

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DAELN - DEPARTAMENTO ACADÊMICO DE ENGENHARIA ELETRÔNICA  
BACHARELADO EM ENGENHARIA ELETRÔNICA

GUILHERME ONO SENS, JOSNEI LUIS OLSZEWSKI JUNIOR

**CWDB: SISTEMA PARA DETECÇÃO DINÂMICA E ANÁLISE DE  
POLUIÇÃO SONORA**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA  
2018

GUILHERME ONO SENS, JOSNEI LUIS OLSZEWSKI JUNIOR

**CWDB: SISTEMA PARA DETECÇÃO DINÂMICA E ANÁLISE DE  
POLUIÇÃO SONORA**

Trabalho de Conclusão de Curso apresentado ao Bacharelado em Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná, como requisito parcial para a obtenção do título de Bacharel de Engenheiro Eletrônico.

Orientadora: Profa. Dra. Keiko Verônica Ono Fonseca  
Universidade Tecnológica Federal do Paraná

CURITIBA  
2018

GUILHERME ONO SENS  
JOSNEI LUIS OLSZEWSKI JUNIOR

## **CWDB: SISTEMA PARA DETECÇÃO DINÂMICA E ANÁLISE DE POLUIÇÃO SONORA.**

Este Trabalho de Conclusão de Curso de Graduação foi apresentado como requisito parcial para obtenção do título de Engenheiro Eletrônico, do curso de Engenharia Eletrônica do Departamento Acadêmico de Eletrônica (DAELN) outorgado pela Universidade Tecnológica Federal do Paraná (UTFPR). Os alunos foram arguidos pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Curitiba, 08 de fevereiro de 2018.

---

Prof. Dr. Robinson Vida Noronha  
Coordenador de Curso  
Engenharia Eletrônica

---

Prof<sup>a</sup>. Dr<sup>a</sup>. Carmen Caroline Rasera  
Responsável pelos Trabalhos de Conclusão de Curso  
de Engenharia Eletrônica do DAELN

### **BANCA EXAMINADORA**

---

Prof<sup>a</sup>. Dr<sup>a</sup>. Keiko Verônica Ono Fonseca  
Universidade Tecnológica Federal do Paraná  
Orientadora

---

Prof. Me. Daniel Rossato de Oliveira  
Universidade Tecnológica Federal do Paraná

---

Prof. Dr. João Luiz Rebelatto  
Universidade Tecnológica Federal do Paraná

---

Prof. Dr. Daniel Fernando Pigatto  
Universidade Tecnológica Federal do Paraná

A folha de aprovação assinada encontra-se na Coordenação do Curso de Engenharia Eletrônica.

À nossa coordenadora, professora Keiko Verônica Ono Fonseca que prontamente nos ajudou mesmo antes do início do projeto, sempre pronta para nos auxiliar. Sem ela, nada disso seria possível.

Nosso maior agradecimento é para Thalita Munique Costa, por todo o conhecimento, tempo e dedicação direcionados ao projeto. Sua contribuição não se mede em palavras ou resultados. É pouco dizer que sem ela não haveria esse projeto.

Ao Luiz Henrique Skodowski por todo o amor e apoio dispensado.

À Suellen Guimarães Mahfoud pelo apoio e amizade incondicional.

Às nossas famílias pela paciência e suporte que nos forneceram durante toda a duração não apenas deste trabalho, mas do curso que agora concluímos com orgulho.

Aos nossos amigos, novos e velhos, passageiros e eternos. Todos os que, de alguma forma, nos moldaram para que um dia nos tornássemos a melhor versão de nós que pudemos ser.

*"You only learn from experience, so as much as someone can tell you things, you have to go out there and make your own mistakes in order to learn". (WATSON, Emma, 2012).*

## RESUMO

SENS, Guilherme; OLSZEWSKI, Josnei. CWdB: Sistema para detecção dinâmica e análise de poluição sonora. 2018. Trabalho de Conclusão de Curso – Bacharelado em Engenharia Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

A poluição sonora é um dos grandes problemas a serem resolvidos pelas cidades inteligentes, as *smart cities*. Este trabalho busca uma solução para o mapeamento de dados sonoros que possa auxiliar no direcionamento de esforços e recursos, humanos e financeiros, para abordar o problema da poluição sonora. O projeto de engenharia aqui executado consiste em integração de *software on-line* aliado a um *hardware* (móvel ou fixo) e de baixo custo para medir o nível sonoro em diversos pontos de uma cidade e permitir o acesso à base dos dados coletados por aplicações voltadas para aumentar a inteligência da cidade. Uma prova de conceito de um sistema foi desenvolvido e resultados apresentados a aplicações de agregação e visualização de dados (na forma de mapas de calor). Procura-se discutir a eficiência de medição das diversas tecnologias utilizadas, para elucidar seu uso do começo ao fim, validando os requisitos iniciais de um projeto de baixo custo, *open source* através do qual pode-se enriquecer as relações entre seres humanos e ambiente urbano.

**Palavras-chave:** Cidades Inteligentes. Poluição Sonora. Detecção Dinâmica. Medição. Decibéis.

## ABSTRACT

SENS, Guilherme; OLSZEWSKI, Josnei. CWdB - Noise pollution dynamic detection and analysis system. 2018. Trabalho de Conclusão de Curso – Bacharelado em Engenharia Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

Noise pollution is one of the greatest issues to be solved by smart cities. This paper seeks a solution on mapping sonic data, which may help delegate effort and resources, either human or material, in order to solve the noise pollution problem. This engineering project consists in an on-line software and a low cost (either mobile or static) hardware capable of acquiring sonic data, in a vast number of points in cities and provide access of a data base to applications focused on making cities smarter. A proof of concept of a system was designed and its results were sent to data clustering and visualization applications (displaying the data in the form of a heat map). All sorts of technologies used try to discuss their measurement efficiency, as to elucidate their use from beginning to end, validating the initial requisites to be a low cost, open source project, on which people and urban environment could enrich their relations.

**Keywords:** Smart cities. Noise pollution. Dynamic detection. Measure. Decibels.

## LISTA DE FIGURAS

Figura 1 – Curvas "fon": curvas de mesma altura que dependem do nível sonoro e da frequência. . . . .	10
Figura 2 – Curvas de compensação (ponderação A, B, C e D) . . . . .	11
Figura 3 – Quadro <i>Kanban</i> . . . . .	15
Figura 4 – Diagrama de funcionamento para o hardware e software . . . . .	16
Figura 5 – Exemplo de utilização do sistema, considerando dois casos fixos (uma residência e um semáforo, por exemplo) e um caso móvel (com o uso de um veículo para transporte), incluindo o sinal GPS, envio para banco de dados e visualização ( <i>site</i> ) . . . . .	17
Figura 6 – Placa de desenvolvimento <i>Blue Pill</i> . . . . .	21
Figura 7 – Placa de desenvolvimento <i>Blue Pill</i> - configuração dos pinos ( <i>Pinout</i> ) . . . . .	21
Figura 8 – Princípio da transdução do microfone capacitivo. . . . .	23
Figura 9 – Esquemático do circuito de amplificação do sinal do microfone . . . . .	24
Figura 10 – Resposta em frequência do microfone KP-903 . . . . .	28
Figura 11 – Circuito de amplificação do sinal do microfone . . . . .	29
Figura 12 – Circuito de detecção de envoltória . . . . .	29
Figura 13 – Circuito de amplificador buffer . . . . .	30
Figura 14 – Circuito de amplificação com saturação do sinal para detecção da frequência . . . . .	31
Figura 15 – Comparação de desempenho entre ASP.NET e NODE.JS . . . . .	33
Figura 16 – <i>Dashboard</i> do Banco de Dados do projeto, visto pelo <i>add-on</i> Heroku Postgres . . . . .	34
Figura 17 – Parte do Banco de Dados do projeto, visto pelo <i>add-on</i> Adminium . . . . .	35
Figura 18 – <i>Dashboard</i> do projeto no Heroku . . . . .	36
Figura 19 – <i>Logs</i> do projeto no Heroku . . . . .	37
Figura 20 – Gráfico de resposta e curva de tendência . . . . .	39
Figura 21 – Composição do protótipo funcional . . . . .	40
Figura 22 – Organização das ligações entre circuitos do protótipo . . . . .	41
Figura 23 – Protótipo montado em um <i>case</i> plástico . . . . .	41
Figura 24 – Visão geral da tela de mapa . . . . .	42
Figura 25 – Detalhe para o mapa de calor ( <i>heatmap</i> ) com <i>markers</i> , representando os pontos encontrados no banco de dados para o intervalo de tempo definido . . . . .	42
Figura 26 – Detalhe para a caixa contendo as opções de filtro e ativação dos <i>markers</i> /mapa de calor nos dados vistos no mapa . . . . .	43
Figura 27 – Tela de análises, contendo os principais recursos para a seleção, filtragem e visualização dos dados contidos no banco . . . . .	43



Figura 28 – Detalhe para o mapa da tela de análises, contendo <i>markers</i> apenas na região escolhida pelo usuário, além de diferenciação entre valores dentro e fora da escala escolhida para o nível sonoro (dB) . . . . .	44
Figura 29 – Detalhe para o gráfico na tela de análises, o qual corresponde a uma visualização em linha do tempo da chegada dos dados, dentro dos filtros e região definidos, sem considerar no entanto o limiar de dB para referência. . . . .	44
Figura 30 – Detalhe para o campo que exibe a quantidade total de pontos recolhidos para a região selecionada, no período de tempo especificado, sem considerar o limiar de dB para referência. . . . .	45
Figura 31 – Detalhe para os campos que exibem as quantidades totais de pontos recolhidos para a região selecionada, no período de tempo especificado, considerando o limiar de dB para referência. . . . .	45
Figura 32 – Registro dos 5992 registros salvos em banco, com todos os dados completos. . . . .	45
Figura 33 – Exemplo de <i>log</i> de novo registro inválido sendo recebido pela API . . . . .	45
Figura 34 – Esquemático completo do protótipo . . . . .	57
Figura 35 – Esquemático do circuito de condicionamento do sinal do microfone . . . . .	58
Figura 36 – <i>Layout</i> do <i>Top Layer</i> . . . . .	59
Figura 37 – <i>Layout</i> do <i>Bottom Layer</i> . . . . .	59
Figura 38 – <i>Layout</i> do <i>Power plane</i> interno . . . . .	60
Figura 39 – <i>Layout</i> do <i>Ground plane</i> interno . . . . .	60
Figura 40 – Esquemático do módulo detector de som RobotDyn . . . . .	61
Figura 41 – Projeto mecânico do módulo detector de som Robotdyn . . . . .	62
Figura 42 – Princípio do microfone capacitivo pré-polarizado com eletreto . . . . .	64
Figura 43 – Estrutura de um microfone dinâmico . . . . .	64
Figura 44 – Resposta em frequência do módulo KY-037 . . . . .	66
Figura 45 – Placa de desenvolvimento Arduino Nano . . . . .	67
Figura 46 – Placa de desenvolvimento FRDM-KL25Z . . . . .	68

## LISTA DE ABREVIATURAS E SIGLAS

ADC	<i>Analog to Digital Converter</i>
AGPS	<i>Assisted Global Positioning System</i>
API	<i>Application Program Interface</i>
ARM	<i>Acorn RISC Machine</i>
CAN	<i>Controller Area Network</i>
CSS	<i>Cascading Style Sheets</i>
DAC	<i>Digital to Analog Converter</i>
DAELN	Departamento Acadêmico de Engenharia Eletrônica
EEPROM	<i>Electrically Erasable Programmable Read-Only Memory</i>
GLONASS	<i>Global Navigation Satellite System</i>
GPIO	<i>General Purpose Input/Output</i>
GPS	<i>Global Positioning System</i>
HTML	<i>HyperText Markup Language</i>
IDE	<i>Integrated Development Environment</i>
I/O	<i>Input/Output</i>
JSON	<i>JavaScript Object Notation</i>
OMS	<i>Organização Mundial da Saúde</i>
OSI	<i>Open Source Initiative</i>
P2P	<i>Peer to Peer</i>
PCB	<i>Printed Circuit Board</i>
PWM	<i>Pulse Width Modulation</i>
RAM	<i>Random Access Memory</i>
RF	<i>Radio Frequency/Rádio Frequência</i>
SQL	<i>Structured Query Language</i>

SPI	<i>Serial Peripheral Interface</i>
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i>
UART	<i>Universal Asynchronous Receiver/Transmitter</i>
USART	<i>Universal Synchronous/Asynchronous Receiver/Transmitter</i>
USB	<i>Universal Serial Bus</i>
Wi-fi	<i>Wireless Fidelity</i>

# SUMÁRIO

<b>1 – INTRODUÇÃO</b>	<b>1</b>
1.1 DELIMITAÇÃO DO TEMA	1
1.2 MOTIVAÇÃO	2
1.3 OBJETIVOS	3
1.3.1 Objetivo Geral	3
1.3.2 Objetivos Específicos	3
<b>2 – FUNDAMENTAÇÃO TEÓRICA</b>	<b>4</b>
2.1 IDENTIFICAÇÃO DO PROBLEMA	4
2.1.1 O Objetivo das Cidades Inteligentes	4
2.1.2 <i>Big Data</i>	5
2.1.3 Ruído e educação	5
2.1.4 Efeitos Fisiológicos do Ruído	6
2.1.5 Sono e Ruído	6
2.1.6 Ruído e Desempenho	6
2.1.7 Principais causas de Poluição Sonora nas cidades	7
2.1.8 O Mapeamento do Ruído Como uma Ferramenta Legal	7
2.2 TRABALHOS RELACIONADOS	8
2.3 FUNDAMENTAÇÃO TÉCNICA	8
2.3.1 Som	8
2.3.2 Poluição Sonora	8
2.3.3 Medição de Ruído Acústico	9
2.3.4 Ambiente de Desenvolvimento Integrado	12
2.3.5 Linguagem de Programação	12
2.3.6 Banco de Dados	13
2.3.7 Serviço Web	13
2.3.8 API	14
2.3.9 <i>Open-Source</i>	14
<b>3 – METODOLOGIA</b>	<b>15</b>
3.1 GESTÃO DE PROJETO	15
3.2 ANÁLISE DE REQUISITOS	15
3.3 <i>HARDWARE</i>	19
3.3.1 Placas de Desenvolvimento	20
3.3.1.1 <i>Blue Pill</i>	20
3.3.1.2 NodeMCU	20

3.3.2	Módulos . . . . .	21
3.3.2.1	Módulo GPS . . . . .	21
3.3.2.2	Módulos de Comunicação sem Fio . . . . .	22
3.3.2.3	Sensores de som . . . . .	22
3.3.3	Microfone . . . . .	23
3.3.4	Amplificador . . . . .	23
3.3.5	Detector de envoltória . . . . .	24
3.3.6	Detecção de frequência . . . . .	24
3.3.7	Compensação do Sinal . . . . .	25
3.4	<i>SOFTWARE</i> . . . . .	25
3.4.1	IDE/Linguagem de Programação . . . . .	25
3.4.2	Serviço <i>Web</i> . . . . .	26
3.4.3	<i>Front-end</i> . . . . .	26
3.4.4	Banco de Dados . . . . .	26
3.4.5	Versionamento . . . . .	26
<b>4 – CWDB - SISTEMA PARA DETECÇÃO DINÂMICA E ANÁLISE DE POLUIÇÃO SONORA . . . . .</b>		<b>27</b>
4.1	<i>HARDWARE</i> . . . . .	27
4.1.1	Microfone . . . . .	27
4.1.2	Condicionamento do sinal . . . . .	28
4.1.2.1	Amplificação . . . . .	28
4.1.2.2	Detector de envoltória . . . . .	28
4.1.2.3	Buffer . . . . .	30
4.1.2.4	Detecção de frequência . . . . .	30
4.1.3	Compensação do sinal . . . . .	31
4.2	<i>SOFTWARE</i> . . . . .	32
4.2.1	Ambiente de Desenvolvimento Integrado (IDE) e Linguagem de Programação . . . . .	32
4.2.2	<i>NODE.JS</i> . . . . .	32
4.2.3	Banco de Dados . . . . .	33
4.2.4	Página HTML . . . . .	34
4.2.5	Serviço <i>Web</i> . . . . .	35
<b>5 – ANÁLISE E DISCUSSÃO DOS RESULTADOS . . . . .</b>		<b>38</b>
5.1	CALIBRAÇÃO . . . . .	38
5.2	PROTÓTIPO . . . . .	38
5.3	IMAGENS DO <i>SOFTWARE</i> - ( <i>CWdB</i> ) . . . . .	42
5.4	ANÁLISE DE CUSTO . . . . .	45

<b>6 – CONCLUSÃO</b> . . . . .	<b>47</b>
6.1 CONSIDERAÇÕES FINAIS . . . . .	47
6.1.1 Comentários adicionais . . . . .	49
6.2 TRABALHOS FUTUROS . . . . .	50
<b>Referências</b> . . . . .	<b>51</b>

**Apêndices** **56**

<b>APÊNDICE A–Projeto do protótipo - (CWdB)</b> . . . . .	<b>57</b>
---	-----------

<b>APÊNDICE B–Documentação do módulo detector de som Robotdyn enviada pelo fabricante mediante solicitação</b> . . . . .	<b>61</b>
--	-----------

<b>APÊNDICE C–Estudo do Funcionamento dos Microfones</b> . . . . .	<b>63</b>
--	-----------

C.1 Microfone Capacitivo . . . . .	63
C.2 Microfone Dinâmico . . . . .	63
C.3 Microfone Piezoelétrico . . . . .	65

<b>APÊNDICE D–Tecnologias Alternativas Consideradas Para o Projeto</b> . . . . .	<b>66</b>
--	-----------

D.1 MICROFONES . . . . .	66
D.1.1 Módulo detector de som Robotdyn . . . . .	66
D.1.2 Módulo detector de som KY-037 . . . . .	66
D.1.3 Módulo de microfone MAX9814 . . . . .	67
D.2 PLACAS DE DESENVOLVIMENTO . . . . .	67
D.2.1 Arduino Nano . . . . .	67
D.2.2 FRDM-KL25Z . . . . .	68
D.2.3 GPS/GPRS AI-Thinker A7 . . . . .	69
D.3 COMUNICAÇÃO SEM FIO . . . . .	69
D.3.1 ESP8266 . . . . .	69
D.3.2 GPRS . . . . .	70
D.3.3 LoRa . . . . .	70
D.3.4 Zigbee . . . . .	70

# 1 INTRODUÇÃO

## 1.1 DELIMITAÇÃO DO TEMA

Muito se fala neste começo de século sobre a inteligência das coisas. Objetos, antes limitados a funções inanimadas, acabam por ganhar vida e agir quase por conta própria, ajudando toda uma sociedade a caminhar para um futuro mais eficiente, limpo e sustentável.

Nesse âmbito, entram as cidades inteligentes (em inglês *smart cities*). Ainda é complexo definir em um único termo o que torna uma cidade de fato inteligente, no entanto, como citado no artigo (HALL et al., 2000, p.2):

*"A city that monitors and integrates conditions of all of its critical infrastructures (...) can better optimize its resources, plan its preventive maintenance activities, and monitor security aspects while maximizing services to its citizens."*

Tradução adaptada: "Uma cidade que monitora e integra condições de todas as suas infraestruturas críticas (...) pode otimizar melhor seus recursos, planejar suas atividades de manutenção preventiva, e monitorar aspectos de segurança enquanto maximiza serviços para os seus cidadãos".

Trata-se de um espaço que utiliza de todos os meios possíveis para interagir consigo mesmo e com aqueles que o utilizam, de modo a melhorar qualidade de vida, o uso dos recursos e aplicar políticas em prol do bem comum.

Um dos grandes desafios é justamente resolver problemas determinados pela grande quantidade de pessoas em um ambiente cada vez mais populoso, como os centros urbanos. Ainda em (HALL et al., 2000, p.1), o autor cita que mais da metade das pessoas vive em centros urbanos, e este número deve continuar a crescer nas próximas décadas. Com uma grande migração da vida rural para a urbana, é de se esperar que grandes desafios sejam resolvidos de formas cada vez mais inteligentes e criativas. Problemas como acúmulo de lixo, congestionamentos e falta de moradia encontram solução em ações como a reciclagem, transporte direcionado e otimizado com a coleta de dados e casas otimizadas para o melhor aproveitamento do espaço.

Não se pode esquecer, no entanto, que grandes aglomerações podem trazer riscos à saúde. Poluição atmosférica pode trazer graves problemas respiratórios. Poluição visual pode causar stress e fadiga. Já a poluição sonora, pode trazer à tona problemas como a surdez com exposições a níveis sonoros mais altos (cerca de 85 dB ou mais por um período de exposição de 8 horas), interferência com a habilidade de fala, distúrbios de sono, cardiovasculares e cerebrais, implicações negativas em tarefas cotidianas como estudar ou trabalhar, além de interferência negativa em habilidades sociais simples (GOINES; HAGLER, 2012).

Combater este problema parece simples à primeira vista, mas é necessária cautela. Não basta apenas corrigir problemas superficiais, é preciso ir além. Investir em políticas públicas e

ações afirmativas no combate à proliferação do ruído, que pode facilmente se multiplicar junto ao crescimento demográfico urbano acelerado. É aqui que se justifica o mapeamento dos níveis sonoros por toda uma cidade, para que se possa atacar o problema onde é mais necessário, auxiliando grandes áreas afetadas de forma mais efetiva.

Organizações como a União Europeia colocam como prioridade o mapeamento em tempo real de áreas para análise desse efeito, incentivando soluções para minimizar a possibilidade de doenças e outros males na população local (ALSINA-PAGÉS et al., 2016). Medidas como estas, em muitos casos, são feitas de forma manual, com um operador se deslocando em pontos específicos da cidade portando um decibelímetro. O processo é lento, custoso e pouco eficiente, se for considerado a frequência e velocidade das medições (ALSINA-PAGÉS et al., 2016), surgindo então a demanda de um sistema que seja capaz de monitorar tantos locais quanto possível, identificando as áreas de maior potencial de risco.

A proposta deste trabalho é estudar e implementar uma rede de sensores de baixo custo para detecção dinâmica de ruídos e análise de poluição sonora, de modo a facilitar a coleta e acesso às informações relevantes. A ideia é encontrar e implementar uma solução de fácil montagem e instalação, que, quando acoplada a um meio móvel (como um ônibus ou similar, de circulação constante) ou fixada em um ambiente, poderá fornecer dados para alimentar um banco de dados de níveis sonoros, que poderá posteriormente auxiliar na identificação das regiões mais afetadas pela poluição sonora, ou mapear os níveis de ruído do local a ser estudado. Com isso, o acesso à informação será facilitado, levantando dados que ficaram em um acervo para posterior análise, contribuindo para alimentar bancos de dados mais complexos (como em sistemas *big data*), onde se pode analisar os efeitos e causas do ruído no meio ambiente.

## 1.2 MOTIVAÇÃO

A principal motivação para a realização do trabalho surgiu da necessidade de agir em prol do benefício coletivo, aliado à apresentação do trabalho de conclusão de curso. O tempo despendido na realização de tal tarefa seria bem empregado caso os autores alcançassem o objetivo de melhorar a qualidade de vida da sociedade, deixando um legado positivo do esforço no período despendido na UTFPR.

A poluição sonora tornou-se presente no cotidiano dos grandes centros, o que não significa que deva ser tolerada. Pelo contrário, deve ser combatida e de forma eficiente.

Motivou-se então encontrar a solução para a questão de onde e como agir para minimizar este problema e suas consequências danosas. Tantas são as soluções possíveis de se praticar, basta apenas escolher a forma e local correto de aplicação. Este é o ponto ao qual se gostaria de chegar, direcionando esforços às áreas com real problema, de forma a facilitar a solução onde mais se precisa. A ideia é fornecer dados para posterior análise, comprovando a eficácia do sistema como parte importante nas *smart cities*.



## 1.3 OBJETIVOS

### 1.3.1 Objetivo Geral

Desenvolver um sistema de *hardware* que, aliado a um *software*, possa fornecer dados da coleta móvel ou fixa do nível sonoro no ambiente que se deseja mapear, e que esses dados sejam acessíveis a qualquer cidadão.

### 1.3.2 Objetivos Específicos

Os objetivos específicos são:

1. Criar um hardware capaz de coletar dados sobre o nível sonoro do ambiente em períodos de tempo pré-determinados, registrando posição e horário do registro;
2. Pré-processar e transferir para um banco de dados remoto (*on-line*) via comunicação sem fio a posição e horário das medidas realizadas;
3. Criar um *software* com componente gráfica para visualização dos dados coletados;

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta justificativas para a realização do trabalho em questão. Também são apresentados fundamentos teóricos estudados necessários para atender o objetivo geral proposto.

### 2.1 IDENTIFICAÇÃO DO PROBLEMA

Esta seção discute a utilidade e importância de um mapeamento temporal dos níveis de ruído em locais específicos ou em locais variados pela cidade. Apresenta-se uma revisão da literatura sobre os objetivos das cidades inteligentes, o problema da poluição sonora nas cidades, os riscos que ela traz à população, a importância e a utilidade do seu mapeamento. O foco da revisão é a poluição sonora e como mapeá-la no contexto de cidades inteligentes.

#### 2.1.1 O Objetivo das Cidades Inteligentes

O termo "cidades inteligentes", também conhecido pela sua tradução em inglês "*smart cities*", se refere à crescente influência das tecnologias computacionais na vida urbana cotidiana. São organizações urbanas que buscam integrar as atividades de segurança pública, gestão de resíduos e da água, administração pública, saúde, educação, segurança, mobilidade, energia e habitação, entre outros. A gestão desses eixos está interligada por meio de ferramentas tecnológicas como a Internet, aplicativos de celulares, sensores, etc. O grande diferencial das *smart cities* é que elas não possuem um padrão de solução, elas observam as necessidades específicas de cada cidade e buscam soluções que resultam em ganhos sociais econômicos e de qualidade de vida (SANTAELLA, 2016).

