

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Adriano Lima de Sá

**Desenvolvimento de uma aplicação Android  
para compartilhamento de preço e qualidade de  
postos de combustível**

**Uberlândia, Brasil**

**2017**

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Adriano Lima de Sá

**Desenvolvimento de uma aplicação Android para  
compartilhamento de preço e qualidade de postos de  
combustível**

Trabalho de conclusão de curso apresentado  
à Faculdade de Computação da Universidade  
Federal de Uberlândia, Minas Gerais, como  
requisito exigido parcial à obtenção do grau  
de Bacharel em Sistemas de Informação.

Orientador: André Ricardo Backes

Universidade Federal de Uberlândia – UFU  
Faculdade de Ciência da Computação  
Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2017

# Resumo

Este trabalho apresenta o planejamento e o desenvolvimento de um sistema que conta com um servidor e um aplicativo que utiliza a plataforma para dispositivos móveis Android, o sistema operacional mais utilizado no mundo em tais dispositivos atualmente ([TECHNOLOGIES, 2017](#)). Esse sistema permitirá aos seus usuários trocar informações de forma colaborativa, sobre os preços e a qualidade dos postos de combustível em uma determinada cidade ou região. Com um grande público alvo, o sistema contará com a ajuda dos seus próprios usuários para manter os preços dos combustíveis atualizados, bem como para cadastrar novos postos, combustíveis, e para avaliar os postos contidos em sua base de dados. Os testes realizados tanto por usuários, com diferentes modelos de aparelhos e versões do sistema operacional, quanto por uma ferramenta disponibilizada pelo próprio Google, foram bem satisfatórios. Ainda cabem melhorias no sistema, as quais deverão ser implementadas futuramente.

**Palavras-chave:** Android, Colaborativo, Combustível, JAVA, Posto de Combustível.

# Lista de ilustrações

Figura 1 – Sistemas Operacionais de dispositivos móveis. . . . .	7
Figura 2 – Variação do preço da gasolina e do etanol ao longo dos anos de 2001 a 2014. . . . .	8
Figura 3 – Evolução da frota nacional de veículos entre os anos de 2003 a 2013. . . . .	9
Figura 4 – Pessoas de 10 anos ou mais de idade que tinham celular. . . . .	9
Figura 5 – Principais telas do Aplicativo Posto de Gasolina . . . . .	16
Figura 6 – Principais telas do Aplicativo Preço dos Combustíveis. . . . .	17
Figura 7 – Principais telas do Aplicativo Easy Gas. . . . .	19
Figura 8 – Casos de uso. . . . .	22
Figura 9 – Banco de dados MeuPosto . . . . .	23
Figura 10 – Telas iniciais do aplicativo. Da esquerda para direita:Home, Mapa, Favoritos. . . . .	25
Figura 11 – Tela de configurações. . . . .	26
Figura 12 – Pesquisa pelo nome de um dado posto. . . . .	27
Figura 13 – Tela com detalhes de um dado posto. . . . .	28
Figura 14 – Possíveis ações na tela de detalhes de um dado posto. O usuário pode realizar o cadastro de um novo combustível, bem como atualizar os preços de todos os combustíveis apresentados. Também pode realizar a avaliação do posto apresentado, e adicioná-lo ou removê-lo dos favoritos. . . . .	29
Figura 15 – Tela de cadastro de um novo posto. . . . .	30
Figura 16 – Tela com mensagem de falha na conexão com a internet, ou GPS desativado. . . . .	31
Figura 17 – Versões do Android utilizadas pelos usuários do teste. . . . .	35
Figura 18 – Respostas para: Ao executar alguma ação (mudar a ordenação dos postos na lista, cadastrar um posto, dentre outras) o aplicativo fecha inesperadamente? . . . . .	36
Figura 19 – Respostas sobre se os dados apresentados pelo aplicativo são suficientes. . . . .	36
Figura 20 – Como os usuários que testaram o aplicativo se conectaram a internet. . . . .	37
Figura 21 – Resultados para os erros detectados no aplicativo MeuPosto em sua versão 2.2 (APK 4). . . . .	37
Figura 22 – Resultados sobre as falhas de segurança detectadas no aplicativo MeuPosto em sua versão 2.2 (APK 4). . . . .	38
Figura 23 – Versões do Android utilizadas nos testes realizados pelo relatório de pré-lançamento do Google Play Console. . . . .	38

Figura 24 – Modelos de aparelhos utilizados nos testes realizados pelo relatório de pré-lançamento do Google Play Console. . . . . 39

# Lista de abreviaturas e siglas

ANP	Agência Nacional do Petróleo, Gás Natural e Biocombustíveis
APK	<i>Android Package File</i>
APP	Aplicativo Móvel
ART	<i>Android Runtime</i>
ASF	<i>Apache Software Foundation</i>
API	<i>Application Programming Interface</i>
DENATRAN	Departamento Nacional de Trânsito
XML	<i>Extensible Markup Language</i>
GNV	Gás Natural Veicular
PHP	<i>Hypertext Preprocessor</i>
IBGE	Instituto Brasileiro de Geografia e Estatística
IPEA	Instituto de Pesquisa Econômica Aplicada
IDE	<i>Integrated development environment</i>
JDK	<i>Java Development Kit</i>
JVM	<i>Java Virtual Machine</i>
OHA	<i>Open Hanset Alliance</i>
PNAD	Pesquisa Nacional por Amostra de Domicílios
SGBD	Sistema de Gerenciamento de Banco de Dados
SO	Sistema Operacional
SDK	<i>Software Development Kit</i>
SGML	<i>Standard Generalized Markup Language</i>
SQL	<i>Structured Query Language</i>

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>7</b>
1.1	Objetivos	10
1.2	Método	10
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>13</b>
2.1	Conceitos Adotados	13
2.2	Trabalhos Correlatos	14
<b>3</b>	<b>DESENVOLVIMENTO</b>	<b>20</b>
3.1	Tecnologias Utilizadas	20
3.2	Atividades Desenvolvidas	21
3.3	Estado Atual do Desenvolvimento e Trabalhos Futuros	31
3.4	Testes e Resultados	34
<b>4</b>	<b>CONCLUSÃO</b>	<b>40</b>
Conclusão		40
	<b>REFERÊNCIAS</b>	<b>41</b>

# 1 Introdução

Nos últimos anos, tem-se notado um crescimento no número de dispositivos móveis e da frota de veículos automotores do país. O mercado de desenvolvimento para tais aplicações está em expansão, uma vez que grande parte da população possui algum dispositivo móvel ou até vários deles, como apontam pesquisas do Instituto Brasileiro de Geografia e Estatística o IBGE (IBGE, 2016a). Devido ao grande número de usuários de smartphones (dispositivo móvel mais utilizado pela população) e ao crescente número de veículos, cria-se um mercado de aplicativos para esse público que precisa ser atendido. Em especial para plataforma Android, sistema operacional (SO) para dispositivos móveis, que é atualmente o SO mais utilizado por esses dispositivos no mundo (TECHNOLOGIES, 2017) (Figura 1), devido a sua capacidade de funcionar em diferentes dispositivos, como celulares, smartphones e tablets. Além de possibilitar que os desenvolvedores criem aplicativos usando a linguagem JAVA, totalmente compatíveis com os modelos que suportam determinada versão do Android.

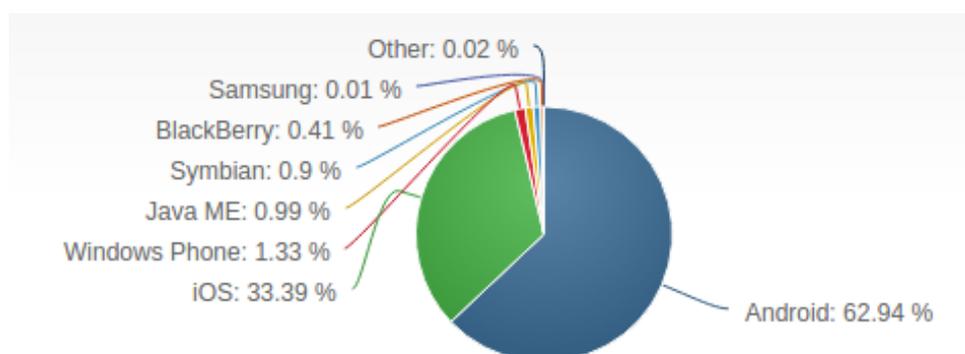


Figura 1 – Sistemas Operacionais de dispositivos móveis.

Fonte: <<http://www.netmarketshare.com/>> Acessado em:12/04/2017

Com a elevação do preço dos combustíveis, torna-se cada vez mais dispendiosa a tarefa de encontrar um revendedor que apresente valores condizentes com a expectativa do consumidor. Principalmente em grandes cidades como Uberlândia, que possui uma população de mais de 660.000 habitantes e devido a uma rotina que exige da população agilidade no cumprimento de suas tarefas. Realizar uma pesquisa para saber qual dentre os vários postos de combustível possui um preço mais convidativo e que ao mesmo tempo esteja próximo ao local onde o consumidor se encontra, atualmente tem se tornado uma tarefa de extrema dificuldade. Nesse sentido, se mostra importante o desenvolvimento de um aplicativo que faça todo o trabalho de pesquisa por preço, local e ordenação das informações apresentando-as para os consumidores.

Um estudo realizado por Souza e Pompermayer (SOUZA; POMPERMAYER, 2015) analisou o comportamento do consumidor brasileiro em relação aos movimentos dos preços do etanol hidratado e da gasolina nos principais estados brasileiros. A Figura 2, retirada do referido estudo, demonstra que há um aumento constante nos valores de ambos os combustíveis (etanol e gasolina) ao longo dos anos de 2001 a 2014, com oscilações pontuais (fonte dos dados ANP).

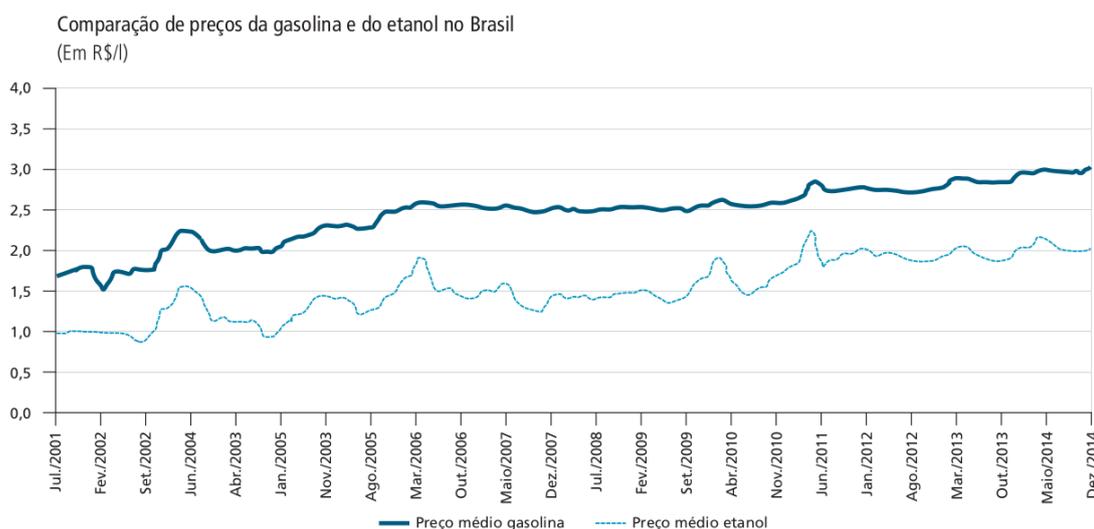


Figura 2 – Variação do preço da gasolina e do etanol ao longo dos anos de 2001 a 2014.

Fonte: Cópia do estudo original de Souza e Pompermayer.

Segundo estudo realizado no ano de 2012 pelo Instituto Brasileiro de Geografia e Estatística (IBGE) e divulgado pelo Instituto de Pesquisa Econômica Aplicada (Ipea) (IPEA, 2016), mais da metade dos domicílios brasileiros (54%) têm carro ou motocicleta para o deslocamento dos moradores. O Departamento Nacional de Trânsito (DENATRAN) (DENATRAN, 2016) aponta que em dezembro de 2015 o país já contava com mais de 90 milhões de veículos. Isso representa aproximadamente 45% da população brasileira, que segundo o IBGE (IBGE, 2016b) o Brasil possui mais de 200 milhões de habitantes atualmente. A Figura 3 apresenta a evolução da frota nacional de veículos ao longo dos anos de 2003 a 2013.

Dados suplementares da Pesquisa Nacional por Amostra de Domicílios (PNAD) em 2013 e liberados pelo IBGE no ano de 2015 (IBGE, 2016a), aponta que três quartos da população brasileira acima de 10 anos têm telefone móvel. Ao todo, o levantamento contabilizou 130,2 milhões (75,2%) de pessoas que possuem um celular, o que pode ser observado na Figura 4.

Mesmo que tivesse havido uma diminuição no crescimento da frota ou do número de smartphones, percebe-se que existe um mercado enorme a ser explorado. Assim, seria de grande valia para a população disponibilizar de forma livre e de fácil acesso os valores dos combustíveis referentes aos postos mais próximos dos consumidores. Segundo dados

### Frota de todos os tipos cresce...

País ultrapassou a barreira dos 80 milhões de veículos em setembro de 2013. Todas as categorias cresceram nos últimos dez anos.

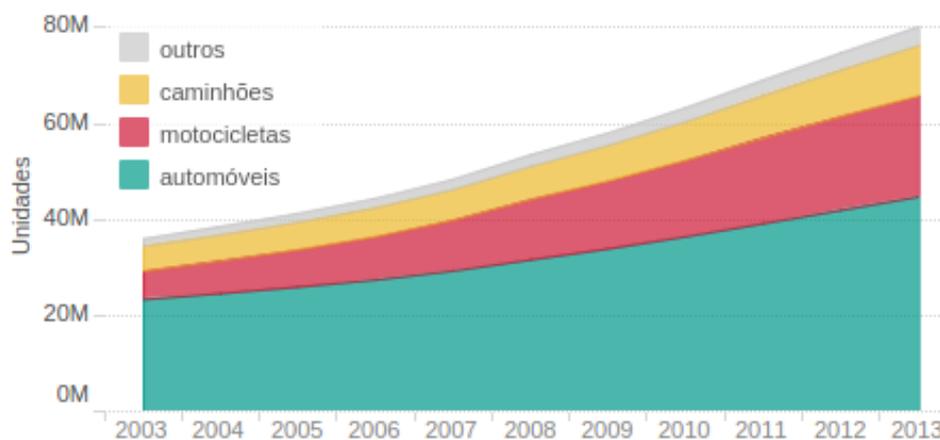


Figura 3 – Evolução da frota nacional de veículos entre os anos de 2003 a 2013.

