

UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
CURSO DE CIÊNCIA DA COMPUTAÇÃO

LUIS PAULO COSTA MORAES

**RASTREAMENTO COLABORATIVO DE
COLETIVOS BASEADO NA LOCALIZAÇÃO
DE USUÁRIOS**

São Luis
2016

LUIS PAULO COSTA MORAES

**RASTREAMENTO COLABORATIVO DE
COLETIVOS BASEADO NA LOCALIZAÇÃO
DE USUÁRIOS**

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para a obtenção do grau de bacharel em Ciência da Computação.

Orientador: Tiago Bonini Borchardt

São Luis
2016

Moraes, Luis Paulo Costa

Rastreamento colaborativo de coletivos baseado na localização de usuários/ Luis Paulo Costa Moraes. – São Luis, 2016.

69f.

Orientador: Tiago Bonini Borchartt

Monografia (Graduação) – Universidade Federal do Maranhão, Curso de Ciência da Computação, 2016.

1. Sistema de Transporte Público 2. Computação Móvel 3. Serviço Baseado em Localização 4. Android 5. Dados Geoespaciais

CDU 004.42

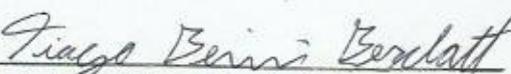
LUIS PAULO COSTA MORAES

**RASTREAMENTO COLABORATIVO DE
COLETIVOS BASEADO NA LOCALIZAÇÃO DE
USUÁRIOS**

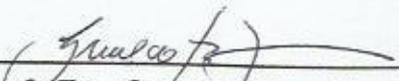
Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para a obtenção do grau de bacharel em Ciência da Computação.

Trabalho aprovado em 1 de Abril de 2016.

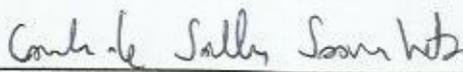
BANCA EXAMINADORA



Prof. Dr. Tiago Bonini Borchardt
(Orientador)
Universidade Federal do Maranhão



Prof. Dr. Geraldo Braz Junior
Universidade Federal do Maranhão



Prof. Dr. Carlos de Salles Soares Neto
Universidade Federal do Maranhão

*Dedico este trabalho a minha querida Avó materna que há poucos meses nós deixou.
A sua perda causou em nós uma dor imensurável, porém momentânea,
pois eu creio que um dia ela estará conosco novamente. .*

Agradecimentos

Agradeço em primeiro lugar a Deus, o autor da minha vida, pois foi Ele quem me deu esta oportunidade e desde sempre tem guiado os meus passos para que eu faça a sua vontade.

Em segundo lugar a minha família, pois ela é minha referência e foi a base formadora do meu caráter em toda minha história, orientando-me para tomar as decisões corretas, sobretudo minha mãe que nos momentos mais difíceis sempre esteve ao meu lado, demonstrando seu carinho e atenção.

Agradeço também a minha noiva que continuamente esteve ao meu lado, sendo paciente e sempre que possível me auxiliando, para que eu continuasse firme nesta grande caminhada.

Agradeço em especial o professor Tiago Bonini Borchardt que sempre mostrou total disponibilidade, acompanhando e direcionando o desenvolvimento deste trabalho. Foi muito gratificante tê-lo como orientador.

Agradeço a todos os amigos do curso de Ciência da Computação, dos mais distantes aos mais próximos, pois através das experiências trocadas ao longo de toda vida acadêmica, ensinamo-nos uns aos outros grandes lições, e sem dúvida, estas serão lembradas por toda a vida.

E por último, e não menos importante, agradeço a todos os professores do Curso de Ciência da Computação da Universidade Federal do Maranhão, pois através dos seus ensinamentos eu pude chegar onde estou. Ao departamento de informática que continuamente visou oferecer um ensino de qualidade a todos os alunos.

"Alexandre, César, Carlos Magno e até Eu mesmo fundamos grandes imperios; Porém, de que dependemos? Da força! Só JESUS CRISTO fundou o seu império sobre o amor e, nesta mesma hora, milhões estão prontos a morrer por ele"
(Napoleão Bonaparte, Imperador Francês)

Resumo

O crescimento demográfico em áreas metropolitanas tem sido constante e eminente. Grandes empresas, polos industriais, centros comerciais, entre outros, são fatores que contribuem para aumentar o número de pessoas, causando o efeito de inchaço populacional nestas regiões. Como consequência, há a necessidade de investir em transporte urbano para locomover diariamente um número significativo de passageiros para os seus destinos. Não diferente das outras capitais, São Luís vem sofrendo com este tipo de fenômeno. Nos últimos anos sua população vem aumentando, fazendo com que a qualidade do transporte público não acompanhe tal crescimento. Além de todo o problema que é inevitável, pouco se investe por parte do poder público em inovações tecnológicas para melhorar o transporte público desta capital, deixando a desejar tanto por parte da infraestrutura dos veículos quanto por parte dos setores administrativos que regem o sistema. Por outro lado, o crescimento tecnológico tem aquecido vários ramos da sociedade, e geralmente é uma alternativa a fim de apresentar soluções para a mesma. Acompanhando esta progressão, está o mercado de *smartphones* que ao longo dos últimos anos vem despertando um grande interesse das empresas para criação de aplicativos móveis nos diferentes setores da sociedade. Com diversas tecnologias agregadas a estes dispositivos e com o aumento do uso de serviços que disponibilizam informações geográficas, inúmeras aplicações são desenvolvidas para satisfazer os mais variados tipos de clientes. Sabendo-se desta realidade, este trabalho de conclusão de curso visa propor um aplicativo móvel que ofereça para usuários de transporte público, um sistema que seja capaz de fornecer informações sobre o itinerário dos ônibus de uma determinada linha, através da exibição de suas rotas em um mapa e da localização aproximada dos coletivos em tempo real, além de disponibilizar o tempo de previsão de chegada dos mesmos até um determinado ponto de ônibus. O aplicativo é desenvolvido para executar na plataforma Android, e o mesmo utilizará a tecnologia GPS e as APIs, *Maps*, *Location* e *Direction* disponibilizadas pela Google para gerar os dados geoespaciais.

Palavras-chave: Sistema de Transporte Público; Computação Móvel; Serviço Baseado em Localização; Android; Dados Geoespaciais.

Abstract

The demographic growth in metropolitan areas have been constant and imminent. Big enterprises, industrial centers, shopping centers, among others, are factors that contribute to increase the number of people, causing the effect of swelling population in these regions. As a result, there is a need to invest in urban transport for carry a significant number of passengers every day at their destinations. No different from other capitals, St. Louis Has been suffering from this type of phenomenon. In recent years its population is increasing, causing the quality of public transport not accompany such growth. In addition to all the trouble that is inevitable, little is invested by the government in technological innovations to improve public transport this capital, falling short by both the infrastructure as of the vehicles as by the administrative sectors that govern the system. On the other hand, technological development has heated various branches of the society, and is usually an alternative to providing solutions for the same. Accompanying this progression is the market of smartphones that over the past few years has aroused great interest of companies to create mobile applications in different sectors of society. With several technologies aggregate to these devices and the increased use of services that provide geographic information, numerous applications are developed to satisfy the most varied types of customers. Knowing this reality, this course conclusion work aims to propose a mobile application that offers for public transport users, a system that be is able to provide information about the itinerary of the bus of a certain line, by displaying your routes in a map and of approximate location of collective in real-time, besides providing the forecast time of arrival of the same up to a certain point by bus. The application is designed to run on the Android platform, it's the even use GPS technology and APIs, Maps, Location and Direction provided by Google to generate geospatial data.

Keywords: Public Transportation System; Mobile Computing; Location Based Service; Android; Geospatial Data.

Lista de ilustrações

Figura 1 – Localização dos terminais de integração da cidade de São Luís, capital do estado do Maranhão.	14
Figura 2 – Aplicativo CittaMobi	16
Figura 3 – Aplicativo Transit	17
Figura 4 – Aplicativo Moovit	18
Figura 5 – Modelo de Comunicação	21
Figura 6 – Gerações de Satélites	24
Figura 7 – Cálculo das Distâncias	26
Figura 8 – Configurações possíveis de PDOP	29
Figura 9 – Arquitetura do sistema operacional Android	31
Figura 10 – Ciclo de Vida de uma Activity	35
Figura 11 – Etapas do processo de construção do aplicativo representado por um diagrama de atividade	40
Figura 12 – Arquitetura da Aplicação	41
Figura 13 – As principais funcionalidades da Aplicação	42
Figura 14 – Interação do usuário com a aplicação	43
Figura 15 – Tela Principal	45
Figura 16 – Usuário fornece sua localização	46
Figura 17 – Os tipos de mapa da aplicação	47
Figura 18 – Cálculo da previsão de chegada de um ônibus até a um ponto de parada	48
Figura 19 – Correção da coordenada gerada fora da via	49
Figura 20 – Precisão da localização GPS (Biblioteca/Terminal)	53
Figura 21 – Precisão da localização GPS (Terminal/CEPRAMA)	54
Figura 22 – Precisão da localização GPS (Terminal/CCET)	55
Figura 23 – Itinerário do Campos Integração. Lado (a) da figura mostra o sentido TU e o lado (b) da figura o sentido UT	56
Figura 24 – Gráfico que compara o tempo estimado com o tempo de chegada para os experimentos do percurso Terminal/Paulo Freire.	58
Figura 25 – Gráfico que compara o tempo estimado com o tempo de chegada para os experimentos do percurso Terminal/CCET.	59

Lista de tabelas

Tabela 1 – Informações Sobre os Modelos dos Satélites	23
Tabela 2 – Informações sobre os sinais que cada modelo de satélite transmite. . . .	26
Tabela 3 – Informações sobre o experimento de precisão GPS (Biblioteca/Terminal)	52
Tabela 4 – Informações sobre o experimento de precisão GPS (Terminal/CEPRAMA)	53
Tabela 5 – Informações sobre o experimento de precisão GPS (Terminal/CCET) .	54
Tabela 6 – A sequência dos pontos de parada do itinerário nos dois sentidos. . . .	56
Tabela 7 – Informações sobre os experimentos de tempo estimado para o percurso Terminal/Paulo Freire.	57
Tabela 8 – Informações sobre os experimentos de tempo estimado para o percurso Terminal/CCET.	58

Lista de siglas

ABINEE Associação Brasileira da Indústria Elétrica e Eletrônica

API Application Programming Interface

CCBS Centro de Ciências Biológicas e da Saúde

CCET Centro de Ciências Exatas e Tecnológicas

CCSO Centro de Ciências Sociais

CEPRAMA Centro de Produção Artesanal do Maranhão

CPU Unidade Central de Processamento

GPS Global Positioning System

HTTP Hyper Text Transfer Protocol

IBGE Instituto Brasileiro de Geografia e Estatística

IDE Ambiente Integrado de Desenvolvimento

JVM Java Virtual Machine

DVM Dalvik Virtual Machine

OHA Open Handset Alliance

RAM Random Access Memory

SMTT Secretaria Municipal de Trânsito e Transporte

SIG Sistema de Informação Geográfica

SIT Sistema Integrado de Transporte

TRE-MA Tribunal Regional Eleitoral do Maranhão

TV Televisão

Wi-Fi Wireless Fidelity

Sumário

1	INTRODUÇÃO	13
1.1	Motivação	14
1.2	Objetivos	15
1.2.1	Objetivo Geral	15
1.2.2	Objetivos Específicos	15
1.3	Soluções Existentes	16
1.3.1	CittaMobi	16
1.3.2	Transit App	17
1.3.3	Moovit	17
1.4	Organização do Trabalho	18
2	FUNDAMENTAÇÃO TEÓRICA	19
2.1	Computação Móvel	19
2.2	Serviço Baseado em Localização	20
2.2.1	Taxonomia	20
2.2.2	Modelo de Comunicação	21
2.3	Sistema de Posicionamento Global	22
2.3.1	Arquitetura	23
2.3.2	Funcionamento	25
2.3.3	Erros que podem afetar o desempenho do sistema	28
2.4	Android	29
2.4.1	O Sistema Operacional	30
2.4.2	Open Handset Alliance	30
2.4.3	Arquitetura	31
2.4.4	Dalvik	32
2.4.5	Ambiente de Desenvolvimento	33
2.4.6	Activity	34
2.4.7	Gerenciando o Ciclo de Vida de uma Atividade	34
2.4.8	Métodos que podem alterar o estado de uma atividade	36
2.5	Google API	37
2.5.1	Google Maps	37
2.5.2	Google Location	37
2.5.3	Google Direction	38

3	SERVIÇO DE LOCALIZAÇÃO DE COLETIVOS	39
3.1	Desenvolvimento	39
3.1.1	Metodologia	39
3.1.2	Arquitetura da Aplicação	40
3.1.3	Modelagem	42
3.1.4	Ferramentas Utilizadas	43
3.2	Aplicativo	44
3.2.1	Principais Funcionalidade	44
3.2.2	Funcionamento	44
3.2.3	Tempo de Previsão de Chegada dos Coletivos	47
3.2.4	Correção do Posicionamento Gerado pelo GPS	48
4	EXPERIMENTOS E ANÁLISE DOS RESULTADOS	50
4.1	Método para a Avaliação	50
4.2	Experimentos	51
4.2.1	Precisão da Localização dos Ônibus Utilizando a Posição GPS	51
4.2.2	Tempo Estimado de Chegada de um Coletivo em um Determinado Ponto de Parada	55
4.3	Avaliação dos Resultados	59
4.3.1	Análise em Relação a Precisão do Sinal GPS	60
4.3.2	Análise em Relação ao Tempo Estimado de Chegada de um Coletivo	61
5	CONCLUSÃO	63
5.1	Trabalhos Futuros	65
	REFERÊNCIAS	67

1 Introdução

Ao olhar em volta, nota-se que os meios de transporte proporcionam locomoção e realizam tarefas básicas para o desenvolvimento da sociedade como um todo. São de grande importância para o crescimento econômico de uma região ou de um país, visto que são responsáveis pelo deslocamento de pessoas, animais, mercadorias, minérios, etc.

Lugares com grande fluxo de pessoas precisam de meios que proporcionem o deslocamento das mesmas. Em São Luís, existem várias empresas de ônibus (SOUSA, 2013) que oferecem aos habitantes da ilha do Maranhão, 168 linhas urbanas, semiurbanas e metropolitanas com propósito de transportar estas pessoas a qualquer lugar da grande São Luís (FERREIRA et al., 2015).

O sistema integrado de transporte, conhecido como SIT São Luís, foi criado em 1996 com o dever de garantir a integração e a locomoção de passageiros por toda a cidade. O SIT é administrado pela SMTT (Secretaria Municipal de Trânsito e Transporte), e se baseia em cinco terminais para fazer o transporte de milhares de pessoas todos os dias. Os terminais que fazem parte do SIT são: o Terminal Praia Grande, o Terminal São Cristóvão, o Terminal Cohab-Cohatrac, o Terminal Cohama-Vinhais e o Terminal Distrito Industrial, como pode ser visto na Figura 1. Os ônibus seguem um esquema de identificação de cores por consórcio. Cada consórcio faz parte de um terminal, com exceção do consórcio Rio Anil, o qual possui apenas linhas não integradas (SOUSA, 2013).

São Luís tem 834,785 km² e possui uma população de aproximadamente 1.073.893 habitantes, conta com uma frota de 347.584 veículos automotivos, no entanto, apenas 5.376 veículos são destinados para o transporte coletivo de pessoas, enquanto 342.208 deles são de carros próprios, caminhões, motocicletas e dentre outros tipos de veículos. (IBGE, 2014).

Percebe-se que São Luís possui uma quantidade muito maior de automóveis em relação ao de transportes coletivos. Por outro lado, o número de usuários de coletivos é maior em relação ao de pessoas que podem ter um carro próprio (não é diferente em muitas outras capitais brasileiras), de acordo com a Assessoria de Comunicação da SMTT, que afirma que a frota atual de ônibus em São Luís é de 874 coletivos e a demanda total de usuários do SIT é cerca de 760 mil pessoas (SMTT, 2016).

Como pode ser visto, apesar de mais de 70% da população metropolitana ludovicense fazer uso de coletivos, poucos ônibus são oferecidos para os cidadãos da capital maranhense. Contudo, a grande questão é como este tipo de serviço está sendo prestado para os passageiros, pois quantidade de coletivos não garante qualidade no serviço oferecido.

Pesquisas apontam que nas mais importantes capitais brasileiras, a maioria dos usuários de transporte público estão insatisfeitos com os serviços prestados (PROTESTE, 2015). No Maranhão, não é diferente, em Imperatriz e municípios vizinhos, muitos usuá-

Figura 1 – Localização dos terminais de integração da cidade de São Luís, capital do estado do Maranhão.



Fonte: Google Maps e Sousa (2013)

rios reclamam de atrasos dos coletivos e de como as empresas responsáveis pelo ofício estão oferecendo o serviço (IMIRANTE, 2011). Em São Luís a situação é a mesma (G1, 2015), pois pouco se investe em infraestrutura para o sistema de transporte que ao longo dos anos tornou rapidamente defasado, do mesmo modo que pouco se estimula em soluções inovadoras para melhorar o serviço, pois quase não se vê recursos tecnológicos inseridos no mesmo.

1.1 Motivação

Pessoas dependem do transporte público para se deslocarem para seus parapeiros e muitas vezes precisam se organizar para fazerem suas tarefas domésticas rapidamente para não perderem o horário dos meios de transporte que os levarão ao lugar desejado. Muitas delas deixam de pegar seus coletivos na hora exata, ocasionando atrasos na chegada aos seus destinos. Outras pessoas ficam longas horas ociosas em pontos de ônibus, por não saberem quando exatamente um coletivo de seu interesse irá passar em um exato lugar. A falta de informação sobre horários dos coletivos que supostamente passaram por determinados pontos de ônibus é ainda um problema de milhares de usuários de transporte

público coletivo.

Sabendo desta realidade, este trabalho visa propor uma solução para o mesmo, desenvolvendo um aplicativo que informará para os usuários de transporte público, estimativas de quando determinados coletivos passarão em um determinado ponto de ônibus, assim como também disponibilizará informações sobre o trajeto da linha de ônibus de seu interesse. Os indivíduos serão consumidores dessas informações provenientes do aplicativo, da mesma maneira que também serão geradores das mesmas, pois a aplicação usará o conceito de software colaborativo.

Desta forma, espera-se que as pessoas que possuírem este aplicativo em seus dispositivos móveis, terão informações suficientes para escolherem as melhores opções de deslocamento para seus destinos, com a garantia de pegar seu transporte com exatidão no ponto mais próximo.

1.2 Objetivos

1.2.1 Objetivo Geral

Este trabalho tem como finalidade desenvolver um aplicativo que auxiliará os usuários de transporte público coletivo da Cidade de São Luís. A implementação do projeto possibilitará que os habitantes da cidade de São Luís tenham acesso às informações das linhas de ônibus que circulam por toda região metropolitana através da exibição de mapas que serão mostrados na aplicação, onde demonstrarão a localização aproximada dos coletivos no decorrer de seus trajetos e estimarão o tempo de chegada dos ônibus até os pontos de parada pertencentes ao itinerário.

1.2.2 Objetivos Específicos

- Aplicar os conhecimentos adquiridos a respeito da linguagem JAVA fazendo um aplicativo para dispositivos móveis que possuem a plataforma Android.
- Utilizar o software Android Studio integrado com a ferramenta SDK (Software Development Kit) para o desenvolvimento da aplicação.
- Fazer uma aplicação utilizando a arquitetura de comunicação cliente-servidor.
- Estudar como as tecnologias para dispositivos móveis estão utilizando e disponibilizando o recurso de mapas nas atuais aplicações.
- Estudar as atuais tecnologias de rastreamento de objetos por satélites e quais estão disponíveis para o uso civil.
- Como o GPS pode emitir erros de precisão de localização, será necessário criar uma rotina que consiga amenizar estas imprecisões e assim o usuário possa sempre

visualizar no mapa do aplicativo coletivos exatamente sobre a via e não próximo a ela.

- Utilizar a API do serviço de mapas da Google para exibir o mapa que será utilizado na aplicação, a API Google Location para fornecer a localização do usuário e a Google Direction para gerar o itinerário e fornecer as previsões de tempo.
- Utilizar um banco de dados para armazenar as informações a respeito da localização dos coletivos, assim como processar e disponibilizar estas informações para os usuários.

