Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Computação e Sistemas

Aplicação de MLOps em Sistemas Embarcados para Análise da Qualidade do Ar

Kevin Lucas de Oliveira Brito

TRABALHO DE CONCLUSÃO DE CURSO

ORIENTAÇÃO:
Igor Muzetti Pereira

Setembro, 2025
João Monlevade–MG

Kevin Lucas de Oliveira Brito

Aplicação de MLOps em Sistemas Embarcados para Análise da Qualidade do Ar

Orientador: Igor Muzetti Pereira

Monografia apresentada ao curso de Engenharia de Computação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina "Trabalho de Conclusão de Curso II".

Universidade Federal de Ouro Preto

João Monlevade

Setembro de 2025

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
INSTITUTO DE CIENCIAS EXATAS E APLICADAS
DEPARTAMENTO DE COMPUTACAO E SISTEMAS

FOLHA DE APROVAÇÃO

Kevin Lucas de Oliveira Brito

Aplicação de MLOps em sistemas embarcados para análise da qualidade do ar

Monografia apresentada ao Curso de Engenharia de Computação da Universidade Federal
de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Engenharia de Computação

Aprovada em 05 de Setembro de 2025

Membros da banca

Dr. Igor Muzetti Pereira - Orientador (Universidade Federal de Ouro Preto)
Dr. Euler Horta Marinho (Universidade Federal de Ouro Preto)
Dr. Racyus Delano Garcia Pacifico (Universidade Federal de Ouro Preto)

Igor Muzetti Pereira, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 23/09/2025

Documento assinado eletronicamente por **Igor Muzetti Pereira**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 23/09/2025, às 16:30, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Racyus Delano Garcia Pacifico**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 24/09/2025, às 16:31, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **0983246** e o código CRC **8B8368F3**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: (31)3808-0819 - www.ufop.br

Agradecimentos

Este trabalho é o resultado de uma jornada que não seria possível sem o apoio, incentivo e contribuição de diversas pessoas às quais gostaria de expressar minha sincera gratidão.

Aos meus pais, Katia Maria de Brito e Agnaldo da Silva, pelo amor incondicional, pelo apoio inabalável e por sempre acreditarem em meu potencial, mesmo nos momentos mais desafiadores. Sua dedicação e sacrifícios foram fundamentais para que eu chegasse até aqui.

Ao meu orientador, Prof. Igor Muzetti, pela orientação excepcional, pela paciência e pela disponibilidade em compartilhar seu conhecimento. Suas críticas construtivas e *insights* foram indispensáveis para o desenvolvimento e aprimoramento deste trabalho.

Aos meus amigos e colegas de turma, em especial meus irmãos da república Cativeiro, pela parceria, pelas discussões enriquecedoras e pelo apoio mútuo ao longo de toda a graduação. Vocês tornaram esta caminhada mais leve e significativa.

Por fim, agradeço a todas as pessoas que, de forma direta ou indireta, contribuíram para a realização deste trabalho e para a minha formação como profissional e cidadão.

A todos, meu mais sincero e eterno obrigado.

"Science is more than a body of knowledge; it is a way of thinking."

— Carl Sagan (1934 – 1996),
in: The Demon-Haunted World: Science as a Candle in the Dark.

Resumo

O monitoramento da qualidade do ar é um desafio crescente em áreas urbanas, devido ao impacto direto na saúde pública e à limitação de estações oficiais de medição, que possuem alto custo e baixa cobertura geográfica. Nesse contexto, este trabalho propõe um sistema integrado de *hardware* e *software* que combina Internet das Coisas (IoT) e *Machine Learning Operations* (MLOps), utilizando um protótipo baseado no microcontrolador ESP32 com sensores MQ-4 (metano), MQ-7 (monóxido de carbono) e DHT22 (temperatura e umidade). O sistema coletou dados ambientais contínuos por 30 dias em condições estáveis e processou-os em um pipeline MLOps automatizado, empregando o algoritmo K-means para análise. Os testes revelaram a capacidade do protótipo em identificar padrões de variabilidade nos gases e condições climáticas, embora a comparação com estações de referência públicas tenha evidenciado limitações na validação cruzada, devido a diferenças metodológicas e geográficas. Os resultados indicam que, mesmo com sensores de baixo custo, é possível obter agrupamentos consistentes para ambientes controlados. Como principais contribuições, destacam-se: (1) a aplicação de MLOps em dispositivos de baixo custo, (2) uma metodologia reprodutível para coleta e processamento de dados ambientais, e (3) uma arquitetura escalável que permite a integração de novos sensores. O trabalho abre perspectivas para sistemas de monitoramento ambiental distribuídos, de baixo custo, com inteligência artificial embarcada.

Palavras-chave: IoT. MLOps. Qualidade do ar. K-means.

Abstract

This work presents an integrated hardware and software system for air quality analysis, combining Internet of Things (IoT) and Machine Learning Operations (MLOps). The system is based on an ESP32 microcontroller equipped with MQ-4 (methane), MQ-7 (carbon monoxide), and DHT22 (temperature and humidity) sensors. Environmental data were continuously collected for 30 days under stable conditions and processed through an automated MLOps pipeline using the K-means algorithm. Comparison with public reference stations highlighted significant challenges in cross-validation due to methodological and geographical differences — particularly discrepancies in local measurement conditions. The main contributions of this work are: (1) the implementation of MLOps in low-cost devices, (2) a reproducible methodology for controlled environments, and (3) a scalable architecture that supports the integration of additional sensors. The study opens perspectives for massively distributed environmental monitoring systems with embedded artificial intelligence.

Keywords: IoT. MLOps. Air Quality. K-means.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

A qualidade do ar tem se tornado uma preocupação crescente devido ao aumento da poluição atmosférica, especialmente por gases como CO (monóxido de carbono) e CH_4 (metano), que, conforme alerta a OMS (FIOCRUZ, 2021), trazem riscos graves à saúde e aos ecossistemas. Embora existam sistemas públicos de monitoramento, como os disponibilizadas pelo INMET (Instituto Nacional de Meteorologia), muitas localidades ainda carecem de soluções acessíveis, em tempo real e com capacidade de análise inteligente.

A integração de tecnologias emergentes como Internet das Coisas (IoT) e práticas MLOps (*Machine Learning Operations*), conforme descrito pelos autores Smith et al. (2023) e Georgios Symeonidis (2022), ainda enfrenta barreiras técnicas, especialmente no que diz respeito à coleta contínua, processamento automatizado, e monitoramento da performance de modelos de aprendizado de máquina. Diante desse cenário, surge a necessidade de uma solução prática que una essas duas áreas para enfrentar o problema da qualidade do ar de forma eficiente e autônoma.

O objetivo geral desse trabalho envolve desenvolver e validar um sistema integrado de *hardware* e software para monitoramento em tempo real da qualidade do ar, baseado na Internet das Coisas (IoT) e em técnicas de Machine Learning Operations (MLOps). O sistema utiliza sensores ambientais de baixo custo acoplados a microcontroladores ESP32 para automatizar a coleta, o processamento e a análise de dados de poluentes atmosféricos, com ênfase na avaliação da eficiência do algoritmo de agrupamento *K-means* na identificação de padrões e anomalias. O projeto visa criar uma solução escalável, de baixo custo e reprodutível para aplicações em monitoramento ambiental.

Dos objetivos específicos, podemos citar:

- Aplicação de conceitos teóricos de MLOps:

  Implementar o ciclo de vida do modelo de *machine learning* seguindo práticas de MLOps, incluindo coleta, processamento, treinamento e monitoramento contínuo de dados, conforme fundamentado por Kreuzberger et al. (2023).

- Implementação de protótipo IoT:

  Desenvolver um protótipo de IoT baseado no microcontrolador ESP32, capaz de coletar e transmitir dados ambientais para uma plataforma central de processamento.

- Implementação do modelo de machine learning: Propor e avaliar um modelo de machine learning para análise preditiva da qualidade do ar, integrado a um pipeline

de MLOps que inclua versionamento de dados, treinamento automatizado e inferência em produção.

- Avaliação do modelo: Avaliar o desempenho do modelo de machine learning por meio de métricas quantitavas (acurácia, precisão) e qualitativas (análise de erros), comparando com *benchmarks* de soluções existentes para monitoramento da qualidade do ar.

O método adotado segue uma abordagem híbrida, combinando desenvolvimento empírico (prototipagem física) com técnicas de engenharia de dados e MLOps. As fases foram delineadas para garantir replicabilidade, escalabilidade e monitoramento contínuo, alinhando-se aos objetivos do sistema proposto.

O sistema proposto foi concebido a partir de uma arquitetura que integra sensores ambientais, um microcontrolador ESP32. A definição da arquitetura geral envolveu a elaboração de diagramas eletrônicos e testes preliminares, que permitiram verificar a viabilidade técnica da solução e sua resistência a interferências em condições reais de coleta. Na etapa de prototipagem, foi desenvolvido um programa de controle em C utilizando o *framework* ESP-IDF[1], com foco em eficiência energética e confiabilidade na leitura dos sensores. Essa abordagem buscou garantir o funcionamento contínuo do dispositivo no cenário de operação. Os dados ambientais coletados foram organizados em um formato padronizado e transmitidos de forma segura via Wi-Fi, utilizando o protocolo MQTT, com sincronização temporal precisa. Esse processo permitiu a integração com o pipeline de análise baseado em MLOps, no qual foram aplicadas etapas de pré-processamento e consistência. Para a análise dos dados, foi adotado o algoritmo de agrupamento K-means, explorado como método de identificação de padrões e anomalias. Além disso, o sistema foi projetado para contemplar o monitoramento de possíveis desvios nos dados e no desempenho do modelo, reforçando a escalabilidade e a robustez da solução desenvolvida.

O restante deste trabalho está organizado em sete capítulos sequenciais. O capítulo 2 aborda três pilares: sensoriamento ambiental, fundamentos de MLOps e trabalhos correlatos. O capítulo 3 detalha as seis fases do método, incluindo critérios de validação e ferramentas. O capítulo 4 apresenta o protótipo físico, métricas de desempenho do modelo e dados coletados em campo. O capítulo 5 relaciona os resultados com os objetivos do trabalho e soluções existentes. O capítulo 6 analisa restrições de *hardware* e viabilidade de escalonamento. Por fim o capítulo 7 sintetiza as contribuições técnicas e propõe direções futuras.

[1] Página do *framework* ESP32-32: https://www.espressif.com/en/products/socs/esp32

2 Referencial Teórico

A integração entre IoT e ML e regulações ambientais tem revolucionado o monitoramento da qualidade do ar, transformando dados brutos em *insights* acionáveis. Enquanto a IoT possibilita a captura distribuída de parâmetros atmosféricos (como CO e CH_4) através de sensores de baixo custo (Atzori et al., 2010), o MLOps garante a confiabilidade dos modelos preditivos desde o treinamento até a produção (Kreuzberger et al., 2023). Essa sinergia é catalisada por normas ambientais (diretrizes da OMS para qualidade do ar), que estabelecem *benchmarks* científicos para calibração ambiental, mas também criam novas fronteiras para cidades inteligentes e saúde preventiva.

2.1 Conceitos Básicos

IoT, sistemas IoT que integram dispositivos físicos com sensores, atuadores e conectividade para coleta e transmissão autônoma de dados (ATZORI; IERA; MORABITO, 2010). No monitoramento ambiental, destacam-se:

A arquitetura do sistema é composta por nós sensores baseados em ESP32, *gateways* de comunicação e plataformas de processamento em ambientes *cloud* e *edge*. As principais aplicações incluem a medição em tempo real de poluentes atmosféricos, como monóxido de carbono (CO), dióxido de nitrogênio (NO_2) e material particulado fino ($PM_{2.5}$), além do monitoramento de condições meteorológicas (GUBBI et al., 2013).

Machine learning operations MLOps é o conjunto de práticas para gerenciar o ciclo de vida de modelos de aprendizado de máquina em produção (KREUZBERGER; KüHL; HIRSCHL, 2023), enquanto ML engloba:

Inteligência Artificial (IA) refere-se a tecnologias capazes de simular processos de cognição humana, aplicadas em diferentes contextos relacionados ao monitoramento ambiental (PAUL; BHATTACHARYA, 2021). Entre suas aplicações destacam-se a predição, por meio de modelos voltados ao *forecast* de poluição, e a otimização, utilizando algoritmos genéticos para definir rotas de coleta de dados de forma mais eficiente. Outra técnica amplamente utilizada é o algoritmo K-means, que permite agrupar dados ambientais em diferentes categorias de similaridade, facilitando a identificação de padrões em variáveis como concentração de poluentes, temperatura e umidade. Essa abordagem possibilita análises exploratórias iniciais em cenários onde não há rótulos previamente definidos, auxiliando na compreensão de tendências e anomalias. Para determinar o número ideal de agrupamentos, utiliza-se com frequência o método do cotovelo, que consiste em calcular a soma dos erros quadráticos internos a cada cluster, conhecida como *Within-Cluster Sum*

of Squares (WSS). Ao plotar a WSS em função do número de clusters, observa-se que a redução dos erros tende a estabilizar a partir de certo ponto, formando um "cotovelo" no gráfico, o qual indica o valor mais adequado de k para o modelo.

No que se refere à integração de sistemas, a conectividade entre componentes heterogêneos é viabilizada pelo uso de protocolos adequados, como o MQTT, que possibilita uma transmissão leve, e o HTTP/REST, empregado em APIs. Além disso, a interoperabilidade entre sistemas é favorecida pela adoção de padrões consolidados de dados ambientais, como CSV ou JSON Schema (DIZIOLI; SANTOS, 2021).

Parâmetros de qualidade do ar correspondem a métricas estabelecidas e reguladas por órgãos internacionais, que servem de referência para a avaliação dos níveis de poluição atmosférica. Entre os principais indicadores, destacam-se o Air Quality Index (AQI), proposto pela Environmental Protection Agency (EPA) (AGENCY, 2020), e as diretrizes definidas pela Organização Mundial da Saúde (OMS) (ORGANIZATION, 2021). No monitoramento da qualidade do ar, a atenção recai especialmente sobre poluentes-chave, como o ozônio (O_3), o metano (CH_4), o dióxido de enxofre (SO_2) e o material particulado em diferentes granulometrias (PM_x).

