

INSTITUTO FEDERAL GOIANO - CAMPUS MORRINHOS
CURSO SUPERIOR DE BACHARELADO EM CIÊNCIA DA
COMPUTAÇÃO

ATHOS FERREIRA DUARTE

ARQUITETURA IOT PARA MONITORAMENTO DE CULTIVOS
AGRÍCOLAS: OTIMIZAÇÃO DE TRANSMISSÃO DE DADOS
EM SISTEMAS EMBARCADOS DE BAIXO CUSTO

MORRINHOS - GO

2026

ATHOS FERREIRA DUARTE

**ARQUITETURA IOT PARA MONITORAMENTO DE CULTIVOS
AGRÍCOLAS: OTIMIZAÇÃO DE TRANSMISSÃO DE DADOS
EM SISTEMAS EMBARCADOS DE BAIXO CUSTO**

Monografia apresentada ao Curso Superior de Bacharelado em Ciência da Computação do Instituto Federal Goiano – Campus Morrinhos, como requisito parcial para obtenção de título de Bacharel em Ciência da Computação.

Área de concentração: Ciência da Computação

Orientador: Prof. Dr. Alexandre Carvalho Silva

MORRINHOS - GO

2026

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema Integrado de Bibliotecas – SIBI/IF Goiano Campus Morrinhos

D812a Duarte, Athos Ferreira.

Arquitetura Iot para Monitoramento de cultivos agrícolas: otimização de transmissão de dados em sistemas embarcados de baixo custo. / Athos Ferreira Duarte. – Morrinhos, GO: IF Goiano, 2026.

75 f. : il. color.

Orientador: Dr. Alexandre Carvalho Silva.

Trabalho de conclusão de curso (graduação) – Instituto Federal Goiano Campus Morrinhos, Bacharelado em Ciências da Computação, 2026.

1. Internet das coisas. 2. Sistemas embarcados (Computadores). 3. Agricultura de precisão. I. Silva, Alexandre Carvalho. II. Instituto Federal Goiano. III. Título.

CDU 004.738

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO

PARA DISPONIBILIZAR PRODUÇÕES TÉCNICO-CIENTÍFICAS

NO REPOSITÓRIO INSTITUCIONAL DO IF GOIANO

Com base no disposto na Lei Federal nº 9.610, de 19 de fevereiro de 1998, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia Goiano a disponibilizar gratuitamente o documento em formato digital no Repositório Institucional do IF Goiano (RIIF Goiano), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IF Goiano.

IDENTIFICAÇÃO DA PRODUÇÃO TÉCNICO-CIENTÍFICA

Tese (doutorado)

Dissertação (mestrado)

Monografia (especialização)

TCC (graduação)

Artigo científico

Capítulo de livro

Livro

Trabalho apresentado em evento

Produto técnico e educacional - Tipo:

Nome completo do autor:

Matrícula:

Título do trabalho:

RESTRIÇÕES DE ACESSO AO DOCUMENTO

Documento confidencial: Não Sim, justifique:

Informe a data que poderá ser disponibilizado no RIIF Goiano: / /

O documento está sujeito a registro de patente? Sim Não

O documento pode vir a ser publicado como livro? Sim Não

DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA

O(a) referido(a) autor(a) declara:

- Que o documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;
- Que obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autoria, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia Goiano os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;
- Que cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia Goiano.

Local

/ /
Data

Assinatura do autor e/ou detentor dos direitos autorais

Ciente e de acordo:

Assinatura do(a) orientador(a)



SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA GOIANO

Ata nº 39/2026 - CCEG-MO/CEG-MO/DE-MO/CMPMHOS/IFGOIANO

ATA DE DEFESA DE TRABALHO DE CURSO

ATA DE DEFESA DA BANCA DE EXAME DE TRABALHO DE CURSO POR VIDEOCONFERÊNCIA

Aos 20 dias do mês de **agosto** de **2026**, às 15 horas foi realizada a Banca de Exame, em formato remoto para a apresentação pública e defesa do trabalho de curso do discente **Athos Ferreira Duarte** intitulado como: **ARQUITETURA IOT PARA MONITORAMENTO DE CULTIVOS AGRÍCOLAS: OTIMIZAÇÃO DE TRANSMISSÃO DE DADOS EM SISTEMAS EMBARCADOS DE BAIXO CUSTO**, como requisito necessário para a conclusão do curso Bacharelado em Ciência da Computação - IFGoiano campus Morrinhos.

A Banca de Exame foi constituída pelos membros: **Prof. Dr. Alexandre Carvalho Silva (orientador)**, **Prof. Dr. Reidner Santos Cavalcante** e **Prof. Dra. Clarice Aparecida Megguer**. Após a análise, emitiram o seguinte resultado:

1 - (☒) **Aprovado**

2 - (☐) **Aprovado com ressalva**

(A Banca Examinadora deve definir as exigências a serem cumpridas pelo aluno na revisão, ficando o orientador responsável pela verificação do cumprimento das mesmas.)

Observações: _____

3 - (☐) **Reprovado com o seguinte parecer:** _____

Morrinhos - GO, 20 de agosto de 2026.

Por ser verdade firmamos a presente:

(Assinado Eletronicamente)

Prof. Dr. Alexandre Carvalho Silva

Orientador(a)

(Assinado Eletronicamente)

Prof. Dr. Reidner Santos Cavalcante

Membro Interno

(Assinado Eletronicamente)

Prof. Dra. Clarice Aparecida Megger

Membro

(Assinado Eletronicamente)

Documento assinado eletronicamente por:

- **Alexandre Carvalho Silva**, PROFESSOR ENS BASICO TECN TECNOLOGICO , em 20/08/2026 16:46:18.
- **Clarice Aparecida Megguer**, PROFESSOR ENS BASICO TECN TECNOLOGICO , em 20/08/2026 16:48:39.
- **Reidner Santos Cavalcante**, PROFESSOR ENS BASICO TECN TECNOLOGICO , em 20/08/2026 17:04:54.

Este documento foi emitido pelo SUAP em 20/08/2026. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifgoiano.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código Verificador: 850686

Código de Autenticação: 726f87d1ed



INSTITUTO FEDERAL GOIANO

Campus Morrinhos

Rodovia BR-153, Km 633, Zona Rural, SN, Zona Rural, MORRINHOS / GO, CEP 75650-000

(64) 3413-7900

AGRADECIMENTOS

Durante o processo de desenvolvimento das aplicações propostas e escrita desta monografia obtive ajuda de diversas pessoas, dentre as quais posso destacar:

A professora Ana Maria Martins Carvalho que durante suas disciplinas ministrou conteúdos relevantes, que me forneceram as habilidades necessárias para desenvolvimento deste trabalho.

Aos professores do curso de Bacharelado em Ciência da Computação, pelos ensinamentos compartilhados ao longo desses anos, que contribuíram direta ou indiretamente para minha formação acadêmica e para a realização deste trabalho.

Ao meu amigo Vitor Henrique Silva de Sá por me auxiliar no planejamento deste trabalho, e em vários aspectos técnicos sempre que eu precisei.

Aos meus amigos João Vitor Guidi de Lima e Amadeus Ferro por sempre compartilharem dos momentos mais importantes para mim durante esses 4 anos de trabalho e aperfeiçoamento.

Aos meus companheiros da turma que foram essenciais para minha permanência no curso.

Aos meus pais que foram minha base de apoio durante esses 4 anos de formação.

E finalmente ao Professor e Orientador Alexandre Carvalho por fazer parte de todas as etapas do desenvolvimento desta pesquisa desde a concepção até os ajustes finais.

RESUMO

Este trabalho aborda o desenvolvimento de um sistema baseado em Internet das Coisas (IoT) para o monitoramento contínuo de variáveis ambientais, com foco em aplicações para ambientes controlados, como estufas agrícolas, considerando restrições de custo e limitações de hardware. Parte-se do problema relacionado à dificuldade de implementar soluções acessíveis que integrem sensoriamento, comunicação e processamento de dados de forma eficiente em contextos com recursos limitados. O objetivo consiste em desenvolver e avaliar uma arquitetura funcional capaz de coletar, transmitir e disponibilizar dados ambientais de maneira contínua. Metodologicamente, a pesquisa é caracterizada como aplicada, com abordagem quali-quantitativa, de natureza exploratória e descritiva, utilizando procedimentos experimentais a partir da construção de um protótipo funcional. A solução desenvolvida integra sensores analógicos, um gateway baseado em Raspberry Pi, comunicação via protocolo MQTT e uma infraestrutura em nuvem para armazenamento e visualização dos dados. Os resultados indicam que o sistema foi capaz de manter fluxo contínuo de informações, com estabilidade operacional e baixo consumo de recursos, além de evidenciar ganhos significativos na eficiência da transmissão de dados após otimizações implementadas. Conclui-se que a arquitetura proposta se mostra viável em contextos controlados, embora apresente limitações relacionadas à escalabilidade, à precisão dos sensores e à necessidade de ajustes para aplicações em maior escala.

Palavras-chave: Internet das Coisas; Agricultura de Precisão; Otimizações de Rede; Sensoriamento Remoto; Telemetria.

ABSTRACT

This study addresses the development of an Internet of Things (IoT)-based system for continuous monitoring of environmental variables, focusing on applications for controlled environments, such as agricultural greenhouses, considering cost constraints and hardware limitations. The research is motivated by the challenge of implementing accessible solutions capable of integrating sensing, communication, and data processing efficiently in resource-constrained environments. The main objective is to develop and evaluate a functional architecture capable of collecting, transmitting, and providing environmental data continuously. Methodologically, the study is classified as applied research, with a qualitative-quantitative approach, exploratory and descriptive objectives, and experimental procedures based on the construction of a functional prototype. The proposed solution integrates analog sensors, a Raspberry Pi-based gateway, MQTT communication protocol, and a cloud infrastructure for data storage and visualization. The results indicate that the system was able to maintain a continuous data flow with operational stability and low resource consumption, as well as significant improvements in transmission efficiency after optimization strategies were implemented. It is concluded that the proposed architecture is viable in controlled environments, although it presents limitations related to scalability, sensor accuracy, and the need for adjustments for large-scale applications.

Keywords: Internet of Things; Precision Agriculture; Network Optimization; Remote Sensing; Telemetry.

FIGURAS

Figura 1 - Arquitetura em camadas para processamento distribuído em sistemas IoT	6
Figura 2 - Arquitetura de comunicação baseada no modelo <i>publish/subscribe</i> do MQTT.	9
Figura 3 - Comparação do tamanho total de mensagens (em MB) entre diferentes formatos de serialização.	12
Figura 4 - Arquitetura geral do sistema IoT proposto	27
Figura 5 - Montagem do protótipo com Raspberry Pi e sensores	29
Figura 6 - Modelo ER: usuários, dispositivos e provisionamento de sensores	35
Figura 7 - Modelo ER: Monitoramento e suas medidas	36
Figura 8 - Fluxo de dados da telemetria no sistema	37
Figura 9 - Fluxo de provisionamento de sensores	39
Figura 10 - Início da execução: Cadastro do Sistema Embarcado	43
Figura 11 - Conexão com o <i>broker</i> , busca de novos sensores e envio de leituras	44
Figura 12 - Tratamento de falha na leitura de sensor	45
Figura 13 - Interface de monitoramento em operação	47
Figura 14 - Tela de registro de dispositivos	48
Figura 15 - Tela de listagem de dispositivos	49
Figura 16 - Seleção do modelo de sensor a registrar	50
Figura 17 - Configuração dos parâmetros da instância do sensor	51
Figura 18 - Comportamento das leituras de umidade do solo	53
Figura 19 - Visualização histórica das leituras monitoradas	55
Figura 20 - Visualização histórica das leituras após aplicação de <i>downsampling</i>	57
Figura 21 - Registro de ocorrências de erro nos sensores	58

TABELAS

Tabela 1 – Comparativo dos trabalhos relacionados frente à abordagem proposta.	23
Tabela 2 – Estimativa de custos dos componentes de hardware do protótipo.	30
Tabela 3 – Consolidado estatístico de uma amostra de 3 minutos de telemetria	54
Tabela 4 – Evolução do tamanho do pacote para envio de 10 leituras	61
Tabela 5 – Consumo de recursos e latência do sistema durante a operação	62

