

**UNIVERSIDADE FEDERAL DO ABC**  
**Especialização em Tecnologias e Sistemas da Informação**



**Estação de Medição da Qualidade do Ar, utilizando Raspberry Pi e Arduino  
com baixo custo.**

**Aluno: Jhony Soares Martins da Silva**  
**Orientador: Prof. Dr. Mario Gazziro**

**São Paulo**  
**2024**

**Jhony Soares Martins da Silva**

**Estação de Medição da Qualidade do Ar, utilizando Raspberry Pi e Arduino  
com baixo custo.**

Monografia apresentada ao Curso de  
Especialização em Tecnologias e Sistemas  
de Informação, para a obtenção do grau de  
Especialista em Tecnologias e Sistemas de  
Informação.

Orientador: Prof. Dr. Mario Alexandre  
Gazziro.

A handwritten signature in blue ink, consisting of several overlapping loops and strokes, positioned above a horizontal line.

Assinatura do Orientador

Santo André – SP  
2024

Ficha catalográfica

Silva, Jhony Soares Martins

Estação de Medição da Qualidade do Ar,  
utilizando Raspberry Pi e Arduino com baixo  
custo – Santo André, SP: UFABC, 2024

## **Agradecimentos**

Agradeço, primeiramente, à minha esposa Laís, por toda ajuda emocional, racional e compreensão durante a realização deste trabalho.

Agradeço ao meu orientador Prof. Dr. Mario Alexandre Gazziro, por todo suporte dado durante a realização deste trabalho.

Por fim, agradeço a todos os colegas, professores e amigos que estiveram presentes do começo ao fim da minha jornada pela especialização, Obrigado.

## RESUMO

O objetivo deste projeto é criar uma estação de monitoramento da qualidade do ar de baixo custo, especialmente voltada para comunidades carentes e sem acesso a medições ambientais. A ideia é entender como a poluição nessas localidades afeta a qualidade do ar e, conseqüentemente, a saúde das pessoas que vivem nesses locais.

Para isso, estou utilizando sensores conectados a um microcontrolador (Arduino) para medir poluentes como NO<sub>2</sub>, CO<sub>2</sub>, fumaça e ozônio, além de temperatura e umidade, que também são essenciais para uma boa qualidade do ar. Esses dados coletados serão enviados para um Raspberry Pi, que funcionará como um sistema de armazenagem de dados em um banco de dados MySQL.

O Raspberry Pi também terá uma instância do Grafana instalada, que é um software que permitirá visualizar os dados coletados de forma clara e em tempo real. Assim, será possível acompanhar a qualidade do ar e tomar medidas para melhorar o ambiente.

Esta estação de monitoramento é uma solução prática e acessível, que visa fornecer dados essenciais para melhorar a qualidade do ar e a saúde das pessoas em áreas carentes.

**Palavras-chave:** Monitoramento de ar, Arduino ,Raspberry Pi, Baixo Custo

## ABSTRACT

The objective of this project is to create a low-cost air quality monitoring station, specifically aimed at underserved communities without access to environmental measurements. The idea is to understand how pollution in these areas affects air quality and, consequently, the health of people living there.

To achieve this, I am using sensors connected to a microcontroller (Arduino) to measure pollutants such as NO<sub>2</sub>, CO<sub>2</sub>, smoke, and ozone, as well as temperature and humidity, which are also essential for good air quality. The collected data will be sent to a Raspberry Pi, which will function as a data storage system using a MySQL database.

The Raspberry Pi will also have an instance of Grafana installed, which is a software that allows for clear and real-time visualization of the collected data. This way, it will be possible to monitor air quality and take measures to improve the environment.

This monitoring station is a practical and accessible solution aimed at providing essential data to improve air quality and the health of people in underserved areas.

**Keywords:** Air Monitoring, Arduino ,Raspberry Pi, Low Cost.

## LISTA DE SIGLAS

**MQ-131** - *Sensor de Ozônio (Ozone Sensor)*

**MQ-135** - *Sensor de Poluentes do Ar (Air Quality Sensor for NO<sub>2</sub>, NH<sub>3</sub>, Benzene, etc.)*

**MQ-7** - *Sensor de Monóxido de Carbono (CO Sensor)*

**MQ-5** - *Sensor de Metano, Gás Natural e Fumaça (Methane, LPG, Smoke Sensor)*

**MQ-2** - *Sensor de Fumaça e Gases Combustíveis (Smoke and Combustible Gas Sensor)*

**DSM501** - *Sensor de Material Particulado (Particulate Matter Sensor)*

**DHT22** - *Sensor de Temperatura e Umidade (Temperature and Humidity Sensor)*

**RPI** - *Raspberry Pi (Computador de Placa Única)*

**Arduino** - *Microcontrolador (Plataforma de Desenvolvimento)*

**Grafana** - *Software de Visualização de Dados (Data Visualization Software)*

**MySQL** - *Sistema de Gerenciamento de Banco de Dados (Database Management System)*

**PM10** - *Material Particulado com Diâmetro Menor que 10 Micrômetros (Particulate Matter with Diameter Less Than 10 Micrometers)*

**PM2.5** - *Material Particulado com Diâmetro Menor que 2.5 Micrômetros (Particulate Matter with Diameter Less Than 2.5 Micrometers)*

**NO<sub>2</sub>** - *Dióxido de Nitrogênio (Nitrogen Dioxide)*

**CO<sub>2</sub>** - *Dióxido de Carbono (Carbon Dioxide)*

**O<sub>3</sub>** - *Ozônio (Ozone)*

**VOC** - *Compostos Orgânicos Voláteis (Volatile Organic Compounds)*

## LISTA DE TABELAS

|  |    |
|--|----|
| <b>Tabela 1:</b> Custo do Projeto.....                     | 31 |
| <b>Tabela 2:</b> Sensores.....                             | 33 |
| <b>Tabela 3:</b> Acompanhamento dos 8 dias do Projeto..... | 38 |

## LISTA DE FIGURAS

|  |    |
|--|----|
| <b>Figura 1:</b> Exemplo de Arduino UNO.....   | 19 |
| <b>Figura 2:</b> Exemplo de RaspBerry Pi 4.....  | 20 |
| <b>Figura 3:</b> Sensor MQ - 135.....  | 21 |
| <b>Figura 4:</b> Sensor MQ - 7.....  | 21 |
| <b>Figura 5:</b> Sensor MQ - 5.....  | 22 |
| <b>Figura 6:</b> Sensor MQ - 131.....  | 22 |
| <b>Figura 7:</b> Sensor AM2302.....  | 23 |
| <b>Figura 8:</b> Sensor DSM501A.....   | 24 |
| <b>Figura 9:</b> Exemplo Node-RED.....   | 25 |
| <b>Figura 10:</b> Exemplo Grafana.....   | 26 |
| <b>Figura 11:</b> Apresentando todos os componentes do sistema.....                                | 27 |
| <b>Figura 12:</b> Exemplo Modelo Físico.....   | 28 |
| <b>Figura 13:</b> Protótipo em Protoboard.....   | 32 |
| <b>Figura 14:</b> Esquemático da montagem do dispositivo.....                                      | 33 |
| <b>Figura 15:</b> Fluxograma do programa feito no Arduino.....                                     | 34 |
| <b>Figura 16:</b> Fluxo do Node-RED.....   | 35 |
| <b>Figura 17:</b> Interface do Grafana.....  | 36 |
| <b>Figura 18:</b> Medição NO <sub>2</sub> .....  | 38 |
| <b>Figura 19:</b> Medição CO.....  | 39 |
| <b>Figura 20:</b> Medição Fumaça.....  | 39 |
| <b>Figura 21:</b> Medição O <sub>3</sub> .....   | 40 |
| <b>Figura 22:</b> Medição de micropartículas de 2.5 µg/m <sup>3</sup> e 10 µg/m <sup>3</sup> ..... | 40 |
| <b>Figura 23:</b> Medição Temperatura e umidade.....   | 41 |
| <b>Figura 24:</b> JSON Gerado pelo NODE-RED.....   | 41 |
| <b>Figura 25:</b> Consulta ao Banco de Dados.....  | 42 |
| <b>Figura 26:</b> Interface do Projeto.....  | 43 |

# SUMÁRIO

|  |           |
|--|-----------|
| <b>SUMÁRIO.....</b>                              | <b>10</b> |
| 1. Introdução.....                               | 11        |
| 1.1. Objetivos.....                              | 12        |
| 1.2. Problema.....                               | 13        |
| 1.3. Justificativa.....                          | 15        |
| 1.4. Organização do Trabalho.....                | 17        |
| 2. Tecnologias Relacionadas.....                 | 18        |
| 2.1. Arduino.....                                | 18        |
| 2.2. Raspberry Pi 4.....                         | 19        |
| 2.3. Sensores.....                               | 20        |
| 2.4. Maria DB.....                               | 24        |
| 2.5. Node-RED.....                               | 24        |
| 2.6. Grafana.....                                | 25        |
| 3. Metodologia e Implementação.....              | 26        |
| 3.1. Visão geral.....                            | 26        |
| <b>3.2. Análise de Requisitos.....</b>           | <b>28</b> |
| 3.2.1. Requisitos Funcionais.....                | 28        |
| 3.2.2. Requisitos Não-Funcionais.....            | 29        |
| 3.3. Arquitetura do Projeto.....                 | 30        |
| 3.4. Custos do Desenvolvimento do Projeto.....   | 30        |
| 3.5. Dispositivo de medição de poluentes.....    | 31        |
| 3.6. Integração via Node-RED.....                | 35        |
| 3.7. Banco de dados Mariadb.....                 | 35        |
| 3.8. Grafana.....                                | 36        |
| 4. Resultados.....                               | 37        |
| 4.1. Desempenho dos Sensores.....                | 37        |
| 4.2. Processamento e Armazenamento de Dados..... | 41        |
| 4.3. Visualização e Análise de Dados.....        | 42        |
| 4.4. Otimização dos Códigos Utilizados.....      | 43        |
| 4.5. Considerações Finais.....                   | 44        |
| 5. Conclusões.....                               | 46        |
| 6. Bibliografia.....                             | 48        |
| <b>7. Apêndice.....</b>                          | <b>49</b> |

# 1. Introdução

Estação de Medição da Qualidade do Ar, utilizando Raspberry Pi e Arduino com baixo custo.

A crescente preocupação com a qualidade do ar e seus impactos na saúde humana e no meio ambiente tem impulsionado a busca por soluções de monitoramento acessíveis e eficazes, especialmente para comunidades com recursos limitados que enfrentam desafios adicionais em acessar informações sobre a qualidade do ar e a presença de partículas nocivas. Em São Paulo este problema é ainda mais perceptível devido a problemas relacionados com a qualidade de vida da população da cidade:

“a comunidade paulista vem sofrendo sérias consequências, estima-se que a expectativa de vida dos moradores da cidade tenha diminuído em três anos, devido ao problema da poluição atmosférica que é incidente na região metropolitana, também há uma diminuição na qualidade de vida das pessoas, que sofrem cada vez mais com problemas de saúde, com destaque para os respiratórios, ocasionados também pelo aumento na poluição atmosférica local.” ( COSTA, HENKES, 2013. p. 251).

Monitorar e coletar dados sobre a qualidade atmosférica mostra-se importante em diversas áreas do desenvolvimento humano nas cidades, não se restringindo apenas aos cuidados com a saúde, como confirma o Ministério do Meio Ambiente:

A gestão da qualidade do ar tem como objetivo garantir que o desenvolvimento sócio-econômico ocorra de forma sustentável e ambientalmente segura. Para tanto, se fazem necessárias ações de prevenção, combate e redução das emissões de poluentes e dos efeitos da degradação do ambiente atmosférico. (BRASIL, MMA, 2024.)

Neste contexto, este trabalho se propõe a desenvolver uma estação de monitoramento de qualidade do ar, utilizando tecnologias como Raspberry Pi e Arduino. O objetivo central é criar uma solução de baixo custo e acessível, capaz de medir diversos parâmetros da poluição atmosférica em regiões carentes. Além de monitorar a concentração de partículas e a presença de gases nocivos, o sistema

também visa identificar poluentes gerados por aglomerações humanas, ampliando assim sua utilidade em ambientes urbanos densamente povoados. Para alcançar esse propósito, serão empregados diferentes sensores compatíveis com a plataforma Arduino, como o Sensor de Partículas PMS5003 e o Sensor de Gases MQ, possibilitando uma monitorização detalhada e abrangente.

