

# Relatório Técnico

## Environment Box - Avaliadora da Qualidade do Ambiente de Trabalho

Allan P. Souza – allanpatrick18@gmail.com  
Giovanni Pietrangelo – gigio.pietrangelo@gmail.com  
Lucas Silvestre Kloss Teles – klossteles@gmail.com  
Thiago Nagaoka – nagaoka.thiago@bol.com.br

Junho de 2017

### Resumo

"Segundo a OIT (Organização Internacional do Trabalho), um profissional passa em média  $\frac{1}{4}$  da vida adulta dentro do ambiente de trabalho. Assim, a qualidade de vida deste profissional está diretamente relacionada as condições desses locais. Este projeto surge pela necessidade de monitorar a qualidade de vida do trabalhador dentro do ambiente onde está inserido, visando auxiliar empresas no monitoramento constante das condições do ambiente de trabalho, com base nas Normas Regulamentadoras (NRs) brasileiras. A integração de várias tecnologias nos permite alcançar este objetivo. Este projeto compreende três grandes partes, denominadas de estações. A primeira, a Estação Remota, consiste de um sistema embarcado responsável pelo envio de medições do ambiente monitorado. Este sistema é composto por sensores, decibelímetro, microprocessador e um módulo de comunicação. Tratando-se da segunda estação, a Estação Base possui toda a estrutura de um Web Service implementada para recepção, tratamento e envio dos dados destinados à Estação Cliente. Esta última sendo um aplicativo desenvolvido exclusivamente para este projeto para auxiliar e alertar os gestores e demais responsáveis sobre as condições físicas no ambiente de trabalho, em especial quando tais condições infringirem os limites definidos pelas NRs. Os resultados de toda essa integração para a construção de um protótipo do projeto idealizado, por fim, mostrarem-se satisfatórios para o objetivo dele. Todas as tecnologias conversam entre si, e cada uma cumpre com a sua finalidade."

## 1 Introdução

A qualidade do ambiente de trabalho vem sendo priorizada cada vez mais na sociedade atual. Se até a Segunda Revolução Industrial era possível encontrar indústrias que sujeitavam seus funcionários a trabalhar sem equipamentos de

segurança e em locais com pouca iluminação e ventilação, hoje, graças as sucessivas manifestações trabalhistas durante os séculos XIX e XX [1], são especificadas por lei as condições ideais do ambiente de trabalho.

Contudo, por mais que tais condições sejam obrigatórias, as organizações estão percebendo por si mesmas a importância que a qualidade destes ambientes possuem para a maior produtividade de seus trabalhadores e competitividade no mercado. Fred Fiedler [2][3], pesquisador da indústria e psicologia organizacional, em sua teoria focada na efetividade de uma organização garante que condições ambientais desfavoráveis no ambiente de trabalho, como excesso de calor, ruídos, falta de umidade, causam desconforto e podem levar a danos consideráveis à saúde, influenciando negativamente o desempenho dos funcionários. A Agência Europeia para a Segurança e a Saúde no Trabalho (OSHA) [4] possui estudos que mostram a influência dos locais de trabalho na saúde do trabalhador, evidenciando que os mesmos com muito ruído, baixa umidade ou que estão com temperatura imprópria, podem causar estresse e problemas de saúde.

Para que houvesse uma padronização nas condições consideradas agradáveis ao trabalho, foram criadas as Normas Regulamentadoras brasileiras. Existem várias normas, mas a NR-15 e a NR-17 são os objetivos deste projeto. A norma NR-15 [5] define os agentes insalubres, limites de tolerância e os critérios técnicos e legais para avaliar e caracterizar as atividades e operações insalubres. Este trabalho foca nos limites definidos por ela, relacionados com a temperatura e a composição dos gases no ar. No que diz respeito a temperatura, a mesma não pode exceder os 27°C no local de trabalho. Quanto a qualidade do ar, é enfatizado o que diz respeito ao monóxido de carbono que não pode exceder os 20 ppm<sup>1</sup>.

A norma NR-17 [6], voltada diretamente a ergonomia, estabelece critérios que permitem a implantação da efetiva adaptação das condições de trabalho às características psicofisiológicas dos trabalhadores [7]. Assim, ela é aplicada de modo a possibilitar o máximo de segurança e conforto. Com relação a esta norma, o presente projeto foca no que diz respeito a umidade do ar, temperatura e intensidade sonora. De acordo com a NR-17, a umidade do ar não pode ser inferior a 40%. Para efeito de conforto, a intensidade dos ruídos não pode ultrapassar 65 dB. E, por fim, a temperatura deve permanecer entre 20°C e 23°C. Nesta última grandeza, observamos que ambas as normas abordam sobre ela. Destacamos que será levado em maior consideração o descrito pela NR-17 devido o seu propósito ser especificamente a ergonomia, ou seja, a organização do trabalho no qual existem interações entre seres humanos e máquinas.

O projeto, denominado de Environment Box, visa auxiliar as empresas na busca de uma melhor produtividade, monitorando constantemente as condi-

---

<sup>1</sup>Partes por milhão (ppm): Especifica a quantidade, em gramas, de soluto presente em 1 milhão de gramas da solução. É uma grandeza que serve para relacionar a massa do soluto com a de soluções que estão muito diluídas. Por exemplo, o ar (solução gasosa) contém 8 ppm de gás hélio. Isso significa que em cada 1 m<sup>3</sup> do ar atmosférico existe 8 mL de hélio.

ções do ambiente de trabalho. Elas poderão ter o controle quanto ao ambiente onde seus funcionários estão inseridos, e principalmente se este está de acordo com as Normas Regulamentadoras brasileiras (NRs). O diagrama da Figura 1 apresenta uma visão geral do projeto, os componentes envolvidos e como eles se relacionam entre si.