QUADROS

Quadro 1 - Classificação metodológica da pesquisa	26
---	----

SIGLAS

Conversor Analógico-Digital (ADC)
Erro Médio Absoluto (MAE)
Erro Percentual Absoluto Médio (MAPE)
Entidade-Relacionamento (ER)
Hypertext Transfer Protocol (HTTP)
Interface de Programação de Aplicações (API)
Inter-Integrated Circuit (I2C)
Internet das Coisas (IoT)
Internet Protocol (IP)
JavaScript Object Notation (JSON)
Low-Power Wide-Area Network (LPWAN)
Message Queuing Telemetry Transport (MQTT)
Quick Response Code (QR Code)
Redes Neurais Artificiais (ANN)
Representational State Transfer (REST)
Unidade Central de Processamento (CPU)
Wide Area Network (WAN)

SUMÁRIO

1 INTRODUÇÃO	1
2 FUNDAMENTAÇÃO TEÓRICA	4
2.1 INTERNET DAS COISAS	4
2.2 ARQUITETURA DE SISTEMAS IOT	5
2.3 COMUNICAÇÃO EM SISTEMAS IOT	7
2.4 PROTOCOLOS DE COMUNICAÇÃO	8
2.5 FORMATO DE DADOS E OTIMIZAÇÕES DE PACOTE	10
2.6 SISTEMAS EMBARCADOS	13
2.7 SENSORIAMENTO EM IOT	14
2.8 AGRICULTURA DE PRECISÃO E MONITORAMENTO AMBIENTAL	15
2.9 TRABALHOS RELACIONADOS	16
3 METODOLOGIA	25
3.1 CLASSIFICAÇÃO DA PESQUISA	25
3.2 ARQUITETURA DO SISTEMA	27
3.3 COMPONENTES DO SISTEMA	28
3.4 MODELAGEM DO BANCO DE DADOS	33
3.5 DESENVOLVIMENTO E LÓGICA DO SISTEMA	36
3.6 USO DE FERRAMENTAS DE INTELIGÊNCIA ARTIFICIAL	40
4 RESULTADOS E DISCUSSÃO	42
4.1 EXECUÇÃO DO SOFTWARE EMBARCADO	42
4.2 OPERAÇÃO DO ECOSISTEMA IMPLEMENTADO	46
4.3 GERENCIAMENTO DE HARDWARE	48
4.4 PROVISIONAMENTO DINÂMICO DE SENSORES	50
4.5 COMPORTAMENTO DAS VARIÁVEIS MONITORADAS	52
4.6 PERSISTÊNCIA E CONSULTA HISTÓRICA DOS DADOS	55
4.7 OCORRÊNCIA DE FALHAS E LIMITES NO FUNCIONAMENTO	58
4.8 OTIMIZAÇÕES NA TRANSMISSÃO DE DADOS	59
4.9 ESTABILIDADE OPERACIONAL, CONSUMO DE RECURSOS E VIABILIDADE DO SISTEMA	62
5 CONCLUSÃO	65
6 TRABALHOS FUTUROS	67
REFERÊNCIAS	68

1 INTRODUÇÃO

O avanço das tecnologias digitais tem ampliado a capacidade de coleta, transmissão e uso de dados em diferentes áreas, especialmente em aplicações que demandam acompanhamento contínuo de variáveis ambientais. Nesse contexto, a Internet das Coisas (IoT) tem sido discutida como uma extensão das redes de comunicação e da computação ubíqua, ao permitir a integração entre dispositivos físicos e sistemas computacionais capazes de operar de forma distribuída (Atzori; Iera; Morabito, 2010; Gubbi *et al.*, 2013). Essa integração possibilita a construção de sistemas voltados à coleta automatizada de dados, ampliando as possibilidades de monitoramento em diferentes contextos, especialmente em aplicações ambientais e agrícolas que demandam acompanhamento contínuo das condições do ambiente (Choudhary, 2024).

A incorporação de sensores, dispositivos embarcados e protocolos de comunicação em uma mesma arquitetura permite a observação contínua de variáveis ambientais. Diferentemente de métodos baseados em medições pontuais, essa abordagem permite a coleta contínua de dados, o que favorece análises baseadas em séries temporais e reduz a dependência de medições isoladas. Esse tipo de estrutura depende da integração entre componentes com diferentes funções, o que envolve decisões relacionadas à arquitetura do sistema e à forma como os dados são transmitidos e processados.

No contexto do cultivo de olerícolas em ambiente protegido, variáveis como temperatura, umidade do solo e qualidade do ar estão associadas às condições de desenvolvimento vegetal, influenciando processos fisiológicos e o comportamento das culturas. O acompanhamento dessas variáveis, no entanto, nem sempre ocorre de forma contínua ou estruturada, o que pode limitar a compreensão das

dinâmicas ambientais envolvidas nesse tipo de sistema produtivo. Essa limitação está relacionada, em muitos casos, à ausência de sistemas que permitam a coleta e organização dos dados de forma acessível.

A utilização de sistemas baseados em IoT tem sido considerada, na literatura, como uma alternativa para viabilizar o monitoramento contínuo dessas condições. Entretanto, a implementação dessas soluções envolve desafios relacionados à definição da arquitetura, à escolha dos componentes e à seleção dos protocolos de comunicação, aspectos que influenciam diretamente o desempenho do sistema (Ray, 2018; Sookhak *et al.*, 2019). Essas decisões tornam-se ainda mais relevantes quando se consideram restrições de custo e limitações de hardware.

Diante desse cenário, observa-se que, embora existam diferentes propostas de sistemas baseados em IoT, ainda há dificuldades na construção de soluções que conciliem simplicidade de implementação, custo reduzido e funcionamento adequado em contextos com recursos limitados. Essa questão evidencia a necessidade de investigar formas de integrar sensoriamento, comunicação e processamento de dados de maneira consistente, considerando as restrições práticas envolvidas.

A partir dessas considerações, este trabalho busca responder ao seguinte problema de pesquisa: como desenvolver um sistema IoT de baixo custo capaz de monitorar condições ambientais relevantes ao cultivo de olerícolas em estufa, considerando as limitações de hardware e as necessidades de comunicação entre os componentes do sistema?

O objetivo geral deste estudo consiste em desenvolver e avaliar um sistema IoT de baixo custo voltado ao monitoramento de condições ambientais aplicadas às olerícolas cultivadas em ambiente protegido (estufa). Para atender a esse objetivo, foram definidos como objetivos específicos: (i) estruturar uma arquitetura baseada em dispositivos embarcados adequada ao contexto do monitoramento; (ii) selecionar protocolos de comunicação compatíveis com as restrições do sistema; (iii) implementar um sistema funcional para coleta e transmissão de dados ambientais; e (iv) analisar o desempenho do sistema desenvolvido à luz das características discutidas na literatura.

A relevância deste estudo está associada à necessidade de compreender, na prática, a integração entre sensores, dispositivos embarcados e protocolos de comunicação em sistemas de monitoramento ambiental. Ao propor o desenvolvimento de uma solução de baixo custo, o trabalho busca contribuir para a análise de alternativas viáveis em contextos com restrições de recursos, articulando os fundamentos teóricos apresentados com sua aplicação em um cenário específico.

Ao longo do trabalho, são apresentados os fundamentos teóricos que sustentam a proposta, seguidos da descrição dos procedimentos adotados para o desenvolvimento do sistema. Na sequência, são discutidos os resultados obtidos a partir da implementação, relacionando-os com os aspectos abordados na literatura. Por fim, apresentam-se as considerações finais, nas quais se retomam os principais pontos do estudo.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 INTERNET DAS COISAS

A Internet das Coisas (IoT) pode ser compreendida como a interconexão de dispositivos físicos capazes de coletar, processar e transmitir dados por meio de redes de comunicação, ampliando o escopo da internet para além de computadores convencionais. Esse conceito foi estruturado em estudos que discutem a integração entre objetos físicos e sistemas digitais, com destaque para a incorporação de sensores e dispositivos embarcados em diferentes contextos de aplicação (Atzori; Iera; Morabito, 2010; Gubbi *et al.*, 2013).

Esses estudos apontam que a IoT envolve não apenas a conectividade entre dispositivos, mas também a capacidade de gerar e utilizar dados de forma integrada. A evolução desse conceito está associada ao avanço das tecnologias de comunicação, processamento e armazenamento, que viabilizam a implementação de sistemas distribuídos em diferentes domínios (Gubbi *et al.*, 2013). Trabalhos mais recentes ampliam essa discussão ao abordar desafios relacionados à interoperabilidade, integração entre plataformas e à aplicação da IoT em domínios específicos, como agricultura e monitoramento ambiental, evidenciando sua utilização em cenários que envolvem coleta contínua de dados e suporte à tomada de decisão (Choudhary, 2024; Singh *et al.*, 2025).

No contexto deste trabalho, a IoT é considerada como base para a construção de sistemas capazes de coletar e disponibilizar dados ambientais. Essa perspectiva permite analisar sua aplicação no monitoramento de variáveis físicas, com foco em soluções que possam ser implementadas com recursos acessíveis. Dessa forma, a abordagem

adotada busca relacionar os conceitos discutidos na literatura com o desenvolvimento do sistema proposto.

2.2 ARQUITETURA DE SISTEMAS IOT

A arquitetura de sistemas IoT é frequentemente descrita por meio de modelos em camadas, que organizam os componentes de acordo com suas funções no sistema. Esses modelos incluem, de forma geral, níveis relacionados à coleta de dados, comunicação, processamento e aplicação, permitindo uma visão estruturada do fluxo de informações (Ray, 2018). Essa organização contribui para a compreensão das interações entre os diferentes elementos do sistema.

De acordo com Gubbi *et al.* (2013), a divisão em camadas facilita o desenvolvimento e a manutenção de sistemas IoT, ao permitir a separação de responsabilidades entre os componentes. Essa abordagem também favorece a escalabilidade, uma vez que possibilita a adaptação ou substituição de partes do sistema sem comprometer sua estrutura geral. A literatura destaca que essa característica é relevante em cenários que demandam evolução contínua das soluções implementadas.

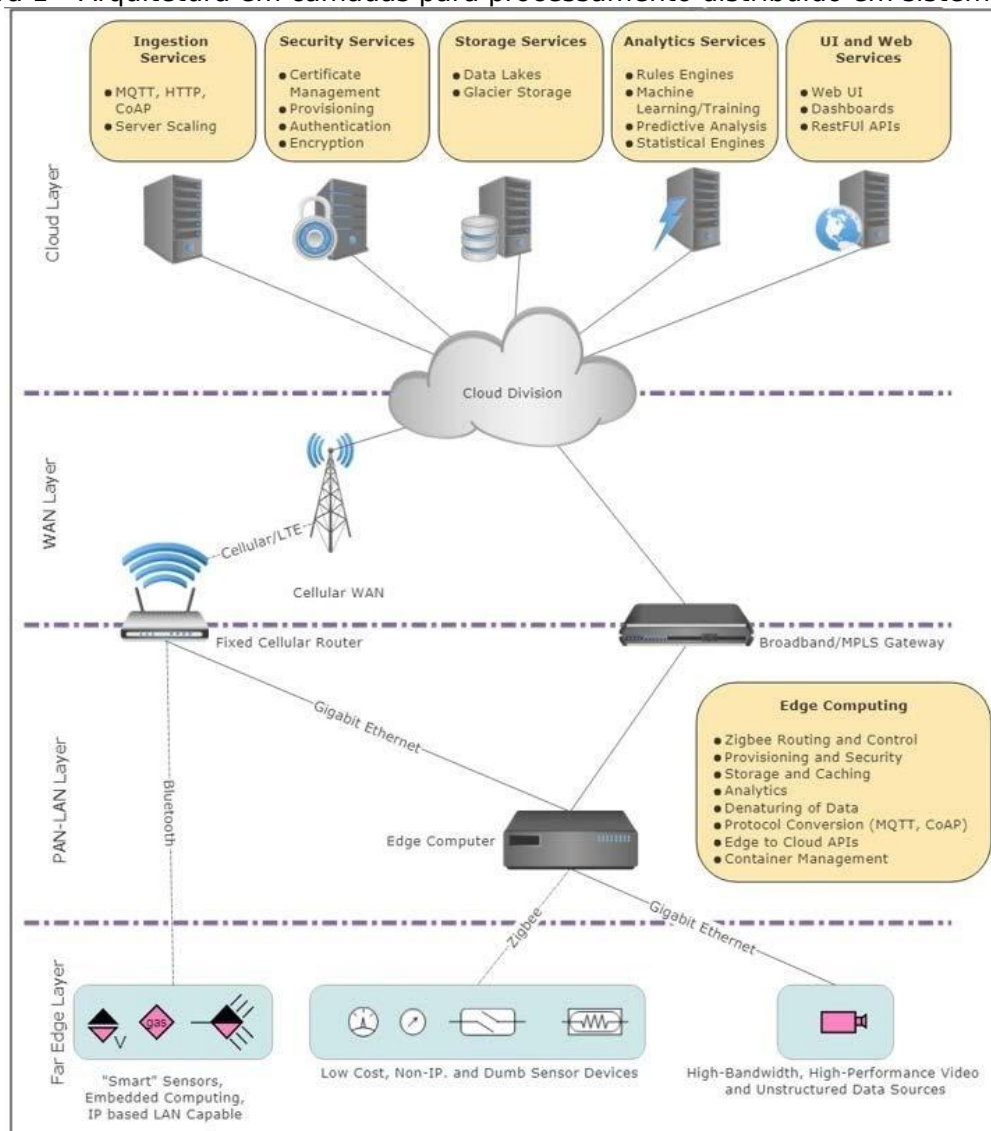
Além dos modelos tradicionais, abordagens mais recentes incorporam mecanismos de processamento distribuído, como *edge* e *fog computing*, que permitem tratar os dados próximos à sua origem. Essa estratégia reduz a dependência de infraestrutura centralizada e pode contribuir para a diminuição da latência na comunicação.

