

UNIVERSIDADE FEDERAL FLUMINENSE
INSTITUTO DE COMPUTAÇÃO

MARCELLO CUNHA DA ROCHA VAZ
RAISSA PEREIRA AMARAL

TRACKBUS: UM APLICATIVO DE APOIO AOS USUÁRIOS DE ÔNIBUS NA CIDADE
DO RIO DE JANEIRO

Niterói
2017

MARCELLO CUNHA DA ROCHA VAZ
RAISSA PEREIRA AMARAL

TRACKBUS: UM APLICATIVO DE APOIO AOS USUÁRIOS DE ÔNIBUS NA CIDADE
DO RIO DE JANEIRO

Monografia apresentada ao Curso de Graduação
em Ciência da Computação da Universidade
Federal Fluminense, como parte dos requisitos
necessários à obtenção do título de Bacharel em
Ciência da Computação.

Orientador: Prof. Dr. Leonardo Gresta Paulino Murta

Niterói
2017

Ficha Catalográfica elaborada pela Biblioteca da Escola de Engenharia e Instituto de Computação da UFF

V393 Vaz, Marcello Cunha da Rocha
Trackbus : um aplicativo de apoio aos usuários de ônibus na
cidade do Rio de Janeiro / Marcello Cunha da Rocha Vaz, Raissa
Pereira Amaral. – Niterói, RJ : [s.n.], 2017.
80 f.

Projeto Final (Bacharelado em Ciência da Computação) –
Universidade Federal Fluminense, 2017.
Orientador: Leonardo Gresta Paulino Murta.

1. Aplicativo móvel. 2. Modalidade urbana. 3. Ônibus. I.
Amaral, Raissa Pereira. II. Título.

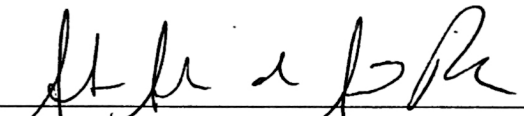
CDD 004

MARCELLO CUNHA DA ROCHA VAZ
RAISSA PEREIRA AMARAL

TRACKBUS: UM APLICATIVO DE APOIO AOS USUÁRIOS DE ÔNIBUS NA
CIDADE DO RIO DE JANEIRO

Monografia apresentada ao Curso de
Graduação em Ciência da Computação da
Universidade Federal Fluminense, como
parte dos requisitos necessários à
obtenção do título de Bacharel em Ciência
da Computação.

BANCA EXAMINADORA



Prof. Dr. ANTONIO AUGUSTO DE ARAGÃO ROCHA



Prof. Dra. ISABEL CRISTINA MELLO ROSSETI



Prof. Dr. LEONARDO GRESTA PAULINO MURTA - Orientador

Niterói
2017

AGRADECIMENTOS (MARCELLO)

Agradeço aos meus pais, Afonso e Rosa, e ao meu irmão, Lucas, por sempre me apoiarem e incentivarem a fazer as coisas que eu gosto, além de providenciarem tudo o que foi preciso para chegar até aqui.

A Raissa, companheira não só deste trabalho, mas de toda a jornada na UFF.

Ao Gabriel Bastos, que além de ser um grande amigo, fez um excelente trabalho com a parte visual do *TrackBus*.

Ao orientador do projeto, Leonardo Murta, por todo o esforço, boa vontade e dedicação nas revisões, além das ideias e dicas essenciais para a criação do *TrackBus*.

Aos integrantes da banca, Isabel Rosseti e Antonio Rocha, que aceitaram avaliar nosso trabalho, mesmo tendo outras responsabilidades e prazos.

A todos os amigos da UFF e da Finxi e todos os professores, que me ajudaram no desenvolvimento pessoal e profissional, que será útil para a vida toda.

AGRADECIMENTOS (RAISSA)

Agradeço, primeiramente, aos meus pais e irmão, Ângela, Dilber e Iago, pelo apoio incondicional em todos os momentos, acreditando mais em mim do que eu mesma, e oferecendo comida japonesa e carinho para as horas de desespero, mesmo a 600 km de distância. Mas sempre alertando: “Fica tranquila, que depois piora”. Eu não estaria aqui sem vocês.

A minha madrinha Edna, por ser a minha segunda mãe até quando não precisava.

Ao meu amigo e parceiro de trabalho Marcello, que esteve comigo desde o primeiro dia da faculdade e topou entrar comigo neste projeto.

Ao nosso professor e orientador, Leonardo Murta, pela disponibilidade e paciência na concepção da ideia, revisões, discussões, enormes contribuições e por apoiar integralmente este trabalho que era uma missão quase impossível.

A Alessandro Campello e Flávia Crizanto, por estarem sempre dispostos a ajudar, seja com dúvidas relativas à estrutura do texto ou ao desenvolvimento do projeto.

Aos amigos da UFF 2012.1 e de todos os outros períodos que se cruzaram no meio de todas as matérias sem uma sequência definida, se eu sobrevivi a todas elas foi graças a vocês.

À turma do Flat 6 por estar comigo no melhor ano da minha vida. Aos amigos e à família de Valadares por sempre representarem a minha casa, e aos de Niterói por fazerem com que aqui seja também a minha casa.

Aos professores da UFF que contribuíram com todos os ensinamentos ao longo de cada período, em especial Isabel Rosseti e Antonio "Guto" Rocha que aceitaram participar da nossa banca para avaliar e contribuir com nosso trabalho.

Aos amigos da SEO Master, TIM e Visagio, por contribuírem para a profissional que eu estou prestes a me formar.

A todos que contribuíram de alguma forma para minha formação, seja pessoal, acadêmica ou profissional.

Muito obrigada!

RESUMO

Dados de 2015 apontam que o número de usuários de ônibus em toda a região sudeste totaliza 40,8 milhões. No Rio de Janeiro, a frota de ônibus corresponde a apenas 0,01% do total de veículos particulares em circulação. O alto número de usuários oposto à frota de veículos disponíveis impacta a população que depende diariamente desse meio de transporte público. Em paralelo a esses números, ocorreu um aumento significativo de *smartphones* no Brasil nesse mesmo período, chegando a 154 milhões em todo o país. Unindo todas essas informações, a fim de aliviar a insatisfação dos usuários com o transporte público, este trabalho fornece um aplicativo para *smartphones* que exibe, em um mapa, diversos ônibus das linhas da cidade do Rio de Janeiro, atualizando sua localização em tempo real. O aplicativo também exibe pontos de parada e pontos turísticos. Além disto, oferece a opção de emitir notificações de proximidade para ambos os pontos. Estas notificações podem emitir sons e a distância pode ser alterada de acordo com a preferência do usuário.

Palavras-chave: ônibus, aplicativo, mobilidade urbana.

ABSTRACT

According to a study made in 2015, 4.8 million people regularly use buses as the primary mean of transport in the southeastern region of Brazil. In Rio de Janeiro, the bus fleet corresponds only to 0,01% of the total vehicles in use. The high number of bus users opposed to the vehicle fleet directly impacts the population that depends on this mean of public transport. Parallel to these numbers, there was a significant increase of smartphones in Brazil at this same period, amassing 154 milion smartphones in the whole country. Gathering all this information, in order to lessen users' dissatisfaction with public transportation, this project provides a smartphone application that shows on a map several bus lines from Rio de Janeiro, updating their location in real time. The application also shows bus stops and tourist attractions. In addition, also offers the option to set up notifications when there are nearby stops or tourist attractions, these notifications may include sound and their distance can be changed according to user preference.

Palavras-chave: ônibus, aplicativo, mobilidade urbana.

SUMÁRIO

Capítulo 1 – Introdução	14
1.1 Motivação	14
1.2 Cenários de utilização	15
1.2.1 João Pedro, o estudante-estagiário	15
1.2.2 Juliana, a trainee.....	15
1.2.3 Lorena, a turista	16
1.3 Objetivo.....	16
1.4 Organização	17
Capítulo 2 – Os aplicativos de ônibus do Rio de janeiro	18
2.1 Introdução	18
2.2 GPS	19
2.3 DATA RIO.....	20
2.4 Moovit.....	22
2.5 Trafi.....	23
2.6 Vá de Ônibus.....	24
2.7 Google Maps.....	25
2.8 Análise Comparativa.....	25
2.9 Considerações Finais	28
Capítulo 3 – TrackBus	30
3.1 Introdução	30
3.2 Listagem de requisitos	30
3.3 Exibição de duas ou mais linhas no mapa	37
3.4 Proximidade em metros	39

3.5 Listagem dos pontos de ônibus	41
3.6 Pontos turísticos	42
3.7 Acompanhamento de viagem.....	44
3.8 Notificações	45
3.9 Considerações finais	47
Capítulo 4 – Decisões de desenvolvimento e implementação	48
4.1 Introdução	48
4.2 O aplicativo Híbrido	48
4.3 Apache Cordova.....	50
4.4 AngularJS.....	51
4.5 Ionic Framework.....	52
4.6 Arquitetura	52
4.7 Bibliotecas Utilizadas	53
4.7.1 Turf	53
4.7.2 NgCordova.....	53
4.7.3 Angular Google Maps.....	54
4.8 Pré-processamento dos dados	54
4.9 Estrutura dos dados	58
4.10 Detalhes de implementação	59
4.10.1 Detalhes da utilização de plugins.....	60
4.10.2 Algoritmo de identificação de pontos de parada.....	60
4.11 Considerações finais	62
Capítulo 5 – Avaliação dos usuários.....	64
5.1 Introdução	64
5.2 Resultados da avaliação	65

5.2.1 Experiência com o aplicativo.....	66
5.2.2 Facilidade de interação	66
5.2.3 Comparação de linhas	66
5.2.4 Notificação de proximidade do ônibus	67
5.2.5 Localização de pontos.....	68
5.2.6 Notificação de proximidade de pontos de parada	68
5.2.7 Notificação de proximidade de pontos turísticos.....	69
5.2.8 utilização Continuada do aplicativo.....	70
5.2.9 Recomendação do aplicativo	70
5.2.10 Curto Resumo do aplicativo.....	71
5.3 Ameaças à validade.....	71
5.4 Considerações finais	72
Capítulo 6 – Conclusão	73
6.1 Contribuições	73
6.2 Limitações.....	73
6.3 Trabalhos futuros	75
Capítulo 7 – Referências Bibliográficas	77
Apêndice A – Formulário para identificação de funcionalidades dos aplicativos disponíveis...	81
Apêndice B – Formulário de mapeamento de funcionalidades do TrackBus.....	81

LISTA DE ILUSTRAÇÕES

Figura 1 – Percentual de usuários que vivem na cidade do Rio e regiões metropolitanas	32
Figura 2 – Frequência de utilização de ônibus pelos usuários.....	32
Figura 3 - Moradores - Relevância de requisitos	33
Figura 4 - Turistas - Relevância de requisitos	33
Figura 5 - Moradores - Relevância de alertas	35
Figura 6 - Turistas - Relevância de alertas.....	35
Figura 7 - Listagem de linhas próximas.....	36
Figura 8 - Listagem de linhas.....	38
Figura 9 – Listagem dos ônibus no mapa	38
Figura 10 - Proximidade em metros do ônibus	40
Figura 11 - Listagem e exibição dos pontos de uma linha.....	41
Figura 12 - Listagem e exibição de pontos turísticos	43
Figura 13 – Configurações de distância para notificação	45
Figura 14 – Notificação de proximidade do ponto	46
Figura 15 - Arquitetura do <i>Cordova</i>	50
Figura 16 - <i>AngularJS Two-Way Data Binding</i>	51
Figura 17 - Bases de dados provenientes do <i>Data.rio</i>	55
Figura 18 - Pseudocódigo	61
Figura 19 - Caso de falha e recuperação do algoritmo	62
Figura 20 - Resultado Avaliação - Facilidade de Interação	66
Figura 21 - Resultado Avaliação – Comparação de linhas	67
Figura 23 - Resultado Avaliação – Identificação de pontos de parada e turísticos	68
Figura 24 - Resultado Avaliação – Notificação de proximidade dos pontos de parada	69
Figura 25 - Resultado Avaliação - Notificação de proximidade de pontos turísticos	69
Figura 26 - Resultado Avaliação - Usaria novamente	70
Figura 27 - Resultado Avaliação - Recomendaria o aplicativo	71
Figura 28 – Situação de falha no modo alternativo de acompanhamento	74

LISTA DE TABELAS

Tabela 1 - Funcionalidade por aplicativo.....	28
Tabela 2 - Requisitos pesquisa de satisfação	31
Tabela 3 - Desenvolvimento nativo vs. híbrido	49
Tabela 4 - Amostra de dados base B3.....	56
Tabela 5 - Estrutura de dados de ônibus	58
Tabela 6 - Estrutura dos dados de pontos de parada.....	59
Tabela 7 - Estrutura dos dados dos pontos turísticos dentro de cada ponto de parada	59
Tabela 8 - Estrutura dos dados dos pontos turísticos.....	59
Tabela 9 – Questionário de avaliação dos usuários	65

CAPÍTULO 1 – INTRODUÇÃO

1.1 MOTIVAÇÃO

De acordo com o IBGE, a cidade do Rio de Janeiro possuía, em 2015, uma frota de 18.205 ônibus, valor que representava aproximadamente 0,01% do total de automóveis particulares circulando pela cidade (IBGE, 2015). O grande número de veículos particulares impacta diretamente no trânsito dessa metrópole, que só perde para São Paulo e Xangai, dentre outras 20 metrópoles, quando analisado o número de horas consumidas no trânsito pela população (IPEA, 2013).

O tempo gasto no trânsito impacta diretamente o cotidiano e a qualidade de vida de muitos cariocas, não só para aqueles que utilizam veículos particulares diariamente, mas, principalmente, para a parcela da população que depende de ônibus e outros tipos de transporte público para se locomover pela cidade. Segundo o Instituto de Pesquisa Econômica Aplicada (IPEA, 2011), 40,8 milhões de pessoas utilizam ônibus na região sudeste.

Qualquer pessoa que dependa de ônibus para se locomover pela cidade do Rio de Janeiro com certeza já se deparou com situações inconvenientes, como a espera excessiva em pontos com pouca segurança, dificuldade em identificar qual linha pegar ou em qual ponto descer para chegar ao seu destino. Esses problemas se potencializam ainda mais quando insere-se, na análise, usuários que não conhecem bem a cidade, como turistas.

Em 2015, o número de *smartphones* superou o de *notebooks* e *tablets* no Brasil (BRIGATTO, 2015), chegando a 154 milhões entre todos os estados. Esse aumento, alinhado com os problemas comumente enfrentados pela população na utilização de transporte público, motivaram a criação de aplicativos para facilitar a mobilidade urbana. Já existem soluções disponíveis que auxiliam os usuários de transporte público a se locomover pela cidade, como os aplicativos *Moovit*, *Trafi*, *Google Maps*, *Vá de Ônibus*, entre outros. Porém apenas o *Vá de Ônibus* oferece a localização dos ônibus em tempo real. A fim de incentivar desenvolvedores e analistas de dados a criarem soluções que proporcionem à população uma visão mais transparente da cidade, a prefeitura disponibiliza no portal *Data.rio*¹ diversas bases de dados

¹ <http://data.rio>

referentes a saúde, educação e transporte do Rio de Janeiro. No que se refere a este trabalho, as bases relacionadas a transporte trazem oportunidade para o desenvolvimento de novos aplicativos que ofereçam dados em tempo real aos usuários de ônibus.

1.2 CENÁRIOS DE UTILIZAÇÃO

Para uma caracterização mais real das diferentes necessidades impostas por diferentes perfis de usuários de ônibus da cidade do Rio de Janeiro, essa seção apresenta três *personas*: um estudante-estagiário, uma trainee e uma turista. Essas *personas* descrevem usuários e situações em que cada um deles vive quando utiliza o sistema de transporte público da cidade. Essas *personas* são a base para o levantamento de requisitos do aplicativo.

1.2.1 JOÃO PEDRO, O ESTUDANTE-ESTAGIÁRIO

João Pedro mora na Barra da Tijuca e estuda na UFRJ. João está quase se formando na faculdade, faz estágio e tem aulas no período noturno.

Como os ônibus para a Barra da Tijuca são pouco frequentes, costuma ficar muito tempo esperando no ponto sozinho. Além de se expor a assaltos ou algum outro tipo de violência enquanto espera, poderia estar estudando ou conversando com os amigos dentro da faculdade. Além disso, como o percurso até a sua casa é longo, João costuma dormir no trajeto de volta. Por esse motivo, ele já perdeu o ponto que precisa descer algumas vezes e acabou parando no ponto final, demorando ainda mais para chegar em casa.

1.2.2 JULIANA, A TRAINEE

Juliana é trainee da Ambev em São Cristóvão e mora em Vila Isabel. Ela tem duas opções de ônibus para chegar ao trabalho, porém com uma dessas opções precisa andar um pouco mais para chegar ao seu destino. Geralmente ela embarca no primeiro ônibus que aparece no ponto, pois o tempo de espera do próximo ônibus, na maioria das vezes, é maior do que o tempo gasto andando até o destino. Entretanto, em alguns casos, logo após embarcar no ônibus em que a viagem leva mais tempo, consegue ver o ônibus da outra linha, que faz um percurso mais rápido, se aproximando do ponto em que estava.

1.2.3 LORENA, A TURISTA

Lorena é uma estudante que está visitando o Rio de Janeiro em suas férias. Lorena está habituada a viajar sozinha e costuma buscar informações de qual o melhor caminho e quais conduções precisa embarcar antes de ir a determinados pontos turísticos. Infelizmente, Lorena está com bastante medo de utilizar o celular na cidade devido ao grande número de assaltos reportados recentemente.

Dessa forma, mesmo sabendo, por exemplo, qual linha de ônibus ela precisa embarcar para chegar ao Corcovado, ela precisaria consultar o celular para localizar um ponto próximo ao seu *hostel* e ainda qual ponto descer, uma vez embarcada no ônibus.

1.3 OBJETIVO

Considerando a motivação levantada na Seção 1.1, o objetivo deste trabalho é desenvolver um aplicativo para celular como prova de conceito que ofereça uma solução prática, acessível e que facilite a vida dos usuários de ônibus da cidade: o *TrackBus*. Apesar de atuar como prova de conceito de algumas funcionalidades específicas, este trabalho visa oferecer um produto real, pronto para ser testado e utilizado. Para definição dos requisitos foi utilizada uma técnica da Engenharia de Software que faz o uso de *personas* para definir pontos focais de atuação (GRUDIN; PRUITT, 2003). Dessa forma, os requisitos do *TrackBus* foram pensados a partir dos problemas enfrentadas pelas *personas* levantadas na Seção 1.2. As *personas* representam também exemplos de uso do aplicativo por usuários com perfis similares que enfrentam dificuldades com transporte público na cidade.

O *TrackBus* ajudará o João Pedro, *persona* descrita na Seção 1.2.1, pois ele poderá monitorar no aplicativo o seu ônibus e só se locomover até o ponto após receber uma notificação de que o veículo está se aproximando. Depois de embarcar no ônibus ele ainda poderá configurar uma nova notificação por áudio que o alertará quando o seu ônibus estiver se aproximando do ponto em que precisa descer, evitando que ele acorde no ponto final novamente. Juliana, *persona* descrita na Seção 1.2.2, também será beneficiada com o *TrackBus*, pois ela poderá selecionar no mapa as duas linhas de ônibus que ela pode embarcar para ir ao trabalho e então analisar se deve ou não esperar pelo ônibus que tem o percurso mais rápido. Lorena, *persona* descrita na Seção 1.2.3, também poderá selecionar a linha que deseja utilizar e, com isso, conseguirá se programar para chegar ao ponto quando o ônibus estiver se aproximando. Para mitigar o problema de

segurança, ela poderá programar uma notificação que a informe por áudio em qual ponto de parada deve descer para chegar ao Corcovado. Assim, o celular poderá ficar guardado no seu bolso durante todo o percurso.

O *TrackBus* utiliza bases de dados fornecidas pelo *Data.rio* para manter os usuários informados sobre os ônibus e seus respectivos pontos turísticos e de parada. Dentre as bases de dados utilizadas está a que fornece informações da localização em tempo real dos ônibus da cidade. Ela é utilizada para oferecer uma das principais funcionalidades do aplicativo: a indicação da posição real dos ônibus no mapa naquele dado momento. Além desta, também são utilizadas bases que informam localizações de pontos de parada e turísticos, seus respectivos nomes e pontos associados a cada linha.