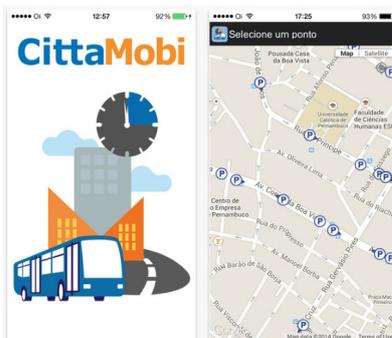
1.3 Soluções Existentes

A seguir, serão apresentadas algumas soluções já consolidadas no mercado sobre aplicativos que oferecem aos usuários informações relacionadas aos meios de transportes de suas regiões.

1.3.1 CittaMobi

CittaMobi é um conjunto de soluções com objetivo de atender às necessidades dos usuários de ônibus. Leva informações de cada ônibus, ponto de parada e linha aos usuários dos coletivos (CITTAMOB, 2014). Este tipo de serviço está presente em algumas cidades brasileiras, e as principais funcionalidades do aplicativo são fornecer a previsão em tempo real dos coletivos em relação aos pontos de parada pertencente a linha dos mesmos, proporcionar os pontos de ônibus mais próximos ao usuário e quais as linhas passam em cada um deles, além de mostrar também quais ônibus são adaptados para deficientes físicos. Este aplicativo está disponível para as versões *mobile*, *web* e em painéis para terminais.

Figura 2 – Aplicativo CittaMobi

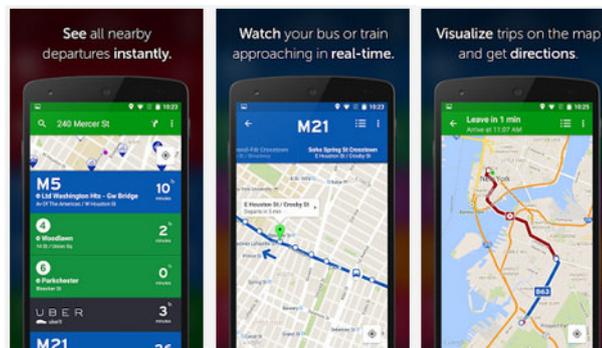


Fonte: CittaMobi (2014)

1.3.2 Transit App

O Transit é um serviço de dados sobre transporte público. Foi lançado primeiramente nos Estados Unidos em 2005, mas hoje está presente em mais de 2,8 mil cidades em todo o mundo, inclusive no Brasil. O recurso fornecer informações sobre todas as opções de transporte e os horários de partida nas proximidades, além de disponibilizar exatamente onde um ônibus ou trem está no mapa em tempo real. Os usuários podem também ter acesso as previsões de chegada dos mesmos, ver horários e itinerários de rota mesmo se estiver sem acesso à internet (*offline*), ter informações sobre os pontos de ônibus ou de metrô, como por exemplo, quais linhas passarão em determinados horários.

Figura 3 – Aplicativo Transit



Fonte: Transit (2015)

1.3.3 Moovit

O moovit é um aplicativo que funciona como uma espécie de comunidade de compartilhamento de informações entre os usuários sobre as condições de transporte. Ao fim de cada trajetória um usuário pode compartilhar para os demais um relatório de qualidade, apontando qualidades e defeitos, como atrasos, veículos excessivamente lotados ou sujo, e também qualificar o desempenho dos motoristas de ônibus e táxi. Dispõe para os usuários cálculos de distância e tempo de duração de uma viagem independente do meio de transporte escolhido. Proporciona também o monitoramento em tempo real do percurso de um veículo, seja ele ônibus, trem ou metrô. Atualmente o moovit está operante em apenas algumas cidades brasileiras, como é o caso de Fortaleza, São Paulo, Rio de Janeiro, Belo Horizonte, Porto Alegre, Curitiba, Campinas, Bauru, Itajaí e João Pessoa (MOOVIT, 2014).

Figura 4 – Aplicativo Moovit



Fonte: Moovit (2014)

1.4 Organização do Trabalho

Esta trabalho está estruturado em cinco capítulos. O primeiro capítulo, a introdução, apresentou o problema, a motivação para solucionar este problema, os objetivos geral e específicos, e por fim algumas soluções similares já existentes no mercado.

No segundo capítulo será abordada toda fundamentação teórica, que por sua vez, serviu de base para o desenvolvimento deste trabalho. Nele será visto conceitos como computação móvel, serviço baseado em localização, sistema de posicionamento global, a plataforma Android e as APIs da Google que foram utilizadas para a construção do aplicativo.

No terceiro capítulo será descrito o desenvolvimento do serviço de localização de coletivos. Neste capítulo serão apresentadas a modelagem do sistema através da demonstração de alguns dos principais diagramas da UML, os procedimentos metodológicos, as etapas, meios e ferramentas que foram utilizadas para o desenvolvimento da aplicação, as suas principais funcionalidades e uma breve descrição do funcionamento da mesma.

No quarto capítulo serão apresentados os testes realizados durante a elaboração da aplicação, assim como os resultados obtidos na fase de testes, analisando os dados obtidos quando o aplicativo foi executado em situações reais.

No quinto capítulo contém a conclusão e algumas propostas de trabalhos futuros, ao final deste, citam-se as fontes bibliográficas utilizadas nesta pesquisa.

2 Fundamentação Teórica

Neste capítulo será abordado o crescimento da computação móvel nos últimos anos, seu objetivo, o conceito de mobilidade e suas principais características. Depois será apresentado os serviços baseado em localização, sua taxonomia e seu modelo de comunicação. Logo após será visto o sistema de posicionamento global, assim como sua arquitetura, seu funcionamento e erros que podem afetar seu desempenho. Também será demonstrado o sistema operacional Android, bem como sua arquitetura, o ambiente de desenvolvimento que envolve esta plataforma e a forma que os ciclos de vida das activities se comportam no decorrer da execução de uma aplicação. E por último falar-se-á sobre as APIs Maps, Location e Direction, cujo estas mesmas serão usadas na construção do aplicativo proposto por este trabalho.

2.1 Computação Móvel

A ciência da computação agrega várias áreas, uma delas é a computação móvel, que ultimamente vem sendo reconhecida como o novo paradigma computacional. Este paradigma visa prover ao usuário acesso a serviços independentemente da sua localização, ou se o mesmo sofrer alguma mudança de local no decorrer do tempo. A questão principal da computação móvel é justamente esta, a mobilidade, a qual a computação tradicional formada por computadores estáticos não consegue prover. As principais classes de itens de computação móvel são os computadores portáteis e os telefones celulares, também conhecidos como dispositivos móveis.

A computação móvel tem como principais objetivos garantir que dispositivos portáteis, independente do tipo, tenha a capacidade de se comunicar a qualquer momento com parte fixa da rede e possivelmente com os outros computadores móveis distribuídos em qualquer lugar do planeta. Isso é possível graças a comunicação sem fio que elimina a necessidade de um aparelho móvel estar conectado a uma infraestrutura fixa. Portanto, é a comunicação sem fio que garante o enorme sucesso da computação móvel nos dias de hoje, e segundo Loureiro e Mateus (2015) os ambientes de computação ubíqua deverão ser a nova forma de trabalho da próxima década, sendo este um cenário desafiador e estimulante que tende a motivar a computação móvel.

Destacam-se como principais características da computação móvel a conectividade, a descentralização, a diversidade, e a simplicidade. Onde, a primeira delas, a conectividade, visa garantir a troca de dados entre os dispositivos conectados a uma rede sem fio. As informações devem ser facilmente trocadas, independentemente do tipo do dado ou do tipo da aplicação. A segunda característica, a descentralização, visa permitir que uma variedade de diferentes dispositivos possam cooperar entre si, mesmo que eles estejam em

lugares distintos, com o propósito de estabelecer uma rede dinâmica de relacionamento. A terceira, está intrínseca na segunda característica, pois a mesma visa prover que os mais variados tipos de dispositivos trabalhem juntos em um mesmo objetivo. E a quarta e última característica da computação móvel, é a simplicidade. Esta qualidade visa oferecer aos usuários sistemas usuais, ou seja, sistemas fáceis de serem utilizados.

Atualmente, a computação móvel esta tendo seu destaque por motivos que não estão somente dentro do interesse científico. Com enorme investimento de grandes empresas no crescente mercado de dispositivos móveis, há um constante aumento no número de vendas de aparelhos, como é o caso dos *smartphones*, que nos últimos anos chegam a representar 76% do mercado de celulares segundo a Abinee (2014), gerando um constante estímulo para o avanço da tecnologia móvel. Em consequência destes fatos, os fabricantes desses aparelhos preocupam-se cada vez mais em agregar variados utensílios aos mesmos, como forma de os tornarem mais funcionais e versáteis, como é o caso dos serviços baseados em localização, providos pela tecnologia de rastreamento por GPS, que está presente na maioria esmagadora dos *smartphones* encontrado no mercado.

2.2 Serviço Baseado em Localização

Serviços baseados em localização (LBS) podem ser definidos como serviços que integram a localização de um dispositivo móvel com outras informações de modo a proporcionar valor agregado ao usuário (SCHILLER; VOISARD, 2004). Estes serviços levam em consideração as informações do usuário, como por exemplo, seu contexto, seu perfil e os locais que frequenta, a fim de fornecer informações que atendam às suas necessidades.

O serviço de localização surgiu na década 1970, quando o Departamento de Defesa dos Estados Unidos começou a operar o sistema de posicionamento global (GPS). No seu início foi concebido para fins militares, mas na década de 1980 começou a ser disponível para o uso civil, a fim de estimular a inovação em torno da tecnologia por satélite. A partir disso, empresas de todo o mundo têm agregado este tipo de serviço aos seus produtos. Contudo, o interesse generalizado sobre o LBS começou a aumentar apenas no fim da década de 1990.

2.2.1 Taxonomia

Existem várias abordagens para classificar aplicações LBS. Uma delas, é se ela é uma aplicação orientada a pessoa ou uma aplicação orientada a dispositivo. A primeira tem como objetivo posicionar uma pessoa ou usar a posição desta para melhorar um serviço, ao passo que a segunda, a localização de apenas uma pessoa não é necessariamente o foco. Em vez de apenas uma pessoa, um objeto ou um grupo de pessoas também pode ser localizado, como por exemplo, o rastreamento de um carro para recuperação após o mesmo ter sido roubado.

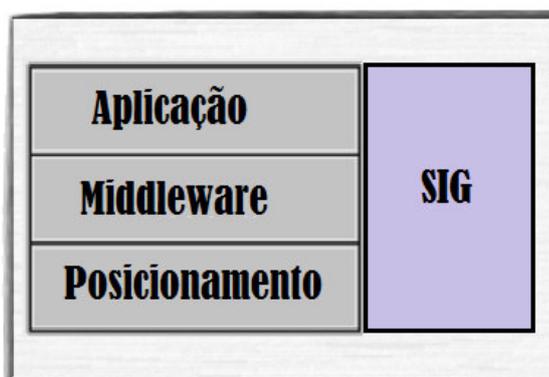
Além desta classificação, os LBS podem ser categorizados como *push services* e *pull services* (SCHILLER; VOISARD, 2004). O serviço *push* acontece quando um usuário recebe informações como resultado de seu paradeiro sem ter de solicitá-lo ativamente. Estas informações podem ser enviadas com ou sem o consentimento prévio do usuário. O serviço *pull* acontece quando o usuário utiliza uma aplicação e ativamente busca (puxa) a informação a partir da rede. Esta informação pode ser a localização de algum lugar, como por exemplo, encontrar o restaurante que serve comida chinesa mais próximo do usuário.

Os serviços mais usados ainda são os *pull services*, especialmente os serviços de informação. Em oposição, os *services push* ainda são discretamente utilizados. As principais razões para esta situação, está primeiramente relacionada por fatores de custo, pois os serviços *push* consomem uma ampla quantidade de recursos de rede por exigirem constantes atualizações de locais dos usuários, elevando o preço do serviço. E a segunda razão, são as preocupações sobre como manter a privacidade do usuário.

2.2.2 Modelo de Comunicação

A realização de um serviço baseado em localização pode ser descrito como um modelo de três camadas. A camada de posicionamento, a camada de middleware e a camada de aplicação, como pode ser visto na figura 5 abaixo. Este modelo foi desenvolvido e apresentado por Schiller e Voisard (2004).

Figura 5 – Modelo de Comunicação



Fonte: Adaptado de Schiller e Voisard (2004)

A primeira camada, a de Posicionamento, é responsável pelo cálculo da posição de um dispositivo (ou usuário). Estas informações são obtidas através do equipamento de determinação de posição (PDE) e de um sistema de informação geográfica (SIG).

O PDE calcula onde um dispositivo está em termos de rede e o SIG traduz essa informação para dados geoespaciais (latitude e longitude). O resultado final deste cálculo é repassado ou para uma plataforma de middleware ou diretamente para um aplicativo.

Apesar de existir a possibilidade de enviar dados diretamente para a camada de aplicação, é aconselhável manter uma camada de middleware para fazer a conexão entre as duas extremidades. O middleware pode reduzir a complexidade da integração dos serviços no modelo, pois é constituído de APIs (*Application Program Interface*). Estas servem de interface entre a aplicação e o dispositivo em si, proporcionando a independência entre as partes.

O objetivo do middleware é mascarar a heterogeneidade e fornecer um modelo mais produtivo para os programadores de aplicativo LBS, uma vez que por meio deste recurso, a camada de posicionamento vê o a camada de aplicação como uma única entidade, sendo necessário apenas uma integração, independente da quantidade de aplicações LBS reais existentes nela. Para a camada de aplicação, as aplicações também veem um único serviço de posicionamento, e através de uma única API, podem localizar dispositivos, obter imagens de mapas, enviar e receber mensagens, assim como tantas outras informações geoespaciais (SCHILLER; VOISARD, 2004).

A terceira e mais alta camada, a de aplicação, é responsável por entregar os dados de localização aos serviços (aplicações) que os solicitaram. Tais serviços integram estes dados em suas ofertas.

Na próxima sessão será apresentado em detalhes o mais clássico exemplo de aplicação LBS existente, o sistema de posicionamento global.

2.3 Sistema de Posicionamento Global

O Sistema de Posicionamento Global (GPS) é um dos sistemas de navegação global por satélite existente e operante no mundo. É de propriedade norte-americana e sua principal funcionalidade é proporcionar a localização de pessoas ou objetos em qualquer lugar da superfície terrestre em tempo real. Além de fornecer a um receptor os dados de localização, pode também proporcionar informações de condições atmosféricas, velocidade de deslocamento, direcionamento e informações horárias.

Foi originalmente denominado NAVSTAR (*Navigation System with Timing and Ranging*). Desenvolvido em 1973 pelo DoD (Departamento de Defesa) do Governo dos Estados Unidos e inicialmente idealizado para uso militar e aberto gratuitamente para uso civil a partir da década de 80 (EMBRAPA, 2013).

O GPS vem sendo usado para diferentes fins: na aviação geral e comercial; na navegação marítima; em armamentos de precisão e etc. Há muito tempo que o GPS contribui para o desenvolvimento de grandes países, porém, com o avanço da tecnologia *mobile* nos últimos anos, este vem sendo agregado a aplicações que visão contribuir na realização de

tarefas do cotidiano das pessoas, como é o caso de aplicações que proporcionam soluções para os sistemas de transporte de superfície.

Atualmente outros sistemas de navegação global por satélite similares ao GPS estão em operação ou em desenvolvimento, como é o caso do Sistema de Navegação Global por Satélite da Rússia (GLONASS), o sistema Europeu GALILEO e o sistema Chinês COMPASS (EMBRAPA, 2013).

O Sistema de Posicionamento Global é composto de três segmentos: o segmento espacial, o segmento de controle e o segmento do usuário. Estes serão apresentados na próxima seção.

2.3.1 Arquitetura

Segmento Espacial

O sistema foi inicialmente composto por 24 satélites operantes e três reservas (que eram usados somente caso algum dos operantes fosse retirado ou viesse a falhar). Em 2011, houve uma expansão espacial onde os três satélites reservas passaram a fazer parte da linha de base da constelação inicial. São distribuídos em seis planos orbitais quase circulares e com períodos de aproximadamente 12 horas, isto é, os satélites dão duas voltas na terra por dia. Possuem uma inclinação de cinquenta e cinco graus em relação à linha do Equador e uma altitude de aproximadamente de 20200 km (GPS, 2015d).

Os satélites podem ser divididos em blocos: os satélites de projeto (protótipos); satélites de produção (operacionais); satélites de reposição (reservas). O primeiro bloco, os protótipos, foram os primeiros a serem lançados no espaço entre os anos de 1978 a 1985 e hoje já estão todos desativados. O segundo bloco, os operacionais, foram lançados entre os anos de 1989 a 1996, e a partir da década de 90 foram lançados satélites operacionais mais sofisticados que os primeiros. O terceiro bloco, os reservas, são os que substituem os satélites do segundo bloco, quando necessário, e o primeiro a ser lançada foi no ano 1997 (EMBRAPA, 2013). Pode-se ver os blocos de satélites existentes e quais foram lançados nos últimos anos na Tabela 6.

Tabela 1 – Informações Sobre os Modelos dos Satélites

BLOCO IIA	BLOCO IIR	BLOCO IIR(H)	BLOCO IIF	GPS III
Projeto com alcance de 7.5 anos de vida	Projeto com alcance de 7.5 anos de vida	Projeto com alcance de 7.5 anos de vida	Projeto com alcance de 12 anos de vida	Projeto com alcance de 15 anos de vida
Lançado em 1990-1997	Lançado em 1997-2004	Lançado em 2005-2009	Lançado desde 2010	Começa a lançar em 2016

Fonte: GPS (2015d)

Para garantir o bom funcionamento do sistema, os Estados Unidos lançou 31 satélites nos últimos anos. Todas as gerações de satélites podem ser vistas na Figura 6 e os tipos de sinais na Tabela 2. As principais características de cada modelo foram apresentadas na Tabela 1.

Figura 6 – Gerações de Satélites



Fonte: GPS (2015d)

Segmento de Controle

É composto por 16 estações monitoras e 12 antenas terrestres, todas espalhadas pelo planeta, juntamente com uma estação de controle mestre e uma estação de controle alternativa, as duas situadas no território americano, a primeira localizada no Colorado e a segunda localizada na Califórnia. As estações monitoras recebem todos os sinais dos satélites, a estação de controle mestre reformula novos dados baseados nos sinais recebidos, e as antenas terrestres enviam os novos dados para os satélites. Todos estes componentes estão espalhados estrategicamente pela superfície terrestre para que o sistema tenha um melhor desempenho (GPS, 2015a).

Toda a estrutura do segmento de controle tem como finalidade: controlar e supervisionar continuamente todo o sistema; atualizar a posição atual dos satélites; e sincronizar o relógio atômico presente em cada um dos satélites. Qualquer instabilidade detectada é corrigida pelo Departamento de Defesa dos Estados Unidos baseada em sistemas de processamento automático.

Segmento de Usuários

É a parte do sistema que interage diretamente com os usuários. O receptor GPS compreende de uma antena para captar os sinais dos satélites e um receptor para decodificar o sinal dos diferentes satélites através de multicanais paralelos. Apesar de hoje no mercado ter diferentes tipos de aparelhos GPS, todos são capazes de receber os sinais dos satélites e

usar essas informações para calcular a posição e o tempo. Geralmente são aparelhos que possuem uma interface para apresentar as informações desejadas pelos usuários, como por exemplo, seu posicionamento, sua altitude, a hora corrente, sua direção, velocidade de seu deslocamento e entre outras funcionalidades variando de equipamento para equipamento.

Atualmente a maioria dos fabricantes de *smartphones* produzem seus aparelhos com receptores GPS embutidos, da mesma forma que no mercado de desenvolvimento de aplicativos já existem centenas de aplicações que usam o sistema GPS como referência.

2.3.2 Funcionamento

Sinais

Dentro de cada satélite existem osciladores que geram uma frequência fundamental $f = 10,23\text{MHz}$. A partir desta frequência são geradas duas outras frequências moduladas pelas portadoras L_1 e L_2 , e através delas são transmitidos os sinais dos satélites. Estes sinais podem ser códigos e mensagens (RODRIGUES, 1996). Atualmente a um programa do governo americano na modernização do sistema GPS onde terá a adição de novos sinais para o uso civil (GPS, 2015c).