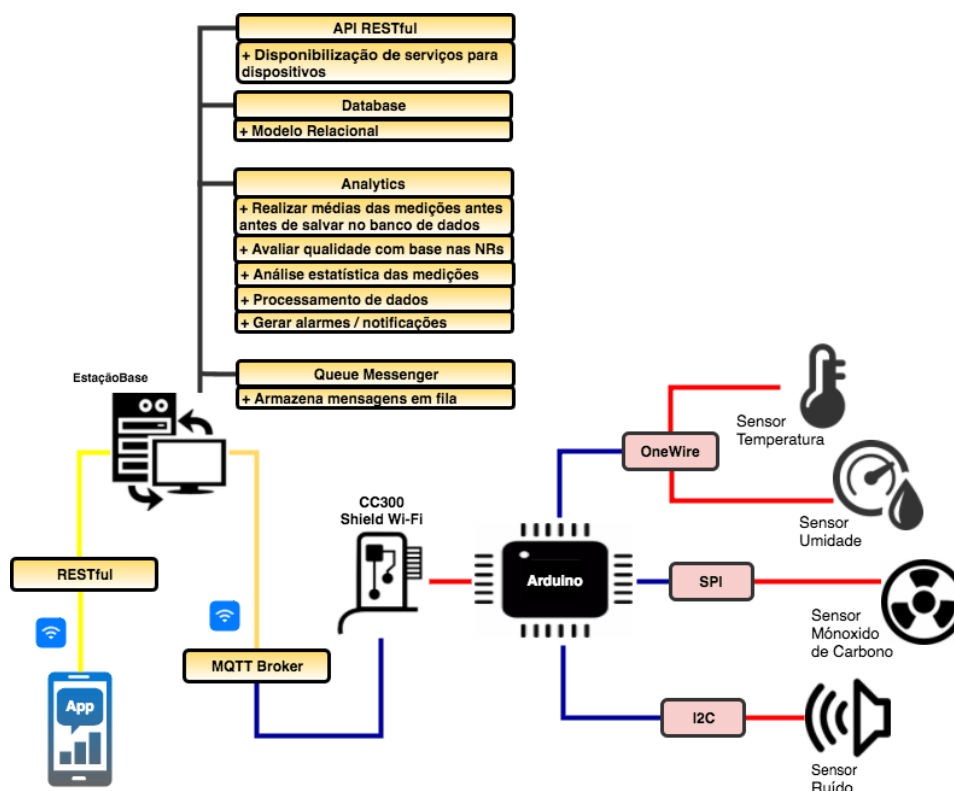


Figura 1: Diagrama de Visão Geral. Fonte: Autoria Própria

O projeto desenvolvido tem como principais requisitos:

#### 1. Funcionais

- (a) O servidor deverá receber requisições do aplicativo.
- (b) O servidor deverá receber os dados provenientes da estação remota de acordo com as configurações definidas para cada um dos sensores.
- (c) O servidor deverá calcular as médias, desvios padrões, variância ou outra análise necessária para compreender se o ambiente está de acordo com as Normas Regulamentadoras.

- (d) O servidor deverá permitir o armazenamento de dados de mais de uma estação remota.
- (e) O aplicativo deverá permitir o cadastro de novas estações remotas.
- (f) O aplicativo deverá gerar gráficos para acompanhamento das medições.
- (g) O aplicativo deverá informar ao usuário através de notificações quanto as condições de trabalho não estão de acordo com as Normas Regulamentadoras.
  - Intensidade sonora não deverá permanecer acima de 65 dB por mais de duas horas
  - Temperatura deverá permanecer entre 20 e 23°C
  - Umidade do não deverá ficar inferior a 40%.
  - Concentração de monóxido de carbono não deverá ultrapassar 40 ppm(partes por milhão).
- (h) A estação remota deverá medir o ruído, a temperatura, o monóxido de carbono e a umidade.

## 2. Não funcionais

- (a) A estação remota deverá ser compacta.
- (b) O aplicativo mostra em cores as medições no histórico de acordo com as faixas de medida.
- (c) O cliente deverá possuir acesso a internet. Caso não possua, não será possível receber notificações (alertas).
- (d) A estação base deverá possuir conexão estável com a internet, permitindo ao usuário atualizar os dados em seu dispositivo móvel.
- (e) O servidor e a estação base deverão se comunicar através de protocolos de comunicação preestabelecidos.
- (f) O usuário deverá possuir um dispositivo iOS (iPhone ou iPad) para poder utilizar o aplicativo.
- (g) A estação remota deverá estar conectado à uma fonte de alimentação.
- (h) A estação remota deverá conter um sensor de ruído, um sensor de temperatura, um sensor de monóxido de carbono e um sensor de umidade.
- (i) A estação remota deverá comunicar-se com a estação base através de uma rede Wi-Fi.

## 2 Tecnologias

O presente tópico trata de duas das três estações que compõem o projeto, e as tecnologias envolvidas em cada uma. São elas a Estação Base e a Estação Remota. A Estação Cliente, no caso o próprio aplicativo que a compõem será apresentado na seção Desenvolvimento. A figura 2 apresenta uma visão em blocos do relacionamento entre cada estação e as tecnologias envolvidas no funcionamento do projeto.



Figura 2: Diagrama de funcionamento do sistema. Fonte: Autoria Própria

### 2.1 Estação Base

A denominada Estação Base (apresentada na figura 1) é a solução utilizada na integração de sistemas e na comunicação entre as nossas aplicações. Ela é responsável por armazenar dados e estabelecer a comunicação entre o sistema embarcado e o aplicativo. Abaixo são descritos as tecnologias que a compõem, o

Message Broker, os protocolos utilizados, o banco de dados escolhido e os *frameworks* utilizados.

### 2.1.1 Message Broker

Sendo um módulo de programa intermediário, o Message Broker traduz uma mensagem de um protocolo formal de mensagens do remetente para o protocolo do receptor. Ele é um padrão de arquitetura para validação, transformação e roteamento de mensagens [8]. Em outras palavras, os Message Brokers são elementos onde as aplicações de *software* se comunicam trocando mensagens formalmente definidas.

### 2.1.2 RabbitMQ

RabbitMQ é um Message Broker cujo objetivo é receber e redistribuir as informações (mensagens) recebidas, armazenando-as em filas, e quando requeridas pela aplicação, redistribuí-las. É de fundamental importância para o projeto, pois é através da comunicação entre o Web Service e o RabbitMQ que iremos obter as medições realizadas e, logo, enviadas pelos dispositivos desenvolvidos.



Figura 3: Broker. Fonte: [9].

### 2.1.3 Protocolo MQTT

MQTT (*Message Queuing Telemetry Transport*), é um protocolo de mensagens leve para sensores e pequenos dispositivos móveis, otimizado para redes TCP/IP não confiáveis ou de alta latência. O esquema de troca de mensagens é fundamentado no modelo *publish-subscribe*. É muito recomendado para uso em dispositivos integrados devido aos seguintes itens:

- Protocolo assíncrono, com vários níveis diferentes de qualidade de serviço, sendo importante nos casos em que as conexões de Internet não são confiáveis.
- Envia mensagens curtas e precisas que o tornam útil para situações de largura de banda baixa.
- Não requer um *software* muito complexo ou que demande muito processamento para implementar um cliente, o que o torna excelente para dispositivos como o Arduino que dispõe de memória limitada.

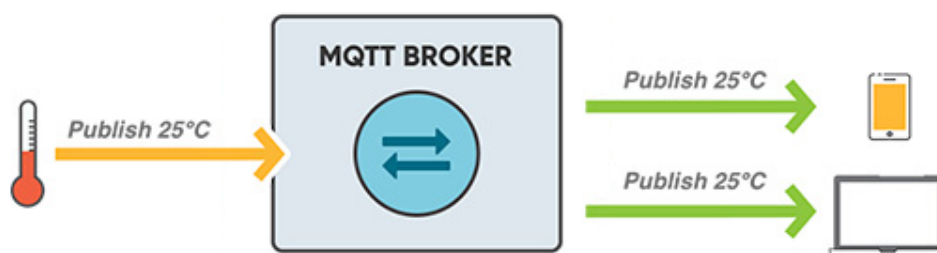


Figura 4: MQTT Broker. Fonte: [9].

