

PROJETO DE GRADUAÇÃO

PROPOSTA DE AUXÍLIO DE RASTREAMENTO AUDITIVO PARA PNE VIA SISTEMA DE INFORMAÇÃO GEOGRÁFICA (SIG)

Luis Eduardo Alkmin La Torre

Brasília, Dezembro de 2018

UNIVERSIDADE DE BRASÍLIA

**FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia
Departamento de Engenharia Elétrica

PROJETO DE GRADUAÇÃO

**PROPOSTA DE AUXÍLIO DE RASTREAMENTO
AUDITIVO PARA PNE VIA SISTEMA DE
INFORMAÇÃO GEOGRÁFICA (SIG)**

Luis Eduardo Alkmin La Torre

Relatório submetido como requisito parcial para obtenção
do grau de Engenheiro de Redes de Comunicação.

Banca Examinadora

Prof. Georges Daniel Amvame Nze, Dr., ENE/UnB
Orientador

Prof. Rafael Timóteo de Sousa Jr., Dr., ENE/UnB
Examinador Interno

Fábio Lúcio Lopes de Mendonça, MSc., ENE/UnB
Examinador Interno

Brasília, Dezembro de 2018

Dedicatória

Dedico este trabalho aos deficientes visuais, por demonstrarem, mesmo com todas as adversidades do mundo, que o real significado de percepção vai além de um sentido do corpo humano.

Luis Eduardo Alkmin La Torre

Agradecimentos

Agradeço à minha mãe Teresinha, ao meu pai César e ao meu irmão Gustavo, por me darem alicerce e todo o apoio para a minha formação tanto pessoal como intelectual. Agradeço aos meus amigos pelo companheirismo e parceria desde a época do ensino médio, vocês são demais! Ao professor Georges, não somente pelo apoio e pela oportunidade de realizar este projeto, mas por toda a sua dedicação e atenção na formação dos alunos do curso de Engenharia de Redes de Comunicação. Agradeço ao Laboratório IoT pela disponibilização do módulo utilizado neste projeto. Por fim, agradeço à todas as pessoas que eu conheci e convivi no meio desse período tão importante da minha vida, e que de alguma forma, contribuíram para o meu crescimento pessoal e intelectual.

Luis Eduardo Alkmin La Torre

RESUMO

Este trabalho tem como propósito a concepção de um dispositivo auditivo de auxílio de locomoção e localização para deficientes visuais via Sistema de Informação Geográfica (SIG), mais precisamente com a utilização de um Sistema Global de Navegação por Satélite – ou GNSS, sigla em inglês para *Global Navigation Satellite System*. Este dispositivo consiste de uma plataforma Raspberry Pi juntamente com um módulo GPS (*Global Positioning System*) que irá fornecer informações faladas sobre localização, terreno e eventuais alertas caso exista algum obstáculo próximo.

Palavras-chave: auxílio, áudio, deficiente visual, localização, Sistema de Informação Geográfica.

ABSTRACT

The purpose of this work is the development of a locomotive and localization aid device for the visually impaired by way of Geographic Information System (GIS). This device consists of a Raspberry Pi platform along with a Global Positioning System (GPS) module that will provide spoken information about location, terrain and possible alerts if there are any near obstacles.

SUMÁRIO

1 INTRODUÇÃO	1
1.1 IDENTIFICAÇÃO E CONTEXTUALIZAÇÃO DO PROBLEMA	1
1.2 OBJETIVOS	2
1.2.1 Objetivo Geral	2
1.2.2 Objetivo Específico	2
1.3 ESTRUTURA DO TRABALHO	2
2 FUNDAMENTAÇÃO TEÓRICA	3
2.1 Raspberry Pi	3
2.2 Módulo GPS (u-blox NEO-6M)	4
2.3 Sistema de Informação Geográfica (SIG)	4
2.4 <i>Global Positioning System</i> (GPS)	5
2.5 Tecnologia 3G	6
2.6 Servidor LAMP	6
2.6.1 Linux	6
2.6.2 Apache	7
2.6.3 MySQL	7
2.6.4 PHP	8
2.6.5 phpMyAdmin	8
2.7 <i>Google Text To Speech</i> (GTTS)	8
2.8 Python	8
3 METODOLOGIA	9
3.1 DELIMITAÇÃO DO TEMA	9
3.1.1 Escolha do <i>hardware</i>	10
3.1.2 Identificação de conexões e Montagem do <i>hardware</i>	10
3.1.3 Comunicação entre módulo e Raspberry	12
3.1.4 Georreferenciamento da UnB no BD	13
3.1.5 Desenvolvimento do programa	17
4 ESTUDO DE CASO	21
4.1 TESTES INICIAIS E PROVA DE CAMPO	21
4.2 DIAGRAMA DE CASO DE USO	22
5 RESULTADOS	24
5.1 DISPOSITIVO DE AUXÍLIO AUDITIVO	24
5.2 CONVERSÃO DE TEXTO EM AVISO AUDITIVO	25
6 CONCLUSÃO	27
REFERÊNCIAS BIBLIOGRÁFICAS	28
APÊNDICE A	30
A1. COMANDOS PARA COMUNICAÇÃO RASPBERRY/MÓDULO	30
A2. COMANDOS PARA INSTALAÇÃO DA PILHA LAMP E PHPMYADMIN	31
A3. INSTALAÇÃO DAS BIBLIOTECAS E CÓDIGO DO PROGAMA <i>AUXILIO.PY</i>	32

LISTA DE FIGURAS

2.1	Raspberry Pi 3 Model B.....	3
2.2	Módulo GPS u-blox NEO-6M	4
2.3	Constelação de satélites do sistema GPS	5
2.4	Histórico de utilização de servidores web pelo mundo.....	7
3.1	Fluxograma de etapas do projeto.....	9
3.2	Diagrama de pinos do GPIO do Raspberry 3	11
3.3	Ligação dos jumpers ao Raspberry	11
3.4	Jumpers soldados ao módulo GPS.....	12
3.5	Demonstração da comunicação entre módulo e Raspberry	12
3.6	Desnível em calçada próximo à entrada do PJC	13
3.7	Mapa dos locais georreferenciados no projeto	15
3.8	Página index.html do servidor Apache no localhost	16
3.9	Página info.php no localhost.....	16
3.10	Tela de login do phpMyAdmin.....	17
3.11	Parte do código relativo aos imports	18
3.12	Parte do código onde as inicializações são feitas	18
3.13	Parte do código contendo os métodos do programa auxilio.py	19
5.1	Diagrama de caso de uso do projeto	23
5.2	Dispositivo de auxílio auditivo	24
5.3	Exemplo de uso do dispositivo.....	25
5.4	Arquitetura do dispositivo	26

LISTA DE TABELAS

3.1	Listagem dos produtos adquiridos para a realização do dispositivo	10
3.2	Base de informações georreferenciadas para o banco de dados	14

LISTA DE SÍMBOLOS

Siglas

ABNT	Associação Brasileira de Normas Técnicas
API	<i>Application Programming Interface</i>
BD	Banco de Dados
BSAS	Bloco de Salas de Aula Sul
CRUD	<i>Create Read Update Delete</i>
FE	Faculdade de Educação
FT	Faculdade de Tecnologia
GEOS	<i>Geometry Engine Open Source</i>
GIS	<i>Geographic Information System</i>
GNSS	<i>Global Navigation Satellite System</i>
GPIO	<i>General-Purpose Input/Output</i>
GPS	<i>Global Positioning System</i>
GTTS	<i>Google Text-To-Speech</i>
HTML	<i>Hypertext Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
ICC	Instituto Central de Ciências
IdA	Instituto de Artes
IMT	<i>International Mobile Telecommunications</i>
IREL	Instituto de Relações Internacionais
IoT	<i>Internet of Things</i>
LAMP	<i>Linux Apache MySQL PHP/Python</i>
LAN	<i>Local Area Network</i>
LED	<i>Light Emitting Diode</i>
NMEA	<i>National Marine Electronics Association</i>
PJC	Pavilhão João Calmon
PNE	Portador de Necessidades Especiais
PPNE-UnB	Programa de Apoio às Pessoas com Necessidade Especiais
RAM	<i>Random Access Memory</i>
RPi	<i>Raspberry Pi</i>
RU	Restaurante Universitário
SGBD	Sistema Gerenciador de Banco de Dados
SIG	Sistema de Informação Geográfica
SO	Sistema Operacional
SQL	<i>Structured Query Language</i>
UnB	Universidade de Brasília

1 INTRODUÇÃO

1.1 IDENTIFICAÇÃO E CONTEXTUALIZAÇÃO DO PROBLEMA

Em um âmbito global, cerca de 253 milhões de pessoas sofrem de algum tipo de deficiência visual, dentre elas, 36 milhões são cegas, de acordo com a Organização Mundial da Saúde [1]. Muitos deficientes visuais lidam com uma qualidade de vida inadequada, além de enfrentar barreiras na educação e emprego [1].

A questão da acessibilidade no Brasil é contemplada na Lei Federal nº 10.098, criada em 2000, e cujo objetivo é a promoção do acesso de pessoas deficientes ou com mobilidade reduzida a locais públicos. A lei inclui no universo dos deficientes não somente os físicos, mas também os deficientes auditivos e visuais, pois eles necessitam da eliminação de barreiras que lhes assegure acesso aos bens culturais e sociais, como também segurança na locomoção [2].

No que se refere especificadamente aos deficientes visuais, a lei menciona a necessidade de implantação de dispositivos sonoros nos semáforos para orientação na travessia de ruas e avenidas com trânsito intenso (art. 9º), além de especificar que os projetos e traçados dos elementos de urbanização públicos e privados de uso da comunidade - itinerário e passagens de pedestres – sigam os parâmetros estabelecidos pelas normas técnicas de acessibilidade da Associação Brasileira de Normas Técnicas – ABNT (art. 5º)[2].

O GNSS refere-se a uma constelação de satélites que fornecem sinais do espaço que transmitem dados de posicionamento e tempo para receptores GNSS. Os receptores então usam esses dados para determinar a localização [3].

O Sistema de Informação Geográfica (SIG), em um enfoque orientado a processos, pode ser definido como coleções de subsistemas integrados, em que dados especiais passam por uma sequência de processos de conversão, coleta, armazenamento e manipulação [4]. A integração de tecnologias SIG e GNSS, se implementada e empregada de modo adequado, com toda certeza possibilitará aos usuários maior eficiência na capacidade de elaboração de análise, no gerenciamento e na otimização dos trabalhos em todas as fases que integram atividades que têm, como componente, o espaço geográfico [4].