Embora diferentes algoritmos de aprendizado de máquina possam ser aplicados ao monitoramento ambiental, como Árvores de Decisão, Redes Neurais ou *Support Vector Machines* (SVM), neste trabalho foi adotado o K-means por sua baixa complexidade computacional e pela capacidade de realizar agrupamentos em *clusters* de forma eficiente. Essa escolha se mostrou adequada principalmente para análises exploratórias iniciais. No entanto, como a base de dados utilizada é a mesma, não haveria impedimento em empregar outros algoritmos; bastaria implementá-los e consumir os mesmos conjuntos de dados para obter resultados comparativos.

2.2 Trabalhos Relacionados

Soluções Baseadas em IoT: Na área de soluções baseadas em IoT, Zhang et al. (2017) propuseram um sistema de monitoramento utilizando sensores de baixo custo e transmissão LoRa, cuja validação contra estações de referência reportou um RMSE (Raiz do Erro Quadrático Médio) de 4.2 $\mu g/m^3$ para $PM_{2.5}$ (ZHANG; LIU; WANG, 2017). De forma complementar, Spinelle et al. (2015) desenvolveram um protocolo específico para a validação de sensores de NO_2 e O_3 em ambientes urbanos, no qual destacaram a importância crítica da calibração contínua para garantir a precisão das medições (SPINELLE; GERBOLES; ALEIXANDRE, 2015).

Modelos Preditivos: No âmbito dos modelos preditivos para calibração de sensores, Zheng et al. (2013) realizaram um uso pioneiro do algoritmo *Random Forest* para correção de dados, conseguindo reduzir o erro de medição em expressivos 40% em relação às

leituras brutas (ZHENG; LIU; HSIEH, 2013). Complementarmente, Marina et al. (2018) conduziram uma revisão abrangente que sistematiza e compara diversas metodologias de calibração aplicáveis a sensores de baixo custo, fornecendo um panorama detalhado do estado da arte no monitoramento ambiental (MARINA; BALDACCI; STRACQUADANIO, 2018).

No contexto dos algoritmos, destacam-se o uso de redes neurais recorrentes para séries temporais *Long Short-Term Memory* (LSTM) e o algoritmo *Random Forest* para a classificação de riscos. Já no âmbito de MLOps, foram empregadas práticas como o versionamento de dados por meio do DVC (Data Version Control), a construção de pipelines utilizando ferramentas como MLflow e Kubeflow, além do monitoramento de *drift*, conforme discutido em (PALEYES; URMA; LAWRENCE, 2022).

Arquiteturas de Edge Computing: No campo das arquiteturas de *Edge Computing*, Mattia et al. (2022) implementaram uma solução de TinyML para detecção de anomalias em sensores de qualidade do ar diretamente em microcontroladores, otimizando o processamento de dados na borda. Avançando na praticidade dessas soluções, Deepak et al. (2023) propuseram um *framework* para atualização *Over-The-Air* (OTA) de modelos de aprendizado de máquina em dispositivos ESP32, abordando um desafio crítico para a manutenção remota e a escalabilidade de implantações de IoT (NADA, a; MATTIA; ROSSI; BIANCHI, 2022).

Lacunas Identificadas: A análise da literatura revela lacunas significativas que esta pesquisa busca abordar. Primeiramente, observa-se uma ausência de *pipelines* de MLOps completos, focados especificamente na etapa de implantação de modelos calibrados. Adicionalmente, há uma escassez de estudos que utilizem conjuntos de dados multi-sazonais com mais de 12 meses, os quais são cruciais para capturar a variabilidade sazonal dos poluentes. Outro ponto crítico é a baixa variabilidade nos conjuntos de dados de treinamento, o que limita a robustez e a generalização dos modelos de calibração para diferentes ambientes. Por fim, nota-se a carência de comparações sistemáticas e contínuas com bases públicas de referência, como as mantidas pela CETESB e EPA, que servem como *gold standard* para a validação.

3 Metodologia

A metodologia adotada nesse trabalho descreve, de forma detalhada, os procedimentos, ferramentas e técnicas utilizadas para o desenvolvimento do sistema proposto, contemplando desde a concepção da arquitetura até a validação final dos resultados. O processo metodológico foi estruturado em etapas sequenciais e interdependentes, abrangendo tanto a implementação física do *hardware* quanto a integração com a camada de *software* e o pipeline de MLOps.

Inicialmente, procedeu-se à prototipagem e montagem do circuito físico utilizando o microcontrolador ESP32 e sensores específicos para detecção de gases, temperatura e umidade. Em WSSseguida definiu-se a arquitetura geral do sistema, contemplando os módulos de coleta de dados, transmissão, processamento e análise. A etapa seguinte consistiu na implementação de um fluxo de comunicação entre o dispositivo e a infraestrutura de processamento, garantindo a transmissão confiável dos dados captados. Estes dados alimentam um fluxo de processamento de aprendizado de máquina automatizado, no qual são processados e analisados por um modelo baseado no algoritmo K-means, possibilitando a classificação do estado da qualidade do ar.

Por fim, a metodologia contempla o monitoramento contínuo do desempenho do modelo, a detecção de possíveis desvios (*drift*) e a validação dos resultados por meio de comparação com bases públicas de referência.

Como ilustrado na Figura 3.1, o processo compreende seis fases iterativas:

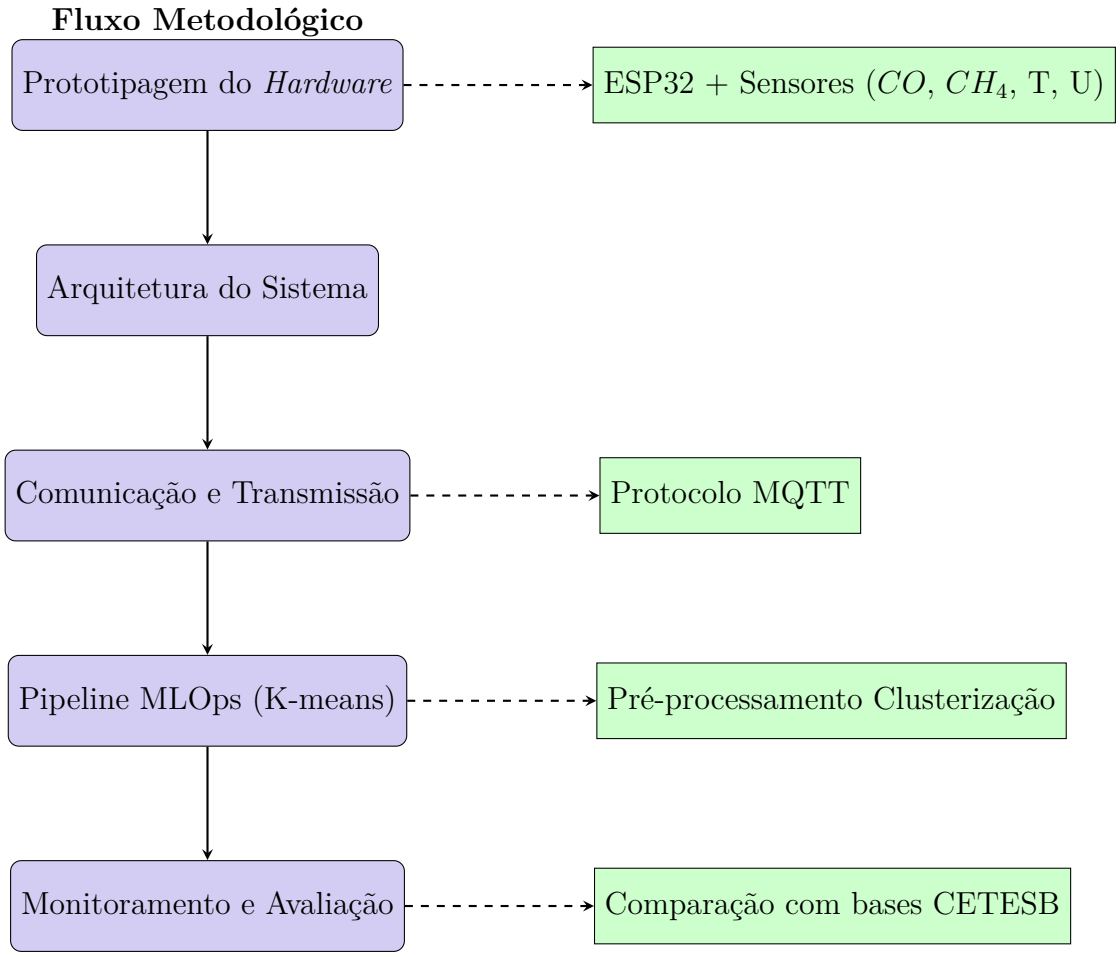

Figura 3.1 – Diagrama detalhado da metodologia proposta, mostrando as 6 fases principais (roxo), processos específicos (verde) e fluxo.

3.1 Arquitetura Geral do Sistema

A arquitetura do sistema proposto é organizada de forma modular, integrando *hardware* e *software* embarcado para realizar a coleta, transmissão, armazenamento e disponibilização dos dados ambientais. A solução é baseada no microcontrolador ESP32, que atua como unidade central do processamento e gerenciamento das tarefas de monitoramento.

O projeto foi estruturado em dois grandes eixos: camada física (*hardware*) e camada lógica (*software*).

Na **camada física**, o ESP32 é conectado aos sensores **MQ-4** (gás metano), **MQ-7** (gás monóxido de carbono) e **DHT22** (temperatura e umidade). Os sensores de gases utilizam a interface Conversor Analógico-Digital (ADC) do microcontrolador, enquanto o DHT22 utiliza comunicação digital. Esses sensores captam amostras periódicas do ambiente, que são processadas localmente antes de serem armazenadas.

Na **camada lógica**, o código-fonte é organizado em módulos independentes, conforme demonstrado na estrutura de pastas do projeto. A pasta `core` concentra funcionali-

dades essenciais, incluindo:

- **network**: gerenciamento de rede Wi-Fi, sincronização de tempo via Simple Network Time Protocol (SNTP) e disponibilização de um servidor web embarcado para acesso local.
- **sensors**: drivers específicos para cada sensor (DHT22, MQ-4, MQ-7), incluindo rotinas de inicialização, leitura periódica e tratamento de dados.
- **storage**: gerenciamento do sistema de arquivos SPIFFS, escrita de dados em formato CSV e funções para fusão e registro de logs.
- **shared**: espaço compartilhado para armazenar as leituras de sensores em memória, facilitando a comunicação de tarefas.

Complementarmente, a pasta `helpers` abriga funcionalidades auxiliares, como o módulo `supervisor`, responsável por monitorar o funcionamento geral do sistema, detectando falhas ou comportamentos inesperados, e encerrando sua execução.

O fluxo operacional inicia-se com a inicialização da memória NVS e do sistema de arquivos SPIFFS, seguida pela conexão com à rede Wi-Fi e sincronização do horário do sistema. Após a configuração dos arquivos CSV e a ativação do servidor web, o sistema inicializa cada sensor e cria tarefas independentes para a leitura contínua de dados. Em paralelo, a tarefa de supervisão garante a integridade e a estabilidade da operação.

Essa organização modular possibilita escalabilidade, manutenção facilitada e a integração transparente entre *hardware* e *software*, além de permitir que futuras expansões sejam incorporadas sem comporometer a estrutura existente.

A seguir, apresentam-se os diagramas fundamentais que representam a organização e o funcionamento do projeto, abrangendo desde a estrutura dos módulos até o fluxo de execução das principais interações.

O diagrama UML da Figura 3.2 detalha a organização estrutural do sistema, com ênfase em:

- **Classes críticas**: Como `SharedSensorData` e `CSVWriter`
- **Relacionamentos**: Dependências entre módulos (`Sensors` → `Storage`)
- **Padrões de projeto**: Identificação de Singleton (`SPIFFSManager`) e Observer (atualização de sensores)

Esta representação evidencia a arquitetura modular adotada, onde:

1. Módulos de baixo nível (`sensors`) não dependem de camadas superiores

2. A comunicação centralizada ocorre via `SharedSensorData`
3. Componentes de rede e armazenamento são isolados em pacotes distintos

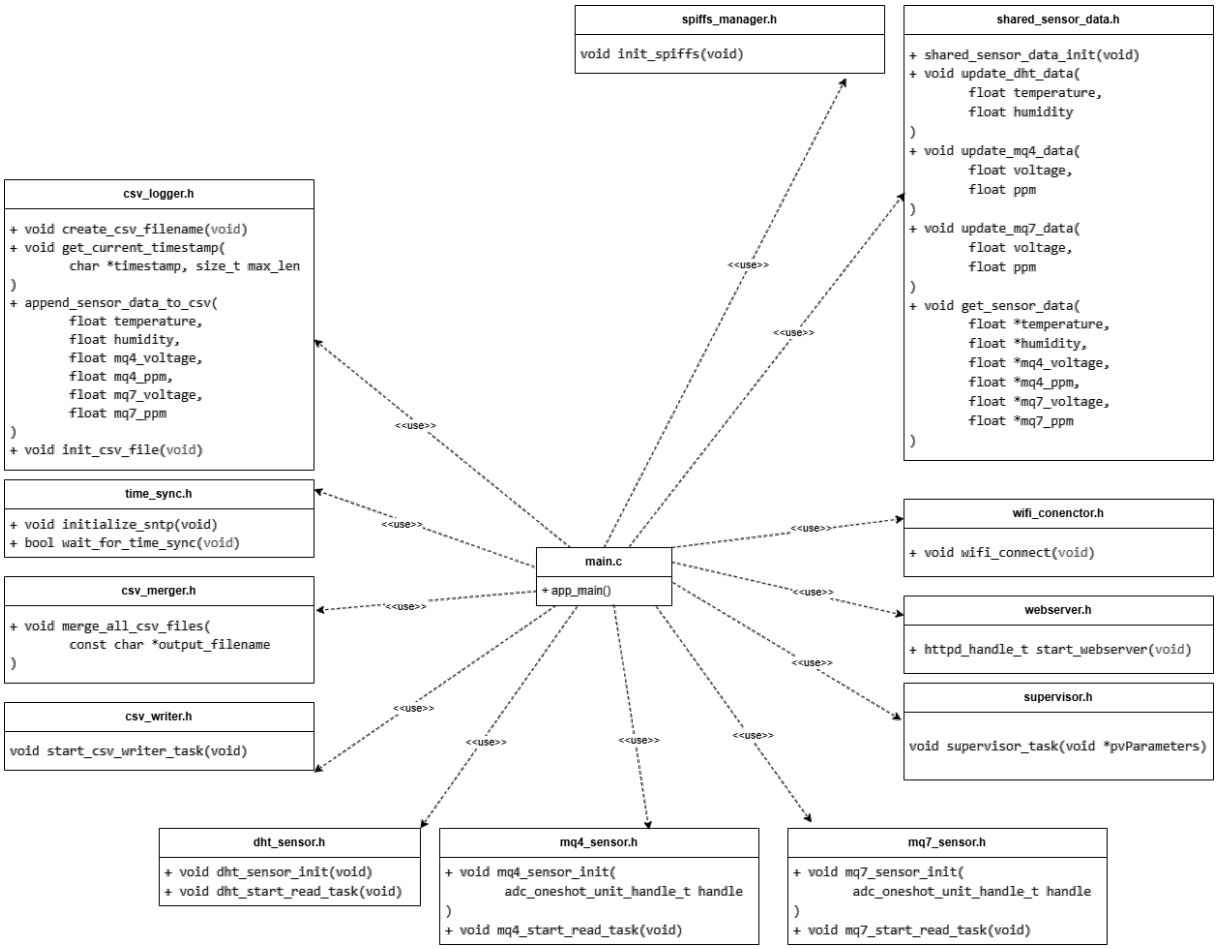

Figura 3.2 – Diagrama de arquivos UML do sistema.

