

**UNIVERSIDADE FEDERAL DO PAMPA**

**RODRIGO MARTINS ALVES**

**SISTEMA DE RASTREAMENTO DE TRANSPORTE PÚBLICO ATRAVÉS DE  
HARDWARE E SOFTWARE**

**Bagé  
2016**

**RODRIGO MARTINS ALVES**

**SISTEMA DE RASTREAMENTO DE TRANSPORTE PÚBLICO ATRAVÉS DE  
HARDWARE E SOFTWARE**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação da Universidade Federal do Pampa, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Computação.

Orientador: Alexandro Gularte Schäfer

**Bagé  
2016**

Ficha catalográfica elaborada automaticamente com os dados fornecidos  
pelo(a) autor(a) através do Módulo de Biblioteca do  
Sistema GURI (Gestão Unificada de Recursos Institucionais) .

A474s Alves, Rodrigo  
SISTEMA DE RASTREAMENTO DE TRANSPORTE PÚBLICO ATRAVÉS DE  
HARDWARE E SOFTWARE / Rodrigo Alves.  
70 p.

Trabalho de Conclusão de Curso(Graduação)-- Universidade  
Federal do Pampa, ENGENHARIA DE COMPUTAÇÃO, 2016.  
"Orientação: Alexandro Schäfer".

1. Mobilidade Urbana. 2. Smartphone. 3. Transporte  
Coletivo. 4. Serviço Web. 5. Arduino. I. Título.

**RODRIGO MARTINS ALVES**

**SISTEMA DE RASTREAMENTO DE TRANSPORTE PÚBLICO ATRAVÉS DE  
HARDWARE E SOFTWARE**

Trabalho de Conclusão de Curso  
apresentado ao Curso de Engenharia de  
Computação da Universidade Federal do  
Pampa, como requisito parcial para  
obtenção do Título de Bacharel em  
Engenharia de Computação.

Trabalho de Conclusão de Curso defendido e aprovado em: 03 de dezembro de  
2016.

Banca examinadora:

---

Prof. Dr. Alexandro Schafer  
Orientador  
UNIPAMPA

---

Prof. Dr. Érico Hoff do Amaral  
UNIPAMPA

---

Prof<sup>a</sup>. Dr<sup>a</sup>. Sandra Dutra Piovesan  
UNIPAMPA

Dedico este trabalho a Deus, minha  
família e namorada.

## **AGRADECIMENTO**

Ao Prof. Dr. Alexandro Schafer e aos professores do curso de engenharia de computação da UNIPAMPA. Agradeço também a empresa Stadtbuss por possibilitar que os equipamentos necessários ao projeto piloto fossem instalados em um de seus ônibus e a prefeitura de Bagé-RS, por fornecer as informações das paradas de ônibus.

## RESUMO

O transporte público é um dos principais meios de transporte utilizado pela população brasileira, no entanto este tipo de transporte ainda é ineficiente no país. Um dos problemas presenciados pela população é o atraso dos ônibus, que pode acontecer por diversos motivos. O rastreamento de ônibus através de GPS e GSM/GPRS pode contribuir para solucionar problemas relacionados com a falta de previsibilidade da chegada e acompanhamento do itinerário do ônibus. Este projeto tem como objetivo o desenvolvimento de um sistema de rastreamento de transporte público que possibilita a estimativa do horário de chegada de um determinado ônibus em uma parada. Para isso, foi integrado um módulo de hardware para determinar a localização do veículo, assim como o desenvolvimento de um aplicativo para *smartphone*. Com isso espera-se que esse projeto possa contribuir para a utilização do transporte público, possibilitando que os passageiros possam programar melhor o seu dia, e empresas possam melhorar a gestão/fiscalização de sua frota. Sendo assim, o passageiro poderá ter um maior controle de seu itinerário, de acordo com a chegada do veículo na parada solicitada.

Palavras-Chave: Mobilidade Urbana, Transporte Coletivo, *Smartphone*, Serviço Web, Arduino, GPS, GSM/GPRS.

## **ABSTRACT**

Public transportation is one of the main means of transport utilized by the population, however it is still inefficient. One of the problems seen by the population is the delay of the bus. This can happen for several reasons such as traffic or accidents on the road. Tracking a bus through GPS and GSM/GPRS can help solve some of the problems related to lack of predictability of arrival, through the planning and management of the bus system. This project proposes the development of a public transportation tracking system, which will estimate the arrival of a bus at a desired bus stop by the passenger as well as allowing the passenger to see the bus in real time. Thus, the passengers will have a better control over their time, according to the arrival of the bus at the requested stop. For this to become a reality, the tracking system will be integrated and programmed into a hardware module to determine the location of the vehicle and a mobile application to visualize the bus on a map. It is expected to make public transport an efficient means of transportation, where the user can have a better planning due to more accurate itinerary and the company can make more concise decisions.

Keywords: Map, Mobility, Public Transportation, App, Web Service, Arduino, GPS, GSM/GPRS.

## LISTA DE FIGURAS

Figura 1 - Estimativa da participação do transporte público no total de viagens das regiões metropolitanas brasileiras. ....	19
Figura 2 - Representação da longitude e latitude, que formam as coordenadas geográficas da Terra.....	21
Figura 3 - Rede GSM.....	22
Figura 4 - Rede GPRS.....	23
Figura 5 - Placa básica de um Arduino Uno. ....	30
Figura 6 - Sequência metodológica para a modelagem do sistema de rastreamento de transporte público. ....	32
Figura 7 - Visão geral do sistema de rastreamento de ônibus.....	33
Figura 8 - Tipos de requisitos não-funcionais. ....	36
Figura 9 - Diagrama do modelo em cascata.....	37
Figura 10 - Diagrama de caso de uso do projeto de rastreamento de ônibus. ....	41
Figura 12 - Diagrama de sequência para acesso ao mapa do usuário.....	48
Figura 13 - Diagrama de atividade de carregar informações sobre um ônibus.....	49
Figura 14 - Diagrama de atividade pesquisa de pontos de ônibus. ....	50
Figura 15 – Modelo ER.....	51
Figura 16 – Arquitetura Web do sistema de rastreamento de transporte público. ...	52
Figura 17 - IDE PyCharm.....	53
Figura 18 - Mapa da linha Damé x Malafaia da empresa Stadbus da cidade de Bagé-RS.....	54
Figura 19 – Parte da linha “Damé x Malafaia” com as suas paradas de ônibus e a localização do veículo.....	54
Figura 20 - Informações das paradas de ônibus a serem mostradas ao passageiros. ....	55
Figura 21 - Foto da vista lateral e vista frontal da parada na Avenida Santa Tecla, nº 1188.....	56
Figura 22- Aplicativo no Android com o menu de opções.....	56
Figura 23 - Modo on-line do aplicativo Android.....	57
Figura 24 - <i>Hardware</i> utilizado neste projeto. ....	58
Figura 25 – Fluxograma do funcionamento do sistema embarcado. ....	58
Figura 26 - Hardware instalado próximo a cabine do motorista. ....	60

Figura 27 - Mapa do percurso realizado pelo ônibus com as velocidades. ....	61
Figura 28 - Velocidades máximas distribuídas durante os horários do dia. ....	62

## LISTA DE TABELAS

Tabela 1 - Tabela de prioridade de requisitos.....	34
Tabela 2 - Tabela de Risco de Requisitos. ....	35
Tabela 3 - Requisitos funcionais.....	39
Tabela 4 - Requisitos não funcionais.....	40
Tabela 5 - Descrição do caso de uso obter posição do ônibus.....	42
Tabela 6 - Descrição do caso de uso enviar dados da posição do ônibus. ....	42
Tabela 7 - Descrição do caso de uso identificar objetos.....	43
Tabela 8 - Descrição do caso de uso identificar rotas dos veículos. ....	43
Tabela 9 - Descrição do caso de uso obter informações sobre o ônibus e parada de ônibus. ....	44
Tabela 10 - Descrição do caso de uso verificar tempo de chegada do próximo ônibus. ....	44
Tabela 11 - Descrição do caso de uso pesquisar pontos de ônibus.....	45
Tabela 12 - Descrição do caso de obter informação sobre o itinerário.....	45

## LISTA DE ABREVIATURAS E SIGLAS

API – *Application Program Interface*  
APN - Nome do Ponto de Acesso  
AT – *Attention*  
AVL - Sistema Automatic vehicle location  
BSC - *Base Station Controller*  
CSS - *Cascading Style Sheets*  
DVM - *Dalvik virtual machine*  
EER - Entidade-Relacionamento Estendido  
ER - Entidade Relacionamento  
ESRI - *Environmental Systems Research Institute*  
GLONASS - *GLObal NAVigation Satellite System*  
GNSS - Navegação Global por Satélite  
GPRS - *General Packet Radio Service*  
GPS - *Global Positioning System*  
GSM - *Global System for Mobile Communications*  
HAL - *Abstração de Hardware*  
HTTP - *HyperText Transfer Protocol*  
IDE - *Integrated Development Environment*  
iOS - *iPhone Operating System*  
JNI - *Java Native Interface*  
JSON - *JavaScript Object Notation*  
LED - *Light Emitting Diode*  
MEO - Meia Órbita da Terra  
MSC - *Mobile Switching Center*  
PHP - *Hypertext Preprocessor*  
PSTN - *Public Switched Telephone Network*  
REST - *Web Service Representational State Transfer*  
RFID - *Radio Frequency IDentification*  
SGBD - Sistema Gerenciador de Banco de Dados  
SOA - Arquitetura Orientada a Serviço  
SSL - *Secure Sockets Layer*

TI – Tecnologia da Informação

TTFF - *Time to First fix*

UML - Linguagem Unificada de Modelagem

WEB - World Wide Web

XML - eXtensible Markup Language

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>15</b>
1.1	Problema de pesquisa	16
1.2	Objetivo	16
1.2.1	Objetivos específicos	16
1.3	Trabalhos correlatos	17
1.4	Estrutura do trabalho	18
<b>2</b>	<b>CONCEITOS GERAIS E REVISÃO DE LITERATURA</b>	<b>19</b>
2.1	Infraestrutura de transporte no Brasil	19
2.2	Tecnologias relacionadas ao posicionamento geográfico e telecomunicação móvel	20
2.2.1	Sistema global de navegação por satélite (GNSS)	20
2.2.2	<i>Global System for Mobile Communications (GSM) / General Packet Radio Service (GPRS)</i>	21
2.3	Tecnologias relacionadas ao desenvolvimento do sistema	23
2.3.1	Arquitetura orientada a serviço	23
2.3.2	<i>Web services</i>	24
2.3.3	<i>Web service REST</i>	24
2.3.4	Python/Django	26
2.3.5	JSON	26
2.3.6	Banco de dados	26
2.3.7	JavaScript	26
2.3.8	Leaflet	27
2.3.9	Bootstrap	27
2.3.10	Android	28
2.3.11	Arduino	29
2.3.12	Comando AT	30
<b>3</b>	<b>METODOLOGIA PESQUISA</b>	<b>32</b>
3.1	Sistema de Rastreamento	32
3.2	Análise dos requisitos	34
3.2.1	Requisitos funcionais	34
3.2.2	Requisitos não funcionais	35
3.2.3	Projeto de software	36

3.3	Definição de ferramentas e tecnologias.....	37
3.4	Desenvolvimento do sistema.....	38
4	IMPLEMENTAÇÃO DO PROJETO .....	39
4.1	Modelagem .....	39
4.1.1	Requisitos funcionais .....	39
4.1.2	Requisitos não funcionais.....	40
4.1.3	Diagrama de caso de uso .....	41
4.1.4	Diagrama de sequência .....	45
4.1.5	Diagrama de atividade .....	49
4.1.6	Desenvolvimento do banco de dados – modelo entidade relacionamento 50	
4.2	Arquitetura modelada .....	52
4.3	Desenvolvimento do sistema.....	53
4.3.1	Serviço Web.....	53
4.3.2	Páginas Web .....	53
4.3.3	Aplicativo .....	56
4.3.4	Desenvolvimento e instalação do hardware .....	57
4.4	Projeto Piloto.....	59
5	CONCLUSÕES E TRABALHOS FUTUROS.....	63
	REFERÊNCIAS .....	65
	APÊNDICE .....	68
	ANEXOS.....	70

## 1 INTRODUÇÃO

Ao longo dos últimos anos aumentou consideravelmente o número de automóveis nas ruas das cidades brasileiras. Esse aumento foi consequência de incentivos fiscais concedidos à população (Lerner, 2009, p. 42). Segundo Lerner (2009, p. 42), as vendas anuais de automóveis aumentaram, entre 1995 e 2000, e dobraram as vendas de motocicletas (atingindo 500 mil por ano). O aumento aconteceu também até 2005, assim, atingindo 1,6 milhões de motocicletas em 2007 (Lerner, 2009, p. 42) e cerca de 3,8 milhões em 2015 de acordo com DENATRAN (2015). Esse aumento resultou em prejuízos na mobilidade urbana e no aumento da emissão de gases nocivos ao meio ambiente.

De acordo com Lerner (2009, p. 43), o nível de poluição por passageiro/KM, quando se utiliza automóvel, é 17 vezes maior do que quando se utiliza ônibus. Já uma motocicleta polui 32,3 vezes mais que o ônibus.

Ainda que o transporte público seja um dos principais meios de locomoção utilizados no Brasil, com cerca de 29% do total de viagens motorizadas nas regiões metropolitanas Brasileiras (ANTP, 2013), ele é ineficiente e impõe aos usuários esperas e superlotação. Um dos principais problemas presenciados pela população é o atraso dos ônibus, o qual pode acontecer por uma série de motivos.

Novas tecnologias têm sido aplicadas em diferentes setores da sociedade para a melhoria de diversos segmentos, e o mesmo deve acontecer para o transporte público, em que sistemas de gerenciamento voltados à tomada de decisões vêm sendo desenvolvidos e utilizados em nível mundial.

A utilização de ferramentas tecnológicas como o rastreamento pode vir a amenizar os problemas relacionados a falta de previsibilidade do transporte coletivo urbano. Novas ferramentas vêm sendo utilizadas na busca pela melhoria dos serviços relacionados ao transporte público em vários países, como Estados Unidos, Austrália e Inglaterra (MAGALHÃES, 2008, p. 23). Como exemplo de ferramentas, pode-se citar o *Radio Frequency IDentification* (RFID), rastreamento de ônibus via *Global Positioning System* (GPS) e *Global System for Mobile Communications* (GSM).

Através do planejamento e gerenciamento da frota, se torna possível a realização de novos meios de organização para empresas de transporte público. A partir da utilização de um sistema de rastreamento, é possível que se tenha um

maior controle sobre uma determinada linha ou frota, tanto por parte da empresa quanto ao passageiro. É possível ainda determinar o tempo de chegada de um ônibus até a próxima parada, o tempo do percurso, em quais paradas não parou, velocidade média, etc.

Auxiliado pela redução do preço dos *smartphones* e pela disponibilidade das redes 3G e 4G, o número de usuários de *smartphones* está crescendo em uma taxa média de 4,7% por ano (BYANSHUL SRIVASTAVA; EMARKETER, 2014). Embora a taxa de crescimento de usuários de telefones móveis tenha atingido um limite máximo nos países em desenvolvimento, com o aumento crescente de usuários na Ásia, Oriente Médio e África, em breve deve-se atingir o número de 4,5 bilhões de usuários. Um relatório da empresa de pesquisa de mercado eMarketer espera que o mercado crescente em países emergentes deve impulsionar o mercado (BYANSHUL SRIVASTAVA; EMARKETER, 2014). Visando o crescente aumento de usuários de smartphones em todo mundo aliado a tecnologia de rastreamento, vê-se então uma grande perspectiva para o desenvolvimento de um projeto que possa contribuir para melhoria do transporte público

## **1.1 Problema de pesquisa**

Um dos principais problemas presenciados pela população é a imprevisibilidade da chegada dos ônibus, essa imprevisibilidade pode ser devido a vários motivos. Isto afeta os usuários deste tipo de transporte, o que pode vir a contribuir para que esses usuários migrem para o transporte particular.

## **1.2 Objetivo**

O objetivo deste projeto é desenvolver um sistema que possibilite que o usuário do transporte público por ônibus possa acompanhar, via *smartphone*, o deslocamento do veículo, bem como estimar o horário de chegada em cada parada.

### **1.2.1 Objetivos específicos**

Os objetivos específicos deste trabalho consistem nos seguintes itens:

- I. Programar um sistema embarcado que utiliza módulos GPS/GSM e microcontrolador, capaz de obter a localização do ônibus e posteriormente envia-la para um servidor;
- II. Desenvolver um sistema Web capaz de armazenar e fornecer a localização de um ônibus em tempo real;
- III. Desenvolver um aplicativo para *smartphone* capaz de mostrar em um mapa a localização real de um ônibus;
- IV. Realizar um teste do protótipo na linha de ônibus “Damé x Malafaia”, no município de Bagé-RS.

### 1.3 Trabalhos correlatos

Sistemas de rastreamento similares ao proposto neste projeto já foram desenvolvidos por outros grupos de pesquisa, para diferentes finalidades, como rastreamento de veículos, antirroubo e gestão de monitoramento inteligente. A seguir apresenta-se brevemente dois sistemas de rastreamento que utilizam tecnologia GPS.

Maurya, Singh e Jain (2012) propuseram o desenvolvimento de um sistema de rastreamento de veículos utilizando tecnologias GPS e GSM para o sistema antirroubo de veículo. O sistema antirroubo proposto por Maurya, Singh e Jain (2012) tem o dispositivo fixado no veículo que envia os dados de localização para um servidor, posteriormente é possível analisar o status do veículo através de um *smartphone*.