Fonte: <<http://infograficos.oglobo.globo.com>> Acessado em:26/06/2017

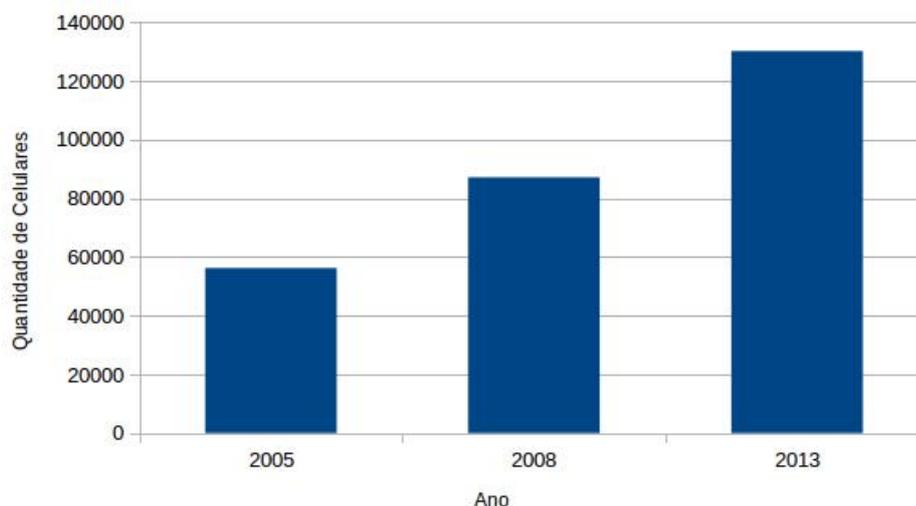


Figura 4 – Pessoas de 10 anos ou mais de idade que tinham celular.

Fonte: <<http://biblioteca.ibge.gov.br>> Acessado em:07/04/2016

da Agência Nacional do Petróleo, Gás Natural e Biocombustíveis (ANP) para o ano de 2015 (ANP, 2016) são mais de 39.000 postos no país. Além da falta de tempo, a grande quantidade desses estabelecimentos comerciais dificulta uma rápida e eficaz coleta de dados referente ao valor dos combustíveis em tais estabelecimentos, fazendo com que uma parte da população de uma cidade, ou até mesmo de um bairro, gaste mais ou menos dinheiro para abastecer seu veículo, uma vez que esses consumidores não estão totalmente cientes dos valores dos combustíveis em sua região. Tal problema seria facilmente resolvido com um aplicativo, que apresentaria na tela do smartphone de todos seus usuários o menor preço de combustível de forma rápida e fácil.

Com a criação do aplicativo espera-se que os motoristas e motociclistas possam diminuir ao máximo seus gastos com combustível, uma vez que estarão cientes de quais postos apresentam o preço mais convidativo aos mesmos e estimular o aumento da concorrência entre os postos de combustível para manter ou aumentar sua gama de clientes.

## 1.1 Objetivos

Neste trabalho, pretende-se utilizar a plataforma Android para o desenvolvimento de um aplicativo nomeado de MeuPosto<sup>1</sup> que permita aos usuários trocar informações, de forma colaborativa, sobre preços e promoções em postos de combustível em uma determinada cidade ou região, além de informações sobre a qualidade dos postos. Nesse sentido, este aplicativo permite formar uma rede colaborativa com dupla função: a de prover uma economia de custos para seus usuários e aumentar a fiscalização dos postos de combustível.

Bem como a utilização da linguagem PHP para o desenvolvimento de um *web service*, o qual ficou responsável por receber as solicitações dos dispositivos móveis e respondê-las com os dados contidos no banco de dados. Esse banco de dados (BD) está alocado em um servidor e é gerenciado pelo MySQL.

## 1.2 Método

Com um grande público alvo, o aplicativo contará com a ajuda dos seus usuários para manter o preços dos combustíveis atualizados, bem como para cadastrar novos postos e combustíveis a sua base de dados, sendo desta forma um aplicativo colaborativo. Assim, os próprios usuários irão garantir a manutenção do aplicativo, atualizando o valor dos combustíveis dos postos que frequentam, bem como avaliando-os. Pretende-se disponibilizar as seguintes funcionalidades aos usuários:

- **Definir o raio de distância máxima entre a atual localização do usuário e os postos de combustível a serem listados:** O aplicativo permitirá que o usuário defina qual é a distância máxima que deseja entre a sua localização atual, a qual será obtida por meio do GPS em seu dispositivo, e os postos de combustível da região onde ele se encontra.
- **Definir o modo de ordenação para apresentação da lista dos postos:** O usuário poderá definir qual será o critério de ordenação para a lista de postos apresentada, podendo ordenar os postos por menor preço praticado ou distância.

<sup>1</sup> MeuPosto, <<https://play.google.com/store/apps/details?id=com.adriano.meuPosto&hl=pt-BR>>

- **Definir o combustível base:** O usuário poderá definir qual é seu combustível mais consumido. Essa informação será utilizada para a apresentação da lista dos postos de combustível.
- **Apresentar a lista com os postos mais próximos ao usuário:** É a principal funcionalidade do aplicativo. Permite a visualização de todos os postos cadastrados na base de dados mais próximos da atual localização do usuário (com distância máxima definida por ele), na ordenação de sua preferência (menor preço ou menor distância). Na lista serão apresentadas informações como: bandeira, endereço e a distância entre cada posto apresentado na lista e a atual localização do usuário, bem como o valor do combustível base.
- **Apresentar os detalhes sobre os postos de combustível:** Ao selecionar um posto apresentado na lista, o usuário poderá visualizar detalhes referentes a esse posto de combustível, tais como: valor de todos os combustíveis com os quais trabalha, endereço, telefone, avaliações e bandeira.
- **Apresentar a localização no mapa:** Ao visualizar os detalhes de um posto de combustível, o usuário poderá utilizar a função mapa. Essa função permitirá ao usuário visualizar no mapa a localização do posto selecionado.
- **Atualizar o valor dos combustíveis:** Ao visitar um posto de combustível e verificar que o preço apresentado no aplicativo não confere com o praticado pelo posto, o usuário poderá atualizar o valor do combustível no aplicativo, repassando assim tal informação aos demais usuários.
- **Avaliar os postos de combustível:** Durante ou após sua visita aos postos de combustível, o usuário poderá avaliar a qualidade de tais estabelecimentos.
- **Cadastrar os postos favoritos:** O aplicativo permitirá que os usuários selecionem quais são seus postos favoritos, possibilitando dessa forma uma rápida consulta aos preços praticados por esses estabelecimentos.
- **Cadastrar novos postos de combustível:** O aplicativo permitirá que os usuários cadastrem novos postos de combustível à base de dados.

Para o desenvolvimento desse projeto foi utilizada a IDE (*Integrated development environment*, em português Ambiente de Desenvolvimento Integrado) oficial para plataforma Android, a Android Studio (DEVELOPERS, 2016) na versão 2.3 para o sistema operacional Windows. Essa IDE além de possuir um inteligente editor de códigos, possui emuladores para se testar os aplicativos desenvolvidos, tornando viável o teste simulado em vários modelos de aparelhos.

O Andoid Studio já possui integrado a ele o *Software Development Kit* (SDK, em português Kit de Desenvolvimento de Software) (DEVELOPERS, 2016). Esse kit inclui documentação, código e utilitários para que os programadores consigam desenvolver suas aplicações de acordo com o padrão de desenvolvimento para o Android. Tanto o SDK quanto a IDE são desenvolvidas pelo Google.

Para compilação dos códigos, uma vez que o Android utiliza a linguagem de programação orientada a objetos JAVA, também é necessário o JAVA Development Kit (JDK), utilizado na versão 8.

## 2 Referencial Teórico

Neste capítulo serão apresentados os principais conceitos a serem adotados neste trabalho, bem como os trabalhos correlatos. Inicialmente apresenta-se o sistema operacional para dispositivos móveis no qual o trabalho será desenvolvido, em seguida as linguagens de programação que serão utilizadas, bem como o banco de dados que será adotado para realizar a persistência dos dados. Na seção seguinte, são apresentados aplicativos já existentes no mercado, os quais apresentam um serviço semelhante ao pretendido por este projeto.

### 2.1 Conceitos Adotados

O sistema operacional (SO) Android foi desenvolvido pela empresa Google em parceria com grandes empresas do mercado de mobilidade, como fabricantes de celulares e operadoras. Esse grupo que ajuda no desenvolvimento de tal plataforma e é liderado pelo Google chama-se *Open Handset Alliance* (OHA), e conta com empresas como: Intel, Samsung, LG, Motorola, Sony, Ericsson, HTC, Sprint Nextel, ASUS, Acer, Dell e Garmin (OHA, 2016). Esse grupo tem dois principais objetivos: criar uma plataforma única e aberta para dispositivos móveis para deixar os consumidores mais satisfeitos com o produto final, e criar uma plataforma moderna e flexível para o desenvolvimento de aplicações corporativas. O resultado dessa união gerou o sistema operacional Android (LECHETA, 2009), no qual este projeto será desenvolvido.

A grande vantagem do Android é o de ser uma plataforma livre e de código aberto (*open source*) (GOOGLE, 2016). Sua licença *Apache Software Foundation* (ASF) (APACHE, 2016) é flexível e permite que cada fabricante possa realizar alterações no código-fonte para customizar seus produtos sem a necessidade de compartilhar tais alterações. O Android também é gratuito, dessa forma os fabricantes podem utilizá-lo para fazer as alterações que desejarem sem custo adicional. Por se tratar de software *open source*, o Android conta com contribuições de desenvolvedores do mundo inteiro para seu código-fonte, seja para adicionarem novas funcionalidades ou até mesmo corrigir falhas encontradas (LECHETA, 2009).

O Android é baseado no Kernel do Linux (LINUX, 2016), que é responsável por gerenciar a memória, os processos, as *threads*, a segurança dos arquivos e pastas, além de redes e *drivers*. Cada aplicativo do Android dispara um novo processo no sistema operacional. Alguns deles podem exibir uma tela para o usuário, enquanto outros podem ficar em segundo plano. Diversos processos e aplicações podem ser executados simultaneamente, e o Kernel do sistema operacional realizará todo o controle da memória. Devido

à esse controle, caso seja necessário o próprio SO poderá encerrar algum processo para liberar memória e recursos, e talvez reiniciar o mesmo processo posteriormente quando a quantidade de memória livre assim permitir (LECHETA, 2009).

Toda segurança do Android é baseada na segurança do Linux. No Android, cada aplicação é executada em um único processo e cada processo, por sua vez, contém uma *thread* dedicada. Para cada aplicação instalada no dispositivo é criado um usuário no sistema operacional, tal usuário terá total acesso a estrutura de diretórios dessa aplicação. Assim, nenhum outro usuário pode ter acesso a ela (LECHETA, 2009).

Para o desenvolvimento de aplicações para o Android, utiliza-se a linguagem de programação JAVA. Porém, o Android não possui a máquina virtual do JAVA, a *JAVA Virtual Machine* (JVM) (ORACLE, 2016a), para rodar suas aplicações, mas sim uma máquina virtual que é otimizada para dispositivos móveis. Até a sua versão 4.4 (KitKat) o Android possuía uma máquina virtual chamada Dalvik. A partir da versão 5.0 (Lollipop) a Dalvik foi substituída pela ART (*Android Runtime*). Ao desenvolver uma aplicação em JAVA, os códigos gerados passam para uma forma intermediária de código denominada *bytecode* após a compilação, que é interpretada pelas JVMs (arquivos com extensão .class). Porém, na compilação no Android, o *bytecode* é convertido para um formato intermediário com extensão .dex. Posteriormente os arquivos .dex e outros recursos como imagens são compactados em um único arquivo com a extensão .apk (*Android Package File*), que representa a aplicação final, a qual pode ser distribuída e instalada nos dispositivos com o sistema operacional Android (LECHETA, 2009).

Para o desenvolvimento do *web service* foi utilizada a linguagem PHP (*Hypertext Preprocessor*) (GROUP, 2016). E para o controle do banco de dados (BD) que se encontra em um servidor, foi utilizado o MySQL que é um sistema de gerenciamento de banco de dados (SGBD) (ORACLE, 2016b), que utiliza a linguagem SQL (*Structured Query Language*, em português Linguagem de consulta estruturada) (WOZ, 2016) como interface.

## 2.2 Trabalhos Correlatos

Na loja de aplicativos do Google, a Google Play <sup>1</sup>, pode-se encontrar alguns aplicativos que possuem o mesmo objetivo deste projeto, dentre eles: Posto de gasolina (LABS, 2016), Preço dos Combustíveis (COMBUSTÍVEIS, 2016) e Easy Gas (APP, 2016). Os três aplicativos são colaborativos, portanto contam com a sua base de usuários para dar manutenção nos postos e preços dos combustíveis, e também são gratuitos.

O aplicativo Posto de Gasolina, Figura 5, oferece a seus usuários a possibilidade de verificarem o menor preço dos combustíveis em todo o Brasil. Essa consulta pode

<sup>1</sup> Google Play <<http://play.google.com/store>>

ser realizada utilizando-se a atual localização do usuário ou digitando um endereço. O aplicativo possibilita a visualização dos preços dos combustíveis no mapa, dessa forma os usuários podem comparar os preços dos combustíveis de acordo com a proximidade entre os postos. Estão disponíveis para consulta os preços da Gasolina, Gasolina Aditivada, Gasolina Premium, Etanol, Diesel, Diesel S10 e GNV (Gás Natural Veicular). Os dados são fornecidos pela ANP (Agência Nacional do Petróleo, Gás Natural e Biocombustíveis) e podem ser atualizados pelos próprios usuários do aplicativo que devem se conectar usando uma conta do Facebook. A atualização de preços anônima não é permitida no aplicativo, e se utiliza a rede social para garantir a identidade real do usuário que atualiza o preço. Os usuários também podem colaborar cadastrando novos postos e depois atualizando os preços dos seus combustíveis.

