



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CAMPUS QUIXADÁ**  
**BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**FRANCISCO FERNANDES JÚNIOR**

**DESENVOLVIMENTO E AVALIAÇÃO DE UM APLICATIVO CAPAZ DE  
GERENCIAR INTERATIVAMENTE LOCAIS DE INTERESSE**

**QUIXADÁ**

**2018**

FRANCISCO FERNANDES JÚNIOR

DESENVOLVIMENTO E AVALIAÇÃO DE UM APLICATIVO CAPAZ DE GERENCIAR  
INTERATIVAMENTE LOCAIS DE INTERESSE

Trabalho de Conclusão de Curso apresentado ao  
Curso de Graduação em Sistemas de Informação  
do Campus Quixadá da Universidade Federal  
do Ceará, como requisito parcial à obtenção do  
grau de bacharel em Sistemas de Informação.  
Área de Concentração: Computação.

Orientador: Prof. Dr. Marcio Espíndola  
Freire Maia

QUIXADÁ

2018

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca Universitária  
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

- F399d Fernandes Júnior, Francisco.  
Desenvolvimento e avaliação de um aplicativo capaz de gerenciar interativamente locais de interesse /  
Francisco Fernandes Júnior. – 2018.  
57 f. : il. color.
- Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá,  
Curso de Sistemas de Informação, Quixadá, 2018.  
Orientação: Prof. Dr. Marcio Espíndola Freire Maia.
1. Android (Programa de computador). 2. Ponto de Interesse. 3. Desenvolvimento. 4. Usabilidade. I.  
Título.

CDD 005

---

FRANCISCO FERNANDES JÚNIOR

DESENVOLVIMENTO E AVALIAÇÃO DE UM APLICATIVO CAPAZ DE GERENCIAR  
INTERATIVAMENTE LOCAIS DE INTERESSE

Trabalho de Conclusão de Curso apresentado ao  
Curso de Graduação em Sistemas de Informação  
do Campus Quixadá da Universidade Federal  
do Ceará, como requisito parcial à obtenção do  
grau de bacharel em Sistemas de Informação.  
Área de Concentração: Computação.

Aprovado em: \_\_\_\_/\_\_\_\_/\_\_\_\_

BANCA EXAMINADORA

---

Prof. Dr. Marcio Espíndola Freire Maia (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Jefferson de Carvalho Silva  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Paulo Armando Cavalcante Aguiar  
Universidade Federal do Ceará (UFC)

À toda minha família, por sua capacidade de acreditar sempre em mim.

## **AGRADECIMENTOS**

Aos meus amigos e familiares, que me apoiaram durante todo o curso.

Agradeço a todos os professores por me proporcionar o conhecimento não apenas racional, mas a manifestação do caráter e afetividade da educação no processo de formação profissional, por tanto que se dedicaram a mim, não somente por terem me ensinado, mas por terem me feito aprender.

“Não importa o quão grande seja a dor, eu continuarei seguindo em frente.”

(Naruto Uzumaki)

## RESUMO

Este trabalho apresenta o desenvolvimento e avaliação de uma aplicação para dispositivos móveis, onde o mesmo une diversas funções relacionadas a mapa dos aplicativos já presentes no mercado, com algumas novas ideias. O aplicativo é capaz de gerar rotas entre dois pontos, manipular pontos de interesse no mapa, criar tipos personalizados de próprio gosto, comunicar-se entre usuários e pontos de interesse e também prover de um tutorial auxiliando cada função disponível. Este tutorial faz parte de algumas das ferramentas que ajudam na avaliação da usabilidade do aplicativo, que é o segundo foco deste trabalho. Todas essas funções tem a finalidade de ajudar os usuários a fazer suas divulgações, como também se orientar em diversos momentos e situações. A aplicação realiza pesquisas em mais de um servidor, primeiramente utiliza-se os servidores da Google para buscar pontos de interesse e seus respectivos tipos, logo após é feito uma mesma busca no servidor próprio, também desenvolvido durante este trabalho.

**Palavras-chave:** Android (Programa de computador). Pontos de Interesse. Desenvolvimento. Usabilidade.

## **ABSTRACT**

This work presents the development and evaluation of an application for mobile devices, where it combines several functions related to the map of the applications already on the market, with some new ideas. The application is able to generate routes between two points, manipulate points of interest on the map, create custom types of own taste, communicate between users and points of interest and also provide a tutorial assisting each function available. This tutorial is part of some of the tools that help in evaluating the usability of the application, which is the second focus of this work. All these functions are intended to help users make their disclosures, as well as to guide themselves in different moments and situations. The application performs searches on more than one server, first use Google's servers to search for points of interest and their types, after the same search is done on the server itself, also developed during this work.

**Keywords:** Android (Program of computer). Points of interest. Development. Usability.

## LISTA DE FIGURAS

Figura 1 – Arquitetura da aplicação . . . . .	30
Figura 2 – Layout Login . . . . .	31
Figura 3 – Layout Inicial . . . . .	32
Figura 4 – Tipos . . . . .	32
Figura 5 – Layout insert_location . . . . .	33
Figura 6 – Layout InfoWindow . . . . .	34
Figura 7 – Layout Detalhes . . . . .	35
Figura 8 – Layout Tutorial . . . . .	35
Figura 9 – Layout Menu Lateral . . . . .	36
Figura 10 – Cabeçalho PHP . . . . .	39
Figura 11 – Consulta SQL . . . . .	39
Figura 12 – Retornando em JSON . . . . .	40
Figura 13 – Exemplo JSON . . . . .	40
Figura 14 – Comunicação com WEB SERVICE . . . . .	41
Figura 15 – Exemplo Volley . . . . .	42
Figura 16 – Questão 2 - Identificação . . . . .	45
Figura 17 – Questão 3 - Identificação . . . . .	46
Figura 18 – Questão 1 . . . . .	46
Figura 19 – Questão 2 . . . . .	47
Figura 20 – Questão 3 . . . . .	48
Figura 21 – Questão 4 . . . . .	48
Figura 22 – Questão 5 . . . . .	49
Figura 23 – Questão 6 . . . . .	49
Figura 24 – Questão 7 . . . . .	50

## **LISTA DE TABELAS**

Tabela 1 – Comparação entre os trabalhos relacionados e o proposto . . . . .	27
--	----

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>16</b>
<b>2.1</b>	<b>Android</b>	<b>16</b>
<b>2.1.1</b>	<i>Activitys</i>	<b>17</b>
<b>2.2</b>	<b>Usabilidade</b>	<b>17</b>
<b>2.2.1</b>	<i>Métricas</i>	<b>19</b>
<b>2.2.2</b>	<i>Técnicas de avaliação</i>	<b>19</b>
<b>2.2.2.1</b>	<i>Heurísticas</i>	<b>19</b>
<b>2.2.2.2</b>	<i>Testes com utilizadores</i>	<b>20</b>
<b>2.3</b>	<b>API</b>	<b>21</b>
<b>2.3.1</b>	<i>Arquitetura Cliente-Servidor</i>	<b>22</b>
<b>2.3.2</b>	<i>API dos Códigos-Fonte</i>	<b>22</b>
<b>2.3.3</b>	<i>API de Legado</i>	<b>23</b>
<b>3</b>	<b>TRABALHOS RELACIONADOS</b>	<b>24</b>
<b>3.1</b>	<b>UbibusRoute - Um Sistema de Identificação e Sugestão de Rotas de Ônibus Baseado em Informações de Redes Sociais</b>	<b>24</b>
<b>3.2</b>	<b>Trocacomigo: Aplicativo de Consumo Colaborativo</b>	<b>25</b>
<b>3.3</b>	<b>RouteMe - Uma Aplicação para Dispositivos Móveis Android para Resolução de Problemas de Rotas de Veículos (PRV)</b>	<b>26</b>
<b>4</b>	<b>METODOLOGIA</b>	<b>28</b>
<b>4.1</b>	<b>Planejamento</b>	<b>28</b>
<b>4.1.1</b>	<i>Requisitos da Aplicação</i>	<b>29</b>
<b>4.1.2</b>	<i>Ambiente e Arquitetura</i>	<b>29</b>
<b>4.2</b>	<b>Construção</b>	<b>31</b>
<b>4.2.1</b>	<i>Desenvolvimento dos Layouts e Activitys</i>	<b>31</b>
<b>4.2.2</b>	<i>Modelos, DAOs, Adapters e Utils</i>	<b>36</b>
<b>4.2.3</b>	<b>WEB SERVICE</b>	<b>38</b>
<b>4.2.4</b>	<i>Comunicação com WEB SERVICE</i>	<b>41</b>
<b>4.3</b>	<b>Testes</b>	<b>42</b>
<b>4.4</b>	<b>Implantação</b>	<b>42</b>

<b>5</b>	<b>AVALIAÇÃO</b> . . . . .	<b>43</b>
<b>5.1</b>	<b>Elaboração dos questionários</b> . . . . .	<b>43</b>
<b>5.2</b>	<b>Interação dos usuários com a aplicação</b> . . . . .	<b>43</b>
<b>5.3</b>	<b>Aplicação do questionário</b> . . . . .	<b>44</b>
<b>6</b>	<b>RESULTADOS</b> . . . . .	<b>45</b>
<b>7</b>	<b>CONCLUSÕES</b> . . . . .	<b>51</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>52</b>
	<b>APÊNDICE A – Requisitos da aplicação</b> . . . . .	<b>54</b>
	<b>APÊNDICE B – Diagrama de Casos de Uso</b> . . . . .	<b>55</b>
	<b>APÊNDICE C – Formulário de Avaliação do Aplicativo</b> . . . . .	<b>56</b>

## 1 INTRODUÇÃO

O rápido crescimento da Internet Móvel tem trazido implicações significativas na forma como as organizações podem fazer seus negócios (TURBAN *et al.*, 2010). Assim, diretores, donos, responsáveis de empresas de softwares e, às vezes, os próprios usuários dessa tecnologia buscam vantagens competitivas por meio de aplicações inovadoras que fazem uso da Internet Móvel. Nesse sentido, o uso de GPS para a rápida localização e navegação, como também para o monitoramento de dispositivos de segurança, é um exemplo de tecnologia bastante utilizada atualmente.

O mercado tecnológico atual está em constante crescimento, logo surgem cada vez mais ideias que podem ser valorizadas e consideradas por uma grande massa da sociedade. As inovações tecnológicas ganharam espaço e revolucionaram o estilo de vida. Sørensen (2011) fala que as tecnologias móveis tocam uma proporção cada vez maior da existência humana, pois em vários lugares onde possamos estar, seja em casa ao deitar, em uma viagem nas férias, no ambiente acadêmico ou no local de trabalho, tem-se conectividade, temos acesso ao mundo virtual, temos nossas aplicações a disposição do usuário. Hoje em dia, os dispositivos e mecanismos eletroeletrônicos são indispensáveis para a maioria das pessoas, algumas por dependência estudantil ou trabalhista, outras por diversão, outras por conforto.

A cada dia que passa, os smartphones, tablets e outros dispositivos estão vindo para aumentar o desenvolvimento de aplicações, devido a considerável evolução tecnológica. Trazendo sempre novas aplicações que conquistam os usuários. Segundo Kalakota e Robinson (2002), as aplicações móveis mudaram a maneira de se viver, de se divertir e de se fazer negócios.

Um exemplo dessas aplicações citadas e que foi abordada neste projeto é o Google Maps, serviço de pesquisa, visualização de mapas e imagens de satélite da Terra. Tal serviço possibilita traçar rotas e calcular distâncias, de forma gratuita e disponível na web, fornecido e desenvolvido pela empresa estadunidense Google. Disponibilizando APIs (Interface de Programação de Aplicações) para desenvolvedores criar suas próprias aplicações, uma grande importância dos mapas é a localização. Por meio deles, podemos encontrar qualquer ponto da superfície terrestre e deslocar-nos até ele. Outra importante função dos mapas é fornecer informações específicas sobre uma determinada localidade em uma perspectiva espacial.

Segundo Nomiso (2008) problemas de complexidade podem gerar um aplicativo de difícil interação ou baixa confiabilidade, gerando conflitos entre os usuários, possivelmente resultando na não utilização da aplicação. Dessa forma, o projeto aqui desenvolvido abordou as

dificuldades encontradas pelos usuários de mapas eletrônicos.

Pessoas têm a dificuldade de encontrar locais, serviços de interesse e oportunidades próximos a elas. Hoje em dia, algumas das vezes, tais pesquisas são feitas de pessoa para pessoa, por meio de conversas, recados e avisos, ou seja, algo bem humano, bem físico, limitando bastante as possíveis respostas. Por exemplo: Determinado indivíduo procura uma casa para alugar ou comprar. Esse indivíduo vai procurar pessoalmente na rua, se vê alguma placa de “ALUGA-SE” ou “VENDE-SE”. Uma outra maneira é ir perguntando a outros indivíduos, sejam eles, colegas, amigos, familiares ou desconhecidos, se sabem onde existem casas disponíveis para aluguel ou venda. Resultando em algo bem cansativo e inseguro. Em relação à serviços, como moto-táxi, oficinas e restaurantes. Em certos momentos não existe a opção de esperar passar um mototáxi, que não dá pra esperar algum amigo indicar uma oficina boa ou um restaurante de qualidade. O desafio está em como fazer essas consultas de forma rápida. Principalmente para pessoas que estão de viagem, é muito difícil saber onde tem um restaurante de qualidade, ou onde ajustar o pneu do carro. E precisará obter informações de terceiros, gerando possíveis atrasos na sua programação, entre outros problemas.

Em relação às oportunidades, pode-se falar dos empregos. Dificuldades encontradas na questão de como desempregados poderiam ver de forma prática onde está precisando de empregado e qual tipo é o emprego. Contudo, se achar um lugar de interesse, surgem dúvidas com a comunicação entre o usuário e o local empregador em relação a salário, entrevistas, currículos etc. Assim, um problema importante é como mostrar de forma fácil, prática, rápida e eficiente pontos de interesse, permitindo o compartilhamento entre usuários.

