



Monitoramento e Controle da Qualidade do Ar em Ambientes Internos com IoT Alinhado a ODS 3 da ONU

Márcia Jandira Baptista de Almeida , Prof.: André Luís de Oliveira

Faculdade de Computação e Informática
Universidade Presbiteriana Mackenzie (UPM) – São Paulo, SP – Brazil

{RA:10290248, marciahandira.almeida@mackenzista.com.br}

Abstract

This paper presents the development of an Internet of Things (IoT) system for monitoring and controlling indoor air quality, directly aligned with Sustainable Development Goal (SDG) 3: Good Health and Well-being. The proposed system utilizes an ESP32 microcontroller, an air quality sensor (MQ-135 or BME680), and a relay module connected to an air extractor or purifier. Data collected from the sensor, including pollutant concentrations, temperature, and humidity, are transmitted to a cloud-based MQTT broker. The system is designed to alert users to critical air quality levels and automatically activate the actuator to improve environmental conditions. This project demonstrates a practical application of IoT to address health challenges related to indoor air pollution, contributing to the promotion of healthier living environments.

Resumo

Este artigo apresenta o desenvolvimento de um sistema de Internet das Coisas (IoT) para monitoramento e controle da qualidade do ar em ambientes internos, diretamente alinhado ao Objetivo de Desenvolvimento Sustentável (ODS) 3: Saúde e Bem-estar. O sistema proposto utiliza um microcontrolador ESP32, um sensor de qualidade do ar (MQ-135 ou BME680) e um módulo relé de 1 canal 5v conectado a um exaustor ou purificador de ar. Os dados coletados pelo sensor, incluindo concentrações de poluentes, temperatura e umidade, são transmitidos para um broker MQTT baseado em nuvem. O sistema é projetado para alertar os usuários sobre níveis críticos de qualidade do ar e ativar automaticamente o atuador para melhorar as condições ambientais. Este projeto demonstra uma aplicação prática da IoT para abordar desafios de saúde relacionados à poluição do ar em ambientes internos, contribuindo para a promoção de ambientes de vida mais saudáveis.

1. Introdução

A qualidade do ar em ambientes internos tem se tornado uma preocupação crescente na saúde pública global. Com a maior parte da população mundial passando uma parcela significativa de seu tempo em espaços fechados, como residências, escritórios e escolas, a exposição a poluentes atmosféricos internos pode ter impactos substanciais na saúde humana. A Organização Mundial da Saúde (OMS) estima que a poluição do ar interior seja responsável por milhões de mortes prematuras anualmente, principalmente devido a doenças respiratórias e cardiovasculares [1]. Essa problemática se alinha diretamente com o Objetivo de Desenvolvimento Sustentável (ODS) 3 da Agenda 2030 da Organização das Nações Unidas (ONU), que visa “Assegurar uma vida saudável e promover o bem-estar para todas e todos, em todas as idades”. Especificamente, a Meta 3.9 do ODS 3 busca “reduzir substancialmente o número de mortes e doenças por produtos químicos perigosos, contaminação e poluição do ar e água do solo” [2]. Nesse contexto, a capacidade de monitorar e controlar ativamente a qualidade do ar em ambientes internos não é apenas uma conveniência, mas uma necessidade premente para a promoção da saúde e a prevenção de doenças.

1.1 Evolução Histórica

Historicamente, o monitoramento da qualidade do ar era uma tarefa complexa e dispendiosa, restrita a equipamentos de laboratório e estações de monitoramento governamentais. Com o avanço da tecnologia, especialmente no campo da eletrônica e da computação embarcada, surgiram sensores mais acessíveis e compactos, tornando viável o monitoramento em tempo real em escala individual e doméstica. A evolução da Internet das Coisas (IoT) revolucionou a forma como interagimos com o ambiente físico, permitindo que dispositivos cotidianos se conectem à internet, coletam e troquem dados. Essa interconexão de objetos inteligentes abriu portas para soluções inovadoras em diversas áreas, incluindo a saúde. Desde os primeiros protótipos de sistemas de automação residencial até os atuais dispositivos vestíveis (wearables) para monitoramento de sinais vitais, a IoT tem demonstrado um potencial transformador na melhoria da qualidade de vida e na gestão da saúde [3].

1.2 Panorama de Estudos Relacionados

Diversos trabalhos correlatos têm explorado a aplicação da IoT no monitoramento ambiental e na saúde. Por exemplo, sistemas baseados em microcontroladores como Arduino e Raspberry Pi têm sido utilizados para coletar dados de sensores de temperatura, umidade e gases em ambientes internos e externos, transmitindo essas informações para plataformas de nuvem para análise e visualização [4]. A comunicação via protocolo MQTT (Message Queuing Telemetry Transport) emergiu como um padrão de fato para a IoT devido à sua leveza, eficiência e capacidade de operar em redes com largura de banda limitada, sendo amplamente empregado em projetos de monitoramento remoto e automação [5]. Projetos que integram sensores de qualidade do ar com plataformas IoT para alertar sobre níveis de poluentes e acionar sistemas de ventilação já foram propostos, demonstrando a viabilidade técnica e a relevância prática de tais soluções [6]. No entanto, a integração de todos esses elementos em um sistema coeso que não apenas monitore, mas também atue proativamente na melhoria da qualidade do ar, com foco explícito nas metas do ODS 3, ainda representa um campo fértil para pesquisa e desenvolvimento. Este trabalho busca preencher essa lacuna, propondo um sistema de IoT que visa diretamente contribuir para a meta de saúde e bem-estar, oferecendo uma solução prática e acessível para o controle da poluição do ar em ambientes internos.

2. Objetivos e Problemas a Solucionar

O principal objetivo deste projeto é desenvolver um sistema de Internet das Coisas (IoT) para o monitoramento e controle autônomo da qualidade do ar em ambientes internos. Este sistema visa mitigar os riscos à saúde associados à poluição do ar interior, alinhando-se diretamente ao Objetivo de Desenvolvimento Sustentável (ODS) 3 da ONU, que busca "Assegurar uma vida saudável e promover o bem-estar para todas e todos, em todas as idades", com foco na Meta 3.9 de reduzir mortes e doenças por produtos químicos perigosos e poluição do ar [2].