O diagrama de pacotes da Figura 3.3 ilustra a organização modular do sistema, refletindo a estrutura de diretórios do projeto e a separação de responsabilidades. Identificam-se dois pacotes principais, compostos por módulos especializados:

- **Pacote `core`**: Agrupa os subsistemas fundamentais da aplicação:
  - `core_sensors`: Drivers e gerenciamento para os sensores (DHT, MQ4, MQ7)
  - `core_shared`: Estruturas de dados e recursos compartilhados entre módulos (`SharedSensorData`)
  - `core_network`: Módulo de comunicação em rede (MQTT, NTP, WebServer)
  - `core_storage`: Módulo de persistência e gerenciamento de dados (SPIFFS, CSV Logger)
- **Pacote `helpers`**: Fornece um utilitário auxiliar para o funcionamento do sistema:

– `supervisor`: Monitoramento de integridade e supervisão contínua do sistema

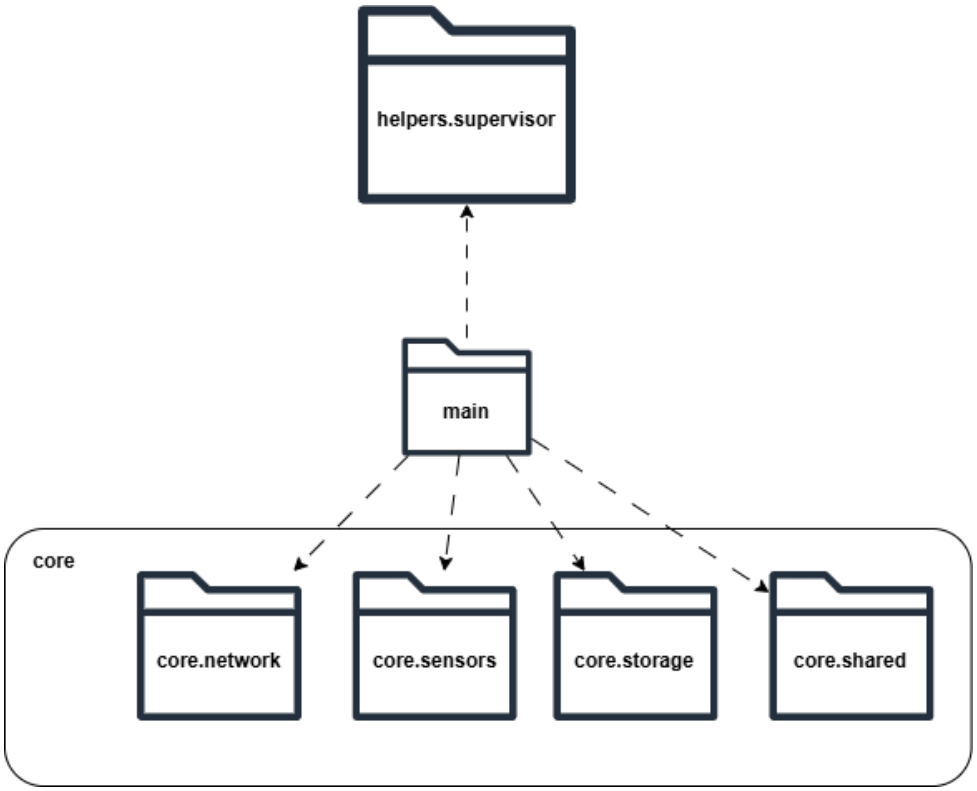

Figura 3.3 – Diagrama de pacotes ilustrando a arquitetura modular do sistema. Os módulos estão agrupados no pacote `core`, responsável pelas funcionalidades essenciais, e no pacote `helpers`, que provê suporte auxiliar.

O diagrama de sequência da (ver Apêndice A.1) detalha o fluxo de inicialização do sistema, ilustrando a coordenação temporal e as dependências entre os componentes críticos. O processo é dividido em duas fases principais.

A primeira fase, de **setup**, é caracterizada pela inicialização não bloqueante de periféricos de baixo nível, como o Non-Volatile Storage (NVS) e o sistema de arquivos SPIFFS. Esta etapa é seguida por uma sequência estrita de inicialização de rede: primeiramente a conexão Wi-Fi é estabelecida; uma vez conectado, o sistema procede com a sincronização do horário via protocolo NTP; e finalmente, somente após o horário estar sincronizado, o servidor web é inicializado.

A segunda fase inicia-se após a sincronização temporal bem-sucedida. Neste momento, ocorre a ativação concorrente das *tasks* responsáveis pela leitura dos sensores (DHT, MQ4 e MQ7). Paralelamente, a tarefa de escrita assíncrona de dados, gerenciada pelo módulo `CSVWriter`, também é iniciada.

Três pontos-chave emergem desta arquitetura: (1) o módulo `Supervisor` monitora continuamente o estado de saúde de todas as tarefas em execução

(representado visualmente por setas tracejadas mostradas no diagrama); (2) a estrutura `SharedSensorData` atua como um buffer de dados centralizado e thread-safe, facilitando a comunicação entre as tarefas de leitura dos sensores e a tarefa de escrita; e (3) existe uma barreira temporal explícita, onde o início da aquisição de dados dos sensores é condicionado à confirmação da sincronização horária (através da função `wait_for_time_sync`), garantindo que todas as medições possuam *timestamps* precisos e válidos.

A escolha do microcontrolador ESP32 como componente central do sistema foi fundamentada em uma análise criteriosa de aspectos técnicos, operacionais e econômicos, visando garantir desempenho adequado e eficiência na captura e transmissão dos dados ambientais.

Entre os principais fatores que justificam a utilização do ESP32, destacam-se:

- **Conectividade integrada**: O ESP32 possui conectividade Wi-Fi nativa, eliminando a necessidade de módulos externos para comunicação, o que simplifica o design do *hardware* e reduz custos e consumo energético.
- **Dual-core**: O microcontrolador possui dois núcleos de processamento independentes, o que permite a execução concorrente de tarefas críticas, como a leitura contínua dos sensores (Core 0) e a gestão do servidor web para interface com o usuário (Core 1), garantindo fluidez e responsividade ao sistema.
- **Periféricos adequados para sensores analógicos e digitais**: O ESP32 dispõe de conversores analógicos-digital (ADC) de 12 bits, essenciais para a leitura precisa dos sensores MQ-4 e MQ-7, que fornecem sinais analógicos proporcionais à concentração dos gases monitorados. Além disso, oferece pinos GPIO digitais configuráveis, utilizados para comunicação com o sensor DHT22 de temperatura e umidade.
- **Custo-benefício**: O ESP32 apresenta um excelente equilíbrio entre recursos avançados e baixo custo de aquisição, tornando-o uma opção viável para soluções embarcadas de monitoramento ambiental, especialmente em aplicações de larga escala ou com restrição orçamentária.

Dessa forma, a integração dos sensores com o microcontrolador explorou plenamente as capacidades do ESP32, possibilitando um fluxo operacional eficiente e confiável, como será detalhado nas próximas seções.

O sistema embarcado desenvolvido realiza a medição da qualidade do ar a partir de um fluxo contínuo e sincronizado que pode ser dividido em cinco etapas fundamentais: captura, transmissão, processamento, análise e alerta. A seguir, detalhamos cada uma

dessas fases, exemplificando seu funcionamento e integração a partir do código-fonte principal e dos módulos desenvolvidos.

1. **Captura de Dados** A captura dos parâmetros ambientais ocorre por meio de sensores especializados conectados ao microcontrolador ESP32, sendo realizada em tarefas independentes executadas pelo FreeRTOS.

   O FreeRTOS é um *kernel* de código aberto que implementa um escalonador preemptivo, permitindo a execução concorrente de múltiplas rotinas em um único núcleo de processamento. Ele aloca pequenos intervalos de tempo (*time slices*) para cada tarefa, criando a ilusão de execução paralela. Tarefas podem ser priorizadas, colocadas em estados de espera e sincronizadas por meio de mecanismos como filas (*queues*) e semáforos.

   A concorrência entre as tarefas de leitura de sensores e a tarefa de escrita (*logging*) exigiu a implementação de um mecanismo robusto para compartilhamento de dados *thread-safe*. Para isso, foi empregado um mutex (*Mutual Exclusion Object*) do FreeRTOS, garantindo integridade nas operações de leitura e escrita sobre a estrutura de dados global que armazena as últimas leituras de todos os sensores.

   O módulo `shared_sensor_data.c` encapsula essa funcionalidade. Seu funcionamento segue o protocolo padrão de acesso a regiões críticas:

   a) **Inicialização**: Ao iniciar o sistema, a função `shared_sensor_data_init()` cria o mutex por meio da primitiva `xSemaphoreCreateMutex()`.

   b) **Atualização (Escrita)**: Cada tarefa de sensor (DHT, MQ4, MQ7), ao obter uma nova leitura válida, chama sua função dedicada. Esta função, antes de modificar as variáveis globais, adquire o mutex com `xSemaphoreTake (sensor_data_mutex, portMAX_DELAY)`. A primitiva `portMAX_DELAY` bloqueia a tarefa indefinidamente até que o mutex seja liberado, garantindo que ela eventualmente executará sua escrita. Após a atualização das variáveis, o mutex é liberado com `xSemaphoreGive(sensor_data_mutex)`.

   c) **Consulta (Leitura)**: A tarefa de logging ou o servidor web, que precisam de uma *snapshot* consistente dos dados, utilizam a função `get_sensor_data()`. Esta função também adquire o mutex antes de copiar todos os valores para variáveis locais do *caller* e libera-o em seguida. Isso previne que uma tarefa de sensor altere os valores (e.g., temperatura) no momento exato em que a tarefa de logging está lendo-os (e.g., umidade), o que resultaria em um conjunto de dados inconsistente (e.g., temperatura nova com umidade antiga).

   Este design assegura que o acesso às variáveis globais `last_temperature`, `last_humidity`, etc., seja atomicamente exclusivo, prevenindo condições de corrida (*race conditions*)

e corrupção de dados. A escolha por um mutex (um semáfono binário) é ideal para este cenário, pois o recurso compartilhado deve ser acessado por apenas uma tarefa por vez, mas por múltiplas tarefas de diferentes prioridades.

- O **sensor DHT22** é responsável por medir temperatura e umidade do ar. Sua leitura é implementada no módulo *dht_sensor.c*, que configura o pino GPIO dedicado e cria uma tarefa (*dht_read_task*) que realiza leituras periódicas a cada 10 segundos. Os dados são validados contra limites mínimos e máximos para garantir a confiabilidade das informações.
- Os sensores **MQ-4** (metano) e **MQ-7** (monóxido de carbono) fornecem sinais análogicos convertidos pelo ADC de 12 bits do ESP32. Cada sensor possui seu módulo específico (*mq4_sensor.c* e *mq7_sensor.c*), que configura canais ADC, calibra automaticamente os sensores em ar limpo ao iniciar e executa tarefas de leitura periódicas. As leituras envolvem conversão da tensão medida em resistência do sensor e, posteriormente, em concentração estimada de gases em partes por milhão (ppm) usando curvas características e fórmulas matemáticas derivadas dos datasheets.
- Para garantir a consistência dos dados capturados, os módulos implementam filtros básicos de validação, descartando leituras fora de faixas físicas plausíveis e que apresentem variações bruscas entre amostras concecutivas.

Todos os dados validados são enviados para um armazenamento compartilhado (*shared_sensor_data.c*), que utiliza mutex do FreeRTOS para sincronização e permite acesso concorrente seguro por múltiplas tarefas.

2. **Transmissão dos Dados** Com os dados atualizados em tempo real, a transmissão é feita pela conectividade Wi-Fi nativa do ESP32, configurada na inicialização do sistema (*wifi_conncect*()), garantindo acesso à rede local.

   Um servidor web embarcado (*start_webserver*()), implementando a aplicação principal, disponibiliza os dados de sensores para consulta remota via interface HTTP, possibilitando monitoramento em tempo real por usuários ou sistemas externos.

   Além disso, os dados são registrados localmente em arquivos CSV no sistema de arquivos SPIFFS, gerenciados por tarefas específicas (*csv_writter.c*), o que permite o armazenamento histórico para análies posteriores e proteção contra eventuais falhas de comunicação.

   A sincronização temporal via protocolo NTP (*initialize_sntp*() e *wait_for_time_sync*()) assegura que todas as leituras estejam devidamente padrnizadas de acordo com o timestamp, essencial para correlação temporal das medições e análises subsequentes.

3. **Processamento dos Dados**

O processamento de dados é a etapa fundamental para garantir a qualidade e confiabilidade das informações que serão usadas nos modelos de aprendizado de máquina. No contexto do projeto, que envolve dados capturados por sensores (temperatura, umidade, gases), o processamento inclui:

- **Leitura dos dados brutos:** Os dados são inicialmente coletados e armazenados em formato CSV, contendo diversas leituras dos sensores ao longo do tempo.
- **Tratamento de valores inválidos:** É comum em dados de sensores a presença de valores nulos (`NaN`) ou valores zerados que indicam falhas ou leituras inválidas. Esses dados são removidos para evitar distorções.
- **Remoção de outliers:** Utilizamos o método do Intervalo Interquartil (IQR) para identificar e eliminar valores atípicos que fogem ao padrão normal dos sensores, garantindo que as análises futuras não sejam enviesadas por ruídos extremos.
- **Ordenação temporal:** Como os dados possuem um componente temporal, eles são ordenados por data e hora para manter a sequência cronológica correta, o que é importante para análises temporais posteriores.