Devido ao fato da Lei Federal nº 10.098 por muitas das vezes não ser respeitada no que diz respeito à segurança na locomoção de deficientes visuais, a proposta de um dispositivo de auxílio auditivo utilizando informações georreferenciadas torna-se válida para a melhoria da segurança no deslocamento, e tendo por consequência, um acréscimo na qualidade de vida do deficiente visual.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Este trabalho tem como principal objetivo a criação de um dispositivo capaz de auxiliar o deficiente visual em seu deslocamento. Desta forma, deve-se atentar ao fato do dispositivo não substituir outros métodos utilizados como referência na locomoção, como: bengala, cão-guia, etc., sendo assim, um complemento para esses meios.

1.2.2 Objetivo Específico

- Georreferenciar a UnB em áreas geográficas e eventuais áreas de risco em um BD;
- Associar o BD de áreas com a localização atual do deficiente visual e emitir uma mensagem específica com informações relevantes acerca do local.

1.3 ESTRUTURA DO TRABALHO

Quanto à organização, este trabalho está dividido em 4 (quatro) partes, sendo eles os seguintes capítulos:

- Capítulo 2: Fundamentação Teórica.

Este capítulo apresenta e define conceitos teóricos primordiais que são utilizados ao longo deste trabalho.

- Capítulo 3: Metodologia.

Aqui são apontadas e explicadas as etapas que envolvem a elaboração deste projeto.

- Capítulo 4: Resultados.

Neste capítulo são denotados os resultados alcançados na composição do projeto.

- Capítulo 5: Conclusão.

Por fim, nesta parte a solução é ratificada para o problema apresentado.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Raspberry Pi

O Raspberry Pi é, basicamente, um computador de baixo custo do tamanho de um cartão de crédito. É uma plataforma pequena capaz de permitir que pessoas de todas as idades explorem a computação, e aprendam a programar em linguagens como Scratch e Python. É capaz de fazer tudo o que você espera de um computador desktop, desde navegar na internet e reproduzir vídeo de alta definição, até fazer planilhas, processamento de texto e jogos [5].

A plataforma possui um processador *quad-core* ARM Cortex-A53 de 1,2 GHz, memória RAM de 1 GB, 100 Base Ethernet, *Wireless LAN* e Bluetooth 4.1 integrados, o que permite a utilização de alguns sistemas operacionais como: Raspbian (Linux), Ubuntu MATE (Linux) e Windows 10 IoT. Por não possuir memória não volátil, como um disco rígido, seu SO deve ser instalado em um cartão de memória Micro SD. Neste projeto foi escolhido o sistema operacional Ubuntu MATE 16.04 por ser considerado uma versão mais leve do Ubuntu e apropriado para o *hardware* do Raspberry.

Estas características permitem que o Raspberry Pi seja utilizado em aplicações que vão além do aprendizado de programação em escolas, sendo a Internet das Coisas um exemplo de aplicação. A Figura 2.1 abaixo apresenta o Raspberry Pi 3 model B com as especificações de hardware discutidas anteriormente.



Figura 2. 1- *Raspberry Pi 3 Model B* [5].

2.2 Módulo GPS (u-blox NEO-6M)

O módulo u-blox NEO-6M, evidenciado pela Figura 2.2, é utilizado em aplicações onde a localização e o posicionamento georreferenciado se torna necessário. Ele utiliza a tecnologia GPS para realizar tal tarefa.

A série de módulos NEO-6 é uma família de receptores GPS autônomos com o mecanismo de posicionamento de alto desempenho u-blox 6. Esses receptores adaptáveis e de baixo custo oferecem inúmeras opções de conectividade em um pacote miniatura de 16 x 12,2 x 2,4 mm. Sua arquitetura compacta e opções de consumo de energia e memória tornam os módulos NEO-6 ideais para dispositivos móveis operados por restrições de bateria e espaço [6].

O motor de posicionamento de 50 canais u-blox 6 possui um Time-To-First-Fix (TTFF) de menos de 1 segundo. O mecanismo de aquisição dedicado, com 2 milhões de correlacionadores é capaz de realizar buscas paralelas em tempo/frequência, permitindo que ele encontre satélites instantaneamente. O design e a tecnologia suprimem as fontes de interferência e reduzem os efeitos de multipercurso, oferecendo aos receptores de GPS NEO-6 um excelente desempenho de navegação, mesmo nos ambientes mais desafiadores [6].

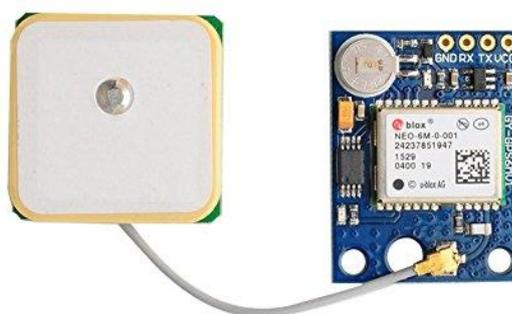


Figura 2. 2 – Módulo GPS u-blox NEO-6M [20].

2.3 Sistema de Informação Geográfica (SIG)

Sistemas de Informação Geográfica (SIG) são sistemas destinados ao tratamento de dados referenciados espacialmente. Estes sistemas manipulam dados de diversas fontes como mapas, imagens

de satélites, cadastros e outras, permitindo recuperar e combinar informações e efetuar os mais diversos tipos de análise sobre os dados [8].

Um software SIG é projetado para entrada, armazenamento, edição, recuperação, análise e saída de dados referenciados geograficamente (dados vinculados explicitamente a pontos conhecidos na superfície da Terra). Em suma, trata-se de objetos localizados espacialmente, distribuições espaciais, e suas análises espaciais. Podemos, por exemplo, ter um banco de dados que contenha estradas, países, cidades, rios e muitas outras características geográficas [7].

2.4 *Global Positioning System (GPS)*

O GPS é um sistema de navegação por rádio baseado no espaço que consiste em uma constelação de satélites e uma rede de estações terrestres usadas para monitoramento e controle. Atualmente, 32 satélites GPS orbitam a Terra a uma altitude de aproximadamente 11.000 milhas, fornecendo aos usuários informações precisas sobre posição, velocidade e tempo em qualquer parte do mundo e em todas condições meteorológicas [9].

Um receptor de GPS deve receber um sinal de pelo menos três satélites, para calcular uma posição 2D (latitude e longitude). Com quatro ou mais satélites visíveis, o receptor pode determinar a posição 3D do usuário (latitude, longitude e altitude), assim como estabelecer a sincronização de relógio com o quarto satélite. Apesar de com apenas 3 satélites em visada o sistema seria capaz de calcular a posição do receptor, a presença do quarto satélite (ou mais) torna-se essencial para o sistema devido ao sincronismo de relógio. Caso não ocorra o sincronismo, todo o sistema se torna obsoleto.

A Figura 2.3 ilustra a constelação de satélites GPS em sua órbita ao redor da Terra. São necessários pelo menos 24 satélites operacionais, em órbitas específicas, para que em qualquer ponto do planeta ao menos 4 satélites estejam visíveis ao receptor.



Figura 2. 3 – Constelação de satélites do sistema GPS [9].

2.5 Tecnologia 3G

A tecnologia de terceira geração (3G) de padrões e tecnologias de telefonia móvel, baseado na família de normas da União Internacional de Telecomunicações, no âmbito do Programa Internacional de Telecomunicações Móveis International Mobile Telecommunications (IMT-2000), trazem grandes inovações tecnológicas e de comportamento humano [10].

As tecnologias 3G permitem às operadoras da rede oferecer a seus usuários uma ampla gama dos mais avançados serviços, já que possuem uma capacidade de rede maior, por causa de uma melhora na eficiência espectral. Entre os serviços há a telefonia por voz e a transmissão de dados a longas distâncias, tudo em um ambiente móvel. As operadoras de telefonia celular implantaram suas redes sempre buscando uma atualização, para estarem de acordo com as grandes exigências do mercado de telecomunicações, onde o usuário busca um padrão de serviços cada vez melhor [10].

A terceira geração de telefonia móvel, na teoria, permite até 7 Mbps de taxa *downlink*. Porém, na prática, devido aos diferentes níveis de conservação e robustez das redes das operadoras de telefonia móvel, e também devido ao estado de deslocamento do usuário (móvel ou estático), a taxa varia entre 400 Kbps e 4 Mbps.

2.6 Servidor LAMP

O termo LAMP faz referência ao conjunto de softwares que fornecem serviços básicos para aplicações *web*. Esta solução reúne o sistema operacional Linux, o servidor *web* Apache, o banco de dados MySQL e as linguagens de programação PHP ou Python.

L – Linux

A – Apache

M – MySQL

P – PHP/Python

2.6.1 Linux

Linux é um Sistema Operacional livre criado por uma comunidade de desenvolvedores de código aberto. Linux serve como uma alternativa para os já conhecidos Windows ou OS/X. Diferente desses sistemas, o Linux é um projeto de muitas fontes onde qualquer um pode alterá-lo [11].

Linux é baseado no antigo Sistema Unix, que usa o princípio de design modular. Neste tipo de sistema, o SO disponibiliza ferramentas simples, enquanto shell scripting e linguagem de comandos

combinam-se para realizar os mais complexos fluxos de trabalho. O Linux suporta um bom número de linguagens de programação, incluindo PHP, Perl, Ruby, Python e outras. Esta diversidade dá ao desenvolvedor a liberdade de escrever seu código na linguagem que ele achar mais confortável [11].

2.6.2 Apache

O *Apache HTTP Server Project* é um esforço de desenvolvimento e manutenção de um servidor *web* HTTP de código aberto para sistemas operacionais modernos, incluindo UNIX e Windows. O objetivo deste projeto é fornecer um servidor seguro, eficiente e extensível que proporcione serviços HTTP em sincronia com os padrões HTTP atuais [12].

O servidor Apache HTTP (“httpd”) foi lançado em 1995 e tem sido o servidor *web* mais popular na internet desde 1996 [12]. A Figura 2.4 apresenta os dados da pesquisa feita pela Netcraft, onde em Junho de 2017, cerca de 46% dos servidores no mundo estavam rodando o Apache, 20% Nginx e 8% Microsoft e Google.

