

CRIAÇÃO DE UM MEDIDOR DE RUÍDO DE BAIXO CUSTO PARA UTILIZAÇÃO EM FÁBRICAS.

Lucas Santos Silva

Graduando em Sistemas de Informação

Ely Fernando do Prado

Mestre em Ciência da Computação

RESUMO

A construção de um medidor de ruídos tem sido motivada devido ao número de pessoas que apresentam problemas de audição oriundas de seu respectivo trabalho. Esse problema, então fomenta o desenvolvimento de novas tecnologias para melhorar a vistorias dentro dessas áreas. Neste sentido, propõe-se o desenvolvimento de um medidor através do Arduino que seja capaz de enviar um relatório, com o intuito de ser mais uma ferramenta no apoio da questão de prevenir situações que possam contribuir para prejudicar a integridade auditiva. Sendo assim, o medidor conta com um sistema web mostrando os picos de ruídos em várias horas do dia, contando com um relatório ruído em decibéis para facilitar o acompanhamento dos sons emitidos no recinto. A construção dessa pesquisa, tem o intuito de chamar atenção sobre os quesitos de poluição sonora e prevenção da perda de acuidade auditiva que impacta negativamente na vida do trabalhador e sua respectiva família.

Palavras-chave: Arduino, medidor de ruídos, audição.

1 INTRODUÇÃO

Muitas vezes o trabalho pode proporcionar uma série de riscos à saúde das pessoas, entre os quais temos o ruído, presente em ampla parte dos métodos de produção de fábricas. Sendo uma exposição possível de medição e de controle, havendo, todavia, vulnerabilidade em relação a monitorização em tais locais de trabalho, de maneira especial nos países em desenvolvimento MEIRA, Tatiane Costa et al (2012). Segundo a Organização Mundial da Saúde (OMS) o ruído coloca-se em terceiro lugar no ranking das causas ocupacionais que mais geram incapacidade em trabalhadores.

Sendo assim, foi manifestada a ideia de desenvolver um medidor de ruídos de baixo custo utilizando o Arduino para tal propósito, sendo este uma placa composta por um microcontrolador, que foi pensada com o intuito da resolução de problemas com diversas opções de uso. Neste projeto foi desenvolvido um equipamento que solucione a seguinte questão: Como desenvolver um medidor de ruídos utilizando Arduino capaz de gerar relatórios a partir dos dados coletados?

O trabalho se mostra relevante, pois pretende-se criar um medidor de ruído usando Arduino para ser utilizado em fábricas que possuem alto índice de ruído com a emissão de relatórios que mostrará os picos de maior evento de ruídos para auxiliar na prevenção de perda de audição dos trabalhadores. Este tema foi escolhido devido ao índice crescente de problemas auditivos segundo a pesquisa de Nelli et al (2015), em que foram avaliados 53 trabalhadores que trabalham expostos a alto índice de ruídos, com aplicação de questionário e análise de exames audiométricos por meio dos resultados dessa pesquisa verificou-se que três funcionários apresentaram perda auditiva induzida por ruído. O intuito de se utilizar o Arduino para solucionar este problema é avaliar se de fato é possível construir um aparelho de baixo custo que consiga auxiliar na prevenção de problema de audição dos trabalhadores e que ainda possa se integrar a um sistema web para posterior emissão de relatórios.

2 REFERENCIAL TEÓRICO

Nesta seção são apresentados os principais conceitos teóricos acerca desta pesquisa, incluindo informações sobre a saúde auditiva, o microcontrolador Arduino e também uma análise de trabalhos correlatos.

2.1 Saúde auditiva

De acordo com a pesquisa Zanon et al. (2016), a exposição excessiva ao alto índice de ruído no ambiente onde se trabalha é um dos agentes mais nocivos à saúde do trabalhador. Os resultados desta exposição podem-se aplicar não só a perda auditiva, mas também a vários outros fatores como a mudança de humor, incapacidade de concentração, dificuldade na compreensão da fala, zumbido no ouvido, tonturas, dores de cabeça frequentes, incapacidade de identificar a origem de fontes sonoras, podendo prejudicar no trabalho, na comunicação e até mesmo no convívio social.

De acordo com Páscoa et al. (2015) na maioria das vezes nem todo ruído pode ser considerado como poluição sonora. O ruído é o conjunto de vibrações que percorrem pelo ar que são causadas por alterações na pressão. Um ruído de 50 decibéis já pode ser considerado indesejável. A exposição a sons maiores como 65 decibéis ou ruídos agudos entre 80 a 85 decibéis pode causar problemas no sistema nervoso. Páscoa et al. (2015) reforça que é recomendável ouvir sons apenas com máximo entre 65 e 70 decibéis, sendo considerado este um limite tolerável para não causar danos. Porém nem sempre é possível o controle do ruído que ouvimos, de modo que há certa flexibilidade ao considerar um som suportável, podendo variar entre 20 a 120 decibéis. Acima de 120 decibéis, já se pode notar um desconforto muito forte, e acima de 130, a pessoa já chega ao limiar da dor.

Segundo Santos e Schuartz et al. (2010) a elevação do som é constituída pela sua frequência. Sons elevados são os quais que mostram maiores frequências, também denominado de sons agudos. Os fenômenos acústicos baixos são aqueles que mostram baixas frequências, sendo chamados de sons graves. A intensidade do som diz respeito à quantidade de energia que a onda sonora transmite. Essa intensidade relacionada à amplitude da onda sonora: quanto maior sua amplitude, maior será sua intensidade. Essa

intensidade é medida em decibéis: ruídos intensos são chamados de ruídos fortes, e os ruídos de baixa intensidade são chamados de ruídos fracos.

De acordo com Cunha e Côtes et al. (2019) a melhor forma de prevenção da perda auditiva seria o uso de equipamentos de proteção individual (EPIs). Se a função que o trabalhador exerce oferece riscos ambientais relacionados à audição, é de total importância fazer o uso de protetores auriculares ou abafadores de som durante o seu período de trabalho. É importante também que se façam medições frequentes através de equipamentos eletrônicos a fim de avaliar a intensidade de ruído presente nos locais de trabalho. Este tipo de aparelho é conhecido como decibelímetro.

2.1.5 Microcontrolador Arduino

Segundo Fabio Costa et al. (2020) a criação do Arduino iniciou na cidade de Ivrea, Itália em 2002, o qual foi projetado no intuito de interagir em projetos acadêmicos com menor custo se comparado com outros equipamentos para prototipagem disponíveis naquele tempo.