#### 2.1.4 Protocolo Avançado de Enfileiramento de Mensagens (AMQP)

AMQP<sup>2</sup> (*Advanced Message Queuing Protocol*) é um protocolo de camada de aplicação padrão aberto para *Message Oriented Middleware*, utilizado para a comunicação entre o Message Broker e o nosso Web Service.

#### 2.1.5 PostgreSQL

O sistema gerenciador de banco de dados objeto-relacional, PostgreSQL<sup>3</sup>, é de fácil instalação, possui interfaces simples e intuitiva, várias possibilidades de extensão (entre a adição de novos dados como funções e operadores), código aberto e abrange uma ampla documentação de apoio. Sua simplicidade no uso e a boa documentação (tal como a experiência profissional da equipe) foram os fatores que levaram a sua escolha para criar o modelo relacional da regra de negócio do *software* para armazenamento de dados coletados pelos dispositivos (sensores e microfone).

#### 2.1.6 Spring Framework

O Spring<sup>4</sup> é um framework de código aberto foi criado com intuito de simplificar a programação em Java. Criado por Rod Johnson [10], como uma alternativa para substituir ou complementar o Enterprise JavaBeans (EJB). Escolhido para o projeto por poder ser utilizado em qualquer aplicação Java, atribuindo algumas funcionalidades importantes como a injeção de dependência (CDI).

#### 2.1.7 Hibernate Framework

O Hibernate<sup>5</sup> é um framework ORM - Object Relational Mapping. É uma ferramenta que nos ajuda a mapear os atributos entre uma base de dados relacionais e o modelo objeto de uma aplicação, através do uso de arquivos XML.

<sup>2</sup>Referência: <https://www.amqp.org/>

<sup>3</sup>Referência: <https://www.postgresql.org/>

<sup>4</sup>Referência: <https://spring.io/>

<sup>5</sup>Referência: <http://hibernate.org/>

## 2.2 Estação Remota

A Estação Remota engloba todo o sistema embarcado deste projeto. Ele conta com um Microcontrolador Arduino UNO, um Shield Wi-Fi CC3000 (apresentado com mais detalhes na subseção Comunicação das Tecnologias), dois sensores e, por fim, um microfone utilizado para implementar o decibelímetro incorporado no projeto (os detalhes encontram-se na subseção Circuito de Captação de Ruído). Nesta seção serão abordados os seguintes sensores: Sensor de monóxido de carbono, sensor de umidade e temperatura, e Microfone Eletreto usado para o projeto do sensor de ruído.

### 2.2.1 Sensor de Monóxido de Carbono

O sensor MQ-2 (figura 5) possui saída analógica e alta sensibilidade para LPG, propano, hidrogênio e monóxido de carbono (CO). Pelo fato de seu material ser composto por  $\text{SnO}_2$ , em ambiente com ar limpo o sensor possui baixa condutividade. Quando o mesmo é exposto em um ambiente com alta concentração destes gases, a sua condutividade aumenta a medida que há maior acúmulo de fluídos. O MQ-2 deve ser alimentado com 5V. O alcance de concentração de gás é compreendido no intervalo de 3ppm à 10000ppm. Com o ajuste da sensibilidade, foi possível calibrá-lo para compreender um intervalo bem menor para a detecção dos 20ppm estabelecidos pela NR-15.

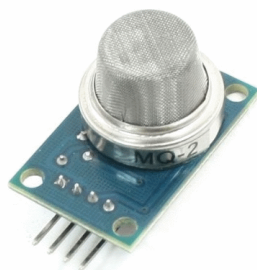


Figura 5: Sensor MQ-2. Fonte: [FONTE]

### 2.2.2 Sensor de Temperatura e Umidade

Sensor digital de umidade e temperatura, DHT11 (figura 6), cujo sinal digital de saída é calibrada. O sensor é de baixo custo e possui estabilidade a longo prazo. Apesar do seu baixo custo, possui alta qualidade, resposta rápida e calibração precisa. O sensor deve ser alimentado com uma tensão entre 3,5V a 5,5V. O período de amostragem é maior que 2 segundos, a faixa de medição da temperatura é de 0°C à 50°C e a de umidade entre 20% à 90% UR (umidade relativa).



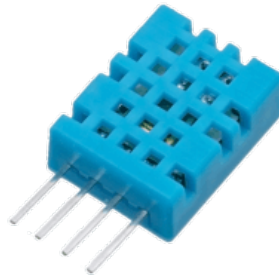


Figura 6: Sensor DHT11. Fonte: [FONTE]

### 2.2.3 Microfone Eletreto

O projeto de um sensor de ruído implica em uma série de análises que devem ser realizadas a fim de definir as características de operação desejadas para o instrumento. Algumas das questões importantes a serem levadas em consideração são as faixas de amplitude e frequência dos sinais que se deseja captar, as quais serão discutidas em mais detalhes nas próximas seções. Para que o nível de ruído seja controlado, deve-se existir algo que capture a energia mecânica e transforme em uma forma de energia que consigamos mensurar facilmente. Para isso foi utilizado o microfone de eletreto, o qual é composto por um sensor capacitivo ligado à base de um transistor **FET** e ambos são instalados em uma cápsula metálica.

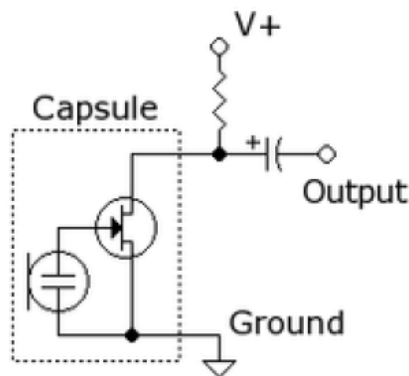


Figura 7: Microfone Eletreto. Fonte: [11].

## 3 Desenvolvimento

Na secção corrente são abordados o desenvolvimento do circuito de Captação do Ruído, a análise espectral do mesmo, como foi feito o firmware que converte

a saída digital do circuito em decibel e como foi feita a integração dos três sensores no *Arduino Uno*.

### 3.1 Circuito de Captação do Ruído

Com o conhecimento do funcionamento de um microfone de eletreto, pode-se projetar um circuito de amplificação e filtro para os sinais desejados. Projetado primeiramente em um software de simulação, **Multisim**<sup>6</sup>, um amplificador com filtros para o sinal almejado, conforme as figuras 8 e 9.