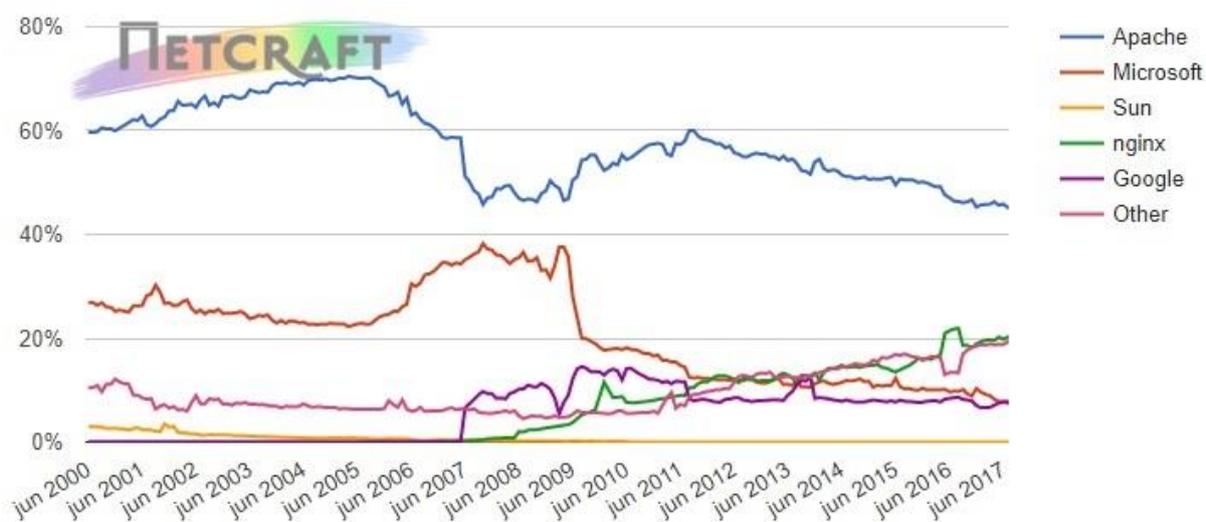


Figura 2.4 Histórico de utilização de servidores web pelo mundo [13].

2.6.3 MySQL

O MySQL é o banco de dados de código aberto mais conhecido no mundo. Com comprovado desempenho, confiabilidade e facilidade de uso, o MySQL tornou-se a principal opção de banco de dados para aplicativos baseados na Web, usado por propriedades da Web de alto perfil, incluindo Facebook, Twitter, YouTube, e outros [14].

Trata-se de um Software Gerenciador de Banco de Dados (SGBD) relacional que utiliza a linguagem SQL (*Structured Query Language* – Linguagem de Consulta Estruturada) para inserir, acessar e gerenciar um banco de dados.

2.6.4 PHP

O PHP (um acrônimo recursivo para PHP: *Hypertext Preprocessor*) é uma linguagem de script *open source* de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento web [15].

A diferença de PHP com relação a linguagens semelhantes a Javascript é que o código PHP é interpretado no servidor, sendo enviado para o cliente apenas HTML (*HyperText Markup Language*) puro. Desta maneira é possível interagir com bancos de dados e aplicações existentes no lado servidor, com a vantagem de não expor o código fonte para o cliente.

2.6.5 phpMyAdmin

O phpMyAdmin é uma ferramenta de *software* livre escrita em PHP, destinada a lidar com a administração do MySQL através da *web*. O phpMyAdmin suporta uma ampla gama de operações no MySQL e no MariaDB. Operações frequentemente usadas (gerenciamento de banco de dados, tabelas, colunas, relações, índices, usuários, permissões, etc.) podem ser realizadas através da interface do usuário, enquanto você ainda tem a capacidade de executar diretamente qualquer instrução SQL [16].

2.7 Google Text To Speech (GTTS)

O Google Text-To-Speech permite que os desenvolvedores sintetizem voz natural com 30 vozes diferentes oferecidas, disponíveis em vários idiomas e variantes. Ele aplica a pesquisa inovadora do DeepMind na WaveNet e as poderosas redes neurais do Google para fornecer a mais alta fidelidade possível. Com esta API fácil de usar, você pode criar interações realistas com seus usuários em vários aplicativos e dispositivos [17].

2.8 Python

Python é uma linguagem de programação interpretada, orientada a objetos e de alto nível, com semântica dinâmica. Suas estruturas de dados incorporadas de alto nível, combinadas com digitação dinâmica e vinculação dinâmica, a tornam muito atraente para o desenvolvimento de aplicativos, bem como para uso como uma linguagem de script ou colagem para conectar os componentes existentes juntos. A sintaxe simples e fácil de aprender do Python enfatiza a legibilidade e, portanto, reduz o custo de manutenção do programa. O Python suporta módulos e pacotes, o que incentiva a modularidade do programa e a reutilização de código. O interpretador Python e a extensa biblioteca padrão estão disponíveis em formato fonte ou binário, sem custo para todas as principais plataformas, e podem ser distribuídos gratuitamente [19].

3 METODOLOGIA

3.1 DELIMITAÇÃO DO TEMA

O presente trabalho tem como proposta desenvolver um dispositivo de auxílio auditivo programado, para a assistência na locomoção de deficientes visuais presentes na área da comunidade acadêmica da UnB, utilizando informação georreferenciada, sendo também aplicável às demais áreas geográficas, visto que o sistema GPS oferece cobertura global. Este dispositivo envia um aviso de áudio ao usuário permitindo que o mesmo saiba com antecedência as situações de riscos que por ventura possam ocorrer durante sua locomoção pela Universidade de Brasília.

O fluxograma da Figura 3.1 apresenta as etapas realizadas para a elaboração do projeto.

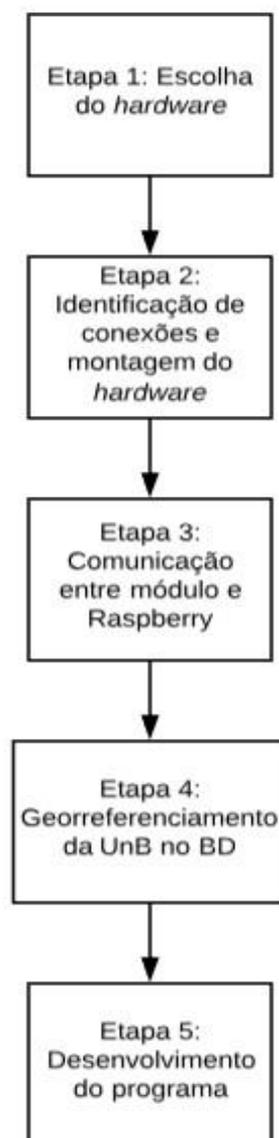


Figura 3. 1 Fluxograma de etapas do projeto.

Etapa 1 – Escolha do *hardware*: consiste no levantamento de equipamentos necessários para a realização do projeto.

Etapa 2 – Identificação de conexões e montagem do *hardware*: analisar os pinos de conexão do módulo e do Raspberry, e realizar as devidas conexões físicas de alimentação e comunicação.

Etapa 3 – Comunicação entre módulo e Raspberry: preparar o SO para que o Raspberry interprete os dados enviados pelo módulo GPS.

Etapa 4 – Georreferenciamento da UnB no BD: criar um Banco de Dados MySQL contendo as áreas de interesse presentes no espaço geográfico da UnB, suas respectivas áreas de risco para a locomoção e as mensagens de aviso.

Etapa 5 – Desenvolvimento do programa: apresenta a ferramenta utilizada para a concepção do código que realiza a interpretação entre BD e localização geográfica, a fim de que o aviso sonoro seja emitido.

3.1.1 Escolha do *hardware*

Tendo o tema sido delimitado, o primeiro passo foi realizar um levantamento de preços dos produtos e a respectiva compra dos mesmos. A Tabela 3.1 abaixo apresenta os produtos adquiridos, a quantidade, o preço unitário e o local onde foram obtidos.

Tabela 3.1 – Listagem dos produtos adquiridos para a realização do dispositivo [autor].

Quantidade	Produto	Preço Unitário (R\$)	Local
1	Raspberry Pi 3 model B	240,00	Ncam informática
1	Módulo GPS u-blox NEO-6M	70,00	Mercado Livre
5	Jumper	0,80	Contato
1	Chave on/off	2,81	Contato
1	Power bank	55,00	Mercado Livre

É válido lembrar que foram sondados módulos de outros fabricantes (como o Breakout Ultimate GPS v3 da Adafruit), e, dentre eles, o u-blox apresentou o melhor custo/benefício.

3.1.2 Identificação de conexões e montagem do *hardware*

Para que o Raspberry se comunique de maneira apropriada com o módulo, as devidas conexões tiveram que ser realizadas.

Em um primeiro momento, foi necessário efetuar a verificação das pinagens do GPIO presente no Raspberry Pi. A Figura 3.2 abaixo apresenta o diagrama de pinos da placa.

Raspberry Pi 3 GPIO Header

Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1 , I ² C)		DC Power 5v	04
05	GPIO03 (SCL1 , I ² C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I ² C ID EEPROM)		(I ² C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

29/02/2016

Figura 3. 2 - Diagrama de pinos do GPIO do Raspberry 3 [18].

Como o NEO-6M trabalha com uma tensão entre 3V e 5V, o pino #1 foi considerado o mais apropriado para a alimentação, pois entrega 3,3V. Os demais pinos utilizados foram os UART0 TX (#08), UART RX (#10) e Terra (#14). Estes pinos UART funcionam como uma interface de comunicação serial do Raspberry.

Desta forma, a conexão física do módulo GPS com o Raspberry Pi foi executada, atentando ao fato de que a entrada RX do módulo deve ser conectada à saída TX do Raspberry e vice-versa. As Figuras 3.3 e 3.4 evidenciam o resultado desta etapa já com os *jumpers* soldados no módulo e devidamente conectados à placa.



Figura 3. 3 – Ligação dos *jumpers* ao Raspberry.

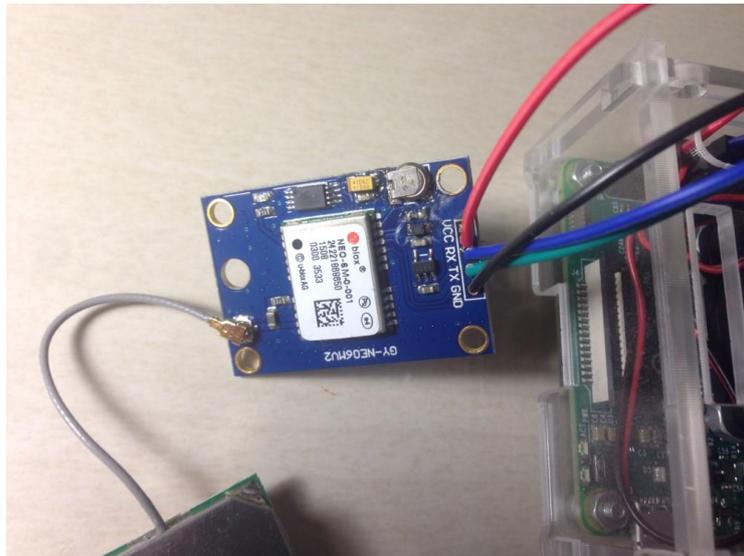


Figura 3. 4 – *Jumpers* soldados ao módulo GPS.