Para melhorar a identificação dos dados referentes aos preços dos combustíveis, tais dados são separados por cores. Essas cores apresentadas no mapa indicam quais são os combustíveis com menor preço representados pela cor verde, com valor intermediário representados pela cor amarela e com alto preço representados pela cor vermelha. Os preços apresentados na cor cinza indicam que não existem informações relativas ao preço para o tipo de combustível procurado. Preços em marrom correspondem a valores que ultrapassaram o limite de tempo de atualização estabelecido pelo usuário. O aplicativo também exibe os postos na forma de uma lista, a qual pode ser ordenada por preço do combustível, distância do posto ou avaliação.

O aplicativo Preço dos Combustíveis, Figura 6, assim como o APP (Aplicativo Móvel) Posto de Gasolina possibilita que seus usuários consultem o preço praticado nos postos de todo o Brasil. Essa consulta pode ser realizada através da localização atual do usuário ou navegando-se no mapa, que exibe os postos mais próximos e com os preços mais competitivos. Além do mapa, os preços também podem ser acessados através de uma lista dos postos de combustível. O aplicativo apresenta os valores de Gasolina, Etanol e Diesel. O aplicativo apresenta as seguintes funcionalidades: Lista de postos por preço ou distância; Mapa com postos e preços; Preços de Gasolina, Etanol e Diesel; Busca pelo endereço ou localização atual; Informações dos postos como endereço, preços e bandeira; Rota para o posto; Compartilhar postos/preços em redes sociais (Facebook e Twitter).

O aplicativo Easy Gas, Figura 7, assim como os demais aplicativos, possibilita que seus usuários consultem o preço praticado pelos postos de combustível de todo o Brasil. Esse aplicativo apresenta como diferencial, frente aos demais aplicativos supra citados, oferecer recompensas aos seus usuários que colaboram com a manutenção do aplicativo, dando pontos aos mesmos. Esses pontos posteriormente podem ser trocados por rifas, com as quais o usuário concorre a prêmios de R\$50,00. O aplicativo apresenta as seguintes funcionalidades: Busca por diferentes combustíveis, incluindo Gasolina Comum, Gasolina Aditivada, Gasolina Premium, Diesel, Etanol e GNV; Classificação dos postos por ban-

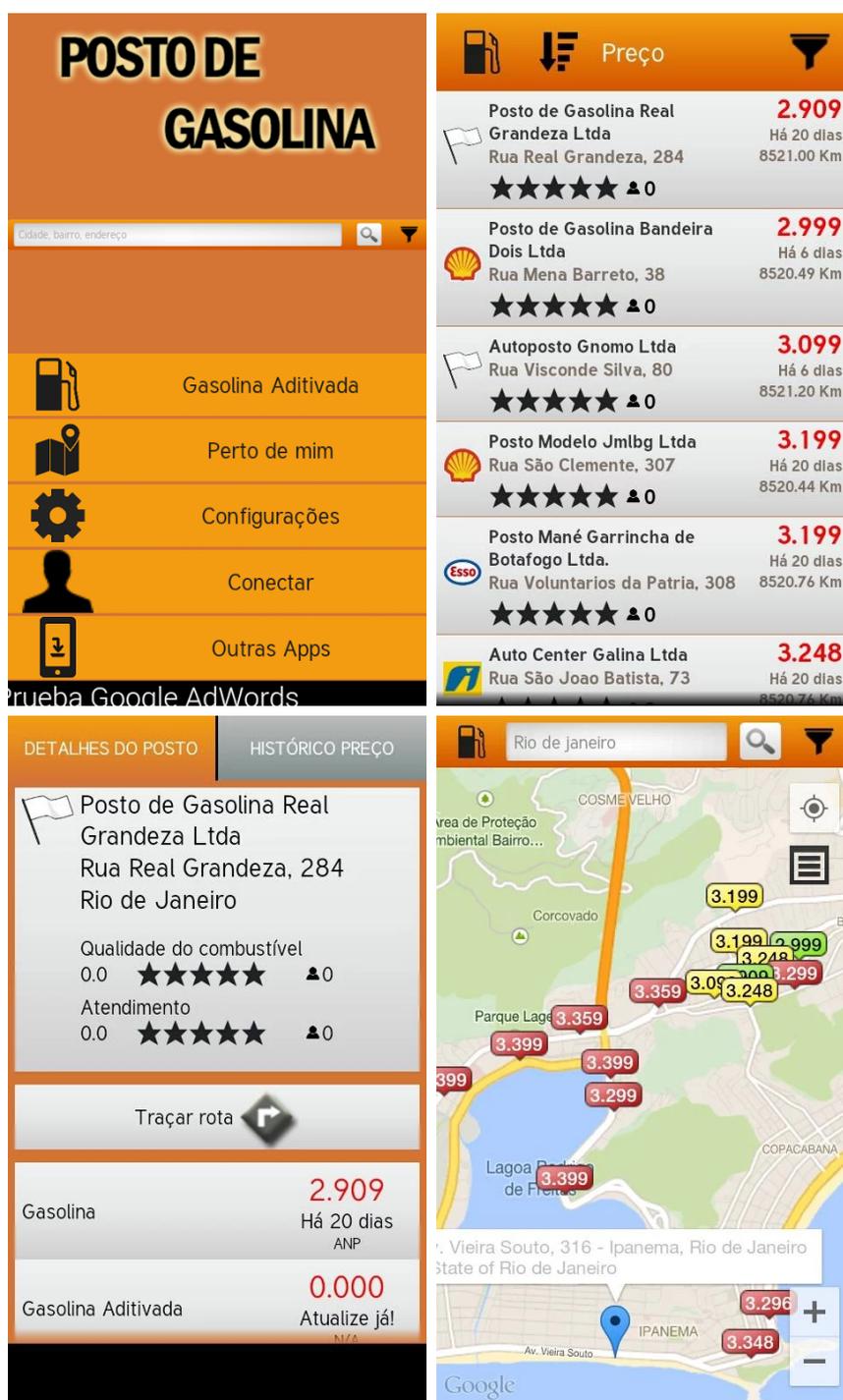


Figura 5 – Principais telas do Aplicativo Posto de Gasolina

Fonte: <<https://play.google.com/store/apps>> Acessado em:12/05/2016

deira; Busca por distância, preço ou valores atualizados mais recentemente; Adicionar postos de gasolina a base de dados; Editar valores dos combustíveis e dados dos postos de combustível; Navegação via GPS até o posto; Adicionar postos aos favoritos; Avaliar os postos combustível; Sistema de recompensas.

Esses aplicativos apresentam alguns pontos negativos que pretendemos resolver

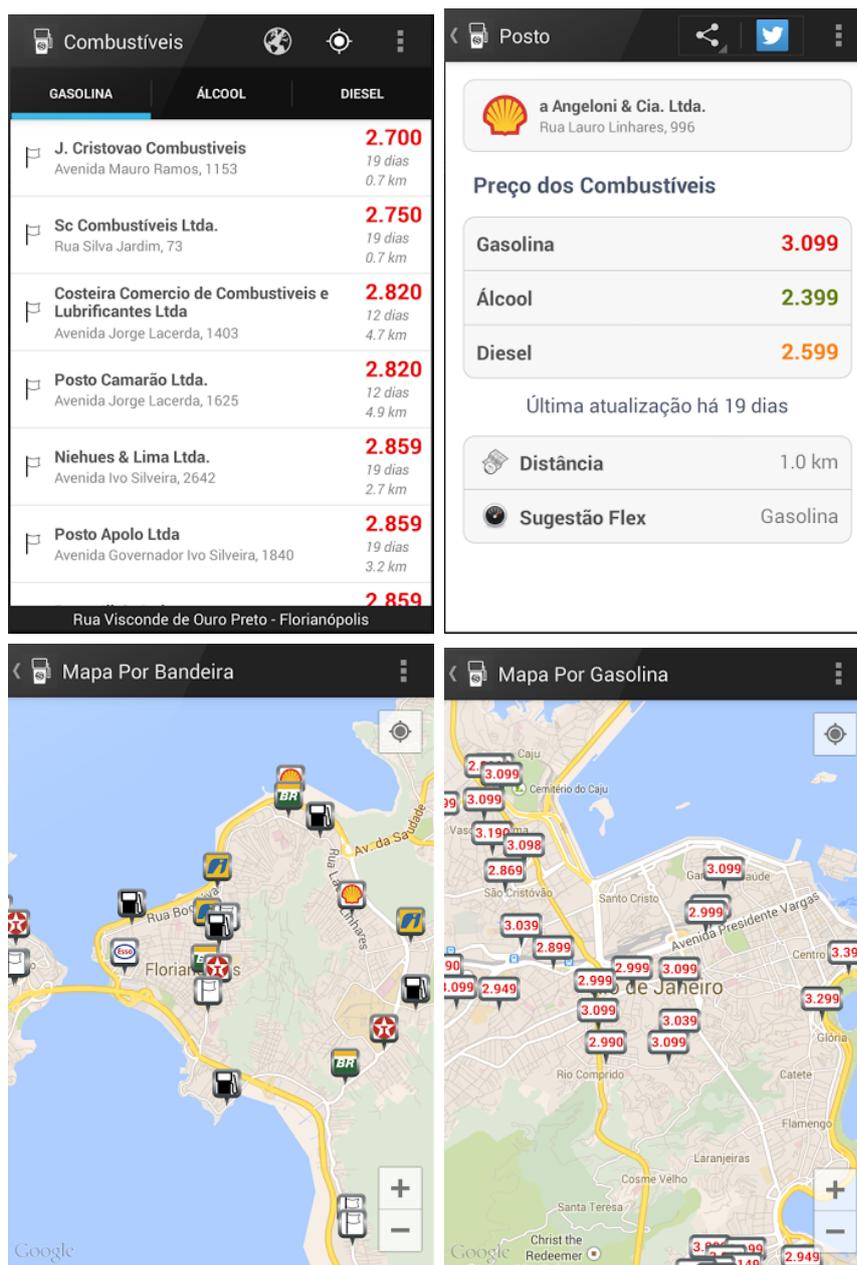


Figura 6 – Principais telas do Aplicativo Preço dos Combustíveis.

Fonte: <<https://play.google.com/store/apps>> Acessado em:12/05/2016

com nossa aplicação. Os três aplicativos não seguem os novos padrões de design fornecidos pelo Google, o Material Design (GOOGLE, 2017b). Isso torna os *layouts* apresentados antiquados ou até inadequados para aplicações mais recentes. Além disso, o aplicativo Posto de Gasolina, por exemplo, não permite que seus usuários façam uma busca pelo nome do posto, o que permitiria a visualização do(s) valor(es) do(s) combustível(is) de um posto que eles já conheçam. A escolha do combustível para a apresentação de seus valores na lista, não é intuitiva dando a entender que apenas é possível verificar os valores da gasolina. Em suas configurações não é possível determinar qual é a distância máxima entre a atual localização do usuário e dos postos apresentados. E para que o usuário possa

realizar a atualização do valor de um combustível apresentado é necessário que o mesmo faça login utilizando sua conta do Facebook.

O aplicativo Preço dos Combustíveis não permite que o usuário faça qualquer configuração. Dessa forma, assim como o aplicativo Posto de Gasolina, não é possível determinar a distância máxima entre o usuário e os postos apresentados no aplicativo. Outro ponto negativo é que são exibidos os preços apenas da gasolina, do etanol e do diesel. Combustíveis como o GNV ficaram de fora. Esse aplicativo não permite que os usuários façam a atualização do preço dos combustíveis apresentados, utilizando apenas como fonte os dados fornecidos pela ANP.

O aplicativo Easy Gas, assim como o aplicativo Posto de Gasolina, apresenta uma forma confusa para definição do combustível sobre o qual serão apresentados os valores na lista com os postos mais próximos do usuário. Assim como os demais aplicativos, também não é possível definir a distância desejada entre o usuário e os postos apresentados. Na lista exibida pelo aplicativo não é apresentada a informação sobre qual combustível se refere os valores apresentados, tornando a experiência com o aplicativo confusa. Algumas vezes o carregamento dessa lista é mais demorado do que outros aplicativos citados. Para que o usuário possa realizar o cadastro de seus postos favoritos, ou editar as informações de um posto apresentado, é necessário que ele faça um cadastro (onde deverá informar um usuário, e-mail e senha) ou login (usuário e senha) caso já tenha se cadastrado. Ou seja, o usuário terá que gastar um pouco mais de tempo, além de ter que fornecer uma informação pessoal.