Em 2015, o Brasil, juntamente com a Alemanha, assinou uma declaração afirmando a intenção de desenvolver uma cooperação para superar desafios comuns em relação a uma urbanização integrada e sustentável (LEAL, 2015). Desde então, o desenvolvimento de cidades inteligentes ganhou uma preocupação maior no país. Como verificado por COSTA, pode-se observar exemplos da evolução das cidades inteligentes, em vários lugares do mundo. Alguns deles são as cidades de Nova York, nos Estados Unidos, e Santander, na Espanha. A Universidade de Nova York inaugurou o Centro para Ciência e Progresso Urbano (*Center for Urban Science and Progress, CSUP*<sup>1</sup>), com o intuito de monitorar luz, temperatura, poluição e outros fatores ambientais através de sensores implantados por toda a cidade, também obtendo informações por meio de mídias sociais e câmeras. O projeto "Smartsantander"<sup>2</sup>, da cidade de Santander, consistiu na implantação de 18 mil sensores de diversos tipos espalhados pela cidade. Esses são

---

<sup>1</sup><http://cusp.nyu.edu/>

<sup>2</sup><http://www.smartsantander.eu/>

responsáveis por monitorar a poluição do ar, sonora e outros fatores ambientais, além de avisar quando o lixo deve ser recolhido, lâmpadas devem ser trocadas e outras funcionalidades.

Uma grande variedade de ferramentas tecnológicas vem sendo desenvolvidas com o objetivo de melhorar diversos serviços prestados à população. Para a eficiência desses processos, são fundamentais a captura, a gestão e a análise de dados (LEAL, 2015). Visto que o conceito de *smart cities* trata justamente de fazer a cidade se interligar à sua população de uma forma cada vez mais inteligente, torna-se essencial tratar dos problemas encontrados nos grandes centros hoje, nos quais os ruídos sonoros estão incluídos.

A cidade de Curitiba já se encontrou entre as 10 cidades consideradas como mais inteligentes do mundo pela Forbes em 2009 um resultado que se deu devido aos seus investimentos desde os anos 80 que vêm sendo copiados por cidades do mundo inteiro (MONZONI; NICOLLETTI, 2015). Um exemplo é o sistema *Bus Rapid Transit* (BRT) projetado para reduzir as emissões de dióxido de carbono, tempo de trajeto e tempo de embarque e desembarque de passageiros associados a um plano diretor de ocupação da terra<sup>3</sup>. A cidade está avançando cada vez mais nesse aspecto de ser uma cidade inteligente, criando conexões e acompanhamento em tempo real do sistema de transporte público, da rede pública de saúde, entre outros avanços tecnológicos (MONZONI; NICOLLETTI, 2015).

### 2.1.2 *Big Data*

O melhor ponto de partida para um planejador urbano é a análise de dados e o século atual nos trouxe, nesse ramo, o desafio da *Big Data*, um contexto onde um número imenso de informações é gerado e essas precisam ser corretamente processadas e verificadas para se tornarem realmente úteis. O acúmulo desse grande número de dados, mesmo que não sejam estruturados, pode aumentar significativamente a compreensão sobre problemas que as cidades enfrentam. Os dados coletados permitem antever situações adversas e então, estudar como evitá-las ou remediá-las antes que aconteçam (COSTA, 2015).

### 2.1.3 Ruído e educação

Um grupo que leva bastante atenção quando o assunto é poluição sonora, é o dos estudantes. Estudos comprovam que ruídos elevados prejudicam as funções cognitivas e podem ser negativamente relacionados ao bem-estar e a motivação (WHO, 2011). A capacidade de educar e atrair recursos humanos é um desafio muito importante para as cidades. Nesse sentido, a qualidade dos espaços, não é só um fator chave de competitividade como será onde os melhores talentos intelectuais e trabalhadores se desenvolverão (VERGARA, 2015). O fator humano é a matéria prima básica da economia atual, o perfil dos cidadãos impacta diretamente as funções, o perfil e o futuro das suas cidades.

---

<sup>3</sup><http://www.ippuc.org.br/>

#### 2.1.4 Efeitos Fisiológicos do Ruído

Altos níveis de ruído colocam o homem em um estado mental de alarme resultante da estimulação do sistema nervoso autônomo, uma reação compreendida já que em todo o reino animal, a audição tem a função principal de ser um sistema de alarme. Este estado traz alguns sintomas fisiológicos percebidos com o aumento dos níveis de ruído:

- Aumento da tensão muscular;
- Redução na velocidade de digestão;
- Aumento no metabolismo;
- Contração dos vasos sanguíneos presentes na pele;
- Aceleração na frequência cardíaca;
- Aumento na pressão sanguínea

(KROEMER; GRANDJEAN, 2005).

#### 2.1.5 Sono e Ruído

O sono é uma necessidade biológica, portanto, transtornos no sono podem causar uma série de problemas de saúde, desta forma, níveis altos de ruídos sonoros estão intimamente ligados a esses problemas, pois podem alterar o sono mesmo nos níveis mais baixos. Um leve aumento no nível indicado já resulta em alteração nos batimentos cardíacos, excitação e movimentação (PIMENTEL-SOUZA, 2000). Ao dormir o ser humano perde a percepção visual, a audição recebe o papel de principal sentido para alertar situações de perigo e o silêncio se torna necessário para o sono ocorrer na melhor qualidade. Próximo à 65dB(A), a função da audição é anulada em parte, trazendo insegurança e como consequência a dificuldade para pegar no sono (PIMENTEL-SOUZA, 2000).

Estudos mostram que o ruído noturno está ligado com sintomas de insônia e outros distúrbios do sono e com o uso de medicações, sintomas que levam à diversas doenças, como citado por PIMENTEL-SOUZA, cujo os estudos inspiraram também os próximos itens discutidos.

#### 2.1.6 Ruído e Desempenho

Percebe-se que a poluição sonora não tem tanta influência em trabalhos manuais, mas quando fala-se em trabalhos que exigem concentração mental, vê-se uma grande influência do nível sonoro do ambiente no desempenho das tarefas. Uma interessante visão sobre o assunto foi apresentada por PIMENTEL-SOUZA:

"Comprometem-se o raciocínio, a comunicação oral, a educação, o bem-estar e a sobrevivência, limitando as potencialidades humanas. O próprio desenvolvimento socioeconômico da sociedade é afetado pela incapacidade de compreender e reagir contra seus acusticamente poluídos meios urbanos, industriais e de lazer, piorados pela alta densidade populacional."

O autor ainda comenta que os países de primeiro mundo foram os primeiros afetados pela poluição sonora e essa percepção se deu pelo aumento no número de acidentes e na improdutividade. Os governos então perceberam que cuidar dos níveis de ruído significa garantir que sua população esteja mais preparada, intelectual e psicologicamente para competirem no mundo globalizado. Pode-se concluir isso também observando os efeitos do ruído no desempenho, os quais [KROEMER](#); [GRANDJEAN](#) sintetizou:

- "1. O ruído sempre interfere nas atividades mentais complexas, assim como nas atividades que exigem muita precisão e nas que exigem interpretação de informação.
2. O ruído pode tornar mais difícil o aprendizado de certos tipos de destreza.
3. Ruídos contínuos e altos níveis de ruído (em regra, acima de 90dB) podem atrapalhar o desempenho mental."

[KROEMER](#); [GRANDJEAN](#) ainda cita situações práticas, como o caso de um setor de digitação que teve seus erros de digitação reduzidos em 30% depois que o nível de ruído do ambiente foi reduzido em 25dB, e o caso de uma fábrica de automóveis que aumentou sua produção em 30% ao reduzir o ruído em 20dB. Mas o autor mesmo lembra que essas análises devem ser feitas com muito cuidado porque a alteração do ambiente de trabalho também pode ter causado um efeito psicológico nos trabalhadores, assim trazendo falsos resultados.

### 2.1.7 Principais causas de Poluição Sonora nas cidades

O tráfego de veículos nas ruas é a principal fonte de ruído nas cidades, podendo-se pontuar como elementos contribuintes os veículos a *Diesel* e veículos que tiveram seus silenciadores alterados para melhorar a sua potência. Nesse contexto, torna-se importante observar o tipo de asfalto utilizado, visto que as diferenças de ruído podem chegar a 11dB entre o asfalto liso e o concreto frizado, para pneus em mesmas condições ([NUNES, 1999](#)). As indústrias e a vizinhança, entram como outras duas causas principais de ruído sendo nas fábricas, os principais agentes os motores, máquinas e ar comprimido. [COSTA](#); [ARAÚJO](#); [BORGES](#) ainda trazem a questão dos cultos religiosos que podem trazer ruídos muito altos e seguem uma rotina, podendo ter essa propagação de ruídos por várias vezes ao dia.

### 2.1.8 O Mapeamento do Ruído Como uma Ferramenta Legal

Existem leis que regulam o barulho em todas as partes do mundo e quando se fala de prejuízos causados pela poluição sonora, do ponto de vista legal, a maior dificuldade consiste na medida. Deve-se ter informações objetivas sobre a intensidade, frequência e faixa espectral para que um juiz possa determinar sobre a proibição da produção de um certo barulho ou a obrigatoriedade do tratamento acústico, entre outras medidas. Deve-se levar em conta também o local, tendo em vista que diferentes níveis de ruído são permitidos para cada zona das cidades (fabris, mistas, etc) ([NEPOMUCENO, 1968](#)). Segundo [COSTA](#); [ARAÚJO](#); [BORGES](#), música alta ainda pode acarretar consequências jurídicas para a pessoa que a reproduz. Ela pode acabar

respondendo por crime ambiental (Lei nº 9.605/98), e pela Lei de Contravenções Penais (Lei 3.681/1941 e infrações administrativas nº 9.503/1997).

## 2.2 TRABALHOS RELACIONADOS

[ALSINA-PAGÉS et al.](#) apresentaram em seu artigo "*Design of a Mobile Low-Cost Sensor Network Using Urban Buses for Real-Time Ubiquitous Noise Monitoring*" (de tradução adaptada: Desenvolvimento de uma rede de sensores móveis e de baixo custo, usando ônibus urbanos para monitoramento em tempo real de ruído ubíquo) uma solução que serviu muito de base para a que foi implementado, já que se assemelham em motivação e objetivos.

## 2.3 FUNDAMENTAÇÃO TÉCNICA

### 2.3.1 Som

O que conhecemos como som são vibrações que se propagam em meios materiais no formato de uma onda, e são reconhecidos pelo ouvido humano ([COSTA; ARAÚJO; BORGES, 2014](#)). Para que uma vibração possa ser identificada pelo ouvido humano como som, ela deve estar compreendida em uma faixa de frequências, que é aproximadamente entre 20Hz e 20kHz.

O som se caracteriza por uma frequência e uma amplitude, conhecidas por tom e altura. A frequência de cada som influencia na percepção da sua altura pelas pessoas que o identificam, além disso, a curva sonora não é linear, assim, utiliza-se uma grandeza logarítmica para representar a intensidade do som, a qual chama-se de decibéis (dB) ([BALBINOT; BRUSAMARELLO, 2013](#)).

### 2.3.2 Poluição Sonora

É chamado de ruído o som indesejado que perturba o ambiente, por atrapalhar a fala, prejudicar a audição ou ser irritante de alguma outra forma. Quando os níveis de ruído se encontram acima de limites estabelecidos, tornam-se uma fonte causadora de diversos distúrbios, dentre os mais comuns ([BALBINOT; BRUSAMARELLO, 2013](#)) :

- Irregularidade no sono;
- Surdez permanente;
- Surdez temporária;
- Mascaramento da voz humana.

Há indícios ainda de que podem haver alterações na pressão sanguínea e secreção de hormônio catecolamínico devido aos níveis de ruído ([WHO, 2011](#)). Também foram encontradas evidências de influência na hipertensão, depressão e outras enfermidades mentais, além de causar fadiga, acidentes e redução da performance ([PIMENTEL-SOUZA, 2000](#)).

Por esses e outros motivos, a preocupação com a poluição sonora vem crescendo em diversos órgãos como a Organização Mundial da Saúde. A OMS recomenda menos de 30 dB(A) para os quartos para garantir uma boa qualidade de sono e menos de 35dB(A) nas salas de aula para garantir boas condições de ensino e aprendizado. Durante a noite, é recomendado ainda um máximo de 40 dB(A) fora dos quartos (WHO, 2011).

AGARWAL ainda completa que em áreas industriais, os níveis de ruído não podem ultrapassar 75dB(A) durante o dia e 70dB(A) durante a noite, áreas comerciais devem se manter abaixo dos 65dB(A) durante o dia e 55dB(A) durante a noite, áreas residenciais não podem exceder 55dB(A) durante o dia e 45dB(A) durante a noite e as chamadas áreas silenciosas, que são as regiões a menos de 100m de hospitais, escolas e tribunais, têm um limite de ruído máximo de 55dB(A) durante o dia e 45dB(A) durante a noite.

De acordo com a União Europeia, cerca de 40% da população dos países que a pertencem, sofrem com o ruído do tráfego acima de 55dB(A), 20% sofrem com um ruído de mais de 65dB(A) durante o dia e 30% com níveis que excedem 55dB(A) durante a noite (WHO, 2011).

Alguns grupos ainda são mais vulneráveis que outros. Crianças precisam de mais horas de sono do que adultos e por isso são mais vulneráveis a distúrbios no sono causados por ruídos. Idosos e pessoas doentes ainda são mais sensíveis aos distúrbios que podem ser causados. Pessoas que trabalham fora do horário comercial, também entram como mais afetadas por sua rotina de sono estressante (WHO, 2011).

### 2.3.3 Medição de Ruído Acústico

Para a medição de ruído acústico, o parâmetro mais utilizado a nível global é a pressão sonora ponderada, que representa a diferença entre a pressão instantânea e a pressão ambiente média em um certo ponto. Porém, a potência ou pressão sonora não são parâmetros que trazem boas representações práticas para o ruído, primeiramente porque a faixa de frequências da pressão sonora é muito maior do que a faixa audível pelo ser humano, além disso, a gama de sensibilidade da audição humana não apresenta comportamento linear, mas sim logarítmico, por conta disso, obtém-se os parâmetros que descrevem o som também por uma razão logarítmica, a qual denomina-se decibel (dB).

Quando se fala em som, é comum a utilização dos termos "tom" e "altura". A tonalidade se refere à frequência do som e a altura se refere a pressão que gera o som. Existe uma variação da percepção da altura em relação à frequência do som (audibilidade). Entre duas tonalidades diferentes mas de mesma potência, a que for mais aguda irá parecer mais alta ao ouvido humano, como podemos perceber nas curvas da (Figura 1) sendo Fom a unidade da definição de audibilidade (ISO 7-27a) (KROEMER; GRANDJEAN, 2005).

Os instrumentos utilizados para medir níveis sonoros, como o microfone e o analisador de espectro, expressam a audição em termos de parâmetros subjetivos que são determinados experimentalmente e que não correspondem exatamente à curva de resposta do ouvido humano.

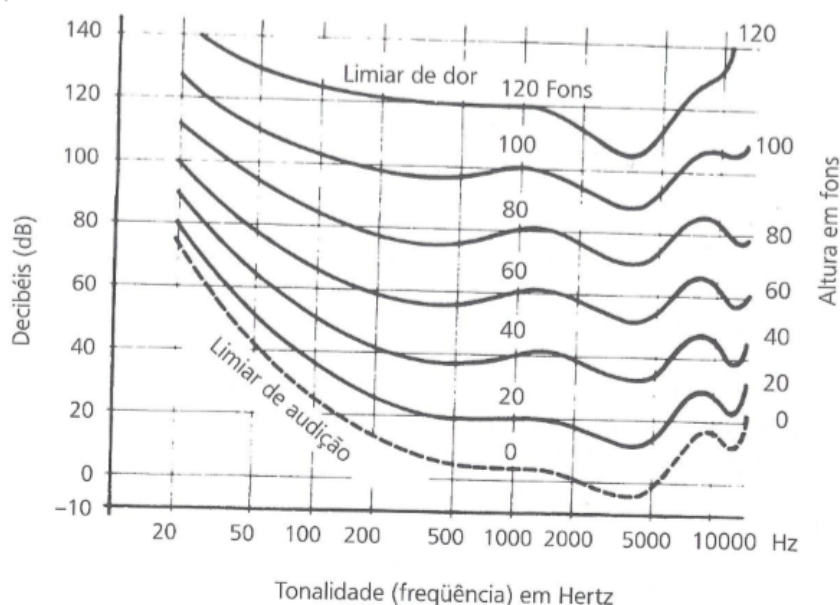


Figura 1 – Curvas "fon": curvas de mesma altura que dependem do nível sonoro e da frequência.

Fonte: KROEMER; GRANDJEAN

Como o homem só é capaz de identificar frequências entre 20Hz e 20kHz, requer-se a ajuda de escalas de ponderação e compensação em frequência, para que as medições se assemelhem às situações reais de desconforto (Figura 2). Essas curvas se encontram nos tipos "A", "B", "C" e "D". A curva "A" é a mais utilizada por se assimilar a curvas de análise de desconforto e risco de lesões auditivas em função do ruído sonoro, e por ser a mais adequada para ruídos contínuos e intermitentes (BALBINOT; BRUSAMARELLO, 2013). Segundo BERECHNUNG..., as fórmulas corretas para a geração das curvas "A", "B" e "C" como vistas na Figura 2 podem ser utilizadas na forma:

$$R_a(f) = \frac{12200^2 * f^4}{(f^2 + 20.6^2) * (f^2 + 12200^2) * \sqrt{f^2 + 107.7^2} * \sqrt{f^2 + 737.9^2}} \quad (1)$$

$$R_b(f) = \frac{12200^2 * f^3}{(f^2 + 20.6^2) * (f^2 + 12200^2) * \sqrt{f^2 + 158.5^2}} \quad (2)$$

$$R_c(f) = \frac{12200^2 * f^2}{(f^2 + 20.6^2) * (f^2 + 12200^2)} \quad (3)$$

$$A = 20 * \log_{10}(R_a(f)) + 2.00dB \quad (4)$$

$$B = 20 * \log_{10}(R_b(f)) + 0.17dB \quad (5)$$

$$C = 20 * \log_{10}(R_c(f)) + 0.06dB \quad (6)$$



Onde:

$f$  = frequência

$R_a$  = valor de referência para a curva A em uma determinada frequência

$R_b$  = valor de referência para a curva B em uma determinada frequência

$R_c$  = valor de referência para a curva C em uma determinada frequência

A = valor de compensação da curva A em uma determinada frequência para um sinal sonoro medido (em dB)

B = valor de compensação da curva B em uma determinada frequência para um sinal sonoro medido (em dB)

C = valor de compensação da curva C em uma determinada frequência para um sinal sonoro medido (em dB)

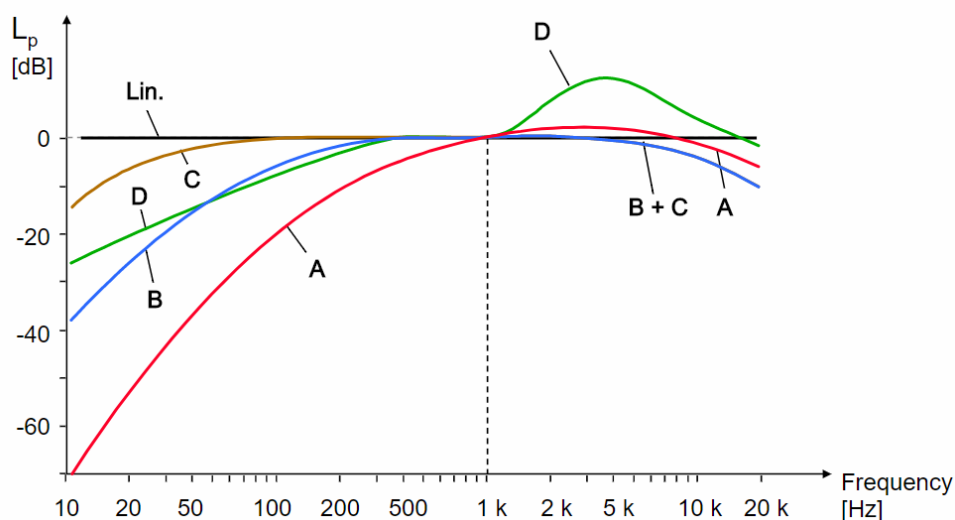


Figura 2 – Curvas de compensação (ponderação A, B, C e D)

Fonte: [BALBINOT; BRUSAMARELLO](#)

Usando as equações 1, 2 e 3, pode-se calcular os valores referência para usar em 4, 5 e 6, respectivamente. São esses valores que complementam o ganho adicional que deverá ser somado a resposta do microfone em uma determinada frequência, aproximando-se do resultado esperado como mostrado na [Figura 2](#).

Quando a medição é feita em ambientes externos, é preferível que o microfone do medidor se encontre a aproximadamente 1,20m do solo, que seja posicionado voltado para a fonte de ruído, que esteja, no mínimo, a 1,20m de distância de obstáculos e haja proteção contra o vento. Em ambientes internos, ainda deve-se considerar a natureza e condições do ambiente, como por exemplo verificar se as janelas se encontram abertas ou fechadas ([BALBINOT; BRUSAMARELLO, 2013](#)).

### 2.3.4 Ambiente de Desenvolvimento Integrado

Um Ambiente de Desenvolvimento (em inglês, *Integrated Development Environment* ou apenas *IDE*) pode ser definido como:

*"An IDE (integrated development environment) is used to write code, test for errors and translate a program."*(BBC, 2018).

Tradução adaptada: Uma IDE (ambiente de desenvolvimento integrado) é usado para escrever código, procurar por erros e traduzir um programa.

Ou seja, um recurso utilizado para escrever, testar/procurar por erros e traduzir um programa (em um conjunto de instruções legíveis pelo ambiente que irá executá-las), em tradução livre. É um conjunto de ferramentas integrado, geralmente acompanhado de uma *interface* amigável para o desenvolvedor, contendo diversos recursos úteis para diversas etapas do desenvolvimento, sejam elas: criação e execução do projeto, correção de erros de sintaxe, auto completar para comandos básicos/comuns, além de ajudar no processo de busca por erros, também conhecido como *debug*.

Alguns exemplos:

- *Microsoft Visual Studio*<sup>®</sup> (MICROSOFT, 2017)
- *Android Studio*<sup>®</sup> (GOOGLE, 2018)
- *XCode*<sup>®</sup> (APPLE, 2018)

Alguns editores de código representam o papel de *IDE* em projetos que não demandam compilação, como é o caso das linguagens *web* (HTML5, CSS e JavaScript, por exemplo). São exemplos de editores:

- *Microsoft Visual Studio Code*<sup>®</sup> (MICROSOFT, 2018),
- *Sublime Text*<sup>®</sup> (SUBLIMETEXT, 2017)
- *Atom*<sup>®</sup> (ATOM, 2017)

### 2.3.5 Linguagem de Programação

Linguagens de programação podem ser definidas como:

*"(...) an algorithm is any well-defined computational procedure that takes some value, or set of values, as input and produces some value, or set of values, as output. An algorithm is thus a sequence of computational steps that transform the input into the output. We can also view an algorithm as a tool for solving a well-specified computational problem".* (CORMEN CHARLES E. LEISERSON, 2009)

Tradução adaptada: (...) um algoritmo é um procedimento computacional bem definido que recebe um ou mais valores como entrada, produzindo um ou mais valores de saída. Um algoritmo é, portanto, uma sequência de passos computacionais que transformam a entrada em saída. Nós podemos ver um algoritmo como ferramenta para resolver um problema computacional bem especificado.

Ou seja, linguagens de programação são as responsáveis por efetivamente implementar instruções que serão realizadas pela máquina, completando uma tarefa específica.

Existem diversos níveis de linguagem, desde os mais baixos, diretamente ligados ao hardware em si, até os mais altos, que se aproximam ao nível de pensamento humano, com grande abstração de comandos.

Cada linguagem apresenta um foco diferente, com otimizações e aprimoramentos em relação a suas semelhantes. Dessa forma, cada uma possui um uso específico para o qual é recomendada, de acordo com suas especificidades e limitações.

São exemplos de linguagem de programação:

- *Java*;
- *C++*;
- *ASP.NET (C#)*;
- *Python*;
- *Lua*;
- *F#*;
- *PHP*;
- *Swift*;
- *Objective-C*.

### 2.3.6 Banco de Dados

Um banco de dados é uma estrutura maleável, podendo conter diversas formas de se armazenar informações, onde cada registro representa um ponto de um determinado conjunto de dados, podendo conter um grande número de entradas.

Muitas opções estão disponíveis no mercado. Citando algumas: *MSSQL*<sup>4</sup>, *MySQL*<sup>5</sup>, *PostgreSQL*<sup>6</sup>, *Cassandra*<sup>7</sup>, *MongoDB*<sup>8</sup>, entre muitas outras. A ideia é sempre a mesma: guardar e pesquisar pelos dados. A diferença fica na forma como os dados são salvos e principalmente referenciados entre si. Alguns bancos, como os *SQL*, permitem o referenciamento de várias tabelas entre si, enquanto os outros tratam os dados de forma mais otimizada, como é o caso do *Cassandra*, que é altamente recomendado para usos em *Big Data*, por tratar grandes quantidades de dados de forma facilitada.

### 2.3.7 Serviço Web

Toda aplicação *web* precisa de um serviço de hospedagem na nuvem (popularmente conhecida por *cloud*). É este serviço quem fornece opções de hospedagem e publicação para o *site*, além de diversas ferramentas extras como configuração de banco de dados, *analytics*, monitoramento de conexões e erros, gerenciamento de servidores, etc.

---

<sup>4</sup><https://www.microsoft.com/en-us/sql-server/sql-server-2016>

<sup>5</sup><https://www.mysql.com/>

<sup>6</sup><https://www.postgresql.org/>

<sup>7</sup><http://cassandra.apache.org/>

<sup>8</sup><https://www.mongodb.com/>

Muitas são as opções disponíveis no mercado para tal tarefa. Algumas delas estão citadas a seguir:

- *Heroku*<sup>®</sup> ([HEROKU, 2018](#)), gratuito/pago;
- *Zeit.co*<sup>®</sup> ([ZEIT.CO, 2018](#)), gratuito/pago;
- *Azure*<sup>®</sup> ([AZURE, 2018](#)), gratuito por tempo limitado;
- *Amazon Web Services (AWS)*<sup>®</sup> ([AMAZON, 2018](#)), gratuito por tempo limitado;
- *Nanobox*<sup>®</sup> ([NANOBOX, 2018](#)), pago.

### 2.3.8 API

A Interface de Programação de Aplicação, ou em inglês *Application Programming Interface* (API), define um meio de acesso à determinada informação por uma aplicação.