O Raspberry Pi desempenha o papel de base local para o armazenamento dos dados coletados, garantindo a disponibilidade das informações mesmo em situações de desconexão temporária da rede de comunicação. Os dados serão disponibilizados em tempo real por meio de um dashboard acessível via rede e também serão armazenados para análises futuras e históricas, fornecendo insights importantes para compreender e enfrentar os desafios da poluição atmosférica. Este trabalho busca contribuir de forma significativa para a conscientização e a adoção de medidas eficazes para lidar com os problemas associados à poluição do ar, oferecendo um instrumento valioso para a mitigação dos impactos ambientais e para a promoção da saúde pública.

## **1.1. Objetivos**

O objetivo deste trabalho é desenvolver uma estação de monitoramento de qualidade do ar utilizando tecnologias acessíveis, como Raspberry Pi e Arduino, com o propósito de avaliar a presença de partículas nocivas e gases poluentes em ambientes urbanos. Os objetivos específicos são:

Projetar e montar a estrutura da estação de monitoramento, integrando sensores de qualidade do ar compatíveis com a plataforma Arduino.

Implementar um sistema de armazenamento local utilizando Raspberry Pi para garantir a disponibilidade dos dados coletados, mesmo em situações de desconexão temporária da rede de comunicação.

Desenvolver um dashboard para visualização em tempo real dos dados coletados, permitindo o acesso remoto e a análise dos resultados por meio de uma interface web acessível.

Realizar testes e validações da estação de monitoramento em diferentes ambientes urbanos, avaliando sua precisão e confiabilidade na detecção de partículas nocivas e gases poluentes.

Analisar os dados coletados ao longo do tempo para identificar padrões de poluição do ar e fornecer insights para a conscientização e tomada de medidas mitigadoras.

Por meio desses objetivos, este trabalho visa contribuir para o desenvolvimento de soluções acessíveis e eficazes de monitoramento da qualidade do ar, com potencial para impactar positivamente a saúde pública e promover a conscientização ambiental.

## **1.2. Problema**

A poluição do ar é um problema que afeta pessoas em todo o mundo, causada por diferentes fontes, como as indústrias, a frota veicular das grandes cidades e também as queimadas. Tais fontes geram diversos poluentes que causam danos à saúde, ao meio ambiente e até mesmo à economia.

A base de evidências para os prejuízos causados pela poluição do ar vem crescendo rapidamente e aponta para danos significativos causados até mesmo por baixos níveis de muitos poluentes atmosféricos. No ano passado, a resposta da OMS se deu por meio da revisão de suas Diretrizes de Qualidade do Ar, de forma a refletir as evidências, tornando-as mais rigorosas, especialmente para material particulado e dióxido de nitrogênio, um movimento fortemente apoiado pela comunidade de saúde, associações médicas e organizações de pacientes. (ONU, 2022)

Reduzindo o campo de observação a nível nacional é possível perceber que a qualidade do ar no Brasil não é medida adequadamente, como aponta o Instituto de Energia e Meio Ambiente (IEMA):

A maioria dos estados brasileiros não realiza qualquer programa de monitoramento da qualidade do ar. Outros o fazem de forma fragmentada e descontínua; seja por não estabelecer prioridade diante do conjunto dos problemas ambientais; seja pelo desconhecimento da importância do

monitoramento como um instrumento direcionador da gestão ambiental.(IEMA, 2022)

De acordo com o G1, dos 27 estados do Brasil, apenas nove realizam o monitoramento da qualidade do ar, sendo mais crítico nas regiões Nordeste, Centro-Oeste e Norte, onde não há nenhum monitoramento.

A maioria das quase 250 estações de monitoramento automático do Brasil está na Região Sudeste, e os seus quatro estados lideram - de longe - o ranking de quantidade de estações.No topo da lista está o Rio de Janeiro, com 65 estações automáticas de monitoramento, seguido por São Paulo, com 62, Minas Gerais, com 54, e Espírito Santo, com 17. (G1, 2024).

Delimitando o território da pesquisa para a Cidade de São Paulo especificamente, onde o monitoramento é realizado pela Companhia Ambiental do Estado de São Paulo (CETESB), ultimamente realizado de duas maneiras:

Medição de Poluentes - Cada poluente é monitorado por equipamento específico. Na REDE AUTOMÁTICA, o ar é amostrado, analisado e os dados são enviados à central localizada na sede da CETESB o que permite divulgação, em tempo real, à população. Há também estações que possuem equipamentos que coletam amostras que são enviadas ao laboratório da CETESB para análise e constituem a denominada REDE MANUAL.(CETESB, 2023)

Ao observar os dados presentes no Relatório da Qualidade do Ar no Estado de São Paulo, nota-se a existência de 18 medidores fixos de qualidade atmosférica, geridos pela Companhia Ambiental do Estado de São Paulo (CETESB), destes, apenas três encontram-se em regiões consideradas periféricas. Considerando a densidade demográfica da cidade e as diferenças encontradas entre as regiões, nota-se que as regiões periféricas encontram-se em defasagem no que se refere à medição de poluentes na atmosfera, principalmente quando se considera a questão das regiões centrais da cidade.

Durante o processo de expansão da cidade, as populações com baixa renda passaram a ocupar e construir suas próprias casas às margens do centro(...) A população mais pobre é a que mais sofre com a poluição atmosférica, por passar mais tempo no trânsito e nas principais vias onde esses poluentes são emitidos e por ter menos tempo e recursos para cuidar da própria saúde". (LIMA, 2019).

Com o crescimento da cidade, a população considerada de baixa renda viu-se obrigada a ocupar as periferias da cidade, que cresceram de forma desordenada e

sem planejamento ambiental. Esta população geralmente encontra-se muito distante dos centros urbanos, onde há a maior quantidade de trabalho e necessita de maior deslocamento para seus locais de trabalho, muitas vezes se utilizando do transporte público e se expondo a maiores níveis de poluição atmosférica neste trajeto, e possuem menor tempo disponível para a busca de serviços de saúde, como pode confirmar o médico especialista em poluição e professor da Universidade de São Paulo , Paulo Saldiva em entrevista para Cesar Samson em 2012:

As casas das comunidades mais carentes são também as mais permeáveis à entrada de poluentes. Finalmente, é nos pontos de ônibus, pontos de alta concentração de poluentes, que a população mais desfavorecida passa longos períodos à espera do transporte(SALDIVA, 2012).

A falta de dados precisos prejudica a compreensão dos impactos na saúde pública e no meio ambiente, bem como a dificuldade ao acesso às informações por parte da população considerada carente da cidade. A poluição do ar é um problema que afeta pessoas em todo o mundo, causada por diferentes fontes, como as indústrias, a frota veicular das grandes cidades e até mesmo queimadas. Tais fontes geram diversos poluentes que causam danos à saúde, ao meio ambiente e até mesmo à economia.

### **1.3. Justificativa**

A preocupação com o ar que respiramos é extremamente importante para questões relacionadas à saúde pública, conseqüentemente o monitoramento da qualidade atmosférica é uma dentre muitas medidas essenciais para que ocorra a conscientização do problema e a tomada de medidas que visem diminuir as causas e conseqüências ocasionadas pela poluição do ar, conforme ressalta o Instituto de Energia e Meio Ambiente (IEMA):

O monitoramento da qualidade do ar, em conjunto com outros instrumentos de política pública, (tais como inventários de fontes emissoras, padrões de qualidade do ar e de emissões, etc), é essencial para assegurar uma

adequada gestão da qualidade do ar e, conseqüentemente, proteger a saúde pública.(IEMA 2022)

A realização deste projeto é fundamentada na necessidade e ampliação das estações de medição da qualidade do ar em regiões periféricas da Cidade de São Paulo, bem como na escassez de investimentos e políticas públicas direcionadas a fornecer informações essenciais sobre um recurso vital: o ar que respiramos.

O monitoramento atmosférico mostra-se extremamente importante em variados campos, não restringindo-se apenas ao campo da saúde pública, mas no âmbito social e econômico da sociedade.

A exposição à poluição do ar causa anualmente milhões de mortes e anos perdidos de vida saudável. Efeitos como as doenças respiratórias e cardiovasculares, baixo peso ao nascer, absenteísmo escolar, câncer de pulmão, aumento do risco de diabetes, perda de massa óssea, restrições à aquisição de habilidades cognitivas necessárias à educação, são alguns dos impactos da poluição do ar relatados na literatura científica.(GOUVEIA, VILLARDI ,2023)

O agrupamento e disseminação de dados referente à qualidade do ar possibilitam que os órgãos públicos possam buscar meios que possam auxiliar na resolução do problema. Para a população, o acesso facilitado à informação favorece que estas ações possam ser cobradas, a fim de buscar uma melhora na qualidade de vida e a diminuição de problemas de saúde.

O monitoramento de qualidade do ar é relevante, pois mensura as concentrações dos poluentes atmosféricos, gera dados sobre as condições atuais da qualidade do ar, constrói um histórico de dados e habilita os tomadores de decisão a planejar ações e políticas públicas no sentido de assegurar a boa qualidade do ar. (VORMITTAG, 2021)

A ausência de acesso a informações precisas sobre a qualidade do ar é uma questão de saúde pública e qualidade de vida. A população, especialmente aquela em situação de vulnerabilidade, necessita de recursos que permitam avaliar e monitorar os níveis de poluentes atmosféricos, dado seu impacto direto na saúde respiratória e no bem-estar geral. Diante desse contexto, este projeto se apresenta como uma resposta concreta e proativa para preencher essa lacuna informacional,

visando proporcionar à população em diferentes regiões do Brasil o acesso a dados confiáveis sobre a qualidade do ar.

Portanto, este projeto não apenas preenche uma lacuna crucial em termos de disponibilidade de dados sobre a qualidade do ar em regiões carentes, mas também representa um passo em direção à promoção da saúde pública e à conscientização ambiental, contribuindo para um futuro mais sustentável e saudável para todos.

## **1.4. Organização do Trabalho**

O estudo está dividido em cinco principais seções. A primeira seção oferece uma visão geral das tecnologias e sensores utilizados para a construção da estação de monitoramento da qualidade do ar, abordando aspectos de sistemas embarcados, plataformas de microcontroladores e comunicação de dados.

O segundo tópico detalha a metodologia e a implementação do projeto, fornecendo uma análise aprofundada sobre a integração dos sensores, como o DHT22, MQ-135, MQ-7, MQ-5, MQ-2 e MQ-131, com o microcontrolador Arduino e o Raspberry Pi. Esta seção explica como os dados são coletados, processados e armazenados, e como a interface de visualização é configurada utilizando o Grafana.

O terceiro tópico apresenta os resultados obtidos com a estação de monitoramento, discutindo as leituras de qualidade do ar e a eficácia dos sensores na detecção de poluentes e condições ambientais. Esta seção inclui gráficos e análises das medições realizadas durante o período de teste.

O quarto tópico aborda as conclusões tiradas a partir dos resultados obtidos, avaliando a precisão dos sensores e a funcionalidade geral do sistema de monitoramento. São discutidas as implicações dos dados coletados para a compreensão da qualidade do ar em comunidades carentes e as potencialidades do sistema para melhorar a saúde ambiental.

Finalmente, o quinto tópico identifica áreas para futuras melhorias e desenvolvimentos, sugerindo possíveis aprimoramentos na precisão dos sensores, na integração do sistema e na expansão da funcionalidade da estação de monitoramento para atender a um público ainda maior.

## **2. Tecnologias Relacionadas**

### **2.1. Arduino**

O Arduino Uno, uma plataforma de prototipagem eletrônica criada em 2005 por Massimo Banzi e David Cuartielles, visa facilitar o desenvolvimento de sistemas interativos acessíveis e de baixo custo. Alimentado pelo microcontrolador ATmega328P, o Arduino Uno oferece 14 pinos digitais de entrada e saída e 6 entradas analógicas, possibilitando uma ampla gama de interações com sensores e atuadores.

Com a capacidade de trocar informações com uma variedade de sistemas eletrônicos, o Arduino possibilita a criação de projetos como monitoramento de temperatura, controle de iluminação e transmissão de dados para sistemas remotos.

Outro destaque é que todo o material - software, bibliotecas e hardware - é de código aberto, permitindo seu uso sem a necessidade de pagamento de royalties ou direitos autorais. A programação é realizada por meio do Arduino IDE, utilizando uma linguagem derivada simplificada de C/C++, baseada no ambiente de programação Processing. A plataforma, sendo de código aberto, possibilita o uso gratuito de seu software, bibliotecas e hardware.

Com sua flexibilidade de hardware e software intuitivo, o Arduino Uno é uma ferramenta ideal para prototipagem ágil e desenvolvimento de projetos inovadores em áreas como robótica, automação e IoT.