O Arduino é uma plataforma de prototipação eletrônica *open-source* que permite criar vários equipamentos eletrônicos e também conta com uma comunidade colaborativa que pode unir pessoas ao redor do mundo, pode-se entender que o Arduino é uma placa dinâmica e pode ser utilizado de várias formas, basta ter uma ideia e saber programar de acordo com a lógica de funcionamento da ideia, ela é composta de uma placa de entrada/saída em um ambiente de desenvolvimento simples que implementa a linguagem *Processing*. Ele apresenta a capacidade de interagir com o mundo físico, seja captando variações ao seu redor ou enviando sinais de controle. Devido a essa importante característica e a relativa facilidade de manuseio, resolveu-se utiliza-lo neste projeto. (ARDUINO, 2019).

2.1.3 Trabalhos correlatos

Santos et.al (2010) descrevem a construção de um decibelímetro para a medição de intensidade sonora, descrevendo os procedimentos e materiais necessários. O projeto é focado na utilização de tecnologias livres para construção de um protótipo, sendo utilizado, portanto, o microcontrolador Arduino. O projeto apresenta uma alternativa de baixo custo comparado a outros decibelímetros comerciais, fazendo leituras da

intensidade sonora e exibindo este valor em um display LCD. Apesar dos autores terem relatado dificuldades na construção e apresentação do artefato, o trabalho contribui como forma de tutorial para construção de aparelhos semelhantes. É importante destacar que devido a problemas na fórmula de conversão da intensidade dos ruídos, o valor mostrado no display LCD não ficou condizente com o valor real medido por um decibelímetro comercial, deixando dúvida se de fato é viável a construção de um aparelho semelhante.

Salles (2017) também elaboraram um projeto com a intenção de desenvolver um decibelímetro para a medição de ruído usando componentes e dispositivos eletrônicos de baixo custo para ser usado no monitoramento de ruído em fábricas. No desenvolvimento foram avaliadas duas formas de construção, sendo uma com base na plataforma Arduino e outra com base no sistema *openWrt* (é uma distribuição Linux embarcada em dispositivos tais como roteadores pessoais) em conjunto com um roteador. Os circuitos de captação do som consistem em componentes de captação de sinais já existente nas lojas comerciais, identificando o que melhor se adaptou a proposta. O sistema de medição possui a proposta de poder ser operado remotamente, de forma que o operador possa controlar as medições sem estar no local fisicamente. Os resultados mostraram que o dispositivo com base no sistema *OpenWrt* apresentou um melhor desempenho, especialmente, em relação sua taxa de amostragem que é maior, não precisando de ajustes. As informações capturadas pelo sensor podem consultadas remotamente em tempo real, mas não foi realizado um estudo sobre o armazenamento destes dados em um banco de dados para consultas futuras.

Vianna e Lima (2014) também desenvolveram um decibelímetro utilizando Arduino para medir a intensidade de ruídos como forma de realizar um monitoramento ambiental coletando e analisando dados sobre a poluição sonora. Foram identificadas dificuldades em descobrir técnicas de medição de ruído de baixo custo, porém, com alto precisão. Diante disto, desenvolveu-se neste estudo decibelímetro capaz de captar e transmitir informações em tempo real a um baixo custo. O sensor ambiental utilizado foi o de nível de pressão sonora, que transmite informações sobre o ruído para um sistema web desenvolvido em PHP. Porém o estudo não apresenta qual foi o formato de exibição de dados adotado e nem mesmo esclarece se este projeto é capaz de proporcionar ao usuário consulta do histórico dos dados informando o período desejado.

Costa e Fonseca (2018) buscaram uma solução para a captura de dados sonoros que possa auxiliar no direcionamento de esforços e soluções, humanas e financeiras, para sanar o problema da poluição sonora e melhorar as relações dos seres humanos com a o espaço urbano. O projeto consiste em integração de software online aliado a um hardware 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 desenvolvida e resultados apresentados para aplicações de associação e visualização de dados na forma de mapas de calor. Ao contrário dos trabalhos anteriores aqui apresentados, ao invés de utilizar o microcontrolador Arduino, optou-se pela utilização do NodeMCU, pois este já vem integrado com módulo wifi, além de ter sido acoplado ao circuito um módulo de GPS para capturar a posição geográfica da captura do sinal. A proposta deste protótipo é espalhar vários destes dispositivos em uma cidade para capturar informações sobre índices de poluição sonora.

Apesar de existirem pesquisas sobre o desenvolvimento de decibelímetros de baixo custo, percebe-se que ainda é necessário mais estudo sobre o tema a fim de avaliar alternativas que consigam trazer uma precisão satisfatória a um preço conveniente. Além disto, os projetos analisados não disponibilizam o armazenamento dos dados coletados em um banco de dados e posterior exibição destes dados em relatórios que facilitem a utilização e o acesso por profissionais que cuidam da saúde no trabalho para fábricas. O projeto que mais se aproxima da proposta deste contexto é o apresentado por Costa e Fonseca (2018), porém não foi analisado o escopo de uso em ambientes fechados, como fábricas, e nem explorada a criação de relatórios que fossem voltados para profissionais que trabalham diretamente analisando os riscos à saúde do trabalhador.

3 MATERIAIS E MÉTODOS

O projeto proposto utiliza o método de pesquisa exploratória, o qual busca investigar formas para se construir um medidor de ruído com capacidade de realizar leituras de intensidade sonoras para posterior visualização dos dados em um sistema web. Desta forma propõe-se o desenvolvimento de um protótipo composto das seguintes partes: aquisição do sinal, ampliação do sinal lido, conversão de analógico

para digital, transformações matemáticas de tensão elétrica para decibéis e visualização do resultado através do sistema web que enviará relatórios de picos de ruídos para gestores de empresas que precisem dessa informação para auxiliá-los a prevenir acidentes de trabalho.

Conforme analisado em trabalhos correlatos, percebe-se que as tarefas de aquisição, ampliação do sinal, conversão de analógico para digital são tarefas relativamente fáceis de serem resolvidas. Contudo o armazenamento destas informações e posterior visualização dos resultados é algo que exige um estudo mais detalhado, especialmente quanto ao formato de exibição dos dados coletados. Portanto um ponto fundamental neste projeto é o desenvolvimento de um sistema web capaz de exibir os dados coletados pelo decibelímetro de maneira que seja útil para profissionais de saúde do trabalhador.