Segundo a pesquisa realizada por Alexandre Gouveia: "API é uma ferramenta que realiza comunicação entre aplicações que desejam compartilhar suas rotinas, ferramentas, padrões e protocolos"(GOUVEIA, 2016).

Basicamente, uma API implementa uma interface entre duas aplicações, sem que essas necessariamente compartilhem qualquer tipo de código proprietário entre si. Uma API para solicitação da previsão do tempo, por exemplo, teria um endereço para acesso que esperaria um ou mais parâmetros para identificar o local de consulta. Ao receber um pedido, esta processaria as informações internas necessárias, retornando apenas o resultado final da solicitação. A aplicação requisitante trata o retorno da melhor forma.

Trata-se então de uma pergunta e resposta, sem a necessidade de interação no processamento intermediário. Novamente de acordo com Gouveia:

"API, portanto, é uma *interface software-to-software*, isso quer dizer que o usuário não vê o processo de pedido ou de resposta. Tudo que o usuário vê é uma integração entre sistemas diferentes, de modo que ele consegue realizar várias ações em um único lugar."(GOUVEIA, 2016)

### 2.3.9 Open-Source

O termo *open-source*, criado pela OSI (*Open Source Initiative*), significa em português "Código aberto" e é um movimento que defende o código-fonte aberto e livre ao desenvolvimento. É comum encontrar essa expressão acompanhada de "*software livre*" tendo em vista que as diferenças entre esses dois movimentos são mínimas. Códigos *open-source* trazem como vantagens a possibilidade de serem compartilhados, reparados pelos seus usuários, entre vários benefícios econômicos (LLC et al., 2001-2017). No contexto das *smart cities* os *open databases* (bancos de dados abertos) ainda entram como uma ferramenta de importância, e por isso é cada vez mais comum ver sua aplicação em cidades de vários países, como já é o caso do Brasil com o "portal da transparência"(GRAFT; VERHULST; YOUNG, 2016).

### 3 METODOLOGIA

Nesta seção, será discutida a metodologia de execução do projeto, incluindo as formas de gestão da divisão e controle de tarefas, análise dos requisitos iniciais de projeto e suas possíveis soluções de aplicação, além de descrever as características das tecnologias escolhidas.

#### 3.1 GESTÃO DE PROJETO

Para controle do desenvolvimento do projeto, foram utilizadas as metodologias SCRUM e *Kanban*, com reuniões diárias e rápidas utilizando de *sprints* que facilitaram a seleção e motivação de próximas tarefas a serem executadas. Utilizou-se um quadro Kanban [Figura 3](#) para facilitar a visualização de todas as tarefas que haviam sido concluídas, as que estavam em progresso e as que ainda precisavam ser feitas, o que facilitou a organização e a visualização do avanço do projeto.

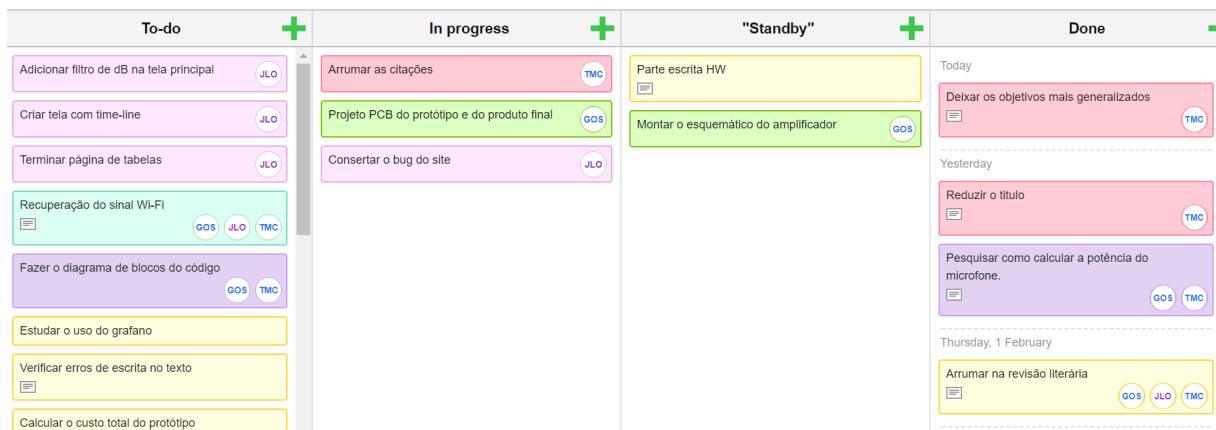


Figura 3 – Quadro *Kanban*

Fonte: Autoria Própria

Os passos para a execução do projeto foram: estudo de sistemas de medição de ruído sonoro, estudo dos componentes básicos destes sistemas, análise de requisitos do sistema proposto, estudo de tecnologias necessárias para atender os requisitos do sistema proposto a partir de integração de componentes, projeto de integração, implementação, testes, análise de resultados, validação. As seções seguintes detalham os componentes e discutem os critérios para a sua escolha.

#### 3.2 ANÁLISE DE REQUISITOS

O problema a ser resolvido consiste em mapear, em tantos pontos quanto for possível, a poluição sonora no ambiente. Para isso, o *hardware* construído leva em consideração a coleta

de dados em posições fixas, que podem ser instaladas em construções e equipamentos urbanos estáticos tais como postes, pontos de ônibus e totens, bem como através de sensores em movimento, como em um veículo, por exemplo. Os dados devem ser armazenados e ficarem acessíveis para processamento e apresentação de resultados de análise. Isso pode ser melhor analisado na [Figura 4](#), onde o sistema é mostrado na forma de um diagrama. No caso específico deste projeto a prova de conceito desenvolvida foca em apresentar combinações de medições geolocalizadas associadas a uma informação de tempo (data, horário), além do próprio valor (em dB(A)) da medição sonora realizada.

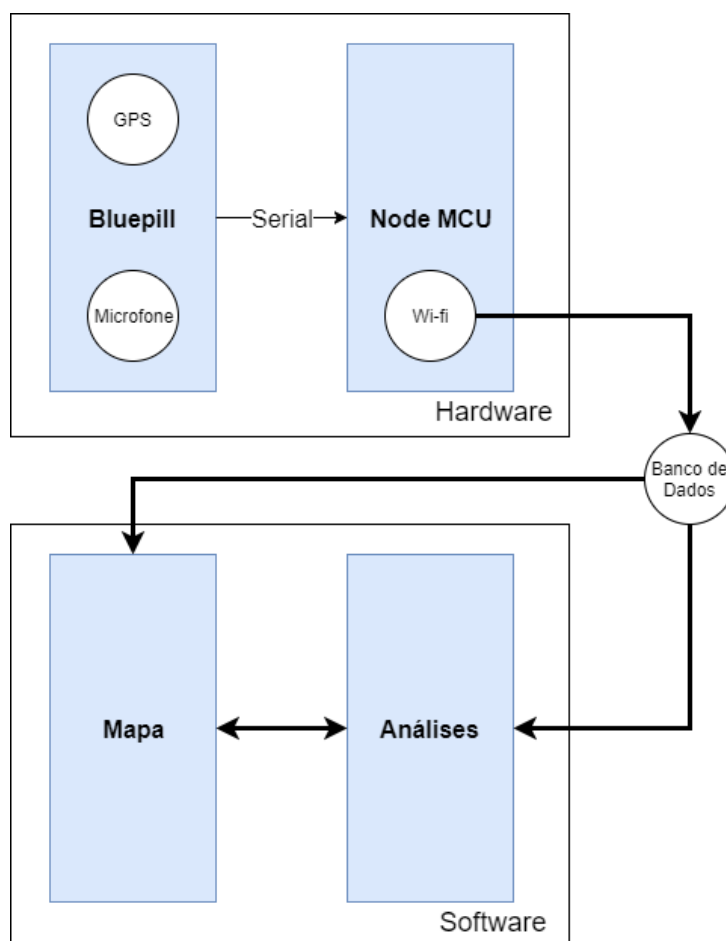


Figura 4 – Diagrama de funcionamento para o hardware e software

Fonte: Autoria própria

Para o cumprimento do objetivo de registrar o local, a data e a hora das medições realizadas, foi necessário avaliar a incorporação de um sistema de posicionamento para aquisição de data, hora, latitude e longitude da medição realizada. As tecnologias possíveis de posicionamento são: GPS (Sistema de posicionamento por satélite desenvolvido pelos Estados Unidos), Beidou (China), o GLONASS (Rússia) e o Galileo (Comunidade Europeia). O registro da posição é importante para posterior visualização do local de coleta dos dados.

O diagrama apresentado na [Figura 5](#) apresenta componentes de um sistema que popula

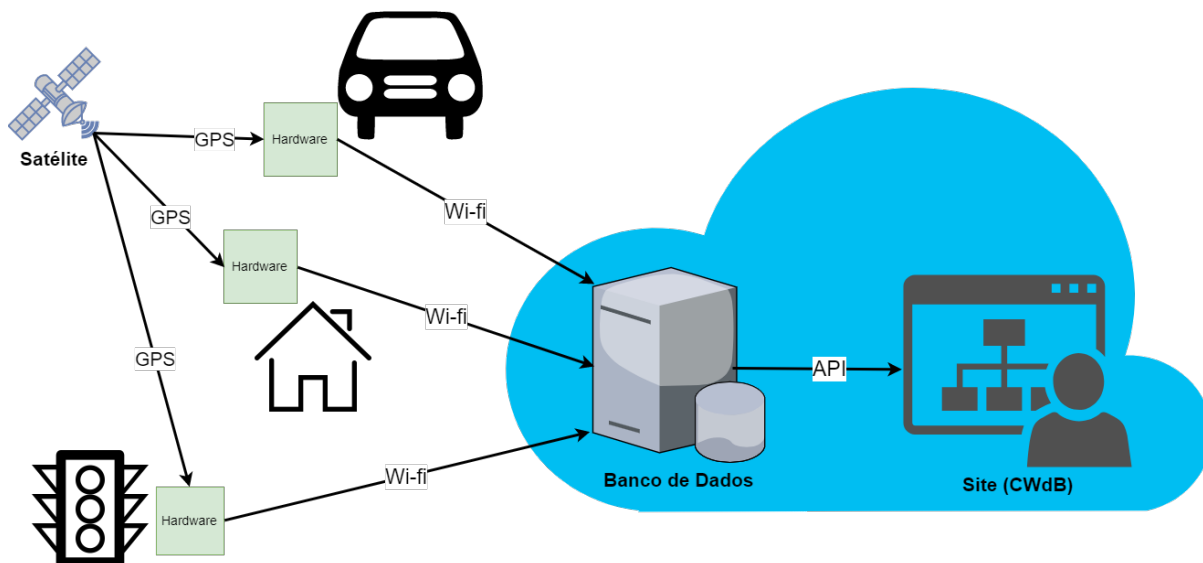


Figura 5 – Exemplo de utilização do sistema, considerando dois casos fixos (uma residência e um semáforo, por exemplo) e um caso móvel (com o uso de um veículo para transporte), incluindo o sinal GPS, envio para banco de dados e visualização (*site*)

Fonte: Autoria própria

um banco de dados sobre ruído através de coleta de nível de ruído audível ao ser humano através de sensores em pontos fixos (semáforo e residência) e através de um sensor móvel (automóvel) cuja geolocalização é obtida através de GPS. Ilustra-se o envio de dados dos sensores a um servidor de banco de dados através de tecnologia sem fio (*Wi-Fi*). O acesso ao servidor de banco de dados se faz via uma API padronizada que disponibiliza aplicações de processamento, análise e visualização dos dados coletados (como clusterização dos dados coletados por área e horário, geração de uma linha do tempo com níveis sonoros, classificação dos dados por intensidade e visualização de pontos coletados em um mapa de calor).

Para contemplar ambos os cenários (sensor fixo e sensor móvel), faz-se necessário um hardware simples, compacto e que seja capaz de se adaptar em diversas situações de conectividade. Como esta pode ou não estar presente a todo o tempo, um *buffer* de memória é essencial para evitar a perda de dados na falta de um *link* direto à *internet*. Uma solução é utilizar duas placas, uma exclusivamente para a aquisição dos dados e outra que monta e envia os pacotes, justificando a divisão entre os dois embarcados, e aumentando a disponibilidade de memória.

A conexão para transmissão de dados deve aproveitar o que já existe de infraestrutura no local e também o que já existe disponível na configuração de hardware escolhida. Neste sentido, a tecnologia *Wi-Fi* se destaca como uma opção para o projeto, evitando uma aquisição de novos componentes, custos fixos de serviços de telefonia para transmissão de dados e aproveitando uma ampla rede em banda não licenciada e de pontos de acesso (pontos de coleta) disponível para uso sem custos adicionais.

Um projeto de integração de tecnologias pode atender os requisitos de fácil uso do sistema pelo usuário final que não deve necessariamente possuir um amplo conhecimento das ferramentas de software. Isso deve principalmente ser alcançado pelo *endpoint* de visualização, que deve cumprir o requisito de ser simples, claro e direto, com capacidade de filtragem e amostragem dos pontos recolhidos. Na [Figura 5](#) o *endpoint* é representado pelo *site* (CWdB).

É esperado uma grande quantidade de pontos, com o intuito de mapear uma área de tamanho variável. Quanto mais pontos, maior deverá ser a capacidade de escalabilidade do banco de dados escolhido. Isso está diretamente ligado ao tema do *Big Data* tratado em sua complexidade de aquisição e tratamento de grandes quantias de dados. No entanto, evitando a sobrecarga de dados e de aquisição tanto no banco quanto em casos de armazenamento interno do *hardware*, as medições podem ser realizadas a uma taxa fixa de 10 segundos, garantindo um bom mapeamento de regiões para o caso de uso móvel.

O microfone é um componente essencial do projeto. É importante que a resposta dele consiga capturar valores nas faixas audíveis aos seres humanos. Deve-se levar em conta às compensações necessárias para se atingir a melhor resposta, similar àquela que de fato representa o ruído perceptível e prejudicial do ambiente. É importante ainda que o sistema tenha uma quantidade de memória disponível para armazenar os dados caso a conexão, principalmente de sensores móveis, com o servidor de banco de dados seja perdida. A transmissão dos dados armazenados se dará por oportunidade, ou seja, assim que uma conexão for reestabelecida e novos dados estiverem aguardando transmissão. O tratamento da perda de sinal pode ser feito com uma verificação constante da conexão ou adotando alguma estratégia de redução de consumo de energia que atenda os critérios de desempenho estabelecidos a priori.

Por fim, para conhecer a posição dinâmica do sensor que coleta o dado de interesse é possível usar a tecnologia de georeferenciamento a partir de dados de GPS. Por uma questão de disponibilidade de componentes no mercado, o uso de módulos GPS é uma forma conveniente de adquirir e gravar o local bem como o *timestamp* de onde e quando é feita cada medição. Essa solução também permite fornecer mais dados sobre as condições a qual a medição foi realizada, como: velocidade a que o sensor se move, direcionalidade (referência cardinal à qual o sensor aponta), quantidade de satélites utilizados na triangulação do posicionamento (validade a qualidade do dado de posição obtido) e altitude relativa ao nível do mar.

Nas seções seguintes, são discutidos os requisitos de *hardware* e *software* de um sistema similar ao representado na [Figura 5](#) e também algumas das soluções existentes consideradas para a integração das tecnologias. Dentre tais soluções, placa de desenvolvimento e funcionalidades disponíveis ([Subseção 3.3.1](#)), sistema de posicionamento e localização ([??](#)), solução de comunicação sem fio ([Subsubseção 3.3.2.2](#)) e sensores de som ([Subsubseção 3.3.2.3](#)) compõe o *hardware*; ambiente de desenvolvimento e linguagem de programação (compreendidos na [Subseção 2.3.4](#) e [Subseção 2.3.5](#), respectivamente), banco de dados ([Subseção 2.3.6](#)), serviço Web ([Subseção 2.3.7](#)) e API ([Subseção 2.3.8](#)) compreendendo soluções para o *software*. Também são discutidas no [Apêndice D](#), as tecnologias alternativas às consideradas no projeto,



mas não implementadas.

### 3.3 HARDWARE

Para que o objetivo do projeto fosse cumprido, havia a necessidade do projeto de um sistema de medição confiável e de baixo custo. Buscando a otimização da razão custo-benefício, foi escolhida a placa de desenvolvimento "*BluePill*" que conta com a quantidade de memória e processamento necessárias ao projeto, além de custar 1/10 do valor da FRDM KL25Z e um preço equivalente ao do Arduino Nano. A escolha desta placa ao invés do Arduino Nano, se deu por diversos motivos. A *Bluepill* conta com um maior número de portas seriais, um ADC de melhor resolução e maior quantidade de portas de entrada e saída. A placa ainda conta com uma arquitetura ARM Cortex M3 que permite trabalhar com um sistema operacional, o que será crucial se o projeto for ampliado e mais sensores forem acoplados. Além disso, a *Blue Pill* conta com mais memórias RAM e *Flash* e um processador de 32 bits, contra 8 bits do *Arduino Nano*.

A equipe decidiu que o armazenamento dos dados seria feito utilizando um banco de dados, então surgiu a necessidade de uma comunicação com o banco escolhido para enviar esses dados, a qual decidimos fazer por *wi-fi* utilizando o protocolo TCP/IP, pela praticidade que essa tecnologia oferece. Adequando-se à essa necessidade de comunicação com a nuvem, o módulo NodeMCU versão 1.0 foi escolhido como a solução mais viável para realização da conexão entre *hardware* e servidor, além do armazenamento dos dados obtidos e processados. Essa escolha se deu devido a testes práticos que mostraram que o ESP8266 não era adequado ao projeto pela sua quantidade de memória e tecnologia defasada em relação às concorrentes estudadas que dificultava a comunicação com a rede em comparação ao nodeMCU.

Os sistemas de medição são compostos, no geral, por 3 elementos principais: o transdutor que corresponde a um elemento de detecção que transforma a quantidade identificada em um sinal elétrico, o condicionador de sinais que transforma o sinal de uma maneira que ele possa ser melhor identificado e um mostrador ou elemento de registro, onde o resultado da medição é acessado (BOLTON, 2002).

O hardware representado na Figura 4 deve ser desenvolvido para atender os objetivos específicos 1 e 2, ou seja, ele deve ser capaz de obter medidas de ruído sonoro em períodos de tempo pré-determinados e o local, data e hora da realização de cada medida. Verificou-se então a necessidade de utilizar uma placa de desenvolvimento, microfone como sensor, módulos GPS e de tecnologia de comunicação sem fio, entre outros componentes. Realizou-se um estudo de alguns modelos de cada componente requerido para que fosse feita uma escolha adequada para atender os requisitos especificados.

### 3.3.1 Placas de Desenvolvimento

O projeto desenvolvido requer uma placa de desenvolvimento, que contenha processador e memória adequados para coordenar e processar os dados obtidos. No anexo [Apêndice D](#) estão apresentadas algumas opções de placa de desenvolvimento que poderiam ser adequadas para o projeto em questão, uma análise que resultou na escolha da *Blue Pill*, como explicado melhor no [Seção 3.1](#).

#### 3.3.1.1 *Blue Pill*

A placa de desenvolvimento *Blue Pill* possui um microcontrolador STM32F103C8T6 e entre seus pontos positivos está o seu baixo custo, visto que seu custo é cerca de dez vezes menor que o de um Arduino Uno oficial e tem como especificações principais ([JEELABS.ORG](http://JEELABS.ORG))

:

- Processador ARM de 32 bits;
- Arquitetura ARM Cortex M3;
- 44 pinos arranjados na placa como mostra a [Figura 7](#);
- 64KB de memória *flash*;
- 20KB de memória RAM;
- 32bits e 72MHz de *clock* interno;
- Faixa de temperatura de -40 à +85 °C;
- USART (3);
- ADC (12bits) 10 canais (1);
- SPI (2);
- I2C (2);
- CAN BUS (2);
- *Timers* (4);
- PWM (15);
- Pinos de programação/*debug*;
- GPIO (33);

#### 3.3.1.2 NodeMCU

Para fornecer ao protótipo a possibilidade de se conectar na rede, foi incluído um NodeMCU, um kit de desenvolvimento *open-source* que contempla entre seus recursos a conexão *wi-fi*. Além disso, oferece ([NODEMCU, 2018a](#)):

- *USB-TTL plug& play*;
- 10 GPIO com suporte PWM, I2C e 1-wire;
- Módulo *Wi-Fi* com certificação FCC e antena PCB;
- API de rede no estilo NodeJS.

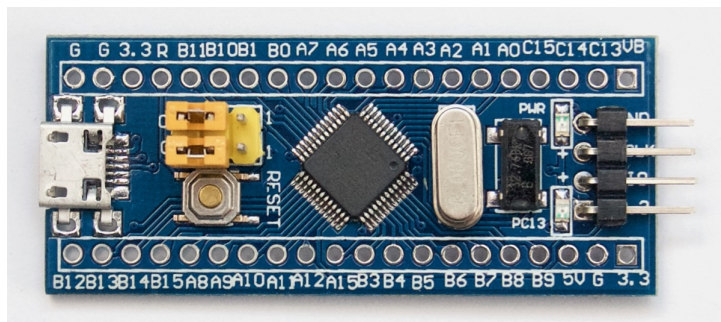


Figura 6 – Placa de desenvolvimento *Blue Pill*

Fonte: (JEELABS.ORG, )

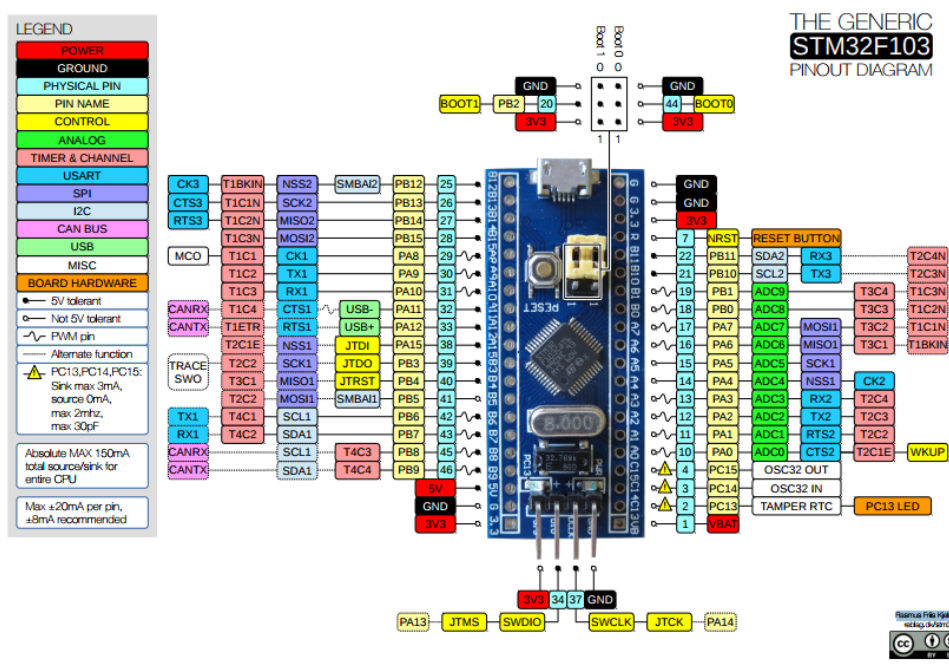


Figura 7 – Placa de desenvolvimento *Blue Pill* - configuração dos pinos (*Pinout*)

Fonte: (KJELDEN, 2017)

A placa em questão ainda oferece a possibilidade de ser programada na IDE e linguagem do Arduino, o que facilita por ter bastante suporte e ser uma linguagem bastante familiar.

### 3.3.2 Módulos

#### 3.3.2.1 Módulo GPS

Para o cumprimento do objetivo de registrar o local, a data e a hora das medições realizadas, o projeto requer a utilização de um módulo GPS. Estudou-se dois modelos possíveis de módulo, os quais serão apresentadas no [Apêndice D](#). Ao final do estudo, foi decidido pelo uso do GY-GPS6MV2 que utiliza a tecnologia NEO-6 u-blox 6, um chip de alta performance que tem, entre suas especificações (U-BLOX, 2018):

- Fonte de alimentação de 2,7v á 3,6v;
- Comunicação UART;
- USB;
- SPI;
- DDC;
- Interrupção externa.

### 3.3.2.2 Módulos de Comunicação sem Fio

O armazenamento de dados deve ser feito em um banco de dados. Para que as informações sejam enviadas para ele, uma boa alternativa é utilizar módulos de comunicação sem fio para realizar a conexão com a nuvem.

Optou-se pela tecnologia *Wi-Fi* pela facilidade que essa escolha traria na implementação do produto, já que é uma tendência das *smart cities* manter seu fluxo de dados e seus cidadãos conectados com sistemas de informação através da promoção da internet em locais públicos (MONZONI; NICOLLETTI, 2015), o que pode se verificar na cidade de Curitiba que já conta com conexão *Wi-Fi* inclusive em ônibus e terminais. A escolha também se deu pela comparação com outras tecnologias, que são apresentadas no [Apêndice D](#), que contempla também uma opção ao módulo utilizado. É fato que essa opção conta com um gasto de energia bem maior se comparado a algumas outras tecnologias e portanto, como esse é um fator de importância, é aconselhável que as outras tecnologias sejam reconsideradas assim que o seus custos de implementação diminuam.

Outra opção seria usar o *Zigbee* (padrão IEEE 802.15.4), que apresenta um menor consumo de energia, limitando a quantidade de dados que pode ser transmitida (COSTA, 2006), que ainda é pouco popular no Brasil e em dispositivos de uso comum à população (como celulares, roteadores e afins). Principalmente por dificuldade no acesso aos módulos, seja por seu preço ou por sua baixa disponibilidade de *access points*, visando um projeto de fácil uso em qualquer ambiente, esta tecnologia não foi escolhida.

### 3.3.2.3 Sensores de som

Para a obtenção dos níveis sonoros, é necessária a utilização de um transdutor, capaz de transformar a energia sonora em alguma outra forma de energia que possa ser tratada pelo circuito a ser desenvolvido, como corrente ou tensão. Esses sensores são chamados de microfones e eles se encontram no mercado em diferentes tipos e direcionados para diversas aplicações. Foi feito um estudo sobre os modelos de microfones mais comuns para que depois houvesse a decisão sobre o qual seria mais apropriado para o caso. Os resultados do estudo são apresentados no [Apêndice C](#). A partir do estudo foi escolhido o modelo capacitivo, como é explicado no [Seção 3.1](#).