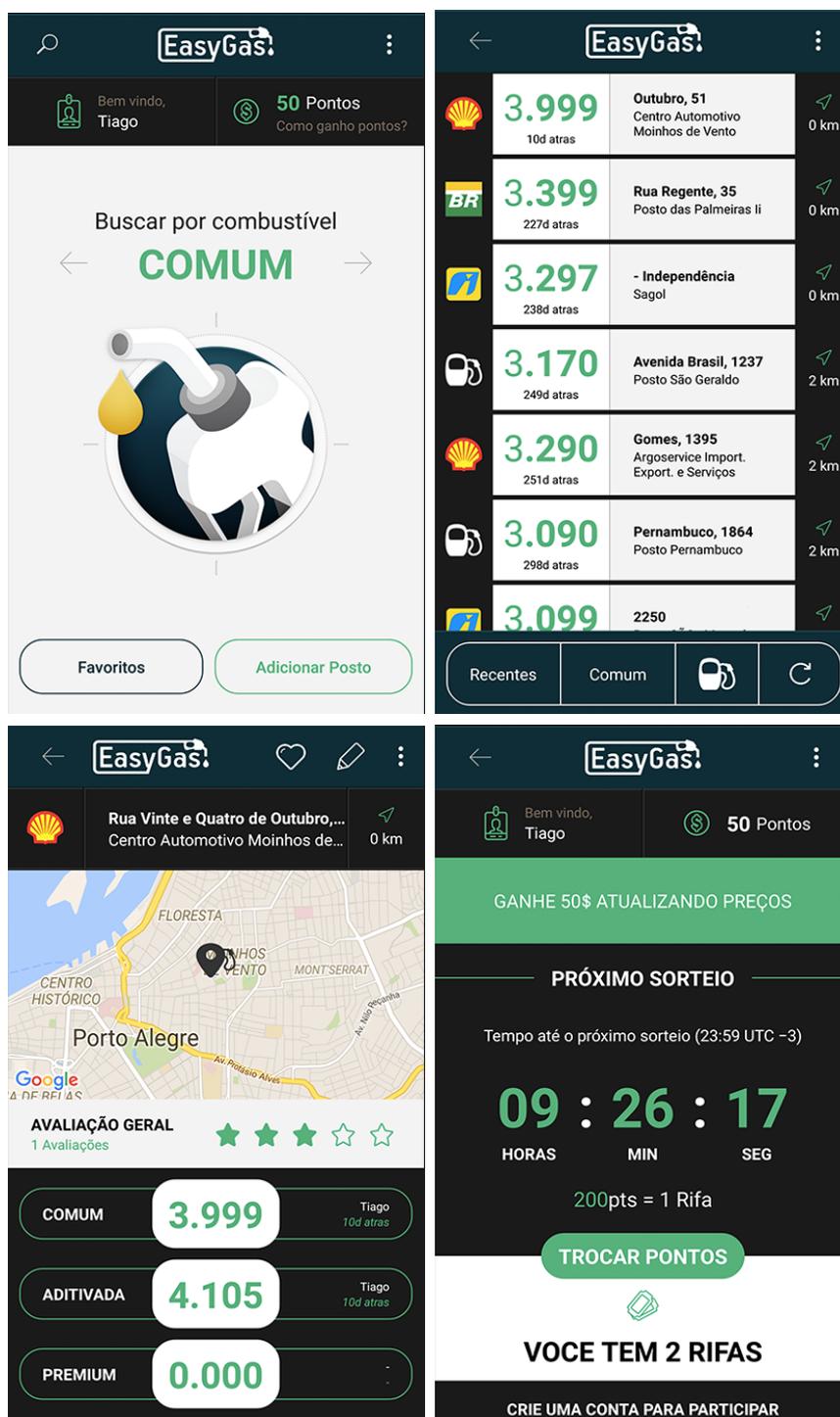


Figura 7 – Principais telas do Aplicativo Easy Gas.

Fonte: <<https://play.google.com/store/apps>> Acessado em:12/05/2016

## 3 Desenvolvimento

Neste capítulo será apresentado a evolução do desenvolvimento deste trabalho. Inicialmente apresentam-se as tecnologias utilizadas no desenvolvimento das aplicações do projeto. Em seguida, são apresentadas as atividades realizadas durante o desenvolvimento deste projeto, desde atividades de planejamento e visão como a criação de diagramas de Casos de Uso até a criação e implementação de telas/layouts e algoritmos. Em seguida, apresenta-se o estado atual do desenvolvimento, com as funcionalidades pretendidas e as de fato concretizadas, e quais são as funcionalidades que ainda se pretende desenvolver futuramente. Por fim apresentam-se os testes realizados e quais foram os resultados obtidos com os mesmos.

### 3.1 Tecnologias Utilizadas

Para o desenvolvimento deste projeto, no que se refere ao aplicativo, utilizou-se o Andoid Studio ([DEVELOPERS, 2016](#)) na versão 2.3 para o sistema operacional Windows. Essa IDE é baseada no IntelliJ IDEA ([JETBRAINS, 2017](#)), uma IDE já conhecida para desenvolvimento que oferece suporte para a linguagem de programação JAVA ([ORACLE, 2017a](#)). O Andoid Studio permite que um mesmo projeto seja disponibilizado para diferentes versões do Android. Este projeto está disponível para os sistemas da versão 16 (Jelly Bean) até a versão 25 (Nougat).

Integrado ao Andoid Studio está o SDK (*Software Development Kit*, em português Kit de Desenvolvimento de Software) ([DEVELOPERS, 2016](#)). Esse kit inclui documentação, código e utilitários para que os programadores consigam desenvolver suas aplicações de acordo com o padrão de desenvolvimento para a plataforma Android. Tanto o SDK quanto a IDE (Andoid Studio) são desenvolvidas pelo Google.

O desenvolvimento de uma aplicação para a plataforma Android, de forma nativa, exige a utilização da linguagem de programação JAVA e o XML (*Extensible Markup Language*) ([W3C, 2017](#)). O JAVA é uma linguagem de programação orientada a objetos desenvolvida na década de 90. Ao contrário de linguagens como o C, que ao serem compiladas geram um binário o qual pode ser executado diretamente pelo computador, o JAVA gera um *bytecode* que deverá ser interpretado por uma máquina virtual (*Java Virtual Machine*, a JVM). Já o XML é uma recomendação da W3C para gerar linguagens de marcação. Ele é um dos subtipos da SGML (acrônimo de *Standard Generalized Markup Language*, em português Linguagem Padronizada de Marcação Genérica) capaz de descrever diversos tipos de dados. Sendo o JAVA utilizado para fazer as operações em *back-end*, e o XML para montar os layouts das telas que são utilizadas no aplicativo, ou

seja, para fazer o *front-end*.

Para a compilação dos códigos, também é necessário o *Java Development Kit* (JDK), utilizado na versão 8. Atualmente o Android Studio também permite que seja utilizada a linguagem C++ juntamente com os códigos em JAVA.

Também foi necessário para o desenvolvimento deste projeto a construção de um *web service* em PHP (*Hypertext Preprocessor*) (GROUP, 2016). Este *web service* ficou responsável por fazer a integração entre os dispositivos móveis (smartphones), os clientes, e o servidor. O PHP é uma linguagem de programação interpretada, livre, e que pode ser mesclada dentro de códigos em HTML. O objetivo principal da linguagem é permitir o desenvolvimento de páginas na *World Wide Web* que serão geradas dinamicamente. O *web service* recebe as solicitações dos clientes e as responde com dados que estão armazenados no banco que se encontra no servidor. Assim, grande parte do processamento fica a cargo do servidor e não dos dispositivos móveis.

Na comunicação entre os dispositivos móveis (aplicativo) e o *web service*, utilizou-se uma padronização para formatação dos textos das mensagens trocadas entre essas aplicações, o JSON. O JSON é uma formatação de texto completamente independente da linguagem de programação, pois usa convenções que são familiares a várias linguagens como o C, JAVA, Python, dentre outras. Ele está constituído em duas estruturas: Uma coleção de pares chave/valor (por exemplo: {"nome":"John", "idade":30}); E uma lista ordenada de valores (por exemplo: ["a","b"]) (ECMA, 2017).

Para o gerenciamento do armazenamento dos dados foi utilizado o MySQL (ORACLE, 2016b) que utiliza a linguagem SQL (*Structured Query Language*, em português Linguagem de Consulta Estruturada) (WOZ, 2016) como interface. O MySQL é um sistema de gerenciamento de banco de dados (SGBD) relacional, ou seja, um sistema que opera sobre um banco de dados (BD), que é uma coleção estruturada de dados. Um banco de dados relacional armazena os dados em tabelas separadas ao invés de colocá-los todos juntos em um grande depósito. As estruturas do banco de dados são organizadas e mantidas em arquivos físicos. O MySQL é open source e é desenvolvido, distribuído e mantido pela Oracle (ORACLE, 2017b).

## 3.2 Atividades Desenvolvidas

Como atividade inicial do projeto foi estabelecido um diagrama de Casos de Uso, o qual serviu para guiar todo o desenvolvimento do aplicativo. Diagramas de Caso de Uso são úteis no início do desenvolvimento do projeto uma vez que conferem uma visão resumida de quem usa ou interage com o aplicativo, os chamados atores, e quais ou com quais recursos funcionais (as funcionalidades) eles poderão interagir. Ou seja, temos uma visão de quem utiliza e o que podem fazer com ele. Dessa forma apresenta-se um diagrama

que documenta o que o aplicativo faz do ponto de vista dos usuários (LANGUAG, 2017).

Na Figura 8 é apresentado o diagrama de Casos de Uso que guiou o desenvolvimento deste projeto. São apresentados dois atores, o usuário e o *web service*, sendo o usuário qualquer pessoa que utilize o aplicativo, e o *web service* uma aplicação responsável por fazer a comunicação entre o aplicativo e o servidor. Cada uma das dez (10) elipses representa um caso de uso que vem a ser uma funcionalidade que deverá ser disponibilizada pelo aplicativo.

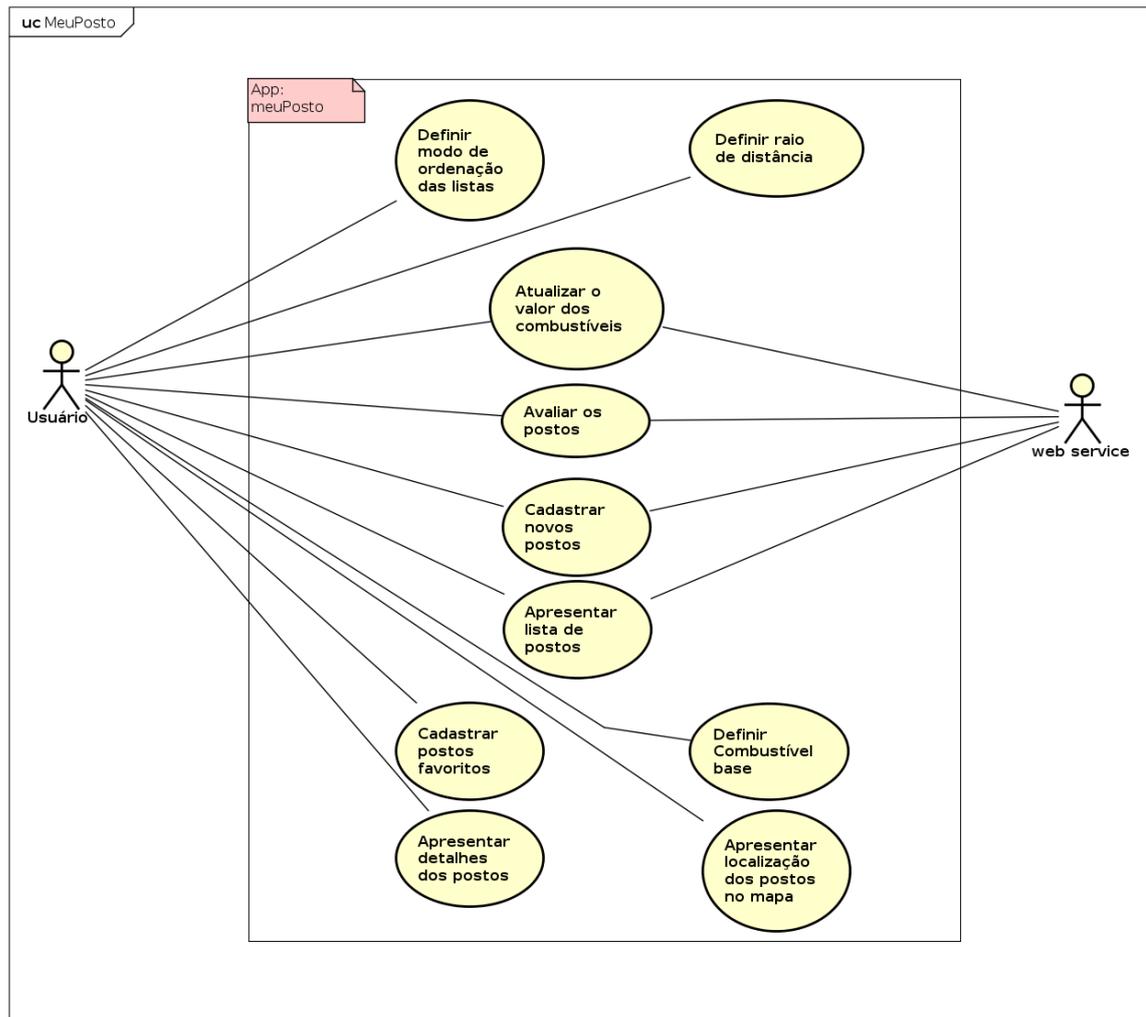


Figura 8 – Casos de uso.

Inicialmente o aplicativo foi desenvolvido apenas para a plataforma Android, uma vez que este vem a ser à aplicação cliente, o front-end com o qual os usuários interagem. A medida que se progrediu o desenvolvimento das telas do aplicativo, e que então se fez necessário a criação do back-end para mostrar as devidas informações e testar as funcionalidades do aplicativo como sugerido na seção Método (1.2), desenvolveu-se então em paralelo o *web service* responsável por fazer a integração da comunicação entre o aplicativo e o banco de dados que se encontra no servidor.

No banco de dados foram criadas 5 (cinco) tabelas, sendo elas: *posto*, *avaliacao*, *combustivel*, *historico* e *preco* (Figura 9). A tabela *posto* armazena todos os postos criados pelos usuários via aplicativo. Ela armazena para cada nova entrada o nome do posto, o endereço (logradouro e número), bairro, cidade, estado, coordenadas geográficas (longitude e latitude), bandeira, telefone, e a média das avaliações feitas pelos usuários. Essa média é calculada a cada nova inserção na tabela *avaliacao*.

