todos os componentes disponíveis no dispositivo, como, por exemplo, Wi-Fi, GPS, Bluetooth, magnetômetro, barômetro e antenas de comunicação.

4.3 CoreMotion

O CoreMotion reporta dados de movimentação e do ambiente, obtidos do *hardware* do dispositivo, o que inclui acelerômetros, giroscópios, pedômetro, magnetômetro e barômetro. Esses dados são disponibilizados para serem utilizados nas aplicações, tanto na forma não-processada, que é o valor bruto da mudança de velocidade em qualquer um dos eixos, quanto na forma processada, que é esse mesmo valor com filtros para suavizar

o movimento.

4.4 ARKit

O *framework* de realidade aumentada escolhido foi o ARKit 3. Ele combina a captura de cenas através da câmera, processamento avançado de cenas e utilidades de visor para simplificar a tarefa de construir a experiência AR.

A tecnologia AR presente no ARKit funciona a partir de um método chamado VIO, onde o *framework* utiliza os sensores do dispositivo, em conjunto com informações visuais da câmera, para rastrear o mundo real. Esse processo é responsável por calcular e armazenar informações e características do mundo 3D para a tela 2D, além de se manter consciente da sua localização, independente do movimento do iPhone.

O código do *framework* é estruturado para que a experiência AR seja descrita como uma sessão, a qual encapsula lógica e dados de um determinado período de atividade.

4.4.1 ARSession

Uma sessão ARSession é responsável por coordenar os cálculos a partir de dados obtidos, os quais são provenientes de câmeras e sensores de movimento. Ela também é responsável por sintetizar o resultado dos cálculos e estabelecer uma correspondência do o espaço existente no mundo real com o espaço AR modelado.

4.4.2 ARFrame

Possui os dados de quadro de vídeo e posição do dispositivo, os quais são passados para a ARSession. Também é responsável por mesclar os dados da imagem obtida com os sensores de movimento para calcular a posição estimada do iPhone.

4.4.3 ARAnchor

É uma posição no mundo real que é mantida independente do movimento ou posição da câmera, pois está ligada a um objeto fixo. Serve para tornar medições mais precisas, pois serve como âncora para adição de novos objetos virtuais.

4.4.4 ARWorldTrackingConfiguration

É a configuração necessária para obter a orientação e posição do dispositivo, além de permitir a detecção de planos e superfícies gravados pela câmera. Também é responsável por conectar o mundo real com o mundo virtual replicado pelo dispositivo, onde todos os objetos virtuais serão inseridos.

4.5 SpriteKit

O SpriteKit é um *framework* de propósito geral para desenhar formas, partículas, textos, imagens e vídeos em duas dimensões. Possui renderização de alta performance ao mesmo tempo que oferece uma API simples para a criação de aplicações que façam uso pesado de processamento gráfico.

A grande vantagem de utilizar esse *framework* é que o ARKit foi pensado e desenvolvido para funcionar de maneira simples, porém completa em conjunto com ele. O ARKit automaticamente transforma a cena AR em uma cena do SpriteKit, então a adição de objetos virtuais que mantenham a posição em relação ao mundo real em uma cena requer apenas a configuração correta de suas posições no SpriteKit.

4.6 Back-end

Por fim, para a criação do *back-end*, que é uma aplicação à parte da aplicação iOS, apenas um *framework* será utilizado, que é o Node.js. Ele foi criado a fim de prover uma ferramenta para construção de aplicações de rede rápidas e escaláveis utilizando código *JavaScript* de maneira assíncrona. Ele é capaz de gerenciar um número elevado de conexões simultâneas com alta taxa de transferência, o que é fundamental para alta escalabilidade (Node.js, 2019). Além disso, esse *framework* possui um gerenciador de

pacotes próprio, o *Node Package Manager* (NPM), o qual possui inúmeros componentes publicamente disponíveis para uso (npm, Inc, 2019). Um desses pacotes é o *Express*. Ele é responsável por abstrair grande parte da lógica de roteamento necessária para criar um servidor, o que torna a aplicação menos complexa (Express, 2019).

Tabela 4.1: Resumo dos frameworks utilizados

Framework	Plataforma	Finalidade
CoreLocation	Aplicativo iOS	Obter dados de localização
CoreMotion	Aplicativo iOS	Obter dados de movimentação
ARKit	Aplicativo iOS	Criar experiência de realidade aumentada
SpriteKit	Aplicativo iOS	Renderizar objetos virtuais
Node.js	<i>Back-end</i>	Criar aplicação de rede
Express	<i>Back-end</i>	Abstrair lógica de roteamento

5 APLICAÇÃO PARA DISPOSITIVOS MÓVEIS

Neste capítulo, serão discutidas boas práticas na criação de uma aplicação e padrões de projeto. Também serão apresentados e discutidos padrões arquiteturais voltados para aplicações *mobile*, com os pontos positivos e negativos de cada um.

5.1 Visão Geral

Aplicações para dispositivos móveis, assim como aplicações para outras tecnologias, possuem técnicas e práticas para manter o projeto estruturado, escalável, testável e de fácil manutenção. A base para obter tais padrões de qualidade se dá através de uma arquitetura sólida, a qual será mantida durante toda a vida útil da aplicação e do projeto.

5.2 Padrões Arquiteturais

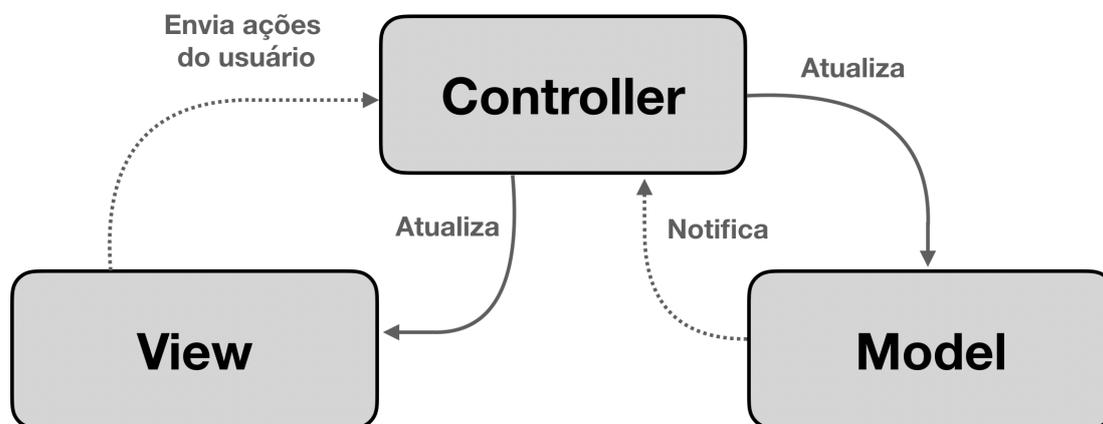
Os padrões de arquitetura de uma aplicação para plataformas *mobile* podem ser divididos, principalmente, em três tipos: MVC, MVP e MVVM. Cada um possui casos de usos e vantagens diferentes, os quais serão abordados a seguir.

5.2.1 MVC

O modelo MVC foi o primeiro modelo a ser estabelecido. Ele funciona através de separações de responsabilidade, onde o componente *Controller* é um mediador entre o componente *View* e o componente *Model*. O *View* é a representação visual dos dados da aplicação e é responsável por notificar o *Controller* de interações do usuário com a aplicação. O *Model* é apenas responsável por guardar dados e notificar o *Controller* sobre mudanças dos mesmos (FOWLER, 2002).

Esse modelo é o mais simples e de mais rápida implementação dos três, porém ele possui problemas de acoplamento de componentes, pois o *Controller* está diretamente ligado ao ciclo de vida do componente *View*, o que dificulta a testabilidade e escalabilidade da aplicação.

Figura 5.1: Diagrama da arquitetura MVC

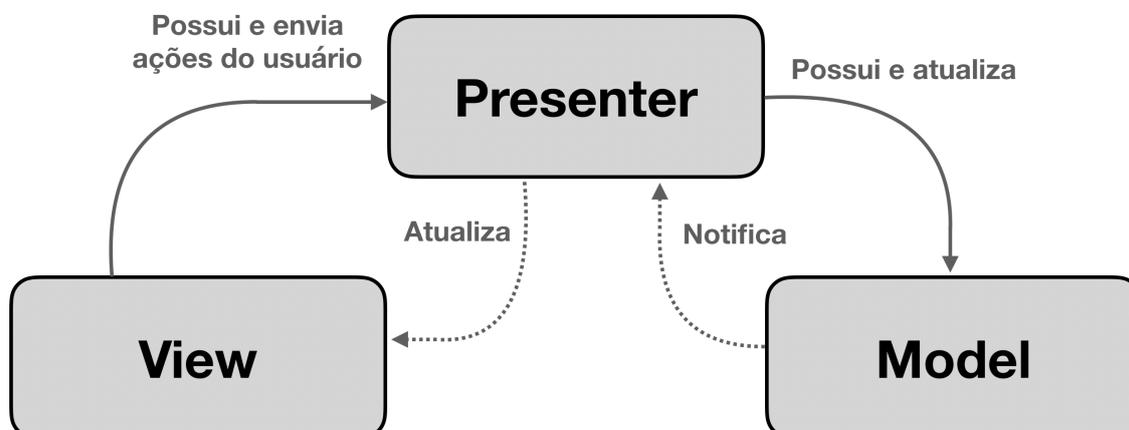


Fonte: O Autor, 2019

5.2.2 MVP

O modelo MVP é derivado do MVC. Seu principal objetivo é resolver o problema de acoplamento entre *View* e *Controller* que existe no MVC. A principal diferença entre os dois modelos é que o *Controller* é substituído pelo componente *Presenter*. O *Presenter*, ao contrário do *Controller*, é possuído pelo componente *View*, e sua responsabilidade é receber notificações de interações de usuário com o *View* e atualizações de dados do *Model* e prover uma forma de comunicação entre os dois. O mesmo possui a lógica de negócios da aplicação e também é responsável por disparar serviços e chamadas de rede e formatar dados para apresentação (KOURAKLIS, 2016).

Figura 5.2: Diagrama da arquitetura MVP



Fonte: O Autor, 2019

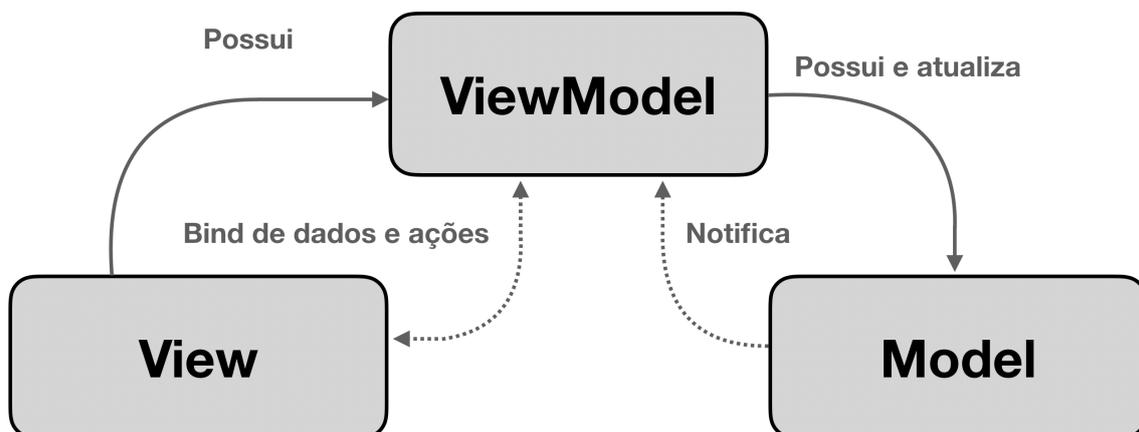
Apesar de a arquitetura MVP solucionar o maior problema do MVC, ela é orientada a interfaces, então todo tipo de comunicação entre *Presenter* e *View* precisa ser expli-

citamente descrita pelo *View*, o que o torna não-escalável. Nessa arquitetura, o problema com testes é solucionado, pois é possível testar comportamento de interface e lógica de negócios separadamente.