Os problemas específicos que este projeto pretende solucionar incluem:

- Detecção de Poluentes: Identificar e quantificar a presença de gases poluentes, material particulado, temperatura e umidade em tempo real em ambientes internos.
- Alerta e Notificação: Fornecer alertas imediatos aos usuários quando os níveis de poluentes excederem limites seguros, permitindo a tomada de ações preventivas ou corretivas.
- Controle Autônomo: Ativar automaticamente atuadores, como exaustores ou purificadores de ar, para melhorar a qualidade do ambiente quando necessário, sem intervenção humana constante.
- Acessibilidade e Custo-benefício: Desenvolver uma solução de baixo custo e fácil implementação, tornando o monitoramento e controle da qualidade do ar acessível a um público mais amplo.
- Coleta e Análise de Dados: Registrar dados históricos da qualidade do ar para análise de tendências, otimização do sistema e suporte a decisões de longo prazo para a melhoria da saúde ambiental.

3. Materiais e Métodos

Esta seção detalha os componentes de hardware que serão empregados para o desenvolvimento do sistema de monitoramento e controle da qualidade do ar em ambientes internos. A arquitetura do sistema é baseada em uma plataforma de prototipagem eletrônica, sensores para coleta de dados ambientais e atuadores para correção das condições.

3.1. Plataforma de Prototipagem Eletrônica

Para a implementação do sistema, será utilizada a plataforma ESP32, devido à sua capacidade de processamento, conectividade Wi-Fi e Bluetooth integradas, e baixo consumo de energia. O ESP32 é um microcontrolador de baixo custo e alta performance, ideal para aplicações de IoT que exigem comunicação com a internet. Sua flexibilidade e o vasto suporte da comunidade facilitam o desenvolvimento e a integração com diversos sensores e atuadores [8], [9]. A escolha do ESP32 é estratégica por garantir a conectividade com a internet, um requisito essencial para a implementação do protocolo MQTT. O módulo ESP32 utilizado é ilustrado na Figura 2.

ESP32

O ESP32 é um System-on-Chip (SoC) desenvolvido pela Espressif Systems, que integra um processador dual-core Tensilica Xtensa LX6, conectividade Wi-Fi 802.11 b/g/n e Bluetooth v4.2 BR/EDR e BLE. Possui uma ampla gama de periféricos, incluindo GPIOs, ADC, DAC, I2C, SPI, UART, PWM, entre outros, o que o torna extremamente versátil para projetos de IoT. A capacidade de operar em modo de baixo consumo de energia é crucial para aplicações que dependem de bateria ou que visam a eficiência energética. A programação do ESP32 pode ser realizada utilizando a IDE do Arduino, MicroPython ou o framework ESP-IDF, oferecendo flexibilidade aos desenvolvedores[9], [10].



3.2. Sensores

Para o monitoramento da qualidade do ar, será empregado o sensor MQ-135, capaz de detectar uma variedade de gases poluentes.

Sensor de Qualidade do Ar MQ-135

O sensor MQ-135 é um sensor de gás semicondutor amplamente utilizado para detectar uma vasta gama de gases, incluindo amônia (NH_3), sulfeto (S), benzeno (C_6H_6), CO_2 e outros gases tóxicos e fumaça. Ele opera com base na alteração da resistência do material sensível (SnO_2) em contato com os gases no ar. A resistência do sensor diminui à medida que a concentração do gás alvo aumenta. O MQ-135 possui alta sensibilidade e tempo de resposta rápido, sendo adequado para aplicações de monitoramento da qualidade do ar em ambientes internos. O módulo do sensor geralmente inclui um circuito de acionamento simples e pode fornecer saídas analógicas e digitais [11].



Fonte: RobotShop; [Winsen Sensor - Datasheet MQ135](#)

3.3. Atuadores

Como atuador, será utilizado um módulo relé para controlar dispositivos externos, como exaustores ou purificadores de ar.

Módulo Relé de 1 Canal 5V

O módulo relé de 1 canal 5V é um dispositivo eletromecânico que permite controlar cargas de alta potência (como exaustores ou purificadores de ar) a partir de sinais de baixa potência, como os fornecidos pelo microcontrolador ESP32. Ele atua como um interruptor controlado eletronicamente, isolando o circuito de controle (ESP32) do circuito de carga (aparelho elétrico). O módulo geralmente inclui um optoacoplador para fornecer isolamento galvânica, protegendo o microcontrolador de picos de tensão. Possui terminais para conexão normalmente aberta (NO), normalmente fechada (NC) e comum (COM), permitindo flexibilidade na configuração do acionamento. A bobina do relé é ativada com 5V, e os contatos podem suportar correntes de até 10A em 250VAC ou 30VDC.



Fonte: Components101 - 5V Single-Channel Relay Module

3.4. Modelo de Montagem (Fritzing)

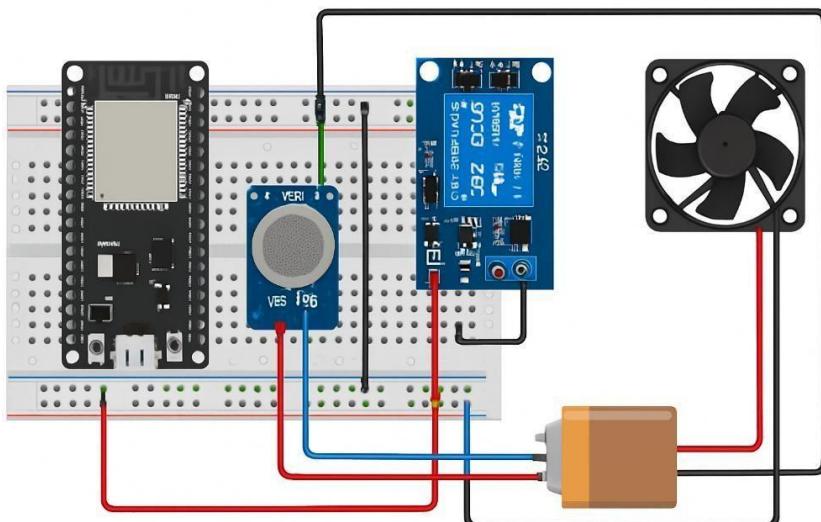
Para a visualização e prototipagem do circuito, utilizou-se o software Fritzing, uma ferramenta de automação de design eletrônico de código aberto que permite a criação de diagramas de protoboard, esquemáticos e PCB. O modelo de montagem proposto, ilustrado na Figura 1, detalha a interconexão do microcontrolador ESP32, do sensor MQ-135 e do módulo relé.