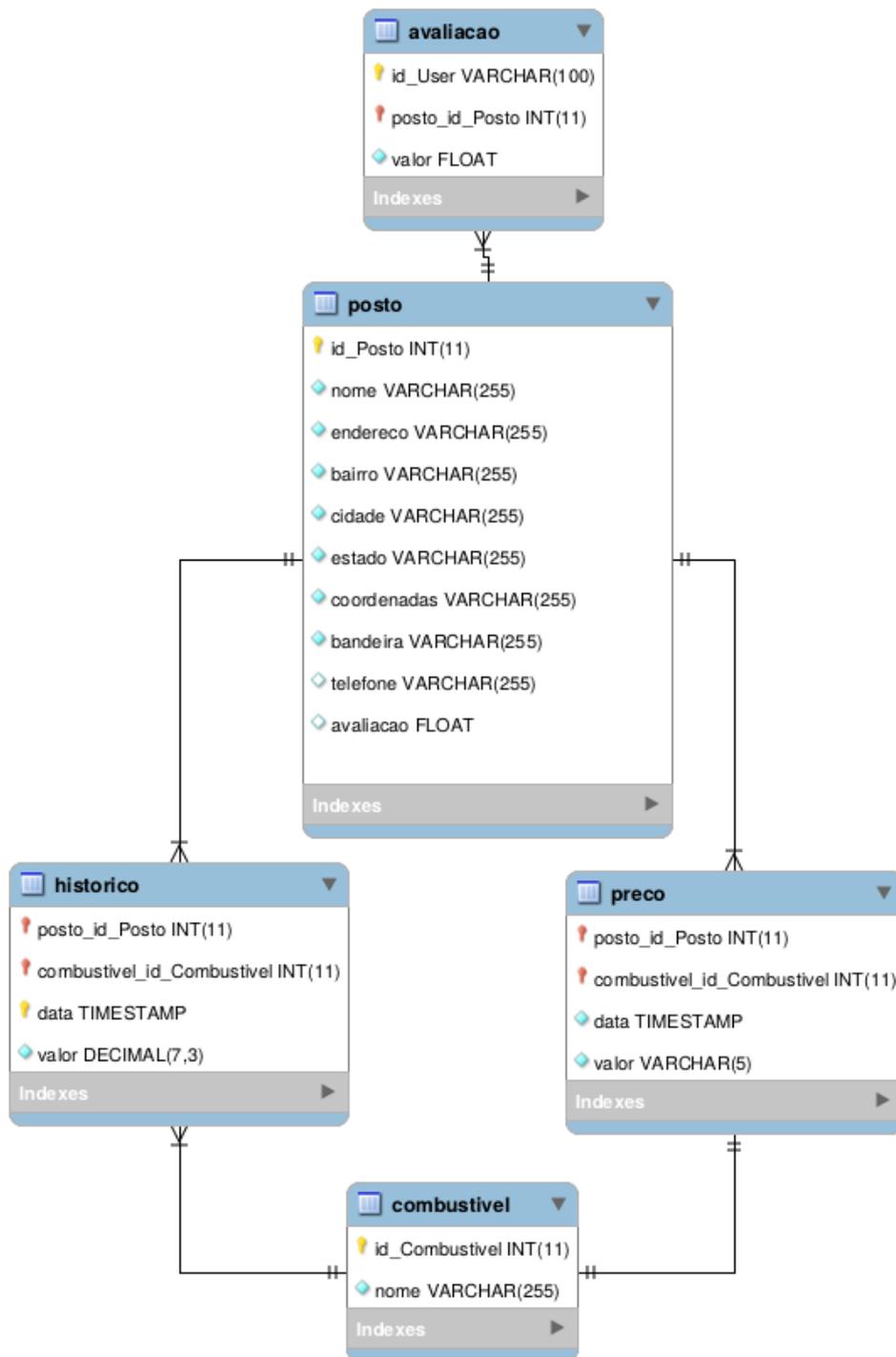


Figura 9 – Banco de dados MeuPosto

A tabela *avaliacao* armazena todas as avaliações feitas por todos os usuários. Para cada avaliação ela armazena o id do usuário, o id do posto e a nota dada pelo usuário. Ao se inserir uma nova linha na tabela, ou ao ser realizado um *update* em alguma delas, é disparado um *trigger* que calcula a média referente a coluna *valor*, para todas as linhas da tabela que contenham os mesmos id's de usuário e de posto da linha atualizada ou inserida. Após se calcular a média, esse valor é lançado na tabela *posto* na forma de um *update* na coluna *avaliacao* para o posto que tenha o mesmo id do posto que disparou o *trigger* na tabela *avaliacao*.

A tabela *combustivel* armazena os nomes dos combustíveis comercializados por grande parte dos postos de combustível do país. São eles: Diesel, Etanol, Gasolina, Gasolina Aditivada e GNV (Gás Natural Veicular).

A tabela *historico* armazena todos os preços sugeridos pelos usuários para todos os combustíveis em todos os postos de combustível, ao longo do tempo. Para isso ela armazena o id do posto, o id do combustível, a data em que esse valor foi salvo e o valor sugerido pelo usuário. Após ser realizada uma nova entrada na tabela, é disparado um *trigger* que analisa se para aquela mesma data existem pelo menos outras duas entradas. Caso positivo, é então calculada a mediana entre os 3 (três) últimos valores lançados para àquela data, para as entradas que contenham os mesmos id's de posto e combustível da nova linha inserida na tabela. Esse valor é então atualizado na tabela *preco* na linha que contenha os mesmos id's de posto e combustível do lançamento realizado na tabela *historico*. Assim, o valor do combustível para o qual os usuários de fato tem acesso somente é alterado quando pelo menos três usuários atualizam o valor de um combustível para um dado posto em um mesmo dia. Dessa forma tenta-se evitar que valores equivocados sejam apresentados para todos os usuários do aplicativo.

A tabela *preco* armazena os preços de todos os combustíveis para todos os postos contidos na base de dados. Essa tabela é responsável por manter o preço que de fato será apresentado para os usuários. Quando um novo posto é criado, ou seja, quando ocorre uma inserção na tabela *posto*, é disparado um *trigger* que cria cinco linhas nessa tabela, uma para cada possível combustível comercializado (Diesel, Etanol, Gasolina, Gasolina aditivada e GNV). Assim como a tabela *historico*, ela também armazena o id do posto, o id do combustível, a data em que o valor foi salvo na tabela e o valor do combustível que é a mediana advinda da tabela *historico*, conforme processo de cálculo descrito no parágrafo anterior.

A tela inicial do aplicativo possui 3 (três) abas, sendo elas: Home, Mapa e Favoritos (Figura 10). Na aba Home é apresentada uma lista com os postos de combustível que estão dentro do raio definido pelo usuário nas configurações (Figura 11). Cada item da lista possui o nome do posto, o endereço, uma imagem com a bandeira do posto, o preço do combustível base do usuário (também definido nas configurações), a distância entre à

atual localização do usuário e o posto do item, e a quanto tempo aquele valor foi atualizado na base de dados.

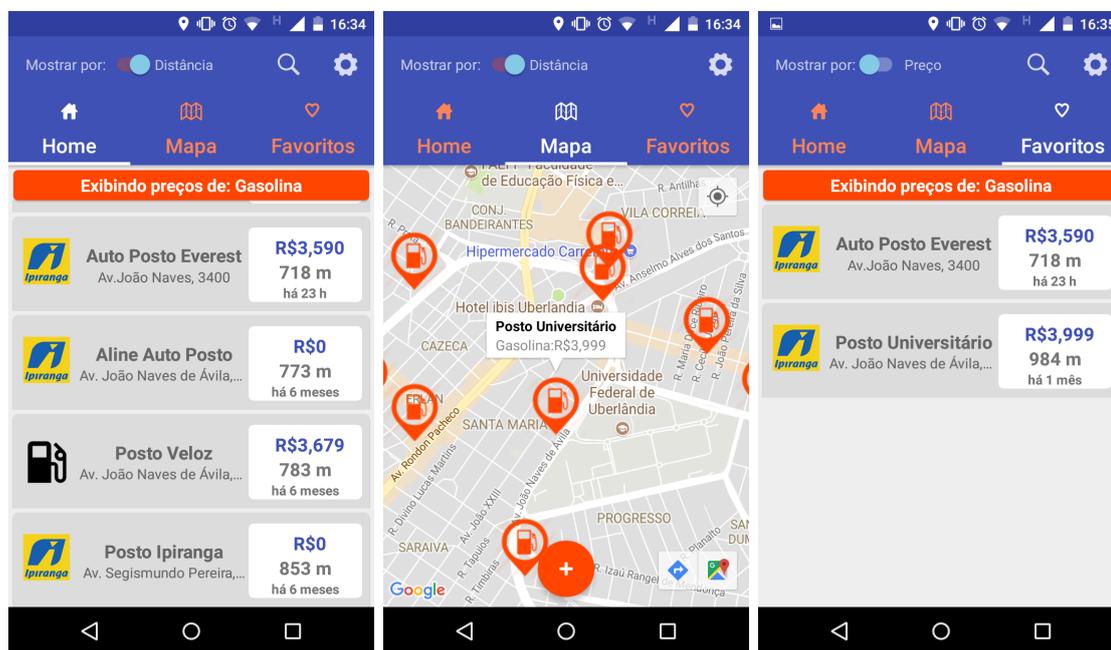


Figura 10 – Telas iniciais do aplicativo. Da esquerda para direita: Home, Mapa, Favoritos.

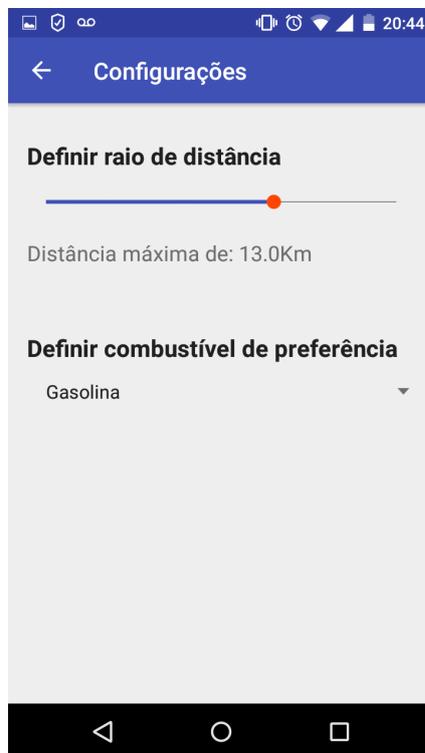


Figura 11 – Tela de configurações.

Na aba Mapa são apresentados os mesmos postos exibidos na aba Home, porém é apresentada a localização de cada um deles no mapa. Quando o usuário clica em um dos

*marker's* é exibido o nome do posto e o valor do combustível base nesse posto. Já na aba Favoritos é exibida uma lista apenas com os postos favoritos do usuário. Dessa forma ele pode ver de forma rápida e prática os preços praticados pelos postos que mais frequenta. Os dados exibidos são os mesmos da lista exibida na aba Home.

A distância entre o usuário e os postos de combustível na base de dados é calculada pelo *web service* de acordo com o algoritmo 1. Para que não seja calculada a distância entre o usuário e todos os postos cadastrados na base de dados, é realizado um filtro (*select*) na tabela *posto*, que retornará todos os postos da base de dados que estejam no mesmo estado e cidade em que o usuário se encontra. Esses dados (cidade, estado, distância máxima permitida e coordenadas geográficas da posição do usuário) são enviadas via requisição GET pelo aplicativo ao *web service*. O *web service* então retorna um JSON com a lista dos postos que passaram pelo filtro, juntamente com os combustíveis e seus respectivos preços praticados por cada um dos postos da lista, bem como a distância entre cada posto e o usuário.

---

**Algorithm 1** Algoritmo calcular distância entre duas coordenadas geográficas

---

**Require:** *latUser* latitude da coordenada onde o usuário se encontra

**Require:** *longUser* longitude da coordenada onde o usuário se encontra

**Require:** *latPosto* latitude da coordenada onde o posto está localizado

**Require:** *longPosto* longitude da coordenada onde o posto está localizado

**Ensure:** *dist* distância em metros entre as coordenadas do usuário e do posto

1:  $\theta = \text{longUser} - \text{longPosto}$

2:  $\text{dist} = \text{seno}(\text{deg2rad}(\text{latUser})) * \text{seno}(\text{deg2rad}(\text{latPosto})) + \text{cosseno}(\text{deg2rad}(\text{latUser})) * \text{cosseno}(\text{deg2rad}(\text{latPosto})) * \text{cosseno}(\text{deg2rad}(\theta))$   
 {*deg2rad()* converte um número em graus ao equivalente em radianos}

3:  $\text{dist} = \text{acos}(\text{dist})$  {*acos()* é a complementar de *cosseno()*, logo  $a == \text{cosseno}(\text{acos}(a))$ }

4:  $\text{dist} = \text{rad2deg}(\text{dist})$  {*rad2deg()* converte o número em radianos ao equivalente em graus}

5:  $\text{miles} = \text{dist} * 60 * 1.1515$  {distância em milhas}

6:  $\text{dist} = \text{miles} * 1.609344 * 1000$  {distância em metros}

7: **return** *dist*

---

Para obter a lista de postos favoritos do *web service*, o aplicativo envia para o mesmo uma requisição GET contendo todos os id's dos postos que o usuário indicou como favorito. Esses id's são gravados com o auxílio de um componente do Android chamado *SharedPreferences* que salva na memória permanente do dispositivo dos usuários todas as configurações solicitadas pelo aplicativo. Também é enviado para o *web service* as coordenadas do usuário para que também possa ser calculada a distância entre os postos favoritos e a localização atual do mesmo.

Para facilitar a visualização dos dados de um determinado posto, uma vez que as listas exibidas no aplicativo podem conter vários deles, o usuário pode fazer uma pesquisa pelo nome do posto em qualquer uma delas, ou seja, as pesquisas podem ser realizadas

nas abas Home e Favoritos. A cada letra digitada na barra de pesquisa são exibidos os postos que contenham em seu nome os caracteres digitados (Figura 12).

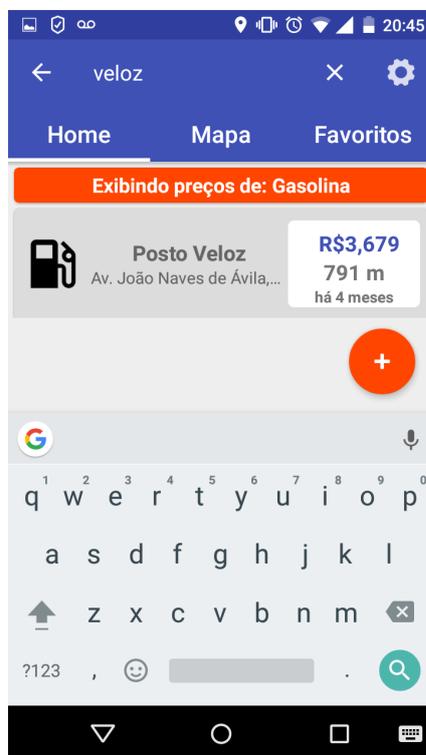


Figura 12 – Pesquisa pelo nome de um dado posto.