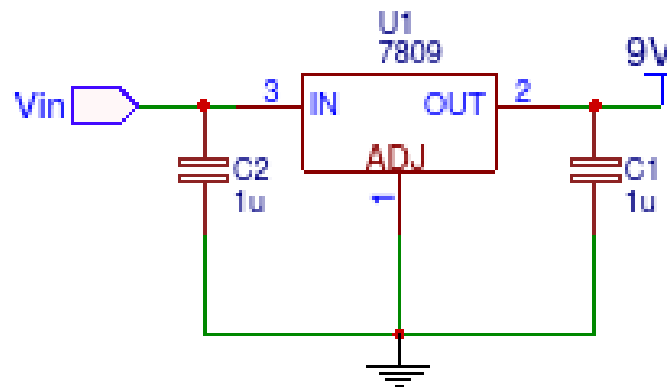


Figura 8: Circuito de alimentação de 9V. Fonte: Autoria própria.

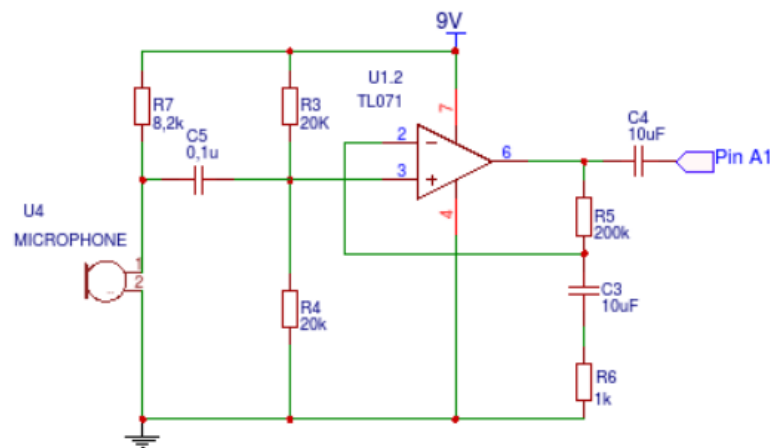


Figura 9: Circuito de captação e amplificação do sinal de ruído. Fonte: Autoria própria.

Observando as figuras 8 e 9, percebe-se que o circuito é alimentado com uma tensão de 12V, pois esta é a tensão do pino **Vin** da placa. É necessário um regulador de tensão de 9V para que a mesma alimente o circuito de polarização

<sup>6</sup>Referência: <http://www.ni.com/multisim/pt/>

do microfone eletreto e o CI do amplificador operacional. A partir da figura 9, é visto que é necessário polarizar o microfone eletreto, pois na sua construção interna existe um transistor FET. Para isso, foi colocado um resistor de  $8.2k\Omega$  entre o positivo do sensor e a tensão de 9V e um capacitor de  $0.1\mu F$ . Nesse momento, o sinal que sai do microfone estará na ordem de mV. Para que o Arduino UNO possa processar esse sinal, será necessário amplificá-lo. Mas antes, é colocado primeiramente um capacitor de acoplamento para eliminar a parte contínua do sinal do microfone. Para a amplificação do sinal, foi utilizado um amplificador operacional TL071, na configuração realimentação negativa, entrada não-inversora e alimentação assimétrica, onde uma é a alimentação com a tensão de 9V, e a outra é o **ground** (tensão de 0V). Observando o circuito da figura 9, o ganho desse amplificador é de 251. Na saída, existe um capacitor de acoplamento de  $10\mu F$  para eliminar a parte contínua do sinal. O sinal do microfone processado por esse circuito estará limitado entre 0V e 6,5V.

### 3.1.1 Análise de Fourier

Com a auxílio do software de simulação **Multisim**, pode-se observar que o circuito projetado do amplificador e filtro passivo implementados atendem aos requisitos da banda de frequência desejada conforme figura 10. Fazendo uma análise do espectro de frequência, é percebida que as senoides que estão na faixa de frequência audível do ouvido humano, 20Hz à 20kHz, não sofrem atenuações acentuadas quando as mesmas são processadas pelo circuito das figuras 8 e 9. Isso mostra que as perdas de informações das ondas nessa faixa não é alta, pois a frequência de corte inferior do circuito, segundo a análise, é de 15Hz. E a frequência de corte superior é de 15kHz.

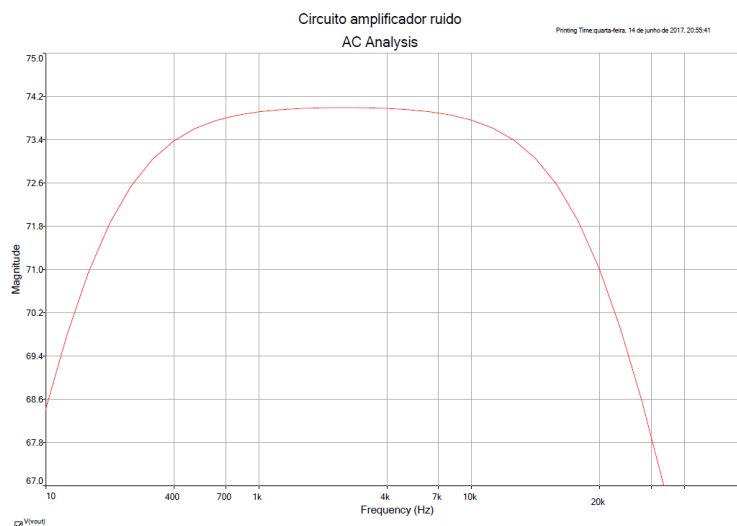


Figura 10: Análise da frequência - Ganho x Frequência. Fonte: Autoria própria.

Tabela 1: Conversão da saída do circuito para decibéis

Pino A1	533	527	560	555	660	581	650	634	680	620
Valor dB	51,4	51,3	54,4	70	69,5	70	70	68,5	55,5	60,4

### 3.2 Conversão de Sinal Digital para Decibel (dB)

Após o condicionamento analógico do sinal advindo dos microfones, é necessário um processamento digital deste sinal para que a tensão medida em um conversor A/D seja utilizada no cálculo do nível de pressão sonora. O dispositivo responsável por esta tarefa é a plataforma Arduino, a qual contém um microprocessador ATmega2560, portas de entrada e saída (algumas delas com conversão analógico-digital), entre outros periféricos, tendo como base a utilização de interpolação polinomial. Para isso, foi utilizado uma frequência de amostragem acima de 20 kHz, seguindo o teorema de Nyquist. Foram colhidas leituras do circuito projetado das figuras 8 e 9 em um ambiente silencioso, anotando os valores em dB de um decibelímetro calibrado como base (*Sound Level Meter - Extech Instruments*). Na tabela 1 é encontrada a tabela com os valores obtidos do circuito e os valores do decibelímetro base.

Com essa tabela em mãos, foi gerada uma equação que relaciona as duas grandezas, a saída do circuito com o valor em decibéis. A equação gerada é apresentada abaixo:

$$y = 31,99 * \ln(x) - 143,0 \quad (1)$$

A variável  $x$  é o valor da saída do circuito já convertido do analógico para digital, ou seja, um valor entre 0 e 1023. E  $y$  é o valor em decibéis.