3.1.3 Comunicação entre módulo e Raspberry

Após a instalação do sistema operacional Ubuntu Mate 16.04 no cartão Micro SD, e com o módulo devidamente conectado ao RPi, o sistema foi ligado. Foi observado que o módulo estava piscando seu LED, o que significa que estava recebendo sinal dos satélites, porém o Raspberry ainda não era capaz de interpretar os sinais recebidos pela GPIO.

Desta forma, uma sequência de comandos foram realizados para que os sinais recebidos pelo “serial0” (UART TX e UART RX) do Raspberry sejam interpretados pelo sistema.

Os comandos executados para a comunicação do Raspberry com o módulo se encontram no Apêndice A1.

A Figura 3.5 demonstra, com a utilização da ferramenta GPSTool, que o Raspberry agora interpreta os dados recebidos na interface “serial0”. Tais dados são informações como: hora, latitude, longitude, altitude, velocidade, número de satélites em visada, entre outros.

```

root@luís-rpi: ~
Time: 2018-03-16T00:19:12.000Z
Latitude: 15.747813 S
Longitude: 47.894384 W
Altitude: 1068.7 m
Speed: 0.0 kph
Heading: 266.7 deg (true)
Climb: 12.0 m/min
Status: 3D FIX (4 secs)
Longitude Err: +/- 22 m
Latitude Err: +/- 11 m
Altitude Err: +/- 34 m
Course Err: n/a
Speed Err: +/- 158 kph
Time offset: 0.485
Grid Square: GH64bg
PRN: Elev: Azim: SNR: Used:
1 15 003 30 Y
3 57 052 25 Y
6 21 250 00 N
7 54 267 30 Y
8 07 045 23 Y
9 37 200 37 Y
11 08 009 24 Y
16 23 127 00 N
18 00 000 14 N
22 36 043 17 Y
23 40 147 24 Y
28 05 340 26 N
30 34 293 27 Y

```

Figura 3. 5 – Demonstração da comunicação entre módulo e Raspberry.

3.1.4 Georreferenciamento da UnB no BD

Para a tarefa de georreferenciar os pontos de risco e alguns locais de interesse presentes na UnB, foi necessária, em um primeiro momento, a realização de um levantamento de dados *in loco*. Durante esta verificação foram encontrados diversos sítios onde a locomoção do deficiente visual estaria comprometida. Tais lugares apresentam pisos irregulares, buracos e/ou obstáculos no caminho.

Mais adiante, foi realizado um levantamento mais detalhado, com o auxílio da estudante Késsia Rodrigues, aluna do curso de Engenharia Química da UnB inscrita no PPNE-UnB, onde ela apontou alguns detalhes e locais que não foram percebidos durante os levantamentos de dados iniciais. Tal ajuda foi importante para o projeto, pois desta forma os dados georreferenciados se tornam corroborados pela vivência de um/a estudante com deficiência visual.

A Figura 3.6 apresenta um exemplo de possível ponto de risco identificado durante a pesquisa.



Figura 3. 6 - Desnível em calçada próximo à entrada do PJC.

Logo após, tendo todas as informações sido coletadas, foi criada uma tabela com todos os locais e mensagens que serão passados em um momento posterior para o banco de dados. A tabela 3.2 representa a tabela que mais tarde será incorporada no banco de dados do projeto. Devido ao fato da coluna mensagem apresentar um campo de texto maior, seu conteúdo será apresentado logo abaixo da tabela.

Tabela 3.2 – Base de informações georreferenciadas para o banco de dados [autor].

ID	nome	lat inf	lat sup	lon esq	lon dir	msg
1	Zona construção	-15.762578	-15.761286	-47.874472	-47.872861	
2	FT	-15.763996	-15.763110	47.873897	47.871744	
3	Estacionamento ICC Norte	-15.762317	-15.761527	-47.871625	-47.870731	
4	RU	-15.764546	-15.763856	-47.870882	-47.870062	
5	Escada FE	-15.767882	-15.767781	-47.871342	-47.871254	
6	Placas ICC Norte	-15.760817	-15.760673	-47.871138	-47.871003	
7	Poste 1	-15.760045	-15.759979	-47.870767	-47.870630	
8	Entrada/Saída 1	-15.760347	-15.760121	-47.870657	-47.870455	
9	Calçada 1	-15.760093	-15.760030	-47.870419	-47.870222	
10	Escada FACE	-15.758781	-15.758596	-47.871432	-47.871287	
11	Obstáculos 1	-15.758747	-15.758550	-47.869766	-47.869649	
12	Calçada 2	-15.759079	-15.758980	-47.870029	-47.869860	
13	Calçada 3	-15.758524	-15.758279	-47.870602	-47.870446	
14	Calçada 4	-15.764665	-15.764489	-47.871140	-47.870956	
15	Calçada 5	-15.766693	-15.766610	-47.871120	-47.870952	
16	Calçada 6	-15.764576	-15.764435	-47.869891	-47.869655	
17	Obstáculos 2	-15.764640	-15.764542	-47.872758	-47.872395	
18	Entrada Estacionamento BSAS	-15.766447	-15.766132	-47.866846	-47.866619	
19	Rampa Reitoria	-15.763297	-15.763050	-47.867535	-47.867166	

1 – Zona Construção – Alerta! Zona de construção.

2 – FT – Área da Faculdade de Tecnologia, alerta para escadas em diagonal entre departamentos, utilize as rampas.

3 – Estacionamento ICC Norte – Área do Estacionamento do I C C Norte, trânsito de veículos.

4 – RU – Área do Restaurante Universitário.

5 – Escada FE – Alerta! Escada na passagem entre os prédios da F E 1 e F E 2.

6 – Placas ICC Norte – Alerta! Placas de trânsito na calçada.

7 – Poste 1 - Alerta! Poste na calçada do ponto de ônibus do P J C.

8 – Entrada/Saída1 – Alerta! Entrada e saída de veículos do subsolo do I C C Norte.

9 – Calçada 1 – Alerta! Piso irregular entre ponto de ônibus e entrada do subsolo do I C C Norte.

10 – Escada FACE – Alerta! Escadaria com piso irregular na entrada da F A C E.

11 – Obstáculos 1 – Alerta! Plantas e placa de travessia de pedestres na calçada do I R E L.

12 – Calçada 2 – Alerta! Calçada com desnível.

13 – Calçada 3 – Alerta! Buraco na calçada lateral do P J C.

14 – Calçada 4 – Alerta! Piso irregular no caminho próximo à faixa de pedestres do R U com o Departamento de Música.

15 – Calçada 5 – Alerta! Piso irregular na frente da placa de oficinas do I D A.

16 – Calçada 6 – Alerta! Piso irregular no caminho entre I C C Sul e R U.

17 – Obstáculos 2 – Alerta! Corrente no caminho da FT para o S G 11.

18 – Entrada Estacionamento BSAS – Alerta! Entrada e saída de veículos na calçada do B S A S.

19 – Rampa Reitoria – Alerta! Rampa de acesso suspensa.

Os locais georreferenciados podem ser visualizados pela Figura 3.7 na qual apresenta um mapa realizado com o auxílio da API da Google My Maps.

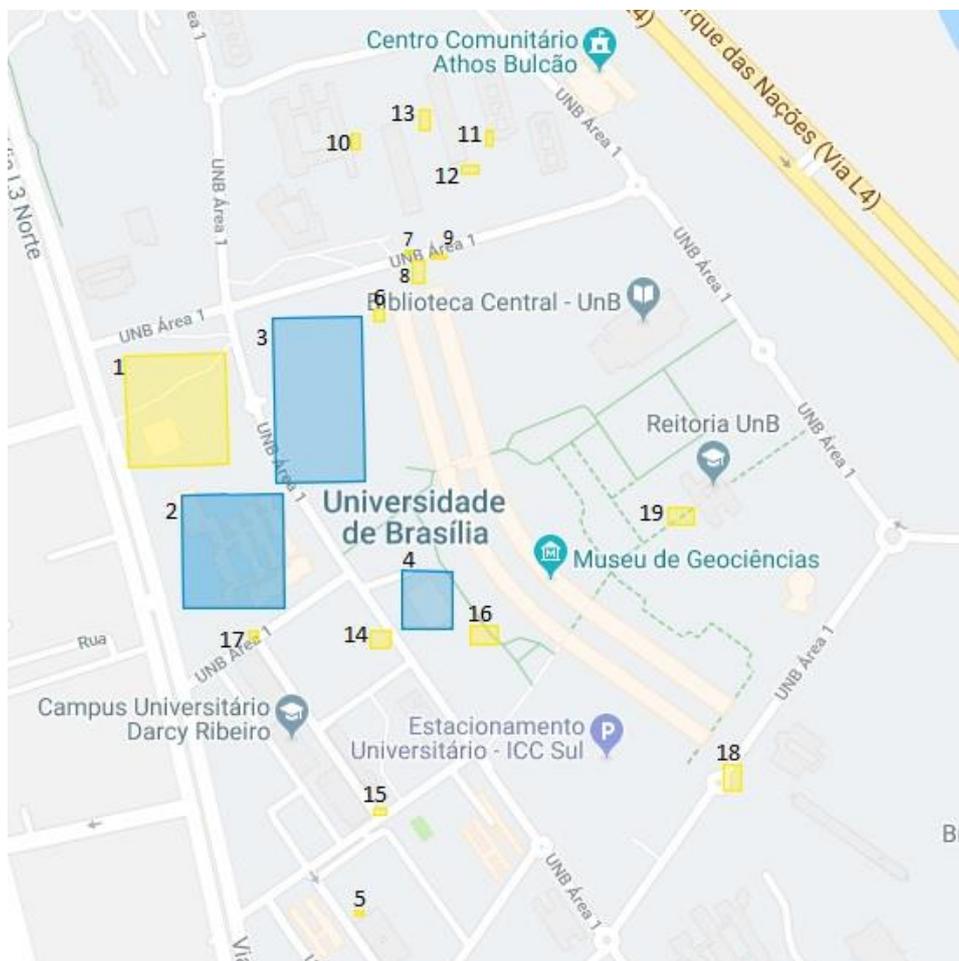


Figura 3.7 – Mapa dos locais georreferenciados no projeto.