Os códigos são sequencias denominadas de código PRN (*Pseudo Radon Noiser*). Existem dois tipos, o código P (Precision) que é modulado pelas duas portadoras L_1 e L_2 e o código C/A (Coarse/Aquisition) modulado apenas pela portadora L_1 . Dos dois códigos, o P é o mais preciso (RODRIGUES, 1996). As mensagens são dados de navegação e sistema, consistindo de efemérides¹, status do sistema e relógios.

$$L_1: 154 \times f = 1575,42 \text{ MHz}$$

$$L_2: 120 \times f = 1227,60 \text{ MHz}$$

Existem duas categorias de sinais que quando ativados limitam a precisão do sistema GPS para pessoas comuns (não autorizadas). Isto tem o propósito de restringir o uso do sistema por civis ou por órgãos que pudessem ameaçar a segurança dos Estados Unidos.

Os dois tipos de sinais que limitam a precisão do sistema são o Anti-Spoofing (AS) e o Selective Availibilty (SA). Em maio de 2000 o presidente dos Estados Unidos Bill Clinton deu ordem para desativar o recurso SA como primeiro passo para modernização do sistema GPS, beneficiando utilizadores comuns por todo o mundo. Em 2007 com o programa de modernização GPS III, satélites serão construído sem o sinal limitador (GPS, 2015b).

Pode-se ver na Tabela 2 os tipos de sinais que cada modelo de setélite pode emitir de acordo com o bloco em que cada um pertence.

¹ É um conjunto de parâmetros, os quais descrevem a órbita de um satélite e sua posição ao longo de uma dada órbita em um dado momento (RGCI, 2007)

Tabela 2 – Informações sobre os sinais que cada modelo de satélite transmite.

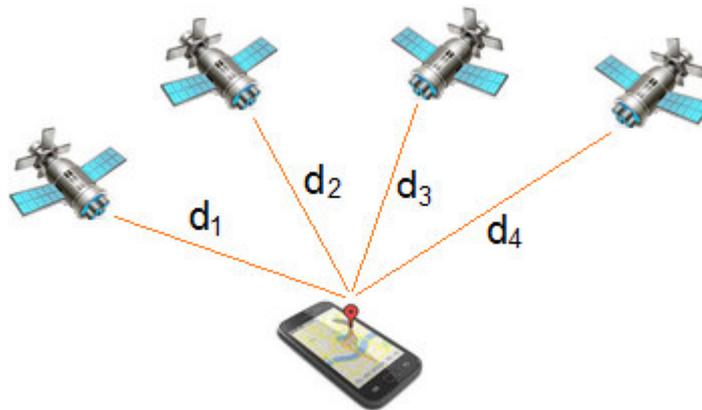
BLOCO IIA	BLOCO IIR	BLOCO IIR(H)	BLOCO IIF	GPS III
Código C/A (Coarse/Aquisition) modulado pela portadora L_1 para uso civil.	Código C / A em L_1 .	Segundo sinal civil na L_2 (L2C).	Terceiro sinal civil na frequência L_5 (L_5).	Quarto sinal civil em L_1 (L_1C).
P (Y) modulado pelas duas portadoras L_1 e L_2 para usuários militares.	P (Y) código em L_1 e L_2 .	Níveis de potência flexíveis para sinais militares.	Maior precisão, intensidade do sinal melhorada e maior qualidade.	Confiabilidade do sinal melhorada, maior precisão e integridade.

Fonte: GPS (2015d)

Determinação das Distâncias

Para se encontrar as distâncias dos satélites visíveis em relação ao usuário precisa saber as coordenadas do posicionamento dos satélites e as coordenadas do usuário. As coordenadas dos satélites são conhecidas, pois os próprios satélites preveem suas trajetórias nas efemérides e transmitem suas coordenadas a cada 30 segundos através de mensagens de navegação (RODRIGUES, 1996).

Figura 7 – Cálculo das Distâncias



Usando conceitos de álgebra linear as distâncias podem ser descobertas através de um sistema de i equações e i incógnitas (i é o número de satélites no sistema), formado pelas equações das distâncias de cada satélite em relação ao usuário (ponto μ) apresentadas abaixo (AVRITZER, 2009, p.27).

$$d_1 = \sqrt{(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2} \quad (1)$$

$$d_2 = \sqrt{(x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2} \quad (2)$$

$$d_3 = \sqrt{(x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2} \quad (3)$$

onde:

$\mu (x, y, z)$ – As coordenadas do usuário.

$p_i (x_i, y_i, z_i)$ - As coordenadas dos satélites.

d_i - As distâncias dos satélites em relação ao usuário.

i - é o número de satélites no sistema.

Substituindo os valores das coordenadas do usuário juntamente com os valores correspondentes às coordenadas da localização de cada satélite se obterá os valores das distâncias de cada satélite em relação a um usuário μ .

Para se saber a posição de um usuário na superfície da terra usa-se as mesmas equações. Mas para isso precisa-se saber:

- a) As coordenadas do posicionamento dos satélites (P_i), no mínimo três satélites para obter o posicionamento bidimensional, e quatro satélites para obter o posicionamento tridimensional. Como a disposição dos satélites em órbita garante que sempre haja pelo menos quatro deles disponíveis em qualquer lugar do planeta (GPS, 2015d), será sempre necessário calcular quatro distâncias, d_1, d_2, d_3, d_4 ;
- b) E a distância do receptor até os satélites (d_i).

Como alternativa, pode-se escrever uma equação genérica baseada nas equações (1), (2) e (3) apresentada acima para se calcular a posição do usuário.

$$d_i = |X_i - X_r| \quad (4)$$

onde:

X_i - vetor de posição de cada satélite (x_i, y_i, z_i);

X_r - vetor de posição do ponto $\mu(x_r, y_r, z_r)$;

d_i - As distâncias dos satélites em relação ao usuário.

i - é o número de satélites no sistema.

Para terminar a velocidade do ponto μ caso ele venha sofrer algum deslocamento ao longo do tempo, basta calcular a taxa de variação da posição em relação ao tempo.

Na prática, para se determinar as distâncias entre os satélites e o receptor do usuário, o sistema GPS relaciona-se o tempo e velocidade de propagação do sinal entre o satélite e o receptor. O tempo de propagação é obtido pela diferença entre o instante que o sinal é emitido pelo satélite (Tt) e o instante que o sinal é ouvido pelo receptor (tr).

Considerando que existe um erro de sincronismo (dtu) que é repassado para as distâncias (d_i), serão gerados valores não exatos para elas, ou seja, gerando as chamadas pseudodistâncias. Um dos métodos para calcular a pseudodistância é o método de medição de código.

Medição de Código

Considerando t como o tempo de propagação do sinal.

$$\tau = (tr - Tt) \quad (5)$$

Pode-se obter a pseudodistância através da equação abaixo:

$$Pd = c.\tau = c.(tr - Tt) \quad (6)$$

onde:

c - é velocidade em que o sinal é emitido (velocidade da luz).

Com as pseudodistâncias calculadas volta-se a equação (4), e acrescentando a ela, o erro de sincronismo do relógio. Desta forma, será possível obter a localização aproximada do usuário.

$$Pd_i = |X_i - X_r| + c.dtu \quad (7)$$

onde:

dtu - é o erro de sincronismo do relógio. Este erro é solucionado quando o receptor rastreia em um mesmo instante o sinal de quatro ou mais satélites para encontrar sua localização.

2.3.3 Erros que podem afetar o desempenho do sistema

Erros de Posição dos Satélites

São inexatidões do local informado do satélite ocasionadas pela diferença de localização real que os satélites estão em dado momento com a localização prevista que eles mesmos calculam em órbita, também conhecido como erros de efemérides. Estes erros alteram a determinação das distâncias e como consequência a localização do usuário final. Se este tipo de erro acontecer, o usuário final pode receber em seu receptor GPS um desacerto de 15 a 25 metros de sua localização real (RODRIGUES, 1996).

Geometria dos Satélites

Está relacionado com o posicionamento dos satélites e como eles estão configurados em um dado instante em relação ao usuário. Dependendo de como estão posicionados no espaço, podem gerar a diluição de precisão na medição da posição do estado final do receptor.

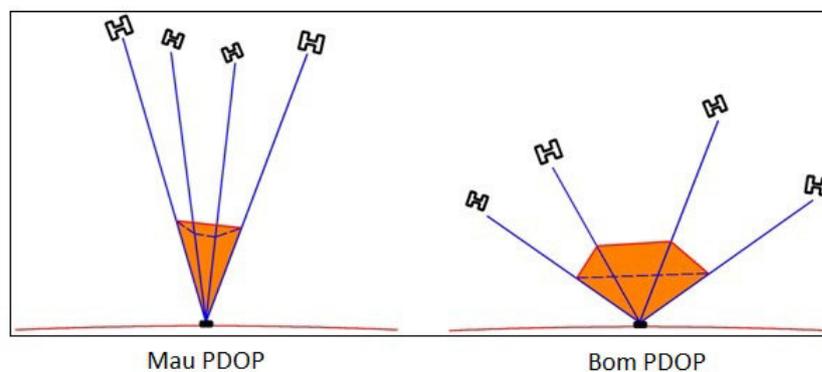
A diluição de precisão (DOP) ou diluição geométrica de precisão (GDOP) está sempre relacionado com o posicionamento tridimensional (PDOP), posicionamento horizontal (HDOP), posicionamento vertical (VDOP), e com a determinação do tempo (TDOP), sendo o mais importante deles o PDOP, usado para medir o desempenho do sistema (RODRIGUES, 1996).

Como dito antes, a geometria dos satélites no espaço é fator determinante para se ter um bom ou mau PDOP e pode ser interpretado como sendo o inverso do volume do tetraedro formado por quatro satélites e o receptor GPS (RODRIGUES, 1996).

$$PDOP = \frac{1}{V} \quad (8)$$

A maioria dos receptores selecionam o melhor PDOP. Será o melhor PDOP o que obtiver o maior volume do tetraedro. O receptor seleciona o melhor PDOP entre os vários satélites visíveis para fazer os cálculos das pseudodistâncias.

Figura 8 – Configurações possíveis de PDOP



Efeito ionosférico

É a camada mais alta da atmosfera, está situada entre 70 e 1000 km acima da superfície terrestre. Possui uma grande densidade de elétrons livres que afetam a propagação das ondas eletromagnéticas. Erros causados pelos efeitos ionosféricos podem variar entre valores menores que um centímetro e a valores maiores que cem metros.

Multipath

O fenômeno acontece quando uma antena pode receber o mesmo sinal por vários caminhos (direções), além do caminho direto até ela. Muitas das perdas de sinal são ocasionadas por efeitos de *multipath*.

2.4 Android

Nas últimas décadas houve um aumento significativo de vendas de celulares, e atualmente, este mercado tende a crescer cada vez mais. Pesquisas apontam que metade da população mundial possui aparelhos celulares ou smartphones. Em números, isso representaria uma quantidade de 3 bilhões de dispositivos móveis em uso no planeta (LECHETA, 2010).

A razão deste próspero mercado tem em vista dois grandes fatores. O primeiro é que cada vez mais os usuários buscam aparelhos com diversos recursos, como: uma moderna e elegante interface visual, bons jogos, GPS, acesso a internet e TV digital. Ao passo que o segundo fator está relacionado ao investimento de grandes empresas, que agora buscam incorporar aplicações moveis a seu dia a dia para agilizar seus negócios e aumentar seus lucros, pois celulares e smartphones podem ocupar um importante espaço em um mundo onde a palavra “mobilidade” está cada vez mais comum (LECHETA, 2010).

2.4.1 O Sistema Operacional

O Android é uma plataforma de desenvolvimento para dispositivos móveis como os Smartphones. Contém um sistema operacional baseado em Linux e utiliza uma máquina virtual customizada arquitetada para otimizar recursos de hardware e memória em um ambiente de dispositivo móvel (LECHETA, 2010).

É um sistema completamente livre e de código aberto (*open-source*), distribuído sob licença Apache 2.0, ou seja, todos podem ter acesso aos códigos-fonte e ao mesmo tempo contribuir para progresso do projeto (SOURCE, 2015).

Foi liderado pela Google desde seu surgimento, acompanhado de muitas outras empresas consolidadas no mercado de telefonia como a Samsung, Motorola, LG, Sony, Ericsson, e muitas outras. E com isso rapidamente se destacou no mercado mobile.

Através da associação destas empresas, o Android é responsável por mover mais de um bilhão de *smartphones* pelo mundo. Apesar deste projeto ser inicialmente desenvolvido para smartphones, ultimamente vem sendo utilizada em outros tipos de dispositivos como, tablets, notebooks e relógios (FERNANDES, 2012). Atualmente o sistema operacional encontra-se na versão 5.0 chamada de Lollipop, além de possuir 975 mil aplicativos publicados no Google Play (ANDROID, 2015).

2.4.2 Open Handset Alliance

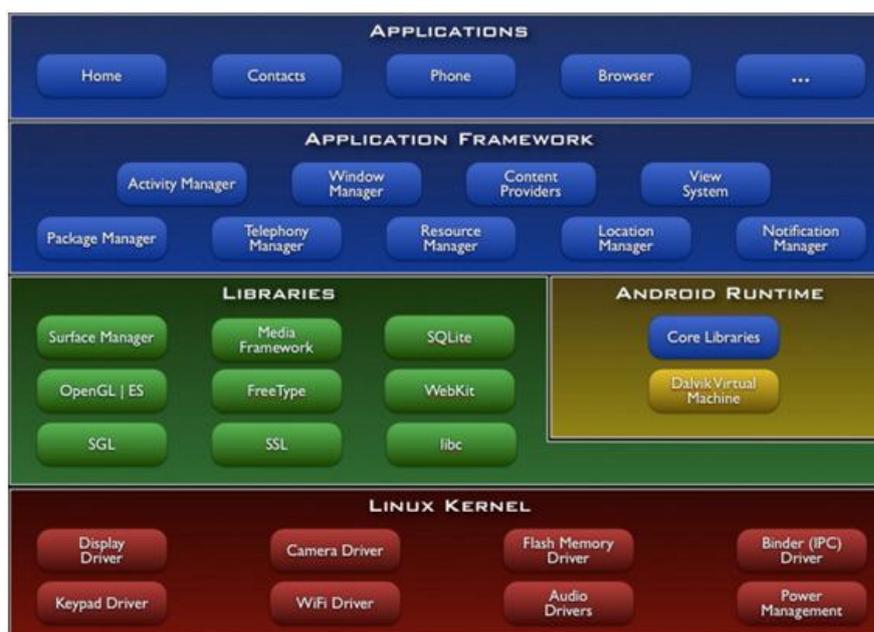
Como já foi dito, a plataforma Android foi formada pela aliança de grandes empresas do mercado de telefonia. A *Open Handset Alliance (OHA)* é um grupo formado por esses gigantes do mercado de celulares e liderado pelo Google.

Teve como principal intenção, padronizar uma plataforma de código aberto e livre para celulares, para atender a todas as expectativas e tendências do mercado atual. Também teve o objetivo de criar uma plataforma moderna e flexível para o desenvolvimento de aplicações corporativas. Como resultado disso, nasceu o sistema operacional Android (LECHETA, 2010).

2.4.3 Arquitetura

É definida em uma arquitetura de 4 camadas, cada uma com sua função bem definida, e todas elas fundamentais para um bom funcionamento de todo o sistema. Roda sobre o *kernel* do Linux, dando suporte para o desenvolvimento de aplicações através de um conjunto de bibliotecas e serviços.

Figura 9 – Arquitetura do sistema operacional Android



Fonte: Bordin (2012)

A descrição de cada camada será apresentada a seguir, começando da mais baixa a até a mais alta em relação ao hardware, como mostra a figura abaixo:

2.4.3.1 Primeira Camada

O *kernel* foi construído baseando-se no Linux 2.6, isso garante que o Android possua características similares a qualquer outro sistema operacional comum que roda em um computador pessoal. A Função do *kernel* é justamente a função de um sistema operacional convencional, ou seja, ele gerencia processos, memória e entrada e saída de dados, além de possuir seu sistema de arquivos, suas configurações de segurança e etc.

O Linux 2.6 foi escolhido por já conter uma grande quantidade de *drivers* de dispositivos sólidos e por gerenciar muito bem a memória e os processos do sistema. É responsável também pela abstração entre o hardware e os aplicativos (FERNANDES, 2012).

2.4.3.2 Segunda Camada

Esta camada é dividida em duas partes, a primeira encontram-se as bibliotecas e a na segunda está o tempo de execução (*Android Run Time*).

a) Bibliotecas e serviços

A camada de biblioteca é um conjunto de instruções que dizem ao dispositivo como lidar com diferentes tipos de dados: áudio, vídeo, gráficos, banco de dados e *browser*. Em sua maior parte são desenvolvidas em C e C++ usadas por diversos componentes do sistema e são expostas a desenvolvedores através da estrutura de aplicativo Android. Além de bibliotecas, aqui também estão os serviços usados em camadas superiores, como máquina virtual *Dalvik*.

b) *Run Time*

A camada de tempo de execução inclui um conjunto de bibliotecas do núcleo Java (*Core Libraries*) e a Máquina Virtual Dalvik (DVM). Para desenvolver aplicações em Android, os programadores devem utilizar a linguagem de programação Java, porém, suas aplicações não serão executadas em uma máquina virtual Java tradicional, e sim na *Dalvik*. Esta permite que cada *thread* rode sua própria instância da MV (máquina virtual), isto garante que nenhuma aplicação seja dependente de outra, e se uma aplicação parar, ela não afetará quaisquer outras aplicações que estão rodando no dispositivo.

2.4.3.3 Terceira Camada

Nesta camada o sistema Android disponibiliza as mesmas APIs usadas no desenvolvimento de aplicações nativas para os desenvolvedores comuns. Todos aqueles que desenvolvem para Android tem a vantagem de ter acesso total a este *framework* como um conjunto de ferramentas básicas, pois a partir dele, o desenvolvedor pode construir aplicações mais robustas que as nativas. Essa camada funciona como um *link* com a camada de bibliotecas do sistema operacional que serão acessadas através de APIs contidas neste *framework*.

2.4.3.4 Quarta Camada

E por última está a camada de aplicação, esta se encontra no topo da pilha de camadas. Ela é composta por um conjunto de aplicações e de funções básicas do sistema. Estas aplicações podem ser nativas ou aplicações desenvolvidas por terceiros. Esta camada tem como principal finalidade prover a interação entre o dispositivo móvel e o usuário.

2.4.4 Dalvik

A máquina virtual *Dalvik* foi construída pelos engenheiros da Google para obter um consumo mínimo de memória e isolamento de processos. Isto é importante por algumas

razões. A primeira é que ela simplifica o gerenciamento de memória, visto que a máquina virtual está baseada em registradores e desenvolvida de forma otimizada para requerer pouca memória. A segunda razão é que ela faz com que nenhuma aplicação seja dependente de outra e se uma aplicação pára, ela não afetará qualquer outra que roda no mesmo dispositivo, ou seja, permite que múltiplas instâncias executem ao mesmo tempo.

Essa máquina virtual foi especialmente criada para rodar aplicações em dispositivos móveis, isto é, em uma CPU lenta, com relativamente pouca RAM, em um sistema operacional sem espaço de troca em disco e em um hardware movido à bateria. É uma máquina de registro, ao contrário das outras JVMs que são máquinas de pilha, o que diminui a quantidade de instruções em cerca de 35% e evita acessos à memória desnecessários (BORSNTEIN, 2008).

As aplicações para o Android são escritas em linguagem Java e utilizam os recursos da *Dalvik*, o código fonte (.java) é compilado em *bytecode* (.class) e então convertido para o formato (.dex), em *Dalvik Executable*, que representa a aplicação do Android compilada. Ao término da compilação, a aplicação, com os arquivos .dex e outros recursos utilizados são compactados em um único arquivo com a extensão .apk (*Android Package File*), que representa a aplicação final pronta para ser distribuída e instalada nos dispositivos.

2.4.5 Ambiente de Desenvolvimento

Não diferente de outras grandes empresas no ramo do desenvolvimento de software, o Android disponibiliza para programadores seu Kit de Desenvolvimento de Software (SDK). Este pacote serve para que programadores externos possam ter uma melhor integração com o software e que consigam desenvolver as suas aplicações de acordo com um padrão de desenvolvimento proposto. Além de dispor também, ferramentas de depuração, emulador, documentação, visualização da utilização de memória e análise de desempenho. O kit de Desenvolvimento de Aplicativos do Android é disponível gratuitamente e pode ser incluído a um ambiente integrado de desenvolvimento (IDE), como por exemplo, o Eclipse.