1.4 ORGANIZAÇÃO

Este trabalho está dividido em outros cinco capítulos, além deste capítulo de introdução. O Capítulo 2 apresenta os aplicativos relacionados e as informações relativas ao funcionamento de GPS e do portal *Data.rio*, ferramentas essenciais para execução deste trabalho. O Capítulo 3 apresenta o *TrackBus*, explica como foram levantados seus requisitos e como são todas as funcionalidades disponíveis no aplicativo. Algoritmos, *plugins*, bases de dados e outras tecnologias que apoiaram a implementação destas funcionalidades são discutidos no Capítulo 4. No Capítulo 5 é apresentada a avaliação do aplicativo por parte de alguns usuários, suas primeiras impressões, avaliação das funcionalidades e sugestões de melhoria. Por último, o Capítulo 6 conclui este trabalho, apresentando suas principais contribuições, relatando suas limitações e levantando possíveis trabalhos futuros.

CAPÍTULO 2 – FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS

2.1 INTRODUÇÃO

Com o grande número de pessoas em áreas urbanas que dependem de transporte público, em especial dos ônibus (MOREIRA, 2015) e com um aumento significativo na quantidade de *smartphones* em uso nos últimos anos (FGV, 2016), é possível perceber que há amplo espaço para desenvolvimento de aplicativos que auxiliam a mobilidade urbana. Nas lojas de aplicativos dos sistemas operacionais *Android* e *iOS* podemos encontrar diversas opções de aplicativos que desempenham funções de auxílio à mobilidade urbana em diversas cidades do mundo, incluindo a cidade do Rio de Janeiro.

Para entender melhor o cenário de aplicativos de mobilidade urbana no Rio de Janeiro, foram realizadas buscas nessas lojas utilizando as palavras-chave “ônibus” e “rio”, a fim de encontrar os aplicativos mais utilizados e mais bem avaliados. Os dois aplicativos com a avaliação mais alta dos usuários foram o *Moovit*² e o *Trafi*³. Outro aplicativo analisado, porém com nota e número de *downloads* baixos, foi o ‘*Vá de Ônibus*’⁴. Esse aplicativo é fornecido pela FETRANSPO (Federação das Empresas de Transportes de Passageiros do Estado do Rio de Janeiro). Como ela é a agência que fornece as informações sobre os ônibus, uma análise mais detalhada no aplicativo disponibilizado por ela mostrou-se desejável.

O *Google Maps*⁵ não está presente nos primeiros resultados da busca, porém é o único dos aplicativos dessa categoria que já está na configuração padrão de todos os sistemas operacionais *Android*. O aplicativo da Google oferece sugestões de rotas por meio de transporte público, similares aos aplicativos analisados, em inúmeras cidades no mundo (GOOGLE MAPS, 2016). A *Google Playstore*, loja do sistema operacional *Android*, é a única que permite ao usuário ver o número exato de *downloads* de um aplicativo. Observando esse número foi

²<http://moovitapp.com/>

³<http://www.trafi.com/>

⁴<http://vadeonibus.com.br/>

⁵<https://www.google.com.br/maps/>

possível constatar que o *Google Maps* detém a maior base de usuários dentre todos os aplicativos citados até o momento.

Neste capítulo, será feita uma análise das funcionalidades, benefícios e limitações de cada um dos aplicativos citados. Primeiramente, na Seção 2.2 é explicada a história e o modo de funcionamento de aparelhos GPS. Na Seção 2.3 é apresentado o *Data.rio*, portal de dados abertos da prefeitura da cidade do Rio de Janeiro. As Seções 2.4 e 2.5 discutem sobre o *Moovit* e o *Trafi*. As Seções 2.6 e 2.7 compreendem o "Vá de ônibus", aplicativo fornecido pela FETRANSPOR, e *Google Maps*, aplicativo com o maior alcance em número de usuários. Ao final, na Seção 2.8, é apresentado um quadro comparativo de todos os principais pontos levantados no capítulo, oferecendo uma análise mais resumida de todos estes serviços. Os requisitos a serem listados nesta última seção são definidos ao longo da análise de cada aplicativo, nas quais as funcionalidades presentes e ausentes em cada um deles são destacadas.

2.2 GPS

Técnicas de posicionamento são pensadas e utilizadas há muitos anos, motivadas pela curiosidade do homem em saber se localizar tanto em regiões próximas, quanto para auxiliar em grandes navegações (MONICO, 2000). Tais técnicas foram se aprimorando em conjunto com o avanço da tecnologia e da necessidade de um posicionamento mais preciso, para então chegar ao GPS disponibilizado comercialmente hoje.

O método mais antigo de orientação é o baseado no posicionamento do sol e dos planetas. Porém, técnicas que dependam desses astros, mesmo quando utilizadas com instrumentos mais avançados (bússola, quadrante de Davis, sextante, etc.) dependem também da habilidade e conhecimento do navegador e de condições climáticas, além de não serem precisas (MONICO, 2000).

Com a imprecisão da navegação celeste e o avanço da eletrônica, sistemas baseados em frequência de ondas de rádio foram desenvolvidos, como o "*Long Range Navigation*" (LORAN), "*Decca Navigator System*" e "*Omega Navigation System*", seguidos pelo primeiro sistema de localização baseado em satélites artificiais, o NNSS (*Navy Navigation Satellite System*). Porém, todos eles apresentavam baixa precisão e/ou custo muito elevado. Em 1970 surgiu o GPS (*Global Positional System*), com a proposta de ser um sistema com boa precisão, facilidade de uso e custos acessíveis para os usuários (MONICO, 2000).

O GPS ou NAVSTAR (*Navigation Satellite With Time and Ranging*) – GPS consiste em 24 satélites orbitando em volta da Terra, transmitindo sinais em duas frequências diferentes. As duas frequências são utilizadas para correção de erros gerados por atrasos de propagação pela ionosfera (MESSRS.; ZOLLER, 1978). O programa para implantação do GPS, desenvolvido pelo Departamento de Defesa dos Estados Unidos, foi aprovado em 1973. O primeiro satélite foi colocado em órbita no ano de 1978 e o 24 ° apenas em agosto de 1993.

Quando lançado, o GPS tinha como principal foco o uso militar e de navegações, porém fornecia para uso civil o sinal com um erro em torno de 50 a 150 metros. A chamada *Selective Availability* (SA) introduzia erros intencionais que adulteravam o valor de um dos sinais transmitidos pelo satélite e “desviavam” a resposta da localização real. Essa era uma prática padrão para evitar que inimigos americanos pudessem utilizar os sinais civis dos receptores de GPS como arma de alta precisão. Entretanto, em maio de 2000, o então presidente Bill Clinton publicou a decisão de descontinuar o uso da SA (U.S COAST GUARD, 2000). A abolição da SA melhorou em até 10 vezes o nível de acurácia na transmissão do sinal (MONICO, 2000).

Do ano 2000 até o momento, os aparelhos receptores dos sinais enviados pelos satélites foram se aprimorando e se tornando cada vez mais populares. Inicialmente eram encontrados somente em dispositivos dedicados à funcionalidade de posicionamento. Contudo, hoje está incorporado na maioria dos *smartphones* e dispositivos móveis. Diversos aplicativos, como, por exemplo, *Waze*, *Foursquare* e *Pokémon GO*, utilizam o GPS dos dispositivos para, respectivamente, traçar rotas, identificar restaurantes e bares próximos e viabilizar jogos com realidade aumentada.

Todas essas aplicações se baseiam em dados passados pelo GPS, informando a latitude e longitude daquele usuário no momento. Os valores dessas coordenadas têm sua precisão associada ao número de casas decimais fornecidas na transmissão. Os dispositivos de GPS comerciais apresentam um máximo de quatro casas de precisão sem interferência no sinal ou sem uso de cálculos de correção diferencial. Essas quatro casas representam uma escala de até 11m de precisão (HUBER, 2015).

2.3 DATA RIO

Na última década, um movimento vem ganhando espaço em diversas áreas de conhecimento e pesquisas: o *Open Data* ou *Public Datasets*. Esse movimento, que prega a

utilização, reutilização e compartilhamento de informações, engloba diversas iniciativas que incentivam o desenvolvimento de *software open source*, *hardware* livre, acesso aberto a publicações científicas, trabalhos criativos abertos, entre muitos outros (MATOS, 2015).

Uma dessas iniciativas do movimento *Open data* é a disseminação do conceito de *Open Government Data* (OGD). Governos de vários países, entre eles o Brasil⁶, começaram a disponibilizar bases de dados livres para consulta e utilização em portais *online*. Essas bases são disponibilizadas principalmente como incentivo para que cientistas de dados, analistas e desenvolvedores possam oferecer análises, soluções e aplicações que beneficiem a sociedade. Além dos benefícios gerais que a abertura de dados possibilita, o movimento OGD argumenta que sua utilização promove também a transparência, a prestação de contas do governo, cria condições para uma entrega de informações mais inclusiva e promove uma maior participação popular (UBALDI, 2013).

Seguindo essa tendência de dados abertos, cidades maiores começaram a disponibilizar dados públicos específicos de seus respectivos domínios, fornecendo também transparência a níveis municipais. Um bom exemplo dessa transparência é o *New York City Open Data*⁷, um portal on-line que disponibiliza bases de dados, mapas e gráficos com informações relativas a diversos setores públicos, como educação, transporte e meio ambiente da cidade de Nova Iorque.

A Prefeitura do Rio de Janeiro também se posicionou favorável ao compartilhamento de dados e lançou em abril de 2014 o portal *Data.rio*⁸. O portal conta com um catálogo de mais de 15 mil arquivos com informações no âmbito de saúde, educação e transporte (ALBUQUERQUE, 2014).

Uma das bases disponibilizadas no portal *Data.rio* é a de modal rodoviário municipal, que mapeia o posicionamento por GPS de todos os ônibus circulando na cidade. Essa informação é disponibilizada em formato *JSON (JavaScript Object Notation)* que é atualizado minuto a minuto. Essa base, que é a fonte principal de obtenção de dados deste trabalho, já foi utilizada em outros trabalhos na UFF. Um deles relativo à predição estatística do tempo de viagem (SILVA, 2016), outro que apresenta uma análise histórica dos dados fornecidos pelas bases (SCHETTINO, 2016), entre muitos outros. Complementar a essa, também são disponibilizadas

⁶<http://dados.gov.br/>

⁷<https://nycopendata.socrata.com/>

⁸<http://data.rio/>

no *Data.rio* bases estáticas com as localizações e descrições representativas de pontos de ônibus e atrações turísticas, e relacionando as linhas e seus respectivos pontos de parada.

2.4 MOOVIT

Fundado no ano de 2011, em Israel, o aplicativo *Moovit* vem conquistando fãs nos 67 países e 1.200 cidades em que ele está disponível (MOOVIT, 2016). O aplicativo, que tem como principal finalidade auxiliar usuários a traçar melhores rotas utilizando serviços de transporte público, já consolida uma base de mais de 45 milhões de usuários ao redor do mundo, sendo, em 2015, foram registrados 4,4 milhões utilizadores apenas no Brasil, caracterizando assim um dos seus maiores mercados (NUNES, 2015). O *Moovit* também possui a segunda maior nota na loja de aplicativos do sistema operacional Android, a *Google Play Store*, quando realizada a busca por aplicativos utilizando palavras-chave relacionadas à mobilidade urbana no Rio.

O aplicativo, também conhecido por “*Waze*⁹ dos ônibus”, opera com dados estáticos de horários dos ônibus fornecidos pelas operadoras de transporte e autoridades, combinados com dados coletados em tempo real via *crowdsourcing* (KIM, 2012). Assim como o *Waze*, o *Moovit* possui um modo de navegação, no qual os usuários podem indicar que embarcaram em um ônibus de uma determinada linha. Esse modo possibilita que eles acompanhem o percurso do ônibus ponto a ponto e sejam notificados quando se aproximam dos pontos de parada do mesmo. Ao utilizar essa funcionalidade, cada usuário fornece anonimamente informações atualizadas da localização real daquele veículo (PURDY, 2012). Como exemplo de uso desses dados fornecidos pelos usuários, podemos supor um caso em que um ônibus está previsto para chegar em um ponto às 17h05, porém, o usuário navegando naquele ônibus ainda está distante da localização do ponto. O sistema automaticamente identifica que ocorrerá um atraso da programação prevista e transmite para os usuários que dependem deste ônibus.

Além das informações anônimas, os usuários também podem inserir pontos de ônibus não cadastrados ou indicar alguma alteração dos que já existem. Todas essas novas edições são revisadas pela equipe do aplicativo em até 48h para serem aprovadas e colocadas na plataforma (MUTTER, 2016). Atualizações das condições dos ônibus e pontos de parada (como lotação, limpeza e acessibilidade) também podem ser inseridas pelo usuário e disponibilizadas para toda a

⁹<https://www.waze.com/> - Aplicativo que sugere as melhores rotas a serem seguidas com veículos particulares baseado em informações passadas pelos usuários.

base cadastrada em tempo real (TARKAN, 2013). Todas estas alterações contribuem para o mapeamento mais preciso da rede de veículos públicos de cada cidade.

Entretanto, apesar de todas as vantagens que o *Moovit* proporciona para seus usuários, ainda existem muitas reclamações por parte dos mesmos disponíveis nas avaliações do aplicativo nas respectivas lojas de aplicativos dos sistemas operacionais *Android* e *iOS*. A maioria delas se deve à falha na estimativa do tempo previsto de chegada do ônibus. Mesmo com a funcionalidade de melhorar a informação disponível por meio de *crowdsourcing*, ainda existem informações imprecisas ou até incorretas, como ônibus que não aparecem no horário previsto ou não aparecem mesmo após um longo período de espera.

2.5 TRAFI

O aplicativo *Trafi*, anunciado como ‘O aplicativo de transporte mais preciso do mundo’ (TRAFI, 2016), utiliza algoritmos de aprendizado de máquina para traçar a melhor rota de acordo com a origem e destino do usuário. Ele ainda inclui rotas que dependam de integração entre diferentes veículos de transporte de massa.

Fundado na Lituânia, *Trafi* foi lançado para as cidades de Rio de Janeiro e São Paulo em 2015 (ROMER, 2015) e tem como um dos seus principais diferenciais a funcionalidade de utilização do aplicativo em modo *off-line*, que permite ao usuário armazenar no dispositivo o resultado da busca pelas rotas que poderá seguir e os horários previstos de chegada do ônibus, não necessitando uma conexão de dados no momento do uso (SALUTES, 2015).

Assim como o *Moovit*, o aplicativo fornece uma estimativa do tempo de chegada do ônibus até o local esperado, conta com a colaboração de usuários para fornecer informações em tempo real sobre trajetos, condições dos veículos e seus pontos de parada, e também fornece a opção de acompanhar o percurso de uma linha selecionada. O destaque, em relação ao seu concorrente, está em indicar a proximidade do próximo ponto por meio de um alerta vibratório.

O *Trafi* ganhou destaque entre usuários quando foi escolhido como um dos vencedores do concurso "Concurso de Transporte da Cidade Olímpica" na categoria "Planejamento de viagens" (RIO, 2016). A premiação foi organizada pela prefeitura do Rio de Janeiro para encontrar aplicativos que facilitem o transporte pela cidade antes e depois dos jogos Rio 2016.

Ao analisar a avaliação dos usuários nas lojas de aplicativos, é possível perceber que o aplicativo *Trafi* enfrenta os mesmos problemas do *Moovit*. Por se basear principalmente no

cronograma fornecido pela Prefeitura, apresentando inclusive a tabela de horários previstos no próprio aplicativo, o *Trafi* muitas vezes falha na previsão do tempo de chegada do ônibus no local. Esse erro se deve por inúmeros motivos que fazem com que os ônibus se atrasem, como acidentes, trânsito intenso, mudanças climáticas, entre outros.

2.6 VÁ DE ÔNIBUS

O “*Vá de ônibus*” é uma ferramenta fornecida pela FETRANSPOR que, quando lançada, estava disponível apenas na versão *web*. Desde seu lançamento até o ano de 2014, o site da ferramenta possuía 200 mil acessos mensais (FETRANSPOR, 2014). A aplicação cobre exclusivamente transporte público por meio ônibus e sugere o melhor itinerário entre pontos de origem e destino definidos pelo usuário utilizando-se desse meio.

A ferramenta também oferece a visão geral do itinerário de uma linha específica. Quando uma linha válida é selecionada, é indicado no mapa os percursos de ida e volta. A ferramenta também mostra no mapa a localização atual dos ônibus em trânsito daquela linha, quando disponível.

Prevendo o aumento crescente de aparelhos celulares e pensando em uma melhor utilização da ferramenta pelos usuários, a FETRANSPOR, em meados de 2014, disponibilizou a aplicativo para os sistemas *Android* e *iOS*. Com o lançamento do aplicativo, a empresa previa um aumento no acesso de 200 mil para 500 mil mensais (FETRANSPOR, 2014).

Além de contar com as funcionalidades já disponíveis no site, o aplicativo indica informações dos percursos percorrido pelas linhas do BRT¹⁰. O aplicativo também se baseia em turistas precisando se localizar na cidade. Ele lista os principais pontos turísticos do Rio, com a opção de colocar cada ponto como origem ou destino de uma nova consulta de itinerário.

Mesmo oferecendo uma solução que atenda ao usuário quando ele precisa de informações rápidas sobre qual ônibus embarcar, a página de downloads da *Google Play Store* conta com uma nota muito baixa (2,7 em um total de 5)¹¹ e inúmeras reclamações dos usuários. As principais reclamações são relacionadas ao aplicativo não executar bem as funcionalidades que se propõe, como indicar a localização dos ônibus e traçar itinerários.

¹⁰Bus Rapid Transit - sistema de transporte coletivo similar ao ônibus, porém com veículos mais extensos e uma faixa exclusiva de acesso

¹¹Dados de acesso em 08/10/2016

2.7 GOOGLE MAPS

O *Google Maps* é uma ferramenta fornecida pela Google e utilizada para identificação de rotas e auxílio à locomoção urbana para veículos particulares ou públicos, pessoas à pé e até de bicicleta. O aplicativo, também disponível em uma versão web, detém uma notória base de usuários e está instalado em mais de 1 bilhão de aparelhos *Android* (D'ONFRO, 2015) . Ele conta ainda com uma base de dados de mapas, imagens de satélite e horários de ônibus relativos a diversas cidades ao redor do mundo (GOOGLE MAPS, 2016)

O *Google Maps* foi lançado pela Google em fevereiro de 2005, após a compra da *Where 2*, empresa que desenvolveu o modelo inicial da aplicação (HUTCHEON, 2015). Desde o ano de lançamento, o aplicativo vem aprimorando sua exibição de mapas e funcionalidades fornecidas aos usuários. Em 2007, uma das funcionalidades adicionadas foi a do *Google Transit*, uma ferramenta que permite o cálculo do trajeto de uma origem até um destino, utilizando apenas transporte público (OEHLER, 2007).

No seu lançamento, o *Google Transit* era disponibilizado apenas na versão web e restrito a algumas poucas cidades. Em paralelo ao aumento da cobertura dessa funcionalidade, mapeando transportes públicos em conjunto com as companhias de transporte de cada cidade e ao aumento do uso de *smartphones*, os aplicativos para *Android* e *iOS* foram lançados. O mapeamento do *Google Transit* chegou ao Rio de Janeiro e regiões metropolitanas em setembro de 2009, terceira cidade do Brasil a receber o serviço, sucedendo apenas São Paulo e Belo Horizonte (QUINTELLA, 2009).

Embora seja o mais completo no que se refere a mobilidade urbana como um todo e também ofereça o mesmo serviço para diversos meios de transporte, o aplicativo fornecido pela Google lista os pontos da linha selecionada, mas não possui nenhum tipo de alerta para indicar ao usuário em que ponto de parada descer e não oferece informações em tempo real dos ônibus da linha. Essas características, ou falta delas, tornam o *aplicativo* uma plataforma para o planejamento de viagens e não se destaca como uma ferramenta para um acompanhamento preciso do percurso.