Devido à maior disseminação do primeiro e maior disponibilidade de módulos e *chips* prontos para tal função, optou-se pela utilização de tecnologia GPS para determinação do

posicionamento das medições.

### 3.3.3 Microfone

Foi escolhido como sensor um microfone de eletreto do tipo capacitivo. A escolha deste sensor foi pelo seu preço acessível, sua alta sensibilidade, baixo ruído interno, alta estabilidade, e baixa distorção, características necessárias e adequadas para o projeto desenvolvido.

Para o circuito desenvolvido, houve a necessidade de converter a capacitância em tensão elétrica, o que pode ser feito utilizando uma carga elétrica constante (amplificador de carga). Essa carga foi introduzida de maneira externa, como é mostrado na [Figura 8](#) (BALBINOT; BRUSAMARELLO, 2013).

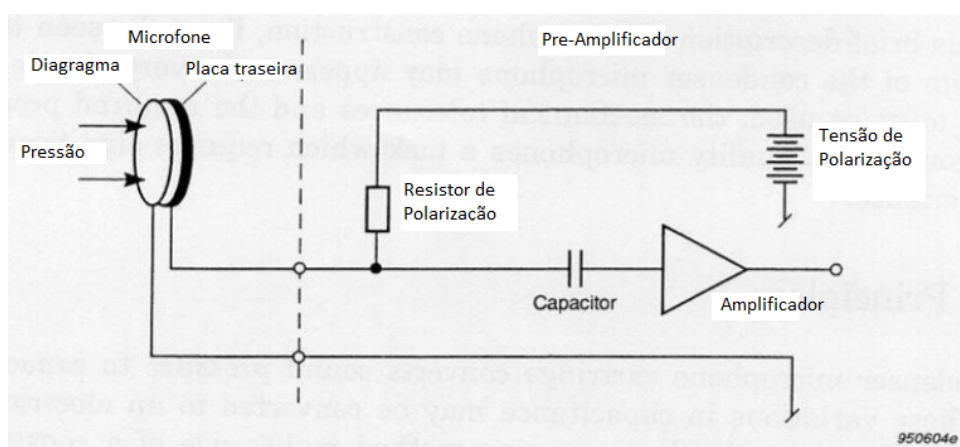


Figura 8 – Princípio da transdução do microfone capacitivo.

Fonte: BALBINOT; BRUSAMARELLO adaptado.

Quatro microfones foram selecionados para a implementação do presente trabalho, os quais são citados abaixo:

- Módulo detector de som RobotDyn;
- Módulo detector de som KY-037;
- Microfone de bancada Knup KP-903;
- Módulo de microfone MAX9814;

### 3.3.4 Amplificador

Circuitos de amplificação têm o objetivo de aumentar a intensidade de sinais elétricos. O sinal elétrico obtido pelo sensor, exige um circuito amplificador de tensão, que deve ser conectado o mais próximo possível do transdutor, para que seja possível o uso de cabos longos e conexões com circuitos de baixa impedância. Escolheu-se utilizar o circuito descrito na [Figura 9](#), sendo seu primeiro estágio um amplificador inversor com levantamento de tensão à metade da

tensão de alimentação do circuito com o intuito de tornar possível a interpretação do sinal sem perda das componentes negativas.

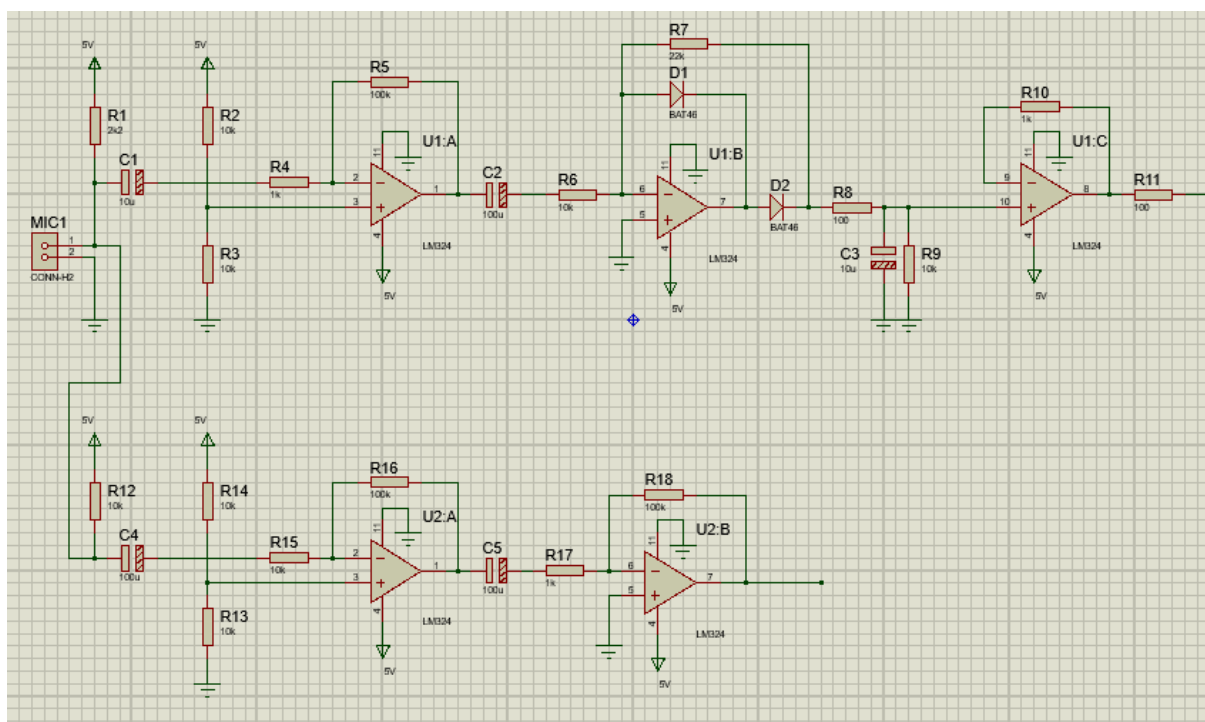


Figura 9 – Esquemático do circuito de amplificação do sinal do microfone

Fonte: Autoria própria

### 3.3.5 Detector de envoltória

O segundo estágio da amplificação do circuito de análise de amplitude do sinal é composto por um detector de envoltória de ganho igual a 2,2. Circuitos detectores de envoltória são detectores de pico com tempo de descarga, desta forma, pode-se acompanhar a envoltória de um sinal oscilatório, permitindo a análise de sua amplitude. Esta solução foi escolhida pela facilidade de implementação e pela necessidade de redução da oscilação do sinal amplificado.

Uma alternativa seria fazer esse detector em *software*, o que acarretaria em um maior poder de processamento necessário pelo microcontrolador, além de menos espaço sobrando para utilizar com o buffer das amostras. Por esse motivo, decidiu-se por utilizar a solução de *hardware*, evitando esse gasto desnecessário no microcontrolador.

### 3.3.6 Detecção de frequência

Um circuito de detecção de frequência foi necessário para a análise de resultados com a curva de compensação A descrita na [Equação \(4\)](#). Para tal, foi pensado em se amplificar

o sinal do microfone ao ponto de saturação, simulando um sinal digital e possibilitando sua análise em uma porta digital da placa de desenvolvimento.

### 3.3.7 Compensação do Sinal

Para eliminar componentes não sensíveis ao ouvido humano e realizar a compensação em direção à curva A (Figura 2), concluiu-se que seria necessário filtrar o sinal obtido. Foram feitos testes com filtros passa-baixas capacitivos e fazendo o tratamento do sinal por software. Após os testes concluiu-se qual é a maneira mais adequada para tratar o sinal no caso em questão.

## 3.4 SOFTWARE

Um dos pontos cruciais do projeto é a implementação da visualização dos dados, aqui representada pela escolha de uma página *web*, tecnologia que embora não tenha sido diretamente tratada durante o período de graduação mostra-se ser de fácil construção. Esta escolha se baseia na facilidade de acesso de qualquer pessoa aos dados, com a inclusão de alguns filtros por data e região, de forma a facilitar o acesso e entendimento do usuário que viesse a fazer uso dos dados coletados.

Através de protocolos de comunicação *web* (POST, GET, etc), deveria haver uma comunicação entre ele e os componentes de *hardware*, portanto seria função também do *software* agregar uma API de coleta de dados. O protocolo de comunicação escolhido foi o TCP/IP, por questão da facilidade de implementação nos dois extremos do projeto, (*software* e *hardware*), além de facilitar a integração entre eles ao se utilizar de uma interface por API, que desacopla as duas partes do sistema.

Não existem requisitos primários de proteção para o acesso dos dados, visto que se trata de um repositório de interesse público. Tal preocupação fica apenas na aquisição das informações através da API, que faz uma verificação nas mensagens recebidas por um formato específico (determinístico). Caso válidos, são direcionados ao banco de dados.

Como última característica, o uso de banco de dados é relevante no cumprimento do requisito de armazenamento dos dados coletados, estando acessíveis a qualquer momento por quem tenha acesso autorizado à estrutura. Este deve apresentar boa escalabilidade e ser ágil o suficiente para que pesquisas mais complexas aconteçam em tempo limitado. Estes requisitos são impostos pela funcionalidade do protótipo de coleta e envio das informações, que gera uma grande quantidade de dados a ser tratada pelos usuários.

### 3.4.1 IDE/Linguagem de Programação

Para este projeto, pensou-se a princípio na utilização do Microsoft Visual Studio 2017 como IDE, disponibilizado de forma gratuita através do programa *Imagine*, voltado para

acadêmicos não apenas da UTFPR mas de qualquer instituição filiada à *Microsoft*. Esta escolha levou ao uso de *backend* escrito em C#.

### 3.4.2 Serviço Web

Planejou-se utilizar a integração entre o a IDE escolhida, *Microsoft Visual Studio 2017* e a plataforma *online* da própria *Microsoft*, o *Azure* (AZURE, 2018). Dessa forma, os passos de publicação e manutenção seriam facilmente integrados em um único local, facilitando os trabalhos posteriores aos de programação.

### 3.4.3 Front-end

Visando simplificar as tarefas e aproveitar o suporte *online*, para este projeto escolheu-se HTML5 como padrão principal para desenvolvimento de visualização, aliado à *scripts* em *JavaScript* e seu *framework*, *JQuery*, além de estilização dos conteúdos em *CSS*. Todos de fácil suporte *online*, com comunidades dedicadas ao seu uso.

### 3.4.4 Banco de Dados

Para este projeto, o *PostgreSQL* foi escolhido por aplicar o conceito de *open-source*, um dos principais motos do projeto como um todo. Ele é capaz de lidar com uma grande quantidade de dados e está facilmente disponível nos principais serviços *web* para hospedagem gratuita, além de ser facilmente portátil entre os mais diversos sistemas operacionais disponíveis.

### 3.4.5 Versionamento

A escolha pelo versionamento através do *Git* para um histórico simplificado das diversas implementações realizadas, além de facilitar o acesso colaborativo de todo o time ao código fonte, seria de essencial importância em todos os passos a seguir, na execução. Portanto, foi planejada a criação de projetos não apenas para o *software*, mas também para o *hardware*. O serviço planejado era o *GitHub*, visto que boa parte dos grandes serviços *web* que são usados para a publicação (*deploy*) do aplicativo, como o *Microsoft Azure*, *Amazon Web Services (AWS)* ou o *Heroku*, possuem integração nativa com esse serviço em especial, o que facilitaria o trabalho do programador.



## 4 CWDB - SISTEMA PARA DETECÇÃO DINÂMICA E ANÁLISE DE POLUIÇÃO SONORA

O nome do projeto, CWdB, vem de uma ligação entre a sigla do aeroporto da cidade de Curitiba, CWB, e a unidade de medida para leitura de nível sonoro, dB, resultando em "CWdB".

Durante o projeto, *hardware* e *software* foram desenvolvidos em paralelo, sempre buscando o cumprimento dos requisitos iniciais, os quais foram verificados constantemente e, sempre que possível, houveram testes de integração entre as plataformas.

### 4.1 HARDWARE

Alguns detalhes sobre a execução do projeto de *hardware* são elucidados na presente seção.

#### 4.1.1 Microfone

Para a construção do *Hardware*, três microfones foram testados junto aos circuitos de condicionamento do sinal de forma a obter os resultados desejados.

As curvas de resposta em frequência dos microfones foram levantadas utilizando áudios com mesmo volume para diferentes frequências, anotando-se a amplitude do sinal elétrico pós-polarização e amplificação. Os valores foram então normalizados com relação à resposta de 1kHz para facilitar a interpretação dos dados. Tais análises podem ser vistas no [Apêndice D](#).

Diferentemente dos módulos detectores de som testados, o microfone Knup KP-903 possui informações fornecidas pelo fabricante. Estas são descritas na Tabela 4.1.1:

<b>Especificação</b>	
Direcionalidade	Omnidirecional
Resposta em frequência	20Hz a 16kHz
Impedância	2.0kOhm
Sensibilidade	-47dB a 1kHz
Tensão de alimentação	4.5V

Tabela 1 – Tabela de informações do fabricante - Microfone Knup KP-903

Fonte: Rótulo do produto

A resposta em frequência do microfone polarizado e amplificado segue na [Figura 10](#).

Pode-se observar que em relação as curvas do [Apêndice D](#), o presente microfone não possui ganhos altos em baixas frequências (20Hz a 1kHz) e um ganho moderado a médias frequências (1 a 8kHz), sendo a opção mais viável ao projeto, reduzindo a necessidade de compensação.

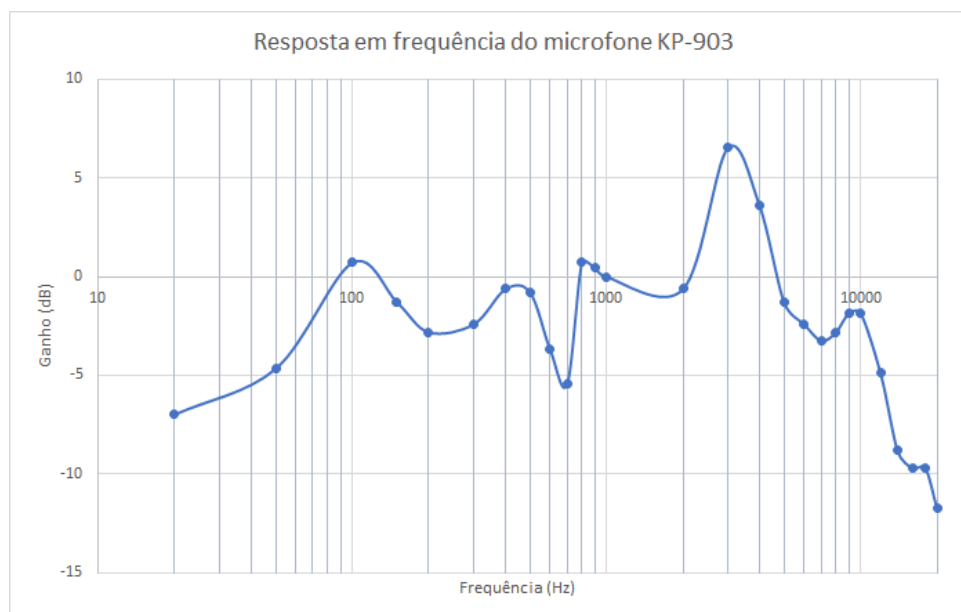


Figura 10 – Resposta em frequência do microfone KP-903

Fonte: Autoria própria

#### 4.1.2 Condicionamento do sinal

As etapas de condicionamento do sinal foram implementadas com amplificadores operacionais com o intuito de preparar o sinal para a etapa de compensação realizada pelo microcontrolador.

##### 4.1.2.1 Amplificação

O circuito de condicionamento do sinal é construído por um resistor de polarização para alimentação do microfone e um capacitor compondo o circuito básico de transdução de um microfone de eletreto. O sinal é então encaminhado a um amplificador operacional inversor de ganho igual a 100 (ou 20dB) e levantamento de tensão em 2,5V (pela tensão de referência na porta não-inversora). A [Figura 11](#) descreve o circuito de tratamento inicial do sinal recebido pelo microfone.

##### 4.1.2.2 Detector de envoltória

De forma a medir a amplitude do sinal sonoro adquirido e condicionado no circuito anterior, foi necessário desenvolver um circuito de detecção de envoltória, reduzindo a oscilação do sinal para melhores resultados. A [Figura 12](#) mostra o circuito em questão. É aplicado um amplificador inversor, de ganho igual a 2,2 vezes, e retificador de precisão meia onda (composto pelos diodos D1 e D2), seguido de um segurador de pico no capacitor C3 e descarga para resetar a envoltória (realizado pelo resistor R9).

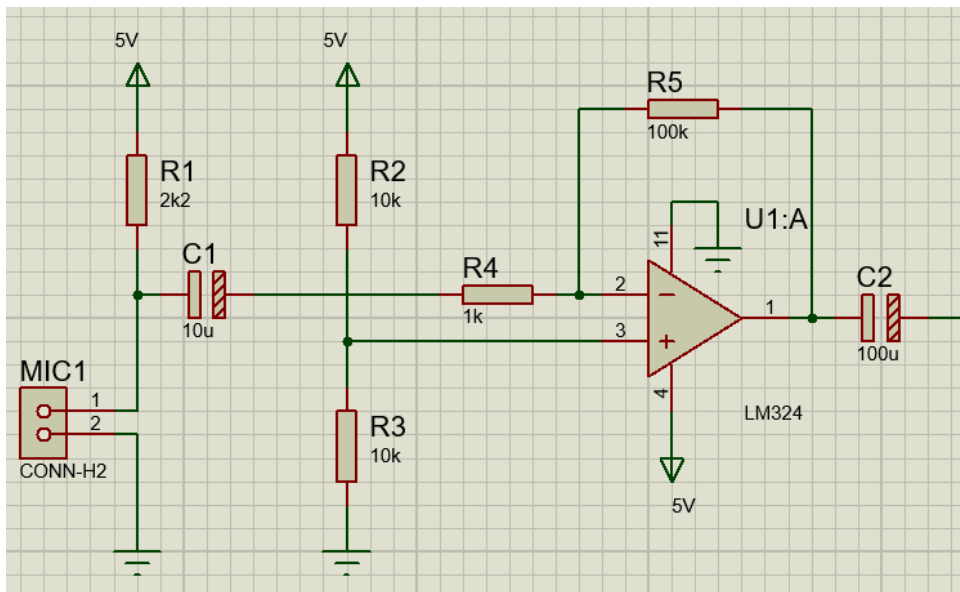


Figura 11 – Circuito de amplificação do sinal do microfone

Fonte: Autoria própria

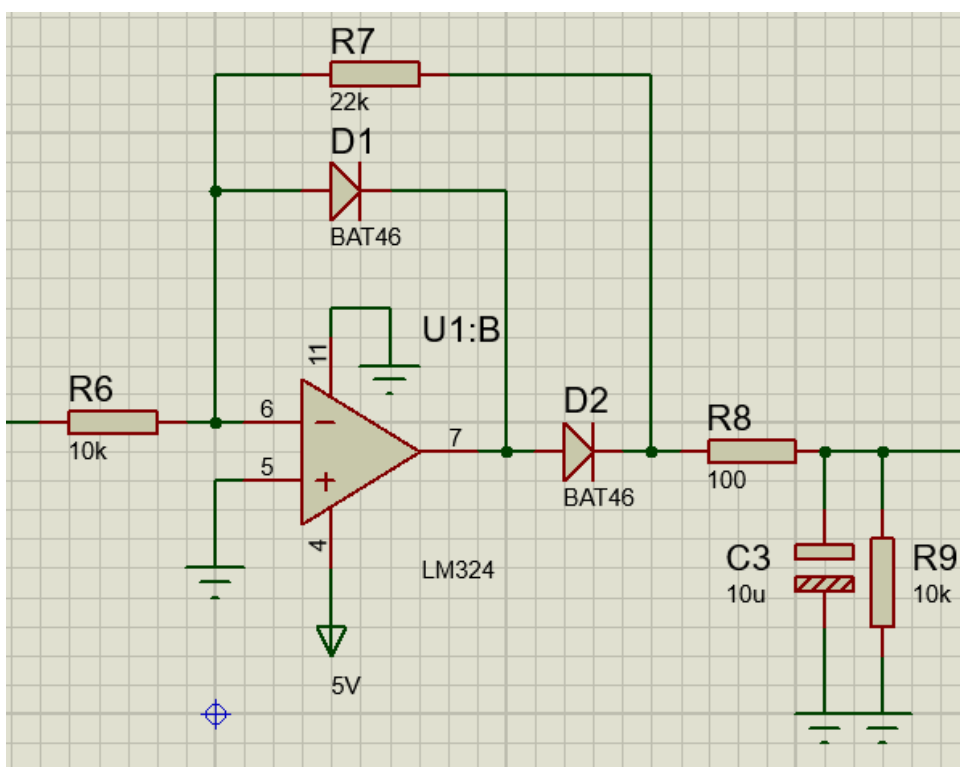


Figura 12 – Circuito de detecção de envoltória

Fonte: Autoria própria

#### 4.1.2.3 Buffer

Um amplificador buffer (evidenciado na [Figura 13](#)) é utilizado entre o detector de envoltória e a placa de desenvolvimento. A alta impedância de saída desta configuração de amplificador impede que o circuito que a segue interfira com o sinal em sua entrada.

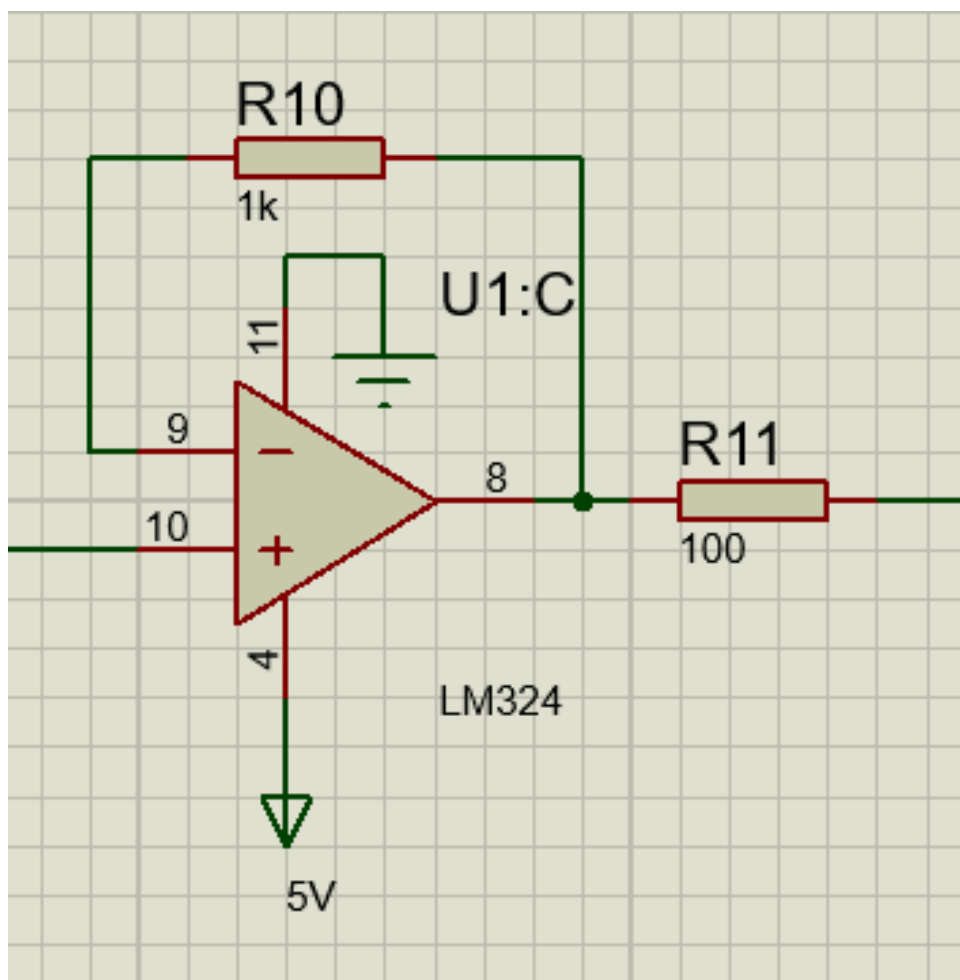


Figura 13 – Circuito de amplificador buffer

Fonte: Autoria própria

#### 4.1.2.4 Detecção de frequência

Um sistema de detecção de frequência foi necessário para a implementação do método de compensação por software. Para tal, utilizou-se um amplificador inversor de ganho igual a 100 após a primeira amplificação do sinal do microfone. Estes valores de ganho foram selecionados de forma a alcançar a saturação do sinal no amplificador, simulando, assim, um sinal digital e permitindo a detecção da frequência deste sinal por meio de uma porta digital da placa de desenvolvimento.

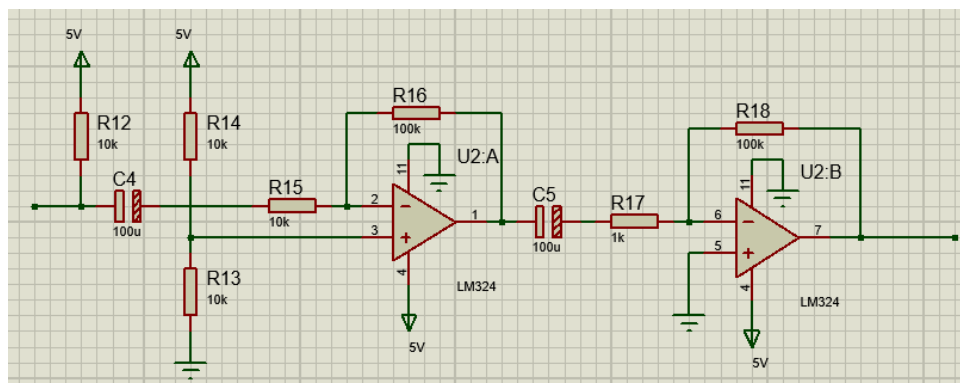


Figura 14 – Circuito de amplificação com saturação do sinal para detecção da frequência

Fonte: A autoria própria

#### 4.1.3 Compensação do sinal

Após a análise do estudo feito por Griffiths (1974), incluiu-se um filtro passa-altas de segunda ordem, com frequência de corte de 1kHz ao projeto, entre o circuito amplificador e o detector de envoltória. Apesar de a resposta obtida ser condizente com a esperada, após testes de campo, concluiu-se que não estava adequada ao projeto pois atenuava componentes em frequências sensíveis ao ouvido humano.