Essa organização em camadas tem sido ampliada em estudos recentes, que propõem arquiteturas mais complexas para lidar com a crescente diversidade de dispositivos e aplicações, incorporando níveis intermediários de processamento e mecanismos de integração entre sistemas distribuídos (Ahoa *et al.*, 2025).

Para ilustrar essa organização, a Figura 1 apresenta uma representação das camadas de processamento em sistemas IoT,

evidenciando a distribuição das funções entre dispositivos de borda, infraestrutura intermediária e serviços em nuvem.

Figura 1 – Arquitetura em camadas para processamento distribuído em sistemas IoT



Fonte: Perry Lea (2020)

A partir dessa representação, observa-se que o processamento pode ser distribuído em diferentes níveis, desde dispositivos mais próximos à coleta dos dados até camadas intermediárias e serviços centralizados. Essa organização amplia as possibilidades de tratamento das informações ao longo do fluxo, permitindo maior flexibilidade na definição da arquitetura do sistema.

No contexto deste trabalho, embora a arquitetura implementada não contemple todas as camadas apresentadas na figura, a utilização de um *gateway* responsável pelo processamento inicial dos dados aproxima-se do conceito de *edge computing*, ao deslocar parte das operações para mais próximo da origem das informações. Essa aproximação contribui para fundamentar as escolhas adotadas na construção da solução proposta.

2.3 COMUNICAÇÃO EM SISTEMAS IOT

A comunicação em sistemas IoT corresponde ao conjunto de mecanismos responsáveis pela troca de dados entre dispositivos, *gateways* e aplicações. Esse processo é essencial para o funcionamento do sistema, uma vez que viabiliza a transmissão das informações coletadas pelos sensores até os níveis em que serão processadas ou analisadas. A forma como essa comunicação ocorre influencia diretamente o desempenho do sistema.

Segundo Ray (2018), os sistemas IoT operam, em muitos casos, com dispositivos que apresentam limitações de processamento, memória e consumo de energia. Essas restrições exigem a utilização de modelos de comunicação garantindo a eficiência na transmissão de dados. Além disso, em aplicações voltadas ao monitoramento ambiental, a comunicação deve considerar a frequência de envio e o volume de dados gerados, uma vez que esses fatores impactam diretamente o consumo energético e a viabilidade do sistema em operação contínua (Faid; Sadik; Sabir, 2020).

Nesse sentido, a comunicação em IoT envolve decisões que vão além da conectividade, incluindo a definição de estratégias adequadas ao tipo de aplicação. A forma como os dados são transmitidos e organizados pode impactar tanto a qualidade das informações quanto a capacidade de resposta do sistema. Essa discussão fundamenta a análise dos protocolos utilizados no desenvolvimento do sistema proposto.

2.4 PROTOCOLOS DE COMUNICAÇÃO

Os protocolos de comunicação definem as regras para a transmissão de dados entre os componentes de um sistema IoT. A literatura destaca que a escolha desses protocolos deve considerar fatores como eficiência, consumo de recursos e tipo de aplicação, especialmente em ambientes com dispositivos limitados (Ray, 2018). Essa decisão influencia diretamente o comportamento do sistema.

O MQTT é um dos protocolos mais utilizados em aplicações IoT, operando sob o modelo *publish/subscribe*, o que permite comunicação assíncrona e reduz a quantidade de dados transmitidos. Cenários de IoT frequentemente exigem a coleta contínua de informações por dispositivos com severas restrições de recursos de *hardware* e energia (Gubbi *et al.*, 2013). Para atender a essa demanda, o MQTT foi projetado como um protocolo de mensageria leve, voltado a redes de alta latência ou conexão instável, priorizando a eficiência no uso de banda e no processamento do dispositivo (Lea, 2020).

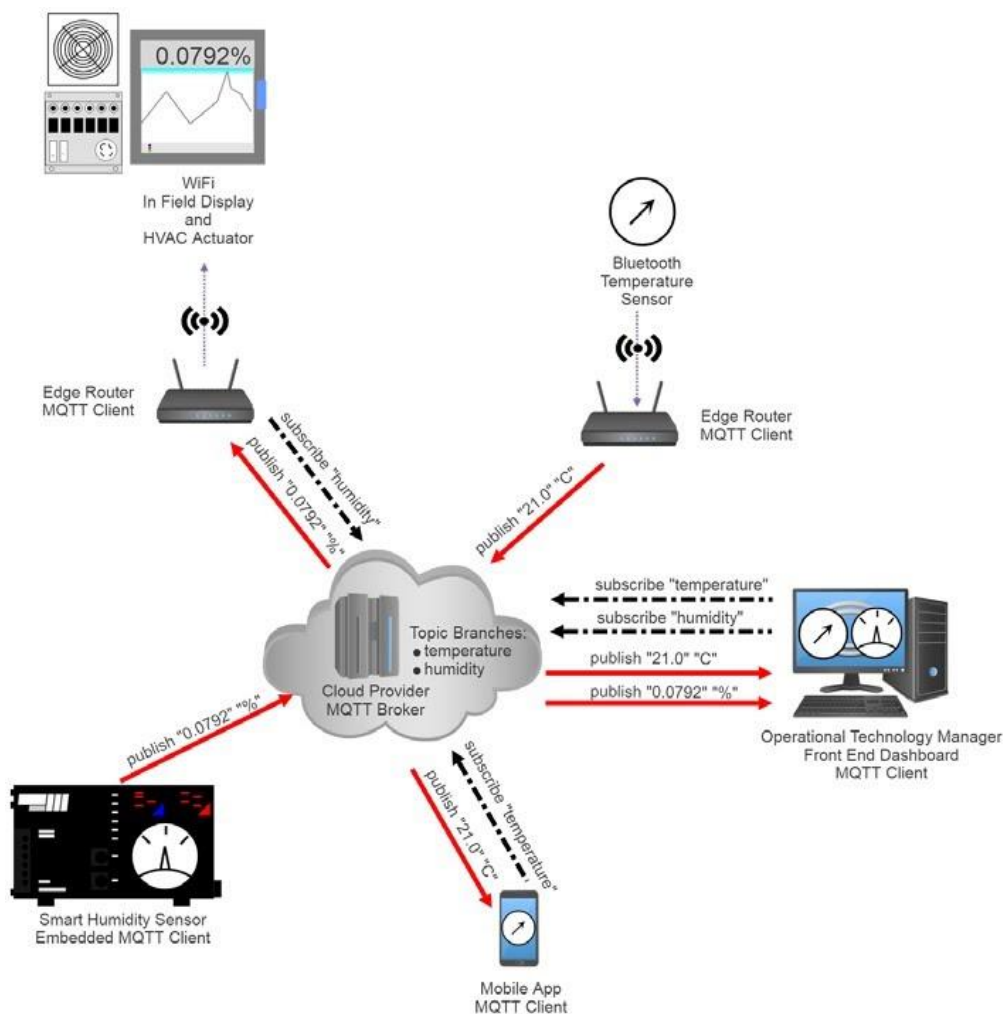
O funcionamento desse modelo depende de um componente central, o *broker*, responsável por conectar os clientes publicadores e assinantes e por rotear as mensagens entre eles. Ao atuar como intermediário, o *broker* dispensa a necessidade de que publicadores e assinantes se identifiquem diretamente por meio de endereços físicos, como o endereço IP, o que favorece a escalabilidade em cenários com grande número de dispositivos conectados (Lea, 2020).

Esse desacoplamento também confere ao modelo uma característica de invariância temporal: uma mensagem publicada por um cliente pode ser lida e processada por um assinante em um momento posterior, sem que ambos precisem estar simultaneamente ativos na rede.

O roteamento realizado pelo *broker* pode ocorrer por diferentes critérios de filtragem, entre eles o assunto da mensagem, associado à estrutura hierárquica dos tópicos, o conteúdo transmitido e o tipo de dado presente no *payload* (Lea, 2020). Essa flexibilidade de filtragem permite

que múltiplos serviços consumam simultaneamente o mesmo fluxo de dados, sem que isso implique conexões diretas entre os dispositivos de origem e cada consumidor final.

Figura 2 - Arquitetura de comunicação baseada no modelo *publish/subscribe* do MQTT.



Fonte: Perry Lea (2020)

O HTTP apresenta uma abordagem mais robusta e amplamente difundida em aplicações *web*, embora envolva maior sobrecarga na comunicação. Sua utilização em sistemas IoT tende a ocorrer em situações em que a integração com serviços externos é necessária e a comunicação não depende de baixa latência. A análise desses protocolos permite compreender as diferenças entre eficiência e robustez, aspecto relevante para a definição da comunicação no sistema desenvolvido.

Em aplicações agrícolas e de monitoramento ambiental, a escolha do protocolo está diretamente associada à necessidade de transmissão contínua de dados com baixo consumo de energia, sendo comum a adoção de estratégias que priorizam eficiência e redução de sobrecarga na comunicação (Faid; Sadik; Sabir, 2020; Singh *et al.*, 2025).

2.5 FORMATO DE DADOS E OTIMIZAÇÕES DE PACOTE

As limitações naturais de recursos nos dispositivos IoT como energia, capacidade de processamento e computação limitadas fazem da otimização uma área teórica primária para a operacionalidade desses sistemas, da qual se apresenta sob diferentes origens, como o tráfego de rede, armazenamento e processamento de dados (Choudhary, 2024).

Os dados trocados entre os módulos de um sistema IoT podem ser descritos com diferentes formatos de serialização, cada um como consequência de características distintas em relação ao tamanho e à legibilidade. O JSON é um formato textual comumente utilizado para representação de dados estruturados, organizados em pares de nome e valor, legíveis para sistemas e programadores (Internet Engineering Task Force, 2017). Essa legibilidade é contrabalançada pela maior extensão dos dados representados, devido ao fato de que a cada mensagem troca-se o nome do campo e o caractere de formatação.

