



**Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Curso de Engenharia Eletrônica**

**Rastreamento veicular com auxílio de dispositivos
móveis**

**Autor: Alan Azevedo Jacundá Ferreira
Willton Oliveira Gonçalves
Orientador: Georges Daniel Amvame Nze**

**Brasília, DF
2014**



Alan Azevedo J. Ferreira
Willton O. Gonçalves

Rastreamento Veicular com auxílio de dispositivos móveis

Monografia submetida ao curso de graduação em Engenharia Eletrônica da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia Eletrônica.

Orientador: Dr. Georges Daniel Amvame Nze

Brasília, DF
2014

CIP – Catalogação Internacional da Publicação

JACUNDÁ FERREIRA, ALAN AZEVEDO; GONÇALVES, WILLTON OLIVEIRA.

Rastreamento veicular com auxílio de dispositivos móveis: UnB, 2014.

Monografia (Graduação) – Universidade de Brasília
Faculdade do Gama, Brasília, 2014. Orientação: Georges Daniel Amvame Nze.

1. Rastreamento de Veículos. 2. Raspberry Pi. 3. SIG I.
Amvame Nze, Georges Daniel. II. Dr.

CDU Classificação



Rastreamento Veicular com auxílio de dispositivos móveis

Alan Azevedo J. Ferreira
Willton O. Gonçalves

Monografia submetida como requisito parcial para obtenção do Título de Bacharel em Engenharia Eletrônica da Faculdade UnB Gama - FGA, da Universidade de Brasília, em 18/08/2014 apresentada e aprovada pela banca examinadora abaixo assinada:

Prof. Doutor: Georges Daniel Amvame Nze, UnB/ FGA
Orientador

Prof. Doutor: André Barros de Sales, UnB/ FGA
Membro Convidado

Prof. Doutor: Leonardo Aguayo, UnB/ FGA
Membro Convidado

Brasília, DF

RESUMO

Os sistemas de navegação por satélite e as redes de telefonia móvel evoluíram significativamente desde que surgiram, possibilitando uma melhoria na comunicação e localização em conjunto com outras tecnologias. Os *smartphones* disponíveis no mercado englobam ambas as tecnologias, proporcionando inúmeras funcionalidades aos dispositivos como, por exemplo, no uso em monitoramentos de posição e rastreamento. Neste projeto, foi implementado um sistema de rastreamento veicular com o intuito de oferecer mais segurança à comunidade da Faculdade FGA – UnB, utilizando um dispositivo móvel dentro do veículo que funciona como o agente de rastreamento. Este, em contínua comunicação com o servidor, é alimentado através de um circuito que carrega sua bateria ininterruptamente. O servidor, por sua vez, utiliza o sistema LAMP *Open Source*, que recebe dados de localização constantemente do dispositivo móvel. A localização é interpretada pelo mapa com o auxílio da API do Google Maps em uma página *web*. O comportamento do sistema foi analisado de acordo com o tipo de rede (WLAN e WWAN) para avaliar a conexão de acordo com parâmetros como a latência, jitter, UDP e TCP. Após avaliação das redes, concluiu-se o melhor uso da rede WLAN por possuir um comportamento mais estável, maiores taxas de transferência de dados e se encaixar no cenário proposto em relação à restrição local, onde o servidor não é configurado com um IP mundial e por isso deve estar conectado na mesma rede do celular para o rastreamento do dispositivo móvel. Por fim, o uso do celular em conjunto com as tecnologias de internet, rede e GPS viabiliza um rastreamento mais efetivo e com maior precisão.

Palavras-chave: Rastreamento veicular, WLAN, WWAN, dispositivo móvel, GPS.

ABSTRACT

The satellite navigation systems and mobile phone networks have evolved significantly since they emerged, enabling an improvement in communication and location technologies in conjunction with other technologies. The smartphones available in the market include both technologies, and they provide numerous functionalities to devices like, for example, its use for position monitoring and tracking. In this project, a vehicle tracking system was implemented in order to provide more security to the community of the Faculty FGA - UNB, using a mobile device in a vehicle that acts as the agent of the trace. This one, in continuous communication with the server, is powered by a circuit that charges its battery continuously. The server, in another hand, uses the Open Source LAMP system, which constantly receives the location data from the mobile device. The location is interpreted by the map with the help of the Google Maps API in a web page. The system behavior was analyzed according to the type of network (WLAN and WWAN) to evaluate it according to the connection parameters such as latency, jitter, UDP and TCP. After the evaluation of the networks, it was concluded the best use of the WLAN due to a more stable performance, higher rates of data transfer and fit into the proposed scenario in relation to the local restriction, where the server is not configured with an world IP and so it must be connected to the same network that the cell is connected so the tracking of the mobile device can happen. Finally, the use of mobile technologies together with internet, network and GPS enables a more effective and more accurate tracking.

Keywords: vehicle tracking, WLAN, WWAN, mobile devices, GPS.

LISTA DE ILUSTRAÇÕES

| | |
|--|----|
| Figura 01 – Constelação sistema GPS (NINJA, 2010) | 21 |
| Figura 02 – Distribuição das estações de rastreamento no mundo (PETER, 2000)..... | 22 |
| Figura 03 – Segmentos do sistema de GPS(CEUB/ICPD, 2014)..... | 23 |
| Figura 04 – Constelação do sistema GLONASS (GLONASS, 2002)..... | 24 |
| Figura 05 – Tipos de rede e suas características (Wifi & Sécurité,2006)..... | 26 |
| Figura 06 – Topologia das redes WLAN (SILVA, 2008)..... | 27 |
| Figura 07 – Canais não sobrepostos. (JAMHOUR, 2013)..... | 29 |
| Figura 08 – Topologia de uma rede WWAN (SVERZUT, 2005)..... | 29 |
| Figura 09 – Estrutura de um SIG (GeoBrasil, 2006)..... | 34 |
| Figura 10 – Trecho do código que gera um mapa centralizado na cidade do Gama-DF | 38 |
| Figura 11 – Mapa exemplo gerado pelo <i>Google Map</i> | 38 |
| Figura 12 – Cenário para o Estudo de Caso..... | 46 |
| Figura 13 – Valores de latência (rede 3G)..... | 48 |
| Figura 14 – Valores de latência (rede WLAN)..... | 48 |
| Figura 15 – Valores de Jitter para <i>upload</i> (rede 3G)..... | 49 |
| Figura 16 – Valores de Jitter para <i>download</i> (rede 3G)..... | 49 |
| Figura 17 – Valores de Jitter para <i>upload</i> (rede WLAN)..... | 49 |
| Figura 18 – Valores de Jitter para <i>download</i> (rede WLAN)..... | 50 |
| Figura 19 – Valores de tráfego TCP para <i>upload</i> (rede 3G)..... | 51 |
| Figura 20 – Valores de tráfego TCP TCP para <i>upload</i> (rede WLAN)..... | 51 |
| Figura 21 – Valores de tráfego TCP para <i>download</i> (rede 3G)..... | 52 |
| Figura 22 – Valores de tráfego TCP para <i>download</i> (rede WLAN)..... | 52 |
| Figura 23 – Valores de vazão UDP para <i>upload</i> (rede 3G)..... | 53 |
| Figura 24 – Valores de vazão UDP para <i>upload</i> (rede WLAN)..... | 53 |
| Figura 25 – Valores de vazão UDP para <i>download</i> (rede 3G)..... | 54 |
| Figura 26 – Valores de vazão UDP para <i>download</i> (rede WLAN)..... | 54 |
| Figura 27 – Cenário do projeto..... | 56 |
| Figura 28 – Modelo de entrega dinâmico (SERVIDOR WEB)..... | 57 |
| Figura 29 – Links dos scripts e variáveis do código (<i>displaymap.php</i>)..... | 58 |
| Figura 30 – código do servidor (<i>displaymap.php</i>)..... | 59 |

| | |
|--|----|
| Figura 31 – Código do arquivo (<i>dbconnect.php</i>)..... | 60 |
| Figura 32 – Erro referente a não conexão com o banco de dados..... | 60 |
| Figura 33 – Parte do código do <i>updateLocation.php</i> | 61 |
| Figura 34 – Fluxograma do funcionamento do servidor web..... | 62 |
| Figura 35 – Servidor conectado com o banco de dados..... | 63 |
| Figura 36 – Status de localização com o celular conectado em redes WLAN..... | 64 |
| Figura 37 – Status de localização com o celular conectado em redes WLAN e antena GPS ativa..... | 65 |
| Figura 38 – Tela do aplicativo..... | 65 |
| Figura 39 – Função <i>startAlarmManager</i> | 66 |
| Figura 40 – Loop que define a variável de rastreamento..... | 66 |
| Figura 41 – Código <i>LoopjHttpClient</i> | 68 |
| Figura 42 – Iniciando o <i>startTracking</i> | 68 |
| Figura 43 – Obtendo a localização..... | 69 |
| Figura 44 – Calculando a distância percorrida..... | 69 |
| Figura 45 – Fluxograma de funcionamento da aplicação (cliente)..... | 70 |
| Figura 46 – Esquemático do circuito simulado..... | 72 |
| Figura 47 – Esquemático completo do circuito..... | 72 |
| Figura 48 – Circuito montado..... | 73 |
| Figura 49 – Circuito em funcionamento carregando o telefone celular..... | 73 |
| Figura 50 – Visualização inicial da página web do servidor..... | 75 |
| Figura 51 – Mensagem de erro ao selecionar o botão “delete” tem uma rota selecionada..... | 76 |
| Figura 52 – Erro ao pressionar o botão “atualizar” quando não há rota selecionada..... | 76 |
| Figura 53 – Rota selecionada de acordo com o usuário cadastrado..... | 77 |
| Figura 54 – Informações do veículo..... | 77 |
| Figura 55 – Definindo o usuário e o endereço de serviço no celular..... | 78 |
| Figura 56. Criando senha do administrador do MySQL..... | 87 |
| Figura 57. Apache em funcionamento..... | 88 |
| Figura 58. Arquivo de configuração do servidor..... | 91 |
| Figura 59 – Arquivo de configuração para detectar dispositivos..... | 95 |
| Figura 60 – Listando os dispositivos conectados..... | 96 |
| Figura 61 – Android SDK Manager..... | 97 |

| | |
|--|-----|
| Figura 62 – Licença dos pacotes selecionados..... | 98 |
| Figura 63 – Celular reconhecido pelo Android Studio..... | 99 |
| Figura 64 – Medida realizada para 3G – SIMET..... | 99 |
| Figura 65 – Medida realizada para WLAN – SIMET..... | 100 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|---------|---|
| 1G | Primeira Geração |
| 2G | Segunda Geração |
| 3G | Terceira Geração |
| 3GPP | Third Generation Partnership Project |
| 4G | Quarta Geração |
| AMPS | Advanced Mobile Phone System |
| API | Application Programming Interface |
| APM | Applications Performance Management |
| C/A | Córses Acquisition |
| CDMA | Code Division Multiple access |
| CIs | Circuitos Integrados |
| CPD | Centro de Processamento de Dados |
| DSSS | Direct Sequence Spread Spectrum |
| DT | Dispositivos sem fio Terminais |
| EB | Estação Base |
| EC | Estação Central |
| EDGE | Enhanced Data rates for GSM Evolution |
| DF | Distrito Federal |
| FDMA | Frequency Division Multiple Access |
| FGA | Faculdade do Gama |
| FHSS | Frequency Hopping Spread Spectrum |
| FM | Frequency Modulation |
| GLONASS | Global Navigation Satellite System |
| GSM | Global System for Mobile communications |
| GNSS | Global Navigation Satellite Systems |
| GPRS | General Packet Radio Service |
| GPS | Sistema de Posicionamento Global |
| HTML | HyperText Markup Language |
| HTTP | Hypertext Transfer Protocol |
| HTTPS | Hyper Text Transfer Protocol Secure |
| IDE | Integrated Development Environment |
| IEEE | Institute of Electrical and Electronics Engineers |

| | |
|-------------|---|
| IP | Internet Protocol |
| ISM | Industrial, Scientific, and Medical bands |
| LAMP | Linux, Apache, MySQL, PHP |
| LED | Light Emissor Diode |
| LTE | Long Term Evolution |
| MAMP | Mac OS, Apache, MySQL, PHP |
| MBWA | Móbile Broadband Wireless |
| MIMO | Multiple Input Multiple Output |
| Navstar GPS | NAVigation System with Time And Ranging – Global Positioning System |
| OFDM | Orthogonal Frequency Division Multiplexing |
| OSM | OpenStreetMaps |
| P | Precise ou Protected |
| PHP | Hypertext Preprocessor |
| PRN | Pseudo Random Noise |
| RF | Radiofrequência |
| SIG | Sistemas de Informações Geográficas |
| SIMET | Sistema de Medição de Tráfego Internet |
| SQL | Structured Query Language |
| TCP | Transmission Control Protocol |
| TDMA | Time Division Multiple Access |
| UDP | User Datagram Protocol |
| UHF | Ultra High Frequency |
| UMTS | Universal Mobile Telecommunications System |
| UnB | Universidade de Brasília |
| VoIP | vídeo e voz sobre IP |
| WAMP | Windows, Apache, MySQL, PHP |
| WCDMA | Wideband Code Division Multiple Access |
| WiFi | Wireless Fidelity |
| WLAN | Wireless Local Area Networks |
| WWAN | Wireless Wide Area Networks |

LISTA DE TABELAS

| | |
|---|-----|
| Tabela 01 – Padrões WLAN e suas características (JAMHOUR, 2013)..... | 28 |
| Tabela 02 – Características do padrão AMPS (Sverzut, 2005)..... | 30 |
| Tabela 03 – Características do padrão TDMA (Sverzut, 2005)..... | 31 |
| Tabela 04 – Características do padrão CDMA (Sverzut, 2005)..... | 31 |
| Tabela 05 – Critérios para seleção da melhor API..... | 35 |
| Tabela 06 – Análise de confiabilidade..... | 36 |
| Tabela 07 – Análise de usabilidade..... | 36 |
| Tabela 08 – Análise de licença..... | 37 |
| Tabela 09 – Análise de suporte..... | 37 |
| Tabela 10 – Cálculo dos critérios da API..... | 37 |
| Tabela 11 – Dados de precisão coletados em diferentes configurações do celular | 74 |
| Tabela 12 – Dados medidos e calculados para a rede 3G..... | 100 |
| Tabela 13 – Dados medidos e calculados para a rede WLAN..... | 101 |

SUMÁRIO

| | |
|---|-----------|
| 1 INTRODUÇÃO | 15 |
| 1.1 Contextualização | 15 |
| 1.2 Motivações | 17 |
| 1.3 Objetivos | 17 |
| 1.1 Objetivos Gerais | 17 |
| 1.2 Objetivos Especificos | 17 |
| 1.3 Organização do Trabalho | 17 |
| 2 FUNDAMENTAÇÃO TEÓRICA | 20 |
| 2.1 GNSS (<i>Global Navigation Satellite Systems</i>) | 20 |
| 2.1.1 GPS | 20 |
| 2.1.1.1 Segmento Espacial | 21 |
| 2.1.1.2 Segmento de Controle | 22 |
| 2.1.1.3 Segmento do Usuário | 23 |
| 2.1.2 GLONASS | 24 |
| 2.1.2.1 Segmento Espacial | 24 |
| 2.1.2.2 Segmento de Controle | 25 |
| 2.1.2.3 Segmento do Usuário | 25 |
| 2.2 Tipos de Rede | 25 |
| 2.2.1 WLAN | 26 |
| 2.2.1.1 Padroes WLAN | 27 |
| 2.2.1.2 Técnicas de Modulação | 28 |
| 2.2.2 WWAN | 29 |
| 2.2.2.1 Primeira geração da tecnologia celular – 1G | 30 |
| 2.2.2.2 Segunda geração da tecnologia celular – 2G | 30 |
| 2.2.2.3 Segunda/Terceira geração da tecnologia celular – 2.5G | 31 |
| 2.2.2.4 Terceira geração da tecnologia celular – 3G | 32 |
| 2.2.2.5 Quarta geração da tecnologia celular – 4G | 32 |
| 2.3 SIG (Sistemas de Informações Geograficas) | 33 |
| 2.3.1 APIs | 34 |
| 2.3.2 Google Maps | 37 |
| 2.4 LAMP(Linux, Apache, MySQL, PHP) | 39 |
| 2.5 Android Studio | 40 |
| 3 METODOLOGIA | 42 |
| 3.1 Métricas | 42 |
| 3.1.1 Latência | 42 |
| 3.1.2 Jitter | 43 |
| 3.1.3 TCP | 43 |
| 3.1.4 UDP | 45 |
| 3.2 Estudo de Caso | 46 |
| 4 RESULTADOS | 55 |
| 4.1 Implementação da Proposta | 56 |
| 4.1.1 Servidor de Rastreamento no Raspiberry PI | 56 |
| 4.1.2 Agente de Rastreamento Via Dispositivo Móvel | 64 |
| 4.1.3 Alimentação do Dispositivo | 71 |
| 4.2 Validação Experimental da Proposta | 74 |
| 4.2.1 Resultados Esperados para o Rastreamento | 74 |
| 4.2.2 Análise dos Resultados | 75 |
| 5 CONCLUSÃO | 80 |
| 5.1 Trabalhos Futuros | 82 |
| REFERENCIAS BIBLIOGRAFICAS | 83 |
| ANEXOS | 86 |

Parte I
Introdução

1. INTRODUÇÃO

Neste capítulo, será abordado de maneira geral o assunto tratado em todo projeto. A contextualização introduz um pouco da história vinculada ao tema do trabalho. No tópico das motivações, tratam-se das problemáticas que levaram a criação do projeto. Os objetivos também são tratados deixando de maneira clara os assuntos que o projeto deve abordar e por fim há um tópico explicitando a organização do trabalho, descrevendo o que cada parte do trabalho discute.

1.1. CONTEXTUALIZAÇÃO

Um dos mais antigos dilemas da humanidade consiste em saber qual a localização que se encontra determinado lugar com a finalidade de alcançá-lo. Para isso deve-se ter conhecimento da localidade atual para assim atingir o objetivo da localidade futura, através do posicionamento georeferenciado.

A necessidade de saber a localidade através do posicionamento global é relevante desde a época das grandes navegações em meados do século XV, em que os grandes navegadores possuíam apenas a orientação dos astros e estrelas no céu, utilizando aparelhos fabricados há muito tempo, como o astrolábio e a bússola. Com isso, muitas tecnologias vêm sendo desenvolvidas desde então, com a finalidade de melhorar a localização e a movimentação no planeta, como é o caso do sistema de posicionamento global (GPS), que é uma ferramenta indispensável para várias aplicações.

Em integração com o GPS, tem-se o sistema de telefonia móvel que evoluiu significativamente desde os primórdios dessa tecnologia. No caso do 1G (primeira geração) onde os dispositivos móveis eram puramente analógicos, até os dias atuais com a tecnologia 4G (quarta geração) onde se tem uma taxa de transferência de dados bem superior às antecessoras.

Atualmente, a tecnologia GPS é bem acessível para a população consumidora de automóveis, em que os veículos comercializados possuem, em sua maioria, este acessório como opcional. Porém, o acesso a esse tipo de tecnologia ainda é caro e possui uma aplicabilidade restrita apenas para localização e não para um monitoramento externo.

O rastreamento veicular é necessário neste contexto que se aplica o monitoramento externo, onde a integração do GPS com sistema de transferência de dados é necessária. Com isso é possível realizar o controle de deslocamento de frotas veiculares, acompanhamento em tempo real da posição dos veículos e no auxílio da recuperação de veículos roubados ou furtados.

Apesar das inúmeras aplicações, o rastreamento veicular se trata ainda de uma tecnologia de alto custo. Existe ainda, a insegurança do serviço de monitoramento ser realizado por terceiros, onde um bem durável e de alto valor como um veículo, é de responsabilidade de pessoas desconhecidas, assim como a segurança dos ocupantes do veículo.

1.2. MOTIVAÇÕES

Um dos maiores problemas de segurança, nos dias atuais, está no roubo ou furto de veículos automotores, um bem durável, de alto custo e que é responsável por altos investimentos em seguradoras ou serviços de rastreamento, por parte do proprietário, para que sejam evitados esses tipos de imprevistos que causam altos prejuízos.

O insucesso da recuperação de veículos roubados, mesmo com o esforço de seguradoras e da Polícia Civil, advém da facilidade de desmanche de um carro ou moto, por exemplo, e da difícil localização desses veículos no período em que são abandonados para serem “esfriados”. Esse processo ocorre para que o criminoso tenha certeza de que o veículo não é rastreado. Segundo a polícia, “é possível desmanchar um carro inteiro em cerca de dez minutos, o que torna a localização do veículo ainda mais complicada” (Globo, 2013).

O Distrito Federal (DF) é a oitava unidade da federação com maior incidência de roubos ou furtos de veículos, o que, proporcionalmente, leva a terceira posição na probabilidade de um veículo ser furtado ou roubado (Roubadosbr, 2013). E no Gama, assim como na Faculdade do Gama da Universidade de Brasília (FGA-UnB), também ocorrem esses fatos, constantemente, com veículos de alunos, professores ou funcionários. Houve casos em que em um intervalo de duas semanas relatou-se três ocorrências de roubos a veículos. Como esse cenário vem ocorrendo constantemente, a proposta deste projeto visa à implementação de um sistema de rastreamento veicular

através das redes WLAN (*Wireless Local Area Networks*) e WWAN (*Wireless Wide Area Networks*), com o uso de softwares livres.

1.3. OBJETIVOS

1.3.1. Objetivos Gerais

O objetivo deste trabalho é a implementação de um sistema de Rastreamento Veicular com o auxílio de Dispositivos Móveis, através de redes *WLAN* e *WWAN*, configurados com o uso de tecnologias em software livre, que possa auxiliar os usuários do campus UnB-Gama no rastreo remoto de eventuais furtos e/ou roubos de seus veículos.

1.3.2. Objetivos específicos

Dentro dos objetivos específicos cabem os seguintes itens:

- Criar um aplicativo para um dispositivo móvel, capaz de enviar as informações necessárias e informar a sua localização e outras informações básicas como velocidade que se encontra o veículo, a precisão da localização informada e a data e hora que foi realizado o rastreamento;
- Criar um servidor web embarcado em um *Raspberry PI* que possui o sistema operacional *Rasbian* (oriundo do sistema Debian-Linux) instalado. Este servidor será capaz de mostrar os dados informados pelo dispositivo móvel em um *website* para que o usuário consiga ver a localização real de seu veículo.
- Definir um ambiente experimental capaz de realizar testes a fim de confirmar o funcionamento do sistema implementado.
- Coletar de forma repetitiva, dados a serem enviados e recebidos via plataforma cliente/servidor pelas redes *WLAN/WWAN* e apresentar a análise e resultados acerca do rastreamento veicular.

1.4. ORGANIZAÇÃO DO TRABALHO

O trabalho está dividido em cinco partes, Parte 1: Introdução, Parte 2: Fundamentação teórica, Parte 3: Metodologia, Parte 4: Resultado e Parte 5: Conclusão.

Na segunda parte tem-se o capítulo 2, que apresenta a fundamentação teórica, abordando a teoria para o bom entendimento de todo o trabalho, citando as tecnologias GPS existentes e os tipos de redes e suas características. Trata também do sistema de informação geográfica, mais especificamente, cita o *Google Maps* e suas funções, apresentando os critérios que levaram a escolha do mesmo. É apresentado o sistema LAMP (Linux, Apache, MySQL e PHP), citando pequenos tutoriais em anexo afim de facilitar a utilização de cada um deles, assim como o *Android Studio*.

No capítulo 3, encontram-se as métricas relacionadas a testes feitos com os dois tipos de rede, em conjunto com a avaliação relacionada ao estudo de caso proposto.

No capítulo 4, é apresentado o servidor implementado, definindo detalhadamente os agentes que o compõe, assim como as modificações realizadas no *layout* do servidor. Na parte do cliente, é definido e explicado todo código que, em conexão com o servidor, deixa claro o funcionamento de todo sistema. É apresentado, também, o dispositivo projetado com a finalidade de carregar a bateria do telefone celular mantendo, assim, a contínua comunicação entre servidor e cliente. Por fim, apresentam-se os resultados relacionados ao bom funcionamento do projeto, bem como o *layout* do servidor implementado, discutindo assim os resultados apresentados.

Na quinta e última parte do trabalho, é apresentada a conclusão do projeto, junto com a proposta de trabalhos futuros relacionados a esta monografia de conclusão de curso.

Parte II
Fundamentação Teórica

2. FUNDAMENTAÇÃO TEÓRICA

Este capítulo trata das tecnologias existentes dos sistemas de posicionamento global (GPS e GLONASS) e dos tipos de rede utilizados (WLAN e WWAN), citando suas principais características. Explicam-se os motivos pelos quais o Google Maps foi escolhido entre as APIs para a aplicação do Sistema de Informação Geográfica. Por fim, cita-se o sistema de software utilizado (LAMP) e o Android Studio, abordando, de maneira objetiva, uma introdução sobre cada um deles.

2.1 GNSS (*Global Navigation Satellite Systems*)

Os sistemas de navegação por satélite tornaram-se parte integrante de todas as aplicações onde a mobilidade desempenha um importante papel. O termo GNSS é utilizado para descrever os sistemas de posicionamento por satélites artificiais que possuem cobertura mundial. Atualmente, os únicos sistemas GNSS que podem ser considerados operacionais são: o sistema Navstar GPS (*NAVigation System with Time And Ranging – Global Positioning System*) e o sistema GLONASS (*Global Navigation Satellite System*). O sistema GNSS europeu Galileo se encontra em desenvolvimento e dentre os citados é o único que possui garantias de operação, devido ao fato de aceitar a combinação de sinais GNSS (Heinrichs, 2005).

O sistema GNSS possui basicamente três segmentos: o segmento espacial, o segmento de controle e o segmento final que é de usuário. Os segmentos citados acima são parecidos nas três tecnologias (GPS, GLONASS e Galileo), e todas juntas compõem o GNSS.

2.1.1. GPS

O NAVSTAR GPS é um sistema de rádio navegação por satélite que fornece as coordenadas precisas de posicionamento e informações sobre navegação e tempo. O sistema GPS impactou consideravelmente temas como posicionamento, navegação, sincronização e aplicações de monitoramento, fornecendo especialmente sinais codificados de satélite que podem ser processados através

de um receptor GPS (Hofmann, 2001). Pode-se dizer que esta é a tecnologia mais utilizada dentre as citadas atualmente.

2.1.1.1. Segmento Espacial

O segmento espacial é constituído pelos satélites GPS, como mostrado na Figura 01. Estes satélites enviam sinais de rádio do espaço para as estações de monitoramento. Os satélites estão dispersos em seis planos orbitais em órbitas quase circulares, com uma altitude de cerca de 20.200 km acima da superfície da Terra, inclinados por 55 graus em relação à linha do equador e com períodos orbitais de cerca de 11 horas e 58 minutos. Com esta configuração, a constelação do sistema GPS garante que no mínimo quatro satélites GPS sejam visíveis em qualquer ponto da superfície terrestre, a qualquer hora (MONICO, 2000).

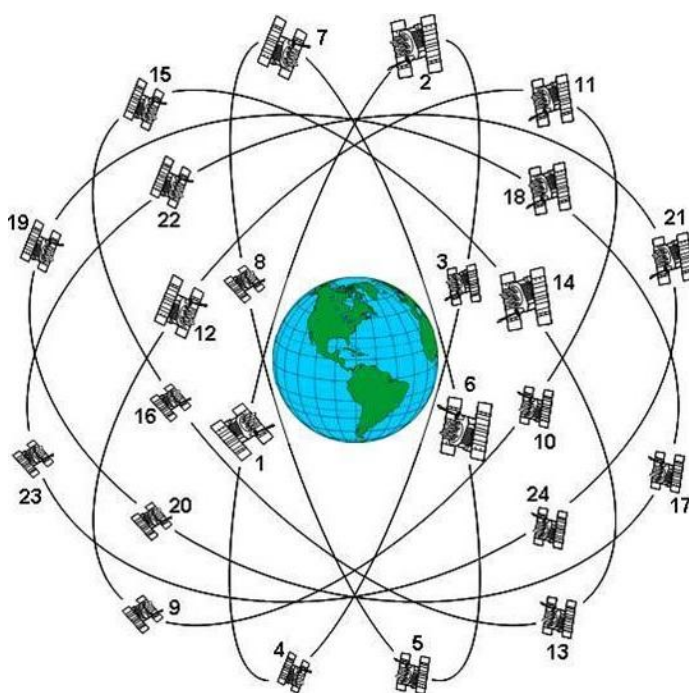


Figura 01 – Constelação sistema GPS (NINJA, 2010)

Em resumo, os sinais gerados a bordo dos satélites são baseados ou derivados da frequência fundamental f_0 de 10.23 Mhz (Hofmann, 2001). O sinal é controlado por relógio atômico e tem estabilidade no intervalo de 10^{-13} . Duas portadoras de sinais (banda L), denominadas L1 e L2 são geradas por fatores multiplicativos de f_0 . A portadora L1 emite ondas de 1575.42 Mhz (10.23×154) e

comprimento de onda de $\lambda \cong 19$ cm. Já a portadora L2 emite ondas de 1227.60 Mhz (10.23×120) e comprimento de onda de $\lambda \cong 24$ cm.

Os códigos PRN (*Pseudo Random Noise*) são modulados sobre as duas portadoras citadas. Um PRN consiste em uma sequência binária (0 e 1 ou +1 e -1) com características aleatórias. O código C/A (*Coarse Acquisition*) é transmitido a uma razão de 1.023 Mhz, modulado somente sobre a portadora L1 e se repete a cada milissegundo. O código P (*Precise or Protected*) é transmitido na razão de 10.23 Mhz modulado nas portadoras L1 e L2, com período de 267 dias. Além desses códigos, existe ainda o código Y, gerado a partir de uma equação secreta (*Anti-spoofing*). A portadora L1 é modulada com os códigos C/A e P (ou Y) (SEGANTINE, 1999).