Como aplicações para a plataforma Android são escritas em Java, a Google recomendava o uso do Eclipse IDE, pois é uma das ferramentas mais sólidas quando se trata de desenvolvimento de aplicativos Java. Para desenvolver usando o Eclipse, o Android oferece um *plugin* personalizado, chamado *textitAndroid Development Tools*. Ele estende os recursos do Eclipse configurando rapidamente o ambiente e deixando-o ideal para o desenvolvimento de projetos Android (DEVELOPER, 2015d). Porém agora, a Google recomenda aos programadores que migrem para o IDE Android Studio (DEVELOPER, 2015e).

O Android Studio é considerado pela Google o IDE oficial para desenvolvimento de aplicativos Android. Apesar de que nas primeiras versões terem evidências de muitos *bugs*, o Android Studio ganha a partir da versão 1.0 sua primeira versão estável, pois com

as correções destes erros, ela se tornou a ferramenta preferencial para os desenvolvedores. O IDE é voltado integralmente para projetos de desenvolvimento Android, diferente do Eclipse. Além de ter possui uma interface bem mais interessante e o *preview* de layouts bem mais robusto, de modo que é possível visualizar simultaneamente vários layouts em diferentes tamanhos tela (DEVELOPER, 2015b), acredita-se que em um futuro bem próximo a maioria dos desenvolvedores usarão essa IDE na construção de seu projetos Android.

Independente do IDE a ser usada, é necessário instalar o JDK (kit de desenvolvimento java). Nele está incluso o *Java Runtime Environment* (Ambiente de Tempo de Execução), o compilador Java e as APIs Java (DEVELOPER, 2015c). Serve principalmente para compilar e executa os aplicativos Java, além fornecer as APIs necessárias para o desenvolvimento de aplicações Java, no qual o Android trabalha.

2.4.6 Activity

Na maioria dos casos representa uma tela da aplicação que e serve como interface para os usuários, onde eles podem interagir com ela realizando algo ou visualizando alguma informação de interesse dos mesmos. Pode-se também dizer que uma *activity* representa uma atividade, ação ou funcionalidade que usuário pode realizar dentro da aplicação (LECHETA, 2010).

Apesar das atividades serem geralmente representadas para os usuários como janelas em telas cheias, elas também pode ser usadas de outras maneiras: Como janelas flutuantes ou incorporadas dentro de outras atividades (DEVELOPER, 2015a).

2.4.7 Gerenciando o Ciclo de Vida de uma Atividade

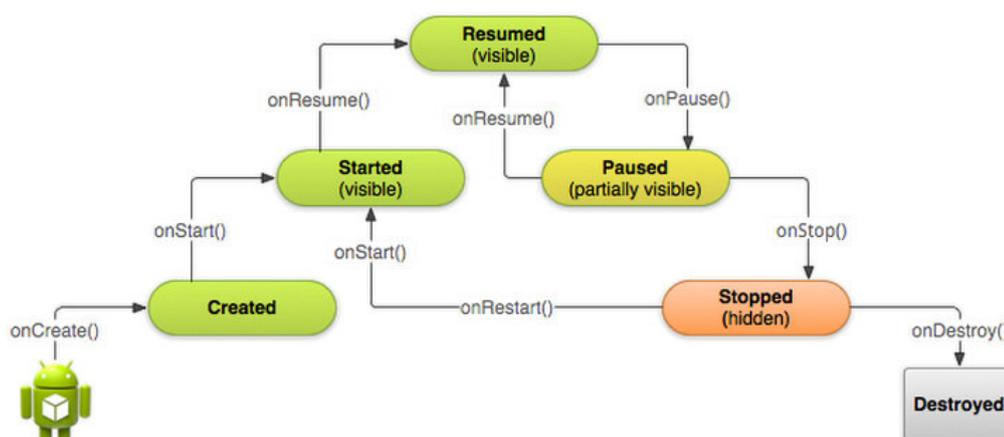
Quando o usuário navega através de um aplicativo, as instâncias das atividades podem assumir diferentes estados em seu ciclo de vida. Ao iniciar a atividade pela primeira vez, ela é direcionada para o primeiro plano recebendo o foco principal, para que o usuário possa interagir com ela. Para que isso aconteça, o sistema Android chama uma série de métodos de ciclo de vida sobre a atividade. Caso o usuário comece uma nova atividade ou um outro aplicativo, o sistema chama um outro conjunto de métodos de ciclo de vida para mover a atividade em foco para o fundo, fazendo esta ficar não visível, porém com a instância e seu estado intacto (DEVELOPER, 2015f).

Diferente de outros paradigmas de programação em que os sistemas são iniciados com um método *main()*, o sistema Android inicia seu código em uma *Activity* (atividade) invocando métodos específicos para essa fase de ciclo de vida (DEVELOPER, 2015f).

Pode-se entender os métodos de ciclo de vida usando uma pirâmide de passos (Figura 10), considerando cada etapa desta pirâmide uma fase do ciclo de vida de uma atividade. Como o sistema cria uma instancia da atividade, cada método move o estado da atividade

em direção ao topo. O topo é ponto ao qual a atividade está sendo executada em primeiro plano. Porém se o usuário começa a deixar a atividade, o sistema chama outros métodos para mover o estado da atividade para baixo da pirâmide com propósito de destruí-la (DEVELOPER, 2015f).

Figura 10 – Ciclo de Vida de uma Activity



Fonte: (DEVELOPER, 2015f)

Apesar de existir diferentes estados, apenas três deles são estáticos, ou seja, as atividades podem existir em apenas um dos três estados em um longo período de tempo. Segundo (DEVELOPER, 2015f) pode-se entender esses estados como está descrito abaixo:

- *Resumed*
Estado em que a atividade está em primeiro plano e o usuário pode interagir com ela.
- *Paused*
Estado em que a atividade está parcialmente escondida por outra atividade. A outra que está em primeiro plano é semitransparente ou não cobre toda a tela. Não pode receber entrada do usuário e nem executar qualquer código.
- *Stopped*
Estado em que a atividade está totalmente oculta, contida no plano de fundo ou no segundo plano. Quando a atividade assume este estado, todas as informações de estado, como variáveis de membro e a instância da atividade são mantidas, porém não pode ser executado qualquer código e muito menos o usuário pode interagir com ela.

Os estados *Created* e *Started*, são estados transitórios e o sistema move-se a partir deles rapidamente e não permite interação alguma com o usuário.

2.4.8 Métodos que podem alterar o estado de uma atividade

Os métodos de chamada que alteram os estados de uma atividade, como viu-se na pirâmide de estados, segundo Lecheta (2010) podem ser resumidamente descritos:

- *onCreate()*
É obrigatoriamente chamado uma única vez. Este método é o ponto de partida de toda aplicação, e somente depois que ele é finalizado, o método *onStart()* é chamada.
- *onStart()*
O método é chamado quando a atividade começa a ser visível pelo usuário e já pode ser vista como “tela” da aplicação.
- *onResume()*
O método é chamado para mover o estado da atividade até o topo da pirâmide de passos. Dessa forma a atividade já está sendo executada como atividade principal e já pode interagir com o usuário.
- *onPause()*
O método é chamado para interromper a atividade que está topo da pilha e salvar o estado da aplicação, para que posteriormente, quando ela voltar a ser executada, tudo possa ser recuperado, se necessário, no método *onResume()*.
- *onStop()*
O método é chamado para encerrar uma atividade que está parcialmente visível ou que já estar totalmente oculta. Se a atividade for reiniciada o método *onRestart()* é chamado. Caso isso nunca ocorra, o sistema operacional chama automaticamente o método *onDestroy()* para remover definitivamente a atividade dessa fila de espera.
- *onRestart()*
O método é chamado quando uma atividade é parada temporariamente e volta a ser inicializada outra vez. Este método é responsável chamar o método *onStart()* de forma automática.
- *onDestroy()*
O método é chamado para encerrar definitivamente uma atividade. Como viu-se ele pode ser chamado pelo sistema operacional para liberar recursos ou pode ser chamado na própria aplicação através do método *finish()*. Com isso o processo que correspondente a esta atividade no sistema operacional é encerrado.

Não há a necessidade de implementar todos os métodos de ciclo de vida, no entanto, é de suma importância entender cada um deles para assegurar que seu aplicativo se comporte de maneira como os usuários esperam, por exemplo:

- Não falhar se o usuário receber um telefonema.
- Não consumir recursos valiosos do sistema quando o usuário não está usando ativamente.
- Não perder o progresso do usuário caso ele deixe o aplicativo e retome a ele posteriormente.
- Não perder o progresso do usuário quando a tela gira entre a orientação horizontal e vertical.

2.5 Google API

É um conjunto de APIs desenvolvidas pela Google a qual permite que aplicações de terceiros possam se comunicar com os seus serviços. Estes serviços são geralmente usados para agregar valor a uma aplicação, porém em outros casos são integralmente essenciais para as mesmas, de modo que sem estas APIs as aplicações não existiriam. A plataforma Android disponibiliza aos seus desenvolvedores, recursos necessários para utilização de todo e qualquer tipo de API já desenvolvida pela Google. A seguir será apresentado as APIs que foram utilizadas para a elaboração do serviço de localização de ônibus.

2.5.1 Google Maps

Serviço o qual permite que usuários inclua mapas e informações de mapeamento personalizado em suas aplicações, possibilitando aos mesmos explorarem o mundo através dos mapas fornecidos pela Google. O usuário pode escolher o tipo de vista do mapa (satélite, terreno e mapas híbridos), personalizá-lo adicionando marcadores sobre o mapa para indicar pontos especiais de interesse e efetuar efeitos de câmera, como por exemplo, rotação, zoom e inclinação que dependendo região, podem apresentar mapas em dimensão 3D (MAPS, 2015).

O *Google Maps* é totalmente gratuito e para ser utilizado em aplicações Android é necessário obter uma chave de autenticação para utilizar o serviço (LECHETA, 2010) e o processo de obtenção de chave é demonstrado na documentação oficial da Google Developes APIs (GEOLOCATION, 2015).

2.5.2 Google Location

Este serviço fornece ao usuário informações necessárias para obter a localização de seu aparelho. Usando esta API, aplicativos podem solicitar a localização conhecida do dispo-

sitivo. Especificamente, é utilizado um provedor de localização fundido para recuperar a localização conhecida do dispositivo. O provedor de localização fundido é uma das APIs de localização da *Google Play Services*. Ele provê uma lista dos provedores de localização disponíveis e registra como receber essas informações sobre a localização, definindo o local através da latitude e da longitude. O provedor de localização fundido também é responsável em otimiza o uso de energia da bateria do dispositivo (LOCATION, 2015).

2.5.3 Google Direction

É um serviço que calcula as direções entre dois locais escolhido (origem e destino) pelo usuário, aplicando quando necessário, o conteúdo da rota calculada em um mapa. Pode-se escolher o modo de tráfego, como por exemplo, a pé, de bicicleta, de carro, dentre outros modos, e se o usuário quiser, pode definir pontos de passagem (*way points*) entre o local de origem e o local de destino. Todos os pontos definidos sejam eles os pontos de origem e destinos ou os *way points*, podem ser especificados como cadeias de textos (endereços), como por exemplo, "Darwin, NT, Austrália" ou como coordenadas geográficas, latitude e longitude (DIRECTION, 2015).

Usa solicitações HTTP para requisitar e retorna o valor de uma consulta. Quando uma aplicação faz um pedido de cálculo de direção, ela pode especificar em que tipo de dado a requisição dever ser respondida, segundo a DIRECTION (2015), essas respostas podem ser retornadas como um arquivo XML ou como um JSON. No conteúdo destes arquivos são incluídas informações sobre o percurso como, todas coordenadas geográficas pertencente a rota calculada, a distância entre os pontos de origem e destino, o tempo de duração do percurso, entre outras.

3 Serviço de Localização de Coletivos

3.1 Desenvolvimento

3.1.1 Metodologia

A Figura 10 mostra as etapas consideradas no desenvolvimento deste trabalho. A priori, foi realizada a elicitación dos requisitos necessários para o desenvolvimento da aplicação. Com esta etapa concluída obteve-se uma documentação para servir como base para a implementação da primeira versão do aplicativo proposto. Nesta primeira etapa foi feita a modelagem do sistema através de três diagramas da UML: o diagrama de casos de uso, e o diagrama de sequência, com objetivo de ter uma documentação descritiva e funcional do sistema. Esta modelagem pode ser vista em detalhes na próxima seção.

Como a aplicação seguiria o clássico modelo de arquitetura cliente-servidor, a fase da implementação foi dividida em duas partes: na primeira houve o desenvolvimento da parte cliente da aplicação, a qual foi desenvolvido o aplicativo que executaria nos dispositivos móveis dos usuários, ao passo que na segunda desenvolveu-se a parte do servidor, a qual seria responsável em gerenciar e prover as informações de localização a respeito dos usuários da aplicação.

Na primeira parte da fase de implementação houve a necessidade de se aprofundar em conceitos mais avançados da linguagem JAVA e estudar também de forma mais detalhada como desenvolver aplicativos para a plataforma Android. Ao conhecer as particularidades do sistema operacional e como construir aplicações para o mesmo, foi necessário estudar também algumas APIs que seriam de extrema importância para elaboração do aplicativo.

Para fazer um aplicativo Android comum processar dados geoespaciais, primeiramente estudou-se como exibir um mapa em um aplicativo, assim como gerar e manipular informações neste mesmo mapa. A API Google Maps foi utilizada para isto e através deste serviço pôde-se desenhar a rota do ônibus, adicionar e personalizar marcadores para o contexto da aplicação, como por exemplo, marcadores para todos os pontos de parada da rota e para os ônibus localizados. Assim os usuários poderão ver a localização aproximada dos ônibus ao longo de suas rotas, bem como outras informações que só farão sentido quando apresentadas no mapa.

Uma vez que os usuários terão que fornecer dados geográficos para alimentar a aplicação, a mesma fará um ambiente colaborativo entre eles a fim de gerar informações sobre a localização dos coletivos. Para isso, foi necessário o uso da API Google Location. Ela foi usada para solicitar a localização dos usuários que estão fazendo uso dos ônibus em um determinado momento. Esta API se comunica com o sistema GPS gerando informações

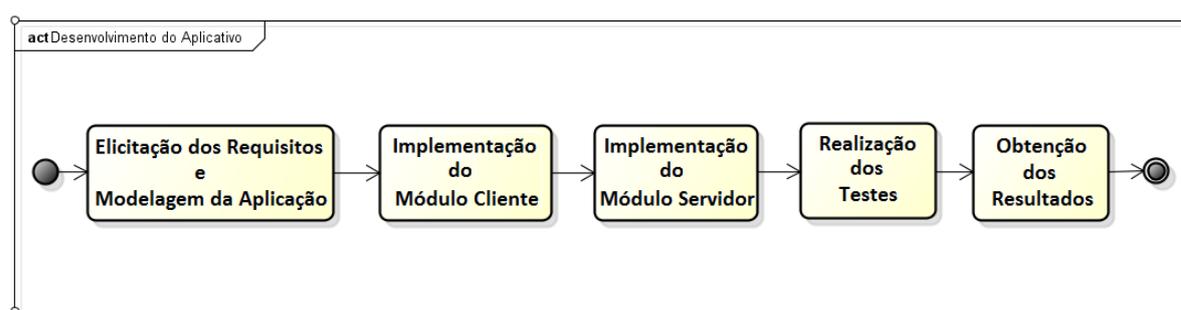
de posicionamento geográfico do dispositivo. Esta API também pode dar informações sobre hora, data e condições climáticas a respeito da região de onde o dispositivo está situado.

Para obter o tempo em que um coletivo percorrerá para se deslocar da sua posição atual até a um ponto de ônibus existente na rota, foi usada a API Google Direction. Através desta API a aplicação envia uma requisição HTTP informando as coordenadas geográficas (ou endereços) da posição atual e da posição final, podendo conter entre o ponto inicial e final, até oito pontos de referência, os quais obrigatoriamente devem fazer parte do percurso. Ela calcula o tempo, a distância, além de outras informações e envia de volta a resposta para aplicação que a solicitou.

Com a primeira versão da aplicação cliente finalizada, começou-se fazer a segunda parte da fase de implementação. Nela houve a necessidade de se aprofundar em conceitos da linguagem PHP para implementar o servidor da aplicação. Viu-se necessidade de conectar aplicação servidora a um banco de dados, pois além de ter que processar muitas informações, uma quantidade significativa teria que ser armazenada para posteriormente ser repassada, pois seria inviável manter essa grande quantidade de informação somente em memória.

Depois da aplicação construída foi feitas uma bateria de testes. Com os resultados destes testes foram obtidas algumas informações sobre o desempenho do aplicativo. Essas informações serão apresentadas e discutidas na próxima seção.

Figura 11 – Etapas do processo de construção do aplicativo representado por um diagrama de atividade

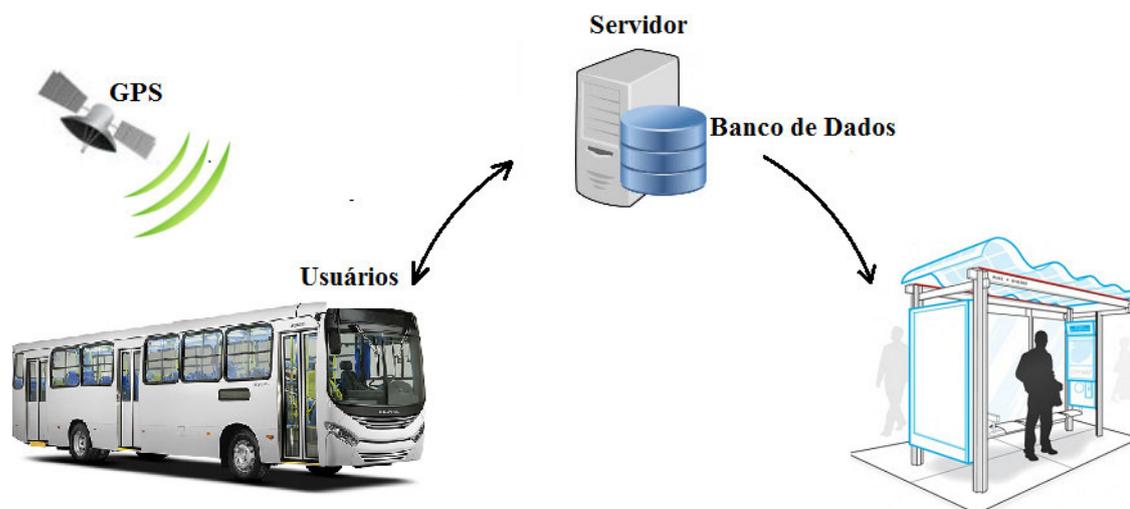


3.1.2 Arquitetura da Aplicação

A arquitetura da aplicação segue o modelo cliente-servidor. Esta estrutura de aplicação distribuída partilha tarefas entre o servidor (o fornecedor de recursos) e os clientes (requerentes dos serviços). Na Figura 12 é apresentada arquitetura desta aplicação.

GPS: É o sistema encarregado de fornecer as coordenadas geográficas para os dispositivos móveis dos usuários. Este sistema se comunica com o receptor GPS presente no dispositivo o qual é responsável em receber o sinal emitido pelos satélites e traduzir este sinal em

Figura 12 – Arquitetura da Aplicação



dados geoespaciais (latitude e longitude) que possam ser úteis para a aplicação. Apesar de o GPS ser um sistema muito eficiente, algumas vezes ele pode emitir dados não tão exatos. Estes erros podem variar em até 25 metros de imprecisão do local real.

Usuário: São todas as pessoas que possuem esta aplicação instalada em seus dispositivos móveis. Através dela o usuário pode fornecer sua localização ou receber a localização de outros usuários que interagem com mesmo sistema. O aplicativo requisita o posicionamento atual através da API Location que se comunica com receptor GPS existente no dispositivo. De tempos em tempos (alguns segundos) a aplicação faz novas requisições para o sistema GPS com o objetivo de receber todos os lugares que o dispositivo (usuário) percorreu no decorrer de um percurso feito por ele e assim continuamente alimentar a aplicação com estas informações, exibido-as em tempo real no mapa do aplicativo por meio da API Google Maps.

Como ser visto na Figura 12 há dois tipos de usuários. O usuário que somente recebe informações do servidor a respeito de outros usuários e o usuário que fornece a sua localização e ao mesmo tempo recebe localização dos outros.