Contudo, a aplicação desenvolvida neste projeto auxilia os usuários em algumas das diversas dificuldades já citadas, pois a aplicação informa detalhes sobre os locais no mapa, permitindo a interação entre os usuários e o local do mapa, dando suporte também a construção de rotas entre pontos escolhidos do mapa e sua localização atual.

**Definição 1.0.1 (Ponto)** *Um ponto no mapa é a coordenada que possui como atributos principais uma latitude e uma longitude.*

Este ponto será usado para definir locais de referências no mapa.

**Definição 1.0.2 (Local de Interesse)** *Um local de interesse é a representação de qualquer ponto baseado em sua descrição, que designará para o usuário que tipo de serviço pertence a tal ponto, e que desperte seu interesse.*

O projeto atende ao público em geral, usuários experientes com o mundo mobile e seus recursos, como também usuários novos e inexperientes, englobando desempregados que visam a aplicação como uma forma para facilitar o ingresso em um novo emprego ou alguma oportunidade. Viajantes e/ou imigrantes em cidades a fim de adquirir informações sobre pontos (restaurantes, oficinas, bancos, pontos turísticos, hotéis) na mesma. Público voltado à vendas, trocas e divulgações com o intuito de adquirir facilidades e novos meios de realizarem seu trabalho.

A seguir, na Seção 2 está a fundamentação teórica, mostrando os conceitos chaves para o projeto, na Seção 3 os trabalhos relacionados, que mostrará comparações dos trabalhos relacionados a este, na Seção 4 os objetivos gerais e específicos da proposta, logo após, na Seção 5, são os procedimentos metodológicos, ou seja, o passo-a-passo de como tudo será feito, como coletar dados, como implementar as ideias e por último as referências.

## 2 FUNDAMENTAÇÃO TEÓRICA

Nesta Seção, serão abordados os principais conceitos relacionados a este trabalho e qual a contribuição de cada conceito para o desenvolvimento do trabalho.

### 2.1 Android

O Android é uma plataforma para tecnologia móvel completa, muito usada atualmente, envolvendo um pacote com programas para celulares, já com um sistema operacional, middleware, aplicativos e interface do usuário (PEREIRA; SILVA, 2009). A história do Android teve início em outubro de 2003, na cidade de Palo Alto na Califórnia, quando quatro amigos, Andy Rubin, Rich Miner, Nick Sears e Chris White decidiram fundar a Android Inc (MIGUEL, 2017). A empresa desenvolvia sistemas operacionais para celulares, mas todos os projetos eram secretos. Cerca de dois anos depois, em agosto de 2005, a Google anunciou a compra da Android Inc. e esse foi um dos primeiros passos da empresa em direção ao mercado de softwares para dispositivos móveis, onde já lidera a algum tempo.

A plataforma Android para desenvolvedores foi construída em 2013 com a intenção de permitir os desenvolvedores criar aplicações móveis, tais que possam aproveitar de forma completa o dispositivo e seus diversos recursos.

A plataforma é baseada no sistema operacional *linux*, com isso ela é composta por conjuntos de ferramentas que atuam em formas similares ao *linux*, mas não pode ser considerado uma distribuição *linux*, pelo fato de não conter diversos padrões que as distribuições possuem (PEREIRA; SILVA, 2009).

O android foi utilizado neste trabalho pelo fato de atualmente ser o sistema operacional mais usado e mais comum em nível mundial. Segundo MENDIOLA (2017) o avanço espetacular do uso da plataforma se justifica também pelo crescimento econômico de países asiáticos que possuem mercados enormes e estão abraçando em massa a plataforma Android.

A aplicação utiliza o android nativo, pois são elementos simples do próprio android e que não necessita de artefatos externos, como uma abordagem híbrida. Com a aplicação voltada para o android nativo, basta apenas criar suas telas características da aplicação, como mensagens e detalhes, mapa e instruções. Outro fator que justifica a abordagem android é a maneira de como o público ver uma aplicação especialmente para o dispositivo android.

### 2.1.1 *Activity*s

*Activity* é um dos elementos do android. Neste presente trabalho abordaremos a mesma com fins de controle e visualização de telas, pelo fato da aplicação estar vinculado com a importância da usabilidade, priorizando sempre o visual de cada tela, por isso deve ter bastante ênfase neste projeto, pelo fato de sua usabilidade estar sendo avaliada.

Uma *activity* é uma classe que deve herdar da classe `android.app.Activity` do próprio android ou alguma subclasse desta (LECHETA, 2013). De forma geral, representa uma tela de uma determinada aplicação, é responsável pelo tratamento de eventos gerados na tela (*view*), como, por exemplo, quando o usuário clica em um botão para enviar o formulário para o servidor, o tratamento deste envio será feito na *activity* desta *view*, ou seja, cada *view*, contém uma *activity* relacionada para seus possíveis eventos.

Segundo Lecheta (2013) e (Silveira, 2010), a classe da *activity* deve implementar o método `onCreate()`, responsável por carregar os layouts XML e outras operações necessárias para a iniciação a fim de executar uma determinada aplicação. O método é executado somente uma vez durante o ciclo de vida da *Activity*.

## 2.2 Usabilidade

A usabilidade é uma área da qualidade de software que tem vindo a acompanhar a evolução tecnológica, permitindo o desenvolvimento de sistemas mais fáceis de usar e mais eficientes (ALVES; PIRES, 2002). A sua importância tem crescido com o avanço da Internet, dos dispositivos cada vez mais variados, e com o recurso cada vez maior a conteúdos multimédia.

A usabilidade é uma necessidade elementar para sistemas de software interativo. Produtos com baixa usabilidade são, geralmente, responsáveis por frustração, ansiedade, baixa produtividade e, em casos extremos, por abandono do software (MEDEIROS; CYBIS, 2000).

O grau de satisfação dos clientes de empresas fornecedoras de software, pelo menos no que se refere ao produto, está ligado, tanto à sua utilidade como ao grau de usabilidade do produto (MEDEIROS; CYBIS, 2000).

Assim, os questionários de avaliação da satisfação de clientes com relação a produtos de software devem investigar também, o grau de usabilidade percebida por seus usuários, com o intuito de qualificar o produto em geral, de acordo com diversas formas de medir tal qualidade.

De acordo com Nielsen (2003), a forma mais comum de se avaliar a usabilidade

de um software é observando a sua interação com o usuário, podendo ser realizada em um laboratório em no próprio local de trabalho.

Para as empresas de software, foi de fundamental importância a adesão da análise de usabilidade como uma parte do processo de software e os engenheiros de usabilidade, permitindo que os produtos finais tenham mais qualidade e logo possam ter um maior sucesso ou de certa forma, uma vantagem para com os concorrentes.

As principais vantagens do estudo da usabilidade de um produto de software são as seguintes:

- Aumentar a produtividade dos utilizadores
- Aumentar os níveis de utilização do produto
- Reduzir a necessidade de formação e de custos de produção de documentação
- Reduzir os custos de suporte técnico
- Reduzir custos e tempo de desenvolvimento
- Minimizar o re-desenvolvimento e as alterações após a finalização

Atualmente o grande foco da usabilidade é a Web, incluindo aplicações mobiles, porque o número de potenciais utilizadores é enorme relativamente a uma distribuição em CD-ROM, que eram o que a algum tempo era comumente medido pela usabilidade e também devido às condicionantes de largura de banda, algo que não constitui preocupação em aplicações locais e o atual mercado (MEDEIROS; CYBIS, 2000).

A usabilidade significa a forma como os utilizadores realizam determinadas tarefas eficientemente, efectivamente e satisfatoriamente. O estudo da interação homem-máquina é uma área que assume, atualmente, uma maior importância, devido a vivermos na era em que a multimédia está presente nos mais diversos meios. Os designers gráficos defendem na sua maioria, que a usabilidade é um obstáculo à criatividade. Do outro lado, os defensores da usabilidade, defendem que é possível conciliar uma interface graficamente apelativa com a funcionalidade.

O trabalho proposto realizará uma avaliação do aplicativo após a fase de implementação e testes estiver completa. O característica utilizada para a avaliação será a usabilidade, levando em consideração algumas das métricas citadas, a fim de enriquecer a avaliação e detalhar a qualidade da aplicação. A usabilidade irá mostrar a facilidade que o usuário tem para com relação ao aplicativo.

### 2.2.1 Métricas

A usabilidade é dividida em várias métricas, onde torna mais fácil a avaliação de determinada aplicação e/ou sistema. Tais métricas são principalmente voltadas para nível de facilidade, número de ocorrências de determinada situação e tempo decorrido de alguma atividade. Algumas métricas que demonstram isso são:

**Facilidade de aprendizagem** o sistema deve ser fácil de assimilar pelo utilizador, para que este possa começar a trabalhar rapidamente;

**Eficiência** o sistema deve ser eficiente para que o utilizador, depois de o saber usar, possa atingir uma boa produtividade;

**Facilidade de memorização** o sistema deve ser facilmente memorizado, para que depois de algum tempo sem o utilizar, o utilizador se recorde como usá-lo;

**Satisfação** o sistema deve ser usado de uma forma agradável, para que os utilizadores fiquem satisfeitos com a sua utilização.

### 2.2.2 Técnicas de avaliação

#### 2.2.2.1 Heurísticas

A usabilidade pode ser avaliada usando várias técnicas. As mais usuais são a da heurística e dos testes com utilizadores. A heurística é uma das formas de avaliar a usabilidade mais econômica e prática, permitindo detectar problemas na fase de desenvolvimento da interface. Evitando-se erros que ao serem detectados depois da implementação iria gerar custos desnecessários (MEDEIROS; CYBIS, 2000).

A avaliação heurística deve ser feita por uma equipe de especialistas, que podem ser engenheiros da usabilidade, programadores, designers ou utilizadores. Este tipo de avaliação pode ser feita com colegas de trabalho para reduzir custos, mas nunca deve ser feita individualmente porque uma só pessoa, pois esta não tem a capacidade de levantar todas as questões heurísticas (MEDEIROS; CYBIS, 2000). Algumas questões que podem ser colocadas, são:

- Navegação – Eu consigo encontrar o caminho em qualquer lugar em que me encontre;
- Funcionalidade – Consigo fazer aquilo que necessito;
- Controle – Sou em quem comanda as operações;
- Linguagem – Eu entendo a terminologia;

- Ajuda e Suporte – Eu obtenho ajuda quando a necessito;
- Feedback – Eu sei o que o sistema está a fazer;
- Consistência – Eu não preciso de aprender novos truques;
- Erros – Cometer erros é algo que é raro acontecer e são fáceis de corrigir;
- Grafismo límpido – Eu consigo reconhecer os elementos que constituem a interface e o design está nítido e apelativo.

As questões de heurística podem ser levantadas por qualquer pessoa e permitem simular o comportamento do utilizador perante a interface. Como não é possível detectar todos os problemas de usabilidade através da heurística, existem então outros tipos de testes, que são os testes com utilizadores, que permitem validar os testes de heurística e corrigir outros erros que não foram detectados.

#### 2.2.2.2 *Testes com utilizadores*

Ao contrário da heurística em que os testes são efetuados com base no design de um produto que ainda não foi desenvolvido, os testes com os utilizadores obriga a que o produto esteja pelo menos em forma de protótipo para poder ser testado (ALVES; PIRES, 2002). É aconselhável que sempre seja feito um protótipo da aplicação, para que resulte em reduções de custos, caso haja alterações profundas na aplicação, e que sirva também para realizar mais rápido um suposto teste.

Na categoria de testes, o objetivo é simular as condições de utilização do software ou aplicação sobre a perspectiva do usuário final (BARTIÉ, 2002). Os testes devem medir o nível de facilidade disponibilizada pela aplicação, de modo a deixar o software mais simples, intuitivo e interativo. Os testes são voltados para a facilidade de navegação entre telas, clareza de textos e mensagens que são exibidas para o usuário, padronização visual, cores, mecanismos de apoio ao usuário entre diversos outros aspectos (BARTIÉ, 2002).

Após o produto concluído, os testadores, devem estar em um local que simula o mais fielmente um ambiente de trabalho provável para o uso de uma tal aplicação. Para cada utilizador deve existir um observador que irá retirar notas de todos os movimentos e reações do utilizador (ALVES; PIRES, 2002). O observador deve ser planejada de modo que o mesmo não interfira no trabalho do utilizador. A seção de teste pode ser gravada para analisar melhor em algum outro momento. O utilizador irá testar a interface tendo por base um conjunto de tarefas que foram definidas como princípios heurísticos. Para cada atividade feita pelo utilizador, o observador

deve monitorar os tempos, cliques, hesitações, movimentos do mouse, expressões do utilizador, etc.

Esta será uma metodologia para avaliar a aplicação proposta neste trabalho, onde poderemos ver a facilidade ou dificuldade dos usuários perante a aplicação, e posteriormente a aplicação de um questionário para avaliar e validar as informações já coletadas.

A usabilidade de uma interface é então quantificada através do estudo estatístico de todos os dados recolhidos (ALVES; PIRES, 2002). Com esses dados, pode-se então iniciar o melhoramento em relação aos problemas detectados antes de o produto ser publicado.

### 2.3 API

Na prática, uma API é simplesmente a exposição de uma série de ferramentas, métodos de programação e protocolos, com o objetivo de facilitar a programação de uma aplicação (LANE, 2012). *Application Programming Interface*, as APIs, podem proporcionar a exposição de apenas uma parte do seu sistema, de maneira que o *backend* fique protegido e você possa fazer a liberação de apenas algumas funcionalidades de seu serviço, como por exemplo, apenas consulta (MALAVASI, 2015).