2.1.1.2. Segmento de Controle

O segmento de controle consiste em um sistema de estações de monitoramento localizadas em todo mundo. É composto por cinco estações monitoras (Hawaii, Kwajalein, Ascension Island, Diego Garcia, Colorado Springs) sendo a estação de controle central em Colorado Springs (Figura 02). As principais tarefas dessas estações é controlar continuamente o sistema de satélites, determinar o tempo GPS, calcular as correções dos relógios dos satélites e atualizar as mensagens de navegação de cada satélite.

Estas estações são equipadas com receptores de dupla frequência e com oscilador externo de alta precisão, os quais rastreiam todos os satélites visíveis e transmitem os dados para a estação de controle central. Esses dados são processados para determinar a órbita dos satélites e as correções dos relógios, com o intuito de atualizar periodicamente as mensagens de navegação.

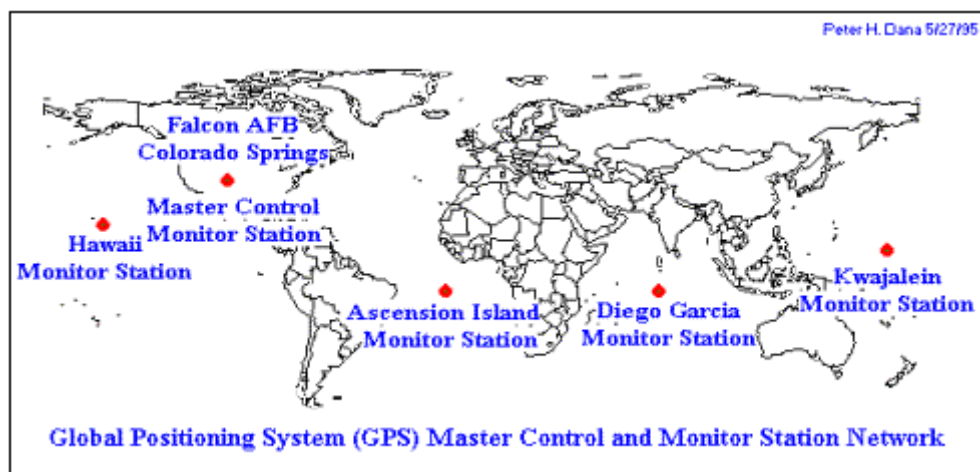


Figura 02 – Distribuição das estações de rastreamento no mundo (PETER, 2000)

2.1.1.3. Segmento do Usuário

O segmento GPS do usuário é composto pelos receptores GPS que convertem os sinais dos satélites em posição, velocidade e o tempo. O segmento do usuário ainda pode ser dividido em civil e militar. Para o uso civil, existe restrição quanto à precisão e equipamentos para usuários autorizados, que permitem uma precisão da ordem de milímetros. Esses receptores são utilizados em estudos geodésicos. A Figura 03 mostra o sistema completo, com todos os segmentos descritos acima.

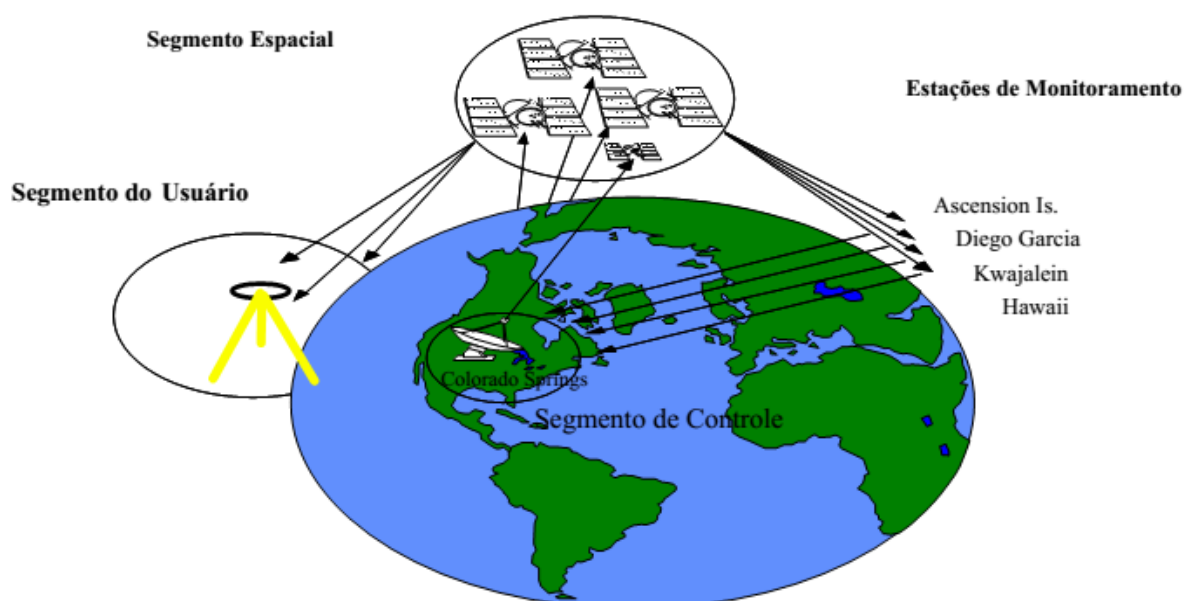


Figura 03 – Segmentos do sistema de GPS(CEUB/ICPD, 2014).

2.1.2. GLONASS

O GLONASS tem como principal objetivo informar a posição do usuário em aspecto tridimensional, velocidade e o tempo em todo o mundo. Assim como o sistema GPS, o sistema GLONASS possui três segmentos: o espacial, o de controle e de usuários, descritos a seguir (GLONASS-ICD, 2002).

2.1.2.1. Segmento Espacial

O segmento espacial do GLONASS é constituído por uma constelação de 24 satélites ativos e um reserva, como mostrado na Figura 04. Os satélites estão divididos em três planos orbitais, com uma altitude de cerca de 19.100 km acima da superfície da Terra, inclinados por 45 graus e com períodos orbitais de aproximadamente 11 horas e 15 minutos.

O sistema GLONASS também transmite sinais em duas bandas (L1 e L2), mas, diferentemente do sistema GPS, cada satélite apresenta sua própria frequência, definidas a partir de uma frequência central dos canais (MONICO, 2000).

Existem dois códigos PRN: o código C/A disponível para todos os usuários civis e transmitido a 0.511 Mhz; e o código P disponível para usuários autorizados e transmitido a 5.11 Mhz. Ambos são modulados nas portadoras L1 e L2, sendo que esta é modulada somente pelo código P.

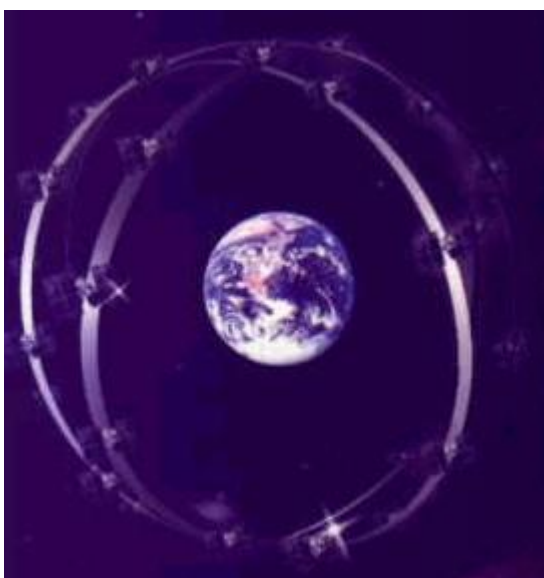


Figura 04 – Constelação do sistema GLONASS (GLONASS, 2002)

2.1.2.2. Segmento de Controle

O segmento de controle, ao contrário do sistema GPS, é totalmente Russo, constituído pelo sistema de controle central na região de Moscou e pelas estações de comando (St. Petersburg, Temopol, Eniseisk e Komsomolsk-na-Amure) [GLONASS, 2002].

As estações de comando têm como principal tarefa rastrear os satélites GLONASS obtendo informações de cada um deles. Tais informações são enviadas para a estação de controle, onde são processados para determinar o estado do relógio do satélite e de suas órbitas. As estações de comando, depois de receber as mensagens da estação de controle, as transmitem novamente aos satélites para que sejam atualizadas as informações de navegação.

Todo sistema é sincronizado através do sincronizador central que se encontra na estação de controle, composta por um relógio atômico de alta precisão, e juntamente com os relógios dos satélites (também relógios atômicos), formam o sistema de tempo do GLONASS.

2.1.2.3. Segmento do Usuário

O sistema do usuário consiste nas antenas e nos receptores que tem como funções principais determinar a posição, velocidade e tempo com precisão. Os instrumentos mais atuais do GLONASS são baseados numa larga escala de CIs (Circuitos Integrados) e processamento digital de sinais, que são mais compactos que as primeiras versões.

A maioria dos fabricantes produzem receptores capazes de rastrear os satélites GPS e GLONASS, o que faz com que o sistema trabalhe em conjunto com as duas tecnologias, tornando assim o sistema mais eficiente. Este é o caso da tecnologia utilizada para localização georreferenciada neste projeto.

2.2 TIPOS DE REDE

As redes de internet possibilitam a comunicação entre dispositivos, sejam eles fixos ou móveis. A comunicação se baseia na troca de informações, através do envio e recebimento de dados entre os dispositivos através da internet. Existem dois tipos de redes: as redes WLAN e WWAN. A WLAN é uma rede de internet local sem fio enquanto a WWAN é uma rede de longas distâncias sem fio.

O estudo destas tecnologias de rede é relevante para o projeto, pois indica as vantagens e desvantagens relacionadas ao cenário proposto no Capítulo 4, onde são avaliados fatores como a mobilidade do dispositivo, a taxa de transferência de dados, o custo do serviço e a faixa de operação. Na Figura 5, observam-se as características das redes em relação aos fatores mencionados.

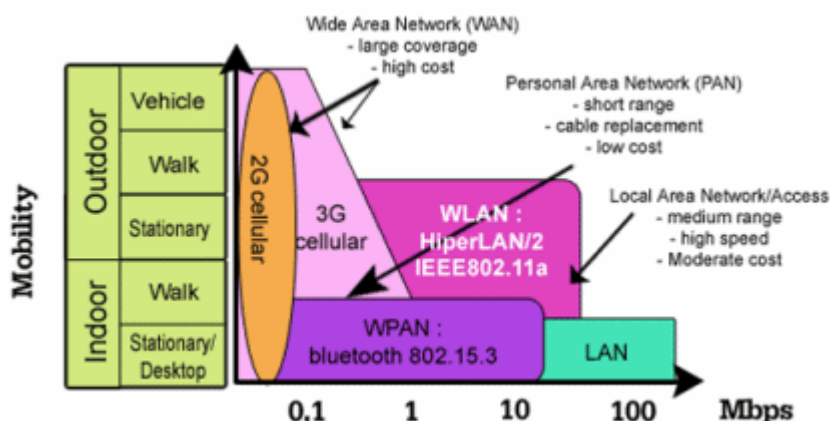


Figura 05 – Tipos de rede e suas características (Wifi & Sécurité,2006)

2.2.1. WLAN

A WLAN, padrão IEEE (*Institute of Electrical and Electronics Engineers*) 802.11 a/b/g/n, atende a uma abrangência de área de cobertura entre 60 e 100 metros *indoor* e 100 a 300 metros *outdoor*, operando na faixa ISM (*Industrial, Scientific, and Medical bands*) na frequência de 2,4 GHz (IEEE 802.11 b/g/n) ou 5 GHz (IEEE 802.11 a/n), e com taxas nominais de transferência de dados de 1 a 2 Mbps (802.11), 11 a 54 Mbps (IEEE 802.11 a/b/g) e até 600 Mbps (IEEE 802.11n, de acordo com a configuração espacial das antenas) (O'HARA, 1999).

Vantagens:

- A maioria dos computadores vem com a tecnologia *WLAN* integrada;
- As conexões são encontradas em residências, escritórios, hotéis, cafeterias e aeroportos;
- Permite a mobilidade do usuário dentro do alcance da rede sem fio;
- Fácil para expansão em pequenas empresas e conexão de mais usuários sem a adição de fios;

Desvantagens:

- A área de cobertura é fixa e geralmente pequena.

- Como usa ondas de rádio, a potência do sinal pode, às vezes, ficar comprometida devido à interferências oriundas de outros sinais de rádio e, ou obstruções.

A Figura 6 exemplifica o funcionamento de uma rede WLAN, onde EB é a *Estação Base* e EC é a *Estação Central* que monitora, gerencia, controla a segurança lógica e estabelece a conexão da rede de dados entre a EB e os *Dispositivos sem fio Terminais* (DT).

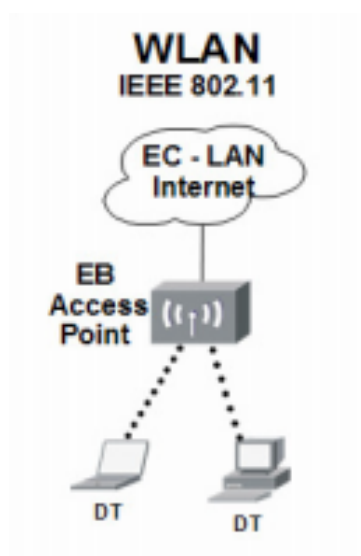


Figura 06 – Topologia das redes WLAN (SILVA, 2008).

2.2.1.1 Padrões WLAN

Os padrões WLAN, também conhecidos como WiFi (*Wireless Fidelity*), são derivados da família IEEE 802.11 e diferem em relação a técnica de modulação, a máxima velocidade de transmissão e a faixa de frequência de operação. (JAMHOUR, 2013)

Na tabela 01, são indicados os padrões com suas respectivas características.

Tabela 01 – Padrões WLAN e suas características. (JAMHOUR, 2013)

| Dados | 802.11a | 802.11b | 802.11g | 802.11n |
|---|---------|----------|---------------|----------------|
| Velocidade de transferência máxima (Mbps) | 54 | 11 | 54 | 600 |
| Número de taxas de transmissão | 8 | 4 | 12 | 576 |
| Alcance máximo (m) | 23 | 30 | 46 | 46 |
| Modulação | OFDM | DSSS,CCK | DSSS,CCK,OFDM | DSSS,CCK,OFDM+ |
| Frequência portadora (GHz) | 5 | 2.4 | 2.4 | 2.4 e 5 |
| Número de antenas | 1 | 1 | 1 | 4 ou mais |
| Largura do canal (MHz) | 20 | 20 | 20 | 20 ou 40 |
| Número de canais | 23 | 3 | 3 | 26 |
| Eficiência espectral (bps/Hz) | 2.7 | 0.55 | 2.7 | 15 a 30 |

2.2.1.2 Técnicas de Modulação

Os padrões WLAN utilizam algumas técnicas de modulação que tem a finalidade de ajustar o sinal transmitido para a frequência padrão. As técnicas de modulação definem uma eficiência espectral de cada padrão e são (JAMHOUR, 2013):

- FHSS (*Frequency Hopping Spread Spectrum*): é utilizado em tecnologias celular. Opera em uma banda de frequência que é dividida em 79 canais de frequência com 1 MHz de largura. Baixa largura de banda com sincronismo na mudança de frequência das estações. É muito sensível às interferências que ocasionam em redução da taxa de operação.
- DSSS (*Direct Sequence Spread Spectrum*): é utilizado nos padrões IEEE 802.11 b/g/n. Opera em uma banda de frequência que é dividida em 14 canais de 22 MHz com sobreposição parcial de 11 canais. É bastante

sensível a interferências, que causam a redução da taxa de operação. Na Figura 07, observam-se os canais não sobrepostos.

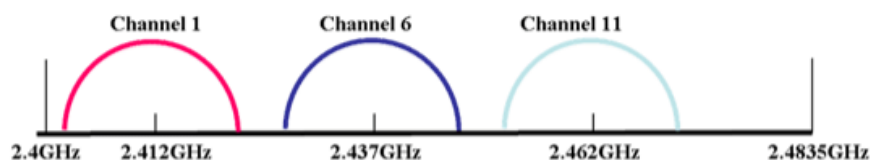


Figura 07 – Canais não sobrepostos. (JAMHOUR, 2013)

- OFDM (*Orthogonal Frequency Division Multiplexing*): é utilizado nos padrões IEE 802.11 a/n na frequência de 5 GHz .Opera em uma banda de frequência que é dividida em canais de 20 MHz, que são subdivididos em 52 canais de aproximadamente 300 KHz. É pouco sensível á interferências, o que o torna bastante robusto na taxa de operação.

2.2.2. WWAN

O WWAN é uma rede sem fio que possibilita a conexão de usuários e dispositivos a redes remotas publicas e, ou privadas que tenham abrangência de grande distancia geográfica. O *mobile Broadband Wireless* (MBWA), padrão IEEE 802.20, que possui mobilidades com taxas de transferências de até 1 Mbps e opera em 3,5 GHz, atende às necessidades das redes WWAN.

A figura 08 exemplifica o funcionamento de uma rede WWAN, onde EB é a *Estação Base*, EC é a *Estação Central* que monitora, gerencia, controla a segurança lógica e estabelece a conexão da rede de dados entre a EB e os *Dispositivos sem fio Terminais* (DT).

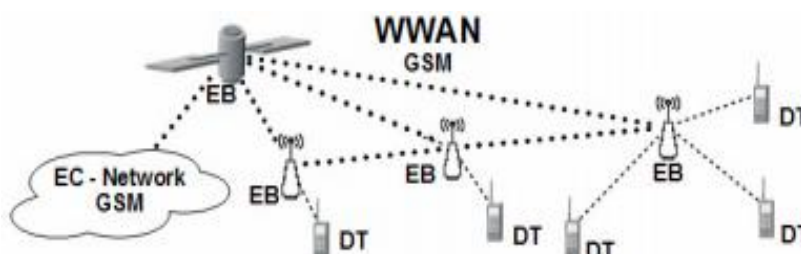


Figura 08 – Topologia de uma rede WWAN (SVERZUT, 2005).

2.2.2.1 Primeira geração da tecnologia celular – 1G

Por ser inovadora, a primeira geração da tecnologia celular (1G) causou uma grande revolução nos meios de comunicações. Antes desta tecnologia existir, o sistema de telefonia móvel possuía uma capacidade muito baixa de tráfego, o que ocasionava em custos muito elevados. Eram sistemas puramente analógicos, e utilizavam modulação FM (*Frequency Modulation*), onde a voz era transmitida em sinais RF (Radiofrequência) em faixas UHF (*Ultra High Frequency*). Com tais especificações, foi criado o padrão AMPS (*Advanced Mobile Phone System*). Este padrão utiliza a tecnologia de acesso múltiplo por divisão de frequência, o FDMA (*Frequency Division Multiply Access*). A tabela 2 mostra as especificações do padrão FDMA. O FDMA é um método de acesso universal, mas que pode ser utilizado de forma isolado, no caso o AMPS.

Tabela 02- Características do padrão AMPS (Sverzut, 2005).

| Tecnologia de múltiplo acesso | FDMA |
|--|------------------------|
| Largura do canal | 30 KHz |
| Usuário por canal | 1 |
| Largura da banda disponível | 25 MHz (894 – 869 MHz) |
| Espaçamento entre os canais | 45 MHz |
| Modulação do canal de voz | FM |
| Numero de canais (controle e tráfego) | 832 |

2.2.2.2 Segunda geração da tecnologia celular – 2G

Devido ao crescimento da tecnologia celular juntamente com a integração dos métodos do AMPS com o FDMA, e a uma necessidade significativa na expansão nos Estados Unidos, surgiu a segunda geração de telefonia celular. A melhoria mais significativa nessa geração foi o aumento de sua capacidade que atendeu as necessidades dos Estados Unidos juntamente com a padronização do sistema, suprimindo as necessidades Europeias.

O sistema deixava então de ser analógico e adentrava no mundo digital (Rappaport, 2009). Nos Estados Unidos foram implementados três novos padrões dessa geração:

- IS-54: AMPS digital ou D-AMPS;

- IS-136: conhecido através da tecnologia de Acesso Múltiplo por Divisão de Tempo (TDMA – *Time Division Multiple Access*), as características deste padrão estão apresentadas na Tabela 03;
- IS-95: conhecido através da tecnologia de Acesso Múltiplo por Divisão de Código (CDMA – *Code Division Multiple access*). Na Tabela 04, apresentam-se as características deste padrão.

Tabela 03- Características do padrão TDMA (Sverzut, 2005).

| Tecnologia de múltiplo acesso | TDMA |
|--|-------------------------------|
| Largura do canal | 30 KHz |
| Usuário por canal | 3 (7,95 Kbits/s ou 6 Kbits/s) |
| Largura da banda disponível | 25 MHz (894 – 869 MHz) |
| Espaçamento entre os canais | 45 MHz |
| Modulação do sinal de voz | $\pi/4$ shifted DQPSK |
| Numero de canais (controle e tráfego) | $3 \cdot 832 = 2496$ |

Tabela 04- Características do padrão CDMA (Sverzut, 2005).

| Tecnologia de múltiplo acesso | CDMA |
|--|--|
| Largura do canal | 1,25 MHz |
| Usuário por canal | Depende da relação sinal/ruído adotado pela rede |
| Largura da banda disponível | 25 MHz (894 – 869 MHz) |
| Espaçamento entre os canais | 45 MHz |
| Modulação do sinal de voz | DQPSK |
| Numero de canais (controle e tráfego) | Depende da relação sinal/ruído adotado pela rede |

Na Europa foi criado o sistema GSM (*Global System for Mobile communications*), que visava sua uniformização. Este sistema passou a ser o principal representante da segunda geração, permitindo o aumento na produção de equipamentos e conseqüentemente atraindo grandes empresas para ingresso no mercado de telecomunicações.

2.2.2.3 Segunda/Terceira geração da tecnologia celular – 2,5G

Os sistemas digitais tiveram grandes melhorias em seus sistemas, mas foram projetados especificamente para transportar voz, o que não atendia a nova realidade de vida e trabalho da sociedade mundial que a cada dia mais vinha utilizando a internet. As redes e os padrões GSM foram atualizados para se adaptar a essa nova realidade, criando a geração 2,5 da telefonia celular.

Este padrão é constituído pela evolução do CDMA, posteriormente atualizado para terceira geração (3G), e o GPRS (*General Packet Radio Service*).

O padrão TDMA e GSM também sofreram evolução, criando o padrão EDGE (*Enhanced Data rates for GSM Evolution*).

2.2.2.4 Terceira geração da tecnologia celular – 3G

O padrão UMTS (*Universal Mobile Telecommunications System*) utiliza a tecnologia WCDMA (*Wideband Code Division Multiple Access*), que representa a evolução das operadoras GSM, GPRS e EDGE. Essa geração teve como modificação a compatibilidade com a interface IS-41, utilizada pelas operadoras TDMA e CDMA. As especificações foram definidas através da parceria da terceira geração da telefonia celular (3GPP – *Third Generation Partnership Project*).

Os principais diferenciais nas arquiteturas que evoluíram da arquitetura 1G estão relacionados às interfaces e protocolos utilizados pela RF. No caso da interface UMTS, as camadas de protocolos são divididas em três: a camada física, a camada enlace e a camada rede. A estrutura dos canais também é formada por três tipos de canal: Canal Lógico, que identifica o tipo de informação que está sendo transportada; canal de transporte, responsável por identificar as características de transporte; e o canal físico que transporta os canais pela interface aérea.

2.2.2.5 Quarta geração da tecnologia celular – 4G

Atualmente, utiliza-se também a quarta geração de telefonia (4G), ou LTE (*Long Term Evolution*), que é um dos últimos avanços alcançados até o momento quando se trata de tecnologia móvel.

A tecnologia LTE aperfeiçoou a eficiência espectral, utilizando uma arquitetura de comunicação voltada apenas para o protocolo IP (*Internet Protocol*) para voz e dados. Com isso, há um aumento na velocidade que pode chegar até 100 Mbps (3GPP, 2010). O sistema LTE atinge taxas elevadas de transferência pelo fato de utilizar o MIMO (*Multiple Input Multiple Output*), proporcionando um desempenho elevado na transmissão já que faz o uso de várias antenas para transmissão e recepção de dados.

2.3 SIG (SISTEMAS DE INFORMAÇÕES GEOGRÁFICAS)

Os Sistemas de Informações Geográficas (SIG) são ferramentas computacionais para Geoprocessamento que realizam análises complexas integrando dados convencionais de expressão espacial de várias fontes. Ainda, é possível a criação de banco de dados geo-referenciados. Ou seja, o termo SIG é aplicado para sistemas que operam de forma computacional a análise de dados geográficos. A capacidade de reunir grande número de dados torna essa ferramenta essencial quando se fala de manipulação de informações geográficas (Geoprocessamento).

As ferramentas para SIG possuem algumas características:

- Capacidade de coletar dados espaciais diversos e processá-los. Estes dados são: levantamentos de campo (provenientes do sistema GPS), mapas, fotogrametria, sensoriamento remoto, entre outros.
- Realização de manipulações a fim de corrigir dados armazenados no banco de dados, bem como atualizá-los.
- Capacidade de analisar os dados armazenados, com possibilidade de executar tarefas definidas pelo usuário, como por exemplo, produzir estimativas de parâmetros para uma determinada simulação a partir dos dados analisados.

A Figura 09 ilustra o relacionamento dos principais componentes presentes em um SIG, que mostra que cada sistema, dependendo de seus objetivos específicos, irá implementar estes componentes de forma diferente. Como se pode observar na interface usuário-máquina, onde o usuário está diretamente ligado ao sistema, define-se como o SIG é operado e controlado. De forma hierárquica, no nível inferior, o sistema deve possuir mecanismos de processamento de dados (entrada, análise, plotagem de mapas). Em um nível mais central, o sistema de gerenciamento de banco de dados espaciais oferece armazenamento e recuperação dos dados.

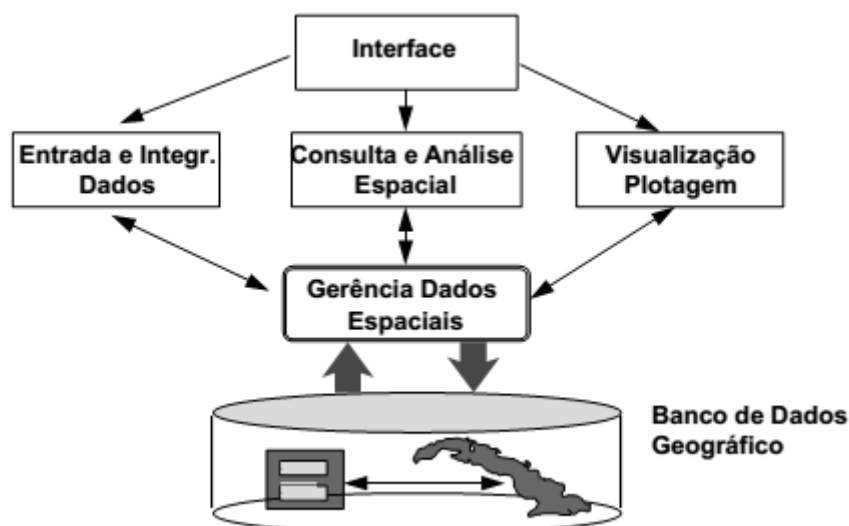


Figura 09 – Estrutura de um SIG (GeoBrasil, 2006)

Existe um grande número de aplicabilidades relacionadas ao uso do GPS integrado ao Sistema de Informações Geográficas, e uma das principais ferramentas de entrada de dados para o SIG é o GPS. Equipamentos GPS permitem coletar não só o posicionamento das regiões de interesse, como também os dados pertinentes.

2.3.1. APIs

A utilização de APIs que facilitam a utilização de mapas em sistemas web é indispensável para integração do sistema de informação geográfica com a proposta de rastreamento presente neste documento. Para tanto, foram selecionadas dentre as APIs existentes algumas que atendiam os requisitos para o problema em questão, estão elas apresentadas abaixo:

- Google Maps

Um serviço gratuito que permite incorporar o *Google Maps* a páginas da Web ou a aplicativos móveis de modo livremente. O serviço deve ser gratuito e publicamente acessível para os usuários finais (Google maps, 2013).

- Open Street Maps

É um projeto de mapeamento colaborativo para criar um mapa livre e editável do mundo. Os mapas são criados usando dados de receptores GPS portáteis, fotografias aéreas e outras fontes livres. Tanto as imagens obtidas

por processamento dos dados e os dados estão disponíveis sob uma licença *Open Database License* (KELLY, 2012).

Com o intuito de facilitar a escolha do API apropriado para o projeto, critérios de seleção foram criados. Cada critério possui uma objetiva descrição, peso e faixa de valores, os pesos e as faixas de valores dadas a cada critério têm por objetivo avaliar de forma correta os interesses reais do projeto, como por exemplo, o critério de confiabilidade que possui o maior peso é o de maior relevância já que o projeto se trata de uma forma de rastreamento e exige segurança.

A maior nota dirá qual o API mais apropriado para ser utilizado no projeto. Na Tabela 05 são descritos os critérios utilizados para selecionar a API.

Tabela 05 – Critérios para seleção da melhor API.

| Critério | Descrição | Peso | Valores Possíveis |
|---------------------------|---|-------------|--------------------------------------|
| Confiabilidade (C) | Integridade de dados, segurança e precisão das informações. | 3 | Baixa (0.0), Média (0.5), Alta (1.0) |
| Usabilidade (U) | Facilidade de uso | 2 | Baixa (0.0), Média (0.5), Alta (1.0) |
| Licença (L) | Propriedade das informações. | 1 | Baixa (0.0), Média (0.5), Alta (1.0) |
| Suporte (S) | Documentação de apoio, comunidade ativa. | 2 | Baixa (0.0), Média (0.5), Alta (1.0) |

Nas Tabelas 06,07, 08 e 09 são apresentadas as análises dos critérios da Tabela 5 para cada uma API. Na Tabela 10 são apresentados os valores obtidos de cada uma das tabelas descritas acima.