Desta forma, o passo seguinte foi realizar a instalação da pilha LAMP no sistema operacional. Devido ao fato do sistema já ser baseado na plataforma Linux, o próximo passo foi efetuar a instalação do servidor Apache, do bando de dados MySQL e do PHP.

Com o terminal do Ubuntu aberto, alguns comandos foram realizados para a instalação correta destas ferramentas.

Apache – A instalação do servidor Apache seguiu a ordem de comandos apresentados no Apêndice A2. A Figura 3.8 apresenta o resultado final da instalação.

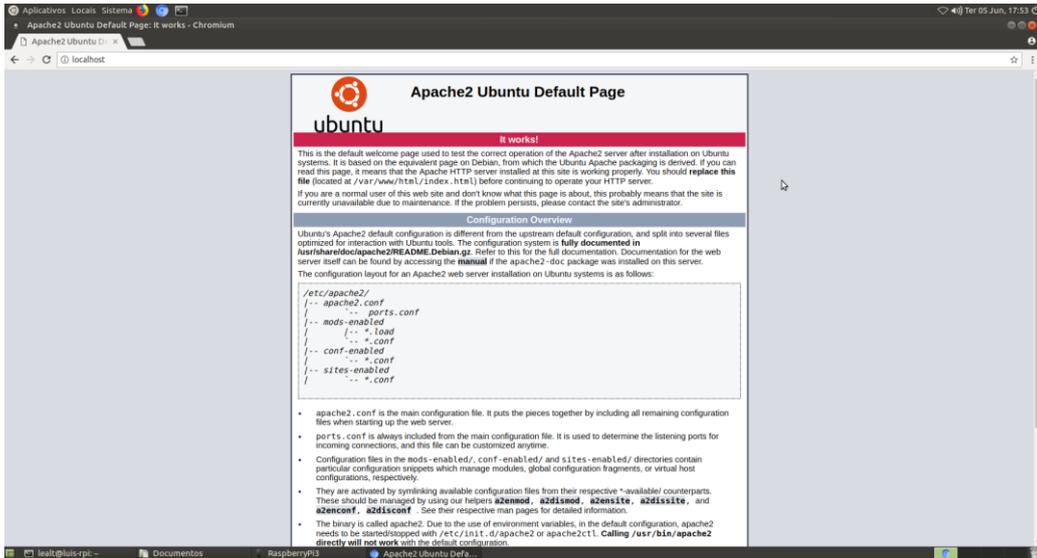


Figura 3. 8 – Página index.html do servidor Apache no localhost.

MySQL - A instalação do banco de dados MySQL seguiu a ordem de comandos apresentados no Apêndice A2.

PHP - A instalação do PHP seguiu a ordem de comandos apresentados no Apêndice A2. Pela Figura 3.9 foi possível verificar que o PHP foi instalado corretamente.



Figura 3. 9 – Página info.php no localhost.

Para uma melhor visualização e gerenciamento do banco de dados MySQL, a ferramenta phpMyAdmin foi instalada após a instalação da pilha LAMP.

phpMyAdmin - A instalação do phpMyAdmin seguiu a ordem de comandos apresentados no Apêndice A2. A Figura 3.10 foi gerada após a instalação.

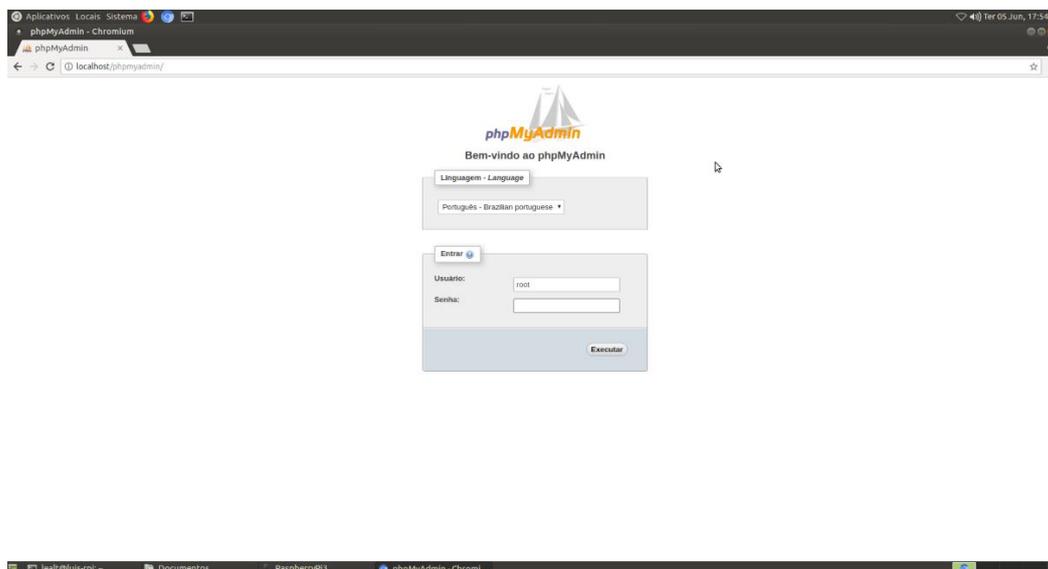


Figura 3. 10 – Tela de login do phpMyAdmin.

Logo após, os dados da Tabela 3.2 foram passados para o banco de dados. Foi gerado um banco de dados com o nome de “bd_coord” com uma tabela de nome “areas”.

O campo “ID” foi criado como sendo do tipo INT com chave de índice PRIMARY e função de AUTO_INCREMENT habilitada. Este campo corresponde ao código de identificação do local.

O campo “nome” foi criado como sendo do tipo VARCHAR, uma *string*, com espaço limitado a 30 caracteres. Este campo pode corresponder ao nome da área ou ao nome do alerta georreferenciado.

Os campos “lat_inf”, “lat_sup”, “lon_esq” e “lon_dir” foram criados como tipo DOUBLE, na qual suportam 8 bytes de armazenamento. Estes campos correspondem a duas coordenadas geográficas que são usadas no algoritmo de criação de áreas.

O campo “msg”, do tipo VARCHAR com espaço limitado a 100 caracteres e codificação do tipo UTF8_GENERAL_CI, foi criado para armazenar a mensagem de informação ou alerta que será dita ao deficiente visual.

3.1.5 Desenvolvimento do programa

Com o banco de dados criado, o próximo passo foi desenvolver a ferramenta que irá fazer a interface de comunicação entre os dados georreferenciados e a informação de localização atual do usuário, tendo como resultado a mensagem de áudio caso o deficiente visual esteja na área mapeada no BD.

Para tal tarefa, foi utilizada a linguagem de programação Python por ser uma linguagem de *script* de código aberto, interpretada, multiplataforma e simples, e também pelo fato de oferecer uma base de biblioteca GPS. Algumas bibliotecas adicionais tiveram que ser instaladas, são elas: “python-gps”, “MySQLdb”, “gtts”, “mpg321” e o “shapely”.

A biblioteca “python-gps” inclui as funções de conexão com o módulo GPS pela comunicação serial.

O “MySQLdb” é uma biblioteca que possui as funções que realizam a conexão e comunicação do programa com o MySQL.

O “gtts” é uma biblioteca que implementa a API do Google *Text To Speech* no programa python. Possui a função que recebe a mensagem escrita e transforma em áudio no formato mp3.

O “mpg321” tem a função de player por linha de comando.

A biblioteca “shapely” é um pacote Python para manipulação e análise de objetos geométricos planares. Ele é baseado nas bibliotecas amplamente distribuídas do GEOS (o mecanismo do PostGIS) e do JTS (do qual o GEOS é portado). Shapely não se preocupa com formatos de dados ou sistemas de coordenadas, mas pode ser facilmente integrado a pacotes que são.

O Apêndice A3 apresenta os comandos para a instalação das bibliotecas e o programa de nome “auxilio.py” que executa toda a lógica de intercomunicação entre o BD e o módulo juntamente com o aviso sonoro.

Em um primeiro momento o código realiza todos os *imports* de bibliotecas necessárias para o programa. A Figura 3.11 apresenta as linhas de 1 a 7 com as bibliotecas necessárias sendo importadas ao programa.

```
1 import RPi.GPIO as GPIO
2 import os
3 import MySQLdb
4 from gps import *
5 from shapely.geometry import Point
6 from shapely.geometry.polygon import Polygon
7 from gtts import gTTS
```

Figura 3. 11 – Parte do código relativo aos *imports*.

A próxima sessão do código busca efetuar as inicializações do GPS, dos pinos GPIO do Raspberry e a conexão com o banco de dados. A Figura 3.12 demonstra esta parte do código pelas linhas de 9 a 28.

```
9 global c
10 global db
11
12 flag = 0
13
14 #Iniciamos o gps
15 session = gps()
16 session.stream(WATCH_ENABLE|WATCH_NEWSTYLE)
17
18 #Iniciamos os pinos do Raspberry pi
19 GPIO.setmode(GPIO.BCM)
20 GPIO.setup(23, GPIO.IN, pull_up_down=GPIO.PUD_UP)
21 GPIO.setup(16, GPIO.OUT)
22
23 #Conexao com Banco de Dados
24 try:
25     db = MySQLdb.connect("localhost","root","mysqlpi","bd_coord")
26     c = db.cursor()
27 except:
28     print ("Servidor nao conectado!")
```

Figura 3. 12 – Parte do código onde as inicializações são feitas.

O pino 23 foi escolhido para implementar a chave *on/off* do programa, dando a opção ao usuário de desligar momentaneamente o aviso sonoro caso ele entre em alguma área mapeada no banco de dados. Já o pino 16 foi escolhido para conectar o LED de suporte que indica quando o programa está rodando normalmente. Este LED pode ser retirado para fins de economia de energia, visto que foi apenas necessário no processo de desenvolvimento do projeto.