Dinkar, Shaikh e Significance (2011) desenvolveram um sistema que utiliza GPS e *General Packet Radio Service* (GPRS) para rastrear veículos baseado na plataforma Web. O sistema de rastreamento Web desenvolvido por Dinkar, Shaikh e Significance (2011) utilizou diferentes tecnologias como Hypertext Preprocessor (PHP), Java Script, XML, MySQL e o *Application Program Interface* (API) do Google Maps para recuperar e exibir a posição do veículo no mapa. As tecnologias aplicadas por Dinkar, Shaikh e Significance (2011) são interessantes pois é possível fazer que todo o sistema seja capaz de armazenar, calcular e apresentar as informações relacionadas ao ônibus.

Algumas das tecnologias utilizadas para implementar o aplicativo Web proposto neste trabalho foram diferentes das tecnologias apresentadas nos projetos

correlatados. Uma das diferenças foi a organização do módulo Web através de padrões de projeto para uma melhor organização do sistema programado. Como também, foram acoplados e programados módulos em *hardware* para a captura da posição do veículo e envio para um servidor.

#### **1.4 Estrutura do trabalho**

Este trabalho está estruturado em cinco capítulos. No capítulo dois, são abordados os conceitos gerais e a revisão bibliográfica, expõe-se as considerações teóricas importantes para desenvolvimento do trabalho. No capítulo três, apresenta-se a metodologia utilizada para o desenvolvimento do projeto. Já no capítulo quatro apresenta-se o modelo proposto com os resultados alcançados ao longo do desenvolvimento do trabalho. Por fim, no capítulo cinco apresenta-se as conclusões e trabalhos futuros.

## 2 CONCEITOS GERAIS E REVISÃO DE LITERATURA

Este capítulo apresenta a revisão bibliográfica realizada para o desenvolvimento do presente trabalho. Abordam-se os seguintes temas:

- a) infraestrutura do transporte no Brasil;
- b) conceitos relacionados às tecnologias de posicionamento geográfico e de telecomunicação móvel;
- c) descrição das ferramentas/linguagens para o desenvolvimento do projeto como estudos acerca da tecnologia Arduino e dos comandos *Attention* (AT).

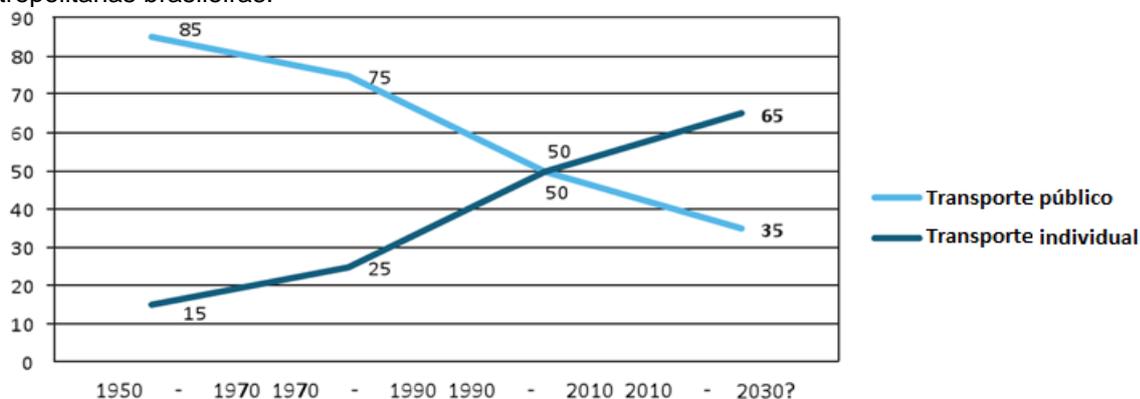
### 2.1 Infraestrutura de transporte no Brasil

Infraestrutura urbana pode ser considerada como um sistema técnico que tem por finalidade prestar serviços necessário para a sociedade (WITOLD ZMITROWICZ; NETO, 1997, p. 5).

Em algumas cidades, a demanda por infraestrutura de transporte é crescente. Nessas cidades, é necessário prever o acréscimo de demanda regional. A infraestrutura do transporte não é somente restringida pelos limites da cidade, devendo estar interligada a sistemas maiores (WITOLD ZMITROWICZ; NETO, 1997, p. 5).

A Figura 1 apresenta a estimativa da participação do transporte público no total de viagens das regiões metropolitanas brasileiras.

Figura 1 - Estimativa da participação do transporte público no total de viagens das regiões metropolitanas brasileiras.



Fonte: IPEA, 2011, p. 26.

No gráfico da figura 1, é possível verificar que, em 2009, a população utilizou mais o transporte individual (TI) do que o transporte público (TP) (IPEA, 2011).

## **2.2 Tecnologias relacionadas ao posicionamento geográfico e telecomunicação móvel**

Neste tópico são abordados conceitos relacionados às tecnologias de posicionamento geográfico e tecnologias de telecomunicação móvel. Essas tecnologias são fundamentais para o desenvolvimento deste projeto, já que é necessário obter a posição do veículo e enviá-la através da rede de comunicação móvel.

### **2.2.1 Sistema global de navegação por satélite (GNSS)**

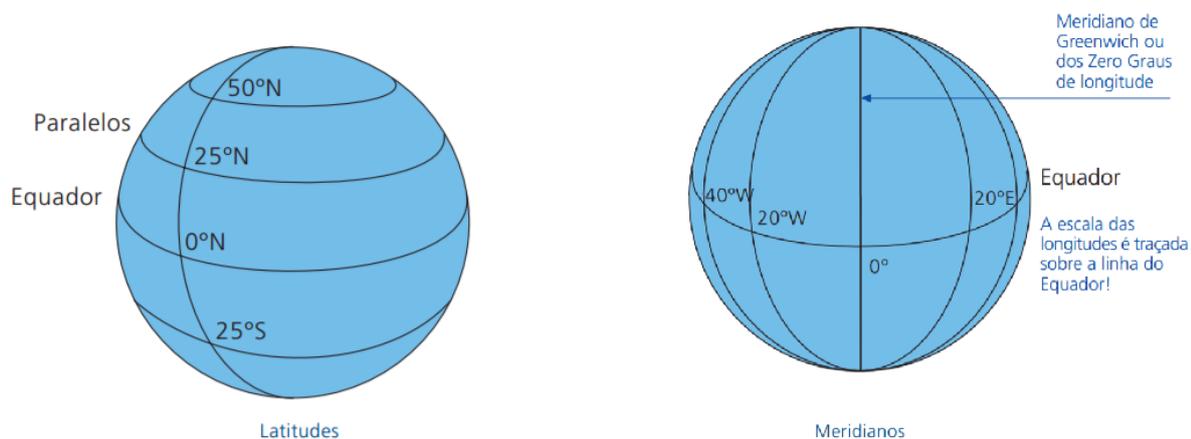
O Sistema de Navegação Global por Satélite (GNSS) são compostos por satélites artificiais com cobertura mundial. Hoje, o GNSS compreende duas grandes constelações que são o sistema de posicionamento global dos Estados Unidos, chamado *Global Positioning System* (GPS), e o *GLObal NAVigation Satellite System* (GLONASS), ou Sistema Global de Navegação por Satélite da Rússia. Há outras constelações, como a da China e a da Europa. Atualmente a maioria dos receptores GNSS utiliza apenas o GPS (HEGARTY, 2012, p. 1).

O GPS é um sistema de navegação por satélite operado pelos Estados Unidos. O primeiro satélite do sistema GPS foi lançado em 1978. A constelação é atualmente superpovoada com 31 satélites operacionais (HEGARTY, 2012, p. 1).

Para determinar as coordenadas de um ponto específico são necessários os cálculos da latitude e da longitude.

Os pontos mais a norte e mais a sul do equador são referenciados ao longo de linhas circulares paralelas desenhadas sobre a Terra, como demonstrada ao lado esquerdo da Figura 2. Essas linhas são os paralelos da terra e a sua posição é medida em graus: o equador é a linha de zero graus, o Polo Norte está a 90° N em relação ao equador, e o Polo Sul a 90° S em relação ao equador. A medida da posição norte-sul chama-se latitude (DILÃO, 1999).

Figura 2 - Representação da longitude e latitude, que formam as coordenadas geográficas da Terra.



Fonte: DILÃO, 1999.

Longitudes são marcadores por posicionamento geográfico que cruza o norte geográfico ao polo sul geográfico, passando pela linha do Equador. Eles se encontram em ambos os polos, e especifica a posição leste e oeste de uma localização. A metade de um círculo longitudinal é conhecido como um meridiano. Meridianos são perpendiculares a cada latitude. A Terra é dividido em 360°, assim como referência, o ponto do Greenwich tem longitude de 0° e o oposto da linha de Greenwich tem longitude de 180° (TIME AND DATE, 2014).

### **2.2.2 Global System for Mobile Communications (GSM) / General Packet Radio Service (GPRS)**

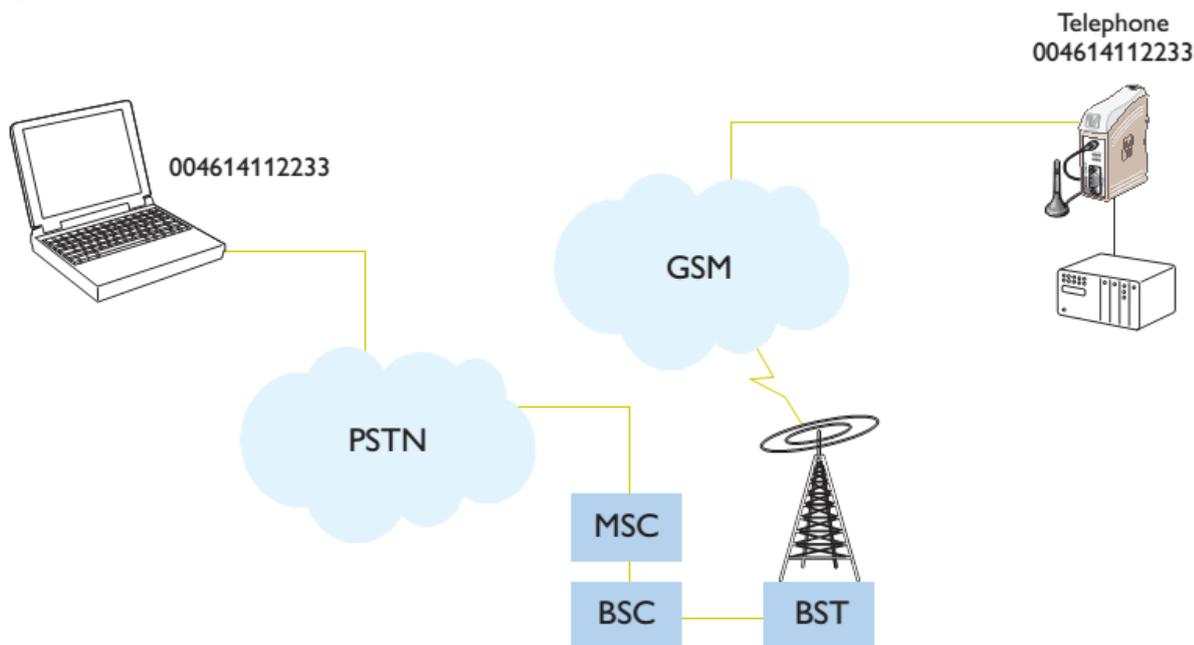
GSM é a sigla de Global System for Mobile Communications, ou Sistema Global para Comunicações Móveis é uma tecnologia utilizada em celulares e outros aparelhos móveis. Entre as tecnologias de telecomunicação sem fio, GSM é a tecnologia mais popular no mundo (WESTERMO, 2005).

GPRS é a sigla de General Packet Radio Services, ou Serviços Gerais de Pacote por Rádio. GPRS é uma extensão do GSM que permite a comutação de pacotes de dados, que pode ser usado para acessar a Internet. GPRS é uma tecnologia que tem o objetivo de aumentar as taxas de transferência de dados entre celulares, facilitando a comunicação e o acesso a rede de telecomunicação.

A possibilidade de utilizar o GSM e GPRS em comunicação de dados é uma alternativa para a comunicação a rádio. Aplicações sem fio são usadas principalmente para a comunicação onde não existem cabos. No entanto, a comunicação utilizando modem GSM ou GPRS requer certas condições básicas (WESTERMO, 2005, p. 93).

O modem GSM deve estar ligado à uma rede GSM. A conexão é feita através de *Mobile Switching Center* (MSC) e a *Base Station Controller* (BSC) saindo de uma linha *Public Switched Telephone Network* (PSTN). Como a ligação GSM é feita através de uma rede de circuito comutado, o circuito está constantemente conectando até que as linhas sejam desconectadas como é mostrada na Figura 3 (WESTERMO, 2005, p. 93).

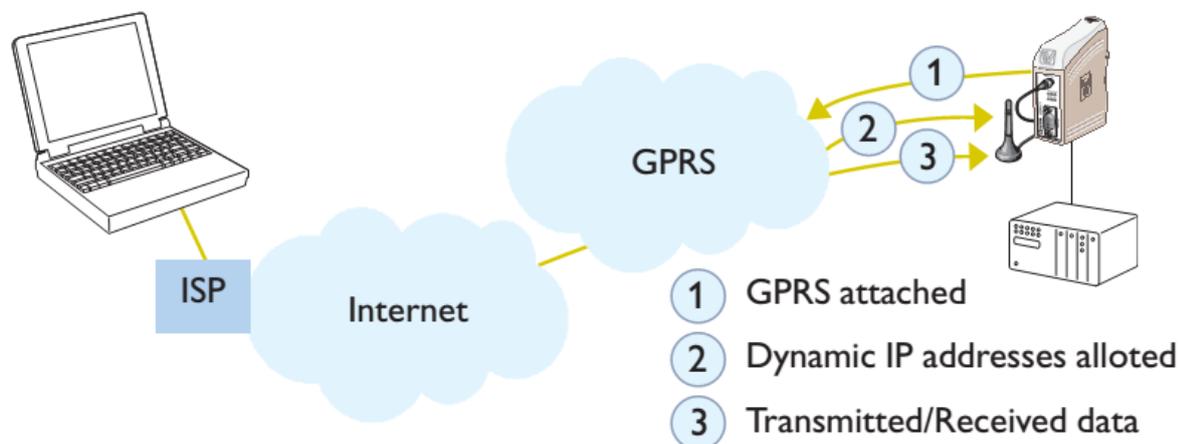
Figura 3 - Rede GSM.



Fonte WESTERMO, 2005, p. 93.

O GPRS é baseado em comunicação IP, o dispositivo ligado deve obter um endereço IP antes que uma conexão possa ser estabelecida. Na Figura 4 tem-se a representação de como é feita a conexão GPRS (WESTERMO, 2005, p. 94).

Figura 4 - Rede GPRS.



Fonte WESTERMO, 2005, p. 93.

## 2.3 Tecnologias relacionadas ao desenvolvimento do sistema

Nesta seção são abordadas as tecnologias utilizadas no desenvolvimento do projeto. São descritas as tecnologias e ferramentas como *framework* Python/Django utilizadas para o desenvolvimento do serviço Web, linguagens de programação, banco de dados, sistema do aplicativo e o *hardware*.

### 2.3.1 Arquitetura orientada a serviço

O sucesso da Arquitetura Orientada a Serviço (SOA) é baseado em três fatores fundamentais: arquitetura, organização, e as experiências do mundo real. A arquitetura de Tecnologia da Informação (TI) é o facilitador do uso do SOA. A adoção bem-sucedida do SOA acelera uma empresa, reduzindo a distância entre a estratégia e o processo, e apoiando sistemas de TI. A arquitetura de TI da organização empresarial é mutuamente dependente. Finalmente, a experiência em particular em infraestrutura de TI (bem sucedida ou e mal sucedida) influencia e valida muitos dos conceitos fundamentais do SOA (DIRK KRAFZIG, KARL BANKE, 2004, p. 20).

De acordo com DIRK KRAFZIG, KARL BANKE (2004, p. 29) para melhorar a agilidade e eficiência de uma arquitetura de software, este deve fornecer características particulares dos itens:

- I. Simplicidade: a arquitetura corporativa deve ser simples, a fim de permitir a comunicação eficiente entre as pessoas chave;
- II. Flexibilidade e facilidade de manutenção: cada sistema empresarial está sujeito a mudanças. Ele deve ser continuamente adaptado às novas exigências devido à necessidade do mercado em evolução, mudanças legais, ou reorganizações empresariais;
- III. Reutilização: essa característica tem sido um dos principais objetivos da engenharia de software ao longo das décadas.

### **2.3.2 Web services**

No início do novo século, o termo "serviços Web" tornou-se extremamente popular, embora tenha sido muitas vezes usado para se referir a diferentes conceitos da computação. O termo "*Web service*" é usado para se referir a módulos de aplicativos que torna acessíveis a outros aplicativos através de protocolos baseados em *eXtensible Markup Language* (XML) (DIRK KRAFZIG, KARL BANKE, 2004, p. 38).

### **2.3.3 Web service REST**

REST é um termo geralmente utilizado para descrever uma interface que transmita dados de um domínio específico sobre *HyperText Transfer Protocol* (HTTP) sem uma camada adicional de mensagem como SOAP ou via *session/cookies*. Sistemas que seguem os princípios REST são conhecidos também como RESTful (MORO; DORNELES; REBONATTO, 2011, p. 40).