2.8 ANÁLISE COMPARATIVA

Todos os aplicativos mencionados possuem vantagens e desvantagens quanto à utilização e benefícios oferecidos aos usuários que dependem diariamente de transporte público. Para

entender melhor quais seriam as necessidades dos usuários e mapear quais funcionalidades estão presentes nos aplicativos, foi realizada uma pesquisa de opinião (os formulários usados na pesquisa de opinião estão disponíveis no Apêndice A). Nessa pesquisa de opinião, respondida por 12 usuários, foi perguntado quais os aplicativos eram mais utilizados para se locomoverem de ônibus pela cidade do Rio e quais funcionalidades os aplicativos possuem ou não. Por último, foi feita uma pergunta aberta questionando qual funcionalidade os usuários mais sentem falta nos aplicativos que utilizam. Todas as respostas a essa última questão, respondida por sete usuários, indicavam necessidade de maior transparência relativa à localização e aos horários dos ônibus.

Dessa forma, avaliando as funcionalidades já existentes nos aplicativos disponíveis, as sugestões de melhorias dos usuários da pesquisa de opinião, e, ainda, pensando nas *personas* citadas na Seção 1.2, este trabalho elencou as funcionalidades mais esperadas em ferramentas desta categoria.

Cada uma dessas funcionalidades é listada a seguir, comentando também sua utilização, benefícios e presença nos aplicativos. Além disso, um quadro comparativo de todas elas é apresentado na Tabela 1, destacando em quais aplicativos elas estão ou não disponíveis.

- 1. Sugere itinerários com base na origem e no destino:** buscar qual a rota e meios de transporte possíveis para chegar ao ponto de destino é uma funcionalidade útil para qualquer pessoa que precisou chegar em algum local que nunca havia ido antes. Por esta necessidade ser comum, todos os aplicativos analisados oferecem opções de itinerários, uma vez informados origem e destino.
- 2. Estima de tempo de chegada do ônibus ao ponto:** uma das funcionalidades mais abordadas pelos usuários nas avaliações dos aplicativos é ter uma previsão de quanto tempo um ônibus demorará até chegar no ponto que ele está. Sabendo desta necessidade, os aplicativos Trafi, Moovit e Google Maps fornecem um tempo estimado de chegada do ônibus mais próximo daquela linha, baseado em informações de horários passadas pela prefeitura de cada cidade. Entretanto, pela falta de linhas exclusivas para ônibus na cidade e pelo trânsito intenso, muitas vezes esta estimativa é falha, gerando insatisfação dos usuários.
- 3. Indica pontos de parada da linha escolhida:** esta funcionalidade está presente em todos os aplicativos, exceto no fornecido pela FETRANSPOR, o *Vá de Ônibus*. A funcionalidade auxilia os usuários a terem conhecimento de todo o trajeto do ônibus,

possibilitando estimar a duração do percurso. O conhecimento sobre o trajeto permite evitar linhas que passem por locais perigosos e auxilia na segurança.

4. **Alertas sonoros e/ou de vibração:** os alertas sonoros ou de vibração do celular funcionam em conjunto com a indicação dos pontos da linha que o usuário está no momento. O aplicativo emite uma notificação quando o ônibus está se aproximando do(s) ponto(s) de parada escolhido(s). Essa funcionalidade auxilia na segurança do passageiro, que não precisa ficar com o celular à vista, olhando a tela durante o percurso, para saber em que momento deverá dar sinal ao motorista para descer. Dos aplicativos analisados, apenas o *Trafi* conta com alertas vibratórios. Contudo, nenhum deles oferece a opção de alertas sonoros.
5. **Informações sobre as condições da linha e/ou do trajeto em tempo real:** por serem aplicativos exclusivos para mobilidade por meio de transporte público e com obtenção de informações via *crowdsourcing*¹², essa funcionalidade é vista apenas no *Moovit* e *Trafi*. Esta funcionalidade é útil para usuários que desejam saber as condições de acessibilidade do ponto, lotação do veículo, entre outros.
6. **Fornece informações específicas para pontos turísticos:** todos os quatro aplicativos analisados possuem ícones no mapa indicando pontos turísticos mais famosos. Entretanto, apenas um deles, o “Vá de ônibus”, tem um serviço específico para o público buscando esses locais. A funcionalidade é útil para usuários explorando a cidade, que não sabem ao certo a região da atração que gostariam de ir.
7. **Indica a localização dos ônibus de uma linha, atualizando em tempo real:** essa funcionalidade permite ao usuário estimar por conta própria o tempo que o ônibus da linha desejada demorará. Como essa estimativa não é automática, é possível ponderar o tempo de acordo com o trânsito daquele horário. O aplicativo “Vá de ônibus” se propõe a indicar essa localização ao usuário, mas, de acordo com as avaliações dos usuários, essa funcionalidade é oferecida com muitas falhas.
8. **Compara proximidade de duas ou mais linhas de ônibus no mapa:** esse comparativo é útil para usuários que tem duas opções de ônibus com trajetos diferentes para o mesmo destino. Essa funcionalidade auxilia na decisão de esperar

¹² Processo para obtenção de recursos que utiliza a própria comunidade.

um próximo ônibus ou não. Porém, nenhum dos aplicativos encontrados possui esta funcionalidade.

9. Notifica ao usuário quando o ônibus escolhido alcança uma certa proximidade:

também visando a redução de exposição ao risco de roubos, essa funcionalidade emite um alerta ao usuário quando o ônibus estiver a uma certa distância dele. Nenhum aplicativo analisado oferece essa função, que além do risco, também reduz o tempo gasto do usuário esperando no ponto.

Tabela 1 - Funcionalidade por aplicativo

#	Funcionalidade	Google Maps	Moovit	Vá de ônibus	Trafi	Total
1.	Sugere itinerários com base na origem e no destino					4
2.	Estima tempo de chegada do ônibus ao ponto			X		3
3.	Indica pontos de parada da linha escolhida			X		3
4.	Alertas sonoros e/ou de vibração	X	X	X		1
5.	Informações sobre as condições da linha e/ou do trajeto em tempo real	X		X		2
6.	Fornece informações específicas para pontos turísticos	X	X		X	1
7.	Indica localização dos ônibus daquela linha, atualizado em tempo real	X	X		X	1
8.	Compara proximidade de duas ou mais linhas de ônibus no mapa	X	X	X	X	0
9.	Notifica ao usuário quando o ônibus escolhido alcança uma certa proximidade	X	X	X	X	0

2.9 CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentados aplicativos que têm como objetivo traçar rotas e auxiliar no percurso dos usuários de transporte público na Cidade do Rio de Janeiro. Também foram apresentadas ferramentas que auxiliam esse ramo de aplicativos, sendo elas o GPS e para o Vá de Ônibus, o *Data.rio*. Ao final foi exibido um quadro comparativo das principais funcionalidades que esses oferecem aos usuários.

Apesar dos aplicativos já contribuírem no dia a dia dos usuários de transporte público, ainda existem pontos reportados por estes mesmos usuários que precisam ser adicionados ou aperfeiçoados.

CAPÍTULO 3 – TRACKBUS

3.1 INTRODUÇÃO

Por meio dos aplicativos levantados no Capítulo 2 e da análise das suas respectivas funcionalidades feita na Seção 2.8, foi possível notar a falta de algumas ferramentas que seriam úteis ao usuário. Mesmo com o incentivo governamental e espaço aberto para desenvolvimento, existem funcionalidades que não estão disponíveis para os usuários nos aplicativos levantados.

A proposta deste trabalho é elaborar como prova de conceito um aplicativo que visa auxiliar diferentes necessidades e perfis de usuários de ônibus na Cidade do Rio de Janeiro. Desta forma, este aplicativo deve ter funcionalidades diferentes das encontradas em outros, mas que sejam relevantes para os usuários de ônibus. Contudo, com o propósito de fornecer uma experiência de utilização adequada e possibilitar a avaliação da prova de conceito, algumas das funcionalidades que já existem em outros *aplicativos* também precisaram ser implementadas.

Este capítulo descreve a abordagem proposta por este trabalho. A Seção 3.2 apresenta uma visão geral dos requisitos avaliados no Capítulo 2. Os requisitos que foram implementados são descritos detalhadamente nas Seções 3.3 a 3.8. Por fim, a Seção 3.9 apresenta algumas considerações finais sobre os requisitos apresentados, e disponibiliza um link para download do *TrackBus* na *Google Play Store*.

3.2 LISTAGEM DE REQUISITOS

A partir das *personas* listadas na Seção 1.2 e dos aplicativos analisados no Capítulo 2, foram pensadas algumas funcionalidades úteis a um aplicativo que preza pelo auxílio aos usuários que dependam de ônibus para se locomover em grandes cidades. Além dos requisitos diretamente envolvidas com as *personas*, também foram enumerados requisitos que por padrão já fazem parte de quase todos aplicativos do ramo.

A fim de validar a necessidade destes requisitos para os usuários, foi realizada uma pesquisa de opinião, identificando se os mesmos seriam úteis ou não. As perguntas foram feitas por meio do formulário que pode ser encontrado no Apêndice B – Formulário de mapeamento de funcionalidades do *TrackBus*. Esta pesquisa de opinião contou com a participação de 98

usuários, com variados perfis. Na pesquisa foi solicitado que o participante classificasse cada um dos requisitos listados na Tabela 2 com um nível de relevância, respondendo de acordo com a sua necessidade, se ele seria útil ou não caso estivesse utilizando o aplicativo. Esta relevância foi indicada com base em uma escala Likert (1932) de cinco níveis, considerando 1 como uma funcionalidade irrelevante e 5 como algo fundamental em um aplicativo com o foco proposto por este trabalho.

Tabela 2 - Requisitos pesquisa de satisfação

#	Requisito
R1.	Sugerir rotas de viagem de acordo com um ponto de início e destino
R2.	Estimar tempo de chegada de um ônibus a um ponto
R3.	Indicar a distância em metros de um ônibus à sua localização atual
R4.	Indicar a localização, em tempo real, dos ônibus de uma linha
R5.	Comparar proximidade de ônibus de diferentes linhas no mapa
R6.	Fornecer endereço e nome de pontos turísticos próximos à uma linha selecionada
R7.	Notificação de quando o ônibus estiver se aproximando
R8.	Notificação da proximidade de um ponto da linha
R9.	Notificação de pontos turísticos próximos dos pontos de parada da linha

A pesquisa também incluiu duas perguntas para mapear o perfil do usuário que estava fazendo aquela classificação: se ele vive ou não na cidade do Rio ou regiões metropolitanas (Figura 1) e qual sua frequência de utilização de ônibus na cidade (Figura 2).

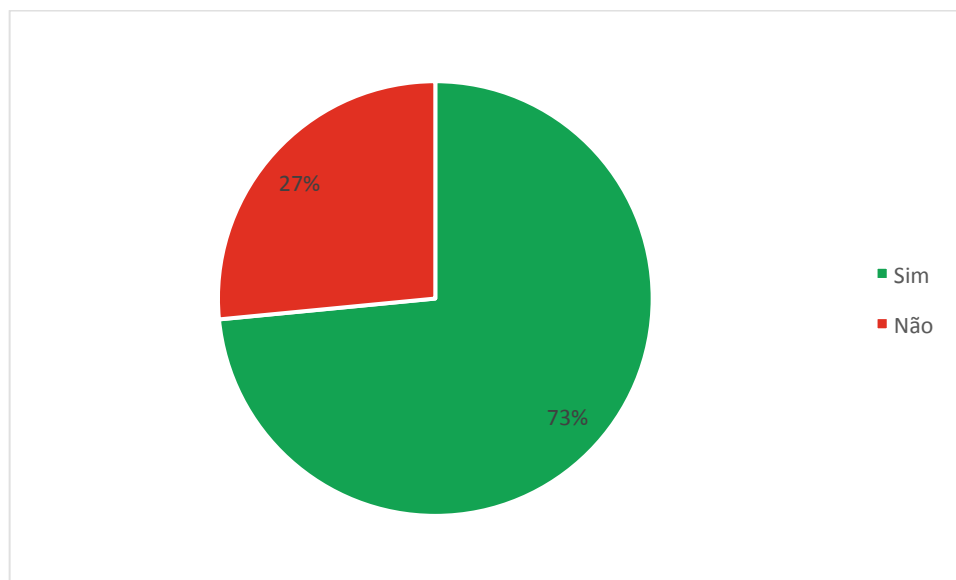


Figura 1 – Percentual de usuários que vivem na cidade do Rio e regiões metropolitanas¹³

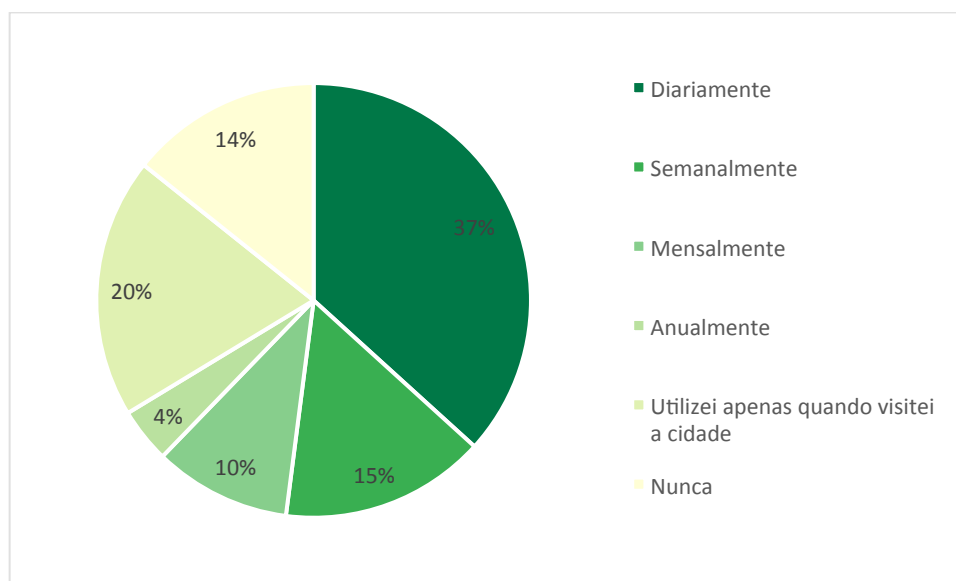


Figura 2 – Frequência de utilização de ônibus pelos usuários

¹³ Todas as escalas de cores utilizadas nos gráficos deste capítulo foram retiradas de <http://colorbrewer2.org/>

Estas duas perguntas foram importantes para filtrar os resultados em dois grandes grupos, o de *Moradores* (64 usuários) e o de *Turistas* (16 usuários). Os resultados podem ser vistos na Figura 3 e Figura 4, respectivamente. Todas as respostas cuja frequência de utilização foi apontada como “Nunca” foram desconsideradas da análise, visto que o objetivo do aplicativo proposto por este trabalho é atingir usuários reais de ônibus, mesmo que esporádicos.

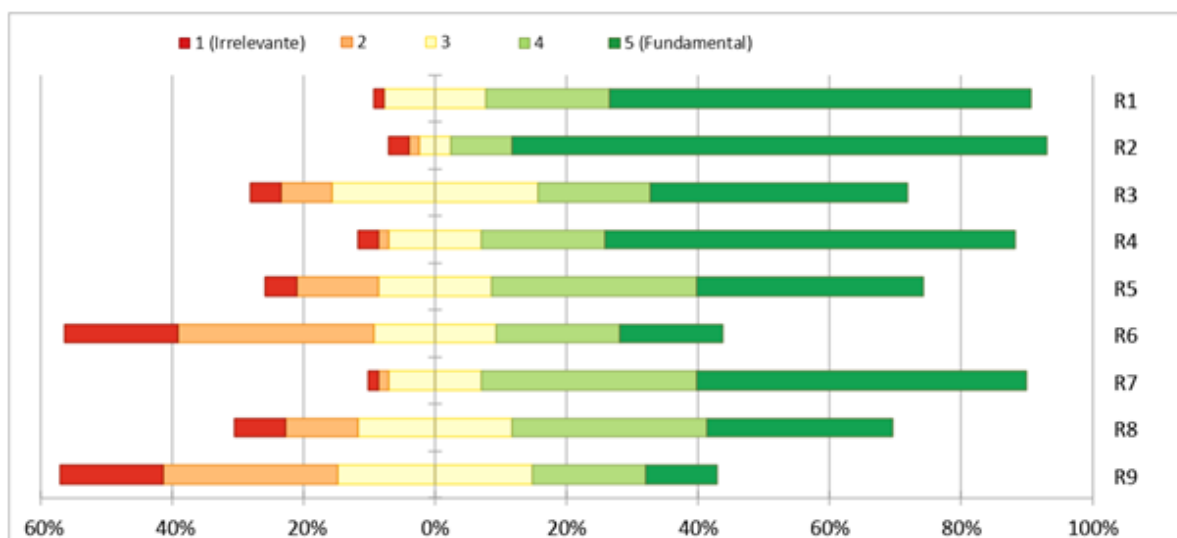


Figura 3 - Moradores - Relevância de requisitos

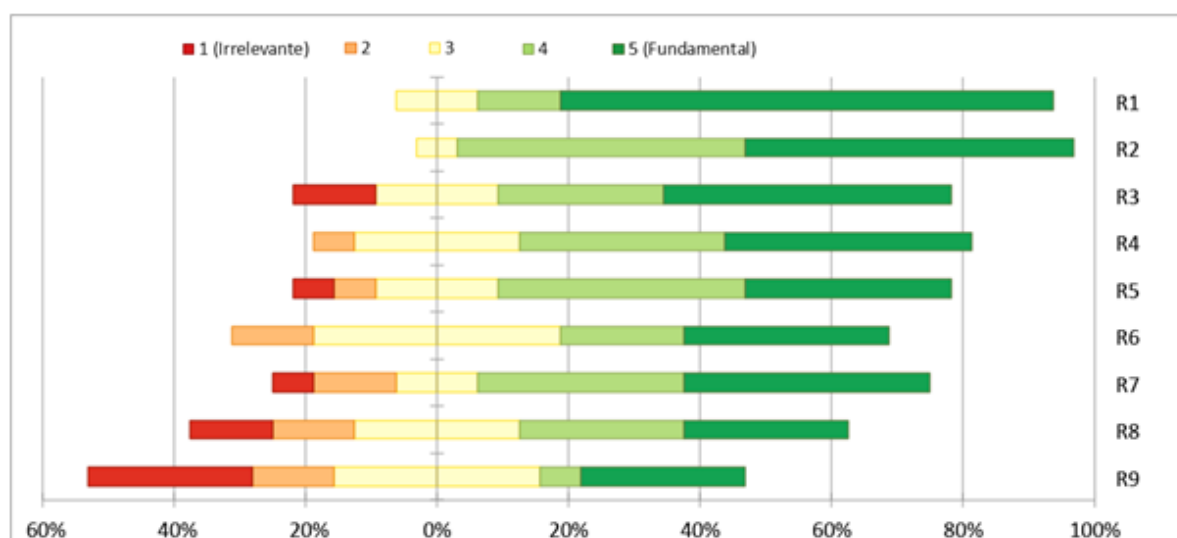


Figura 4 - Turistas - Relevância de requisitos

Analisando os dois gráficos e os resultados de cada um dos requisitos, podemos notar que R1 e R2 são fundamentais para a maioria dos usuários, independente do grupo em que se enquadram. Entretanto, vimos no Capítulo 2 que estas duas funcionalidades estão presentes na maioria dos aplicativos disponíveis atualmente para *download*. Como este trabalho se propõe a implementar novas funcionalidades em aplicativos no ramo de transporte de ônibus, como prova de conceito, foi uma decisão de projeto não incluir funcionalidades já comuns à maioria deles, e que possuíam um alto esforço de implementação. Entendemos que a nossa prova de conceito pode vir a ser incorporada aos aplicativos existentes, caso os resultados obtidos sejam satisfatórios.

É possível observar também no gráfico que os requisitos R3, R4 e R5 são considerados importantes para ambos os grupos de usuário que responderam ao questionário. Estes requisitos representam as funcionalidades principais que são incluídas no *TrackBus* e serão melhor detalhadas nas seções a seguir.