Figura 1: Exemplo de Arduino UNO. Fonte: Google

## 2.2. Raspberry Pi 4

O Raspberry Pi 4, lançado em 2019 pela Raspberry Pi Foundation, é uma plataforma de computação avançada projetada para promover a educação em ciência da computação e viabilizar projetos de tecnologia e automação residencial com um desempenho superior a um custo acessível. Equipado com um processador ARM Cortex-A72 quad-core de 1,5 GHz e opções de 2 GB, 4 GB ou 8 GB de RAM, o Raspberry Pi 4 oferece um desempenho significativamente melhorado em relação aos modelos anteriores, garantindo uma maior capacidade de processamento e multitarefa.

Com uma variedade ampliada de portas, incluindo USB 3.0, USB 2.0, HDMI duplo, Ethernet Gigabit e GPIO, do Raspberry Pi 4 proporciona uma conectividade robusta e a capacidade de integrar diversos dispositivos e sensores externos. Este modelo também suporta armazenamento em massa por meio de um slot para cartão microSD e a possibilidade de conectar discos rígidos via USB, ampliando suas opções de armazenamento.

Sua compatibilidade com sistemas operacionais como Raspberry Pi OS (anteriormente conhecido como Raspbian), Ubuntu e outras distribuições Linux oferece flexibilidade no ambiente de desenvolvimento. A capacidade do Raspberry Pi 4 de rodar sistemas operacionais modernos e suportar a execução de várias

aplicações simultaneamente o torna uma plataforma ideal para uma ampla gama de projetos, desde automação residencial até servidores e aplicações complexas de Internet das Coisas (IoT).

Com sua combinação de desempenho melhorado, versatilidade e acessibilidade, o Raspberry Pi 4 se estabelece como uma ferramenta poderosa para a criação de soluções inovadoras em diversas áreas, desde educação e prototipagem até produtos comerciais e automação.



Figura 2 : Exemplo de RaspBerry Pi 4. Fonte: Multilógica Shop

### 2.3. Sensores

Os sensores AM2302, MQ-135, MQ-7, MQ-2, MQ-131 e DSM501A são os sensores escolhidos para o monitoramento de gases e qualidade do ar desse projeto. Projetados para serem compatíveis com a plataforma Arduino, esses sensores oferecem uma maneira acessível e eficaz de detectar e medir diferentes gases no ambiente.

O MQ-135 é um sensor de gases utilizado para a detecção de poluentes atmosféricos, incluindo gases tóxicos e contaminantes como dióxido de nitrogênio (NO<sub>2</sub>), amônia (NH<sub>3</sub>), benzeno (C<sub>6</sub>H<sub>6</sub>) e outros compostos orgânicos voláteis (COVs). Sua alta sensibilidade e resposta rápida o tornam ideal para monitorar a qualidade do ar em ambientes internos e externos, detectando uma ampla gama de poluentes que podem impactar a saúde humana e a qualidade do ambiente. O MQ-135 é amplamente utilizado em sistemas de monitoramento de qualidade do ar e em dispositivos que visam a proteção contra a exposição a gases nocivos



Figura 3 : Sensor MQ - 135. Fonte: Ebay

O Sensor MQ-7 é utilizado para detectar concentrações de monóxido de carbono (CO) no ar. Esse sensor, o MQ-7, pode ser usado em sistemas de monitoramento de qualidade do ar e detecção de vazamentos de CO.



Figura 4 : Sensor MQ - 7. Fonte: Ebay

O MQ-5 é capaz de detectar uma variedade de gases inflamáveis e tóxicos, como GLP, propano, álcool e fumaça. Sua faixa de detecção e capacidade de

resposta rápida o tornam uma escolha para uma variedade de aplicações, desde sistemas de alarme de incêndio até detecção de vazamentos de gás em ambientes industriais.



Figura 5 : Sensor MQ -5. Fonte: Ebay

O MQ-131 é um sensor de gás projetado para medir a concentração de ozônio ( $O_3$ ) em ambientes variados. O ozônio é um gás oxidante presente na estratosfera, onde protege a Terra dos raios ultravioleta, mas em concentrações elevadas na troposfera, pode ser prejudicial à saúde humana e ao meio ambiente. A monitoração da qualidade do ar é crucial para entender e controlar os níveis de poluentes atmosféricos, incluindo o ozônio.



Figura 6 : Sensor MQ -131. Fonte: Ebay

O AM2302 é um sensor digital de temperatura e umidade, amplamente utilizado em projetos de monitoramento ambiental devido à sua precisão e facilidade

de integração. Ele oferece medições confiáveis de temperatura em um intervalo de -40 a 80 graus Celsius, com uma precisão de  $\pm 0.5$  graus Celsius. Para a umidade, o AM2302 cobre uma faixa de 0% a 100% com uma precisão de  $\pm 2-5\%$  RH.

Esse sensor é ideal para aplicações que requerem monitoramento constante das condições ambientais, como em sistemas de automação residencial, estufas, e equipamentos de climatização. O AM2302 se conecta facilmente a microcontroladores como o Arduino e o Raspberry Pi, e é conhecido por sua estabilidade e baixo custo.

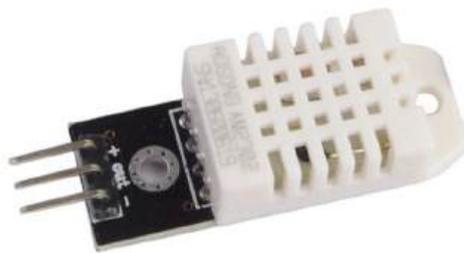


Figura 7 : Sensor AM2302. Fonte: Ebay

O DSM501A é um sensor de partículas utilizado para medir a concentração de poeira no ar. Projetado para monitorar a qualidade do ar, ele detecta partículas em suspensão de diferentes tamanhos, tipicamente PM10 e PM2.5, que são críticos para avaliar a poluição atmosférica.

Esse sensor opera com base em um princípio óptico, onde um feixe de luz é projetado através do fluxo de ar. Partículas no ar dispersam a luz, e o sensor detecta essa dispersão para quantificar a concentração de partículas. A saída do DSM501A é um sinal analógico que pode ser convertido em valores de concentração de partículas usando um microcontrolador, como o Arduino.

Sua alta sensibilidade e resposta rápida o tornam adequado para aplicações em sistemas de monitoramento da qualidade do ar, ajudando a identificar níveis de poluição e seus impactos potenciais na saúde. Comumente utilizado em ambientes

internos e externos, o DSM501A fornece dados importantes para estudos ambientais e iniciativas de controle de poluição.



Figura 8 : Sensor DSM501A. Fonte: Ebay

## 2.4. Maria DB

O MariaDB é um banco de dados de código aberto de alta performance, desenvolvido pela comunidade como um fork do MySQL. É amplamente utilizado para armazenar dados em diversos formatos, incluindo aplicações em áreas como monitoramento de operações, dados coletados de sensores IoT e análises em tempo real.

## 2.5. Node-RED

O Node-RED é uma ferramenta de código aberto em formato visual, desenvolvida pela IBM Emerging Technology, desenvolvida para criar e conectar dispositivos, além de ser amplamente utilizada em hardwares, APIs e web services. Esta ferramenta utiliza um editor baseado em navegador, simples e compatível com todos os web browsers, permitindo a criação de fluxos de aplicação por meio de nós (ou nodes).

O Node.js é um ambiente de execução que possibilita a programação em JavaScript dentro do editor do Node-RED, utilizando rich text. O Node-RED inclui diversa variedade de nós padrão, mas também permite a adição de nós extras para

diversas funcionalidades, como integração com clouds, dashboards, sensores, entradas/saídas (I/O), funções personalizadas, acesso a mídias sociais e bancos de dados.

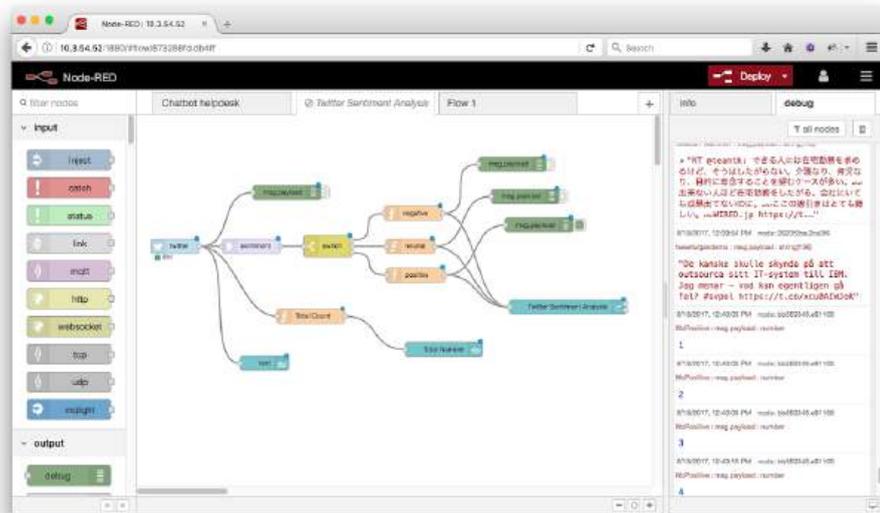


Figura 9 : Exemplo RedNode. Fonte: IBM

## 2.6. Grafana

O Grafana é uma plataforma de código aberto desenvolvida para visualização de dados e monitoramento em tempo real. Utilizada em áreas como monitoramento de operações, análise de dados de sensores IoT e dashboards analíticos, Grafana permite a criação de painéis interativos que exibem informações de maneira clara e intuitiva.

A plataforma oferece suporte robusto para diversos tipos de fontes de dados, incluindo bancos de dados como MariaDB, possibilitando a conexão e a visualização direta dos dados armazenados. A configuração do Grafana é simples e flexível, permitindo a personalização dos painéis de acordo com as necessidades específicas dos usuários.

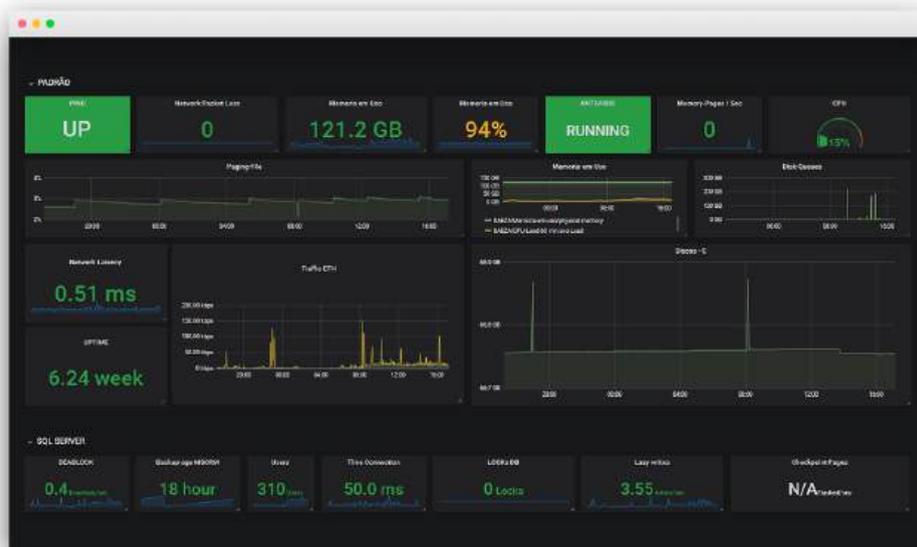


Figura 10 : Exemplo Grafana. Fonte: IBM

### 3. Metodologia e Implementação

#### 3.1. Visão geral

O projeto da Estação de Medição da Qualidade do Ar, desenvolvido com Raspberry Pi e Arduino, tem como objetivo criar um sistema acessível para monitorar a qualidade do ar em regiões com falta de tecnologia adequada. O projeto visa fornecer dados em tempo real sobre a qualidade do ar, auxiliando na saúde pública e na conscientização ambiental.

A estação utiliza uma variedade de sensores conectados ao Arduino para medir diferentes parâmetros da qualidade do ar. Os sensores empregados são:

- MQ-135: Para medir a concentração de gases como NO<sub>2</sub> e outros poluentes.
- MQ-7: Para a detecção de monóxido de carbono (CO).
- MQ-2: Para detectar uma gama de gases, incluindo fumaça e gases combustíveis.
- MQ-131: Para a medição de ozônio (O<sub>3</sub>).
- DMS501-A: Para medir partículas em suspensão no ar, como PM<sub>10</sub> e PM<sub>2.5</sub>.
- AM2302: Para medir a temperatura e a umidade do ambiente.