Conexões Detalhadas:

Componente	Pino de Saída	Pino de Entrada no ESP32	Descrição da Conexão
Sensor MQ-135	VCC	3.3V (ou 5V se o módulo tiver regulador)	Alimentação do sensor.
Sensor MQ-135	GND	GND	Terra.
Sensor MQ-135	A0 (Analógico)	GPIO 34 (VP)	Leitura da concentração de gases.
Módulo Relé	VCC	5V (ou 3.3V dependendo do módulo)	Alimentação da bobina do relé.
Módulo Relé	GND	GND	Terra.
Módulo Relé	IN (Sinal)	GPIO 23	Sinal de controle para acionar o atuador (exaustor/purificador).

Figura: Diagrama de Montagem do Protótipo (Simulação Fritzing) O diagrama de montagem do protótipo, que gerado no Fritzing, é detalhado visualmente na Figura Modelo de Montagem.

Modelo de Montagem (Fritzing)



4. Métodos

Esta seção descreve as ferramentas e metodologias que serão empregadas para a concretização do trabalho final, abrangendo o protocolo de comunicação e o ambiente de desenvolvimento.

4.1. Protocolo de Comunicação MQTT

Para a comunicação entre o sistema embarcado (ESP32) e a nuvem, será utilizado o protocolo MQTT (Message Queuing Telemetry Transport). O MQTT é um protocolo de mensagens leve, projetado para dispositivos com recursos limitados e redes com baixa largura de banda e alta latência, tornando-o ideal para aplicações de IoT. Ele opera com um modelo de publicação/assinatura, onde os dispositivos (clientes) publicam mensagens em tópicos específicos e outros dispositivos ou serviços (clientes) assinam esses tópicos para receber as mensagens. Um broker MQTT central gerencia a distribuição das mensagens, garantindo que elas cheguem aos assinantes corretos. Este protocolo é eficiente em termos de consumo de energia e dados, características essenciais para o monitoramento contínuo da qualidade do ar [7], [13].

4.2. Ferramentas e Metodologias de Desenvolvimento

O desenvolvimento do software embarcado será realizado utilizando a IDE do Arduino, que oferece um ambiente de programação simplificado e uma vasta biblioteca de funções para o ESP32. A linguagem de programação será C/C++. Para a gestão dos dados na nuvem, será configurado um broker MQTT, onde o ESP32 publicará as leituras dos sensores. Uma plataforma de visualização de dados (por exemplo, Thingspeak, Adafruit IO ou um dashboard customizado) será utilizada para exibir os dados em tempo real e armazenar o histórico para análises futuras. A metodologia de desenvolvimento seguirá um ciclo iterativo, com fases de prototipagem, testes e refinamento, garantindo a funcionalidade e a robustez do sistema [10], [14].

Para a gestão e visualização dos dados, foi utilizada a plataforma Node-RED, que atuou como broker MQTT e ferramenta para criação de um dashboard interativo. A metodologia de desenvolvimento seguiu um ciclo iterativo, com fases de prototipagem, testes e refinamento.

4.3. Integração e Arquitetura do Sistema

A integração entre o hardware e a plataforma de controle é realizada utilizando o Node-RED como intermediário. O microcontrolador ESP32, simulado na plataforma Wokwi, conecta-se à rede Wi-Fi e publica as leituras do sensor em um tópico MQTT (ex: `wokwi/ppm`). O Node-RED, por sua vez, assina este tópico para receber os dados em tempo real. Para o controle do atuador, o Node-RED publica mensagens em um segundo tópico (ex: `wokwi/exaustor`). O ESP32 assina este tópico e, ao receber uma mensagem, aciona ou desativa o pino correspondente ao atuador. Essa arquitetura desacoplada garante flexibilidade e escalabilidade ao sistema.

5. Funcionamento

O sistema de monitoramento e controle da qualidade do ar opera de forma autônoma e proativa, seguindo um ciclo contínuo de sensoriamento, processamento, comunicação e atuação. O projeto atende aos critérios básicos de uso de ao menos um atuador (Módulo Relé) e um sensor (MQ-135), além da comunicação com a internet via protocolo MQTT.