Servidor: É a parte da aplicação que recebe as coordenadas dos usuários, corrige-as quando necessário e depois as repassa para os demais. Ao longo de um percurso feito por um usuário em um coletivo, a aplicação gera em um intervalo de tempo pré-definido várias coordenadas e no mesmo ritmo as envia via GET para o servidor, o mesmo recebe as informações dos usuários e faz a correção do ponto (latitude/longitude) enviado pelo usuário utilizando a API GeoPHP. A aplicação servidor procura o ponto mais próximo existente na rota do ônibus em relação ao ponto gerado pelo usuário. Logo após, o servidor guarda tanto o ponto recebido quanto o ponto corrigido em um banco de dados remoto. Em seguida o servidor envia de volta todos os ônibus localizados para aplicação cliente, ou seja, a localidade de todos os usuários que estão colaborando para localizar os coletivos.

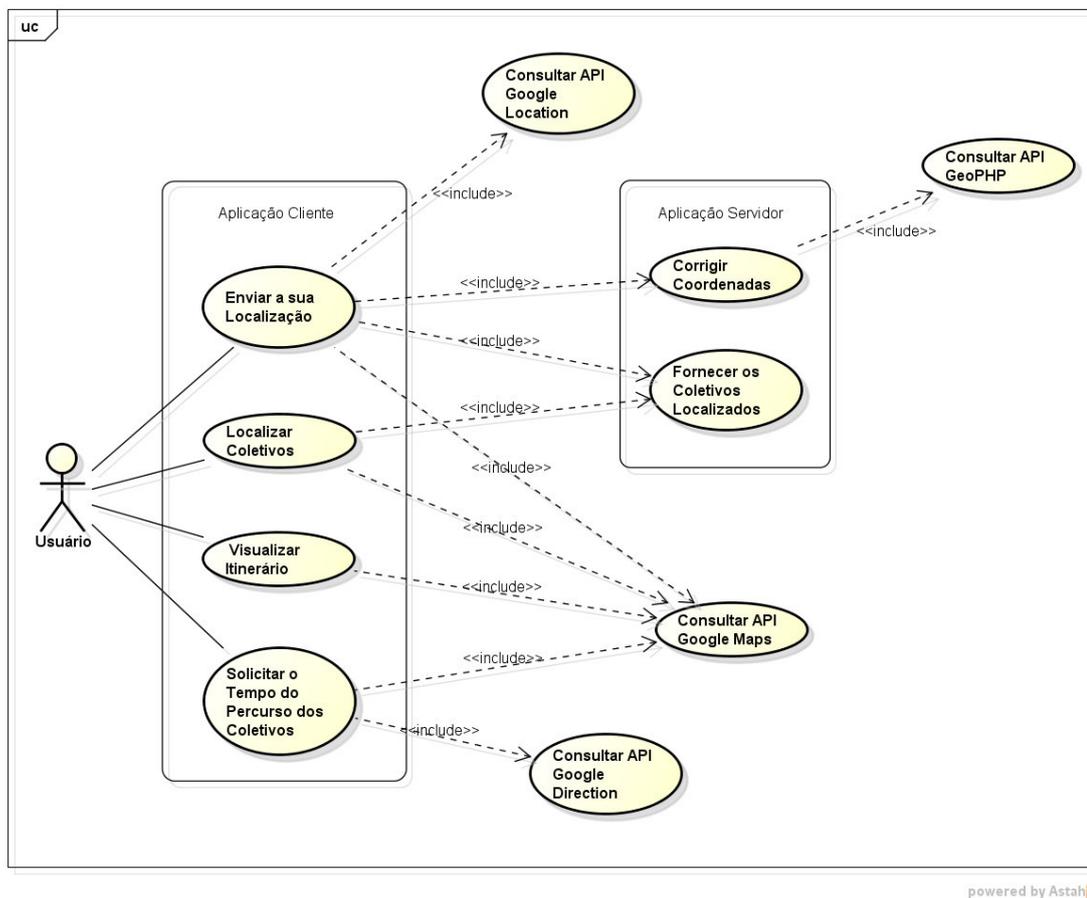
3.1.3 Modelagem

Para melhor entendimento da aplicação, foi realizada a modelagem do sistema utilizando o software Astah. Durante esta etapa foram desenvolvidos os diagramas de caso de uso e o diagrama de sequência do aplicativo. Cada um deles será apresentado a seguir.

Diagrama de Caso de Uso

Este diagrama tem como objetivo descrever as principais funcionalidades da aplicação e a interação dessas funcionalidades com os usuários da mesma. A Figura 13 apresenta o diagrama de caso de uso.

Figura 13 – As principais funcionalidades da Aplicação



powered by Astah

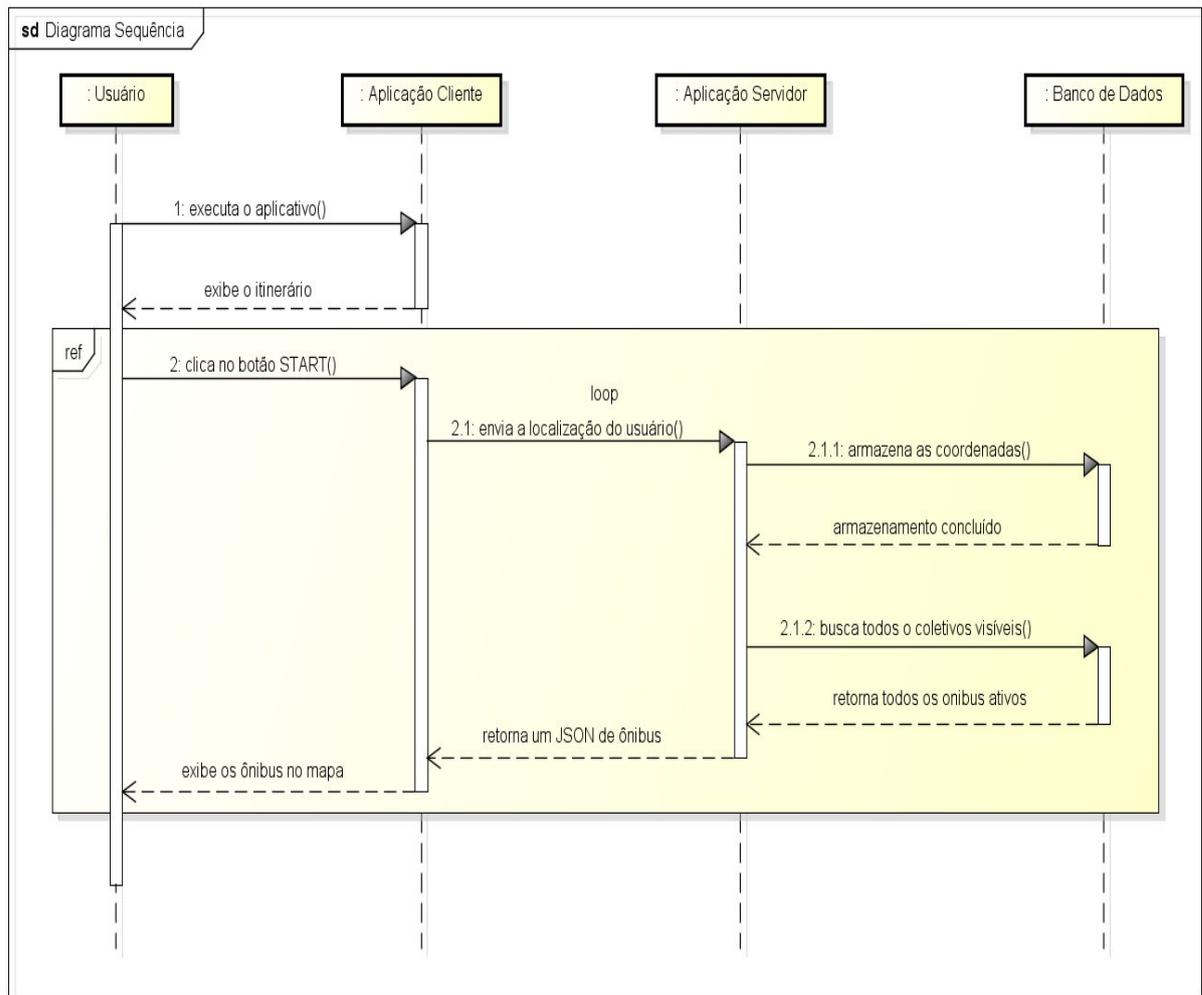
Diagrama de Sequência

Este diagrama tem como objetivo mostrar como as interações entre os atores (usuário, aplicativo e banco de dados) acontecem e como as mensagens de solicitação e repostas ocorrem no decorrer do tempo para a realização de uma operação existente na aplicação.

É mostrado a sequência de interações entres os atores do sistema quando usuário estiver fornecendo a sua localização para aplicação. Esta operação foi escolhida pois dentre as

outras é a funcionalidade que melhor representa o comportamento do sistema após uma ação do usuário no mesmo. A Figura 14 mostra o diagrama de sequência.

Figura 14 – Interação do usuário com a aplicação



powered by Astah

3.1.4 Ferramentas Utilizadas

Para o desenvolvimento desta aplicação foram utilizadas as seguintes ferramentas:

- A linguagem Java no desenvolvimento da aplicação para plataforma Android.
- A linguagem PHP para implementação do Servidor.
- O bando de dados MySQL para armazenar e disponibilizar informações sobre a localização dos usuários.
- As APIs Google Maps, Google Location, Google Direction e GeoPHP.
- Os Softwares Astah, Android Studio, Netbeans e MySQL WorkBench.

3.2 Aplicativo

3.2.1 Principais Funcionalidade

- Exibir no mapa o itinerário de uma linha de ônibus.
- Exibir no mapa todos os pontos de parada da linha de um coletivo.
- Exibir no mapa todos os ônibus que estão sendo localizados baseado na localização de usuários que estão fazendo uso dos coletivos em um determinado instante.
- Receber coordenadas geográficas dos usuários.
- Fornecer o tempo estimado dos ônibus presentes na rota até a um determinado ponto de parada.
- Corrigir possíveis erros de localização gerados pelo Sistema de Posicionamento Global (GPS).

3.2.2 Funcionamento

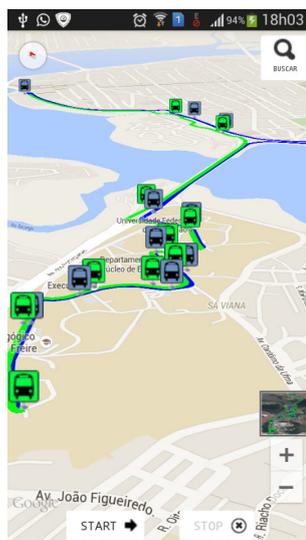
O protótipo da aplicação foi desenvolvido para funcionar em *smartphones* que possuem o sistema operacional Android. O protótipo possui apenas uma tela, onde será exibido um mapa onde conterà o itinerário juntamente com todos os pontos de parada da mesma linha. Todas essas informações são geradas localmente no dispositivo, isto é, sem a necessidade de conexão com a internet. Somente será necessário o uso da internet, caso o usuário queira saber onde algum coletivo está localizado ou caso ele mesmo queira colaborar com a sua localidade para fornecer a posição de um coletivo.

A tentativa foi deixar um layout bem simplificado (Figura 15) onde fosse fácil do usuário conseguir desfrutar de todas funcionalidades da aplicação de forma bem prática e assim ser capaz de receber informações do aplicativo ou enviar informações para ele com praticidade.

Quando o usuário executar a aplicação em seu dispositivo, será apresentado um mapa onde terá algumas informações inclusas nele (Figura 15). Este mapa está direcionado para a região onde a linha do ônibus utilizada como estudo de caso está geograficamente situada. O usuário poderá navegar sobre toda a extensão do mapa, assim como modificar a posição, inclinação ou o zoom em que o mapa é apresentado com apenas alguns simples toques na tela do dispositivo.

Como já dito antes, o mapa apresenta a rota nos dois sentidos em que um coletivo pode percorrer ao executar a sua linha. Para isso, foi usado duas cores para diferenciar quando um coletivo está se deslocando a um determinado sentido e uma outra cor quando estiver se deslocando no sentido contrário do primeiro. As linhas coloridas sobrepostas nas ruas do mapa são para facilitar a leitura quanto ao sentido em que um coletivo visível

Figura 15 – Tela Principal



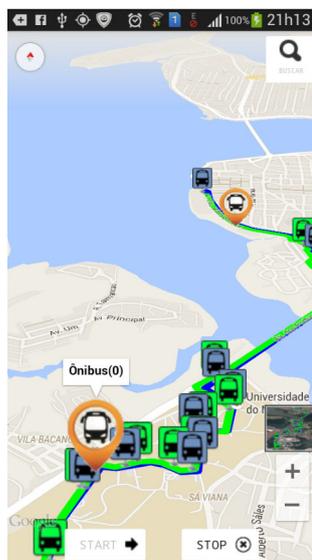
se desloca ao longo de seu itinerário. Seguindo o mesmo princípio do sentido das rotas, os pontos de parada dos ônibus também são exibidos em duas cores distintas e cada um deles possuirá a mesma cor do sentido em que ele pertence no itinerário.

Além das informações já inclusas no mapa, a tela principal apresenta alguns botões para receber comandos do usuário. Um deles é o botão “START”, o qual é responsável em acionar a aplicação para entrar em comunicação com o sistema GPS, para o mesmo fornecer para a aplicação as coordenadas geográficas ao longo de um trajeto. Cada vez que o aplicativo recebe um dado geográfico provido do GPS, ele imediatamente envia-os para o servidor da aplicação por meio da internet. Todos estes passos só serão executados se o usuário habilitou o provedor de localização do aparelho (GPS) e a rede Wi-Fi ou dados móveis do dispositivo, enquanto não houver feito isso, ao apertar o botão “START”, o aplicativo não fará nada, apenas exibirá uma mensagem de alerta pedindo para o usuário iniciar o recurso ainda não habilitado.

Quando o usuário pressionar o “START” ele deve estar ciente que a partir de agora sua localização estará visível para todos os usuários interessados em saber onde um ônibus se encontra. Na tela de todos os dispositivos que estiverem executando o aplicativo, aparecerá um marcador personalizado com uma imagem de um ônibus (Figura 16) informando a localização deste. Como ele está fornecendo sua localização para informar onde o ônibus se encontra, todos os usuários, inclusive ele mesmo, verão onde exatamente um coletivo está em dado instante na rota. Este botão só deverá ser pressionando quando o usuário estiver dentro de um coletivo. Ao apertá-lo ele fica desabilitado até o usuário apertar o botão “STOP”.

O botão “STOP” é responsável em parar a execução de passos que o botão “START” começou. Quando o usuário pressioná-lo, ele estará informando que não está mais dentro das dependências de um coletivo, ou seja, é momento ao qual ele informa a aplicação que

Figura 16 – Usuário fornece sua localização



desceu do ônibus, então o aplicativo encerra sua comunicação com o GPS e consequentemente para de enviar as coordenadas desse usuário para o servidor, pois ele não é mais uma referência para localizar um coletivo. O “STOP” só fica habilitado logo após o botão “START” ser pressionado, enquanto o “START” não é pressionado “STOP” permanece desabilitado para o usuário.

Como pode ser visto estes dois botões são os responsáveis em manter a aplicação informada sobre quando deve-se pegar as coordenadas geográficas de um usuário e quando não se deve pegá-las. Eles representam uma viagem feita por um usuário em um coletivo, onde este ciclo é iniciado a partir do momento em que o usuário entra em um coletivo até o momento que ele sai do mesmo. Somente através destes dois botões é possível rastrear um ônibus e fornecer a localização do mesmo para todos os usuários envolvidos no mesmo sistema.

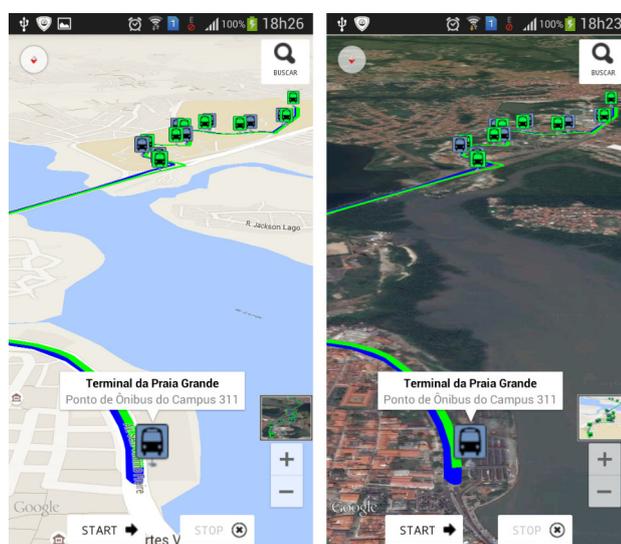
O botão “Buscar” representado por uma lupa localizado no canto superior esquerdo da tela (Figura 16) é o botão responsável em requisitar todos os ônibus localizados pela aplicação. No momento em que o usuário aperta este botão, todos os ônibus que estão sendo rastreados pelo servidor da aplicação serão exibidos no mapa do aplicativo. De maneira quase similar quando é pressionado o “START”, a aplicação também envia uma requisição ao servidor solicitando todos os ônibus localizados através de uma chamada GET, mas com uma pequena diferença, nesta funcionalidade não é enviado nenhum dado geoespacial para o servidor, isto é, o aplicativo apenas pede para que o servidor envie todos os ônibus que estão sendo rastreados baseado na localização de terceiros.

Para acionar esta utilidade também é necessário habilitar o provedor de localização do aparelho (GPS) e o Wi-Fi ou os dados móveis do dispositivo. Depois de acionar este botão o mapa fica atualizando em um curto intervalo de tempo (poucos segundos) a posição dos

marcadores que estão representando os ônibus no mapa.

Existem mais dois botões na tela principal da aplicação, o botão que muda o tipo de apresentação do mapa e o botão para acionar o zoom no mapa. O botão “Mudar o Tipo do Mapa” representado por uma imagem de um mapa de tamanho reduzido situado em cima do botão zoom, muda o tipo de representação do mapa ou para o tipo satélite, onde o mapa é apresentado através de fotos tiradas de satélites que dão uma visão mais realista do ambiente, ou para o tipo normal, onde o mapa é apresentado com características artificiais do ambiente real, ou seja, é representado como uma ilustração (Figura 17).

Figura 17 – Os tipos de mapa da aplicação



O botão de zoom é uma alternativa para o usuário, pois através de simples toques na tela o aplicativo pode se aproximar ou se distanciar de um ponto de interesse do usuário sem a necessidade dele pressionar este botão, mas como a maioria das aplicações que usam mapas encontradas no mercado apresentavam esta função mesmo conhecedoras que na própria tela do dispositivo possui tela sensível ao toque (*touchscreen*), este botão também foi adicionado nesta aplicação para seguir tal padrão de mercado.

3.2.3 Tempo de Previsão de Chegada dos Coletivos

Fornecer para os usuários uma previsão de tempo de chegada de um coletivo até um determinado ponto de parada da rota (Figura 18). Para o usuário requisitar este tempo, bastará ele clicar em algum marcador de ponto de parada existente no mesmo sentido em que um ou mais ônibus estão trafegando em um exato momento dos quais os coletivos ainda não passaram.

Quando o usuário clicar em um ponto de parada, a aplicação envia uma solicitação HTTP para o servidor da API Google Direction e então ela calculará a direção entre o local de origem (local onde ônibus se encontra no mapa) e o local de destino (local de um

Figura 18 – Cálculo da previsão de chegada de um ônibus até a um ponto de parada



ponto de parada da rota também presente no mapa) devolvendo a resposta à aplicação. Nesta resposta há várias informações sobre o cálculo de uma direção, inclusive o tempo estimado que o usuário levará para transitar do ponto de origem até ao ponto de destino. O modo de viagem também deverá ser incluso na URL que faz a requisição HTTP para a API. Para esta aplicação foi usando o modo de viagem padrão (*driving*), o qual informa a Google Direction que o tempo a ser calculado em relação a uma direção deverá ser baseado em um trajeto feito por um veículo automotivo.

Serão somente calculados os tempos entre o local do ônibus até os pontos de parada, os pontos cujo o ônibus ainda não passou ao longo do seu percurso. Caso ele já tenha passado por um ponto de parada e o usuário clicar nele, será apresentado uma mensagem de alerta informando que não há nenhum ônibus na rota.