De acordo com Moro, Dorneles e Rebonatto (2011, p. 40) as principais funcionalidades do serviço REST são:

- I. *Cache*: com a utilização de *cache*, os dados de uma resposta oriundos de uma requisição ao servidor são demarcados como *cacheable* (possível de ser utilizada no futuro) ou *noncacheable* (passíveis ou não

- de utilização da cache). Uma resposta demarcada como *cacheable*, será reutilizada como resposta para futuras requisições equivalentes;
- II. Cliente-Servidor: essa característica é utilizada em aplicações *Web*. Um servidor possui um conjunto de serviços disponíveis, que escuta requisições a estes serviços. Um cliente é aquele que deseja um serviço disponível no servidor. Sendo que, este pode tanto rejeitar como executar o serviço solicitado, e retornar uma resposta ao cliente;
  - III. *Web Services* REST utiliza o protocolo HTTP para trabalhar. Sendo que este provê uma interface uniforme com no mínimo quatro métodos básicos para as quatro operações mais comuns: GET, PUT, POST e DELETE;
  - IV. *Stateless* (sem estado): a comunicação entre cliente-servidor deve ser feita sem o armazenamento de qualquer tipo de estado no servidor. Assim, estados de sessão devem ser totalmente mantidos no cliente;
  - V. Multicamada: com o objetivo de aprimorar a escalabilidade da internet, foi adicionado ao REST a característica de divisão em camadas. Sendo que as camadas servem para separar diferentes unidades de funcionalidade;
  - VI. *Code-On-Demand*: essa é uma característica opcional do REST que possibilita aos clientes a funcionalidade de fazer *download* e executar diretamente código no lado cliente, além de permitir *debugar* sem paralisar o serviço;
  - VII. Interface Uniforme: o que diferencia a característica REST de outros estilos é a sua ênfase em uma interface uniforme entre os componentes (cliente e servidor). O objetivo é de obter uma interface uniforme, o REST define quatro requisitos de interface:
    - Identificação de recursos ou *Uniform Resource Identifier* (URI);
    - Manipulação de recursos por meio de representações;
    - Mensagens auto descritivas;
    - Hiperlinks como mecanismo de estado da aplicação.

Como a filosofia REST tornou-se popular, novas estruturas estão surgindo para tornar fácil projetar um serviço RESTful. Estes são os *frameworks* mais

populares: Ruby on Rails, Restlet (para Java) e Django (para Python) (RICHARDSON; RUBY, 2008, p. 339).

#### **2.3.4 Python/Django**

Django é um *framework* que facilita o desenvolvimento de aplicações Web e serviços Web em Python (RICHARDSON; RUBY, 2008, p. 355). Devido a essa facilidade de desenvolvimento e a praticidade em utilizá-lo, foi escolhido o Django como o *framework* para desenvolvimento do serviço Web.

#### **2.3.5 JSON**

A maioria dos serviços Web utilizam os dados no tipo XML, mas o uso serializado de dados do tipo *JavaScript Object Notation* (JSON) está crescendo. JSON é geralmente consumido por aplicações Ajax, e este tipo de dados é uma maneira simples e independente de formatação (RICHARDSON; RUBY, 2008, p. 45).

#### **2.3.6 Banco de dados**

De acordo com Elmasri e Navathe (2005), um sistema de banco de dados é composto por um Banco de Dados (BD) e um Sistema Gerenciador de Banco de Dados (SGBD).

Rigaux et al., (2002) define banco de dados como uma grande coleção de dados inter-relacionados armazenados dentro de um ambiente computacional. Um Sistema Gerenciador de Banco de Dados (SGBD) pode ser definido como um conjunto de softwares que gerenciam a estrutura do banco de dados e controla o acesso aos dados armazenados no banco de dados. O SGBD facilita o processo de definição (especificação dos tipos de dados, estruturas e limitações), construção, manipulação, consulta e atualização do banco de dados.

#### **2.3.7 JavaScript**

JavaScript é uma linguagem de programação interpretada e orientada a objetos. Sintaticamente, o núcleo da linguagem JavaScript assemelha-se com as

linguagens C, C++ e Java, como construções de programação condicional *if*, o *loop while*, e o operador *&&*. JavaScript é uma linguagem fracamente tipada, o que significa que as variáveis não precisam serem especificadas. O núcleo da linguagem JavaScript suporta os dados do tipo números, strings e valores booleanos como dados primitivos (FLANAGAN, 2006, p. 30).

JavaScript é comumente usada em navegadores Web. Nesse contexto, o núcleo da linguagem permite interação com o usuário, como exemplo controlar o navegador Web e alterar o conteúdo do documento que aparece na janela do navegador Web. É comumente chamado de JavaScript do lado cliente para enfatizar que os scripts são executados pelo computador cliente ao invés de pelo servidor Web. O núcleo da linguagem JavaScript e seus tipos de dados embutidos são objetos que segue normas internacionais.(FLANAGAN, 2006, p. 30).

### 2.3.8 Leaflet

Leaflet é uma biblioteca escrita em JavaScript de código aberto para criação de mapas interativos compatíveis com *smartphone*. Ao longo dos últimos anos, JavaScript tornou se a linguagem de mapeamento on-line. Em 2013, a biblioteca Leaflet foi o centro das atenções na conferência JS.geo. Com essa biblioteca é possível criar mapas simples com menos de três linhas de códigos em JavaScript, ou pode-se criar complexos e interativos mapas com centenas de linhas de código (III, 2014, p. 44).

### 2.3.9 Bootstrap

O Bootstrap é uma estrutura de interface gráfica em código aberto para projetar sites e aplicativos web. A estrutura utiliza o sistema de *design* responsivo que é um método para tirar proveito de toda tela do dispositivo. Por exemplo, *tablets* podem obter um *layout* otimizado, aproveitando o formato da tela retrato ou paisagem. Para atingir diferentes larguras, Bootstrap utiliza consultas de mídia *Cascading Style Sheets* (CSS) e JavaScript para medir a largura da janela do navegador e em seguida, usando estruturas condicionais, muda partes das folhas de estilo que são carregados. Usando a largura da janela do navegador, Bootstrap pode

então otimizar o conteúdo usando uma combinação de alturas e larguras (SPURLOCK, 2014, p. 7).

### 2.3.10 Android

O Android é um sistema operacional voltado para hardware móvel, como telefones e outros dispositivos de computação limitados, como netbooks e tablet computadores (JORDAN; GREYLING, 2011, p. 2).

O conceito e a plataforma foram uma ideia da Android Inc., uma pequena empresa startup do Palo Alto, Califórnia, que foi adquirida pela Google em 2005. Seu objetivo era criar um sistema operacional pequeno, estável, flexível e facilmente atualizado para aparelhos que seriam altamente atraente para os fabricantes de aparelhos e operadoras de telefonia (JORDAN; GREYLING, 2011, p. 2).

As versões Android 1.x até 2.x são destinadas principalmente a smartphones, considerando que a versão Android 3.x foi a primeira plataforma operacional projetada especificamente com o suporte em alta qualidade para computadores e *tablets*. A plataforma Android foi originalmente lançado em novembro de 2007. A inauguração coincidiu com o anúncio da formação do Open Handset Alliance, um grupo de empresas que compartilham o objetivo de promover padrões abertos para plataformas de *smartphone*, como o Android. Em outubro de 2008, o Android foi lançado sob a licença de código aberto Apache 2.0 (JORDAN; GREYLING, 2011, p. 2).

A plataforma do Android é descrita por JORDAN, GREYLING ( 2011, p. 21) a seguir:

- I. As camadas de abstração de hardware (HALs) são projetadas para proteger a plataforma operacional da grande variedade de hardware existente no mercado. Os drivers do dispositivo e o do kernel Linux formam a estrutura HAL na plataforma Android. Ela cuida de operações do núcleo do sistema, como controle de hardware, gestão de processos, rede e da gestão de memória;
- II. As bibliotecas *Runtime* em C/C++ executam diretamente no kernel do HAL e fornecem serviços básicos para as aplicações e para o Android. Esses serviços incluem suporte a gráficos (2D, 3D, SGL, OpenGL), gerenciamento de exibição, reprodução de vídeo e mídia de áudio,

armazenamento de dados estruturados (SQLite), libc, suporte embutido Web navegador via WebKit, e Secure Sockets Layer (SSL) para redes seguras;

- III. Sistema *Runtime* no Android. O Dalvik virtual machine (DVM) é muito importante para aplicações *Runtime*. Cada instância DVM está hospedada no seu próprio processo do sistema do kernel Linux;
- IV. O aplicativo *Framework* do Android suporta o desenvolvimento de aplicações. Esse framework possibilita a criação de aplicações através das classes em Java disponíveis, além de englobar uma ampla gama de gerenciadores de recursos de *hardware*. Esses recursos incluem a interface do usuário, reconhecimento de local, detecção de aceleração, câmera, telefonia, as notificações do sistema, etc;
- V. A camada de aplicações é onde geralmente é entregue o conjunto de funcionalidade dos aplicativos. A concepção aberta da plataforma permite aos desenvolvedores reutilizar, estender ou substituir completamente os principais aplicativos fornecidos pelo Android com suas próprias criações. Todos os aplicativos do Android são tratados igualmente pelo sistema. As aplicações Android são geralmente escritos em Java, e as bibliotecas de código nativo podem ser carregadas e chamadas via Java Native Interface (JNI), se necessário.

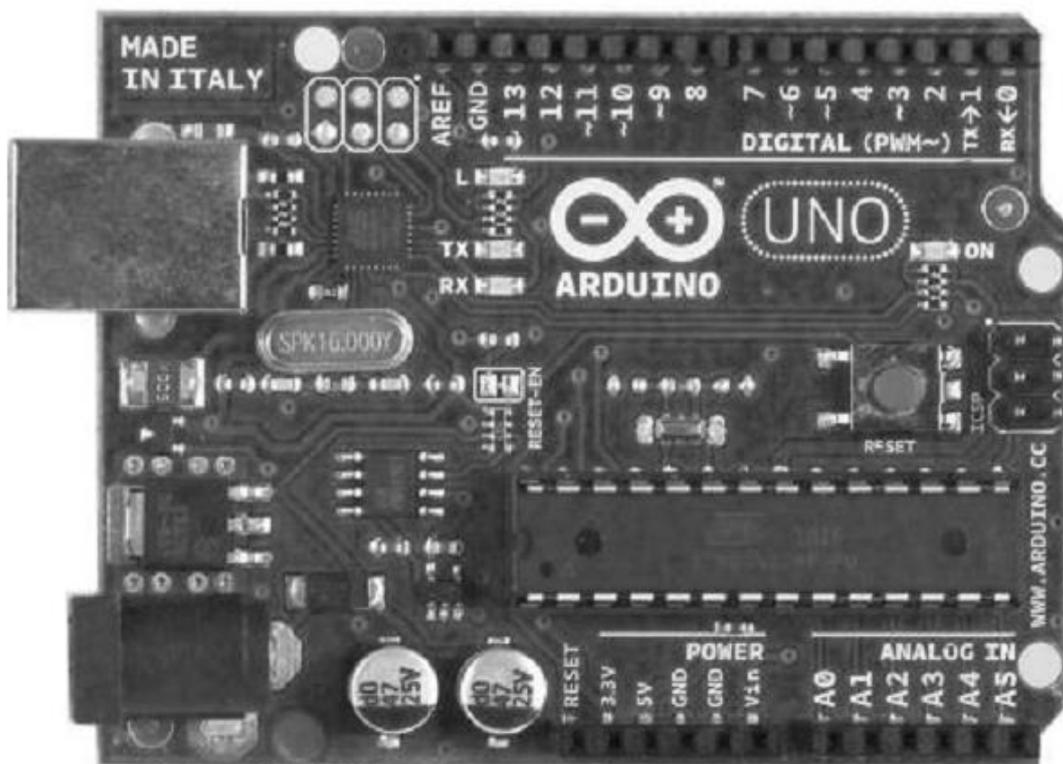
### 2.3.11 Arduino

O ambiente Arduino foi projetado para iniciantes que não têm nenhuma experiência com software ou eletrônica. Com o Arduino, pode-se construir projetos que podem responder a um controle, som, toque e movimento. Arduino é usado para criar uma incrível variedade de projetos, incluindo instrumentos musicais, robôs, esculturas de luz, jogos, e até mesmo roupas interativas (MARGOLIS, 2011, p. 1).

A placa Arduino exibida na Figura 5 é onde o código é executado. A placa só pode controlar e responder a energia elétrica aplicada na placa, assim os componentes específicos estão ligados a ele para permitir a interação com o mundo real. Esses componentes podem ser sensores, que convertem algum aspecto do mundo físico à energia elétrica, ou atuadores, que recebem energia elétrica da placa e então converte algo para o mundo real. Exemplos de sensores incluem chaves,

acelerômetros e sensores de distância por ultrassom. Atuadores são elementos como Light Emitting Diode (LEDs), alto-falantes, motores e displays (MARGOLIS, 2011, p. 2).

Figura 5 - Placa básica de um Arduino Uno.



Fonte: MARGOLIS, 2011, p. 3

O microcontrolador do Arduino é um pequeno computador em um único circuito integrado que contém um núcleo de processador, memória de dados, conversor A/D e periféricos de entrada/saída. Microcontroladores são muito menores e simplificados de que computadores, de modo que eles podem incluir todas as funções requeridas em um único chip (MARGOLIS, 2011).

### 2.3.12 Comando AT

Os comandos do Tipo Atenção (AT) são usados para controlar dispositivos, na maioria modems. Estes comandos são derivados dos comandos Hayes que foram utilizados pelos modems inteligentes Hayes. Os comandos Hayes começam com um AT para indicar a atenção do modem. Os modems dial-up e wireless,

dispositivos que envolvem comunicação entre máquina a máquina, utilizam comandos ATs para interagir com computadores. Estes incluem o comando Hayes definido como um subconjunto, juntamente com outros comandos estendido AT (VERMA, 2013).

De acordo com VERMA (2013), os comandos ATs de uma rede GSM/GPRS ou de telefone celular pode ser usado para acessar as seguintes informações e serviços:

- I. Informação e configuração referente ao *smartphone* ou um modem de um cartão SIM;
- II. Serviços de SMS;
- III. Serviços de MMS;
- IV. Serviços de fax;
- V. Dados e voz através de chamada pela rede móvel.

Os comandos Hayes são chamados comandos básicos e os comandos específicos a uma rede GSM são chamados comandos estendido AT.

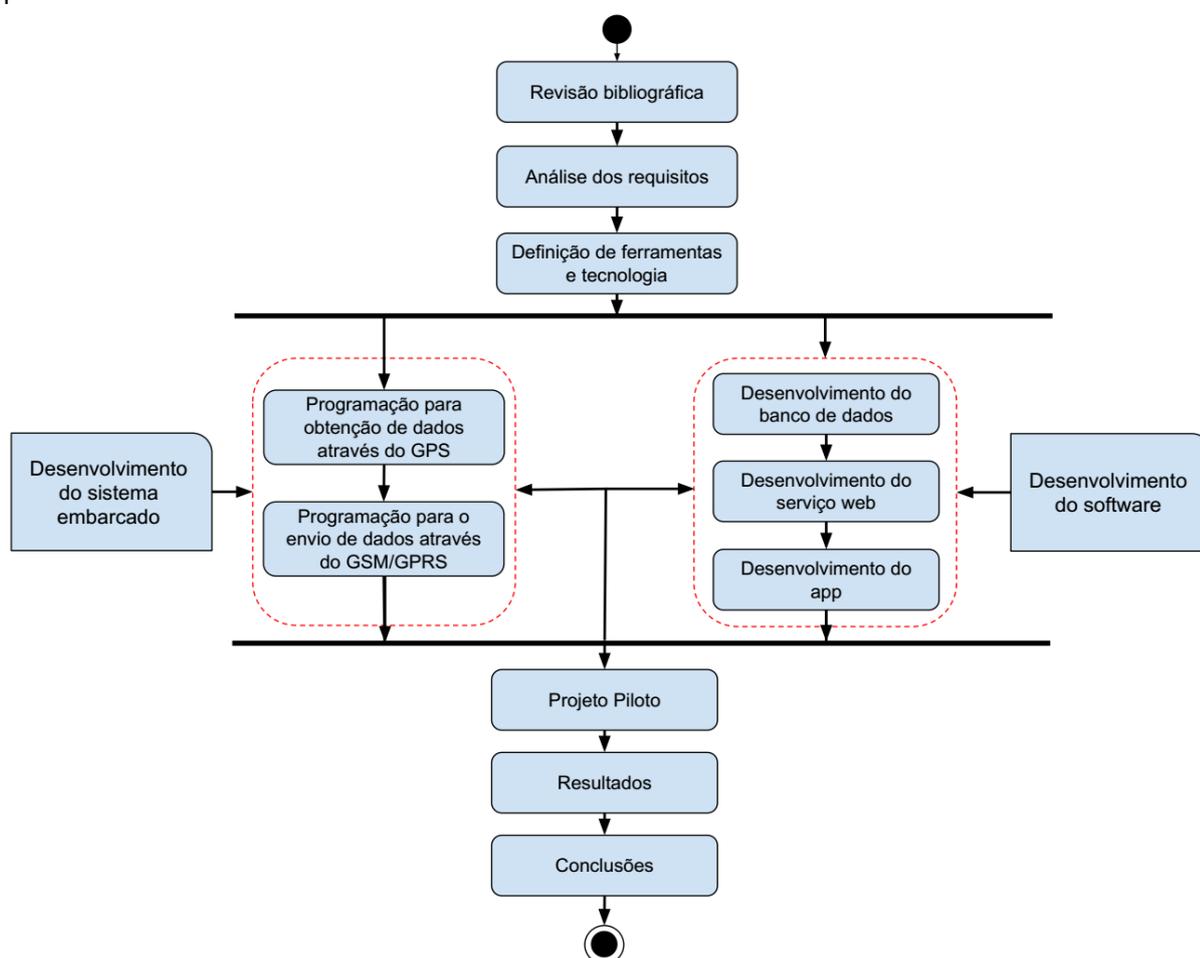
Os tipos de comando AT são:

- I. Comandos de teste - usado para verificar se um comando é suportado ou não pelo modem;
- II. Comando leitura - usado para obter as configurações do telefone;
- III. Comandos de envio - usado para modificar as configurações do telefone ou modem de Internet móvel para uma operação;
- IV. Comandos de execução - usados para realizar uma operação.