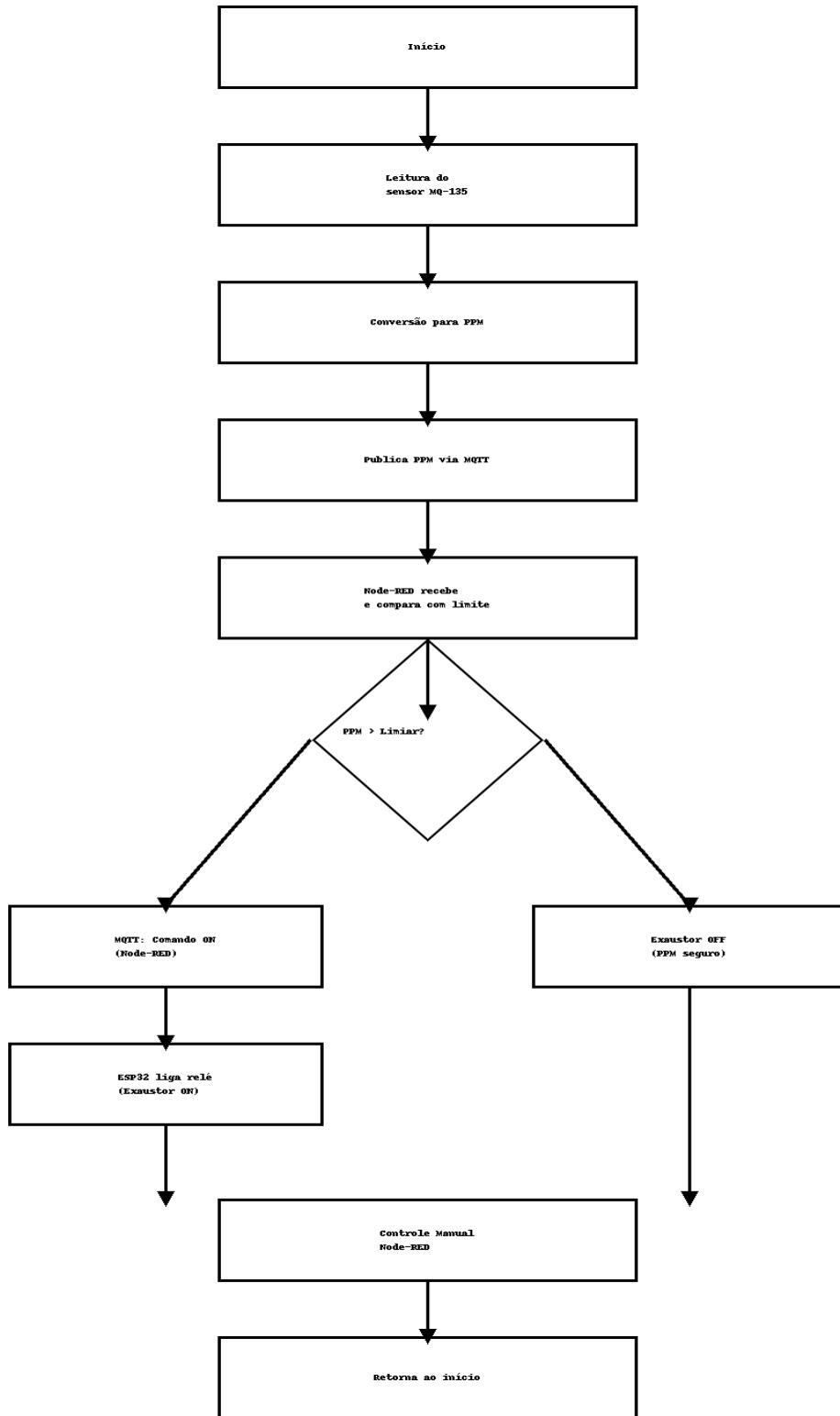


Figura 1: Fluxograma de Funcionamento do Sistema

O ciclo de operação é o seguinte:

1. Sensoriamento e Processamento: O ESP32 realiza leituras do sensor MQ-135 em intervalos regulares. A leitura analógica é processada e convertida em uma métrica de concentração de poluentes (PPM - Partes Por Milhão).
2. Comunicação e Alerta: O valor de PPM é publicado via MQTT. No Node-RED, o valor é recebido e comparado com um limite pré-estabelecido. Se o limite for excedido, um alerta visual é gerado no dashboard.
3. Ação Autônomo: Se a leitura de poluentes (PPM) exceder o limite de segurança, o Node-RED publica uma mensagem para o tópico do atuador, fazendo com que o ESP32 envie um sinal ALTO (High) para o pino de controle do relé, ativando o exaustor. O atuador permanece ligado até que as leituras retornem a um nível seguro.
4. Controle Manual: O usuário pode, a qualquer momento, utilizar o dashboard no Node-RED para ligar ou desligar o exaustor manualmente, sobrepondo-se à lógica automática.

5.1. Sensoriamento e Processamento de Dados

O Sensor MQ-135 é o principal componente para a coleta de dados. O sensor é conectado a uma porta analógica (GPIO 34) do ESP32. O microcontrolador realiza a leitura da tensão de saída do sensor, que é inversamente proporcional à qualidade do ar (quanto maior a concentração de poluentes, menor a resistência e maior a tensão lida).

- 1 Leitura Contínua: O ESP32 realiza leituras do sensor MQ-135 em intervalos regulares (e.g., a cada 5 segundos).
- 2 Calibração e Conversão: A leitura analógica (valor bruto de 0 a 4095) é processada e convertida em uma métrica de concentração de poluentes (como PPM - Partes Por Milhão), utilizando uma curva de calibração específica para o MQ-135. Esta conversão é crucial para transformar o sinal elétrico em uma informação ambientalmente significativa.

- 3 Análise de Limite: O valor de PPM é comparado com um limite de segurança pré-estabelecido (limiar de acionamento), que define o ponto crítico de qualidade do ar.

5.2. Comunicação e Alerta (Protocolo MQTT)

A comunicação com a nuvem é realizada exclusivamente através do protocolo MQTT [7], garantindo eficiência e confiabilidade. O MQTT é ideal para dispositivos IoT devido ao seu baixo overhead e modelo de comunicação assíncrona.

- 4 Conexão Wi-Fi e Broker: O ESP32 estabelece uma conexão com a rede Wi-Fi local e, em seguida, conecta-se a um Broker MQTT (servidor).
- 5 Publicação de Dados: A cada ciclo de leitura, o ESP32 atua como um Publisher, publicando os dados de qualidade do ar (PPM) em um tópico específico (e.g., casa/qualidade_ar/leitura). A mensagem é um pacote de dados leve contendo o valor medido e um timestamp. A utilização de Quality of Service (QoS) 1 (At Least Once) pode ser configurada para garantir a entrega da mensagem, mesmo em redes instáveis.
- 6 Visualização e Alerta: Um aplicativo móvel ou um dashboard na nuvem atua como um Subscriber, assinando o tópico. Ao receber a mensagem, o sistema de visualização atualiza o status em tempo real e, se necessário, dispara um alerta (notificação push, e-mail) ao usuário.

5.3. Acionamento Autônomo (Atuador)

O controle autônomo do Exaustor de Ar Doméstico é a função de atuação do sistema.

- 7 Condição de Acionamento: Se a leitura de poluentes (PPM) exceder o limite de segurança estabelecido (analisado na seção 4.1), o ESP32 inicia o processo de acionamento.
- 8 Sinal para o Relé: O ESP32 envia um sinal digital ALTO (High) para o pino de controle (GPIO 23) do Módulo Relé de 1 Canal 5V.
- 9 **Ativação do Atuador: O sinal do ESP32 energiza a bobina do relé, que fecha o contato e completa o circuito de alta tensão, ligando o Exaustor de Ar Doméstico.r. O módulo relé atua como um isolador galvânico, protegendo o microcontrolador do circuito de potência.
- 10 Desativação: O atuador permanece ligado até que as leituras do MQ-135 retornem a um nível seguro (abaixo do limite de acionamento). Nesse ponto, o ESP32 envia um sinal digital BAIXO (Low) para o relé, desativando o atuador.

Este ciclo garante que o sistema não apenas monitore, mas também tome ações corretivas imediatas para preservar a qualidade do ar, alinhando-se ao objetivo de controle autônomo do projeto. A lógica de controle implementada no ESP32 permite uma resposta rápida e localizada à degradação da qualidade do ar interno.

6. Resultados

Nesta seção, são apresentados os resultados práticos obtidos com a simulação do protótipo.