Tabela 06 – Análise de confiabilidade

| APIs | Peso | Justificativa |
|-----------------------|-------------|--|
| Google Maps | Alta | Qualquer usuário pode editar as informações dos mapas, entretanto uma equipe do Google precisa aprovar as modificações antes delas irem ao ar, isso garante uma confiabilidade maior das informações. |
| OpenStreetMaps | Média | Qualquer usuário pode editar as informações dos mapas, e essas informações são colocadas no ar instantaneamente, há uma pequena equipe que fica analisando as informações após elas estarem no ar, mas se a informação estiver errada pode levar cerca de dois dias para sair do ar. |

Tabela 07 – Análise de usabilidade

| APIs | Peso | Justificativa |
|-----------------------|-------------|---|
| Google Maps | Alta | A usabilidade do mapa da Google é muito intuitiva e não há dificuldades para encontrar o local que se procura. |
| OpenStreetMaps | Alta | A usabilidade do mapa do OSM é muito parecida à Google Maps, bastante intuitiva e sem barreiras para encontrar informações. |

Tabela 08 – Análise de licença

| APIs | Peso | Justificativa |
|----------------|--------------|--|
| Google Maps | Proprietário | Todas as informações dadas pelos usuários se tornam propriedade do Google. |
| OpenStreetMaps | Livre | As informações dadas pelos usuários são públicas. |

Tabela 09 – Análise de suporte

| APIs | Peso | Justificativa |
|----------------|-------|---|
| Google Maps | Alta | O Google Maps possui muita documentação da sua API, além de vídeos e “mostruários”. |
| OpenStreetMaps | Média | O OpenStreetMaps possui uma documentação fraca se comparada a do Google Maps, entretanto, sua comunidade é mais ativa e pronta para ajudar. |

Tabela 10 – Cálculo dos critérios da API

| APIs | Confiabilidade | Usabilidade | Licença | Suporte | Total |
|----------------|----------------|-------------|---------|---------|-------|
| Google Maps | 3 | 3 | 3 | 3 | 24 |
| OpenStreetMaps | 2 | 3 | 3 | 2 | 21 |

Levando em consideração os valores obtidos na Tabela 10, a API escolhida foi o *Google Maps*

2.3.2. Google Maps

Atualmente a API do *Google Maps* se encontra na versão três. A API Javascript do *Google Maps* permite incorporá-lo em suas páginas, como já ressaltado. A versão três, de acordo com o site do *Google Maps*, foi especialmente desenvolvida para ser mais rápida e direcionada aos dispositivos móveis.

A terceira versão oferece diversos utilitários para manipulação de mapas e para a adição de conteúdo ao mapa por meio de diversos serviços. Isso possibilita a criação de um mapa com vários recursos diferentes. Na Figura 10 pode-se observar um exemplo básico de um código gerado pelo *Google Maps*, onde os

únicos parâmetros estáticos modificados foram dentro da função “.*LatLong()*”, para obtenção das coordenadas de *Latitude* e *Longitude*, que nesse exemplo, centralizou-se na cidade do Gama-DF (Figura 11).

```
<!DOCTYPE html>
<html>
<head>
<script src="http://maps.googleapis.com/maps/api/js?
key=AIzaSyDY0kkJiTPVd2U7aTOAwhc9ySH6oHxOIYM&sensor=false">
</script>

<script>
function initialize()
{
var mapProp = {
center:new google.maps.LatLng(-16.017023,-48.064084),
zoom:13,
mapTypeId:google.maps.MapTypeId.ROADMAP
};
var map=new google.maps.Map(document.getElementById("googleMap")
,mapProp);
}

google.maps.event.addDomListener(window, 'load', initialize);
</script>
</head>

<body>
<div id="googleMap" style="width:500px;height:380px;"></div>
</body>
</html>
```

Figura 10 – Trecho do código que gera um mapa centralizado na cidade do Gama-DF

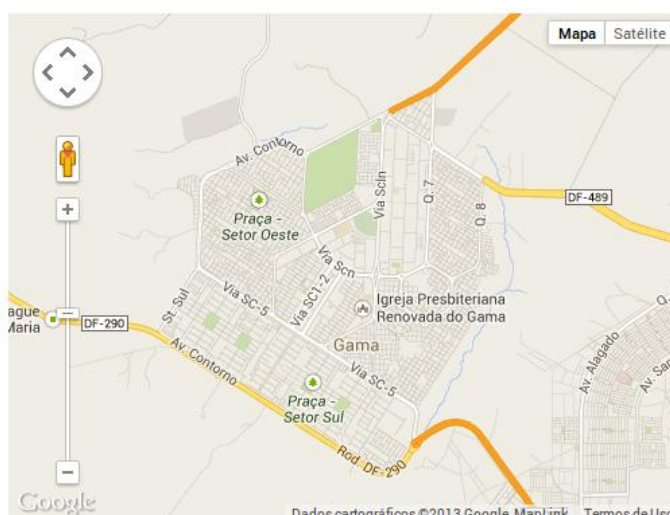


Figura 11 – Mapa exemplo gerado pelo Google Maps.

2.4. LAMP (LINUX, APACHE, MYSQL, PHP).

O LAMP é uma lista de softwares de código aberto que inclui o sistema operacional GNU/LINUX, o servidor Apache, banco de dados MySQL (*Structured Query Language*) e o PHP (*Hypertext Preprocessor*) e mostrou ser um dos que mais crescem no que se diz respeito a aplicações web, pelo fato de ser um software livre (HESHMATI, 2006). Como todos estes *softwares* são de código aberto, eles podem ser usados de forma livre. Outras vantagens em usar *softwares* livres são:

1. Menor custo – produtos de código aberto normalmente tem licenças grátis ou de custos bem menores como também um custo de operação bem reduzido, fazendo com que o produto final fique mais viável economicamente falando.
2. Confiabilidade e desempenho – grande parte dos usuários de aplicações web utiliza o LAMP através de uma variedade de plataformas antes de serem certificadas para a produção. Quando os erros são encontrados, são corrigidos rapidamente.
3. Segurança – como descrito no item anterior, uma grande quantidade de usuários testam os softwares de código aberto, fazendo com que a vulnerabilidade em questões de segurança seja logo identificada.

Existem outros softwares também como o WAMP (Windows, Apache, MySQL, PHP) e o MAMP (MacOS, Apache, MySQL, PHP). A diferença entre estes é o sistema operacional: no caso do LAMP, temos o Linux; o WAMP, o Windows; e por fim o MAMP que deriva de Mac OS. Portanto, projeto em questão foi desenvolvido com a arquitetura LAMP.

Para instalação do LAMP e criação de um servidor web, devem-se seguir as informações contidas no Anexo I. Os testes para verificação da funcionalidade dos pacotes do LAMP estão disponíveis no Anexo II.

2.5. ANDROID STUDIO

Pode-se dizer que hoje a plataforma mais popular do mundo é o *Android*. Não é de se admirar que existam várias plataformas de programação para ela, como o Eclipse, NetBeans, a IDE (*Integrated Development Environment*) e o Android Studio, do Google. O Android Studio é uma plataforma de desenvolvimento parecida com o Eclipse, porém mais simples e rápida, como afirma a própria Google.

Para instalação e configuração do *software* devem-se seguir as informações contidas no Anexo III.

Parte III
METODOLOGIA

3. METODOLOGIA

Neste capítulo, serão abordadas as métricas que foram adotadas para o acompanhamento e avaliação dos tipos de redes que podem ser usadas para o rastreamento veicular. Primeiramente, inicia-se o capítulo abordando conceitos relativos à latência, jitter, UDP (*User Datagram Protocol*) e TCP (*Transmission Control Protocol*) para avaliação das redes WLAN e WWAN em locais específicos. Por fim, o estudo de caso indica a avaliação das redes através do monitoramento de repetidos testes, com o software SIMET (Sistema de Medição de Tráfego Internet), em conjunto com cálculos estatísticos e gráficos.

3.1. MÉTRICAS

Nos últimos anos, a necessidade de estabelecer procedimentos em Gestão de Desempenho de Aplicativos (APM, na sigla em inglês) tornou-se evidente para muitas organizações. Infelizmente, muitas vezes isso só acontece depois que a aplicação já foi criada e seu desempenho foi baixo. Aplicativos pobres em desempenho podem levar a uma perda significativa de dinheiro e a um aumento de investimento em tempo e recursos para solucionar os problemas (TCPING, 2007).

O desempenho do aplicativo também é influenciado pela rede em que ele será inserido, daí a importância de se realizar testes que considerem também a performance deste fator. Para isso, podem-se mensurar alguns elementos da rede a fim de se testar o seu comportamento, como a latência, o jitter e por meio dos protocolos TCP e UDP. Essas medidas são importantes para o projeto, com o intuito de avaliar o desempenho da transferência de dados das redes e assim prever o comportamento de todo o sistema de comunicação entre cliente e servidor.

3.1.1. Latência

A latência (ou atraso da rede) pode ser definida como o tempo que um sinal, ou um pacote de dados, demora a atravessar determinada parte da rede, começando pelo dispositivo que iniciou a transmissão e finalizando no dispositivo de destino (TCPING, 2007).

A latência medida da fonte do sinal até o seu destino é chamada de “latência de ponta a ponta” ou “latência de mão única”. A “latência de ida e volta” é a soma da latência de ponta a ponta com o tempo que leva para o sinal de resposta

retornar ao destino. A latência é medida em milissegundos e é composta de três elementos principais. São eles (TCPING, 2007):

- Atraso de propagação: tempo que um sinal demora a atravessar fisicamente o caminho de rede, está relacionado com a distância geográfica entre o dispositivo transmissor e o receptor e também a velocidade do sinal propagado, que é de aproximadamente dois terços da velocidade da luz em fios ou fibras ópticas.
- Atraso de serialização: é o atraso medido a partir do transmissor para conduzir um sinal ou pacote de dados para a fila de saída. Para um dado tamanho de pacote e largura de banda disponível, o atraso de serialização é uma constante e pode ser facilmente calculado dividindo o tamanho do pacote (em bits) pela largura de banda disponível (em bits por segundo);
- Atraso de fila: é o total de atrasos causados quando um pacote é recebido, armazenado e transmitido por filas. É o maior atraso variável presente nas redes modernas, e depende do número de filas em rota e os comprimentos de fila - em uma rede muito carregada e congestionada, as filas serão longas e o atraso de fila será elevado em conformidade. Em muitos casos, este é o principal componente na latência total. Em redes não congestionadas, este atraso pode ser considerado insignificante.

3.1.2 Jitter

O jitter está relacionado à variação da latência da rede, e pode ser definido como o desvio padrão da média da latência. Na maioria dos casos, é causado por rápidas mudanças nas condições da rede e rota (TCPING, 2007).

3.1.3 TCP

O TCP é um protocolo IP com base em vários mecanismos de entrega garantida. Uma transmissão TCP inicia quando dispositivo emissor começa uma sessão com o dispositivo receptor (*syn*). Em seguida, ele espera até que o reconhecimento seja recebido (*syn/ack*), antes de a transmissão real começar. Depois que a sessão está configurada e a transmissão começa, o receptor mantém-se informando o remetente quais pacotes foram recebidos através do envio de um pacote de confirmação (*ack*). As notificações são enviadas depois que certo número de pacotes foi recebido, e se as confirmações não forem recebidas

depois de algum tempo (*time out*), os pacotes são enviados novamente (TCPING, 2007).

O protocolo TCP utiliza outro mecanismo chamado *Sliding Window*, referente ao tamanho da janela de recursos (*Window Size*). Este mecanismo dinâmico permite que as partes que estão se comunicando ajustem a capacidade atual da conexão. Isto é possível por meio do ajuste do número de pacotes que são enviados antes que um pacote de confirmação seja necessário. O tamanho da janela de dados que estão sendo enviados é gradualmente aumentado até que a taxa de envio exceda a taxa ótima para a conexão de rede. Neste ponto, alguns pacotes serão descartados pela rede (perda de pacotes), e quando o remetente percebe que alguns dados foram perdidos, ele reage cortando a taxa de envio pela metade. Este processo de redução e aumento de tamanho das janelas continua até que se atinja uma taxa de transmissão ótima (TCPING, 2007).

As características de *Window Size* e *Sliding Window* tornam o protocolo TCP extremamente sensível à latência. Quando a latência da rede aumenta, o remetente tem que manter-se inativo por mais tempo esperando que os pacotes de confirmação cheguem, em vez de enviar pacotes constantemente. Uma alta latência também faz com que o processo de *Sliding Window* trabalhe mais lentamente, uma vez que ele também depende da taxa em que são recebidos pacotes de confirmação. Isto resulta numa relação inversamente proporcional entre a latência da rede e a taxa de transferência TCP. Com o aumento da latência da rede, o rendimento da transferência TCP diminui (TCPING, 2007).

Como explicado anteriormente, com o aumento na latência, o remetente fica inativo por longo tempo, à espera que o receptor reconheça os dados antes da transmissão poder concluir-se totalmente. Durante esse tempo, o receptor também espera e cria buffers de pacotes até que possa montar mensagens inteiras. Quando várias sessões devem ser manuseadas simultaneamente (por exemplo, quando um servidor é acessado por vários usuários ao mesmo tempo), é requerido um constante aumento no uso de buffer, o que pode, eventualmente, causar degradação de desempenho global em relação ao emissor (TCPING, 2007).

3.1.4 UDP

O UDP é um protocolo IP que oferece uma maneira direta de enviar e receber sinais ou pacotes. Ao contrário do TCP, não se tem um controle de fluxo ou sistema de garantia de entrega. Ele não fornece o sequenciamento de pacotes e não pode dividir e remontar mensagens. Um dispositivo, ao enviar pacotes UDP, assume que o sinal enviado chegou ao destino, sem qualquer alerta de confirmação de envio (TCPING, 2007).

O protocolo UDP é muito eficiente no que se refere à economia de recursos de processamento, pois geralmente realiza a troca de pequenos pacotes de dados e pouca remontagem de mensagens. Este protocolo também é tipicamente usado para aplicações de *streaming* de mídia (vídeo e voz sobre IP – VoIP, em inglês), onde um pacote perdido ocasionalmente tem pouca relevância em relação a perda de dados de pacotes TCP (TCPING, 2007).

Ao contrário do TCP, a latência e taxas de transferência são completamente independentes no tráfego UDP. Ou seja, o valor da latência não influencia no rendimento da transferência de dados no protocolo. Se o valor da latência permanecer constante (jitter nulo), a única consequência que a latência pode ocasionar é um aumento da demora de todo o fluxo (*lag*) (TCPING, 2007).

Com alta latência, mesmo que esteja em um padrão fixo, haverá um efeito significativo sobre um aplicativo utilizado em tempo real. É evidente que seria muito difícil realizar uma ligação utilizando o VoIP através de um *link* com alta latência, pois o mecanismo de envio e recebimento será relacionado com o tempo de espera de acordo com o valor da latência.

Porém, o UDP é sensível ao valor do jitter. O jitter pode trazer efeitos graves às aplicações UDP, especialmente voz e vídeo sobre IP. Jitters de aproximadamente 50 milissegundos ou um pouco menores podem ocasionar em aumento da latência e perda de pacotes. Este elemento faz com que os pacotes cheguem ao seu destino com um tempo diferente do que eles se originaram, e às vezes até mesmo em uma ordem diferente. Assim, palavras podem soar de forma intermitente e distorcida, a um grau que a mensagem pode tornar-se completamente incompreensível.

3.2. ESTUDO DE CASO

Com o intuito de desenvolver uma aplicação para rastreamento de forma efetiva, foram realizados testes com as redes WLAN e WWAN para saber a influência da velocidade de transmissão de dados de cada cenário. Com isso, é necessário encontrar os valores para avaliar o melhor cenário e tornar o projeto o mais viável possível.

Para tanto, se pode observar na Figura 12 o cenário do estudo de caso, em que são avaliadas as redes 3G de um celular e WLAN de um computador pelo software SIMET (Sistema de Medição de Tráfego Internet).

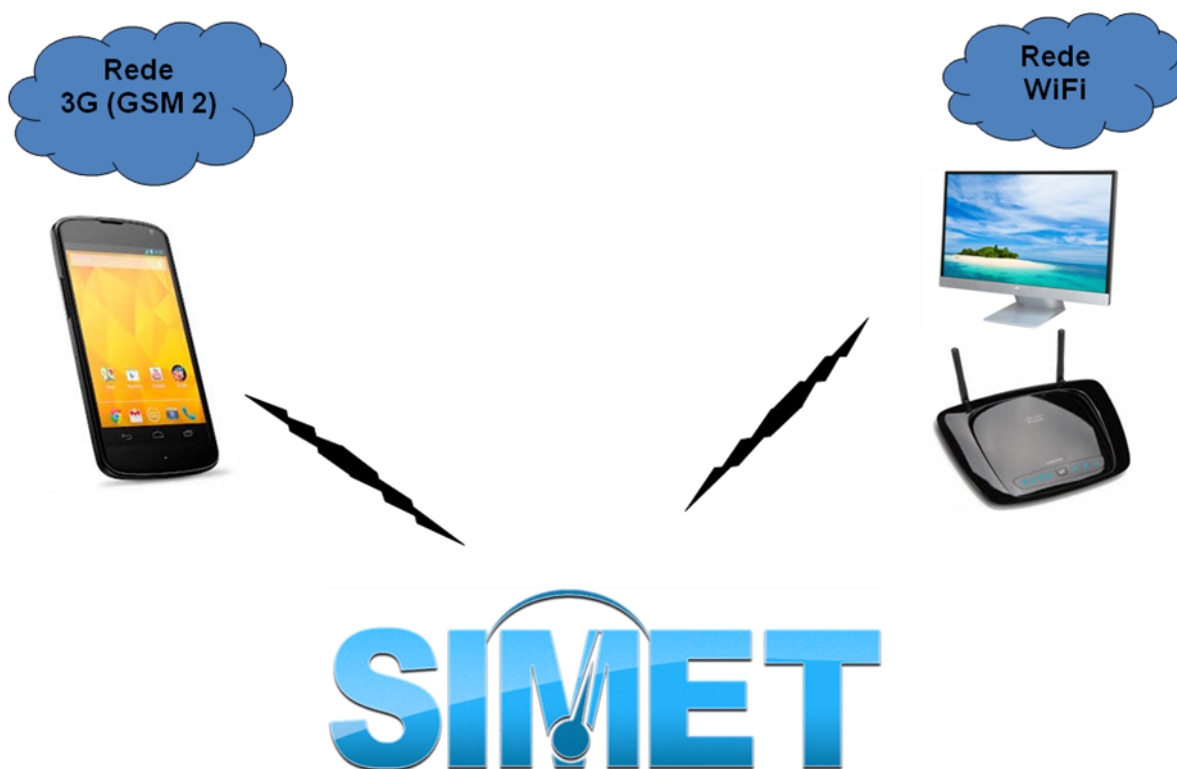


Figura 12 – Cenário para o Estudo de Caso.

O primeiro ambiente monitorado foi a rede de dados 3G, da operadora rede GSM 2, no estacionamento da FGA-UnB.

O segundo ambiente monitorado deveria ser a rede WLAN da FGA, mas não foi possível realizar testes na rede por conta de bloqueios feitos pelo CPD (Centro de Processamento de Dados) local. Tentou-se estabelecer um diálogo com o

responsável do local para que se pudesse obter estes dados, mas sem sucesso. Portanto, para efeitos de comparação, o monitoramento foi realizado em uma rede WLAN de uma residência comum, em que a operadora de serviços é a GVT com plano de 15MB contratado. O computador foi fixado em local em que o sinal estava na margem de 40 a 60% para simular uma localização em que se encontrasse um veículo no estacionamento da FGA e os pontos de acesso Wifi no interior do prédio.

Utilizou-se o software de monitoramento online SIMET. Este é um sistema que realiza testes de desempenho em redes com acesso a Internet, através de servidores espalhados dentro dos Pontos de Troca de Tráfego Internet. Para isso, mediram-se os valores da latência, jitter, tráfego TCP e vazão UDP em um intervalo de aproximadamente dois minutos, coletando valores de cada variável. Com isso, calcula-se a média e a mediana relativa a todos os valores medidos no intervalo de tempo. (SIMET, 2013)

Com o objetivo de obter dados mais precisos, o teste foi feito com bases estatísticas do desvio padrão e média dos valores medidos a cada 30 minutos. O processo para medir os valores relativos à rede era de aproximadamente 2 minutos e por isso foi definido, que a cada 30 minutos, deveriam ser medidas três amostras de cada variável para os cálculos estatísticos, o que resultaria em um monitoramento contínuo da rede por aproximadamente 6 minutos. O número de amostras igual a três é explicado pelo tempo extenso de se obter cada dado.

Em resumo, foram monitoradas as redes WWAN e WLAN, a cada 30 minutos, medindo-se os parâmetros por três vezes seguidas e, a partir de cada valor, foram calculados a média e o desvio padrão das amostras. Com os dados estatísticos calculados, plotou-se gráficos para melhor avaliação e análise.

Foram feitas cinco medidas intercaladas a cada trinta minutos, totalizando 15 amostras. Não foi possível obter mais medidas, pois o plano de dados contratado da operadora GSM 2 atingiu o limite e a velocidade de transmissão de dados foi bastante reduzida. Para o teste da rede WLAN, foi seguido o mesmo padrão para efeitos comparativos. No Anexo VII se encontra uma das medidas realizadas com o software SIMET para o ambiente 3G e no Anexo VIII se encontra uma das medidas realizadas para o ambiente WLAN.

Nas figuras 13 e 14, podem-se observar os valores obtidos através do cálculo da média das amostras medidas, em que medida é definida pelo

monitoramento da rede a cada 30 minutos. Observam-se valores altos de latência para rede 3G com significativos valores de desvio padrão da latência em comparação com os dados obtidos para a rede WLAN. Com a análise dos gráficos abaixo é possível concluir que a rede 3G é mais instável e lenta, uma vez que a latência é o tempo que pacotes de dados vão do emissor para o receptor (latência de ponta a ponta).

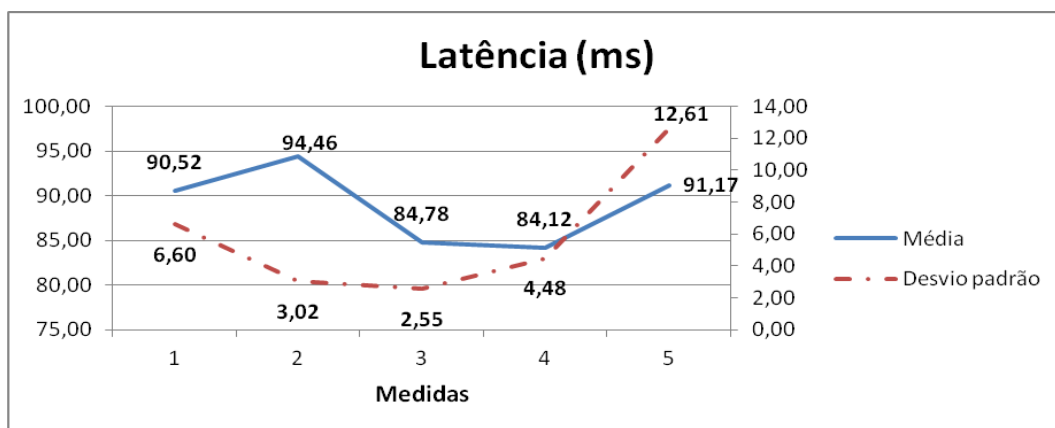


Figura 13 – Valores de latência (rede 3G)

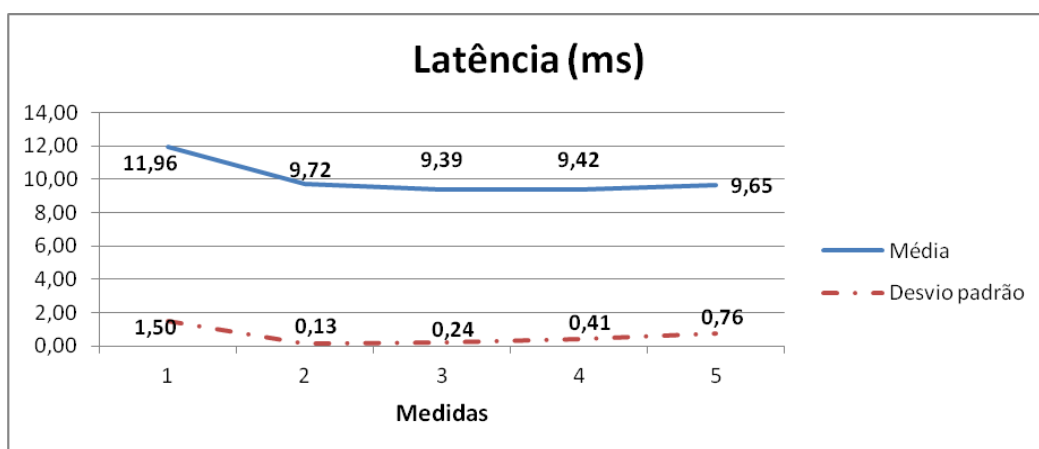


Figura 14 – Valores de latência (rede WLAN).

Nas figuras 15, 16, 17 e 18 podem-se observar os valores obtidos através do cálculo da média das amostras medidas, em que cada medida é definida pelo monitoramento da rede a cada 30 minutos. Observam-se valores significativamente mais altos de Jitter na rede 3G que na rede WLAN, tanto para *download* como para *upload*. Isto pode ser explicado analisando os gráficos anteriores (figuras 13 e 14) em que os valores de latência para rede 3G variam de forma mais acentuada, o que significa maiores valores de desvio padrão. Sendo o Jitter, um parâmetro

definido como o desvio padrão médio da latência, conclui-se que a rede 3G é mais instável que a rede WLAN no intervalo de tempo monitorado.

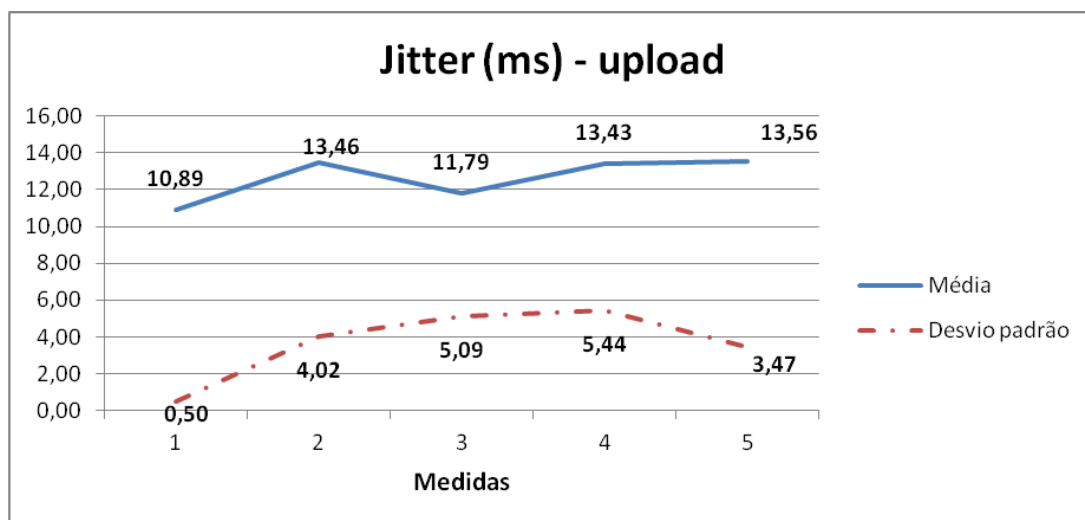


Figura 15 – Valores de Jitter para *upload*(rede 3G)

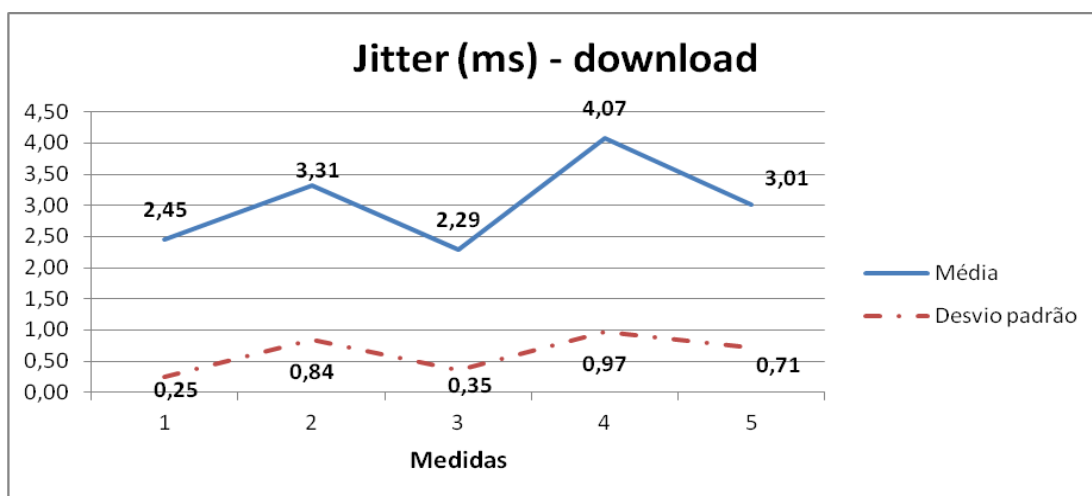


Figura 16 – Valores de Jitter para *download* (rede 3G)

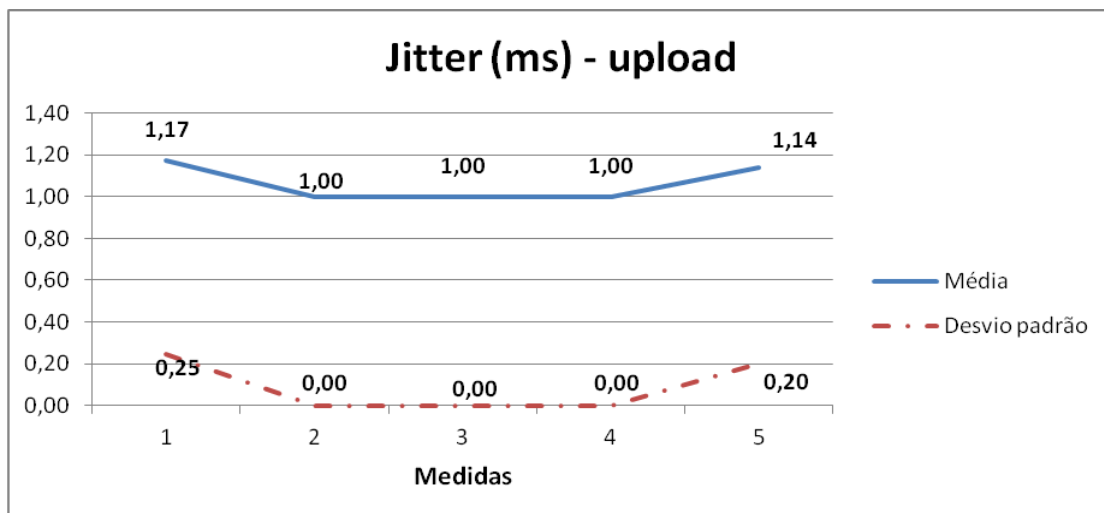


Figura 17 – Valores de Jitter para *upload* (rede WLAN)

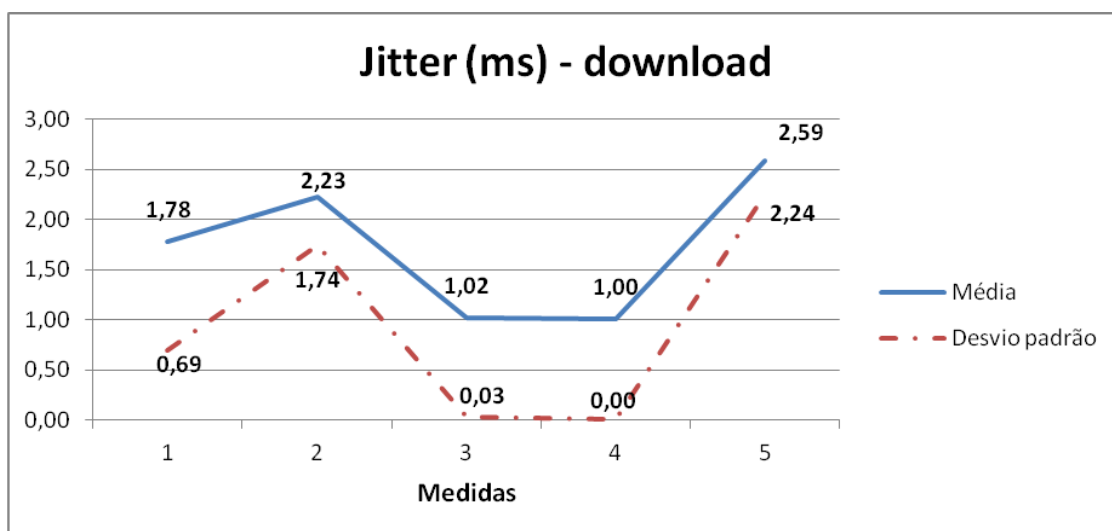


Figura 18 – Valores de Jitter para *download* (rede WLAN)

Nas figuras 19, 20, 21 e 22 observam-se os valores de tráfego TCP menores para a rede 3G em relação à rede WLAN em ambos os casos (*download e upload*) assim como os valores de desvio padrão que é explicado pela maior linearidade do gráfico relacionado à rede WLAN. Isso é explicado devido ao comportamento inversamente proporcional da latência e a taxa de transferência. As redes 3G possuem maior latência e, conseqüentemente, menor velocidade de tráfego TCP. Com isso, se pode concluir que as redes WLAN são mais estáveis. É possível notar uma queda significativa de velocidade na rede 3G nas duas últimas medidas que podem ser explicadas devido ao horário em que foram obtidas, após as 18 horas,

onde há um maior número de dispositivos utilizando a antena local e assim sobrecarregando a rede.