### 3.3 Integração dos Sensores

Toda a integração é feita através de um *software* programado para o Arduino UNO. Os sensores são plugados nos pinos de entrada analógica e digital. O *software* é configurado para que os mesmos sejam tratados como entrada do sistema. Após isso, são feitas as leituras dos sensores a cada iteração do sistema. Nas figuras 11 e 12 é mostrado um fluxograma da primeira versão do software que foi manuseado para testes dos sensores e coleta de dados. Nesta figura é mostrado o diagrama de classes, este foi projetado para fácil manuseio dos sensores nos software.

### 3.4 Comunicação entre as Tecnologias

A comunicação entre o nosso *software* e *hardware* é efetuada via conexão Wi-Fi utilizando o Shield Adafruit CC3000 [12] que a torna acessível ao Arduino. A Adafruit provê uma biblioteca com vários exemplos básicos da utilização do CC3000 na própria plataforma do Arduino. O CC3000, segundo a empresa, é eletricamente simples de usar [13]. O módulo Wi-Fi requer uma conexão SPI,

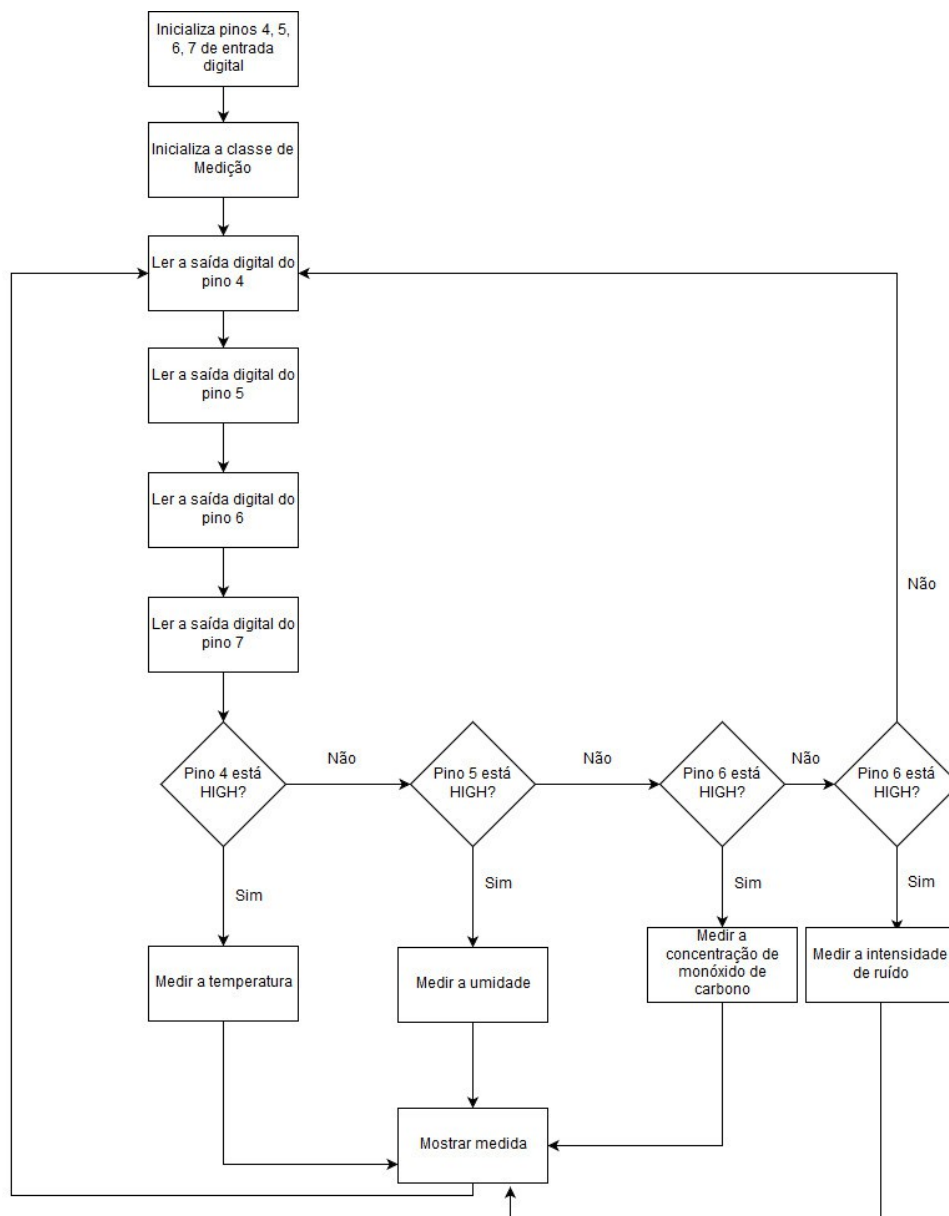


Figura 11: Fluxograma do software. Fonte: Autoria própria.

incluindo um *clock* (CLK), dados de entrada de um microcontrolador (MOSI) e dados de saída para o microcontrolador (MISO). Ele também usa uma linha de seleção de chip, ou *chip-select line* (CS), para a SPI indicando uma transferência de dados como iniciada, quando a mesma ocorrer.

Após a realização das medições é necessário enviar os dados da estação remota para a estação base. No Arduino é feita a implementação de um servidor HTTP somente com o essencial para responder à requisições simples. Devido

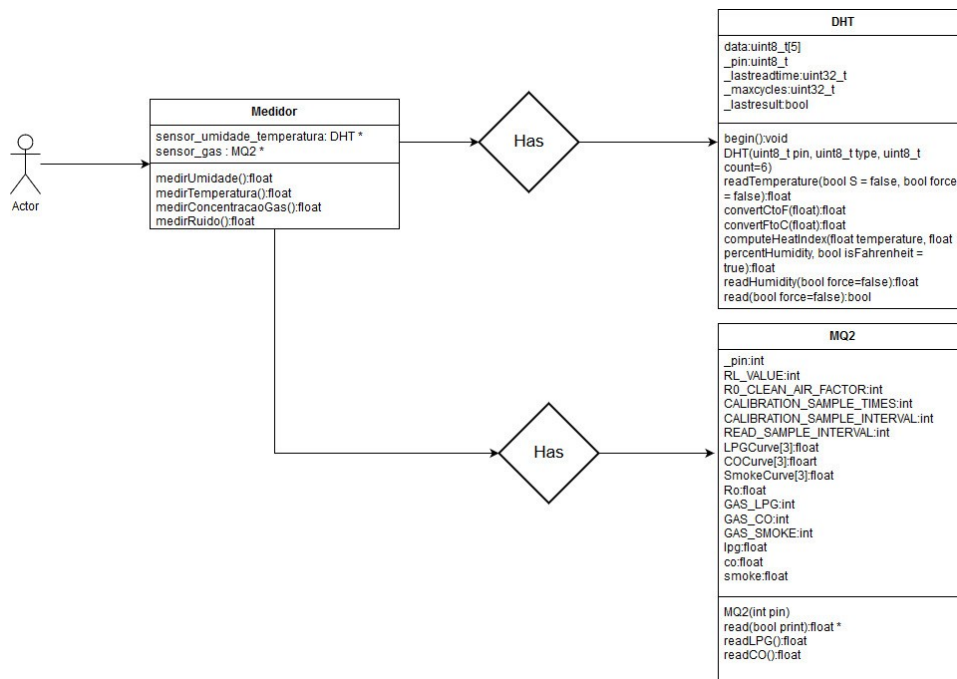


Figura 12: Diagrama de classes Fonte: Autoria própria.