É uma Interface, cheia de ferramentas que possibilitam a Programação de Aplicações. A API é o livro de regras, que estabelece os padrões de comunicação que poderão ser usados naquela interação (PELOI, 2016). Se eu quero programar uma aplicação que use os dados do Twitter, eu devo usar o livro de regras conforme definido pelo Twitter, pela API do Twitter, da mesma forma com o Facebook, Instagram, Google Maps.

É claro que a API pode fazer muitas outras coisas, ainda mais se ela estiver atrelada à uma Ferramenta de Gerenciamento de APIs (como o *API Management Suite*). Vamos dar um exemplo simples de API e fazer o paralelo com a comunicação sem APIs, na vida real.

APIs são apenas uma peça do que a empresa chamaria de Arquitetura Orientada a Serviços (S O A). As APIs não são novas. Mas por volta de 2000, uma parte da experiência SOA deixou a empresa e encontrou um ambiente mais fértil no mundo das *startups* (PELOI, 2016).

Algumas das justificativas pelo qual as APIs dispoem seus dados, são:

- Gerar inovação sobre seus próprios dados, de uma maneira que não poderia sozinho;
- Gerar desenvolvimento de sua própria plataforma em novos meios;
- Gerar mais mercado e expandir o público-alvo;

- Organização do setor de TI, em que os consumidores das APIs são as outras equipes internas da empresa, e as APIs expostas funcionam como portais de troca de dados entre setores.

As APIs facilitam muito a implementação de muitas aplicações, como neste trabalho será preciso a utilização de mapas e rotas, a decisão de utilizar APIs era indispensável. As APIs ajudam também a deixar o aplicativo padronizado, ou seja, equivalente com os atuais do mercado, pois muitos aplicativos atuais, utilizam alguma API para a construção e/ou funcionamento. As APIs utilizadas serão o Google SignIn para realizar o login pelo google, com o intuito de deixar tudo integrado na própria aplicação; Google Maps, para a visualização, edição e manipulação de mapas e pontos; e também a Google Directions para requisitar, receber e utilizar as rotas que o google oferece para os desenvolvedores.

### **2.3.1 Arquitetura Cliente-Servidor**

O cliente é o lado responsável por fazer pedido de recursos. Esses pedidos são chamados de requisições, e é feito à API. A API recebe tal requisição e consulta um computador especial, responsável por organizar a chegada das requisições e buscar os recursos requisitados, esse computador especial é o lado servidor (PELOI, 2016). Esse é o modelo Web APIs, que funcionam em aplicações *web* e/ou *mobile*. Contudo, também existem as APIs locais, como as dos sistemas operacionais ou que integram diferentes aplicativos.

Cliente e servidor são as duas extremidades ligadas por uma API (PELOI, 2016). Um programa executa no cliente, fazendo as requisições e outro no servidor, respondendo as requisições e mandando de volta ao cliente, ou seja, a API é a interface. Uma API envolve diversos outros aspectos de alta relevância, como controle de tráfego, segurança, autenticação, formatos de dados, design e o próprio gerenciamento.

### **2.3.2 API dos Códigos-Fonte**

As APIs dos códigos-fonte oferecem bibliotecas de objetos, classes e diversas outras coisas para a implementação de novas aplicações. As API dos códigos-fonte geralmente são utilizadas no desenvolvimento de projetos para criar um aplicativo composto (JOHNSON, 2014). As chamadas do aplicativo em desenvolvimento para a API, é feita na maioria das vezes de forma simples e similar com outras, e são feitas de acordo com as normas do ambiente do aplicativo.

### **2.3.3 *API de Legado***

Uma variedade de abordagens de legado para os aplicativos de interfaces fazerem uso de arquivos e ideias unidimensionais, protocolos de objeto remoto, interfaces de sistema operacional, API de hardwares, protocolos de comunicação, filas de mensagem e outros meios (JOHNSON, 2014). Tal legado, representa que esse grupo é um conjunto de protocolos antigos e de longa duração, que podem ser usado em diversas aplicações atuais de acordo com o contexto das mesmas.

### 3 TRABALHOS RELACIONADOS

O presente trabalho baseia-se em informações, métodos, conceitos e técnicas coletadas de livros, artigos e trabalhos científicos coletados das áreas de programação java e android, cálculo e análise de rotas, criação de pontos no mapa e análises de usabilidade. Os trabalhos de Lima *et al.* (2012), Thomas e Favaretto (2017) e Antunes (2015) são os principais contribuidores para o desenvolvimento deste trabalho.

Assim como nesses três trabalhos citados, o trabalho desenvolveu uma aplicação na plataforma android, calculando rotas, inserindo e editando pontos no mapa. Este trabalho possui um diferencial, o mesmo possibilita que usuários troquem mensagens entre eles e os pontos de interesse do mapa, a fim de manter uma comunicação fácil entre os dois usuários. Entretanto, o trabalho desenvolvido visou não somente implementar apenas tais funcionalidades, mas também aplicar uma avaliação para medir a usabilidade da aplicação, usando métricas e questionários online.

#### 3.1 UbibusRoute - Um Sistema de Identificação e Sugestão de Rotas de Ônibus Baseado em Informações de Redes Sociais

Esse trabalho apresenta o UbibusRoute, um sistema móvel que considera informações contextuais dinâmicas do trânsito provenientes de redes sociais e apresenta informações sobre as rotas aos usuários, utilizando a API do Google Maps, como fonte de geolocalização e a API do Twitter como provedora de informações dos cidadãos. Ou seja, a partir dos tweets que usuários fizerem, a aplicação tem a capacidade de recolher e analisar tais tweets para melhorar as rotas dos ônibus.

O trabalho de Lima *et al.* (2012) descreve que é possível gerar rotas no mapa de acordo com informações externas. Essas rotas serão utilizadas para auxiliar automóveis, de uma forma que os mesmos percorra de uma maneira melhor e sem muitas complicações, tais como, congestionamento, acidentes e etc. O trabalho desenvolvido também possui a funcionalidade para desenhar e calcular rotas possíveis, pois mostra graficamente a distância entre pontos do mapa e o ponto atual do usuário. Segundo Lima *et al.* (2012), o cenário atual do trânsito nas grandes cidades brasileira vem piorando a cada momento e com o crescimento do número de carros particulares nas vias, motos e outros veículos, os congestionamentos tornaram-se cada vez mais frequentes. Com isso, a função de calcular e desenhar rotas pode ajudar bastante os

cidadãos nos dias atuais, facilitando a locomoção tanto de pessoas, quanto qualquer tipo de veículo.

Segundo Lima *et al.* (2012), os Sistemas Inteligentes de Transporte se transformaram em uma alternativa que atrai o público, ao mesmo tempo são considerados viáveis para resolver alguns problemas e superar desafios na área de transporte das grandes cidades, essa evolução se deve em parte ao acelerado crescimento da Tecnologia da Informação e Comunicação. A aplicação foi desenvolvida e colocada no mercado. Porém, existem limites, já que o usuário não tem a liberdade de inserir pontos para ser usados, ou seja, o trabalho é voltado apenas para identificação e recomendação de rotas para meios de transportes públicos. O público alvo para o aplicativo é restrito, pois necessita da relação com os meios de transportes públicos, podendo apenas utilizar e ajudar na recomendação que o aplicativo tem o objetivo de disponibilizar, diferenciando assim do aplicativo desenvolvido neste trabalho.

### **3.2 Trocacomigo: Aplicativo de Consumo Colaborativo**

Thomas e Favaretto (2017) mostram em seu trabalho que foram realizados estudos sobre consumo, consumismo e hiperconsumo, traçando suas diferenças em relação ao tipo de consumo defendido no trabalho, abordado como consumo colaborativo. Thomas e Favaretto (2017) decidiram construir uma aplicação mobile capaz de conectar usuários que possivelmente possam ter os mesmos interesses. Essa conexão foi feita com base na geolocalização dos usuários, interessados em vender, trocar, alugar ou doar seus bens de consumo reutilizáveis. Thomas e Favaretto (2017) supõe que essa pode ser uma concorrência para o mercado atual visto nas redes sociais.

Uma forma de mostrar os possíveis pontos para troca, doações e aluguel, é de forma visual no mapa, como fez Thomas e Favaretto (2017). Assemelhando a este trabalho onde também é preciso inserir no mapa tais pontos, onde o usuário pode visualizar seu interesse sobre tal ponto. A relação entre os dois trabalhos trata simplesmente desta criação de pontos, pois o próprio usuário que tem a liberdade de criar seu ponto no mapa e o mesmo tem também o controle sobre ele, em relação a edição e remoção. Contudo, ambos visam o capitalismo como algo difícil de se lidar. Tentando tratar essa dificuldade com trocas, doações e conversações entre clientes e proprietários de serviços como vendas e moto taxis, a fim de chegar a acordos financeiros razoáveis para os dois lados, por meio de conversas e também comparações.

Porém o aplicativo desenvolvido no trabalho de Thomas e Favaretto (2017) não

aplica o desenvolvimento e cálculos de possíveis rotas para os usuários, também limitando a aplicação apenas para visualização de pontos de venda, troca, aluguel ou doação e seus detalhes, já que não existe uma comunicação entre o usuário e o responsável pelo ponto escolhido.

Por fim, o aplicativo desenvolvido foi entregue ao público com o intuito de disseminar a ideia do consumo colaborativo, estimulando a sociedade para refletir e realizar um consumo cada vez mais sustentável, de certa forma, garantindo um mundo mais verde para todos.

### **3.3 RouteMe - Uma Aplicação para Dispositivos Móveis Android para Resolução de Problemas de Rotas de Veículos (PRV)**

Antunes (2015) propôs o trabalho a seguir, que foi realizado no âmbito do Mestrado em Computação Móvel do Instituto Politécnico da Guarda. Antunes (2015) utilizou algumas ferramentas e serviços disponíveis, em especial o Google Maps, para aquisição de dados georreferenciados usados na definição dos PRV e também características de interatividade com o utilizador, onde o mesmo possa construir e resolver os PRVs. Contudo, o projeto consiste em desenvolver uma aplicação para dispositivos móveis Android, de modo a permitir a qualquer utilizador proceder à resolução de PRV através desse tipo de dispositivos.

Há diversas semelhanças entre esse trabalho e o trabalho proposto. A questão de rota foi abordada de maneira similar. Tais rotas precisam ser calculadas e de certa forma consideradas uma boa escolha entre dois pontos. Uma questão abordada no trabalho de Antunes (2015) é a maneira de passar em todos os pontos necessários de uma forma não tão dispendiosa, assemelhando a este projeto que também pode ter essa ocasião de um usuário precisar de passar por mais de um ponto. Logo, o usuário deve analisar e ver qual seria o cálculo dessa rota. Se for um serviço de moto táxi, dependendo do resultado desse cálculo, os usuários podem comunicar-se e chegar a um acordo financeiro, pois ambos saberão o quanto foi percorrido durante o serviço.

O trabalho de Antunes (2015) está voltado apenas para a solução de problema de rotas de veículos, onde tais soluções devem ser a partir cálculos sob rotas de forma que possa resultar na melhor alternativa ou próximo a melhor. Outro fator que denota a diferença do trabalho aqui desenvolvido com o de Antunes (2015) é que seu aplicativo não possui a liberdade para o usuário criar seus próprios pontos, também não possui uma comunicação clara e bidirecional entre clientes e fornecedores.

A Tabela 1 mostra uma comparação entre os trabalhos relacionados e ao proposto.

Tabela 1 – Comparação entre os trabalhos relacionados e o proposto

	<b>Comunicação entre pontos</b>	<b>Cálculo de Rotas</b>	<b>Pontos no Mapa</b>	<b>Controle da Usabilidade</b>	<b>Tema</b>
UbibusRoute	Não	Sim	Não	Não	Rotas para transportes públicos com o auxílio da mineração de texto
TrocaComigo	Não	Não	Sim	Não	Visualização de pontos para troca, aluguel, venda ou doação
RouteMe	Não	Sim	Sim	Não	Cálculo para melhoramento de rotas
Presente Trabalho	Sim	Sim	Sim	Sim	Desenhos de rotas e criação de pontos no mapa para visualização e comunicação entre usuários

Fonte: Elaborada pelo autor.

## 4 METODOLOGIA

Para o desenvolvimento do aplicativo apresentado neste trabalho, foi utilizado o Scrum Solo como base para a metodologia de desenvolvimento. O projeto de desenvolvimento de uma aplicação possui várias etapas. As etapas utilizadas neste trabalho foram a de Planejamento, de Construção, de Testes e de Implantação. As sprints foram realizadas no período de 28 dias (quatro semanas), notando que as mesmas não tiveram muitas tarefas pelo fato de não ser realizadas por várias pessoas. A seguir, foram divididas as etapas em várias partes, cada subseção detalha o que foi feito em cada etapa do desenvolvimento, seguindo a ordem cronológica das atividades. De forma que cada etapa representa uma parte do desenvolvimento em geral, gerando artefatos que caracterizam cada uma facilitando e auxiliando o processo de desenvolvimento.

### 4.1 Planejamento

Para o desenvolvimento na plataforma Android, utilizamos a linguagem JAVA, realizando as operações dos controllers, serviços entre outros componentes. Essa é a linguagem que representa o lado do cliente, pois, toda a interface basicamente é implementada pelo XML e controlada e realizada pelo JAVA.

Para o lado do servidor, foi usada a linguagem PHP, muito comum hoje em dia por suas facilidades e ferramentas que auxiliam na comunicação entre cliente e o servidor. O sistema de gerenciamento de banco de dados utilizado foi o MYSQL, que também é bem comum trabalhar em conjunto com o PHP e com o JAVA, alguns sites de hospedagem possuem suporte apenas para tal SGBD, como por exemplo o Hostinger, que usaremos neste trabalho.

A API do Google Maps foi usada com a finalidade de trazer funcionalidades do mapa para a aplicação, com o principal objetivo de fazer a conversão entre endereços e coordenadas no mapa.