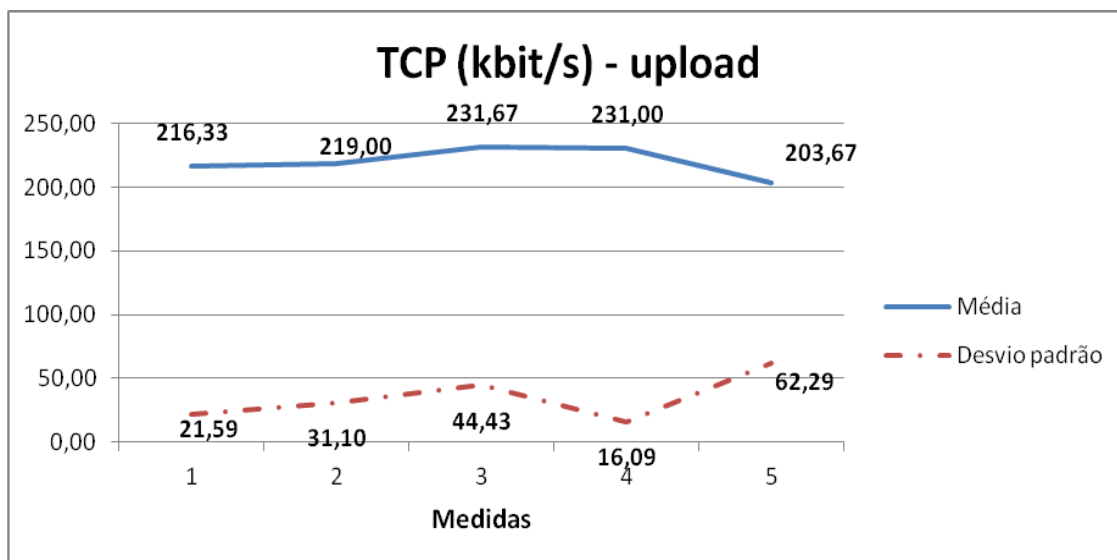


Figura 19 – Valores de tráfego TCP para *upload* (rede 3G)

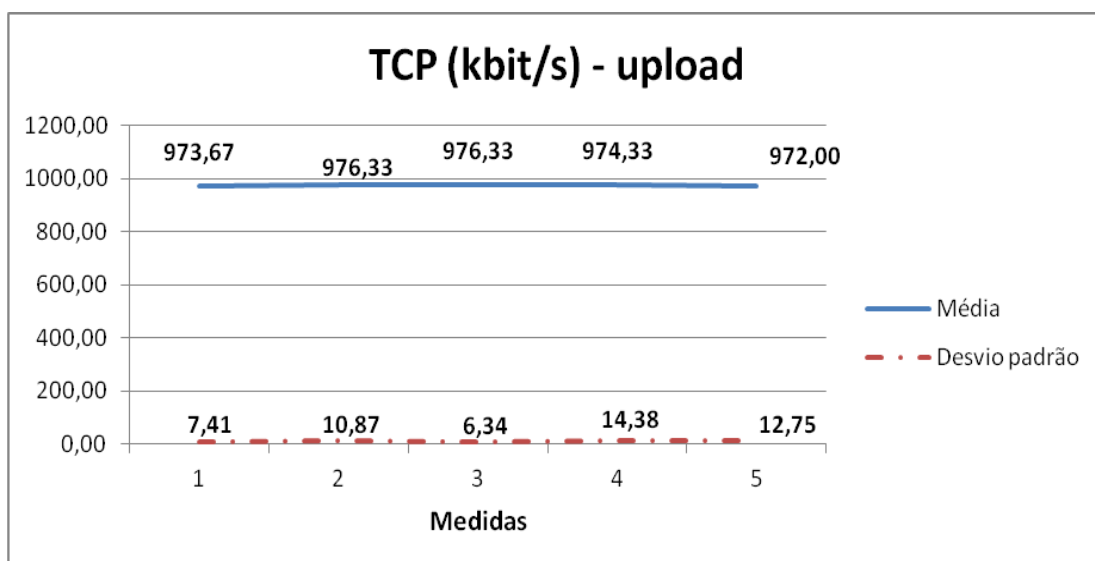


Figura 20 – Valores de tráfego TCP para *upload* (rede WLAN).

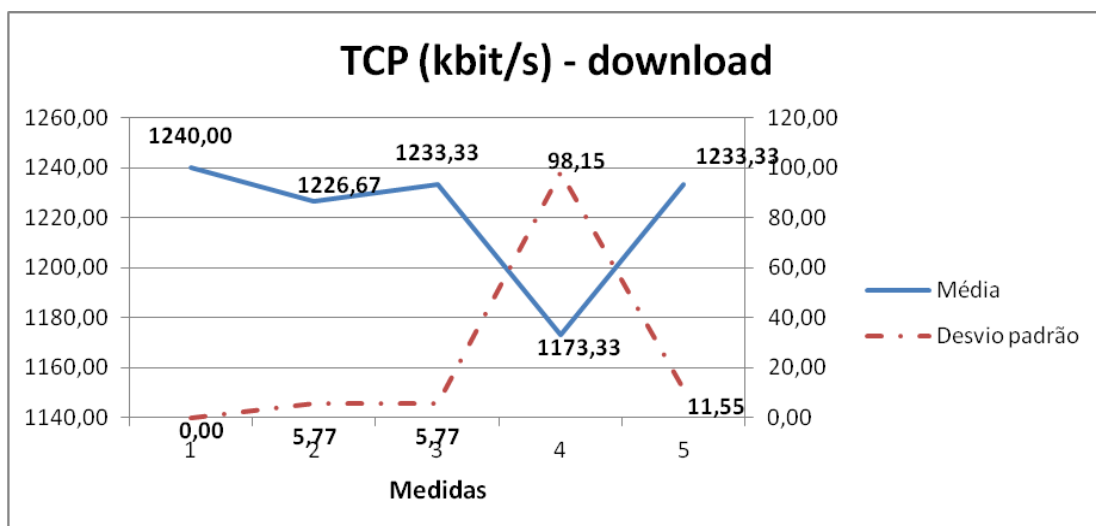


Figura 21 – Valores de tráfego TCP para *download* (rede 3G)

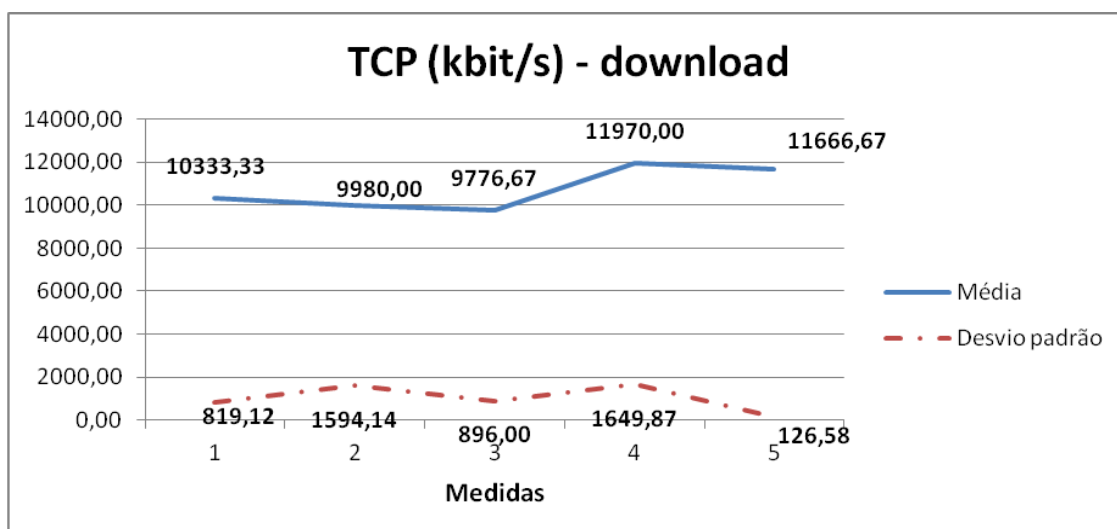


Figura 22 – Valores de tráfego TCP para *download* (rede WLAN)

Nas figuras 23, 24, 25 e 26 observam-se os valores de vazão UDP menores para a rede 3G em relação à rede WLAN em ambos os casos (*download* e *upload*), assim como menores valores de desvio padrão, explicados pela maior linearidade do gráfico relacionado à rede WLAN. Como este protocolo é independente de qualquer outro parâmetro, com exceção da latência que define o tempo em que o pacote de dados chega ao seu destino, a vazão UDP é relativamente maior que a velocidade de tráfego TCP. Esta velocidade significativamente maior pode ser explicada pelo fato do protocolo UDP não necessitar de alertas de envio completo do sinal, o que melhora o processamento e o torna mais veloz. Este comportamento pode ser observado nos gráficos das figuras abaixo onde também

é possível notar um maior desvio padrão nas últimas medidas da rede 3G ocasionadas devido à instabilidade da rede proveniente do maior número de usuários na rede local.

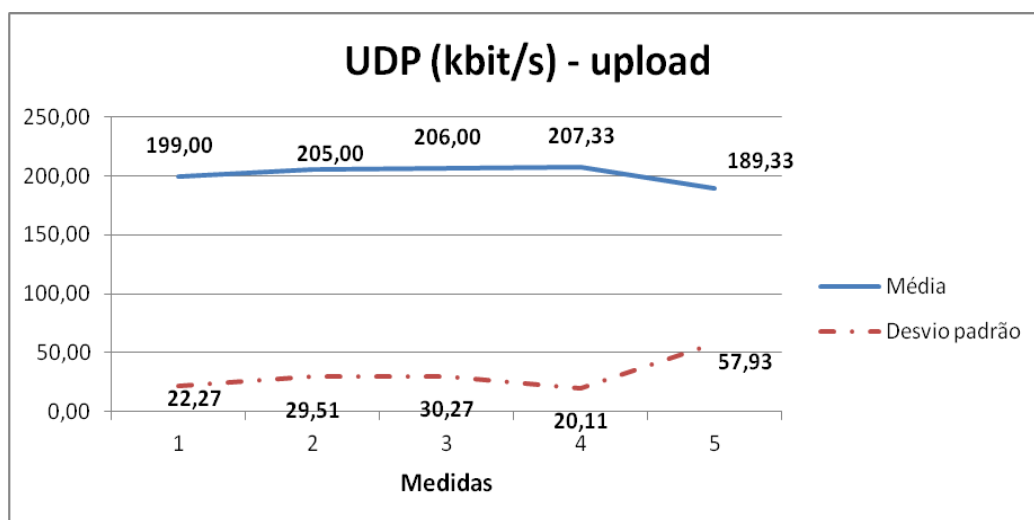


Figura 23 – Valores de vazão UDP para *upload* (rede 3G)

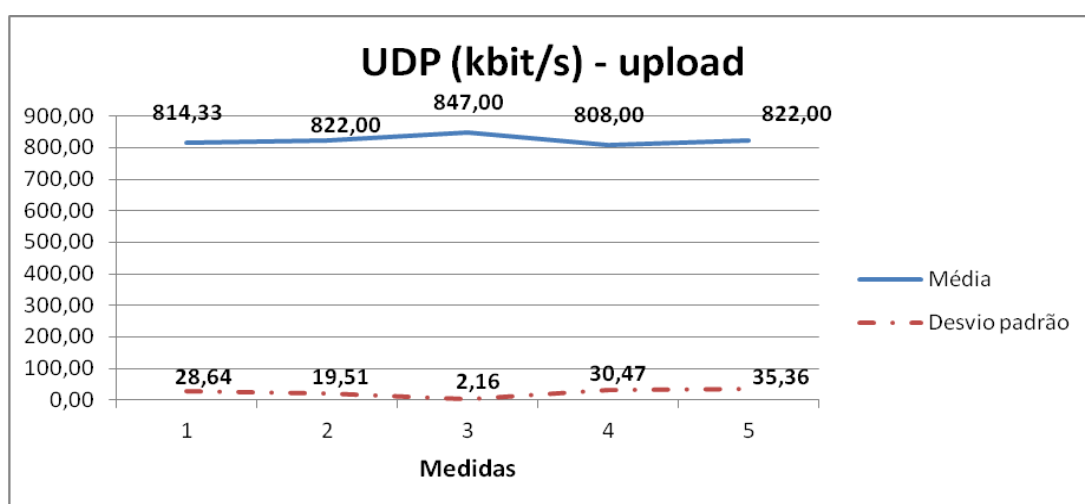


Figura 24 – Valores de vazão UDP para *upload* (rede WLAN)

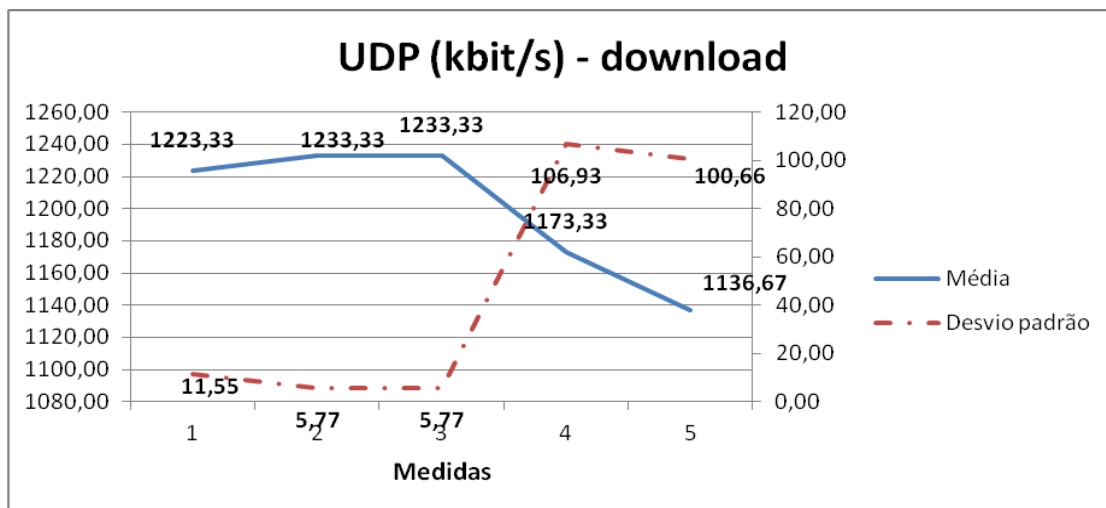


Figura 25 – Valores de vazão UDP para *download* (rede 3G)

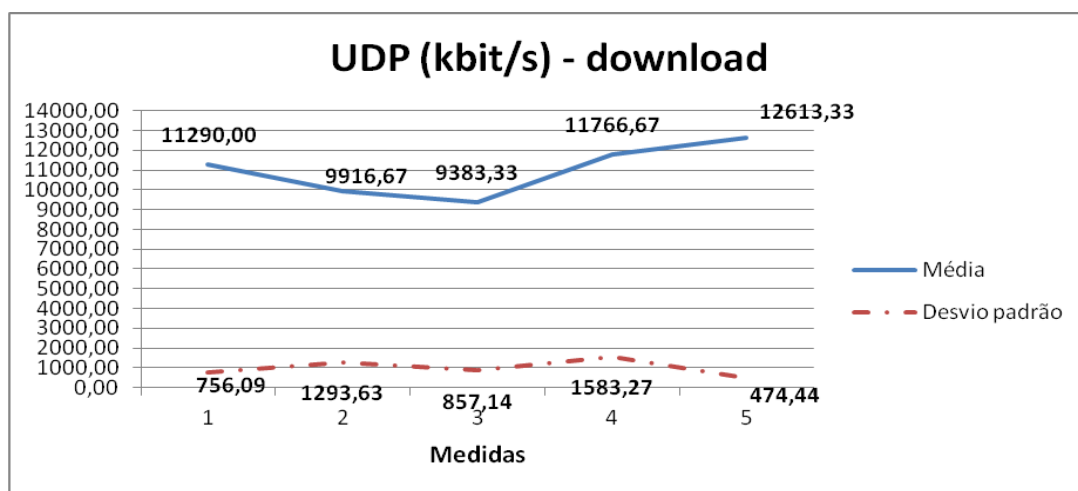


Figura 26 – Valores de vazão UDP para *download* (rede WLAN)

Após todas as avaliações das redes que podem ser utilizadas para o rastreamento do veículo, concluiu-se que há uma maior confiabilidade da rede WLAN por ser mais estável e possuir maior taxa de transferência de dados. Porém, a desvantagem da utilização de redes WLAN ocorre em relação à limitação de sua área de cobertura e da dependência contínua de energia, que pode ser minimizada com o uso de *no-breaks* na rede de abastecimento de energia dos *access point*. Ainda é possível concluir que o protocolo TCP é o mais vantajoso por garantir que o pacote de dados chegará ao destino, por meio dos mecanismos de entrega que este protocolo possui. Os dados referentes aos valores medidos se encontram nas tabelas do Anexo IX e Anexo X.

Parte IV
RESULTADOS

4.1. IMPLEMENTAÇÃO DA PROPOSTA

Neste capítulo, é abordado o detalhamento do funcionamento de todo o sistema que compõe o servidor e o cliente, já implementados. Para que o celular esteja sempre em comunicação com o servidor e não afete o funcionamento do projeto, foi desenvolvido um carregador ligado diretamente à bateria do carro, que alimenta constantemente o dispositivo móvel. Por fim, apresentam-se os resultados do projeto onde se observa o funcionamento de todo o sistema de rastreo veicular.

Na Figura 27, ilustra-se o cenário de funcionamento do projeto em que o celular, localizado no veículo e conectado na mesma rede WLAN do *raspberry PI*, se comunicam para obter a localização mais precisa possível. A identificação de roubo ou furto do veiculo é interpretada através da comunicação entre o dispositivo e o servidor quando for interrompida.

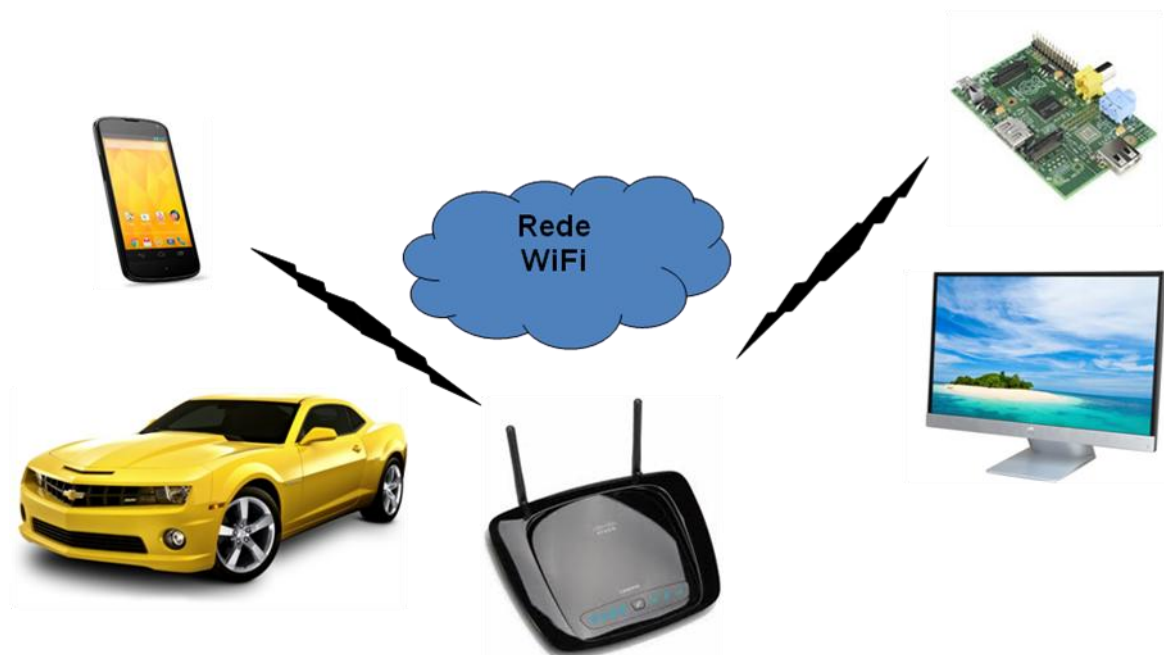


Figura 27 – Cenário do projeto

4.1.1. Servidor de Rastreamento no Raspberry PI

Um servidor web é um programa de computador responsável por aceitar pedidos HTTP (*Hipertext Transfer Protocol*) de clientes e servi-los com respostas HTTP, incluindo dados como documentos HTML com objetos embutidos (páginas web). As páginas de internet podem ter várias extensões em seus arquivos e cada

servidor interpreta um grupo de extensões de arquivos. A extensão HTML é comum a todas.

O projeto em questão utiliza as extensões *.js* e *.php*. Arquivos *.js* contém programação em *JavaScript* e são utilizados em geral pelos navegadores e editáveis com qualquer editor de texto. Arquivos *.php* contém uma linguagem de programação que permite transladar para a web o conteúdo armazenado em bases de dados.

Este servidor possui como origem do conteúdo enviado uma resposta dinâmica a um pedido HTTP, pois o processamento e troca de dados ocorre eventualmente acessando a base de dados na qual o resultado obtido é enviado para o servidor pelo cliente. A vantagem deste tipo de resposta é na possibilidade de ela poder ser programada por alguns tipos de linguagem de programação. A Figura 28 ilustra o funcionamento de um modelo de entrega dinâmica entre o servidor e o banco de dados.

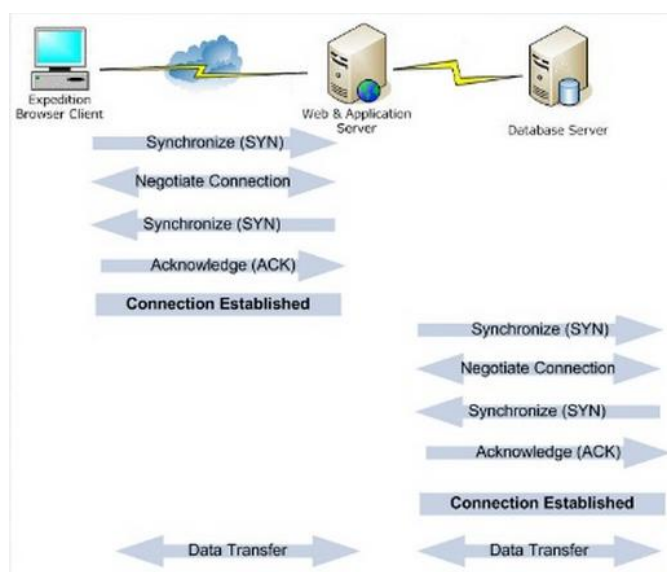


Figura 28 – Modelo de entrega dinâmica (SERVIDOR WEB).

O servidor em questão é o Apache que, após sua instalação via terminal no *Raspberry Pi*, foi configurado para atender os requisitos necessários para o funcionamento do projeto, conforme mostrado no Anexo I.

Com o site do servidor funcionando, deve-se prosseguir com a configuração relacionada ao rastreamento de um dispositivo móvel, incluindo os arquivos que farão a conexão com o cliente e o banco de dados para então obter a localização

com melhor precisão do cliente. Estes arquivos devem estar no diretório /www/var/rastreamento e são:

- `displaymap.php` : este é o arquivo principal que conterà toda a informação da pagina acessada para se ver a localização do cliente. Será exibida uma pagina em que se pode selecionar a rota de acordo com o usuário cadastrado no banco de dados, o tempo de atualização, o *zoom* relativo ao mapa e a opção de excluir uma rota. É possível ainda, realizar o cadastro de um usuário com informações do mesmo, citando o veículo e sua placa. Para que o funcionamento ocorra conforme descrito anteriormente, no código referente à página principal, são inclusos os *links* para chamar outros arquivos que estão relacionados ao *layout* da pagina, do mapa e em relação à estética de cada elemento. São incluídos *scripts* da api do Google *maps*, do arquivo *Javascript* do mapa e o *link* que inclui a configuração de posicionamento de cada elemento da pagina. O arquivo para carregar as rotas armazenadas no banco de dados é chamado para que se mostre a localização no mapa. Na figura 29, observa-se a chamada dos arquivos que definem o layout do mapa principal (*maps.js*) e do tipo de mapa definido pelo usuário. Neste caso, o tipo usado é o *Google Maps (google.js)*. O arquivo *styles.css* foi totalmente modificado para atender o layout final da página do servidor com suas funcionalidades.

```

1 <!DOCTYPE htmlPUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
2 "http://www.w3.org/TR/html4/loose.dtd">
3 <html>
4 <head>
5   <title>Rastreamento veicular com auxilio de dispositivos móveis</title>
6   <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
7
8   <script src="javascript/custom-google-loader-1.0.0.min.js"></script>
9   <script src="javascript/maps.js"></script>
10  <script src="javascript/leaflet-0.7.2/leaflet.js"></script>
11  <script src="javascript/leaflet-plugins/google.js"></script>
12  <script src="javascript/leaflet-plugins/bing.js"></script>
13  <link href="javascript/leaflet-0.7.2/leaflet.css" rel="stylesheet" type="text/css" />
14  <link href="styles/styles.css" rel="stylesheet" type="text/css" />
15  <script type="text/javascript">
16
17
18      google.load('jquery', '1.11.0');
19
20      var time = 3000;
21      var numofitens = 7;
22      var routeSelect;
23      var refreshSelect;
24
25      var map;
26      var intervalID;
27      var newInterval;
28      var currentInterval;
29      var zoomLevelSelect;
30      var zoomLevel;
--

```

Figura 29 – Links dos scripts e variáveis do código (*displaymap.php*).