5.2.3 MVVM

O modelo MVVM é o mais recente e se propõe a solucionar os problemas dos outros dois, MVC e MVP. Novamente, o novo componente substitui o *Controller* e o *Presenter*, e se chama *ViewModel*. Como o nome sugere, ele é uma representação independente do componente *View* e do seu estado. Ele também resolve o problema de interfaces através de *databinding*, o que significa que o *ViewModel* é responsável por invocar mudanças no *Model* e se atualizar de acordo com as mudanças do mesmo. Além disso, ao contrário do MVP, que invocava mudanças no *View* através de interfaces de comunicação, o *View* se atualiza sozinho apenas observando um *stream* de dados, os quais foram atualizados de acordo com as mudanças do *Model* (KOURAKLIS, 2016).

Figura 5.3: Diagrama da arquitetura MVVM



Fonte: O Autor, 2019

Com base nos benefícios de cada uma das arquiteturas e também nos problemas existentes em cada, a arquitetura escolhida para o desenvolvimento da aplicação foi a MVVM, que, apesar de mais complexa, possui maior escalabilidade e testabilidade dentre as três.

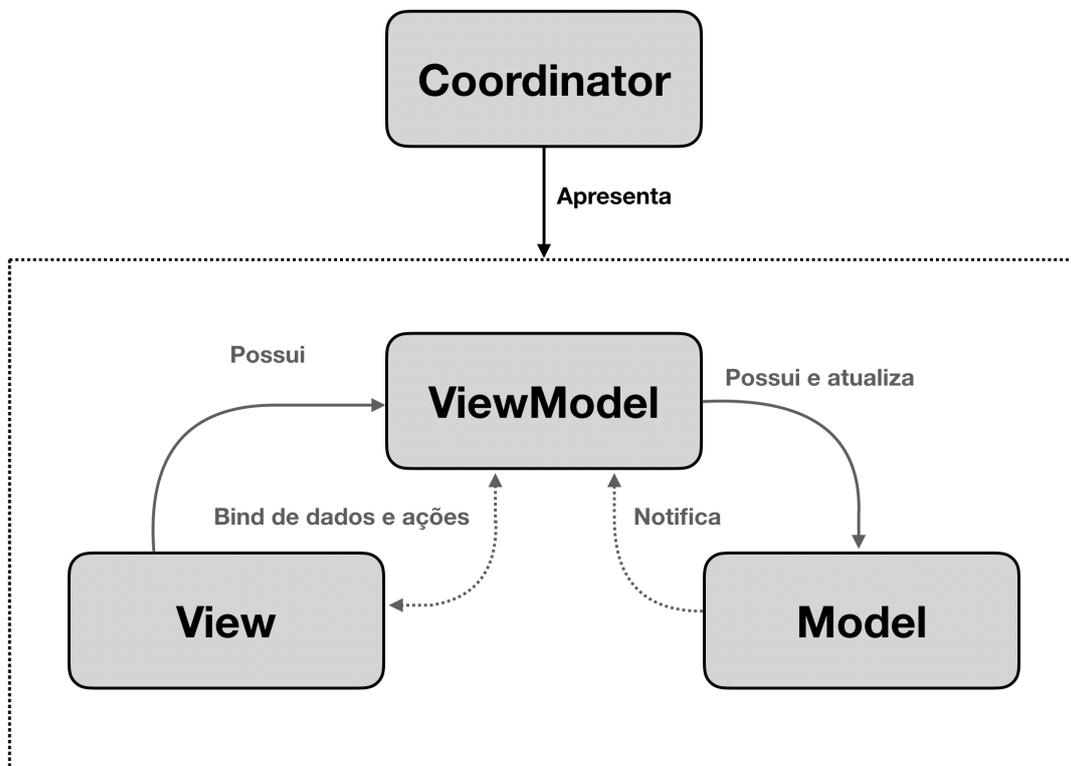
5.3 Estrutura

Com a arquitetura escolhida, existe uma última decisão a ser tomada, que é como estruturar a aplicação em relação ao fluxo de navegação.

5.3.1 Fluxo de navegação

Para o fluxo de navegação, o padrão a ser utilizado em conjunto com o MVVM é o *Coordinator*. Sem utilizar um padrão de fluxo, os componentes de *View* acabam precisando conhecer uns aos outros para criar uma navegação na aplicação. O padrão *Coordinator* propõe que cada *View* possua um componente chamado de *Coordinator* que é exclusivamente responsável pela sua lógica de navegação. Para esse novo componente, é delegada a responsabilidade de saber qual o próximo passo de um fluxo, enquanto o componente *View* somente o avisa de que deseja navegar.

Figura 5.4: Diagrama da arquitetura MVVM utilizando Coordinators



Fonte: O Autor, 2019

A partir da implementação desse padrão, consegue-se desacoplar ainda mais a aplicação e resolver o último problema estrutural da mesma.

6 SOLUÇÃO PROPOSTA

Neste capítulo, será apresentada a aplicação que foi desenvolvida, abordando a sua estrutura, tecnologias utilizadas e funcionalidades presentes.

6.1 Visão geral

O sistema foi dividido em duas aplicações principais, o *front-end* e o *back-end*. O *front-end* consiste em um aplicativo iOS, que é instalado em iPhones e possui a responsabilidade de coletar dados sobre posição e movimentação do usuário, bem como receber dados externos, processá-los e criar uma interface para o usuário com base neles.

O fluxo de utilização dessa aplicação pode ser resumido em três passos, os quais serão melhor detalhados e explicados ao longo do capítulo:

- **Home.** É uma tela onde o usuário seleciona qual o tipo de ambiente que ele gostaria de buscar, como, por exemplo, restaurantes ou universidades, que redireciona para a tela de AR.
- **Experiência AR.** Essa tela é a parte da aplicação que busca mudar a maneira comum de visualizar mapas, pois, ao invés de apenas mostrar os pontos de interesse buscados na primeira parte e mostrá-los no formato de lista ou mapa, a aplicação insere múltiplas marcações flutuantes ao redor do usuário em um mundo real simulado através do celular do mesmo, utilizando a imagem da câmera para montar a experiência de aumentar a realidade. O usuário poderá mover o celular em qualquer direção, enxergando marcações que estão na direção apontada. Toda marcação é clicável, o que leva ao terceiro passo da aplicação.
- **Detalhe de um estabelecimento.** Quando uma marcação é clicada, uma tela de detalhe é apresentada. Essa tela possui nome do estabelecimento, descrição, avaliações, fotos, horário de funcionamento e distância para o local.

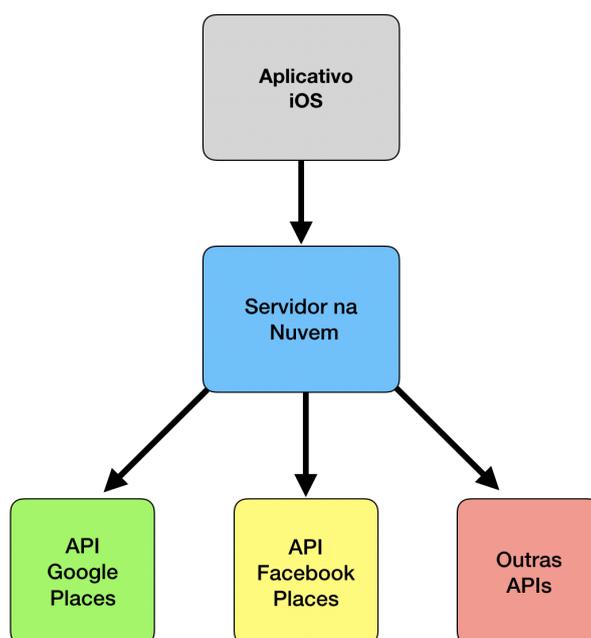
O *back-end* consiste em um servidor Node que é responsável por receber chamadas de rede, buscar dados em fontes diversas, processá-los, modelá-los e retorná-los para a aplicação *front-end*.

6.2 Back-end

O *back-end* é um servidor armazenado na nuvem, que utiliza os *frameworks Node.js* e *Express*. Ele é responsável por receber, processar e modelar dados para a aplicação. A comunicação com o dispositivo é feita através da construção de uma API, que permite que o aplicativo faça buscas utilizando as informações obtidas no próprio dispositivo, sem a necessidade de comunicar diretamente com APIs externas. Ele também é responsável por prover diferentes rotas para obtenção de informações. A principal funcionalidade do servidor é receber um pedido de pontos de interesse para uma determinada localização, buscar informações sobre essas localizações em diferentes APIs públicas, receber essas informações e modelá-las para que o aplicativo possa consumi-las sem a necessidade de um processamento posterior.

Para atingir tal objetivo, foi necessário que o servidor fosse feito de maneira modular. Cada API pública a ser adicionada como fonte de informação tem um adaptador para que os dados obtidos sejam compatíveis com os outros já armazenados previamente. Diferentes APIs podem ser agregadas ou removidas do processo de obtenção de informações. Outra funcionalidade do servidor é a de prover para o aplicativo uma lista de tipos de estabelecimentos possíveis para determinada localização, a fim de que o aplicativo seja capaz de realizar filtros com os mesmos. A figura 6.1 contém um diagrama arquitetural do servidor.

Figura 6.1: Estrutura do servidor



Fonte: O Autor, 2019

6.2.1 API do Google Places

A primeira API escolhida para montar o objeto de resposta foi a do Google Places. Essa é responsável por fornecer informações como latitude, longitude, nome do estabelecimento, nome da rua e avaliação dentro da plataforma Google. A API também é responsável por fornecer a lista de tipos de lugares, além de permitir buscas filtradas por esses tipos.

6.2.2 API do Facebook Places

A outra API escolhida para a aplicação foi a do Facebook. A partir dela, é possível buscar informações sobre um lugar baseado em sua localização geográfica e nome, informações que são obtidas via a primeira API. Com a API do Facebook, é possível obter fotos do lugar, comentários sobre o mesmo, descrição, horário de funcionamento, entre outros. Apesar de ser uma API independente da API do Google, a busca realizada nela é com base em dados previamente obtidos na primeira API, a fim de complementar esses dados com informações mais detalhadas.

6.3 Aplicativo iOS

O aplicativo é responsável por coletar os dados de localização e movimentação dos sensores, utilizando-se de APIs e *frameworks* nativos do sistema operacional. Além disso, o aplicativo também tem a responsabilidade de calcular e exibir anotações na tela do dispositivo baseado nesses dados coletados previamente, juntamente com informações externas que serão obtidas por chamadas de rede. Por fim, dois fatores foram levados em consideração para a construção da aplicação, que são a privacidade do usuário e o consumo de bateria pelo próprio aplicativo.

6.3.1 Posição

A habilidade de estimar precisamente a posição do usuário é fator chave para o funcionamento da aplicação. Portanto, empregar uma tecnologia de posicionamento (também conhecida como *location-awareness*) (TSENG et al., 2001) se fez necessário.

Embora existam inúmeras maneiras de estimar a posição, elas podem ser classificadas em duas categorias: auto-posicionamento e posicionamento remoto.