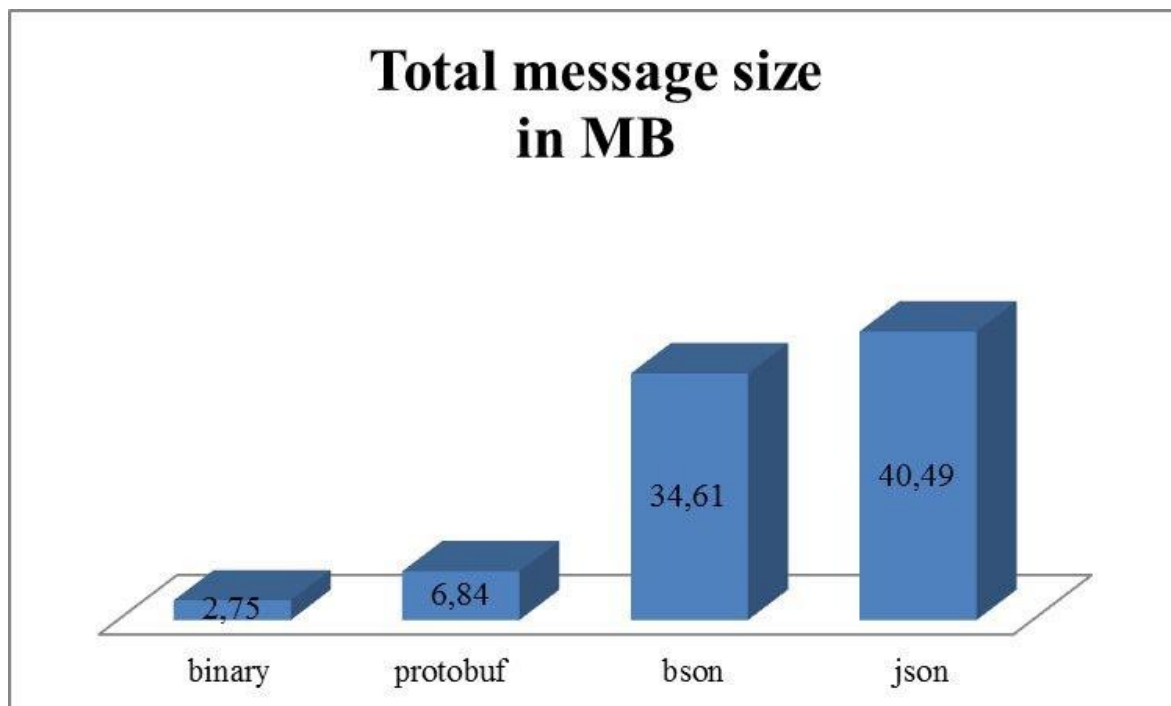
Formas de serialização binária pretendem diminuir essa sobrecarga representando os mesmos dados de maneira mais compacta, abrindo mão da leitura explícita em prol de uma maior eficiência na transmissão.

O Protocol Buffers, criado pelo Google, é um mecanismo de serialização independente de linguagem e plataforma, no qual a estrutura dos dados deve ser definida previamente por meio de um esquema, permitindo que cada mensagem seja transmitida em uma forma reduzida em relação às representações textuais equivalentes (Google, [s.d.]).

Popić *et al.* (2016), ao analisarem o uso do Protocol Buffers em sistemas de comunicação IoT, observaram reduções significativas no volume de tráfego da rede em comparação aos formatos textuais, característica que se torna bastante relevante para dispositivos com limitações de processamento e largura de banda.

O protocolo de comunicação, por sua vez, também contém mecanismos de otimização destinados a minimizar os metadados que são transmitidos. Como mencionado anteriormente, o roteamento de mensagens do MQTT é baseado na estrutura hierárquica dos tópicos do *broker*, que são relevantes para filtragem e entrega das mensagens entre publicadores e assinantes. Esta estrutura, entretanto, pode resultar em identificadores de nomes bastante longos, que se repetem para cada mensagem enviada. Para redução deste custo, versões mais recentes do protocolo têm permitido que nomes de tópicos longos sejam substituídos por identificadores numéricos, a partir da primeira publicação, o denominado Topic Alias, que reduz a repetição de metadados em mensagens posteriores e o *overhead* de transmissão em condições de alta frequência de envio (Lea, 2020).

Figura 3 - Comparação do tamanho total de mensagens (em MB) entre diferentes formatos de serialização.



Fonte: Adaptado de Popić et al. (2016).

Lea (2020), ao analisar os *gateways* de borda, elenca o armazenamento em *cache* de pacotes antes de sua transmissão como um procedimento eficaz em um cenário de comunicação não confiável, permitindo tentativas de reenvio e favorecendo a resiliência da comunicação.

Além da resiliência, o autor propõe a presença de *gateways* agregadores (*aggregating gateways*), responsáveis pela agregação de leituras provenientes de diferentes sensores para um menor número de fluxos enviados ao *broker* MQTT na nuvem. Isso reduz a sobrecarga da comunicação e o número de conexões simultâneas abertas ao servidor, diminuindo a necessidade da frequência de comunicação entre o dispositivo e a infraestrutura do receptor.

O protocolo de comunicação, juntamente com o esquema de serialização, pode ainda ser aperfeiçoado por meio de técnicas voltadas para a codificação mais eficiente da informação temporal das leituras. O *delta encoding* parte do pressuposto de que, em sistemas de

monitoramento contínuo, as oscilações entre leituras consecutivas tendem a ser pequenas, o que torna mais eficiente representar cada marcação de tempo como um *delta* em relação a um referencial do que transmitir o *timestamp* completo a cada amostra (Pelkonen *et al.*, 2015).

Essa eficácia depende da regularidade dos intervalos de amostragem: quanto mais próximos entre si os *deltas* temporais, menor a quantidade de *bits* necessária para representá-los em comparação ao *timestamp* absoluto.

A escolha das técnicas de representação e transmissão de dados é determinante para a viabilidade de sistemas IoT, sobretudo em contextos com restrições de energia e conectividade, como o agrícola, que exige equilíbrio entre granularidade e frequência de envio (Faid; Sadik; Sabir, 2020). Os autores destacam a vida útil do nó sensor como parâmetro crítico, situando a otimização do projeto de *software*, formato, protocolo e agregação de dados entre as principais áreas de interesse para sistemas energeticamente eficientes (Faid; Sadik; Sabir, 2020).

Essas decisões, portanto, não se limitam a aspectos técnicos isolados, mas condicionam a sustentabilidade operacional do sistema como um todo.

2.6 SISTEMAS EMBARCADOS

Os sistemas embarcados constituem elementos centrais em aplicações IoT, sendo responsáveis pela execução de tarefas específicas em dispositivos com recursos computacionais limitados. Esses sistemas integram *hardware* e *software* de forma dedicada, permitindo a coleta, o processamento inicial e a transmissão de dados diretamente no dispositivo. De acordo com Lea (2020), a presença desses componentes possibilita a descentralização do processamento para a borda da rede (*edge computing*), característica recorrente em arquiteturas distribuídas

Plataformas como ESP32 e Raspberry Pi são amplamente utilizadas em projetos de IoT devido à sua capacidade de integração com sensores e módulos de comunicação. A utilização desses dispositivos está associada à execução de tarefas próximas à origem dos dados, o que reduz a dependência de processamento centralizado e contribui para a eficiência da comunicação. Essa abordagem se relaciona com estratégias de *edge computing*, discutidas na literatura como forma de otimizar o fluxo de dados em sistemas distribuídos (Lea, 2020; Singh *et al.*, 2025).

A adoção de sistemas embarcados exige a consideração de aspectos como consumo de energia, capacidade de processamento e estabilidade operacional. Esses fatores influenciam diretamente o comportamento do sistema e sua viabilidade em diferentes contextos de aplicação. No âmbito deste trabalho, esses elementos são considerados na definição da arquitetura proposta, uma vez que impactam a forma como os dados são coletados, processados e transmitidos.

2.7 SENSORIAMENTO EM IOT

O sensoriamento representa a etapa inicial dos sistemas IoT, sendo responsável pela coleta de dados do ambiente por meio de dispositivos específicos. Esse processo é compreendido como uma das camadas fundamentais das arquiteturas IoT, pois constitui o ponto de entrada das informações no sistema e viabiliza a representação de fenômenos físicos em formato digital (Ray, 2018). Sensores realizam a conversão de grandezas físicas em sinais que podem ser processados por sistemas eletrônicos.

A literatura destaca que a qualidade dos dados coletados está diretamente relacionada às características dos sensores utilizados, incluindo precisão, calibração e estabilidade operacional (Fissore *et al.*, 2024). Esses fatores influenciam a confiabilidade das informações e impactam as análises realizadas a partir dos dados obtidos. Em sistemas distribuídos, essas limitações do *hardware* devem ser consideradas em

conjunto com as severas restrições dos dispositivos embarcados e das condições de comunicação (Gubbi *et al.*, 2013).

No contexto deste trabalho, o sensoriamento está voltado à coleta de dados ambientais associados ao cultivo de olerícolas, permitindo o acompanhamento de variáveis como temperatura e umidade. Essa aplicação depende da integração entre sensores e dispositivos embarcados, de modo que as leituras sejam coletadas e encaminhadas ao sistema de forma contínua. A forma como esse processo é estruturado influencia diretamente o funcionamento do sistema como um todo.

2.8 AGRICULTURA DE PRECISÃO E MONITORAMENTO AMBIENTAL

A aplicação de tecnologias digitais no contexto agrícola tem sido associada ao conceito de agricultura de precisão, que busca otimizar o uso de recursos a partir da coleta e análise de dados ambientais. Nesse contexto, a Internet das Coisas tem sido utilizada como uma das principais abordagens para viabilizar o monitoramento contínuo de variáveis como temperatura, umidade e condições do solo, permitindo uma compreensão mais detalhada do ambiente produtivo (Choudhary, 2024).

Estudos voltados à aplicação de IoT na agricultura indicam que esses sistemas têm contribuído para a automatização da coleta de dados e para a melhoria na tomada de decisão em ambientes agrícolas. A integração entre sensores, dispositivos embarcados e plataformas de análise permite o acompanhamento em tempo real das condições ambientais, favorecendo práticas mais eficientes no uso de recursos como água e energia (Singh *et al.*, 2025; Chavare *et al.*, 2025).

Nesse contexto, aplicações como sistemas de irrigação inteligente demonstram que o monitoramento contínuo de variáveis ambientais pode contribuir diretamente para a otimização do uso de água e para a manutenção de condições adequadas ao desenvolvimento das culturas, reduzindo desperdícios e ampliando a eficiência produtiva (Gamal *et al.*, 2023).

No caso específico do cultivo em ambiente controlado, a literatura aponta que a estufa amplifica os benefícios do monitoramento contínuo, pois o controle climático já é parte constitutiva do sistema produtivo. Culturas olerícolas, cultivadas predominantemente sob esse regime, tendem a depender de manutenção de faixas relativamente estreitas de temperatura e umidade para evitar perdas de produtividade e qualidade, o que torna a automação da coleta de dados um componente relevante da gestão desse tipo de produção (Hernández-Morales *et al.*, 2023).

2.9 TRABALHOS RELACIONADOS

Os trabalhos revisados nesta seção ilustram uma tensão recorrente na literatura sobre monitoramento ambiental com sistemas IoT de baixo custo: diferentes propostas endereçam problemas distintos dentro desse escopo, mas poucas os endereçam simultaneamente. Os trabalhos reunidos nesta seção foram selecionados por tocarem, cada um a seu modo, em questões que esta pesquisa também precisa responder, desde a arquitetura do sistema até a forma como os dados circulam entre suas camadas.

Entre os trabalhos revisados, o de Hernández-Morales *et al.* (2023) é o que trata do cenário de aplicação mais próximo ao pretendido por esta pesquisa, o cultivo protegido. Os autores propõem um sistema de monitoramento IoT voltado a cultivos protegidos, organizando a solução em quatro camadas funcionais: percepção, rede, processamento e aplicação, todas orientadas pela preocupação com o custo de implantação.

O ponto que mais chama atenção no trabalho é o tratamento dado à heterogeneidade climática interna à estufa: ao invés de multiplicar sensores para compensar as variações, os autores recorrem à teoria do empacotamento de círculos (*circle packing*) para calcular a distribuição que maximiza a cobertura com o menor número de dispositivos. Pelos

resultados obtidos em uma estufa de 60 m², seis unidades foram suficientes para cobrir 100% do espaço monitorado com alta precisão, havendo ainda a possibilidade de cobrir a mesma área com apenas duas unidades ao se expandir o raio de alcance tolerando um erro ligeiramente maior.

A transmissão dos dados é feita via Sigfox, tecnologia LPWAN adotada pelo alcance e pelo baixo consumo energético. No plano analítico, o trabalho vai além do monitoramento passivo, valendo-se de redes neurais artificiais (ANN) para compor um mecanismo de previsão térmica.

Três camadas e oito neurônios no estrato intermediário permitiram antecipar o comportamento térmico interno com 24 horas de antecedência e erro abaixo de 1,8°C. A validação se estendeu por três meses numa estufa tomateira localizada no México, reunindo mais de 12.500 amostras.

Partir do mesmo orçamento restrito não leva necessariamente à mesma arquitetura, e o contraste entre as duas propostas deixa isso claro. O sistema dos autores opera por meio de redes sem fio de amplo alcance, com comunicação direta entre sensor e estação rádio base via Sigfox, protocolo que, por utilizar espectro não licenciado, impõe restrições técnicas, como o limite comum de 140 mensagens de *uplink* diárias por dispositivo.

Essa limitação se justifica no contexto de uma operação extensiva, onde coleta periódica e antecipação de tendências compensam a baixa cadência de amostragem. A presente pesquisa parte de outra necessidade: acompanhar continuamente variáveis ambientais em contextos operacionais enxutos, nos quais a frequência de amostragem não é apenas uma variável de ajuste, mas uma condição determinante para a qualidade das informações produzidas.