Também não será disponibilizado uma estimativa de tempo para os pontos de parada que não fazem parte do mesmo sentido ao qual o coletivo se desloca, como por exemplo, se o usuário clica em ponto de parada azul (sentido UFMA/Terminal da Praia Grande), porém um ou mais ônibus estão percorrendo a rota sobre a linha verde (sentido Terminal da Praia Grande/UFMA) nenhuma informação sobre os ônibus que trafegam sobre a linha verde será apresentada para o usuário neste ponto de parada, mas sim uma mensagem de alerta informando que não há nenhum ônibus trafegado neste sentido.

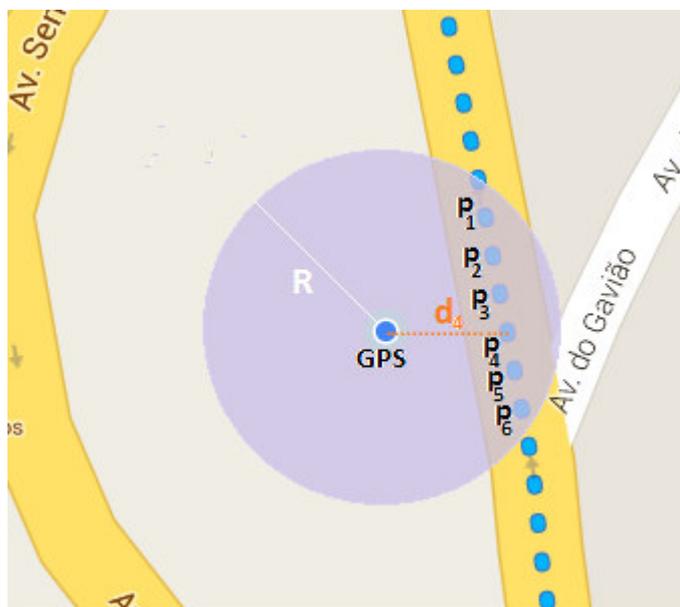
3.2.4 Correção do Posicionamento Gerado pelo GPS

Para cada coordenada geográfica gerada pelo receptor GPS em um dado instante, terá uma outra coordenada relacionada a esta primeira, e para cada relação, será retornado um ponto (p_i) medindo uma distância (d_i) entre as duas coordenadas envolvidas. A segunda coordenada, a gerada pelo servidor da aplicação, é a coordenada geográfica corrigida caso

a primeira varie muito da localização real do usuário. Estas coordenadas geralmente são corrigidas devido a erros de precisão que o sistema GPS pode emitir (vide na subseção 2.3.3 da Fundamentação Teórica).

Utilizando a Figura 19 para ilustrar como funciona a rotina de correção das coordenadas imprecisas, vê-se que quando receptor GPS emite a coordenada geográfica da posição de onde usuário está, ele pode cometer erros de precisão. Como o usuário estará dentro de um coletivo, espera-se que a localização deste coletivo seja exatamente sobre a via, porém o receptor GPS acaba emitindo uma localização distante da via (ponto azul nomeado de GPS). E para mascarar esta possível imprecisão de localização do usuário, é necessário que haja uma correção e assim apresentar o coletivo exatamente sobre via no mapa do aplicativo.

Figura 19 – Correção da coordenada gerada fora da via



Depois que a coordenada do receptor GPS do aparelho é emitida, faz-se uma busca por pontos pertencentes ao itinerário em um raio (R) de 45 metros em relação a coordenada emitida pelo receptor GPS. Todos aqueles que fizerem intercessão com a área do círculo relacionada à coordenada gerada pelo GPS, (p_1, p_2, \dots, p_6) serão coletados e medido as distâncias (d_1, d_2, \dots, d_6) que cada um deles tem em relação a coordenada do GPS. O que obter a menor distância (d_4) será o ponto retornado para o usuário (p_4), ou seja, o ponto corrigido pelo servidor da aplicação (Figura 19).

4 Experimentos e Análise dos Resultados

A etapa de experimentos e análises dos resultados foi a responsável em verificar o comportamento e medir o desempenho do aplicativo proposto. A mesma foi iniciada imediatamente após o término da etapa de desenvolvimento. Nela, foi utilizado o estudo de caso da linha do ônibus campus integração (número 311) para coletar as informações necessárias e logo após fazer as avaliações sobre a aplicação. Esta linha foi escolhida por ser relativamente pequena quando comparada com outras linhas, e por fazer parte do cotidiano dos universitários e conseqüentemente bem conhecida por todos.

4.1 Método para a Avaliação

Foram definidos dois métodos avaliativos para fazer a análise do aplicativo. O primeiro foi observar o comportamento do aplicativo em relação à precisão do sinal recebido pelo receptor GPS do dispositivo, enquanto o segundo foi comparar o tempo de previsão feito pelo aplicativo em relação ao tempo real em que um coletivo gasta para percorrer um determinado trajeto. Como já foi dito, todos os percursos utilizados foram feitos usando a linha de ônibus do campus integração.

Houve também a necessidade de implementar uma rotina que coletasse várias coordenadas geográficas (latitude e longitude) no decorrer de um percurso. Estas coordenadas eram geradas pelo receptor GPS do aparelho em um pequeno intervalo de tempo repetidas vezes, e a cada vez que geradas, armazenadas ordenadamente em um arquivo de saída, onde os mesmos seriam usados futuramente como simulações para efetuar os testes.

Em outras palavras, foram feitas simulações do que aconteceria na realidade, ao usuário entrar no ônibus o mesmo iria habilitar o aplicativo para gerar sua posição de tempos em tempos até que o mesmo chegasse ao lugar desejado e assim encerrasse a interação com a aplicação, ou seja, cada arquivo seria equivalentemente a uma viagem feita por um usuário em um ônibus.

Estas simulações foram nomeadas de percursos, os quais são eles: O percurso feito do Terminal da Praia Grande até o Centro Pedagógico Paulo Freire foi nomeado de Terminal/Paulo Feire; O percurso feito do Terminal da Praia Grande até o Centro de Ciências Exatas e Tecnológicas (CCET) foi nomeado de Terminal/CCET; O percurso feito da Biblioteca Central até o terminal Praia Grande foi nomeado de Biblioteca/Terminal; E o percurso feito do Terminal da Praia Grande até o ponto de parada que fica em frente ao CEPRAMA foi nomeado de Terminal/CEPRAMA.

Resumidamente todos os percursos podem se enquadrar em duas categorias, os que fazem o percurso sentido Terminal até a UFMA (linha de cor azul do itinerário – Figura 18) e os que fazem o percurso sentido UFMA até o Terminal (linha de cor verde do itinerário – Figura 18). Para simplificar será chamado apenas pela sigla TU para primeira categoria e pela sigla UT para segunda.

Todos os percursos foram utilizados nas duas baterias de testes, alguns para o primeiro método e outros para o segundo método. Os percursos Terminal/Paulo Feire e Terminal/CCET foram usados para verificar a margem de erro da previsão de tempo de chegada de um coletivo até a um ponto de parada qualquer, fazendo uma comparação entre a previsão de tempo gerado pelo aplicativo em relação ao tempo real em que este coletivo demorou para chegar até a um determinado ponto de parada.

Já os percursos Biblioteca/Terminal, Terminal/CEPRAMA e o Terminal/CCET, sendo este último o único usado nas duas baterias de testes, foram utilizados para medir as margens de erro quando o aplicativo gera dados geoespaciais sobre a posição de um usuário, comparando a distância entre a coordenada geográfica gerada pelo receptor GPS e a coordenada geográfica corrigida gerada pelo servidor da aplicação em dado momento.

4.2 Experimentos

De acordo com o que foi descrito na Seção 4.1, os testes foram divididos em duas etapas levando em consideração os dois tipos de métodos usados para avaliar o aplicativo. Os primeiros experimentos a serem apresentados serão os experimentos relacionados a taxa de variação da distância entre as coordenadas geográficas geradas pelo receptor GPS em relação as coordenadas geográficas corrigidas geradas pelo servidor da aplicação. E na sessão subsequente será apresentada os experimentos feitos em relação a previsão de tempo de chegada de um coletivo até um ponto de parada fornecida pelo aplicativo.

4.2.1 Precisão da Localização dos Ônibus Utilizando a Posição GPS

Nas tabelas abaixo serão representados alguns pontos selecionados aleatoriamente de cada percurso. Por exemplo, na Tabela 4.2.1, a coluna nomeada de coordenada GPS tem os valores de latitude e longitude associada às coordenadas coletadas para este percurso (simulação). A coluna nomeada de coordenada corrigida possui os valores de latitude e longitude das coordenadas corrigidas no decorrer deste experimento, conforme viu-se na seção 3.2.4 Correção do Posicionamento Gerado pelo GPS.

A primeira coluna enumera todas as relações (r_1, r_2, \dots, r_n) que foram feitas entre as duas coordenadas envolvidas. E por último, a distância em metros entre as duas

coordenadas (quarta coluna). Estas distâncias (d_i) são justamente as variações que as coordenadas corrigidas possuem em relação as coordenadas emitidas pelo receptor GPS.

Para deixar mais claro, foram gerados gráficos para cada simulação. Neles são mostrados todos os pontos do percurso em relação à distância que os pontos desse percurso assumiram. Todos os valores são similares aos valores das tabelas, mas dessa vez serão apresentados todos os pontos de cada simulação.

Percurso Biblioteca/Terminal

Os dados usados para este percurso foram coletados no dia 09/06/2015 às 16:59. Todos os pontos foram coletados entre a Biblioteca Central situada na Universidade Federal do Maranhão até o Terminal da Praia Grande situado na avenida Senador Vitorino Freire, isto é, este é um percurso da categoria UT.

Para este percurso foram coletados 170 pontos, mas serão mostrados apenas 16 pontos para exemplificar o problema encontrado. Estes 16 pontos podem ser visto na Tabela 4.2.1.

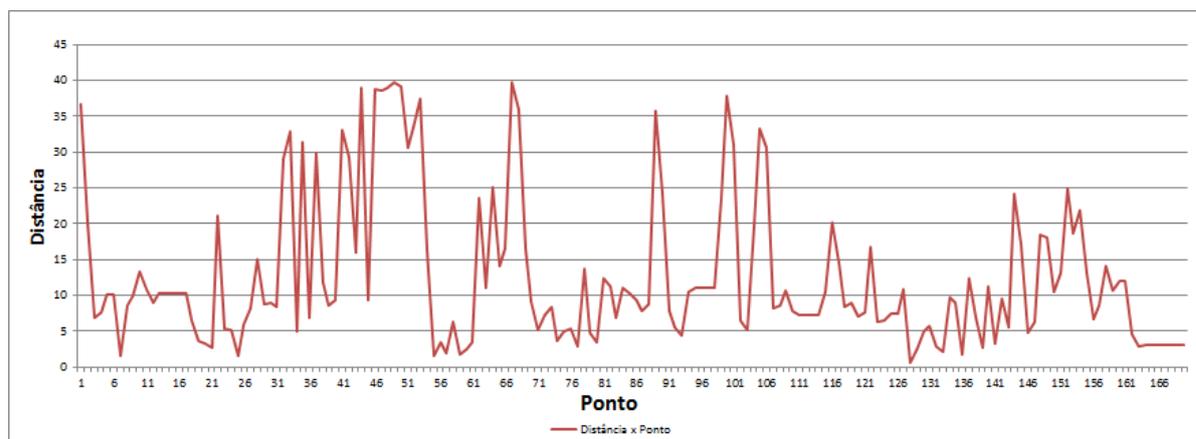
Tabela 3 – Informações sobre o experimento de precisão GPS (Biblioteca/Terminal)

Relação (r_i)	Coordenada do GPS	Coordenada Corrigida	Distâncias (d_i)
r_1	latitude:-2.5549/longitude:-44.3070	latitude:-2.5549/longitude:-44.3065	36,74
r_{13}	latitude:-2.5531/longitude:-44.3059	latitude:-2.5532/longitude:-44.3060	10,33
r_{26}	latitude:-2.5519/longitude:-44.3071	latitude:-2.5519/longitude:-44.3072	5,98
r_{39}	latitude:-2.5488/longitude:-44.3048	latitude:-2.5488/longitude:-44.3047	8,62
r_{54}	latitude:-2.5441/longitude:-44.3007	latitude:-2.5440/longitude:-44.3006	16,38
r_{67}	latitude:-2.5447/longitude:-44.2976	latitude:-2.5446/longitude:-44.2980	39,66
r_{78}	latitude:-2.5447/longitude:-44.2966	latitude:-2.5449/longitude:-44.2966	13,74
r_{82}	latitude:-2.5444/longitude:-44.2977	latitude:-2.5443/longitude:-44.2976	11,3
r_{95}	latitude:-2.5414/longitude:-44.3006	latitude:-2.5415/longitude:-44.3006	11,08
r_{107}	latitude:-2.5393/longitude:-44.3009	latitude:-2.5394/longitude:-44.3009	8,28
r_{112}	latitude:-2.5388/longitude:-44.3010	latitude:-2.5389/longitude:-44.3011	7,16
r_{125}	latitude:-2.5373/longitude:-44.3016	latitude:-2.5374/longitude:-44.3016	7,5
r_{139}	latitude:-2.5362/longitude:-44.3036	latitude:-2.5362/longitude:-44.3035	2,72
r_{152}	latitude:-2.5337/longitude:-44.3062	latitude:-2.5339/longitude:-44.3061	24,98
r_{161}	latitude:-2.5310/longitude:-44.3070	latitude:-2.5310/longitude:-44.3069	11,96
r_{170}	latitude:-2.5308/longitude:-44.3069	latitude:-2.5308/longitude:-44.3069	3,02

Com os dados coletados para este percurso, foi feito um gráfico (Figura 20). Nele está representado todas as distâncias calculadas para cada ponto da simulação em questão, ou seja, os 170 pontos envolvidos neste experimento.

O eixo x do gráfico mostra todos os pontos do percurso e o eixo y do gráfico mostra a distância em metros entre a cada ponto gerado pelo receptor GPS neste experimento em relação ao ponto real pertencente a rota do ônibus. Essa relação está representada pela linha vermelha apresentada no gráfico da Figura 20.

Figura 20 – Precisão da localização GPS (Biblioteca/Terminal)



Percurso Terminal/CEPRAMA

O percurso Terminal/CEPRAMA foi feito no dia 16/06/2015 às 12:39. Todos os pontos foram coletados entre o Terminal da Praia Grande situado na avenida Senador Vitorino Freire até o ponto de parada que fica em frente ao Centro de Produção Artesanal do Maranhão (CEPRAMA) situado no bairro Madre Deus. Este ponto de parada fica do lado oposto ao CEPRAMA. É um percurso pertencente a categoria TU, e o mesmo possui 55 pontos, mas para exemplificar, serão mostrados apenas 16 pontos na Tabela 4.

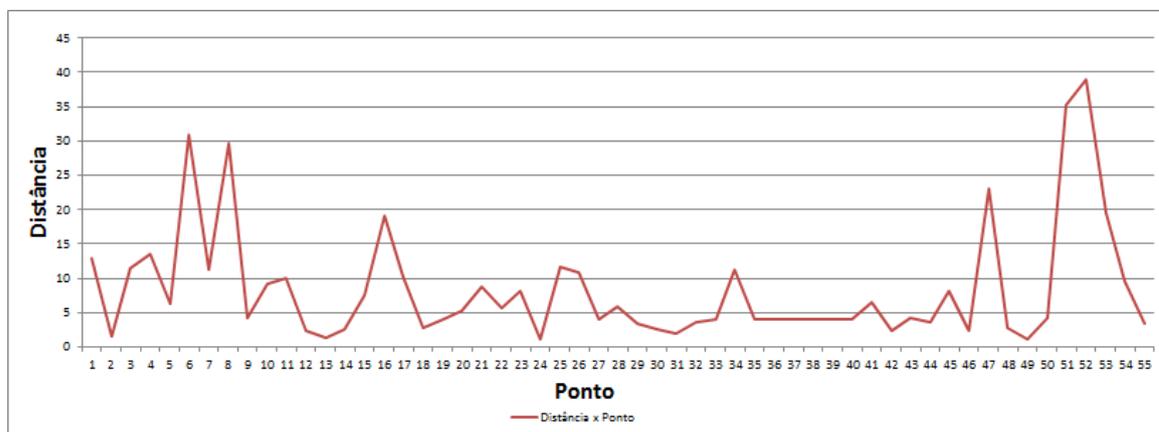
Tabela 4 – Informações o sobre experimento de precisão GPS (Terminal/CEPRAMA)

Relação	Coordenada do GPS	Coordenada Corrigida	Distâncias
r_1	latidude:-2.5326/longitude:-44.3069	latidude:-2.5326/longitude:-44.3068	13
r_6	latidude:-2.5336/longitude:-44.3064	latidude:-2.5334/longitude:-44.3065	30,86
r_{11}	latidude:-2.5352/longitude:-44.3056	latidude:-2.5352/longitude:-44.3055	10,02
r_{15}	latidude:-2.5361/longitude:-44.3044	latidude:-2.5361/longitude:-44.3044	7,59
r_{19}	latidude:-2.5365/longitude:-44.3033	latidude:-2.5364/longitude:-44.3033	4,08
r_{22}	latidude:-2.5367/longitude:-44.3028	latidude:-2.5366/longitude:-44.3028	5,68
r_{27}	latidude:-2.5369/longitude:-44.3024	latidude:-2.5369/longitude:-44.3023	3,97
r_{33}	latidude:-2.5376/longitude:-44.3020	latidude:-2.5375/longitude:-44.3020	4,07
r_{36}	latidude:-2.5379/longitude:-44.3016	latidude:-2.5379/longitude:-44.3016	4,04
r_{39}	latidude:-2.5379/longitude:-44.3016	latidude:-2.5379/longitude:-44.3016	4,04
r_{42}	latidude:-2.5380/longitude:-44.3015	latidude:-2.5380/longitude:-44.3016	2,35
r_{45}	latidude:-2.5394/longitude:-44.3012	latidude:-2.5394/longitude:-44.3013	8,19
r_{47}	latidude:-2.5402/longitude:-44.3014	latidude:-2.5404/longitude:-44.3014	23,13
r_{51}	latidude:-2.5412/longitude:-44.3009	latidude:-2.5410/longitude:-44.3010	35,17
r_{53}	latidude:-2.5418/longitude:-44.3008	latidude:-2.5419/longitude:-44.3008	19,45
r_{55}	latidude:-2.5423/longitude:-44.3006	latidude:-2.5424/longitude:-44.3006	3,33

O gráfico deste percurso apresentará os valores das distâncias de cada ponto coletado neste experimento. O eixo x do gráfico mostra todos os pontos coletados para o percurso, enquanto o eixo y do gráfico exibe as distância em metros relativo a cada ponto do percurso

Terminal/CEPRAMA. Essa relação está representada pela linha vermelha, como pode ser visto no gráfico abaixo (Figura 21).

Figura 21 – Precisão da localização GPS (Terminal/CEPRAMA)



Percurso Terminal/CCET

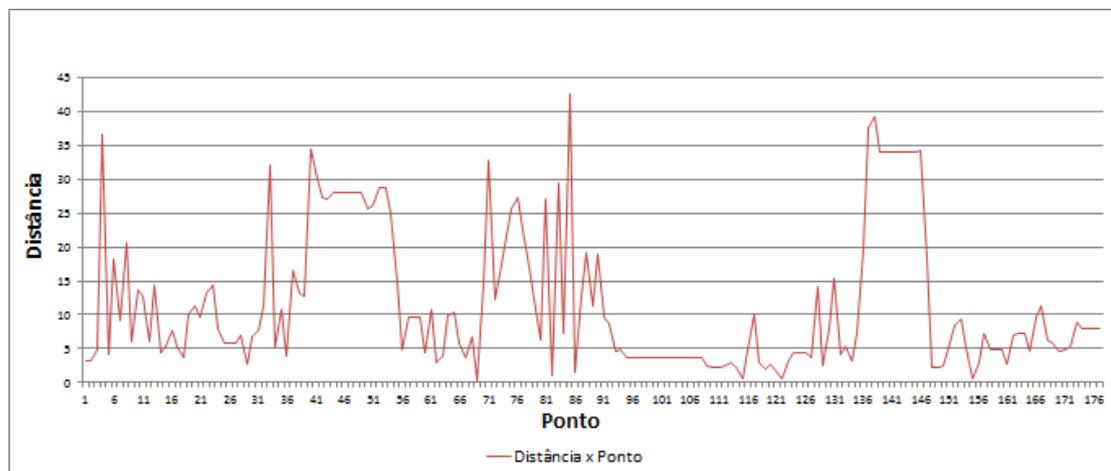
Os dados usados para este percurso foram coletados no dia 09/06/2015 às 9:57. Todos os pontos foram coletados do Terminal da Praia Grande até o Centro de Ciências Exatas e Tecnológicas situado na Universidade Federal do Maranhão (UFMA). É um percurso da categoria TU e nele foram coletados 177 pontos, mas serão mostrados apenas 18 na Tabela 5.