Para a avaliação do aplicativo foi utilizado a usabilidade como princípio de qualidade do mesmo, as métricas utilizadas para analisar a usabilidade são nível qualitativo de facilidade de uso de uma função, número de problemas de compreensão de interface detectados e número de consultas ao sistema de ajuda, com a intenção de saber se o usuário está satisfeito com a interface e outros pontos da aplicação.

#### **4.1.1 Requisitos da Aplicação**

Existem alguns requisitos para o aplicativo, alguns com intuito de trazer as qualidade de alguns apps já existentes e outros visando as ideias e melhorias para o aplicativo em si. A tabela de requisitos está detalhada mais abaixo no Apêndice A.

Os requisitos foram definidos no planejamento do trabalho e atualizados e corrigidos de acordo com a evolução do mesmo. Cada requisito possui as diversas tarefas associadas a ele, ou seja, para concluir o requisito “Efetuar login pelo Google”, é preciso configurar o ambiente para a API do Google Sign In, criar o layout, realizar as comunicações de activitys, entre outras.

A verificação dos requisitos foram feitas assim que cada um era completo, porém será mais detalhada na etapa de Testes.

O diagrama de caso de uso mostra a relação das atividades com seus respectivos atores e pode ser analisado no Apêndice B.

#### **4.1.2 Ambiente e Arquitetura**

Utilizamos algumas ferramentas e ambientes para o desenvolvimento da aplicação. Como o Android Studio, que é o ambiente de desenvolvimento integrado (IDE) oferecido pela Google para desenvolvedores que desejam criar aplicações para a plataforma Android. A ferramenta tem como base o IntelliJ IDEA e ele oferece um ambiente completo para lidar com as mais variadas aplicações criadas para o SO citado. Outro ambiente é o Android SDK, que possui diversas bibliotecas e ferramentas para criar, testar e debugar os aplicativos criados para a plataforma Android. A IDE utilizada para integrar algumas dessas ferramentas será o Android Studio.

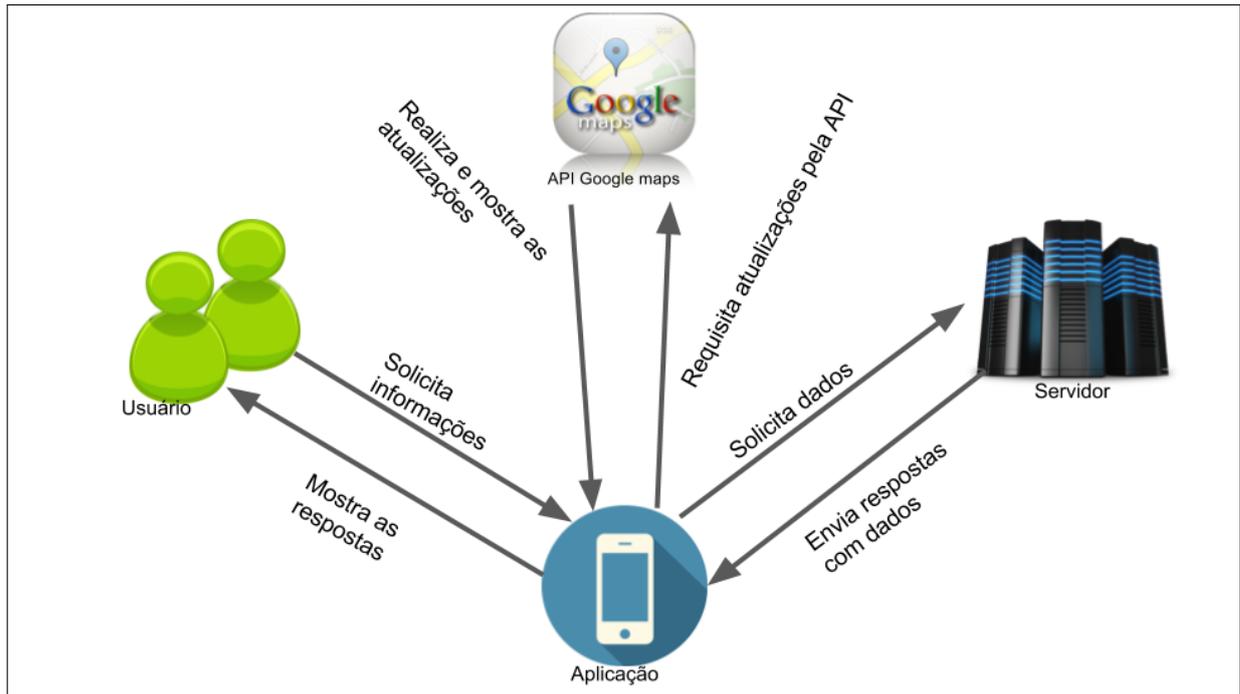
Utilizamos o Sublime Text 3 para a codificação juntamente com o android studio, por questões de performance do notebook. Ao utilizar o sublime text, a compilação e instalação foram feitas pelo terminal utilizando o gradlew e o adb respectivamente.

Utilizamos também o Filezilla para manipular os arquivos do servidor PHP e o Mega.nz para subir o app final e disponibilizar para o público.

A aplicação desenvolvida neste trabalho usa a API do Google Maps para o tratamento e conversões em relação a coordenadas e endereços, ajudando a fornecer informações adicionais para os locais permitindo a interação do usuário com o mapa. O usuário visualiza a interface da aplicação solicitando possíveis funcionalidades que estará presente na mesma, após essa

solicitação, dependendo do tipo de solicitação, é requisitado dados de diferentes servidores utilizando ou não a API do Google Maps. Quando a solicitação do usuário é em relação ao mapa será feito duas requisições, primeiramente é feito a conexão via API aos servidores da Google, retornando os pontos que a API já contém e todas ferramentas a serem utilizadas. A Figura 1 mostra a arquitetura da proposta aqui sugerida.

Figura 1 – Arquitetura da aplicação



Fonte: o autor.

Os locais são adicionados a partir da tela do mapa, ao clicar em um ponto no mapa, você pode inserir um novo local de interesse, ao inserir, as informações desse local são dirigidas ao servidor PHP. Em relação a comunicação, funciona de forma simples, as mensagens serão providas a partir da tela de detalhes, ao clicar em um local de interesse, abrirá a tela de detalhes de tal ponto, ao enviar uma mensagem, a mesma será enviada para o banco, o usuário que enviou a mensagem, a mensagem e o destinatário, que no caso é o local de interesse.

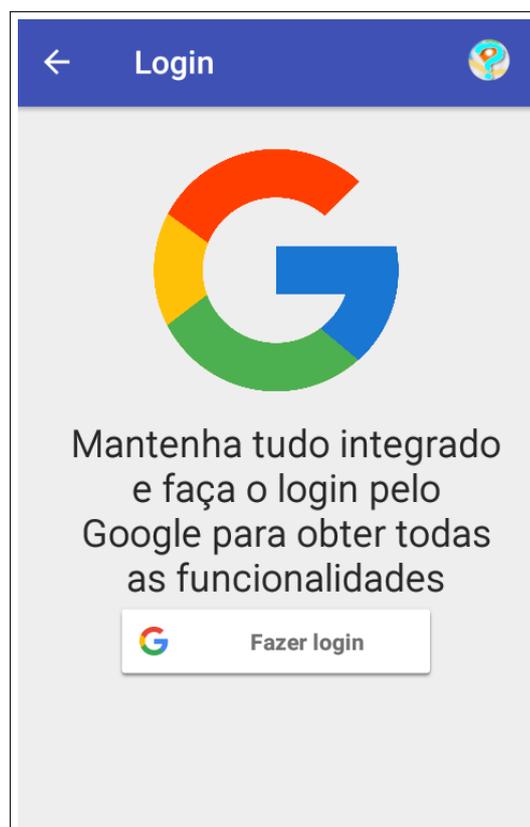
Em relação às rotas, serão tratadas apenas na API, ou seja, não guardaremos rotas em bancos de dados, serão calculadas em tempo de execução da aplicação.

## 4.2 Construção

### 4.2.1 Desenvolvimento dos Layouts e Activitys

O Layout de login tem uma breve descrição da relação do google com o aplicativo e um botão possibilitando o login com o google utilizando a Google Sign-In API. Na parte superior uma ActionBar *app\_bar\_login.xml*, onde mostra o nome da atual tela, a seta para voltar para a tela anterior e um menu *menu\_login.xml* que possui um item que redireciona o usuário para a tela de tutorial. O usuário deslogado, só pode visualizar a tela de ajuda, a tela com o mapa, seus locais e calcular rotas mas não terá a permissão para interagir com os mesmos. Após logar, é redirecionado para a página inicial da aplicação, porém somente se o login for bem sucedido. Com o usuário logado, o mesmo tem a permissão para criar locais e enviar mensagens para esses locais, assim como as funcionalidades que o usuário não logado possui.

Figura 2 – Layout Login



Fonte: o autor.

A tela inicial tem um botão que rediciona o usuário para a tela de mapa e algumas instruções necessárias para a tela, como também o menu lateral deslizante detalhado mais a frente. Na parte superior existe um ActionBar chamado *app\_bar\_main.xml*, onde mostra o nome

da atual tela e um *menu main.xml* que possui um item que redireciona o usuário para a tela de tutorial.

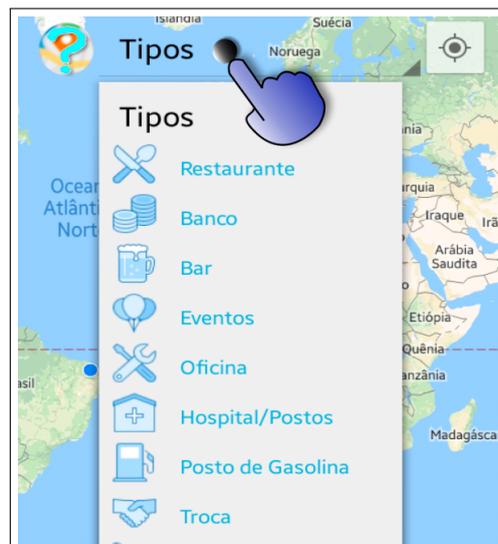
Figura 3 – Layout Inicial



Fonte: o autor.

A tela de mapa mostra todos os locais de interesse após filtrados por tipos, com um select na barra superior do mapa e um icone que redireciona o usuário para a tela de tutorial.

Figura 4 – Tipos



Fonte: o autor.

Ao dar um clique longo em um determinado ponto do mapa, surge um fragmento chamado *insert\_location.xml* que possibilita a criação de um ponto com as coordenadas que o

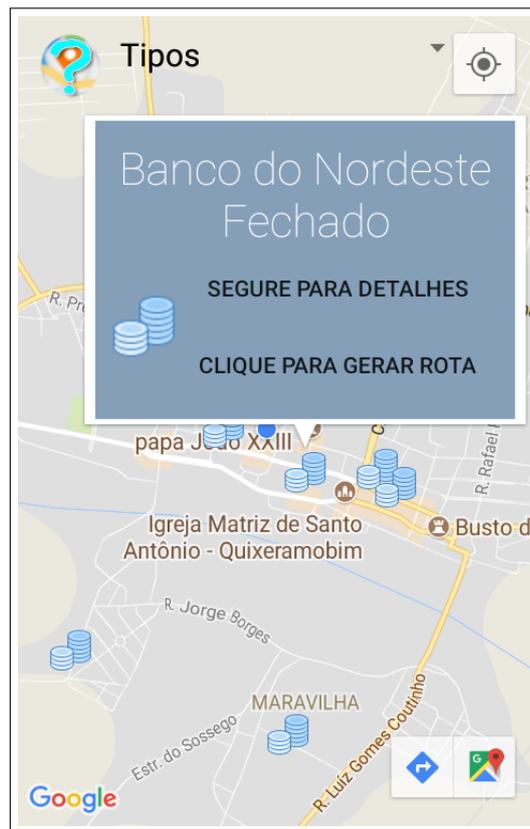
ponto representa.

Figura 5 – Layout insert\_location

Fonte: o autor.

Nesse layout também temos a representação dos markers, que são os pontos de interesses, cada marker é representado por uma janela de informações chamada de *info\_window.xml*, nela contém o nome e o horário de funcionamento ou a condição de aberto ou fechado de tal ponto. Com a janela de informações de determinado ponto de interesse aberta, podemos efetuar duas funções importantes da aplicação, ao clicar na janela a aplicação desenhará no mapa uma rota indo da sua localização atual até o ponto selecionado, ao clicar de forma longa, a aplicação lhe redirecionará para a tela de detalhes daquele ponto. Pode-se editar a latitude e longitude de um ponto de interesse caso você seja proprietário, basta localizá-lo no mapa e clicar longo e arrastar.

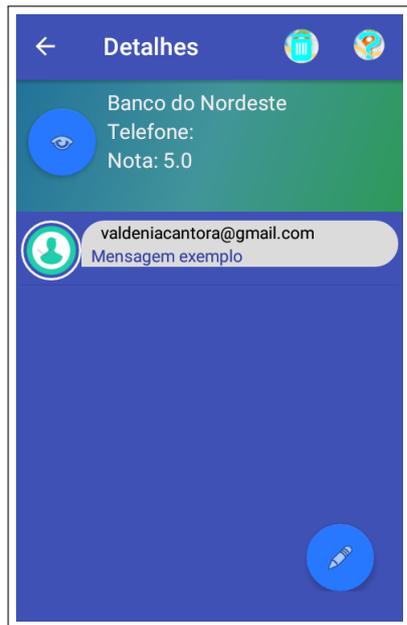
Figura 6 – Layout InfoWindow



Fonte: o autor.