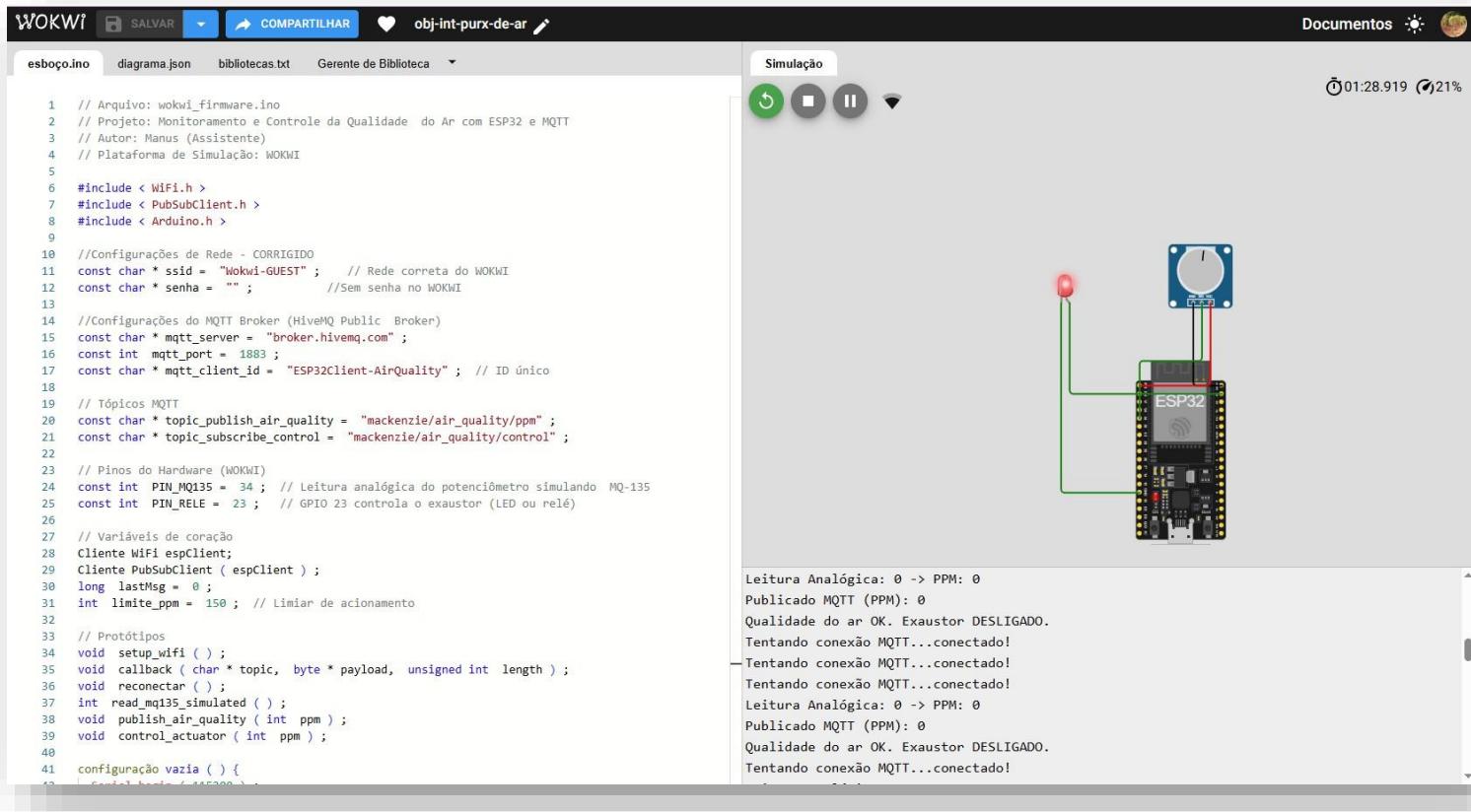


Figura 2: Simulação do protótipo na plataforma Wokwi

Figura 3: Fluxo de controle e visualização no Node-RED

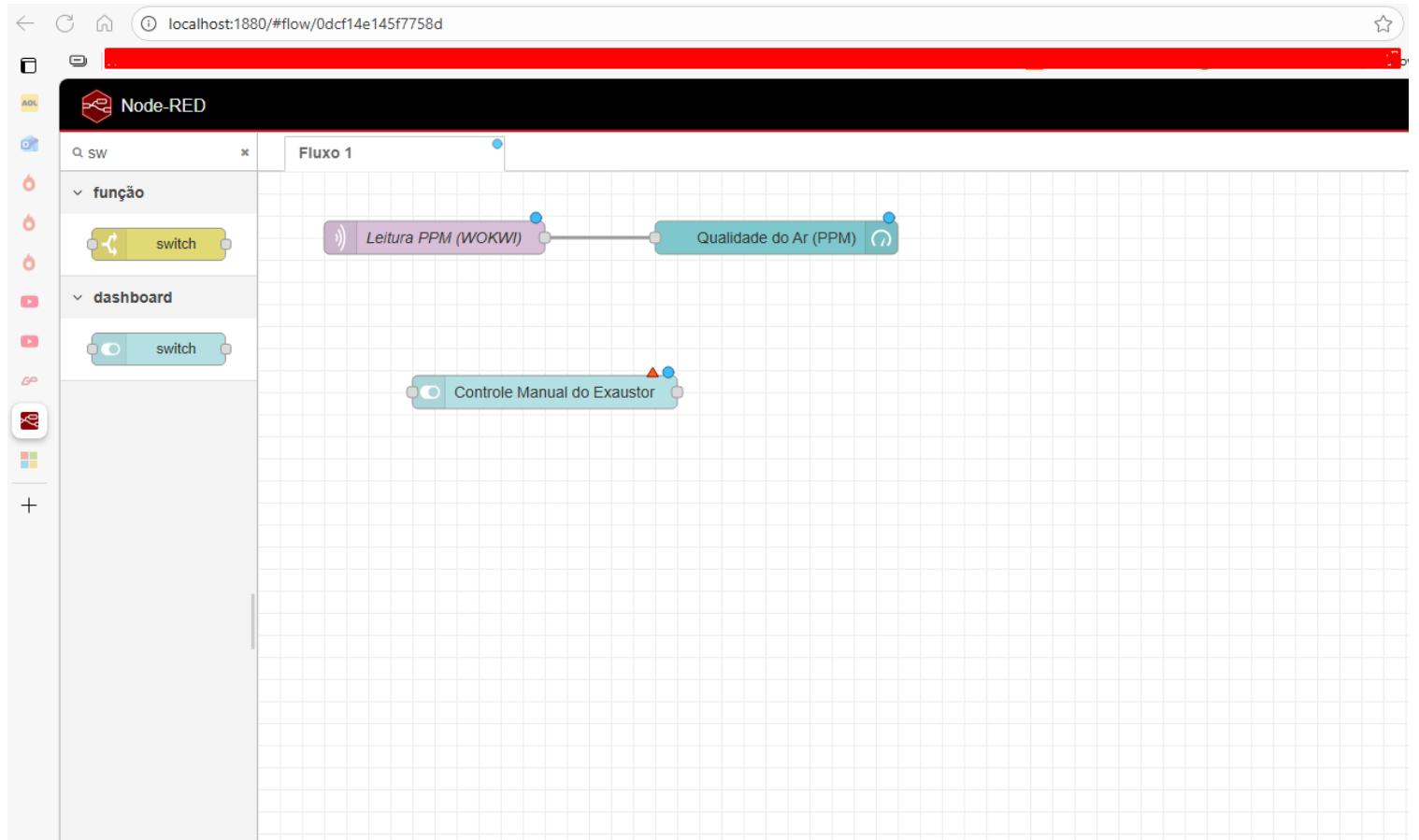


Fig 3.1- Leitura de baixa concentração de PPM e exaustor desligado

Demonstração do sensor simulando baixa poluição (55 PPM), enviando o valor via MQTT para o Node-RED Dashboard. O