Tabela 5 – Informações o sobre experimento de precisão GPS (Terminal/CCET)

Relação	Coordenada do GPS	Coordenada Corrigida	Distâncias
r_1	latITUDE:-2.5327/longITUDE:-44.3068	latITUDE:-2.5327/longITUDE:-44.3068	$d_1 = 3,2$
r_7	latITUDE:-2.5348/longITUDE:-44.3059	latITUDE:-2.5348/longITUDE:-44.3058	9,23
r_{14}	latITUDE:-2.5366/longITUDE:-44.3028	latITUDE:-2.5366/longITUDE:-44.3028	4,34
r_{21}	latITUDE:-2.5376/longITUDE:-44.3018	latITUDE:-2.5375/longITUDE:-44.3018	9,72
r_{35}	latITUDE:-2.5395/longITUDE:-44.3013	latITUDE:-2.5395/longITUDE:-44.3012	10,81
r_{42}	latITUDE:-2.5417/longITUDE:-44.3009	latITUDE:-2.5419/longITUDE:-44.3008	27,31
r_{56}	latITUDE:-2.5419/longITUDE:-44.3008	latITUDE:-2.5419/longITUDE:-44.3008	4,86
r_{63}	latITUDE:-2.5425/longITUDE:-44.3006	latITUDE:-2.5425/longITUDE:-44.3006	3,9
r_{77}	latITUDE:-2.5470/longITUDE:-44.3033	latITUDE:-2.5486/longITUDE:-44.3046	22,58
r_{84}	latITUDE:-2.5494/longITUDE:-44.3053	latITUDE:-2.5495/longITUDE:-44.3054	7,16
r_{98}	latITUDE:-2.5519/longITUDE:-44.3076	latITUDE:-2.5519/longITUDE:-44.3076	3,71
r_{112}	latITUDE:-2.5520/longITUDE:-44.3075	latITUDE:-2.5519/longITUDE:-44.3075	2,52
r_{126}	latITUDE:-2.5535/longITUDE:-44.3061	latITUDE:-2.5535/longITUDE:-44.3061	4,33
r_{133}	latITUDE:-2.5547/longITUDE:-44.3063	latITUDE:-2.5548/longITUDE:-44.3063	5,29
r_{147}	latITUDE:-2.5548/longITUDE:-44.3075	latITUDE:-2.5548/longITUDE:-44.3076	18,24
r_{154}	latITUDE:-2.5560/longITUDE:-44.3076	latITUDE:-2.5560/longITUDE:-44.3076	5,11
r_{163}	latITUDE:-2.5564/longITUDE:-44.3075	latITUDE:-2.5564/longITUDE:-44.3075	7,27
r_{177}	latITUDE:-2.5572/longITUDE:-44.3085	latITUDE:-2.5572/longITUDE:-44.3084	8,06

Fazendo a mesma relação dos experimentos anteriores, para o percurso Terminal/CCET também foi feito um gráfico (Figura 22) onde será apresentado a variação das distâncias relativo a cada ponto obtido nesta simulação.

Figura 22 – Precisão da localização GPS (Terminal/CCET)



4.2.2 Tempo Estimado de Chegada de um Coletivo em um Determinado Ponto de Parada

O presente aplicativo tem uma funcionalidade muito comum quando trata-se de aplicações que usam sistemas baseados em localização. Aplicações com essa natureza geralmente apresentam rotinas que oferecem a localidade de certos objetos ou lugares de interesse do usuário, e em muitos casos, as mesmas podem calcular a distância entre um determinado objeto em relação a atual posição do usuário ou até mesmo calcular a distância entre dois lugares distintos sem levar em consideração a posição do usuário. Neste não é diferente, mas pra ser mais específico, foi desenvolvido uma rotina que apresentasse previsões sobre o tempo de chegada de coletivos em determinado ponto de parada presente em um itinerário, utilizando-se de informações geográficas, como por exemplo, a distância entre dois pontos geoespaciais.

Para esta importante funcionalidade serão feitos testes para medir o desempenho da aplicação quanto ao percentual de erro que a mesma pode sofrer quando gera previsões de tempo de chegada para um determinado coletivo em um certo percurso. No atual experimento categorizou-se três tipos de distâncias levando como referência os pontos de parada do itinerário do ônibus campus integração. Este itinerário possui 22 pontos de parada, 11 pontos para o sentido UFMA/Terminal (UT) e 11 pontos para o sentido Terminal/UFMA (TU). Os dois sentidos podem ser visto na Figura 23 e a sequência completa dos pontos de parada dos dois sentidos do itinerário, está descrito na Tabela 6.

Figura 23 – Itinerário do Campos Integração. Lado (a) da figura mostra o sentido TU e o lado (b) da figura o sentido UT

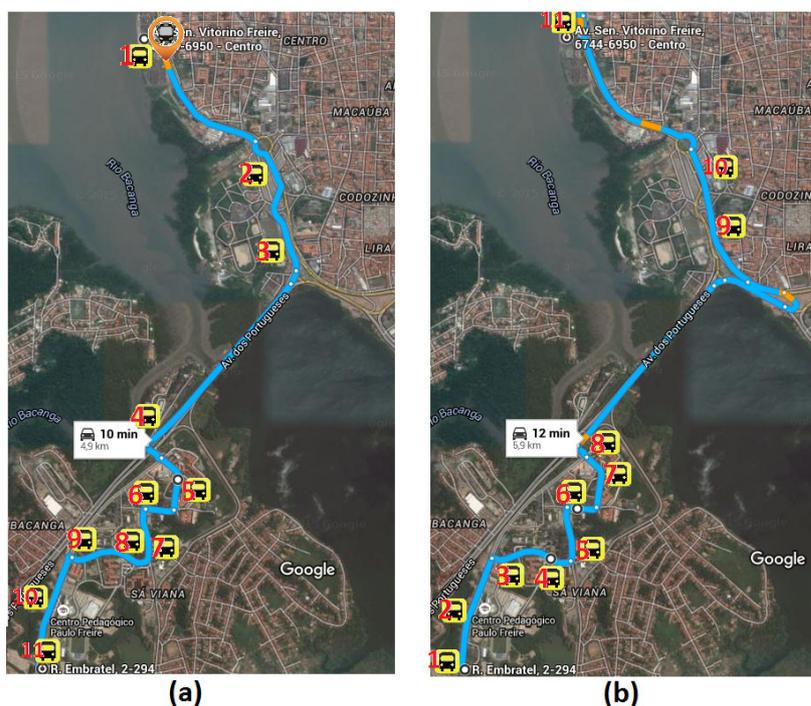


Tabela 6 – A seqüência dos pontos de parada do itinerário nos dois sentidos.

Sentido	Seqüência de Pontos de Parada
UFMA-Terminal (UT)	(1)Ponto Final, (2)Paulo Freire, (3)CCSO, (4)CCET, (5)Área de Vivência, (6)Biblioteca Central, (7)CCBS, (8)Entrada da UFMA(9)CEPRAMA, (10)TRE-MA e (11)Terminal Praia Grande
Terminal-UFMA (TU)	(1)Terminal Praia Grande, (2)Anel Viário, (3)CEPRAMA(lado oposto), (4)Corpo de Bombeiros, (5)CCBS (lado oposto), (6)Biblioteca Central, (7)Área de Vivência, (8)RU, (9)CCSO, (10)Paulo Freire e (11)Ponto Final

Como dito antes, todas as distâncias utilizadas neste experimento foram classificadas em três categorias: a de longa distância, a de média distância e a de curta distância. Um experimento que possui no mínimo 9 pontos de parada será classificado em um experimento de longa distância. Por exemplo, se um ônibus estiver no Terminal Praia Grande (ponto de parada de número 1 do sentido TU) e o usuário desejar saber quanto tempo este ônibus gastará para chegar no CCBS (ponto de parada de número 9 do mesmo sentido TU), este será um experimento de longa distância (oitava linha da Tabela 4.2.2).

Agora se o usuário quiser saber em quanto tempo um ônibus vai demorar para chegar em determinado ponto de parada cuja a soma de todos os pontos que fazem parte deste trajeto contabilizarem entre 4 a 8 pontos de parada, este experimento será categorizado de média distância. Da mesma forma será feito para o experimento de curta distância, porém a soma de todos os pontos de parada desse trajeto terá no máximo 3 pontos de parada.

Em todos os casos de teste a rotina de previsão de tempo sempre foi disparada quando

um coletivo se encontrava próximo a um ponto de parada. Como pode ser visto na parte (a) da Figura 23 o ícone do ônibus próximo ao ponto de parada de número 11.

Percurso Terminal/Paulo Freire

Os dados usados para este percurso foram coletados no dia 16/06/2015 às 17:39. Todos os pontos foram coletados entre o terminal da Praia Grande situado na avenida Senador Vitorino Freire até o Centro Pedagógico Paulo Freire situado na Universidade Federal do Maranhão. Neste percurso foram feitos 10 experimentos como pode ser visto na Tabela 4.2.2.

As duas primeiras linhas da tabela são os experimentos de curta distância. Os experimentos de média distância estão registrados a partir da terceira linha e vão até a sétima linha, enquanto os experimentos de longa distância estão registrados na oitava e nona linha.

Já em relação as colunas, a Tabela 4.2.2 apresenta primeiramente a coluna referente a quantidade de pontos de parada usados para cada teste. Logo após, na segunda e terceira coluna estão o tempo de previsão e o tempo de chegada. Na quarta coluna está a diferença entre o tempo de previsão e o tempo de chegada. E na última coluna, está a sequência de pontos de parada percorridos em cada experimento (baseado na numeração feita pela Tabela 6).

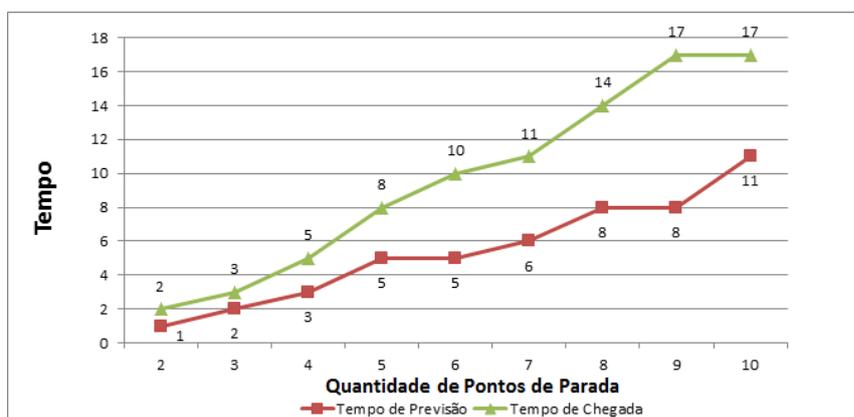
Tabela 7 – Informações sobre os experimentos de tempo estimado para o percurso Terminal/Paulo Freire.

Quantidade	(t_1)	(t_2)	$(t_2 - t_1)$	Sequência dos Pontos de Parada
2	1 min (17h40)	2 min (17h41)	1 min	(5) e (6).
3	2 min (17h41)	3 min (17h42)	1 min	(6), (7) e (8).
4	3 min (17h42)	5 min (17h44)	2 min	(6), (7), (8) e (9).
5	5 min (17h44)	8 min (17h47)	3 min	(3), (4), (5), (6) e (7).
6	5 min (17h44)	10 min (17h49)	5 min	(5), (6), (7), (8), (9) e (10).
7	6 min (17h45)	11 min (17h50)	5 min	(2), (3), (4), (5), (6), (7) e (8).
8	8 min (17h47)	14 min (17h53)	6 min	(1), (2), (3), (4), (5), (6), (7) e (8).
9	8 min (17h47)	17 min (17h56)	9 min	(1), (2), (3), (4), (5), (6), (7), (8) e (9).
10	11 min (17h50)	17 min (17h56)	6 min	(1), (2), (3), (4), (5), (6), (7), (8), (9) e (10).

As informações relacionadas na Tabela 4.2.2 foram usadas para gerar o gráfico da Figura 24. Este gráfico mostra a comparação entre o tempo estimado pelo aplicativo com o tempo real realizado por um coletivo. Nele vê-se o traçado de duas linhas: A

linha vermelha mostra os valores para cada experimento realizado em relação ao tempo estimado. já a linha verde mostra os valores para cada experimento realizado em relação ao tempo real.

Figura 24 – Gráfico que compara o tempo estimado com o tempo de chegada para os experimentos do percurso Terminal/Paulo Freire.



No gráfico (Figura 24) o eixo x exibe a quantidade de pontos de parada para cada experimento. O eixo y exibe uma escala que marca os tempos mínimo e máximo envolvidos nos experimentos deste percurso. Relacionado os dois eixos tem-se os resultados da primeira bateria de experimentos.

Percurso Terminal/CCET

Já para este percurso os dados usados foram coletados no dia 09/06/2015 às 9:57. Todos os pontos foram colhidos entre o terminal da Praia Grande até o Centro de Ciências Exatas e Tecnológicas situado na UFMA. Foram feitos um total de 8 experimentos para este percurso (Tabela 8).

Tabela 8 – Informações sobre os experimentos de tempo estimado para o percurso Terminal/CCET.

Quantidade	(t_1)	(t_2)	$(t_2 - t_1)$	Sequência dos Pontos de Parada
2	2 min (9h59)	2 min (9h59)	0 min	(1) e (2).
3	2 min (9h59)	2 min (9h59)	0 min	(3), (4) e (5).
4	4 min (10h01)	4 min (10h01)	0 min	(5), (6), (7) e (8).
5	5 min (10h02)	4 min (10h01)	-1 min	(2), (3), (4), (5) e (6).
6	5 min (10h02)	6 min (10h03)	1 min	(3), (4), (5), (6), (7) e (8).
7	8 min (10h05)	7 min (10h04)	-1 min	(1), (2), (3), (4), (5), (6) e (7).
8	8 min (10h05)	8 min (10h05)	0 min	(1), (2), (3), (4), (5), (6), (7) e (8)

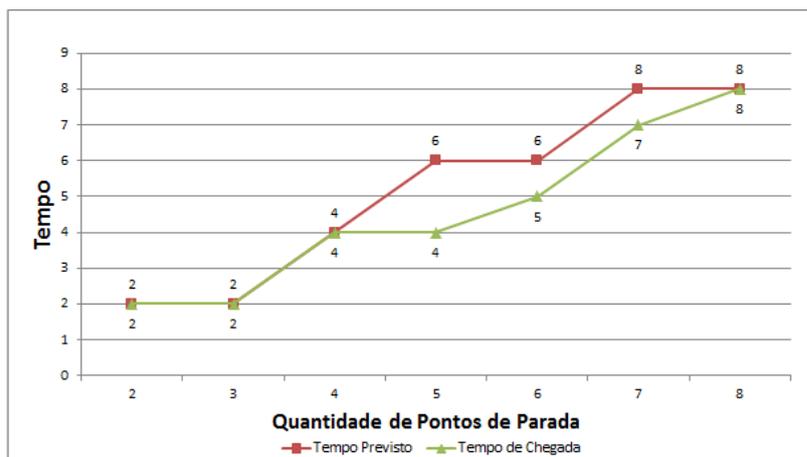
Análogo à tabela anterior, os teste foram agrupados de acordo com a três categoria usadas, porém como este percurso é de menor tamanho, foram realizados apenas experimentos de curta e média distância.

Nas linhas um e dois da tabela estão todos os experimentos de curta distância e da terceira até a última linha da tabela estão todos os experimentos de média distância. Quanto as colunas, a tabela atual segue a mesma estrutura da tabela do percurso anterior.

Para o gráfico ilustrado na Figura 25 estão exibidos todos os experimentos feito para este percurso, mostrando a relação entre o tempo real e tempo estimado. A linha verde mostra todos os tempos reais de chegada de cada experimento e a linha vermelha mostra todos os tempos previstos pelo aplicativo para cada experimento.

Os eixos x e y exibem a quantidade de pontos de parada usados em cada experimento e o tempo relativo para cada teste respectivamente.

Figura 25 – Gráfico que compara o tempo estimado com o tempo de chegada para os experimentos do percurso Terminal/CCET.



4.3 Avaliação dos Resultados

Seguindo a mesma lógica da sessão anterior, serão apresentadas as análises dos resultados em duas etapas. Primeiramente será analisado os resultados dos experimentos quanto a previsão do sinal GPS, mostrando e analisando cada percurso usado para esta bateria de testes. Logo após será analisado os resultados dos experimentos quanto ao tempo de previsão de chegada de um coletivo até um determinado ponto de parada, também mostrando e analisando cada percurso usado para última bateria de testes.

4.3.1 Análise em Relação a Precisão do Sinal GPS

Percurso Biblioteca/Terminal

As distâncias calculadas para este experimento se comportaram com uma precisão regular no que diz respeito ao valor desejado (próximo de zero - Figura 20). No entanto, houveram variações consideráveis em determinados intervalos deste percurso, como por exemplo, do ponto 32 até o ponto 53. Em outras situações houveram pequenas variações que ocorreram entre os pontos 62 a 67, 99 a 101, 105 a 106, 149 a 155. E de forma isolada, estas variações ocorreram em alguns pontos. São eles: 1, 2, 89 e 90.

Resumidamente esses pontos e intervalos de pontos obtiveram uma média de distâncias de 20 a 39 metros de variação entre a coordenada geográfica gerada pelo receptor GPS e a coordenada geográfica corrigida.

Percurso Terminal/CEPRAMA

Para este experimento, as variações de distâncias se comportaram de forma satisfatória, ou seja, a maioria dos pontos ficaram próximos de zero e apenas 5 dos 55 pontos deste percurso apresentaram valores de distância acima de 20 metros. Os quais foram eles, o ponto 6 com o valor de 30,86 metros, o ponto 8 com o valor de 29,6 metros, o ponto 47 com o valor de 23,13 metros, o ponto 51 com o valor de 35,17 metros e o ponto 52 com o valor de 38,86 metros. Como pode ser visto na Figura 21.

Percurso Terminal/CCET

O terceiro e último experimento aproximou-se do primeiro, de forma que também foi considerado um experimento com resultado regular no que diz respeito as distâncias calculadas, possuindo alterações bem acentuadas em alguns intervalos do percurso. O primeiro deles ocorreu entre o ponto 40 e o ponto 54, possuindo valores mínimo e máximo de 24,97 metros e de 34,44 metros respectivamente. Outro intervalo relevante aconteceu deste do ponto 137 até o ponto 146, contendo valores mínimo de 33,92 metros e máximo de 39,16 metros. E por último, um pequeno intervalo do ponto 71 até o ponto 77 com valor mínimo de 21,52 e o valor máximo de 27,37.

Além dos intervalos de variações significativas, também ocorreram pequenas oscilações isoladas com valores acima de 27 metros. Como por exemplo, o ponto 4 com o valor de 36,56 metros, ponto 33 com o valor de 32,09 metros, o ponto 81 com 27,01 metros, o ponto 83 com o valor de 29,59 metros e o maior deles, o ponto 85 com o valor de 42,62 metros (Figura 22).

4.3.2 Análise em Relação ao Tempo Estimado de Chegada de um Coletivo

Percurso Terminal/Paulo Freire

Para o percurso Terminal/Paulo Freire os experimentos de curta distância obtiveram um acréscimo de um minuto em relação ao tempo de previsão feito pelo aplicativo. Por exemplo, no experimento com três pontos de parada, teve-se o tempo estimado de dois minutos, mas o tempo de chegada do coletivo foi de três minutos. Os experimentos de média distância tiveram erros de previsão de três a seis minutos, sendo o melhor caso, o experimento com quatro pontos de parada e o pior caso o de oito pontos de parada. Já os experimentos de longa distância que foram os experimentos com nove e dez pontos de parada, apresentaram erros de previsão maior ou igual a seis minutos, chegando a ter até nove minutos de diferença entre o tempo previsto e o tempo real.

Pode-se ver esse comportamento através do gráfico da Figura 24 gerado sobre os experimentos deste percurso, onde as linhas que representam o tempo de chegada (verde) e o tempo de previsão (vermelho), nos primeiros experimentos estão bem próximas e progressivamente começam a se distanciar uma da outra. Isto mostra que nos experimentos de menor quantidade de pontos de parada não sofreram grandes diferenças e cada vez que se aumentou o quantidade de pontos de parada as previsões passaram a ser menos precisas.