Logo após, são definidos os métodos do programa. Cada método é chamado para realizar uma tarefa específica. A Figura 3.13 exibe as linhas de 31 a 52 contendo os três métodos que são chamados durante a execução do programa.

```
31 #Funcao que emite um pulso no LED
32 def registro():
33
34     #Pulso apos gravar
35     GPIO.output(16, GPIO.LOW)
36     time.sleep(0.2)
37     GPIO.output(16, GPIO.HIGH)
38
39 #Funcao da logica de calculo da area
40 def ponto_area(lat1, lon1, lat2, lat3, lon2, lon3):
41
42     ponto = Point(lat1, lon1)
43     area = Polygon([(lat3, lon2), (lat3, lon3), (lat2, lon3), (lat2, lon2)])
44     return(area.contains(ponto))
45
46 #Funcao que emite a mensagem ao PNE
47 def fala(mensagem):
48
49     AudioDoTexto = gTTS(mensagem, lang='pt')
50     AudioDoTexto.save("/tmp/audio.mp3")
51     os.system("mpg321 /tmp/audio.mp3")
52     os.remove("/tmp/audio.mp3")
```

Figura 3.13 – Parte do código contendo os métodos do programa *auxilio.py*.

O método “registro()” é chamado para emitir um pulso no LED alocado no pino 16 do GPIO. Este pulso indica que a chave *on/off* se encontra na posição ligada, de modo que o aviso sonoro seja emitido ao deficiente visual. Com a chave desligada, o LED não emite pulso e o aviso não é emitido.

O método “ponto_area()” recebe como parâmetro as coordenadas geográficas do módulo juntamente com duas coordenadas geográficas da área mapeada no BD. Esta função retorna *true* caso o ponto atual do módulo esteja dentro da zona georreferenciada.

Já o método “fala()” recebe como parâmetro a *string* de mensagem contida no BD, de modo que o áudio seja sintetizado pela API do Google, guardado dentro dos arquivos temporários do sistema, executado pelo player “mpg321” e removido do diretório. Esta é uma das funções mais importantes do projeto, nela o áudio é emitido para o usuário. Por fazer uso da API GTTS, uma conexão do Raspberry com a internet se torna necessária.

Em seguida, o programa cai em um laço *while* de parâmetro *true*, fazendo com que ele caia em uma espécie de *loop* infinito. Este laço contém um operador condicional *if/else* principal que analisa o sinal elétrico do pino 23 do GPIO controlado pela chave. Caso esteja ligada, o programa primeiro recebe o valor de latitude e longitude apresentado pelo módulo GPS e guarda em duas variáveis respectivas. Depois, ele realiza uma consulta SQL no banco de dados com o intuito de receber todos os valores da

tabela “áreas”. Um laço *for* faz a chamada do método “*ponto_area()*” em cada iteração. Se o retorno for positivo, uma chamada para o método “*fala()*” é feito e a mensagem escrita no BD será falada ao PNE. Caso a chave esteja desligada o programa entra no modo de espera explicado anteriormente.

É necessário lembrar que a única função do dispositivo é implementada pelo programa, desta forma, ao ligar o Raspberry sem uma tela o usuário só possui a opção de interação com a chave, pois o programa roda em segundo plano no sistema, retirando a necessidade de uso do terminal ou outro método de execução.

Uma vez ligado, não existe outro meio de parar a execução do programa *auxilio.py* que não seja pelo desligamento forçado da fonte de energia do dispositivo ou o eventual consumo total da bateria do *power bank*. Isto torna o dispositivo mais fechado ao usuário final, ele somente interage com a chave seletora.

4 ESTUDO DE CASO

4.1 TESTES INICIAIS E PROVA DE CAMPO

Uma vez que todos os processos de implementação de *hardware* e *software* do projeto foram implementados, o passo seguinte foi a realização de testes onde cada função do dispositivo foi verificado. Estas funções envolvem principalmente a exatidão dos dados geográficos adquiridos pelo módulo GPS e a emissão das mensagens em formato de áudio para o usuário.

Para este teste, o dispositivo foi ligado em uma fonte de alimentação fixa com a utilização da rede cabeada doméstica, sendo que duas áreas foram inseridas no banco de dados. A área de ID 1 na tabela incluía o ponto de localização do dispositivo e a área de ID 2 não fazia inclusão do ponto, representando, por consequência, duas áreas geográficas distintas sem interseção.

Foi observado que o dispositivo emitiu somente a mensagem contida na área de ID 1. Ao retirar a área que incluía a localização do dispositivo no BD, nenhuma mensagem foi emitida. Desta forma, ficou comprovada a precisão do módulo juntamente com o algoritmo de checagem do BD e emissão de áudio.

Para um segundo teste, agora em uma área externa e com o dispositivo conectado à rede 3G de um telefone móvel e alimentado por um *power bank*, foram incluídas novamente duas áreas ao BD. Ao ligar o dispositivo e acionar a chave, nenhuma mensagem foi enunciada, posto que o módulo não se encontrava dentro de uma das áreas georreferenciadas. Ao adentrar em uma das áreas, o dispositivo começou a emitir a mensagem após um período de tempo, ocasionando uma certa latência ao sistema.

Esta latência variou entre alguns segundos, e pode ocorrer devido à alguns fatores como: conexão com a rede 3G, posicionamento da antena do módulo e fatores climáticos. Acredita-se que este atraso possa ser tratado aplicando melhorias no tipo de conexão com a rede móvel e na qualidade recepção do sinal dos satélites pela antena do dispositivo.

Desta forma, para a última série de testes, agora na Universidade de Brasília e já com alguns locais presentes no banco de dados, a antena foi posicionada de uma forma que ocorra uma melhor linha de visada com os satélites GPS. Foi observada uma melhora no atraso de emissão da mensagem, dando indícios de que os fatores apontados na fase de testes anterior, de fato, podem atuar no desempenho do dispositivo.

Quanto à atuação do Raspberry Pi juntamente com o módulo GPS não ocorreram falhas de funcionamento ou erros na obtenção dos dados e execução do programa *auxilio.py* quando o dispositivo foi utilizado em uma área aberta.

Quando o dispositivo adentra em uma edificação, a sua antena perde a linha de visada com os satélites, ocasionando perda da comunicação do módulo com o sistema GPS, e uma interrupção momentânea na leitura dos dados feita pelo programa. Ao voltar para uma área aberta, ou melhorar a

relação sinal-ruído no ambiente, o módulo recupera a conexão com os satélites e o programa volta a funcionar. Por este motivo, questões meteorológicas também devem ser levadas em consideração.

Outro fato importante, além da latência, é o tempo necessário para que o módulo reconheça o sinal dos satélites após a primeira inicialização do sistema. Este tempo recebe o nome de *Time-To-First-Fix* ou *Cold Start* e está prevista no manual do fabricante como sendo aproximadamente 27 segundos em condições ideais de ganho da antena.

4.2 DIAGRAMA DE CASO DE USO

A compreensão geral do funcionamento do sistema pode ser ilustrada pelo diagrama de caso apresentado pela Figura 5.1 abaixo.

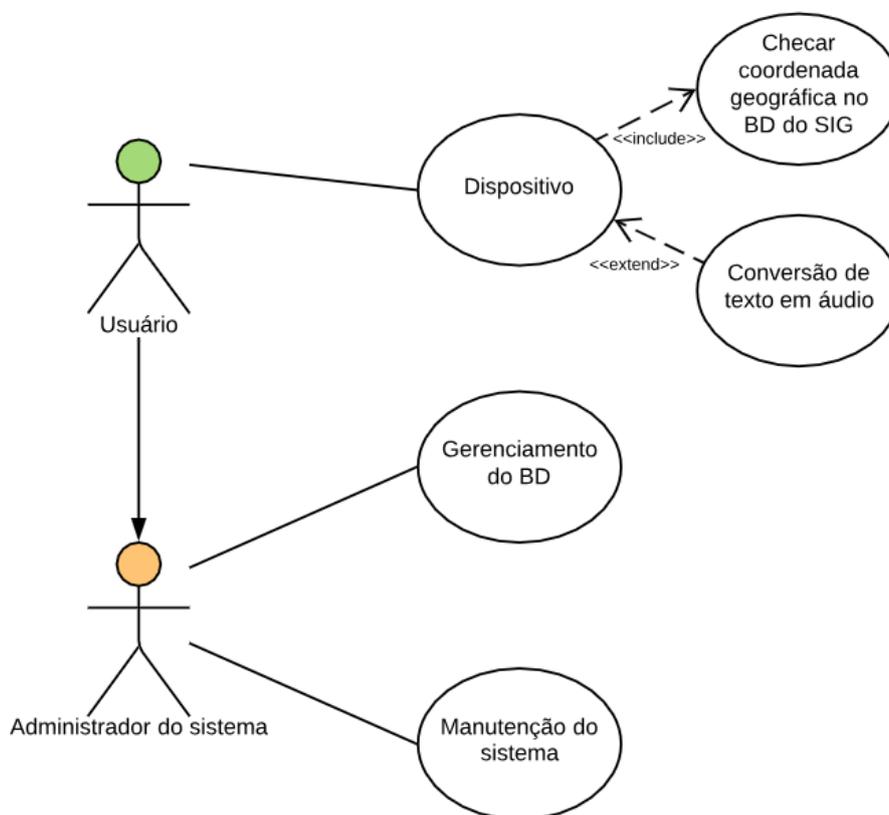


Figura 5. 1 – Diagrama de caso de uso do projeto.

Pelo diagrama é possível observar a presença de dois agentes atuando no sistema de rastreamento auditivo. Estes dois agentes são: o usuário e o administrador do sistema. Esta divisão torna o uso do dispositivo mais simples e intuitiva para o deficiente visual.

As tarefas delegadas ao administrador do sistema dependem da interação do usuário com o mesmo, de modo que, ao presenciar algum erro ou a necessidade de inclusão de algum local de risco ao sistema, o usuário informa ao administrador. O administrador tendo como base as informações dadas pelo usuário, realiza ações de manutenção do sistema do dispositivo (*hardware* e/ou *software*) ou CRUD no banco de dados.

O usuário consiste basicamente de uma pessoa PNE utilizando o sistema de rastreamento auditivo por meio de informações georreferenciadas. Para eles o sistema deve funcionar de forma similar a uma *blackbox*, ou “caixa preta”, ou seja, a complexidade do sistema abaixo da interação com o usuário deve ser oculta. Assim, o sistema se torna de simples uso, cabendo ao PNE apenas as tarefas de utilizar o dispositivo e reportar possíveis falhas do sistema ou atualização de locais georreferenciados ao administrador.

5 RESULTADOS

5.1 DISPOSITIVO DE AUXÍLIO AUDITIVO

Após todo o processo de desenvolvimento do sistema, o conjunto físico final do dispositivo foi composto pelos componentes presentes na Figura 5.2, onde estão apresentados o Raspberry Pi, o módulo GPS e o *power bank* de alimentação do dispositivo.