3.1 Entrevistas e pesquisa de campo

Foi entrevistado um profissional de segurança do trabalho para a definição dos requisitos do sistema que foram desenvolvidos neste projeto. Foi utilizada a técnica de entrevista aberta, em que não havia uma programação prévia das perguntas a serem realizadas, deixando a entrevista mais informal e aberta a novas informações. Esta entrevista foi bastante útil para entender as necessidades do stakeholder e definir os requisitos do sistema. De forma geral foi constatado nesta entrevista que um sistema capaz de mostrar os dados coletados sobre intensidade de ruído ao longo de um determinado período seria muito útil, pois permite que o profissional de saúde possa fazer análises sem estar necessariamente presente na fábrica todo o período de tempo da medição.

3.2 Definição dos requisitos do sistema

De acordo com Sommerville (2011), os requisitos funcionais descrevem quais são as funções que um sistema deve realizar e são úteis para compreender a necessidade dos usuários. Foram identificados os seguintes requisitos funcionais:

- Cadastrar usuários: o sistema deve permitir o cadastro de usuário para acessar o sistema e consultar seus dados;

- Gerar relatório sobre ruído: o sistema deve gerar relatório sobre a intensidade sonora ao longo de um determinado período informado pelo usuário;
- Medir nível de ruído: o sistema deve medir o nível de ruído através de sensores;
- Enviar dados para o servidor: o sistema deve automaticamente enviar os dados coletados sobre a intensidade sonora para um servidor a fim de disponibilizar estes dados para consulta remota;

Requisitos não funcionais são relacionados às características do sistema em definições de usabilidade, desempenho, segurança, confiabilidade, e que não estão diretamente ligadas com os serviços disponibilizados pelo sistema ao usuário, e sim ao funcionamento interno do sistema (SOMMERVILLE, 2011). Desta forma, foram identificados os seguintes requisitos não funcionais:

- Apenas usuários autenticados no sistema poderão consultar o relatório sobre ruído;
- O sistema deverá funcionar através da internet;
- A captura de dados deverá ser realizada utilizando um microcontrolador Arduino;

3.3 Modelagem do Sistema

Após a definição dos requisitos, foi realizada a modelagem do sistema através do diagrama de casos de uso e também foi modelado o banco de dados para persistência e consulta dos dados.

O sistema foi modelado por meio do UML (*Unified Modeling Language*), mais precisamente o diagrama de casos de uso. O diagrama de casos de uso é capaz de descrever o cenário que mostra as funcionalidades do sistema sob a perspectiva do usuário (ator). De acordo com Booch (2012), o diagrama de casos de uso tem um papel central para a modelagem do comportamento do sistema, subsistema ou classe. Com ele, é possível mostrar ao usuário as principais funcionalidades do sistema. Na Figura 1 é mostrado o Diagrama de Casos de Uso do Sistema desenvolvido neste projeto.

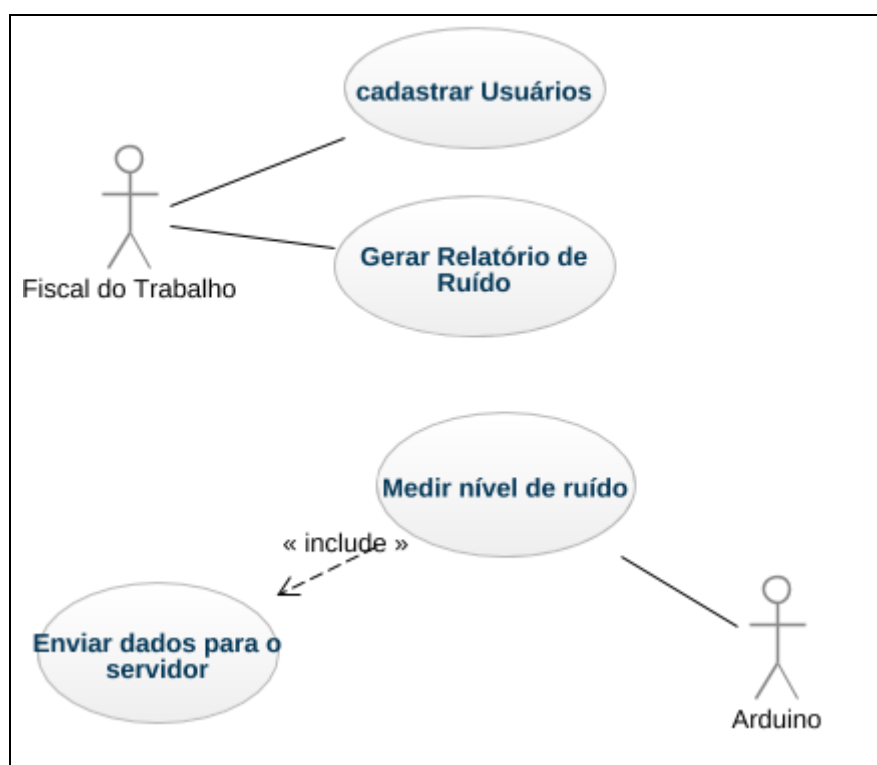


Figura 1: Diagrama de caso de uso. Fonte: Elaborado pelo autor.

Para projetar o banco de dados feita a modelagem do diagrama entidade e relacionamento. Neste diagrama é possível ilustrar como as entidades vão se relacionar entre si dentro do sistema. Na Figura 2 é apresentado o diagrama contendo as tabelas, colunas, além do relacionamento entre as tabelas do banco, sendo representado graficamente o banco de dados responsável pelo gerenciamento dos usuários e os dados coletados pelos sensores.

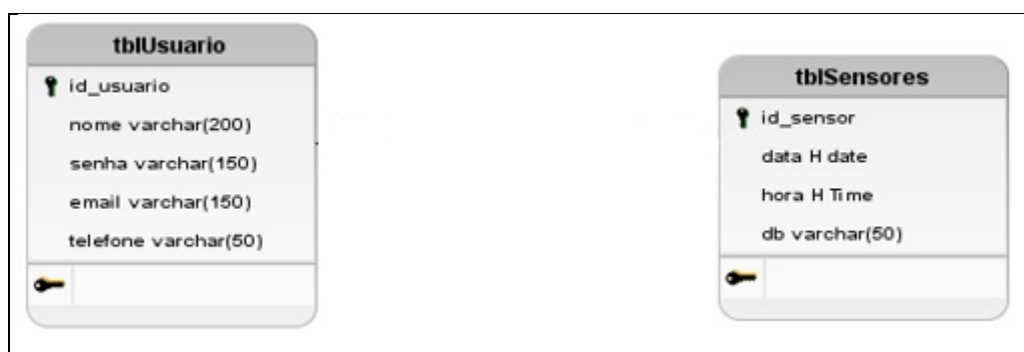


Figura 2: Representação gráfica do banco de dados dos usuários. Fonte: Elaborado pelo autor.