Para dar conta desse fluxo sem depender de redes de amplo alcance, optou-se por um nó de borda encarregado de centralizar a telemetria local. Há, além disso, uma diferença de maturidade analítica entre as duas propostas: enquanto o sistema de Hernández-Morales *et al.* (2023) já demonstra aptidão para antecipar condições futuras, esta

pesquisa se ocupa da etapa anterior a qualquer modelagem preditiva, a saber, assegurar que a coleta esteja estruturada, a transmissão estável e os dados confiáveis o bastante para embasar investigações subsequentes.

Wan *et al.* (2022) abordam um problema pouco explorado nos trabalhos revisados por esta pesquisa. Boa parte dos sistemas de monitoramento existentes cobrem bem os parâmetros de qualidade da água, mas pouco fazem para acompanhar o crescimento das plantas de forma automatizada. O trabalho responde a essa lacuna com um sistema IoT modular que integra computação de borda ao monitoramento de um ambiente produtivo que combina aquicultura e cultivo vegetal em ciclo fechado.

A arquitetura segue o modelo ponta-borda-nuvem. Um microcontrolador STM32 faz a leitura dos parâmetros hídricos e ambientais (oxigênio dissolvido, pH, temperatura e CO₂, entre outros), sem que isso distinga o trabalho do que já existe na literatura. O que diferencia é o que os autores colocam na camada de borda: um Raspberry Pi com câmera e modelo de visão computacional treinado em PyTorch extrai altura e espessura do caule em menos de dez segundos, sem contato físico com o cultivo.

Processar as imagens localmente não é uma escolha arbitrária; é a resposta direta ao problema de banda que elas gerariam se enviadas em formato bruto para a nuvem. Os dados processados seguem via MQTT para armazenamento em MySQL na Ali-cloud e chegam ao usuário por um mini-programa móvel com monitoramento em tempo real, histórico, alarmes e controle de atuadores.

Dois meses de testes no *campus* da China Agricultural University confirmaram quase zero perda de pacotes até 90 m, erro inferior a 5% nas variáveis ambientais e MAE de 0,093 mg/L para oxigênio dissolvido; o módulo de crescimento vegetal registrou MAE de 0,758 cm na altura das plantas, com viés atribuído à calibração da captura de imagens.

O Raspberry Pi e o MQTT aparecem em ambos os trabalhos pela mesma razão prática: atendem bem ao perfil de prototipagem com recursos limitados, e a estruturação em camadas com separação entre

coleta, processamento e armazenamento é terreno comum. O que diverge é a função atribuída à borda: Wan *et al.* (2022) a transformam em unidade de análise visual, enquanto este trabalho a mantém como intermediadora do fluxo sensorial, sem inteligência local.

Essa diferença não é acidental: reflete contextos de aplicação distintos. O sistema de aquaponia foi calibrado para espécies e condições específicas, ao passo que a presente arquitetura deliberadamente evita esse vínculo, o que amplia seu potencial de reuso, mas desloca a análise para outro ponto da cadeia.

A ausência de processamento local reflete, portanto, uma escolha com consequências projetadas para aquele contexto de aplicação, distinta da escolha assumida nesta pesquisa. Em frequências de amostragem elevadas, a telemetria sensorial contínua pode gerar tráfego suficiente para saturar enlaces restritos, cenário para o qual este trabalho investiga a reestruturação das mensagens transmitidas como estratégia de mitigação, tanto no formato quanto no agrupamento e na codificação dos dados.

Chong *et al.* (2023) construíram um sistema IoT para cultivo doméstico de cogumelos que vai além do monitoramento: o sistema acionava autonomamente resfriadores termoelétricos, aquecedores e umidificadores sempre que os sensores detectavam desvio das condições ideais de temperatura e umidade. Os dados circulavam pela plataforma comercial Blynk, e registros fotográficos do desenvolvimento dos fungos eram enviados periodicamente ao Dropbox. A validação dos sensores resultou em MAPE inferior a 10% para todas as variáveis monitoradas, margem compatível com aplicações de monitoramento ambiental contínuo.

O que distingue esse trabalho do presente estudo não é o objeto de cultivo, mas a pergunta que cada um se propõe a responder. Chong *et al.* (2023) investigam se atuadores físicos conseguem manter um microambiente agrícola dentro de condições ideais, e respondem afirmativamente. Este trabalho parte de uma questão diferente: como estruturar a transmissão de dados ambientais para que ela seja viável sob restrições reais de *hardware* e rede?

Essa diferença de foco se desdobra na arquitetura. A dependência do Blynk em Chong *et al.* (2023) demonstra uma delegação da infraestrutura de dados. Embora coerente com a concentração de esforços no controle físico, o custo dessa delegação é o controle sobre como os dados são organizados, por quanto tempo persistem e sob quais condições podem ser recuperados. O presente trabalho assumiu essa complexidade, com *backend* próprio em Spring Boot e PostgreSQL e interface em React JS, porque a telemetria aqui é o objeto da investigação, não seu subproduto.

É nesse ponto que os dois trabalhos mais se afastam. Chong *et al.* (2023) concentram a investigação no controle físico do ambiente e delegam a infraestrutura de dados a uma plataforma comercial, o que é coerente com o escopo do problema que os autores se propuseram a responder: o custo de transmissão dos dados coletados simplesmente não integra as questões de pesquisa que orientam o trabalho.

Este trabalho assume a posição inversa. A telemetria é o próprio objeto de investigação, e não um subproduto do controle físico, o que exige controle direto sobre como as mensagens são estruturadas antes de serem transmitidas. Isso é facilmente observado pelas otimizações aplicadas na transmissão dos dados, que resultaram em uma redução de 89% no volume transmitido no cenário testado.

Fissore *et al.* (2024) partem de uma constatação amplamente documentada: pessoas passam cerca de 90% do tempo em ambientes fechados, e a qualidade ambiental desses espaços, térmica, acústica, luminosa e do ar, tem efeitos mensuráveis sobre saúde e produtividade. O problema que os autores identificam não é a ausência de sistemas de monitoramento. Além disso, a principal lacuna que buscam preencher frente às plataformas existentes é a garantia de confiabilidade e rastreabilidade metrológica dos dados gerados por sensores de baixo custo, o que exige um rigoroso processo de calibração.

A resposta dos autores foi o PROMET&O, cuja principal aposta é tratar o sistema como um problema de engenharia integral, com decisões de projeto explícitas em várias camadas. No *hardware*, Fissore *et al.* (2024)

desenvolveram um dispositivo multissensorial baseado em microcontrolador, onde o posicionamento físico foi pensado para minimizar interferências mútuas, garantindo o isolamento térmico entre fontes de calor e elementos sensores por meio de uma parede divisória interna. Essas decisões revelam uma preocupação com a qualidade do dado na origem, antes de qualquer estratégia de transmissão.

A contribuição mais diretamente relevante para esta pesquisa está na estratégia de aquisição. A solução opera no dispositivo: o microcontrolador agrega cada série de leituras em parâmetros estatísticos (como a determinação dos percentis $L10$ e $L90$ para o domínio acústico) e transmite apenas esse resumo no intervalo de reporte configurado. Trabalhos anteriores enviavam apenas a média; Fissore *et al.* (2024) argumentam que isso implica perda de informação sobre eventos transitórios, como aberturas de janelas ou picos de poluentes. No banco de dados, um *scheduler* realiza pré-agregações em diferentes granularidades, como tempo real, três horas, um dia, três dias, uma semana e um mês. Isso elimina o custo de recalcular agregações sob demanda nas consultas históricas.

Por fim, na camada de interface, o sistema utiliza uma aplicação *web* que incorpora visualizações do Grafana, permitindo aos usuários o acesso em tempo real aos dados medidos e aos índices calculados, além de integrar a coleta de *feedback* subjetivo de conforto.

O PROMET&O é, entre os trabalhos examinados, o que mais se aproxima desta pesquisa em termos de abrangência arquitetural. Ambos os sistemas cobrem a cadeia completa da coleta à visualização, utilizam MQTT, adotam banco de dados relacional e desenvolveram interfaces *web* próprias integradas a ferramentas de *dashboard*. Essa convergência não é coincidência: indica que as duas propostas partem da mesma premissa de que o controle sobre a infraestrutura é condição para investigar o comportamento do sistema, não apenas para utilizá-lo.

É justamente essa proximidade que a diferencia por completo quanto ao aspecto do tratamento dos dados. O PROMET&O superou o desafio do volume de dados na camada de coleta: os valores brutos são

descartados no dispositivo, e apenas suas representações estatísticas são transmitidas. A solução é eficiente, mas irreversível: os dados brutos normalmente não são enviados ao servidor e não podem ser resgatados para as análises que necessariamente dependem da sequência original das leituras coletadas.

No presente trabalho, faz-se o oposto: mantêm-se os valores individuais, e a redução de volume é buscada por meio da reestruturação da mensagem, com a substituição do JSON pelo Protocol Buffers, o agrupamento das leituras em lote e a codificação da diferença temporal entre registros consecutivos. Com essa abordagem, alcançou-se uma redução significativa no volume transmitido sem descartar nenhuma leitura individual, mantendo-se a série histórica completa disponível para análises futuras

O que distingue o PROMET&O do presente estudo de caso não é uma superioridade técnica, mas uma combinação específica de condições: sensoriamento do tipo analógico contínuo, *gateway* de borda limitado em recursos e transmissão sem descarte das leituras, tudo sob custo acessível. É nesse cruzamento que esta pesquisa se concentra.

Tabela 1 – Comparativo dos trabalhos relacionados frente à abordagem proposta.

Trabalho	Foco Principal	Arquitetura / Tecnologia	Diferença de Abordagem
Hernández-Morales <i>et al.</i> (2023)	Predição térmica em estufas.	Rede Sigfox e Redes Neurais (ANN).	Prioriza longo alcance com amostragem periódica, enquanto este trabalho foca em amostragem contínua.
Wan <i>et al.</i> (2022)	Monitoramento integrado de aquaponia.	Visão na borda e nuvem comercial.	Utiliza a borda para análise visual e delega infraestrutura, enquanto aqui a borda roteia o fluxo sensorial.
Chong <i>et al.</i> (2023)	Controle autônomo de cultivo.	Atuadores físicos e plataforma <i>Blynk</i> .	Centrado no controle físico do ambiente (atuadores), ao passo que esta pesquisa investiga a própria telemetria.
Fissore <i>et al.</i> (2024)	Qualidade ambiental interna.	Agregação na borda, MQTT e <i>Grafana</i> .	Reduz tráfego agregando leituras na origem, enquanto este otimiza a mensagem para preservar os dados brutos.
Esta pesquisa	Telemetria contínua em redes restritas.	Backend próprio, <i>Protobuf</i> e codificação.	Mantém controle integral da infraestrutura e foca na compactação estrutural para preservar a série histórica.

Fonte: Elaborado pelo autor (2026).

Como sintetizado na tabela, reunidos, esses quatro trabalhos delimitam o espaço em que esta pesquisa se posiciona. Hernández-Morales *et al.* (2023) otimizam cobertura sensorial e antecipam tendências climáticas, mas sob cadência de amostragem incompatível com o acompanhamento contínuo pressuposto aqui. Wan *et al.* (2022) e Chong *et al.* (2023) mostram que ganhos de automação e inteligência local

costumam vir acoplados a um domínio específico ou à delegação da infraestrutura a terceiros. Fissore *et al.* (2024) reduz drasticamente o volume de dados sem renunciar a arquitetura própria, mas ao custo do descarte irreversível das leituras individuais. Nenhum desses trabalhos reúne, ao mesmo tempo, amostragem contínua, controle integral da cadeia de dados e preservação da série original sob restrição real de rede, é nessa combinação que se situa o sistema descrito no capítulo seguinte.

3 METODOLOGIA

3.1 CLASSIFICAÇÃO DA PESQUISA

A definição da abordagem metodológica deste estudo busca tornar explícito o percurso adotado na construção e na análise do sistema proposto, permitindo compreender de que forma as decisões técnicas se articulam aos fundamentos teóricos que sustentam a investigação. Para isso, a pesquisa é caracterizada a partir das dimensões de natureza, abordagem, objetivos e procedimentos técnicos, conforme categorias consolidadas na literatura de metodologia científica.