Se analisarmos apenas a Figura 3, podemos inferir que de todos os requisitos analisados, apenas R6 e R9 são considerados irrelevantes por muitos usuários. Mesmo assim, estes dois requisitos foram avaliados como fundamentais por um pequeno grupo de usuários. Ambos são diretamente relacionados a informações sobre pontos turísticos, que não agregam valor a moradores da cidade no dia-a-dia. Contudo, por outro lado, ao analisar os resultados da Figura 4, podemos ver que este ponto se torna fundamental para um maior número de usuários, uma vez que as respostas foram fornecidas por turistas da cidade do Rio.

Os requisitos R7, R8 e R9 motivaram a criação de um outro bloco de perguntas de relevância na pesquisa com usuários, relacionadas aos tipos de notificação. Como as notificações podem se apresentar de formas distintas, além da notificação padrão na tela, foi solicitado que os usuários selecionassem na mesma escala dos requisitos, a relevância de três tipos diferentes de alertas: vibratório, sonoro e por voz. Os resultados, separados pelos dois grupos definidos anteriormente, são exibidos na Figura 5 e na Figura 6.

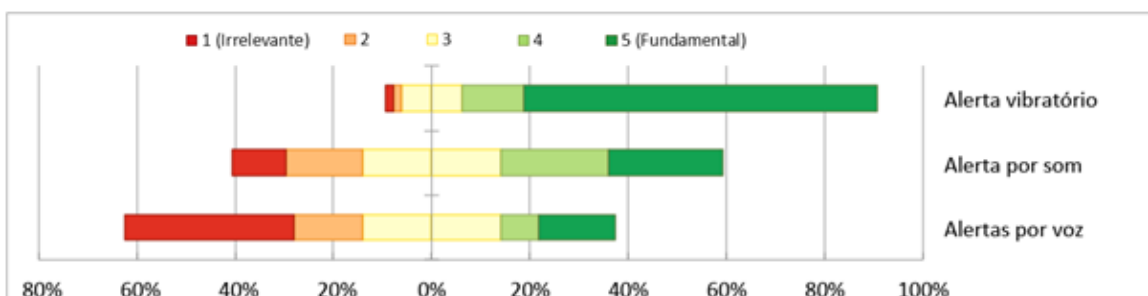


Figura 5 - Moradores - Relevância de alertas

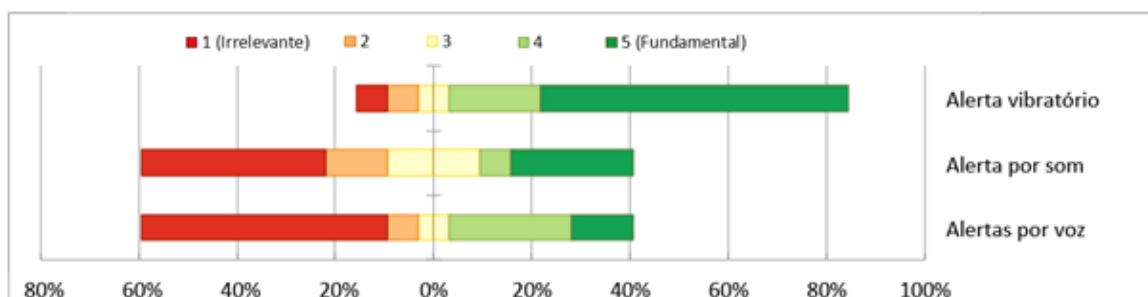


Figura 6 - Turistas - Relevância de alertas

O único requisito que não se destacou em nenhum dos grupos foi o de “notificações por voz”. Porém, por ter sido considerado fundamental por alguns poucos participantes e por seu custo de desenvolvimento ser baixo, foi decidido incluí-lo também no projeto. Por este motivo, relevância considerável e baixo custo de implementação, algumas funcionalidades já existentes em aplicativos disponíveis também foram incorporadas ao *TrackBus*, agregando um maior valor ao protótipo funcional.

Com a pesquisa, foi possível observar que todos os requisitos levantados são considerados fundamentais para, pelo menos, um grupo pequeno de pessoas, gerando uma média positiva na maioria deles. Nas próximas seções cada uma das funcionalidades implementadas será discutida com mais detalhe.

Atualmente, a maioria dos aplicativos no mercado, com o mesmo propósito deste trabalho, oferece uma solução comum para busca de linhas e itinerários. O usuário, para encontrar linhas que atendam à sua necessidade, precisa fornecer os locais de origem e destino da sua viagem, para então serem apresentadas opções de rotas e linhas possíveis.

O *TrackBus* propõe uma abordagem que oferece uma solução alternativa a quem já sabe quais opções de linhas o atendem. Existem situações em que o usuário já está acostumado a

frequentemente pegar uma mesma linha, e não precisa todos os dias inserir origem e destino, uma vez que ele deseja apenas saber a distância que um ônibus daquela linha está dele.

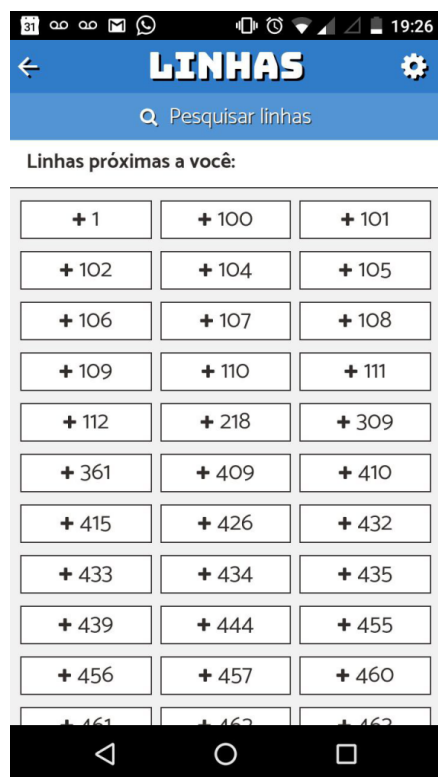


Figura 7 - Listagem de linhas próximas

Como pode ser visto na **Figura 7**, o aplicativo oferece um conjunto de linhas filtradas de acordo com a proximidade dos ônibus da linha ao usuário. Todas as linhas que possuem pelo menos um ônibus dentro de um raio de 2 km do usuário são exibidas. Esta distância não é configurável e foi considerada razoável para o usuário ter uma visão geral de linhas próximas a ele, sem precisar consultar o mapa. Caso o usuário não encontre a linha desejada entre as filtradas, também é possível buscar pelo número da linha dentre todas disponíveis.

Este filtro de linhas de ônibus foi preferido à escolha por texto livre como uma forma de mitigação de erros de digitação. Além disto, alguns números de linhas fornecidas pelo *Data.rio* são descritas de forma diferente do que é exibido no visor dos ônibus, o que pode causar confusão ou simplesmente impedir o usuário de encontrar a linha que deseja. O filtro ocorre de forma dinâmica, enquanto o usuário digita o número da linha, procurando reduzir problemas como o citado anteriormente.

A funcionalidade de filtrar linhas próximas foi sugerida por alguns dos participantes que responderam à pesquisa de opinião apresentada na Seção 3.2. Uma das perguntas da pesquisa

pedia ao participante para dizer qual funcionalidade seria relevante para ele, mas que não foi listada nos requisitos iniciais. Um deles indicou “Linhas que passam pelo lugar onde estou”, o que é a principal característica desta funcionalidade, além de outras sugestões que também são atendidas com a aplicação desta listagem.

3.3 EXIBIÇÃO DE DUAS OU MAIS LINHAS NO MAPA

Esta funcionalidade foi pensada baseando-se principalmente na *persona* da Seção 1.2.2, e em usuários que passam por situações parecidas. Diariamente alguns usuários de transporte público tem a possibilidade de pegar duas ou mais linhas de ônibus para chegarem no destino desejado. É esperado que estas linhas cumpram trajetos diferentes a partir de determinados pontos, que tenham frequência e número de ônibus em trânsito diferentes.

Estas diferença no trajeto indicam que provavelmente uma das linhas chegará antes no destino final. Porém, pode ser que a frequência de ônibus desta linha seja menor do que da outra linha, o que aumenta o tempo de espera pelo ônibus da linha com o trajeto mais curto. Pensando em possibilidades como esta, o *TrackBus* implementa uma funcionalidade que permite ao usuário visualizar mais de uma linha no mapa ao mesmo tempo, como pode ser observado na Figura 8. Com o resultado da busca, é atribuído a cada linha um ícone de cor diferente para ser exibido no mapa como pode ser visto na Figura 9.



Figura 8 - Listagem de linhas

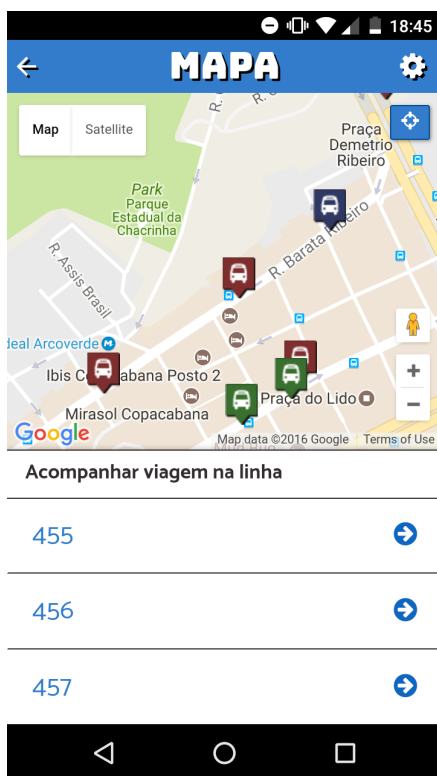


Figura 9 – Listagem dos ônibus no mapa

Este formato de exibição possibilita o usuário ponderar e decidir se será mais vantajoso esperar o ônibus da linha com o trajeto mais rápido ou se deve embarcar logo no ônibus da linha com o percurso mais longo, mas que já está se aproximando.

O aplicativo permite que o usuário selecione até três linhas no mapa. Esta decisão foi tomada sob duas óticas: uma em relação à usabilidade do aplicativo e outra analisando as possibilidades de escolha no mundo real. Na primeira, caso o usuário selecione um número muito grande de linhas para aparecer no mapa, a visualização clara destas linhas pode ser prejudicada, impactando na escolha dos ônibus e até no desempenho. Já na segunda, assumimos que é incomum existir mais de três linhas ativas que atendam o usuário para o mesmo destino.

Além disto, também incluímos informações adicionais para facilitar a utilização do aplicativo pelo usuário, como o número da linha do ônibus selecionado e a distância deste para usuário.

3.4 PROXIMIDADE EM METROS

Conforme mencionado no Capítulo 2, uma das maiores reclamações nas páginas dos aplicativos existentes hoje é a falha na estimativa temporal de chegada do ônibus ao ponto. Esta estimativa fornecida é muitas vezes baseada nos horários previstos no cronograma de chegada daquele ônibus naquele ponto. Porém, em uma cidade com trânsito tão variável como o do Rio de Janeiro, estes horários são difíceis de serem cumpridos.

Para dar mais transparência ao usuário e diminuir esta insatisfação relativa à estimativa do tempo, o *TrackBus* exibe a distância, em metros, de um ônibus ao usuário. Entendemos que este é um processo que regride alguns passos na automatização e implantação de facilidades ao usuário, que não precisaria estimar ele mesmo quanto tempo o ônibus levará para chegar ao ponto que deseja. Entretanto, com as ferramentas disponíveis hoje, e com a estrutura do sistema de ônibus na cidade, uma maneira de fornecer uma informação mais relevante e verídica ao usuário é por meio da distância real daquele ônibus a ele. Com esta informação, e sabendo o horário, o dia da semana e observando as condições do trânsito naquele momento, é possível que o usuário, principalmente o que faz o mesmo trajeto diariamente, calcule aproximadamente quando ele deverá se dirigir ao ponto para alcançar o ônibus.

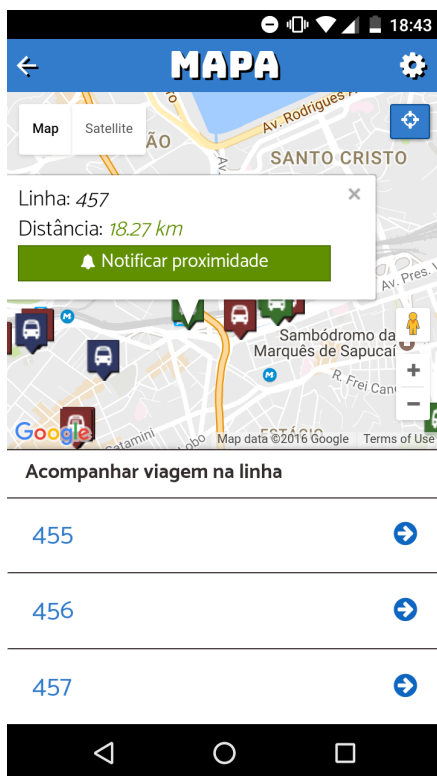


Figura 10 - Proximidade em metros do ônibus

O aplicativo permite que o usuário veja na tela a redução da distância do ônibus até a sua localização, como é mostrado na Figura 10. Mas ainda, é possível programar uma notificação para alertar o usuário quando o ônibus estiver se aproximando.

Da maneira que a funcionalidade foi implementada, uma vez que o usuário requisita ser notificado da proximidade de um ônibus, a cada ciclo de atualização de sua posição geoespacial, a distância entre ele e cada ônibus escolhido é calculada. Caso esta distância seja menor ou igual ao parâmetro que indica a distância mínima para ser notificado, o usuário recebe uma notificação no seu dispositivo. A distância em metros para esta notificação pode ser definida pelo usuário na tela de configurações, que será discutida na Seção 3.7.

3.5 LISTAGEM DOS PONTOS DE ÔNIBUS

Uma das funcionalidades atendidas é a listagem de todos os pontos de ônibus de uma linha. Esta funcionalidade tem o objetivo de auxiliar usuários em duas diferentes situações: (1) quando o usuário está buscando por uma linha que passe por um ponto próximo do destino que precisa e (2) quando o usuário já sabe qual linha deve pegar e o nome do ponto que precisa descer, baseado em uma pesquisa anterior ou informações de amigos, entretanto não sabe quando chegará neste ponto durante a viagem.

A listagem de pontos pode ser vista quando é selecionada uma linha qualquer. Todos seus pontos são marcados no mapa, e seus respectivos nomes são listados, como pode ser visto na Figura 11.

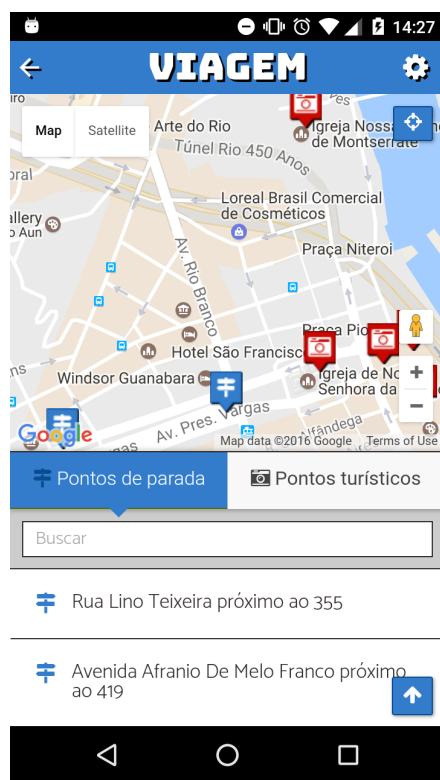


Figura 11 - Listagem e exibição dos pontos de uma linha

A primeira situação é atendida tanto com o mapa, caso o usuário saiba previamente a região geográfica que precisa ir, quanto com a listagem nominal dos pontos, caso ele saiba o nome e número do destino. Além do nome dos pontos, marcadores indicando sua localização também são exibidos no mapa, assim como os de pontos turísticos que serão comentados na próxima seção.

Já a segunda situação permite que o usuário acompanhe a sua localização sendo atualizada no mapa, e se aproximando do ponto que precisa descer. Entretanto, ela é melhor atendida quando utilizada em conjunto com a funcionalidade de acompanhamento de viagem e suas respectivas notificações, que são detalhadas nas seções 3.7 e 3.8, respectivamente.

A requisição para a listagem de pontos de parada, assim como turísticos, citados na Seção 3.6, não é feita ao servidor do *Data.rio*. Ambas são realizadas ao sistema de arquivos local. A forma como estes arquivos são gerados será comentada na Seção 4.8.

3.6 PONTOS TURÍSTICOS

Segundo o aplicativo *TripAdvisor*¹⁴, o Rio de Janeiro é a cidade do Brasil preferida por turistas. Além desta preferência, a cidade também vem ganhando grande evidência mundial nos últimos anos, sediando eventos importantes em diversos âmbitos da sociedade, como a Jornada Mundial da Juventude, Rio +20, Copa do mundo e Olimpíadas. O aumento do número turistas, tem como consequência um paralelo aumento do número de pessoas que não estão acostumadas com os meios de transporte público, mas ainda assim gostariam de utilizar os mesmos para se locomover pela cidade.

Pensando nestas pessoas, este trabalho propõe uma forma de encontrar pontos turísticos, a quem deseja explorar as atrações que a cidade oferece. Além das informações de localização dos ônibus, indicação da proximidade de pontos já mencionados, que agregam tanto ao turista quanto ao usuário cotidiano.

¹⁴<https://www.tripadvisor.com.br/> - Aplicativo que mapeia e fornece um sistema de avaliação para cidades, restaurantes, atrações turísticas, entre outros, em sua maioria fornecidas por viajantes.



Figura 12 - Listagem e exibição de pontos turísticos

Utilizando o *TrackBus* é possível que o usuário, ao selecionar uma linha, veja todos os pontos turísticos em que um ônibus daquela mesma linha parará próximo. Esta listagem pode ser vista na Figura 12. Como explicado na Seção 3.2, inicialmente apenas as linhas que têm ônibus próximos à localização atual do usuário são listadas. Desta forma, é possível analisar se o ponto turístico que usuário deseja ir está na listagem, indicando que uma determinada linha atenderá sua requisição.

Além disto, esta listagem de pontos turísticos também auxilia em uma definição de roteiro mais eficiente. Uma vez que a linha passe por mais de uma atração turística, é possível que o visitante se organize e defina qual visitar primeiro. Ou ainda, a funcionalidade possibilita a descoberta de atrações que às vezes não estavam no roteiro, mas que pela proximidade com outra que estava, pode ser visitada.

Entretanto, caso o usuário não queira receber notificações, ou mesmo visualizar no mapa os pontos das atrações turísticas, é possível remover esta funcionalidade na tela de configurações.

3.7 ACOMPANHAMENTO DE VIAGEM

Não só os turistas, mas também os moradores da cidade precisam ir a lugares sem saber onde devem descer do ônibus para chegar. A maioria dos usuários frequentes ou esporádicos de ônibus já passou por situações em que sabiam qual linha deveriam embarcar, mas tinham apenas uma referência de onde descer. Antes do avanço da tecnologia (isto é, GPS) este problema era resolvido solicitando ajuda para passageiros do ônibus, trocadores ou motoristas. Porém, já existem mecanismos que possibilitam que este processo seja automatizado, não dependendo de pessoas, que as vezes podem esquecer de informar ou não conhecer o destino desejado.

O *TrackBus* oferece a opção de ‘Acompanhar viagem’, como pode ser vista na Figura 9, que quando ativada indica ao aplicativo que o usuário embarcou em um ônibus daquela determinada linha. A partir deste momento, o aplicativo começa a fazer verificações de proximidade aos pontos de ônibus daquela linha e aos pontos turísticos próximos destes pontos de ônibus. Se acionado pelo usuário, quando a sua localização, fornecida pelo GPS do seu *smartphone*, começa a se aproximar de algum dos pontos da linha, é mostrada na tela uma notificação indicando a distância do usuário ao ponto mais próximo. Esta distância mínima para notificação também pode ser configurada pelo usuário, como pode ser visto na Figura 13, de acordo com o que melhor o convém.