3.4 Aquisição do sinal

Para transformar um sinal sonoro em um valor de tensão utilizou-se o módulo sensor de som KY-038, encontrado facilmente em lojas de componentes eletrônicos a um baixo custo. Este módulo é muito utilizado devido ao seu tamanho reduzido e alta sensibilidade. Para funcionar corretamente, ele necessita de uma alimentação de 3 a 24 volts e consome aproximadamente 1mA de corrente elétrica.

O sensor KY-038 mostrado na Figura 3 é capaz de medir a intensidade sonora do ambiente ao seu redor, variando o estado de sua saída digital caso detectado um sinal sonoro. Ele possui um microfone condensador elétrico e o seu limite de detecção pode ser ajustado através do potenciômetro presente no sensor que regula a saída digital D0.



Figura 3: Sensor LCD. Fonte: <http://www.arduino.com>

3.5 Conversão analógico para digital

O Arduino trabalha com uma frequência de 16 MHz, porém seu circuito conversor A/D (analógico para digital) trabalha em uma frequência menor, podendo o usuário escolher um fator de divisão entre 2 e 128, fornecendo assim uma frequência de amostragem, que na prática fica entre 8 e 125 kHz, aproximadamente.

Nesse projeto foi utilizado um fator de divisão de 16, resultando em uma frequência de amostragem de 50 kHz. Essa forma foi escolhida pelo fato que a taxa de amostragem deve ser igual ou maior que duas vezes a frequência do som a ser amostrado, de acordo com a SANTOS, Nuno Pessanha (2009, p. 67). Considerando o limite da audição humana entre 10 e 22 kHz, a taxa de amostragem deve ser de no mínimo 44 kHz, resultando na escolha de 50 kHz.

3.6 Programação do Arduino

O software de controle do Arduino foi desenvolvido de maneira que consiste em ler um valor da saída do conversor A/D, converter em um equivalente em dB e enviar para o sistema web. Para isso, o código foi escrito na linguagem C++, compilado e enviado ao Arduino pela interface USB, sendo utilizado o programa de edição do próprio Arduino, compilação e comunicação fornecido pela empresa fabricante do microcontrolador.

A Figura 4 apresenta a primeira montagem do circuito do projeto, medindo o som do ambiente e exibindo no display LCD o valor em decibéis.

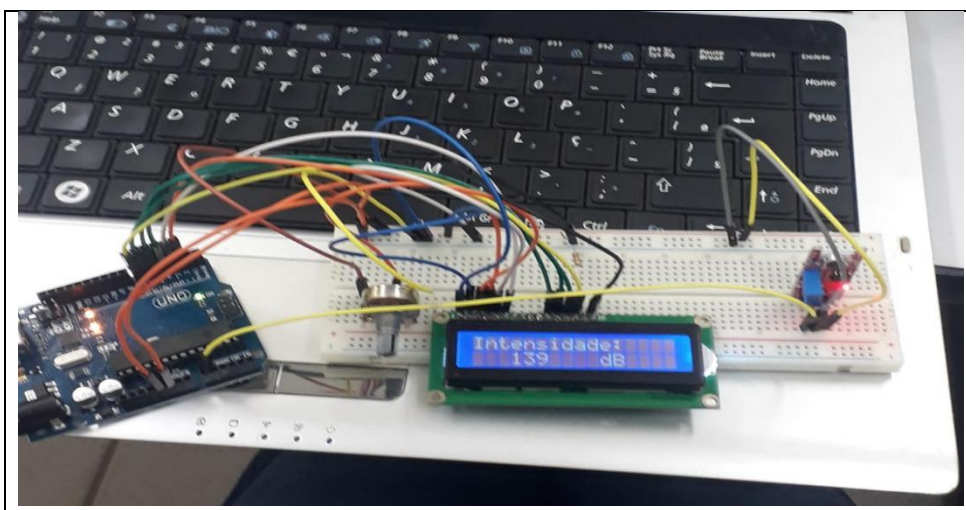


Figura 4: Circuito do Projeto. Fonte: Elaborado pelo autor.

A Figura 5 apresenta o projeto com sua montagem sem o display LCD, para que fosse colocado o Ethernet Shield w5100 para conexão com a internet a fim de enviar seus dados para o servidor.

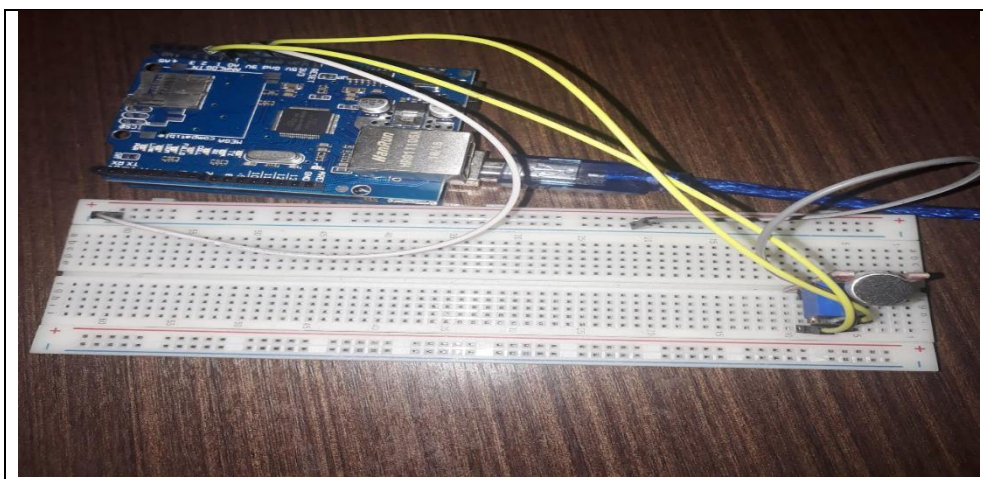


Figura 5: Circuito do Projeto. Fonte: Elaborado pelo autor.

O projeto foi codificado utilizando a IDE do Arduino na Linguagem C++. Nos Apêndices A e B. São apresentadas as partes do código responsável por medir o som do ambiente em decibéis, o qual é exibido no sistema web, nele está o cálculo matemático para a conversão de intensidade sonora para ser calculado em decibéis.