ao módulo não criar *hotspot*, é usado o protocolo DHCP. Este é um protocolo de serviço TCP/IP que oferece configuração dinâmica de terminais possibilitando a concessão de endereços IP. O servidor DHCP gerencia uma faixa fixa de IPs disponíveis assim como outras informações e parâmetros necessários, por exemplo, o DNS. Quando o dispositivo conecta-se na rede, ele envia um pacote UDP com um pedido de configurações DHCP. O servidor DHCP que capturar este pacote irá responder com um outro pacote de configurações que contenha um endereço IP, uma máscara de rede e outros dados opcionais. O endereço IP é atribuído ao dispositivo que receberá as requisições do aplicativo pelo respectivo endereço [14].

### 3.5 Web Service

Ao receber as medições vindas do Message Broker, é necessário realizar o tratamento da mensagem. Foi necessário definir um padrão de envio das medições, caso contrário o dispositivo de envio mandaria as medições de uma forma e o Web Service não conseguiria identificar quais dados que foram enviados. Somente após este tratamento que podemos de fato salvar as medições no banco de dados. Os alertas gerados para o aplicativo, respeitam a Normas Regulamentadoras brasileiras e por motivos de processamento e desempenho decidimos realizar a verificação das medições direto no banco de dados. Ao detectar que um limite imposto pela NR foi descumprido, envia-se uma notificação para o

aplicativo, através do *Apple Push Notification Service*.

### 3.6 Aplicativo

O aplicativo desenvolvido para esta etapa do projeto (protótipo) foi feito para a plataforma iOS exclusivamente, criado no ambiente de desenvolvimento integrado e software livre da Apple Inc., denominado Xcode. Para a sua implementação, é utilizada a linguagem de programação Swift, lançada pela Apple em 9 de setembro de 2014 com sua primeira aparição em junho do mesmo ano na WWDC (*Apple Worldwide Developers Conference*) [15].

O aplicativo é responsável por fornecer retorno inteligente das medições realizadas pelas estações remotas, apresentando gráficos interativos relativos aos sensores em geral e à cada um em específico. Por fim, ele é necessário também para alertar os usuários sobre as infrações das Normas Regulamentadoras brasileiras, ou seja, quando os fatores conhecidos como temperatura, umidade, ruído e monóxido de carbono ultrapassam os limites inferiores e superiores delimitados pelas normas. Torna-se possível, assim, um monitoramento mais eficiente do ambiente de trabalho.

## 4 Resultados

Dentre os resultados esperados encontram-se o próprio ambiente controlado usado pela equipe na validação e calibração dos sensores, tal como as medições obtidas por eles, apresentadas nos gráficos na subseção Análise dos Dados. Por fim, são apresentadas duas telas em específico, na subseção Gráfico e Notificação, referentes ao aplicativo que o tornam parte importante do projeto.

### 4.1 Bancada de Testes

Durante o desenvolvimento do projeto, percebeu-se a necessidade da validação dos dados coletados através dos sensores e do decibelímetro. Tendo isto em vista, projetamos e desenvolvemos uma bancada de testes, onde possuímos o controle sobre um ambiente fechado e sobre todos os fatores ligados ao projeto, sendo eles a temperatura, umidade, monóxido de carbono e o ruído. A bancada de testes possui alguns componentes necessários para que a equipe possa criar um ambiente variável e controlado ao mesmo tempo. Estes componentes são:

- **Lâmpada Incandescente e Dimmer:** A lâmpada irá realizar o aumento da temperatura do ambiente, enquanto que o dimmer é responsável por variar gradualmente a temperatura.
- **Cooler:** Responsável por realizar a mudança de umidade no ambiente.
- **Incenso:** Através da queima do incenso é possível realizar o aumento de monóxido de carbono dentro do ambiente.

Sem a criação da bancada de testes, seria impraticável a validação das medições obtidas assim como o real funcionamento dos sensores em especial, pois dependeríamos de mudanças climáticas para realizar alguns dos testes.



Figura 13: Bancada de testes. Fonte: Autoria Própria.

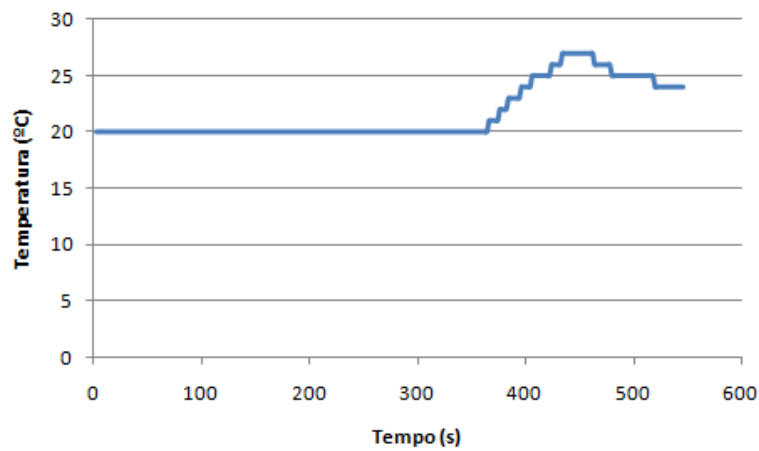
## 4.2 Análise dos Dados

Com o preparo da bancada de testes, foi possível efetivar a coleta de dados para a análise dos mesmos sobre efeito de fatores ambientais controlados. Nos gráficos da figura 14 é observado o comportamento do sensor de temperatura e umidade *DHT11*. Para as leituras da temperatura, a lâmpada incandescente da bancada de teste permanece desligada, mantendo a temperatura ambiente do local, que era de 20 °C. Após as primeiras leituras o dimmer da lâmpada foi ligado, com isso, percebemos um acréscimo na temperatura, chegando em torno de 26°C, 27°C. Após a captura destes dados, desligamos a lâmpada e ligamos o *cooler*, para que a temperatura no ambiente fosse diminuída gradativamente, como podemos observar no gráfico. Tal gráfico, por fim, mostra que o sensor de temperatura se comporta como esperávamos.