Na Figura 30, se pode observar o código responsável pela modificação do *layout* do servidor em relação ao original. Primeiramente, se chama a função *load()* que carrega as informações do banco de dados. A variável *globalNav* contém a barra de navegação de abas á serem selecionadas. Na variável *pageNav* tem-se o código que representa a janela lateral para cadastro de usuário para veículo e placa. A variável *selectionLinks* define os links para informações diversas do rastreamento. Todas as variáveis estão indicadas no arquivo *styles.css* para indicar a posição referencial ou absoluta na pagina principal do servidor.

```

69 <body onload="load()">
70 <div id="masthead">
71 <h1 id="siteName">Rastreamento Veicular </h1>
72 <div id="globalNav">
73  
74 <div id="globalLink">
75 <a href="displaymap.php" id="gl1" class="glink">Principal</a>
76 <a href="autores.php" id="gl2" class="glink">Quem somos</a>
77 <a href="projeto.php" id="gl3" class="glink">0 projeto</a>
78 <a href="contato.php" id="gl7" class="glink">Contato</a>
79 </div>
80 <!--end globalLinks-->
81 </div>
82 <!-- end globalNav -->
83 </div>
84 <!-- end masthead -->
85 <div id="pagecell1">
86 <!--pagecell1-->
87 <div id="pageName">
88 <h2>Cadastro</h2>
89 
90 </div>
91 <div id="pageNav">
92 <div class="relatedLinks">
93 <h3>Usuário</h3>
94 <hr />
95 <form id="form" method="get" action="">
96 <label for="host">Veiculo :</label><br />
97 <input type="text" name="host" id="host" size="15" maxlength="15" />
98 <label for="community">Placa :</label><br />
99 <input type="text" name="community" id="community" size="15" maxlength="15" /><br />
100 <input name="botao" type="submit" value="Atualiza">
101 </form>
102 </div>
103 <div id="sectionLinks">
104 <a href="velocidade.php?host=<?php echo $host?&localhost=<?php echo $community?>" target="conteudo">Velocidade</a>
105 <a href="distancia.php?host=<?php echo $host?&localhost=<?php echo $community?>" target="conteudo">Distância</a>
106 <a href="data.php?host=<?php echo $host?&localhost=<?php echo $community?>" target="conteudo">Data</a>
107 <a href="precisao.php?host=<?php echo $host?&localhost=<?php echo $community?>" target="conteudo">Precisão</a>
108 <a href="extras.php?host=<?php echo $host?&localhost=<?php echo $community?>" target="conteudo">Informações Extras</a>
109 </div>
110 </div>
111 </div>
112 <!--end pagecell1-->

```

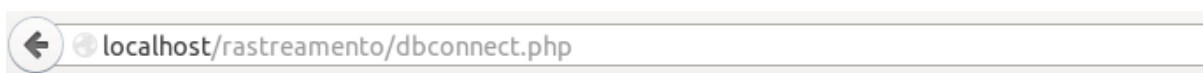
Figura 30 – código do servidor (*displaymap.php*)

- *dbconnect.php*: este arquivo é responsável pela conexão do banco de dados com o servidor em que, caso algum parâmetro não esteja certo, o compilador retorna um erro quando há o acesso do arquivo pelo navegador web, como se pode observar na Figura 32. Se estiver tudo certo, o retorno é vazio. Na Figura 31, se observa os parâmetros que foram alterados no código para funcionar de acordo com o projeto proposto e são:

- \$dbhost: deve-se atribuir este parâmetro como 'localhost' que é o endereço do servidor do banco de dados.
- \$dbuser: deve-se atribuir este parâmetro como 'root' que é o usuário de acesso do login do banco de dados.
- \$dbpass: deve-se atribuir este parâmetro com a senha para acesso do banco *mysql* configurado em sua instalação.
- \$dbname: deve-se atribuir este parâmetro como o nome do banco de dados criado.

```
dbconnect.php x
1 <?php
2
3 $dbhost = 'localhost';
4 $dbuser = 'root';
5 $dbpass = '280590';
6 $dbname = 'gpstracker';
7
8 $mysqli = new mysqli($dbhost, $dbuser, $dbpass, $dbname);
9
10 if (mysqli_connect_errno()) {
11     echo "Connection failed: " . mysqli_connect_error();
12     exit();
13 }
14
15 ?>
```

Figura 31 - Código do arquivo (*dbconnect.php*).



Connection failed: Access denied for user 'root'@'localhost' (using password: YES)

Figura 32 – Erro referente a não conexão com o banco de dados.

- *updatelocation.php*: este arquivo é responsável pela conexão com o cliente na rede local, no caso o celular, para que se possa chamar a função que extrai as informações de localização do dispositivo contidas na aplicação do celular. Neste arquivo, é chamado o arquivo *dbconnect.php* e em caso de não conexão com o banco de dados, é retornada a variável de erro e ocorre o travamento de todo o processo. Na figura 33, observa-se parte do código do *updatelocation.php* que indica as variáveis a serem extraídas da aplicação.

```

19     $query = 'CALL prcSaveGPSLocation(\'
20         . $latitude . '\',\'
21         . $longitude . '\',\'
22         . $speed . '\',\'
23         . $direction . '\',\'
24         . $distance . '\',\'
25         . $date . '\',\'
26         . $locationMethod . '\',\'
27         . $phoneNumber . '\',\'
28         . $sessionID . '\',\'
29         . $accuracy . '\',\'
30         . $extraInfo . '\',\'
31         . $eventType . '\')';
32
33     if (!$mysqli->multi_query($query)) {
34         die('$mysqli->multi_query: ' . $mysqli->error);
35     }
36
37     $mysqli->close();
38
39     echo '0';

```

Figura 33 – Parte do código do *updatelocation.php*.

- *getroutes.php*: este arquivo é responsável por chamar a função do cliente que pegue todos os pontos de localização do cliente no tempo em que o mesmo foi monitorado. Neste arquivo, é chamado o arquivo *dbconnect.php* e em caso de não conexão com o banco de dados é retornada a variável de erro e ocorre o travamento de todo o processo.
- *getrouteformap*: este arquivo é responsável por chamar a função do cliente que pegue a rota estabelecida pela junção de todos os pontos de localização estabelecidos através do tempo em que o mesmo foi monitorado. Neste arquivo, é chamado o arquivo *dbconnect.php* e em caso de não conexão com o banco de dados é retornada a variável de erro e ocorre o travamento de todo o processo.
- *deleteroute.php*: este arquivo é responsável por deletar rotas armazenadas no banco de dados. Este procedimento é feito através do botão “delete” presente na pagina principal do servidor. Neste arquivo, é chamado o arquivo *dbconnect.php* e em caso de não conexão com o banco de dados, é retornada a variável de erro e ocorre o travamento de todo o processo.

Em resumo, o funcionamento do servidor pode ser melhor entendido a partir do fluxograma que se encontra na Figura 34.

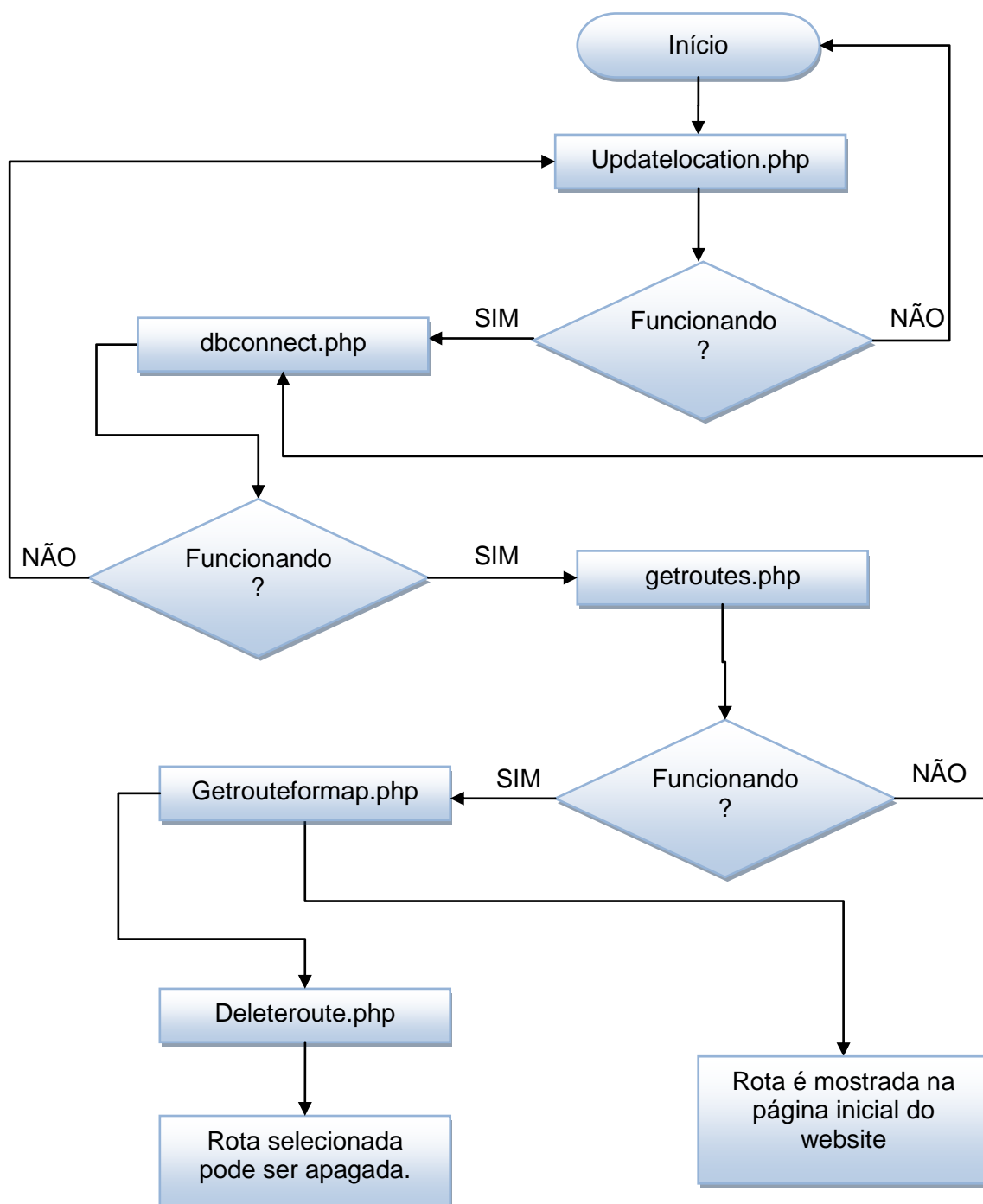


Figura 34 – Fluxograma do funcionamento do servidor web.

Diretório *mysql*: este diretório conterà o arquivo com extensão *.sql* que será instalado no bando de dados do servidor.

- Diretório *javascript*: este diretório conterà todos os arquivos *.js* que serão incluídos no arquivo *displaymap.php*
- Diretório *styles*: este diretório conterà o arquivo *.css* que será responsável pela edição dos elementos da pagina principal relacionados a posição e tamanho.
- Diretório *images*: este diretório conterà os arquivos de imagem da pagina principal.

Com o servidor instalado, ativo e funcionando, deve-se instalar o banco de dados no servidor que se encontra no Anexo V.

Após a instalação do banco de dados, se houver êxito, deve-se ter como resultado o ilustrado na Figura 35.

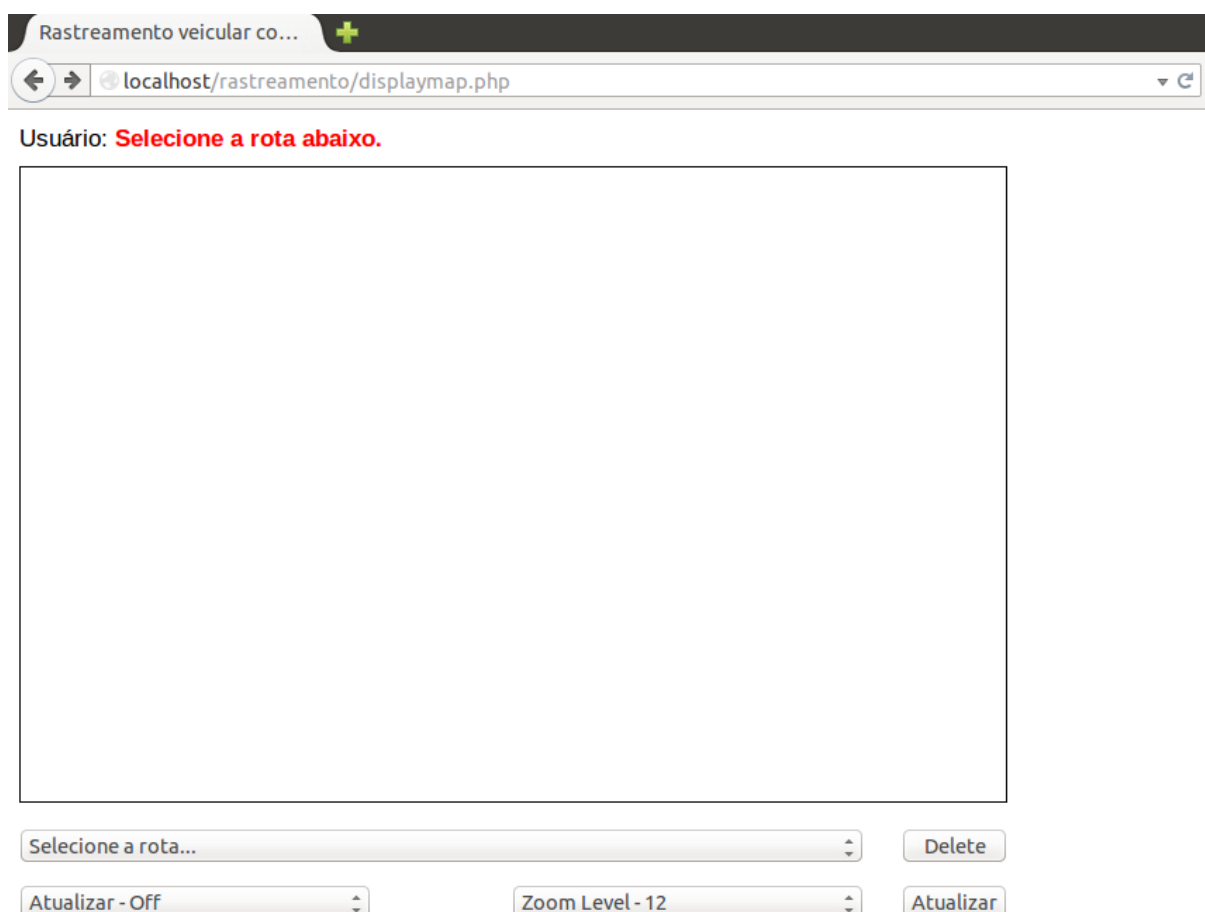


Figura 35 - Servidor conectado com o banco de dados.

4.1.2. Agente De Rastreamento Via Dispositivo Móvel

Em um sistema cliente-servidor, o servidor, como já mencionado, é um dispositivo que prover serviços à rede, enquanto o cliente é o agente que acessa esses serviços. No presente contexto, o dispositivo móvel (celular) é o agente de rastreamento pelo fato de se localizar no veículo em um lugar discreto e de difícil acesso.

O celular deverá ser equipado com o sistema operacional *Android* baseado na plataforma Linux, projetado pela *Google* para o gerenciamento de todos os processos dos aplicativos e do *hardware* do dispositivo. O dispositivo deverá ter um *hardware* compatível com acesso à rede 3G e, para melhor precisão de localização, antena GPS. A alimentação do celular deverá ser interligada com a bateria do veículo para que o mesmo nunca desligue e possa ser continuamente monitorado.

Ao desenvolver uma aplicação deve-se, primeiramente, configurar o dispositivo para o desenvolvimento do aplicativo. Para isso, devem-se seguir alguns passos, que se encontram no Anexo VI.

Com a aplicação rodando no celular, é possível ver o status do aplicativo rodando em fases no celular, ou seja, é possível ver a compilação do programa através do *Android Studio*. O primeiro teste foi ativado apenas a configuração do celular que utiliza redes WLAN locais para localizar a posição do dispositivo. Quando se ativa a localização através do botão "*Tracking is on*" no celular, observa-se o status "*startTracking*" e "*onConnected*" e logo após a posição indicando as coordenadas de latitude, longitude e precisão, respectivamente, através da *API* do *Google Maps*. Logo após, observa-se o status de conexão com o servidor retornando a variável 200 como sendo afirmativa. Como se pode observar na Figura 36, as coordenadas foram:

- latitude: -15.8348533
- longitude: -47.9755165
- precisão: 76.5 m

```
2831-2831/com.websmithing.gpstracker D/LocationService: startTracking
2831-2831/com.websmithing.gpstracker D/LocationService: onConnected
2831-2831/com.websmithing.gpstracker E/LocationService: position: -15.8340533, -47.9755165 accuracy: 76.5
2831-2831/com.websmithing.gpstracker D/dalvikvm: GC_FOR_ALLOC freed 446K, 6% free 9613K/10124K, paused 16ms, to
2831-2831/com.websmithing.gpstracker E/GpsTrackerActivity: checkIfWebsiteIsReachable onSuccess statusCode: 200
2831-2831/com.websmithing.gpstracker E/LocationService: sendLocationDataToWebsite onSuccess statusCode: 200
```

Figura 36 - Status de localização com o celular conectado em redes WLAN.

No segundo teste foram ativadas as configurações de GPS e redes WLAN para localização. Observa-se como resultado, na Figura 37, as coordenadas:

- latitude: -15.8339983
- longitude: -47.9755629
- precisão: 58.5 m

```
2831-2831/com.websmithing.gpstracker D/LocationService: startTracking
2831-2831/com.websmithing.gpstracker D/LocationService: onConnected
2831-2831/com.websmithing.gpstracker E/LocationService: position: -15.8340533, -47.9755165 accuracy: 76.5
2831-2831/com.websmithing.gpstracker D/dalvikvm: GC_FOR_ALLOC freed 446K, 6% free 9613K/10124K, paused 16ms, to
2831-2831/com.websmithing.gpstracker E/GpsTrackerActivity: checkIfWebsiteIsReachable onSuccess statusCode: 200
2831-2831/com.websmithing.gpstracker E/LocationService: sendLocationDataToWebsite onSuccess statusCode: 200
```

Figura 37 - Status de localização com o celular conectado em redes WLAN e antena GPS ativa.

Lembrando que os testes foram feitos em local fechado, o que atenua o sinal GPS, e com isso a precisão não melhorou consideravelmente. Observa-se o status “cancelAlarmManger” quando o botão “Tracking is off” foi selecionado no dispositivo.

Para entender como funciona a aplicação, deve-se analisar o funcionamento de todos os arquivos Java.

1. GpsTrackerActivity – Este arquivo define a tela do aplicativo, onde existem dois campos de textos que são editáveis, um para o nome de usuário e outro para o endereço do servidor web HTTP. Há também, um conjunto de opções que permite o usuário a definir o intervalo de rastreo, que varia de 1 a 60 minutos e um botão de que ativa o rastreamento do dispositivo. Na figura 38, ilustra a tela principal da aplicação no dispositivo móvel.

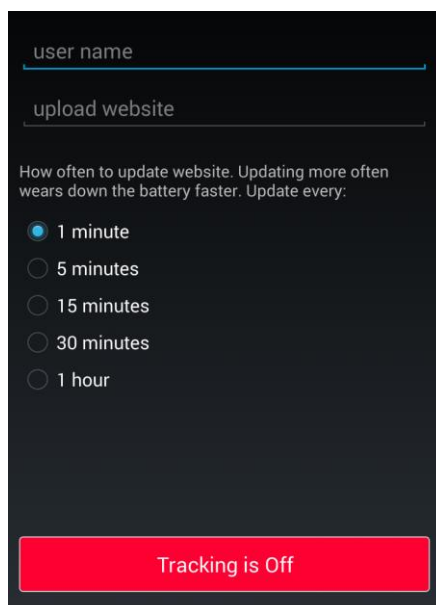


Figura 38- Tela do aplicativo.

O funcionamento desta parte da aplicação está baseada na atividade principal que é iniciada no loop *onCreate*, onde um arquivo é aberto , o *sharedPreference* e a variável *currentlyTracking* é definida. Esta é uma variável importante e será usada em todo o aplicativo. Quando o *onResume* é chamado, outros dois métodos são chamados. O primeiro é o *displayUserSettings* que tem a finalidade de restaurar a interface do usuário para seu estado anterior, a outra é o *setTrackingButtonState* que exibe o estado atual do botão de rastreamento, se está ativado ou não. Quando um usuário ativa o rastreamento pelo botão, a variável *currentlyTracking* é definida como verdadeira e o método *startAlarmManager* é chamado, como se pode observar nas figuras 39 e 40.

```
private void startAlarmManager() {
    Log.d(TAG, "startAlarmManager");

    Context context = getBaseContext();
    alarmManager = (AlarmManager)context.getSystemService(Context.ALARM_SERVICE);
    gpsTrackerIntent = new Intent(context, GpsTrackerAlarmReceiver.class);
    pendingIntent = PendingIntent.getBroadcast(context, 0, gpsTrackerIntent, 0);

    alarmManager.setRepeating(AlarmManager.ELAPSED_REALTIME_WAKEUP,
        SystemClock.elapsedRealtime(),
        intervalInMinutes * 60000, // 60000 = 1 minuto
        pendingIntent);
}
```

Figura 39 - Função startAlarmManager.

```

if (currentlyTracking) {
    Toast.makeText(getApplicationContext(), R.string.tracking_has_now_stopped, Toast.LENGTH_LONG).show();

    cancelAlarmManager();

    currentlyTracking = false;
    editor.putBoolean("currentlyTracking", false);
    editor.putString("sessionID", "");
} else {
    Toast.makeText(getApplicationContext(), R.string.tracking_has_now_started, Toast.LENGTH_LONG).show();

    startAlarmManager();

    currentlyTracking = true;
    editor.putBoolean("currentlyTracking", true);
    editor.putFloat("totalDistanceInMeters", 0f);
    editor.putBoolean("firstTimeGettingPosition", true);
    editor.putString("sessionID", UUID.randomUUID().toString());
}

```

Figura 40 - Loop que define a variável de rastreamento.

Neste método, um *AlarmManager* é criado e o intervalo de tempo é definido na variável *intervalInMinutes*. Esta variável é definida pelo usuário ao escolher as opções de tempo de rastreamento presentes na tela do aplicativo. O *AlarmManager* é uma classe de interface para serviços de sistema de alarme do telefone e permite executar o código em um momento específico. Neste método, cria-se uma intenção pendente que permite o *AlarmManager* executar o código usando permissões definidas pela aplicação. Em resumo, o que ocorre é que o *AlarmManager* irá executar o método *OnReceive* da classe *GpsTrackerAlarmReceiver* de acordo com o tempo definido pelo usuário para rastreio.

2. *GpsTrackerAlarmReceiver* - Esta classe tem um método *OnReceive* que é chamado periodicamente a partir de um *AlarmManager*. Quando o método é chamado, o serviço é iniciado em segundo plano. Esta classe está diretamente ligada a classe auxiliar *WakefulBroadcastReceiver* que cria um bloqueio despertar parcial que garante que o *cpu* não fique em modo *standby* enquanto o serviço de fundo está em execução. Essa permissão é definida no arquivo *AndroidManifest*.
3. *LocationService* - Este serviço usa os serviços de localização do Google Play para obter a localização do dispositivo e enviá-la para o site de atualização do servidor usando uma biblioteca cliente HTTP.. O serviço, em seguida, encerra em si. Essa classe é executada na *thread* principal do aplicativo rodando em segundo plano e continua mesmo se o aplicativo é desligado. Uma vez que é executado na *thread* principal, todas as operações de longa execução devem ser executadas simultaneamente para

se evitar possíveis “travamentos” do sistema provenientes de varias aplicações ocorrendo ao mesmo tempo.

- A operação de obter a localização a partir do Google Play utiliza o método *requestLocationUpdates* da classe *LocationClient* para começar a receber dados de localização do telefone. Quando o telefone tem uma localização que atenda aos critérios, logo em seguida, ele retorna esse local de volta para o método *OnLocationChanged*. Este processo ocorre de maneira assíncrona para evitar possíveis “travamentos” da operação.
- A operação de enviar as informações para o servidor utiliza uma biblioteca *LoopjHttpClient* que permite fazer chamadas HTTP de forma assíncrona e manipular para ser chamada a partir de um serviço de fundo. Na figura 40 pode-se observar o código referente ao descrito acima.

```
public class LoopjHttpClient {
    private static AsyncHttpClient client = new AsyncHttpClient();

    public static void get(String url, RequestParams params, AsyncHttpResponseHandler responseHandler) {
        client.get(url, params, responseHandler);
    }

    public static void post(String url, RequestParams requestParams, AsyncHttpResponseHandler responseHandler) {
        client.post(url, requestParams, responseHandler);
    }
}
```

Figura 41 - Código LoopjHttpClient.

Assim, quando o serviço é iniciado, o *onCreate* é chamado e, em seguida, o *onStartCommand*. Esta operação é chamada para evitar a criação de outro serviço se, por acaso, outra operação tentar iniciá-lo. Então é iniciada a operação *startTracking*, como se pode observar na Figura 42.

```
private void startTracking() {
    Log.d(TAG, "startTracking");

    if (GooglePlayServicesUtil.isGooglePlayServicesAvailable(this) == ConnectionResult.SUCCESS) {
        locationClient = new LocationClient(this, this, this);

        if (!locationClient.isConnected() || !locationClient.isConnecting()) {
            locationClient.connect();
        }
    } else {
        Log.e(TAG, "unable to connect to google play services.");
    }
}
```

Figura 42 - Iniciando o startTracking.

A variável *currentlyProcessingLocation* controla para que o *startTracking* não seja chamado duas vezes, quando se programa para o rastreo ocorrer a cada minuto, pois pode demorar mais de um minuto para conseguir a primeira localização de satélite quando se ativa o GPS do celular.

Quando o *AlarmManager* chama o método *OnReceive*, inicia-se o serviço de fundo *LocationService* logo após. Se o intervalo de rastreo estiver definido para um minuto, então é chamado o método *onStartCommand* na classe *LocationService*. Enquanto o telefone estiver em execução, o *AlarmManager* continuará a iniciar esse serviço quantas vezes forem necessárias, para obter a localização do telefone e, em seguida, enviar o local para o seu servidor através do arquivo *updateLocation.php*.

4. *GpsTrackerBootReceiver* - Essa classe também possui um método *OnReceive* que é chamado, mas este é chamado quando o telefone é reiniciado. Quando o método é chamado, ele cria um novo *AlarmManager*. O *AlarmManager* inicia o serviço de fundo periodicamente. Esta classe se estende ao *BroadcastReceiver* e está relacionada com outros aplicativos do dispositivo com a finalidade de impedir que a aplicação pare se caso elas sejam chamadas como, por exemplo, quando o celular receber uma ligação.

O próximo passo é a criação do *locationClient*. Quando o cliente está conectado, cria-se o *LocationRequest* no método de retorno *OnConnected*.

O intervalo definido é de 1 segundo e a prioridade é *PRIORITY_HIGH_ACCURACY*, ou seja, obter a maior precisão possível em menor tempo. Então, o telefone irá obter a sua localização a partir do Google play proveniente de qualquer método de rastreamento (GPS, WLAN, 3G) com precisão de no máximo 100 metros. Na Figura 43, observa-se o *loop* responsável por obter a localização do dispositivo.

```
public void onLocationChanged(Location location) {
    if (location != null) {
        Log.e(TAG, "position: " + location.getLatitude() + ", " + location.getLongitude() + " accuracy: " + location.getAccuracy());

        if (location.getAccuracy() < 100.0f) {
            stopLocationUpdates();
            sendLocationDataToWebsite(location);
        }
    }
}
```

Figura 43 - Obtendo a localização.

No loop *sendLocationDataToWebsite* é verificado se o parâmetro *firstTimeGettingPosition* é setado. Se for, então se salva a latitude e longitude, se não, cria-se um objeto de localização com a latitude e longitude salvas anteriormente. Quando os parâmetros forem definidos, usa-se o método *distanceTo* da classe *Location* para descobrir a distancia desde a última atualização local através do incremento da variável *distance*. Com isso é possível calcular a distância percorrida, como se pode observar na figura 44.

```

boolean firstTimeGettingPosition = sharedPreferences.getBoolean("firstTimeGettingPosition", true);
if (firstTimeGettingPosition) {
    editor.putBoolean("firstTimeGettingPosition", false);
} else {
    Location previousLocation = new Location("");
    previousLocation.setLatitude(sharedPreferences.getFloat("previousLatitude", 0f));
    previousLocation.setLongitude(sharedPreferences.getFloat("previousLongitude", 0f));

    float distance = location.distanceTo(previousLocation);
    totalDistanceInMeters += distance;
    editor.putFloat("totalDistanceInMeters", totalDistanceInMeters);
}

```

Figura 44 - Calculando a distância percorrida.

Quando o *AlarmManager* atinge seu intervalo novamente, o serviço de fundo é recriado e todo o processo começa novamente. Este processo tem maior precisão em intervalos de rastreo maiores que cinco minutos, em intervalos de um minuto, pode não ser possível que os serviços de localização do *Google Play* obtenham tempo suficiente para uma correção local que atenda os critérios.

Com isso o loop de todo o processo é reiniciado de acordo com o tempo que o usuário definir na tela do aplicativo. Em resumo, todo o funcionamento da aplicação pode ser melhor compreendido de acordo com o fluxograma da Figura 45.