Resolveu-se também testar um filtro passa altas de primeira ordem, com frequência de corte em 194Hz, o qual experimentalmente apresentou melhor resposta em frequência, pois considera frequências sensíveis ao ouvido humano que eram descartadas pelo filtro com a frequência de corte em 1kHz.

Em seguida, foram feitas tentativas com tratamento digital do sinal, fazendo a compensação diretamente no código da placa. Um dos métodos testados foi aplicar as fórmulas de geração das curvas A, B e C discutidos no Capítulo 2. Os resultados obtidos foram condizentes com o esperado.

Além da compensação para a resposta se adequar à faixa de frequência desejada, houve a necessidade de tratar o sinal para que fosse obtida a resposta logarítmica, correspondente ao comportamento do som. Para isso utilizou-se um método no qual os valores obtidos pelo circuito construído e por um decibelímetro compensado pela curva A, da marca ICEL e modelo DL-1100 (ICEL, 2018), foram comparados. Mantendo ambos os aparelhos de medição (o projetado e a referência de calibração) a 1,2 metros de uma fonte sonora reproduzindo um sinal sonoro de 1kHz, alterou-se a intensidade do sinal (volume) gradativamente, anotando os resultados obtidos do decibelímetro e na saída do ADC do microcontrolador. A frequência de 1kHz foi escolhida para a calibração pelo fato de, segundo a curva A (Figura 2), esta frequência estar no patamar de corte da compensação (significando uma compensação de 0dB sobre o sinal medido). Gerou-se o gráfico de resposta, referente à comparação, e a partir desta a curva de tendência logarítmica a sua respectiva fórmula. Comparou-se os resultados, aplicando, então

a fórmula adquirida ao código.

## 4.2 SOFTWARE

Todo o projeto de *software* foi desenvolvido e versionado sobre o Git no *link*: <https://github.com/josneijr/CWdB>.

### 4.2.1 Ambiente de Desenvolvimento Integrado (IDE) e Linguagem de Programação

Apesar de o projeto ter sido planejado no *Microsoft Visual Studio* com desenvolvimento em ASP.NET, houve a troca para o *Microsoft Visual Code* e o *NODE.JS*. A ideia foi simplificar os passos da execução, a começar pela nova IDE que segue os moldes da escolha anterior, mas que simplifica as opções disponíveis, limitando-se a agregar apenas aquilo que foi realmente utilizado pelo projeto em questão.

A grande vantagem do *Visual Studio Code* (MICROSOFT, 2017) sobre alternativas como o *Sublime Text* (SUBLIMETEXT, 2017) e *Atom* (ATOM, 2017) fica pelos recursos de auto completar nativo à todas as ferramentas utilizadas, suporte à um crescente número de tecnologias de programação e pacotes como NPM, além da interface amigável para quem já está familiarizado com os produtos da *Microsoft*, sem se ater à um alto custo agregado, por se tratar de um editor gratuito. Visto se tratar de uma ferramenta de edição, o terminal integrado em conjunto as ferramentas tratadas a seguir como o NPM, Express e o próprio Git, se tornou um importante aliado nas tarefas de versionamento, *debug*, publicação e *log* dos dados.

### 4.2.2 NODE.JS

A mudança da linguagem ASP.NET para a NODE.JS se mostrou uma ótima escolha. A complexidade e principalmente custo agregado da ASP.NET tornou a opção altamente não recomendável para este projeto em especial. A escolha do NODE.JS permitiu aproveitar toda a integração *open-source* das outras ferramentas que serão explanadas a seguir. Concluiu-se que muitas são as vantagens dessa linguagem que se não é melhor do que aquela pensada a princípio, ao menos se iguala quando comparadas em questões como desempenho, segurança e portabilidade. Podemos listar:

- **DESEMPENHO:** apesar de ser um sistema single-thread, sua gestão de notificações é bem escrita, apresentando um desempenho similar àquele encontrado no ASP.NET, como podemos verificar na [Figura 15](#). Com isso, pode-se concluir que o uso de uma ou outra depende apenas de fatores como: IDE, familiaridade de uso e simplicidade da ferramenta.
- **OPEN-SOURCE:** por ser um serviço aberto, é muito fácil encontrar suporte para essa linguagem que é tão difundida na execução de projetos *back-end*. Além disso, a evolução da linguagem, que depende dos usuários que contribuem para o projeto, possibilitou agregar recursos para os diversos tipos de tecnologia utilizados em uma aplicação *web*,

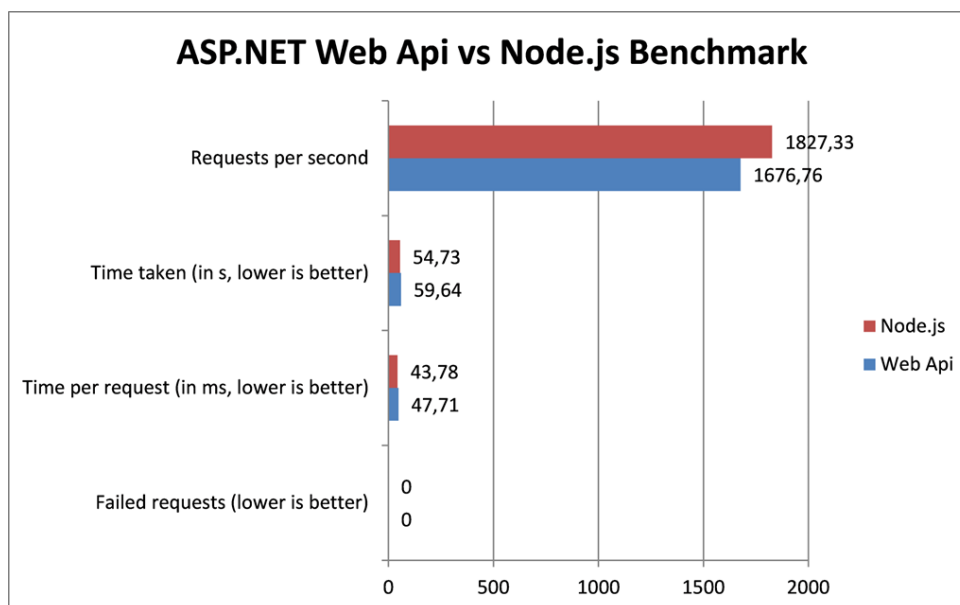


Figura 15 – Comparação de desempenho entre ASP.NET e NODE.JS

Fonte: [PETROV](#)

acompanhando novas tendências de uso como o *big-data* que tende a ser o futuro do projeto em questão.

O último motivo para a escolha específica do NODE.JS como linguagem de *back-end*, consiste em sua ampla gama de bibliotecas que facilitam diversas tarefas corriqueiras. Para este projeto, por exemplo, foram utilizadas:

1. **Express:** ([EXPRESSJS, 2017](#)) para a criação do servidor, rotas (redirecionamento de páginas solicitadas), definição de comandos HTML como GET e POST.
2. **EJS (Embedded Javascript Templates):** ([EJS, 2017](#)) permite reaproveitar partes de uma página em outra, como em campos comuns da página (menus e afins). Além disso, facilita o uso de variáveis na interface entre *back* e *front-end*.
3. **Sequelize:** ([SEQUELIZE, 2017](#)) responsável por um alto nível de abstração no acesso ao banco, além de permitir uma rápida troca de tecnologia, caso necessário, já que aceita boa parte dos bancos SQL como MySQL, PostgreSQL, SQLite e MSSQL.

#### 4.2.3 Banco de Dados

A escolha pelo PostgreSQL ([POSTGRESQL, 2017](#)) beneficiou o projeto com sua facilidade de uso, além de ser um projeto *open source*, sendo essa a sua maior vantagem sobre as outras opções SQL conhecidas. Outra vantagem é a integração com o servidor *web* utilizado, o Heroku, que possui um *add-on* específico para criação de banco de dados em PostgreSQL.

Podemos listar as vantagens relevantes à esse projeto, como visto em ([POSTGRESQL, 2017](#)). São elas:

1. **Código aberto:** por se tratar de um projeto com fins acadêmicos, a gratuidade e

facilidade de acesso devem ser fatores primordiais na escolha de qualquer tecnologia a ser utilizada.

2. **Confiabilidade:** é utilizado o método do MVCC (*Multi Version Concurrency Control*, onde todas as consultas de escrita e leitura ocorrem de forma simultânea e em harmonia, sem que ocorra a sobreposição, como explicado por (SMITH CHRISTOPHER BROWNE, 2007)). Nele, ao realizar uma consulta, o banco retorna com uma versão estática ou *snapshot* de seu conteúdo no momento da consulta. Caso ocorra alguma alteração durante o processamento dos dados, esta não influenciará no que já houver sido passado para o requisitante, nem tão pouco irá impedir a escrita dos novos dados. Como o sistema terá uma série de usuários e equipamentos em constante comunicação, torna-se essencial levar esse quesito em consideração.

O acesso ao banco ocorre diretamente no projeto criado no Heroku, previamente tratado nesse projeto. Para isso, basta escolher o *add-on* correspondente aquele onde o banco foi referenciado, chamado "*Heroku Postgres*". A *dashboard* com todas as informações básicas é mostrada na Figura 16. Nela, é possível conferir a quantidade de pontos adicionados no banco no momento da consulta, o número de tabelas (uma única para todas as amostras) e a quantidade de espaço utilizada pelo componente.

Já na Figura 17 podemos conferir o exemplo de algumas linhas salvas no banco, já com informações salvas pela interface entre o *hardware* e a API do software. O acesso tanto para visualização como edição dos dados (caso necessário) é facilitado com esse *add-on*, também de fácil acesso por estar ligado ao projeto pelo Heroku.

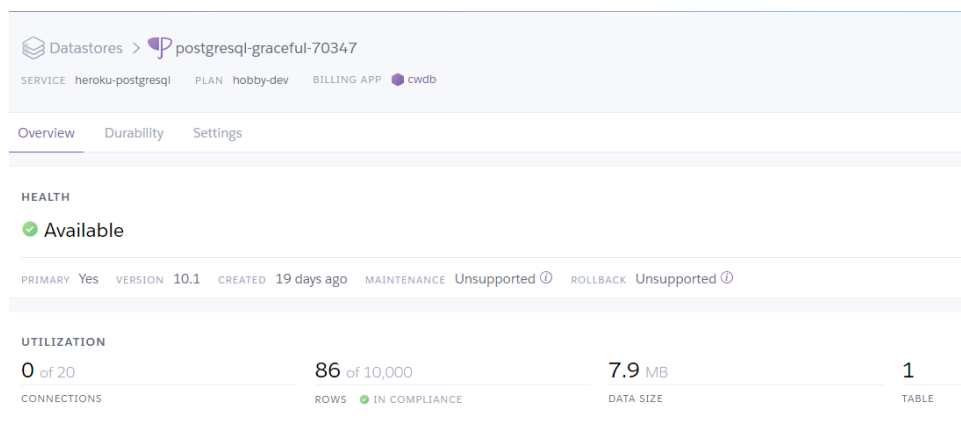


Figura 16 – *Dashboard* do Banco de Dados do projeto, visto pelo *add-on* Heroku Postgres

#### 4.2.4 Página HTML

A interface da página *web* criada aproveitou o *framework* do *Bootstrap* (BOOTSTRAP, 2017) para automatizar a edição e responsividade do *site*, disponível então para multiplataformas em diversos tamanhos de dispositivos. Isso é uma parte importante do processo de criação, visto que hoje, mais do que nunca, os celulares ocupam uma fatia importante do mercado de



Id	Data	Latitude	Longitude	Amplitude	Createdat	Updatedat
86	January 22, 2018 00:08	-25.41	-49.32	7.84	January 22, 2018 00:08	January 22, 2018 00:08
85	January 22, 2018 00:08	-25.41	-49.32	4.64	January 22, 2018 00:08	January 22, 2018 00:08
84	January 22, 2018 00:08	-25.41	-49.32	6.67	January 22, 2018 00:08	January 22, 2018 00:08
83	January 22, 2018 00:08	-25.41	-49.32	2.15	January 22, 2018 00:08	January 22, 2018 00:08
82	January 22, 2018 00:08	-25.41	-49.32	2.47	January 22, 2018 00:08	January 22, 2018 00:08
81	January 22, 2018 00:08	-25.41	-49.32	7.08	January 22, 2018 00:08	January 22, 2018 00:08
80	January 22, 2018 00:08	-25.41	-49.32	4.37	January 22, 2018 00:08	January 22, 2018 00:08

Figura 17 – Parte do Banco de Dados do projeto, visto pelo *add-on* Adminium

acessos à *internet*, principalmente no Brasil, onde em 2016 houve a duplicação no número de usuários móveis em relação a 2014 (VITÓRIA, 2017).

Os elementos de *UI* são então baseados em componentes prontos. O mapa vem da integração com a API pública do *Google Maps* (GOOGLE, 2017), que oferece a possibilidade de se criar um mapa com diversos recursos (mapa de calor, markers, etc), sem qualquer custo para um projeto em criação como este. Uma alternativa, caso o projeto tome grandes proporções, seria mudar para uma ferramenta totalmente gratuita, como o *OpenStreetMap* (OPENSTREETMAPS, 2017).

Por último, as requisições entre o *front-end* e *back-end* são todas tratadas em *Ajax* (AJAX, 2017), uma ferramenta do *framework* JQuery (JQUERY, 2017). Com ele, é possível serializar um objeto, transformando-o junto de todas as suas propriedades em uma representação gráfica, quando temos a necessidade de enviá-lo para uma API qualquer.

O elemento de gráfico, presente na tela de análises, vem de uma ferramenta bastante popular para tal fim chamada *Chart.JS* (CHART.JS, 2018). Apesar de não planejada a princípio, foi de grande auxílio para atingir o objetivo de criar uma forma visualmente eficiente de para mostrar os dados aos usuários.

#### 4.2.5 Serviço *Web*

Apesar do planejamento inicial pelo uso do *Azure*, houve a troca pelo *Heroku* como principal serviço *web* utilizado pelo projeto. Isso se deu através da análise de dois requisitos: facilidade de hospedar a página e o banco em um mesmo local, além do custo zero para essas operações, por período ilimitado. A única mudança do escopo inicial para que essa hospedagem fosse possível consistiu na troca entre C# e NODE.JS como tecnologia para o *back-end*, como já tratado aqui. Existe suporte não oficial à hospedagem C# na plataforma assim como outras diversas outras tecnologias, mas como o intuito era justamente ter o suporte oficial, além de facilidade no processo de integração, preferiu-se trocar para o NODE.JS.

É claro que existe o lado negativo em se usar o Heroku. Por se tratar de um serviço

gratuito, existe um limite de banda, instâncias simultâneas e a impossibilidade de se usar um domínio customizado, constituem algumas delas. Nenhuma dessas características interferiu nesse projeto, e caso fosse necessário, poder-se-ia optar por um aumento no plano escolhido para este projeto junto à ferramenta, que passaria a ter um custo baixo em relação às outras plataformas, suportando então recursos como: escalabilidade, domínio personalizado, suporte direcionado e maior disponibilidade do servidor. Portanto, uma solução maleável para qualquer fosse o tamanho alcançado por este projeto sem gerar qualquer custo a princípio, perfeito para a validação do protótipo sugerido.

A estrutura de todo projeto no Heroku é bastante similar, quando se trata de *User Interface (UI)*. A *dashboard* principal é composta das principais informações e atalhos às principais funções que o programador pode vir a precisar. Temos *links* aos add-ons, último status de *deploy* do código, que nesse caso é ligado ao projeto através da ferramenta de versionamento já vista, o GitHub, últimas interações com o serviço, entre outras, como pode ser visto na [Figura 18](#).

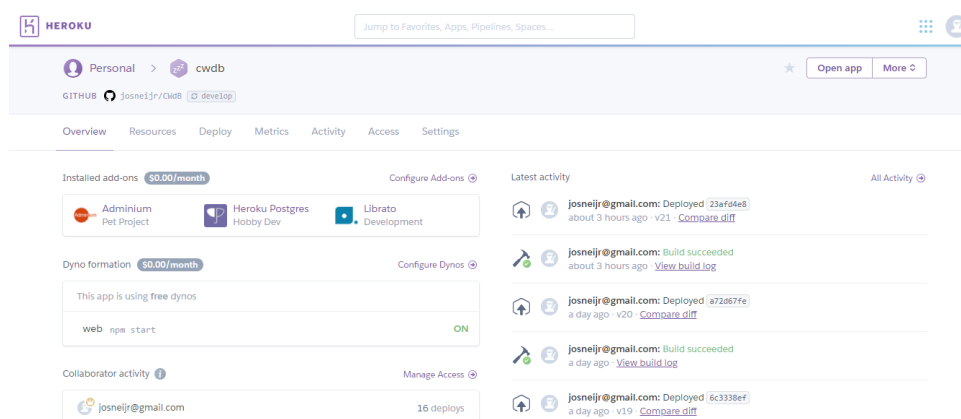


Figura 18 – *Dashboard* do projeto no Heroku

Outra ferramenta muito útil da plataforma é a visualização dos logs com as mensagens de informação, *debug* e erro programadas. É nela que temos todos os status das requisições de entrada e saída do *back-end*, previamente inclusos para facilitar o processo de monitoramento do *app web*. Um exemplo pode ser conferido na [Figura 19](#).

Application Logs ALL PROCESSES ↕

```
+00:00') RETURNING *;  
2018-01-27T00:33:17.203121+00:00 app[web.1]: Executing (default): INSERT INTO "samples"  
("id","data","latitude","longitude","amplitude","createdAt","updatedAt") VALUES (DEFAULT,'2018-01-27  
00:33:07.000 +00:00',0.00000,0.00000,0.045665,'2018-01-27 00:33:17.196 +00:00','2018-01-27 00:33:17.196  
+00:00') RETURNING *;  
2018-01-27T00:33:17.203412+00:00 app[web.1]: Executing (default): INSERT INTO "samples"  
("id","data","latitude","longitude","amplitude","createdAt","updatedAt") VALUES (DEFAULT,'2018-01-27  
00:32:57.000 +00:00',0.00000,0.00000,0.060073,'2018-01-27 00:33:17.196 +00:00','2018-01-27 00:33:17.196  
+00:00') RETURNING *;  
2018-01-27T01:09:05.303967+00:00 heroku[web.1]: Idling  
2018-01-27T01:09:05.304735+00:00 heroku[web.1]: State changed from up to down  
2018-01-27T01:09:06.021081+00:00 heroku[web.1]: Stopping all processes with SIGTERM  
2018-01-27T01:09:06.117739+00:00 heroku[web.1]: Process exited with status 143
```

Figura 19 – Logs do projeto no Heroku

## 5 ANÁLISE E DISCUSSÃO DOS RESULTADOS

O presente capítulo visa apresentar os resultados do trabalho desenvolvido.

### 5.1 CALIBRAÇÃO

Os resultados da calibração para uma resposta logarítmica discutida na [Subseção 4.1.3](#) seguem descritas na [Tabela 2](#) e [Figura 20](#).

dBA	ADC
41	0.004545
52	0.011212
60	0.023939
66	0.030303
73.5	0.063636
77.5	0.118182
82.6	0.193939
85.3	0.260606
89.8	0.387879
91.2	0.421212
92.7	0.484848
94.5	0.554545
95.6	0.569697
97.6	0.593939

Tabela 2 – Tabela de comparação dos resultados obtidos no ADC do microcontrolador e medidos por um decibelímetro

Na [Figura 20](#), é possível observar duas curvas. A curva de tendência gerada a partir dos dados coletados é representada pela linha pontilhada em preto. A proximidade das duas respostas provou viável o uso deste método.

A [Equação \(7\)](#) descreve a relação encontrada pelos valores medidos e aplicada ao código de conversão.

$$dBA = 10.904 * \ln(ADC) + 101.26 \quad (7)$$

### 5.2 PROTÓTIPO

O [Apêndice A](#) apresenta o esquemático completo do protótipo desenvolvido, bem como a representação do *layout* de PCB sugerido. O protótipo funcional foi confeccionado em duas placas universais, às quais foram soldados os componentes e suas respectivas conexões.

As [Figuras 21](#) e [22](#) mostram a construção do protótipo funcional. A [Figura 23](#) demonstra o protótipo montado em um *case* plástico, protegendo os componentes sem causar interferência à recepção de GPS ou comunicação *Wi-Fi*.

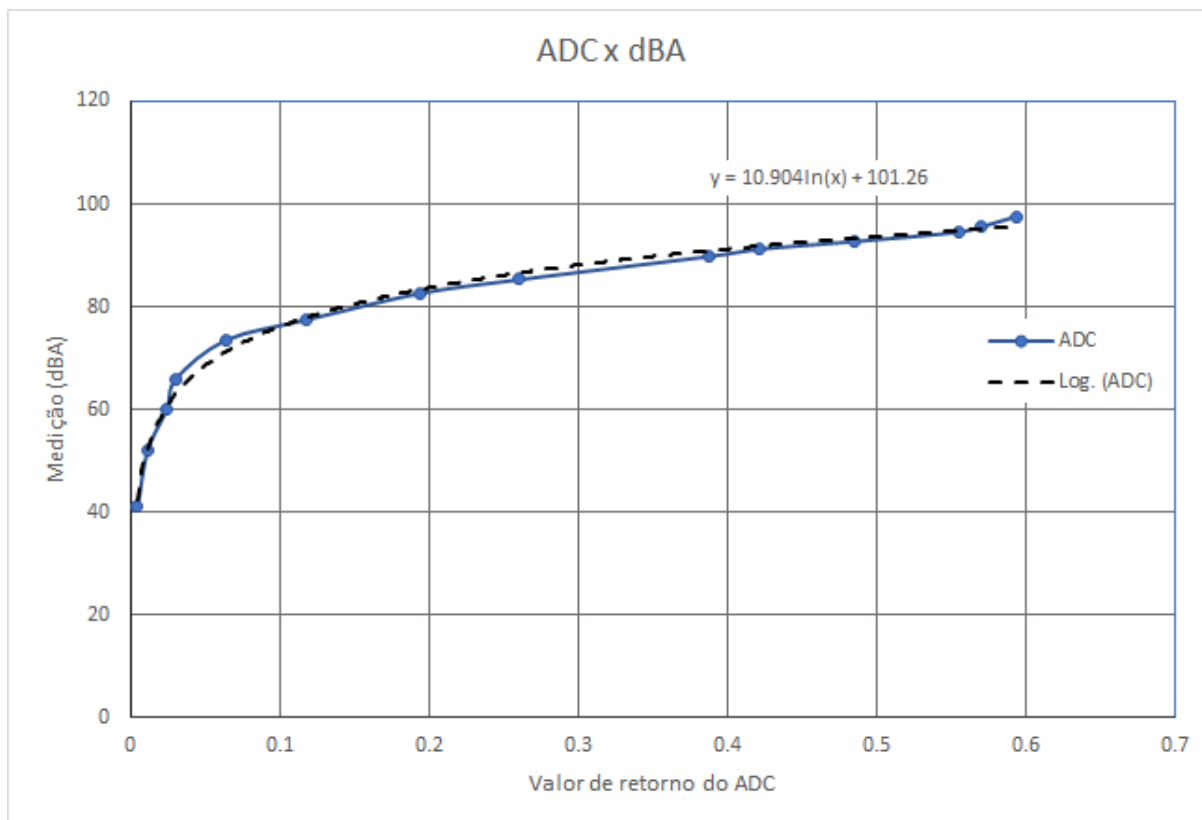


Figura 20 – Gráfico de resposta e curva de tendência

Fonte: Autoria própria

Durante o desenvolvimento, algumas considerações foram tomadas a partir da análise de resultados de testes do circuito. Em relação à solução de condicionamento do sinal, um osciloscópio foi utilizado para avaliar as características (frequência, amplitude, forma de onda e ruído) do sinal em cada etapa do processo. Foi notado um desempenho esperado no sinal após o detector de envoltória, com aumento gradual de tensão conforme o aumento da intensidade sonora aplicada ao microfone. Avaliando o detector de frequência, observou-se que a conexão do amplificador diretamente ao circuito que precede o detector de envoltória (primeiro estágio de amplificação) resultava em uma queda de tensão na envoltória. A solução (representada pela Figura 14) foi a replicação do primeiro amplificador e do circuito de polarização do microfone para garantir alimentação do sinal e impedir a interferência com a envoltória gerada.

Seguindo para a etapa de tratamento do sinal, de forma a facilitar a organização em um objeto JSON dos dados adquiridos do microfone e GPS, a placa de desenvolvimento *Blue Pill* estrutura-os no seguinte formato: "timestamp": "%02d/%02d/%02d %02d:%02d:%02d", "lat": "%05.5f", "long": "%05.5f", "value": "%03.6f"}"

Sendo:

- "timestamp" caracterizado pelos valores de: dia/mês/ano hora:minutos:segundos;
- "lat": o valor de latitude fornecido pelo GPS;



Figura 21 – Composição do protótipo funcional

Fonte: Autoria própria

- "long": o valor de longitude fornecido pelo GPS;
- "value": o valor em dbA da intensidade sonora do ambiente.

Desta forma, após enviado à NodeMCU via uma implementação de comunicação serial por software, o dado pode ser serializado e enviado ao banco caso haja conexão à Internet disponível ou enviado a um *buffer* circular até o momento da próxima conexão.