Ao final, o conjunto de dados limpo e ordenado é salvo em arquivo próprio para ser consumido pela etapa seguinte de análise.

4. **Análise dos Dados** Nesta etapa, aplicamos técnicas de aprendizado de máquina para identificar padrões relevantes nos dados limpos, facilitando a compreensão do comportamento dos sensores e a segmentação das leituras em grupos similares.

   - **Clusterização com K-means:** O algoritmo K-means é utilizado para agrupar as leituras em *clusters* — grupos que compartilham características similares em termos de temperatura, umidade e concentração de gases. O número de *clusters* é definido baseado no método do cotovelo (*Elbow Method*), que ajuda a balancear a complexidade do modelo e a qualidade dos grupos formados.
   - **Pré-processamento:** Antes da clusterização, as variáveis são normalizadas usando `StandardScaler` para que todas tenham a mesma escala, evitando que variáveis com grande magnitude influenciem indevidamente a formação dos *clusters*.
   - **Treinamento e armazenamento do modelo:** O modelo treinado e o *scaler* são armazenados para que possam ser reutilizados para classificação de novas amostras.
   - **Visualização:** São gerados diversos gráficos que auxiliam na interpretação dos resultados, incluindo:

  - Gráficos 2D que mostram a relação entre variáveis sensoriais para cada cluster.
  - Pairplot para análise conjunta das distribuições e correlações entre as variáveis nos diferentes *clusters*.
  - Gráfico de médias dos sensores por cluster para entender o perfil médio de cada grupo.

  Essa etapa é crucial para validar se os *clusters* representam realmente diferentes estados ambientais ou condições de operação, além de facilitar a comunicação dos resultados.

5. **Alerta** Com o modelo de clusterização treinado e avaliado, a etapa de alerta tem o papel de monitorar em tempo real ou em novos conjuntos de dados as condições ambientais detectadas pelos sensores, usando a classificação por *clusters* para identificar situações críticas.

   - **Classificação:** Novas leituras são processadas pelo mesmo pipeline de pré-processamento (mesmo scaler) e passadas para o modelo K-means treinado para previsão do cluster ao qual pertencem.
   - **Definição de critérios para alerta:** Cada cluster pode ser associado a um estado operacional ou ambiental, como normal, atenção, ou crítico. Por exemplo, um cluster que contenha leituras com altos níveis de gás pode ser definido como crítico.
   - **Geração de mensagens de alerta:** Quando uma nova leitura é classificada em um cluster definido como crítico, um alerta é disparado. Isso permite intervenções rápidas e eficientes.

   Essa etapa traduz a análise técnica em ações práticas e imediatas para garantir segurança, eficiência e monitoramento contínuo dos sensores.

3.2 Prototipagem e Montagem do *Hardware*

Nesta etapa do projeto, o foco foi a construção do sistema físico de aquisição de dados ambientais por meio de sensores específicos, integrados ao microcontrolador ESP32. O processo envolveu a seleção dos componentes, montagem do circuito, definição dos protocolos de comunicação, calibração dos sensores e validação das leituras iniciais.

Para o desenvolvimento do protótipo foram utilizados os seguintes componentes principais:

Tabela 1 – Componentes principais utilizados no protótipo

Componente	**Função**
ESP32	Microcontrolador com conectividade Wi-Fi e Bluetooth, responsável pela leitura dos sensores e envio dos dados.
Sensor MQ-4	Detecção de gás metano (CH_4) no ambiente.
Sensor MQ-7	Detecção de monóxido de carbono (CO).
Sensor DHT22	Medição de temperatura e umidade relativa do ar.
Protoboard e cabos jumper	Montagem temporária e conexões entre componentes.

O circuito foi projetado e documentado utilizando a ferramenta Fritzing, que permitiu a criação do diagrama esquemático das conexões entre os sensores e o microcontrolador ESP32. O esquema detalha as ligações dos sensores às portas analógicas e digitais do ESP32, assim como a alimentação e o aterramento do sistema, garantindo a integridade elétrica e a segurança dos componentes.

Abaixo é apresentada a simulação do circuito implementado para o projeto.

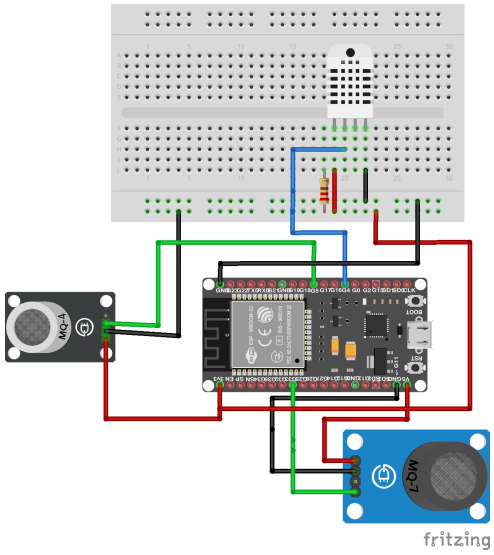

Figura 3.4 – Simulação do circuito Fritzing.

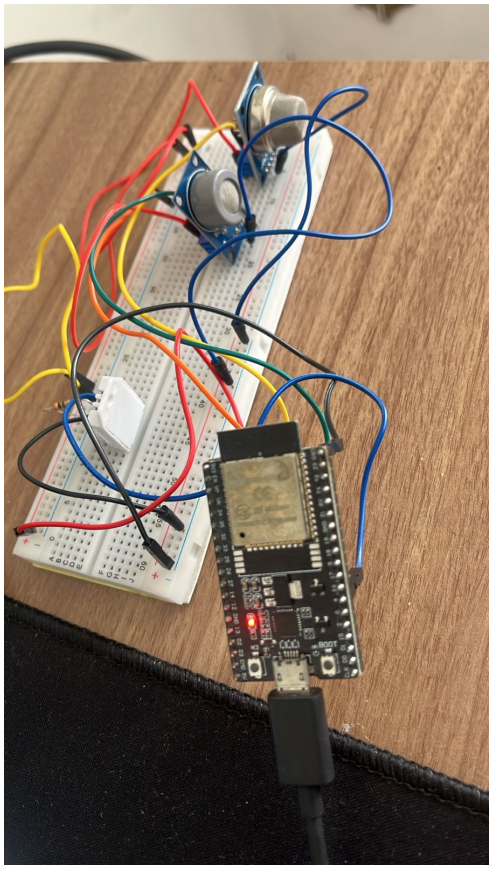

Figura 3.5 – Protótipo Desenvolvido.

Os sensores de gás MQ-4 e MQ-7 necessitam de um procedimento inicial de pré-aquecimento para estabilização do sensor e obtenção de leituras precisas. Este tempo varia entre 20 a 60 minutos, conforme especificação dos fabricantes, para que o elemento sensor aqueça internamente e atinja a sensibilidade ideal.

Após o pré-aquecimento, foi realizado o ajuste dos valores base, considerando as leituras em ambiente livre de contaminantes para estabelecer a linha de base. Esse processo é fundamental para reduzir ruídos e compensar variações ambientais, como temperatura e umidade, que possam influenciar na resposta dos sensores.

Com o circuito montado e sensores calibrados, foram realizados testes iniciais para a validação das leituras coletadas pelo ESP32. Utilizando código-fonte específico para leitura analógica dos sensores MQ e decodificação digital do DHT22, os valores foram capturados, exibidos via monitor serial e armazenados para posterior análise.

Esses testes confirmaram a funcionalidade do sistema de aquisição e serviram como base para o desenvolvimento do *software* de pré-processamento e análise, garantindo que os dados coletados fossem confiáveis para as etapas subsequentes do projeto.

3.3 Coleta e Trasmissão dos Dados

Nesta etapa, abordamos a configuração do ESP32 para estabelecer a comunicação em rede, o protocolo utilizado para o envio dos dados coletados pelos sensores, o formato em que esses dados são transmitidos, além da definição da frequência de amostragem adotada para garantir a atualização adequada das informações.

O microcontrolador ESP32 foi configurado para conectar-se a uma rede Wi-Fi local utilizando o *framework* ESP-IDF (*Espressif IoT Development Framework*). As credenciais da rede (SSID e senha) são armazenadas em variáveis durante a fase de desenvolvimento e utilizadas pela pilha Wi-Fi do ESP-IDF (`esp_wifi.h`) para estabelecer a conexão durante a inicialização do dispositivo. Esta abordagem oferece maior controle sobre os parâmetros de conexão e eventos de rede em comparação com abstrações de mais alto nível.

Para garantir a estabilidade da conexão, o sistema implementa rotinas de reconexão automática em caso de queda de sinal ou perda temporária da rede, assegurando a continuidade do envio dos dados para o servidor ou broker.

O protocolo escolhido para a transmissão dos dados foi o Message Queuing Telemetry Transport (MQTT), devido à sua leveza, baixo consumo de banda e robustez para aplicações IoT em tempo real.

```
I (816) wifi:state: auth -> assoc (0x0)
I (826) wifi:state: assoc -> run (0x10)
I (956) wifi:connected with CATIVEIRO, aid = 9, channel 11, BW20, bssid = e8:45:8b:36:88:10
I (956) wifi:security: WPA2-PSK, phy: bgn, rssi: -31
I (966) wifi:pm start, type: 1

I (966) wifi:dp: 1, bi: 102400, li: 3, scale listen interval from 307200 us to 307200 us
I (976) WIFI: Connected to AP
I (1026) wifi:AP's beacon interval = 102400 us, DTIM period = 3
I (1336) wifi:<ba-add>idx:0 (ifx:0, e8:45:8b:36:88:10), tid:0, ssn:0, winSize:64
I (2386) esp_netif_handlers: sta ip: 192.168.15.68, mask: 255.255.255.0, gw: 192.168.15.1
I (2386) WIFI: Got IP:192.168.15.68
I (2606) wifi:<ba-add>idx:1 (ifx:0, e8:45:8b:36:88:10), tid:7, ssn:6, winSize:64
I (2796) TIME_SYNC: Waiting for system time to be set... (2/10)
I (4796) TIME_SYNC: Time synchronized: Sat Aug 30 13:36:07 2025

I (4796) SPIFFS: Initializing SPIFFS
```

Figura 3.6 – Prompt de inicialização do pipeline, com conexão Wi-Fi.

A Figura 3.6 apresenta o prompt de inicialização do pipeline no microcontrolador. Nesse processo, a placa tenta se conectar à rede Wi-Fi utilizando as credenciais configuradas no código. Em caso de sucesso, é realizada a sincronização temporal via protocolo SNTP, garantindo registros consistentes. São feitas múltiplas tentativas de conexão; se todas falharem, a execução é interrompida. Caso contrário, ocorre a inicialização do sistema de gerenciamento SPIFFS, com a criação de um novo arquivo CSV para armazenamento, e em seguida é iniciada a captura dos parâmetros ambientais.

Cada entrada no arquivo de log CSV (`merged.csv`) contém um *snapshot* completo do estado dos sensores, incluindo tanto os valores finais processados quanto os dados brutos

intermediários, seguindo a estrutura ilustrada no exemplo da Tabela 2:

- `Timestamp`: Marco temporal no formato `AAAA-MM-DD HH:MM:SS`, registrando o momento exato da aquisição dos dados.
- `Temperature(C)`: Valor de temperatura ambiente em graus Celsius (ºC).
- `Humidity(%)`: Valor de umidade relativa do ar em porcentagem (%).
- `MQ4_Voltage(V)`: Tensão analógica bruta (em Volts) lida do sensor MQ-4.
- `MQ4_PPM`: Concentração de gás metano (CH_4) em partes por milhão (ppm), calculada a partir da tensão bruta do MQ-4.
- `MQ7_Voltage(V)`: Tensão analógica bruta (em Volts) lida do sensor MQ-7.
- `MQ7_CO_PPM`: Concentração de monóxido de carbono (CO) em partes por milhão (ppm), calculada a partir da tensão bruta do MQ-7.

A decisão de incluir tanto as tensões brutas quanto os valores convertidos em PPM oferece transparência total ao processo de aquisição e calibração. Isso permite a verificação dos dados crus, a recalibração *offline* com novos parâmetros e a análise detalhada do comportamento dos sensores, facilitando a interoperabilidade e o processamento posterior por ferramentas externas.

Timestamp	**Temp. (ºC)**	**Umid. (%)**	**MQ4 (V)**	**MQ4 (PPM)**	**MQ7 (V)**	**MQ7 (PPM)**
2025-07-08 07:25:54	0.0	0.0	0.00	0.00	0.00	0.00
2025-07-08 07:55:54	17.7	71.0	0.59	4.49	0.37	0.64
2025-07-08 08:25:54	17.7	71.2	0.60	4.76	0.37	0.64
2025-07-08 08:55:54	17.7	71.3	0.61	4.86	0.37	0.65
2025-07-08 09:25:54	17.7	71.2	0.60	4.62	0.37	0.64

Tabela 2 – Exemplo de registros extraídos do arquivo CSV com variáveis ambientais e gases monitorados.

A frequência de coleta dos dados foi definida em um intervalo fixo de 30 minutos. Este valor foi escolhido para equilibrar a necessidade de um monitoramento significativo (quase em tempo real para certas aplicações) com a conservação crítica de recursos do dispositivo, como vida útil da bateria, desgaste da memória *flash* (SPIFFS).

Nota: A presença de valores de tensão bruta (e.g., `MQ4_Voltage(V)`) junto aos valores convertidos (e.g., `MQ4_PPM`) é uma prática essencial para a validação científica e engenharia de dados. Ela permite isolar falhas: um valor de PPM anômalo pode ser causado por um problema no sensor (detectado pela tensão bruta anormal) ou por um erro no algoritmo de conversão (detectado se a tensão for normal, mas o PPM calculado não).

Adicionalmente, os valores iniciais zerados (0.00) são um artefato esperado do período de desconexão dos sensores de gás após uma inicialização do sistema.

A Figura 3.7 ilustra o fluxo completo de dados no sistema, desde a captação inicial pelas leituras dos sensores até a etapa de análise automatizada utilizando técnicas de clusterização. Esse diagrama permite visualizar de forma integrada como as diferentes fases do processo — aquisição, armazenamento, transmissão e processamento — se relacionam para viabilizar o monitoramento ambiental contínuo e a extração de padrões relevantes a partir das informações coletadas.