As tecnologias de auto-posicionamento funcionam através do recebimento de sinais de antenas emissoras geograficamente distribuídas, que podem ser tanto terrestres quanto satélites. Com esses sinais recebidos, o dispositivo é capaz de calcular a sua própria posição, possibilitando que aplicações possam utilizar tal informação calculada pelo mesmo para a tomada de decisões onde o quesito posição é relevante. Já a tecnologia de posicionamento remoto funciona a partir de medições de sinais que saem do emissor, chegam no dispositivo e voltam para o emissor, ou seja, os emissores sabem onde está o dispositivo (ZEIMPEKIS; GIAGLIS; LEKAKOS, 2003).

O modelo de auto-posicionamento é o mais apropriado para a aplicação Compass, pois é necessário que se saiba a localização do próprio dispositivo. Existem inúmeras técnicas para o receptor determinar a sua posição, sendo a mais comum entre elas o GPS.

O GPS é um sistema mundial de navegação baseado em satélites, onde esses transmitem a informação de navegação e o receptor as utiliza para calcular a sua posição, com uma precisão de menos de 10 metros. Existem algumas limitações para o uso do GPS, pois, para o mesmo operar de maneira satisfatória, é necessário visão não-obstruída do céu e sinal de pelo menos três satélites (DJUKNIC; RICHTON, 2001). Como a aplicação Compass visa o uso externo e não interno, a tecnologia de GPS é adequada e foi utilizada durante o desenvolvimento.

6.3.2 Movimentação

Além de saber onde está posicionado o aparelho, é vital para a aplicação poder determinar a direção, sentido e intensidade do movimento de um usuário. Para isso, foi utilizado um dos sensores do aparelho, o acelerômetro, pois ele possui as informações necessárias para determinar a orientação do dispositivo. Para obter informações sobre movimentos sendo aplicados ao dispositivo, o *framework* utilizado foi o *CoreMotion*, introduzido no capítulo 4. Ele é responsável por prover acesso à informação de rotação, aceleração e orientação do dispositivo.

6.3.2.1 Acelerômetro

Um acelerômetro é um dispositivo que mede forças de aceleração. Essas forças podem ser constantes, como, por exemplo, a força da gravidade, ou podem ser dinâmicas, causadas por algum movimento ou vibração externo ao dispositivo (ALSAEDI, 2016). Existem múltiplos tipos de acelerômetros desenvolvidos, porém os aparelhos que utilizam a aplicação trabalham com o modelo de três eixos.

O acelerômetro de três eixos proporciona medições simultâneas em três direções ortogonais, que, no caso, representam os eixos x , y e z no plano cartesiano. Uma medição estática da gravidade incidente sobre o acelerômetro pode ser analisada e transformada em um ângulo de inclinação do dispositivo, enquanto medições dinâmicas podem ser transformadas em direção e sentido do movimento do mesmo (ALLAN, 2011).

6.3.3 Localização

Para obter informações sobre a posição do dispositivo, o *framework* utilizado foi o *CoreLocation*, que foi introduzido no capítulo 4. Ele é responsável por prover a posição geográfica do dispositivo utilizando alguns sensores do mesmo. Para utilizá-lo, é necessário que o usuário permita a captação de tais dados. Existem diferentes níveis de precisão para a *API*, que serão mostrados juntamente com uma breve descrição na Tabela 6.1.

Tabela 6.1: Níveis de precisão na medição da posição

Classificação	Descrição
kCLLocationAccuracyBestForNavigation	O maior nível de precisão, porém utiliza outros sensores para auxiliar aplicativos de navegação
kCLLocationAccuracyBest	O maior nível de precisão sem auxílio de outros sensores
kCLLocationAccuracyNearestTenMeters	Precisão de até 10 metros
kCLLocationAccuracyHundredMeters	Precisão de até 100 metros
kCLLocationAccuracyKilometer	Precisão de até 1 quilômetro
kCLLocationAccuracyThreeKilometers	Precisão de até 3 quilômetros

Quanto mais precisa for a medição, maior será o consumo de energia. Como a aplicação está baseada em medições precisas, o parâmetro escolhido será o de *kCLLocationAccuracyBestForNavigation*, pois o mesmo utiliza sensores como o giroscópio, acelerômetro e magnetômetro para reduzir erros e aumentar a precisão da medição.

6.3.4 Privacidade

A localização atual do usuário é um dado bastante sensível e pessoal, portanto não deve ser armazenado por nenhuma aplicação que não esteja no próprio dispositivo. A comunicação com o servidor precisa desses dados para fazer a chamada baseada em localização, mas o mesmo apenas executa buscas com essa informação, sem se preocupar com o armazenamento dela.

6.3.5 Consumo de bateria

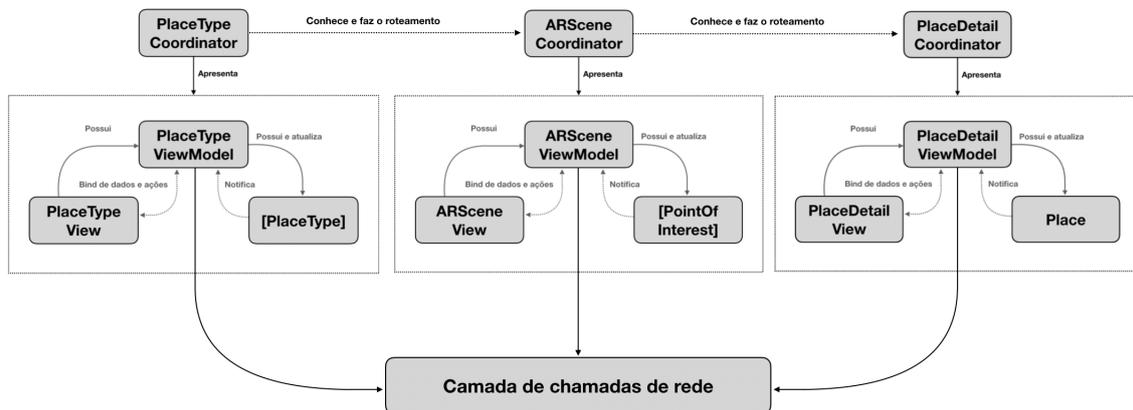
Um fator importantíssimo em qualquer aplicação é a otimização do consumo de bateria. A aplicação possui basicamente três estados: *foreground*, *background* e não inicializada. Uma estimativa de consumo pode ser feita para cada um dos estados:

- **Foreground.** Quando a aplicação está em *foreground*, ela está inicializada e é a aplicação sendo utilizada no momento. Nesse caso, todas as medições e cálculos estão sendo executados, então a estimativa de impacto na bateria pode ser considerada moderada/alta;
- **Background.** Quando a aplicação está em *background*, ela está inicializada mas outra aplicação está sendo utilizada no momento. Como a aplicação trabalha com a visualização de dados em tempo real, não há necessidade de processamento enquanto ela não está sendo utilizada, então a estimativa de impacto na bateria pode ser considerada mínima;
- **Não inicializada.** Quando a aplicação não está inicializada, o impacto na bateria é nulo.

6.3.6 Estrutura da aplicação

Baseada em boas práticas de arquitetura, engenharia de *software* e utilizando os *frameworks* mencionados, a estrutura da aplicação está demonstrada na figura 6.2. Utilizou-se o padrão arquitetural MVVM, em conjunto com o *Coordinator*, com uma camada de chamadas de rede separada, com as quais somente as *ViewModels* podem se comunicar. Além disso, as *ViewModels* são, por definição, responsáveis pela comunicação entre o *View* e o *Model* dos seus respectivos componentes.

Figura 6.2: Estrutura da aplicação



Fonte: O Autor, 2019

Como já explicado anteriormente, a aplicação pode ser dividida em três principais partes: a seleção de tipos de estabelecimento, a experiência em realidade aumentada e o detalhe de um estabelecimento.

6.3.7 Seleção do tipo de estabelecimento

Essa parte da aplicação é responsável por se comunicar com o *back-end* e obter uma lista de tipos de lugares disponíveis a serem consultados. Consiste em uma interface simples, com poucas interações, porém é o ponto inicial da aplicação. Além disso, é responsável por obter a lista de lugares em determinado alcance para o tipo escolhido, a fim de repassá-los para a experiência AR. A interface pode ser conferida na figura 6.3.

De acordo com a figura 6.2, o componente *PlaceTypeViewModel* é o responsável por realizar a comunicação com o servidor. Com a resposta obtida, ele atualiza o seu componente de *Model*, que é uma lista de *PlaceType*. Quando essa lista termina de atualizar, ela avisa novamente o seu *ViewModel*, que preenche o *View* através de *databinding*, mostrando ao usuário a lista de tipos disponíveis. De forma similar, o *PlaceTypeViewModel* realiza comunicação com o servidor para obter a lista de lugares próximos, porém repassa essa informação ao *PlaceTypeCoordinator*, que direciona a aplicação para o próximo componente.

Figura 6.3: Tela inicial da aplicação



Fonte: O Autor, 2019

6.3.8 Experiência AR

Essa é a principal parte da aplicação. Primeiramente, recebe uma lista de objetos que contém a latitude e longitude de cada um, que são provenientes da primeira parte da aplicação. A lista de lugares é recebida através do *ARSceneCoordinator*, o qual é responsável por iniciar a realidade aumentada e repassar esses dados para o *ARSceneViewModel*. A partir desse componente, as informações de lugares são processadas com auxílio dos *frameworks* ARKit e SpriteKit, que transformam esses dados de localização em posições dentro de um mundo simulado. Com as posições calculadas, o componente *ARSceneView* adiciona âncoras (*ARAnchor*) no mundo para melhorar a precisão de rastreamento e, por fim, adiciona o objeto virtual (*SKNode*) na cena. Com os objetos já adicionados, a aplica-

ção, a partir do componente *ARSceneViewModel*, começa a obter dados de localização e movimentação utilizando os *frameworks* CoreLocation e CoreMotion. Para cada atualização de posição, a distância para cada um dos lugares é recalculada e o componente ARSceneView é avisado, o qual atualiza o valor mostrado na interface.

Para exemplificar um caso de uso, a aplicação foi utilizada no bairro Jardim Botânico, em Porto Alegre. Com base na coordenada geográfica do dispositivo (latitude: -30.051797056635827, longitude: -51.184657382996335), uma busca por universidades num raio de 1 quilômetro foi realizada na API do Google Places. Os resultados dessa chamada de rede estão demonstrados na tabela 6.2.