Esses sensores enviam os dados para um Arduino Uno, que gerencia o funcionamento dos sensores e a coleta das informações.

O Raspberry Pi serve como o núcleo do sistema, gerenciando a recepção e o armazenamento dos dados em um banco de dados local. Este banco de dados organiza e processa as informações coletadas pelo Arduino, que se comunica com o Raspberry Pi através de uma conexão serial. Para a visualização dos dados, o Grafana é utilizado, uma plataforma de código aberto que cria painéis interativos e dinâmicos. Esses painéis exibem as leituras da qualidade do ar de forma clara e intuitiva, permitindo o monitoramento contínuo e a análise dos dados em tempo real. Abaixo está o fluxograma do funcionamento do projeto:

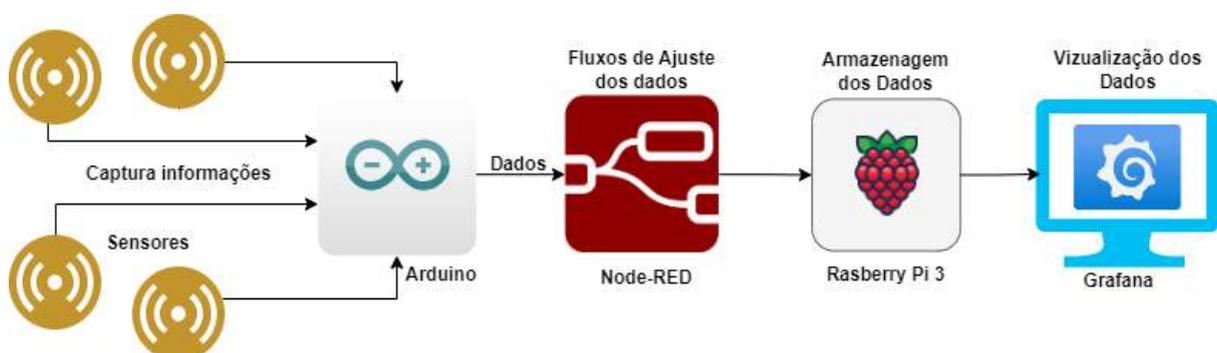


Figura 11 : Apresentando todos os componentes do sistema. Fonte: própria

Os dispositivos físicos que constituem a estação de medição de qualidade do ar são representados na Figura 12 abaixo.

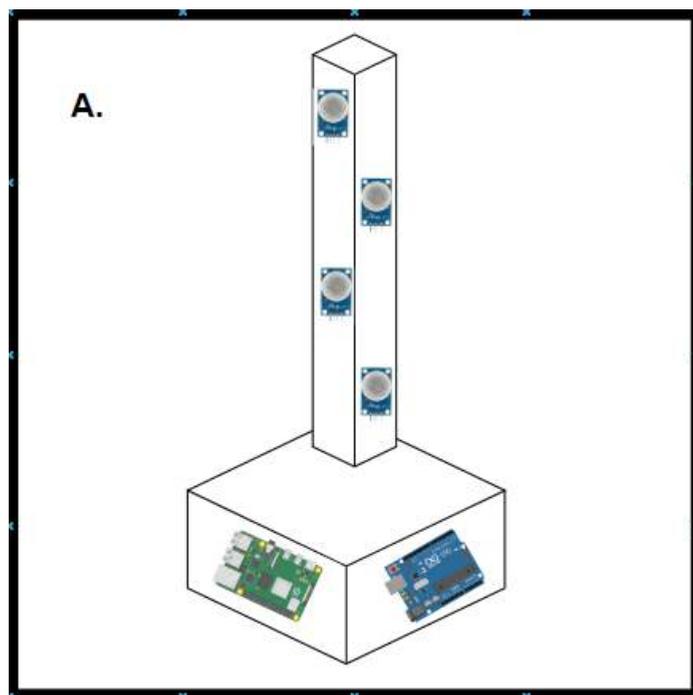


Figura 12 : Exemplo Modelo Físico. Fonte: própria

## 3.2. Análise de Requisitos

### 3.2.1. Requisitos Funcionais

**RF01:** Medir e transmitir dados de vários sensores (PM2.5, PM10, CO, NO2, O3, SO2, temperatura, umidade) simultaneamente num microcontrolador.

**RF02:** Armazenar dados dos sensores como séries temporais em um banco de dados MariaDB e InfluxDB.

**RF03:** Gerenciar o acesso de usuários ao sistema, permitindo diferentes níveis de permissões.

**RF04:** GUI (Graphical User Interface) para explorar dados das séries temporais graficamente e em tabelas, utilizando Grafana.

**RF05:** Análise de séries temporais e previsão de comportamentos futuros baseados nos dados coletados.

**RF06:** Exportar os dados coletados para planilhas para análises externas e relatórios.

**RF07:** Realizar cálculos das médias das medições durante um período específico, como o realizado durante os 8 dias de teste.

**RF08:** Atualizar as medições dos sensores a cada 5 minutos e refletir essas atualizações em tempo real na GUI.

### **3.2.2. Requisitos Não-Funcionais**

RNF01: Custo baixo (Max. R\$1000).

RNF02: Interface de fácil utilização, com dashboards intuitivos e acessíveis.

RNF03: Atender aos padrões de qualidade do ar do órgão CONAMA.

RNF04: Possuir alertas configuráveis para quando os níveis de poluentes excederem determinados limites.

RNF05: Garantir a segurança dos dados armazenados no banco de dados.

RNF06: O sistema deve ser capaz de operar de maneira confiável em um ambiente fechado.

### 3.3. Arquitetura do Projeto

Como pode ser visto nas Figuras 11 e 12, a arquitetura final do projeto, incluindo o modelo do protótipo, possui quatro módulos, consistindo de:

**Microcontrolador Integrado:** Um microcontrolador (Arduino) integrado com sensores para a medição de gases e outras grandezas definidas na pesquisa, enviando dados através da comunicação serial entre o Arduino e o Raspberry Pi.

**Serviço Node-Red:** Um serviço Node-Red responsável pela integração entre os dispositivos físicos e o banco de dados. O serviço inclui uma configuração para receber os dados dos sensores, filtrar as informações do JSON recebido e encaminhá-las para o módulo do banco de dados.

**Banco de Dados:** Um banco de dados MariaDB onde as informações são armazenadas no formato de séries temporais a partir do JSON recebido do módulo integrado na aplicação Node-Red.

**Instância do Grafana:** Uma instância do Grafana configurada no Raspberry Pi, um aplicativo com interface gráfica que possui integração nativa com o banco de dados, utilizando controle de acesso com login e senha para visualizar e analisar os dados.

### 3.4. Custos do Desenvolvimento do Projeto

Para criar uma estação de monitoramento de qualidade do ar com um custo reduzido, foi estabelecida uma meta de valor de R\$1.000,00. Abaixo, segue uma tabela comparativa dos custos dos sensores e sistemas embarcados utilizados no desenvolvimento deste projeto:

| Sensor        | Valores    |               |
|---------------|------------|---------------|
| MQ-135        | R\$        | 27,00         |
| MQ-131        | R\$        | 80,00         |
| MQ-7          | R\$        | 9,00          |
| MQ-5          | R\$        | 9,00          |
| DSM501-A      | R\$        | 47,00         |
| AM2302        | R\$        | 20,00         |
| Arduino       | R\$        | 170,00        |
| Raspberry pi4 | R\$        | 600,00        |
| <b>Total</b>  | <b>R\$</b> | <b>962,00</b> |

Tabela 1 : Custo do Projeto. Fonte: própria

Embora o custo total tenha sido de R\$962,00, o que representa uma economia em relação à meta de valor estabelecida de R\$1.000,00, a abordagem econômica foi fundamental para garantir a viabilidade financeira do projeto. Isso possibilitou a implementação de uma solução eficaz para a medição da qualidade do ar dentro de um orçamento controlado.

Para contextualizar, esta abordagem de baixo custo é particularmente relevante quando comparada com estações de medição de qualidade do ar profissionais, cujos preços podem variar de R\$2.000,00 a R\$50.000,00. A diferença de preço reflete a variedade de funcionalidades, precisão e robustez das soluções profissionais, que são projetadas para atender a necessidades específicas e ambientes exigentes. Este projeto demonstra que é possível obter uma solução eficiente e funcional para monitoramento de qualidade do ar a um custo significativamente reduzido, adequando-se a orçamentos mais restritos sem comprometer a eficácia.

### 3.5. Dispositivo de medição de poluentes

Para a construção do dispositivo de medição dos poluentes, optou-se pelo uso de um microcontrolador Arduino UNO R3 devido à sua disponibilidade e baixo custo. O Arduino UNO R3 é equipado com um microcontrolador ATmega328, oferecendo 32KB de memória flash e 2KB de SRAM para armazenamento de variáveis. Ele conta com seis portas analógicas e vinte portas digitais, podendo

operar em tensões de 3V a 5V, o que proporciona flexibilidade e eficiência na integração com diversos sensores.