Quanto à natureza, trata-se de uma pesquisa aplicada, uma vez que está orientada à resolução de um problema específico por meio do desenvolvimento de uma solução funcional. Conforme Gil (2008), esse tipo de pesquisa tem como finalidade gerar conhecimento voltado à prática, o que se evidencia na proposta de construção de uma arquitetura baseada em Internet das Coisas para o monitoramento de variáveis ambientais no cultivo de plantas.

Em relação à abordagem, a pesquisa apresenta caráter quali-quantitativo. A dimensão qualitativa está associada às decisões de projeto, envolvendo a definição da arquitetura, a organização das camadas e a escolha dos mecanismos de comunicação e processamento. A dimensão quantitativa decorre da observação do comportamento do sistema em funcionamento, considerando aspectos como fluxo de dados, estabilidade da comunicação e resposta das leituras ao longo do tempo. Lakatos e Marconi (2017) destacam que a articulação entre essas abordagens é adequada quando há integração entre análise interpretativa e observação empírica.

No que se refere aos objetivos, o estudo é classificado como exploratório e descritivo. A dimensão exploratória está relacionada à investigação da viabilidade da arquitetura proposta, enquanto a dimensão

descritiva se manifesta na apresentação do funcionamento do sistema e na organização de seus elementos constitutivos.

Quanto aos procedimentos técnicos, a pesquisa assume caráter experimental, uma vez que envolve a construção e a observação de um protótipo em funcionamento. De acordo com Lakatos e Marconi (2017), a pesquisa experimental se caracteriza pela análise dos efeitos decorrentes de uma intervenção sobre determinado objeto, o que, neste estudo, ocorre por meio da implementação do sistema e da observação de seu comportamento em operação.

Com o objetivo de estruturar essas escolhas, apresenta-se o Quadro 1.

Quadro 1 - Classificação metodológica da pesquisa

Dimensão	Classificação	Fundamentação teórica
Natureza	Aplicada	Gil (2008)
Abordagem	Quali-quantitativa	Marconi e Lakatos (2017)
Objetivos	Exploratória e descritiva	Gil (2008)
Procedimentos técnicos	Experimental	Marconi e Lakatos (2017)

Fonte: Elaborado pelo autor com base em Gil (2008) e Lakatos e Marconi (2017).

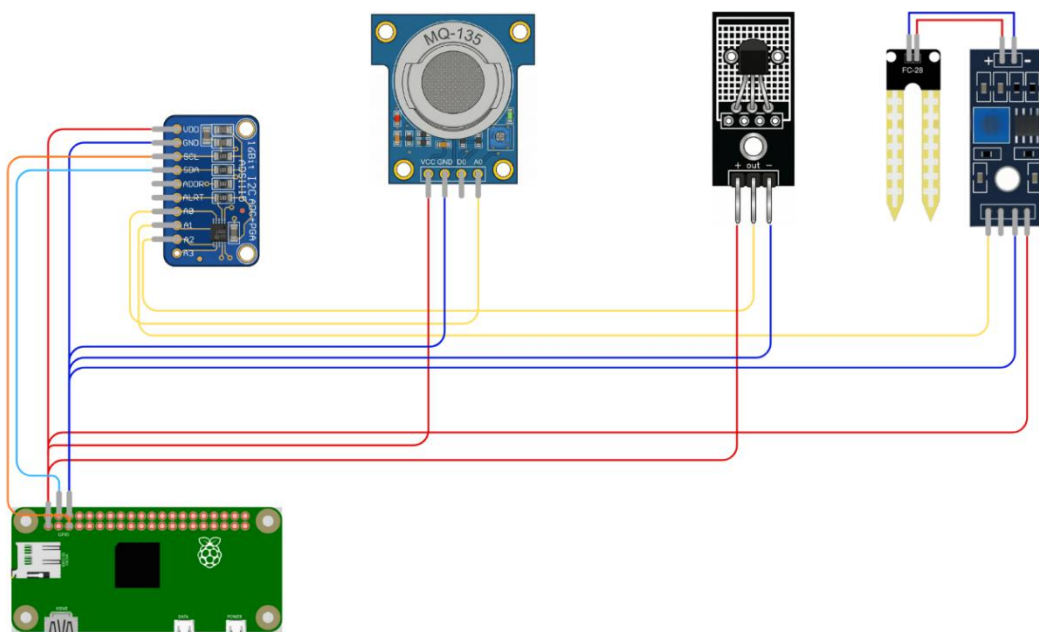
A classificação apresentada no Quadro 1 permite compreender o delineamento metodológico adotado de forma integrada, evidenciando como as diferentes dimensões da pesquisa se articulam na construção do estudo. Essa organização não apenas explicita o enquadramento teórico-metodológico, mas também orienta as etapas de desenvolvimento e análise do sistema proposto, garantindo coerência entre os objetivos definidos, os procedimentos adotados e a forma de interpretação dos resultados.

3.2 ARQUITETURA DO SISTEMA

A organização da arquitetura estrutura o funcionamento da proposta, pois define como os dados são captados, processados, transmitidos e disponibilizados ao usuário. Nesse sentido, o sistema foi estruturado em camadas, de modo a distribuir responsabilidades e permitir que cada etapa do fluxo de dados seja tratada de forma específica dentro do ecossistema.

Para tornar essa organização mais clara, a Figura 4 apresenta a estrutura geral da arquitetura desenvolvida, evidenciando a relação entre os componentes físicos, o *gateway*, a infraestrutura em nuvem e a interface de visualização.

Figura 4 - Arquitetura geral do sistema IoT proposto



Fonte: Elaborado pelo autor (2026).

A partir da representação apresentada, é possível compreender que o sistema se inicia na camada de captação, onde sensores realizam a leitura de variáveis ambientais, como temperatura, umidade do solo e qualidade do ar. Esses sinais são convertidos por meio do módulo ADS1115, permitindo sua utilização pelo sistema computacional.

A seleção dessas variáveis está alinhada às condições tipicamente monitoradas em ambientes de cultivo protegido, nos quais o controle climático ativo depende do acompanhamento constante desses parâmetros.

Na sequência, os dados são encaminhados à camada de borda, onde o Raspberry Pi Zero W atua como *gateway*. Nesse ponto, ocorre não apenas a leitura das informações, mas também seu tratamento inicial, incluindo organização e preparação para transmissão. Essa etapa concentra parte do processamento próximo à origem dos dados, reduzindo a dependência da infraestrutura em nuvem.

A camada seguinte corresponde à nuvem, responsável pelo recebimento, processamento e armazenamento das informações. Por fim, a camada de apresentação permite que os dados sejam visualizados pelo usuário, integrando informações em tempo real e registros históricos. A estrutura em camadas, portanto, não apenas organiza o sistema, mas define o percurso que os dados percorrem desde a coleta até sua visualização.

3.3 COMPONENTES DO SISTEMA

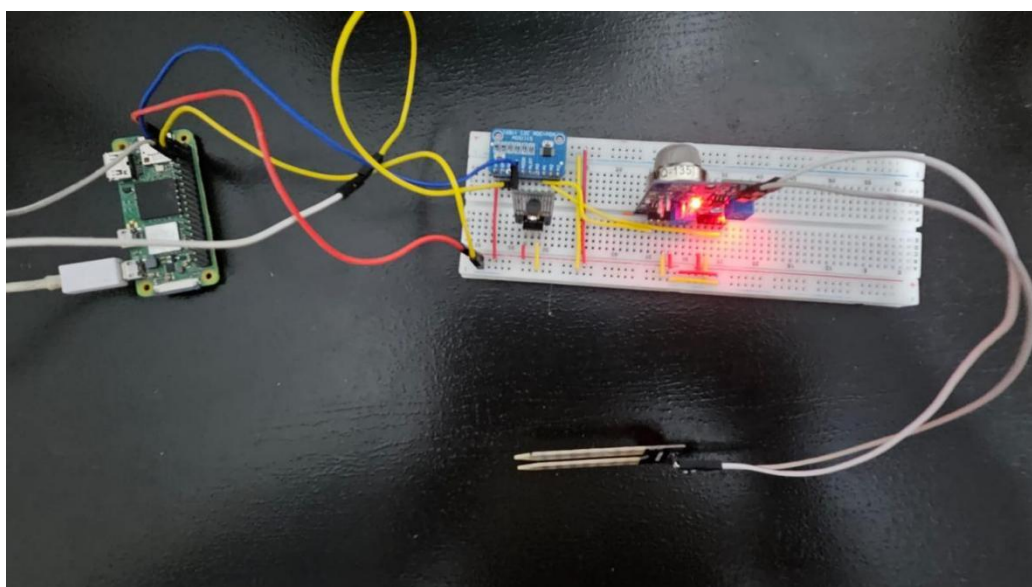
A definição dos componentes do sistema foi estabelecida a partir da necessidade de garantir a viabilidade técnica da solução, mantendo compatibilidade com a proposta de desenvolvimento de uma arquitetura de baixo custo. Essa escolha envolveu a seleção de dispositivos capazes de operar de forma integrada, considerando tanto a captação das variáveis ambientais quanto a transmissão e o processamento dos dados ao longo das camadas do sistema.

Na camada de borda, o Raspberry Pi Zero W foi adotado como unidade de processamento, desempenhando o papel de *gateway* entre os sensores e a infraestrutura em nuvem. Sua utilização permite a execução local das rotinas de leitura, bem como a organização inicial dos dados

antes de sua transmissão. A captação das variáveis ambientais é realizada por sensores analógicos, cuja integração ao sistema ocorre por meio do conversor analógico-digital ADS1115, responsável por transformar os sinais elétricos em valores digitais interpretáveis pelo sistema computacional.

Com o objetivo de evidenciar a implementação prática dessa configuração, a Figura 5 apresenta o protótipo desenvolvido, no qual se observa a integração entre o dispositivo de processamento, os sensores e os elementos de conexão utilizados.

Figura 5 - Montagem do protótipo com Raspberry Pi e sensores



Fonte: Elaborado pelo autor (2026).

Na montagem apresentada, os sensores encontram-se conectados ao *gateway* por meio de uma protoboard, que viabiliza a organização das conexões e a distribuição de energia entre os componentes. Essa configuração permite a leitura contínua das variáveis ambientais, ao mesmo tempo em que mantém flexibilidade para ajustes e reconfigurações durante o desenvolvimento do sistema.

Foram utilizados sensores específicos para cada variável monitorada, considerando suas características de operação e compatibilidade com a proposta do sistema: o LM35DZ para medição de temperatura, o YL-69 para detecção de umidade do solo e o MQ-135 para

análise da qualidade do ar. A escolha desses componentes considerou seu baixo custo e a facilidade de integração com plataformas embarcadas, aspectos centrais para a proposta deste trabalho.

Em termos de viabilidade econômica dessa seleção de *hardware*, a Tabela 2 apresenta a relação dos componentes utilizados na camada de borda e seus respectivos custos médios de mercado.

Tabela 2 – Estimativa de custos dos componentes de hardware do protótipo.

Componente	Preço Estimado (R\$)
Raspberry Pi Zero W	180,00 a 250,00
Módulo ADS1115	25,00 a 40,00
Protoboard	15,00 a 30,00
Sensor LM35DZ	10,00 a 15,00
Sensor YL-69	10,00 a 15,00
Sensor MQ-135	20,00 a 35,00
Cabos Jumper	10,00 a 15,00
Custo Total Aproximado	270,00 a 400,00

Fonte: Elaborado pelo autor (2026).

Como se observa na Tabela 2, o custo total estimado para a montagem da unidade de borda situa-se em um patamar comercialmente acessível, o que se alinha à intenção inicial de evitar a dependência de *hardwares* industriais de alto valor. Uma vez estruturada a infraestrutura física, a coordenação dessas leituras passa a depender diretamente da lógica de controle embarcada no dispositivo.

No plano de *software*, a linguagem Python foi adotada na camada de borda para a implementação das rotinas de leitura dos sensores, tratamento inicial dos dados e envio das informações para a rede. Tal opção resulta da execução do agente em uma distribuição Linux instalada diretamente no *gateway*, o que assegura um ambiente de

execução estável para a continuidade de aplicações, sem dependências com as camadas intermediárias de *firmware*.

É nesse agente que reside a inteligência local do sistema: ele faz a identificação dos instrumentos conectados, interpreta as leituras físicas e aplica um tratamento inicial nos dados, descartando ruídos elétricos e valores fora dos limites esperados, antes de repassá-los para a rede. Este encaminhamento é realizado por meio de uma conexão persistente com o *broker* de mensageria, no qual as leituras são publicadas já agregadas em lotes e com o *payload* reduzido, já que a informação de *timestamp* completo de cada amostra é substituída por *offsets* relativos a um único *timestamp* de tempo-base.