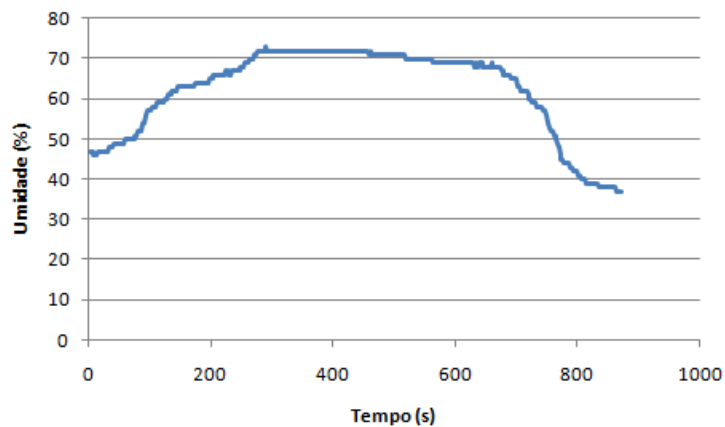
As leituras da umidade do ar, foram realizadas colocando um pedaço de papel úmido na bancada de testes e ligando a lâmpada incandescente, pois como a bancada de testes é fechada, o aumento da temperatura provoca a evaporação da água presente no papel úmido. Com isso, a umidade do ambiente aumenta, como podemos observar no figura 14b. Analisando as primeiras leituras observamos um aumento na umidade do ar.

O comportamento do sensor MQ2 é evidenciado na figura 15a. Esse comportamento foi obtido, colocando no interior da bancada de testes um incenso, fazendo com que houvesse um acréscimo de concentração de monóxido de carbono (CO), porém não fechamos completamente a bancada de testes, pois o incenso aumenta de forma muito rápida a concentração de monóxido de carbono.





(a) Gráfico do DHT11 coletando a temperatura. Fonte: Autoria Própria

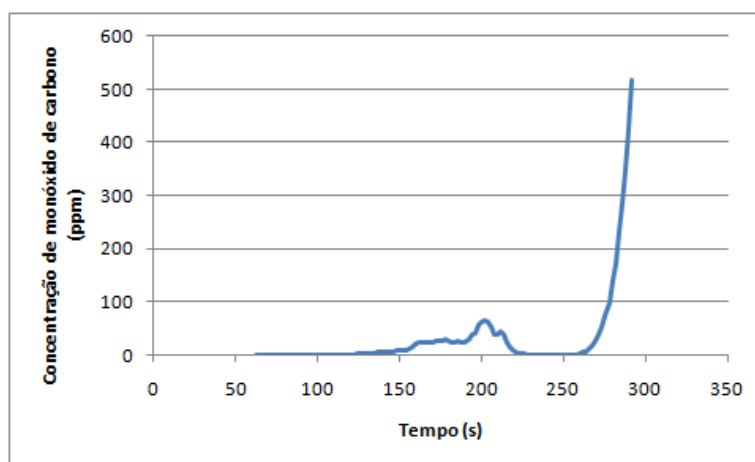


(b) Gráfico do DHT11 coletando a umidade. Fonte: Autoria Própria.

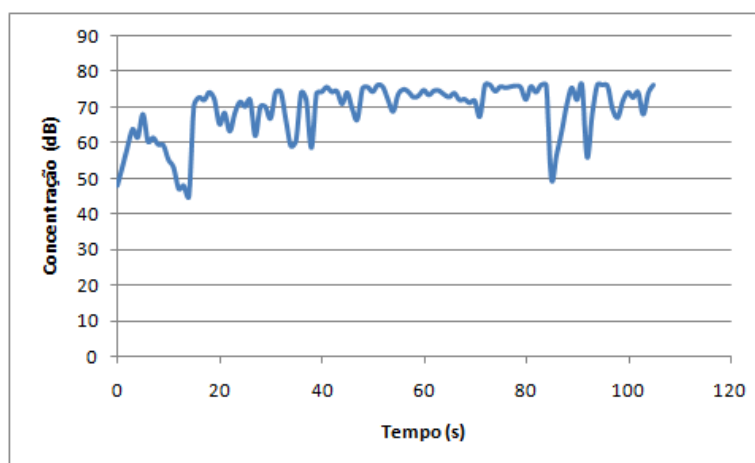
Figura 14: Gráficos dos sensores de temperatura e umidade.

Percebe-se que o comportamento é exponencial, quanto maior a quantidade de gás emitida pelo incenso, maior era a concentração em **ppm** registrada pelo sensor.

A figura 15b mostra o comportamento do circuito de captação de ruído. Esses dados foram obtidos colocando o sensor em um ambiente silencioso, tendo como fonte de ruído uma caixa de som. Os primeiros dados coletados foram obtidos com a caixa de som desligada. Em seguida ligamos a caixa de som e novos dados foram coletados. Em paralelo, coletamos dados a partir de um decibelímetro comercial, devidamente calibrado. O erro percentual entre os dados obtidos pelo decibelímetro produzido pela equipe e o comercial foi de 10%.



(a) Gráfico do sensor MQ2 medindo a concentração de monóxido de Carbono. Fonte: Autorial Própria



(b) Gráfico do sensor de ruído. Fonte: Autorial Própria.

Figura 15: Gráficos sensor MQ2 e sensor de Ruído.

### 4.3 Gráfico e Notificações

Muitos aplicativos são idealizados tendo em vista uma necessidade ou uma dor dos usuários. O aplicativo concebido para o projeto vem da necessidade de informar de forma mais prática, intuitiva e simples o gestor, ou outros responsáveis, a respeito das condições físicas (temperatura, umidade, ruído e inclusive a concentração de monóxido de carbono) do seu local de trabalho. Ou seja, para o gestor será possível monitor e controlar melhor esse seu ambiente (visando possivelmente uma maior produtividade). Na figura 16a, temos umas das formas inteligentes de retornar as informações referentes as medições. O gráfico consiste em informar o usuário sobre quais das condições físicas está mais pró-

xima de infringir as Normas Regulamentadoras brasileiras. Na figura 16b, por fim, apresenta um dos pilares do aplicativo, o recebimento de notificações. A Estação Remota, envia os dados à Estação Base, onde os mesmos são processados e analisados. Caso o Web Service identifique uma infração, é enviada uma notificação alertando sobre qual das condições ultrapassou os limites impostos pelas NRs.