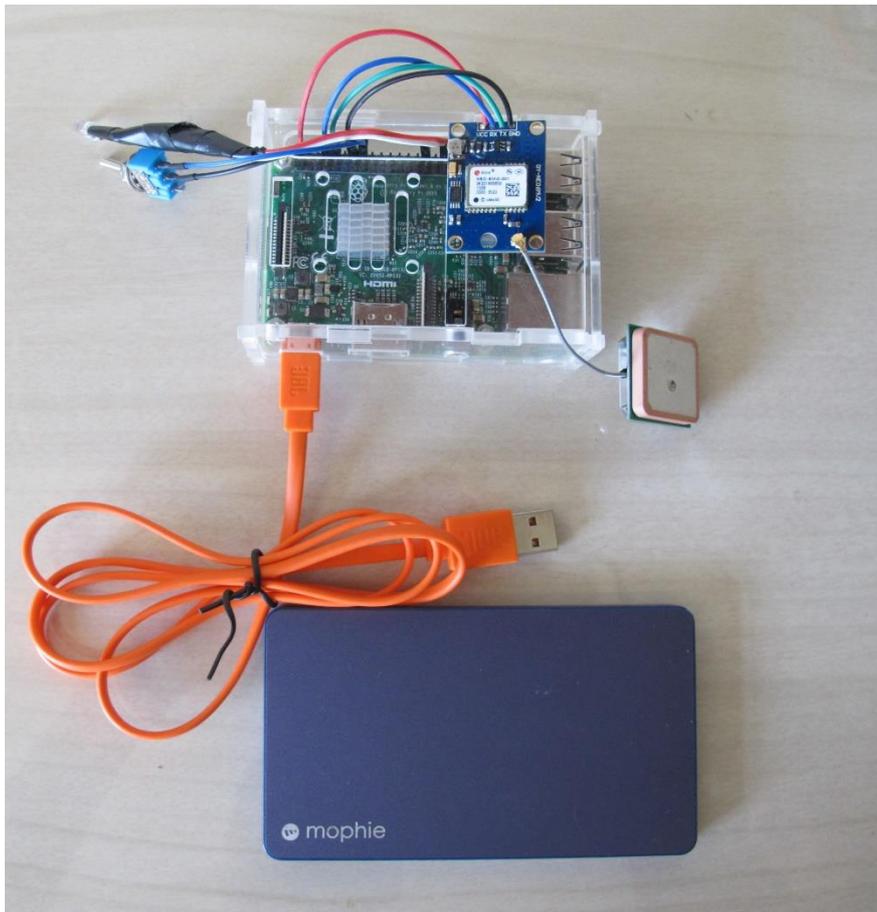


Figura 5. 2 – Dispositivo de auxílio auditivo.

Para a fonte de emissão de áudio existem diversos aparelhos que podem ser utilizados, como fones de ouvidos, *headphones* ou caixas de som, basta que o mesmo utilize uma entrada de áudio do tipo P2 para ser conectada ao jack de saída de áudio do Raspberry Pi. Devido ao fato do deficiente visual utilizar da audição para se orientar, o aparelho mais recomendado seria um fone de ouvido, pois este aparelho possibilita a escuta de sons externos além do que está sendo reproduzido.

O dispositivo poderá ficar em uma mochila, em algum compartimento se possível, basta atentar ao fato de que a antena deve estar à mostra e em linha de visada com o céu para que o seu ganho seja otimizado e se obtenha uma melhor relação sinal-ruído. A Figura 5.3 ilustra um usuário utilizando o dispositivo.



Figura 5. 3 – Exemplo de uso do dispositivo.

Além da mochila, outros itens como bolsas de cintura e doleiras foram analisados para finalidade de abrigar o dispositivo, porém, estes outros itens apresentaram resultados não satisfatórios quanto à localização da antena, possível desconexão dos cabos e maior aquecimento do Raspberry Pi por estar mais junto ao corpo do usuário.

Deste modo, o dispositivo foi concebido a partir da ligação entre *hardware* e *software*, tendo como principais integrantes o banco de dados com informações georreferenciadas, o Raspberry Pi e o módulo GPS. Esta ligação é realizada pelo programa *auxilio.py* escrito na linguagem de programação Python.

É válido lembrar que este dispositivo possui um elevado potencial de aprimoramento no que se refere ao sistema de interação com os usuários e entre os usuários. Pretende-se expandir o conceito deste sistema para uma plataforma de navegação em comunidade, tornando algo semelhante ao que a aplicação “Waze” realiza hoje.

5.2 CONVERSÃO DE TEXTO EM AVISO AUDITIVO

A tarefa principal deste dispositivo, como evidenciado no decorrer deste documento, é a emissão de um aviso por meio de voz ao deficiente visual. Esta tarefa é fundamental para o funcionamento e validação do dispositivo.

Para realizar esta função, o uso da API do Google de nome “Google Text-To-Speech (GTTS)”, neste aparelho, se torna essencial.

A Figura 5.4 abaixo apresenta um diagrama de funcionamento do dispositivo, sendo que o produto final é o aviso de áudio caso o deficiente visual adentre uma área de risco.

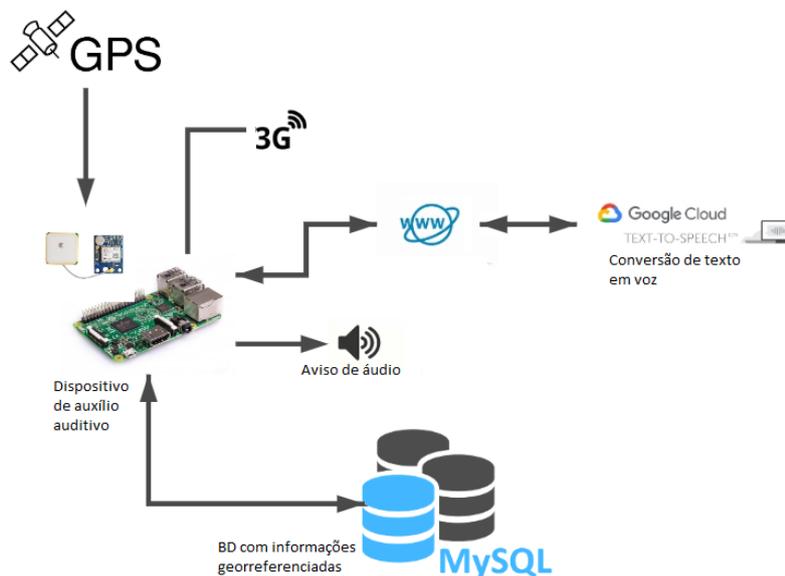


Figura 5. 4 – Arquitetura do dispositivo.

De acordo com o diagrama acima, a tarefa de gerar um aviso sonoro só irá ocorrer após alguns passos serem executados, esses passos consistem em: uma coordenada geográfica recebida pelo sensor coincidir com as coordenadas dentro de uma área georreferenciada no BD que contém o SIG, após isso, o programa realiza o passo seguinte que se resume em pegar o texto do aviso desta área específica no BD e mandar para a função da biblioteca GTTS do Google em Python, que utilizará a API na internet e devolverá o texto do aviso na forma de um arquivo em formato mp3 com uma voz sintetizada.

Desta forma, a conversão de texto em áudio ocorre a partir da ferramenta GTTS na *web*.

6 CONCLUSÃO E TRABALHOS FUTUROS

A temática abordada neste trabalho teve como objetivo a elaboração de um dispositivo de auxílio auditivo para deficientes visuais com a utilização de informações georreferenciadas. Para a realização deste projeto, ferramentas como Raspberry Pi, tecnologia GPS e SIG com banco de dados foram utilizados como base. O projeto se baseou como exemplo, o georreferenciamento de locais e zonas do Campus Darcy Ribeiro da UnB com a finalidade elaborar um sistema que ajude o deficiente visual em sua locomoção e localização, e que previna eventuais acidentes que possam ocorrer com os mesmos ao transitar em uma área de risco.

O critério utilizado na concepção deste projeto consistiu na pesquisa com pelo menos um estudante PNE com o propósito de averiguar as características relevantes para a locomoção e localização de um deficiente visual, além de elaborar um banco de dados com locais de relevância fundamentada. Também foi parte constituinte deste projeto a realização de uma pesquisa bibliográfica acerca das tecnologias presentes no sistema, de modo que fosse obtido um embasamento para a escolha das ferramentas que melhor se adequam no propósito de criar um produto final eficiente.

Como dito anteriormente, o uso da Universidade de Brasília como exemplo de aplicação do projeto serviu como uma prova de conceito, sendo importante identificar que o fato da tecnologia GPS possuir uma cobertura mundial possibilita a utilização do dispositivo em qualquer área georreferenciada no globo. Como o uso do dispositivo na UnB se tornou válido, outras áreas podem ser anexadas ao sistema fazendo com que o mesmo se torne mais amplo e se estenda para outros sistemas que utilizem dados de geolocalização.

Para trabalhos futuros deste projeto, pode ser considerado a criação de uma plataforma que possibilite o rastreamento do dispositivo e o armazenamento do trajeto feito pelo usuário, e apresente em uma página *web* em forma de mapas.

Como consequência deste rastreamento da localização geográfica do usuário, um sistema de acionamento de emergência pode ser implementado, possibilitando que o deficiente visual envie alguma espécie de mensagem de alerta contendo sua localização caso algum acidente ocorra.

Este projeto possui uma forte base para o desenvolvimento de um sistema maior e mais robusto para os seus usuários. No futuro poderá ser realizado o aperfeiçoamento do dispositivo para sistemas de navegação como já é utilizado em aplicações como o “Waze”, por exemplo, onde seus próprios usuários interagem com o sistema e entre si por meio de comunidade.