3.7 Desenvolvimento da Aplicação Web

Após processado e convertido, o sinal deve ser mostrado ao usuário através de um sistema web. O sistema web deve possuir um gráfico para mostrar o maior pico de decibéis durante o período que estão sendo captadas as informações, as que são armazenadas no banco de dados e exibidos através de um relatório, assim mostrando ao usuário do sistema os decibéis que está sendo lido pelo Arduino. Nesta página ainda pode-se escolher o período de datas que deseja gerar este relatório. Para o desenvolvimento da interface, foram utilizadas as seguintes tecnologias: HTML5 para construção da estrutura das páginas do sistema; PHP para a programação de interações do usuário com o sistema, com o intuito de deixá-las mais dinâmicas; CSS e o framework Bootstrap para definir a aparência da interface das páginas. Além disso, foi criado o banco de dados através do gerenciador de bancos de dados MySQL.

No HTML cada elemento (TAG) tem sua forma de abertura de comando e sua forma de fechamento, indicando assim, os limites dos comandos. Essa linguagem funciona em todos os diferentes dispositivos e também plataformas, e é interpretada por qualquer navegador disponível no mercado. (LUBBERS, 2013).

A lógica do sistema e a integração com o banco de dados foram desenvolvidos usando a linguagem de programação PHP. O qual permite manipular todo o conteúdo das páginas web no servidor, de imediato, antes mesmo da página ser enviada ao navegador do usuário. O funcionamento se baseia em um *script* PHP sendo executado no servidor, e pode alterar ou gerar códigos HTML sem restrições. (BEIGHLEY, 2013).

Foi utilizado no projeto o framework Bootstrap para melhor apresentação dos componentes deixando o site mais bonito visualmente permitindo que o site fique responsivo para facilitar o manejo do sistema.

O MySQL foi usado neste projeto para implementação do banco de dados. Ele foi escolhido por ter um excelente desempenho, por ser estável e também pela facilidade de uso. De acordo com Manzano (2011), o MySQL é um SGBD (Sistema Gerenciador de Banco de Dados) utilizado por grandes empresas como NASA, Silicon Graphics, Motorola, Yahoo, devido a sua escalabilidade e flexibilidade, alto desempenho e proteção de dados.

4. Resultados obtidos

Para avaliar a precisão dos dados coletados pelo sensor foram realizados testes com este componente. Nestes testes foram gerados ruídos para que o sinal fosse capturado por pelo sensor KY-38 e também por um decibelímetro comercial modelo Minipa Msl-1355, fornecendo então uma tabela que correlaciona a saída do sensor e nível sonoro em decibéis medido pelo decibelímetro, conforme mostrado na Tabela 1. Esta tabela contém a coluna (Saída do sensor Ky-38 (em dB)) que mostra os valores de decibéis que foram capturadas pelo sensor KY-38 e a coluna (Decibelímetro de referência (em dB)) que apresenta os valores em decibéis obtidos através de um decibelímetro comercial.

Tabela 1: Tabela de nível sonoro do sensor KY-38 e nível sonoro medido pelo decibelímetro de referência

Saída do sensor KY-38 (em dB)	Decibelímetro de referência (em dB)
60	64
55	60
80	75
40	47
47	50
38	30
58	56

Tabela 1. Fonte: Elaborado pelo autor.

O sistema web foi desenvolvido usando a linguagem de programação PHP para o backend, além do HTML para o frontend, juntamente com o CSS e o framework Bootstrap para a aparência do sistema. Para o armazenamento dos dados, foi usado o gerenciador de banco de dados MySQL.

Na página inicial após o usuário logar. Utilizando login e senha, é exibido uma interface intuitiva e simples sem muita poluição visual para simplificar o entendimento do sistema e também pode-se visualizar um campo que o usuário pode selecionar a data que ele deseja gerar o relatório de decibéis que foi capturado pelo Arduino. Além de exibir um gráfico, o sistema mostra as 10 primeiras captações de decibéis do dia que são atualizados no gráfico automaticamente a cada 15 minutos, no gráfico pode-se visualizar o maior decibéis e o menor do dia.

Na parte superior da tela, contém um navegador que é fixo em todas as páginas do sistema, o qual contém o menu para a navegação, podendo assim utilizar todas as funcionalidades do sistema, tais como cadastrar usuários e ver demonstrativos de decibéis. Na Figura 6 é mostrada a tela com as funcionalidades citadas.