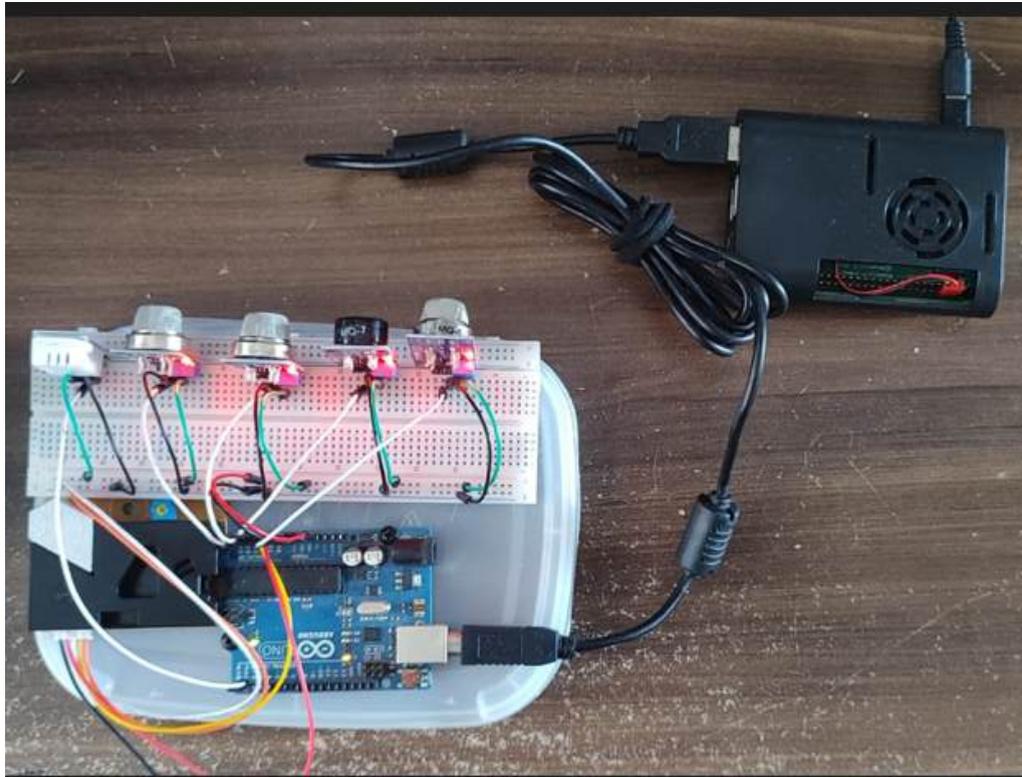


Figura 13 : Protótipo em Protoboard. Fonte: própria

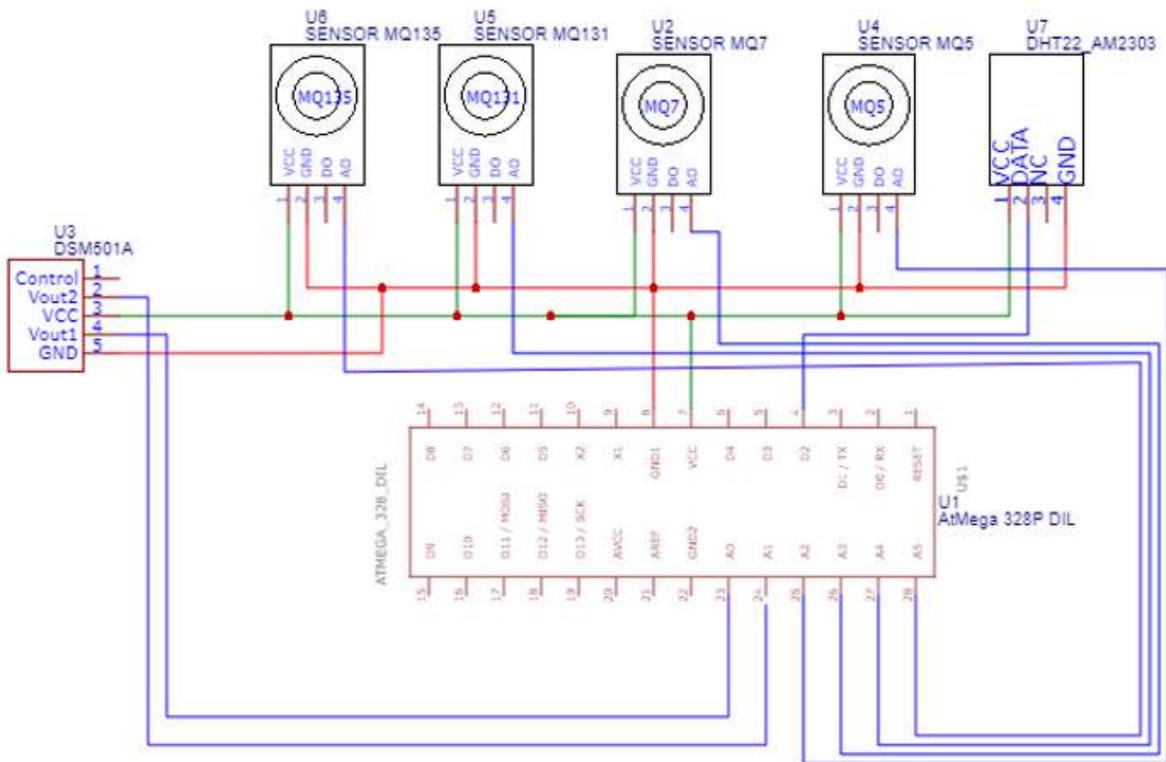


Figura 14 : – Esquemático da montagem do dispositivo. Fonte: própria

Os sensores adquiridos para o projeto podem ser vistos na Tabela 1. Para garantir a medição precisa de diferentes poluentes e condições ambientais, foram selecionados sensores específicos para cada tipo de poluente:

| Sensor   | MP 10 | MP 2.5 | SO2 | NO2 | Fumaça | CO2 | Umidade | Temperatura |
|----------|-------|--------|-----|-----|--------|-----|---------|-------------|
| MQ-135   |       |        |     | X   |        |     |         |             |
| MQ-131   |       |        | X   |     |        |     |         |             |
| MQ-7     |       |        |     |     |        | X   |         |             |
| MQ-5     |       |        |     |     | X      |     |         |             |
| DSM501-A | X     | X      |     |     |        |     |         |             |
| AM2302   |       |        |     |     |        |     | X       | X           |

Tabela 2 : Sensores. Fonte: própria

- **MQ-135:** Para medir a concentração de gases como NO2 e outros poluentes.
- **MQ-7:** Para a detecção de monóxido de carbono (CO).
- **MQ-5:** Para detectar uma gama de gases, incluindo fumaça e gases combustíveis.
- **MQ-131:** Para a medição de ozônio (O3).
- **DSM501-A:** Para medir partículas em suspensão no ar, como PM10 e PM2.5.

- **AM2302:** Para medir a temperatura e a umidade do ambiente.

Durante o desenvolvimento do programa para o microcontrolador, foi utilizada a IDE do Arduino para escrever o código-fonte, por ser simples e facilitar o carregamento no dispositivo. Uma das dificuldades encontradas nesta etapa do projeto foi converter os valores analógicos dos sensores para concentrações conhecidas, o que exigiu o uso de bibliotecas específicas.

Para os sensores MQ, foi utilizada a biblioteca GasSensor para converter os valores analógicos medidos em concentrações em ppm. O sensor DSM501-A, que retorna valores de partículas suspensas e poeira, utiliza uma leitura digital mais complexa, consumindo mais memória do que os sensores analógicos.

O programa final possui o seguinte fluxo de operação, apresentado na Figura 13:

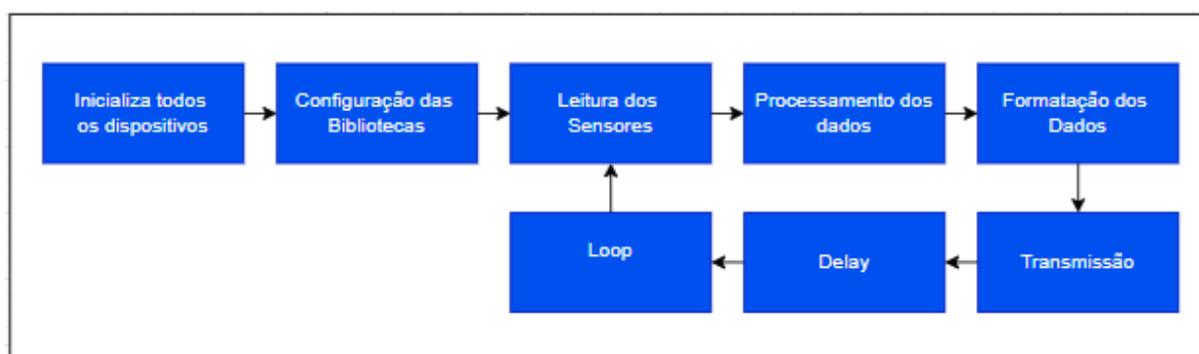


Figura 15 : Fluxograma do programa feito no Arduino. Fonte: própria

- Inicializa todos os dispositivos do Arduino e os sensores.
- Utiliza funções específicas das bibliotecas para definir os parâmetros das curvas de conversão.
- Inicia a leitura das portas analógicas do Arduino para receber os valores medidos pelos sensores.
- Chama as funções que determinam a concentração correta de cada poluente em ppm.
- Insere os dados coletados em um objeto em bytes, que é colocado na fila de transmissão para ser enviado para a aplicação em Node-RED.
- Após um atraso de um minuto, o processo de leitura das medições dos sensores é repetido em loop.

### 3.6. Integração via Node-RED

O módulo Node-RED desempenha funções cruciais na integração e armazenamento dos dados coletados pelos sensores. Suas principais funções são: receber os dados do dispositivo de medição, converter essas informações para o formato JSON, filtrar e formatar os dados para torná-los compatíveis com as tabelas do banco de dados e, por fim, armazená-los na instância do banco de dados MariaDB. O diagrama dos nós responsáveis por cada uma dessas funções pode ser visto na Figura 15.

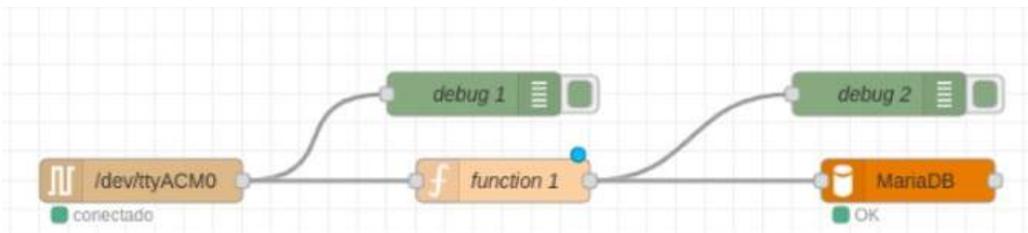


Figura 16 : Fluxo do Node-Red. Fonte: própria

O nó MQTT, denominado “Dados in”, é responsável por conectar-se ao dispositivo remotamente através da rede The Things Network. Ele recebe a mensagem contendo os dados e a passa para o próximo nó, que realiza a conversão dos dados para o formato JSON. Em seguida, o objeto JSON é encaminhado para um nó de processamento em JavaScript, que extrai as medidas e identificadores necessários. Esses dados são então inseridos no banco de dados através do nó de inserção, integrado e conectado ao MariaDB.

### 3.7. Banco de dados Mariadb

A escolha do banco de dados MariaDB foi feita devido às suas características que atendem bem aos requisitos do projeto. O MariaDB é um banco de dados relacional que oferece flexibilidade na criação e manipulação de tabelas, suporte para operações complexas de consulta e uma integração robusta com o Node-RED para armazenamento e leitura de dados. Além disso, o MariaDB é amplamente

reconhecido por sua facilidade de instância, configuração e navegação pelas tabelas.

No projeto, os dados são configurados para serem inseridos no banco a cada 5 minutos. Esses dados são organizados automaticamente nas colunas apropriadas, conforme o formato do objeto JSON recebido. As tabelas são estruturadas com base nos identificadores do dispositivo de origem, seguidos pelos valores das concentrações dos respectivos gases, garantindo uma organização eficiente e acessível para análise futura.

### 3.8. Grafana

O dashboard foi configurado para apresentar uma visão abrangente dos dados coletados pelos sensores. Ele inclui gráficos individuais para cada tipo de medição realizada, como as concentrações de gases específicos, bem como um gráfico comparativo que exibe todos os gases medidos simultaneamente. Esta configuração permite uma análise detalhada e eficiente das diferentes variáveis monitoradas, conforme ilustrado na Figura 17.



Figura 17 : Interface do Grafana. Fonte: própria.

Durante o desenvolvimento do dashboard, a principal dificuldade encontrada foi a adaptação às queries do MariaDB para a integração com o Grafana. Embora as queries do MariaDB sejam relativamente semelhantes às queries comuns do MySQL, houve um período de aprendizado para ajustar as consultas e garantir que os dados fossem corretamente formatados e apresentados no Grafana. Apesar desse desafio inicial, a semelhança entre as sintaxes facilitou a adaptação, permitindo a criação de visualizações eficazes e a integração completa dos dados com o painel do Grafana.

## **4. Resultados**

O desenvolvimento da estação de medição da qualidade do ar alcançou com sucesso os objetivos estabelecidos, resultando em uma solução integrada e eficaz para o monitoramento ambiental. Este projeto envolveu a combinação de sensores, microcontroladores e plataformas de software para criar um sistema robusto e confiável que mede, processa e visualiza dados sobre a qualidade do ar. A seguir, são apresentados os principais resultados obtidos, incluindo detalhes sobre a otimização dos códigos utilizados e a eficácia geral do sistema.

### **4.1. Desempenho dos Sensores**

A escolha e a integração dos sensores foram aspectos cruciais para o sucesso do projeto. Cada sensor foi selecionado com base em suas características específicas, contribuindo significativamente para a obtenção de dados detalhados e precisos sobre a qualidade do ar:

Foi realizado um acompanhamento dos dados durante oito dias em um ambiente fechado, utilizando o protótipo montado em protoboard. O objetivo foi observar as variações e o desempenho do dispositivo ao longo desse período. As medições coletadas foram utilizadas para calcular a média dos valores registrados

durante os oito dias, os quais foram posteriormente exportados para uma planilha para análise e documentação.

| Data       | PM2.5 ( $\mu\text{g}/\text{m}^3$ ) | PM10 ( $\mu\text{g}/\text{m}^3$ ) | Fumaça (ppm) | CO (ppm) | NO2 (ppm) | O3 (ppm) | SO2 (ppm) | Temperatura ( $^{\circ}\text{C}$ ) | Umidade (%) |
|------------|------------------------------------|-----------------------------------|--------------|----------|-----------|----------|-----------|------------------------------------|-------------|
| 01/07/2024 | 4.09                               | 3.49                              | 4.87         | 4.87     | 4.81      | 4.87     | 0.00      | 23.60                              | 43.00       |
| 02/07/2024 | 4.08                               | 3.54                              | 4.87         | 4.87     | 4.82      | 4.88     | 0.01      | 23.60                              | 44.40       |
| 03/07/2024 | 4.08                               | 3.51                              | 4.88         | 4.86     | 4.81      | 4.87     | 0.03      | 23.60                              | 43.60       |
| 04/07/2024 | 4.09                               | 3.49                              | 4.87         | 4.87     | 4.81      | 4.87     | 0.03      | 23.60                              | 43.00       |
| 05/07/2024 | 4.09                               | 3.49                              | 4.88         | 4.87     | 4.81      | 4.88     | 0.04      | 23.70                              | 43.00       |
| 06/07/2024 | 4.09                               | 3.48                              | 4.87         | 4.87     | 4.81      | 4.87     | 0.03      | 23.60                              | 42.90       |
| 07/07/2024 | 4.09                               | 3.49                              | 4.87         | 4.87     | 4.81      | 4.87     | 0.02      | 19.20                              | 43.00       |
| 08/07/2024 | 4.09                               | 3.49                              | 4.88         | 4.87     | 4.81      | 4.88     | 0.01      | 22.00                              | 43.00       |

Tabela 3 : Acompanhamento dos 8 dias do Projeto. Fonte: própria.