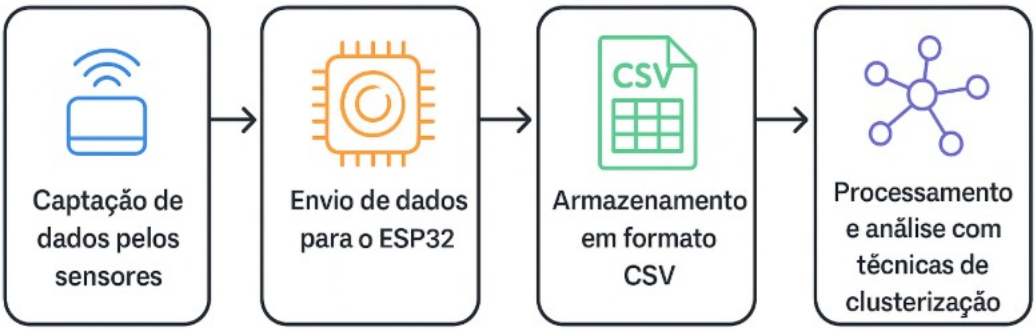

Figura 3.7 – Fluxo de dados no sistema: desde a captação pelos sensores até a análise automatizada utilizando técnicas de clusterização.

3.4 MLOps

O pipeline MLOps é um conjunto integrado de processos e ferramentas que possibilitam a automação e o monitoramento do ciclo de vida dos modelos de aprendizado de máquina, desde a ingestão dos dados até o *deploy* e manutenção em produção.

No presente trabalho, o pipeline foi estruturado para executar de forma automatizada em tempo de execução, processando os dados à medida que são coletados pelos sensores. Isso significa que as etapas de pré-processamento, agrupamento e armazenamento ocorrem de maneira contínua e integrada, sem necessidade de intervenção manual durante o fluxo. Essa característica torna o sistema mais robusto e escalável, permitindo adaptações futuras com outros algoritmos ou novos sensores, mantendo a mesma lógica de automação.

Para o desenvolvimento deste pipeline, foram implementados *scripts* em Python personalizados, responsáveis por gerenciar as etapas de tratamento dos dados, treinamento e inferência dos modelos. A escolha por *scripts* específicos se deu pela flexibilidade necessária para adaptar as rotinas às particularidades do projeto.

No processamento e análise dos dados, foram empregadas bibliotecas como Pandas e NumPy. A modelagem e avaliação utilizaram recursos da scikit-learn, incluindo algoritmos de agrupamento (*K-Means*), pré-processamento (*StandardScaler*) e métricas de desempenho

(MAE, MSE e silhouette score). A visualização de dados foi conduzida com Matplotlib e Seaborn.

Para versionamento e reutilização de modelos e transformadores, empregou-se a biblioteca `joblib`, permitindo o salvamento e carregamento eficiente dos artefatos. Adicionalmente, módulos padrão do Python, como `datetime`, `os` e `logging`, foram utilizados para controle de fluxo, manipulação de arquivos e registro de eventos.

É importante destacar que o processo de agrupamento por meio do algoritmo K-means não interfere na operação dos sensores. Os dispositivos permanecem apenas na função de coleta e envio de dados ambientais, sem qualquer alteração no seu funcionamento interno. O algoritmo é executado posteriormente, sobre os dados já coletados e armazenados, de forma a identificar padrões e agrupamentos. Dessa forma, garante-se que o *hardware* mantenha sua simplicidade e confiabilidade, enquanto a inteligência do sistema é concentrada na camada de processamento.

A estrutura do pipeline compreende as seguintes etapas principais:

1. Ingestão dos dados

   Os dados brutos provenientes dos sensores são coletados e armazenados em arquivos CSV, que servem como base para todo o processo de análise. A ingestão consiste na leitura destes arquivos no ambiente de processamento, garantindo a disponibilidade das informações atualizadas para as etapas subsequentes.

2. Pré-processamento e limpeza

   Nesta etapa, os dados passam por rotinas de limpeza que incluem a remoção de valores nulos ou inválidos, exclusão de *outliers* utilizando o método do intervalo interquartil (IQR) e ordenação cronológica das amostras. Essas ações garantem a qualidade e consistência das informações para o treinamento do modelo.

   Para otimizar o desempenho do algoritmo de clusterização K-means, as variáveis foram padronizadas utilizando o `StandardScaler` do Scikit-learn. A padronização consiste em transformar os dados para que tenham média zero e variância unitária, facilitando a convergência do modelo e a comparabilidade entre diferentes features.

3. Armazenamento

   Os dados processados e enriquecidos com as informações de cluster são armazenados em arquivos CSV em diretórios específicos do projeto, que funcionam como um repositório organizado (*data lake* local).

Embora o *deploy* automático para ambientes de produção não esteja implementado nesta fase, a estrutura do pipeline facilita a integração com ferramentas CI/CD e orques-

tradores, permitindo que em versões futuras o fluxo seja completamente automatizado e monitorado.

3.5 Implementação do Modelo de ML

Nesta etapa, o algoritmo escolhido para análise e clusterização dos dados coletados foi o K-means, um dos métodos de agrupamento não supervisionado mais utilizados devido à sua simplicidade e eficiência computacional.

3.5.1 Algoritmo Escolhido: K-means

O K-means é um algoritmo que busca particionar os dados em K *clusters*, minimizando a soma das distâncias quadráticas entre os pontos e os centróides dos grupos. Ele é especialmente adequado para detectar grupos de dados que possuem formas aproximadamente esféricas e tamanhos semelhantes.

Para definir o valor ideal de K (número de *clusters*), foi aplicada uma técnica clássica:

Método do Cotovelo (*Elbow Method*): Consiste em calcular a soma dos quadrados intra-cluster *Within-Cluster Sum of Squares* (WSS) para diferentes valores de K e identificar o ponto em que a redução do erro deixa de ser significativa, formando um "cotovelo" no gráfico. Este ponto indica um bom equilíbrio entre complexidade do modelo e qualidade da clusterização.

A partir dessas análises, optou-se por um valor de K=4, que apresentou um balanceamento adequado entre qualidade dos agrupamentos e interpretabilidade dos resultados.

Dado que os dados dos sensores podem conter leituras errôneas, ruídos ou valores discrepantes, adotou-se uma estratégia de pré-processamento para mitigar o impacto destes dados no modelo:

- Remoção de Outliers: Utilizou-se o método do intervalo interquartil (IQR) para identificar e remover pontos que estejam significativamente distantes dos quartis, considerados como outliers. Essa abordagem ajuda a evitar que valores extremos distorçam a formação dos *clusters*.
- Filtro de Valores Nulos e Zeros: Dados com valores nulos ou zero (que indicam falha na leitura dos sensores) foram eliminados para garantir a integridade das informações.

Embora o K-means seja amplamente utilizado e apresente baixo custo computacional, os resultados iniciais não foram satisfatórios para este projeto. Uma possível explicação

é que a maior parte dos dados foi coletada em um mesmo ambiente, resultando em baixa variabilidade nas leituras, o que dificultou a separação clara entre *clusters*.

3.6 Monitoramento e Operacionalização do Modelo

Para garantir a robustez e a eficiência contínua do modelo de ML implementado, é fundamental estabelecer um processo estruturado de monitoramento e operacionalização, visando detectar e reagir a mudanças nos dados e no desempenho do modelo em produção.

Embora o K-means seja um algoritmo não supervisionado, algumas métricas internas são utilizadas para avaliar a qualidade dos *clusters* e monitorar a estabilidade do modelo ao longo do tempo:

- Inércia *Within-Cluster Sum of Squares*: Mede a soma das distâncias quadráticas entre os pontos e seus respectivos centróides, indicando a coesão interna dos *clusters*.
- *Silhouette Score*: Avalia a separação entre os *clusters*, auxiliando a identificar se as fronteiras entre grupos continuam bem definidas.

Além disso, no contexto operacional, a qualidade dos dados coletados (completude, ausência de valores nulos, frequência correta) também é monitorada para assegurar a integridade do pipeline.

Com o tempo, é comum que os dados de entrada sofram alterações em suas distribuições estatísticas — fenômeno conhecido como *data drift* — que pode comprometer a efetividade dos *clusters* gerados. Para mitigar esse risco, são implementadas rotinas de monitoramento que:

- Detectam alterações na distribuição dos dados: Comparando estatísticas descritivas (média, variância, quartis) e características dos *clusters* entre janelas temporais distintas.
- Avaliam a estabilidade dos *clusters*: Verificando mudanças significativas nos centróides ou na composição dos *clusters*, que indicam possível degradação do modelo.

Essas análises permitem identificar precocemente situações em que o modelo não está mais representando adequadamente os dados atuais.

3.7 Implementação e Disponibilidade do Código-Fonte

Todo o código-fonte, scripts de análise, modelos de ML e documentação complementar estão disponíveis publicamente em um repositório GitHub, visando garantir a transparência, reprodutibilidade e o avanço aberto do projeto.

- **Repositório:** <https://github.com/kevin504-max/MLOps>

4 Resultados

Nesta seção são apresentados os resultados obtidos com a implementação e avaliação do sistema de monitoramento ambiental proposto. Com o objetivo de aprimorar a eficiência do algoritmo de classificação, o final dos testes foi conduzido condições induzidas, que consistiu na manipulação intencional de variáveis ambientais, como aumento da umidade e elevação dos níveis de gases, com o intuito de avaliar a sensibilidade e a resposta do sistema, além de aumentar a variabilidade dos parâmetros captados, otimizando o desempenho do algoritmo. Nesse cenário, foram aplicadas técnicas de *cluster*ização a fim de explorar padrões nos dados e otimizar o agrupamento de amostras, utilizando as abordagens de *Elbow* e *Silhouette* para determinar o número ideal de *clusters* e validar a consistência dos agrupamentos obtidos. A seguir, detalham-se as características dos dados coletados e os resultados alcançados com as diferentes metodologias de análise.

As medições foram realizadas em um intervalo recorrente de 30 minutos ao longo do período de 30 dias. Esse intervalo, no entanto, não se manteve perfeitamente regular, uma vez que a coleta dependia de condições de vigilância e monitoramento do protótipo em funcionamento. Os registros foram obtidos em minha residência, abrangendo diferentes ambientes, como cômodos internos, quintal e áreas abertas e arejadas, de forma a capturar variações nas condições ambientais de cada espaço. Em média, cada dia contou com cerca de 40 a 45 registros de variáveis ambientais, resultando em uma base de dados suficientemente densa para a aplicação das técnicas de *cluster*ização e para a validação das análises realizadas. Essa estratégia de amostragem buscou equilibrar a disponibilidade de dados e a capacidade de armazenamento e processamento do sistema.

Quanto ao armazenamento local, a memória interna do dispositivo é limitada (4 MB de flash), sendo utilizada apenas de forma temporária para buffer dos dados em caso de falhas de conectividade, já que o fluxo principal é direcionado diretamente ao servidor para processamento. É importante destacar que os sensores MQ-4 e MQ-7 foram calibrados previamente à execução do algoritmo K-means, garantindo que os valores captados correspondessem a concentrações próximas das condições reais de medição. Assim, o K-means não atua como etapa de calibração dos sensores, mas sim como técnica de agrupamento aplicada sobre os dados já coletados e calibrados.

4.1 Abordagem com *Elbow Method*

Para determinar o número ideal de *clusters* na análise de agrupamento dos dados ambientais, utilizamos o método do cotovelo (*Elbow Method*) aplicado ao algoritmo K-Means. Inicialmente, selecionamos as variáveis mais relevantes para o monitoramento,

incluindo temperatura, umidade, concentração de metano ($MQ4\_PPM$) e monóxido de carbono ($MQ7\_CO\_PPM$). Em seguida, normalizamos os dados utilizando o StandardScaler, de modo a padronizar a escala das variáveis e evitar que características com magnitudes maiores dominassem o cálculo das distâncias. O K-Means foi então executado para valores de K variando de 1 a 10, registrando-se o WSS (Within-*cluster* Sum of Squares ou inércia), que indica a soma das distâncias quadráticas entre cada ponto e o centróide do *cluster* correspondente. Ao plotar o WSS em função de K, procuramos o ponto em que a redução da inércia começa a se estabilizar, caracterizando o "cotovelo" da curva, o qual sugere o número de *clusters* que melhor representa a estrutura subjacente dos dados, equilibrando a complexidade do modelo e a homogeneidade dos agrupamentos. A figura resultante do *Elbow Method* foi gerada e salva para análise visual, permitindo uma escolha fundamentada do número de *clusters* a ser utilizado nas etapas subsequentes de *cluster*ização.

A Figura 4.1 mostra a curva característica do método do cotovelo, onde se observa uma queda acentuada da inércia entre 1 e 4 *clusters*, redução mais suave a partir de 4 *clusters* e o ponto de inflexão claramente marcado em $K = 4$.

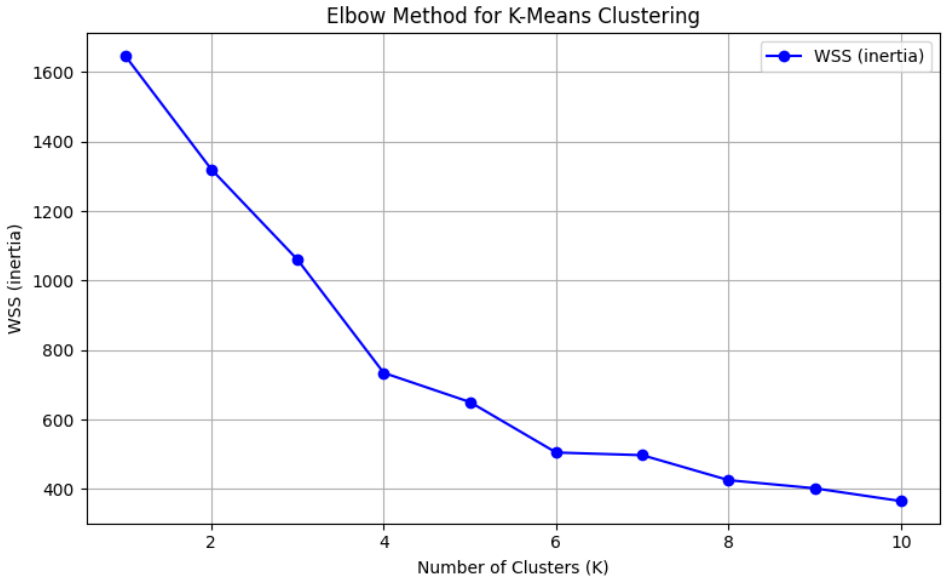

Figura 4.1 – Gráfico do método Elbow para determinação do número ótimo de *clusters* em condições controladas. O eixo vertical representa a inércia (WSS) na faixa de 400 a 1600, enquanto o eixo horizontal mostra o número de *clusters* testados (1 a 10). O ponto de cotovelo em $K = 4$ está destacado.