Tabela 6.2: Resultados da chamada de rede para coordenada geográfica

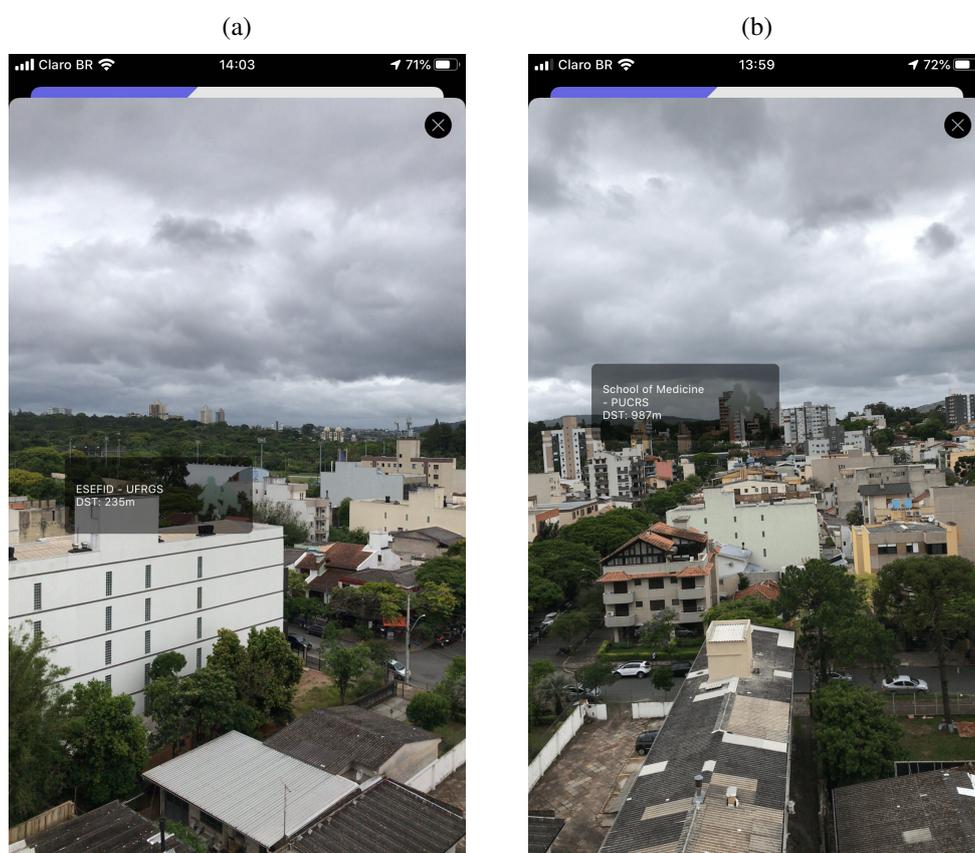
Latitude	Longitude	Nome
-30.0509476	-51.1825412	ESEFID - UFRGS
-30.0449028	-51.1894731	ITEC
-30.0547163	-51.1754905	School of Medicine - PUCRS
-30.0549105	-51.1830015	Instituto de Pós-Graduação do RS
-30.0501752	-51.1852768	Galpão de Artes - Porto Alegre
-30.0583980	-51.1855125	Colégio Brasileiro de Cirurgiões

Com os resultados obtidos, é possível adicionar cada um dos pontos no mundo simulado pela aplicação, usando a coordenada do dispositivo como ponto central e calculando a distância entre dois pontos geográficos para cada um dos resultados. Com isso, ao mover o dispositivo para a direção de uma das universidades, uma marcação flutuante estará em cima do local onde a universidade se encontra, mostrando o nome do lugar e a distância até o mesmo.

A figura 6.4 exemplifica dois dos resultados obtidos, os quais foram adicionados à realidade aumentada. Ambas as fotos foram tiradas do mesmo lugar, utilizando a aplicação e movendo a câmera do celular de um ponto a outro. A primeira representa a ESEFID (Escola de Educação Física, Fisioterapia e Dança da UFRGS), enquanto a segunda representa a Escola de Medicina da PUCRS.

Para comparar a experiência, a figura 6.5 possui um mapa com as duas universidades plotadas em conjunto com a localização do dispositivo. Nele, é possível identificar a distância e direção de cada uma das universidades para o ponto de referência. Com esses dados obtidos, é possível comparar a informação com o que foi gerado nas telas de realidade aumentada, a fim de verificar a precisão e acurácia das medições.

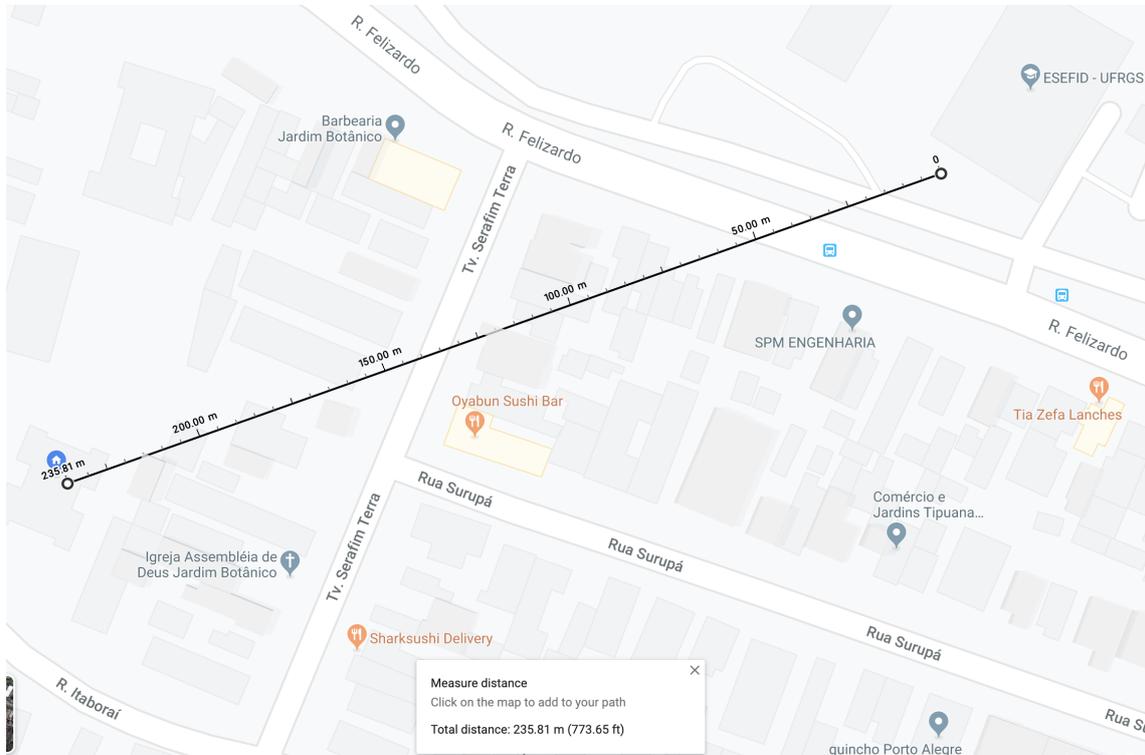
Figura 6.4: Experiência AR mostrando (a) a ESEFID e (b) a PUCRS



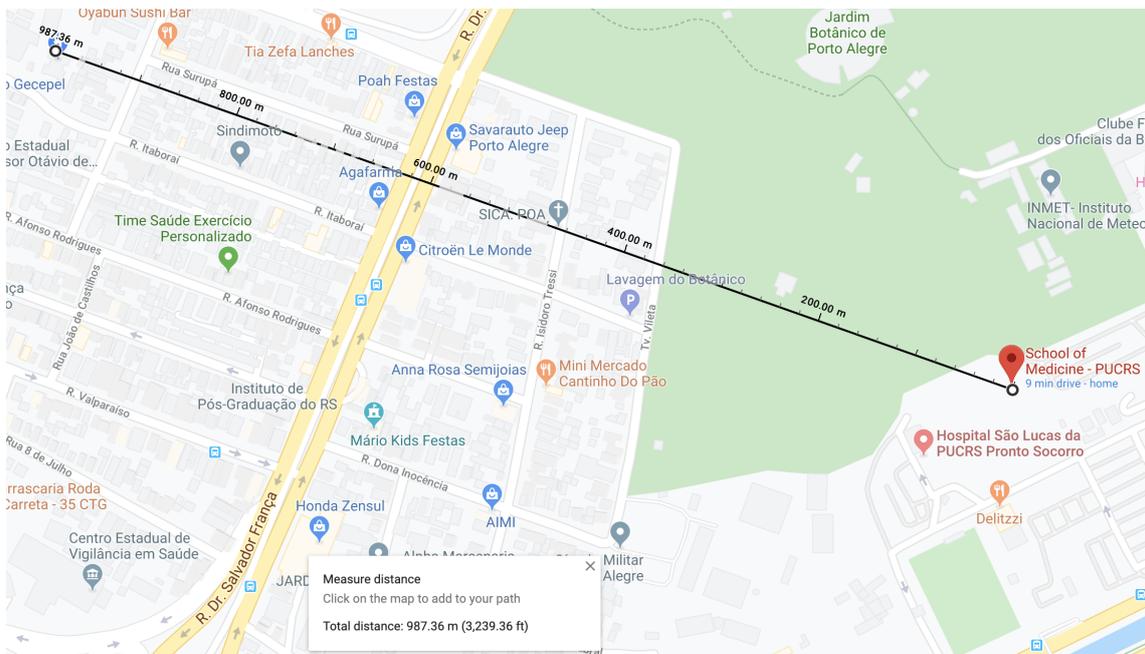
Fonte: O Autor, 2019

Figura 6.5: Medições de distância entre ponto de referência e (a) a ESEFID e (b) a PUCRS

(a)



(b)



Fonte: Google Maps, 2019

Como pode ser observado, a distância linear para o ponto de referência de ambas as universidades vai de encontro à distância medida pela aplicação. Também pode-se observar na primeira imagem da ESEFID que a mesma aparece ao fundo de sua marcação, então pode-se esperar que a PUCRS esteja na direção correta da sua também.

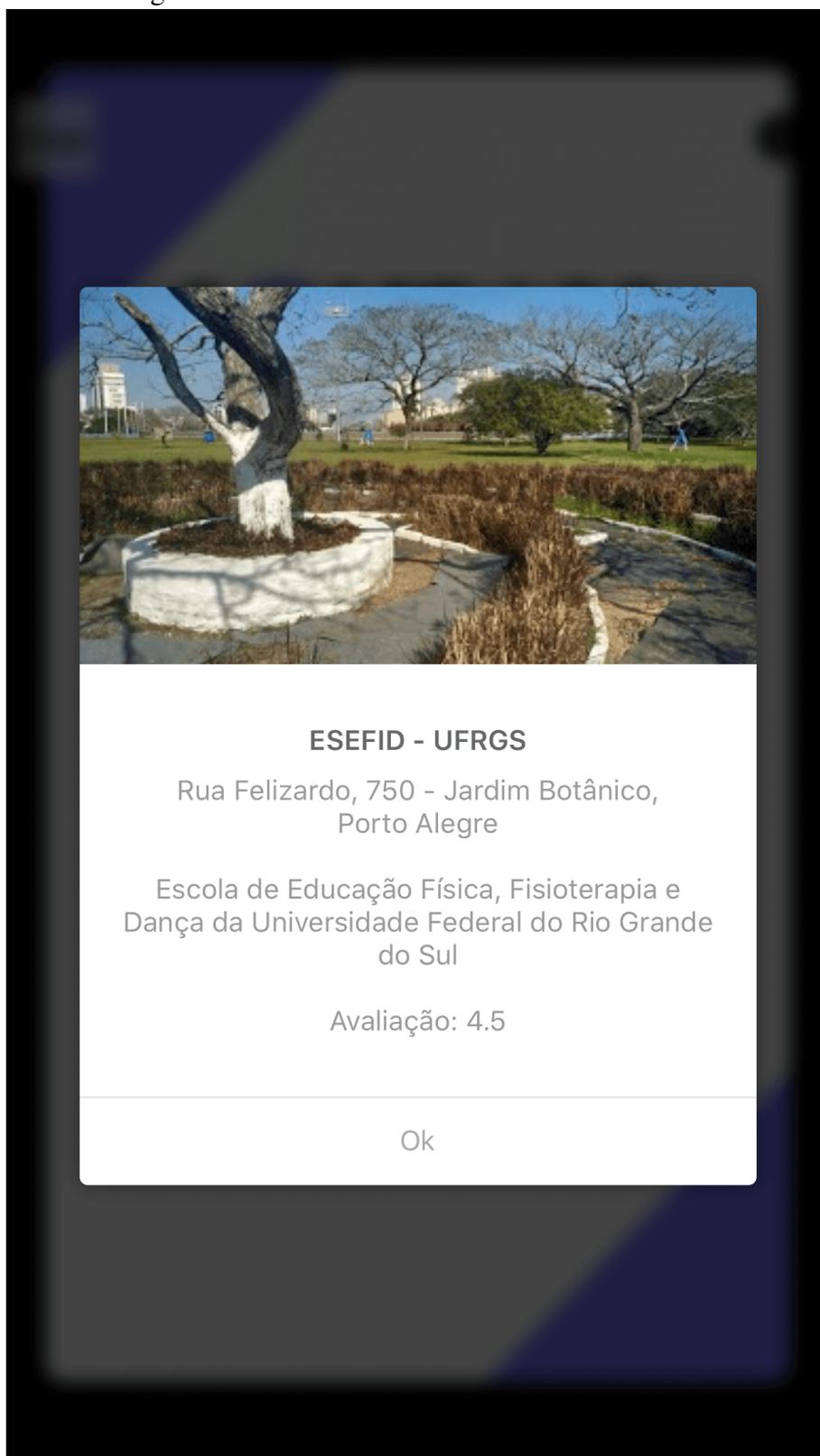
Além das duas universidades demonstradas, existem outras que estão em outras direções. Para encontrá-las na aplicação, basta mexer a câmera até que a sua marcação apareça, indicando a sua direção e distância. Inicialmente, a busca foi feita para universidades no raio de 1 quilômetro, mas esse parâmetro pode ser ajustado a fim de obter mais ou menos resultados.