**MQ-135:** Este sensor foi utilizado para medir a concentração de gases como óxidos de nitrogênio ( $\text{NO}_2$ ) e outros poluentes atmosféricos. O MQ-135 é conhecido por sua capacidade de detectar uma ampla gama de gases nocivos, o que é essencial para a avaliação abrangente da qualidade do ar. No projeto, ele forneceu dados consistentes sobre a presença de poluentes gasosos, possibilitando uma análise detalhada e a identificação de fontes de poluição, abaixo segue imagens da captura do gás mostrados no compilador do arduino e também no Grafana.



Figura 18 : Medição NO2. Fonte: IDE Arduino

**MQ-7:** O sensor MQ-7 é especializado na detecção de monóxido de carbono (CO). Este gás é um dos principais indicadores de poluição atmosférica e tem implicações significativas para a saúde pública. O MQ-7 demonstrou alta sensibilidade e precisão na medição do CO, fornecendo dados cruciais para a monitorização da qualidade do ar e ajudando a identificar níveis potencialmente

perigosos deste poluente. Abaixo segue imagens da captura do gás mostrados no compilador do arduino e também no Grafana.



Figura 19 : Medição CO. Fonte: IDE Arduino

**MQ-5:** O MQ-5 é um sensor versátil que detecta uma ampla gama de gases, incluindo fumaça e gases combustíveis. Sua flexibilidade e capacidade de detectar diferentes tipos de poluentes gasosos foram fundamentais para a obtenção de uma visão abrangente da qualidade do ar. O sensor MQ-5 contribuiu significativamente para identificar fontes de poluição em ambientes variados, melhorando a eficácia do sistema de monitoramento. Abaixo segue imagens da captura do gás mostrados no compilador do arduino e também no Grafana.



Figura 20 : Medição Fumaça. Fonte: IDE Arduino

**MQ-131:** Este sensor é especializado na medição de ozônio (O<sub>3</sub>). O ozônio é um poluente atmosférico com impactos significativos na saúde e no meio ambiente. O MQ-131 forneceu dados precisos sobre as concentrações de ozônio, permitindo uma análise detalhada e a identificação de condições atmosféricas que podem afetar a qualidade do ar. Abaixo segue imagens da captura do gás mostrados no compilador do arduino e também no Grafana.



Figura 21 : Medição O3. Fonte: IDE Arduino

**DSM501-A:** O sensor DSM501-A foi utilizado para medir partículas em suspensão no ar, especificamente PM10 e PM2.5. A detecção de partículas é crucial para avaliar a poluição particulada, que pode ter sérios efeitos sobre a saúde respiratória. O DSM501-A fornece medições detalhadas e precisas das partículas no ar, contribuindo para uma compreensão mais completa da qualidade do ar e ajudando a identificar períodos de alta poluição particulada. Abaixo segue imagens da captura do gás mostrados no compilador do arduino e também no Grafana.



Figura 22 : Micropartículas de 2.5 µg/m³ e 10 µg/m³. Fonte: IDE Arduino

**AM2302:** O sensor AM2302 foi utilizado para medir a temperatura e a umidade do ambiente. A temperatura e a umidade têm impacto direto na operação dos sensores e nas condições ambientais. O AM2302 forneceu dados valiosos que ajudaram a interpretar as medições dos outros sensores, considerando as variações climáticas que podem influenciar a qualidade do ar.



Figura 23 : Temperatura Umidade. Fonte: IDE Arduino

## 4.2. Processamento e Armazenamento de Dados

O Node-RED desempenhou um papel central na integração e processamento dos dados dos sensores. A integração eficiente dos dados foi crucial para garantir a precisão e a confiabilidade das informações armazenadas e analisadas. As principais conquistas nesta etapa foram:

**Eficiência na Integração:** O Node-RED foi configurado para receber dados dos sensores via Serial, processar essas informações e convertê-las para o formato JSON. Este processo garantiu a integração eficiente com o banco de dados MariaDB, possibilitando um fluxo contínuo e organizado de dados.

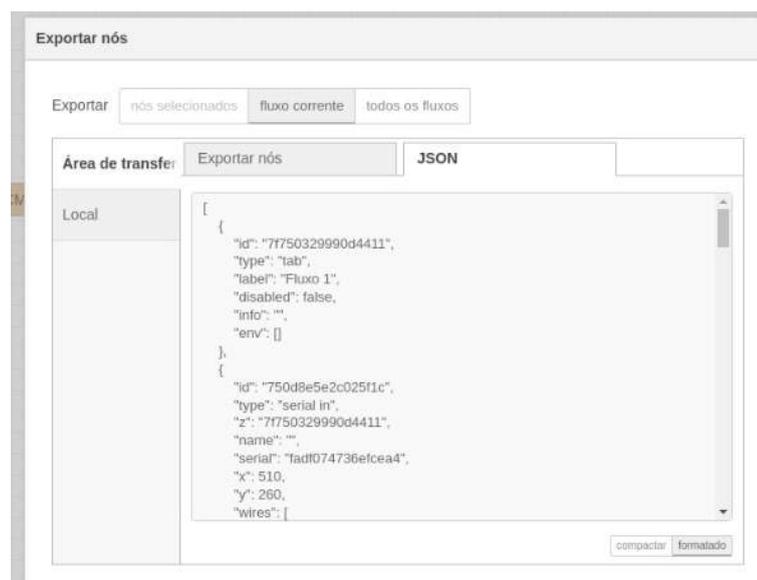


Figura 24 : JSON Gerado pelo NODE-RED. Fonte: NODE-RED.

Armazenamento no MariaDB: O banco de dados MariaDB foi escolhido pela sua capacidade de gerenciamento de dados relacionais. Os dados foram armazenados a cada minuto, organizados em tabelas com identificadores de dispositivos e concentrações dos gases. Esta estrutura facilitou a análise e o acesso às informações, permitindo uma visualização clara e acessível dos dados coletados.

```
MariaDB [sensor_database]> SELECT * FROM sensor_data;
```

| id | mq7  | mq135 | mq131 | mq5  | temperature | humidity | pm25  | timestamp           |
|----|------|-------|-------|------|-------------|----------|-------|---------------------|
| 1  | 1.14 | 1.35  | 0.89  | 2.49 | 24.6        | 48.8     | 0.62  | 2024-09-01 12:33:42 |
| 2  | 1.15 | 1.35  | 0.87  | 2.51 | 24.6        | 47.6     | 0.62  | 2024-09-01 12:34:42 |
| 3  | 1.12 | 1.33  | 0.86  | 2.47 | 24.6        | 47.7     | 0.62  | 2024-09-01 12:35:43 |
| 4  | 1.11 | 1.32  | 0.86  | 2.45 | 24.5        | 48.4     | 0.62  | 2024-09-01 12:36:43 |
| 5  | 1.11 | 1.3   | 0.85  | 2.42 | 24.5        | 47.9     | 0.62  | 2024-09-01 12:37:43 |
| 6  | 1.11 | 1.3   | 0.86  | 2.41 | 24.5        | 47.9     | 0.62  | 2024-09-01 12:38:43 |
| 7  | 1.11 | 1.3   | 0.87  | 2.41 | 24.5        | 48.3     | 0.62  | 2024-09-01 12:39:43 |
| 8  | 1.1  | 1.3   | 0.87  | 2.42 | 24.4        | 48.5     | 0.62  | 2024-09-01 12:40:43 |
| 9  | 1.1  | 1.28  | 0.85  | 2.39 | 24.5        | 47.2     | 0.62  | 2024-09-01 12:41:43 |
| 10 | 1.11 | 1.3   | 0.86  | 2.41 | 24.6        | 47.4     | 0.62  | 2024-09-01 12:42:43 |
| 11 | 1.1  | 1.32  | 0.86  | 2.42 | 24.7        | 47.5     | 37.38 | 2024-09-01 12:43:43 |
| 12 | 1.11 | 1.33  | 0.86  | 2.46 | 24.7        | 47.9     | 0.62  | 2024-09-01 12:44:43 |
| 13 | 1.12 | 1.34  | 0.87  | 2.48 | 24.7        | 47.3     | 0.62  | 2024-09-01 12:45:43 |
| 14 | 1.11 | 1.31  | 0.86  | 2.43 | 24.7        | 47.4     | 0.95  | 2024-09-01 12:46:43 |
| 15 | 1.1  | 1.32  | 0.87  | 2.43 | 24.8        | 47.5     | 0.62  | 2024-09-01 12:47:43 |
| 16 | 1.11 | 1.33  | 0.87  | 2.44 | 24.8        | 46.8     | 13.76 | 2024-09-01 12:48:43 |
| 17 | 1.1  | 1.32  | 0.87  | 2.43 | 24.9        | 46.8     | 4.65  | 2024-09-01 12:49:43 |
| 18 | 1.11 | 1.3   | 0.85  | 2.41 | 25          | 45.7     | 0.62  | 2024-09-01 12:50:44 |
| 19 | 1.11 | 1.33  | 0.86  | 2.45 | 25.1        | 46.1     | 14.92 | 2024-09-01 12:51:43 |

Figura 25 : Consulta ao Banco de Dados. Fonte: MariaDB.

### 4.3. Visualização e Análise de Dados

O Grafana foi utilizado para criar um dashboard detalhado, permitindo a visualização e análise dos dados coletados. Os principais resultados nesta etapa incluíram:

Gráficos Detalhados: O dashboard do Grafana foi configurado para exibir gráficos individuais para cada sensor, além de gráficos comparativos que mostram as concentrações de todos os gases simultaneamente. Esta configuração proporcionou uma visão abrangente da qualidade do ar e facilitou a interpretação dos dados.

Alertas Configurados: Foram configurados alertas para notificar quando os valores medidos excederem os limites estabelecidos. Esses alertas foram essenciais para a rápida identificação de condições anômalas e para a tomada de decisões informadas sobre a qualidade do ar.



Figura 26 : Interface do Projeto. Fonte: própria

Facilidade de Uso: A interface gráfica do Grafana permitiu uma análise intuitiva dos dados, facilitando a identificação de padrões e tendências na qualidade do ar. A integração com o MariaDB garantiu que as visualizações fossem atualizadas em tempo real, proporcionando uma visão precisa e atualizada dos dados.

#### 4.4. Otimização dos Códigos Utilizados

Durante o desenvolvimento do projeto, a otimização dos códigos foi uma etapa crucial para garantir o desempenho e a eficiência do sistema. As principais ações de otimização incluíram:

Refatoração do Código: O código foi otimizado para reduzir o consumo de memória e melhorar a eficiência do processamento. Funções redundantes foram removidas e o código foi organizado para facilitar a manutenção e a escalabilidade.

Melhoria na Conversão de Dados: Foram implementadas melhorias na conversão dos dados dos sensores, garantindo que os valores analógicos fossem transformados em concentrações precisas com o mínimo de impacto no desempenho do sistema.

Ajustes na Comunicação entre Componentes: A comunicação entre o Arduino, Node-RED e MariaDB foi otimizada para garantir uma transmissão de dados eficiente e sem falhas. Foram feitos ajustes na configuração do MQTT e no processamento dos dados para reduzir a latência e melhorar a integridade das informações.

## **4.5. Considerações Finais**

Os resultados obtidos ao longo deste projeto evidenciam a eficácia e a robustez da solução desenvolvida para o monitoramento da qualidade do ar. A integração de sensores precisos, combinada com um processamento eficiente dos dados e a apresentação de visualizações detalhadas, permitiu criar uma ferramenta abrangente e eficaz para a avaliação ambiental. A implementação bem-sucedida desta solução não apenas alcançou seus objetivos iniciais, mas também estabeleceu uma base sólida para futuras melhorias e expansões.

Durante o desenvolvimento, foi possível observar a importância de uma abordagem integrada que une diferentes tecnologias para obter um monitoramento ambiental preciso. A capacidade dos sensores em fornecer dados confiáveis, aliada à eficiência do processamento desses dados, garantiu uma análise detalhada e precisa das condições do ar. As visualizações geradas pelo sistema permitem uma compreensão clara e imediata das variações na qualidade do ar, facilitando a tomada de decisões informadas para a gestão ambiental.