### 3 METODOLOGIA PESQUISA

A sequência metodológica utilizada para a modelagem do sistema de rastreamento de transporte público através de um sistema embarcado, sistema Web e um aplicativo para *smartphone* é apresentada no diagrama de atividade em Linguagem Unificada de Modelagem (UML) na Figura 6. Esta pesquisa pode ser definida, quanto a natureza do objetivo como uma pesquisa quantitativa e experimental (SILVEIRA; CÓRDOVA, 2009).

Figura 6 - Sequência metodológica para a modelagem do sistema de rastreamento de transporte público.

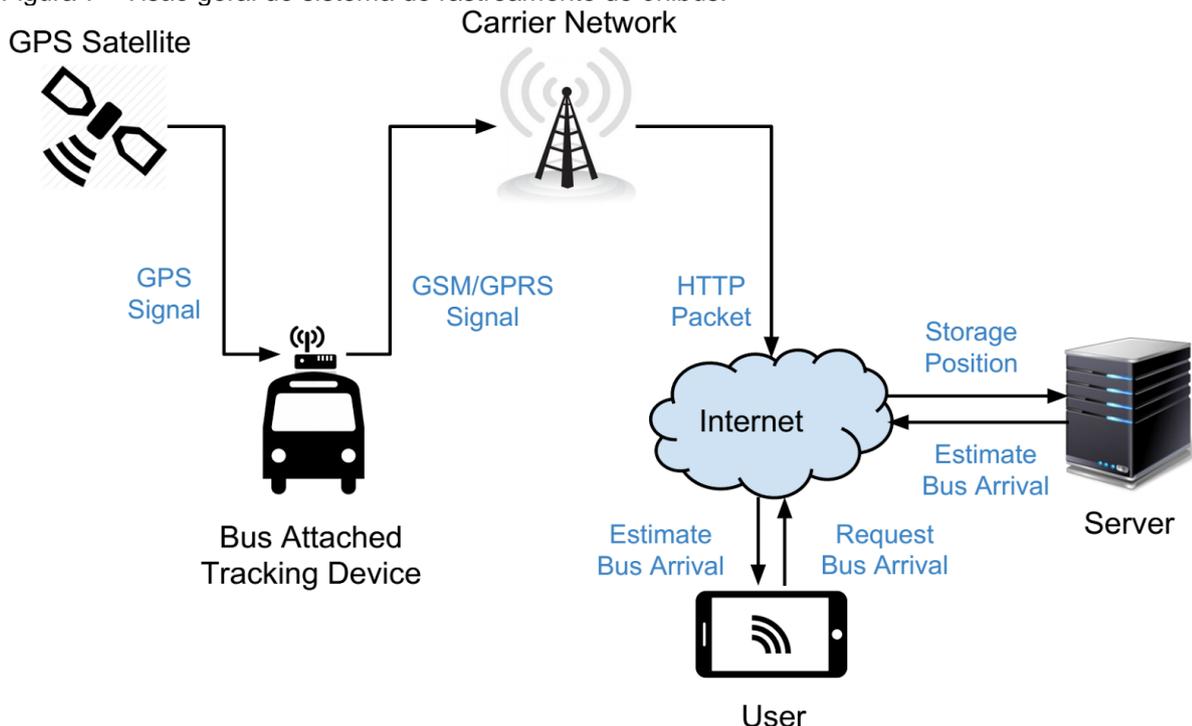


Fonte: Próprio Autor, 2016.

#### 3.1 Sistema de Rastreamento

A partir dos requisitos do projeto e com a disponibilidade da tecnologia, foi elaborado um gráfico na Figura 7 que descreve a visão geral do sistema de rastreamento.

Figura 7 - Visão geral do sistema de rastreamento de ônibus.



Fonte: Próprio Autor, 2016.

Como representado na Figura 7, o fluxo de dados começa com o satélite GPS que necessita de pelo menos três satélites para fazer a triangulação da posição exata do dispositivo. Com os satélites triangulados é possível obter a posição com o uso de um receptor GPS. Quando o receptor GPS capta a sua própria posição, um microcontrolador comunica com o receptor. Por conseguinte, o microcontrolador lê essa posição e cria uma requisição HTTP para enviar a posição obtida ao servidor. O envio da posição utiliza a rede de celular GSM/GPRS. Para processar esse pedido, uma página no servidor é invocada com a posição do objeto e outros parâmetros como hora e quantidade de satélites. Em seguida, uma coleção de *scripts* irá processar e armazenar a posição do objeto. Após a posição do objeto ser armazenada no banco de dados, o passageiro poderá solicitar a hora de chegada do veículo usando seu próprio *smartphone*. Quando solicitado a hora da chegada de um ônibus pelo passageiro, é invocado uma página Web no servidor para restaurar e estimar o tempo de chegada do mesmo na parada solicitada.

Com os requisitos bem definidos o projeto foi modelado e por fim feito o teste com o projeto piloto. Com os resultados do projeto piloto, foi ajustado o modelo previamente modelado até atingir o objetivo final e possa chegar ao produto ideal.

## 3.2 Análise dos requisitos

É um processo de obtenção dos requisitos de sistema através da observação de sistemas existentes, discussões com potenciais usuários e compradores, análise de tarefas, etc. Isso pode envolver o desenvolvimento de um ou mais modelos de sistema e protótipos. A análise dos requisitos ajuda o analista a compreender o sistema a ser especificado (SOMMERVILLE, 2007, p. 50).

### 3.2.1 Requisitos funcionais

Conforme Sommerville (2007, p. 79) os requisitos de um sistema são descrições das funcionalidades do sistema e suas restrições operacionais. Os requisitos são necessidades dos clientes que ajudam a resolver algum problema como por exemplo controlar um dispositivo, enviar um pedido ou encontrar informações. O processo de descobrir, analisar, documentar e verificar esses serviços e restrições é chamado de engenharia de requisitos. Em conformidade com Sommerville (2007, p. 81) os requisitos funcionais de um sistema descrevem o que o sistema deve fazer. Os requisitos dependem de diversos fatores alguns desses fatores podem depender do tipo de usuário que irá utilizar o sistema ou o tipo de software que está sendo desenvolvido e a abordagem da organização ao redigir os requisitos.

Para a melhor classificação das prioridades, foram criadas três tipos de prioridades que estão descritas na Tabela 1.

Tabela 1 - Tabela de prioridade de requisitos.

E – Essencial	Este tipo de requisito deve ser implementado para que o sistema funcione. A condição é essencial para o sucesso do projeto.
I – Importante	Sua não implementação afeta a satisfação do usuário porém caso não seja implementado, o sistema entra em funcionamento do mesmo modo.
D – Desejável	Este tipo de requisito não compromete o funcionamento do sistema.

Fonte: mobilab, 2015, p. 8.

Foram definidos a porcentagem e a probabilidade de risco durante a implementação. Pode ocorrer eventos indesejáveis, como o não cumprimento da programação, orçamento de custo ultrapassado e probabilidade de enfrentar sérias dificuldades durante a implementação do requisito devido à falta de experiência no domínio e/ou nas tecnologias necessárias (MOBILAB, 2015, p. 8). A porcentagem de risco é detalhada na Tabela 2.

Tabela 2 - Tabela de Risco de Requisitos.

A – Alto	Risco > 50%
M – Médio	Risco entre 10% e 50%
B – Baixo	Risco < 10%

Fonte: mobilab, 2015, p. 8.

### 3.2.2 Requisitos não funcionais

Os requisitos não funcionais são aqueles que não estão diretamente relacionados com às funções específicas fornecidas pelo sistema. Eles podem estar relacionados com às propriedades emergentes do sistema, como por exemplo confiabilidade, tempo de resposta e espaço de armazenamento (SOMMERVILLE, 2007, p. 82).

Na Figura 8 estão descritos os tipos de requisitos não funcionais que se dividem em requisitos de produto, requisitos organizacionais e requisitos externos. Esses requisitos estão fortemente ligados com as propriedades emergentes do sistema.

Figura 8 - Tipos de requisitos não-funcionais.



Fonte:(SOMMERVILLE, 2007, p. 82).

Pode se dizer que um caso de uso se refere a um contrato que descreve o comportamento do sistema sob várias condições. Em que o sistema responde a uma solicitação de acordo com os atores que têm influências externas no projeto (PRESSMAN, 2006, p. 129).

O diagrama de sequência em UML indica como os eventos provocam transições de objeto para objeto. Ele representa os eventos que causam o fluxo de um objeto para outro como sequência ou pode ser considerado também uma versão abreviada do caso de uso (PRESSMAN, 2006, p. 179).

Em UML, é possível utilizar diagramas de atividades para modelar a sequência das ações que ocorrem em um sistema ou é utilizado para descrever o que acontece em um determinado fluxo de trabalho de um processo de negócio.

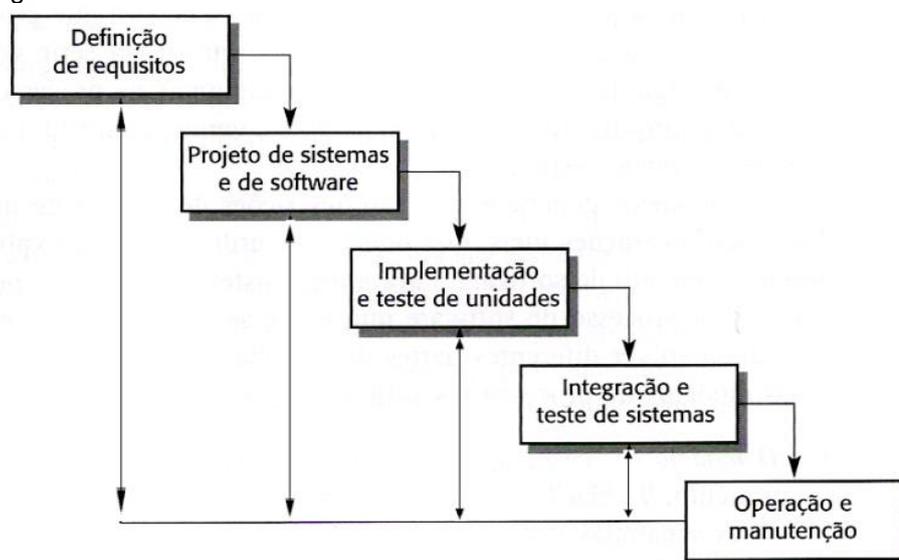
### 3.2.3 Projeto de software

Um modelo de desenvolvimento de software é uma representação abstrata de um processo de software. Cada modelo tem suas diferentes perspectivas, por isso, fornece somente informações parciais sobre este processo. Um modelo é uma abstração do processo que pode ser usada para explicar diferentes abordagens para o desenvolvimento de software (SOMMERVILLE, 2007, p. 43).

Neste trabalho foi utilizado o modelo em cascata visto que os requisitos são bem definidos. O diagrama do modelo em cascata é representado na Figura 9.

O modelo em cascata é um modelo que é dividido em fases, onde a fase seguinte não deve começar antes que a fase anterior tenha terminado. O modelo em cascata deve ser usado apenas quando os requisitos forem bem compreendidos e houver pouca probabilidade de mudanças radicais durante o desenvolvimento do sistema. Todavia, o modelo em cascata é o tipo de modelo de processo usado em outros projetos de engenharia (SOMMERVILLE, 2007, p. 45). Já Pressman (2006, p. 39) diz que um modelo em cascata sugere uma abordagem sistemática e sequencial para o desenvolvimento de softwares que começa com a especificação dos requisitos e terminado na manutenção progressiva do software acabado.

Figura 9 - Diagrama do modelo em cascata.



Fonte: SOMMERVILLE, 2007, p. 44.

### 3.3 Definição de ferramentas e tecnologias

As ferramentas e linguagens que foram utilizadas neste projeto foram definidas por diferentes aspectos como experiência do programador, praticidade de programação dentre outros.

No desenvolvimento do hardware foi utilizado um dispositivo que integra um módulo GPS que permite determinar a localização de um veículo, pessoa ou outros objetos, assim possibilitando registrar a posição do mesmo em intervalos regulares. Os dados de localização obtidos foram transmitidos para um servidor de banco de dados por meio das tecnologias GSM/GPRS. Posteriormente, as informações das posições foram plotadas em um mapa para a visualização do objeto em movimento.

Neste projeto foi utilizado o Arduino Uno R3. O Arduino foi responsável por configurar o dispositivo GPS/GSM/GPRS, ler a localização GPS e enviar a localização para o servidor usando as tecnologias GSM/GPRS. O Arduino utilizado constituiu em um kit que tem integrado um microcontrolador ATmega328P, possibilitando a comunicação com o dispositivo GPS/GSM/GPRS.

O dispositivo SIM808 utilizado, é um sistema completo Quad-Band GSM/GPRS que combina a tecnologia GPS para a navegação por satélite. O dispositivo hardware depende de uma fonte de tensão que fornece pelo menos 2 amperes para alimentar o Arduino Uno R3 e o SIM808.

O software utilizado para escrever, compilar e transferir o código para o microcontrolador foi o programa de código aberto Arduino Software *Integrated Development Environment (IDE)*.

### 3.4 Desenvolvimento do sistema

Para o desenvolvimento da plataforma Web, foram utilizadas as seguintes linguagens de programação e tecnologias Python/Django, JavaScript, HTML, CSS, Bootstrap e Leaflet. Essas tecnologias são amplamente utilizadas, por oferecerem rápida implementação e fácil manutenção. Como exemplo o *framework* Django que é utilizado para desenvolvimento ágil baseado na linguagem Python, além de utilizar padrões da programação orientada a objetos e outros padrões de projeto.

Para o desenvolvimento do aplicativo foi dividido em duas partes: *online* (com internet) e *off-line* (sem internet). No qual o modo *online* tem-se acesso à localização em tempo real do ônibus e à previsibilidade da chegada a uma parada específica. No modo *off-line* é possível visualizar o itinerário pré-determinado das linhas de ônibus.

No desenvolvimento do sistema em hardware, foi acoplado no Arduino um *shield* SIM808, esse dispositivo é capaz de obter a posição do veículo através do GPS e então enviá-la, via GSM/GPRS, para o servidor de banco de dados. A inteligência do hardware foi implementada no Arduino, pois todo o processamento é realizado diretamente nele antes de enviar os dados da localização para o servidor.

## 4 IMPLEMENTAÇÃO DO PROJETO

Neste tópico apresenta-se as etapas da implementação do sistema. Inicialmente apresenta-se a modelagem do sistema, em seguida são descritas as tecnologias utilizadas. Por fim o desenvolvimento do sistema e o projeto piloto são detalhados.

### 4.1 Modelagem

Na modelagem foram criados os diagramas de caso de uso, sequência, atividades e o Entidade Relacionamento (ER) que serão descritos neste tópico.

#### 4.1.1 Requisitos funcionais

Na Tabela 3 apresentam-se os requisitos funcionais levantados a partir das necessidades para o desenvolvimento do sistema de rastreamento de transporte público.

Tabela 3 - Requisitos funcionais.

ID	REQUISITOS FUNCIONAIS	PRIORIDADE	RISCO
RF01	Obtenção da posição do ônibus em uma determinada frequência.	I	A
RF02	Envio dos dados da posição do ônibus em uma determinada frequência.	I	A
RF03	Apresentação das informações em uma interface de mapas georreferenciados.	E	M
RF04	Exibição das rotas do plano de operação dos veículos fretados em uma interface de mapas.	I	M
RF05	Obtenção de informações sobre o ônibus e parada de ônibus, como exemplo se a parada de ônibus é coberta, se o ônibus é adaptado para cadeirantes dentre outros.	I	M
RF06	Verificação do tempo de chegada dos ônibus.	I	M

RF07	Opção de pesquisa de pontos de ônibus.	D	M
RF08	Função no aplicativo mobile (Android) para obter informação sobre o itinerário em modo off-line.	D	M

Fonte: Próprio Autor, 2016.

#### 4.1.2 Requisitos não funcionais

Na Tabela 4 apresentam-se os requisitos não funcionais levantados.

Tabela 4 - Requisitos não funcionais.