Desse modo, fica demonstrado que a camada de borda concentra não apenas a captação, como também a filtragem e a compressão dos dados, entregando para a nuvem um fluxo já consolidado para o processamento.

Além disso, a comunicação entre a camada de borda e a infraestrutura em nuvem tem o *broker* EMQX como intermediário, quando as leituras publicadas pelo *gateway* são roteadas para os serviços consumidores. Essa mediação favorece a comunicação assíncrona entre os dispositivos, possibilitando o envio das telemetrias de forma constante, sem influenciar o ritmo de processamento da camada de nuvem.

Além de intermediar o tráfego entre a borda e o *backend*, o EMQX fornece os dados também diretamente à interface ao usuário, que se conecta a ele para mostrar as leituras em tempo real sem depender de uma nova consulta ao banco de dados a cada atualização. Essa característica reduz a latência percebida na visualização e evidencia o papel do *broker* como eixo central na arquitetura, conectando simultaneamente os três principais domínios do sistema: a captação na borda, o processamento na nuvem e a apresentação ao usuário.

Na camada de nuvem, o *backend* foi desenvolvido utilizando o *framework* Spring Boot, sendo responsável pela disponibilização de uma API REST e pela intermediação com o banco de dados, armazenado no PostgreSQL, onde os dados são mantidos sob estrutura de tabela. Esse

serviço consome as mensagens que chegam do *broker*, processando cada leitura e armazenando-a de forma organizada, o que assegura a rastreabilidade entre o momento em que é captada e o registro definitivo no banco de dados.

A escolha do PostgreSQL como sistema de armazenamento relaciona-se à necessidade de manter a integridade relacional entre os dados, para que o histórico das variáveis monitoradas seja preservado de forma consistente e consultável ao longo do tempo. Essa persistência estruturada justifica as demais funções da API REST na arquitetura, que vão além do recebimento da telemetria.

A comunicação entre as camadas ocorre por meio dos protocolos MQTT e HTTP, adotados de acordo com a natureza das operações realizadas. O MQTT sustenta o canal contínuo de telemetria já descrito, favorecendo a comunicação assíncrona entre os dispositivos de borda e a infraestrutura em nuvem. O HTTP é utilizado para as operações vinculadas à API REST: consultando o histórico de leituras armazenadas no banco de dados e gerenciando a topologia do sistema, com o cadastramento de novos sensores e o ajuste dos parâmetros de configuração.

Essa segmentação de responsabilidades entre os dois protocolos diz respeito à natureza dos dados que trafegam em cada canal, um aspecto relevante para a compreensão da comunicação do sistema em desenvolvimento.

A interface do sistema foi desenvolvida em React JS, biblioteca que estrutura a camada de apresentação, composta em componentes reutilizáveis, cada um representando uma parte específica do painel de controle de monitoramento. Essa abordagem confere caráter modular à interface, no qual cada sensor cadastrado passa a ser implementado em um elemento visual próprio, incluindo o valor instantâneo da leitura e o histórico da sua evolução. Na medida em que novos instrumentos são acrescentados à topologia do sistema, a interface torna-se automaticamente ciente disso, não necessitando de adaptação estrutural para tal.

Do ponto de vista funcional, a interação do usuário dá-se através de painéis que integram, simultaneamente, o acompanhamento das variáveis monitoradas em tempo real, e a consulta aos registros históricos das leituras. Essa dualidade possibilita que o usuário observe tanto o comportamento atual do ambiente, como o seu comportamento em intervalos mais longos, sendo que uma forma de visualização não exclui a outra.

Além do monitoramento, a interface agrega as funções de gerenciamento do sistema, possibilitando o cadastro de novos sensores, o ajuste de seus parâmetros e a organização da infraestrutura à medida que as necessidades mudam no cultivo.

3.4 MODELAGEM DO BANCO DE DADOS

A estrutura relacional mencionada anteriormente é detalhada a seguir, evidenciando como os dados foram organizados para sustentar tanto a leitura em tempo real do estado atual do cultivo quanto a reconstrução de séries históricas. Essa dualidade orienta praticamente todas as decisões de design que estruturam o modelo, que é apresentado em dois blocos: o primeiro voltado à identidade dos usuários, dispositivos e ao provisionamento dos sensores; o segundo, ao monitoramento agrônomo e à persistência da telemetria.

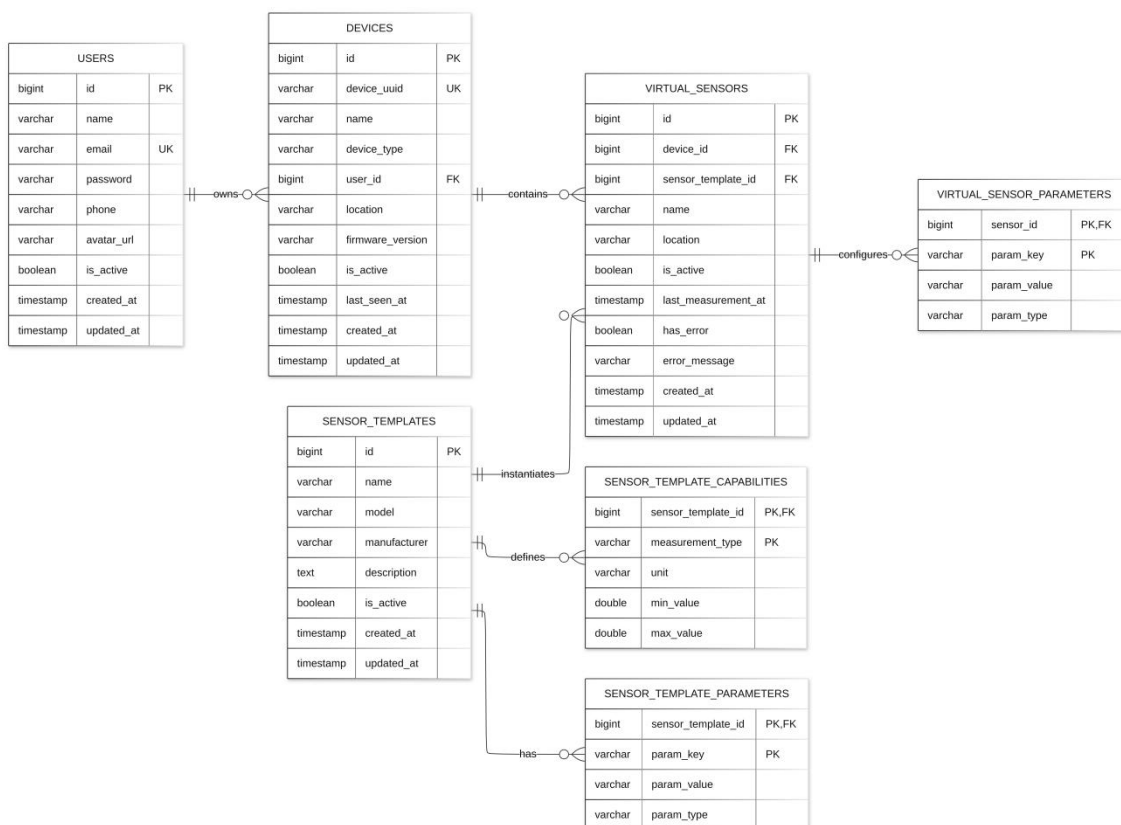
O pilar dessa estrutura é a entidade de usuários, encarregada de consolidar o isolamento lógico da plataforma. Cada usuário é único proprietário de sua infraestrutura física e de seu histórico de cultivo, de modo que todas as outras entidades decorrem desse pilar, o que permite a rastreabilidade de cada leitura registrada até o usuário responsável. Nessa abordagem, o registro de dispositivos guarda, na nuvem, a identidade de cada equipamento de borda instalado em campo, um identificador único e o instante da última comunicação, o que viabiliza o monitoramento contínuo da disponibilidade do equipamento.

Nesse encadeamento, a decisão mais relevante da modelagem é a separação entre a definição de um sensor modelo e sua instância concreta, sendo a primeira um catálogo de *hardware* homologado apresentando capacidades de medição, unidades e parâmetros padrão, e a segunda associando este catálogo com um dispositivo real, cujos parâmetros não seguem necessariamente a determinação do seu modelo, mas sim aquilo que foi definido pelo usuário. Essa separação é o que viabiliza o provisionamento dinâmico de sensores (que será detalhado mais adiante) sem a necessidade de alterar a estrutura do banco a cada novo sensor cadastrado.

Essa flexibilidade, no entanto, também possui sua desvantagem: a resolução de um parâmetro será dependente de uma camada adicional de verificação, em que as configurações definidas pelo usuário se sobrepõem aos padrões do catálogo, recorrendo-se a estes apenas na falta de inserção explícita de valores. A mesma instância concreta também retém informações sobre falhas interceptadas na borda e sobre o último instante de leitura recebida do coletor, não somente sustentando a consistência dos dados persistidos, mas permitindo acompanhar a saúde da própria infraestrutura de sensoriamento enquanto opera.

A Figura 6 apresenta o modelo relacional correspondente a esse primeiro bloco, evidenciando a hierarquia entre usuários, dispositivos, catálogo de sensores e suas instâncias.

Figura 6 - Modelo ER: usuários, dispositivos e provisionamento de sensores



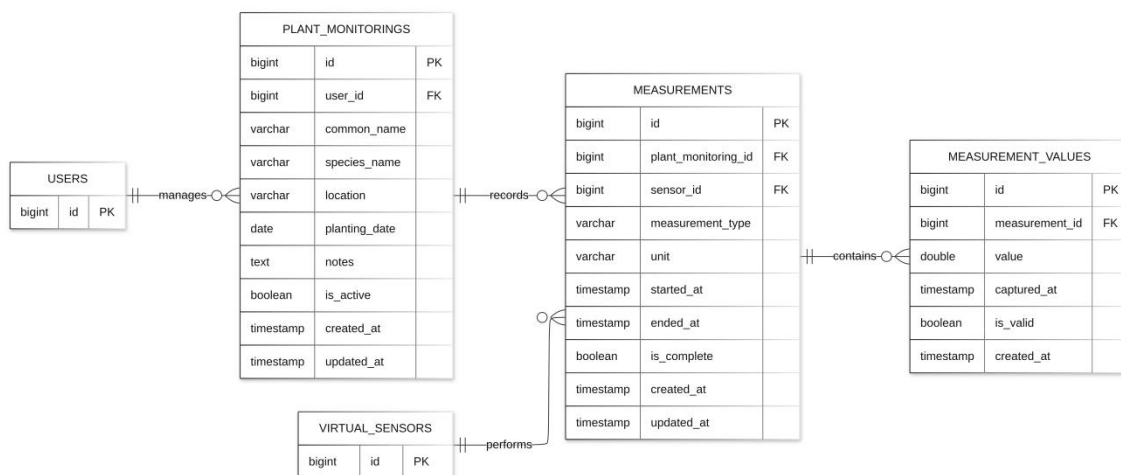
Fonte: Elaborado pelo autor (2026).

O monitoramento agrônômico, por sua vez, é configurado paralelamente à própria topologia eletrônica, de tal modo que a troca de um sensor físico não compromete a identidade da planta, nem a integridade do seu histórico. A persistência da telemetria segue a mesma lógica de desacoplamento: uma entidade opera como cabeçalho de cada sessão de coleta, vinculando o monitoramento, o sensor e a variável observada, enquanto outra guarda apenas a carga útil, ou seja, os valores numéricos e os instantes de captura. Em outras palavras, o foco recai sobre a variável climática registrada, e não sobre o instrumento que a registra, o que permite que sensores diferentes realizem a coleta da mesma grandeza sem alterar o contexto agrônômico.

Tal separação não é apenas uma conveniência de organização: ela evita a repetição de metadados em cada registro, o que favorece tanto a inserção contínua de leituras quanto as consultas por intervalo de tempo, aspecto que se torna crucial quando o volume de dados armazenados cresce durante a operação do sistema.

A Figura 7 apresenta o modelo relacional desse segundo bloco, no qual o monitoramento agrônômico se conecta ao sensor responsável pela coleta, mantendo a separação entre o cabeçalho da sessão de medição e os valores numéricos efetivamente coletados.