A tela de detalhes possui alguns detalhes como nome e telefone do ponto, também um *ListView* de mensagens para aquele ponto, logo abaixo existe um campo que representa o envio de mensagens. Na parte superior existe um *ActionBar* *app\_bar\_detalhes.xml*, onde mostra o nome da atual tela, a seta para voltar para a tela anterior e um menu *menu\_detalhes.xml* que possui um item que redireciona o usuário para a tela de tutorial e outro item que possibilita a exclusão do ponto selecionado (caso o usuário seja proprietário do mesmo). Para enviar uma mensagem para um local de interesse, o usuário deve estar logado e efetuar um clique longo em cima do local, após ser redirecionado para a tela de detalhes, o usuário deve clicar no campo da parte inferior do layout, abre-se um pequeno fragmento chamado *insert\_message.xml* para escrever a mensagem, com um botão de enviar e outro botão para cancelar. Após clicar em qualquer botão, é dado o feedback do processo realizado e redirecionado para a tela de mapa. A função de editar usa o mesmo layout de inserir uma mensagem, enquanto para inserir uma resposta utilizamos o *insert\_answer.xml*. As funções de responder, editar e deletar mensagens podem ser encontradas no menu *menu\_message.xml* e aparecem ao clicar longo em determinada mensagem ou resposta.

Figura 7 – Layout Detalhes



Fonte: o autor.

A tela de ajuda mostra para o usuário todas as instruções que poderão ser necessárias para as funcionalidades da aplicação. Desde o login, até a criação de rotas, pontos e troca de mensagens. Esse layout foi feito utilizando a API de AppIntro, onde em cada slide possui um botão de próximo, um botão de pular e uma view para lhe informar a quantidade e o progresso dos slides.

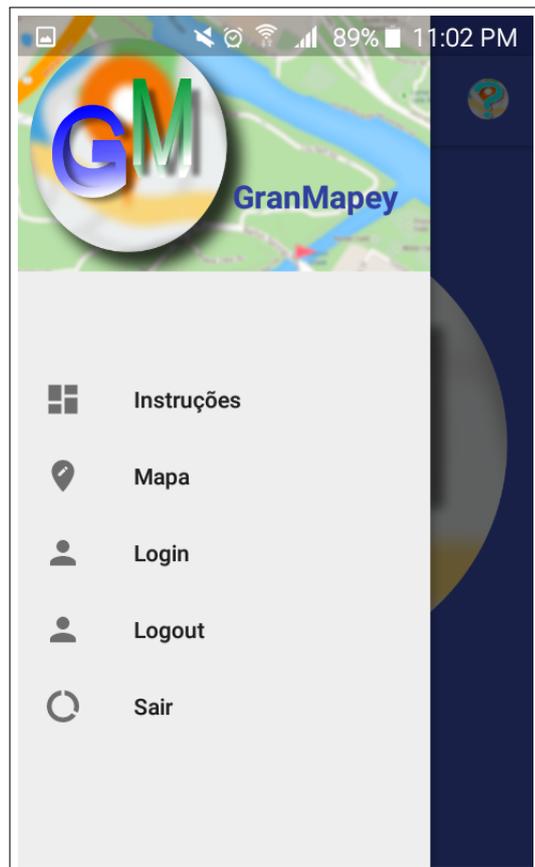
Figura 8 – Layout Tutorial



Fonte: o autor.

Existe o menu lateral deslizante no padrão Google Navigation Drawer do android chamado de *activity\_main\_drawer.xml*, pode ser encontrado na tela principal da aplicação, onde mostra o nome da aplicação, e as principais funções, como itens para redirecionamentos para a tela de login, a tela de mapa, a tela de instruções, a função logout e a função sair.

Figura 9 – Layout Menu Lateral



Fonte: o autor.

#### 4.2.2 Modelos, DAOs, Adapters e Utils

Os modelos utilizados pelo aplicativo ficam localizados em um determinado pacote, “*br.com.example.juniorf.tcc.MODEL*” e servem para manipular tanto o back-end quanto o banco de dados, de acordo com seus atributos e métodos.

- DetalhesGoogle - São os detalhes que vem do google, após realizar uma requisição ao Place Details com o *id\_reference* do ponto, a aplicação recebe um Json com diversos dados, são selecionados alguns como o endereço completo e formatado (*String address*), o level de preço (*String priceLevel*), o website do ponto selecionado (*String webSite*) e a situação do ponto, se está aberto ou fechado (*Boolean openNow*). Todos esses dados são mostrados ao clicar nos

detalhes de um determinado ponto.

- Mensagem - O modelo responsável pelas mensagens e respostas, é composto pelo identificador único da classe (*Integer id*), um campo de texto para a mensagem (*String texto*), um campo que contém o email de origem da mensagem (*String emailOrigem*) e outro com um email de destino (*String emailDestino*), um campo que guarda o nome do local do ponto para futuras requisições (*String local*) e um booleano (*boolean answer*) que diz se a classe é uma Mensagem ou Resposta. Essa diferença influencia na parte de organização da listView da tela Detalhes.
- MyLocation - É o principal modelo da aplicação, representando o ponto de interesse, contendo um identificador único (*int id*), um título (*String nome*), um telefone (*String telefone*), a latitude (*double lat*) e a longitude (*double lng*), que representam a coordenada do ponto no mapa, um tipo (*String tipo*), que denota e classifica os pontos, um campo com o proprietário do ponto (*String email*), um identificador com associação com o id dos servidores do Google (*String id\_reference*) e um outro campo com o horário de funcionamento (*String horario\_funcionamento*) do ponto.
- Tipo - É o modelo responsável pelo auxílio da classificação dos pontos de interesse (*MyLocation*), contendo um identificador único (*Integer id*), um campo contendo o valor a ser mostrado na front-end (*String tipo*) e outro valor usado como termo de busca (*String type*), foi feita essa separação pelo fato de ser preciso realizar as buscas em dois servidores diferentes com o mesmo atributo, o do Google e o desenvolvido neste trabalho.

O pacote “*br.com.example.juniorf.tcc.DAO*”, guarda os arquivos com o padrão DAO (*Data Access Object*) da aplicação, onde são armazenadas as operações de comunicação com o servidor php e o banco de dados *MySQL*. Todos os métodos utilizam o *framework Volley* para as requisições ao servidor, os principais métodos usados nos DAOs são: *insert* para inserir, *update* para atualizar, *delete* para apagar e o *getAll* para obter todos os dados de determinada tabela do banco de dados.

O *adapter* utilizado para as mensagens ficam localizados em um determinado pacote, “*br.com.example.juniorf.tcc.ADAPTER*” e é responsável por preencher a lista com as mensagens e respostas da tela de Detalhes.

O pacote “*br.com.example.juniorf.tcc.CONFIG*” contém uma classe de configuração

geral da aplicação e para a utilização do *framework Volley* para as requisições necessárias.

A aplicação utiliza diversas urls, tanto para comunicar-se com as APIs Google, como ao próprio servidor, logo, tais urls estão armazenadas em uma interface chamada *Codes* no pacote “*br.com.example.juniorf.tcc.CONSTANTS*”, contendo uma classe com métodos que são úteis para diversas outras classes, como verificar conexão com a internet, verificar GPS ativo e exibir mensagem de erro, essa classe denomina-se *UtilMethods*.

O pacote “*br.com.example.juniorf.tcc.LOGIN*” contém apenas a *activity* de Login(*LogInActivity.java*), com o intuito de organização dos arquivos da aplicação, assim como o pacote “*br.com.example.juniorf.tcc.MAPS*” contendo a *activity* de mapa (*MapsActivity.java*), o pacote “*br.com.example.juniorf.tcc.DETALHES*” com a *activity* de detalhes(*DetalhesActivity.java*) e ainda o pacote “*br.com.example.juniorf.tcc.TUTORIAL*” com a *activity* do tutorial da aplicação (*TutorialActivity.java*), com todos os slides necessários para o tutorial.

### 4.2.3 WEB SERVICE

O WEB SERVICE foi desenvolvido em conjunto com a aplicação, a cada tabela ou dado precisos na aplicação era realizado um *update* no servidor com novos arquivos e funções. Basicamente existe um *CRUD* para cada tabela e alguns arquivos de requisições especiais com alguns parâmetros para busca. O *CRUD* representa o *create* de criação, *read* para leitura, *update* de atualização e *delete* para apagar os registros.

O servidor foi desenvolvido na linguagem PHP, que tem ótimas funcionalidades e atributos para se trabalhar com manuseio de dados, a linguagem já possui diversas estruturas para manipular o JSON, já que é a formatação necessária para manipulação do framework no *back-end* (*Volley*).

O servidor está online desde o início do desenvolvimento. Para ler, escrever e atualizar os arquivos do servidor foi preciso utilizar uma ferramenta chamada *Filezilla* que ajuda a realizar uma conexão FTP, com o intuito de obter e atualizar os arquivos do servidor.

A aplicação utiliza o servidor a cada requisição, logo, o mesmo é fundamental para o bom desempenho do app. O WEB SERVICE trabalha em conjunto com os servidores Google, em relação aos pontos de interesses e tipos.

Existe um cabeçalho em cada arquivo PHP do servidor, contendo a conexão com o banco de dados (*MYSQL*), logo após, existe as consultas e os retornos para a aplicação, a Figura 10 mostra como é feito o cabeçalho:

Figura 10 – Cabeçalho PHP

```

try{
    $db = new PDO('mysql:host=mysql.hostinger.com.br;dbname=
u694924616_bd;charset=utf8', 'u694924616_root', '
rootroot');

}catch (PDOException $e) {

    echo "Not Connected";
    die();
}

```

Fonte: o autor.

O *PDO (PHP Data Objects)* é uma extensão que auxilia no trabalho do php com o banco de dados. A variável "*db*" armazena a conexão realizada pelo objeto PDO com o primeiro atributo sendo o *MySQL* o sistema gerenciador do banco de dados, seguido com o host e o nome do banco, o segundo atributo é o usuário administrador do banco e o terceiro atributo a senha referente ao usuário preenchido. Caso a conexão não tenha sucesso, é retornado para a aplicação uma falha de conexão.

A parte da consulta é onde deve ser realizado a parte de manipulação dos dados do banco, como consultas sql, a figura 11 mostra uma consulta que retorna todos os pontos de interesses filtrados por tipo cadastrados no banco. Primeiro recebemos da aplicação o tipo a ser utilizado na consulta, logo após, criamos a variável "*query*" que prepara a consulta a ser executada. Seguindo a *query* será executada.

Figura 11 – Consulta SQL

```

$type = $_POST["typez"];

$query = $db->prepare("SELECT * FROM Location where Tipo = '
$type'");
$query->execute();

```

Fonte: o autor.

E por último a parte de retorno, onde os dados já estão todos em suas variáveis e prontos para serem retornados para a aplicação, a Figura 12 mostra esse processo:

Figura 12 – Retornando em JSON

```
if($query->rowCount() >= 0){
    $data = $query->fetchAll(PDO::FETCH_ASSOC);
    echo json_encode($data);
}else{
    $json['success'] = 0;
    $json['message'] = 'No Data found';
    $json['myintro'] = '';

    echo json_encode($json);
}
```

Fonte: o autor.

Primeiramente é realizado um teste, para ver se a *query* não está vazia, caso seja aprovado no teste, uma variável (*data*), recebe um *array* multidimensional como retorno do *fetchALL* (função do PDO). Após a variável *data* for preenchida, retornaremos a aplicação os valores em forma de JSON, utilizando uma própria função do PHP (*json\_encode*). Se a condição da *query* não vazia for falsa, retornaremos para a aplicação um outro JSON contendo a mensagem de dados não encontrados. A Figura 13 mostra um exemplo do arquivo JSON obtido em uma requisição na entidade de Tipos.

Figura 13 – Exemplo JSON

```
[
  {
    "id": "1",
    "Tipo": "Tipos",
    "type": "0"
  },
  {
    "id": "2",
    "Tipo": "Restaurante",
    "type": "restaurant"
  },
  {
    "id": "3",
    "Tipo": "Banco",
    "type": "bank"
  }
]
```

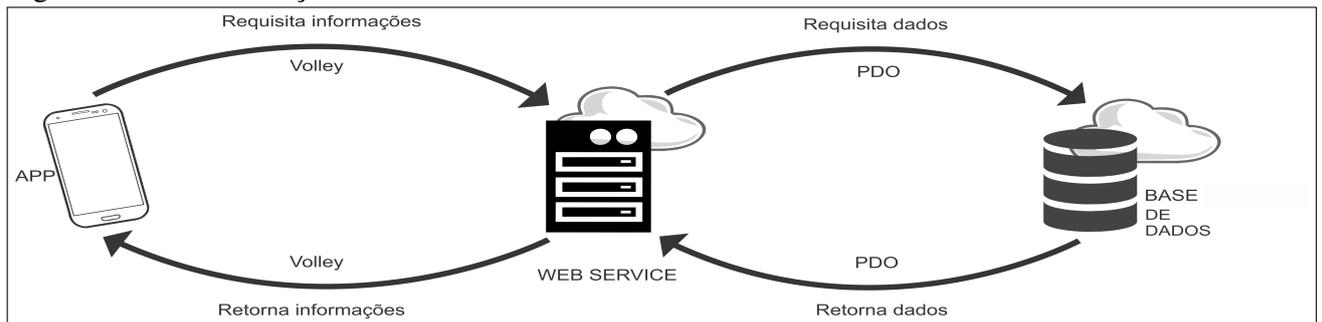
Fonte: o autor.

O resultado acima mostra o resultado de uma consulta sql que retorna os três primeiros tipos cadastrados no banco de dados com a finalidade apenas de demonstração do resultado JSON.

#### 4.2.4 Comunicação com WEB SERVICE

A comunicação da parte da aplicação para o WEB SERVICE é feita utilizando o *framework Volley*, processando as requisições HTTP e recebendo os responses também com o *Volley*. Do WEB SERVICE para a base de dados é utilizado o PDO, tanto para as requisições e responses. O processo é demonstrado na Figura 14.