Durante os testes de comunicação serial, identificou-se que utilizando a taxa de transmissão (*baud rate*) padrão da NodeMCU de 115200bps havia perda de dados. Um ajuste da *baud rate* para 19200bps na comunicação entre placas foi necessário para garantir

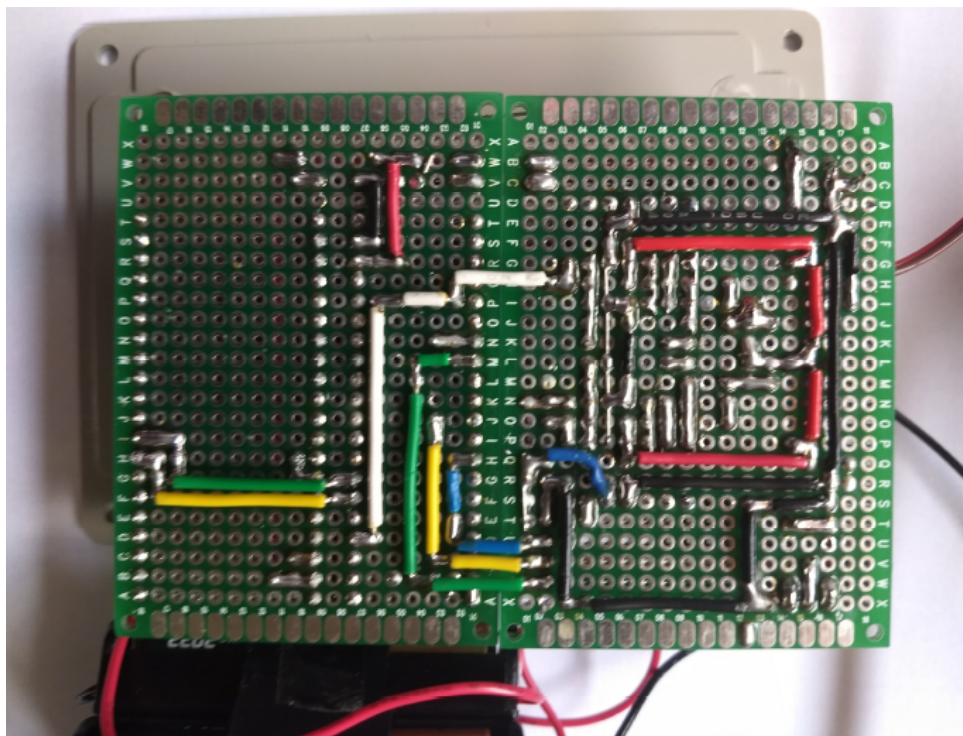


Figura 22 – Organização das ligações entre circuitos do protótipo

Fonte: Autoria própria

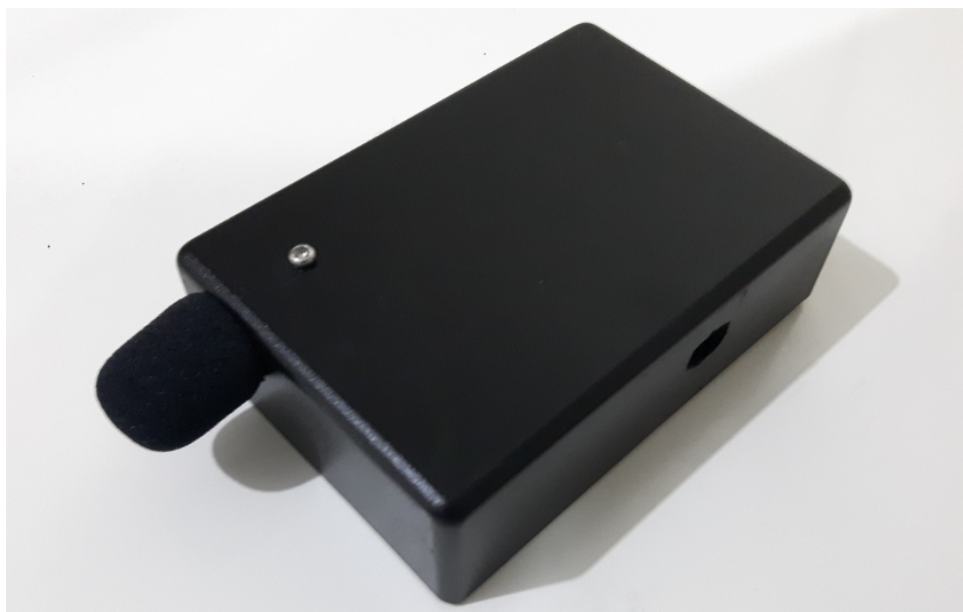


Figura 23 – Protótipo montado em um case plástico

Fonte: Autoria própria

estabilidade de transmissão e recepção. Após este processo, o dado é enviado à API feita com o *software* deste mesmo projeto.

### 5.3 IMAGENS DO SOFTWARE - (CWdB)

O *software* desenvolvido mostra de forma amigável ao usuário as informações coletadas pelos sensores na forma de um mapa de calor e marcadores. A [Figura 24](#) e a [Figura 25](#) exemplificam a visualização dos dados na página inicial do site.

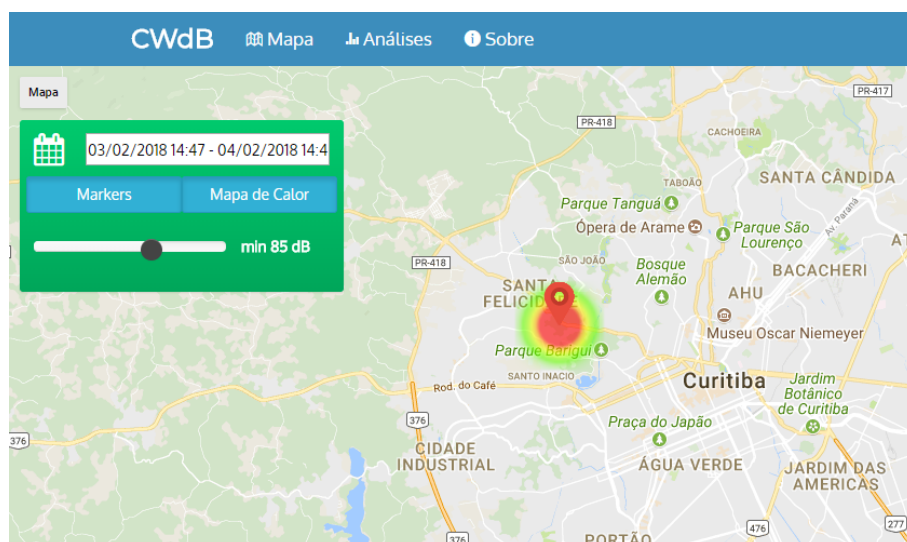


Figura 24 – Visão geral da tela de mapa

Fonte: Autoria própria

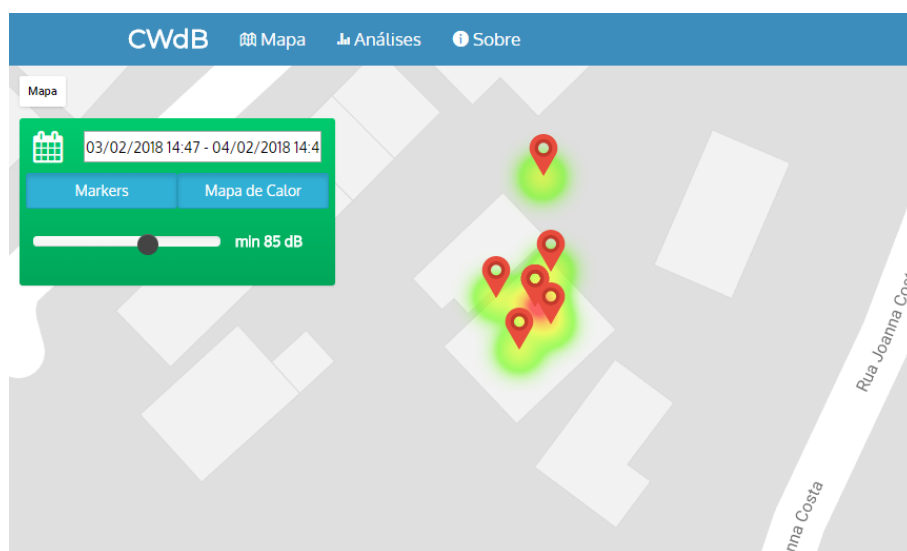


Figura 25 – Detalhe para o mapa de calor (*heatmap*) com *markers*, representando os pontos encontrados no banco de dados para o intervalo de tempo definido

Fonte: Autoria própria

A [Figura 26](#) mostra o sistema de refinamento da visualização das informações disponibilizadas no *heatmap*. Sendo possível selecionar a faixa de tempo e intensidade sonora mínima



dos dados coletados e possibilitando ao usuário escolher a visualização dos marcadores e do mapa de calor.

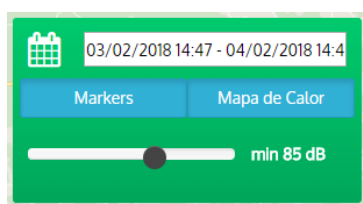


Figura 26 – Detalhe para a caixa contendo as opções de filtro e ativação dos *markers*/mapa de calor nos dados vistos no mapa

Fonte: Autoria própria

Na tela de análises, aqui representada em completo na Figura 27 podem ser analisados os pontos em exibição visual, detalhados no mapa da Figura 28. A Figura 29 detalha o campo com o número total de pontos, como no exemplo citado de 299, sendo 163 abaixo do nível definido nos filtros (70dB) e 136 pontos acima, como pode ser visto na Figura 31.

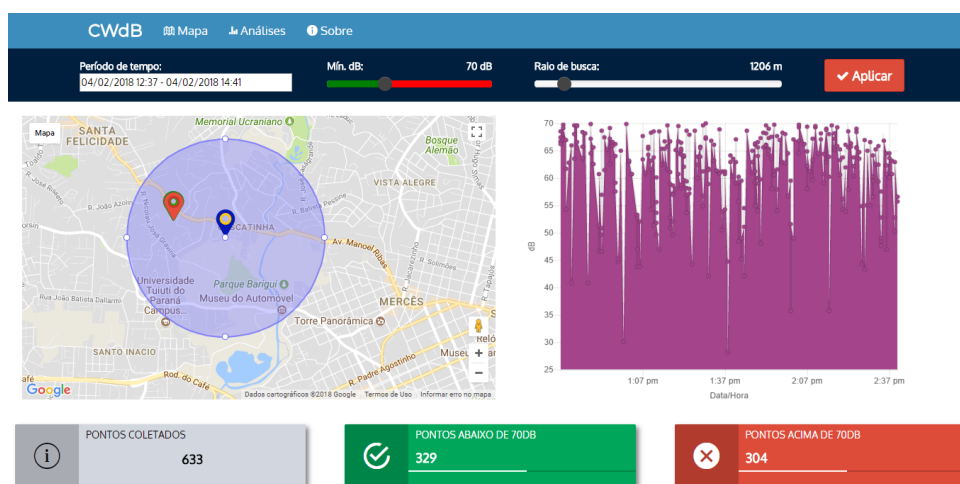


Figura 27 – Tela de análises, contendo os principais recursos para a seleção, filtragem e visualização dos dados contidos no banco

Fonte: Autoria própria

Já os testes com o banco mostraram que ele é capaz de chegar em 5992 registros sem qualquer problema de desempenho, tanto na leitura como na escrita de dados. O registro desse feito pode ser visto na Figura 32, com as informações necessárias presentes em cada linha (data e hora, valor em dB e posição, além dos registros de data e hora de chegada dos dados em banco). Esses registros foram recebidos do protótipo de *hardware*, via comunicação estabelecida entre eles pela API.

Importante citar também os *logs* recebidos por cada ação do *software*, seja na recepção de uma requisição de acesso por um usuário, seja por uma recepção de novos dados pelo

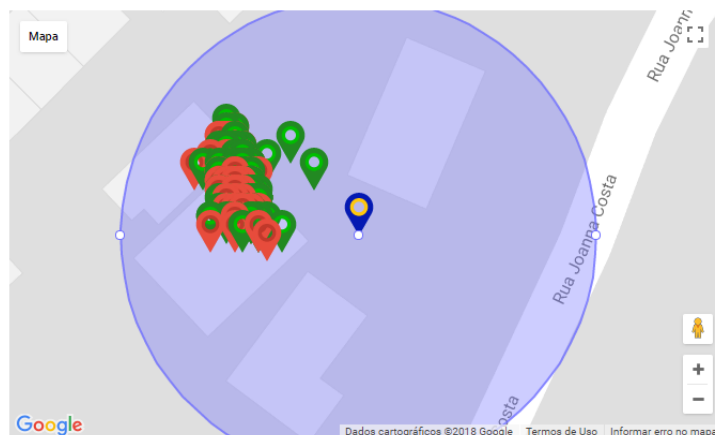


Figura 28 – Detalhe para o mapa da tela de análises, contendo *markers* apenas na região escolhida pelo usuário, além de diferenciação entre valores dentro e fora da escala escolhida para o nível sonoro (dB)

Fonte: Autoria própria

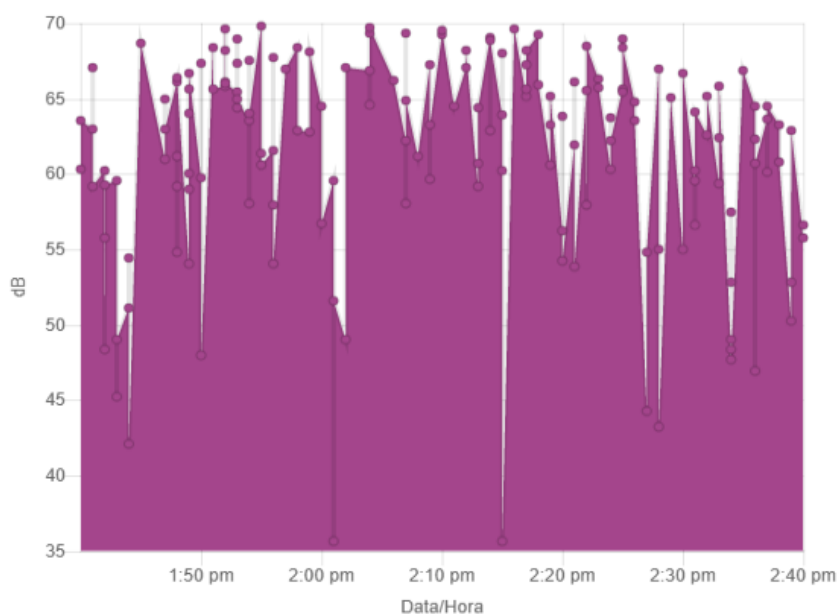


Figura 29 – Detalhe para o gráfico na tela de análises, o qual corresponde a uma visualização em linha de tempo da chegada dos dados, dentro dos filtros e região definidos, sem considerar no entanto o limiar de dB para referência.

Fonte: Autoria própria

*hardware*. Um exemplo desses *logs* pode ser visto em [Figura 33](#), onde um registro inválido é detectado na recepção da API. Essa visualização auxilia no processo de *debug* do *hardware*, visto que qualquer problema nele pode facilmente ser analisado sem que esteja conectado a um monitor serial do computador, por exemplo.

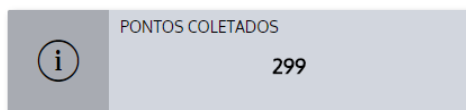


Figura 30 – Detalhe para o campo que exibe a quantidade total de pontos recolhidos para a região selecionada, no período de tempo especificado, sem considerar o limiar de dB para referência.

Fonte: Autoria própria



Figura 31 – Detalhe para os campos que exibem as quantidades totais de pontos recolhidos para a região selecionada, no período de tempo especificado, considerando o limiar de dB para referência.

Fonte: Autoria própria

Id	Data	Latitude	Longitude	Amplitude	Createdat	Updatedat
5,992	February 04, 2018 20:24	-25.41	-49.32	53.22	February 04, 2018 20:24	February 04, 2018 20:24
5,991	February 04, 2018 20:23	-25.41	-49.32	79.25	February 04, 2018 20:23	February 04, 2018 20:23
5,990	February 04, 2018 20:23	-25.41	-49.32	77.70	February 04, 2018 20:23	February 04, 2018 20:23
5,989	February 04, 2018 20:23	-25.41	-49.32	72.47	February 04, 2018 20:23	February 04, 2018 20:23
5,988	February 04, 2018 20:23	-25.41	-49.32	58.21	February 04, 2018 20:23	February 04, 2018 20:23
5,987	February 04, 2018 20:23	-25.41	-49.32	60.89	February 04, 2018 20:23	February 04, 2018 20:23

Figura 32 – Registro dos 5992 registros salvos em banco, com todos os dados completos.

Fonte: Autoria própria

```

Application Logs
2018-02-08T22:35:34.513868+00:00 heroku[router]: at=info method=POST path="/receiveData" host=cwdb.herokuapp.com request_id=2c947eb2-8512-4940-97f1-e724468ebfe9
 fwd="189.40.74.20" dyno=web.1 connect=2ms service=7ms status=200 bytes=200 protocol=http
2018-02-08T22:35:34.511215+00:00 app[web.1]: New data arrived at 2018-2-8 22:35:34
2018-02-08T22:35:34.511360+00:00 app[web.1]: Number of arrived samples: 1
2018-02-08T22:35:34.511474+00:00 app[web.1]: ts: 02/08/2018 22:35:33
2018-02-08T22:35:34.511542+00:00 app[web.1]: lat/long: 0.00000/0.00000
2018-02-08T22:35:34.511652+00:00 app[web.1]: value: 51.592190
2018-02-08T22:35:34.511716+00:00 app[web.1]: SAMPLE SEM GPS!
    
```

Figura 33 – Exemplo de log de novo registro inválido sendo recebido pela API

Fonte: Autoria própria

## 5.4 ANÁLISE DE CUSTO

A Tabela 3 evidencia os custos dos materiais utilizados na construção do protótipo descrito na Seção 5.2.

<b>Componente</b>	<b>Preço (R\$)</b>
Blue Pill	16,00
NodeMCU	11,00
GPS GY-GPS6MV2	23,00
Amplificadores	2,00
Barras de pinos	6,00
Componentes passivos	1,50
Microfone	15,00
Placas padrão	5,00
Baterias	6,00
Casing	10,00
Total	95,50

Tabela 3 – Tabela de custos dos materiais utilizados na construção do protótipo

Alcançando um custo total de R\$95,50, o protótipo manteve-se abaixo do valor (de aproximadamente R\$150,00 em Curitiba) de decibelímetros com resposta equivalente encontrados no mercado durante a realização deste trabalho.

## 6 CONCLUSÃO

### 6.1 CONSIDERAÇÕES FINAIS

Houve êxito no cumprimento do objetivo inicial, que consistiu na construção de um sistema (prova de conceito) capaz de identificar, mapear e armazenar dados sobre ruídos sonoros, assim como em todos os objetivos específicos.

A implementação do *hardware* provou-se desafiadora desde a pesquisa e delimitação de requisitos ao desenvolvimento de um protótipo funcional. A fase de pesquisa dos possíveis componentes e testes subsequentes para delimitação dos produtos a serem usados demandou mais do que o esperado. No entanto, definidos os componentes e configuração de circuitos a serem utilizados a integração seguiu com êxito.

Devido às respostas irregulares dos microfones testados, o condicionamento e compensação do sinal adquirido necessitaram de constantes pesquisa e ajustes. Contudo, optou-se por utilizar o método de compensação por meio de uma curva de compensação gerada empiricamente, simplificando o circuito de tratamento (neste caso, reduzindo a necessidade de implementação de filtros no circuito que tiveram resultados insatisfatórios em testes realizados durante o desenvolvimento do projeto.

Uma das grandes dificuldades encontradas foi a escolha da fonte de alimentação para o sensor móvel. O módulo GPS utilizado no projeto demanda bastante corrente e testes com pilhas comuns se mostraram ineficientes. O problema foi resolvido utilizando um *power bank*, que traz as vantagens de ser recarregável.

Os resultados com o *software* foram satisfatórios, com o cumprimento de todos os requisitos propostos inicialmente. Visto que estes são mais qualitativos, o cumprimento dos itens propostos é determinado através do uso do que foi desenvolvido, pelos autores do projeto e também, informalmente, por pessoas externas a este.

Houve sucesso em criar uma *interface* direta e intuitiva, como foi verificado ao apresentar a página para pessoas externas ao desenvolvimento. O mesmo se aplica com o uso de tabelas e filtros para exibição dos dados, cumprindo o requisito de análise e tratamento dos dados obtidos.

A criação da API de recebimento de dados foi mais simples do que o esperado, graças a compatibilidade do *hardware* com o envio de informações serializadas em *JSON*. O sistema é atualizado com a chegada de novos dados, sendo portanto indicado para uma análise rápida.

A implementação do protótipo do sistema desenvolvido comprovou sua potencial eficiência em trazer um impacto positivo na cidade, sendo possíveis aplicações, por exemplo, de identificação de áreas de riscos à saúde da comunidade, aquisição de informações para serem analisadas futuramente (alimentação de sistemas *Big Data*, por exemplo), indicador urbano como ferramenta legal que ajuda a provar e identificar casos de ruídos que desrespeitam as leis

ambientais e de trânsito, indicador de outros fatores como industrialização e urbanização das áreas mapeadas, entre outras.

O sistema desenvolvido foi testado e foi observada uma resposta satisfatória com relação à curva de referência, ou seja, a resposta foi corretamente compensada segundo a curva A, as intensidades de som se assemelham às identificadas pelo ouvido humano e os valores obtidos correspondem aos lidos em instrumentos de medição.

Foi atendido o objetivo de guardar dados de posição (*GPS*), data e hora e valor de cada medição. Tais informações estão armazenadas em um banco de dados online, disponíveis para acesso através do *software* desenvolvido, ou através de uma API de fácil consulta por qualquer outra aplicação.

Com relação aos módulos de comunicação e localização utilizados, *Wi-Fi* e *GPS* respectivamente, houve êxito desde as primeiras implementações, graças a todo suporte *on-line* encontrado (NODEMCU, 2018a) (NODEMCU, 2018b) (U-BLOX, 2018). A comunicação via protocolo TCP/IP se mostrou conveniente para a aplicação.

A escolha da tecnologia de banco de dados *PostgreSQL* para o armazenamento das informações, se mostrou uma boa escolha por trazer a opção de trabalhar com um número muito maior de dados, com relação a outras tecnologias estudadas, sem que o desempenho seja prejudicado, um cenário que não é aplicado ao protótipo em questão, mas que é crucial para que o projeto seja implementado como uma ferramenta de análise em *smart cities*, onde uma grande quantidade de dados é esperada para uma aplicação similar.

O protótipo desenvolvido tem as características de ser leve, de fácil portabilidade e de custo acessível. O protótipo pesa 270g, o que o torna prático e o possibilita ser colocado em diversos lugares, fixos ou móveis. No entanto, caso o equipamento seja instalado na parte exterior de veículos, um lugar adequado deve ser buscado para que o vento não cause interferências significativas nas medições, bem como o próprio motor do veículo. O custo total do protótipo foi de aproximadamente R\$: 95,50, um valor que pode ser reduzido por produção em larga escala e readequação do projeto de forma a integrar os controladores em uma PCB única sem a necessidade de placas de desenvolvimento. Comparando com o preço de um decibelímetro de resposta equivalente (aproximadamente R\$ 150,00 em Curitiba em 2018) como referência de seu baixo custo, considerando possuir outras capacidades que não apenas medir o nível sonoro, como receber dados do GPS e se comunicar com a API para armazenamento dos dados *on-line*.

Para a visualização dos dados coletados, foi criado um *site*<sup>1</sup> onde as informações podem ser filtradas, e visualizadas de uma forma clara e amigável. As ferramentas utilizadas atenderam aos requisitos iniciais por contemplar uma grande quantidade de pontos coletados, sem queda de desempenho significativa, principalmente no que tange o acesso e processamento dos dados. A usabilidade desta ferramenta foi testada informalmente por diferentes pessoas, o que foi de grande ajuda, pois os relatos foram cruciais para que o site fosse estruturado

---

<sup>1</sup><http://cwdb.herokuapp.com>

da maneira que se encontra, e que a compreensão da representação gráfica dos dados fosse validada.

O objetivo secundário de utilizar ferramentas abertas (*opensource*) foi realizado com sucesso, comprovando a possibilidade da aplicação do projeto em regiões que dispõem de poucos recursos financeiros para esse tipo de investimento, tão necessário no futuro das cidades inteligentes.

### 6.1.1 Comentários adicionais

Para o cumprimento dos objetivos, houve a necessidade da busca do conhecimento em áreas como engenharia de som e gestão de projetos. O projeto poderia ser refinado com um conhecimento ainda maior em áreas como *design*, engenharia de *software* entre outras. Importante ressaltar a falta de disciplinas de integração, usando como base a grade de Engenharia da Computação, na mesma universidade (UTFPR). Houve um grande desafio a ser cumprido principalmente na parte de implementação do *software*, onde diversos conhecimentos externos precisaram ser buscados de forma independente para que o projeto pudesse ser realizado. Conclui-se, nesse tópico, que o suporte necessário para tais ferramentas tão presentes no dia-a-dia da população como um todo deve ser revisto. Mesmo com essa limitação, foi possível um grande avanço nesta área, como pode ser comprovado através do uso do produto pelos desenvolvedores e também por pessoas externas ao projeto, com *feedbacks* positivos ao objetivo desejado.

Os métodos ágeis utilizados para a gestão do processo de desenvolvimento foram de crucial importância para que o tempo utilizado na execução do trabalho fosse melhor aproveitado e que o resultado final fosse mais organizado. Metodologias que podem ser aplicadas não só para um projeto dessa estrutura, mas também para organização de tarefas cotidianas.

O projeto ainda foi de grande importância para a equipe, pois muito se aprendeu sobre as ferramentas utilizadas e foi uma grande oportunidade de colocar em prática conceitos aprendidos durante toda a graduação, em busca do benefício de toda a população da cidade de Curitiba. Ainda, adquiriu-se consciência de problemas e objetivos que podem ser solucionados em trabalhos futuros e da importância de utilizar os conhecimentos obtidos em busca do benefício coletivo.

Foi muito interessante aplicar não apenas o aprendido em sala de aula, mas também conhecimentos externos, agregados em estágios e projetos profissionais e pessoais desenvolvidos fora da universidade. Concluímos, por fim, que saímos com uma quantidade de conhecimento ampla e sólida, no que tange as diversas áreas da Engenharia Eletrônica, contemplando tanto *software*, *firmware* e *hardware*.

## 6.2 TRABALHOS FUTUROS

Uma sugestão seria adequar a leitura dos níveis sonoros para a faixa de frequência sensível por animais de estimação, com o objetivo de garantir a saúde física e mental deles. É um tema que seria de grande importância, pois, como a faixa de frequência audível para eles é diferente da dos humanos, ruídos prejudiciais para os animais podem nem ser detectados pelos seus donos.