gauge indica boa qualidade do ar e o exaustor permanece desligado.

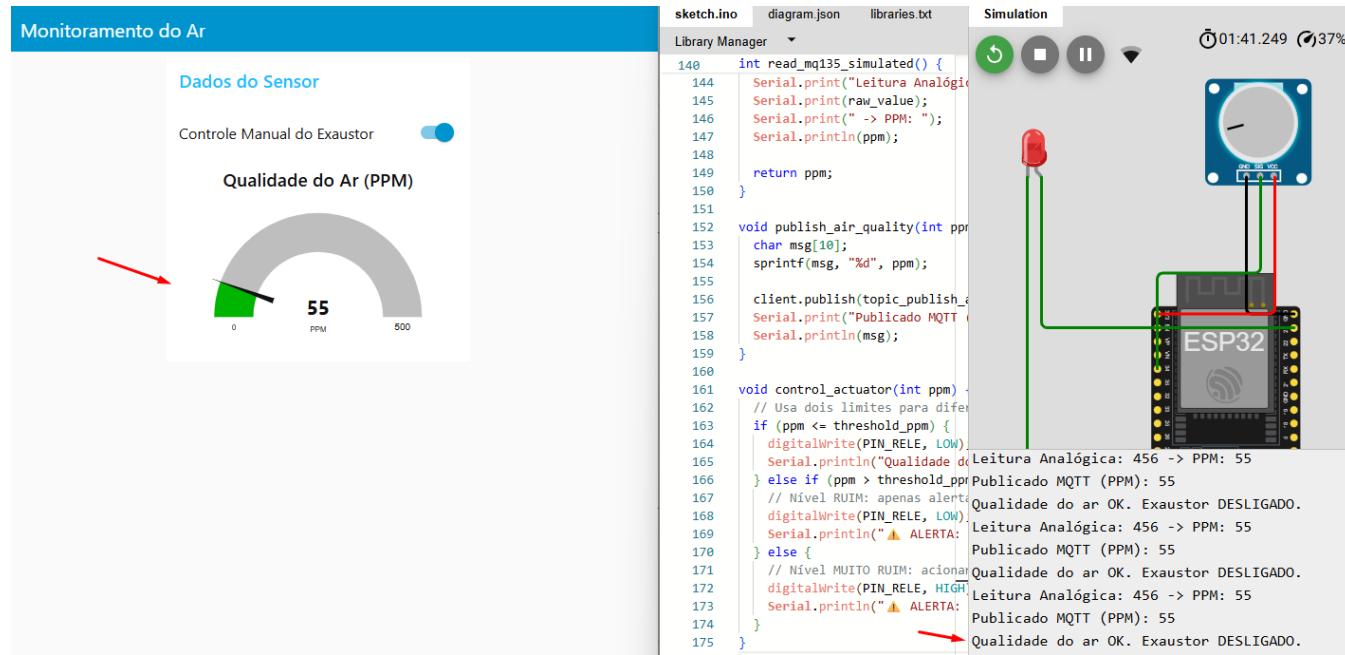


Fig. 3.2 - Recebimento da mensagem MQTT e interpretação no Node-RED

Fluxo no Node-RED recebendo os dados publicados pelo ESP32 via MQTT. O painel de depuração exibe a mensagem “Qualidade do ar OK”, comprovando o recebimento correto do tópico mackenzie/air_quality/ppm.