Figura 13 – Configurações de distância para notificação

Esta abordagem possui uma utilização interativa das listagens discutidas nas seções 3.5 e 3.6. Além de apenas visualizar os pontos importantes daquela linha, o usuário pode ser alertado de quando estes estão se aproximando. Este alerta também reduz a exposição do *smartphone* do usuário durante o percurso, vantagem que é discutida na Seção 3.8.

3.8 NOTIFICAÇÕES

Não só o número de turistas vem crescendo no Rio de Janeiro, mas também o aumento nos índices de criminalidades vem assustando cada vez mais moradores e visitantes. Segundo dados do ISP (Instituto de Segurança Pública) do Rio de Janeiro, os roubos de rua (são considerados: roubo a transeunte, roubo de aparelho celular e roubo em coletivo) cresceram em 34,1% no acumulado de janeiro a julho de 2016, relativo ao mesmo período no ano anterior (NASCIMENTO, 2016).

Para a utilização de aplicativos de transporte público pela cidade, é necessário estar com o celular e deixá-lo à vista quando é se deseja checar se está se aproximando do destino, ou se o ônibus está chegando. Esta utilização expõe o usuário a um possível roubo, que não é incomum

na cidade. Pensando em uma forma de tentar proteger um pouco o usuário desta exposição, foram introduzidos mecanismos de notificação ao *TrackBus*.

Com o *TrackBus*, o usuário pode configurar três tipos de notificações antes e durante o deslocamento:

- Alertas apenas vibratórios
- Alertas com som
- Alertas por voz

As notificações podem ser utilizadas para alertar sobre a proximidade do ônibus quando o usuário está no ponto. Quando este embarcou, podem ser configuradas notificações de aproximação de pontos de ônibus e pontos turísticos de interesse. Ao selecionar o modo de navegação “Acompanhar viagem”, o usuário pode também selecionar sobre quais os pontos deseja ser notificado. Um exemplo do funcionamento desta seleção pode ser observado na Figura 14.

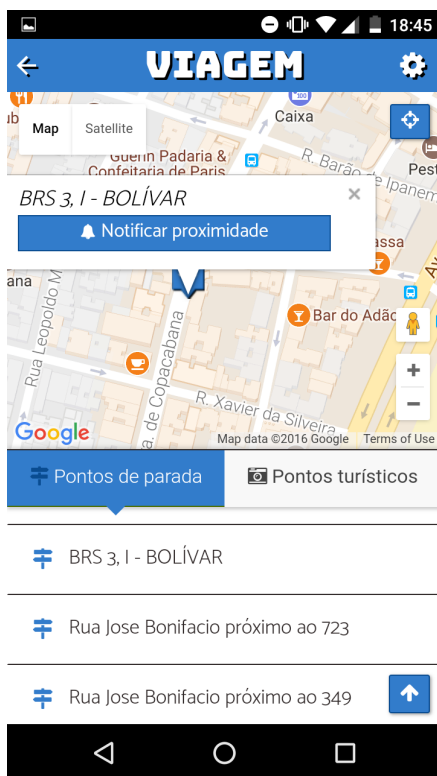


Figura 14 – Notificação de proximidade do ponto

Desta forma, o usuário pode buscar ainda em um local seguro a linha que deseja e selecionar o ponto desta que deseja descer, ou alguma atração que deseja visitar. Suponhamos

que este usuário tenha selecionado o tipo de notificação por voz, que apresenta um texto descritivo da notificação. Quando ele observar na tela que o ônibus está se aproximando, pode guardar o aparelho e utilizar apenas fones de ouvido, que o expõem menos do que andar com o celular à vista. Quando atingir determinada localização, o usuário ouve uma notificação por voz que seu ônibus está chegando e uma vez andando no ônibus, recebe a notificação indicando que o ponto selecionado por ele é a próxima parada.

3.9 CONSIDERAÇÕES FINAIS

Neste capítulo foi apresentada a abordagem proposta por este trabalho. Foram apresentadas suas funcionalidades uma a uma e discutidos os benefícios que cada uma delas proporciona aos usuários. O TrackBus já está disponível para download na *Google Play Store*: https://play.google.com/store/apps/details?id=com.uff.trackbus&hl=pt_BR. Apesar de poder ser disponibilizado também para o sistema operacional *iOS*, o alto custo de obtenção da licença de desenvolvedor exigido pela empresa Apple impossibilitou o lançamento do aplicativo na loja do sistema operacional. A descrição detalhada da implementação será melhor comentada no Capítulo 4.

CAPÍTULO 4 – DECISÕES DE DESENVOLVIMENTO E IMPLEMENTAÇÃO

4.1 INTRODUÇÃO

Este capítulo tem como objetivo discutir com mais detalhes como o aplicativo e suas respectivas funcionalidades, discutidas no Capítulo 3, foram implementadas. O *TrackBus* é um aplicativo híbrido, ou seja, portátil para dispositivos compatíveis com sistemas operacionais *Android* e *iOS*. A Seção 4.2 apresenta uma visão geral com vantagens e desvantagens da escolha deste tipo de desenvolvimento. As seções seguintes, 4.3, 4.4 e 4.5, detalham todos os *frameworks* utilizados no desenvolvimento.

Em seguida, nas seções 4.6, 4.7 e 4.8 serão discutidas a arquitetura do sistema e quais bibliotecas e *scripts* de pré-processamento foram utilizadas para auxiliar no desenvolvimento das funcionalidades, respectivamente. A estrutura dos dados utilizados pelo aplicativo é descrita na Seção 4.9, enquanto na Seção 4.10 são discutidos os detalhes da implementação de cada funcionalidade. Por fim, na Seção 4.11 é fornecida a conclusão, expondo alguns pontos positivos e negativos do projeto.

4.2 O APLICATIVO HÍBRIDO

O *TrackBus* é um aplicativo que se destina a diversos perfis de usuários. E exatamente por serem muitos perfis, seria ideal o aplicativo estar disponível para os sistemas operacionais mais populares: *Android* e *iOS*. Porém, cada sistema possui uma linguagem e ambientes de desenvolvimento diferentes (*Android* utiliza *Java*, enquanto o *iOS* utiliza *Objective-C* ou *Swift*). Para fazer um aplicativo nativo para cada uma das plataformas seria necessário desenvolvê-lo em duas linguagens diferentes, cada uma com suas bibliotecas específicas e processos de implantação distintos. Além do tempo de aprendizado das linguagens, o tempo total de desenvolvimento passa a ser maior e a manutenção também é mais complexa, uma vez que qualquer código deverá ser feito em ambas as linguagens de forma separada e podem existir comportamentos específicos de cada plataforma.

Entretanto é possível utilizar *frameworks*, baseados em uma linguagem em comum, geralmente *JavaScript*, para gerar o aplicativo correspondente para cada plataforma.

Na, Tabela 3 é exibido um quadro comparativo de vantagens e desvantagens de cada uma das alternativas de desenvolvimento, nativa e híbrida.

Tabela 3 - Desenvolvimento nativo vs. híbrido

#	Característica	Desenvolvimento Nativo	Desenvolvimento Híbrido
1	Custo	Toda implementação e manutenção deve ser feita para cada plataforma, tornando o custo de desenvolvimento alto	Implementação única ¹⁵ para todas as plataformas com tecnologias muito difundidas na web, diminuindo o custo de desenvolvimento
2	Reusabilidade de código e Portabilidade	Aplicações funcionam apenas para a plataforma em que foram desenvolvidas	A aplicação é portátil para qualquer plataforma suportada pelo framework híbrido
3	Acesso a funcionalidades do aparelho	Acesso direto a todas as funcionalidades nativas	Permite acesso a todas as funcionalidades por meio de APIs do framework
4	Interface com o usuário	Componentes de interface nativos, similares a outras aplicações do aparelho	Interface similar à nativa
5	Desempenho	Tem acesso direto às funcionalidades da plataforma, são executadas como aplicativos nativos, resultando em melhor desempenho.	São executadas por meio de um WebView e possuem camadas adicionais para realizar integrações nativas, prejudicando o desempenho

Analisando a Tabela 3 em ordem de importância das funcionalidades nela listadas, temos primeiramente que considerar qual das duas alternativas forneceria maior alcance de usuários. Com o desenvolvimento nativo conseguimos isto com o custo de desenvolver dois aplicativos iguais para plataformas diferentes, o que é mais complexo do que desenvolver apenas um aplicativo para todas as plataformas, como é feito no desenvolvimento híbrido.

A integração com as funcionalidades nativas do aparelho é extremamente necessária, porém ambas as alternativas fornecem esta funcionalidade de forma satisfatória, o mesmo acontece com a interface com o usuário.

Por fim, o desempenho do aplicativo não é um ponto de muita importância, pois apesar de serem realizados cálculos geoespaciais complexos, estes não são feitos com frequência.

Tendo considerado estes pontos, optamos por implementar o *TrackBus* utilizando *frameworks* para desenvolvimento híbrido.

¹⁵ Trechos de código podem ser específicos por plataforma

4.3 APACHE CORDOVA

Um dos *frameworks* escolhidos para utilização no desenvolvimento do *TrackBus* foi o *Apache Cordova*¹⁶. O *Cordova* é um *framework open-source* para desenvolvimento de aplicativos móveis que permite a utilização de tecnologias padrão na *web* como *HTML* (*HyperText Markup Language*), *CSS* (*Cascading Style Sheets*) e *JavaScript*. Por meio desse *framework* é possível acessar funções nativas dos dispositivos, como, por exemplo, sensores, câmera, lista de contatos, entre outros. O acesso é realizado por meio de uma *API JavaScript* que realiza as chamadas de procedimentos nativos do sistema operacional, como pode ser visto na Figura 15.

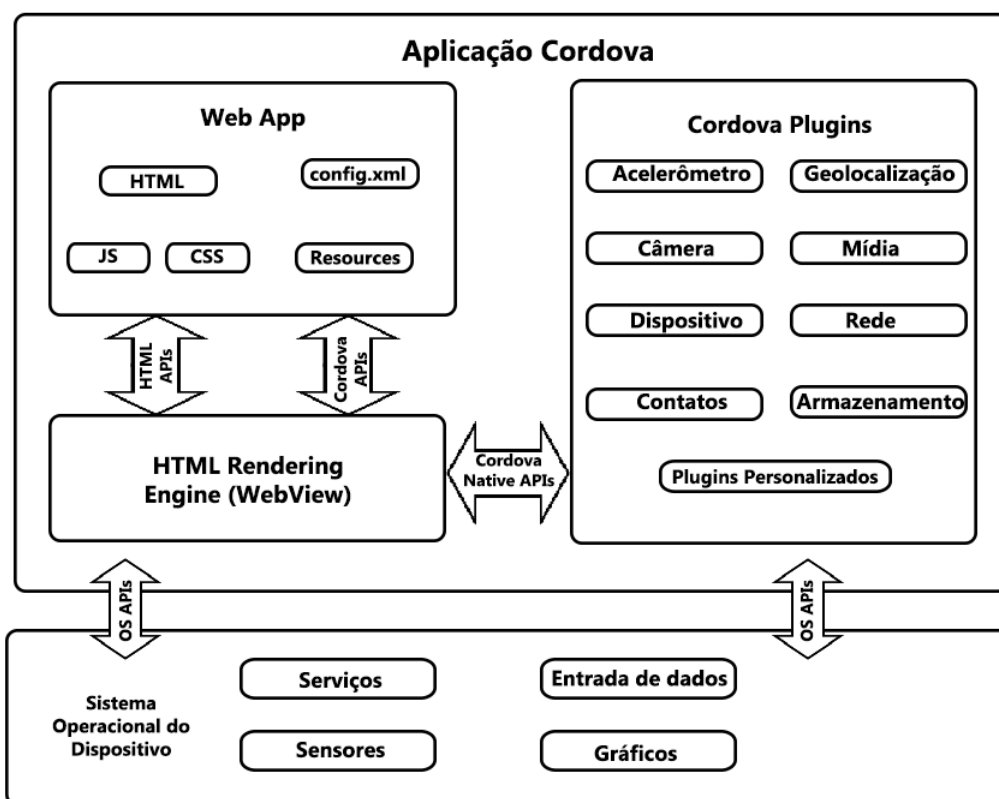


Figura 15 - Arquitetura do *Cordova*¹⁷

Também é possível utilizar *plugins* fora da biblioteca padrão do *Cordova*, sendo possível escrevê-los ou simplesmente utilizar algum já pronto.

O *Cordova* é responsável por transformar o projeto em um formato reconhecido por cada plataforma, o que torna possível o desenvolvimento de aplicativos híbridos.

¹⁶ <https://cordova.apache.org/>

¹⁷ Imagem adaptada de <https://cordova.apache.org/docs/en/latest/guide/overview/>

4.4 ANGULARJS

*AngularJS*¹⁸ é um *framework JavaScript* criado com o propósito de facilitar manipulações nos documentos *HTML*. O *framework* utiliza a estrutura *MVW*(*Model-View-Whatever*)¹⁹, uma variante do *MVC* (*Model-View-Controller*), significando que a plataforma permite um certo grau de liberdade de escolha de estrutura de projeto.

A facilidade da manipulação do *DOM* (*Document Object Model*) é possível porque o *AngularJS* permite estender a sintaxe padrão do *HTML* para suportar novos operadores. A extensão do *HTML* é feita pela compilação do documento em *HTML* válido e blocos de código *JavaScript* transparentes ao desenvolvedor. O processo de compilação é feito esporadicamente e é chamado de *digest cycle*. Neste ciclo, todas as variáveis que estão sendo observadas são atualizadas no *template*, ou seja, no documento *HTML*.

Esta atualização em tempo real das variáveis observadas é chamada de *two-way data binding* e pode ser vista com detalhes na Figura 16.

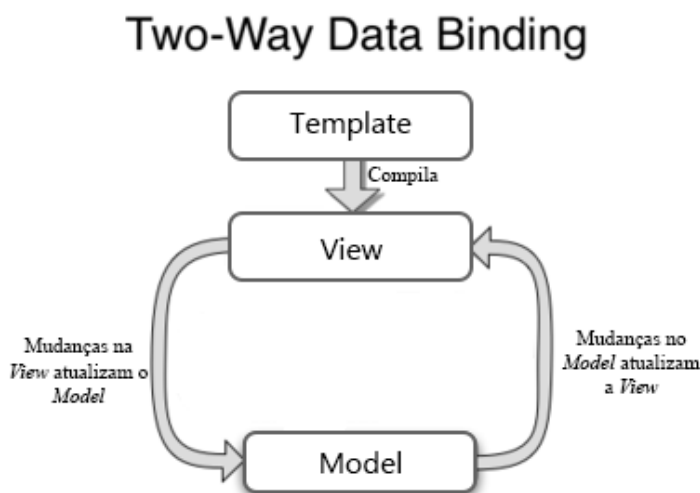


Figura 16 - *AngularJS* Two-Way Data Binding

O *framework* também fornece mecanismos de injeção de dependências e uma vasta biblioteca de módulos que implementam funcionalidades essenciais para qualquer aplicação *web*.

¹⁸ <https://angularjs.org/>

¹⁹ <https://plus.google.com/+AngularJS/posts/aZNVhj355G2>

4.5 IONIC FRAMEWORK

Outro *framework* escolhido para o projeto foi o *Ionic*²⁰. Ele funciona em conjunto com o *Cordova* e também é *open-source*. O papel do *Ionic* é tornar o aplicativo o mais parecido possível com uma implementação nativa. Para isto, são fornecidas várias soluções prontas para a interface, que além de melhorar a experiência do usuário, facilitam a implementação.

O desenvolvimento com o *Ionic* é feito utilizando *AngularJS*, pois este, como explicado na Seção 4.4, torna mais fáceis e funcionais as interações dinâmicas com o *HTML*, e aplicativos móveis se beneficiam com isso devido ao tamanho reduzido da área de exibição de conteúdo. O *Ionic* também disponibiliza módulos feitos com *AngularJS* para facilitar e agilizar ainda mais o desenvolvimento do aplicativo.

4.6 ARQUITETURA

A arquitetura do *TrackBus* consiste apenas de um aplicativo híbrido, feito com o *framework Ionic 1* em conjunto com o *framework AngularJS*. No momento em que o desenvolvimento do aplicativo foi iniciado, o *Ionic 2*²¹ já estava disponível, porém em estado Beta, ainda não recomendado para produção. Por este motivo, o *Ionic 1* foi a opção definida para seguir a implementação.

A possibilidade de hospedar as bases de dados usadas pelo aplicativo em um servidor foi descartada, pois apesar da disponibilidade inconstante e dados incorretos, para o escopo definido para este projeto os serviços fornecidos pelo *Data.rio* são suficientes. Além disto, um servidor geraria mais custos e maior complexidade na implementação.

Os dados dinâmicos, como listagem das linhas e posição dos ônibus, são obtidos do *Data.rio* por meio de requisições HTTP com os dados no formato JSON. Como a posição dos ônibus é atualizada a cada minuto na base do *Data.rio*, para que o usuário não perca uma atualização por ter iniciado o aplicativo poucos segundos antes da atualização da base, optamos por tentar atualizar a posição cada trinta segundos. Para dados estáticos, como pontos turísticos e pontos de parada, armazenamos os dados pré-processados diretamente no aplicativo como arquivos também no formato JSON.

²⁰ <https://ionicframework.com/>

²¹ <http://ionic.io/2>

4.7 BIBLIOTECAS UTILIZADAS

Além dos frameworks discutidos anteriormente, algumas bibliotecas foram utilizadas para propósitos específicos, como discutido a seguir.

4.7.1 TURF

Para cálculos geoespaciais foi utilizado o *Turf*²², biblioteca feita em *JavaScript* que utiliza *GeoJSON* como estrutura principal de dados. *GeoJSON*²³ é um formato para compartilhamento de dados geoespaciais baseado em *JSON*. Esse formato é definido pela RFC7946, publicada em Agosto 2016, da *IETF (Internet Engineering Task Force)*. O formato já existia desde 2008 (BUTLER, 2008), porém ainda não havia sido padronizado.

O *Turf* possui diversas funcionalidades para manipular informações geoespaciais. Utilizamos especificamente duas delas: a de cálculo de distância entre dois pontos, que é realizada pela fórmula de Haversine (ROBUSTO, 1957), e a verificação se um ponto está dentro de um polígono.

O cálculo de distância é usado frequentemente para verificar a aproximação de um ônibus ao usuário ou de um usuário a um ponto de parada ou turístico. A verificação de pontos dentro de um polígono é usada para descobrir quais linhas de ônibus possuem veículos próximos ao usuário. Como polígono, utilizamos um círculo com o centro sendo a posição atual do usuário e um raio de 2 km. Este círculo, junto com a posição de cada ônibus de cada linha, são passados como parâmetro da função e são retornadas as linhas que estão dentro do raio estipulado.

4.7.2 NGCORDOVA

O *framework Ionic* utiliza o *Cordova* para fazer integração nativa por meio de funções *JavaScript* com os sistemas dos dispositivos móveis, como é descrito na Seção 4.3. O *framework Angular*, também presente no projeto em conjunto com o *Ionic*, utiliza amplamente o conceito de *promises*²⁴ para resolver chamadas assíncronas, no lugar dos *callbacks*²⁵, mais utilizados em *JavaScript* nativo.

²² <http://turfjs.org/>

²³ <https://tools.ietf.org/html/rfc7946>

²⁴ Uma *promise* representa o resultado de uma operação assíncrona, podendo estar pendente ou ter sido cumprida ou rejeitada. *Promises* excluem a necessidade de passar funções *callback* como parâmetro.

²⁵ *Callbacks* são funções que são passadas como parâmetros para outras funções. São utilizadas para executar trechos de código após a finalização de um procedimento assíncrono.

Para que a integração com as funcionalidades nativas do aparelho seja feita de forma transparente para o usuário, ou seja, sem que haja interrupções na usabilidade do aplicativo, todas as chamadas nativas devem ser assíncronas. Para que as chamadas assíncronas sejam resolvidas corretamente, é necessário passar uma função *callback* como parâmetro, que é executada ao fim da execução do *plugin* nativo.

O *ngCordova* é uma biblioteca *JavaScript* que fornece *wrappers*²⁶ para *plugins* do *Cordova*, ajudando a manter boas práticas no projeto, transformando as funções nativas que recebem *callbacks* em *promises*, deixando o código mais uniforme e consistente.