ID	REQUISITOS NÃO FUNCIONAIS	DESCRIÇÃO	PRIORIDADE	RISCO
RFN01	Acesso por diversos dispositivos.	Através de diversos dispositivos será possível acessar o sistema de rastreamento de ônibus. Esse é um requisito de usabilidade.	I	M
RFN02	Design responsivo.	O sistema Web deve se adaptar a tela de acordo com o tamanho da mesma. Esse é um requisito de usabilidade.	I	M
RFN03	O sistema deve ter uma interface simples e objetiva.	A partir da utilização de elementos gráfico como mapas e ícones para assim tornar um sistema mais interativo. Esse requisito é um requisito de usabilidade.	I	M
RFN04	Eficiência na obtenção dos dados.	Com uma determinada frequência o mapa deve atualizar os elementos, e o rastreador deve enviar a posição do veículo em uma	I	M

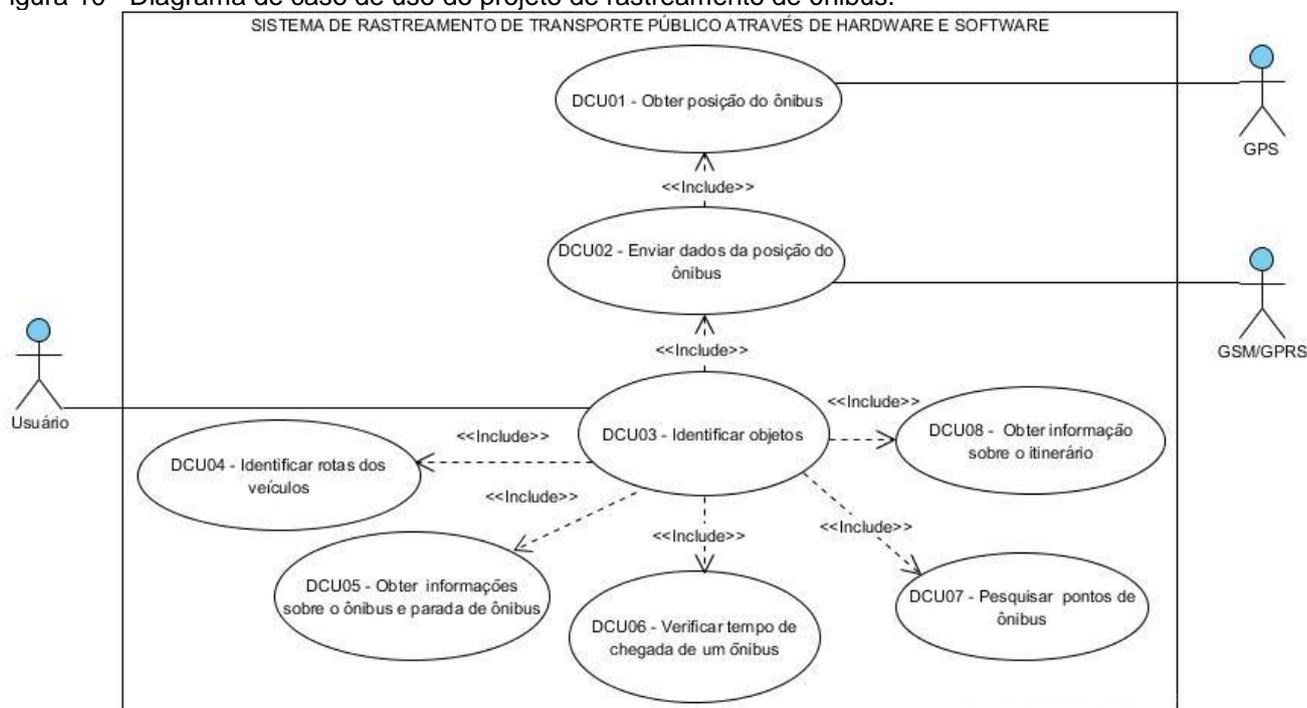
	determinada frequência. Esse requisito é um requisito de desempenho.	
--	--	--

Fonte: Próprio Autor, 2016.

### 4.1.3 Diagrama de caso de uso

O diagrama de casos de uso representa as funcionalidades do projeto e as suas interações com os atores. O diagrama de casos de uso deste projeto está descrito na Figura 10.

Figura 10 - Diagrama de caso de uso do projeto de rastreamento de ônibus.



Fonte: Próprio Autor, 2016.

As próximas tabelas descrevem os casos de uso deste projeto. Para obter a posição do veículo se faz necessário o uso do ator GPS, que é o responsável pela obtenção da posição do veículo que posteriormente outro ator possa utilizar essa informação como é mostrado na Tabela 5.

Tabela 5 - Descrição do caso de uso obter posição do ônibus.

Nome do caso de uso	DCU01 - Obter posição do ônibus.
Ator principal	GPS
Descrição	O modulo GPS obtém a posição do ônibus com uma determinada frequência.
Pré-condições	Os satélites devem estar triangulados com no mínimo de três satélites.
Pós-condições	Localização obtida e outras informações como altitude, hora, número de satélites.

Fonte: Próprio Autor, 2016.

O acesso à internet acontece através das tecnologias GSM/GPRS. Através dessas tecnologias é possível utilizar o protocolo HTTP, que tem a função de fazer requisição a uma página Web. Assim, a página Web registra no banco de dados a última posição do ônibus obtida pelo GPS, como é descrito na Tabela 6.

Tabela 6 - Descrição do caso de uso enviar dados da posição do ônibus.

Nome do caso de uso	DCU02 - Enviar dados da posição do ônibus.
Ator principal	GSM/GPRS
Descrição	O envio da localização do ônibus através das tecnologias GSM/GPRS para comunicar com o servidor Web.
Pré-condições	Obtenção da localização e utilização da rede GSM/GPRS.
Pós-condições	Registro da posição do ônibus armazenada na base de dados.

Fonte: Próprio Autor, 2016.

A Tabela 7 apresenta a descrição do caso de uso para identificar objetos. Foi utilizado o *framework* Leaflet que possibilita plotar em um mapa objetos para identificação dos pontos de parada e ônibus. Assim o usuário pode identificar cada objeto de uma maneira simples.

Tabela 7 - Descrição do caso de uso identificar objetos.

Nome do caso de uso	DCU03 - Identificar objetos.
Ator principal	Leaflet/Usuário
Descrição	Deve ser plotado no mapa informações como paradas de ônibus e a última localização do ônibus.
Pré-condições	São necessárias as posições das instancias de cada objeto.
Pós-condições	Os objetos plotados em distintos ícones.

Fonte: Próprio Autor, 2016.

A identificação das rotas é importante para obter uma melhor visualização do itinerário, suas especificações estão descritas na Tabela 8.

Tabela 8 - Descrição do caso de uso identificar rotas dos veículos.

Nome do caso de uso	DCU04 - Identificar rotas dos veículos.
Ator principal	Leaflet/Usuário
Descrição	Cada rota especificada deve ser identificada por uma linha no mapa.
Pré-condições	As coordenadas das rotas devem ser definidas e então deve ser carregadas no mapa.
Pós-condições	As linhas no mapa apresentam rotas com as suas determinadas cores para distinguir as diferentes rotas.

Fonte: Próprio Autor, 2016.

As informações sobre o veículo e o ponto de parada devem ser fornecidas ao usuário, por exemplo se o veículo é adaptado para deficientes ou se a parada de ônibus é coberta ou não, o caso de uso “Obter informações sobre o ônibus e parada de ônibus” é descrita na Tabela 9.

Tabela 9 - Descrição do caso de uso obter informações sobre o ônibus e parada de ônibus.

Nome do caso de uso	DCU05 - Obter informações sobre o ônibus e parada de ônibus.
Ator principal	Usuário
Descrição	Informações sobre o ônibus e parada de ônibus devem ser fornecidas ao usuário do sistema.
Pré-condições	As informações armazenadas no banco de dados devem ser apresentadas ao usuário quando o mesmo clica no item desejado.
Pós-condições	As informações mostradas no formato texto ou imagem.

Fonte: Próprio Autor, 2016.

O caso de uso “Verificar tempo de chegada do próximo ônibus” está descrita na Tabela 10. A função do cálculo do tempo do próximo ônibus é útil ao passageiro, pois o passageiro tem uma estimativa de quando o ônibus chegará à uma parada desejada.

Tabela 10 - Descrição do caso de uso verificar tempo de chegada do próximo ônibus.

Nome do caso de uso	DCU06 - Verificar tempo de chegada do próximo ônibus
Ator principal	Usuário
Descrição	A partir de um cálculo deve ser estimado o tempo aproximado em que o ônibus demorará para chegar em uma parada de ônibus.
Pré-condições	Posição atual do ônibus, parada desejada pelo usuário e velocidade média do veículo.
Pós-condições	O tempo de chegada para ser exibido ao passageiro.

Fonte: Próprio Autor, 2016.

O usuário poderá pesquisar pontos de ônibus a partir de uma lista, isso é um meio rápido para pesquisar as paradas de ônibus. Esse caso de uso é descrito na Tabela 11.

Tabela 11 - Descrição do caso de uso pesquisar pontos de ônibus.

Nome do caso de uso	DCU07 - Pesquisar pontos de ônibus.
Ator principal	Usuário
Descrição	A partir de um painel de busca o usuário do sistema poderá pesquisar uma parada de ônibus ou um ônibus.
Pré-condições	Serão listadas todas as paradas de ônibus e ônibus possíveis para busca.
Pós-condições	As informações da parada de ônibus ou do ônibus serão mostradas ao usuário do sistema.

Fonte: Próprio Autor, 2016.

Caso o usuário tenha o aplicativo instalado no celular, mas esteja sem plano de dados, ele poderá obter o cronograma *off-line* do itinerário em uma determinada rota. Esse caso de uso é descrito na Tabela 12.

Tabela 12 - Descrição do caso de obter informação sobre o itinerário.

Nome do caso de uso	DCU09 - Obter informação sobre o itinerário.
Ator principal	Usuário
Descrição	No modo off-line (sem internet) o usuário pode obter informação do cronograma de uma rota.
Pré-condições	O aplicativo deve estar instalado no aparelho móvel.
Pós-condições	Itinerário disponíveis no formato tabular.

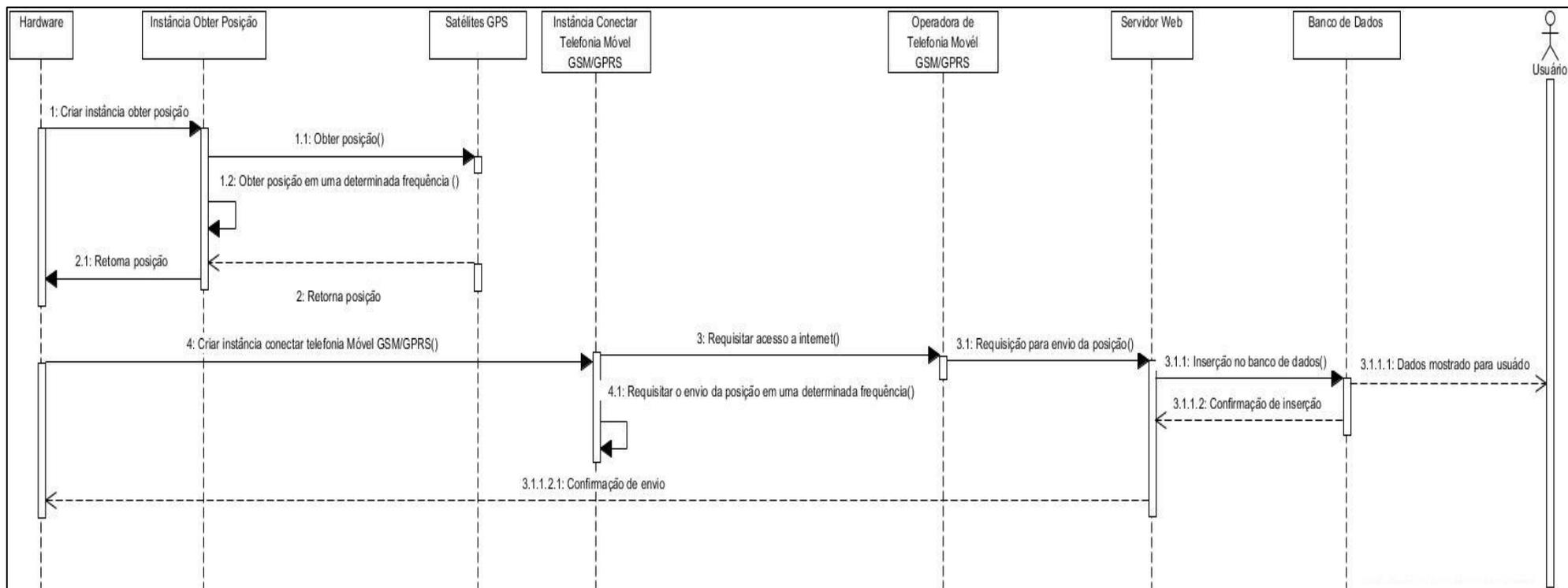
Fonte: Próprio Autor, 2016.

#### 4.1.4 Diagrama de sequência

Na Figura 11 é apresentado o diagrama de sequência do sistema embarcado implementado no rastreador, inicialmente é realizado a triangulação de no mínimo três satélites para obtenção da posição do ônibus. Após a triangulação dos satélites, o rastreador obtém a posição do veículo com uma determinada frequência. A posição obtida é armazenada no banco de dados através de uma conexão da rede de telefonia móvel. Desta maneira, a localização do ônibus estará disponível para ser apresentada ao usuário.

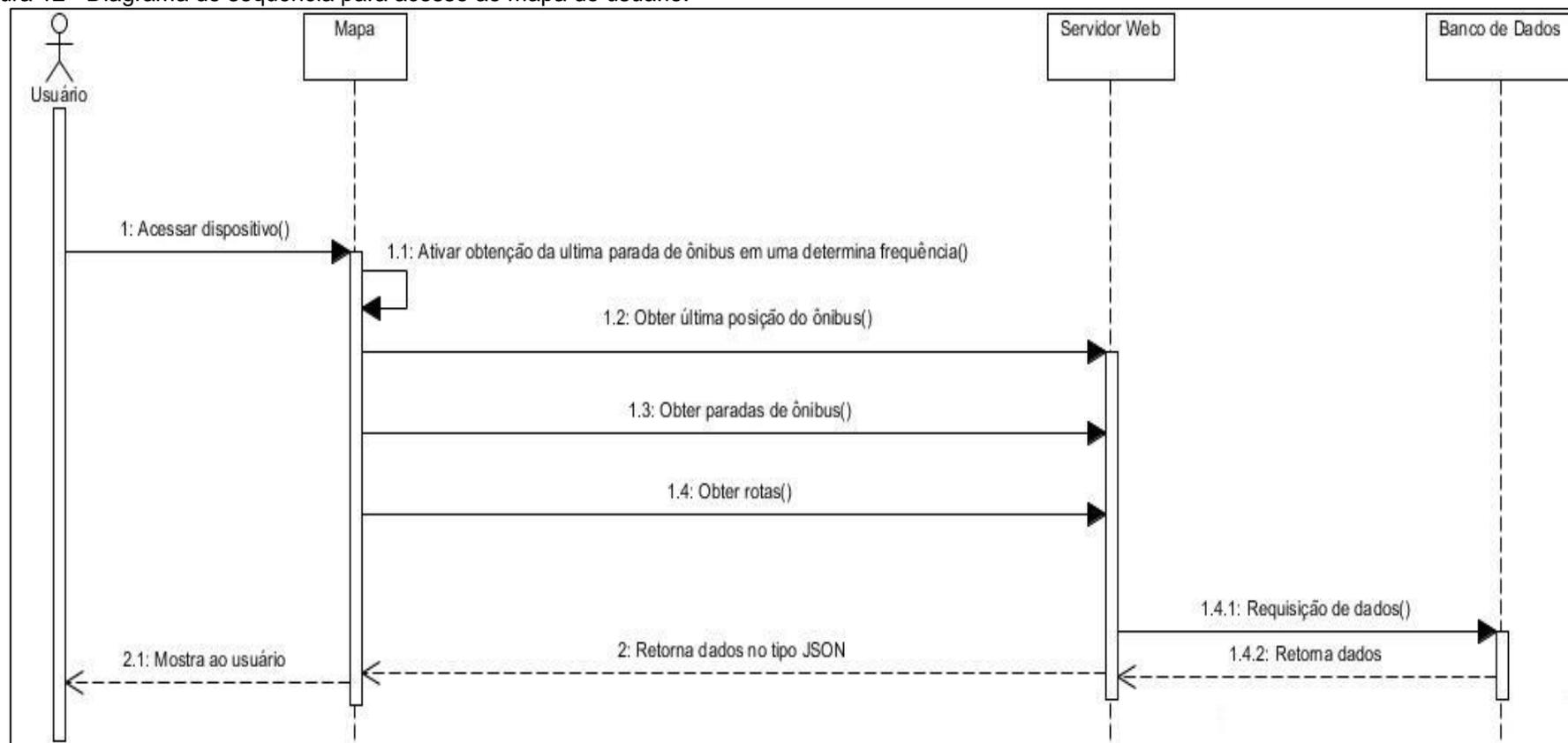
Na Figura 12 tem-se o diagrama de sequência para o acesso ao mapa do usuário. Quando o usuário abre o mapa, um conjunto de *scripts* realizam requisições no servidor Web para obtenção dos objetos para serem plotados no mapa. Posteriormente esses objetos são plotados no mapa. Já a função de atualização da posição do ônibus é requisitada frequentemente para o acompanhamento da localização do ônibus.

Figura 11 - Diagrama de seqüência do sistema embarcado implementado no rastreador.



Fonte: Próprio Autor, 2016.

Figura 12 - Diagrama de sequência para acesso ao mapa do usuário.



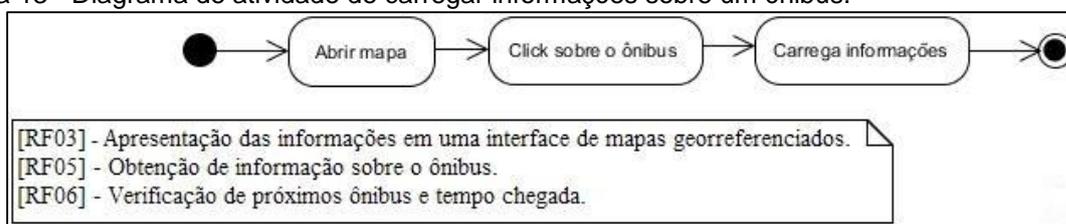
Fonte: Próprio Autor, 2016.