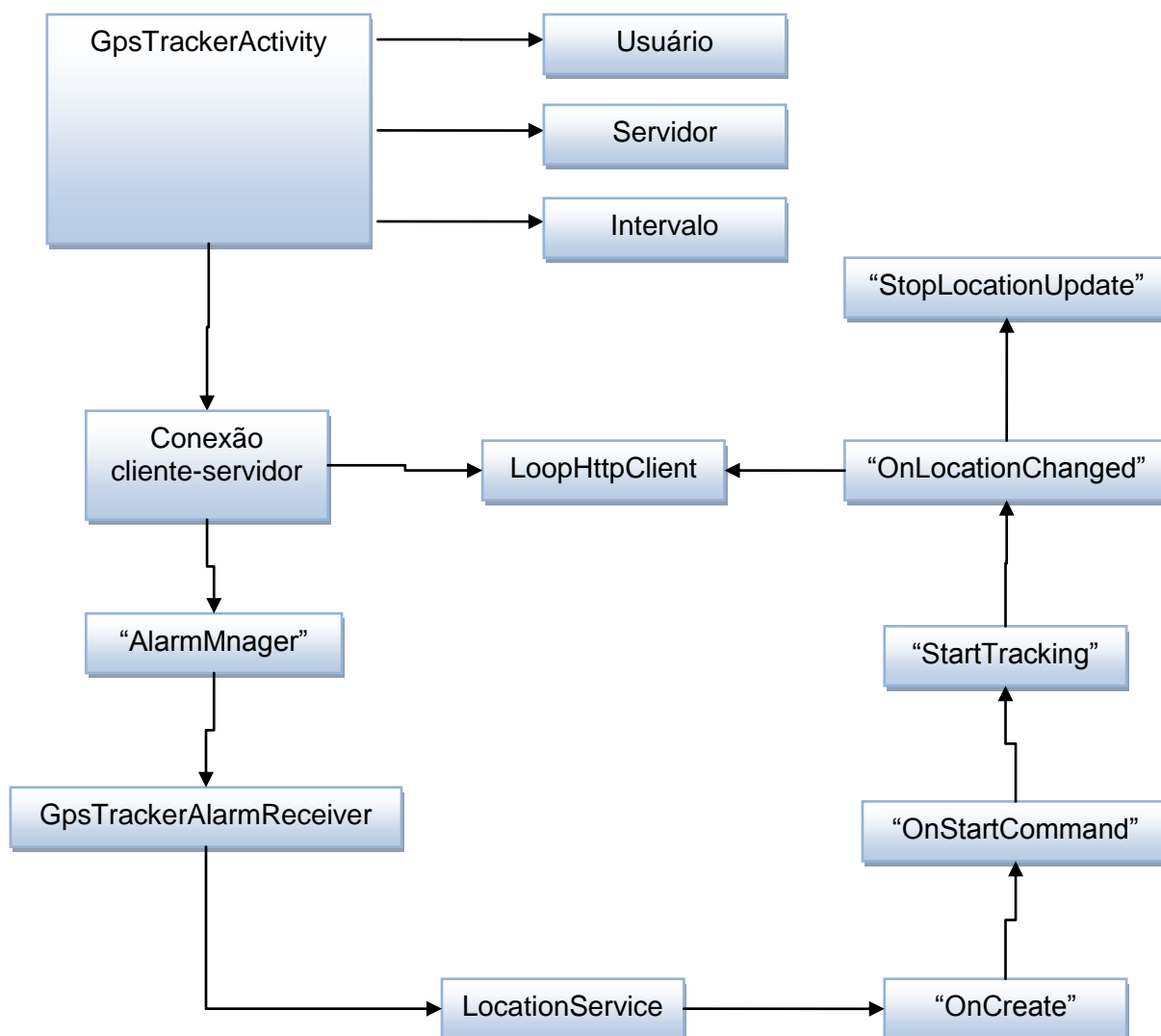


Figura 45 - Fluxograma de funcionamento da aplicação (cliente).

4.1.3. Alimentação do Dispositivo

O agente de rastreamento apresentado será um dispositivo móvel (telefone celular) e, para que ele possa estar em constante funcionamento, é necessária uma fonte externa de alimentação para que possa carregar a bateria do telefone celular. Este *hardware* responsável pelo carregamento deverá estar localizado em local discreto no veículo, como por exemplo, atrás do painel. Já que se trata de rastreamento veicular, pode-se utilizar a bateria do próprio veículo como fonte externa, necessitando apenas de um circuito que liga o telefone celular diretamente a bateria. Foi proposta então, neste caso, uma alimentação proveniente do sistema de som do veículo, que é de 12V.

Na Figura 46 tem-se um esquemático do circuito projetado, simulado utilizando o *software* CircuitMaker, comprovando que na saída temos aproximadamente 5V.

Pode-se observar na Figura 46 o regulador de tensão (7805A), responsável por regular a tensão de alimentação de entrada de 12 V para 5 V (tensão ideal para carregar a bateria do telefone celular). O regulador de tensão tem uma característica importante: mesmo havendo variações na tensão de entrada ou na corrente de saída, a tensão de saída sempre ficará constante, o que evita possíveis instabilidades no comportamento da tensão e ocasiona diminuição da vida útil da bateria. Os capacitores são responsáveis por filtrar as tensões de entrada e saída, reduzindo os ruídos de RF e tornando a tensão de saída mais estável. Por fim, colocou-se um LED (*Light emitter diode*) em associação com um resistor para efeitos estéticos, indicando o funcionamento do circuito. Pode-se observar o circuito completo na Figura 47.

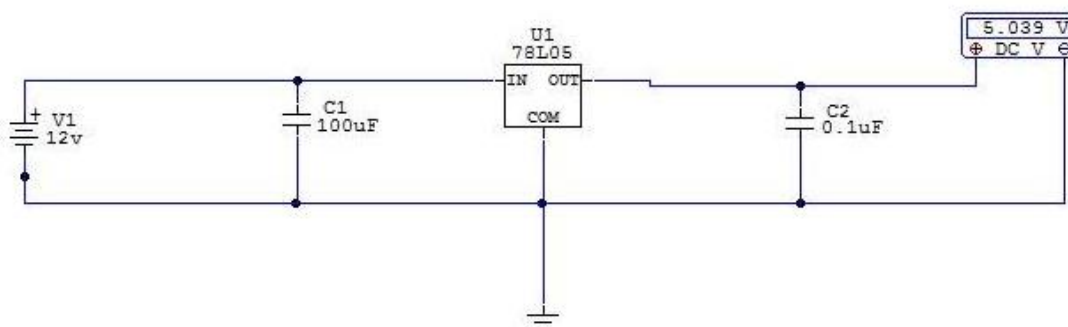


Figura 46 – Esquemático do circuito simulado.

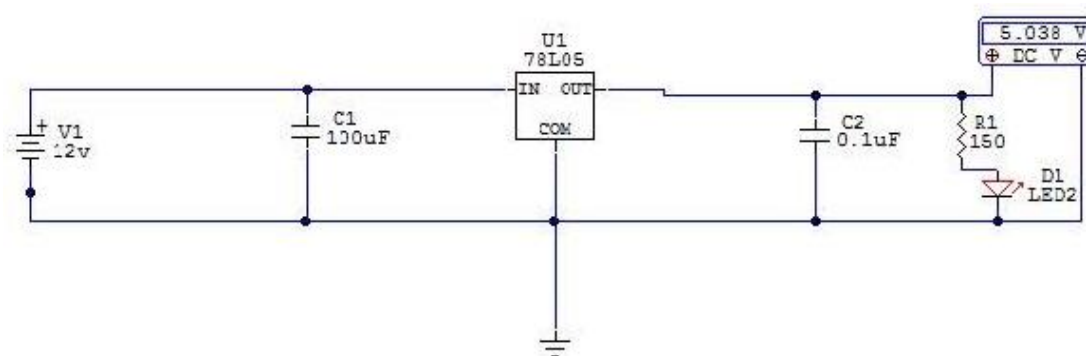


Figura 47 – Esquemático completo do circuito

Em uma análise nodal, capacitores tem comportamento de circuito aberto em tensão contínua, portanto a corrente I_{R1} (corrente que passa pelo resistor R1) é calculada:

$$I_{R1} = \frac{V_{out} - V_{LED}}{R1} = \frac{(5 - 2.1)V}{150\Omega} = 19.3mA$$

Para carregar uma bateria é recomendável que a corrente que a alimenta seja de aproximadamente 70% de sua capacidade total para baterias acima de 2.9V. Como a tensão das baterias dos telefones celulares, atualmente, são de 3.7V, este carregador é viável para baterias de capacidade total abaixo de 2100 mAh (Eletronics LAB, 2014).

Sabendo que a corrente máxima de saída do regulador de tensão, de acordo com o *datasheet* disponível no anexo XI, é de 1.5 A. Então a corrente que irá alimentar a bateria do telefone celular é:

$$I_B = I_{OUT} - I_{R1} = 1.5 - 19.3mA = 1.48A$$

Para o cálculo da potência dissipada P_D no regulador de tensão, afim de projetar um circuito que opere de forma ideal, considera-se a seguinte equação que se encontra no Anexo XI:

$$P_D = (V_{OUT} - V_{IN}) \times I_{OUT} = (12 - 5) \times 1.5 = 10.5W$$

Portanto, como a potência dissipada máxima é de 15W (TEXAS INSTRUMENTS, 2014), o circuito opera de forma ideal.

Na figura 48 tem-se o circuito montado. Na figura 49, observa-se o funcionamento do circuito carregando o dispositivo móvel.

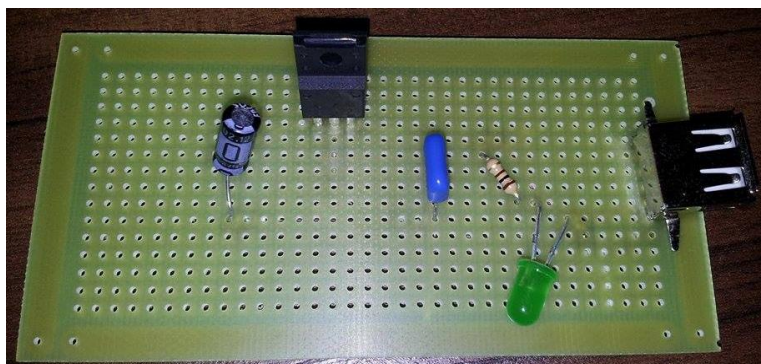


Figura 48 – Circuito montado.

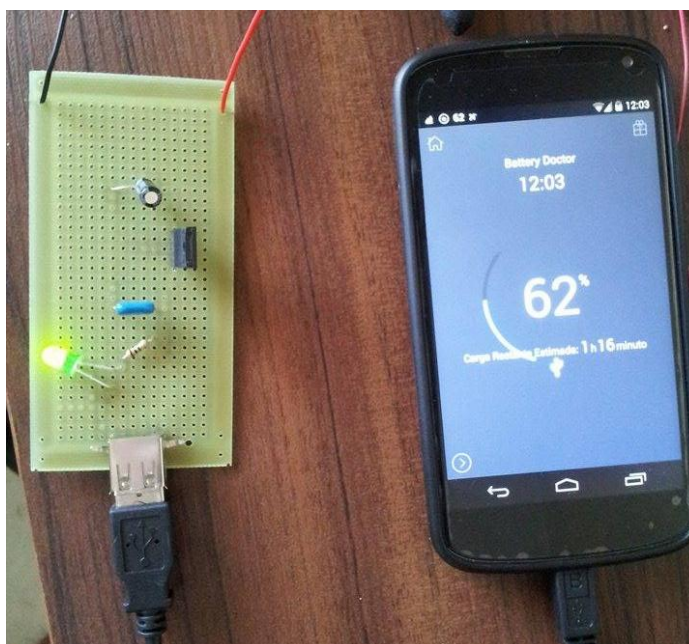


Figura 49 – Circuito em funcionamento carregando o telefone celular.

4.2. VALIDAÇÃO EXPERIMENTAL DA PROPOSTA

4.2.1. Resultados esperados para o rastreamento

De posse dos dados obtidos, é necessário realizar uma análise dos resultados em campo a fim de compará-los com os obtidos experimentalmente.

Experimentalmente, o objetivo era obter a melhor precisão possível, ou seja, próxima de zero metro. Mas, em campo, obteve-se uma precisão que varia de acordo com as configurações estabelecidas no dispositivo móvel, onde a localização se tornava mais precisa quando havia uso de redes mais rápidas e com habilitação da antena GPS do celular. Esta precisão poderia sofrer uma melhora de até 37% se configurada nos meios mais efetivos. Na tabela 11, observam-se os dados coletados a respeito do descrito acima.

Tabela 11 – Dados de precisão coletados em diferentes configurações do celular

| Testes | Velocidade de <i>Download</i> | Velocidade de <i>Upload</i> | Localização (<i>Wi-fi</i>) | Localização (<i>GPS</i>) | Precisão |
|----------------|-------------------------------|-----------------------------|------------------------------|----------------------------|----------|
| Teste 1 | 13,93 Mb/s | 1,45 Mb/s | Sim | Não | 45 m |
| Teste 2 | 13,92 Mb/s | 1,45 Mb/s | Sim | Sim | 28 m |
| Teste 3 | 475 Kb/s | 80 Kb/s | Sim | Não | 54 m |
| Teste 4 | 475 Kb/s | 80 Kb/s | Sim | Sim | 39 m |

Ainda, experimentalmente, o objetivo era que o servidor se comunicasse com o celular a partir de qualquer lugar com cobertura de redes móveis. Para tanto, seria necessário adquirir um *IP* mundial para o servidor, onde seria possível obter a localização do celular em qualquer lugar do planeta. Como não foi possível adquirir este *IP*, o uso de servidores locais (*localhost*) foi suficiente, como prova de conceito em campo, para justificar a veracidade desta premissa.

Com isso é possível aplicar estas condições para o estudo de caso do *Campus* UnB-Gama, onde o furto de veículos tornou-se algo frequente. Com o dispositivo móvel presente no veículo e em constante comunicação com o servidor local, é possível inferir que o veículo foi furtado quando a conexão entre servidor e cliente for cessada.

4.2.2. ANÁLISE DOS RESULTADOS

Dadas as condições apresentadas anteriormente, o resultado obtido para o servidor que comunicará com o dispositivo móvel tem como visualização inicial a Figura 50.

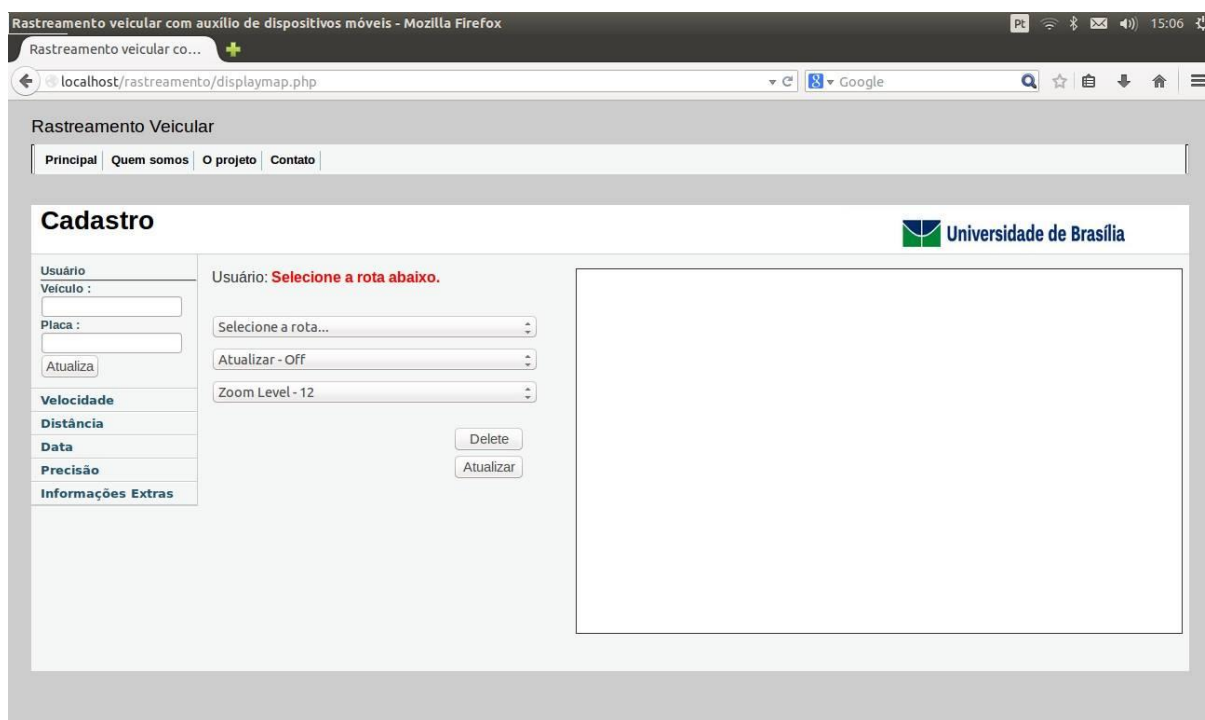


Figura 50. Visualização inicial da página web do servidor

O botão “Delete” tem a finalidade de excluir uma rota armazenada no banco de dados. Caso seja pressionado este botão sem a escolha de uma rota, retorna-se uma mensagem de erro, como se pode observar na Figura 51.

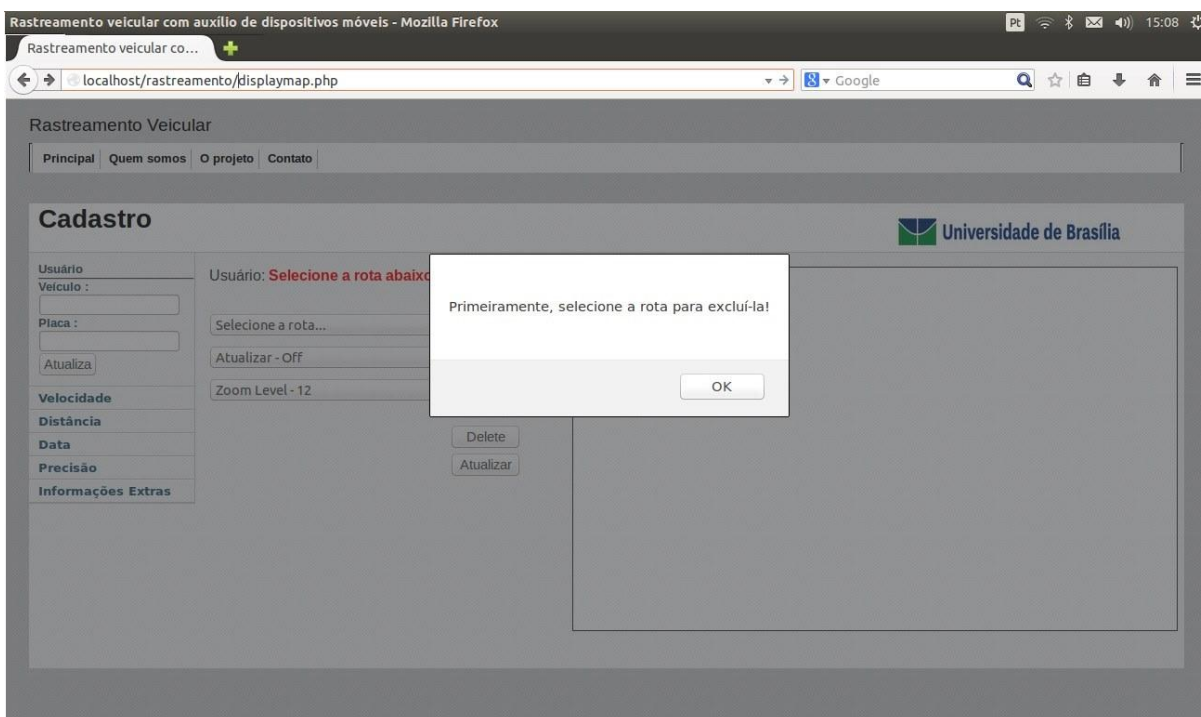


Figura 51 - Mensagem de erro ao selecionar o botão “delete” tem uma rota selecionada.

O botão “Atualizar” é selecionado quando se deseja atualizar a localização manualmente. Caso este botão seja pressionado sem ter selecionado uma rota, retorna-se uma mensagem de erro semelhante ao caso do botão “Delete” como mostra a Figura 52

Selecione a rota primeiramente.

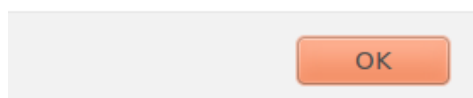


Figura 52 - Erro ao pressionar o botão “atualizar” quando não há rota selecionada.

A aba que especifica o tempo de atualização automática da rota, ao ser pressionada sem uma rota estabelecida, retorna o mesmo erro anterior. O mesmo ocorre ao selecionar a aba em que se define o *zoom* do mapa.

Ao selecionar a rota estabelecida por um cliente cadastrado no banco de dados, onde o nome de usuário é definido no dispositivo móvel através do aplicativo *android*, tem-se o resultado na Figura 53.

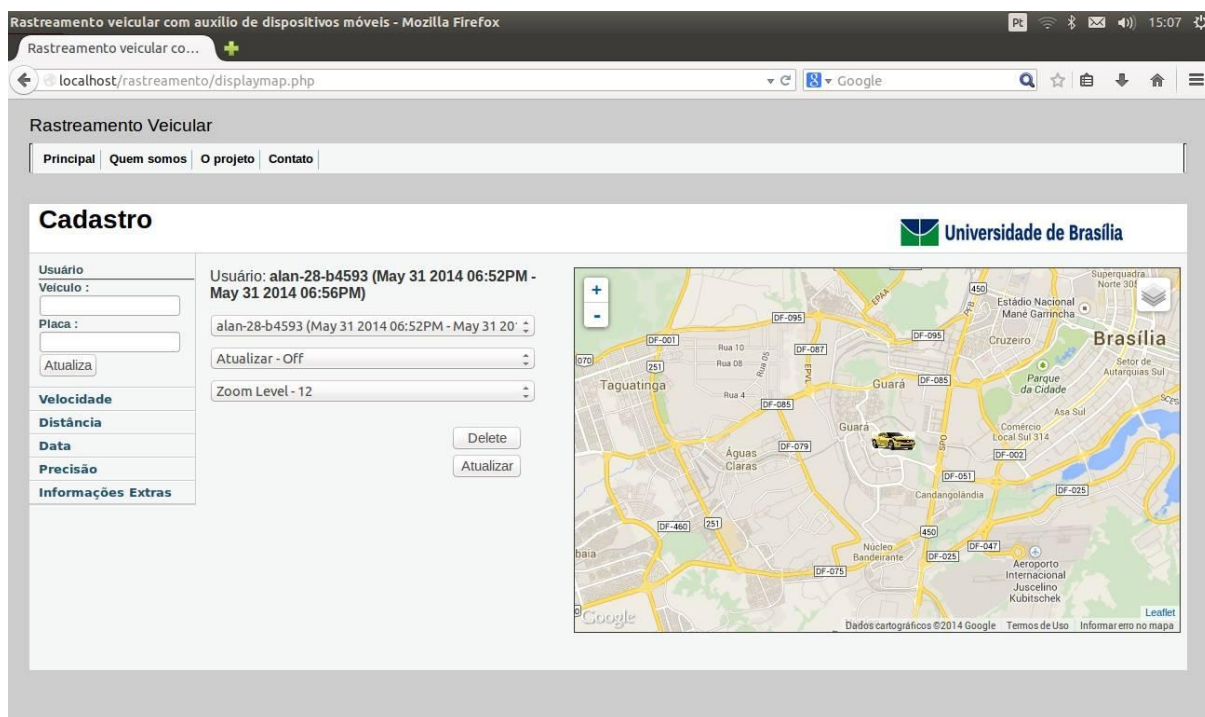


Figura 53 - Rota selecionada de acordo com o usuário cadastrado.

Ao selecionar o ícone do veículo no mapa, obtêm-se informações relacionadas ao tempo em que o dispositivo móvel, e conseqüentemente o veículo, foi monitorado. As informações são: velocidade, distância percorrida, data em que o veículo parou de ser monitorado, usuário, precisão e informações extras.

A Figura 54 mostra essas informações:

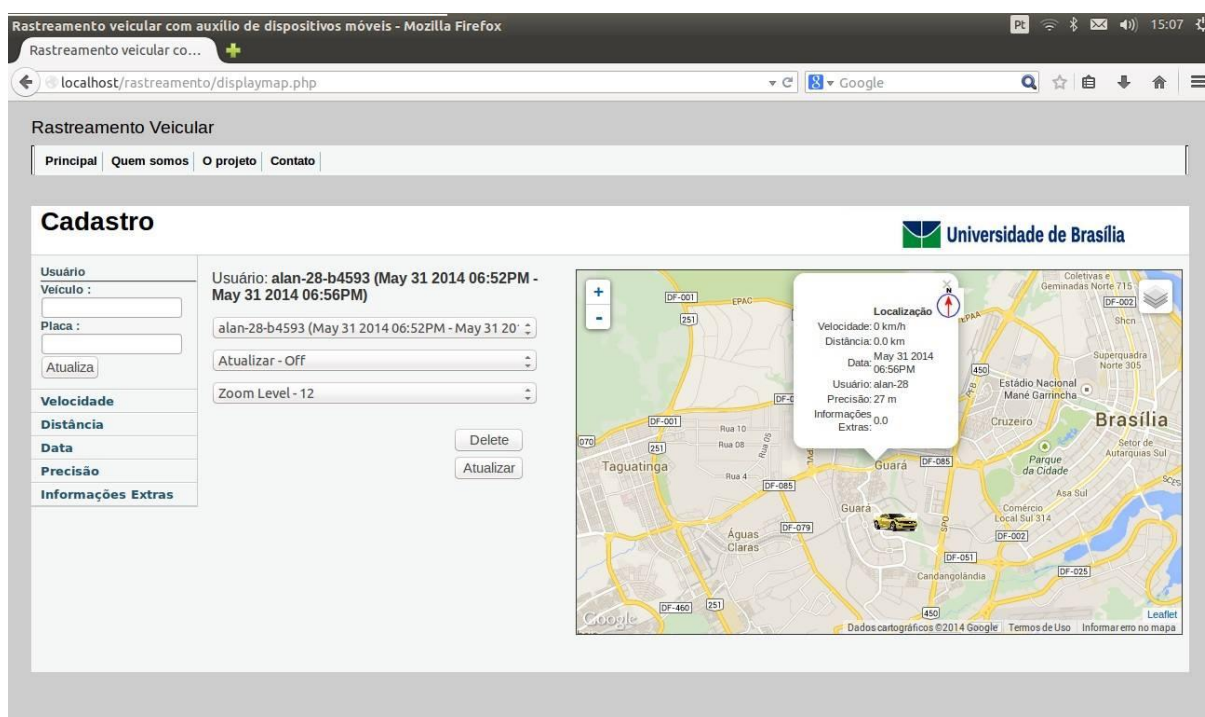
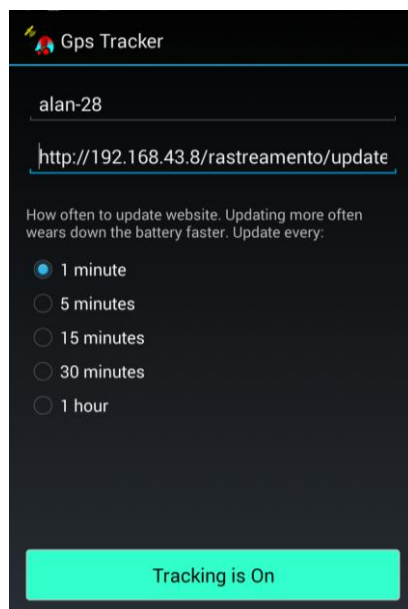


Figura 54 - Informações do veículo.

No dispositivo móvel, coloca-se o nome de usuário e o endereço do servidor web local (<http://192.168.43.8/rastreamento/updateslocation.php>), como se pode ver na Figura 55.



- Figura 55 - Definindo o usuário e o endereço de serviço no celular.

Parte V
CONCLUSÃO

5. CONCLUSÃO

O crime relacionado ao furto e roubo de veículos vem crescendo de forma preocupante desde que a tecnologia automotiva se inseriu no mercado e se tornou mais acessível para a população. Por outro lado, a tecnologia GPS foi criada e desenvolvida e com isso, pode ser usada em conjunto com as redes de telefonia móvel para auxiliar na procura por veículos roubados, que é um bem durável de alto custo.

As tecnologias GPS foram desenvolvidas a partir de sistemas GNSS que descrevem o posicionamento dos satélites artificiais em órbita no planeta. A tecnologia NAVSTAR GPS – ou apenas GPS, como é conhecido popularmente –, é a mais utilizada por dispositivos de rastreamento. A disposição em órbita dos satélites permite que a localização seja estabelecida com o auxílio de pelo menos quatro satélites na superfície terrestre. O monitoramento de todas as ações dos satélites é controlado através de estações de controle espalhadas pelo mundo. Em conjunto, receptores GPS se comunicam com os satélites para finalizar o processo de localização georreferenciada. O GLONASS é a tecnologia russa e é utilizada em alguns dispositivos. Os satélites estão dispostos em um plano orbital tridimensional, o que torna a comunicação com os receptores mais precisa. O monitoramento dos satélites é de responsabilidade da estação de controle russa. Neste projeto, o dispositivo usado como agente de rastreamento possui ambas as tecnologias estudadas, o que melhora a eficiência de todo o sistema.

Foi apresentado como proposta deste projeto um sistema de rastreamento e localização de veículos que possui um servidor *web* implementado no *Raspberry Pi*. Este servidor é localizado na central de controle, com o objetivo de rastrear veículos através de dados recebidos por um dispositivo móvel localizado no veículo. Este dispositivo deverá ser equipado com tecnologia GPS e acesso às redes de dados *WLAN* e *WWAN*. O método de localização consiste no envio de dados do dispositivo, localizado no veículo, para as estações base de rede de telefonia móvel que, através da triangulação de antenas, e o auxílio do *GPS* envia à central de controle a localização do dispositivo.

Com a proposta apresentada, foi necessária a realização de alguns testes com os possíveis tipos de redes para avaliar a influência de cada uma na

transmissão de dados. Foram monitoradas as redes 3G de um celular da operadora GSM 2 e rede WLAN residencial, com o auxílio do *software* SIMET. Os parâmetros (latência, jitter, vazão UDP e tráfego TCP) foram avaliados com diversas amostras em diferentes tempos para estudo do comportamento de cada rede em longo prazo. Esta avaliação foi baseada em cálculos estatísticos de média e desvio padrão das amostras para avaliar a estabilidade de cada cenário e, após a construção de gráficos foi possível avaliar cada parâmetro e concluir que as redes WLAN e o protocolo referente a comunicação e tráfego TCP são mais eficientes para o projeto em questão.