6.3.9 Detalhe do estabelecimento

Enfim, para cada objeto virtual adicionado, existe um reconhecimento de toque associado. Quando um objeto é clicado, o *ARSceneView* intercepta esse toque e avisa o seu componente *ARSceneViewModel*, o qual obtém os dados necessários daquele local na sua lista de *PointOfInterest* e avisa o *ARSceneCoordinator*. Esse é responsável por direcionar a aplicação para a tela de detalhe do local desejado. Essa parte da aplicação é responsável por chamar a rota de detalhes de um estabelecimento selecionado, através do componente *PlaceDetailViewModel*, armazenar os dados no objeto *Place* e mostrar tais informações para o usuário no componente *PlaceDetailView*, como pode ser observado na figura 6.6, novamente utilizando a ESEFID como exemplo.

Nessa tela, é possível navegar entre as fotos do estabelecimento, bem como identificar a rua e a avaliação do mesmo. Quando disponível, a tela ainda mostra uma descrição do lugar.

Figura 6.6: Tela de detalhe de um estabelecimento



Fonte: O Autor, 2019

7 RESULTADOS E DISCUSSÕES

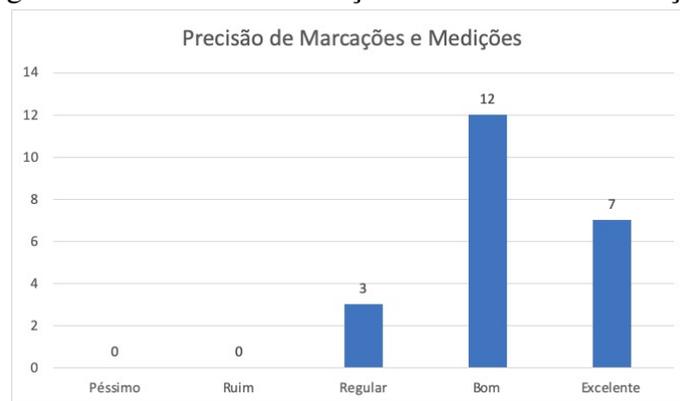
A solução proposta foi implementada na linguagem Swift na parte do aplicativo iOS, rodando em celulares com processador A9 ou superior. Para medir o impacto da solução proposta, realizaram-se testes com 22 usuários, os quais responderam um questionário ao fim do uso da aplicação.

O perfil dos usuários pode ser definido como adultos entre 23 e 30 anos, dos quais 20 são homens e 2 são mulheres. Todos trabalham em empresa de tecnologia e 80% trabalha como desenvolvedor na sua empresa. A respeito de conhecimento sobre a tecnologia AR, 10 usuários responderam que conheciam o conceito e apenas 1 usuário respondeu que já tinha trabalhado com ela. Para testar a aplicação, os usuários receberam uma versão do aplicativo, a qual foi instalada no seu celular.

Os testes duraram três dias, nos quais o aplicativo foi utilizado durante horário de almoço, saídas noturnas e até mesmo dentro de shoppings. A grande maioria utilizou o aplicativo em movimento, testando atualizações de distância e posicionamento das marcações. A acurácia das informações buscadas pela aplicação também foi examinada, a fim de determinar se a informação correspondia com o ambiente mostrado.

O gráfico na figura 7.1 mostra a precisão das medições, bem como a acurácia das marcações flutuantes, segundo os usuários.

Figura 7.1: Precisão de medições e acurácia de marcações



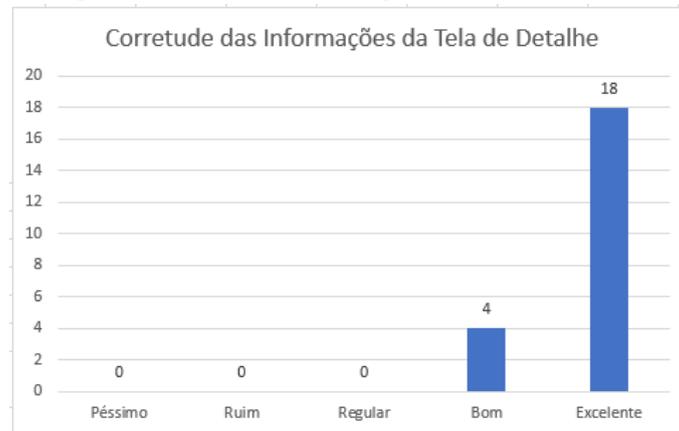
Fonte: O Autor, 2019

Como é possível observar, a maioria das medições foi satisfatória, porém alguns resultados tiveram índices inferiores ao esperado devido a imprecisões nas leituras de GPS do dispositivo, principalmente se o mesmo foi utilizado em local fechado.

Já na figura 7.2, mediu-se o nível de correção das informações disponibilizadas na tela de detalhe de um estabelecimento, as quais são obtidas a partir da combinação de

duas APIs externas pelo *back-end*.

Figura 7.2: Nível de correção da tela de detalhe

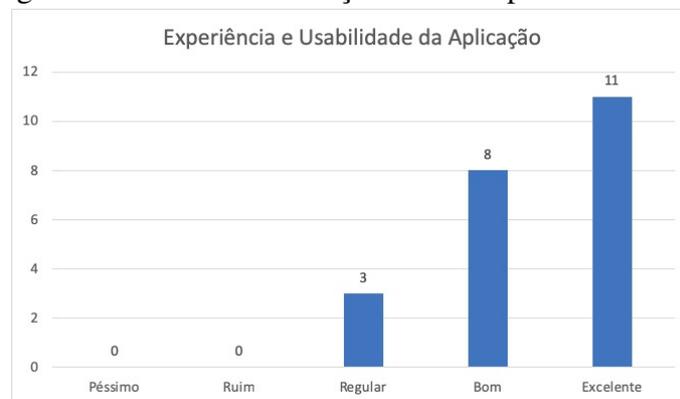


Fonte: O Autor, 2019

De acordo com as respostas obtidas, a API está funcionando de maneira satisfatória, deixando pouca margem para erros.

O nível de satisfação do usuário ao utilizar a aplicação está descrito na figura 7.3. Foram levados em consideração *bugs*, *crashes* ou travamentos, além da experiência do usuário de forma geral.

Figura 7.3: Nível de satisfação com a experiência de uso

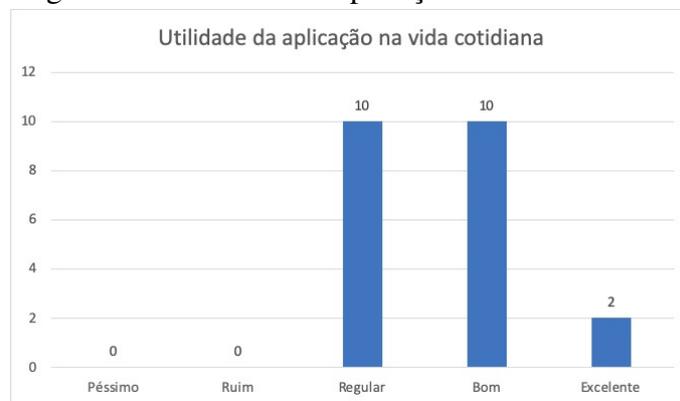


Fonte: O Autor, 2019

Com as respostas obtidas, é possível afirmar que a aplicação funciona e possui uma boa experiência de uso, porém existem alguns pontos de melhoria a serem trabalhados, como, por exemplo, mudanças na interface visual da experiência AR, baseadas em informações do lugar, onde a distância poderia ser um fator decisivo no tamanho da anotação, ou a avaliação do mesmo poderia afetar a cor da marcação. Outra mudança sugerida foi que a aplicação guardasse buscas recentes, a fim de possibilitar que o usuário utilize a aplicação mesmo quando o dispositivo não possui acesso à internet.

Por fim, na figura 7.4, procurou-se estimar o quão útil a aplicação foi ou pode vir a ser no cotidiano de cada usuário.

Figura 7.4: Utilidade da aplicação na vida cotidiana



Fonte: O Autor, 2019

Apesar de bem avaliada em todos os outros quesitos, a aplicação ainda não pode ser considerada extremamente útil para todos. Acredita-se que com adição de mais funcionalidades e mudanças de interface, como, por exemplo, integração com sistemas de navegação já existentes ou até mesmo a implementação de um navegador em realidade aumentada, tal problema possa ser parcialmente resolvido. Outra possível alteração seria na página de detalhe, onde usuários relataram que poderiam existir mais informações, como, por exemplo, *links* para redes sociais, ou até mesmo formas de contato. O questionário completo se encontra no Apêndice A.

8 CONCLUSÃO

Este trabalho se propôs a criar uma nova forma de interação com o ambiente ao redor do usuário. Através da utilização da realidade aumentada, criou-se uma nova maneira de interagir com mapas e obter informações de estabelecimentos próximos. Até o atual momento, baseado no conhecimento disponível, a aplicação é pioneira na proposta. A aplicação almeja impulsionar a utilização da tecnologia nas mais variadas áreas do conhecimento.

O trabalho ainda discutiu arquitetura de aplicações *mobile*, bem como boas práticas e técnicas de engenharia de *software* para manter as aplicações com parâmetros aceitáveis de qualidade. Além disso, explicou-se como a realidade aumentada surgiu e o que é necessário para implementá-la.

Os resultados mostraram que, apesar de válida e funcional, a aplicação ainda pode se beneficiar de otimizações, adições e correções. Existem limitações no rastreamento da posição do usuário, o que torna a aplicação, por vezes, um pouco imprecisa. Para trabalhos futuros, o aumento na precisão das marcações é um bom ponto de partida. Além disso, a adição de mais funcionalidades, bem como melhorias na experiência de uso, pode impulsionar o nível de utilidade da aplicação, ponto que mais demonstrou fragilidade nas avaliações dos usuários. Por fim, a adição de mais fontes de informação pode complementar a aplicação no quesito de quantidade de informações disponível, ponto que também foi mencionado durante a pesquisa com os usuários.