Caso o usuário queira ver mais detalhes de um dado posto, basta que ele clique em um dos itens de qualquer uma das listas exibidas nas abas Home e Favoritos, que ele será redirecionado a uma nova tela (*activity*). Nessa nova tela (Figura 13) são exibidas todas as informações disponíveis sobre àquele posto na base de dados. Além das informações que já são exibidas no item da lista como o nome, o endereço e a bandeira do posto, agora também são exibidos o telefone do posto assim como a quantidade e a média das avaliações dadas pelos usuários para aquele posto. Também são exibidos os preços de todos os combustíveis cadastrados para esse posto, e não apenas o combustível base como é apresentado nas listas. Nessa tela o usuário pode cadastrar um novo combustível para esse posto (Figura 14) clicando no botão com ícone de +. O usuário também pode atualizar o preço de um dos combustíveis exibidos clicando sobre o nome ou preço de um deles, avaliar o posto clicando sobre o ícone das estrelas exibido logo abaixo do telefone do posto, ou torná-lo favorito clicando no ícone do coração exibido no canto superior direito da tela. Além disso, o usuário pode visualizar o posto em questão no mapa, sendo ele apresentado de forma individual, ao contrário da aba Mapa que apresenta todos os postos dentro do raio definido pelo usuário.

As ações de atualizar o preço de um dado combustível e avaliar um dado posto, requerem a comunicação com o *web service* para que de fato tais alterações sejam replica-

das para todos os outros usuários. Para atualizar o valor de um combustível o aplicativo envia uma requisição via POST para o *web service* passando-se o id do posto, o id do combustível e o novo valor informado pelo usuário. Esse valor é então salvo na tabela *historico*. Após ser realizada a inserção dessa nova informação na tabela, será avaliado se o valor desse combustível para esse posto, passados pelo requisição, deve ser atualizado na tabela *preco* como já descrito anteriormente. Para a avaliação de um dado posto é enviada uma requisição também via POST ao *web service* passando-se o id do posto, id do usuário, e nota dada por ele ao posto. O id do usuário é obtido com o auxílio de um método do Android que retorna um id único que cada aparelho possui. Quando o *web service* recebe a requisição para avaliar um posto ele primeiramente verifica se aquele usuário já tinha realizado uma avaliação para esse mesmo posto antes, dependendo do resultado ele faz um *insert* ou um *update* na tabela *avaliacao* (caso o usuário já tenha feito uma avaliação realiza-se um *update*, caso contrário *insert*). Após ser realizado o *insert* ou o *update* é disparado um *trigger* que calcula a média das avaliações para àquele posto, e essa média é então salva na tabela *posto* (é realizado um *update* na coluna *avaliacao* para o posto que tenha o mesmo id do posto que disparou o *trigger*).



Figura 13 – Tela com detalhes de um dado posto.

Nas abas Home e Mapa é exibido um botão com ícone de + e clicando nesse botão o usuário é redirecionado para uma nova tela de cadastro. Nessa tela (Figura 15) o usuário precisa apenas preencher o nome do posto (entrada de carácter obrigatório), escolher a bandeira do posto, e se possível o telefone do posto (entrada não obrigatória). Os dados relativos ao endereço do posto são preenchidos automaticamente dada a localização atual

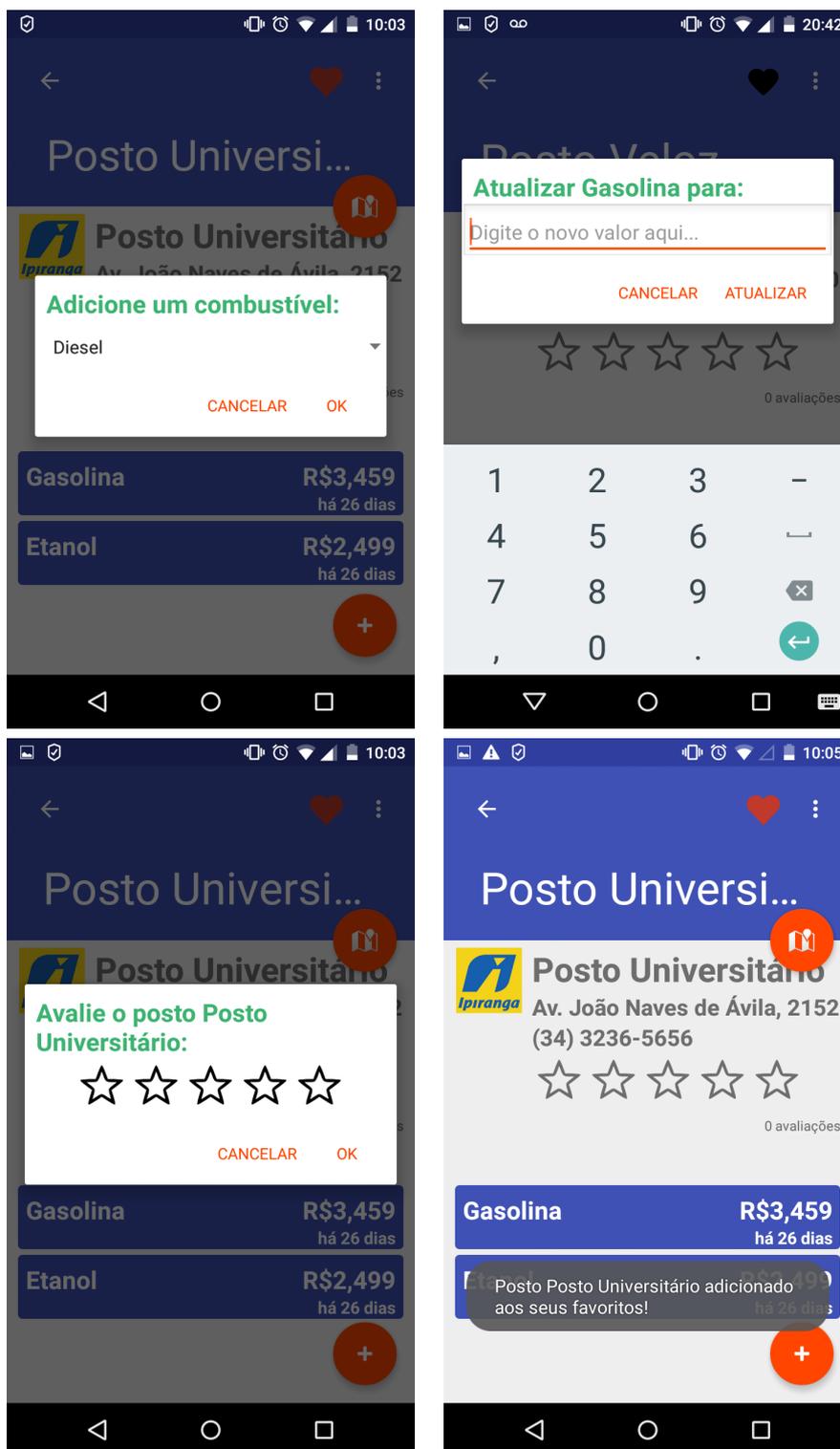


Figura 14 – Possíveis ações na tela de detalhes de um dado posto. O usuário pode realizar o cadastro de um novo combustível, bem como atualizar os preços de todos os combustíveis apresentados. Também pode realizar a avaliação do posto apresentado, e adicioná-lo ou removê-lo dos favoritos.

do usuário. Dessa forma, a ideia é que ao estar abastecendo seu veículo ou motocicleta em um dado posto, ao o usuário detectar que esse posto não está cadastrado na base de

dados do aplicativo, ele então o faça. Após preencher os dados, basta que o usuário clique no botão com o ícone de um disquete para enviar tais informações para o *web service* que irá tentar salvar essas informações no banco de dados que se encontra no servidor. Caso já exista na base de dados um posto com as mesmas coordenadas do novo posto, essa nova entrada é rejeitada. Se o usuário estiver sem conexão com a internet, ou se o GPS estiver desativado, o usuário não é redirecionado para essa tela, visto que eles são necessários para o perfeito funcionamento desses recursos. Em vez disso é exibida uma mensagem pedindo para o usuário verificar tais recursos (Figura 16).

Para que o cadastro de um novo posto possa de fato ser realizado faz-se necessário que o aplicativo faça uma requisição via POST ao *web service*. Nessa requisição o aplicativo envia todos os dados obtidos do posto, ou seja, o nome, endereço, bairro, cidade, estado, coordenadas geográficas, telefone e bandeira. Caso não exista outro posto na base de dados com as mesmas coordenadas geográficas do posto informado, ele é então gravado na base de dados e é retornada uma mensagem de sucesso.



A imagem mostra a interface de usuário para o cadastro de um novo posto. O título da tela é "Cadastrar Posto". O formulário contém os seguintes campos:

- Nome do posto:** Um campo de texto com o placeholder "Digite o nome do posto...".
- Selecione a bandeira:** Um menu suspenso com "Petrobrás" selecionado e o logo da Petrobrás.
- Endereço:** Um campo de texto com o placeholder "Rua Irmã Dúlce" e o número "55".
- Bairro:** Um campo de texto com o placeholder "Santa Mônica".
- Cidade:** Um campo de texto com o placeholder "Uberlândia".
- Telefone:** Um campo de texto com o placeholder "Digite o telefone do posto...".

Um botão verde com um ícone de disquete (salvar) está localizado no canto inferior direito da tela.

Figura 15 – Tela de cadastro de um novo posto.

Clicando no ícone das engrenagens na tela inicial ou no ícone  na tela de detalhes de um dado posto, o usuário é redirecionado para a tela de configurações (Figura 11). Nessa tela o usuário pode definir o raio de distância, o qual refere-se a distância máxima entre o usuário e um dado posto contido na base de dados. Também é possível definir o combustível base ou combustível de preferência, ou seja, o combustível mais utilizado pelo usuário. Ao sair dessa tela as configurações são salvas automaticamente. O usuário



Figura 16 – Tela com mensagem de falha na conexão com a internet, ou GPS desativado.

também pode ser redirecionado para essa tela caso o aplicativo tente buscar postos na base de dados e a lista retornada pelo *web service* seja vazia, nesse caso um dos possíveis problemas pode ser o raio estar com um valor pequeno, então o aplicativo sugere para que o usuário aumente o raio de distância.

### 3.3 Estado Atual do Desenvolvimento e Trabalhos Futuros

As funcionalidades propostas para o aplicativo, ainda na fase de planejamento, foram as seguintes (seção 1.2):

1. Definir o raio de distância máxima entre a atual localização do usuário e os postos de combustível a serem listados;
2. Definir o modo de ordenação para apresentar a lista dos postos;
3. Definir o combustível base;
4. Apresentar a lista com os postos mais próximos ao usuário;
5. Apresentar os detalhes sobre os postos de combustível;
6. Apresentar a localização no mapa;
7. Permitir que o usuário atualize o valor dos combustíveis;

8. Permitir que o usuário avalie os postos de combustível;
9. Cadastrar os postos favoritos;
10. Cadastrar novos postos de combustível.

As funcionalidades 1 e 3 estão disponíveis na tela de configurações (Figura 11). As funcionalidades 2, 4, e 6 estão disponíveis na tela inicial/principal do aplicativo (Figura 10). Clicando em um posto de qualquer uma das listas apresentadas nas abas Home ou Favoritos, ambas da tela inicial do aplicativo, o usuário será redirecionado para uma nova tela onde poderá visualizar todos os detalhes do posto selecionado, ou seja, poderá visualizar todas as informações contidas em nossa base de dados sobre aquele determinado posto de combustível (funcionalidade 5). Nessa tela ele também poderá visualizar o posto em questão no mapa (funcionalidade 6). Também poderá atualizar o valor dos combustíveis do posto (funcionalidade 7), bem como avaliá-lo (funcionalidade 8), e incluir ou retirar o posto da lista de seus favoritos clicando no ícone de coração (funcionalidade 9, Figuras 13 e 14). Ainda nas abas Home ou Mapa, o usuário irá visualizar um botão com um ícone de +. Clicando nele o usuário será redirecionado para uma tela onde poderá fazer o cadastro de um posto (funcionalidade 10, Figura 15).

Além das funcionalidades propostas, no decorrer deste projeto conseguiu-se desenvolver outras funcionalidades, como a possibilidade dos usuários poderem realizar uma pesquisa pelo nome do posto em qualquer uma das listas apresentadas pelo aplicativo nas abas Home e Favoritos (Figura 12). Também implementou-se o cadastro de novos combustíveis para um dado posto. Essa funcionalidade está disponível na tela de detalhes do posto (Figura 13), a qual contém um botão com um ícone de +. Clicando nesse botão aparece um *pop-up* com uma lista com todos os prováveis combustíveis comercializados pelo posto. Assim, basta o usuário selecionar o combustível e clicar em ok.

Para que os usuários não precisassem fechar o aplicativo e abri-lo novamente para visualizar as últimas alterações realizadas na base de dados, eles podem realizar uma operação de atualização semelhante a de aplicativos como o Facebook <sup>1</sup>. Essa atualização pode ser realizada em qualquer uma das listas, aba Home e ou Favoritos, bastando o usuário clicar na tela no aparelho e fazer um movimento de cima para baixo (estando ele em uma das referidas abas).