Além de atingir seus objetivos, o projeto proporcionou uma rica experiência e uma série de insights valiosos sobre o campo do monitoramento ambiental. A jornada de desenvolvimento revelou a complexidade e a importância de monitorar a

qualidade do ar, especialmente em contextos urbanos e industriais onde a poluição pode ter impactos significativos na saúde pública. A integração de tecnologias avançadas neste projeto demonstrou como a inovação pode ser aplicada para enfrentar desafios ambientais modernos e melhorar a qualidade de vida das pessoas.

A solução desenvolvida oferece uma abordagem prática e acessível para o monitoramento da qualidade do ar, destacando a relevância de tecnologias avançadas na gestão e melhoria das condições ambientais. O sistema não só fornece uma resposta eficiente às necessidades atuais, mas também abre portas para o desenvolvimento de novas funcionalidades e a integração com outras plataformas e tecnologias.

Em resumo, este projeto reafirma a importância do avanço tecnológico na área ambiental e ilustra como, com a aplicação adequada de tecnologias, podemos fazer uma diferença significativa na preservação e melhoria do nosso ambiente. A experiência adquirida ao longo do desenvolvimento não apenas contribuiu para o avanço na área de tecnologia ambiental, mas também proporcionou uma perspectiva enriquecedora sobre o impacto positivo que a tecnologia pode ter na qualidade do ar e na saúde pública. O caminho percorrido destaca a capacidade da inovação em enfrentar desafios globais e oferece um modelo para futuros projetos que visem melhorar a qualidade de vida e o bem-estar das comunidades.

## 5. Conclusões

Este trabalho teve como objetivo o desenvolvimento e implementação de uma estação de medição da qualidade do ar, que se mostrou eficaz na avaliação e monitoramento de diversos poluentes atmosféricos. Através da integração de tecnologias modernas e da otimização de processos, o projeto conseguiu fornecer uma solução robusta e confiável para a análise ambiental.

Durante o desenvolvimento, foram enfrentados diversos desafios técnicos, que foram superados com sucesso através de ajustes precisos e melhorias contínuas. A escolha dos componentes e a configuração do sistema foram realizadas com o objetivo de garantir a máxima eficiência e precisão nas medições. A utilização de sensores especializados, combinada com uma arquitetura de software eficiente, permitiu a obtenção de dados detalhados e relevantes sobre a qualidade do ar.

Um aspecto crucial do projeto foi a otimização do código, que envolveu a refatoração e ajuste das funções para garantir um desempenho ágil e eficiente. A conversão dos dados dos sensores foi aprimorada, garantindo que os sinais analógicos fossem convertidos com precisão para valores significativos, minimizando o impacto no desempenho geral do sistema. A comunicação entre os diferentes componentes do sistema foi ajustada para reduzir a latência e garantir a integridade dos dados.

O processamento e armazenamento dos dados foram realizados de forma eficiente utilizando a plataforma Node-RED, que possibilitou a integração entre os sensores e o banco de dados MariaDB. A configuração do Node-RED permitiu a recepção e formatação dos dados no formato JSON, que foram então armazenados de maneira organizada e acessível no MariaDB. Essa abordagem facilitou a análise e a visualização das informações, permitindo uma gestão eficaz dos dados coletados.

A visualização dos dados foi realizada com o auxílio do Grafana, que ofereceu uma interface gráfica intuitiva para a análise dos dados. Os gráficos individuais e comparativos configurados no Grafana proporcionaram uma visão detalhada da qualidade do ar, facilitando a identificação de padrões e tendências. A integração com o banco de dados garantiu que as visualizações fossem atualizadas em tempo real, proporcionando uma visão precisa e atualizada dos níveis de poluição.

Além dos aspectos técnicos, o projeto também trouxe contribuições significativas para a área de monitoramento ambiental. A implementação da estação de medição demonstrou a importância da integração de tecnologias avançadas na gestão da qualidade do ar e ofereceu uma solução prática e acessível para a avaliação dos poluentes atmosféricos. A experiência adquirida ao longo do desenvolvimento destacou a necessidade de uma abordagem contínua e adaptativa para lidar com os desafios ambientais.

Em resumo, o trabalho desenvolvido resultou em uma solução eficaz e abrangente para o monitoramento da qualidade do ar. A combinação de sensores precisos, processamento eficiente de dados e visualização detalhada estabeleceu uma base sólida para futuras melhorias e expansões. A estação de medição criada não apenas atingiu os objetivos propostos, mas também forneceu uma ferramenta valiosa para pesquisadores, profissionais e gestores ambientais, contribuindo para uma melhor compreensão e mitigação dos impactos da poluição atmosférica.

## 6. Bibliografia

BRASIL. Ministério do Meio Ambiente. **Qualidade do Ar**. Disponível em : <https://antigo.mma.gov.br/cidades-sustentaveis/qualidade-do-ar.html#content>.

Acesso em: 20 abr.2024

CETESB. **Qualidade do Ar - Poluentes**. São Paulo. [s.d]. Disponível em: <https://cetesb.sp.gov.br/ar/poluentes/>. Acesso em 26 abr. 2024

CETESB. **Qualidade do ar no Estado de São Paulo 2022**. São Paulo. 2023. Disponível em : <https://cetesb.sp.gov.br/ar/wp-content/uploads/sites/28/2023/07/Relatorio-de-Qualidade-do-Ar-no-Estado-de-Sao-Paulo-2022.pdf> . Acesso em 20 abr.2024

COSTA, F. A de A; HENKES, J. A. Uma Breve Análise da Qualidade do Ar da Cidade de São Paulo. **Revista Gestão & Sustentabilidade Ambiental**, Florianópolis, ano 2013, v. v.2, ed. n.2, p. 250-264, abr/set 2013. Disponível em: [https://portaldeperiodicos.animaeducacao.com.br/index.php/gestao\\_ambiental/article/download/1450/1107/2657](https://portaldeperiodicos.animaeducacao.com.br/index.php/gestao_ambiental/article/download/1450/1107/2657). Acesso em: 20 abr. 2024.

GOUVEIA, N; VILLARDI, J.W.R. **Poluentes Atmosféricos Advindos do Espaço Urbano e Regiões Metropolitanas como Determinantes para a Saúde das Populações, e o Papel das Políticas Públicas**. Estudos Avançados da Universidade de São Paulo. São Paulo, jul/dez 2023. Disponível em: <http://www.iea.usp.br/pesquisa/projetos-institucionais/usp-cidades-globais/artigos-digitais/politicas-publicas-poluentes-atmosfericos> . Acesso em: 21/04/2024.

IEMA, **Recomendações Para a Expansão e a Continuidade das Redes de Monitoramento da Qualidade do Ar no Brasil**. São Paulo. jul 2022. Disponível em: [https://energiaeambiente.org.br/wp-content/uploads/2022/07/IEMA\\_policypaper\\_qualidadedoar.pdf](https://energiaeambiente.org.br/wp-content/uploads/2022/07/IEMA_policypaper_qualidadedoar.pdf). Acesso em 26 abr. 2024.

LIMA, B. A. A. in PINOTTI. F. **Consequências da degradação ambiental são piores na periferia**. Agência Universitária de Notícias. São Paulo, nov 2019. Disponível em: <https://aun.webhostusp.sti.usp.br/index.php/2019/11/29/consequencias-da-degradacao-ambiental-sao-piores-nas-periferias/> . Acesso em 21 abr. 2024

PEIXOTO, R. **Apenas 13 estados do país têm estações automáticas de qualidade do ar, mostra estudo**. [S.l]. 27. Fev 2024. G1. Disponível em: <https://g1.globo.com/meio-ambiente/noticia/2024/02/27/apenas-13-estados-do-pais-t-em-esta-coes-automaticas-de-qualidade-do-ar-mostra-estudo.ghtml>. Acesso em: 21 abr. 2024.

ONU. **Novos dados da OMS revelam que bilhões de pessoas ainda respiram ar insalubre** | As Nações Unidas no Brasil. (n.d.). 23 mar 2024. Disponível em:<https://brasil.un.org/pt-br/176864-novos-dados-da-oms-revelam-que-bilh%C3%B5es-de-pessoas-ainda-respiram-ar-insalubre> Acesso em: 21 abr. 2024

SALDIVA, P. **Pobres são os mais atingidos pela poluição urbana, diz médico da USP**. Entrevistador: César Samson. Instituto Humanitas Unisinos. São Paulo. ago 2013. Disponível em: <https://www.ihu.unisinos.br/noticias/512065-pobres-sao-os-mais-atingidos-pela-poluicao-urbana-diz-medico-da-usp>. Acesso em 21 abr. 2024

VORMITTAG, E. da M. P. A. de A., Cirqueira, S. S. R., Wicher Neto, H., & Saldiva, P. H. N. **Análise do monitoramento da qualidade do ar no Brasil**. São Paulo, jul 2023. Estudos Avançados. Disponível em: <https://doi.org/10.1590/s0103-4014.2021.35102.002>. Acesso : 20 abr. 2024

## 7. Apêndice

### APÊNDICE A - Testes dos sensores utilizados.

#### A.1 Introdução

Este apêndice documenta os testes realizados nos sensores utilizados no projeto de monitoramento da qualidade do ar. Os sensores envolvidos são o MQ-7 (CO<sub>2</sub>), MQ-135 (NO<sub>2</sub>), MQ-131 (O<sub>3</sub>), MQ-5 (Fumaça), DHT22 (Temperatura e Umidade) e DSM501A (PM<sub>2.5</sub>). O objetivo dos testes foi validar a precisão, a confiabilidade e a funcionalidade dos sensores em condições reais e simuladas.

#### A.2 Sensores e Configurações

##### A.2.1 MQ-7 (CO<sub>2</sub>)

- Descrição: Sensor de monóxido de carbono (CO) e dióxido de carbono (CO<sub>2</sub>).
- Pino de Conexão: A0 no Arduino.
- Fator de Calibração: 0.5 para conversão de leitura analógica para ppm.

### **A.2.2 MQ-135 (NO2)**

- Descrição: Sensor para detectar dióxido de nitrogênio (NO2) e outros gases poluentes.
- Pino de Conexão: A1 no Arduino.
- Fator de Calibração: 0.5 para conversão de leitura analógica para ppm.

### **A.2.3 MQ-131 (O3)**

- Descrição: Sensor para detecção de Ozônio (O3).
- Pino de Conexão: A2 no Arduino.
- Fator de Calibração: 0.5 para conversão de leitura analógica para ppm.

### **A.2.4 MQ-5 (Fumaça)**

- Descrição: Sensor para detectar fumaça e gás metano.
- Pino de Conexão: A3 no Arduino.
- Fator de Calibração: 0.5 para conversão de leitura analógica para ppm.

### **A.2.5 DHT22 (Temperatura e Umidade)**

- Descrição: Sensor de temperatura e umidade.
- Pino de Conexão: D2 no Arduino.
- Tipo: DHT22.

### **A.2.6 DSM501A (PM2.5)**

- Descrição: Sensor de partículas finas no ar (PM2.5).
- Pino de Conexão: D4 no Arduino.
- Tipo de Saída: Pulso.

## **A.3 Metodologia de Teste**

### **A.3.1 Configuração dos Sensores**

Todos os sensores foram conectados ao Arduino conforme descrito nas seções anteriores. O Arduino foi programado para ler os dados dos sensores e transmitir os resultados para um monitor serial.

### **A.3.2 Procedimentos de Teste**

- Testes de Precisão: Comparação das leituras dos sensores com valores conhecidos para verificar a precisão dos dados.
- Testes de Repetibilidade: Coleta de múltiplas leituras em condições estáveis para verificar a consistência das medições.
- Testes de Calibração: Ajuste dos fatores de calibração dos sensores para garantir que os dados estejam dentro dos intervalos esperados.
- Testes em Condições Variadas: Exposição dos sensores a diferentes condições ambientais (como variações de temperatura e umidade) para avaliar o desempenho.

## **A.4 Resultados dos Testes**

### **A.4.1 MQ-7 (CO<sub>2</sub>)**

- Precisão: As leituras foram consistentes com os valores esperados quando comparadas a uma fonte de CO<sub>2</sub> conhecida.
- Repetibilidade: As medições repetidas em condições estáveis mostraram variações mínimas.

### **A.4.2 MQ-135 (NO<sub>2</sub>)**

- Precisão: O sensor apresentou uma boa correspondência com os valores conhecidos para NO<sub>2</sub>.
- Repetibilidade: As leituras foram consistentes, com pequenas variações que podem ser atribuídas a flutuações naturais no ambiente.

### **A.4.3 MQ-131 (O<sub>3</sub>)**

- Precisão: Leitura precisa dos níveis de O<sub>3</sub> com boas correlações com fontes conhecidas.
- Repetibilidade: A repetição das medições mostrou estabilidade e confiabilidade.