O comportamento típico do gráfico confirma que 4 *clusters* representam o melhor compromisso entre complexidade do modelo e qualidade do agrupamento para nossos dados coletados. A inércia inicial de aproximadamente 1600 para um único *cluster* cai drasticamente até $K = 4$, onde a curva começa a se estabilizar, indicando que *clusters* adicionais não melhorariam significativamente a compactação dos grupos.

A Figura 4.2 apresenta o resultado da *cluster*ização K-Means aplicado à relação entre temperatura e concentração de monóxido de carbono (MQ7_CO_PPM) nas condições do experimento. O *score* de *Silhouette* de 0.36 indica uma estrutura de agrupamento pouco razoável, com forte sobreposição entre os *clusters*, porém com distinção perceptível de alguns grupos formados.

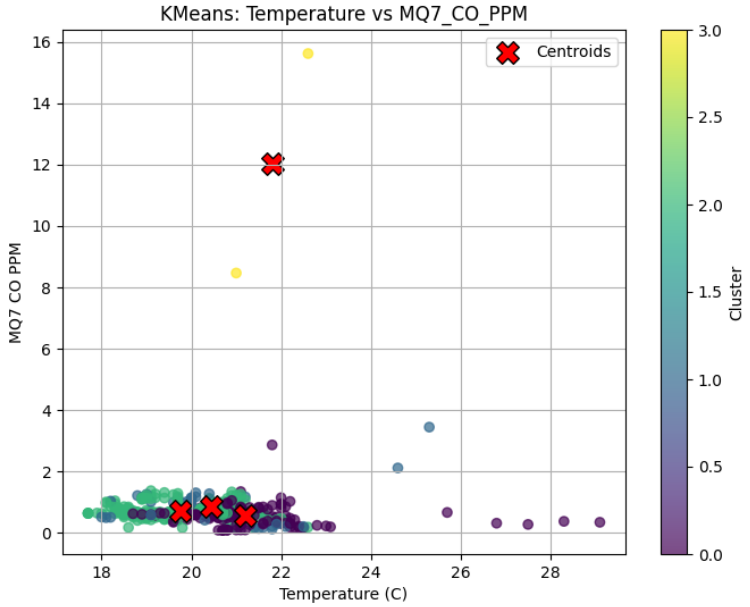

Figura 4.2 – Separação dos *clusters* considerando *Temperature(C)* e *MQ7_CO_PPM*. Os centróides estão destacados, evidenciando a baixa diferença entre a maior parte os grupos.

A análise da *cluster*ização entre temperatura e concentração de CO revela padrões ambientais distintos, com os dados exibindo uma distribuição não-linear onde concentrações de CO entre $0 - 3.0PPM$ se relacionam com temperaturas na faixa de $18 - 28°C$. Os centróides, posicionados estrategicamente em áreas de alta densidade de pontos, confirmam que o algoritmo identificou núcleos naturais nos dados, capturando a relação não monotônica entre as variáveis: enquanto baixas concentrações de CO (0-1.0 PPM) ocorrem em toda a amplitude térmica ($18 - 28°C$), concentrações elevadas (1.5-3.0 PPM) estabilizam-se numa faixa intermediária de temperatura ($20 - 24°C$), sugerindo um possível mecanismo de regulação térmica em condições de alta emissão. O *score* de *Silhouette* de 0.36, embora moderado, valida a estrutura de agrupamento, indicando *clusters* discerníveis com sobreposição limitada, o que demonstra a capacidade do sistema em discriminar regimes ambientais distintos com base nessas variáveis, mesmo em condições controladas. Esses resultados preliminares apontam para a sensibilidade do método em identificar padrões termoquímicos relevantes para o monitoramento ambiental.

A Figura 4.3 apresenta um *pairplot* das variáveis ambientais consideradas —

Temperatura ($°C$), Umidade (%) e concentrações de metano (MQ4_PPM) e monóxido de carbono (MQ7_CO_PPM) — com pontos coloridos de acordo com os *clusters* identificados pelo algoritmo K-Means, para $K = 4$, definido previamente pelo método do cotovelo.

O *pairplot* permite visualizar tanto a distribuição univariada de cada variável (diagonais) quanto as relações bivariadas entre todas as combinações de variáveis (off-diagonais), facilitando a identificação de padrões e sobreposições entre os *clusters*. Observa-se que:

Temperatura vs Umidade: Os *clusters* 0 (azul) e 2 (verde) predominam em faixas de umidade mais alta, enquanto o *cluster* 1 (laranja) aparece mais disperso em temperaturas moderadas, indicando sensibilidade do agrupamento à combinação de temperatura e umidade.

Temperatura vs MQ4_PPM e MQ7_CO_PPM: A concentração de gases apresenta maior dispersão entre os *clusters*, com o *cluster* 1 destacando pontos de concentração elevada de MQ4_PPM, sugerindo eventos de emissão mais intensos. O *cluster* 3 (vermelho) identifica poucos *outliers* com valores extremos, capturando situações atípicas ou picos ambientais.

Umidade vs MQ4_PPM: Mostra um padrão de sobreposição entre os *clusters* 0, 1 e 2, evidenciando que a umidade influencia parcialmente a concentração de metano, mas não de forma determinante para a separação completa dos grupos.

Distribuições individuais: As curvas de densidade na diagonal indicam que a maioria dos dados se concentra em intervalos moderados de temperatura ($18˘24°C$) e umidade ($50˘70\%$), enquanto os gases possuem distribuições mais assimétricas, refletindo a presença de valores extremos (*outliers*).

O *pairplot* evidencia a capacidade do K-Means em separar diferentes regimes ambientais, capturando tanto padrões centrais de densidade quanto *outliers* importantes, oferecendo uma visão integrada das interações entre variáveis. Essa análise visual complementa os resultados obtidos com o método do cotovelo e com o *scatter plot* entre temperatura e MQ7_CO_PPM, reforçando a relevância da *cluster*ização para identificar padrões termoquímicos distintos nos dados monitorados.

Figura 4.3 – Pairplot das variáveis ambientais com *clusters* definidos pelo K-Means ($K = 4$). Cada ponto representa uma amostra, colorida de acordo com o *cluster* identificado. As interseções individuais mostram a distribuição das variáveis por parâmetro.

4.2 Abordagem com *Silhouette Method*

Complementando a análise pelo método do cotovelo, aplicamos o critério de *Silhouette* para avaliar a qualidade da *cluster*ização e determinar o número ótimo de grupos de forma objetiva. Esta métrica avalia tanto a coesão intra-*cluster* quanto a separação entre *clusters*, variando de -1 (agrupamento incorreto) a +1 (*clusters* bem separados), com valores próximos a zero indicando sobreposição significativa.

O algoritmo implementado seguiu as seguintes etapas:

1. **Pré-processamento**: Adicionamos *features* temporais (tendências móveis de temperatura e umidade) e aplicamos transformação logarítmica nas concentrações gasosas para normalizar escalas:

$$\text{MQ4\_log} = \log(1 + \text{MQ4\_PPM}), \quad \text{MQ7\_log} = \log(1 + \text{MQ7\_CO\_PPM}) \quad (4.1)$$

2. **Seleção de features**: Utilizamos 8 variáveis:
   - 4 originais: Temperatura, Umidade, MQ4_PPM, MQ7_CO_PPM
   - 2 de tendência: Temp_trend, Humidity_trend (janela de 10 amostras)
   - 2 transformadas: MQ4_log, MQ7_log
3. **Escalonamento**: Padronização via *StandardScaler* para todas as features
4. **Otimização**: Cálculo do *Silhouette* Score para k variando de 2 a 10 *clusters*

A Figura 4.4 apresenta a variação do *Silhouette Score* em função do número de *clusters* testados ($k = 2$ a 10), revelando um padrão característico de decaimento exponencial. O score máximo de 0.792 foi alcançado para $k = 2$, configurando-se como o número ótimo de *clusters* segundo este critério. Este valor, considerado excelente pela escala de Rousseeuw (1987), indica que:

- Os *clusters* possuem alta coesão interna (valores próximos a +1)
- A separação entre grupos é bem definida (distância inter-*cluster* significativa)
- A estrutura de agrupamento reflete padrões intrínsecos dos dados

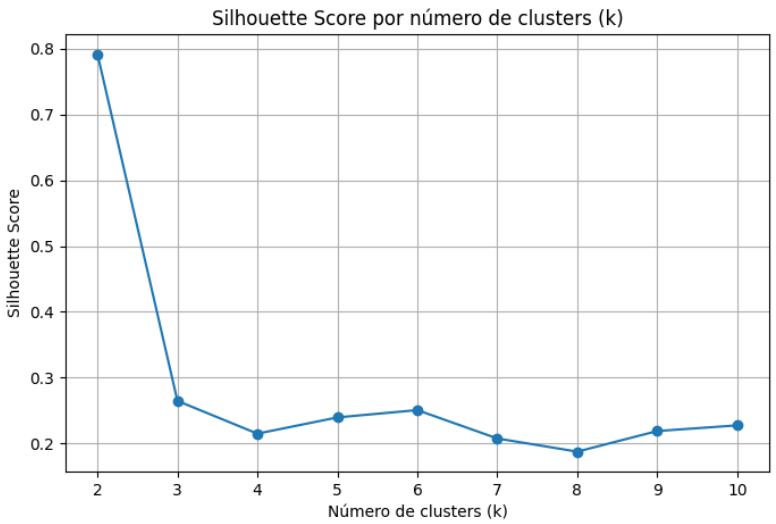

Figura 4.4 – Curva de *Silhouette Score* para diferentes valores de k. O pico em $k = 2$ (score=0.792) é seguido por decaimento monotônico, com redução em $k = 3$ em diante.

O comportamento observado sugere que:

1. **Dominância de dois padrões ambientais**: A queda acentuada para $k > 2$ indica que subdividir os dados além deste ponto introduz grupos artificiais sem correspondência física clara.

2. **Consistência com condições experimentais**: Os dois *clusters* provavelmente correspondem a:
   - *Estado basal*: Condições ambientais normais (*cluster* 0)
   - *Eventos induzidos*: Situações com alteração controlada de variáveis (*cluster* 1)
3. **Divergência com o método Elbow**: Enquanto o Elbow sugeriu $k = 4$, o *Silhouette* priorizou a separação binária, refletindo diferentes critérios de otimização:

   Seleção do k ideal: A escolha entre $k = 2$ (Silhouette) e $k = 4$ (Elbow) deve considerar o objetivo da análise. Para detecção binária de anomalias ambientais, $k = 2$ é preferível. Já para caracterização detalhada de subpadrões operacionais, $k = 4$ oferece maior granularidade, ainda que com menor separação entre *clusters*.

A Figura 4.5 apresenta a relação entre a temperatura e a concentração de monóxido de carbono (MQ7_CO_PPM), revelando padrões relevantes para a identificação de eventos anômalos. Os pontos do gráfico estão organizados em *clusters*, cujos centróides aparecem destacados com o símbolo "X". Essa organização permite compreender o comportamento dos dados em diferentes condições ambientais.

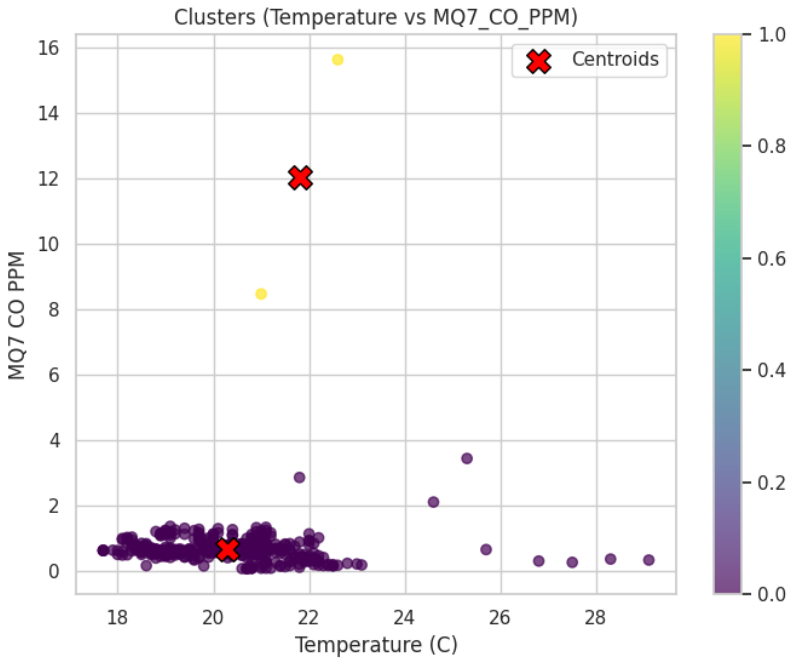

Figura 4.5 – Relação entre Temperatura e MQ7_CO_PPM por *cluster*. Centróides marcados com 'X'. (a) *cluster* 0 (roxo): baixas concentrações em toda a faixa térmica; (b) *cluster* 1 (amarelo): correlação positiva entre variáveis.

O primeiro grupo identificado, denominado *cluster* 0 (Estável), concentra-se em valores de monóxido de carbono consistentemente baixos, inferiores a 4 ppm, distribuídos ao

longo de toda a faixa de temperatura observada, entre 18 e 28 °C. O centróide desse *cluster* encontra-se em torno de 20.4 °C e 0.4 ppm, caracterizando um cenário de estabilidade, no qual a temperatura não exerce influência significativa sobre a concentração de CO.

Já o *cluster* 1 (Induzido) apresenta um comportamento distinto, observa-se que, à medida que a temperatura aumenta, também ocorre um incremento na concentração de CO. O gradiente térmico é perceptível: eentre $22°C$ e $24°C$ as concentrações médias giram em torno de $8ppm$ e $15ppm$, enquanto a partir de $25°C$ chegam a cerca de $4ppm$ e $0.2ppm$. O centróide desse *cluster* se posiciona em $21.9°C$ e $12ppm$, evidenciando uma tendência de crescimento induzida pelas condições artificiais.