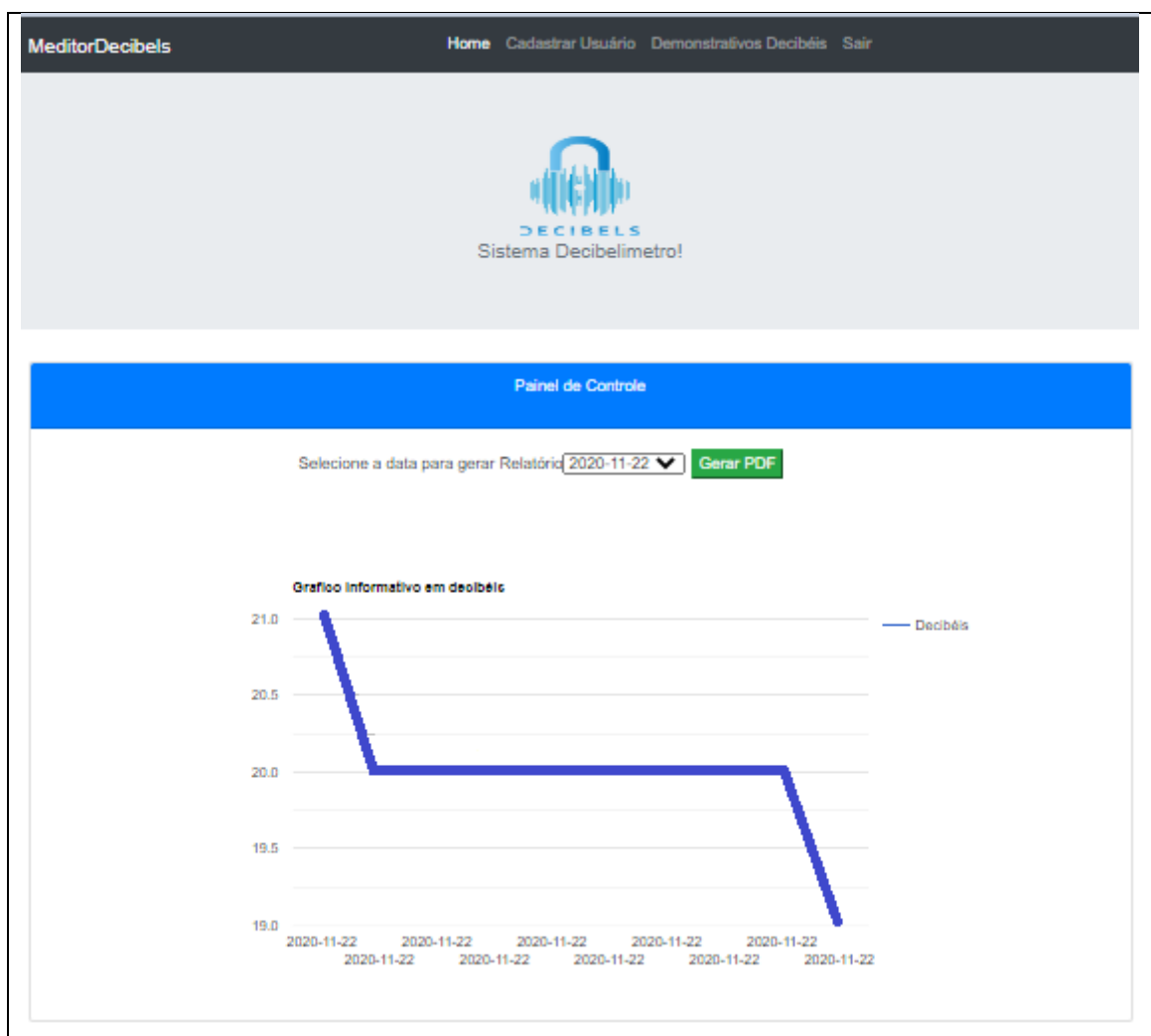
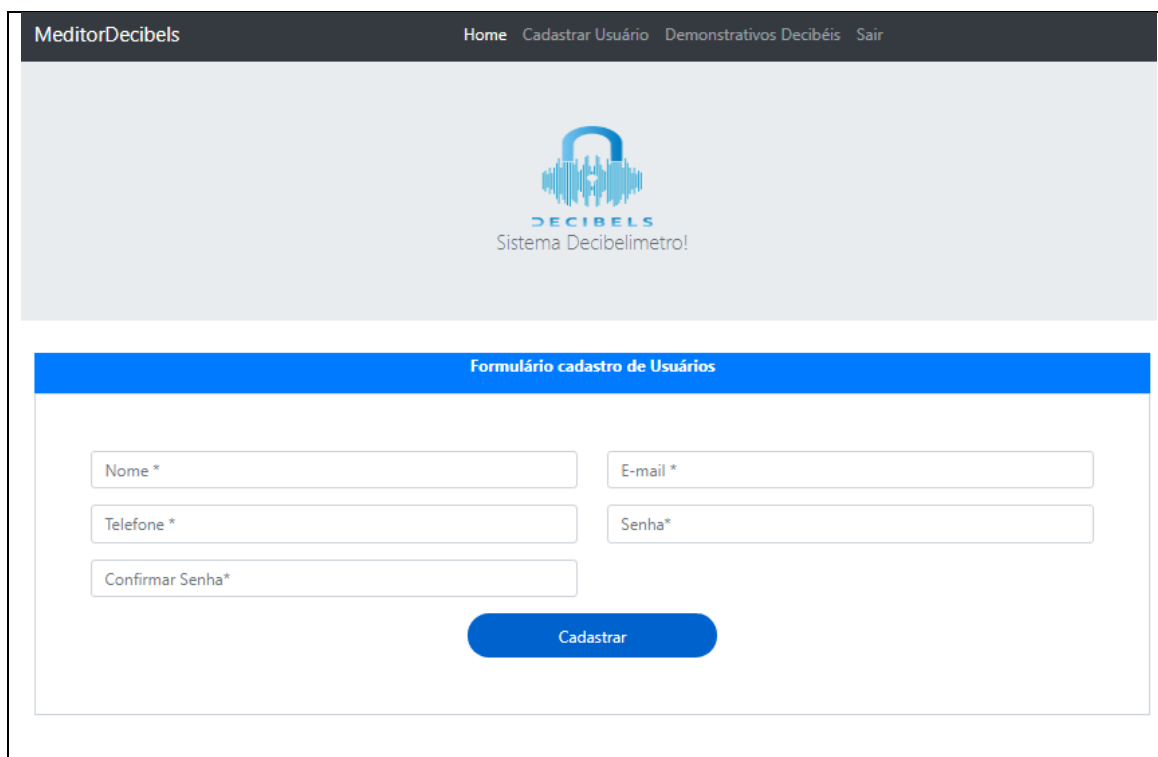


Figura 6: Representação da página inicial do sistema. Fonte: Elaborado pelo autor.

Como anteriormente citado, o navegador que fica na parte de cima da tela, contém a opção que é possível realizar o cadastro de novos usuários. Nessa tela, é possível cadastrar dados pessoais como nome, e-mail e telefone.

Nesta página, existe uma validação em relação ao e-mail, sendo que caso o mesmo seja inválido, o sistema não cadastra o usuário e também se o usuário digitar a senha diferente do confirmar senha ele não é cadastrado. Na Figura 7 é exibida a tela de cadastro de usuários.



A imagem mostra a interface web do sistema 'MeditorDecibels'. No topo, há uma barra de navegação com links: Home, Cadastrar Usuário, Demonstrativos Decibéis e Sair. Abaixo, o cabeçalho principal apresenta o logo 'DECIBELS' com o subtítulo 'Sistema Decibelímetro!'. O formulário de cadastro, intitulado 'Formulário cadastro de Usuários', contém campos para: Nome *, E-mail *, Telefone *, Senha * e Confirmar Senha *. Um botão azul 'Cadastrar' está posicionado na base do formulário.

Figura 7: Representação da página de cadastro de usuários. Fonte: Elaborado pelo autor.

No navegador superior, clicando em Demonstrativo Decibéis, é encontrado uma lista contendo 10 capturas do sensor, mostrando a data, hora e o decibéis que foi gerado em determinado momento da captura. Esses dados são atualizados a cada 15 minutos modificando a lista. Na Figura 8 é exibida esta tela de Demonstrativo em Decibéis.