Portanto, ao definir o tipo de rede WLAN para o rastreamento, foi necessário aplicá-la na metodologia do projeto. Assim, o sistema cliente - servidor funcionou de forma satisfatória e atendeu todos os requisitos esperados. O servidor se comunica de forma efetiva com o celular (cliente), interpretando sua localização através da *API* do *Google Maps*, validando assim a proposta inicial do projeto. Percebe-se que a precisão no rastreamento do dispositivo está diretamente ligada à configuração do mesmo, em que a velocidade de transferência dos dados móveis e o auxílio do sistema *GPS* estão proporcionalmente ligados à uma maior efetividade do sistema, como mencionado na Tabela 11. Por este motivo, o dispositivo móvel, sempre que possível, deverá estar conectado a uma rede *WLAN* e com o sistema *GPS* habilitado, tornando o rastreamento mais preciso. Então, quando o dispositivo não estiver em zona de cobertura *WLAN*, a conexão se altera automaticamente para rede *WWAN*, que apesar de ser menos precisa, ainda atende as especificações de rastreamento do projeto. Esta ação somente é possível se o servidor for acessível em qualquer máquina, através de um IP mundial.

Portanto, como o servidor tem suas funcionalidades limitadas em *localhost*, o monitoramento dos veículos ocorrerá através da comunicação entre o dispositivo e o servidor se ambos estiverem conectados a mesma rede *WLAN*. Como o servidor é local, sua área é delimitada através da cobertura de sinal dos pontos de acesso do campus UnB-Gama instalados para este fim. Quanto maior o número de pontos de acessos *WLAN* interligados, maior a área coberta para o rastreamento e monitoramento dos veículos.

A partir do momento em que a comunicação entre cliente-servidor for cessada, o usuário poderá interpretar que o seu veículo foi furtado. Com isso, ele

poderá tomar medidas junto às autoridades pertinentes. Por fim, como o objetivo inicial do projeto se resume na criação de um sistema para rastreamento de veículos com o auxílio de dispositivos móveis, subentende-se que esse foi atingido com sucesso, mesmo que ainda haja algumas limitações.

4.1. TRABALHOS FUTUROS

Como proposta de trabalhos futuros a fim de aprimorar ainda mais o projeto, pode-se levantar algumas questões importantes:

- Adicionar novas funcionalidades ao mapa do *website*, como por exemplo, um aviso de alerta quando o veículo começar a se mover ou se ultrapassar alguma área delimitada pelo usuário e habilitar o cadastro de usuários em conexão com o banco de dados, onde pode se inserir fotos e dados de identificação mais detalhados.
- Expandir a área de rastreamento, com o sistema funcionando fora do espaço da FGA. Para isso, necessita-se somente de um IP mundial, para que o *website* possa ser acessado a partir de qualquer dispositivo com acesso a internet, e não somente em *localhost*.
- Implementar um servidor de modo que este seja mais seguro, configurando-o com senhas pessoais e com o uso do protocolo HTTPS (*Hyper Text Transfer Protocol Secure*), que garante mais segurança em relação a possíveis invasões no servidor.
- Implementar outro Raspberry Pi como agente de rastreamento, com localização discreta dentro do veículo. Este dispositivo deverá possuir hardware externo de antenas GPS e modem com acesso a redes 3G para se comunicar com outro Raspberry Pi, sendo este o servidor e possuindo a mesma função do abordado neste projeto.

REFERÊNCIAS

3GPP. **Technical Specification 36.300: Evolved Universal Terrestrial Radio Access (E-ULTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN)**. [S.l.], 2010.

CARROS MAIS ROUBADOS. Disponível em: <http://www.roubadosbr.com.br/mais_roubados.php>. Acesso em 08/11/2013.

CEUB/ICPD. **Instituto CEUB de Pesquisa e Desenvolvimento**. Disponível em: <<http://www.ufscar.br/~debe/geo/paginas/tutoriais/pdf/gps/GPS%20e%20Cartografia%20Basica.pdf>>. Acesso em 02/08/2014.

RASTREAMENTO VEICULAR. Disponível em: <<http://www.multirastreamento.com.br/>>. Acesso em 16/05/2014.

RASTREAMENTO VEICULAR. Disponível em: <http://www.concox.com.br/products/GPS_tracking_system.html>. Acesso em 16/05/2014.

O'HARA, BOB; PETRICK, AL. **802.11 Handbook – A Designer's Companion**. 1^o ed. New York: The Institute of Electrical and Electronics Engineers, Inc. Jan 1999.

SILVA, DIEGO CESAR VALENTE E. **Análise experimental da aplicação da tecnologia de radio sobre fibra em redes IEEE 802.15.4**. Disponível em: <http://www.bibliotecadigital.puc-campinas.edu.br/tde_busca/arquivo.php?codArquivo=532>. Acesso em 20/11/2013.

REDES WWAN. Disponível em: <<http://comum.rcaap.pt/bitstream/123456789/4983/1/Politecnica%201.pdf#page=14>>. Acesso em 10/11/2013.

SVERZUT, J.U. **Redes GSM, GPRS, EDGE e UMTS, Evolução a Caminho da Terceira Geração (3G)**. São Paulo: Ed. Érica, 2005.

TESTE MOSTRA VELOCIDADE CONEXÃO 4G BRASÍLIA. Disponível em: <<http://tecnologia.uol.com.br/noticias/redacao/2013/05/13/teste-mostra-desempenho-do-4g-nas-seis-cidades-sede-da-copa-das-confederacoes.htm>>. Acesso em 10/11/2013.

SERVIDOR WEB. Disponível em: <<http://pt.slideshare.net/raphaelamarques/servidor-web-8225869>>. Acesso em 08/11/2013.

RAPPAPORT, T. S. **Comunicações sem fio, princípios e práticas**. Segunda edição. [S.l.]:Pearson, 2009.

ROUBO E FURTO DE VEÍCULOS. Disponível em: <<http://g1.globo.com/sao-paulo/noticia/2013/05/roubo-e-furto-de-veiculos-crescem-em-sp-e-recuperacao-fica-estagnada.html>>. Acesso em 08/11/2013.

TCPING, 2007. **The Network Impact on Application Performance**. Disponível em: <http://www.tcping.com/files/wp_performance_metrics.pdf>. Acesso em 01/08/2014.

Wifi & Sécurité. Disponível em: <http://reseau-wifi.blogspot.com.br/2006_11_01_archive.html>. Acesso em 01/08/2014.

NINJA. **Núcleo Infanto-juvenil de Aviação**. Disponível em: <<http://ninja-brasil.blogspot.com.br/2010/09/navegacao-aerea.html>>. Acesso em 02/08/2014.

SIMET. **Sistema de Medição de Trafégo Internet**. Disponível em: <<http://simet.nic.br/>>. Acesso em 18/07/2014.

PETER H. Dana. **Global Positioning System Overview**. Disponível em: <http://www.colorado.edu/geography/gcraft/notes/gps/gps_f.html>. Acesso em 05/08/2014.

Heinrichs G., Germany G., et al. **A Hybrid Galileo/UMTSReceiver Architecture for Mass-Market Applications**. Disponível em: <http://www.gawain-eceivers.com/publications/lfEN_Paper_GAWAIN_GNSS2005.pdf>. Acesso em 04/08/2014.

Hofmann-Wellenhof B., Lichtenegger H., and Collins J. **Global Positioning System: Theory and Practice**. 5 ed. New York: Springer Verlag Wien.

MONICO, J.F.G. **Posicionamento pelo NAVSTAR-GPS: descrição, fundamentos e aplicações**. São Paulo: Editora UNESP, p287.

SEGANTINE, P. C. L. **GPS – Sistema de Posicionamento Global**. Apostila didática da Universidade de São Paulo Escola de Engenharia de São Carlos, Departamento de Transportes, 1999.

KELLY, C. BOECK. **Aplicação WEB para Consultas de Itinerários de Transporte Público com Visualização no Google Maps**. Dissertação (Graduação) — Universidade Regional de Blumenau, 2012.

LAMP Stack: Na Open-Source Architecture for Web Based Applications. Disponível em: <http://www.dixite.com/docs/lamp/dixite-lamp-en-0_3.pdf>. Acesso em 23/06/2014.

GeoBrasil. **Tutorial sobre Banco de Dados Geográficos**. Disponível em: <http://www.dpi.inpe.br/TutorialBdGeo_GeoBrasil2006.pdf>. Acesso em 01/08/2014.

Geoprocessamento. **Tópicos Avançados em Sistemas de Informações Geográficas**. Disponível em: <<http://pt.scribd.com/doc/42670298/Apostila-Geoprocessamento>>. Acesso em 25/07/2014.

GLONASS-ICD. **GLONASS Interface Control Document**. Disponível em: <http://www.glonass-center.ru/ICD02_e.pdf>. Acesso em 04/08/2014.

Fundamentos de Geoprocessamento. Disponível em: <<http://www.ltc.ufes.br/geomaticsce/Modulo%20Geoprocessamento.pdf>>. Acesso em 02/08/2014.

Google maps. Disponível em: <<https://developers.google.com/maps/licensing?hl=pt-br>>. Acesso em 28/07/2014.

GPRS e EDGE. Disponível em: <<http://www.hardware.com.br/dicas/gprs-edge.html>>. Acesso em 10/11/2013.

GPS TRACKER. Disponível em: <<https://www.websmithing.com/gps-tracker/>>. Acesso em 10/04/2014.

JAMHOUR. **Comunicação sem Fio WLAN (802.11)**. Disponível em: <<http://www.ppgia.pucpr.br/~jamhour/Pessoal/Atual/WLAN-Parte1.pdf>>. Acesso em 31/07/2014.

Eletronics LAB. **How To rebuild a li-lon battery pack**. Disponível em: <http://www.electronic-lab.com/articles/Li_lon_reconstruct/How%20to%20rebuild%20a%20Li-lon%20pack.pdf>. Acesso em 05/08/2014.

TEXAS INSTRUMENTS. **Datasheet A7800 series positive-voltage regulators**. Disponível em: <<https://www.sparkfun.com/datasheets/Components/LM7805.pdf>>. Acesso em 01/08/2014.

Parte VI
ANEXOS

- ANEXO I: Instalação do LAMP e criação do servidor web

Tendo em mãos um computador com o sistema operacional LINUX (no presente trabalho foi utilizado o Rasbian) instalar o pacote de softwares do LAMP é uma tarefa relativamente fácil, para executar esta tarefa segue-se a sequencia de passos abaixo:

1. Com o sistema operacional Rasbian em execução, abra o seu terminal (tecla de atalho ctrl+alt+t).
2. Digite o comando:

```
sudo apt-get install lamp-server
```

Após digitar o comando pressione a tecla *ENTER* e quando for solicitado, digite a sua senha de SU (Super Usuario).

3. Será feita então uma listagem no terminal com os pacotes necessários, digite a tecla "S" e tecla *ENTER* para confirmar a instalação.
4. Durante o processo de instalação do MySQL será pedido para criar uma senha para o administrador, digite a senha e tecla *ENTER*.

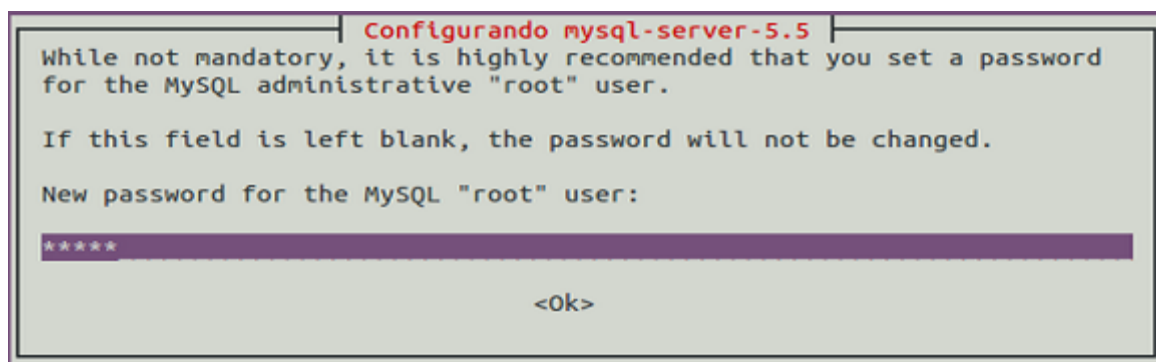


Figura 56. Criando senha do administrador do MySQL.

5. Novamente será pedido para digitar a senha, repita a senha digitada anteriormente para finalizar a instalação.

- ANEXO II: Teste para verificar funcionalidade dos pacotes instalados
-

- Teste Apache

Com o S.O. em execução, abra o navegador de sua preferência (no presente trabalho foi utilizado o Firefox), na barra de endereços digite:

```
http://localhost/
```

Caso o navegador apresente a página abaixo, o Apache está funcionando.

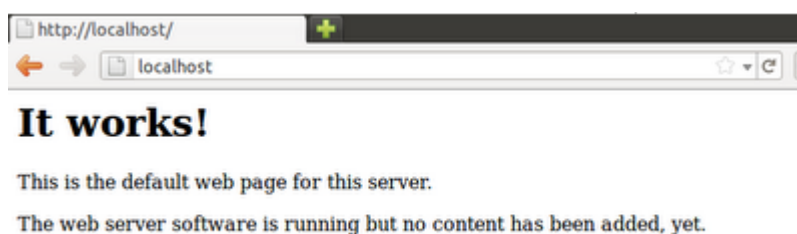


Figura 57. Apache em funcionamento.

Caso o navegador não apresente esta tela como saída, o Apache pode ser reiniciado com o comando abaixo:

```
sudo /etc/init.d/apache2 restart
```

- Teste PHP

Para fazer o teste do PHP, primeiro se cria um arquivo .php vazio no endereço:

```
/var/www
```

Posteriormente inserir o trecho de código abaixo, dentro desse arquivo criado.

```
<?php phpinfo(); ?>
```

Os passos descritos acima podem ser feitos diretamente pelo terminal, realizando esses dois comandos:

```
echo "<?php phpinfo(); ?>" | sudo tee /var/www/ teste.php  
sudo /etc/init.d/apache2 restart
```

Após realizado estes comandos, deve-se abrir o navegador e digitar o endereço:

```
http://localhost/teste.php
```

- Configurando o MySQL

Como se trata de um servidor local, o banco de dados MySQL deve usar o mesmo endereço *IP* que a máquina, no caso 127.0.0.1 (endereço de *IP* local). Isso pode ser configurado através do terminal:

```
cat /etc/hosts | grep localhost
```

Esse comando deve gerar a saída:

```
~$ cat /etc/hosts | grep localhost  
127.0.0.1 localhost  
:::1 ip6-localhost ip6-loopback
```

Outro ponto importante que deve ser verificado é o endereço de ligação, digitando o seguinte comando:

```
cat /etc/mysql/my.cnf | grep bind-address
```

Esse comando deve gerar a saída:

```
cat /etc/mysql/my.cnf | grep bind-address  
bind-address = 127.0.0.1
```

Caso alguma das saídas retornem um endereço de IP diferente do descrito acima o arquivo `my.cnf` deve ser editado:

```
sudo gedit /etc/mysql/my.cnf
```

A linha que contém “*bind-address*” deve ser editada, colocando o endereço 127.0.0.1.

- ANEXO III: Instalação e execução do Android Studio

Para instalação deste software, tem-se que seguir os passos citados abaixo:

1. Instalar o *Java JDK* com a versão 6 ou superior, para verificar a versão instalada execute o comando no terminal (*javac -version*). Caso haja a necessidade de instalá-lo:

```
sudo apt-get-repository ppa:webupd8team/Java
sudo apt-get update
sudo apt-get install oracle-java7-installer (versão 1.7.0_45)
```

2. Baixar o arquivo ***android-studio-bundle-<version>.tgz*** disponível no link abaixo:

```
http://developer.android.com/sdk/installing/studio.html
```

3. Após feito o download, o arquivo deve ser descompactado, dentro de uma pasta apropriada para as aplicações e de fácil acesso, preferencialmente. Para iniciar o Android Studio, abra o terminal e navegue até a pasta onde foi descompactado o arquivo, depois entre em:

```
Android-studio/bin/
```

4. Agora basta executar o arquivo *studio.sh*, assim ele será iniciado.

```
./studio.sh
```

5. Este software necessita da instalação do Github que é um servidor onde enviamos as versões do projeto, para instalá-lo execute o comando no terminal:

```
sudo apt-get install git-core
```

Depois de executar estes passos o Android Studio estará pronto para desenvolver qualquer aplicação para celulares Android.

- ANEXO IV: Configuração do Servidor no Raspberry PI

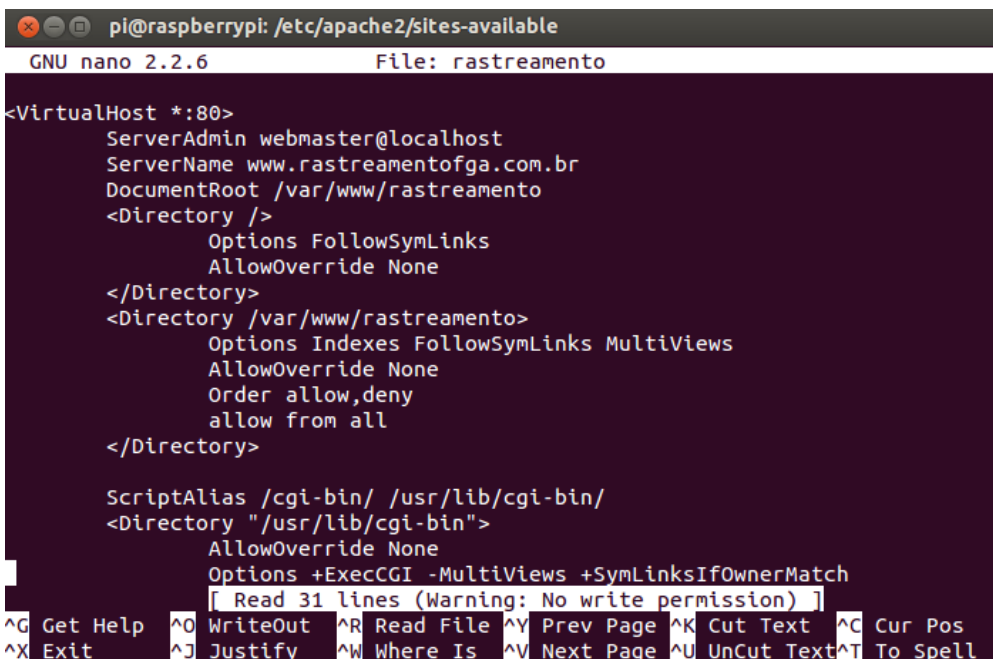
1. Criou-se um diretório chamado “rastreamento” em /www/var/ com os comandos:

```
sudo cd /www/var
sudo mkdir rastreamento
```

2. Ir ao diretório de configuração do apache (*sites-available*), criar um arquivo de texto chamado rastreamento com os comandos:

```
cd /etc/apache2/sites-available
sudo nano rastreamento
```

3. No arquivo criado, configurá-lo para que o site do servidor tenha endereço referente ao diretório criado no primeiro passo, através da porta 80 e possua um endereço em HTTP correspondente. O arquivo então será editado da seguinte forma:



```
pi@raspberrypi: /etc/apache2/sites-available
GNU nano 2.2.6 File: rastreamento
<VirtualHost *:80>
  ServerAdmin webmaster@localhost
  ServerName www.rastreamentofga.com.br
  DocumentRoot /var/www/rastreamento
  <Directory />
    Options FollowSymLinks
    AllowOverride None
  </Directory>
  <Directory /var/www/rastreamento>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    allow from all
  </Directory>

  ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
  <Directory "/usr/lib/cgi-bin">
    AllowOverride None
    Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
  </Directory>
</VirtualHost>
[ Read 31 lines (Warning: No write permission) ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Figura 58. Arquivo de configuração do servidor.

4. Salva-se o arquivo e cria-se um link simbólico no diretório `/etc/apache2/sites-enabled` como nome de rastreamento, afim de chamar o arquivo criado no passo anterior para que não haja necessidade de editar o arquivo contido em ambos os diretórios. Para isso, devem-se seguir os comandos:

```
cd /etc/apache2/sites-enabled
sudo ln -s ../sites-available/rastreamento rastreamento
```

5. Configurado o site do servidor, deve-se reiniciar o apache e ativar o servidor recém-criado com os comandos:

```
sudo /etc/init.d/apache2 restart
sudo ensite apache2
```

Feito o procedimento acima, o site está criado e poderá ser acessado por qualquer maquina ou dispositivo em “*localhost*”, digitando no navegador *web* o endereço `http://localhost/rastreamento` ou através do *ip* da maquina que contém o servidor habilitado, neste caso o *Raspberry Pi*. Se o servidor estiver conectado a uma rede *WLAN*, outros dispositivos que estiverem conectados na mesma rede poderão acessar o conteúdo web do servidor digitando no navegador `http://ip-wlan0-do-servidor/rastreamento`. Para descobrir o *ip* da rede *WLAN* deve-se digitar o comando no terminal:

```
sudo ifconfig wlan0
```

- ANEXO V: Instalação do banco de dados MySQL

Para entrar no banco de dados *mysql* e criar um banco de dados chamado *gpstracker* deve-se seguir os passos a partir das linhas de comando no terminal do *raspberrypi*:

```
mysql -h localhost -u root -p
```

(se, ao instalar o *mysql*, for configurada uma senha para usuário *root*, inserir a senha configurada)

```
create database gpstracker;  
exit;
```

Após criar o banco e sair, deve-se incluir os dados do arquivo do diretório “*mysql*” no banco de dados, para isso usam-se os comandos:

```
cd /var/www/rastreamento/mysql  
mysql -h localhost -u root -p gpstracker < gpstracker-03-14-14.sql
```

(se, ao instalar o *mysql*, for configurada uma senha para usuário *root*, inserir a senha configurada).

Se não houver nenhum erro, o banco de dados está instalado. O próximo passo é criar um usuário geral para que as informações do cliente sejam armazenadas. Para isso deve-se entrar no *mysql* e seguir com os comandos:

```
mysql -h localhost -u root -p gpstracker
```

(se, ao instalar o *mysql*, for configurada uma senha para usuário *root*, inserir a senha configurada)

```
use gpstracker;  
GRANT ALL PRIVILEGES ON gpstracker.* TO  
'gpstracker_user'@'localhost' IDENTIFIED BY 'gpstracker';  
exit;
```

Para que se possam ver todas as informações armazenadas no banco de dados, deve-se entrar no banco novamente e digitar o comando:

```
select * from gpslocations order by gpsLocationID desc;
```

E então serão listados todos os usuários cadastrados no banco de dados. Saia do *mysql* e se tudo estiver correto, deve-se acessar a pagina web do servidor com o endereço:

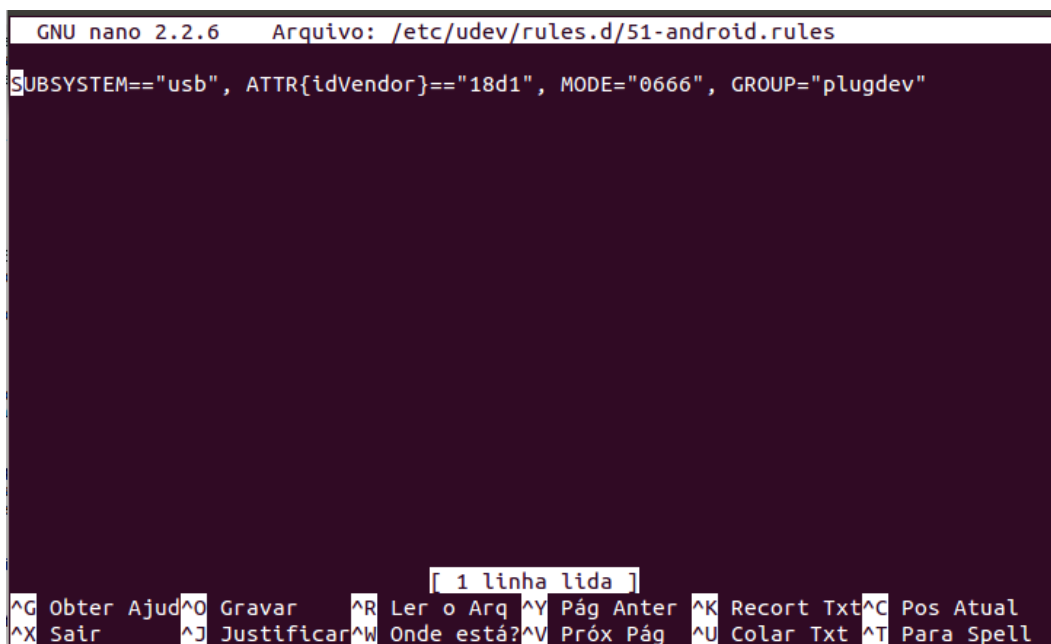
```
http://localhost/rastreamento/displaymap.php
```


- ANEXO VI: Configuração e instalação do aplicativo no dispositivo móvel

1. Primeiramente, deve-se ativar a depuração USB do dispositivo que receberá a aplicação. Para isso, devem-se ativar as opções de desenvolvedor em que o procedimento para ativá-lo, depende da versão do *android*.
2. Configurar o sistema para detectar o dispositivo quando conectado via cabo USB. Para isso, deve-se ir ao diretório */etc/udev/rules.d* e criar um arquivo chamado *51-android.rules* com os comandos:

```
cd /etc/udev/rules.d/  
sudo nano 51-android.rules
```

3. Então se deve editar o arquivo de acordo com a imagem abaixo, lembrando que a configuração a seguir é referente aos aparelhos da linha *Nexus* que possuem a *Google* como fabricante, para demais aparelhos deve-se mudar o parâmetro *AATTR{idVendor}* para o código relacionado ao fabricante do dispositivo celular.



```
GNU nano 2.2.6 Arquivo: /etc/udev/rules.d/51-android.rules  
SUBSYSTEM=="usb", ATTR{idVendor}=="18d1", MODE="0666", GROUP="plugdev"  
[ 1 linha lida ]  
^G Obter Ajud ^O Gravar ^R Ler o Arq ^Y Pág Anter ^K Recort Txt ^C Pos Atual  
^X Sair ^J Justificar ^W Onde está? ^V Próx Pág ^U Colar Txt ^T Para Spell
```

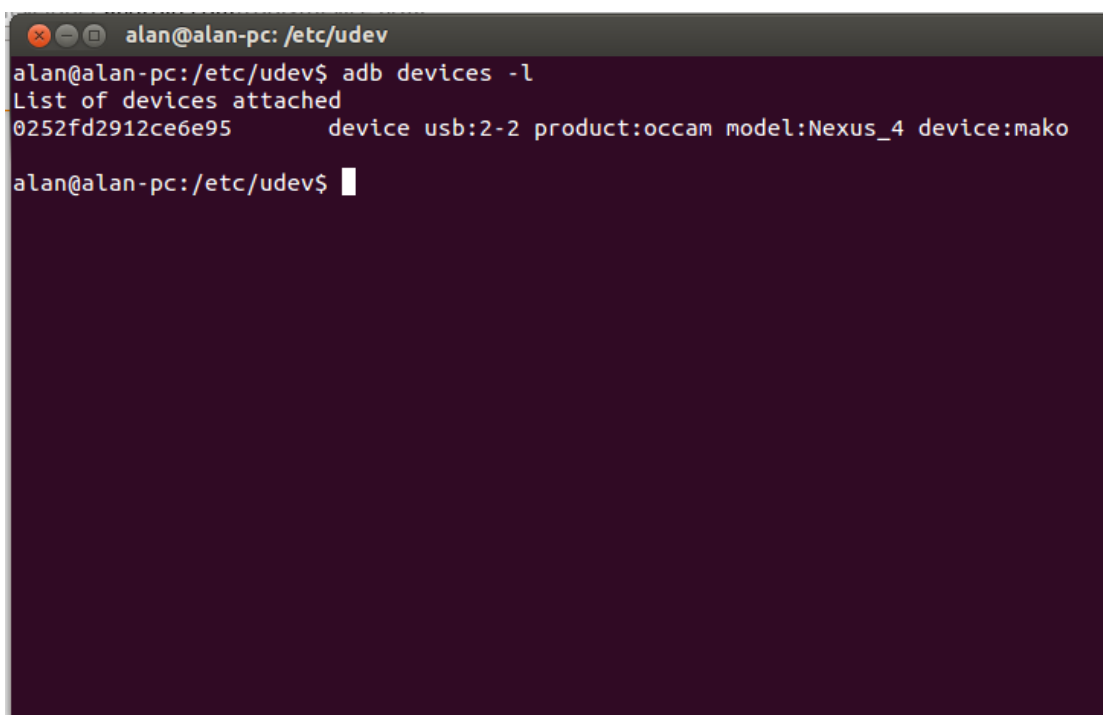
Figura 59 – Arquivo de configuração para detectar dispositivos.

```
chmod a+r /etc/udev/rules.d/51-android.rules
```

Para verificar se o dispositivo está sendo reconhecido pela máquina, deve-se executar o comando para mostrar os dispositivos conectados. O comando é:

```
adb devices -l
```

O resultado pode ser observado na figura 59.