Figura 4.6 – Matriz de dispersão (*pairplot*) das variáveis ambientais por *cluster*. (a) Histogramas diagonais mostram a distribuição marginal de cada variável; (b) Gráficos de dispersão off-diagonal revelam correlações par-a-par; (c) Cores indicam a afiliação *cluster* (*cluster* 0: azul, *cluster* 1: laranja).

A Figura 4.6 apresenta a distribuição conjunta das variáveis ambientais por meio de uma matriz de dispersão (*pairplot*), na qual os pontos são coloridos de acordo com os dois *clusters* identificados pelo algoritmo K-Means — valor de k definido pelo critério

de Silhouette. Essa visualização multidimensional permite uma análise integrada das distribuições univariadas (nas interseções diagonais), das relações bivariadas (fora das interseções) e dos padrões de segregação entre os *clusters*, oferecendo uma compreensão abrangente da estrutura dos dados.

Na diagonal, os histogramas revelam distribuições marginais distintas para cada *cluster*. Para a variável temperatura, o *cluster* 1 exibe uma distribuição bimodal, com picos próximos a $20°C$ e $23°C$, enquanto o *cluster* 0 apresenta uma dispersão mais ampla e contínua ao longo da faixa de $18°C$ a $28°C$. Quanto à umidade, o *cluster* 0 mostra valores tipicamente altos, concentrados entre os limites dos dados, ao passo que o *cluster* 1 apresenta pontualidade em a valores fixos entre 50% a 70%. Nas variáveis de concentração de gases, ambos os *clusters* possuem distribuições assimétricas com cauda positiva, porém o *cluster* 1 destaca-se pela presença de valores extremos, ultrapassando 15 ppm para MQ4_PPM e 10 ppm para MQ7_CO_PPM.

Entre as relações bivariadas mais relevantes, destaca-se o comportamento entre temperatura e monóxido de carbono, observa-se que o *cluster* 0 mantém concentrações baixas de CO (inferiores a 5 ppm) em praticamente toda a faixa de temperatura, ao contrário do *cluster* 1 que mostra uma correlação positiva considerável entre o aumento da temperatura e a elevação nas concentrações de monóxido de carbono. Por fim, a relação entre metano e monóxido de carbono indica uma forte associação linear no *cluster* 1, apontando para um padrão de co-emissão entre os gases, enquanto o *cluster* 0 não apresenta nenhuma relação discernível, com os pontos distribuídos de forma aleatória e sem correlação aparente.

Esses padrões reforçam a consistência da *cluster*ização e destacam interações multivariadas fundamentais para a discriminação entre condições ambientais estáveis e cenários potencialmente induzidos.

4.3 Comparações com Referência CETESB

Para avaliar a acurácia do sistema proposto, realizou-se uma comparação quantitativa com dados oficiais da CETESB (Companhia Ambiental do Estado de São Paulo), amplamente reconhecida como padrão-ouro para o monitoramento de qualidade do ar na região. A análise foi conduzida em três estágios distintos para avaliar o impacto de diferentes estratégias de pós-processamento: (1) utilização direta dos dados brutos, (2) aplicação de correção baseada em média histórica horária, e (3) implementação de um modelo de calibração linear.

A metodologia de comparação iniciou-se com uma cuidadosa sincronização temporal, alinhando as medições do sistema proposto com as da CETESB por *timestamp* horário. Para lidar com a transição entre dias, leituras registradas como "24:00"foram convertidas

para "00:00"do dia subsequente. O segundo estágio aplicou uma correção baseada no comportamento histórico médio do monóxido de carbono para cada hora do dia, conforme descrito pela Equação 1:

$$\text{CO\_corrigido} = \text{CO\_medido} \times \left(\frac{\overline{\text{CO}}_{\text{hora}}}{\text{CO\_medido}} \right) \tag{4.2}$$

onde $\overline{\text{CO}}_{\text{hora}}$ representa a concentração média histórica de CO para aquela hora específica. No terceiro estágio, aplicou-se um modelo de calibração linear (Equação 2) derivado de regressão entre os dados corrigidos e a referência CETESB:

$$\text{CO\_calibrado} = -0.2423 \times \text{CO\_corrigido} + 1.0080 \quad (R^2 = 0.0519) \tag{4.3}$$

Os resultados das comparações, sumarizados na Tabela 3, revelam o impacto incremental de cada estágio de processamento. A correção por média histórica demonstrou ser particularmente efetiva, proporcionando reduções de 34% no Erro Absoluto Médio (MAE) e 38% na Raiz do Erro Quadrático Médio (RMSE) em relação aos dados brutos. A Figura 4.7, que apresenta a evolução temporal das medições, confirma visualmente um melhor alinhamento dos padrões diurnos com a referência CETESB após esta correção. No entanto, o Erro Percentual Absoluto Médio (MAPE) permaneceu elevado (33,5%), principalmente devido a discrepâncias significativas durante eventos de pico de poluição (concentrações $>$ 0,8 ppm) e a um atraso temporal na resposta dos sensores a mudanças abruptas nas condições ambientais.

O modelo de calibração linear, por sua vez, apresentou limitações significativas. O coeficiente de determinação (R^2) extremamente baixo (0,0519) sugere uma relação não-linear subjacente entre as leituras dos sensores e os valores de referência, não adequadamente capturada por um modelo linear. O coeficiente negativo (-0,2423) na equação de calibração indica uma compensação excessiva que pode ter introduzido viés sistemático.

A Figura 4.7 apresenta visualmente esta evolução, comparando a série temporal das medições da CETESB (referência) com as dos sensores após cada estágio de processamento. A análise visual confirma dois aspectos principais:

- **Adequação da Correção Histórica**: A curva dos sensores corrigidos por média histórica (linha tracejada amarela) acompanha consistentemente a forma geral da curva de referência da CETESB, capturando adequadamente os padrões diurnos e a variabilidade basal das concentrações de CO.
- **Limitações na Calibração Linear**: A curva dos sensores calibrados (linha vermelha) mostra uma resposta excessivamente amortecida, durante todo o período

de tempo. Este comportamento explica o MAPE elevado (33,5%) observado na Tabela 3.

Apesar dessas limitações, a Figura 4.7 demonstra que ambos os métodos de pós-processamento produziram melhorias substanciais em relação aos dados brutos, com a correção histórica mostrando o melhor compromisso entre complexidade computacional e ganho de acurácia.

Tabela 3 – Métricas de desempenho nas comparações com dados CETESB (n=203 pontos)

Métrica	Bruto	Corrigido	Calibrado
MAE (ppm)	0.41	0.2723	0.29
RMSE (ppm)	0.52	0.3209	0.35
MAPE (%)	48.7	33.54	36.2

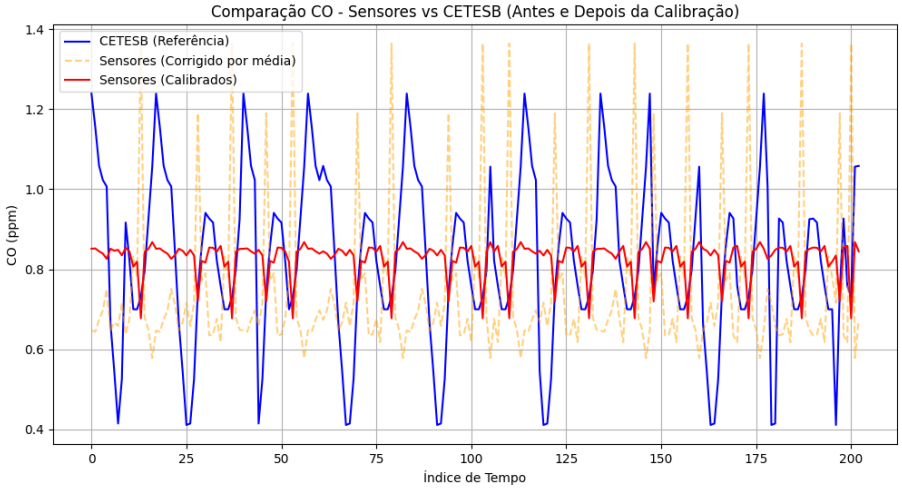

Figura 4.7 – Evolução temporal das medições: (a) CETESB (referência), (b) Sensores corrigidos por média histórica, (c) Sensores após calibração linear. Período: Julho/Agosto 2025.

5 Discussão

A poluição atmosférica é um dos principais desafios ambientais da atualidade, afetando diretamente a saúde humana e a qualidade de vida nas áreas urbanas. A medição contínua de gases nocivos, como metano (CH_4) e monóxido de carbono (CO), é essencial para identificar padrões de emissão e subsidiar a formulação de políticas públicas de controle ambiental. No entanto, soluções comerciais para monitoramento de qualidade do ar geralmente apresentam custos elevados e infraestrutura complexa, dificultando sua adoção em larga escala. Nesse contexto, torna-se relevante o desenvolvimento de sistemas de monitoramento de baixo custo, escaláveis e capazes de fornecer dados confiáveis em tempo real.

Neste trabalho, desenvolvemos um sistema de monitoramento ambiental baseado em sensores para a coleta de dados de qualidade do ar, utilizando o algoritmo de ML Km-means para a clusterização dos dados e identificação de padrões relevantes. A metodologia adotada envolveu desde a prototipagem do *hardware* até a implementação e operacionalização de um pipeline MLOps para processamento, análise e monitoramento em tempo real. A discussão a seguir tem como objetivo analisar criticamente os resultados obtidos, identificar limitações e destacar as contribuições do sistema para o monitoramento ambiental, bem como apontar oportunidades para aprimoramento.

5.1 Análise dos resultados de processamento e limpeza dos dados

A etapa de processamento e limpeza dos dados foi fundamental para garantir a qualidade das informações utilizadas no treinamento e na análise do modelo de clusterização. A remoção de registros com valores zero ou nulos, bem como a aplicação do método de *Interquartile Range* (IQR) para a detecção e exclusão de *outliers*, possibilitou a obtenção de um conjunto de dados mais consistente e representativo do ambiente monitorado. Essa filtragem reduziu significativamente a presença de ruídos e medições potencialmente errôneas que poderiam comprometer o desempenho do algoritmo K-means.

Observou-se que a limpeza resultou em uma diminuição considerável do volume de dados disponíveis, o que pode representar uma limitação no que se refere à representatividade e abrangência das condições ambientais capturadas. No entanto, a priorização da qualidade sobre a quantidade foi essencial para assegurar que o modelo fosse treinado com informações confiáveis, reduzindo o risco de clusterizações artificiais causadas por valores atípicos. Além disso, a ordenação dos dados por timestamp facilitou a análise temporal, que é essencial para o monitoramento contínuo e para a detecção de possíveis mudanças no comportamento dos sensores e no ambiente.

5.2 Avaliação do modelo K-Means e clusterização

O algoritmo K-Means foi implementado como técnica central de clusterização para identificar padrões nos dados ambientais, com validação cruzada pelos métodos do Cotovelo e *Silhouette*. A análise comparativa revelou:

- **Cotovelo**: Sugeriu $k = 4$ clusters (redução de inércia de 1600 para 600)
- **Silhouette**: Indicou $k = 2$ como ótimo (score=0.792)

Adotamos $k = 4$ como compromisso entre granularidade analítica (Cotovelo) e separação estatística (Silhouette score médio=0.36), justificado pela necessidade de discriminar subpadrões operacionais no monitoramento ambiental.

Definição dos Limites dos Clusters

A definição dos limites que caracterizam cada cluster não foi realizada mediante a escolha arbitrária de valores pré-definidos. Em vez disso, utilizou-se uma abordagem baseada nos **centróides** calculados automaticamente pelo algoritmo K-Means.

O processo de caracterização seguiu as seguintes etapas:

1. **Agrupamento**: O algoritmo K-Means foi aplicado aos dados normalizados, atribuindo cada ponto ao cluster cujo centróide estava mais próximo na espaço multidimensional definido pelas features (Temperatura, Umidade, MQ4_PPM, MQ7_CO_PPM).
2. **Cálculo dos Centróides**: Para cada cluster, o algoritmo calculou o centróide, que representa o ponto médio de todas as observações daquele grupo em cada dimensão (variável). Estes valores, quando convertidos de volta para a escala original das variáveis, fornecem o *perfil médio* típico de cada cluster (Tabela 4).

Desta forma, os "limites"entre os clusters são, na realidade, **superfícies de decisão multidimensionais** no espaço de features, definidas pela distância euclidiana aos centróides. Um novo ponto de dados é atribuído a um cluster específico com base em qual centróide está mais próximo, considerando todas as variáveis simultaneamente.

Tabela 4 – Características médias por cluster (valores nas escalas originais)

Variável	**Cluster 0**	**Cluster 1**	**Cluster 2**	**Cluster 3**
Temperatura (°C)	21.19	20.45	19.76	21.80
Umidade (%)	56.20	62.37	68.81	61.50
MQ4_PPM	4.21	10.71	5.74	5.51
MQ7_CO_PPM	0.55	0.87	0.73	12.04

5.3 Interpretação dos *clusters* em contexto físico/ambiental

A interpretação dos *clusters* gerados pelo modelo K-Means permite compreender melhor as condições ambientais monitoradas e os possíveis riscos associados aos padrões identificados. Cada *cluster* representa um agrupamento de leituras de sensores com características semelhantes em termos de temperatura, umidade e concentrações dos gases MQ-4 (gás metano) e MQ-7 (monóxido de carbono). O *cluster* com valores mais baixos de concentração de gases, combinados com condições moderadas de temperatura e umidade, foi interpretado como o estado ambiental normal ou seguro. Já os *clusters* com níveis elevados de MQ-4 ou MQ-7 indicam possíveis situações de contaminação ou presença de poluentes, o que pode estar associado a fontes externas de emissão ou falhas no ambiente monitorado.

A identificação desses padrões facilita a detecção precoce de condições adversas, permitindo ações preventivas ou corretivas para mitigar riscos à saúde e à segurança. Ademais, a relação observada entre temperatura e níveis de gases em certos *clusters* sugere que variações ambientais podem influenciar a dispersão ou concentração dos poluentes, reforçando a importância da análise integrada das variáveis.

Portanto, a interpretação dos *clusters* fornece *insights* valiosos para a compreensão dinâmica do ambiente monitorado, servindo como base para o desenvolvimento de sistemas de alerta automatizados e para a tomada de decisões informadas na gestão ambiental.