O projeto do sensor móvel que mapeia o ruído na cidade, pode ser incrementado identificando a variação no limiar de risco, diminuindo o limiar para o aconselhado pela OMS para pontos próximos a escolas e hospitais. O projeto ainda poderia ser melhorado com o estudo de uma tecnologia de georreferenciamento mais econômica, visto que o utilizado gasta muita energia.

Quando ao banco de dados, a sugestão é verificar o desempenho do PostgreSQL em uma situação cotidiana, onde percebe-se a chegada de uma grande quantidade de dados vinda dos diversos sensores espalhados por uma cidade inteligente. É necessário analisar o desempenho não apenas para o armazenamento, mas também tratamento dos dados que precisam ser adquiridos com agilidade e confiança.

Poderia ser criado um aplicativo para a análise dos resultados obtidos. No que tange a página *web* para exibição final ao usuário, muita coisa pode ser melhorada ou até mesmo adicionada. Entre as muitas possibilidades, pode-se citar:

- Melhoria na forma como é apresentada a seleção de áreas vizinhas para o usuário;
- Incrementar o número de possibilidades para os filtros na tela principal do mapa;
- Personalização das telas de acordo com a necessidade de cada usuário.
- Integração dos dados com APIs externas, onde pode haver uma consulta de qualquer outro *site* externo aos dados previamente liberados para tal fim.

Para refinamento do projeto, novas áreas de expertise podem trabalhar com aspectos diferentes do protótipo, como por exemplo:

- **Design:** O protótipo foi pensado apenas em seu quesito de funcionamento, sem considerar questões estéticas, cruciais caso o produto chegue ao usuário final. É sugerido criar um invólucro menor e desenhado exclusivamente para esta função, aprimorando não apenas a aparência do produto, mas também sua usabilidade (com fácil acesso aos componentes, como carregamento das baterias, por exemplo);
- **Engenharia de Computação:** seria ideal acrescentar os conhecimentos de um engenheiro de computação, com possíveis ideias para a *interface* do usuário, seja com a mesma estrutura utilizada ou não, além de tópicos como a escolha de banco, forma de salvar os dados, criação e gestão do sistema e assuntos relacionados;
- **Engenharia do Som:** visto se tratar de um projeto totalmente ligado a análise de sinais sonoros, a análise por um engenheiro especializado pode trazer vantagens principalmente na análise e utilização de um equipamento otimizado para a captação dos níveis sonoros.



## Referências

- AGARWAL, S. K. **Noise Pollution**. [S.l.]: S.B. Nangia, 2009. Citado na página 9.
- AJAX. 2017. Disponível em: <<http://api.jquery.com/jquery.ajax/>>. Acesso em: 30 de Janeiro de 2018. Citado na página 35.
- ALSINA-PAGÉS, R. M. et al. Disentangling dynamics, computation and cognition. **National Center for Biotechnology Information Search database**, 2016. Disponível em: <<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5298630/#!po=0.862069>>. Citado 2 vezes nas páginas 2 e 8.
- AMAZON. **Amazon Web Services**. 2018. Disponível em: <<https://aws.amazon.com/pt/>>. Acesso em: 25 de Janeiro de 2018. Citado na página 14.
- APPLE. **Xcode**. 2018. Disponível em: <<https://developer.apple.com/xcode/>>. Acesso em: 04 de Fevereiro de 2018. Citado na página 12.
- ARDUINO. **Arduino Nano**. 2018. Disponível em: <<https://store.arduino.cc/usa/arduino-nano>>. Acesso em: 04 de Fevereiro de 2018. Citado na página 67.
- ATOM. 2017. Disponível em: <<https://atom.io/>>. Acesso em: 30 de Janeiro de 2018. Citado 2 vezes nas páginas 12 e 32.
- AZURE. 2018. Disponível em: <<https://azure.microsoft.com/pt-br/>>. Acesso em: 25 de Janeiro de 2018. Citado 2 vezes nas páginas 14 e 26.
- BALANCING the ZigBee/802.15.4 link budget. 2006. Disponível em: <<http://industrial.embedded-computing.com/pdfs/Atmel.Fall06.pdf>>. Citado na página 70.
- BALBINOT, A.; BRUSAMARELLO, V. **Instrumentação e Fundamentos de Medidas**. 2. ed. Rio de Janeiro: LTC, 2013. Citado 6 vezes nas páginas 8, 10, 11, 23, 63 e 64.
- BBC. **Programming software and the IDE**. 2018. Disponível em: <<https://www.bbc.co.uk/education/guides/zgmpr82/revision/4>>. Acesso em: 04 de Fevereiro de 2018. Citado na página 12.
- BERECHNUNG der Bewertungsfilter-Kurven. 2002. Disponível em: <<http://www.sengpielaudio.com/BerechnungDerBewertungsfilter.pdf>>. Acesso em: 30 de Janeiro de 2018. Citado na página 10.
- BOLTON, W. **Instrumentação e Controle**. [S.l.]: Hemus, 2002. Citado na página 19.
- BOOTSTRAP. 2017. Disponível em: <<https://getbootstrap.com/>>. Acesso em: 30 de Janeiro de 2018. Citado na página 34.
- CHART.JS. **Chart.JS**. 2018. Disponível em: <<http://www.chartjs.org/>>. Acesso em: 04 de Fevereiro de 2018. Citado na página 35.
- CO.LTD, A. T. **A6/A7/A6C User Manual**. 2016. Disponível em: <[https://www.makerfabs.com/desfile/files/A6\\_A7\\_A6C\\_datasheet-EN.pdf](https://www.makerfabs.com/desfile/files/A6_A7_A6C_datasheet-EN.pdf)>. Acesso em: 04 de Fevereiro de 2018. Citado na página 69.

- CORMEN CHARLES E. LEISERSON, R. L. R. C. S. T. H. **Introduction to Algorithms**. [s.n.], 2009. Disponível em: <[http://ressources.unisciel.fr/algoprogram/s00a/root/aa00/module1/res/%5BCormen-AL2011%5DIntroduction\\_To\\_Algorithms-A3.pdf](http://ressources.unisciel.fr/algoprogram/s00a/root/aa00/module1/res/%5BCormen-AL2011%5DIntroduction_To_Algorithms-A3.pdf)>. Acesso em: 08 de Fevereiro de 2018. Citado na página 12.
- COSTA, C. A. **Cidades Inteligentes e Big Data**. 2015. Disponível em: <[http://fgvprojetos.fgv.br/sites/fgvprojetos.fgv.br/files/cadernos\\_fgvprojetos\\_smart\\_cities\\_bilingue-final-web.pdf](http://fgvprojetos.fgv.br/sites/fgvprojetos.fgv.br/files/cadernos_fgvprojetos_smart_cities_bilingue-final-web.pdf)>. Acesso em: 17 de Janeiro de 2018. Citado 2 vezes nas páginas 4 e 5.
- COSTA, F. R. d. N.; ARAÚJO, C. C. S. C.; BORGES, I. T. **Uso abusivo do som: crime ou contravenção penal?** 2014. Disponível em: <<https://jus.com.br/artigos/33158/uso-abusivo-do-som-crime-ou-contravencao-penal>>. Acesso em: 30 de Janeiro de 2018. Citado 2 vezes nas páginas 7 e 8.
- COSTA, L. A. M. M. R. A. A. Evolução das redes sem fio: Um estudo comparativo entre bluetooth e zigbee. 2006. Disponível em: <<http://www.unipac.br/site/bb/tcc/tcc-a010b188f93af4c28ca9af23b9e3c476.pdf>>. Citado na página 22.
- EJS. 2017. Disponível em: <<http://www.embeddedjs.com/>>. Acesso em: 25 de Janeiro de 2018. Citado na página 33.
- EXPRESSJS. 2017. Disponível em: <[expressjs.com/pt-br/](http://expressjs.com/pt-br/)>. Acesso em: 25 de Janeiro de 2018. Citado na página 33.
- GOINES, L.; HAGLER, L. Noise pollution: A modern plague. In: . [S.l.: s.n.], 2012. Citado na página 1.
- GOOGLE. **Google Maps**. 2017. Disponível em: <<https://www.google.com.br/maps>>. Acesso em: 30 de Janeiro de 2018. Citado na página 35.
- GOOGLE. **Android Studio**. 2018. Disponível em: <<https://developer.android.com/studio/index.html>>. Acesso em: 04 de Fevereiro de 2018. Citado na página 12.
- GOUVEIA, A. **O que é uma API?** 2016. Disponível em: <<http://www.up.edu.br/blogs/engenharia-da-computacao/2016/07/01/o-que-e-uma-api/>>. Acesso em: 04 de Fevereiro de 2018. Citado na página 14.
- GRAFT, A.; VERHULST, S.; YOUNG, A. **BRAZIL'S OPEN BUDGET TRANSPARENCY PORTAL MAKING PUBLIC HOW PUBLIC MONEY IS SPENT**. 2016. Disponível em: <<http://odimpact.org/case-brazils-open-budget-transparency-portal.html>>. Acesso em: 04 de Fevereiro de 2018. Citado na página 14.
- GRIFFITHS, J. D. **The Design of a Low Cost Sound Level Meter**. Julho 1974. Dissertação (Mestrado) — USAF Academy, Colorado, United States, 1974. Disponível em: <<https://nepis.epa.gov>>. Acesso em: 20 de Janeiro de 2018. Citado na página 31.
- HALL, R. et al. The vision of a smart city. In: \_\_\_\_\_. [s.n.], 2000. Disponível em: <<http://www.osti.gov/scitech/servlets/purl/773961>>. Citado na página 1.
- HEROKU. 2018. Disponível em: <<http://www.heroku.com/>>. Acesso em: 25 de Janeiro de 2018. Citado na página 14.
- ICEL. **ICEL DL-1100**. 2018. Disponível em: <[http://www.icel-manaus.com.br/produto\\_descricao.php?id=718](http://www.icel-manaus.com.br/produto_descricao.php?id=718)>. Citado na página 31.

- JEELABS.ORG. **Let's start with a Blue Pill**. Citado 2 vezes nas páginas 20 e 21.
- JQUERY. 2017. Disponível em: <<https://jquery.com/>>. Acesso em: 30 de Janeiro de 2018. Citado na página 35.
- KJELDTSEN, R. F. **Blue Pill Pinout**. 2017. Disponível em: <<http://reblog.dk/wordpress/wp-content/uploads/2016/07/The-Generic-STM32F103-Pinout-Diagram.pdf>>. Acesso em: 04 de Fevereiro de 2018. Citado na página 21.
- KROEMER, K. H. E.; GRANDJEAN, E. **Manual de Ergonomia: Adaptando o Trabalho ao Homem**. 5. ed. Porto Alegre: Bookman, 2005. Citado 4 vezes nas páginas 6, 7, 9 e 10.
- LEAL, C. I. S. **Prefácio - Cadernos FGV Projetos - Cidades Inteligentes e Mobilidade Urbana**. 2015. Disponível em: <[http://fgvprojetos.fgv.br/sites/fgvprojetos.fgv.br/files/cadernos\\_fgvprojetos\\_smart\\_cities\\_bilingue-final-web.pdf](http://fgvprojetos.fgv.br/sites/fgvprojetos.fgv.br/files/cadernos_fgvprojetos_smart_cities_bilingue-final-web.pdf)>. Acesso em: 17 de Janeiro de 2018. Citado 2 vezes nas páginas 4 e 5.
- LLC, S. et al. **Software Livre e de Código Aberto**. 2001–2017. Disponível em: <[https://pt.opensuse.org/Software\\_Livre\\_e\\_de\\_C%C3%B3digo\\_Aberto#O\\_qu.C3.AA.C3.A9\\_Software\\_livre.3F](https://pt.opensuse.org/Software_Livre_e_de_C%C3%B3digo_Aberto#O_qu.C3.AA.C3.A9_Software_livre.3F)>. Acesso em: 04 de Fevereiro de 2018. Citado na página 14.
- LORA Technology. 2018. Disponível em: <<http://www.microchip.com/design-centers/wireless-connectivity/embedded-wireless/lora-technology>>. Citado na página 70.
- LORA vs Zigbee , Difference between LoRa and Zigbee. 2018. Disponível em: <<http://www.rfwireless-world.com/Terminology/LoRa-vs-Zigbee.html>>. Citado na página 70.
- MAX9814. 2018. Disponível em: <<https://datasheets.maximintegrated.com/en/ds/MAX9814.pdf>>. Acesso em: 04 de Fevereiro de 2018. Citado na página 67.
- MBED, A. **FRDMKL25Z**. 2017. Disponível em: <<https://os.mbed.com/platforms/KL25Z/>>. Acesso em: 04 de Fevereiro de 2018. Citado na página 68.
- MICROSOFT. **Microsoft Visual Studio 2017**. 2017. Disponível em: <<https://www.visualstudio.com/pt-br/>>. Acesso em: 30 de Janeiro de 2018. Citado 2 vezes nas páginas 12 e 32.
- MICROSOFT. **Microsoft Visual Studio Code**. 2018. Disponível em: <<https://code.visualstudio.com/>>. Acesso em: 04 de Fevereiro de 2018. Citado na página 12.
- MONZONI, M.; NICOLLETTI, M. **A Cidade Para os Cidadãos: Mobilidade, Energia e Agricultura Urbana**. 2015. Disponível em: <[http://fgvprojetos.fgv.br/sites/fgvprojetos.fgv.br/files/cadernos\\_fgvprojetos\\_smart\\_cities\\_bilingue-final-web.pdf](http://fgvprojetos.fgv.br/sites/fgvprojetos.fgv.br/files/cadernos_fgvprojetos_smart_cities_bilingue-final-web.pdf)>. Acesso em: 17 de Janeiro de 2018. Citado 2 vezes nas páginas 5 e 22.
- NANOBOX. 2018. Disponível em: <<https://nanobox.io/>>. Acesso em: 25 de Janeiro de 2018. Citado na página 14.
- NEPOMUCENO, L. X. **Acústica Técnica**. São Paulo: ETEGIL, 1968. Citado na página 7.
- NODEMCU. 2018. Disponível em: <[http://www.nodemcu.com/index\\_en.html](http://www.nodemcu.com/index_en.html)>. Acesso em: 04 de Fevereiro de 2018. Citado 2 vezes nas páginas 20 e 48.

- NODEMCU. **PROGRAMMING ESP8266 ESP-12E NODEMCU V1.0 WITH ARDUINO IDE INTO WIRELESS TEMPERATURE LOGGER**. 2018. Disponível em: <<http://www.instructables.com/id/ESP8266-NodeMCU-v10-ESP12-E-with-Arduino-IDE/>>. Acesso em: 04 de Fevereiro de 2018. Citado na página 48.
- NUNES, M. F. de O. Poluição sonora em centros urbanos: O ruído de tráfego veicular. 1999. Disponível em: <[http://www.abepro.org.br/biblioteca/ENEGEP1999\\_A0068.PDF](http://www.abepro.org.br/biblioteca/ENEGEP1999_A0068.PDF)>. Citado na página 7.
- OPENSTREETMAPS. 2017. Disponível em: <<https://www.openstreetmap.org>>. Acesso em: 04 de Fevereiro de 2018. Citado na página 35.
- PETROV, I. I. **Comparison of ASP.NET and Node.js for Backend Programming**. 2017. Disponível em: <<https://gist.github.com/ilyaigpetrov/f6df3e6f825ae1b5c7e2>>. Acesso em: 17 de Janeiro de 2018. Citado na página 33.
- PIMENTEL-SOUZA, F. Efeito do ruído no homem dormindo e acordado. **XIX ESBA, SOBRAC-2000, BH**, 2000. Disponível em: <<http://labs.icb.ufmg.br/lpf/pimentel,sobrac2000.html>>. Acesso em: 2018.01.23. Citado 2 vezes nas páginas 6 e 8.
- POSTGRESQL. **PostgreSQL Advantages**. 2017. Disponível em: <<https://www.postgresql.org/about/advantages/>>. Acesso em: 25 de Janeiro de 2018. Citado na página 33.
- POSTGRESQL. 2017. Disponível em: <<https://www.postgresql.org/about/advantages/>>. Acesso em: 25 de Janeiro de 2018. Citado na página 33.
- SANTAELLA, L. **Temas e dilemas do pós-digital: a voz da política**. [S.l.]: Pia Sociedade de São Paulo - Editora Paulus, 2016. Citado na página 4.
- SEQUELIZE. 2017. Disponível em: <<http://docs.sequelizejs.com/>>. Acesso em: 25 de Janeiro de 2018. Citado na página 33.
- SMITH CHRISTOPHER BROWNE, L. K. S. G. Comparativo de desempenho entre bancos de dados de código aberto. 2007. Disponível em: <[https://wiki.postgresql.org/wiki/Why\\_PostgreSQL\\_Instead\\_of\\_MySQL:\\_Comparing\\_Reliability\\_and\\_Speed\\_in\\_2007](https://wiki.postgresql.org/wiki/Why_PostgreSQL_Instead_of_MySQL:_Comparing_Reliability_and_Speed_in_2007)>. Acesso em: 2018.01.25. Citado na página 34.
- SUBLIMETEXT. **Sublime Text**. 2017. Disponível em: <<https://www.sublimetext.com/>>. Acesso em: 30 de Janeiro de 2018. Citado 2 vezes nas páginas 12 e 32.
- SYSTEMS, E. **Datasheet ESPRESSIF SMART CONNECTIVITY PLATFORM: ESP8266**. 2013. Disponível em: <<http://www.electroschematics.com/wp-content/uploads/2015/02/esp8266-datasheet.pdf>>. Acesso em: 04 de Fevereiro de 2018. Citado na página 69.
- U-BLOX. **Datasheet NEO-6 u-blox 6 GPSModules**. 2018. Disponível em: <<https://www.u-blox.com>>. Acesso em: 04 de Fevereiro de 2018. Citado 2 vezes nas páginas 21 e 48.
- VALLE, S. d. **Microfones**. 2. ed. Rio de Janeiro: Música e Tecnologia, 2002. Citado 3 vezes nas páginas 63, 64 e 65.
- VERGARA, A. **Os Territórios Inteligentes**. 2015. Disponível em: <[http://fgvprojetos.fgv.br/sites/fgvprojetos.fgv.br/files/cadernos\\_fgvprojetos\\_smart\\_cities\\_bilingue-final-web.pdf](http://fgvprojetos.fgv.br/sites/fgvprojetos.fgv.br/files/cadernos_fgvprojetos_smart_cities_bilingue-final-web.pdf)>. Acesso em: 17 de Janeiro de 2018. Citado na página 5.

VITÓRIA, F. **Dobra participação do acesso à internet por dispositivos móveis no Brasil**. 2017. Disponível em: <<https://noticias.r7.com/cidades/folha-vitoria/dobra-participacao-do-acesso-a-internet-por-dispositivos-moveis-no-brasil-06092017>>. Acesso em: 30 de Janeiro de 2018. Citado na página 35.

WHO, W. H. O. **Noise- Data and Statistics**. 2011. Disponível em: <<http://www.euro.who.int/en/health-topics/environment-and-health/noise/data-and-statistics>>. Acesso em: 22 de Janeiro de 2018. Citado 3 vezes nas páginas 5, 8 e 9.

WIFI Vs. Cellular: Differences and Uses For M2M Applications. 2018. Disponível em: <<https://www.link-labs.com/blog/wifi-vs-cellular-differences-for-m2m>>. Citado na página 70.

ZEIT.CO. **Zeit.co**. 2018. Disponível em: <<https://zeit.co/>>. Acesso em: 25 de Janeiro de 2018. Citado na página 14.