MeditorDecibels		
Home Cadastrar Usuário Demonstrativos Decibéis Sair		
		
Demonstrativos em Decibéis		
Data	Hora	Decibéis
15/03/2020	16:05:39	25
15/03/2020	16:05:33	26
15/03/2020	16:05:27	26
15/03/2020	16:05:20	27
15/03/2020	16:05:14	26
15/03/2020	16:05:08	25
15/03/2020	16:05:01	26
15/03/2020	16:04:55	27
15/03/2020	16:04:48	27
15/03/2020	16:04:42	27

Figura 8: Representação da página de Demonstrativos Decibéis. Fonte: Elaborado pelo autor.

O produto final foi testado com um decibelímetro comercial. Desta forma, o decibelímetro desenvolvido pelo autor passou por um processo de calibração, que foi da seguinte forma: O nível sonoro foi medido usando ambos, o comercial e o desenvolvido pelo autor, e o valor fornecido pelo primeiro serve de referência para validar o valor fornecido pelo segundo (o desenvolvido pelo projeto).

Dessa forma durante a calibração do decibelímetro pode-se perceber uma limitação do mesmo, com uma margem de erro que varia de 5 decibéis pra cima ou para abaixo em comparação ao decibelímetro comercial.

4.1 Tabela de custos

Um dos pontos importantes para se analisar neste trabalho é a avaliação dos custos para se construir o decibelímetro. A Tabela 2 apresenta dos dados das cotações realizadas em outubro de 2020 nas seguintes lojas:

- Filipeflop: <https://www.filipeflop.com/>
- Eletrônica Castro: <https://www.eletronicacastro.com.br/>
- Eletrogate: <https://www.eletrogate.com/>

Tabela 3: Custos para o desenvolvimento do decibelímetro

Item	Felipeflop	Eletrônica Castro	Eletrogate
Arduino uno	R\$ 59,99	R\$ 57,90	R\$ 56,80
Ethernet Shield w5100	R\$ 39,99	R\$ 43,90	R\$ 49,80
Sensor KY-38	R\$ 9,99	R\$11,50	R\$ 12,90
Componentes	R\$ 3,00	R\$ 3,20	R\$ 2,50
Total	R\$ 112,97	R\$ 116,50	R\$122,00

Fonte: O autor (2020).

Apesar dos decibelímetros comerciais não possuírem sistema web integrado, foram realizadas cotações de preço destes aparelhos a fim de se fazer uma comparação que comprove se de fato este trabalho apresenta um protótipo de baixo custo quando comparado aos produtos existentes no mercado. Desta forma foram realizadas em outubro de 2020 três cotações de decibelímetros existentes no mercado nas seguintes lojas:

- Guia de compra: <https://guiadecompra.org/>
- iTesT: <https://www.itest.com.br/>
- Akso: <https://loja.akso.com.br/>

Tabela 4: Custos de decibelímetros comerciais

Decibelímetros	Guia de compra	iTesT	Akso
Decibelímetro Minipa Msl-1355	R\$ 475,00	R\$ 498,00	R\$490,00
Decibelímetro CBRN08872	R\$155,00	R\$ 174,00	R\$ 158,00

Fonte: O autor (2020).

5 Considerações finais

A principal limitação do projeto foi o valor obtido de decibéis que contém uma margem de erro em relação ao valor real, medido com um decibelímetro comercial, porém esta faixa de erro não foi calculada. A montagem do aparelho não apresentou problemas e o sistema web integrado ao Arduino funcionou de maneira esperada com uma interface intuitiva e sem muitas funcionalidades, assim ficando de maneira simples e fácil de ser compreendida para até as pessoas mais leigas no assunto. O código utilizado para programar o Arduino não teve falhas na lógica.

Na demonstração do projeto utilizou-se um cabo USB alimentado pelo notebook, fornecendo a alimentação de energia ao circuito. Observou-se que este procedimento não é o ideal. A tensão fornecida pela saída USB não é garantida em 5v e apresenta oscilações assim afetando a medição dos decibéis, desta forma recomenda-se que utilize uma fonte de tensão separada para alimentar o circuito.

Apesar de várias dificuldades na construção e o desenvolvimento do sistema o objetivo foi alcançado e foi obtido conhecimento sobre o assunto, sendo este conhecimento essencial para futuros projetos envolvendo eletrônica. Este trabalho apresenta os materiais, o procedimento e as informações necessárias para que possa ser reproduzido e melhorado em projetos futuros por terceiros. Dentre estes projetos futuros, vale destacar melhorias na exibição de dados e na precisão dos dados coletados pelos sensores.

REFERÊNCIAS

ARDUINO. **Arduino**. 2019. Disponível em: <http://www.arduino.cc>. Acesso em: 25 out. 2019.

BEIGHLEY, Lynn; MORRISON, Michael. **Use a Cabeça! PHP & MySQL**. 2. ed. Rio de Janeiro: Alta Books, 2013.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML Guia do Usuário**. 2. ed. Rio de Janeiro: Elsevier Editora Ltda, 2012.

COSTA, Thalita Munique; FONSECA, Keiko Verônica Ono. **Sistema de detecção e mapeamento de níveis de ruído sonoro**. 2018. 83 f. TCC - Curso de Engenharia Eletrônica, Universidade Tecnológica Federal do Paraná, Curitiba, 2018. Disponível em: <http://repositorio.roca.utfpr.edu.br/jspui/handle/1/10128>. Acesso em: 25 out. 2020.

CUNHA e CÔRTEZ. Perda Auditiva Induzida Pelo Ruído Ocupacional. **HUMANIDADES E TECNOLOGIA (FINOM)**, v. 1, n. 16, p. 507-521, 2019. Acesso em: 23 de outubro de 2020.

SANTOS, Nuno Pessanha. **ESCOLA NAVAL**. Curitiba: Pessanha Santos, 2009. 69 p. Disponível em: https://www.isegi.unl.pt/docentes/vlobo/escola_naval/MFC/Tutorial%20Arduino.pdf. Acesso em: 25 out. 2019.

BOOTSTRAP. **BOOTSTRAP**. 2020. Disponível em: <https://getbootstrap.com/docs/4.3/about/overview/>. Acesso em: 14 mar. 2020.

COSTA, Fabio. **História Arduino**. 2020. Disponível em: <https://fabiocosta.net/arduino/historia-do-arduino/>. Acesso em: 21 out. 2020.

GRAPH. **GRAPH** graph. 2019. Disponível em: <http://www.padowan.dk/graph/>. Acesso em: 25 out. 2019.

COMPRA, Guia de. **Guia de Compra**. 2020. Disponível em:

<https://guiadecompra.org/como-escolher-o-decibelmetro-certo/#Conexao>. Acesso em: 17 set. 2020.