#### 4.1.5 Diagrama de atividade

Na Figura 13 apresenta-se o diagrama de atividade para carregar as informações de um ônibus. Esse diagrama tem a mesma estrutura de atividade para carregar as informações de uma parada de ônibus. As etapas são descritas a seguir:

- I. O usuário abre o mapa;
- II. Após aberto o mapa, o usuário clica sobre o ônibus;
- III. Todas as informações referentes ao ônibus são carregadas.

Figura 13 - Diagrama de atividade de carregar informações sobre um ônibus.

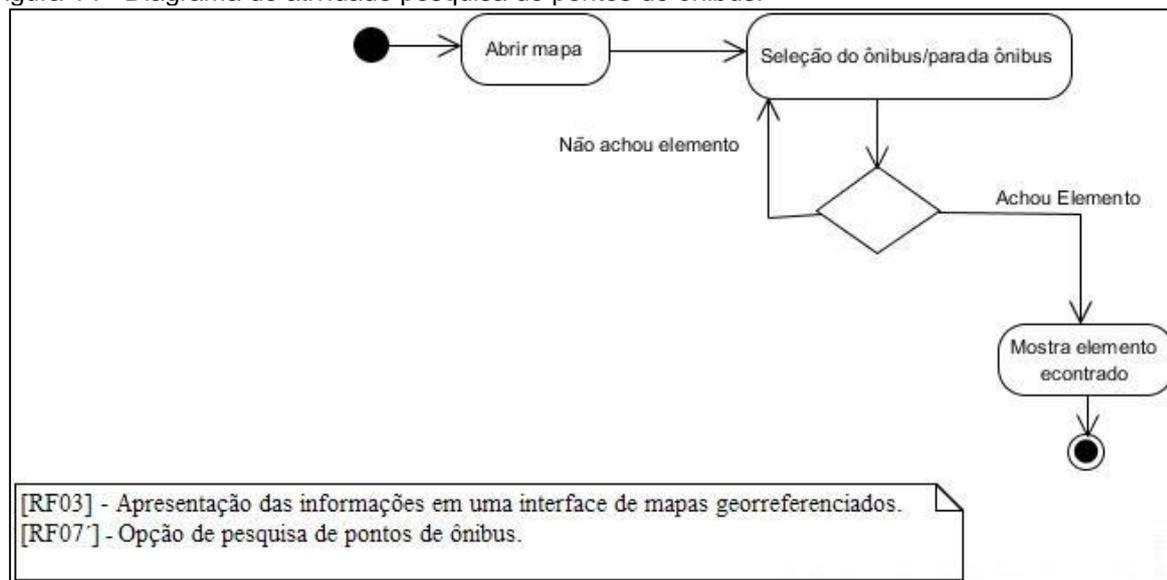


Fonte: Próprio Autor, 2016

A Figura 14 mostra o diagrama de atividade de uma pesquisa de ponto de ônibus que consiste das seguintes etapas:

- I. Usuário abre o mapa;
- II. Usuário digita o código/nome do ônibus ou parada de ônibus em uma caixa de pesquisa;
- III. Caso for encontrado o elemento desejado, então ele é mostrado para o usuário. Caso não for encontrado, o usuário pode fazer outras tentativas para encontrar o elemento desejado.

Figura 14 - Diagrama de atividade pesquisa de pontos de ônibus.



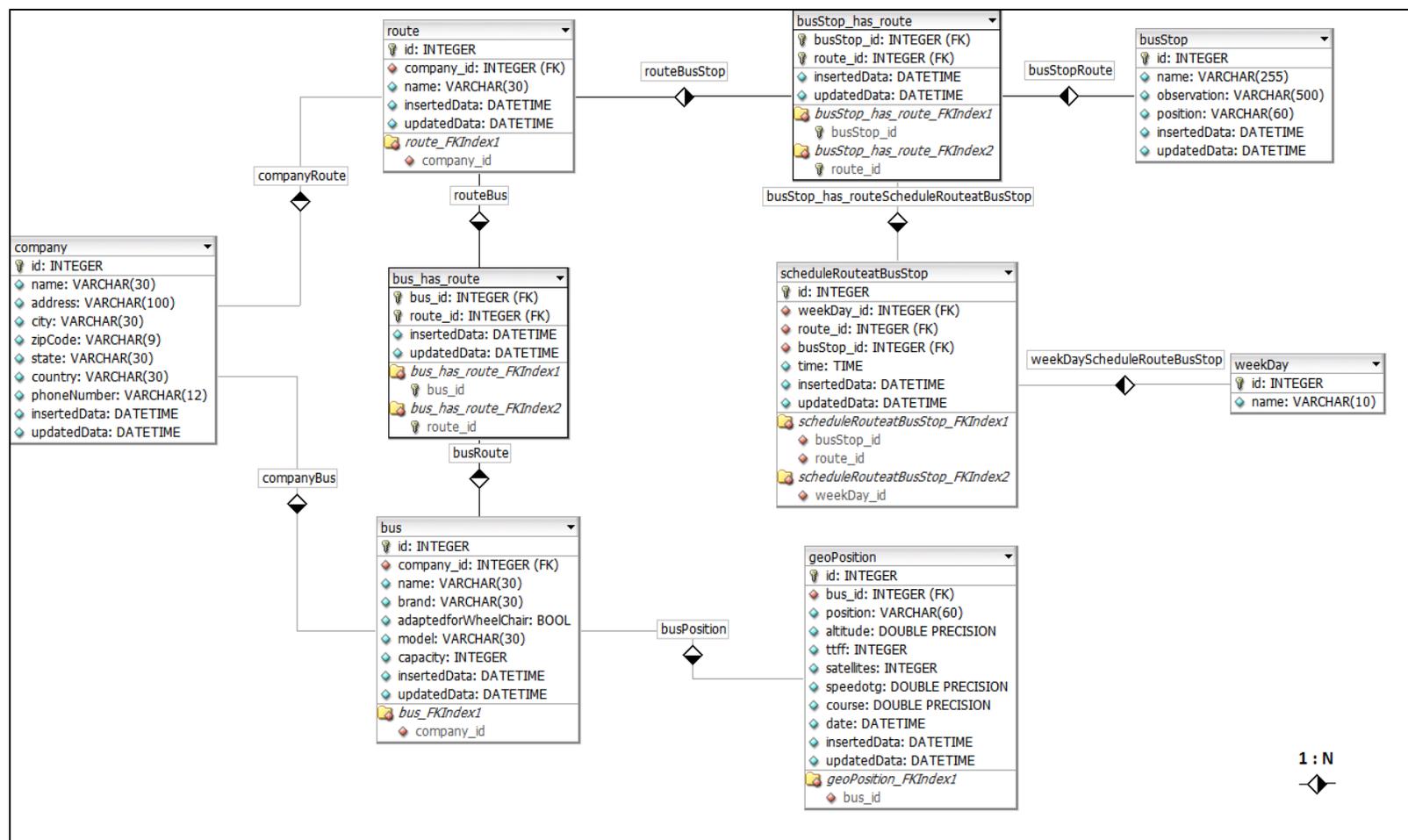
Fonte: Próprio Autor, 2016.

#### 4.1.6 Desenvolvimento do banco de dados – modelo entidade relacionamento

O modelo ER do banco de dados está representado na Figura 15, neste modelo foram criadas nove tabelas na notação Entidade-Relacionamento Estendido (EER). As tabelas estão relacionadas de modo que elas garantam a integridade dos dados.

As tabelas “geoPosition” e “busStop” contém colunas de posições do tipo de dados geográficos, através desse tipo de dados é possível utilizar funções já existentes de processamento de localização, como cálculo de distância entre dois pontos geográficos e seleções de dados em regiões específicas.

Figura 15 – Modelo ER.



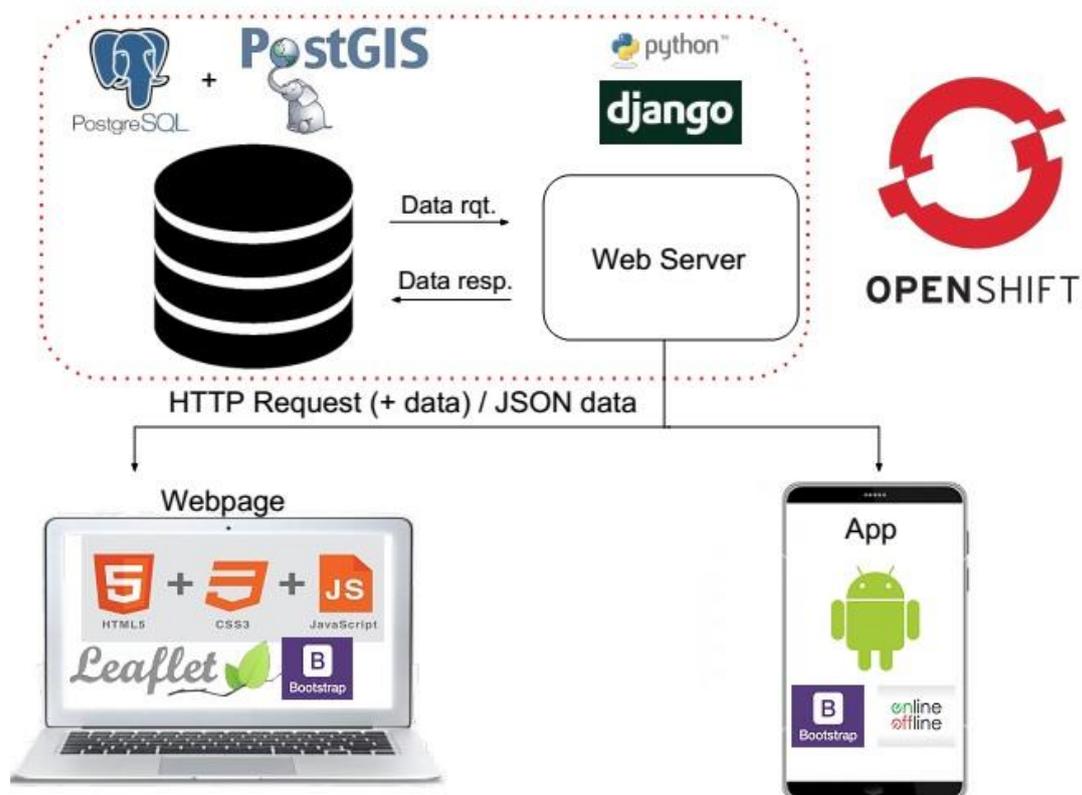
Fonte: Próprio Autor, 2016.

## 4.2 Arquitetura modelada

Na Figura 16 apresenta-se a arquitetura Web do funcionamento do sistema de rastreamento de transporte público, utilizada para a elaboração das funcionalidades contempladas no escopo do projeto. Na arquitetura Web são apresentadas as tecnologias e protocolos utilizados para o desenvolvimento do sistema Web e do aplicativo mobile. A requisição de dados entre a interface do usuário e o *webserver* utiliza o protocolo HTTP, após são retornados os dados no formato JSON. A interface Web utiliza as seguintes tecnologias: HTML5, CSS, Javascript, Leaflet e bootstrap. O aplicativo para Android foi programado em linguagem nativa do sistema Java.

Utilizou-se o PostgreSQL como Sistema de Gerenciamento de Banco de Dados (SGBD) e a sua extensão espacial (PostGIS) para armazenamento das posições no PostgreSQL.

Figura 16 – Arquitetura Web do sistema de rastreamento de transporte público.



Fonte: Próprio Autor, 2016.

### 4.3 Desenvolvimento do sistema

Neste tópico são detalhados o desenvolvimento do sistema Web, do aplicativo e do hardware.

#### 4.3.1 Serviço Web

Para o desenvolvimento e implementação desse projeto, foi necessário o acesso externo ao servidor Django hospedado em Openshift, através do link <http://rytmuv2-rytmuv.rhcloud.com/>.

Para facilitar a programação foi utilizado o software PyCharm, que é um sistema de desenvolvimento integrado que usa diversos recursos como análise de código, *debugger*, sistema integrado de teste e suporte para o desenvolvimento no *framework* Django. Na Figura 17 é mostrado o IDE Pycharm com os arquivos e funções utilizadas no projeto.

Figura 17 - IDE PyCharm.

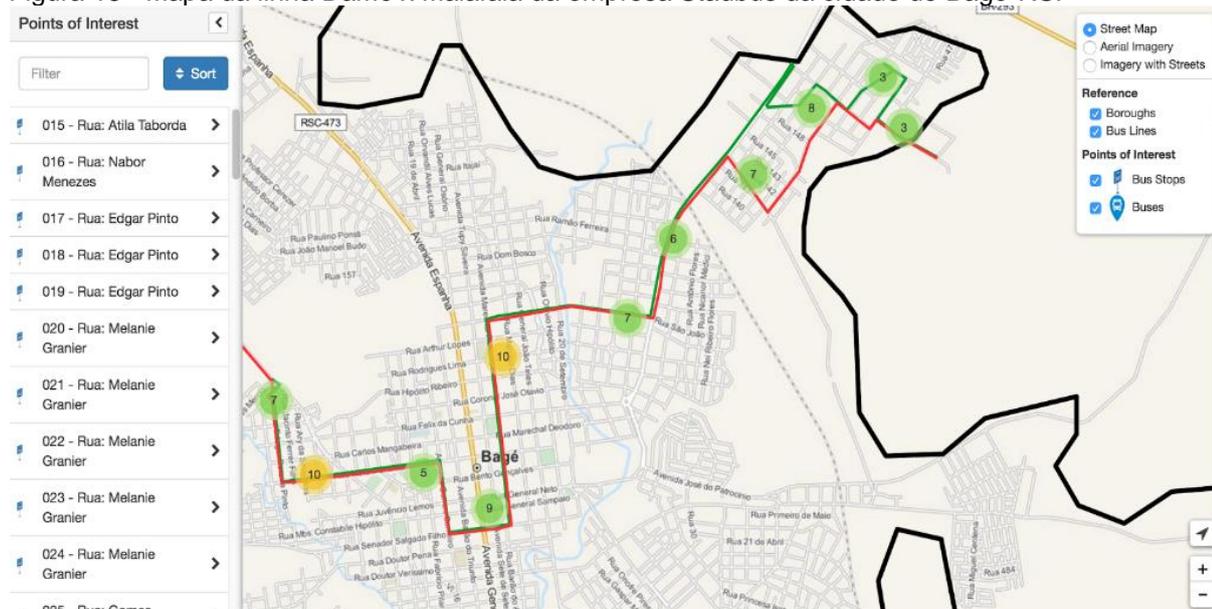


Fonte: Próprio Autor, 2016.

#### 4.3.2 Páginas Web

Foram desenvolvidas páginas Web com o auxílio do *framework* LeafLet da linha “Damé x Malafaia” da empresa Stadtbuss da cidade de Bagé-RS. O mapa é composto de *layers* nos quais foram plotados as paradas de ônibus e o sentido da linha, na Figura 18 apresenta-se o mapa desenvolvido.

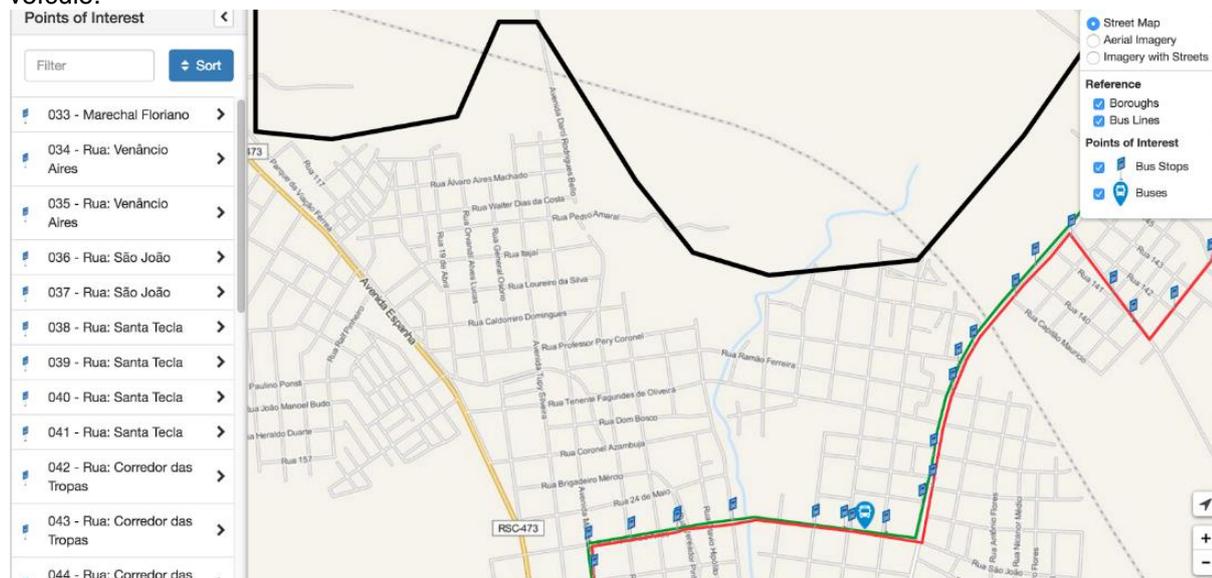
Figura 18 - Mapa da linha Damé x Malafaia da empresa Stadbus da cidade de Bagé-RS.



Fonte: Próprio Autor, 2016.

Com o apoio da prefeitura de Bagé-RS e da empresa Stadbus, obteve-se a localização geográfica das oitenta e oito paradas de ônibus da linha “Damé x Malafaia”. A Figura 19 mostra parte da linha com suas paradas e a localização do ônibus.