### **A.4.4 MQ-5 (Fumaça)**

- Precisão: O sensor foi capaz de detectar a presença de fumaça e metano com precisão razoável.
- Repetibilidade: As leituras foram consistentes ao longo do tempo.

### **A.4.5 DHT22 (Temperatura e Umidade)**

- Precisão: As leituras de temperatura e umidade foram precisas e consistentes com os valores esperados.
- Repetibilidade: As medições mostraram pouca variação em condições estáveis.
- 

### **A.4.6 DSM501A (PM<sub>2.5</sub>)**

- Precisão: O sensor de PM<sub>2.5</sub> forneceu leituras confiáveis das partículas no ar, alinhadas com expectativas baseadas em padrões de qualidade do ar.
- Repetibilidade: As medições foram estáveis, com pequenas variações dependendo das condições ambientais.

## **A.5 Considerações Finais**

Os testes realizados confirmaram a eficácia e a confiabilidade dos sensores utilizados no projeto. Cada sensor foi capaz de fornecer dados precisos e repetíveis, o que é crucial para um sistema de monitoramento da qualidade do ar. Ajustes finos

e calibrações adicionais podem ser necessárias para condições específicas, mas o sistema geral demonstrou um desempenho sólido.

A integração dos dados dos sensores e o processamento eficaz foram essenciais para criar uma solução robusta e confiável. O sucesso dos testes reforça a validade do sistema desenvolvido e oferece uma base sólida para futuras melhorias e expansões.

## **APÊNDICE B - Códigos Fonte.**

### **Código-fonte do Arduino.**

```
#include <DHT.h>

// Definição dos pinos dos sensores
const int mq7Pin = A0; // Pino do sensor MQ-7 (CO2)
const int mq135Pin = A1; // Pino do sensor MQ-135 (NO2)
const int mq131Pin = A2; // Pino do sensor MQ-131 (SO2)
const int mq5Pin = A3; // Pino do sensor MQ-5 (Fumaça)

// Definição do pino do sensor DSM501A
const int dsm501aPin = 4; // Pino do sensor DSM501A (PM2.5)

// Definição do pino e tipo do sensor DHT
#define DHTPIN 2 // Pino de dados do sensor DHT
#define DHTTYPE DHT22 // Definir o tipo de sensor (DHT11 ou DHT22)
DHT dht(DHTPIN, DHTTYPE);

// Variáveis para armazenar os valores lidos dos sensores
int mq7Value = 0;
int mq135Value = 0;
int mq131Value = 0;
int mq5Value = 0;
float temperature = 0.0;
float humidity = 0.0;
unsigned long duration;
float ratio = 0;
float concentration = 0;

// Intervalo de leitura (1 minuto em milissegundos)
const unsigned long interval = 60000; // 60.000 ms

// Variáveis para controle de tempo
unsigned long previousMillis = 0;
```

```

// Funções para conversão de leitura dos sensores MQ para ppm
float mqToPPM(int sensorValue, float calibrationFactor) {
    // O valor de calibração deve ser ajustado com base nas especificações do sensor
    e condições ambientais
    float voltage = sensorValue * (5.0 / 1023.0); // Convertendo o valor do sensor para
    tensão (5V é a tensão de referência do Arduino)
    return (voltage / calibrationFactor); // Converta a tensão para ppm usando o fator de
    calibração
}

void setup() {
    // Inicializa a comunicação serial
    Serial.begin(9600);
    // Inicializa o sensor DHT
    dht.begin();
    // Inicializa o pino do DSM501A
    pinMode(dsm501aPin, INPUT);
}

void loop() {
    unsigned long currentMillis = millis();

    if (currentMillis - previousMillis >= interval) {
        previousMillis = currentMillis;

        // Lê os valores dos sensores
        mq7Value = analogRead(mq7Pin);
        mq135Value = analogRead(mq135Pin);
        mq131Value = analogRead(mq131Pin);
        mq5Value = analogRead(mq5Pin);

        // Lê a temperatura e umidade do sensor DHT
        temperature = dht.readTemperature();
        humidity = dht.readHumidity();

        // Verifica se a leitura do DHT foi bem-sucedida
        if (isnan(temperature) || isnan(humidity)) {
            Serial.println("Falha na leitura do DHT");
        } else {
            // Converte os valores dos sensores para ppm
            float mq7PPM = mqToPPM(mq7Value, 0.5); // Ajuste o fator de calibração
            conforme necessário
        }
    }
}

```

```

float mq135PPM = mqToPPM(mq135Value, 0.5); // Ajuste o fator de calibração
conforme necessário
float mq131PPM = mqToPPM(mq131Value, 0.5); // Ajuste o fator de calibração
conforme necessário
float mq5PPM = mqToPPM(mq5Value, 0.5); // Ajuste o fator de calibração
conforme necessário

// Lê o valor do DSM501A
duration = pulseIn(dsm501aPin, LOW);
ratio = duration / (interval * 10.0); // Calcula a razão de ocupação
concentration = 1.1 * pow(ratio, 3) - 3.8 * pow(ratio, 2) + 520 * ratio + 0.62; //
Concentração em µg/m³

// Envia os valores dos sensores para o monitor serial
Serial.print("MQ-7 (CO2): ");
Serial.print(mq7PPM);
Serial.print(" ppm, MQ-135 (NO2): ");
Serial.print(mq135PPM);
Serial.print(" ppm, MQ-131 (SO2): ");
Serial.print(mq131PPM);
Serial.print(" ppm, MQ-5 (Fumaça): ");
Serial.print(mq5PPM);
Serial.print(" ppm, Temperature: ");
Serial.print(temperature);
Serial.print(" °C, Humidity: ");
Serial.print(humidity);
Serial.print("%, PM2.5 (DSM501A): ");
Serial.print(concentration);
Serial.println(" µg/m³");
}
}
}

```

### **Código-fonte Json do RED-NODE.**

```

[
  {
    "id": "6fb34501bf3b8363",
    "type": "function",
    "z": "7f750329990d4411",
    "name": "function 1",
    "func": "let payload = msg.payload.trim(); // Remove espaços e caracteres
especiais no início e no final da string\n\n// Regex para capturar os valores\nlet

```

```

mq7Match      =      payload.match(/MQ-7\s*\(\(CO2\)\):\s*(\[\d.\]+\)\s*ppm/);\nlet
mq135Match    =      payload.match(/MQ-135\s*\(\(NO2\)\):\s*(\[\d.\]+\)\s*ppm/);\nlet
mq131Match    =      payload.match(/MQ-131\s*\(\(SO2\)\):\s*(\[\d.\]+\)\s*ppm/);\nlet
mq5Match      =      payload.match(/MQ-5\s*\(\(Fumaça\)\):\s*(\[\d.\]+\)\s*ppm/);\nlet
temperatureMatch    =      payload.match(/Temperature:\s*(\[\d.\]+\)\s*°C/);\nlet
humidityMatch = payload.match(/Humidity:\s*(\[\d.\]+\)%/); // Atualizado para capturar
a      porcentagem      de      umidade\nlet      pm25Match      =
payload.match(/PM2\5\s*\(\(DSM501A\)\):\s*(\[\d.\]+\)\s*µg\Wm³/);\n\n// Extrair os
valores ou definir como 0 se não for encontrado\nlet mq7 = mq7Match ?
parseFloat(mq7Match[1]) : 0;\nlet mq135 = mq135Match ?
parseFloat(mq135Match[1]) : 0;\nlet mq131 = mq131Match ?
parseFloat(mq131Match[1]) : 0;\nlet mq5 = mq5Match ? parseFloat(mq5Match[1]) :
0;\nlet temperature = temperatureMatch ? parseFloat(temperatureMatch[1]) : 0;\nlet
humidity = humidityMatch ? parseFloat(humidityMatch[1]) : 0;\nlet pm25 =
pm25Match ? parseFloat(pm25Match[1]) : 0;\n\n// Converter o timestamp para o
formato 'YYYY-MM-DD HH:MM:SS'\nlet timestamp = new
Date().toISOString().replace('T', ' ').split('.')[0]; // Remove frações de segundos e
'T'\n\n// Configura o tópico e o payload para inserção no banco de dados\nmsg.topic
= \"INSERT INTO sensor_data (mq7, mq135, mq131, mq5, temperature, humidity,
pm25, timestamp) VALUES (?, ?, ?, ?, ?, ?, ?, ?)\";\nmsg.payload = [mq7, mq135,
mq131, mq5, temperature, humidity, pm25, timestamp];\n\nreturn msg;\n",
  "outputs": 1,
  "timeout": 0,
  "noerr": 0,
  "initialize": "",
  "finalize": "",
  "libs": [],
  "x": 760,
  "y": 260,
  "wires": [
    [
      "4c1a903af44a7059",
      "556edf27294ff93a"
    ]
  ]
}
]

```

### Código-fonte Function do RED-NODE.

```

let payload = msg.payload.trim(); // Remove espaços e caracteres especiais no início
e no final da string

```

```

// Regex para capturar os valores

```

```

let mq7Match = payload.match(/MQ-7\s*\(\CO2\):\s*(\d+)\s*ppm/);
let mq135Match = payload.match(/MQ-135\s*\(\NO2\):\s*(\d+)\s*ppm/);
let mq131Match = payload.match(/MQ-131\s*\(\SO2\):\s*(\d+)\s*ppm/);
let mq5Match = payload.match(/MQ-5\s*\(\Fumaça\):\s*(\d+)\s*ppm/);
let temperatureMatch = payload.match(/Temperature:\s*(\d+)\s°C/);
let humidityMatch = payload.match(/Humidity:\s*(\d+)\%/); // Atualizado para
capturar a porcentagem de umidade
let pm25Match = payload.match(/PM2.5\s*\(\DSM501A\):\s*(\d+)\s*µg\m³/);

// Extrair os valores ou definir como 0 se não for encontrado
let mq7 = mq7Match ? parseFloat(mq7Match[1]) : 0;
let mq135 = mq135Match ? parseFloat(mq135Match[1]) : 0;
let mq131 = mq131Match ? parseFloat(mq131Match[1]) : 0;
let mq5 = mq5Match ? parseFloat(mq5Match[1]) : 0;
let temperature = temperatureMatch ? parseFloat(temperatureMatch[1]) : 0;
let humidity = humidityMatch ? parseFloat(humidityMatch[1]) : 0;
let pm25 = pm25Match ? parseFloat(pm25Match[1]) : 0;

// Converter o timestamp para o formato 'YYYY-MM-DD HH:MM:SS'
let timestamp = new Date().toISOString().replace('T', ' ').split('.')[0]; // Remove frações
de segundos e 'T'

// Configura o tópico e o payload para inserção no banco de dados
msg.topic = "INSERT INTO sensor_data (mq7, mq135, mq131, mq5, temperature,
humidity, pm25, timestamp) VALUES (?, ?, ?, ?, ?, ?, ?, ?)";
msg.payload = [mq7, mq135, mq131, mq5, temperature, humidity, pm25, timestamp];

return msg;

```

### **Código-fonte Banco de dados do Mariadb.**

```

-- Criação do banco de dados
CREATE DATABASE IF NOT EXISTS sensor_database;

-- Seleção do banco de dados
USE sensor_database;

-- Criação da tabela sensor_data
CREATE TABLE IF NOT EXISTS sensor_data (
  id INT AUTO_INCREMENT PRIMARY KEY,
  mq7 FLOAT,
  mq135 FLOAT,
  mq131 FLOAT,
  mq5 FLOAT,

```

```

    temperature FLOAT,
    humidity FLOAT,
    pm25 FLOAT,
    timestamp DATETIME
);

-- Criação da tabela para armazenar informações de sensores
CREATE TABLE IF NOT EXISTS sensor_info (
    id INT AUTO_INCREMENT PRIMARY KEY,
    sensor_name VARCHAR(255),
    sensor_type VARCHAR(255),
    location VARCHAR(255),
    calibration_info TEXT
);

-- Criação da tabela para armazenar logs de erros
CREATE TABLE IF NOT EXISTS error_logs (
    id INT AUTO_INCREMENT PRIMARY KEY,
    error_message TEXT,
    timestamp DATETIME DEFAULT CURRENT_TIMESTAMP
);

-- Inserção de exemplo na tabela sensor_data
INSERT INTO sensor_data (mq7, mq135, mq131, mq5, temperature, humidity,
pm25, timestamp)
VALUES (1.11, 1.32, 0.85, 2.44, 24.50, 45.80, 0.62, NOW());

```