Figura 14 – Comunicação com WEB SERVICE



Fonte: o autor.

Primeiramente precisamos informar o *Volley* que faremos uma requisição utilizando o método `newRequestQueue(Context t)`, logo após, criamos o objeto de requisição, contendo o método (`GET` ou `POST`), a url e um novo *Listener* para o *response*. Para o *Listener* ser utilizado, precisa implementar dois métodos do *Response*, que são o `onResponse(String response)`, responsável por tratar a string quando obtida e o `onErrorResponse(VolleyError error)`, responsável em dizer caso algum erro tenha ocorrido. O terceiro método do *Listener* é o alocado para parâmetros, sendo setado em um *HashMap*. Por último, as configurações finais, incluindo um tempo máximo de espera. A Figura 15 demonstra o processo detalhado acima, com o *request* e *response*.

Figura 15 – Exemplo Volley

```

public void update(final int id , final String msg, Context t){
    RequestQueue mRequestQueue = Volley.newRequestQueue(t);
    StringRequest request = new StringRequest(Request.Method.POST, urlJsonUpdateMessage,
        new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                send_ok();
            }
        }, new Response.ErrorListener(){
            @Override
            public void onErrorResponse(VolleyError error) {
                error();
            }
        })
    {
        protected Map<String, String> getParams() throws AuthFailureError {
            Map<String, String> parameters = new HashMap<String, String>();
            parameters.put("id", String.valueOf(id));
            parameters.put("texto", msg);
            return parameters;
        }
    };
    int socketTimeout = 30000; //30 seconds - change to what you want
    RetryPolicy policy = new DefaultRetryPolicy(socketTimeout, DefaultRetryPolicy.DEFAULT_MAX_
request.setRetryPolicy(policy);
    ApplicationController.getInstance().addToRequestQueue(request);
}

```

Fonte: o autor.

### 4.3 Testes

Os testes foram feitos de acordo com o desenvolvimento da aplicação. Realizados no final de cada história de usuário completada, sendo feito o teste funcional e o teste de caixa preta, que basicamente se resume em ver se a aplicação faz o que se propõe a fazer, se está fazendo o que os requisitos pedem que façam. Ao final de cada sprint, foi feito novamente o teste da versão atual da aplicação, levando sempre em consideração a validação dos requisitos e a facilidade de utilização das funcionalidades e do sistema em geral.

### 4.4 Implantação

Foi utilizado a plataforma do Mega (Mega.nz) para upload do aplicativo, apesar de utilizar o plano gratuito da mesma, a plataforma não deixa a desejar em relação a hospedagem de um arquivo leve como um aplicativo deste trabalho. O aplicativo está disponível online no link <<https://mega.nz/#!VVIBxAxQ!WGOGSxktis0zGQxc8nJBNtxNlrayR0x58zULIF5O2KI>> , para qualquer pessoa fazer o download de forma gratuita, para que possam utilizá-la e avaliá-la em termos usuais, de acordo com as métricas já discutidas.

## 5 AVALIAÇÃO

Esta seção mostra como foi realizada a avaliação do aplicativo GranMapey, contendo informações sobre a elaboração dos questionários, a interação dos usuários com a aplicação, a aplicação do questionário e a consolidação dos dados.

### 5.1 Elaboração dos questionários

Foi elaborado um questionário online com 10 questões. Incluindo perguntas básicas como nome, idade, curso/semestre e perguntas voltadas as métricas da usabilidade do aplicativo.

- Qual o nível de facilidade você teve para realizar as tarefas sugeridas para a avaliação?
- Quantas vezes não compreendeu ou teve problemas com a interface da aplicação?
- Quantas vezes você consultou a tela de ajuda para obter informações sobre o uso da aplicação?
- Quantos erros você encontrou durante a execução das tarefas?
- Para cada ação do usuário o aplicativo oferece feedback imediato e adequado sobre seu status?
- O App possui padrões que facilitam seu uso?
- Como qualificaria a usabilidade do app GranMapey?

### 5.2 Interação dos usuários com a aplicação

Após o questionário pronto e a aplicação já disponível, o questionário foi divulgado nas redes sociais e nos aplicativos de *chat* e comunicação (*Facebook, Slack, Whatsapp e Telegram*), sendo direcionado a um grande público para a avaliação. Após a divulgação, foi realizado uma interação entre usuários com a aplicação, tais usuários dispulseram para analisar e avaliar a aplicação, logo deve ser a primeira vez que os usuários tenha acesso ao aplicativo. Foram sugeridas algumas tarefas para o usuário realizar na aplicação, algumas tarefas simples e outras consideradas mais complexas, com o intuito de analisar se até mesmo as tarefas mais complexas estão intuitivas e na característica usável. A seguir, as tarefas sugeridas.

- Crie um ponto de interesse no mapa, com um novo tipo e próximo ao seu local atual.
- Encontre o ponto que você criou, com auxílio da pesquisa por tipos e altere a

localização do local onde você salvou.

- Pesquise pelo tipo "Banco" e insira uma mensagem.
- Edite, responda e apague a mensagem inserida no item anterior.
- Escolha um ponto do tipo "Restaurante" e veja se o mesmo está aberto no momento da pesquisa.
- Selecione o banco mais próximo a você e gere uma rota até ele.
- Veja o tutorial do aplicativo, com todas as instruções de uso.

### **5.3 Aplicação do questionário**

Logo após a interação do usuário com a aplicação, o mesmo respondeu honestamente o questionário online que visa a qualidade no quesito usabilidade da aplicação. O usuário preencheu online o simples questionário com as perguntas citadas nas seções acima de acordo com sua experiência com a aplicação. O questionário foi preenchido por 86 participantes, sejam eles profissionais da área de TI, como profissionais e estudantes de outras áreas. No início do formulário havia uma breve apresentação do propósito do mesmo, como também instruções e restrições para o prosseguir da avaliação. Como exemplo de instruções, temos as tarefas já citadas acima, e como restrições, a versão mínima do android para a aplicação.

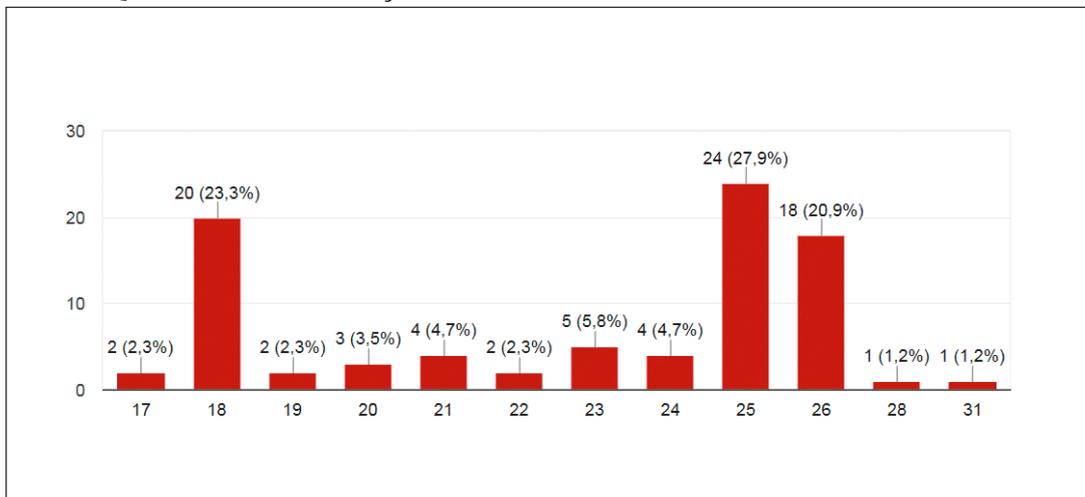
## 6 RESULTADOS

Após todas as amostras de questionário serem concluídas, analisamos os questionários. Essa análise foi feita principalmente em cima das questões de métricas de usabilidade, gerando uma média entre todos os questionários, formas de satisfações, e de certa forma notas para tal avaliação.

Como já mencionado, o questionário possui 10 perguntas, sendo 3 (três) de caráter identificativo e as outras 7 (sete) voltadas a usabilidade do aplicativo. A primeira pergunta foi referente ao nome, sendo opcional, caso o individuo não quisesse identificar-se. A segunda(idade) e a terceira(profissão) foram obrigatórias para tentar analisar e moldar um padrão de acordo com as respostas. Detalharemos a seguir apenas questões destinadas a identificação dos avaliadores.

A primeira pergunta obteve apenas 16 respostas, que foi a pergunta do nome, onde o individuo poderia ou não preencher. A segunda, sua idade, onde pode-se notar que a maior quantidade de avaliadores tinham entre 18 e 26 anos, de acordo com a figura abaixo, incluindo estudantes e profissionais.

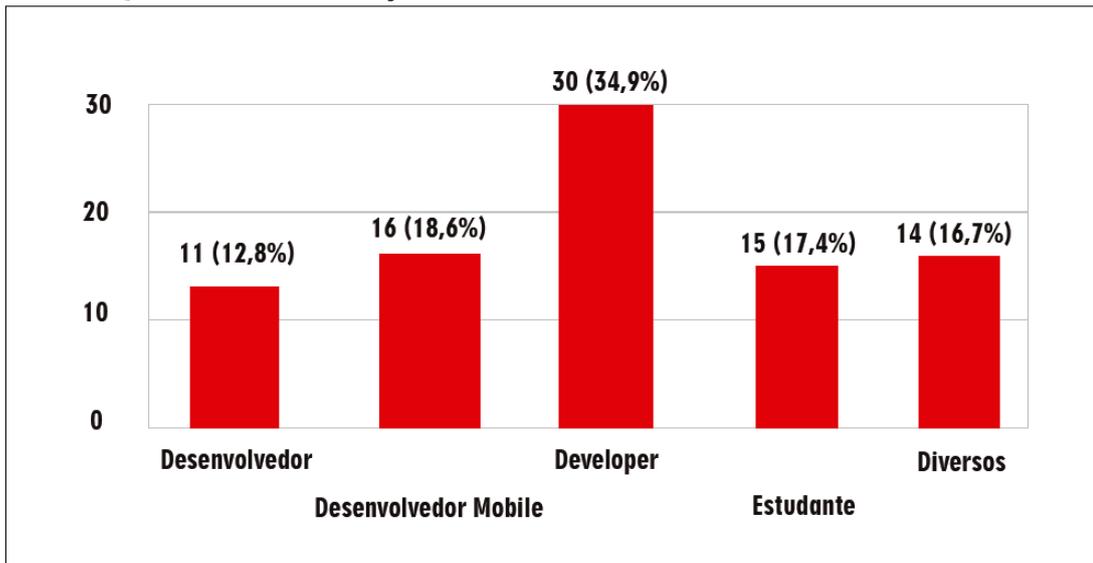
Figura 16 – Questão 2 - Identificação



Fonte: o autor.

A terceira pergunta foi a profissão do participante que resultou em quase todos os casos em um profissional da TI, acompanhando no gráfico abaixo, 30 foram denotados como "Developer", 16 como *Desenvolvedor Mobile*, 15 como *Estudante* e 11 como *Desenvolvedor*. Tais resultados são justificados pelos meios de divulgação do questionário de avaliação, como redes sociais e aplicativos de conversas.

Figura 17 – Questão 3 - Identificação

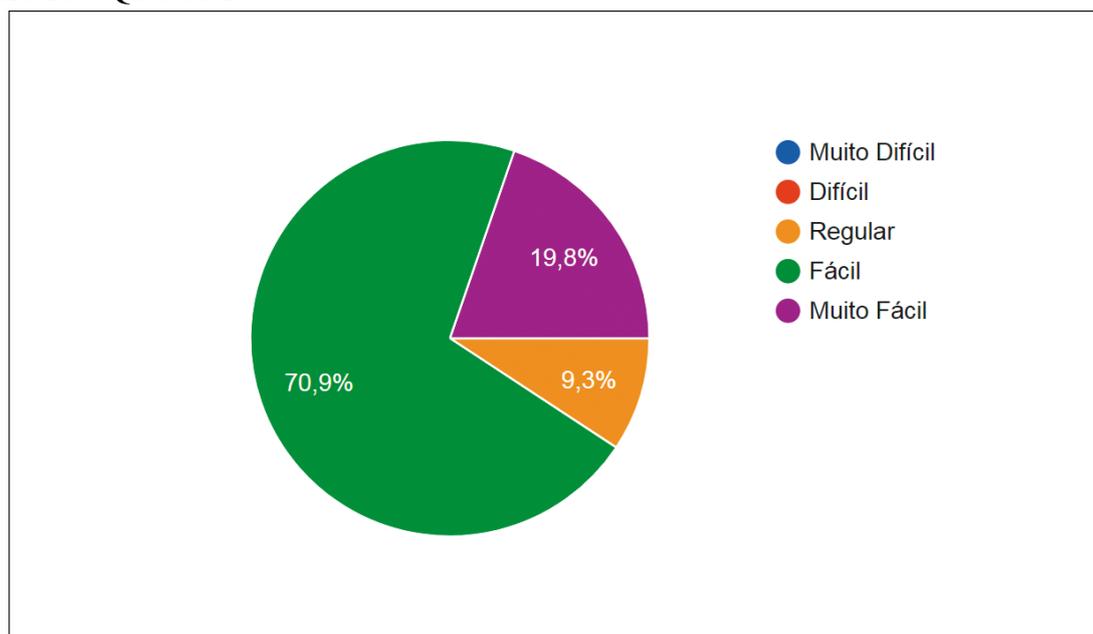


Fonte: o autor.

A seguir as questões destinadas a medição de usabilidade do aplicativo.

A primeira pergunta foi: "*Qual o nível de facilidade você teve para realizar as tarefas sugeridas para a avaliação?*", onde procuramos obter informações sobre o grau de dificuldade perante as tarefas variadas sugeridas no início do questionário e realizadas no decorrer do mesmo. O gráfico da questão 1, mostra o resultado em porcentagem para esta questão.