## Apêndices

## APÊNDICE A – Projeto do protótipo - (CWdB)

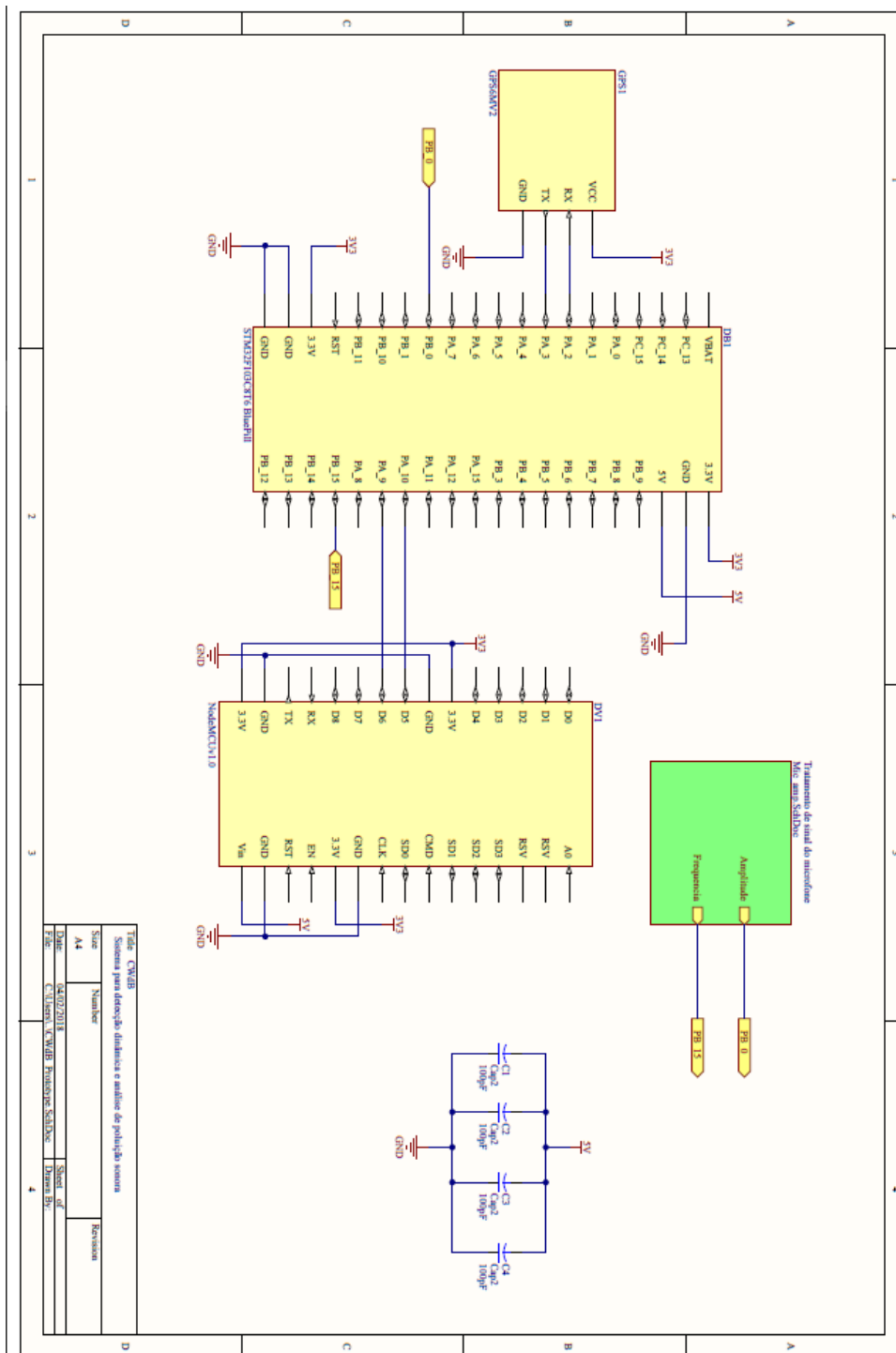


Figura 34 – Esquemático completo do protótipo

Fonte: Autoria própria

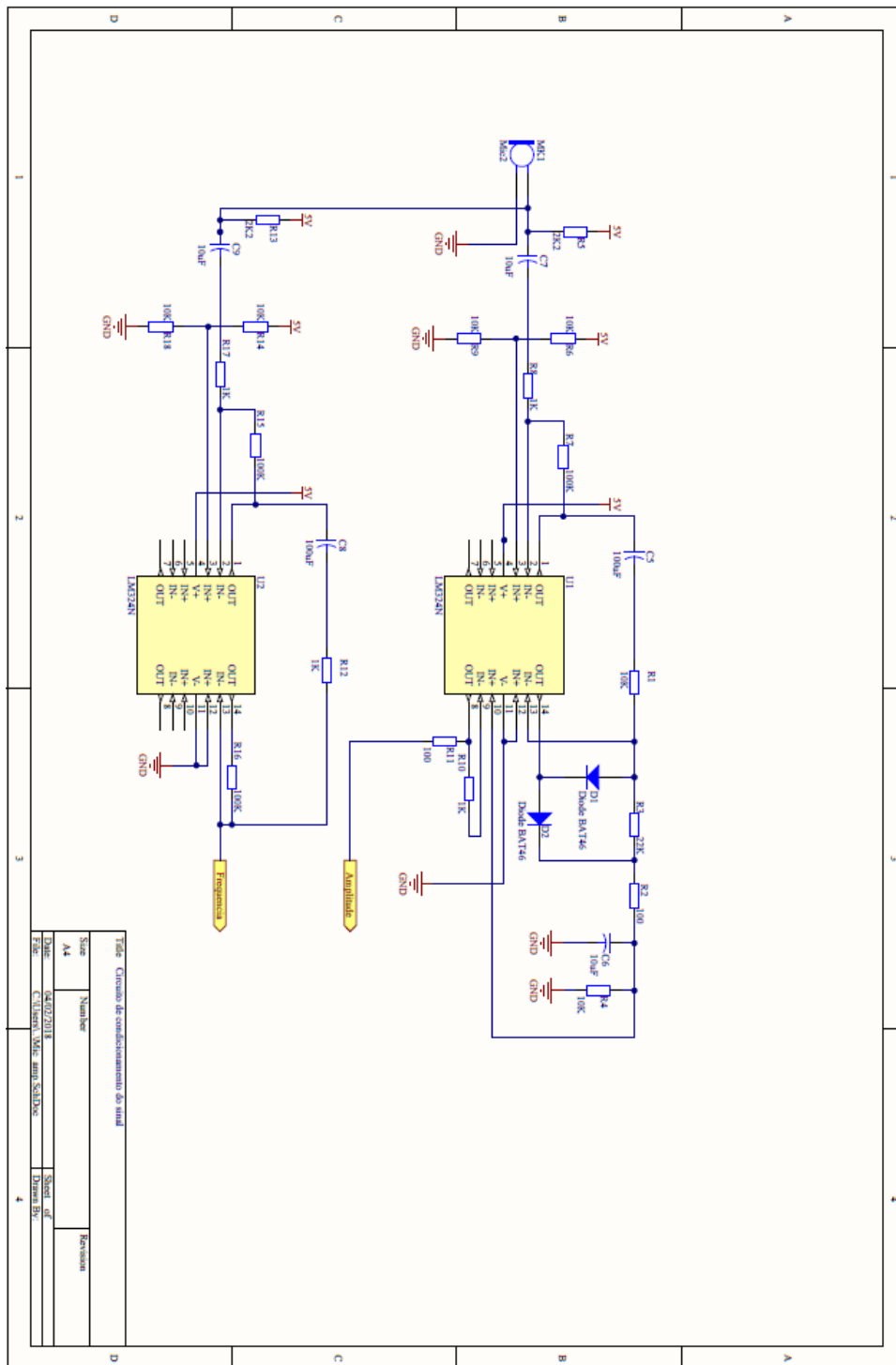


Figura 35 – Esquemático do circuito de condicionamento do sinal do microfone

Fonte: Autoria própria



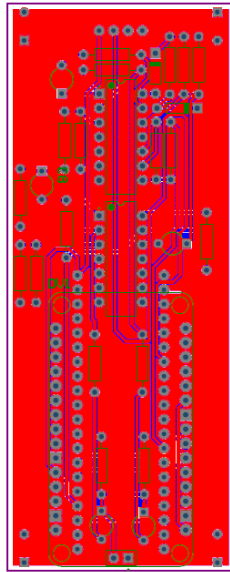


Figura 36 – Layout do Top Layer

Fonte: Autoria própria

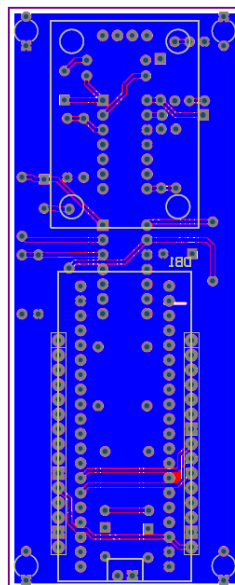


Figura 37 – Layout do Bottom Layer

Fonte: Autoria própria

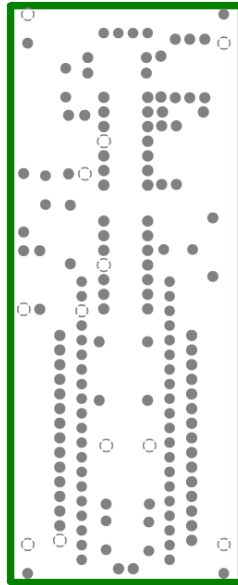


Figura 38 – *Layout do Power plane interno*

Fonte: Autoria própria

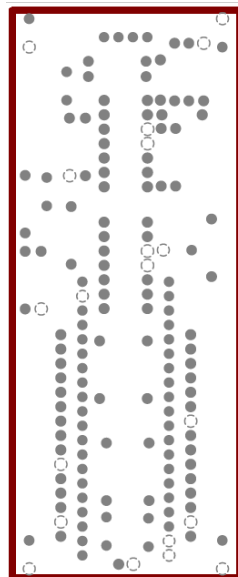


Figura 39 – *Layout do Ground plane interno*

Fonte: Autoria própria

**APÊNDICE B – Documentação do módulo detector de som Robotdyn enviadas pelo fabricante mediante solicitação**

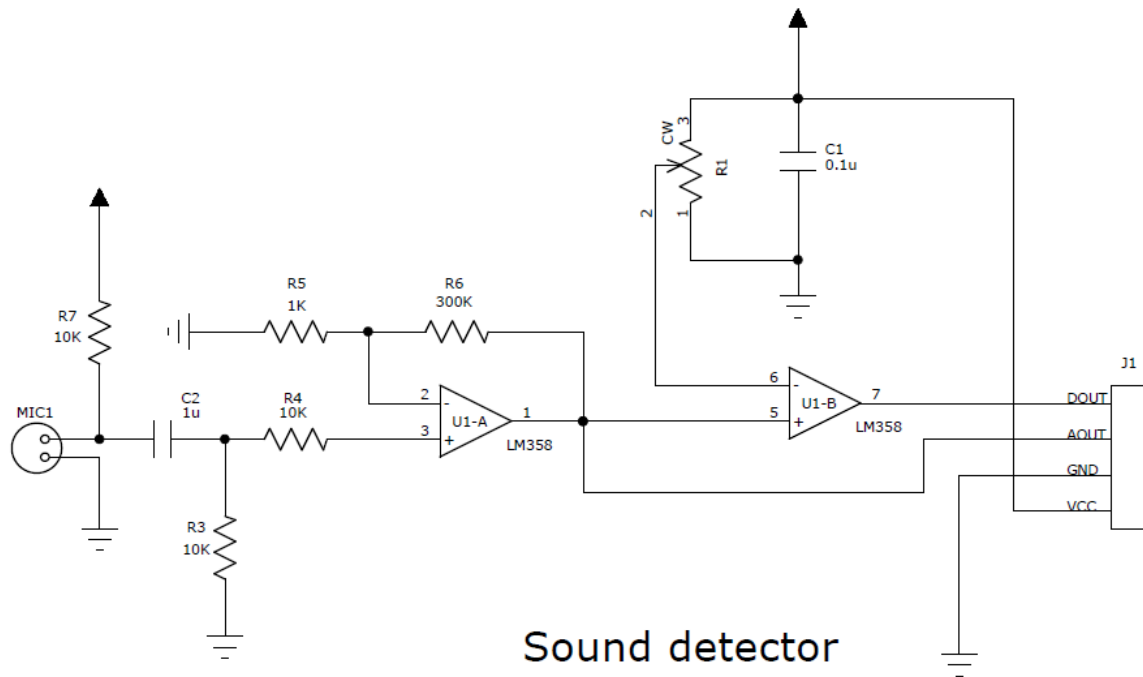


Figura 40 – Esquemático do módulo detector de som RobotDyn

Fonte: Acervo do fabricante : RobotDyn

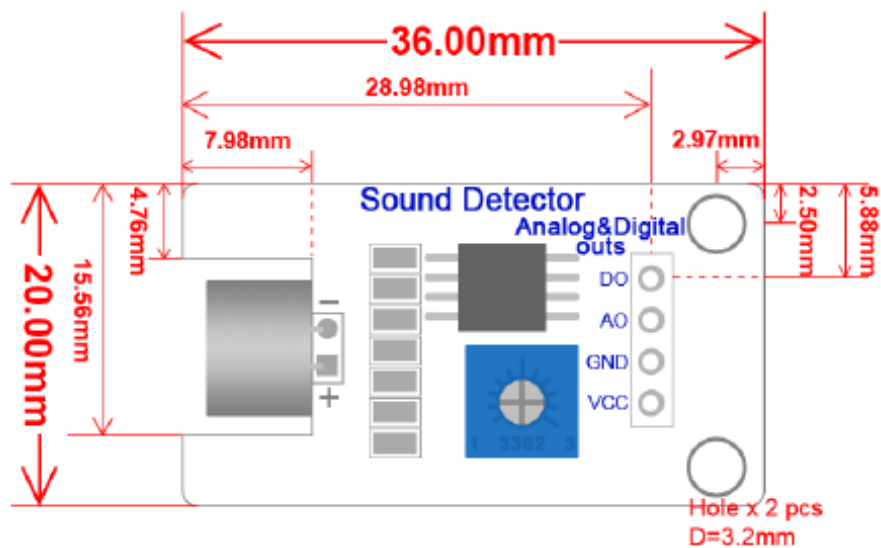


Figura 41 – Projeto mecânico do módulo detector de som Robotdyn

Fonte: Acervo do fabricante : RobotDyn

## APÊNDICE C – Estudo do Funcionamento dos Microfones

### C.1 Microfone Capacitivo

O microfone capacitivo, também chamado de condensador, é um dispositivo que funciona com o movimento de um diafragma com relação a um plano fixo, que produz uma capacitância correspondente a esta movimentação. Um circuito elétrico fica responsável por transformar essa variação de capacitância em tensão.

O dispositivo é constituído de um isolante elétrico e um prato montado como um diafragma paralelo, dentro de um encapsulamento metálico (Figura 42). Um capacitor ativo é formado pelo diafragma e o prato e a capacitância é alterada quando a placa do diafragma é deslocada pela pressão sonora. Os valores são dados pelas seguintes expressões:

$$Q_0 = E * C \quad (8)$$

$$(E_0 + e) * \left(\frac{\epsilon * A}{D_0 + d}\right) = E_0 * \left(\frac{\epsilon * A}{D_0}\right) \quad (9)$$

$$e = E_0 * \frac{d}{D_0} \quad (10)$$

Sendo  $Q_0$  a carga constante da placa do capacitor,  $E$  a tensão elétrica instantânea entre as placas,  $C$  a capacitância instantânea entre as placas,  $E_0$  a tensão elétrica de polarização,  $e$  o valor instantâneo da pressão de saída,  $\epsilon$  a constante dielétrica do ar,  $A$  a área da placa do capacitor,  $D_0$  a distância entre as placas no repouso e  $d$  o deslocamento do diafragma [BALBINOT; BRUSAMARELLO](#).

### C.2 Microfone Dinâmico

O funcionamento do microfone dinâmico ocorre da seguinte forma: uma membrana leve tem a propriedade de vibrar de acordo com o som que incide sobre ela, essa membrana é acoplada à uma bobina móvel que é submetida à um campo magnético forte, gerado por um ímã redondo (Figura 43). Quando o som incide neste conjunto, o conjunto se movimenta, variando o fluxo magnético na bobina, o que produz uma corrente elétrica proporcional à variação de pressão no diafragma ([VALLE, 2002](#)).

A sensibilidade deste tipo de microfone é considerada baixa pois ele tende a ignorar sinais de baixa intensidade e a distorcer sinais de alta intensidade. Por outro lado, é um microfone de fácil operação, pois não precisa com uma fonte de alimentação externa para funcionar ([VALLE, 2002](#)).

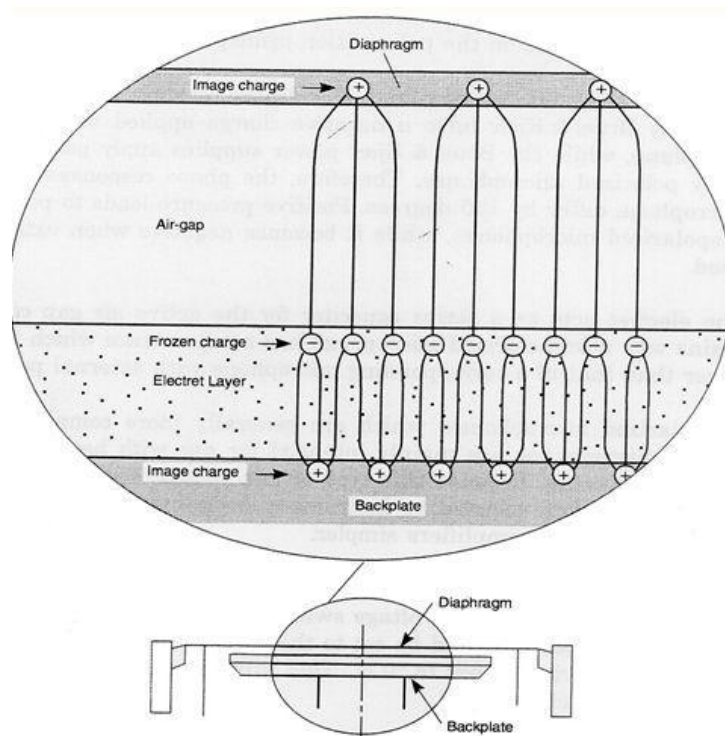


Figura 42 – Princípio do microfone capacitivo pré-polarizado com eletreto

Fonte: [BALBINOT; BRUSAMARELLO](#)

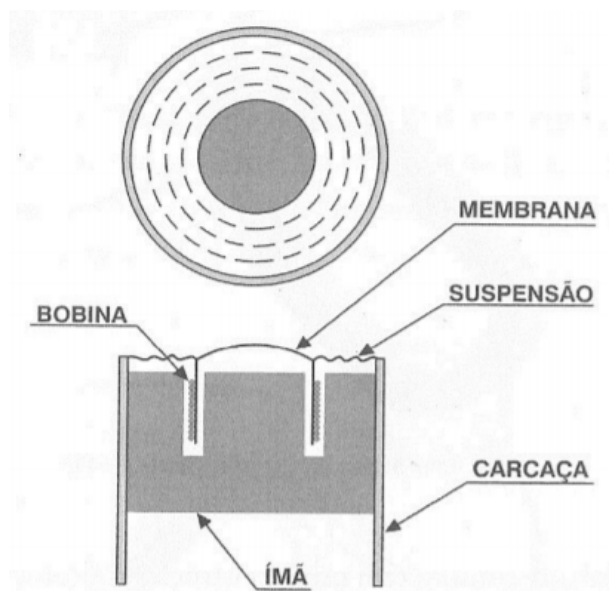


Figura 43 – Estrutura de um microfone dinâmico

Fonte: [Valle \(2002\)](#)

### C.3 Microfone Piezoelétrico

Os microfones piezoelétricos funcionam pela propriedade que alguns materiais têm de gerar tensão elétrica quando são deformados, assim, quando se acopla um desses materiais a uma membrana elétrica, tem-se um microfone. Esse microfone está entrando em desuso por oferecer uma baixa qualidade de som, mas o princípio ainda é muito usado para obter ruídos direto da fonte ([VALLE, 2002](#)).

## APÊNDICE D – Tecnologias Alternativas Consideradas Para o Projeto

### D.1 MICROFONES

#### D.1.1 Módulo detector de som Robotdyn

O módulo em questão já possui um amplificador integrado e circuito de polarização do microfone como mostra o [Apêndice B](#).

O circuito deste módulo permite adquirir um sinal analógico amplificado, bem como um sinal digital que resulta da saída de um comparador, ao qual tem seu *threshold* (valor de comparação) variável ao usuário. As tentativas de levantamento da curva de resposta em frequência deste módulo se deram inconclusivas, uma vez que o ganho de 300 vezes já montado ao circuito leva o sinal a saturação na tensão de alimentação (5V) mesmo a baixas intensidades sonoras.

#### D.1.2 Módulo detector de som KY-037

O segundo microfone testado foi o módulo detector de som KY-037. Um detector de som genérico com saída analógica não amplificada e sem polarização. Assim como o módulo anterior, também possui uma saída digital resultante de uma amplificação e comparação do sinal. A [Figura 44](#) mostra a resposta em frequência levantada.

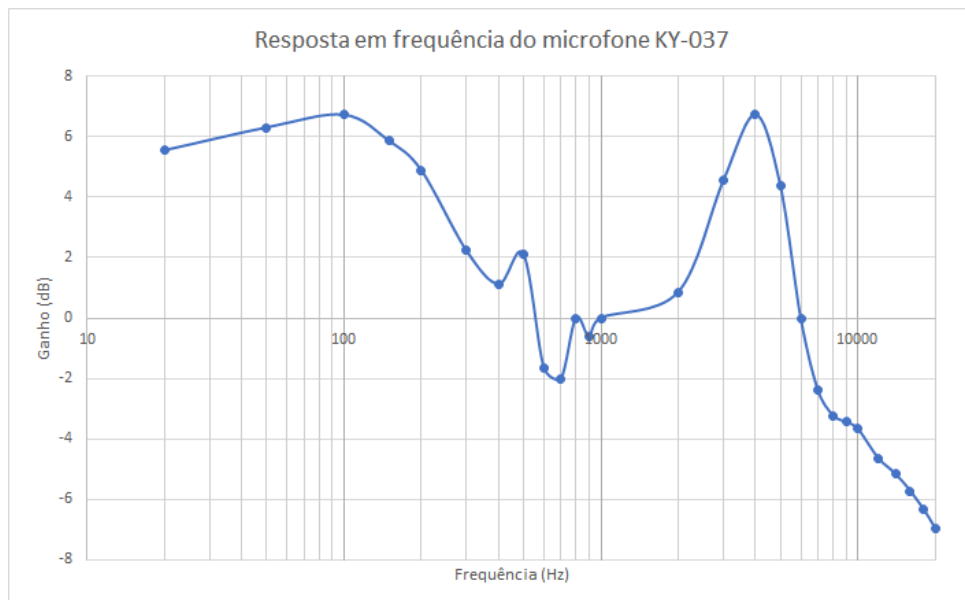


Figura 44 – Resposta em frequência do módulo KY-037

Fonte: Autoria própria



### D.1.3 Módulo de microfone MAX9814

O módulo de microfone MAX9814 possui ganho automático configurável de 20, 40 ou 60dB, impedindo a saturação do sinal ([MAX9814, 2018](#)). No entanto, o módulo adquirido não pode ser testado devido a atrasos no envio do exemplar adquirido.

## D.2 PLACAS DE DESENVOLVIMENTO

### D.2.1 Arduino Nano

O Arduino Nano é uma compacta placa de desenvolvimento *open-source* que inclui um processador ATmega328P, Entre suas especificações, estão ([ARDUINO, 2018](#)):

- Arquitetura AVR;
- Tensão de Operação de 5V;
- 32kB de memória *flash*;
- 2kB de memória RAM;
- Clock de 16 MHz;
- Pinos I/O analógicos (8) ;
- EEPROM de 1 KB;
- Corrente DC nos pinos I/O de até 40 mA;
- Tensão de entrada de 7-12 V;
- Pinos de I/O digitais (22);
- Saída PWM;

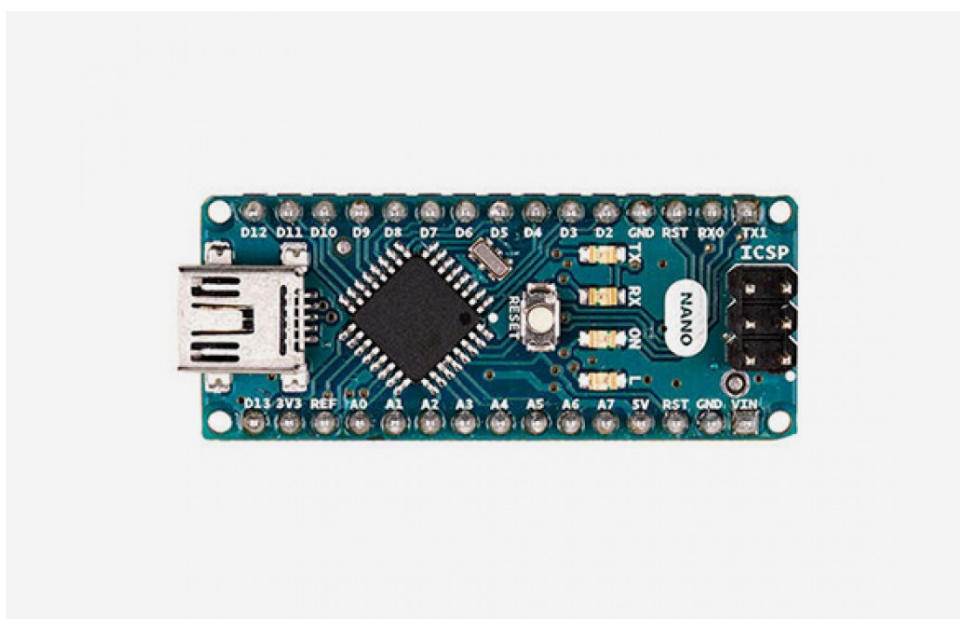


Figura 45 – Placa de desenvolvimento Arduino Nano

Fonte: ([ARDUINO, 2018](#))

## D.2.2 FRDM-KL25Z

Outra opção de placa de desenvolvimento existente no mercado é a FRDM-KL25Z. É considerada uma placa de baixo custo pelos recursos que ela traz. Dentre eles (MBED, 2017):

- Processador de 32 bits ARM Cortex M0+;
- Baixo consumo de energia;
- Interface de *debug* para a programação em memória *flash*;
- 128KB de memória *flash*;
- 16KB de memória RAM;
- USB (*Host/Device*);
- SPI (2);
- I2C (2);
- UART (3);
- PWM (TPM);
- ADC (16 bits);
- DAC (1x 12bits);
- Sensor de toque;
- GPIO (66);

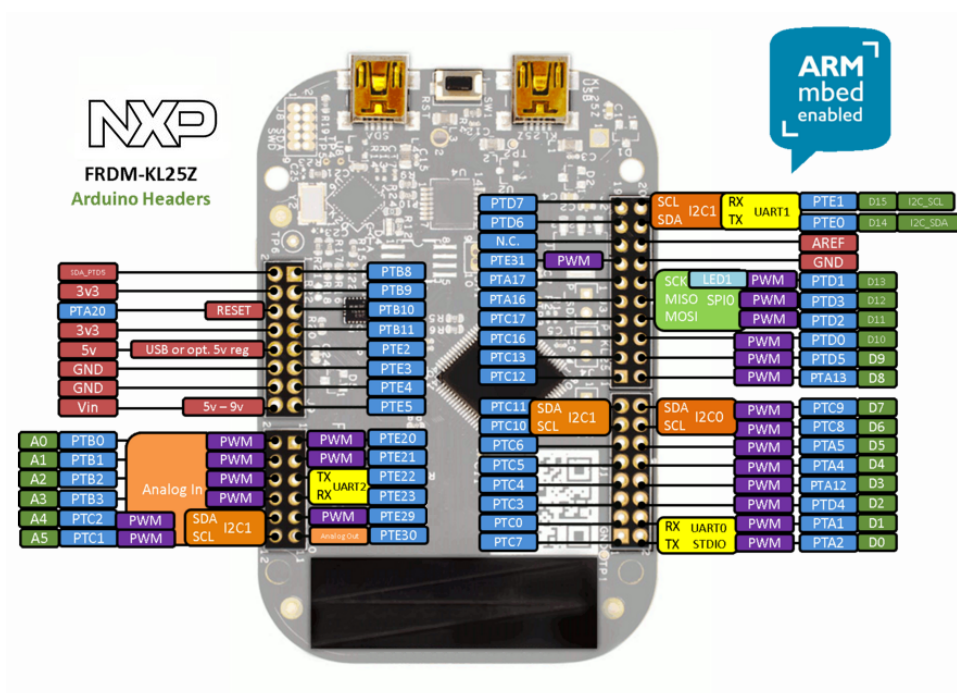


Figura 46 – Placa de desenvolvimento FRDM-KL25Z

Fonte: (MBED, 2017)

### D.2.3 GPS/GPRS AI-Thinker A7

O AI-Thinker A7 é um módulo de comunicação GPS/GPRS que traz um grande número de recursos, entre eles (CO.LTD, 2016):

- Faixa de temperatura entre  $-30^{\circ}$  à  $+80^{\circ}$
- Tensão de trabalho: 3.3V-4.2V;
- Tensão de alimentação: 3.4V;
- GPRS classe 10;
- Sensibilidade:  $<-105$ ;
- Suporta chamadas de voz;
- Suporta mensagens SMS;
- GPIO: 2,8v;
- Suporta duas portas serial;
- Suporta comandos TCP/IP;
- Suporta áudio digital e analógico;
- Suporta GPS+AGPS;

O módulo em questão não foi priorizado na implementação do projeto devido à falta de documentação e suporte de integração a outros sistemas por conta do fabricante.

## D.3 COMUNICAÇÃO SEM FIO

Dentre as tecnologias consideradas, pode-se citar:

- *Wi-Fi*;
- GPRS;
- LORA;
- Zigbee

Dos quais, o primeiro foi selecionado como solução a ser implementada.

### D.3.1 ESP8266

O ESP8266 é um módulo que oferece uma solução de comunicação *Wi-Fi* e conta com as seguintes características (SYSTEMS, 2013):

- Protocolo 802.11 b/g/n;
- *Wi-Fi Direct* (P2P), soft-AP;
- Protocolo integrado TCP/IP;
- Amplificador integrado;
- PLL e reguladores integrados;
- Saída de +19.5dBm;
- Sensor de temperatura integrado;
- Suporte para diversas antenas;
- Processador de 32-bits;

- SDIO 2.0, SPI, UART;
- Transmite pacotes em  $< 2\text{ms}$ ;

### D.3.2 GPRS

A comunicação GPRS foi uma das primeiras opções consideradas para o projeto, graças ao seu longo alcance e possibilidade de comunicação TCP/IP, podendo, portanto enviar dados diretamente ao banco. No entanto, esta solução demanda um custo junto a uma operadora de serviços de telefonia móvel para operar, aumentando os custos de projeto e produto final. Além disto, esta tecnologia demanda um consumo de energia mais elevado em comparação com as demais descritas na presente sessão ([WIFI... , 2018](#)), sabendo que as tecnologias LoRa e Zigbee foram desenvolvidas para baixo consumo de energia ([LORA... , 2018b](#)).

### D.3.3 LoRa

LoRa é uma solução de comunicação por RF em baixas frequências (433, 868 e 915MHz), obtendo baixas taxas de transmissão (de 0.3 a 100kbps)([LORA... , 2018b](#)) e longo alcance (até 10 milhas ou 16km) ([LORA... , 2018a](#)). No entanto, é necessário implementar um transmissor e um receptor para que um sistema baseado em Lora funcione. Ainda assim, o envio das informações ao banco de dados requer uma implementação de conexão à Internet (um módulo *Wi-Fi*, por exemplo), à qual o receptor LoRa deveria ser integrado. Sendo assim, optou-se por reduzir passos do sistema e a necessidade de implementação de diversos pontos de recepção. Assume-se que áreas urbanas já possuem implementadas uma rede de pontos de acesso *Wi-Fi*.

### D.3.4 Zigbee

A tecnologia Zigbee foi a última considerada na questão de comunicação sem fio. Sendo uma solução de baixo consumo de energia e alcance de cerca de 500m ([BALANCING... , 2006](#)), à primeira vista, o Zigbee parece ser a solução ideal para este projeto. No entanto, o custo da tecnologia ainda é elevado, em alguns casos, um módulo de desenvolvimento Zigbee chega a custar cerca de 10 vezes mais em comparação ao NodeMCU utilizado neste trabalho. Assim como o LoRa, o ZigBee ainda não possui uma rede disseminada o suficiente para tornar viável o seu uso.