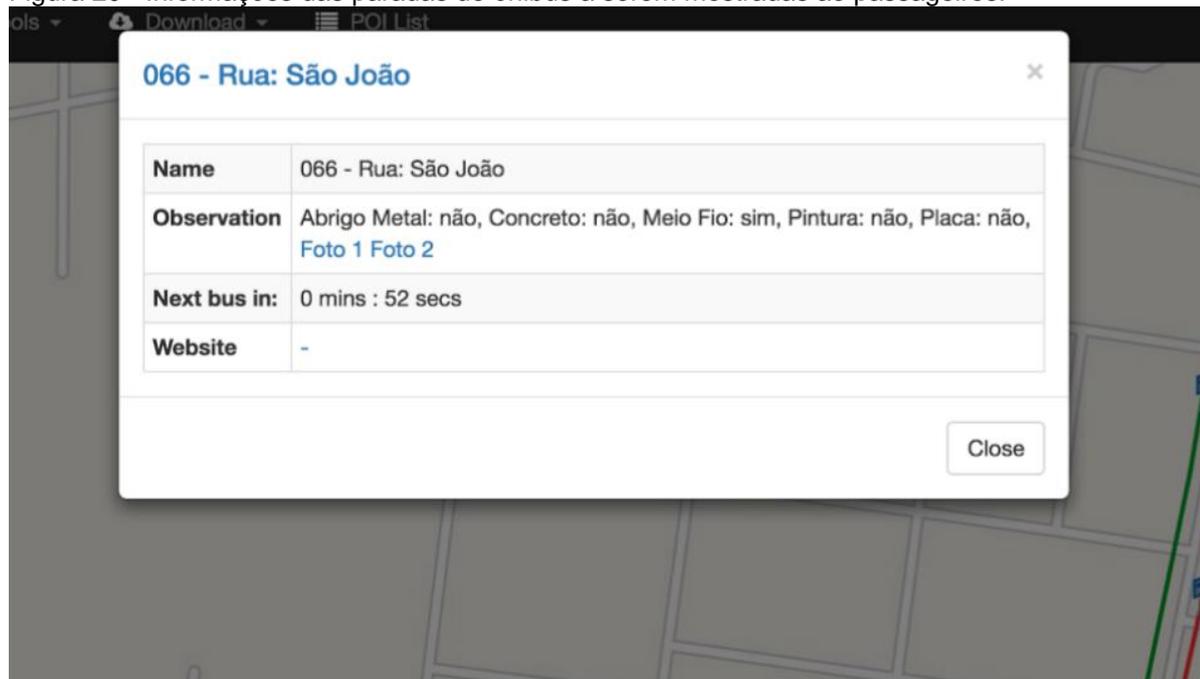
Figura 19 – Parte da linha “Damé x Malafaia” com as suas paradas de ônibus e a localização do veículo.



Fonte: Próprio Autor, 2016.

O passageiro poderá clicar sobre as paradas de ônibus para obter as informações das previsões de chegada do veículo nas mesmas. A Figura 20 mostra as informações de uma parada de ônibus. Essas informações são: id, nome da parada, existência de abrigo de metal, existência de abrigo de concreto, existência de meio fio, pintura do abrigo, existência de sinalização, fotos e a previsão de chegada do próximo ônibus.

Figura 20 - Informações das paradas de ônibus a serem mostradas ao passageiros.



Fonte: Próprio Autor, 2016.

Na Figura 21 a e b apresentam-se fotos da parada de ônibus na Avenida Santa Tecla, nº 1188. Nessa figura são apresentadas as vistas lateral e frontal da parada. Através dessas fotos o passageiro poderá identificar as paradas de ônibus com maior facilidade.

Figura 21 - Foto da vista lateral e vista frontal da parada na Avenida Santa Tecla, nº 1188.

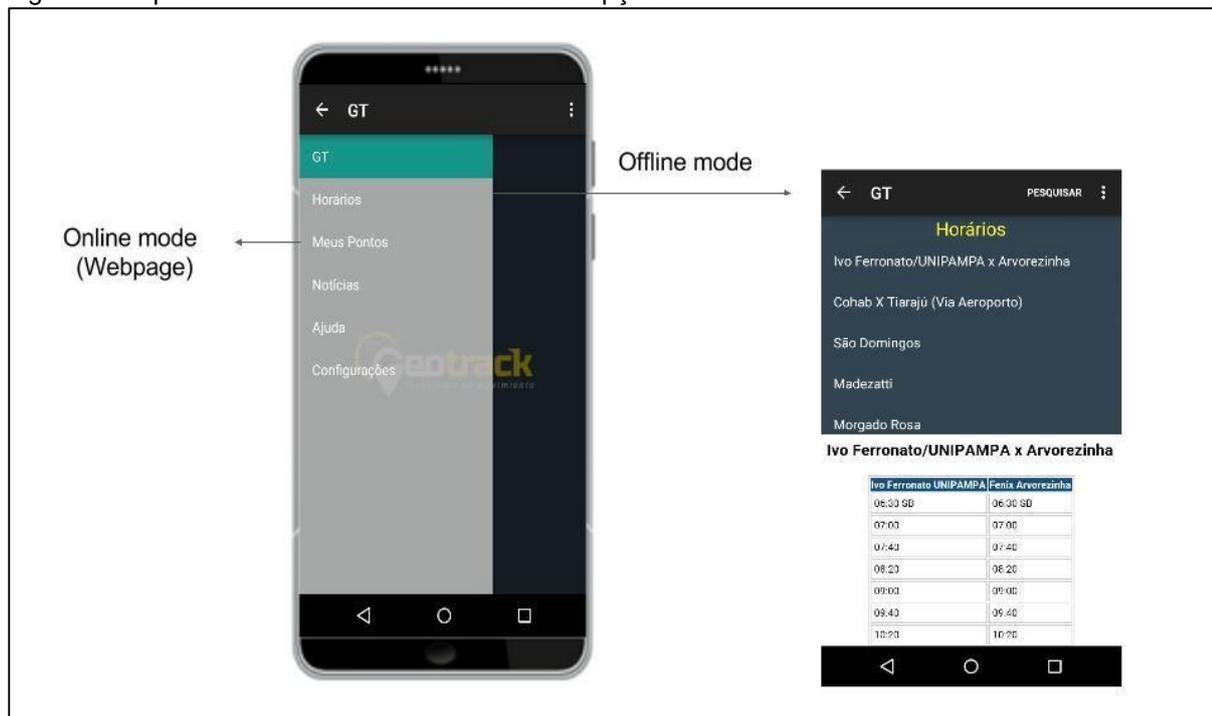


Fonte: Próprio Autor, 2016.

### 4.3.3 Aplicativo

Na Figura 22 apresenta-se a tela do aplicativo Android desenvolvido com o menu de opções. A utilização de menu no aplicativo favorece a navegação nas opções disponíveis.

Figura 22- Aplicativo no Android com o menu de opções.



Fonte: Próprio Autor, 2016.

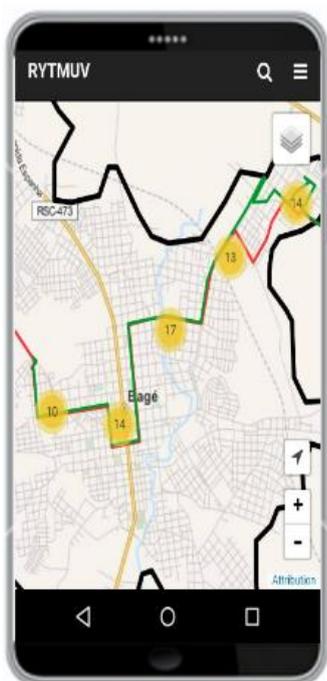
Para agilizar o processo de criação do aplicativo, foi utilizado o *framework* Bootstrap. O *framework* Bootstrap é responsável por redimensionar a tela do aplicativo de acordo com cada *smartphone*.

A Figura 23 mostra o modo on-line do aplicativo, esse modo é responsável por carregar a página do mapa desenvolvido. Na Figura 23 são mostradas as opções:

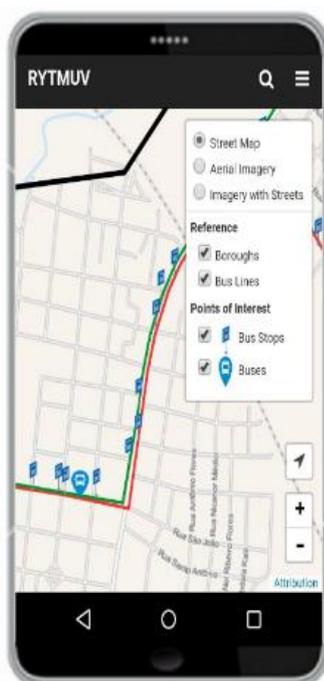
- mapa com os cluster das paradas de ônibus;
- mapa com as paradas de ônibus e a localização do ônibus;
- informações da parada de ônibus.

Figura 23 - Modo on-line do aplicativo Android.

a) Clusters paradas de ônibus



b) Paradas de ônibus/ônibus



c) informações parada de ônibus



Fonte: Próprio Autor, 2016.

#### 4.3.4 Desenvolvimento e instalação do hardware

Na Figura 24 apresenta-se uma fotografia do *hardware* utilizado. O módulo SIM 808 foi acoplado no Arduino UNO através de um *shield*. Foi adquirido uma caixa acrílica para proteger o circuito de sujeiras e água, evitando a danificação das placas. O preço total do hardware foi de aproximadamente R\$300,00.

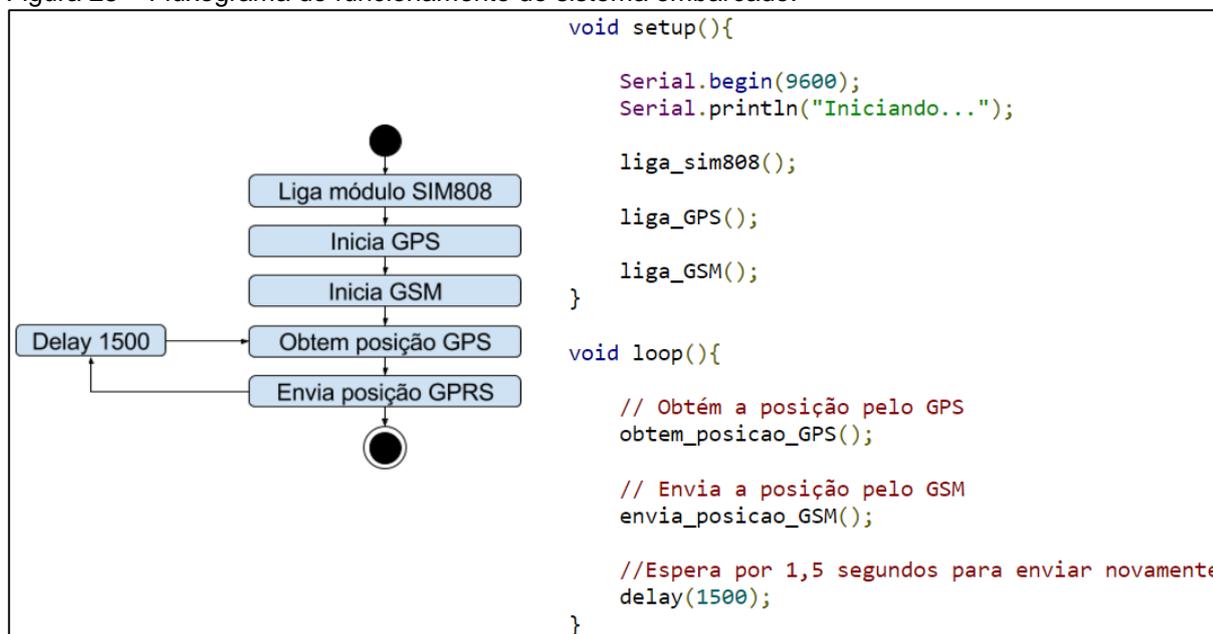
Figura 24 - Hardware utilizado neste projeto.



Fonte: Próprio Autor, 2016.

O fluxograma da Figura 25 ilustra o funcionamento do sistema embarcado desenvolvido.

Figura 25 – Fluxograma do funcionamento do sistema embarcado.



Fonte: Próprio Autor, 2016.

Primeiramente é necessário inicializar e configurar os elementos do módulo, três elementos são inicializados: a placa SIM808, módulo GPS e módulo GSM. Para ligar o SIM808 é necessário pressionar o botão I/O da placa SIM808 por 3

segundos, logo depois é enviado um simples comando AT do microcontrolador para o SIM808, caso o retorno seja “OK”, então a placa foi ligada corretamente. Após é ligado o módulo GPS e realizado a triangulação dos satélites. Na configuração do módulo GSP, são informados o Nome do Ponto de Acesso (APN) do provedor de internet, usuário e senha. Após a correta inicialização e configuração desses módulos é possível obter a posição do GPS e enviá-la para o servidor através da rede GSM.

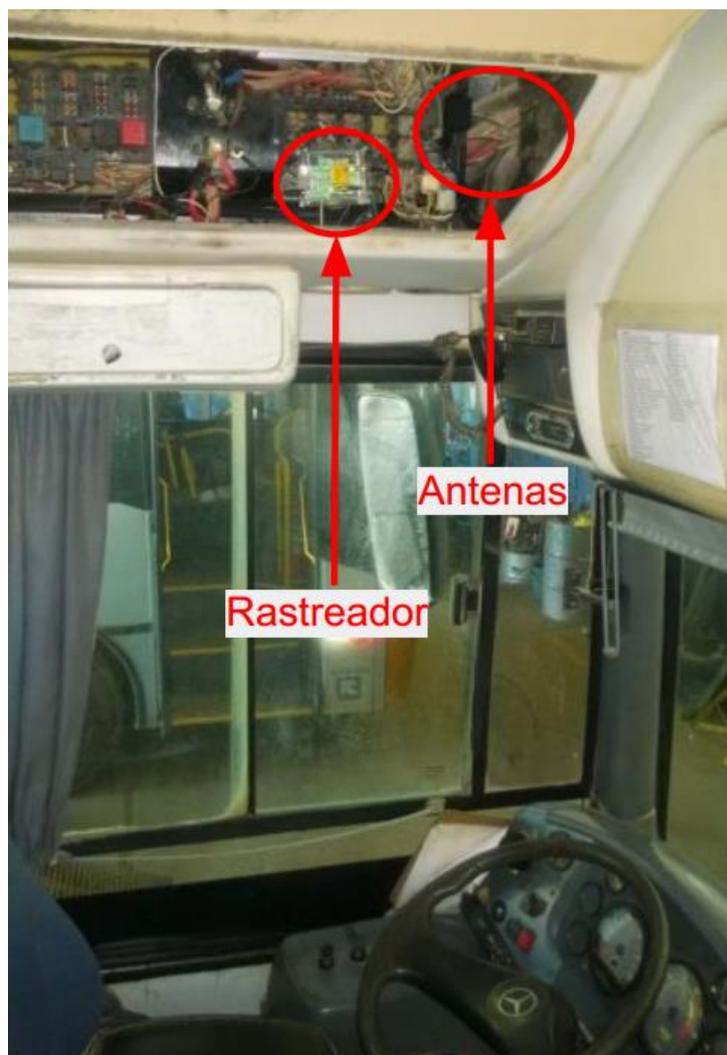
#### **4.4 Projeto Piloto**

Com o intuito de verificar se o sistema funcionaria de maneira satisfatória foi realizado um teste do sistema desenvolvido, em um ônibus da linha urbana da cidade de Bagé-RS. Foi utilizada uma operadora que fornece o serviço GPRS na cidade, a escolha da operadora foi baseada na qualidade do sinal GPRS. A cobertura do sinal GPRS oferecida pela operadora na cidade de Bagé-RS é demonstrada no mapa do Anexo A. O hardware foi instalado em um ônibus da empresa Stadtbuss na linha “Damé x Malafaia”, o rastreador foi conectado na parte elétrica do veículo que forneceu uma tensão de 12 volts para a alimentação do mesmo. Durante a instalação do hardware, contou-se com o acompanhamento e com o auxílio do mecânico responsável pela frota.

Durante quatro dias, o rastreador enviou a localização para o servidor. Nesse período foram armazenados mais de cinquenta mil registros na base de dados. O envio da posição do ônibus para o servidor foi realizado aproximadamente 4 segundos, essa amostragem mostrou-se adequada para a visualização do veículo em movimento no mapa. Caso a frequência de envio seja mais alta, o pacote de dados contratado poderia esgotar, inviabilizando o envio das posições. Nos experimentos, foi gasto 1MB/hora, caso o ônibus opere por 16horas/dia em 1 mês será gasto 480MB. A operadora oferece diversas franquias, a franquia que melhor se adequa a esse consumo tem um valor de R\$24,99 mensal por 600MB/mês.

A Figura 26 indica o local da instalação do rastreador. Ele foi instalado próximo a cabine do motorista e da central do veículo. O dispositivo foi fixado com cintas plásticas. Já as antenas foram anexadas com os imãs das próprias antenas.

Figura 26 - Hardware instalado próximo a cabine do motorista.



Fonte: Próprio Autor, 2016.