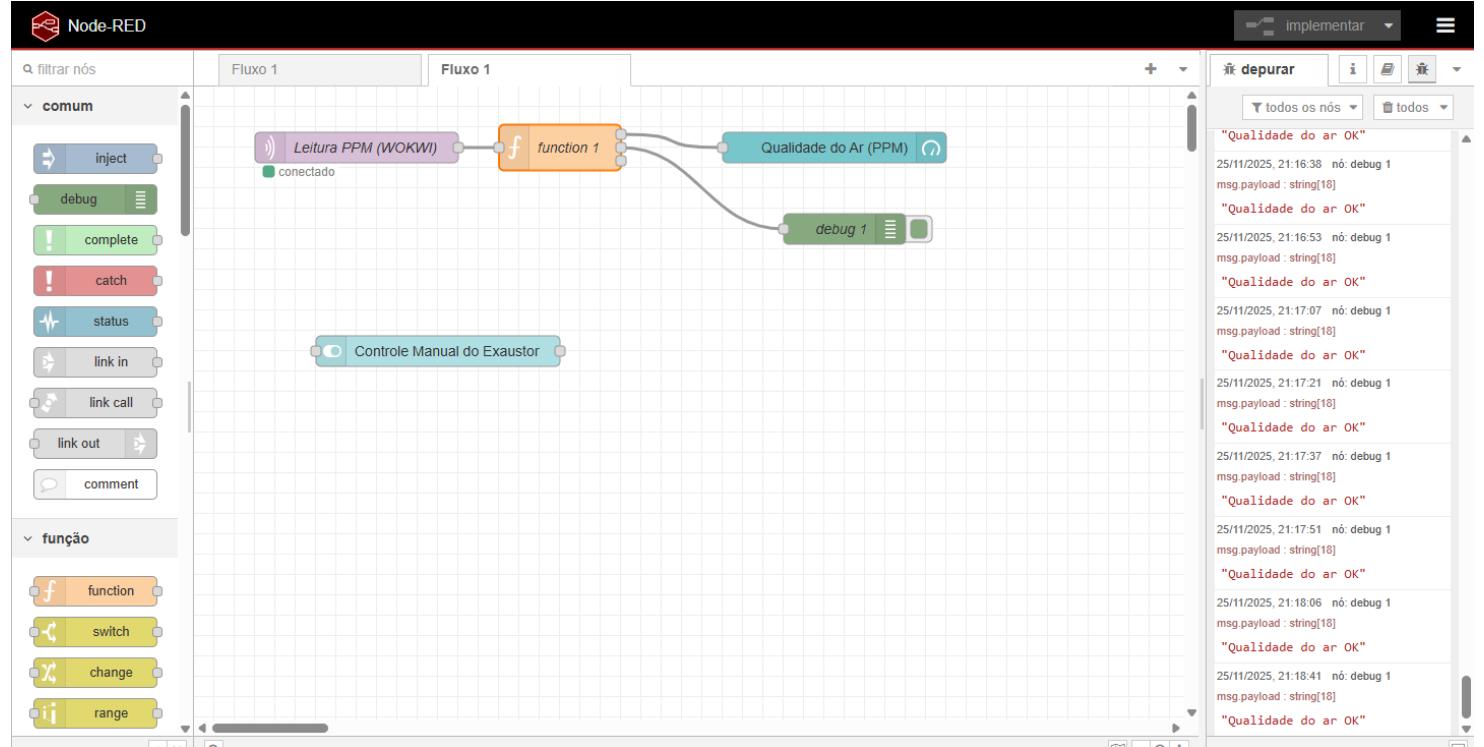


Fig. 3.3 - Detecção de nível muito ruim (318 PPM) e acionamento automático do exaustor

Valor alto de PPM enviado pelo ESP32 ao broker MQTT. O Node-RED exibe a faixa vermelha no gauge e o ESP32 aciona o LED (exaustor), indicando qualidade de ar muito ruim.

Wokwi simulation environment showing the Node-RED interface and the ESP32 sketch code. The sketch code includes functions for reading analog values, publishing air quality to MQTT, and controlling an actuator based on the PPM level. The Node-RED interface shows a gauge with a red segment indicating a value of 318 PPM, and a red arrow points to the LED pin in the circuit diagram.

```

int read_mq135_simulated() {
  Serial.print("Leitura Analógica: ");
  Serial.print(raw_value);
  Serial.print(" -> PPM: ");
  Serial.println(ppm);
  return ppm;
}

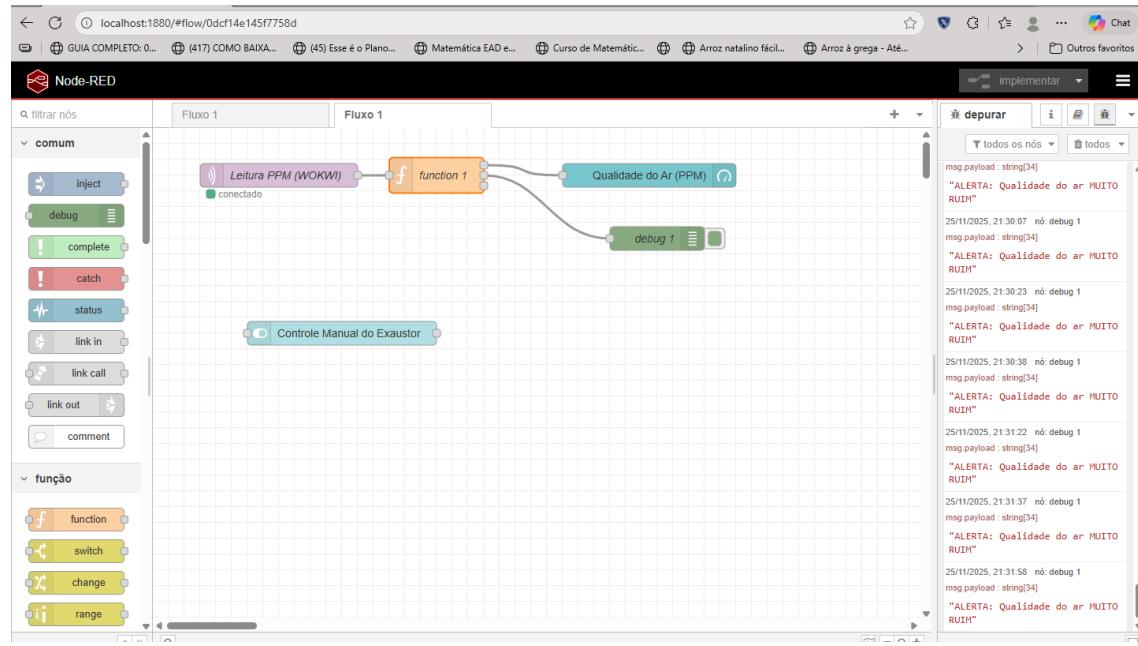
void publish_air_quality(int ppm) {
  char msg[10];
  sprintf(msg, "%d", ppm);

  client.publish(topic_publish_air_quality, msg);
  Serial.print("Publicado MQTT: ");
  Serial.println(msg);
}

void control_actuator(int ppm) {
  // Usa dois limites para diferenciar entre ruim e muito ruim
  if (ppm <= threshold_ppm) {
    digitalWrite(PIN_RELAY, LOW);
    Serial.println("Qualidade do ar OK");
  } else if (ppm > threshold_ppm) {
    // Nível RUIM: apenas alerta
    digitalWrite(PIN_RELAY, HIGH);
    Serial.println("ALERTA: Qualidade do ar RUIM.");
  } else {
    // Nível MUITO RUIM: aciona exaustor
    digitalWrite(PIN_RELAY, HIGH);
    Serial.println("ALERTA: Qualidade do ar MUITO RUIM.");
    Exaustor LIGADO.
    Tentando conexão MQTT...conectado!
  }
}
  
```

Fig. 3.4 - Node-RED exibindo alerta crítico com base nos dados MQTT

No Node-RED, o nó Function processa o valor recebido e emite o alerta “Qualidade do ar MUITO RUIM” no depurador, confirmando o funcionamento da lógica de categorização e a comunicação MQTT.