```
alan@alan-pc: /etc/udev
alan@alan-pc:/etc/udev$ adb devices -l
List of devices attached
0252fd2912ce6e95      device usb:2-2 product:occam model:Nexus_4 device:mako
alan@alan-pc:/etc/udev$
```

Figura 60 – Listando os dispositivos conectados.

Neste momento, tem-se o software para desenvolver aplicações *android* e o dispositivo conectado para receber estas aplicações. Com o *Android Studio* aberto, selecionar *File > Import Project* e selecionar a pasta *android* localizada no diretório *phoneClients*. Feito isso, no menu ir em *Tool > Android > SDK Manager*. Selecionar *Tools, Extras* e o pacote referente à versão *Android* que o celular possui, neste caso o *android 4.4.2*. Ao selecionar para instalar os pacotes, aparecerá em uma tela para que sejam marcadas as licenças clicando em *Accept License* em todos os pacotes e então os mesmos serão baixados e instalados para completar a compatibilidade da aplicação com o celular ao clicar em *Install*.

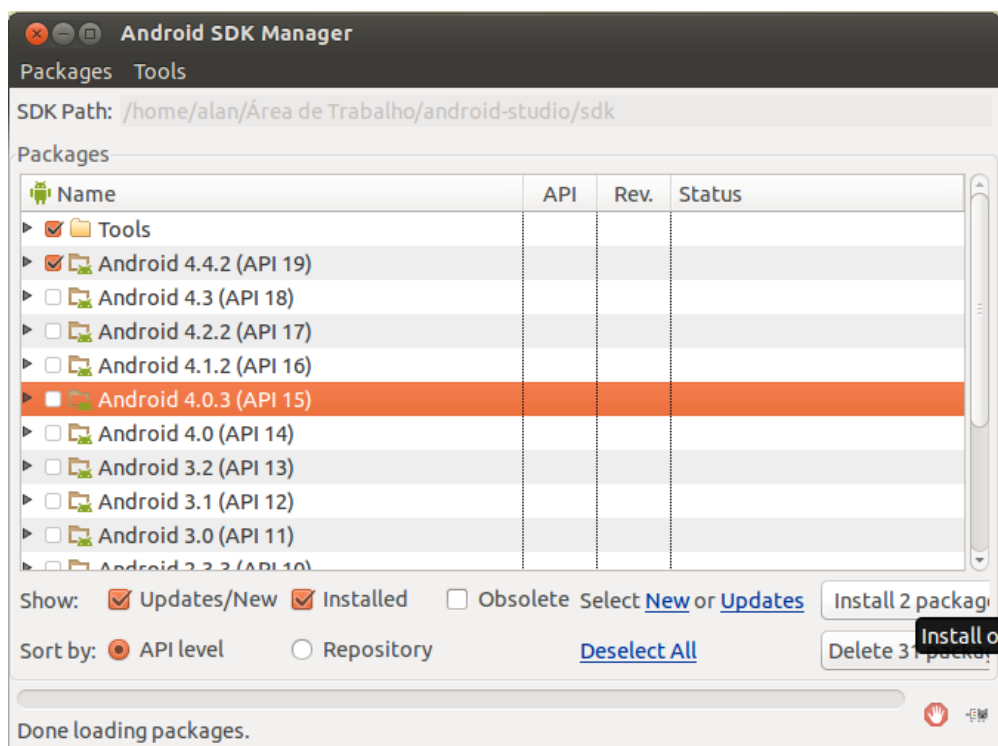


Figura 61 – Android SDK Manager.

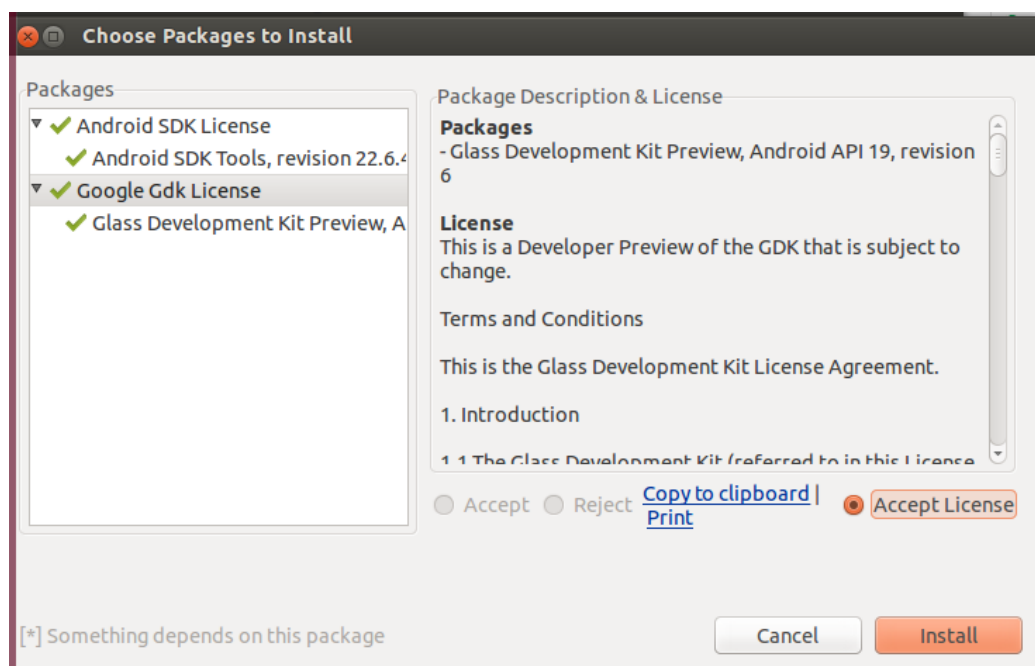


Figura 62 – Licença dos pacotes selecionados.

Com o celular em conexão USB com o computador, ele deverá agora ser reconhecido pelo computador no canto inferior esquerdo. Deve-se selecionar *Run* para iniciar a aplicação no celular e então se iniciará a instalação da aplicação no celular. Após a instalação deve-se colocar o nome de usuário no primeiro campo e o endereço do servidor local no segundo campo e selecionar *Trancking is On*.

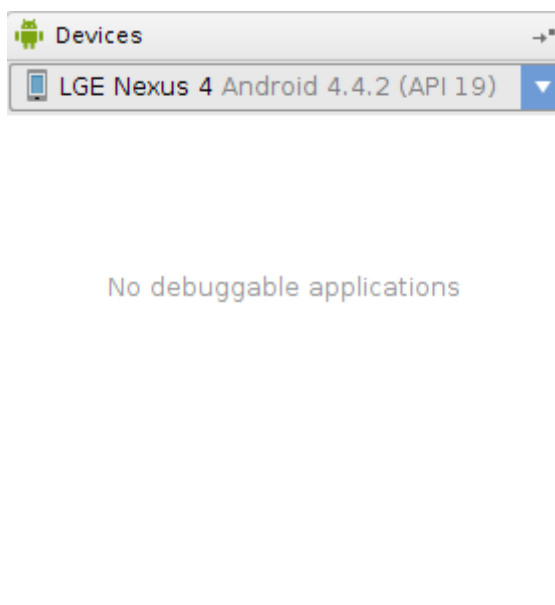


Figura 63 – Celular reconhecido pelo Android Studio.

- ANEXO VII: Medida realizada para rede 3G - SIMET



Resumo dos Resultados
 Velocidade: 1.23 Mbit/s
 Latência: 84.95 ms
 Perda de Tráfego: 0.00%
 Banda Contratada: 1.00 Mbit/s

CEP: 72444-240

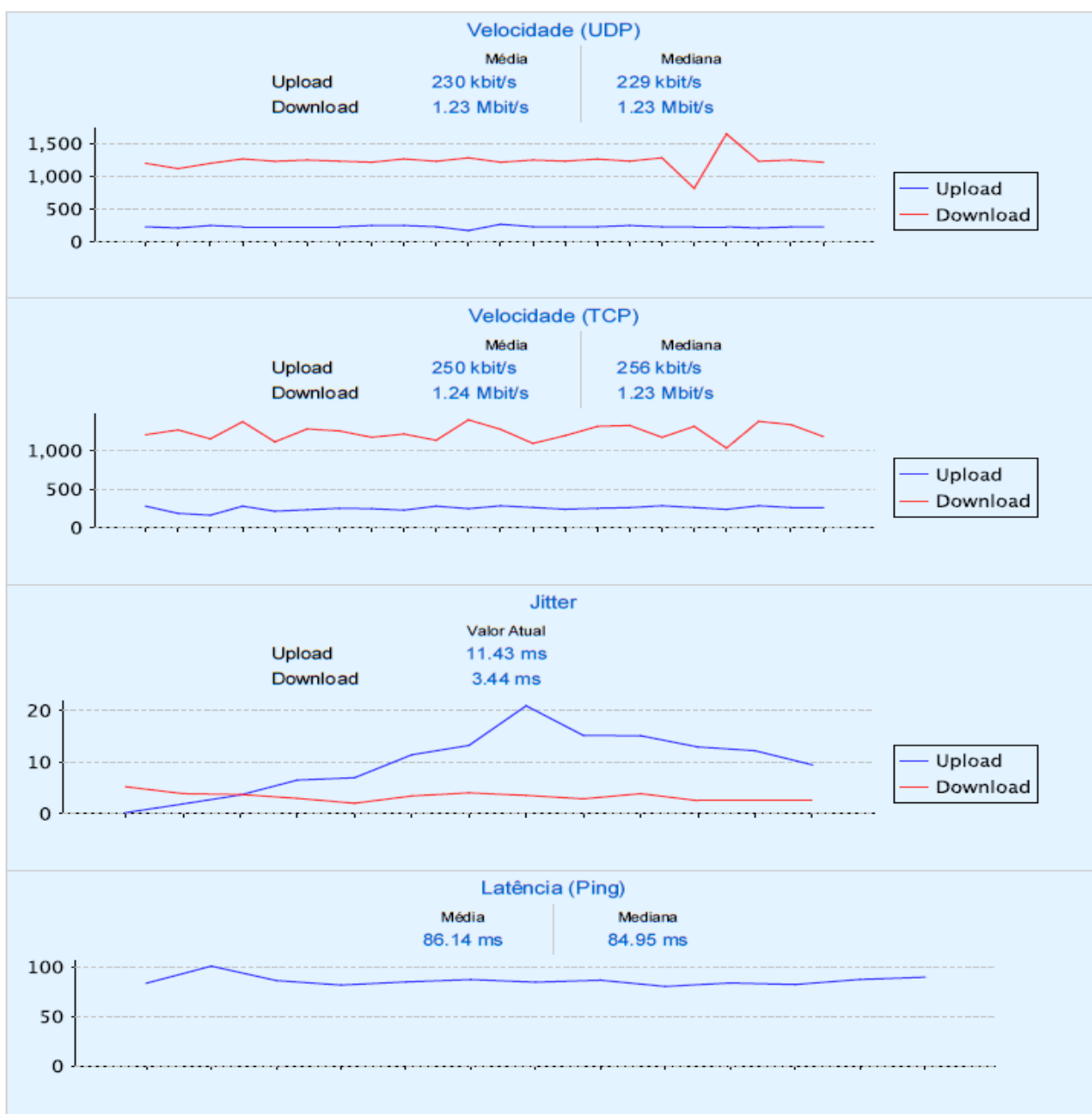
Banda contratada: 1.00 Mbit/s

Local de acesso: Em outro lugar

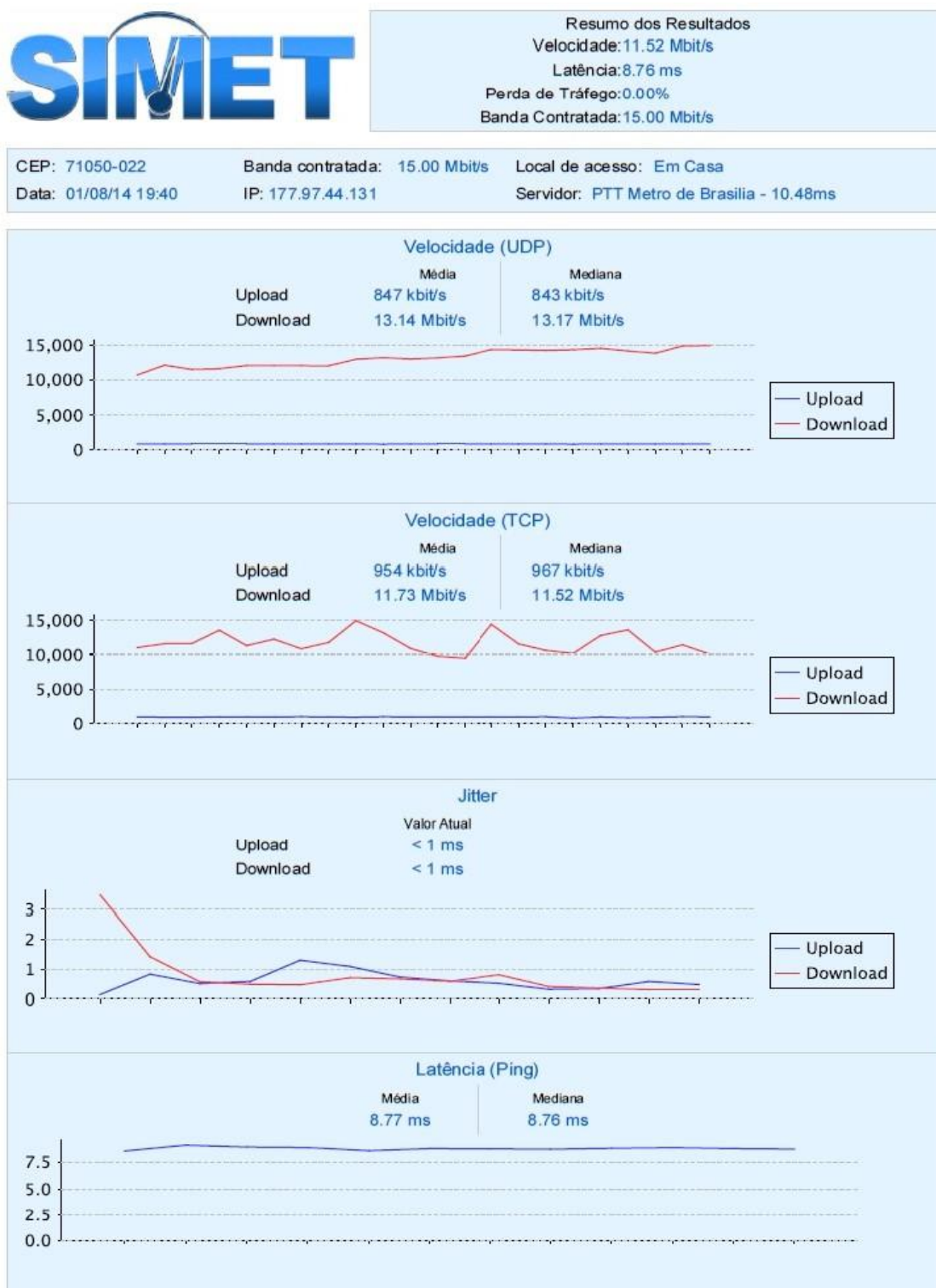
Data: 31/07/14 17:20

IP: 187.116.245.33

Servidor: PTT Metro de Sao Paulo - ALOG - 88.09



- ANEXO VIII: Medida realizada para rede WLAN - SIMET



- ANEXO IX: Dados medidos e calculados para a rede 3G – operadora VIVO

| MÉDIA | | | | | | | | | | MEDIANA | | | | | | | | | |
|-----------------|---------------|-------------|----------|--------------|----------|--------------|----------|---------------|-------------|--------------|----------|--------------|--------|---------|--|--|--|--|--|
| Medida 1 | Latência (ms) | Jitter (ms) | | TCP (kbit/s) | | UDP (kbit/s) | | Latência (ms) | Jitter (ms) | TCP (kbit/s) | | UDP (kbit/s) | | | | | | | |
| | | Upload | Download | Upload | Download | Upload | Download | | | Upload | Download | | | | | | | | |
| Amostra 1 | 90,56 | 10,51 | 2,49 | 239 | 1240 | 223 | 1230 | 80,42 | 10,51 | 2,49 | 240 | 1240 | 220 | 1240 | | | | | |
| Amostra 2 | 83,91 | 11,46 | 2,68 | 214 | 1240 | 195 | 1230 | 79,48 | 11,46 | 2,68 | 255 | 1180 | 191 | 1230 | | | | | |
| Amostra 3 | 97,1 | 10,71 | 2,19 | 196 | 1240 | 179 | 1210 | 85,52 | 10,71 | 2,19 | 233 | 1290 | 172 | 1250 | | | | | |
| Média | 90,52 | 10,89 | 2,45 | 216,33 | 1240,00 | 199,00 | 1223,33 | 81,81 | 10,89 | 2,45 | 242,67 | 1236,67 | 194,33 | 1240,00 | | | | | |
| Desvio padrão | 6,60 | 0,50 | 0,25 | 21,59 | 0,00 | 22,27 | 11,55 | 3,25 | 0,50 | 0,25 | 11,24 | 55,08 | 24,17 | 10,00 | | | | | |
| Medida 2 | | | | | | | | | | | | | | | | | | | |
| Medida 2 | Latência (ms) | Jitter (ms) | | TCP (kbit/s) | | UDP (kbit/s) | | Latência (ms) | Jitter (ms) | TCP (kbit/s) | | UDP (kbit/s) | | | | | | | |
| | | Upload | Download | Upload | Download | Upload | Download | | | Upload | Download | | | | | | | | |
| Amostra 1 | 94,49 | 17,61 | 3,04 | 246 | 1220 | 224 | 1230 | 85,62 | 17,61 | 3,04 | 261 | 1210 | 229 | 1240 | | | | | |
| Amostra 2 | 91,43 | 13,19 | 2,64 | 185 | 1230 | 171 | 1230 | 87,26 | 13,19 | 2,64 | 238 | 1240 | 172 | 1240 | | | | | |
| Amostra 3 | 97,47 | 9,59 | 4,26 | 226 | 1230 | 220 | 1240 | 87,61 | 9,59 | 4,26 | 254 | 1230 | 218 | 1230 | | | | | |
| Média | 94,46 | 13,46 | 3,31 | 219,00 | 1226,67 | 205,00 | 1233,33 | 86,83 | 13,46 | 3,31 | 251,00 | 1226,67 | 206,33 | 1236,67 | | | | | |
| Desvio padrão | 3,02 | 4,02 | 0,84 | 31,10 | 5,77 | 29,51 | 5,77 | 1,06 | 4,02 | 0,84 | 11,79 | 15,28 | 30,24 | 5,77 | | | | | |
| Medida 3 | | | | | | | | | | | | | | | | | | | |
| Medida 3 | Latência (ms) | Jitter (ms) | | TCP (kbit/s) | | UDP (kbit/s) | | Latência (ms) | Jitter (ms) | TCP (kbit/s) | | UDP (kbit/s) | | | | | | | |
| | | Upload | Download | Upload | Download | Upload | Download | | | Upload | Download | | | | | | | | |
| Amostra 1 | 81,84 | 17,63 | 2,44 | 181 | 1230 | 172 | 1230 | 77,95 | 17,63 | 2,44 | 214 | 1220 | 172 | 1240 | | | | | |
| Amostra 2 | 86,36 | 8,26 | 1,89 | 264 | 1230 | 216 | 1240 | 85,44 | 8,26 | 1,89 | 264 | 1220 | 220 | 1230 | | | | | |
| Amostra 3 | 86,14 | 9,48 | 2,54 | 250 | 1240 | 230 | 1230 | 84,95 | 9,48 | 2,54 | 256 | 1230 | 229 | 1230 | | | | | |
| Média | 84,78 | 11,79 | 2,29 | 231,67 | 1233,33 | 206,00 | 1233,33 | 82,78 | 11,79 | 2,29 | 244,67 | 1223,33 | 207,00 | 1233,33 | | | | | |
| Desvio padrão | 2,55 | 5,09 | 0,35 | 44,43 | 5,77 | 30,27 | 5,77 | 4,19 | 5,09 | 0,35 | 26,86 | 5,77 | 30,64 | 5,77 | | | | | |
| Medida 4 | | | | | | | | | | | | | | | | | | | |
| Medida 4 | Latência (ms) | Jitter (ms) | | TCP (kbit/s) | | UDP (kbit/s) | | Latência (ms) | Jitter (ms) | TCP (kbit/s) | | UDP (kbit/s) | | | | | | | |
| | | Upload | Download | Upload | Download | Upload | Download | | | Upload | Download | | | | | | | | |
| Amostra 1 | 81,47 | 9,2 | 5,11 | 226 | 1230 | 213 | 1230 | 81,48 | 9,2 | 5,11 | 244 | 1220 | 210 | 1230 | | | | | |
| Amostra 2 | 81,61 | 19,57 | 3,92 | 218 | 1230 | 185 | 1240 | 80,07 | 19,57 | 3,92 | 232 | 1240 | 210 | 1240 | | | | | |
| Amostra 3 | 89,29 | 11,52 | 3,19 | 249 | 1060 | 224 | 1050 | 85,03 | 11,52 | 3,19 | 261 | 1140 | 229 | 1090 | | | | | |
| Média | 84,12 | 13,43 | 4,07 | 231,00 | 1173,33 | 207,33 | 1173,33 | 82,19 | 13,43 | 4,07 | 245,67 | 1200,00 | 216,33 | 1186,67 | | | | | |
| Desvio padrão | 4,48 | 5,44 | 0,97 | 16,09 | 98,15 | 20,11 | 106,93 | 2,56 | 5,44 | 0,97 | 14,57 | 52,92 | 10,97 | 83,86 | | | | | |
| Medida 5 | | | | | | | | | | | | | | | | | | | |
| Medida 5 | Latência (ms) | Jitter (ms) | | TCP (kbit/s) | | UDP (kbit/s) | | Latência (ms) | Jitter (ms) | TCP (kbit/s) | | UDP (kbit/s) | | | | | | | |
| | | Upload | Download | Upload | Download | Upload | Download | | | Upload | Download | | | | | | | | |
| Amostra 1 | 105,68 | 10,87 | 3,13 | 134 | 1220 | 123 | 1150 | 100,72 | 10,87 | 3,13 | 142 | 1250 | 119 | 1230 | | | | | |
| Amostra 2 | 84,91 | 17,48 | 2,24 | 223 | 1240 | 215 | 1230 | 82,89 | 17,48 | 2,24 | 256 | 1230 | 210 | 1240 | | | | | |
| Amostra 3 | 82,91 | 12,34 | 3,65 | 254 | 1240 | 230 | 1030 | 82,57 | 12,34 | 3,65 | 261 | 1260 | 229 | 1100 | | | | | |
| Média | 91,17 | 13,56 | 3,01 | 203,67 | 1233,33 | 189,33 | 1136,67 | 88,73 | 13,56 | 3,01 | 219,67 | 1246,67 | 186,00 | 1190,00 | | | | | |
| Desvio padrão | 12,61 | 3,47 | 0,71 | 62,29 | 11,55 | 57,93 | 100,66 | 10,39 | 3,47 | 0,71 | 67,31 | 15,28 | 58,80 | 78,10 | | | | | |

- ANEXO X: Dados medidos e calculados para rede WLAN – operadora GVT

| MÉDIA | | | | | | | | | | | | MEDIANA | | | | | | | | | | | |
|---------------|---------------|-------------|----------|--------------|----------|--------------|----------|---------------|-------------|----------|--------------|----------|--------------|----------|--|--|--|--|--|--|--|--|--|
| Amostra 1 | Latência (ms) | Jitter (ms) | | TCP (kbit/s) | | UDP (kbit/s) | | Latência (ms) | Jitter (ms) | | TCP (kbit/s) | | UDP (kbit/s) | | | | | | | | | | |
| | | Upload | Download | Upload | Download | Upload | Download | | Upload | Download | Upload | Download | Upload | Download | | | | | | | | | |
| Amostra 1 | 9,84 | 1 | 2,68 | 984 | 10650 | 778 | 11990 | 9,84 | 1 | 2,68 | 985 | 11680 | 804 | 12070 | | | | | | | | | |
| Amostra 2 | 13,03 | 1 | 1 | 967 | 11140 | 848 | 11640 | 12,27 | 1 | 1 | 979 | 13640 | 843 | 12020 | | | | | | | | | |
| Amostra 3 | 13,01 | 1,52 | 1,66 | 970 | 9210 | 817 | 10240 | 10,12 | 1,52 | 1,66 | 962 | 10700 | 832 | 9850 | | | | | | | | | |
| Média | 11,96 | 1,17 | 1,78 | 973,67 | 10333,33 | 814,33 | 11290,00 | 10,74 | 1,17 | 1,78 | 975,33 | 12006,67 | 826,33 | 11313,33 | | | | | | | | | |
| Desvio padrão | 1,50 | 0,25 | 0,69 | 7,41 | 819,12 | 28,64 | 756,09 | 1,09 | 0,25 | 0,69 | 9,74 | 1222,27 | 16,42 | 1034,93 | | | | | | | | | |
| MÉDIA 2 | | | | | | | | | | | | MEDIANA | | | | | | | | | | | |
| Amostra 1 | Latência (ms) | Jitter (ms) | | TCP (kbit/s) | | UDP (kbit/s) | | Latência (ms) | Jitter (ms) | | TCP (kbit/s) | | UDP (kbit/s) | | | | | | | | | | |
| | | Upload | Download | Upload | Download | Upload | Download | | Upload | Download | Upload | Download | Upload | Download | | | | | | | | | |
| Amostra 1 | 9,69 | 1 | 4,69 | 983 | 10750 | 796 | 9220 | 9,3 | 1 | 4,69 | 985 | 10650 | 837 | 10630 | | | | | | | | | |
| Amostra 2 | 9,57 | 1 | 1 | 961 | 11430 | 827 | 11730 | 9,48 | 1 | 1 | 980 | 11730 | 823 | 12670 | | | | | | | | | |
| Amostra 3 | 9,89 | 1 | 1 | 985 | 7760 | 843 | 8800 | 9,73 | 1 | 1 | 985 | 8239 | 843 | 8710 | | | | | | | | | |
| Média | 9,72 | 1,00 | 2,23 | 976,33 | 9980,00 | 822,00 | 9916,67 | 9,50 | 1,00 | 2,23 | 983,33 | 10206,33 | 834,33 | 10670,00 | | | | | | | | | |
| Desvio padrão | 0,13 | 0,00 | 1,74 | 10,87 | 1594,14 | 19,51 | 1293,63 | 0,18 | 0,00 | 1,74 | 2,36 | 1459,31 | 8,38 | 1616,91 | | | | | | | | | |
| MÉDIA 3 | | | | | | | | | | | | MEDIANA | | | | | | | | | | | |
| Amostra 1 | Latência (ms) | Jitter (ms) | | TCP (kbit/s) | | UDP (kbit/s) | | Latência (ms) | Jitter (ms) | | TCP (kbit/s) | | UDP (kbit/s) | | | | | | | | | | |
| | | Upload | Download | Upload | Download | Upload | Download | | Upload | Download | Upload | Download | Upload | Download | | | | | | | | | |
| Amostra 1 | 9,31 | 1 | 1,06 | 974 | 10830 | 844 | 8220 | 9,29 | 1 | 1,06 | 984 | 11200 | 843 | 7420 | | | | | | | | | |
| Amostra 2 | 9,14 | 1 | 1 | 970 | 9860 | 849 | 10260 | 9,09 | 1 | 1 | 985 | 9250 | 843 | 10110 | | | | | | | | | |
| Amostra 3 | 9,72 | 1 | 1 | 985 | 8640 | 848 | 9670 | 9,77 | 1 | 1 | 985 | 8740 | 843 | 9680 | | | | | | | | | |
| Média | 9,39 | 1,00 | 1,02 | 976,33 | 9776,67 | 847,00 | 9383,33 | 9,38 | 1,00 | 1,02 | 984,67 | 9730,00 | 843,00 | 9070,00 | | | | | | | | | |
| Desvio padrão | 0,24 | 0,00 | 0,03 | 6,34 | 896,00 | 2,16 | 857,14 | 0,29 | 0,00 | 0,03 | 0,47 | 1060,09 | 0,00 | 1179,86 | | | | | | | | | |
| MÉDIA 4 | | | | | | | | | | | | MEDIANA | | | | | | | | | | | |
| Amostra 1 | Latência (ms) | Jitter (ms) | | TCP (kbit/s) | | UDP (kbit/s) | | Latência (ms) | Jitter (ms) | | TCP (kbit/s) | | UDP (kbit/s) | | | | | | | | | | |
| | | Upload | Download | Upload | Download | Upload | Download | | Upload | Download | Upload | Download | Upload | Download | | | | | | | | | |
| Amostra 1 | 9,02 | 1 | 1 | 984 | 14270 | 849 | 13960 | 8,87 | 1 | 1 | 985 | 14680 | 843 | 14200 | | | | | | | | | |
| 10,63 | 9,26 | 1 | 1 | 985 | 11160 | 799 | 11060 | 9,26 | 1 | 1 | 985 | 11340 | 843 | 10890 | | | | | | | | | |
| Amostra 3 | 9,98 | 1 | 1,01 | 954 | 10480 | 776 | 10280 | 9,79 | 1 | 1,01 | 961 | 10780 | 804 | 10550 | | | | | | | | | |
| Média | 9,42 | 1,00 | 1,00 | 974,33 | 11970,00 | 808,00 | 11766,67 | 9,31 | 1,00 | 1,00 | 977,00 | 12266,67 | 830,00 | 11880,00 | | | | | | | | | |
| Desvio padrão | 0,41 | 0,00 | 0,00 | 14,38 | 1649,87 | 30,47 | 1583,27 | 0,38 | 0,00 | 0,00 | 11,31 | 1721,73 | 18,38 | 1646,35 | | | | | | | | | |
| MÉDIA 5 | | | | | | | | | | | | MEDIANA | | | | | | | | | | | |
| Amostra 1 | Latência (ms) | Jitter (ms) | | TCP (kbit/s) | | UDP (kbit/s) | | Latência (ms) | Jitter (ms) | | TCP (kbit/s) | | UDP (kbit/s) | | | | | | | | | | |
| | | Upload | Download | Upload | Download | Upload | Download | | Upload | Download | Upload | Download | Upload | Download | | | | | | | | | |
| Amostra 1 | 10,63 | 1,42 | 5,76 | 980 | 11780 | 772 | 12710 | 10,07 | 1,42 | 5,76 | 984 | 11970 | 841 | 13110 | | | | | | | | | |
| Amostra 2 | 8,77 | 1 | 1 | 954 | 11730 | 847 | 13140 | 8,76 | 1 | 1 | 967 | 11520 | 843 | 13170 | | | | | | | | | |
| Amostra 3 | 9,56 | 1 | 1 | 982 | 11490 | 847 | 11990 | 9,52 | 1 | 1 | 983 | 11510 | 843 | 11980 | | | | | | | | | |
| Média | 9,65 | 1,14 | 2,59 | 972,00 | 11666,67 | 822,00 | 12613,33 | 9,45 | 1,14 | 2,59 | 978,00 | 11666,67 | 842,33 | 12753,33 | | | | | | | | | |
| Desvio padrão | 0,76 | 0,20 | 2,24 | 12,75 | 126,58 | 35,36 | 474,44 | 0,54 | 0,20 | 2,24 | 7,79 | 214,53 | 0,94 | 547,38 | | | | | | | | | |