REFERÊNCIAS

- ALEM, L.; HUANG, W. **Recent Trends of Mobile Collaborative Augmented Reality Systems**. Springer New York, 2011. (SpringerLink : Bücher). ISBN 9781441998453. Available from Internet: <<https://books.google.com.br/books?id=HGxZytjBBnsC>>.
- ALLAN, A. **Basic Sensors in iOS: Programming the Accelerometer, Gyroscope, and More**. [S.l.]: O'Reilly Media, 2011.
- ALSAEDI, M. A. S. Development of 3d accelerometer testing system. **Dissertations and Theses. Paper 3371.**, 2016. Available from Internet: <https://pdxscholar.library.pdx.edu/open_access_etds/3371>.
- Azuma, R. et al. Recent advances in augmented reality. **IEEE Computer Graphics and Applications**, v. 21, n. 6, p. 34–47, Nov 2001.
- AZUMA, R. T. A survey of augmented reality. **Presence: Teleoperators Virtual Environments**, v. 6, p. 355 – 385, 1997. ISSN 1054-7460. Available from Internet: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.35.4614&rep=rep1&type=pdf>>.
- BOTELLA, C. et al. Mixing realities? an application of augmented reality for the treatment of cockroach phobia. **Cyberpsychology behavior : the impact of the Internet, multimedia and virtual reality on behavior and society**, v. 8, p. 162–71, 05 2005.
- CARMIGNIANI, J. et al. Augmented reality technologies, systems and applications. **Multimedia Tools and Applications**, v. 51, p. 341–377, 12 2010.
- CIPRESSO, P. et al. The past, present, and future of virtual and augmented reality research: A network and cluster analysis of the literature. **Frontiers in Psychology**, v. 9, p. 2086, 11 2018.
- Daniel Juarez Garcia. **Florence Travel Guide, Italy**. 2019. <<https://apps.apple.com/us/app/florence-travel-guide/id355465182>>. Acesso em 08/2019.
- DJUKNIC, G. M.; RICHTON, R. E. Geolocation and assisted gps. **IEEE Computer Society**, v. 34, p. 123–125, 2001. ISSN 0018-9162. Available from Internet: <http://www.cs.columbia.edu/~drexel/CandExam/Geolocation_assistedGPS.pdf>.
- Escobedo, L. et al. Using augmented reality to help children with autism stay focused. **IEEE Pervasive Computing**, v. 13, p. 38–46, 2014. ISSN 1536-1268.
- Express. **Routing**. 2019. <<http://expressjs.com/en/guide/routing.html>>. Acesso em 10/2019.
- FOWLER, M. **Patterns of Enterprise Application Architecture**. USA: Addison-Wesley Longman Publishing Co., Inc., 2002. ISBN 0321127420.
- FUCHS, H.; BISHOP, G. **Research Directions in Virtual Environments**. Chapel Hill, NC, USA, 1992.
- GAMMA, E. **Design patterns : elements of reusable object-oriented software**. [S.l.]: Reading, Mass. : Addison-Wesley, [1995] ©1995, 1995.

GIGANTE, M. A. 1 - virtual reality: Definitions, history and applications. In: EARNSHAW, R.; GIGANTE, M.; JONES, H. (Ed.). **Virtual Reality Systems**. Boston: Academic Press, 1993. p. 3 – 14. ISBN 978-0-12-227748-1. Available from Internet: <<http://www.sciencedirect.com/science/article/pii/B9780122277481500093>>.

JUAN, M.-C. et al. Using augmented reality to treat phobias. **Computer Graphics and Applications, IEEE**, v. 25, p. 31– 37, 12 2005.

KOURAKLIS, J. Mvvm as design pattern. In: _____. [S.l.: s.n.], 2016.

MEKNI, M.; LEMIEUX, A. Augmented reality : Applications , challenges and future trends. In: . [S.l.: s.n.], 2014.

MERRY, K.; BETTINGER, P. Smartphone gps accuracy study in an urban environment. **PLoS ONE**, v. 14, p. e0219890, 07 2019.

MILGRAM, P.; KISHINO, F. A taxonomy of mixed reality visual displays. **IEICE Trans. Information Systems**, vol. E77-D, no. 12, p. 1321–1329, 12 1994.

Niantic. **Pokémon GO**. 2019. <<https://www.pokemongo.com/en-us/>>. Acesso em 08/2019.

Node.js. **About Node.js**. 2019. <<https://nodejs.org/en/about/>>. Acesso em 10/2019.

npm, Inc. **About npm**. 2019. <<https://docs.npmjs.com/about-npm/>>. Acesso em 10/2019.

OLIVEIRA, M. P. de; SILVA, F. P. da. Use of virtual reality and augmented reality in learning objects: a case study for technical drawing teaching. **International Journal of Education and Research**, v. 7, p. 21–32, 2019. ISSN 2411-5681. Available from Internet: <<https://lume.ufrgs.br/handle/10183/194935>>.

Orlando/Orange County Convention & Visitors Bureau, INC. **Visit Orlando**. 2019. <<https://www.visitorlando.com/en/plan-your-trip/visit-orlando-destination-app>>. Acesso em 08/2019.

Sá, M. de; CHURCHILL, E. Mobile augmented reality: Exploring design and prototyping techniques. In: **Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services**. New York, NY, USA: ACM, 2012. (MobileHCI '12), p. 221–230. ISBN 978-1-4503-1105-2. Available from Internet: <<http://doi.acm.org/10.1145/2371574.2371608>>.

SHATTE, A.; HOLDSWORTH, J.; LEE, I. Mobile augmented reality based context-aware library management system. **Expert Systems with Applications**, v. 41, n. 5, p. 2174 – 2185, 2014. ISSN 0957-4174. Available from Internet: <<http://www.sciencedirect.com/science/article/pii/S0957417413007471>>.

SUTHERLAND, I. **The Ultimate Display.[1965]** In: **PACKER, R; JORDAN, K.(ed.). Multimedia: From Wagner to Virtual Reality**. [S.l.]: WW Norton & Company. 396p, 2001.

SWANSON, S.; TAYLOR, M. B. Greendroid: Exploring the next evolution in smartphone application processors. **IEEE Communications Magazine**, v. 4, p. 112 – 119, 2011. ISSN 1558-1896. Available from Internet: <<https://cseweb.ucsd.edu/~swanson/papers/IEEEComm2011GreenDroid.pdf>>.

TSENG, Y.-C. et al. Location awareness in ad hoc wireless mobile networks. **IEEE Computer Society**, v. 34, p. 46 – 52, 2001. ISSN 1558-0814. Available from Internet: <<http://mail.tku.edu.tw/jingo/wireless/paper/8-13.pdf>>.

ZEIMPEKIS, V.; GIAGLIS, G. M.; LEKAKOS, G. A taxonomy of indoor and outdoor positioning techniques for mobile location services. **ACM SIGecom Exchanges - Mobile commerce**, v. 3, p. 19–27, 2003. ISSN 1551-9031. Available from Internet: <http://www.sigecom.org/exchanges/volume_3/3.4-Zeimpekis.pdf>.

**APÊNDICE A — QUESTIONÁRIO DIRECIONADO AOS USUÁRIOS DA
APLICAÇÃO IOS**

Pesquisa pós utilização da aplicação Compass

PARA CADA UMA DAS PERGUNTAS ABAIXO MARCAR **APENAS UMA RESPOSTA**.

01. O quão corretas estavam posicionadas as marcações flutuantes?

- Péssimo
- Ruim
- Regular
- Bom
- Excelente

02. O quanto as informações da tela de detalhes correspondiam ao estabelecimento selecionado?

- Péssimo
- Ruim
- Regular
- Bom
- Excelente

03. Como foi a usabilidade da aplicação, levando em consideração bugs, crashes, travamentos e experiência em geral?

- Péssimo
- Ruim
- Regular
- Bom
- Excelente

04. O quão útil você consideraria a aplicação para uso cotidiano?

- Péssimo
- Ruim
- Regular
- Bom
- Excelente

APÊNDICE B — TRABALHO DE GRADUAÇÃO I

Compass: Aplicação para Visualização de Pontos de Interesse em Realidade Aumentada

Matheus Stefanello Luz¹, Claudio Fernando Resin Geyer¹

¹Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brasil

{msluz, geyer}@inf.ufrgs.br

Abstract. *The Compass application was created in order to offer an alternative to the way people look at maps nowadays, that is widely spread by other applications. Using modern concepts of augmented reality, movement detection and route prediction, this application aims to create an easy and interactive way to show to the user what is around them and possible points of interest for them.*

Resumo. *A aplicação Compass está sendo desenvolvida a fim de oferecer uma alternativa à visualização padrão de mapas e estabelecimentos que é adotada hoje em dia, difundida amplamente por outras aplicações. Utilizando-se de conceitos modernos de realidade aumentada, detecção de movimentos e predição de rotas, a aplicação almeja criar uma maneira fácil e interativa de mostrar ao usuário o que está em sua volta, bem como possíveis pontos de interesse para o mesmo.*

1. Introdução

Realidade Aumentada é uma tecnologia que oferece um novo paradigma para interagir com um computador. Essa é tratada como uma variação de *Virtual Reality (VR)*, que é definida como uma tecnologia de imersão onde, uma vez que o usuário está inserido no mundo virtual, não consegue mais ver o mundo real a sua volta. A Realidade Aumentada, por sua vez, almeja o contrário disso, adicionando e suplementando o mundo real ao invés de mudá-lo completamente. O seu principal objetivo é aprimorar a percepção e interação do usuário através da adição de modelos 3D em um ambiente real, a fim de misturar objetos reais com objetos virtuais [Azuma 1997].

Atualmente, os computadores estão cada vez menores e mais poderosos. Os *smartphones* já são vistos como a nova geração de computadores e uma grande parte da população mundial faz uso deles diariamente [Swanson and Taylor 2011]. Nessa rotina de uso, é muito comum a utilização de aplicativos de mapas para locomoção e aplicativos de busca de estabelecimentos para lazer. Com base nessa rotina e na crescente evolução do poder de processamento dos *smartphones*, a aplicação Compass busca mudar a forma com que o usuário realiza tais buscas e interage com os aplicativos, se valendo dos conceitos da realidade aumentada.

1.1. Motivação

O estudo e uso dessa tecnologia de realidade aumentada não é novo. Por outro lado, o número de aplicações comuns e populares se valendo dela ainda é baixo, e apenas recentemente recebeu um aumento considerável no seu uso, o que se deve muito ao rápido avanço das tecnologias de *smartphones* [Shatte et al. 2014]. Existe uma vasta gama de áreas onde a realidade aumentada poderia ser utilizada para melhorar a experiência que o público tem em situações cotidianas. Uma dessas situações é a de viagens para lugares novos e desconhecidos. Mapas comuns podem auxiliar a encontrar estabelecimentos que possam ser do interesse do usuário e, além disso, ainda existem aplicações para sugestões de lugares a serem visitados. Porém, mesmo com todo o avanço na tecnologia dos *smartphones* e de realidade aumentada, não se encontrou na literatura uma solução que busque inserir o usuário na própria cidade de uma maneira diferente, se valendo de informações valiosas como sua posição, suas preferências e pontos de interesse da própria cidade.

Uma das propostas do Compass é mesclar os dois mundos: o mundo real, em que o usuário está visitando e não possui pleno conhecimento de rotas, ruas e lugares, e o mundo virtual, que possui dados como fotos, revisões, avaliações, recomendações, horários e planejamento de rotas. Uma vez que a aplicação é facilmente executada em dispositivos móveis, não é requerido nenhum equipamento adicional, apenas o próprio aparelho pessoal.