Os dados gerados pelo rastreador foram coletados e armazenados no banco de dados. As informações armazenadas foram:

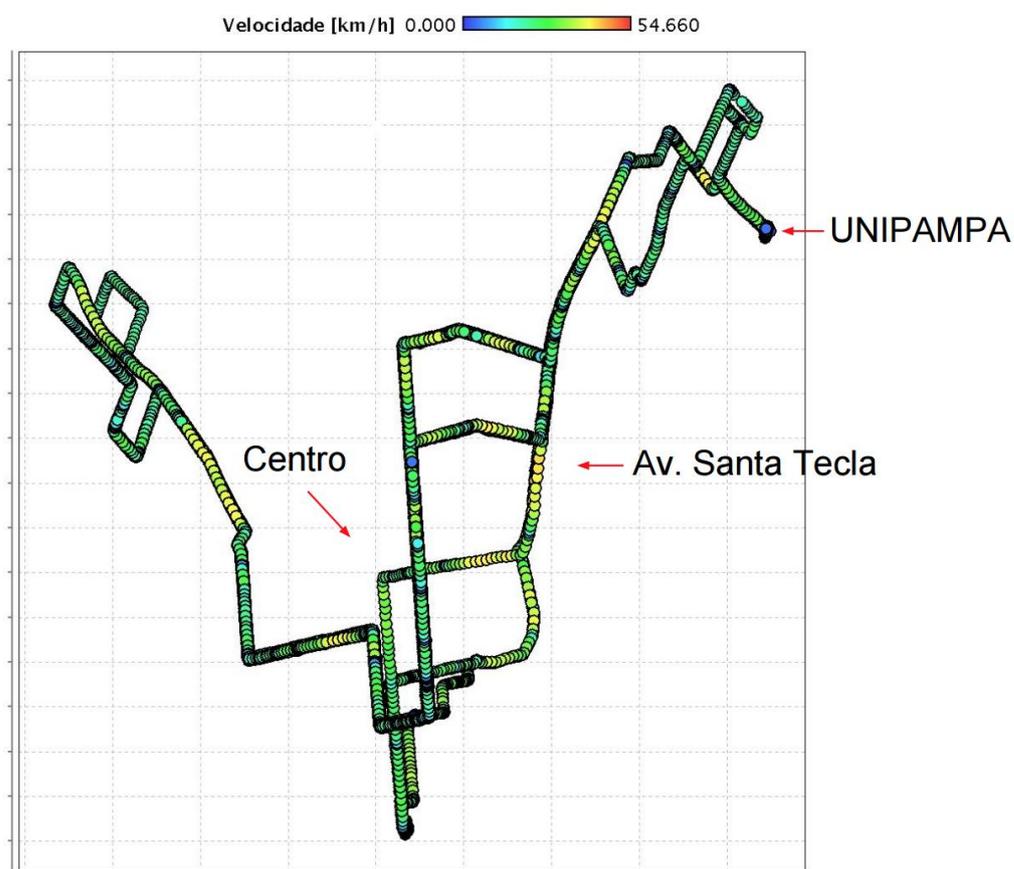
- I. Latitude;
- II. Longitude;
- III. Altitude;
- IV. Data/hora - No formato AnoMêsDiaHorasMinutosSegundos.
- V. (*Time to First fix*) TTFF - Tempo para estabilizar a procura pelos satélites em segundos;
- VI. Satélites – Número de satélites localizado;
- VII. Velocidade – Velocidade sobre a terra;

### VIII. Direção – Em graus, de 0 a 360 graus.

Para o cálculo da chegada do veículo na parada solicitada foi utilizada como referência a velocidade média de 18 Km/h, proposta em Meirelles (2000). O cálculo do tempo de chegada do veículo é realizado dividindo a distância entre a posição atual do ônibus e a parada pela velocidade média. A velocidade média da linha estudada foi de 14,90 Km/h, desta forma a previsão da chegada do ônibus será adiantada. Essa antecipação favorece ao passageiro chegar na parada com antecedência.

A partir das posições e velocidades coletadas, foi plotado o mapa da Figura 27 através do programa RapidMiner Studio que auxilia a exploração de dados.

Figura 27 - Mapa do percurso realizado pelo ônibus com as velocidades.



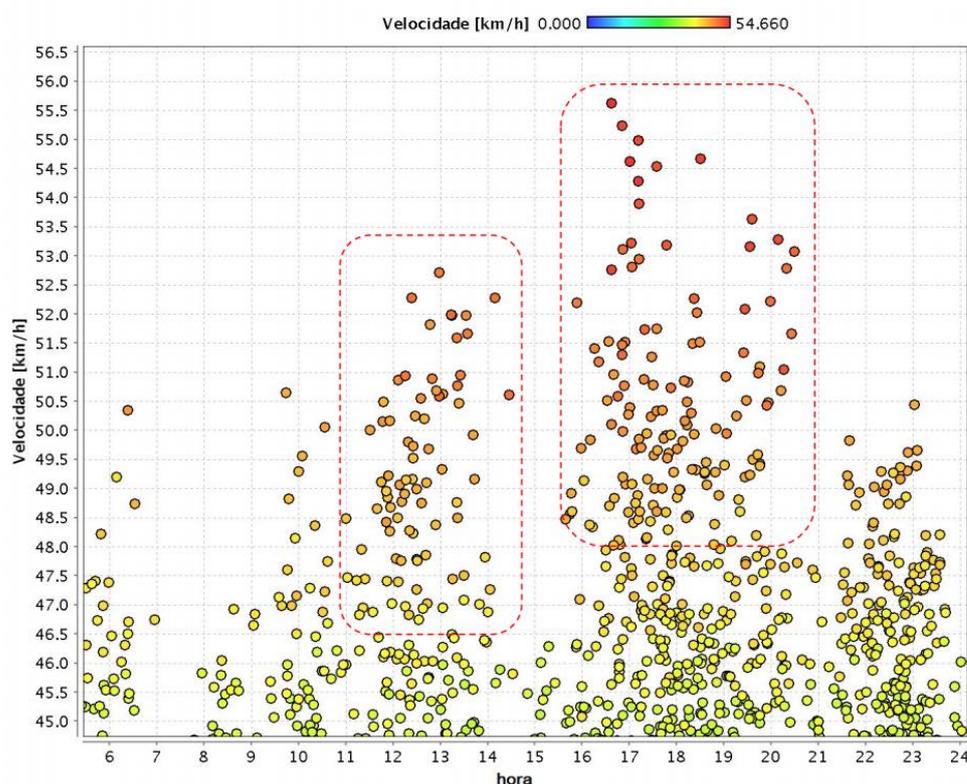
Fonte: Próprio Autor, 2016.

É possível observar, na Figura 27, que nas retas o veículo teve maiores velocidades (em amarelo). Esse tipo de informação pode ser útil para os gestores controlarem a velocidade do veículo a fim de evitar acidentes e multas por alta

velocidade. Além disso, a informação pode ser incorporada em um algoritmo e ser utilizada para melhorar a previsibilidade de chegada do ônibus na parada.

Na Figura 28 foram plotadas as velocidades máximas durante os horários do dia. No gráfico dessa figura, é possível verificar duas divisões. A primeira divisão apresenta pico de velocidade entre as 11 horas e 14 horas. O outro pico de velocidade foi verificado entre as 16 horas e 20 horas. Estes picos podem ocorrer devido aos congestionamentos de saída para almoço dos comércios e escola; e do fechamento dos comércios e início das aulas nas instituições de ensino. Deste modo, os motoristas compensam os horários gastos nos congestionamentos, aumentando as velocidades nas retas. Através deste tipo de gráfico, os gestores podem determinar os horários para disponibilizar ônibus de reforço.

Figura 28 - Velocidades máximas distribuídas durante os horários do dia.



Fonte: Próprio Autor, 2016.

## 5 CONCLUSÕES E TRABALHOS FUTUROS

O objetivo geral desse trabalho foi o desenvolvimento de um sistema de rastreamento para solucionar o problema da imprevisibilidade da chegada do ônibus. Para isso foram desenvolvidos um sistema Web e um aplicativo, capazes de acessar a localização real do ônibus. Após o desenvolvimento do sistema Web e do aplicativo, foi implementado o sistema embarcado para obter a posição do veículo e enviá-la para um servidor. Por fim, foi realizado o teste do hardware em uma linha no município de Bagé-RS.

Para o desenvolvimento do sistema Web e do aplicativo foram utilizadas diferentes tecnologias que auxiliaram o desenvolvimento do sistema. Essas tecnologias foram o Django, Bootstrap e Leaflet. Elas agilizaram o desenvolvimento do sistema e garantiu a estruturação do projeto. Com a boa organização dos códigos, foi possível reutilizar grande parte do sistema. Como por exemplo, foram programadas uma única vez as consultas ao banco de dados, assim tanto a página web quanto o aplicativo acessaram as mesmas consultas.

A implementação do sistema embarcado foi dividida em diversos estágios. Os estágios foram: ligar módulo SIM808, iniciar GPS, iniciar GPRS, obter posição do GPS, enviar posição através do GPRS. Um estágio só poderia prosseguir para o próximo se o atual fosse executado sem erro. Na programação do sistema embarcado, diversos problemas aconteceram devido a configurações incorretas dos GPS e GPRS. Para identificar os erros nas configurações, foi utilizada a saída serial do módulo SIM808, assim foi possível identificar o código de cada erro retornado pelo SIM808.

Para a realização do teste do hardware na linha de ônibus, foram contatadas as duas empresas de transporte público do município: Anversa e Stadtbuss. Para efetivar o teste em um ônibus, deparou-se com algumas dificuldades como: indisponibilidade de ônibus para teste, dificuldade de contatar os responsáveis da empresa e a necessidade de um responsável da empresa para a realização do teste. Apesar desses desafios, foi possível realizar a instalação do rastreador em um ônibus da empresa Stadtbuss. O hardware comportou-se como esperado, sendo possível o rastreamento do mesmo pelo aplicativo e a estimativa da chegada do ônibus utilizando uma velocidade média de 18 Km/h baseado no trabalho de Meirelles (2000).

Além de ser disponibilizado a localização do ônibus em tempo real para os passageiros através dos dados coletados, é possível gerar conhecimento para o gestor da empresa. Assim, o gestor da empresa pode tomar decisões mais concisas para evitar gastos desnecessários. Além dos gráficos plotados neste trabalho, outras informações podem ser correlacionadas com a posição do veículo como motoristas e lotação, assim possibilitando extrair mais informações.

Todos os requisitos levantados previamente foram atendidos e todo o sistema funcionou de maneira aceitável. Porém, devido ao alto custo de implementação do módulo GPS/GSM, para trabalhos futuros deve ser levado em consideração a aquisição de rastreadores comerciais com preços mais acessíveis. Nas pesquisas realizadas, foram encontrados rastreadores que atendem aos requisitos do sistema com valores que variam entre R\$50,00 a R\$100,00. Além da questão do preço, deve ser levado em consideração, em trabalhos futuros, a aquisição de rastreadores com sistema embarcado implementado, para assim agilizar o desenvolvimento do projeto como todo.

Além do conhecimento técnico adquirido durante a pesquisa e o desenvolvimento do sistema, foi possível entender, parcialmente, o funcionamento do transporte público na cidade de Bagé. Foi possível verificar a necessidade de melhorias nesse setor, e uma das maneiras de aprimorar o transporte público seriam através de ferramentas tecnológicas. Portanto espera-se que passageiros possam se beneficiar dessas tecnologias para melhorar a qualidade dos serviços de transporte público prestados à população, que merece ser tratada com respeito e com dignidade.

## REFERÊNCIAS

ANTP, A. N. DE T. P.-. **Sistema de Informações da Mobilidade Urbana Relatório Geral 2013**. São Paulo: ANTP, 2013.

BYANSHUL SRIVASTAVA; EMARKETER. **2 Billion Smartphone Users By 2015: 83% of Internet Usage From Mobiles [Study]**. Disponível em: <<http://dazeinfo.com/2014/01/23/smartphone-users-growth-mobile-internet-2014-2017/>>. Acesso em: 23 abr. 2016.

DENATRAN, D. N. DE T.-. **Frota municipal de veículos**. Brasília, 2015.

DILÃO, R. **Latitudes e Longitudes**. Lisboa: Ministério da Ciência e da Tecnologia, , 1999.

DINKAR, A. S.; SHAIKH, S. A.; SIGNIFICANCE, P. **Design and Implementation Of Vehicle Tracking System Using**. v. 1, n. 3, p. 1–7, 2011.

DIRK KRAFZIG, KARL BANKE, D. S. **Enterprise SOA: Service-Oriented Architecture Best Practices**. Prentice Hall, 2004.

ELMASRI, R.; NAVATHE, S. B. **Sistemas de banco de dados**. 4. ed. São Paulo: Person, 2005.

FLANAGAN, D. **JavaScript: The Definitive Guide**. Pacific Northwest between the cities of Seattle, Washington: O'Reilly Media, 2006.

HEGARTY, C. J. **GNSS signals - An overview**. 2012 IEEE International Frequency

Control Symposium, IFCS 2012, Proceedings, 2012

III, P. C. **Leaflet.js Essentials**. Birmingham B3 2PB, UK: Packt Publishing Ltd., 2014.

IPEA. **A mobilidade urbana no Brasil**. p. 35, 2011.

JORDAN, L.; GREYLING, P. Android Projects. **Environment**, p. 418, 2011.

LERNER, J. **Avaliação comparativa das modalidades de transporte público urbano**. Nacional das Empresas de transporte Urbano, p. 92, 2009.

MAGALHÃES, C. T. DE A. **AVALIAÇÃO DE TECNOLOGIAS DE RASTREAMENTO POR GPS PARA MONITORAMENTO DO TRANSPORTE PÚBLICO POR ÔNIBUS**. Rio de Janeiro, 2008.

MARGOLIS, M. **Arduino Cookbook**. Palo Alto: O'Reilly Media, 2011.

MAURYA, K.; SINGH, M.; JAIN, N. **Real Time Vehicle Tracking System using GSM and GPS Technology-An Anti-theft Tracking System**. International Journal of Electronics and Computer Science Engineering, v. 1, p. 1103–1107, 2012.

MOBILAB. **Monitoramento de transporte coletivo privado**São PauloMobilab - Prefeitura de São Paulo, , 2015. Disponível em: <<http://mobilab.prefeitura.sp.gov.br/>>

MORO, T. D.; DORNELES, C. F.; REBONATTO, M. T. **Web services WS- \* versus Web Services REST**. REIC - Revista de Iniciação Científica, v. 11, n. 1, p. 36–51, 2011.

PRESSMAN, R. **Engenharia de Software**. 6a. edição, Addison-Wesley/Pearson, 2006.

RICHARDSON, L.; RUBY, S. **RESTful Web Services**. Nova York: O'Reilly Media, 2008.

RIGAUX, P.; SCHOLL, M.; VOISARD, A. **Spatial databases - With application to GIS**. San Diego: Morgan Kaufmann Publishers, 2002.

SILVEIRA, D. T.; CÓRDOVA, F. P. **A pesquisa científica**. Porto Alegre: UFRGS, 2009.

SOMMERVILLE, I. **Engenharia de Software**. Pearson, 2006.

SPURLOCK, J. **Bootstrap**. Simon St. ed. Sebastopol, CA: O'Reilly Media, Inc, 2014.

TIME AND DATE. **What are Longitudes and Latitudes?** Disponível em: <<http://www.timeanddate.com/geography/longitude-latitude.html>>. Acesso em: 22 abr. 2016.

VERMA, P. **GSM/GPRS module**. Disponível em: <<https://sites.google.com/site/puneetverma24/gsm-gprs>>. Acesso em: 24 abr. 2016.

WESTERMO. **Industrial Data Communication Westermo Handbook Theoretical and General Applications**. Eskilstuna, Sweden: Westermo, 2005.

WITOLD ZMITROWICZ; NETO, G. D. A. **Infra-Estrutura Urbana** São Paulo EPUSP, 1997.

## APÊNDICE

### APÊNDICE A – Saída do terminal do software embarcado

Starting...

AT  
OK

AT+CGPSPWR=1  
AT+CGPSRST=0

GPS Ready  
+CPIN: READY

AT+CGPSSTATUS?  
OK

AT+CGPSSTATUS?  
AT+CREG?  
AT+SAPBR=3,1,"Contype","GPRS"  
AT+SAPBR=3,1,"APN","gprs.oi.com.br"  
AT+SAPBR=3,1,"USER","oi"  
AT+SAPBR=3,1,"PWD","oi"  
AT+SAPBR=0,1  
AT+SAPBR=0,1

AT+CGPSINF=0  
+CGPSINF:  
0,3118.313589,5403.840468,256.185459,20160616184245.000,38,12,0.0000  
00,0.000000

AT+HTTPINIT  
AT+HTTPPARA="CID",1  
AT+HTTPPARA="URL","http://rytmuv2-  
rytmuv.rhcloud.com/GET\_insertgeoPosition/  
?bus\_id=1&lat=5403.840468&lon=3118.313589&alt=256.185459&time=201606  
16184245.000&TTF=38&sat=12&speedOTG=0.000000&course=0.000000"  
AT+HTTPACTION=0

Done!

AT+HTTPTERM  
AT+CGPSINF=0

+CGPSINF:  
0,3118.313589,5403.840468,256.185467,20160616184249.000,38,12,0.0000  
00,0.000000

```
0
AT+HTTPINIT
AT+HTTPPARA="CID",1
AT+HTTPPARA="URL","http://rytmuv2-
rytmuv.rhcloud.com/GET_insertgeoPosition/
?bus_id=1&lat=5403.840468&lon=3118.313589&alt=256.185467&time=201606
16184249.000&TTFF=38&sat=12&speedOTG=0.000000&course=0.000000"
AT+HTTPACTION=0
Done!
AT+HTTPTERM
```

## ANEXOS

ANEXO A – Mapa de cobertura do Sinal GPRS em Bagé-RS – Fonte  
opensignal.com.