Figura 18 – Questão 1



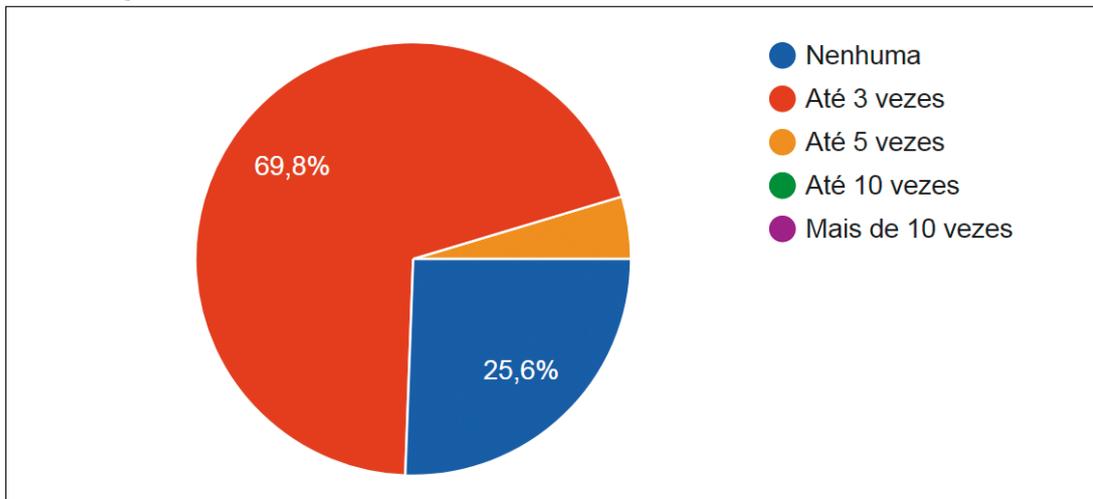
Fonte: o autor.

De acordo com o gráfico, obtivemos 70,9% da avaliação qualificada como *Fácil*,

19,8% como *Muito Fácil* e 9,3% como *Regular*. Logo, segundo os avaliadores, as tarefas foram tranquilamente executadas na aplicação desenvolvida neste trabalho.

A segunda pergunta foi: "*Quantas vezes não compreendeu ou teve problemas com a interface da aplicação?*". Esta pergunta tem o intuito de descobrir se a aplicação possui falhas que influenciam no mau entendimento das funcionalidades e da aplicação em geral. O gráfico da questão 2 mostra o resultado em porcentagem para esta questão.

Figura 19 – Questão 2

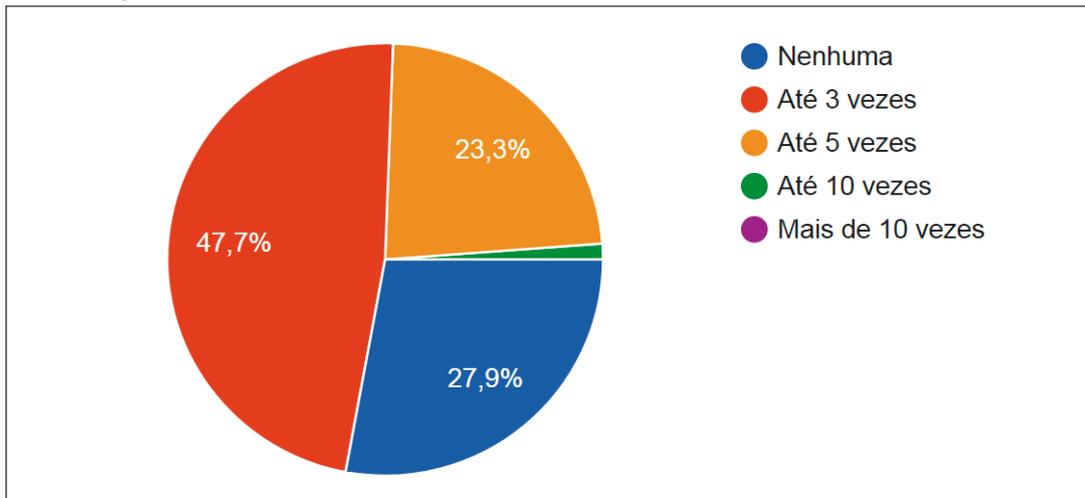


Fonte: o autor.

Segundo o gráfico acima, temos que 69,7% dos participantes avaliadores não compreenderam ou tiveram problemas com a interface da aplicação até 3 vezes, enquanto 4,7% não compreenderam até 5 vezes e os outros 25,6% não obtiveram complicações ou problemas perante a aplicação.

A terceira pergunta foi: "*Quantas vezes você consultou a tela de ajuda para obter informações sobre o uso da aplicação?*". Essa pergunta serviu para medir se os maus compreendimentos tinham soluções na própria aplicação, ou seja, se o usuário usaria de fato tal funcionalidade para aprender a utilizar o aplicativo. O gráfico da questão 3 mostra o resultado em porcentagem para esta questão.

Figura 20 – Questão 3

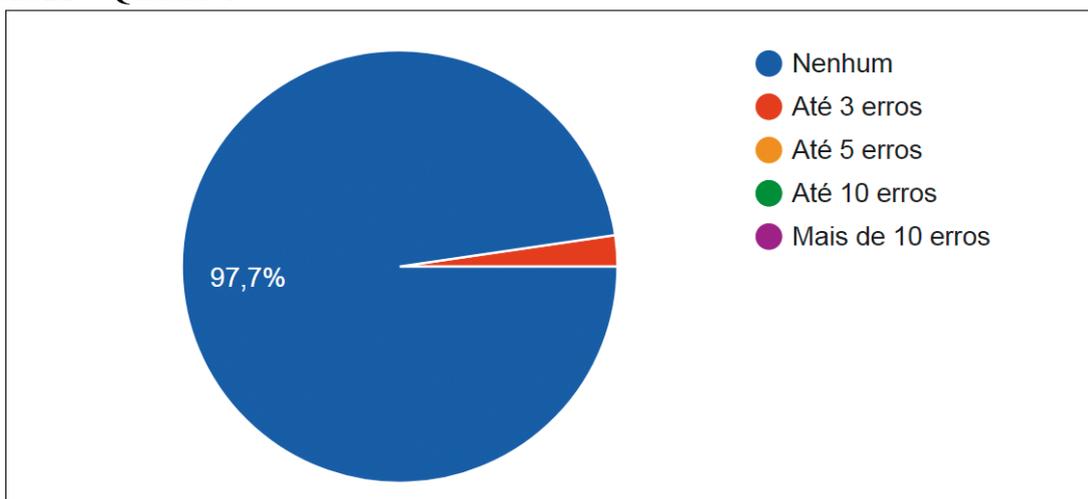


Fonte: o autor.

Com isso, temos 27,9% de participantes que não consultaram a ajuda, enquanto 47,7% consultaram até 3 vezes, 23,3% até 5 vezes e 1,2% consultaram até 10 vezes.

A quarta pergunta foi: "*Quantos erros você encontrou durante a execução das tarefas?*". Essa é uma medida que influenciará bastante no resultado final da usabilidade, mostrando se na atual versão da aplicação existem erros que dificultam o entedimento do usuário. O gráfico da questão 4 mostra o resultado em porcentagem para esta questão.

Figura 21 – Questão 4



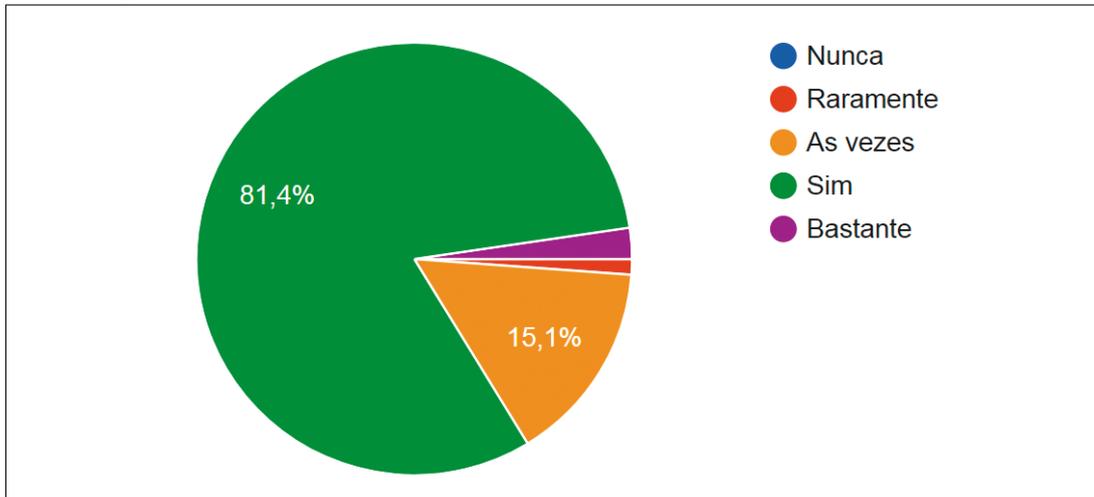
Fonte: o autor.

Contudo, 97,7% dos participantes não encontraram erros, enquanto apenas 2,3% encontraram até 3 erros na aplicação durante a execução das tarefas designadas no questionário.

A quinta pergunta foi: "*Para cada ação do usuário o aplicativo oferece feedback imediato e adequado sobre seu status?*". Um outro fator importante para uma boa usabilidade é

o feedback, e nessa questão é a medida deste fator em relação a todas as operações e funções da aplicação. O gráfico da questão 5 mostra o resultado em porcentagem para esta questão.

Figura 22 – Questão 5

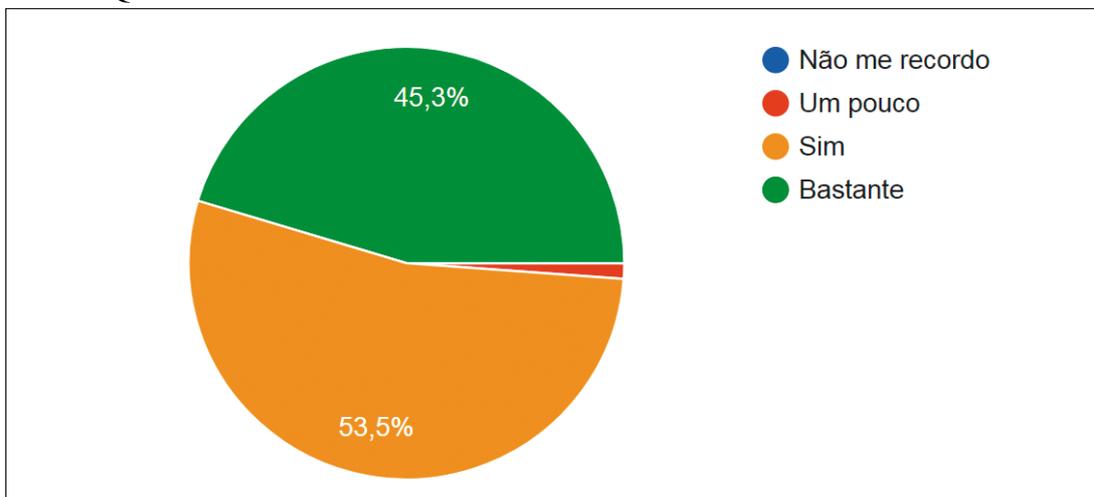


Fonte: o autor.

De acordo com o gráfico, 81,4% dos avaliadores concordaram que o aplicativo retorna informações a cada operação, 2,3% disseram que isso ocorre bastante, enquanto 15,1% que só ocorre as vezes e apenas 1,2% que acontece raramente.

A sexta pergunta foi: "*O App possui padrões que facilitam seu uso?*". Padronizar a interface pode tornar mais fácil a aprendizagem de uma aplicação, esta questão mede tais padrões e conseqüentemente a facilidade de aprendizagem. O gráfico da questão 6 mostra o resultado em porcentagem para esta questão.

Figura 23 – Questão 6

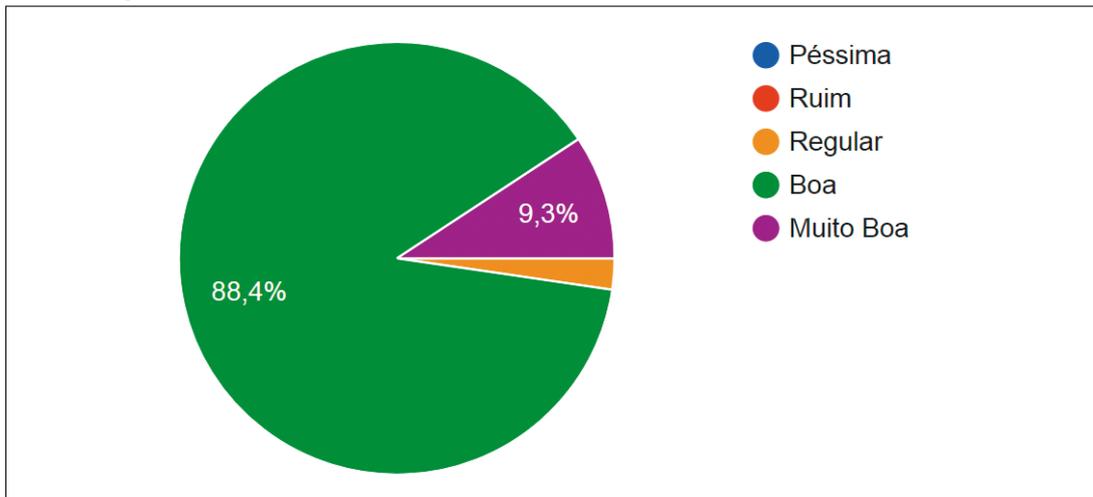


Fonte: o autor.