Futuramente espera-se implementar as seguintes funcionalidades no aplicativo:

- **Desenvolver uma nova forma de cadastro dos postos de combustível:** Além da forma de cadastro atual, que preenche automaticamente a localização do posto a ser cadastrado com a atual localização do usuário, os usuários poderão utilizar o mapa para mostrar a localização do posto que desejam cadastrar;

<sup>1</sup> Facebook <<https://play.google.com/store/apps/details?id=com.facebook.katana&hl=pt-BR>>

- **Alterar o modo de visualização da aba Mapa:** Essa aba, além de apresentar a localização dos postos com os *marker's*, também deverá apresentar uma linha com a melhor rota entre a atual localização do usuário e cada um dos postos apresentados nela;
- **Alterar o ícone do *marker* na aba Mapa:** Também na aba Mapa, alterar o ícone dos *marker's* para todos os postos que forem mal classificados por um grande número de usuários do aplicativo, além de emitir algum alerta sobre esse fato;
- **Disponibilizar outras formas de ordenação das listas (abas Home e Favoritos):** Os usuários poderão ordenar as listas dos postos de combustível por nome, bandeira, endereço, tempo de atualização do valor do combustível base, dentre outras possíveis formas;
- **Utilizar API do Google:** O *web service* deverá utilizar a API (*Application Programming Interface*, em português Interface de Programação de Aplicação) do próprio Google para fazer o cálculo da distância entre a atual localização do usuário e os postos de combustível cadastrados na base de dados.

Atualmente o aplicativo se encontra na versão 2.2, a qual está publicada na loja oficial para a plataforma Android, a Google Play ou Play Store <sup>2</sup>. Da versão inicial do aplicativo (versão 1.0) para a versão 2.0, a grande mudança foi quanto ao uso de uma biblioteca que facilitou o processo de comunicação entre o aplicativo e o *web service*, a *Squareup Retrofit 2* (SQUARE, 2017). Essa biblioteca implementa um cliente HTTP *type-safe* para o Android e para o JAVA. Tem-se a vantagem de se recuperar e carregar JSON através das requisições HTTP de uma forma facilitada, uma vez que com ela é possível configurar um conversor para serialização dos dados. Nesse projeto foi utilizado o GSON, uma biblioteca JAVA desenvolvida pelo Google que pode ser usada para converter objetos JAVA para uma representação em JSON e vice-versa (GOOGLE, 2017a). Então, ao receber uma lista de postos em formato JSON por exemplo, é realizada uma conversão automática para uma lista contendo objetos postos em JAVA, o que antes era feito de forma manual (versão 1.0).

Na versão 1.0 utilizava-se o objeto `URLConnection` para se fazer as requisições HTTP ao *web service*. Dessa forma era criada uma classe para cada diferente tipo de requisição que deveria ser realizada, uma vez que as repostas advindas do *web service* eram tratadas manualmente. Com a utilização da biblioteca *Squareup Retrofit 2* bastou criar uma interface com todos os tipos de requisição, e então criar um objeto que implementasse essa interface juntamente com o conversor de JSON.

<sup>2</sup> Google Play <<http://play.google.com/store>>

Da versão 2.0 para a mais recente (versão 2.2) foram realizadas apenas correções de layout, a fim de deixar o aplicativo mais agradável para o usuário.

## 3.4 Testes e Resultados

Com o aplicativo e o *web service* funcionais, criou-se um app beta para que usuários reais pudessem testá-lo. Desta forma, seria possível verificar se ocorreriam falhas na utilização do aplicativo. Para isso, foi criado um questionário com as seguintes perguntas:

1. Qual a sua versão do Android?
  - Android 4.0 - Ice Cream Sandwich
  - Android 4.1/4.2/4.3 - Jelly Bean
  - Android 4.4 - KitKat
  - Android 5.0/5.1 - Lollipop
  - Android 6.0 - Marshmallow
  - Outra
2. Qual o tamanho da tela do seu smartphone?
  - 4 polegadas ou menor
  - 4,5 polegadas
  - 5 polegadas
  - 5,5 polegadas ou maior
3. Em alguma tela a informação não aparece completamente?
4. Ao executar alguma ação (mudar a ordenação dos postos na lista, cadastrar um posto, dentre outras) o aplicativo fecha inesperadamente?
5. Os dados apresentados são suficientes para o correto entendimento das informações?
6. Houve algum problema na localização por GPS?
7. As cores utilizadas no app são adequadas?
8. Como você se conecta a internet?
  - 3G
  - 4G
  - Wifi

9. Considerando que sua conexão com a internet esteja funcionando corretamente, alguma ação demora mais do que você gostaria?
10. Alguma ação não está funcionando como deveria?

Dos itens de 3 a 7, 9 e 10 as opções eram apenas sim ou não. Sendo que para os itens 3, 4, 9 e 10 os usuários (doze) poderiam especificar qual foi o problema caso respondessem sim.

A Figura 17 apresenta as respostas para a pergunta: Qual a sua versão do Android? (item 1) Observa-se que grande parte dos usuários que realizaram os testes e que responderam o questionário utiliza uma versão mais antiga do Android, a 4.4 (uma vez que o Android já se encontra na versão 7). Mas na segunda colocação houve um empate entre uma versão mais atual, a 6.0, e a opção "outra", que englobará a versão 7.

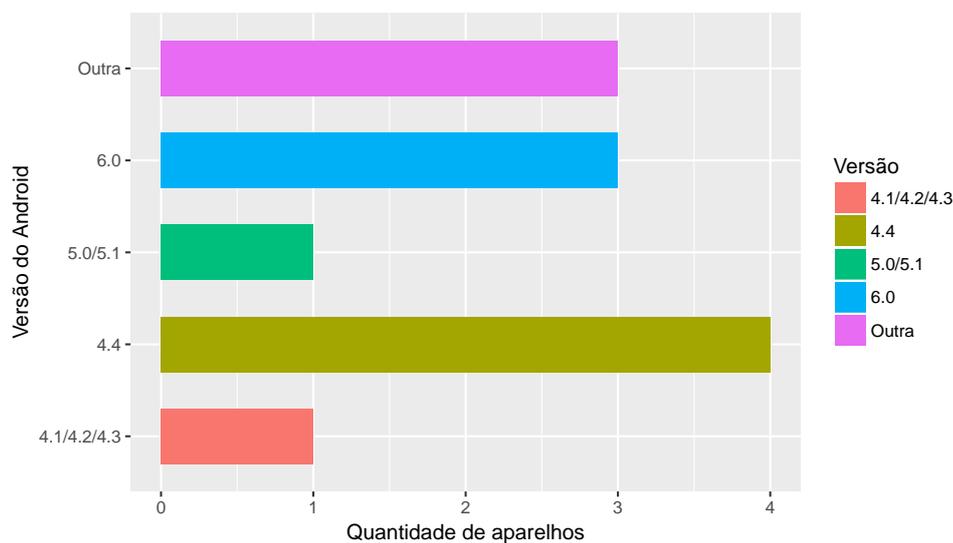


Figura 17 – Versões do Android utilizadas pelos usuários do teste.

A Figura 18 apresenta as respostas para a pergunta: Ao executar alguma ação (mudar a ordenação dos postos na lista, cadastrar um posto, dentre outras) o aplicativo fecha inesperadamente? (item 4). 83,33% dos usuários responderam que não, ou seja, a maioria deles não tiveram grandes problemas ao utilizarem o aplicativo. O fechamento inesperado de um app é um grande fator desmotivador para os usuários, o que infelizmente ocorreu para 16,67% dos usuários que realizaram os testes.

A Figura 19 apresenta as respostas para a pergunta: Os dados apresentados são suficientes para o correto entendimento das informações? (item 5). 58,33% dos usuários responderam que sim e 41,67% responderam que não. Logo, para mais da metade dos usuários dos testes as informações que são apresentadas pelo app são suficientes para que eles consigam entendê-las.

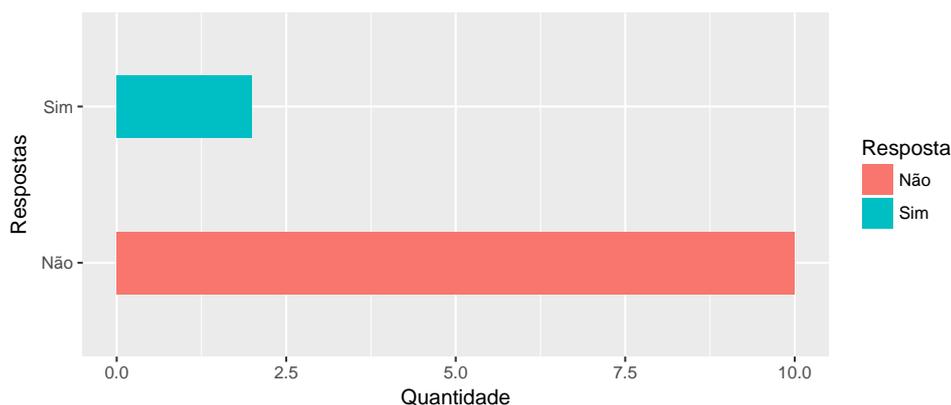


Figura 18 – Respostas para: Ao executar alguma ação (mudar a ordenação dos postos na lista, cadastrar um posto, dentre outras) o aplicativo fecha inesperadamente?

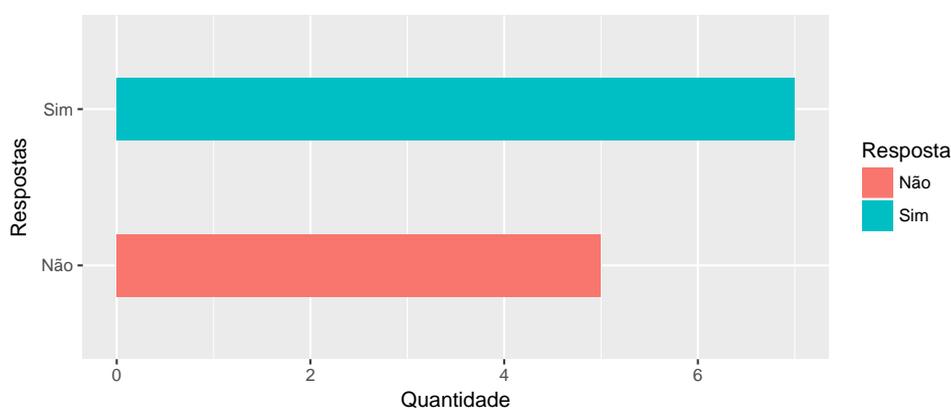


Figura 19 – Respostas sobre se os dados apresentados pelo aplicativo são suficientes.

A Figura 20 apresentam as respostas para a pergunta: Como você se conecta a internet? (item 8). 5 (cinco) dos 12 (doze) usuários responderam que utilizam apenas o Wifi para se conectar a Internet (41,67%). Em segundo lugar temos 3G; 4G; Wifi com 25%, em seguida aparecem 3G; Wifi com 16,67%, por fim 3G e 4G; Wifi com 8,33%.

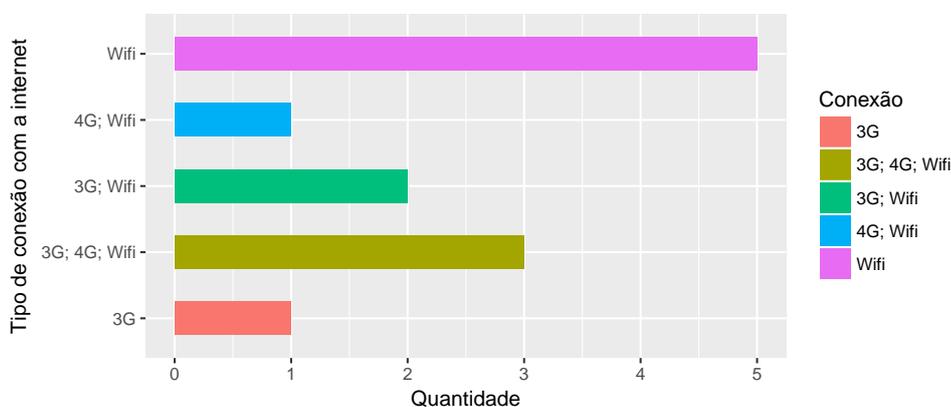


Figura 20 – Como os usuários que testaram o aplicativo se conectaram a internet.

Esses testes foram realizados tanto com a versão 1.0 quanto com a versão 2.0 do aplicativo. A medida em que se evoluía no desenvolvimento do aplicativo e que as mudanças se mostravam satisfatórias em testes internos (Alfa), atualizava-se então o apk ao qual os usuários de teste tinham acesso.

Também foram realizados testes através de ferramentas disponibilizadas pelo próprio Google para desenvolvedores que publicam seus aplicativos na loja oficial de aplicativos para a plataforma Android. Ao publicar um aplicativo como sendo uma versão Alfa ou Beta, o desenvolvedor pode habilitar um mecanismo que realiza testes de forma automatizada, chamado Relatório de pré-lançamento, o qual pode ser habilitado ou desabilitado nas configurações do Google Play Console da conta do desenvolvedor <sup>3</sup>.

Tais testes foram realizados na versão mais recente do aplicativo (versão 2.2). A Figura 21 apresenta em quantos aparelhos foram realizados os testes, em quantos deles houve algum problema e em quantos não houve. Já a Figura 22 apresenta o resultado para o teste de segurança, no qual é possível observar que não foram encontradas vulnerabilidades conhecidas no aplicativo.

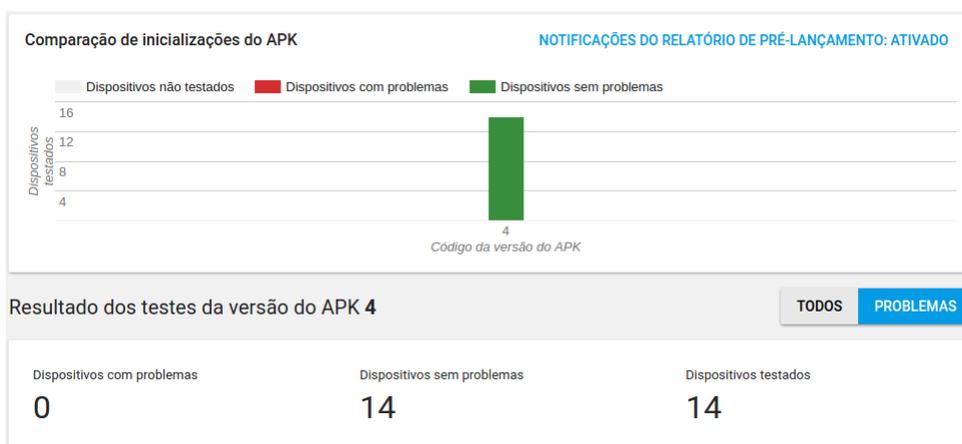


Figura 21 – Resultados para os erros detectados no aplicativo MeuPosto em sua versão 2.2 (APK 4).