1.2. Objetivos

A criação da aplicação tem como objetivo principal proporcionar uma alternativa para a maneira atual de localização em ambientes não conhecidos. Através de marcações flutuantes ao redor do usuário com os pontos relevantes da cidade, o Compass busca a imersão do usuário no contexto a sua volta, levando o mesmo a realizar a opção por estabelecimentos baseado em mais do que proximidade, considerando lugares frequentemente visitados, lugares preferidos e interesses do próprio usuário, fotos, vídeos e outros atributos que são relevantes na tomada de decisão. Com base em todas as informações disponíveis, tanto do usuário quanto dos pontos de interesse, a aplicação busca mostrar a cidade por outros olhos através da tela de um *smartphone*.

O texto está organizado da seguinte forma: na seção 2, é citado o contexto em que o trabalho está inserido, bem como trabalhos e estudos relacionados. Na seção 3, é abordada a metodologia utilizada pela aplicação na obtenção, processamento e visualização de dados. Na seção 4, é demonstrado o andamento do projeto até o presente momento. Por fim, na seção 5, é proposto um cronograma para a realização do TG2.

2. *Background* e visão geral

A tecnologia de realidade aumentada tem como suas raízes o campo de pesquisa de interfaces em computação [Mekni and Lemieux 2014], sendo considerada uma tecnolo-

gia proveniente dos estudos de realidade virtual. O termo realidade virtual é, segundo [Ong and Nee 2013], definido como um ambiente artificial, o qual é experimentado através de estímulos sensoriais, como, por exemplo, visões e sons. Estes são provenientes de um computador e as ações realizadas por um usuário afetam parcialmente o que acontece nesse ambiente. Já o termo realidade aumentada é usado para descrever sistemas que possuem as seguintes características: 1) combinam o real com o virtual num ambiente real; 2) possuem interação em tempo real; 3) registro em três dimensões, onde o alinhamento preciso de objetos reais e virtuais é imprescindível [Azuma et al. 2001].

2.1. Trabalhos Relacionados

O aumento no poder de processamento e redução do tamanho dos computadores, juntamente com o crescente número de pessoas utilizando dispositivos móveis, tornou possível uma nova classe de aplicações que exploram o contexto ao redor de um usuário [Land and Zimmerman 2014]. A realidade aumentada aplicada a dispositivos móveis possibilita que a tecnologia seja utilizada fora de uma área pré-determinada com equipamentos já montados [Alem and Huang 2011]. Idealmente, isso possibilita a adição de camadas de informação a qualquer ambiente, em qualquer lugar [de Sá and Churchill 2012]. Na seção 1, mencionou-se uma parte da vasta gama de possíveis aplicações utilizando a tecnologia de realidade aumentada. Pesquisadores das mais diversas áreas propuseram soluções que almejam mudar a maneira como enxergamos determinados assuntos, como, por exemplo, educação, medicina, turismo e até mesmo jogos eletrônicos.

As propostas na área de educação são inúmeras. Como descreve [Yilmaz 2018], a tecnologia de realidade aumentada aplicada na educação pode enriquecer a experiência de aprendizado, pois ela possibilita o aumento na interação de quem está aprendendo com o conteúdo a ser aprendido, além de tornar o processo de aprendizado mais atrativo, efetivo, significativo e motivante. O nível das aplicações pode variar desde o ensino fundamental até o ensino superior, como pode ser visto, por exemplo, na proposta de um projeto para a disciplina de Desenho Técnico para os cursos de engenharia [de Oliveira and da Silva 2019], onde um objeto a ser estudado é projetado virtualmente para que os alunos possam compreendê-lo de maneira mais assertiva.

Um jogo que se popularizou recentemente e utiliza recursos de realidade aumentada é o Pokémon GO [Niantic]. Combinando técnicas de sobreposição do mundo real com o mundo virtual, juntamente com informações de localização do usuário, o jogo oferece uma jogabilidade e experiência que são totalmente diferentes dos jogos que não fazem uso de tal tecnologia.

Na área médica, essas aplicações têm encontrado seu espaço no tratamento de problemas psicológicos como, por exemplo, na aplicação sugerida por [Juan et al. 2005], em que usuários com aracnofobia podem ser expostos a modelos 3D de aranhas sem a necessidade da interação real com o animal, proporcionando, assim, meios de superar o medo. Em outra aplicação sugerida por [Escobedo et al. 2014], a tecnologia é empregada a fim de ajudar no tratamento e acompanhamento de crianças com autismo que possuem dificuldades para manter o foco e a atenção em determinados assuntos.

A área de turismo é a que mais se assemelha com a proposta desse artigo. Existem algumas aplicações consolidadas que são responsáveis por mapear e informar sobre pontos turísticos ou históricos de uma determinada região. O aplicativo *Visit Orlando* [Orlando/Orange County Convention & Visitors Bureau, INC], por exemplo, mapeia os pontos de interesse dos parques de diversões mais famosos da cidade de Orlando utilizando a tecnologia de realidade aumentada, tornando mais fácil a navegação e visitação de seus clientes. Já o aplicativo *Florence Travel Guide* [Daniel Juarez Garcia] é responsável por mostrar ao usuário a rica história da cidade de Florença através de marcações flutuantes espalhadas por toda a cidade.

3. Metodologia

O sistema será dividido em duas aplicações principais, o *front-end* e o *back-end*. O *front-end* consiste em um aplicativo iOS, que será instalado em iPhones e iPads, e possui a responsabilidade de coletar dados sobre posição e movimentação do usuário, bem como receber dados externos, processá-los e criar uma interface para o usuário com base neles. O *back-end* consiste em um servidor Node que será responsável por receber chamadas de rede, buscar dados em fontes diversas, processá-los, modelá-los e retorná-los para a aplicação *front-end*.

3.1. Posição

A habilidade de estimar precisamente a posição do usuário é fator chave para o funcionamento da aplicação. Portanto, empregar uma tecnologia de posicionamento (também conhecida como *location-awareness*) [Tseng et al. 2001] é um pré-requisito. Embora existam inúmeras maneiras de estimar a posição, elas podem ser classificadas em duas categorias: auto-posicionamento e posicionamento remoto.

As tecnologias de auto-posicionamento funcionam através do recebimento de sinais de antenas emissoras geograficamente distribuídas, que podem ser tanto terrestres quanto satélites. Com esses sinais recebidos, o dispositivo é capaz de calcular a sua própria posição, possibilitando que aplicações possam utilizar tal informação calculada pelo mesmo para a tomada de decisões onde o quesito posição é relevante. Já a tecnologia de posicionamento remoto funciona a partir de medições de sinais que saem do emissor, chegam no dispositivo e voltam para o emissor, ou seja, os emissores sabem onde está o dispositivo [Zeimpekis et al. 2003].

O modelo mais apropriado para a aplicação Compass é o modelo de auto-posicionamento, pois é necessário que se saiba a localização do próprio dispositivo. Existem inúmeras técnicas para o receptor determinar a sua posição, sendo a mais comum entre elas o GPS (Global Positioning System).

O GPS é um sistema mundial de navegação baseado em satélites, onde esses transmitem a informação de navegação e o receptor as utiliza para calcular a sua posição, com uma precisão de menos de 10 metros. Existem algumas limitações para o uso do GPS,

pois, para o mesmo operar de maneira satisfatória, é necessário visão não-obstruída do céu e sinal de pelo menos três satélites [Djuknic and Richton 2001]. Como a aplicação Compass visa o uso externo e não interno, a tecnologia de GPS é adequada e será utilizada pela mesma.

3.2. Movimentação

Além de saber onde está posicionado o aparelho, é vital para a aplicação poder determinar a direção, sentido e intensidade do movimento de um usuário. Para isso, será utilizado um dos sensores do aparelho, o acelerômetro, pois ele possui as informações necessárias para determinar a orientação do dispositivo.

3.2.1. Acelerômetro

Um acelerômetro é um dispositivo que mede forças de aceleração. Essas forças podem ser constantes, como, por exemplo, a força da gravidade, ou podem ser dinâmicas, causadas por algum movimento ou vibração externo ao dispositivo [Alsaedi 2016]. Existem múltiplos tipos de acelerômetros desenvolvidos, porém os aparelhos que utilizarão a aplicação trabalham com o modelo de três eixos.

O acelerômetro de três eixos proporciona medições simultâneas em três direções ortogonais, que, no caso, representam os eixos x , y e z no plano cartesiano. Uma medição estática da gravidade incidente sobre o acelerômetro pode ser analisada e transformada em um ângulo de inclinação do dispositivo, enquanto medições dinâmicas podem ser transformadas em direção e sentido do movimento do mesmo [Allan 2011].

3.3. Aplicativo iOS

O aplicativo será responsável por coletar os dados de localização e movimentação dos sensores, utilizando-se de *APIs (Application Programming Interfaces)* nativas do sistema operacional. Além disso, o aplicativo também terá a responsabilidade de calcular e exibir anotações na tela do dispositivo baseado nesses dados coletados previamente, juntamente com informações externas que serão obtidas por chamadas de rede. Por fim, dois fatores deverão ser levados em consideração pela aplicação, que são a privacidade do usuário e o consumo de bateria pelo próprio aplicativo.

3.3.1. API de Localização

Para obter informações sobre a posição do dispositivo, a *API* nativa a ser utilizada será a *CoreLocation* [Apple a]. Ela é responsável por prover a posição geográfica do dispositivo utilizando alguns sensores do mesmo. Para utilizá-la, é necessário que o usuário permita a captação de tais dados. Existem diferentes níveis de precisão para a *API*, que serão mostrados juntamente com uma breve descrição na Tabela 1.

Tabela 1. Níveis de precisão na medição da posição

Classificação	Descrição
kCLLocationAccuracyBestForNavigation	O maior nível de precisão, porém utiliza outros sensores para auxiliar aplicativos de navegação
kCLLocationAccuracyBest	O maior nível de precisão sem auxílio de outros sensores
kCLLocationAccuracyNearestTenMeters	Precisão de até 10 metros
kCLLocationAccuracyHundredMeters	Precisão de até 100 metros
kCLLocationAccuracyKilometer	Precisão de até 1 quilômetro
kCLLocationAccuracyThreeKilometers	Precisão de até 3 quilômetros

É preciso levar em consideração o impacto no consumo de energia de cada um desses parâmetros, pois, quanto mais precisa for a medição, maior será o gasto. Como a aplicação está baseada em medições precisas, o parâmetro escolhido será o de *kCLLocationAccuracyBestForNavigation*, pois o mesmo utiliza sensores como o giroscópio, acelerômetro e magnetômetro para reduzir erros e aumentar a precisão da medição.

3.3.2. API de Movimentação

Para obter informações sobre movimentos sendo aplicados ao dispositivo, a *API* nativa a ser utilizada será a *CoreMotion* [Apple b]. Ela é responsável por prover acesso à informação de rotação, aceleração e orientação do dispositivo.

3.3.3. Privacidade

A localização atual do usuário é um dado bastante sensível e pessoal, portanto não deverá ser armazenado por nenhuma aplicação que não esteja no próprio dispositivo. A comunicação com o servidor precisará desses dados para fazer a chamada baseada em localização, mas o mesmo apenas executará buscas com essa informação, sem se preocupar com o armazenamento dela.

3.3.4. Consumo de bateria

Um fator importantíssimo em qualquer aplicação é a otimização do consumo de bateria. A aplicação possui basicamente três estados: *foreground*, *background* e não inicializada. Uma estimativa de consumo pode ser feita para cada um dos estados:

- **Foreground.** Quando a aplicação está em *foreground*, ela está inicializada e é a aplicação sendo utilizada no momento. Nesse caso, todas as medições e cálculos estão sendo executados, então a estimativa de impacto na bateria pode ser considerada moderada/alta.
- **Background.** Quando a aplicação está em *background*, ela está inicializada mas outra aplicação está sendo utilizada no momento. Como a aplicação trabalha com a visualização de dados em tempo real, não há necessidade de processamento

enquanto ela não está sendo utilizada, então então a estimativa de impacto na bateria pode ser considerada mínima.