A demonstração completa do funcionamento do sistema está disponível em vídeo, acessível pelo link:
<https://youtu.be/wdpeUGbhRQ>

Medições:

Para avaliar a performance do sistema, foram realizadas medições de latência (tempo de resposta). Os resultados estão compilados na tabela abaixo.

Nº da Medida	Tipo de Medição	Tempo de Resposta (ms)
1	Sensor – Broker MQTT	123
2	Sensor – Broker MQTT	150
3	Sensor – Broker MQTT	135
4	Sensor – Broker MQTT	140
Média	Sensor – Broker MQTT	136,25
1	Broker MQTT - Atuador	180
2	Broker MQTT - Atuador	200
3	Broker MQTT - Atuador	175
4	Broker MQTT - Atuador	190
Média	Broker MQTT - Atuador	186,25



Figura 4: Gráfico de Tempo de Resposta do Sistema

O código-fonte, e a documentação completa do projeto estão disponíveis no seguinte repositório GitHub
<https://github.com/Marcia-Almeida/Objetos-Inteligentes---Projeto-Monitoramento-e-Controle-da-Qualidade-do-Ar>

7. Conclusões

Ao final deste trabalho, conclui-se que os objetivos propostos foram alcançados com sucesso. Foi desenvolvido e validado um protótipo funcional de um sistema IoT para monitoramento e controle da qualidade do ar, utilizando uma plataforma de simulação que permitiu testar a lógica de hardware e software de forma integrada.

O principal desafio enfrentado foi de natureza logística, relacionado à aquisição dos componentes físicos. A dificuldade em encontrar o microcontrolador e os sensores localmente, somada ao tempo de entrega de compras online, apresentou um risco ao cronograma do projeto. A solução adotada foi a utilização da plataforma de simulação Wokwi, que se mostrou uma alternativa robusta e eficaz, permitindo a implementação e o teste completo do firmware e da integração com o broker MQTT.

A principal vantagem do sistema proposto é sua alta acessibilidade e baixo custo, por se basear em componentes amplamente disponíveis e software de código aberto. A arquitetura desacoplada, utilizando MQTT, oferece grande flexibilidade e escalabilidade. Como desvantagem, a dependência de uma conexão Wi-Fi ativa representa um ponto crítico de falha para a comunicação com a nuvem.

Para futuras melhorias, o projeto poderia ser aprimorado com a implementação de um sistema de armazenamento de dados históricos em um banco de dados de séries temporais (como InfluxDB), permitindo análises de tendências a longo prazo. Adicionalmente, poderiam ser criados alertas mais sofisticados (via Telegram ou e-mail) e, fundamentalmente, a transição do protótipo simulado para uma montagem física robusta.

Referências

- [1] ORGANIZAÇÃO MUNDIAL DA SAÚDE (OMS). Air pollution. Disponível em: https://www.who.int/health-topics/air-pollution#tab=tab_1. Acesso em: 12 set. 2025.
- [2] NAÇÕES UNIDAS NO BRASIL. Objetivo 3: Saúde e Bem-Estar. Disponível em: <https://brasil.un.org/pt-br/sdgs/3>. Acesso em: 12 set. 2025.
- [3] AL-FUQAH, A.; GUERRAOUI, B.; ALEDROOS, A. The Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys & Tutorials*, v. 17, n. 4, p. 2347-2376, 2015.
- [4] SINGH, M.; SINGH, A. K.; SINGH, R. P. IoT based smart environment monitoring system using Arduino and Raspberry Pi. In: 2017 International Conference on Computer, Communications and Electronics (Comptelix). IEEE, 2017. p. 1-6.
- [5] HONG, J.; KIM, J.; LEE, S. Design and Implementation of IoT-Based Smart Home System Using MQTT Protocol. In: 2017 International Conference on Information and Communication Technology Convergence (ICTC). IEEE, 2017. p. 1146-1148.
- [6] KASHYAP, N.; GUPTA, A.; GUPTA, S. IoT based air quality monitoring system. In: 2017 International Conference on Computing, Communication and Automation (ICCCA). IEEE, 2017. p. 1-4.
- [7] IBM. MQTT: Por que é bom para IoT?. Disponível em: <https://www.ibm.com/developerworks;br/library/iot-mqtt-why-good-for-iot/index.html>.
- [8] ESPRESSIF. ESP32 Series Datasheet. Disponível em: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.
- [9] ESPRESSIF. ESP32 Technical Reference Manual. Disponível em: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf.
- [10] ESPRESSIF. ESP-IDF Programming Guide. Disponível em: <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/index.html>.
- [11] WINSEN SENSOR. MQ135 (Ver1.4) - Manual. Disponível em: [https://www.winsen-sensor.com/d/files/PDF/Semiconductor%20Gas%20Sensor/MQ135%20\(Ver1.4%20\)%20-%20Manual.pdf](https://www.winsen-sensor.com/d/files/PDF/Semiconductor%20Gas%20Sensor/MQ135%20(Ver1.4%20)%20-%20Manual.pdf).
- [12] SONGLERELAY. SRD-05VDC-SL-C Datasheet. Disponível em: <https://www.alldatasheet.com/datasheet-pdf/pdf/1132639/SONGLERELAY/SRD-05VDC-SL-C.html>.
- [13] HIVESMQ. MQTT Essentials. Disponível em: <https://www.hivemq.com/mqtt-essentials/>.
- [14] ARDUINO. Arduino IDE. Disponível em: <https://www.arduino.cc/en/software>.

- [15] NODE-RED. Node-RED. Disponível em: <https://nodered.org/>.
- [16] WOKWI.Wokwi – Online Simulator for Arduino and ESP32. Disponível em: <https://wokwi.com/>.
- [17] FRITZING. Fritzing. Disponível em: <https://fritzing.org/>.
- [18] EMBARCADOS. Ferramentas para Design de Circuitos Eletrônicos. Disponível em: <https://www.embarcados.com.br/ferramentas-para-design-de-circuitos-eletronicos/>.
- [19] EADUINO. Tutorial: Conhecendo o Fritzing (Parte 1). Disponível em: <http://www.eaduino.com.br/wp-content/uploads/downloads/2013/07/2013-Conhecendo-o-Fritzing-parte-I.pdf>.