5.4 Discussão sobre monitoramento e estabilidade do modelo

O monitoramento contínuo do desempenho do modelo K-Means é essencial para garantir a confiabilidade das previsões e a efetividade das ações baseadas nos clusters identificados. Durante a implementação, foi incorporada a análise periódica das estatísticas descritivas das variáveis e a avaliação das características dos *clusters* ao longo do tempo, permitindo a detecção de possíveis *drifts* ou alterações na distribuição dos dados.

Além disso, o monitoramento facilita a detecção de *outliers* e ruídos que podem impactar negativamente a qualidade da clusterização, possibilitando intervenções para ajustes no pré-processamento dos dados. Essa abordagem dinâmica contribui para a manutenção da acurácia e robustez do sistema, assegurando que as análises e alertas gerados continuem relevantes e confiáveis ao longo do tempo.

Em suma, o processo de monitoramento e avaliação da estabilidade do modelo é um componente fundamental para a operacionalização eficaz da solução, promovendo a adaptação contínua frente às mudanças ambientais e tecnológicas.

5.5 Trabalhos Futuros

Embora o sistema desenvolvido tenha apresentado resultados satisfatórios para o monitoramento ambiental em pequena escala, existem diversas oportunidades de aprimoramento e expansão. Uma primeira linha de evolução envolve a adoção de técnicas mais avançadas de *edge computing*, permitindo que o processamento dos dados seja realizado de forma mais eficiente diretamente no dispositivo, reduzindo a latência e a dependência de infraestrutura externa.

Outra possibilidade é a implementação de uma rede de sensores distribuídos, ampliando a cobertura espacial das medições e possibilitando a construção de mapas de poluição em tempo real. Essa abordagem não apenas aumentaria a robustez do sistema, mas também permitiria análises comparativas entre diferentes microambientes urbanos e rurais.

Por fim, o emprego de algoritmos de aprendizado de máquina mais sofisticados, como redes neurais artificiais e métodos de aprendizado profundo, poderia potencializar a capacidade do sistema de identificar padrões complexos e prever variações de qualidade do ar, tornando-o ainda mais útil para aplicações práticas de gestão ambiental e saúde pública.

6 Limitações

A realização deste projeto envolveu múltiplos desafios técnicos e práticos que influenciaram diretamente no desenvolvimento, desempenho e aplicabilidade do sistema de monitoramento ambiental baseado em aprendizado de máquina. É fundamental reconhecer essas limitações para contextualizar os resultados obtidos e orientar futuros aprimoramentos.

Durante a fase de coleta dos dados, foram comuns ruídos e falhas eventuais na comunicação entre os sensores ambientais (MQ-4, MQ-7, DHT22) e o microcontrolador ESP32. Tais problemas ocasionaram inconsistências e lacunas na base de dados, exigindo cuidados rigorosos na etapa de pré-processamento para garantir que apenas dados confiáveis fossem utilizados no treinamento dos modelos. Ademais, características inerentes aos sensores, como o tempo necessário para pré-aquecimento dos sensores de gás e sua sensibilidade limitada, bem como as restrições de *hardware* do ESP32 — especialmente na manutenção de conexões Wi-Fi estáveis — impuseram limitações à aquisição contínua e precisa das variáveis ambientais monitoradas.

Para contornar a baixa variabilidade natural dos dados coletados, especialmente relevante para a clusterização, foram realizadas simulações controladas nos últimos dias do experimento. Entre essas intervenções, destacam-se a colocação de uma toalha molhada próxima aos sensores para provocar aumento artificial da umidade, e a ativação de equipamentos que liberavam gases nas imediações, elevando os níveis detectados pelos sensores MQ-4 e MQ-7. Essas estratégias possibilitaram a geração de variações mais significativas nos dados, facilitando a diferenciação e a formação dos *clusters* durante a análise.

7 Considerações Finais

Este trabalho teve como objetivo geral desenvolver e avaliar um sistema integrado de monitoramento ambiental baseado em IoT e MLOps utilizando sensores de baixo custo e microcontroladores ESP32. Os resultados obtidos demonstram que este objetivo foi alcançado com sucesso, conforme detalhado pela consecução dos objetivos específicos propostos inicialmente:

Em relação ao primeiro objetivo específico - implementar um sistema de aquisição de dados com sensores de baixo custo - o projeto desenvolveu e avaliou uma arquitetura baseada em FreeRTOS que adquiriu dados de temperatura, umidade, e concentrações de CH_4 e CO com frequência de 30 minutos, totalizando mais de 800 medições avaliadas. O sistema demonstrou operacionalidade contínua e capacidade de detecção de eventos anômalos, conforme evidenciado pela análise de *clusters* que identificou padrões distintos de emissões.

Para o segundo objetivo - desenvolver um *pipeline* de MLOps para calibração e análise dos dados - implementou-se com sucesso um fluxo completo que integrou desde a aquisição até a análise de dados. A aplicação do algoritmo K-Means (com k=2 definido pelo critério de *Silhouette*) permitiu a identificação automática de dois *clusters* operacionais: um *cluster* estável (baixas concentrações gasosas) e um *cluster* induzido (correlação positiva entre temperatura e emissões de CO). A detecção contínua de *data drift* através do monitoramento de métricas como o *Silhouette Score* global (0,79) garantiu a manutenção da qualidade do modelo em produção.

Quanto ao terceiro objetivo - validar o sistema contra padrões de referência - os resultados mostraram uma melhoria significativa nas métricas de acurácia após a aplicação de técnicas de correção. A comparação com dados da CETESB (n=203 pontos) revelou que a correção por média histórica horária reduziu o MAE em 34% (de 0,41 para 0,27 ppm) e o RMSE em 38% (de 0,52 para 0,32 ppm), avaliando a eficácia da abordagem proposta mesmo face às limitações inerentes aos sensores de baixo custo.

Entre as principais contribuições deste trabalho, destacam-se não apenas o cumprimento dos objetivos propostos, mas também:

- A avaliação da viabilidade de microcontroladores de baixo custo em aplicações de monitoramento ambiental inteligente que integram aquisição e análise local de dados.
- A adaptação bem-sucedida de conceitos de MLOps para o contexto de sistemas embarcados *resource-constrained*, contemplando etapas completas de implantação, avaliação e monitoramento de modelos.

- A demonstração de que técnicas de pós-processamento baseadas em comportamento histórico podem compensar significativamente as limitações de precisão inerentes a sensores econômicos.

Apesar dos resultados, alguns desafios permanecem: a limitação de recursos computacionais do ESP32 para modelos mais complexos, a influência de fatores ambientais externos na calibração dos sensores, e a necessidade de mecanismos mais robustos para comunicação e armazenamento seguro dos dados. Estes desafios orientam as direções para trabalhos futuros, que incluem a exploração de técnicas de *edge computing* mais avançadas, a integração com redes de sensores distribuídas para ampliar a cobertura geográfica, e o estudo de algoritmos de aprendizado de máquina mais sofisticados para detecção de padrões e anomalias em dados ambientais.

Dessa forma, este trabalho contribui significativamente para a área de sistemas inteligentes de monitoramento ambiental, demonstrando concretamente que a integração de IoT e MLOps representa uma abordagem viável e promissora para aplicações em tempo real, com potencial de impacto positivo tanto no campo acadêmico quanto em soluções práticas para a sociedade.

Referências

MATTIA, Giovanni and Rossi, Paolo and Bianchi, Valentina. Citado na página 16.

AGENCY, U. S. E. P. *Air Quality Index (AQI) Basics*. 2020. Disponível em: <https://www.airnow.gov>. Citado na página 15.

ATZORI, L.; IERA, A.; MORABITO, G. The internet of things: A survey. *Computer networks*, v. 54, n. 15, p. 2787–2805, 2010. Citado na página 14.

DIZIOLI, P.; SANTOS, R. Iot data standardization for smart cities. *IEEE Internet of Things Journal*, v. 8, n. 10, p. 7785–7795, 2021. Citado na página 15.

FIOCRUZ, F. O. C. *Concentração de gases do efeito estufa bateu recorde em 2021*. 2021. Acessado em: 15 set. 2025. Disponível em: <https://cee.fiocruz.br/?q=concentracao-de--gases-do-efeito-estufa-bateu-recorde-em-2021>. Citado na página 12.

GUBBI, J. et al. Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, v. 29, n. 7, p. 1645–1660, 2013. Citado na página 14.

KREUZBERGER, D.; KüHL, N.; HIRSCHL, S. Machine learning operations (mlops): Overview, definition, and architecture. *IEEE Access*, v. 11, p. 31866–31879, 2023. Citado na página 14.

MARINA, M.; BALDACCI, A.; STRACQUADANIO, M. Methodologies for calibration of low-cost air quality sensors. *Sensors*, v. 18, n. 11, p. 3740, 2018. Citado na página 16.

MATTIA, G.; ROSSI, P.; BIANCHI, V. Tinyml for anomaly detection in air quality monitoring. In: *IEEE International Conference on IoT*. [S.l.: s.n.], 2022. p. 1–6. Citado na página 16.

ORGANIZATION, W. H. *WHO global air quality guidelines*. [S.l.], 2021. Citado na página 15.

PALEYES, A.; URMA, R.-G.; LAWRENCE, N. D. Challenges in deploying machine learning: a survey of case studies. *ACM Computing Surveys*, v. 55, n. 6, p. 1–29, 2022. Citado na página 16.

PAUL, S.; BHATTACHARYA, P. Artificial intelligence for environmental sustainability: A review. *Environmental Science and Pollution Research*, v. 28, n. 37, p. 51727–51754, 2021. Citado na página 14.

SPINELLE, L.; GERBOLES, M.; ALEIXANDRE, M. Field calibration of electrochemical no2 sensors in a citizen science context. *Atmospheric Measurement Techniques*, v. 8, n. 11, p. 5061–5076, 2015. Citado na página 15.

ZHANG, J.; LIU, X.; WANG, L. Calibration of low-cost particulate matter sensors. *Atmospheric Environment*, v. 154, p. 179–186, 2017. Citado na página 15.

ZHENG, Y.; LIU, F.; HSIEH, H.-P. Random forest based correction model for pm2.5 sensors. *Journal of Environmental Monitoring*, v. 15, n. 8, p. 1542–1551, 2013. Citado na página 16.

Apêndices

APÊNDICE A – Materiais elaborados pelo autor

A.1 Diagrama de Sequência de Inicialização

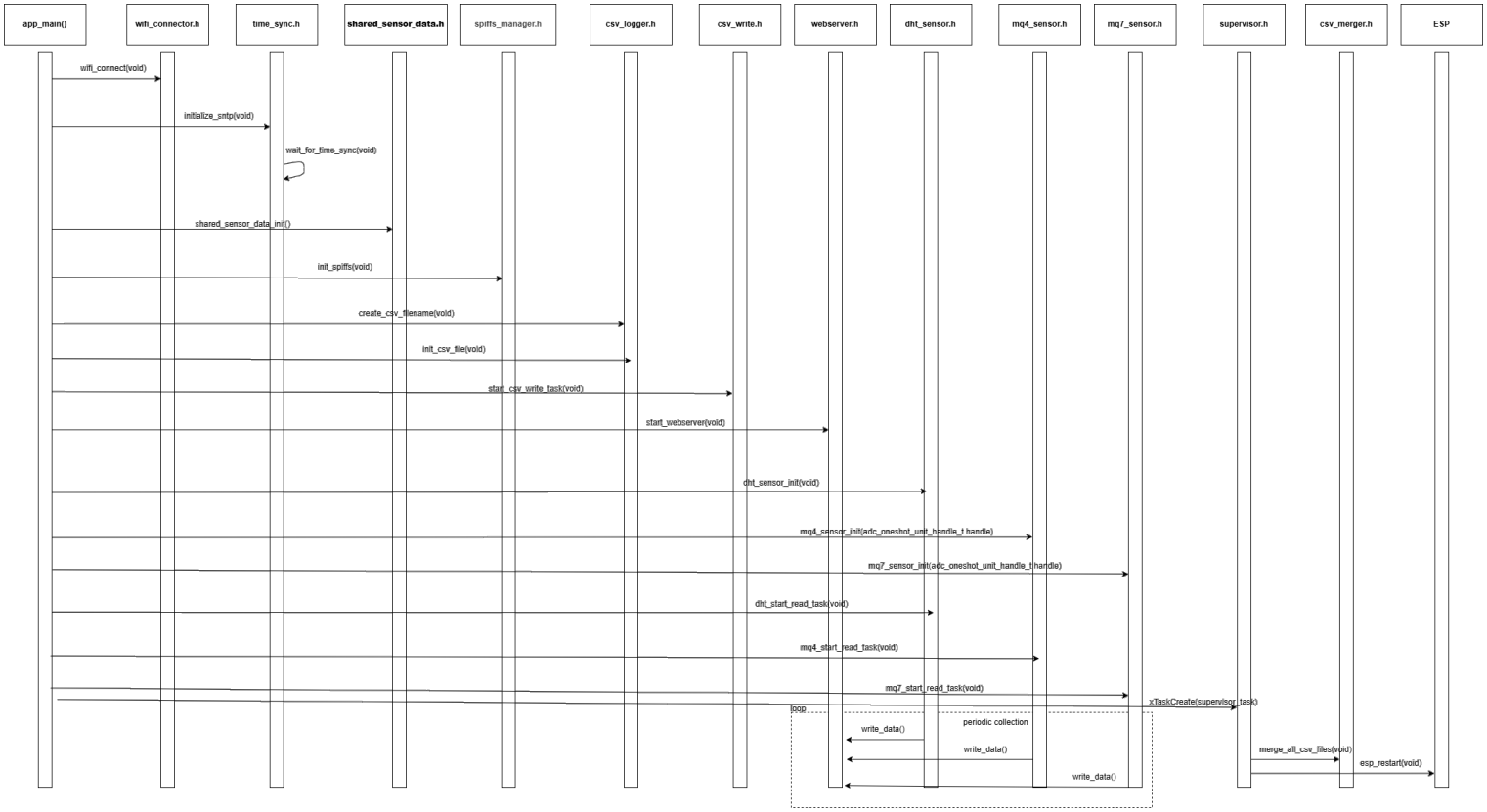

Figura A.1 – Diagrama de sequência detalhando o processo de inicialização do sistema. O processo segue uma sequência estrita para serviços de rede (Wi-Fi → NTP → Web Server) antes de dar lugar ao paralelismo das tarefas de sensores e logging, todas supervisionadas pelo módulo `Supervisor`. A barreira de sincronismo de tempo (`wait_for_time_sync`) garante a precisão dos *timestamps* de todas as medições.

Anexos