4.7.3 ANGULAR GOOGLE MAPS

Em 2008 o *Google Maps* foi lançado para plataformas móveis e já possuía integração com o GPS do aparelho. Como informado na Seção 2.7, é um aplicativo amplamente utilizado e conhecido, por isto optamos pela integração com este serviço.

O *Google Maps* também disponibiliza uma *API JavaScript* para desenvolvedores utilizarem os seus serviços em aplicações *web*. O aplicativo *TrackBus* conta com a *API* do *Google Maps* para exibir o mapa com marcadores, indicando a localização do usuário, dos ônibus e dos pontos de turísticos e de parada. Pelo mesmo motivo citado na Seção 4.7.2, a fim de manter boas práticas no projeto, utilizamos o *Angular Google Maps*, além deste prover otimizações para a exibição de marcadores no mapa.

A utilização dos serviços da *API* do *Google Maps* é gratuita até 25.000 carregamentos de instâncias do mapa por dia, o que atende suficientemente bem as nossas necessidades, dado que o aplicativo é proposto como uma prova de conceito.

4.8 PRÉ-PROCESSAMENTO DOS DADOS

Conforme mencionado na Seção 2.3, o *Data.rio* disponibiliza diversas bases de dados relativas aos ônibus da Cidade do Rio de Janeiro. Quatro destas bases são utilizadas como fonte de informações do *TrackBus*. Suas descrições são exibidas a seguir e um esquema resumido da estrutura pode ser visto na Figura 17:

B1. Localização do ônibus e linhas em tempo real²⁷

B2. Localização de todos os pontos de ônibus da cidade, com nomes descritivos²⁸

²⁶ Objetos que encapsulam outros objetos, podendo adicionar novas funcionalidades.

²⁷B1 - <http://dadosabertos.rio.rj.gov.br/apiTransporte/apresentacao/rest/index.cfm/ônibus>

B3. Pontos de parada de cada uma das linhas, e sua ordem no percurso²⁹

B4. Localização endereço e nomes de atrações turísticas³⁰

B1. GPS Ônibus	B2. Pontos de ônibus	B3. Pontos por linha	B4. Pontos turísticos
+ DATAHORA: datetime	+ STOP_ID: varchar	+ LINHA: varchar	+ NOME: varchar
+ ORDEM: varchar	+ STOP_CODE: varchar	+ DESCRICAO: varchar	+ ENDEREÇO: varchar
+ LINHA: varchar	+ STOP_NAME: varchar	+ AGENCIA: varchar	+ NÚMERO: numeric
+ LATITUDE: numeric	+ STOP_DESC: varchar	+ SEQUENCIA: numeric	+ BAIRRO: varchar
+ LONGITUDE: numeric	+ STOP_LAT: numeric	+ LATITUDE: numeric	+ TELEFONE: varchar
+ VELOCIDADE: numeric	+ STOP_LON: numeric	+ LONGITUDE: numeric	+ LATITUDE: numeric
+ DIRECAO: string	+ ZONE_ID: numeric		+ LONGITUDE: numeric
	+ STOP_URL: varchar		
	+ LOCATION_TYPE: varchar		
	+ PARENT_STATION: numeric		
	+ STOP_TIMEZONE: timestamp		
	+ PLATFORM_CODE: numeric		

Figura 17 - Bases de dados provenientes do *Data.rio*³¹

Porém, para facilitar a execução de algumas das funcionalidades propostas é necessário que informações contidas nestas diferentes bases estejam relacionadas entre si. Mais precisamente, para a função de listagem dos pontos de ônibus (Seção 3.5) e pontos turísticos (Seção 3.6) de uma linha e para identificação destes pontos ao longo do percurso (Seção 3.7), é necessário que as bases B2, B3 e B4 estejam diretamente relacionadas.

Precisamos de informações completas sobre os pontos que a linha selecionada para, quais os pontos turísticos próximos àquele ponto, e ainda quais são todos os pontos turísticos que a linha se aproxima durante seu percurso. Entretanto, não existem identificadores que permitam a junção das bases, como chaves estrangeiras, por exemplo. Os parâmetros que permitem esta junção são os valores de latitude e longitude, comuns às três bases em questão.

Uma vez que B2, B3 e B4 não são atualizadas regularmente, e que realizar as junções em tempo de execução seria custoso e prejudicaria o desempenho do aplicativo, optamos por realizar um pré-processamento para obter as relações entre os dados que faltavam.

²⁸B2 - <http://data.rio/dataset/onibus-gtfs>

²⁹B3 - <http://data.rio/dataset/pontos-de-parada-de-onibus/resource/9e182b08-67ee-4f9a-ac13-d8fee1a106b7>

³⁰B4 - <http://data.rio/dataset/pontos-turisticos-e-culturais>

³¹ Nomes das colunas das bases iguais aos fornecidos pelo portal *Data.rio*

Apesar dos valores de latitude e longitude estarem presentes em todas as tabelas, eles foram obtidos em diferentes medições, o que, provavelmente influenciado pelo erro do GPS discutido na Seção 2.2, não gerou valores idênticos, mesmo se tratando de pontos equivalentes. Para tratamento destas inconsistências, foi necessária a criação de dois *scripts* que comparam os valores de latitude e longitude, considerando suas posições geográficas e a distância real entre os pares analisados. O cálculo desta distância foi realizado por meio da fórmula de *Haversine* (ROBUSTO, 1957), fórmula que determina a distância entre dois pontos considerando a curvatura da Terra.

O primeiro *script* foi criado para identificar nomes descritivos, fornecidos pela base B2, associadas aos pontos de parada de cada linha. A base B3, por possuir a listagem de todas as linhas e seus respectivos pontos de parada, serviu como ponto inicial das junções. Para cada tupla de B3, foi buscado um ponto em B2 que estivesse a uma distância menor que 22 metros dela. Os 22 metros foram definidos de acordo com o erro do GPS discutido na Seção 2.2, de 11 metros. Uma vez que estamos utilizando duas medições distintas, o intervalo foi dobrado. Desta forma, todos os pontos comparados que obtivessem um resultado de distância menor que a definida eram incorporados à estrutura final de pontos daquela linha, agora com uma indicação semântica daquele ponto no percurso.

Apesar da base B3 ter uma coluna “DESCRICAO”, que deveria ser suficiente para identificar semanticamente aquele ponto de parada, como pode ser visto na Tabela 4, este valor é na verdade uma descrição da linha, e não dos pontos individualmente.

Tabela 4 - Amostra de dados base B3

Linha	Descrição	Agência	Sequência	Latitude	Longitude
474	474-JACARE X JARDIM DE ALAH	Fetranspor	13	-22.8997	-43.2285
474	474-JACARE X JARDIM DE ALAH	Fetranspor	46	-22.8992	-43.2448
474	474-JACARE X JARDIM DE ALAH	Fetranspor	50	-22.9005	-43.2522
474	474-JACARE X JARDIM DE ALAH	Fetranspor	7	-22.896	-43.2406

Entretanto, existem tuplas da base B3 sem relação com tuplas em B2 considerando uma distância menor de 22 m. Para casos como este, é atribuído à descrição do ponto a mensagem "Nome do ponto não identificado". Além disto, podem também acontecer casos em que existam mais de um ponto dentro do intervalo de distância considerado. Nestes casos, o algoritmo retorna a descrição do primeiro ponto que aparecer na listagem, independentemente se existe ou não

outro com uma distância menor, possibilitando a ocorrência de falsos positivos no processamento.

Ainda no primeiro *script*, incorporada a iteração inicial na base B3, também foi feita uma busca na base B4 por todos os pontos turísticos a uma distância de 2 km do ponto analisado. Todos os pontos turísticos neste raio são incorporados à lista de pontos turísticos próximos ao ponto de ônibus. Esta associação foi feita para informar ao usuário quais atrações turísticas, podendo caminhar caso decida descer naquele ponto. Apesar da distância média aceitável para caminhar até um ponto ser de aproximadamente 800 metros (NEW JERSEY DEPARTMENT OF TRANSPORTATION, 1994), a distância de 2 km foi escolhida para dar liberdade ao usuário configurar no aplicativo um valor maior (até 2 km), caso não se importe em caminhar mais do que 800 metros.

Os resultado gerados por este *script* foram salvos em arquivos JSON individuais para cada linha analisada, onde cada arquivo possui informações de cada um dos pontos da mesma. A estrutura final do resultado gerado pode ser vista na Tabela 6 e Tabela 7.

O segundo *script* precisou ser implementado para a funcionalidade do aplicativo que lista todos os pontos turísticos de uma linha. Ele também foi iniciado com a base B3 e segue uma implementação similar à segunda etapa do *script* de pontos de parada, comparando a distância dos pontos de cada linha aos pontos turísticos fornecidos na base B4. A distância máxima para inclusão do ponto também é de 2km, pelo mesmo motivo mencionado anteriormente. A diferença para o primeiro *script* é que ao invés de listar as atrações próximas de cada ponto, é gerada uma lista de todos os pontos turísticos que estão próximos àquela linha ao longo de todo este trajeto. Este *script* também gera um arquivo JSON para cada linha, e a sua estrutura final pode ser vista na Tabela 8.

Todos arquivos gerados por este pré-processamento foram incluídos dentro do próprio aplicativo, permitindo um acesso rápido às informações e retirando uma alta carga de processamento de aparelhos que geralmente são executados em plataformas com hardware limitado. Com isto, o aplicativo se torna mais rápido e econômico em termos de energia, sem prejudicar a validade dos dados.

4.9 ESTRUTURA DOS DADOS

Antes da implementação do aplicativo foi preciso analisar o formato dos dados enviados pelo *Data.rio*. No caso das informações sobre as linhas de ônibus, a documentação³² está disponível e pôde ser facilmente acessada no próprio site do *Data.rio*. Os dados estão expostos como indica a Tabela 5 e são utilizados da mesma forma no aplicativo, sem alterações.

Tabela 5 - Estrutura de dados de ônibus

#	Campo	Descrição	Tipo
1	DataHora	Data e hora da coleta do dado	DATETIME
2	Ordem	Identificação alfanumérica encontrada na lateral do ônibus	VARCHAR
3	Linha	Linha do ônibus	VARCHAR
4	Latitude	Latitude do ônibus na coleta (GPS, WGS84)	NUMERIC
5	Longitude	Longitude do ônibus na coleta (GPS, WGS84)	NUMERIC
6	Velocidade	Velocidade do ônibus na hora da coleta do dado	NUMERIC
7	Direção	Número em graus em relação ao norte, que representa a direção do veículo (campo calculado)	NUMERIC

Porém, no caso das informações sobre os pontos de parada e pontos turísticos, a documentação não está disponível. Também não há documentação para o arquivo *GTFS* (*General Transit Feed Specification*), que possui informações adicionais sobre as linhas de ônibus. Com o pré-processamento destes dados, foram gerados arquivos com estrutura diferente da que é fornecida pela Prefeitura. A nova estrutura dos arquivos de pontos de parada é exibida na Tabela 6 e na Tabela 7. Os dados da listagem de pontos turísticos são definidos na Tabela 8.

³² http://dadosabertos.rio.rj.gov.br/apitransporte/apresentacao/pdf/documentacao_gps.pdf

Tabela 6 - Estrutura dos dados de pontos de parada

#	Campo	Descrição	Tipo
1	Linha	Linhas dos ônibus	VARCHAR
2	Descrição	Nome completo do ponto	VARCHAR
3	Sequência	Número que indica a ordem do ponto no trajeto	NUMERIC
4	Latitude	Latitude do ponto	NUMERIC
5	Longitude	Longitude do ponto	NUMERIC
6	Pontos Turísticos	Lista de pontos turísticos próximos ao ponto	OBJECT

Tabela 7 - Estrutura dos dados dos pontos turísticos dentro de cada ponto de parada

#	Campo	Descrição	Tipo
1	Nome	Nome do ponto turísticos	VARCHAR
2	Distância	Distância do ponto turístico ao ponto de parada, em metros	NUMERIC

Tabela 8 - Estrutura dos dados dos pontos turísticos

#	Campo	Descrição	Tipo
1	Nome	Nome do ponto turísticos	VARCHAR
2	Latitude	Latitude do ponto turístico	NUMERIC
3	Longitude	Longitude do ponto turístico	NUMERIC
4	Endereço	Endereço do ponto turístico	VARCHAR

Além das definições dadas na Tabela 6, é importante lembrar que os ônibus geralmente possuem um trajeto de ida e um de volta, que não são necessariamente iguais. Assim, é possível que os números de sequência sejam duplicados, ou seja, possua mais de um ponto de parada com o mesmo número de sequência.

4.10 DETALHES DE IMPLEMENTAÇÃO

Para implementação dos requisitos listados na Seção 3.2 foram utilizados diferentes *plugins* do *Cordova* e bibliotecas. Os *plugins* entraram no projeto com o objetivo de facilitar e padronizar a implementação das funcionalidades nativas, visto que alguns não pertencem à biblioteca padrão do *Cordova*. As bibliotecas *Turf* e *Angular Google Maps* foram utilizadas para

auxiliar cálculos e exibições geoespaciais. Detalhes da utilização de ambos são dados na Seção 4.10.1. Na Seção 4.10.2 é descrito o algoritmo para decidir o ponto atual de parada e o próximo de acordo com a posição do usuário.

4.10.1 DETALHES DA UTILIZAÇÃO DE PLUGINS

Para as notificações, três *plugins* do *Cordova* (ver Seção 4.3) são utilizados em conjunto. Primeiramente, é utilizado o *plugin* que permite exibição de notificações *push* no dispositivo. Este *plugin* também permite que sejam emitidos alertas sonoros, ao mesmo tempo que ocorre a notificação. Assim que a notificação é exibida no dispositivo, podem ser utilizados outros dois meios de alerta: vibratórios e por voz. Estes alertas são possíveis devido ao *plugin* de vibração e o de *text-to-speech*.

A posição geoespacial é informada ao aplicativo por meio de outro *plugin* do *Cordova* que realiza a integração nativa com o serviço de geolocalização do dispositivo. As operações geoespaciais são todas realizadas pelo *Turf* (ver Seção 4.7.1).

Para que fosse possível emitir notificações quando o aplicativo não está em primeiro plano ou a tela do dispositivo está desligada, foi utilizado um *plugin* para que o aplicativo fosse executado em *background*. O aplicativo funciona em *background* apenas em momentos que o usuário pode solicitar e receber notificações, ou seja, durante a listagem de ônibus ou pontos no mapa.

Cada marcador exibido no mapa pelo *Angular Google Maps* (ver Seção 4.7.3) precisa de um identificador único para ser referenciado em outro momento. O aplicativo possui marcadores para ônibus, pontos turísticos e pontos de parada. Os ônibus são identificados por seu atributo “ordem”, para pontos turísticos e pontos de parada, os atributos “nome” e “descrição” são utilizados como identificadores. Todos os marcadores são posicionados de acordo seus atributos de latitude e longitude.

4.10.2 ALGORITMO DE IDENTIFICAÇÃO DE PONTOS DE PARADA

Para que fosse possível notificar o usuário do próximo ponto de parada, foi preciso definir um algoritmo que nos permitisse identificar qual seria este ponto. Como visto na Tabela 6, cada ponto de parada possui um identificador de sequência, que geralmente não é único, pois a

rota no sentido oposto também possui números de sequência distribuídos da mesma forma. O algoritmo para definir o próximo ponto de parada é descrito na Figura 18.

Esse algoritmo recebe como entrada o ponto de parada atual do usuário. O ponto de parada atual é calculado como sendo o ponto mais próximo, dentre todos os pontos de parada. O cálculo é realizado desta forma quando é iniciado o acompanhamento de viagem e quando o usuário escolhe ser notificado sobre a proximidade de um ponto de parada ou turístico. Depois disto, a cada atualização da posição geoespacial do usuário, é calculada a distância para o próximo ponto de parada. Caso a distância seja menor do que a configurada pelo usuário, o ponto atual é atualizado e passa a ser o próximo ponto. Desta forma, caso o ponto esteja na lista de pontos que o usuário deseja ser notificado, a notificação é exibida.

```

Seja P a lista de todos os pontos de parada
Seja C a lista dos candidatos a próximo ponto de parada
Seja p o ponto de parada atual
Seja q o próximo ponto de parada

Para cada elemento e em P, faça:
    Se e.sequência == (p.sequência + 1), faça:
        C.adicionar(e)
    Fim se
Fim para

q <- C[0]
Para cada elemento e em C, faça:
    Se distânciaAoUsuário(e) < distânciaAoUsuário(q), faça:
        q <- e
    Fim se
Fim para

Retornar q

```

Figura 18 - Pseudocódigo

O algoritmo, porém, falha em um caso específico: quando o ponto de parada no sentido oposto ao usuário, caso esteja entre os candidatos a próximo ponto, está mais próximo do que o ponto no mesmo sentido. Neste caso é retornado um ponto no sentido oposto da navegação como próximo ponto.

Entretanto, não é uma falha fatal, pois a próxima vez que o algoritmo for executado, a distância para um ponto no sentido oposto com o próximo número de sequência tende a estar

cada vez maior. Desta forma, o algoritmo tende a voltar para um estado correto nas iterações seguintes.

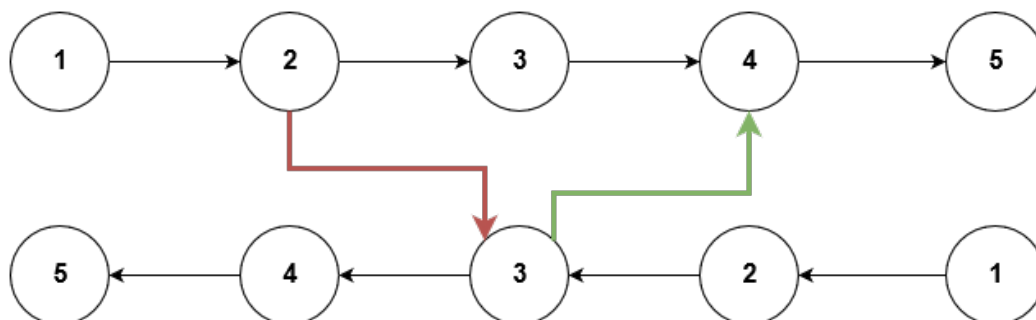


Figura 19 - Caso de falha e recuperação do algoritmo

Como é possível ver na Figura 19, as setas pretas indicam o caminho esperado, a vermelha o caminho tomado no caso do erro descrito acima e a verde após a recuperação do algoritmo.

A notificação de proximidade dos pontos turísticos é feita de acordo com o ponto de parada atual do usuário. Como pode ser visto na Tabela 6, cada ponto de parada possui uma lista de pontos turísticos próximos. O aplicativo, então, busca nesta lista quais pontos o usuário deseja ser notificado. Caso a distância para o ponto de parada atual seja menor ou igual ao parâmetro de proximidade de pontos turísticos configurado pelo usuário, a notificação é exibida.

4.11 CONSIDERAÇÕES FINAIS

Como pode ser visto nas seções anteriores, o aplicativo é dependente dos dados fornecidos pelo *Data.rio*. Durante o desenvolvimento foram encontradas algumas inconsistências no dados, como por exemplo, ônibus sem identificador de linha e pontos de parada ordenados incorretamente. Também observamos que o serviço ficou indisponível algumas vezes. Todos estes problemas afetam diretamente o aplicativo, às vezes podendo até impedir o uso do mesmo.

Apesar das inconsistências e a instabilidade da base de dados, os algoritmos apresentados podem ser reproduzidos em outras bases com estruturas semelhantes e espera-se que o resultado seja igualmente positivo.

Por outro lado, o aplicativo não sofreu problemas de desempenho, mesmo sendo híbrido e realizando operações geoespaciais. Todas as funcionalidades propostas foram implementadas

sem ressalvas e julgamos que a usabilidade esteja simples o suficiente para atender o público alvo.

Foram feitos testes de aceitação para validar o funcionamento e a facilidade de uso do aplicativo. Os resultados dos testes são apresentados no Capítulo 5.