Também deve ser estudado o fato da evolução das tecnologias, um exemplo é a própria plataforma Raspberry Pi, que está em constante aperfeiçoamento, de modo que suas especificações técnicas estão melhorando ao passo dos anos desde a sua criação.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] WORLD HEALTH ORGANIZATION. Disponível em: <<http://www.who.int/blindness/en/>>. Acesso em: 18/06/2018.
- [2] MONTEIRO, J. L. Os desafios dos cegos nos espaços sociais: um olhar sobre a acessibilidade. Seminário de Pesquisa em Educação da Região Sul IX ANPED SUL, Caxias do Sul, p. 7. 2012. Disponível em: <<http://www.ucs.br/etc/conferencias/index.php/anpedsul/9anpedsul/paper/viewFile/1081/649>>. Acesso em: 18/06/2018.
- [3] EUROPEAN GLOBAL NAVIGATION SATELLITE SYSTEMS AGENCY. Disponível em: <<https://www.gsa.europa.eu/european-gnss/what-gnss>>. Acesso em: 18/06/2018.
- [4] MONICO, J. F.G. Posicionamento Pelo GNSS: descrição fundamentos e aplicações. 2ª ed. São Paulo: Editora UNESP, 2007.
- [5] THE RASPBERRY PI FOUNDATION. Disponível em: <<https://www.raspberrypi.org/>>. Acesso em: 18/06/2018.
- [6] U-BLOX. Datasheet: NEO-6 u-blox 6 GPS Modules. Disponível em: <https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdf>. Acesso em: 18/06/2018.
- [7] DEMERS, M. N. *Fundamentals of Geographic Information Systems*. 2ª ed. Hoboken: John Wiley & Sons, Inc, 2009.
- [8] PAREDES, E. A. Sistema de Informação Geográfica. São Paulo: Érica, 1994.
- [9] FEDERAL AVIATION ADMINISTRATION. Disponível em: <https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/gps/>. Acesso em: 18/06/2018.
- [10] CARDOSO, M. A. C. Tecnologias 3G: uma visão prática. Trabalho de Conclusão de Curso de Especialização (Curso de Especialização em Tecnologias, Gerência e Segurança de Redes de Computadores) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2008. [Orientador: Prof. Dr. Juergen Rochol]. Disponível em: <<http://www.lume.ufrgs.br/handle/10183/15978>>. Acesso em: 18/06/2018.
- [11] LENOVO GROUP LTD. Disponível em: <<https://www.lenovo.com/br/pt/faqs/notebook-faqs/linux/>>. Acesso em: 18/06/2018.
- [12] THE APACHE SOFTWARE FOUNDATION. Disponível em: <<https://httpd.apache.org/>>. Acesso em: 18/06/2018.
- [13] NETCRAFT LTD. Disponível em: <<https://news.netcraft.com/archives/2018/01/19/january-2018-web-server-survey.html>>. Acesso em: 18/06/2018.
- [14] MySQL. Disponível em: <<https://www.mysql.com/about/>>. Acesso em: 18/06/2018.
- [15] THE PHP GROUP. Disponível em: <http://php.net/manual/pt_BR/intro-what-is.php>. Acesso em: 18/06/2018.
- [16] phpMyAdmin. Disponível em: <<https://www.phpmyadmin.net/>>. Acesso em: 18/06/2018.

- [17] GOOGLE CLOUD TEXT TO SPEECH. Disponível em: <<https://cloud.google.com/text-to-speech/>>. Acesso em: 18/06/2018.
- [18] My Electronics Lab. Disponível em: <<https://myelectronicslab.com/raspberry-pi-3-gpio-model-b-block-pinout/>>. Acesso em: 18/06/2018.
- [19] PYTHON SOFTWARE FOUNDATION. Disponível em: <<https://www.python.org/doc/essays/blurb/>>. Acesso em: 07/09/2018.
- [20] Arduino Store. Disponível em: <<http://www.arduinstores.net/product/diymall-ublox-neo-6m-gps-module-with-eprom-for-mwcaeroquad-with-antenna-for-arduino-flight-control-aircraft/>> Acesso em: 18/06/2018.

APÊNDICE A

A1. COMANDOS PARA COMUNICAÇÃO RASPBERRY/MÓDULO

Para a capturar informações do módulo, uma série de comandos foram realizados no terminal do Ubuntu MATE.

O primeiro passo foi habilitar a entrada serial (UART) do Pi no arquivo “config.txt”. Para isso, os seguintes comandos foram utilizados:

```
sudo nano /boot/config.txt  
  
enable_uart=1
```

Depois, foi necessário a remoção da configuração que utiliza a interface serial0 do Raspberry.

```
sudo nano /boot/cmdline.txt
```

O conteúdo removido da linha foi:

```
console = serial0, 11520
```

Logo em seguida, o sistema foi reiniciado e foi verificada a comunicação do módulo com o Pi.

```
sudo cat /dev/serial0
```

Uma série de códigos no formato NMEA foram gerados na tela e atualizados a cada segundo, indicando que agora o Raspberry recebe os dados dos satélites GPS e estes dados podem ser tratados pelo sistema para a utilização em *softwares* de posicionamento.

Para uma melhor visualização inicial destes dados, a ferramenta GPSD foi instalada no sistema.

```
sudo apt-get install gpsd gpsd-clients python-gps  
  
sudo nano /etc/default/gpsd
```

As seguintes configurações foram implementadas:

```
START_DAEMON="true"  
  
GPSD_OPTIONS="-n"  
  
DEVICES="/dev/serial0"  
  
USBAUTO="true"  
  
GPSD_SOCKET="/var/run/gpsd.sock"
```

Para desativar qualquer serviço que ainda esteja utilizando a comunicação serial0, os seguintes comandos foram utilizados:

```
sudo systemctl stop serial-getty@serial0.service
sudo systemctl disable serial-getty@serial0.service
```

Logo após, o serviço GPSD foi iniciado.

```
sudo systemctl start gpsd.socket
sudo systemctl enable gpsd.socket
cgps -s
```

A2. COMANDOS PARA INSTALAÇÃO DA PILHA LAMP E phpMyAdmin

Para instalar a pilha LAMP no Linux Ubuntu MATE, os seguintes comandos foram realizados pelo terminal:

O primeiro comando a ser utilizado deve realizar uma atualização no repositório de pacotes do sistema operacional.

```
sudo apt-get update
```

Apache:

Para instalar o servidor Apache basta executar um comando.

```
sudo apt-get install apache2
```

A instalação irá criar um diretório html em /var/www/ onde representará o caminho de *localhost* no navegador. Este diretório pode armazenar arquivos html, php, etc.

Para um uso mais confiável do servidor Apache, alguns atributos como firewall devem ser configurados. Não houve necessidade de configuração destes atributos neste projeto em questão.

MySQL:

Com o apt-get novamente, o banco de dados MySQL foi instalado.

```
sudo apt-get install mysql-server
```

Durante a instalação foi pedida a criação de uma senha para o usuário “root” do servidor. Este usuário é conhecido como administrador pelo banco de dados, e possui uma maior gama de privilégios no sistema.

PHP:

O comando abaixo foi utilizado para instalar o PHP e suas dependências para rodar no sistema.

```
sudo apt-get install php php-mysql libapache2-mod-php
```

Com a finalidade de checar a instalação, o arquivo `info.php` foi criado em `/var/www/html` e acessado pelo navegador com o endereço “localhost/info.php”

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

phpMyAdmin:

Para uma melhor manutenção e gerência do banco de dados, o phpMyAdmin foi instalado seguindo alguns comandos.

```
sudo apt-get install phpmyadmin
```

Durante a instalação foi selecionado o servidor “Apache2” e quando perguntado sobre a utilização do “dbconfig-common” foi selecionada a opção “yes”.

Logo após foram utilizados alguns comandos para instalação de dependências e um *link* do diretório phpMyAdmin para o diretório do servidor Apache para que a ferramenta seja acessada pelo navegador no *localhost*.

```
sudo apt-get install php-mbstring php-gettext
sudo phpenmod mcrypt
sudo phpenmod mbstring
ln -s /usr/share/phpmyadmin /var/www/html
systemctl restart apache2
```

A3. INSTALAÇÃO DAS BIBLIOTECAS E CÓDIGO DO PROGRAMA *auxilio.py*

Para instalar as bibliotecas utilizadas pelo programa *auxilio.py*, os seguintes comandos foram executados a partir do terminal do Ubuntu:

```
sudo apt-get install python-gps
sudo apt-get install python-mysql
sudo apt-get install python-pip
sudo pip install --upgrade setuptools
sudo pip install gtts
sudo apt-get install mpg321
sudo apt-get install python-shapely
```

Código do programa *auxilio.py*:

```
import RPi.GPIO as GPIO
import os
import MySQLdb
from gps import *
from shapely.geometry import Point
from shapely.geometry.polygon import Polygon
from gtts import gTTS

global c
global db

flag = 0

#Iniciamos o gps
session = gps()
session.stream(WATCH_ENABLE|WATCH_NEWSTYLE)

#Iniciamos os pinos do Raspberry pi
GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.setup(16, GPIO.OUT)

#Conexao com Banco de Dados
try:
    db = MySQLdb.connect("localhost","root","mysqlpi","bd_coord")
    c = db.cursor()
except:
    print ("Servidor nao conectado!")

#Funcao que emite um pulso no LED
def registro():

    #Pulso apos gravar
    GPIO.output(16, GPIO.LOW)
    time.sleep(0.2)
    GPIO.output(16, GPIO.HIGH)

#Funcao da logica de calculo da area
def ponto_area(lat1, lon1, lat2, lat3, lon2, lon3):
```

```

    ponto = Point(lat1, lon1)
    area = Polygon([(lat3, lon2), (lat3, lon3), (lat2, lon3), (lat2, lon2)])
    return(area.contains(ponto))

#Funcao que emite a mensagem ao PNE
def fala(mensagem):

    AudioDoTexto = gTTS(mensagem, lang='pt')
    AudioDoTexto.save("/tmp/audio.mp3")
    os.system("mpg321 /tmp/audio.mp3")
    os.remove("/tmp/audio.mp3")

while True:
    input_state = GPIO.input(23)
    if input_state == False:
        print('Registrando Pontos')
        GPIO.output(16, GPIO.HIGH)
        report = session.next()
        if report.keys()[0] == 'epx' :
            lat = float(report['lat'])
            lon = float(report['lon'])

            registro()
            try:
                c.execute("SELECT ID, nome, lat_inf, lat_sup, lon_esq,
lon_dir, msg FROM areas")
                data = c.fetchall()
                for row in data :
                    ID = row[0]
                    nome = str(row[1])
                    lat_inf = float(row[2])
                    lat_sup = float(row[3])
                    lon_esq = float(row[4])
                    lon_dir = float(row[5])
                    msg = str(row[6])
                    dentro = ponto_area(lat, lon, lat_inf, lat_sup, lon_esq,
lon_dir)

                    if dentro == True and flag == 0:
                        fala(msg)
                        flag = 1
                    if dentro == False and flag == 1:
                        flag = 0
            except:
                print ("Erro ao ler Banco de Dados!")
                time.sleep(0.8)
        else:
            print('Nao Esta Registrando Nada')
            GPIO.output(16, GPIO.OUT)
            time.sleep(1)

```