- **Não inicializada.** Quando a aplicação não está inicializada, o impacto na bateria é nulo.

3.4. *Back-end*

O *back-end* será um servidor armazenado na nuvem, utilizando os *frameworks Node.js* e *Express*. O *Node.js* foi criado a fim de prover uma ferramenta para construção de aplicações de rede rápidas e escaláveis utilizando código *JavaScript* de maneira assíncrona. Ele é capaz de gerenciar um número elevado de conexões simultâneas com alta taxa de transferência, o que é fundamental para alta escalabilidade [Node.js]. Além disso, esse *framework* possui um gerenciador de pacotes próprio, o *NPM (Node Package Manager)*, o qual possui inúmeros componentes publicamente disponíveis para uso [npm, Inc]. Um desses pacotes é o *Express*. Ele é responsável por abstrair grande parte da lógica de roteamento necessária para criar um servidor, o que torna a aplicação menos complexa [Express].

O servidor irá receber, processar e modelar dados para a aplicação. A comunicação com o dispositivo será feita através da construção de uma *API*, que permitirá que o aplicativo faça buscas utilizando as informações obtidas no próprio dispositivo, sem a necessidade de comunicar diretamente com *APIs* externas. Ele também será responsável por prover diferentes rotas para obtenção de informações. A principal funcionalidade do servidor será receber um pedido de pontos de interesse para uma determinada localização, buscar informações sobre essas localizações em diferentes *APIs* públicas, receber essas informações e modelá-las para que o aplicativo possa consumi-las sem a necessidade de um processamento posterior.

Para atingir tal objetivo, é imprescindível que o servidor seja feito de maneira modular. Cada *API* pública a ser adicionada como fonte de informação precisará de um adaptador para que os dados obtidos sejam compatíveis com os outros já obtidos previamente. Diferentes *APIs* podem ser agregadas ou removidas do processo de obtenção de informações. Outra funcionalidade do servidor será prover para o aplicativo uma lista de tipos de estabelecimentos possíveis para determinada localização, a fim de que o aplicativo seja capaz de realizar filtros com os mesmos.

4. Andamento

Durante a realização deste trabalho, foram realizados alguns protótipos a fim de validar as soluções propostas isoladamente e, após isso, validar a integração entre as mesmas. Primeiramente, criou-se uma aplicação iOS que se comunica com uma *API* pública externa, a qual retorna pontos de interesse não modelados e não processados para uma determinada coordenada geográfica. A *API* utilizada foi a do *Google Places*, e o aplicativo utiliza a resposta dessa *API* para preencher um mapa 2D com marcações nos lugares desejados,

o que valida a hipótese de busca de estabelecimentos baseado em geolocalização, como pode ser visto na Figura 1.

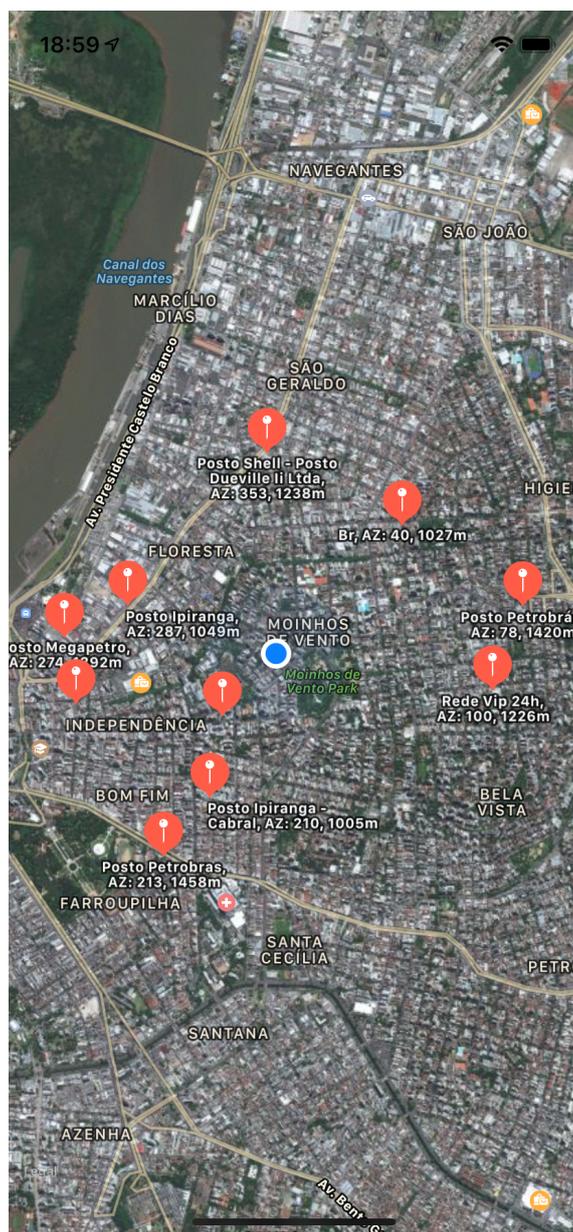


Figura 1. Protótipo de aplicativo para marcação 2D de pontos de interesse filtrados por coordenada geográfica (latitude: -30.0250869, longitude: -51.2034952) e tipo de estabelecimento (postos de gasolina)

Após a prototipação do aplicativo, criou-se um servidor *Node* que será responsável por fazer exatamente a mesma busca, e alterou-se o aplicativo para que ele se comunicasse com essa nova *API*. Com isso, a comunicação entre servidor e aplicativo foi estabelecida e validada.

Por fim, depois da criação da aplicação *back-end*, um estudo sobre a implementação de interfaces em realidade aumentada tornou-se necessário. Juntamente com o estudo, algumas provas de conceito foram realizadas utilizando código nativo da plataforma iOS e alguns *frameworks* também nativos, como, por exemplo, o *ARKit* da *Apple*. Durante a prototipação utilizando os *frameworks* de realidade aumentada, alguns objetos 3D foram plotados em um mesmo plano, a fim de que fosse validada a proposta de múltiplos objetos ao redor de um usuário, utilizando sua posição atual. Os objetos foram distribuídos utilizando essa posição como referencial, e não coordenadas geográficas reais. O detalhamento da implementação, bem como a modelagem da aplicação utilizando coordenadas geográficas reais, serão feitos no TG2.

5. Cronograma

Tabela 2. Cronograma de atividades para o TG2

	Jul	Ago	Set	Out	Nov	Dez
Modelagem da implementação	x	x	x			
Implementação do aplicativo iOS		x	x	x		
Implementação do back-end			x	x		
Validação do funcionamento da aplicação					x	
Elaboração da monografia		x	x	x	x	
Apresentação						x

A **modelagem da implementação** contempla as definições do projeto, como, por exemplo, decisões sobre linguagens de programação, plataformas que serão utilizadas e arquitetura do sistema. As etapas de **implementação do aplicativo iOS** e de **implementação do back-end** compõe a fase de desenvolvimento do software proposto como um todo, bem como sua integração, validação e homologação. A **validação do funcionamento da aplicação** será feita de forma a se utilizar o *software* com diferentes usuários e verificar a eficácia da ferramenta desenvolvida, no que diz respeito à satisfação do usuário em relação à solução proposta. Juntamente com o desenvolvimento da aplicação, a **elaboração da monografia** será feita ao longo do semestre. Por fim, a **apresentação** será na data definida pela universidade.

Referências

Alem, L. and Huang, W. (2011). *Recent Trends of Mobile Collaborative Augmented Reality Systems*. SpringerLink : Bücher. Springer New York.

- Allan, A. (2011). *Basic Sensors in iOS: Programming the Accelerometer, Gyroscope, and More*. O'Reilly Media.
- Alsaedi, M. A. S. (2016). Development of 3d accelerometer testing system. *Dissertations and Theses. Paper 3371*.
- Apple. CoreLocation. <https://developer.apple.com/documentation/corelocation>. Acesso em 08/2019.
- Apple. CoreMotion. <https://developer.apple.com/documentation/coremotion>. Acesso em 08/2019.
- Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., and MacIntyre, B. (2001). Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, 21(6):34–47.
- Azuma, R. T. (1997). A survey of augmented reality. *Presence: Teleoperators Virtual Environments*, 6:355 – 385.
- Daniel Juarez Garcia. Florence Travel Guide, Italy. <https://apps.apple.com/us/app/florence-travel-guide/id355465182>. Acesso em 08/2019.
- de Oliveira, M. P. and da Silva, F. P. (2019). Use of virtual reality and augmented reality in learning objects: a case study for technical drawing teaching. *International Journal of Education and Research*, 7:21–32.
- de Sá, M. and Churchill, E. (2012). Mobile augmented reality: Exploring design and prototyping techniques. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services, MobileHCI '12*, pages 221–230, New York, NY, USA. ACM.
- Djuknic, G. M. and Richton, R. E. (2001). Geolocation and assisted gps. *IEEE Computer Society*, 34:123–125.
- Escobedo, L., Tentori, M., Quintana, E., Favela, J., and Garcia-Rosas, D. (2014). Using augmented reality to help children with autism stay focused. *IEEE Pervasive Computing*, 13:38–46.
- Express. Routing. <http://expressjs.com/en/guide/routing.html>. Acesso em 10/2019.
- Juan, M.-C., Alcañiz Raya, M., Monserrat, C., Botella, C., Baños, R., and Guerrero, B. (2005). Using augmented reality to treat phobias. *Computer Graphics and Applications, IEEE*, 25:31– 37.
- Land, S. and Zimmerman, H. (2014). Synthesizing perspectives on augmented reality and mobile learning. *TechTrends*, 58.
- Mekni, M. and Lemieux, A. (2014). Augmented reality : Applications , challenges and future trends.
- Niantic. Pokémon GO. <https://www.pokemongo.com/en-us/>. Acesso em 08/2019.
- Node.js. About Node.js. <https://nodejs.org/en/about/>. Acesso em 10/2019.
- npm, Inc. About npm. <https://docs.npmjs.com/about-npm/>. Acesso em 10/2019.

- Ong, S. and Nee, A. (2013). *Virtual and Augmented Reality Applications in Manufacturing*. Springer London.
- Orlando/Orange County Convention & Visitors Bureau, INC. Visit Orlando. <https://www.visitorlando.com/en/plan-your-trip/visit-orlando-destination-app>. Acesso em 08/2019.
- Shatte, A., Holdsworth, J., and Lee, I. (2014). Mobile augmented reality based context-aware library management system. *Expert Systems with Applications*, 41(5):2174 – 2185.
- Swanson, S. and Taylor, M. B. (2011). Greendroid: Exploring the next evolution in smartphone application processors. *IEEE Communications Magazine*, 4:112 – 119.
- Tseng, Y.-C., Wu, S.-L., Liao, W.-H., and Chao, C.-M. (2001). Location awareness in ad hoc wireless mobile networks. *IEEE Computer Society*, 34:46 – 52.
- Yilmaz, R. M. (2018). Augmented reality trends in education between 2016 and 2017 years. *State of the art virtual reality and augmented reality knowhow*, 81:97.
- Zeimpekis, V., Giaglis, G. M., and Lekakos, G. (2003). A taxonomy of indoor and outdoor positioning techniques for mobile location services. *ACM SIGecom Exchanges - Mobile commerce*, 3:19–27.