CAPÍTULO 5 – AVALIAÇÃO DOS USUÁRIOS

5.1 INTRODUÇÃO

Para a avaliação do *TrackBus*, foram convidadas sete pessoas para testar o aplicativo. Os avaliadores estão na faixa etária de 20 a 30 anos. Todos já possuíam experiência anterior com *smartphones* e foram escolhidos por conveniência, sem critérios rígidos. Não foi dado nenhum treinamento nem roteiro de utilização do aplicativo. Esperávamos que cada avaliador utilizasse o aplicativo da forma que usaria em um dia comum.

Dois dos avaliadores não conseguiram inicializar o aplicativo corretamente por problemas de incompatibilidade do aplicativo com a versão do sistema operacional. Solicitamos que todos os usuários que conseguiram utilizar o aplicativo com sucesso respondessem um formulário com 11 perguntas indicando como foi a experiência de utilização.

Foram feitas perguntas tanto para saber se o aplicativo funcionou corretamente, quanto abertas, para avaliar de forma geral as impressões que eles tiveram após utilizarem. As perguntas fechadas deveriam ser respondidas de acordo com a escala Likert (1932), em que os usuários poderiam selecionar as opções “Concordo totalmente”, “Concordo”, classificadas como positivas, “Não concordo nem discordo”, classificada como neutra, “Discordo” e “Discordo totalmente”, respostas que indicam uma experiência negativa. Além destas, também foi incluída a opção “Não se aplica” para o caso do usuário não ter testado tal funcionalidade. Já as perguntas abertas deveriam ser respondidas em campos de texto livre. As perguntas apresentadas aos usuários são listadas na Tabela 9.

Tabela 9 – Questionário de avaliação dos usuários

#	Descrição	Tipo
Q1.	Descreva com poucas palavras, como foi sua experiência geral com o aplicativo TrackBus.	Aberta
Q2.	O TrackBus foi fácil de interagir?	Fechada
Q3.	Você conseguiu comparar mais de uma linha no mapa?	Fechada
Q4.	A notificação de aproximação do ônibus ao ponto foi executada no tempo esperado?	Fechada
Q5.	Você conseguiu encontrar o ponto que precisava com facilidade?	Fechada
Q6.	Uma vez embarcado no ônibus, a notificação de aproximação do ponto foi executada no tempo esperado?	Fechada
Q7.	Uma vez embarcado no ônibus, a notificação de aproximação do ponto turístico foi executada no tempo esperado?	Fechada
Q8.	Você usaria novamente o TrackBus?	Fechada
Q9.	Você recomendaria o TrackBus para amigos?	Fechada
Q10.	Em caso afirmativo, como você contaria pra ele?	Aberta
Q11.	Resuma o TrackBus em uma ou duas palavras	Aberta

Na Seção 5.2 são apresentados e discutidos os resultados de cada pergunta, enquanto na Seção 5.3 são discutidas as ameaças à validade da avaliação. Por fim, as conclusões e considerações finais são expostas na Seção 5.4.

5.2 RESULTADOS DA AVALIAÇÃO

Nesta seção são apresentados e discutidos os resultados das perguntas que foram respondidas pelos avaliadores. As perguntas são apresentadas na ordem proposta para que os avaliadores respondessem. No total, cinco dos usuários que testaram o aplicativo responderam ao formulário. As perguntas fechadas contam com um gráfico de pizza representando os resultados obtidos nas mesmas.

5.2.1 EXPERIÊNCIA COM O APLICATIVO

Em relação a pergunta Q1, as respostas ficaram divididas entre reações positivas e negativas. Dois dos avaliadores responderam que o aplicativo precisa de melhorias na interface com o usuário, apesar de não informarem quais melhorias, e correção de problemas técnicos, como falha ao emitir notificações e travamentos. Os outros três elogiaram o aplicativo como “Fácil e direto” e “Aplicativo simples e intuitivo”. Com este resultado não conseguimos obter nenhuma conclusão satisfatória, para isso precisaríamos de mais avaliadores ou melhor descrição dos problemas encontrados.

5.2.2 FACILIDADE DE INTERAÇÃO

De acordo com os resultados obtidos da pergunta Q2, na maioria das respostas obtivemos resultados positivos em relação a facilidade de uso do aplicativo: “Concordo” (2) e “Concordo Totalmente” (2). Apenas um dos avaliadores respondeu como “Não concordo nem discordo”, o que pode indicar que ocorreu alguma dificuldade durante o uso, mas que não afetou de forma séria a usabilidade.

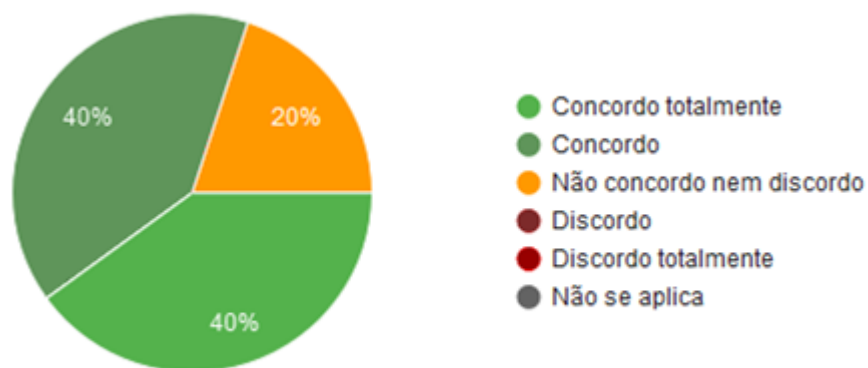


Figura 20 - Resultado Avaliação - Facilidade de Interação

5.2.3 COMPARAÇÃO DE LINHAS

Todos os avaliadores que utilizaram esta funcionalidade indicaram, na pergunta Q3, que funcionou da forma esperada e conseguiram comparar a distância entre dois ou mais ônibus no mapa. Um dos avaliadores indicou que “Não se aplica”, assumindo-se que o mesmo não testou a

funcionalidade. Podemos inferir, de acordo com as respostas, que a funcionalidade funciona conforme o esperado.

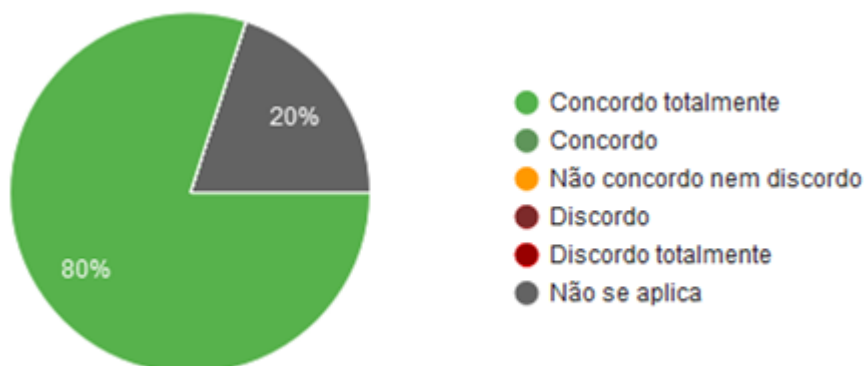


Figura 21 - Resultado Avaliação – Comparação de linhas

5.2.4 NOTIFICAÇÃO DE PROXIMIDADE DO ÔNIBUS

Na pergunta Q4, a respeito da notificação da proximidade dos ônibus, um dos avaliadores respondeu como “Discordo”, indicando que o mesmo não recebeu a notificação no tempo esperado, enquanto os outros que receberam responderam com “Concordo” (1) e “Concordo Totalmente” (3). Por falta de informações adicionais, não conseguimos dizer se a falha foi por parte do usuário na utilização do aplicativo ou por parte do aplicativo. De ambas as formas, o problema deve ser analisado com mais detalhes. No caso de falha do usuário, a interface aplicativo pode não ter sido clara o suficiente para indicar o que estava ocorrendo no momento. No caso de falha do aplicativo, devemos descobrir se a falha foi de fato do sistema ou se ocorreu por causa do recebimento de dados inconsistentes do *Data.rio*.

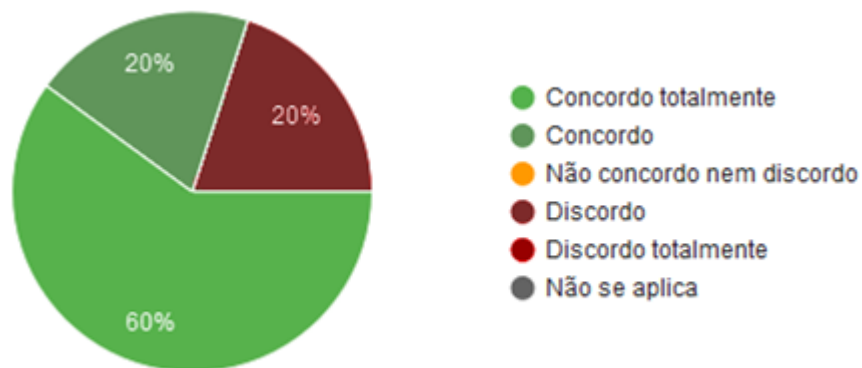


Figura 22 - Resultado Avaliação - Notificação de proximidade

5.2.5 LOCALIZAÇÃO DE PONTOS

Todos os avaliadores responderam a pergunta Q5 positivamente com “Concordo” (3) e “Concordo Totalmente” (2). Nenhum deles passou por problemas para encontrar o ponto desejado no mapa. Com isto, podemos concluir que a interface está satisfatória neste quesito e a funcionalidade é executada sem falhas.

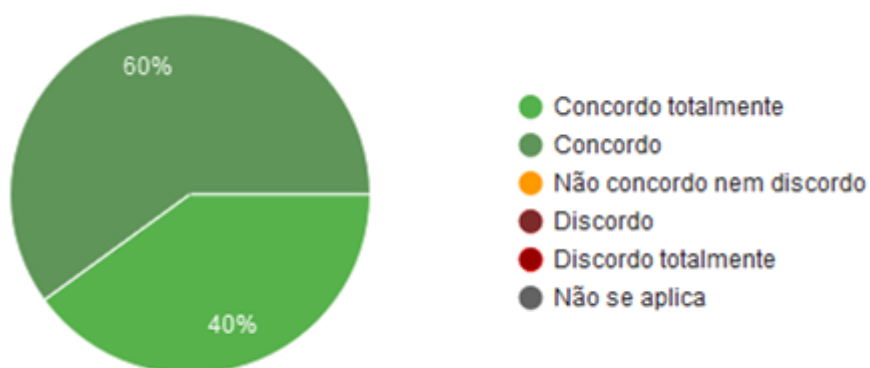


Figura 23 - Resultado Avaliação – Identificação de pontos de parada e turísticos

5.2.6 NOTIFICAÇÃO DE PROXIMIDADE DE PONTOS DE PARADA

Apenas três usuários testaram a funcionalidade descrita na pergunta Q6. Os outros dois indicaram que “Não se aplica”. Dos que testaram, observamos que um deles não recebeu a

notificação da proximidade do ponto de parada no tempo esperado. Novamente, como descrito na Seção 5.2.4, não conseguimos identificar o ponto de falha apenas com estas informações e seriam necessários mais testes para certificar o comportamento correto da funcionalidade.

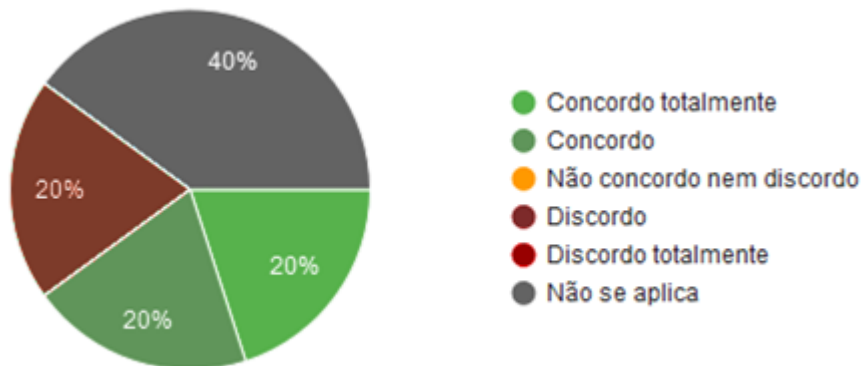


Figura 24 - Resultado Avaliação – Notificação de proximidade dos pontos de parada

5.2.7 NOTIFICAÇÃO DE PROXIMIDADE DE PONTOS TURÍSTICOS

A funcionalidade descrita na pergunta Q7 foi testada por apenas dois avaliadores. Visto que é uma funcionalidade direcionada a turistas, é esperado que não seja utilizada diariamente por todos os usuários. Os dois avaliadores que testaram a notificação de pontos turísticos responderam positivamente com “Concordo” (1) e “Concordo Totalmente” (1) à pergunta, o que provavelmente indica que está funcionando de forma esperada. Porém, acreditamos que mais testes podem ser feitos com usuários visitando a cidade, que seriam o principal foco da funcionalidade, para uma validação mais sólida.

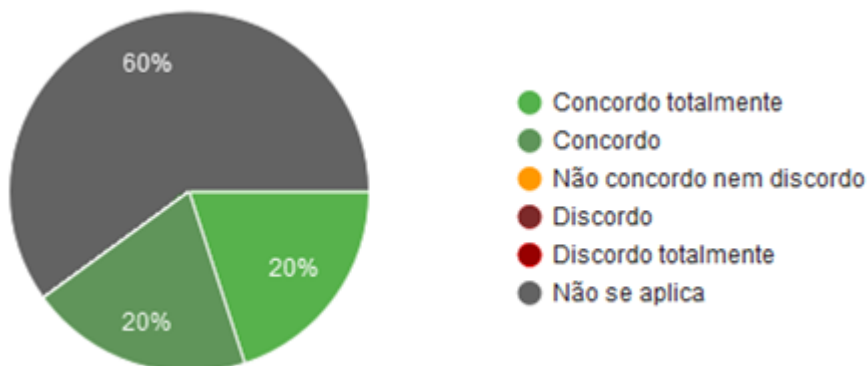


Figura 25 - Resultado Avaliação - Notificação de proximidade de pontos turísticos

5.2.8 UTILIZAÇÃO CONTINUADA DO APLICATIVO

A pergunta Q8 procurava saber se os avaliadores usariam de novo o *TrackBus*. Todos responderam positivamente com “Concordo” (1) e “Concordo Totalmente” (4). Como conclusão, podemos assumir que o aplicativo foi bem recebido pelos avaliadores e todos consideraram pelo menos uma funcionalidade útil o suficiente para motivar uma nova utilização.

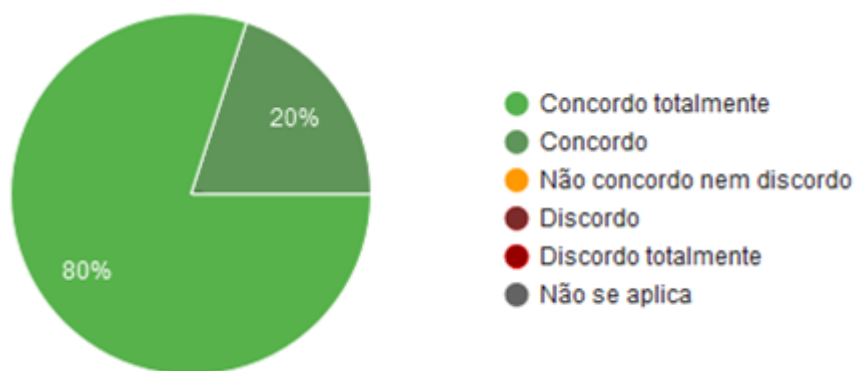


Figura 26 - Resultado Avaliação - Usaria novamente

5.2.9 RECOMENDAÇÃO DO APLICATIVO

Esta seção inclui os resultados das perguntas Q9 e Q10, pois ambas são relacionadas, questionando se recomendariam o aplicativo para os amigos, e em caso positivo, como fariam esta recomendação. Todos os avaliadores indicaram com “Concordo” (2) e “Concordo Totalmente” (3) que recomendariam o *TrackBus* para seus amigos. Em mais de uma mensagem de recomendação para os amigos, é possível perceber que os avaliadores recomendariam o aplicativo por causa da funcionalidade de visualizar os ônibus no mapa, listada na Seção 3.3. Um dos avaliadores também disse que o aplicativo é “Ideal para turistas”.

Podemos supor que a funcionalidade de visualizar os ônibus no mapa foi indicada por eles porque todos os avaliadores são passageiros diários ou frequentes de ônibus. Sendo assim, julgam ser importante saber sobre a proximidade dos ônibus.

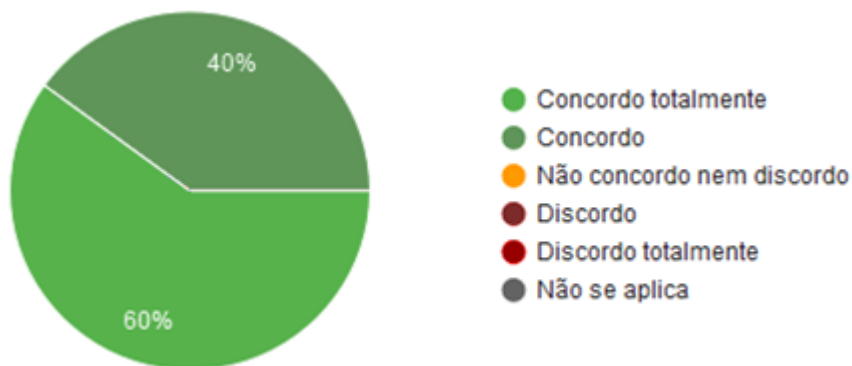


Figura 27 - Resultado Avaliação - Recomendaria o aplicativo

5.2.10 CURTO RESUMO DO APLICATIVO

Na pergunta Q11, pedimos que os avaliadores descrevessem o aplicativo com uma ou duas palavras. Podemos juntar três avaliações (“Simples e prático”, “Utilidade”, “Eficaz”) em um grupo, pois todas estas dizem respeito a forma em que são exibidas as informações, além de considerarem estas informações úteis. Os outros avaliadores responderam “Grande potencial” e “Interessante”, ambos indicam que o aplicativo possui funcionalidades interessantes e um deles diz que essas funcionalidades podem ser mais exploradas.

5.3 AMEAÇAS À VALIDADE

Como três usuários não conseguiram, por problemas técnicos citados na Seção 5.1, executar o aplicativo corretamente, a quantidade de avaliações foi pequena. Sendo assim, pode não representar de forma correta a recepção do aplicativo em um público mais geral.

Os avaliadores foram escolhidos por conveniência, sendo todos experientes com *smartphones* e conhecidos dos autores deste projeto. Um deles teve participação no projeto como colaborador, sendo responsável por grande parte da identidade visual do aplicativo.

Cada pessoa avaliou apenas uma vez o aplicativo, então as informações que possuímos são referentes apenas à primeira impressão dos usuários.

Como não houve treinamento prévio, é possível que algumas percepções negativas tenham ocorrido pela falta de experiência com o aplicativo. Contudo, o usuário típico não lê

manuais antes do uso de aplicativos, então esta decisão reflete de forma mais precisa o público alvo do aplicativo.

Ainda assim, esperamos que as avaliações representem uma visão mais geral do que testes realizados pelos desenvolvedores do aplicativo.

5.4 CONSIDERAÇÕES FINAIS

No geral, o *TrackBus* recebeu avaliações positivas sobre o funcionamento e experiência de uso.

Percebemos que o aplicativo precisa de mais rodadas de testes mais amplos e correções para que todos os requisitos propostos funcionem corretamente. Mesmo assim, o aplicativo funciona de forma satisfatória na maioria dos casos de uso.

CAPÍTULO 6 – CONCLUSÃO

6.1 CONTRIBUIÇÕES

Como principal contribuição deste projeto temos o aplicativo *TrackBus*, funcional, validado por usuários e com código livre disponível em <https://github.com/gems-uff/trackbus>. Esperamos que o aplicativo seja capaz de auxiliar todos os usuários de *smartphones Android* que utilizam transporte público diariamente ou para conhecer a Cidade do Rio de Janeiro.

Também, com maior exposição do projeto, mais desenvolvedores podem se interessar pelos dados disponíveis no *Data.rio* e desenvolver novos *softwares* para facilitar o dia a dia dos habitantes da cidade. Com maior utilização das bases de dados, tanto por parte de usuários quanto de desenvolvedores, espera-se que a Prefeitura tenha maior interesse em manter os dados atualizados e com maior disponibilidade.