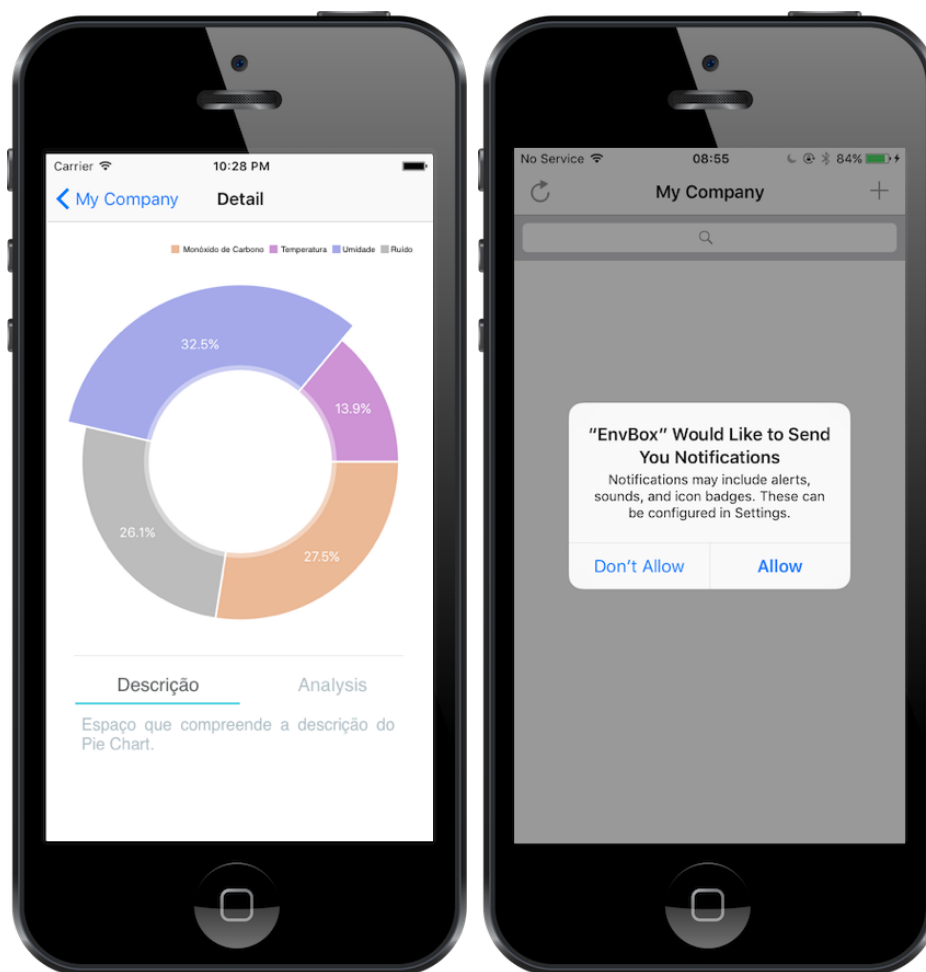


Figura 16: Imagens do aplicativo desenvolvido.

## 5 Considerações Finais

Durante todo o percurso, diversas dificuldades foram encontradas. Mesmo em face delas, o projeto pode chegar a sua conclusão de forma satisfatória, cumprindo com o seu objetivo. Entre as principais dificuldades podemos destacar:

1. Compreensão do funcionamento dos sensores *DHT11*, umidade e temperatura, e *MQ2*, Monóxido de Carbono para que a biblioteca de medição fosse desenvolvida.
2. Projeto do hardware do amplificador do sinal que é produzido pelo *microfone eletreto*.
3. Gerar uma equação onde, quando o valor lido do circuito fosse colocado como entrada, retornasse o menor erro possível de leitura quando comparado a um decibelímetro base.

## Agradecimentos

Agradecemos ao Professor Dr. Guilherme Alceu Schneider e ao Professor Dr. João Alberto Fabro pelo suporte durante todo o projeto, sugestões e avaliações, em especial para a construção do presente Relatório Técnico.

Agradecemos ao Professor Dr. Marcelo de Oliveira Rosa por nos conceder um decibelímetro profissional que nos possibilitou validar e calibrar o nosso decibelímetro, implementado neste projeto.

Agradecemos ao Engenheiro André Mariano e ao Engenheiro Joed Lopes, da empresa Exati Tecnologia, pelo auxílio durante a etapa de idealização do projeto, principalmente pelas análises e sugestões das tecnologias a serem usadas.

## Referências

- [1] Portal Brasil. Evolução das relações trabalhistas, 2014. Disponível em: <http://www.brasil.gov.br/economia-e-emprego/2011/04/evolucao-das-relacoes-trabalhistas>.
- [2] SOUZA, A. P.; MINETTI, L. J.; MACHADO, C. C.; FIEDLER, N. C.; TIBIÇÁ, C. G. Avaliação da carga física de trabalho exigida em operações de colheita florestal., 1998. Revista *Árvore*, Viçosa.
- [3] FIEDLER, N. C.; RODRIGUES, T. O.; MEDEIROS, M. B. Avaliação das condições de trabalho, treinamento, saúde e segurança de brigadistas de combate a incêndios florestais em unidades de conservação do distrito federal., 2006. Revista *Árvore*, Viçosa.

- [4] Agência Europeia para a Segurança e a Saúde no Trabalho. O impacto do ruído no trabalho., 2006. Disponível em: <https://osha.europa.eu/pt/tools-and-publications/publications/factsheets/57>.
- [5] Normas Regulamentadoras brasileiras. Nr-15 atividades e operações insalubres, 1990. Disponível em: <http://www.uneb.br/pgdp/files/2010/09/NR-15-Atividades-e-operacoes-insalubres1.pdf>.
- [6] Normas Regulamentadoras brasileiras. Nr-17 ergonomia, 2007. Disponível em: [https://www.pncq.org.br/uploads/2016/NR\\_MTE/NR%2017%20-%20ERGONOMIA.pdf](https://www.pncq.org.br/uploads/2016/NR_MTE/NR%2017%20-%20ERGONOMIA.pdf).
- [7] Ministério do Trabalho. Manual de aplicação da norma regulamentadora nº 17, 1994. Disponível em: [http://www.simucad.dep.ufscar.br/simucad/dn\\_manualnr17.pdf](http://www.simucad.dep.ufscar.br/simucad/dn_manualnr17.pdf).
- [8] Enterprise Integration Patterns. Hub and spoke [or] zen and the art of message broker maintenance, 2003. Disponível em: [http://www.enterpriseintegrationpatterns.com/ramblings/03\\_hubandspoke.html](http://www.enterpriseintegrationpatterns.com/ramblings/03_hubandspoke.html).
- [9] CloudAMQP. Mqtt, 2017. Disponível em: <https://www.cloudamqp.com/>.
- [10] Pivotal. Spring framework, 2017. Disponível em: <https://spring.io>.
- [11] Escrito por Diversos Colaboradores. Projeto e construção de um decibelímetro, 2015. Disponível em: <http://www.newtoncbraga.com.br/index.php/eletronica/57-artigos-e-projetos/13196-projeto-e-cosntrucao-de-um-decibelimetro-col0110>.
- [12] Prometec. Jugando con wifi shield cc3000, 2015. Disponível em: <http://www.prometec.net/wifi-cc3000/>.
- [13] Rick Lesniak. Adafruit cc3000 wifi, 2013. Disponível em: <https://learn.adafruit.com/adafruit-cc3000-wifi/overview>.
- [14] FOROUZAN, B. A.; MOSHARRAF, F. Redes de computadores: Uma abordagem top-down, 2013. AMGH Editora.
- [15] Apple Inc. The swift programming language, 2014. Disponível em: [https://developer.apple.com/library/content/documentation/Swift/Conceptual/Swift\\_Programming\\_Language/](https://developer.apple.com/library/content/documentation/Swift/Conceptual/Swift_Programming_Language/).