Isso pode ter ocorrido por várias razões, primeiro precisa-se levar em consideração o horário desta simulação (17:39). Pois fora realizada em um momento onde há maior fluxo de universitários fazendo uso do coletivo e ocasionando maior tempo gasto pelo ônibus em embarque e desembarque de usuários em vários pontos de parada. A segunda razão também vinculada ao horário da simulação, está relacionada ao grande número de veículos nas vias, ocasionando lentidão sobre a fluidez do trânsito e provocando uma diferença significativa entre o tempo de previsão e o tempo de chegada do coletivo, como pode ser visto nas linhas da quinta coluna da Tabela 4.2.2.

Percurso Terminal/CCET

Como neste percurso houveram apenas experimentos de curta e média distância (os quais são experimentos que menos apresentam erros), na maioria dos casos o aplicativo teve um bom desempenho e apenas no experimento com cinco, seis e sete pontos de parada, apresentaram variações entre o tempo real e o tempo estimado.

Através do gráfico da Figura 25, pode-se ver que as duas linhas que representam os dois tempos em questão, estão na maioria dos casos bem próximas. Isto quando não estão sobrepostas (Figura 25). E quando ocorreram os casos de erros de previsão, em vez de o tempo de chegada ser maior que o tempo previsto, ocorreu o contrário, onde as diferenças entre os dois tempos foram menor que zero (quinta coluna da Tabela 8).

As observações feitas para os experimentos deste percurso, mais uma vez são sobre

o horário em que foi feito tal simulação. Pois esta ocorreu às 9:57 horas da manhã, e por isso as previsões feitas pelo aplicativo na sua maioria foram bem próximas das reais. Diferentemente dos experimentos do percurso anterior, este percurso foi realizado em um momento onde o fluxo de passageiros e o trânsito de veículos é menor, evitando que o coletivo parasse muitas vezes ao longo percurso. Devido a isto, o ônibus manteve a velocidade constante na maior parte de sua trajetória, e por isso, as previsões feitas pelo aplicativo foram em sua maioria satisfatórias.

5 Conclusão

Apesar de muito comuns em outras cidades brasileiras, a cidade de São Luís ainda não possui aplicações que disponibilizam informações a respeito do transporte público em tempo real. Até então, pouco se investiu para mudar esta realidade e muito deixa a desejar quando fala-se de sistema de transporte público nesta capital. Mesmo com a população na zona metropolitana crescendo no decorrer dos últimos anos e conseqüentemente a frota de coletivos, ainda não se tem um serviço de qualidade e boa parte da população ainda espera por melhorias neste setor.

Sabendo desta realidade, este trabalho lança uma proposta para tentar mudar esta situação ou pelo menos amenizá-la na medida do possível, pois com as informações de tempo de chegada ou com disponibilização da localização aproximada dos coletivos, ainda não é possível mudar todo cenário negativo do sistema de transporte, uma vez que o mesmo ainda precisa de grandes melhorias em diferentes áreas. Porém se esta proposta chegar nas mãos de quem diariamente usa coletivos, sem dúvida irá melhorá a rotina de milhares de pessoas.

Com a implementação desta aplicação, deve-se trazer em um futuro bem próximo, alternativas para melhorar a interação entre os usuários e o sistema de transporte público da cidade de São Luís, e em especial, os frequentadores da Universidade Federal do Maranhão. As pessoas interessadas em saber sobre os ônibus de sua importância poderão receber ou gerar informações sobre os mesmos.

Todas essas informações serão apresentadas através de um aplicativo para *smartphone*. Nele terá uma tela com um mapa onde apresentará a localização de todos os coletivos cujo a posição será baseada na localização de colaboradores. Para este trabalho os colaboradores serão em sua maioria os universitários da UFMA, pois o caso de teste usado neste trabalho envolveu uma das linhas de ônibus que rodam dentro da cidade universitária.

Além da localização dos coletivos também será possível calcular o tempo de chegada de um ônibus até um determinado ponto de ônibus. Basta o usuário clicar em qualquer ícone de ponto de ônibus cujo o coletivo a qual se quer saber o tempo, esteja no mesmo sentido do ponto de ônibus clicado.

Para a criação de tal aplicação, foi necessário fazer o levantamento de requisitos e uma modelagem antes mesmo de começar a implementar o aplicativo. Logo viu-se a necessidade de dividir a fase de implementação em duas partes. Primeiramente fez-se a parte da aplicação que executaria nos *smartphones* que possuísse a plataforma Android. Esta parte da aplicação é o aplicativo que será usado pelos usuários. Depois foi implementado o servidor para interagir com parte cliente da aplicação, pois este serviço web seria responsável em colher dados de localização de todos os usuários e corrigi-los quando necessário.

A preferência para desenvolver tal aplicação usando a plataforma Android fez com que a curva de aprendizagem obtivesse um comportamento muito satisfatório e eficiente, refletindo em um bom e rápido desenvolvimento. Outro fator positivo em usar esta plataforma, é o uso da linguagem JAVA para o desenvolvimento. Esta linguagem é bem conhecida e geralmente usada em trabalhos no decorrer de todo curso de ciência da computação.

A combinação de linguagens já conhecidas, ótimas ferramentas de desenvolvimento e com utilizações de conceituadas APIs, pôde-se fazer uma aplicação com a arquitetura cliente-servidor que mantivesse uma boa comunicação entre as partes.

Em relação ao aplicativo final, chegou-se a um resultado agradável, uma aplicação onde tivesse todas as funcionalidades cruciais exigidas. Todas as informações relevantes para os usuários de coletivos foram disponibilizadas na interface do aplicativo: Localização dos pontos de parada e de coletivos pertencente ao itinerário estudado neste trabalho, o tempo de chegada de um coletivo até um ponto de parada de interesse do usuário e a exibição de todo itinerário mesmo quando o usuário estiver *offline*.

Na interface também são apresentados alguns botões. Estes são responsáveis por enviar solicitações para o servidor e para as APIs utilizadas. Através deles o usuário interage com a aplicação para disponibilizar sua localização como referência para localizar um coletivo ou somente solicita os coletivos já localizados. Sem contar com o botão que aplica zoom sobre o mapa e o botão que muda o tipo de exibição do mapa.

Depois do aplicativo pronto, foram feitos alguns experimentos para verificar como o mesmo iria se comportar. Para efetuar os experimentos, foi necessário obter dados. Estes dados foram coletados através de viagens feitas nas rotas dos coletivos que seriam usadas nos testes. A fazer um trajeto na rota usada para os experimentos, foram coletados os dados geoespaciais ao longo de todo o percurso, e armazenados em alguns arquivos. Estes mesmos arquivos foram usados para simular as viagens na etapa dos experimentos e assim observar o comportamento do aplicativo.

Dois tipos de testes foram feitos para medir o desempenho da aplicação. O primeiro foi observar o comportamento do aplicativo em relação à precisão do sinal recebido pelo receptor GPS do dispositivo. O segundo foi comparar o tempo de previsão feito pelo aplicativo em relação ao tempo real em que um coletivo gasta para percorrer um determinado trajeto.

Na primeira bateria de testes os resultados dos experimentos se comportaram de maneira razoável. Para os percursos usados nesta bateria, dos três experimentos realizados, o primeiro e último apresentaram alterações bem acentuadas em alguns intervalos do percurso, apresentando um número maior de variações entre a coordenada geográfica gerada pelo receptor GPS e a coordenada geográfica corrigida, ao passo que no segundo experimento teve um melhor retrospecto, a maioria dos pontos ficaram próximos de zero e somente 5 dos 55 pontos deste percurso apresentaram variações consideradas significativas.

O interessante é que mesmo com variações acentuadas ou pequenas, o aplicativo sempre corrige as imprecisões. Em todas situações de todos os percursos usados para esta bateria de testes, a aplicação nunca lançou um coletivo fora da via. Isto leva a concluir que esta rotina se comportou de maneira aceitável.

Para a segunda bateria de testes foram usados dois percursos. Para o primeiro percurso foram feitos 10 experimentos de curta, média e longa distância. Os experimentos de menor quantidade de pontos de parada não sofreram grandes diferenças, mas cada vez que se aumentou a quantidade de pontos de parada, as previsões passaram a ser menos precisas. Já no segundo percurso usado, foram feitos 8 experimentos de curta e média distância. Na maioria dos casos o aplicativo teve um bom comportamento e apenas em três experimentos apresentaram variações entre o tempo real e o tempo estimado.

Vários fatores externos à aplicação fizeram com que os experimentos de tempo estimado assumissem tais resultados. Todos os fatores estão relacionados ao horário em que os dados dos percursos foram coletados, pois dependendo do horário, o fluxo de veículos nas vias e a quantidade de usuários que fazem uso dos coletivos pode variar. Se as tais situações ocorrem apresentando um grande número de passageiros e um grande número de veículos nas vias, fará com que ocorra inconstância na previsão do tempo estimado. Agora se em outro horário o trânsito de passageiros e veículos nas vias for menor, as previsões tentem a mostrar pouca variação levando esta rotina apresentar valores ideais.

Vendo estas observações, leva-se a crer que tal aplicativo pode ser usado em situações reais, mas que ainda existem trabalhos futuros a serem desenvolvidos. Apesar de algumas melhorias que ainda devem ser feitas, tal aplicação já pode chegar nas mãos de usuários para testes mais minuciosos e assim ter-se uma grande variedade de *feedbacks* que contribuirão para melhorar mais ainda o aplicativo.

O intuito sempre foi levar o aplicativo para o público em geral mesmo que não ocorresse imediatamente. Por isso neste trabalho optou-se primeiramente a adotar um estudo de caso e a partir daí aumentar o número de rotas gradativamente assim que forem projetadas novas versões para a aplicação, fazendo com que ela seja acessível para a maior quantidade possível de usuários.

Apesar de ser um pequeno passo, trata-se de uma iniciativa para um futuro promissor, a fim de contribuir para o avanço tecnológico desta capital. Espera-se que em futuro não tão distante, este aplicativo possa estar disponível para a sociedade em geral, visando o bem estar e conforto da mesma, além de torná-la a par de tais recursos que lhe possa proporcionar vantagens em seu dia a dia.

5.1 Trabalhos Futuros

Como o aplicativo proposto ainda está na sua primeira versão, o ideal seria ter todas as funcionalidades bem executadas para trazer resultados satisfatórios. Por isso a aplicação

ainda não apresenta uma aparência tão agradável e convidativa visualmente, pois o foco foi desenvolver seus requisitos funcionais. Então, deverá ter uma preocupação com o visual do aplicativo para que futuramente apresente uma interface bem mais atrativa para quem for usá-lo.

Falando em requisitos funcionais, dois dos principais deles também devem ser melhorados. O algoritmo de correção do sinal GPS na maioria dos casos funcionou perfeitamente, mas se por acaso o aplicativo for executado em um *smartphone* que possua uma tecnologia de receptor GPS mais antiga, a precisão da localização pode sofrer variações maiores do que as experimentadas neste trabalho, pois o algoritmo pode se comportar de maneira inesperada, uma vez que ele faz uma verificação de pontos da rota em um raio limitado por 45 metros, e se o receptor gerar variações maiores que essa, ele pode não retornar bons resultados para o usuário.

Da mesma maneira, o algoritmo de previsão de chegada dos ônibus precisa ser melhorado, pois nos experimentos apresentados, viu-se que para alguns casos ele variou muito além do desejado, mesmo que tenham havido fatores externos que fizeram com que o tempo estimado variasse muito em relação ao tempo real. Então será necessário desenvolver um algoritmo que consiga agregar todas essas situações adversas como variáveis do problema.

Além de tais situações adversas, uma alternativa para o melhoramento dos resultados da rotina de tempo estimado, seria fazer uma coleta de dados sobre os tempos estimados que são gerados nas rotas ao longo do tempo. Tendo essas informações armazenadas para serem usadas posteriormente, a rotina faria comparações entre estimativas similares, isto é, que envolvem o mesmo percurso e os mesmos pontos de parada. Desta maneira acredita-se que a aplicação faria previsões mais precisas.

Referências

ABINEE. **Smartphones já representam 76% do mercado de celulares.** São Paulo, 2014. Disponível em: <<http://www.abinee.org.br/noticias/com282.htm>>. Acesso em: 24 fev. 2015.

ANDROID. **A história do Android.** Android, 2015. Disponível em: <<http://www.android.com/history/>>. Acesso em: 09 mai. 2015.

AVRITZER, D. **Geometria Analítica e Álgebra Linear: Uma Visão Geométrica.** Belo Horizonte: Editora UFMG, 2009. Disponível em: <<http://www.mat.ufmg.br/ead/acervo/livros/Geometria%20Analitica%20e%20Algebra%20Linear%20-%20Uma%20Visao%20Geometrica%20-%20TII.pdf>>. Acesso em: 21 jul. 2015.

BORDIN, M. V. **Introdução a Arquitetura Android.** Dissertação (Artigo) — Sociedade Educacional Três de Maio, Rio Grande do Sul, Julho 2012. Disponível em: <<http://sites.setrem.com.br/stin/2012/anais/Maycon.pdf>>. Acesso em: 3 abr. 2016.

BORSNTEIN, D. **Dalvik VM Internals.** [s.n.], 2008. Disponível em: <<https://sites.google.com/site/io/dalvik-vm-internals>>. Acesso em: 16 jun. 2015.

CITTAMOBI. **Informação do seu ônibus, na hora certa.** [S.l.], 2014. Disponível em: <<http://www.cittamobi.com.br/>>. Acesso em: 15 jun. 2015.

DEVELOPER. **Activity.** [S.l.], 2015. Disponível em: <<https://developer.android.com/reference/android/app/Activity.html>>. Acesso em: 13 mai. 2015.

_____. **Android Studio Overview.** [S.l.], 2015. Disponível em: <<http://developer.android.com/tools/studio/index.html>>. Acesso em: 23 mai. 2015.

_____. **Como posso começar a desenvolver programas Java com o Java Development Kit (JDK)?** [S.l.], 2015. Disponível em: <https://www.java.com/pt_BR/download/faq/develop.xml>. Acesso em: 23 mai. 2015.

_____. **Installing the Eclipse Plugin.** [S.l.], 2015. Disponível em: <<http://developer.android.com/sdk/installing/installing-adt.html>>. Acesso em: 23 mai. 2015.

_____. **Migrating to Android Studio.** [S.l.], 2015. Disponível em: <<http://developer.android.com/sdk/installing/migrate.html>>. Acesso em: 23 mai. 2015.

_____. **Starting an Activity.** [S.l.], 2015. Disponível em: <<https://developer.android.com/training/basics/activity-lifecycle/starting.html>>. Acesso em: 13 mai. 2015.

DIRECTION. **The Google Directions API.** [S.l.], 2015. Disponível em: <<https://developers.google.com/maps/documentation/directions/>>. Acesso em: 17 jun. 2015.

EMBRAPA. **Emprapra Monitoramento por Satélite.** [S.l.], 2013. Disponível em: <http://www.sat.cnpm.embrapa.br/conteudo/missao_gps.php>. Acesso em: 19 mai. 2015.

- FERNANDES, N. C. **Sistema Operacional Android**. Dissertação (Artigo) — Universidade Federal do Fluminense, São Luís, Julho 2012. Disponível em: <<http://www.midiacom.uff.br/~natalia/2012-1-sisop/tgrupo1.pdf>>. Acesso em: 17 fev. 2015.
- FERREIRA, A. J. de A. et al. **Percepção Socioambiental na Região Metropolitana da Grande São Luís-Maranhão**. Dissertação (Artigo) — Universidade Federal do Maranhão, São Luís, 2015. Disponível em: <<http://www.secid.ma.gov.br/files/2015/04/Percep%C3%A7%C3%A3o-Ambiental-na-RMGSL.pdf>>. Acesso em: 17 jan. 2016.
- G1. **Passageiros reclamam de condições dos terminais de ônibus de São Luís**. [S.l.], 2015. Disponível em: <<http://g1.globo.com/ma/maranhao/noticia/2015/01/passageiros-reclamam-de-condicoes-dos-terminais-de-onibus-de-sao-luis.html>>. Acesso em: 25 mar. 2015.
- GEOLOCATION. **The Google Maps Geolocation API**. [S.l.], 2015. Disponível em: <<https://developers.google.com/maps/documentation/geolocation/>>. Acesso em: 17 jun. 2015.
- GPS. **Control Segment**. [S.l.], 2015. Disponível em: <<http://www.gps.gov/systems/gps/control/>>. Acesso em: 13 mai. 2015.
- _____. **GPS Modernization**. [S.l.], 2015. Disponível em: <<http://www.gps.gov/systems/gps/modernization/>>. Acesso em: 21 mai. 2015.
- _____. **New Civil Signals**. [S.l.], 2015. Disponível em: <<http://www.gps.gov/systems/gps/modernization/civilsignals/>>. Acesso em: 15 mai. 2015.
- _____. **Space Segment**. [S.l.], 2015. Disponível em: <<http://www.gps.gov/systems/gps/space/>>. Acesso em: 13 mai. 2015.
- IBGE. **Instituto Brasileiro de Geografia e Estatística**. [S.l.], 2014. Disponível em: <<http://www.cidades.ibge.gov.br/xtras/perfil.php?lang=&codmun=211130&search=maranhao|sao-luis|infograficos:-informacoes-completas>>. Acesso em: 12 mar. 2015.
- IMIRANTE. **Usuários do transporte coletivo estão insatisfeitos**. [S.l.], 2011. Disponível em: <<http://imirante.globo.com/imperatriz/noticias/2011/08/17/usuarios-do-transporte-coletivo-estao-insatisfeitos.shtml>>. Acesso em: 23 mar. 2015.
- LECHETA, R. R. **Google Android: aprenda a criar aplicações para dispositivos móveis com Android SDK**. São Paulo: Novatec, 2010.
- LOCATION. **Getting the Last Known Location**. [S.l.], 2015. Disponível em: <<https://developer.android.com/training/location/retrieve-current.html>>. Acesso em: 17 jun. 2015.
- LOUREIRO, A. A. F.; MATEUS, G. R. In: _____. **Introdução à Computação Móvel**. [s.n.], 2015. Disponível em: <http://homepages.dcc.ufmg.br/~loureiro/cm/docs/cm_livro_1e.pdf>. Acesso em: 28 jun. 2015.
- MAPS. **Adding Maps**. [S.l.], 2015. Disponível em: <<https://developer.android.com/training/maps/index.html#features>>. Acesso em: 17 jun. 2015.

MOOVIT. **Moovit: Metro, Ônibus e Trens**. [S.l.], 2014. Disponível em: <<http://www.techtodo.com.br/tudo-sobre/moovit.html>>. Acesso em: 12 jun. 2015.

PROTESTE. **Associação Brasileira de Defesa do Consumidor: Transporte público deixa a desejar**. [S.l.], 2015. Disponível em: <<http://www.proteste.org.br/familia/nc/noticia/transporte-publico-deixa-a-desejar>>. Acesso em: 23 mar. 2015.

RGCI. **Revista de Gestão Costeira Integrada (JICZM - Journal of Integrated Coastal Zone Management)**. [S.l.], 2007. Disponível em: <<http://www.aprh.pt/rgci/glossario/gps.html>>. Acesso em: 21 mai. 2015.

RODRIGUES, R. de O. **O Sistema GPS e algumas de suas aplicações**. Dissertação (Monografia) — Universidade Federal do Maranhão, São Luís, 1996.

SCHILLER, J.; VOISARD, A. **Location-based Services**. San Francisco, CA, USA: Morgan Kaufmann, 2004.

SMTT. **Secretaria Municipal de Trânsito e Transportes**. [S.l.], 2016. Disponível em: <http://www.saoluis.ma.gov.br/subportal_noticia.asp?id_noticia=16670>. Acesso em: 20 fev. 2016.

SOURCE. **The Android Source Code**. [S.l.], 2015. Disponível em: <<http://source.android.com/source/index.html>>. Acesso em: 09 mai. 2015.

SOUSA, B. L. M. de. **Transporte Coletivo Público na cidade de São Luís - MA: Comparações e pós-implantação do SIT - Sistema Integrado de Transportes**. Dissertação (Dissertação) — Universidade Federal de São Carlos, São Carlos, 2013.

TRANSIT. **O Maneira mais rápida para chegar em sua cidade**. [S.l.], 2015. Disponível em: <<http://transitapp.com/>>. Acesso em: 15 jun. 2015.