6.2 LIMITAÇÕES

Como principal limitação indica-se o *Data.rio*. Apesar de ser a nossa fonte principal de dados, identificamos alguns problemas com os dados e a disponibilidade do serviço. Entre os problemas dos dados, podemos observar, principalmente, que existem ônibus sem informações sobre qual linha eles pertencem. Isto pode afetar o aplicativo na exibição dos ônibus no mapa, pois não é possível identificar diretamente a linha destes ônibus sem identificação.

Como foi visto na Seção 4.8, a base B3 contém os dados dos pontos de parada e cada ponto possui um campo “descrição”. Esperávamos que esse campo contivesse os nomes de cada ponto, porém encontramos apenas o nome da linha. Visto que esta informação não é útil para o usuário identificar o nome do ponto, foi necessário fazer um pré-processamento dos dados, integrando as informações das bases. Entretanto, este processo é passível de erros, como é descrito na Seção 4.8, sendo possível que alguns pontos de parada sejam omitidos do usuário durante a utilização do aplicativo.

Outra limitação pode ser oriunda do erro de 11 metros das medições do GPS, citado na Seção 2.2. Como o erro é relativamente baixo se comparado com as distâncias previstas para as notificações de proximidade, não esperamos que isto afete significativamente o aplicativo. Da

mesma forma, a fórmula de Haversine considera a Terra como uma esfera e não um geóide, o que introduz mais erros na geolocalização do usuário.

Durante a avaliação dos usuários, descrita no Capítulo 5, notamos que o algoritmo desenvolvido para a identificação do próximo ponto, que é explicado em detalhes na Seção 4.10.2, necessita que o ponto atual passado como parâmetro seja de fato o ponto em que o usuário se encontra. Caso o ponto atual seja identificado incorretamente, o algoritmo não falha, porém não retornará o próximo ponto do usuário e sim do ponto informado equivocadamente como atual. Para mitigar este problema, permitimos que o usuário selecione um modo alternativo de acompanhamento de viagem. Neste modo o número de sequência do ponto não é considerado para emitir notificações, sendo usada apenas a distância. Assim, a notificação sempre ocorrerá dado que o usuário está na distância definida para a notificação. Por não considerar o número de sequência dos pontos, é possível que o usuário seja notificado de um ponto que está dentro da distância de notificação mas não é o próximo ponto de parada.

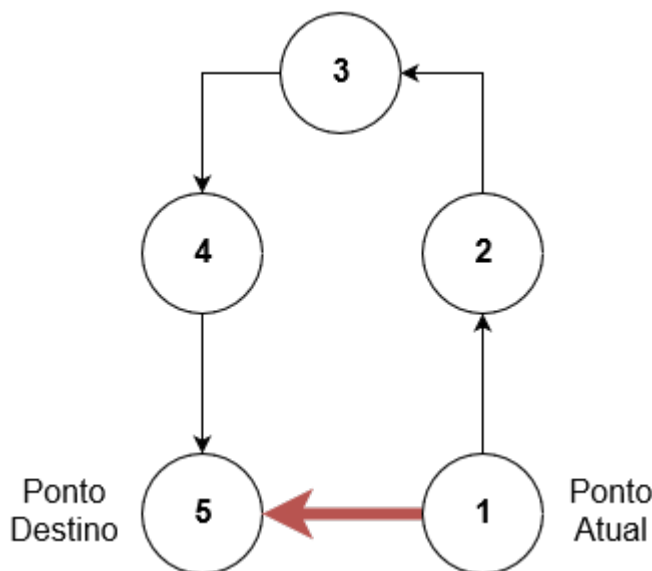


Figura 28 – Situação de falha no modo alternativo de acompanhamento

Como pode ser visto na Figura 28, o usuário se encontra no ponto 1 e deseja chegar no ponto 5, porém antes de chegar no ponto desejado, o ônibus ainda passaria pelos pontos 2, 3 e 4. Caso o número de sequência dos pontos não seja considerado, o usuário será notificado no ponto 1 da proximidade do ponto 5. Sendo assim, esta não é uma solução definitiva, mas pode ajudar usuários com problemas para receber notificações.

Dada a pequena quantidade de testes realizados, é possível que o comportamento do aplicativo seja diferente do esperado em alguns dispositivos específicos, pois é grande a diversidade de aparelhos com o sistema *Android*.

O aplicativo não foi publicado na *Apple Store* por falta de recursos financeiros, tanto para a obtenção de um *iPhone*, quanto para a licença de desenvolvedor da *Apple*. Porém, dado que é um aplicativo híbrido, como explicado na Seção 4.2, esperamos que funcione sem problemas na plataforma *iOS*.

6.3 TRABALHOS FUTUROS

Podemos enumerar duas principais vertentes para trabalhos futuros: a melhoria do aplicativo com novas funcionalidades e a melhoria da qualidade dos dados buscados do *Data.rio* e inclusão de novas informações.

Para o aplicativo, podem ser implementadas as funcionalidades presentes em outros aplicativos semelhantes, descritos no Capítulo 2, como a estimativa de tempo de viagem e tempo de chegada do ônibus, apresentar dados sobre lotação e estado de conservação dos ônibus, entre outras. A estimativa de viagem poderia ser feita por meio do trabalho de predição estatística de tempo de viagem, realizado por SILVA (2016), que apresentou resultados positivos comparados com outras abordagens utilizadas, sendo possível diminuir a insatisfação dos usuários com as estimativas imprecisas dos outros aplicativos. Para isso, assim como para apresentar os dados sobre lotação e estado de conservação dos ônibus, precisaríamos de um servidor com alta disponibilidade, além da interface adequada no aplicativo.

A implantação de um servidor aumenta a possibilidade para incluir várias novas funcionalidades, que vão desde melhorar a qualidade dos dados recebidos pelo *Data.rio* e até incluir novos dados. Como foi percebido durante o projeto, citado na seção 4.11, a disponibilidade do servidor do *Data.rio* é menor do que o ideal. Uma forma de mitigar o problema seria armazenar em um servidor de alta disponibilidade um *cache* das informações passadas pela Prefeitura, podendo repassar para o usuário informações antigas, indicando a data da última atualização, para que o mesmo possa fazer suas estimativas. O *cache* de informações também possibilita realizar certos tratamentos dos dados fornecidos, além da estimativa de tempo de chegada. Um exemplo de tratamento seria identificar os ônibus sem identificador de

linha de acordo com seu trajeto, verificando quais outros ônibus percorrem um trajeto semelhante aos sem identificador.

Para tornar o aplicativo mais completo, também podem ser incluídas informações sobre o estado do trânsito. Estas informações podem ser adquiridas pelo serviço do *Google Maps*. Com estas novas informações, seria possível realizar previsões mais precisas do tempo de viagem.

Durante a avaliação dos requisitos pelos usuários na Seção 3.2, alguns sugeriram outras funcionalidades. A mais sugerida entre elas é a exibição do custo da passagem de cada ônibus e o custo total da viagem, incluindo descontos fornecidos pelo Bilhete Único. Também foi sugerida a opção de marcar linhas de ônibus como favoritas, para facilitar a usabilidade, principalmente de usuários diários do aplicativo. Além destas, também foram sugeridas funcionalidades relacionadas a exibição da rota de viagem do percurso atual e cálculo da melhor rota considerando um ponto de origem e destino, exibindo quais linhas de ônibus devem ser utilizadas e uma filtragem por linhas mais rápidas.

Por fim, apesar do nome do aplicativo, *TrackBus*, é possível expandir a quantidade de modais de transporte urbano suportados. O *Data.rio* também fornece informações sobre as barcas, BRT, trens, metrô e bicicletas do projeto *Bike Rio*.

REFERÊNCIAS BIBLIOGRÁFICAS

ALBUQUERQUE, R. **Prefeitura lança portal de serviços personalizado e abre base de dados do município para o cidadão.** Disponível em: <<http://www.rio.rj.gov.br/web/guest/exibeconteudo?id=4669376>>. Acesso em: 11 nov. 2016.

BRIGATTO, G. **Número de smartphones passa o de computadores e tablets no Brasil.** Disponível em: <<http://www.valor.com.br/empresas/4010440/numero-de-smartphones-passa-o-de-computadores-e-tablets-no-brasil>>. Acesso em: 8 out. 2016.

BUTLER, H. **GeoJSON Specification.** Disponível em: <<http://geojson.org/geojson-spec.html>>. Acesso em: 10 dez. 2016.

D'ONFRO, J. **Here are all the Google services with more than a billion users.** Disponível em: <<http://www.businessinsider.com/google-services-with-1-billion-users-2015-10>>. Acesso em: 24 dez. 2016.

FETRANSPOR. **Portal Fetranspor – Mobilidade com Qualidade – Vá de Ônibus: Aplicativo inova com realidade aumentada para clientes do transporte público,** 15 jul. 2014. Disponível em: <<https://www.fetranspor.com.br/noticias/va-de-onibus-aplicativo-inova-com-realidade-aumentada-para-clientes-do-transporte-publico>>. Acesso em: 8 out. 2016

FGV. **Pesquisa Anual do Uso de TI nas Empresas.** Disponível em: <<http://eaesp.fgvsp.br/sites/eaesp.fgvsp.br/files/pesti2016gvciappt.pdf>>. Acesso em: 10 ago. 2016.

GOOGLE MAPS. **Cities Covered.** Disponível em: <<https://maps.google.com/landing/transit/cities/>>. Acesso em: 8 out. 2016.

GRUDIN, J.; PRUITT, J. **Personas, Participatory Design and Product Development: An Infrastructure for Engagement. DUX '03 Proceedings of the 2003 conference on Designing for user experiences,** p. 1–8, jun. 2003.

HUBER, W. A. **Measuring accuracy of latitude and longitude? - Geographic Information Systems Stack Exchange.** Disponível em: <<http://gis.stackexchange.com/questions/8650/measuring-accuracy-of-latitude-and-longitude>>. Acesso em: 10 nov. 2016.

HUTCHEON, S. **The Untold Story About the Founding of Google Maps.** Disponível em: <<https://medium.com/@lewgus/the-untold-story-about-the-founding-of-google-maps-e4a5430aec92#.3jl135oqh>>. Acesso em: 21 fev. 2016.

IBGE. **IBGE | Cidades | Infográficos | Rio de Janeiro | Rio de Janeiro | Frota de veículos.** Disponível em:

<<http://cidades.ibge.gov.br/painel/frota.php?lang=&codmun=330455&search=rio-de-janeiro|rio-de-janeiro|infogr%E1ficos:-frota-municipal-de-ve%EDculos%27>>. Acesso em: 14 dez. 2016.

IPEA. **SIPS traz avaliação do brasileiro sobre transporte.** Disponível em:

<http://www.ipea.gov.br/portal/index.php?option=com_content&view=article&id=6894>.

Acesso em: 21 mar. 2016.

IPEA. **Ricos e pobres perdem cada vez mais tempo no trânsito.** Disponível em:

<http://www.ipea.gov.br/portal/index.php?option=com_content&view=article&id=17212>.

Acesso em: 21 mar. 2016.

KIM, R. **Transit app Moovit takes a page from Waze's crowdsourcing playbook.** Disponível em:

<<https://gigaom.com/2012/12/27/transit-app-moovit-takes-a-page-from-wazes-crowdsourcing-playbook/>>. Acesso em: 16 set. 2016.

LIKERT, R. A technique for the measurement of attitudes. v. 22, p. 1–55, 1932.

MATOS, D. **O Poder do Open Data | Ciência e Dados.** Disponível em:

<<http://www.cienciaedados.com/o-poder-do-open-data/>>. Acesso em: 11 nov. 2016.

MESSRS., R. J. M.; ZOLLER, C. J. Principle of Operation of NAVSTAR and System Characteristics. p. 95–106, jun. 1978.

MONICO, J. F. G. **Posicionamento pelo NAVSTAR-GPS.** 1. ed. [s.l: s.n.]. v. 1

MOOVIT, A. **Use Moovit in Your City.** Disponível em: <<http://moovitapp.com/cities/>>. Acesso em: 8 out. 2016.

MOREIRA, M. **Um em cada quatro brasileiros usa o ônibus como principal meio de transporte.** tipo. Disponível em: <<http://agenciabrasil.ebc.com.br/geral/noticia/2015-10/um-em-cada-quatro-brasileiros-usa-o-onibus-como-principal-meio-de-transporte>>. Acesso em: 7 out. 2016.

MUTTER, P. **Israeli transit app Moovit expands crowdsourcing community.** Disponível em:

<<http://www.geektime.com/2016/08/09/israeli-transit-app-moovit-expands-crowdsourcing-to-give-users-more-info-for-their-commute/>>. Acesso em: 23 set. 2016.

NASCIMENTO, K. **ISP divulga dados do mês de junho.** Disponível em:

<<http://www.isp.rj.gov.br/Noticias.asp?ident=358>>. Acesso em: 25 nov. 2016.

NEW JERSEY DEPARTMENT OF TRANSPORTATION. **PLANNING FOR TRANSIT-FRIENDLY LAND-USE.** jun. 1994.

NUNES, E. C. **“Waze do ônibus”, Moovit precisa do usuário para driblar falta de dados e redes - Especiais - iG.** Disponível em: <<http://tecnologia.ig.com.br/especial/2015-06-03/waze->

do-onibus-moovit-precisa-do-usuario-para-driblar-falta-de-dados-e-redes.html>. Acesso em: 8 out. 2016.

OEHLER, C. **Google Transit Graduates from Labs**. Disponível em: <<http://google-latlong.blogspot.com/2007/10/google-transit-graduates-from-labs.html>>. Acesso em: 22 fev. 2016.

PURDY, K. **Moovit Wants To Fix Your Nightmare Commute**. Disponível em: <<https://www.fastcompany.com/3004264/moovit-wants-fix-your-nightmare-commute>>. Acesso em: 16 set. 2016.

QUINTELLA, M. **Chegou a busca por Transporte Público para o Rio de JaneiroO blog do Google Brasil**, 24 set. 2009. Disponível em: <<https://brasil.googleblog.com/2009/09/chegou-busca-por-transporte-publico.html>>. Acesso em: 8 out. 2016

RIO, P. **Prefeitura anuncia aplicativos de mobilidade urbana vencedores de concurso internacional**. Disponível em: <<http://www.rio.rj.gov.br/web/guest/exibeconteudo?id=6031814>>. Acesso em: 8 out. 2016.

ROBUSTO, C. C. The Cosine-Haversine Formula. **The American Mathematical Monthly**, v. Vol. 64, No. 1, p. 38–40, jan. 1957.

ROMER, R. **App de transporte público Trafi lança versão offline para mercado brasileiro - Startups**. Disponível em: <<https://canaltech.com.br/noticia/startups/app-de-transporte-publico-trafi-lanca-versao-offline-para-mercado-brasileiro-51501/>>. Acesso em: 8 out. 2016.

SALUTES, B. **Conheça o TRAFI: o aplicativo de transporte público que funciona em modo offline**. Disponível em: <<http://www.androidpit.com.br/trafi-aplicativo-transporte-offline>>. Acesso em: 8 out. 2016.

SCHETTINO, B. D. P. **An Infrastructure for Bus Data Analyses applied To The Rio De Janeiro City**. Dissertação de Mestrado—[s.l.] Universidade Federal Fluminense - UFF, dez. 2016.

SILVA, A. C. **PREDIÇÃO ESTATÍSTICA DO TEMPO DE VIAGEM DE ÔNIBUS A PARTIR DE DADOS HISTÓRICOS**. Trabalho de Conclusão de Curso—[s.l.] Universidade Federal Fluminense - UFF, 2016.

TARKAN, F. **O Moovit chega ao Rio de JaneiroShare for the Future**, 12 abr. 2013. Disponível em: <<https://shareforthefuture.wordpress.com/2013/04/12/o-moovit-chega-ao-rio-de-janeiro/>>. Acesso em: 23 set. 2016

TRAFI, A. **TRAFI - World's most accurate public transport app**. Disponível em: <<http://www.trafi.com/>>. Acesso em: 23 set. 2016.

UBALDI, B. Open Government Data: Towards Empirical Analysis of Open Government Data Initiatives. **OECD Working Papers on Public Governance, No. 22, OECD Publishing**, p. 60, 27 maio 2013.

U.S COAST GUARD. STATEMENT BY THE PRESIDENT REGARDING THE UNITED STATES' DECISION TO STOP DEGRADING GLOBAL POSITIONING SYSTEM ACCURACY.

Disponível

em:

<<http://www.navcen.uscg.gov/?pageName=gpsSelectiveAvailability>>. Acesso em: 5 mar. 2016.

APÊNDICE A – FORMULÁRIO PARA IDENTIFICAÇÃO DE FUNCIONALIDADES DOS APLICATIVOS DISPONÍVEIS

Este formulário teve o objetivo de identificar os aplicativos mais comuns entre os usuários de ônibus, e validar quais funcionalidades estes aplicativos possuem ou não.

1. Qual aplicativo você costuma utilizar para se locomover de ônibus pela cidade do RJ? (Google Maps/ Moovit/ Trafi/ Vá de ônibus/ Outro)
2. O app sugere linhas possíveis para se pegar, dados origem e destino? (Sim/Não)
3. Escolhida uma linha, a localização dos ônibus desta linha é atualizada em tempo real no mapa? (Sim/Não)
4. O app indica a distância entre você e um destes ônibus em metros? (Sim/Não)
5. O app estima o tempo de chegada do ônibus até você? (Sim/Não)
6. O app informa se existem pontos turísticos próximos a você? (Sim/Não)
7. O app informa quais os pontos de parada do ônibus que você embarcou? (Sim/Não)
8. O app notifica quando você precisa descer do ônibus? (Sim/Não)
9. Como são estas notificações? (Na tela (notificação push)/ Por áudio/ Vibração)
10. O que você acha de uma funcionalidade que permita comparar no mapa a proximidade de ônibus de duas linhas diferentes? (Usaria bastante/ Acho legal, mas não usaria tanto/ Qual a vantagem disso no app?)
11. O que você mais sente falta neste app? Qual funcionalidade você acha que utilizaria muito caso existisse? (opcional – texto livre para resposta)

APÊNDICE B – FORMULÁRIO DE MAPEAMENTO DE FUNCIONALIDADES DO *TRACKBUS*

Este formulário foi utilizado como meio de realizar a pesquisa de opinião que gerou os resultados analisados no Capítulo 3. A pesquisa inicialmente faz uma identificação do usuário

que está respondendo, onde ele deve informar a sua localidade e frequência de utilização de ônibus. Com seu perfil identificado, ele é questionado sobre a relevância de funcionalidades de aplicativos com a mesma proposta que a apresentada neste trabalho.

1. Você vive na cidade do Rio de Janeiro ou regiões metropolitanas? (Sim/Não)
2. Qual a sua frequência de utilização de ônibus na cidade do Rio? (Diariamente/ Semanalmente/ Mensalmente/ Anualmente/ Nunca/ Apenas quando visito a cidade)

Numa escala onde 1 é totalmente irrelevante e 5 fundamental, diga o quanto lhe seria útil esta funcionalidade:

3. Sugerir rotas de viagem de acordo com um ponto de início e destino (1 a 5)
4. Estimar tempo de chegada de um ônibus a um ponto (1 a 5)
5. Indicar a distância em metros de um ônibus à sua localização atual (1 a 5)
6. Indicar a localização, em tempo real, dos ônibus de uma linha (1 a 5)
7. Comparar proximidade de ônibus de diferentes linhas no mapa (1 a 5)
8. Fornecer endereço e nome de pontos turísticos próximos à uma linha selecionada (1 a 5)
9. Notificação de quando o ônibus estiver se aproximando (1 a 5)
10. Uma vez embarcado no ônibus, notificação da proximidade de um ponto desta linha (1 a 5)
11. Uma vez embarcado no ônibus, notificação de pontos turísticos próximos dos pontos de parada desta linha (1 a 5)

Indique na mesma escala a utilidade de cada forma de notificação (referente a R7, R8 e R9)

12. Alerta vibratório (1 a 5)
13. Alerta por som (1 a 5)
14. Alertas por voz (com um texto descritivo) (1 a 5)