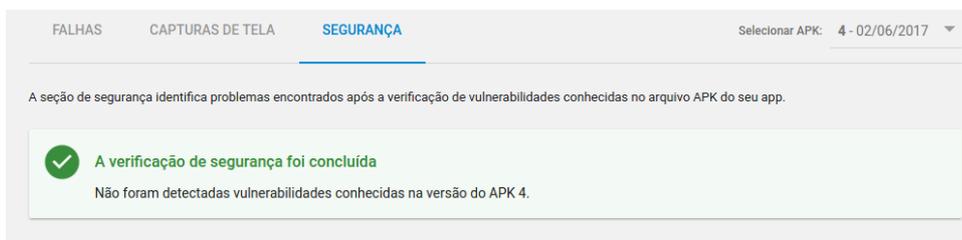


Figura 22 – Resultados sobre as falhas de segurança detectadas no aplicativo MeuPosto em sua versão 2.2 (APK 4).

<sup>3</sup> Google Play Console <<https://play.google.com/apps/publish/>>

A Figura 23 apresenta as versões do Android que foram utilizadas para se realizar os testes automatizados do Google. Observa-se que foram realizados testes utilizando desde uma versão mais antiga do Android a 4.4 até a uma versão que ainda se encontra em *preview*, ou seja que ainda está em fase de desenvolvimento (em Beta), o Android O. Assim como nos testes realizados pelos usuários (Figura 17) grande parte dos testes foi realizada na versão 4.4.

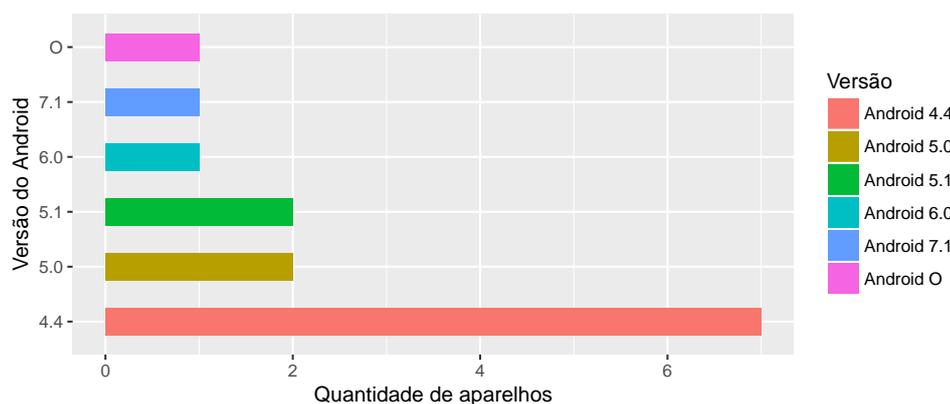


Figura 23 – Versões do Android utilizadas nos testes realizados pelo relatório de pré-lançamento do Google Play Console.

A Figura 24 apresenta quais foram os modelos de aparelho utilizados para os testes. Observa-se que praticamente foi utilizado um modelo para cada teste, repetindo-se apenas os modelos Nexus 5 e Pixel, duas vezes cada um.

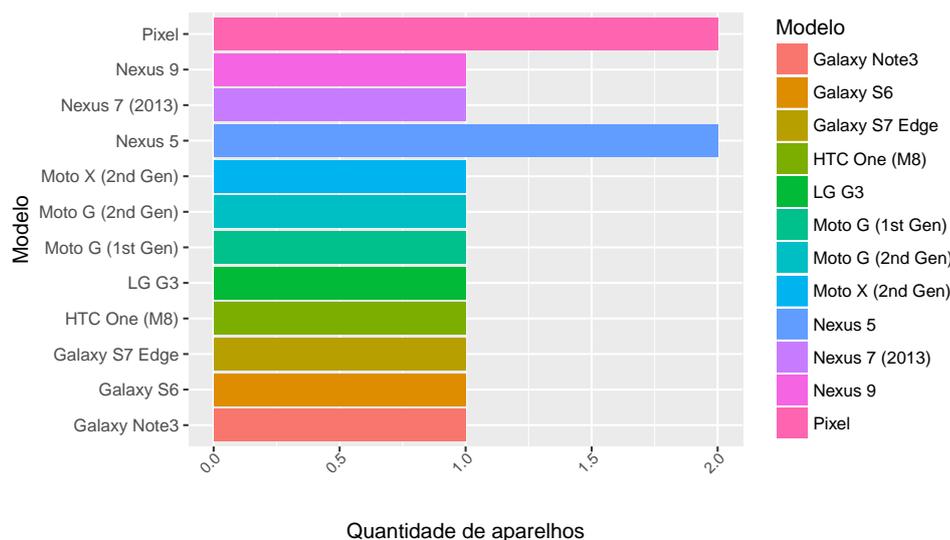


Figura 24 – Modelos de aparelhos utilizados nos testes realizados pelo relatório de pré-lançamento do Google Play Console.

## 4 Conclusão

No decorrer deste projeto foi idealizado e implementado um aplicativo para a plataforma de dispositivos móveis Android para permitir aos seus usuários trocar informações, de forma colaborativa, sobre preços e promoções em postos de combustível em uma determinada cidade ou região, além de informações sobre a qualidade dos postos. Nesse sentido, criou-se um sistema que conta com um aplicativo e um servidor os quais permitem formar uma rede colaborativa com dupla função: a de prover uma economia de custos para seus usuários e aumentar a fiscalização dos postos de combustível.

Afim de se testar a eficácia da aplicação, foi disponibilizada de forma livre e gratuita o apk (extensão para arquivos executáveis na plataforma Android) do aplicativo para qualquer pessoa que quisesse utilizar/testar o mesmo. Ao mesmo tempo disponibilizou-se um questionário para que se pudesse verificar as constatações dos usuários. Também foi realizado um teste automatizado disponibilizado pelo Google ao se publicar um aplicativo na loja oficial Google Play/Play Store. Em ambos os testes os resultados foram satisfatórios, uma vez que não foram constatadas grandes falhas, como o fechamento inesperado do aplicativo ou um possível travamento do aparelho. Nos testes de segurança do Google também não foram encontradas vulnerabilidades no aplicativo.

Para que o desenvolvimento deste projeto fosse possível fez necessário um conhecimento multidisciplinar. Disciplinas como Lógica para Computação (GSI005), Estrutura de Dados (GSI006), Programação Orientada a Objetos 1 e 2 (GSI015, GSI020), Modelagem de Software (GSI025), Redes de Computadores (GSI023), Banco de Dados 1 e 2 (GSI016, GSI021) e Computação Móvel (GBC100) foram de suma importância para esse projeto. E proporcionaram o completo desenvolvimento do mesmo com resultados satisfatórios. Os conhecimentos adquiridos em tais disciplinas por si só já possibilitariam grande parte do desenvolvimento, cabendo apenas buscar alguns conhecimentos específicos quanto ao uso de algumas bibliotecas e ou componentes mais específicos utilizados pelo aplicativo.

Pretende-se continuar o desenvolvimento deste projeto a fim de se garantir uma experiência cada vez mais agradável para os usuários, além de se utilizar componentes e ou bibliotecas mais recentes que tornem o aplicativo mais rápido e eficiente. Funcionalidades como uma nova forma de cadastro dos postos de combustível, novas formas de ordenação das listas das abas Home e Favoritos, e a utilização da API do próprio Google para fazer o cálculo da distância entre a atual localização do usuário e os postos de combustível, já estão sendo planejadas e logo deverão estar disponíveis para todos os usuários.

# Referências

- ANUÁRIO Estatístico 2015. 2016. Disponível em: <<http://www.anp.gov.br/?pg=76798&m=anu%Elrio&t1=&t2=anu%Elrio&t3=&t4=&ar=0&ps=1&1460142202145>>. Acesso em: 08 abr 2016. Citado na página 9.
- APACHE. *Apache Software Foundation*. 2016. Disponível em: <<http://www.apache.org/foundation/>>. Acesso em: 12 maio 2016. Citado na página 13.
- APP, E. G. *Easy Gas*. 2016. Disponível em: <<https://play.google.com/store/apps/details?id=com.devnavts.easygas>>. Acesso em: 12 maio 2016. Citado na página 14.
- COMBUSTÍVEIS, P. dos. *Preço dos Combustíveis*. 2016. Disponível em: <<https://play.google.com/store/apps/details?id=br.com.precodoscombustiveis.pdc>>. Acesso em: 12 maio 2016. Citado na página 14.
- DEPARTAMENTO NACIONAL DE TRÂNSITO. *Frota Nacional (Dezembro 2015)*. 2016. Disponível em: <<http://www.denatran.gov.br/frota2015.htm>>. Acesso em: 07 abr 2016. Citado na página 8.
- DEVELOPERS, G. *Tools Android Studio and SDK*. 2016. Disponível em: <<http://developer.android.com/>>. Acesso em: 11 abr 2016. Citado 3 vezes nas páginas 11, 12 e 20.
- ECMA. *Introdução ao JSON*. 2017. Disponível em: <<http://www.json.org/json-pt.html>>. Acesso em: 07 Julho 2017. Citado na página 21.
- GOOGLE. *Android Open Source Project*. 2016. Disponível em: <<https://source.android.com/>>. Acesso em: 12 maio 2016. Citado na página 13.
- GOOGLE. *Gson User Guide*. 2017. Disponível em: <<https://sites.google.com/site/gson/gson-user-guide>>. Acesso em: 04 Junho 2017. Citado na página 33.
- GOOGLE. *Material Design para Android*. 2017. Disponível em: <<https://developer.android.com/design/material/index.html?hl=pt-br>>. Acesso em: 07 Julho 2017. Citado na página 16.
- GROUP, T. P. *Documentation*. 2016. Disponível em: <<http://php.net/docs.php>>. Acesso em: 12 maio 2016. Citado 2 vezes nas páginas 14 e 21.
- INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. *Acesso à Internet e à Televisão e Posse de Telefone Móvel Celular para Uso Pessoal*. 2016. Disponível em: <<http://biblioteca.ibge.gov.br/visualizacao/livros/liv93373.pdf>>. Acesso em: 07 abr 2016. Citado 2 vezes nas páginas 7 e 8.
- INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. *Projeção da população do Brasil e das Unidades da Federação*. 2016. Disponível em: <<http://www.ibge.gov.br/apps/populacao/projecao/>>. Acesso em: 07 abr 2016. Citado na página 8.

- INSTITUTO DE PESQUISA ECONÔMICA APLICADA. *Comunicado Ipea 2013*. 2016. Disponível em: <[http://www.ipea.gov.br/agencia/index.php?option=com\\_content&view=article&id=20330&catid=161](http://www.ipea.gov.br/agencia/index.php?option=com_content&view=article&id=20330&catid=161)>. Acesso em: 08 abr 2016. Citado na página 8.
- JETBRAINS. *IntelliJ IDEA*. 2017. Disponível em: <<https://www.jetbrains.com/idea/>>. Acesso em: 12 abr 2017. Citado na página 20.
- LABS, T. *Posto de Gasolina*. 2016. Disponível em: <<https://play.google.com/store/apps/details?id=com.teuflabs.postodegasolina>>. Acesso em: 12 maio 2016. Citado na página 14.
- LANGUAG, U. M. *UML Use Case Diagrams*. 2017. Disponível em: <<http://www.uml-diagrams.org/use-case-diagrams.html>>. Acesso em: 03 Junho 2017. Citado na página 21.
- LECHETA, R. R. *Aprenda a criar aplicações para dispositivos móveis com o Android SDK*. 4. ed. São Paulo, SP, BR: NOVATEC, 2009. Citado 2 vezes nas páginas 13 e 14.
- LINUX KERNEL ORGANIZATION. *The Linux Kernel Archives*. 2016. Disponível em: <<https://www.kernel.org/>>. Acesso em: 12 maio 2016. Citado na página 13.
- OPEN HANSET ALLIANCE. *Open HANset Alliance*. 2016. OHA. Disponível em: <[http://www.openhandsetalliance.com/oha\\_members.html](http://www.openhandsetalliance.com/oha_members.html)>. Acesso em: 12 maio 2016. Citado na página 13.
- ORACLE. *Java*. 2016. Disponível em: <<https://www.java.com>>. Acesso em: 12 maio 2016. Citado na página 14.
- ORACLE. *Why MySQL*. 2016. Disponível em: <<https://www.mysql.com/why-mysql/#pt-0-0>>. Acesso em: 12 maio 2016. Citado 2 vezes nas páginas 14 e 21.
- ORACLE. *Java Documentation*. 2017. Disponível em: <<https://docs.oracle.com/en/java/>>. Acesso em: 12 abril 2017. Citado na página 20.
- ORACLE. *What is MySQL*. 2017. Disponível em: <<https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html>>. Acesso em: 12 abril 2017. Citado na página 21.
- SOUZA, J. G. d. M.; POMPERMAYER, F. M. Variações no preço do etanol em comparação ao preço da gasolina: uma análise da resposta do consumidor. Instituto de Pesquisa Econômica Aplicada (Ipea), 2015. Citado na página 8.
- SQUARE, I. *Retrofit*. 2017. Disponível em: <<http://square.github.io/retrofit/>>. Acesso em: 04 Junho 2017. Citado na página 33.
- TECHNOLOGIES, M. S. S. for I. *Mobile/Tablet Operating System Market Share*. 2017. NET APPLICATIONS. Disponível em: <<http://www.netmarketshare.com/operating-system-market-share.aspx?qprid=8&qpcustomd=1>>. Acesso em: 12 abr 2017. Citado 2 vezes nas páginas 2 e 7.
- W3C. *Extensible Markup Language (XML)*. 2017. Disponível em: <<https://www.w3.org/TR/REC-xml/#sec-intro>>. Acesso em: 12 abril 2017. Citado na página 20.
- WOZ, S. *What is sql*. 2016. Disponível em: <<http://sqlwoz.com/what-is-sql/>>. Acesso em: 12 maio 2016. Citado 2 vezes nas páginas 14 e 21.