HTML.2020. Disponível em: < <https://html.com/> Acesso em: 27 de abril de 2020.

LUBBERS, Peter; ALBERS, Brian; SALIM, Frank. Programação Profissional em HTML5.1. ed. Rio de Janeiro: Alta Books editora, 2013.

MANZANO, José Augusto N. G; TOLEDO. MySQL 5.1 interativo Guia Prático de Orientação e Desenvolvimento. 3. ed. São Paulo: Editora Érica Ltda, 2011.

MEIRA, Tatiane Costa et al. **Exposição ao ruído ocupacional**: reflexões a partir do campo da Saúde do Trabalhador. 2012. Disponível em:

<https://repositorio.ufba.br/ri/bitstream>. Acessado em 02 de novembro de 2020

MILLMAN, J. et al. **Eletrônica: dispositivos e circuitos** volume 2. São Paulo:

McGraw-Hill, 1981. NELLI, Máira Pietraroia et al. Estudo da audição e incômodo auditivo em trabalhadores do setor canavieiro expostos a ruído e produtos químicos. 2015.

Páscoa, **Impacto do ruído no desempenho cognitivo** 2020. Disponível em

<<https://repositorio-aberto.up.pt/bitstream/10216/80098/2/36350.pdf>. Acesso em 22/10/2020.

PAZ, E. C. dá et al. **Estudo comparativo da percepção do ruído urbano**. 2005. Rev. Saúde Pública, São Paulo, v. 39, n. 3. Disponível em: <<http://www.scielo.org/scielo>.

[php?script=sci_arttext&pid=S0034-89102005000300019&nrm=iso](http://www.scielo.org/scielo.php?script=sci_arttext&pid=S0034-89102005000300019&nrm=iso)>. Acesso em: 25 de outubro de 2019.

PEDRONI, V. A. **Circuitos Eletrônicos. Rio de Janeiro: Livros Técnicos e**

Científicos Editora S.A., 1986. PROCESSING. Processing. 2019. Disponível em: <<http://www.processing.org>>. Acesso em: 25 de outubro de 2019.

PHP.php.2020. Disponível em: <https://www.php.net/manual/pt_BR/intro-what-is.php>. Acesso em 14 de março de 2020.

GOOGLE CHART.2020. Disponível em:

<https://developers.google.com/chart/glossary>. Acesso em 14 de março de 2020.

SALLES, Laís Monique Mendes. **Desenvolvimento de um dispositivo de medição de ruído com base no sistema OpenWrt.**2017. Disponível em

<https://repositorio.unesp.br/handle/11449/154132>>. Acesso em 25 outubro de 2020.

SANTOS, Andre Luiz Cirino dos *et al.* **MEDIÇÃO DA INTENSIDADE SONORA A PARTIR DA CONSTRUÇÃO DE UM DECIBELÍMETRO.** 2010. 37 f. - Curso de Engenharia de Computação, Universidade Tecnológica Federal do Paraná, Curitiba, 2010. Disponível em: <http://paginapessoal.utfpr.edu.br/msergio/portuguese/ensino-de-fisica/oficina-de-integracao-ii/oficina-de-integracao-ii/Monog-10-2-Decibelimetro.pdf>. Acesso em: 22 out. 2020.

SOMMERVILLE, Ian. **Engenharia de Software.** 9. ed. São Paulo: Pearson Education do Brasil, 2011. WIRING. Wiring. 2019. Disponível em: <http://wiring.org.co>. Acesso em: 25 de outubro de 2019.

TOCCI, R. J. et al. **Sistemas digitais: princípios e aplicações.** 10. ed. São Paulo:

Pearson Prentice Hall, 2007. Tradução: Claudia Martins. Revisão Técnica: João Antônio Martino. Vianna e Lima (2014). **APLICAÇÃO DO ARDUÍNO NO**

MONITORAMENTO AMBIENTAL. Disponível em

<http://www.peteq.feq.ufu.br/jorneq/anais2014/trabalhos/A34.pdf>. Acessado em: 25 outubro de 2020.

ZONON at al. (2016). **Universidade Tecnológica Federal do Paraná,** Curitiba.

Disponível em: <http://repositorio.roca.utfpr.edu.br/jspui/handle/1/15264>. Acesso em 21 de outubro de 2020.

APÊNDICE A – Código em Linguagem C++ responsável por medir o som do ambiente.

```
#include <SPI.h>
#include <Ethernet.h>
#include <MySQL_Connection.h>
#include <MySQL_Cursor.h>
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
byte servidor[] = {192, 168, 1, 104};
#define portaHTTP 80
EthernetClient clienteArduino;
const int PinoSensor = A0; //pino Analógico de Entrada 0
const int PinoDigital = 0;
//int Estado = 0;
int ValorSensor = 0;
int valorMaior = 0;
float tensao = 0;
int dB = 0;
int cont = 0;
void setup() {
  Serial.begin(9600);
  Ethernet.begin(mac);
  if(Ethernet.begin(mac)==0){
    Serial.println("Falha ao conectar a rede");
    Ethernet.begin(mac);
  }
  Serial.print("Conectado a rede, no ip: ");
  Serial.println(Ethernet.localIP());
}
void loop() {
  while (cont < 10000)
  {
    ValorSensor = analogRead(PinoSensor);
    if (ValorSensor > valorMaior)
    {
      valorMaior = ValorSensor;
    }
    cont++;
  }
}
```

APÊNDICE B – Código em Linguagem C++ responsável por medir o som do ambiente.

```
cont = 0;
tensao = valorMaior / 1023.0 * 4.53;
//dB = 103.1*tensao - 115,4;
dB = 87.1 * tensao - 75, 4;
if (dB < 0)
{
    dB = 0;
}
//Serial.print("Tensao: ");
//Serial.print(tensao);
//Serial.println(" V");
Serial.print("Intensidade: ");
Serial.print(dB);
Serial.println("dB");
valorMaior = 0;
delay(100);
Serial.println("Conectando ao servidor...");
if(clienteArduino.connect(servidor, portaHTTP)){
    clienteArduino.print("GET /Aplicacao/salvar.php");
    clienteArduino.print("?dB=");
    clienteArduino.print(dB);
    clienteArduino.println(" HTTP/1.0");
    clienteArduino.println("Host: 192.168.1.104");
    clienteArduino.println("Connection: close");
    clienteArduino.println();
    clienteArduino.stop();

}else{
    Serial.println("falha na conexao com o servidor");
    clienteArduino.stop();
    delay(5000);
}
}
```