Seguindo o gráfico acima, 53,5% disseram Sim e concordaram que a aplicação avaliada possui padrões, 45,3% denotaram que existe muitos padrões, enquanto 1,2% viram poucos padrões no decorrer da execução das tarefas.

A sétima pergunta foi: "*Como qualificaria a usabilidade do app GranMapey?*". Esta pergunta serve para generalizar a usabilidade da aplicação, seguindo a experiência adquirida no decorrer do questionário. O gráfico da questão 7 mostra o resultado em porcentagem para esta questão.

Figura 24 – Questão 7



Fonte: o autor.

Como última questão, o gráfico representa a avaliação geral, com uma boa taxa de qualidade, 88,4% dos avaliadores qualificaram em termos de usabilidade a aplicação como Boa, 9,3% como Muito Boa e apenas 2,3% como Regular.

Um fato a ser observado, foi que, a taxa de falta de entendimento da interface foi bem equivalente à taxa de consultas a tela de ajuda, ou seja, pode-se concluir que mesmo em sistemas e aplicações mais robustas, que podem resultar em uma complexidade ao utilizar a aplicação, pode-se utilizar recursos como este, de ajuda, que auxiliam o usuário em tempo de execução da aplicação/sistema.

Os resultados foram satisfatórios, algumas respostas denotaram que a aplicação necessita de melhorias, porém em termos usável a aplicação não deixou a desejar para os avaliadores.

## 7 CONCLUSÕES

Muitos usuários de aplicações Android reclamam de suas interfaces, de seus estilos de letras, de falta de padrões e de informações. A cada dia que passa, é mais comum um usuário trocar de um aplicativo A para um aplicativo B similar, por motivos apenas de interface, ou seja, o visual influencia fortemente a escolha das pessoas, logo, a importância de um aplicativo de boa aparência é enorme.

O trabalho aqui descrito, após o estudo de aplicações existentes relacionadas ao uso de mapas e comunicação, se propôs a obter conhecimentos necessários e desenvolver um aplicativo, capaz de unir diversas funcionalidades de outros aplicativos, como inserir novas que seriam fundamentais para integrar em um todo. Tudo isso com o intuito de deixar mais fácil a vida do usuário, ou seja, achar o que deseja em um único app, e o mesmo ser fácil de usar.

Para atingir o objetivo do trabalho descrito, foi preciso dividi-lo em algumas partes, tais como, o estudo de aplicações similares, o levantamento de requisitos, a validação dos requisitos, a fase de projeto, a elaboração da aplicação, o teste da aplicação, a implantação, a elaboração do questionário, avaliação e os resultados.

Concluindo, o aplicativo desenvolvido neste trabalho, passou por algumas avaliações, e ao final, a maior porcentagem da avaliação geral de sua usabilidade foi referente a medição de "Boa", com isso, possivelmente pode gerar disputas com outros aplicativos hoje do mercado, pelo fato de estar com uma boa usabilidade e com funcionalidades agradáveis.

## REFERÊNCIAS

- ALVES, P.; PIRES, J. A. A usabilidade em software educativo: princípios e técnicas. In: **VI Congresso Iberoamericano, IV Simpósio Internacional de Informática Educativa, VII Taller Internacional de software Educativo, 2002**. [S.l.: s.n.], 2002.
- ANTUNES, R. **RouteMe-Uma Aplicação para Dispositivos Móveis Android para Resolução de Problemas de Rotas de Veículos**. [S.l.: s.n.], 2015.
- BARTIÉ, A. **Garantia da qualidade de software**. [S.l.]: Gulf Professional Publishing, 2002.
- JOHNSON. **Definição e classificação por tipo da Interface de Programação de Aplicativos (API)**. 2014. Disponível em: <<http://blog.magicsoftware.com.br/definicao-e-classificacao-por-tipo-da-interface-de-programacao-de-aplicativos-api/>>. Acesso em: 28 abr. 2017.
- KALAKOTA, R.; ROBINSON, M. **M-business Tecnologia Movel E Estrategia**. [S.l.]: Bookman, 2002.
- LANE, K. History of apis. **API Evangelist**, v. 20, 2012.
- LECHETA, R. R. **Google Android: aprenda a criar aplicações para dispositivos móveis com o Android SDK**. 3ª ed. [S.l.]: Novatec Editora, 2013.
- LIMA, V. *et al.* Ubibusroute: Um sistema de identificação e sugestão de rotas de ônibus baseado em informações de redes sociais. **Simpósio Brasileiro de Sistemas de Informação, São Paulo**, p. 516–527, 2012.
- MALAVASI. **As evoluções e maravilhas da integração através de APIs**. 2015. Disponível em: <<https://imasters.com.br/apis/as-evolucoes-e-maravilhas-da-integracao-atraves-de-apis/?trace=1519021197&source>>. Acesso em: 26 abr. 2017.
- MEDEIROS, M. A.; CYBIS, W. d. A. Método de avaliação de usabilidade de software a partir da satisfação de usuários e da aplicação de quesitos da norma iso 9241. In: **IHC'2000–III Workshop sobre Fatores Humanos em Sistemas de Computação**. [S.l.: s.n.], 2000. p. 93–101.
- MENDIOLA, J. Z. **Android já é o sistema operacional mais usado do mundo**. 2017. Disponível em: <[https://brasil.elpais.com/brasil/2017/04/04/tecnologia/1491296467\\_396232.html](https://brasil.elpais.com/brasil/2017/04/04/tecnologia/1491296467_396232.html)>. Acesso em: 29 jul. 2017.
- MIGUEL. **A evolução do Android desde o início**. 2017. Disponível em: <<https://vidroid.com.br/news/a-evolucao-do-android-desde-o-inicio>>. Acesso em: 25 abr. 2017.
- NIELSEN, J. **Usability 101: Introduction to usability**. [S.l.: s.n.], 2003.
- NOMISO, L. S. **A importância da usabilidade nas informações apresentadas em hipertexto**. [S.l.: s.n.], 2008.
- PELOI. **O que são APIs - Parte 2: Como uma api funciona**. 2016. Disponível em: <<http://sensedia.com/blog/apis/o-que-sao-apis-parte-2-como-uma-api-funciona/>>. Acesso em: 26 abr. 2017.
- PEREIRA, L. C. O.; SILVA, M. L. da. **Android para desenvolvedores**. [S.l.]: Brasport, 2009.

SØRENSEN, C. **Enterprise mobility**: tiny technology with global impact on work. [S.l.]: Springer, 2011.

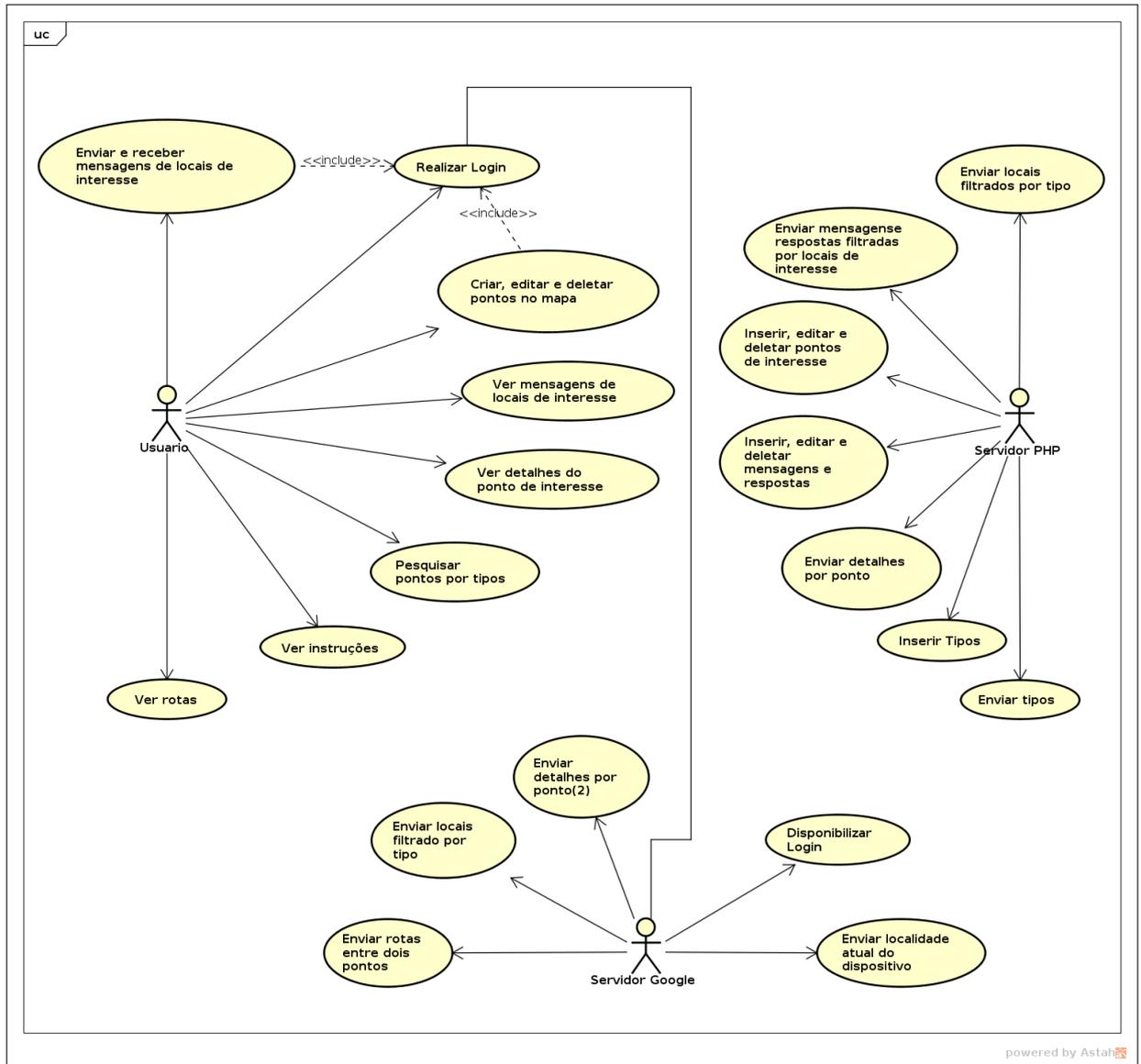
THOMAS, F.; FAVARETTO, R. M. Trocacomigo: Aplicativo de consumo colaborativo. **Anais SULCOMP**, v. 8, 2017.

TURBAN, E. *et al.* **Tecnologia da Informação para Gestão**: Transformando os negócios na economia digital. [S.l.]: Bookman, 2010.

## APÊNDICE A – REQUISITOS DA APLICAÇÃO

- R01 - Efetuar login pelo Google
- R02 - Pesquisar por tipos de pontos de interesse
- R03 - Inserir, deletar e editar um ponto no mapa
- R04 - Comunicar-se com outros usuários proprietários de pontos de interesse
- R05 - Visualizar rota entre meu ponto atual e o ponto interessado
- R06 - Visualizar o sistema de ajuda em qualquer tela da aplicação
- R07 - Visualizar detalhes de um ponto de interesse, como nome, tipo, telefone, horário de funcionamento e descrição

## APÊNDICE B – DIAGRAMA DE CASOS DE USO



Fonte: o autor.

## APÊNDICE C – FORMULÁRIO DE AVALIAÇÃO DO APLICATIVO

### Avaliando o App GranMapey

Esse formulário será para avaliar em termos de usabilidade o aplicativo denominado como GranMapey. Ao participar você estará ajudando e contribuindo para o meu TCC do curso de Sistemas de Informação na Universidade Federal do Ceará. Não é obrigado se identificar, apenas respondê-lo.

Baixe o aplicativo no link:

<https://mega.nz/#!VMBxAxQ!WGOGSxktis0zGQxc8nJBNbxNlrayR0x58zULIF5O2KI>

Instale em seu dispositivo (com android superior a 4.0.0) e execute as seguintes tarefas :

- 1 - Crie um ponto de interesse no mapa, com um novo tipo e próximo ao seu local atual.
- 2 - Encontre o ponto que você criou, com auxílio da pesquisa por tipos e altere a localização do local onde você salvou.
- 3 - Pesquise pelo tipo "Banco" e insira uma mensagem em um banco próximo a sua localização.
- 4 - Edite, responda e apague a mensagem inserida no item (3)
- 5 - Escolha um ponto do tipo "Restaurante" e veja se o mesmo está aberto no momento da pesquisa.
- 6 - Selecione o banco mais próximo a você e gere uma rota até ele.
- 7 - Veja o tutorial do aplicativo, com todas as instruções de uso.

Nome: \_\_\_\_\_ (Opcional)

Idade: \_\_\_\_\_

Profissão: \_\_\_\_\_

Qual o nível de facilidade você teve para realizar as tarefas sugeridas para a avaliação?

- Muito Difícil
- Difícil
- Regular
- Fácil
- Muito Fácil

Quantas vezes não compreendeu ou teve problemas com a interface da aplicação?

- Nenhuma
- Até 3 vezes
- Até 5 vezes
- Até 10 vezes
- Mais de 10 vezes

Quantas vezes você consultou a tela de ajuda para obter informações sobre o uso da aplicação?

- Nenhuma
- Até 3 vezes
- Até 5 vezes
- Até 10 vezes
- Mais de 10 vezes

Quantos erros você encontrou durante a execução das tarefas?

- Nenhum
- Até 3 erros
- Até 5 erros
- Até 10 erros
- Mais de 10 erros

Para cada ação do usuário o aplicativo oferece feedback imediato e adequado sobre seu status?

- Nunca
- Raramente
- Às vezes
- Sim
- Bastante

O App possui padrões que facilitam seu uso?

- Não me recordo
- Um pouco
- Sim
- Bastante

Como qualificaria a usabilidade do app GranMapey?

- Péssima
- Ruim
- Regular
- Boa
- Muito Boa

Fonte: o autor.