Figura 7 – Modelo ER: Monitoramento e suas medidas



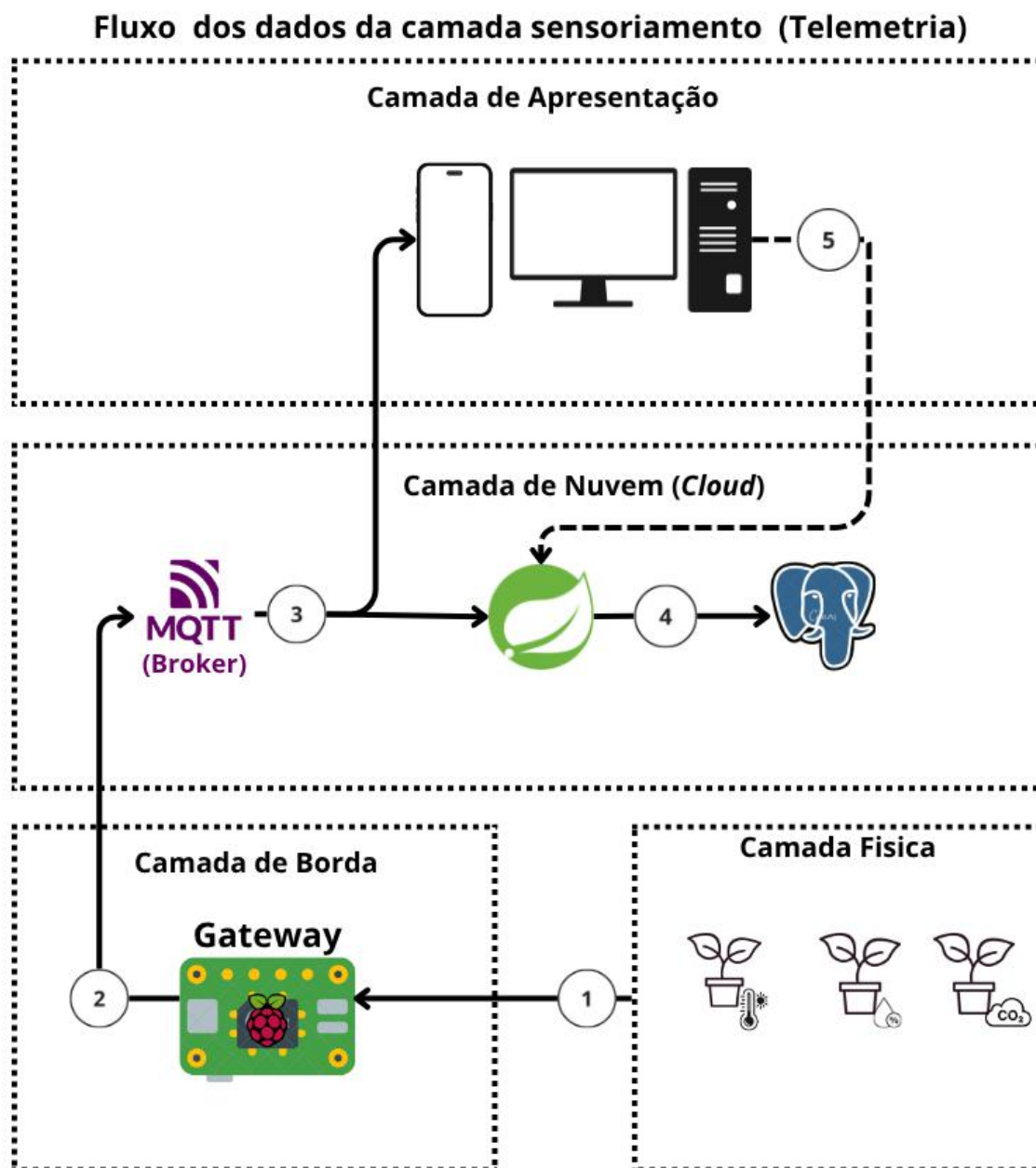
Fonte: Elaborado pelo autor (2026).

3.5 DESENVOLVIMENTO E LÓGICA DO SISTEMA

O desenvolvimento do sistema foi conduzido com o objetivo de estruturar o percurso dos dados desde sua captação no ambiente até sua disponibilização ao usuário, considerando a integração entre os componentes físicos e as camadas computacionais. Essa organização permite compreender o sistema não apenas pelos elementos que o compõem, mas pelo modo como as informações circulam entre eles.

Para explicitar esse funcionamento, a Figura 8 apresenta o fluxo de dados da telemetria, evidenciando o caminho percorrido pelas informações ao longo das camadas do sistema.

Figura 8 – Fluxo de dados da telemetria no sistema



Fonte: Elaborado pelo autor (2026).

Conforme representado, os dados são inicialmente obtidos pelos sensores acoplados às plantas, que realizam a leitura das variáveis ambientais, como temperatura, umidade e níveis de CO₂ (1). Essas leituras são encaminhadas ao *gateway*, implementado sobre um Raspberry Pi na camada de borda, que recebe, lê e organiza inicialmente as informações

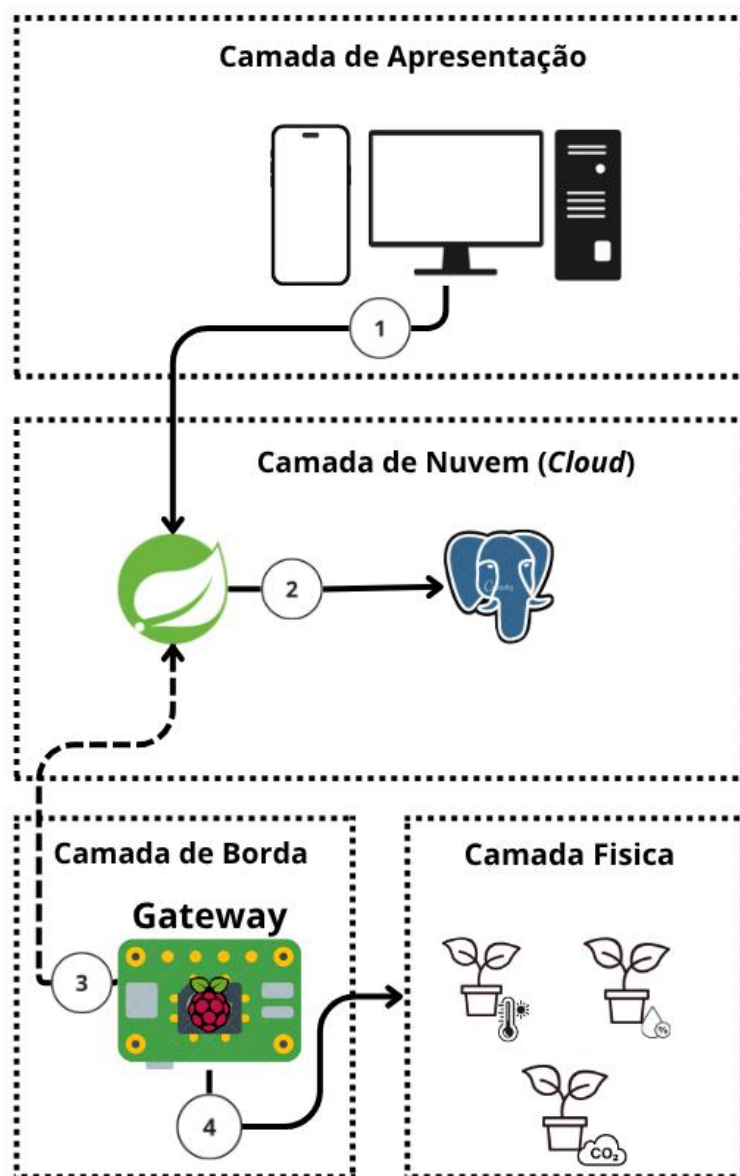
(2). Nesse momento, são realizadas operações de tratamento preliminar dos dados, tendo em vista a preparação para transmissão.

Na sequência, as leituras tratadas no *gateway* são publicadas por meio do protocolo MQTT, cujo *broker* exerce a função de intermediário entre a camada de borda e a infraestrutura em nuvem, permitindo o fluxo contínuo das mensagens (3). Na camada de nuvem, os dados recebidos pelo *broker* são processados por um serviço Web API (desenvolvido em Spring, como representado no ícone da Figura 8), e então, são armazenados em um banco de dados relacional (4), permitindo a recuperação dos dados em outro momento.

Finalmente, na camada de apresentação, os dados armazenados são disponibilizados ao usuário por meio de dispositivos como *smartphones* e *desktops* (5), possibilitando a verificação das leituras em tempo real e consulta posterior dos registros históricos, o que proporciona ampliação das possibilidades de análise do comportamento do sistema ao longo do tempo.

Além do fluxo de telemetria, o sistema contempla a configuração dinâmica dos sensores, permitindo a incorporação de novos dispositivos ao ambiente monitorado. Esse processo é representado na Figura 9.

Figura 9 – Fluxo de provisionamento de sensores

Fluxo de provisionamento dinâmico de sensores

Fonte: Elaborado pelo autor (2026).

A representação evidencia como a definição e a organização dos sensores são realizadas de forma estruturada, integrando a etapa de configuração à lógica geral de funcionamento do sistema. Conforme indicado na Figura 8, o processo se inicia na camada de apresentação, quando o usuário realiza, através da interface, o cadastro de um sensor já instalado fisicamente em um ambiente monitorado (1). Para isso, essa solicitação é transmitida até a camada nuvem, onde o serviço Web API(Spring) a processa e registra as informações do novo dispositivo em

um banco de dados (2). Nesse registro, o provisionamento se torna dinâmico e automático: o *gateway* (Raspberry Pi) consulta a API para obter os dados do sensor recém-cadastrado (3) e, a partir deles, inicia a leitura das variáveis microclimáticas junto à camada física (4), de forma que o novo dispositivo é incorporado no fluxo de telemetria, sem requerer uma intervenção manual na infraestrutura existente. Esse mecanismo torna a arquitetura flexível, viabilizando a expansão do sistema de monitoramento sem exigir alterações diretas na estrutura central da aplicação.

Focalizando nessa organização, a análise é realizada em função do funcionamento do sistema, levando em consideração o fluxo dos dados entre as camadas e como atuam os elementos nesse fluxo. A observação se foca na continuidade da comunicação, estabilidade da transmissão das informações e integração entre os elementos constituintes da arquitetura.

Esse direcionamento permite examinar o sistema como um conjunto estruturado, no qual as etapas de cadastro, provisionamento, coleta, tratamento, transmissão, armazenamento e visualização se relacionam de maneira contínua. A análise não se restringe à verificação isolada dos componentes, mas considera o funcionamento integrado da solução proposta.

Dessa forma, a metodologia estabelece as condições para a avaliação da arquitetura em uso, orientando a observação do sistema em operação e sua capacidade de manter o fluxo de dados de maneira consistente. Essa avaliação é conduzida considerando a continuidade do fluxo de dados, a estabilidade da comunicação entre as camadas e a consistência das informações coletadas ao longo do tempo.

3.6 USO DE FERRAMENTAS DE INTELIGÊNCIA ARTIFICIAL

No desenvolvimento deste trabalho, foi utilizada a ferramenta de Inteligência Artificial Claude (desenvolvida pela Anthropic), exclusivamente para fins de revisão ortográfica e gramatical do texto

produzido. A utilização da ferramenta limitou-se à correção de aspectos linguísticos e de escrita, não tendo sido empregada na geração de conteúdo técnico-científico, na definição da arquitetura proposta, na análise dos dados ou na elaboração das conclusões apresentadas neste estudo, cuja autoria e responsabilidade são integralmente do autor.

4 RESULTADOS E DISCUSSÃO

4.1 EXECUÇÃO DO SOFTWARE EMBARCADO

O funcionamento do software embarcado corresponde, na prática, à camada de borda descrita na arquitetura do sistema. É nesse agente que a leitura bruta dos sensores se converte em dado transmissível, e é nele que reside a primeira camada de decisão sobre o que efetivamente chega à nuvem, antes de qualquer processamento posterior. As execuções registradas a seguir percorrem os estados centrais desse comportamento: o momento em que o dispositivo ainda não possui identidade na plataforma, a operação já consolidada de leitura e envio, e a resposta do agente diante de um sensor que falha.

A primeira execução ocorre em um estado no qual o dispositivo ainda não possui identidade na plataforma. É esse momento que a Figura 10 demonstra: ao não encontrar um registro correspondente na API, o agente cria uma nova entrada e apresenta, simultaneamente, um QR Code e um código de pareamento em texto plano, duas formas de representar o mesmo identificador recém-gerado.

Figura 10 - Início da execução: Cadastro do Sistema Embarcado

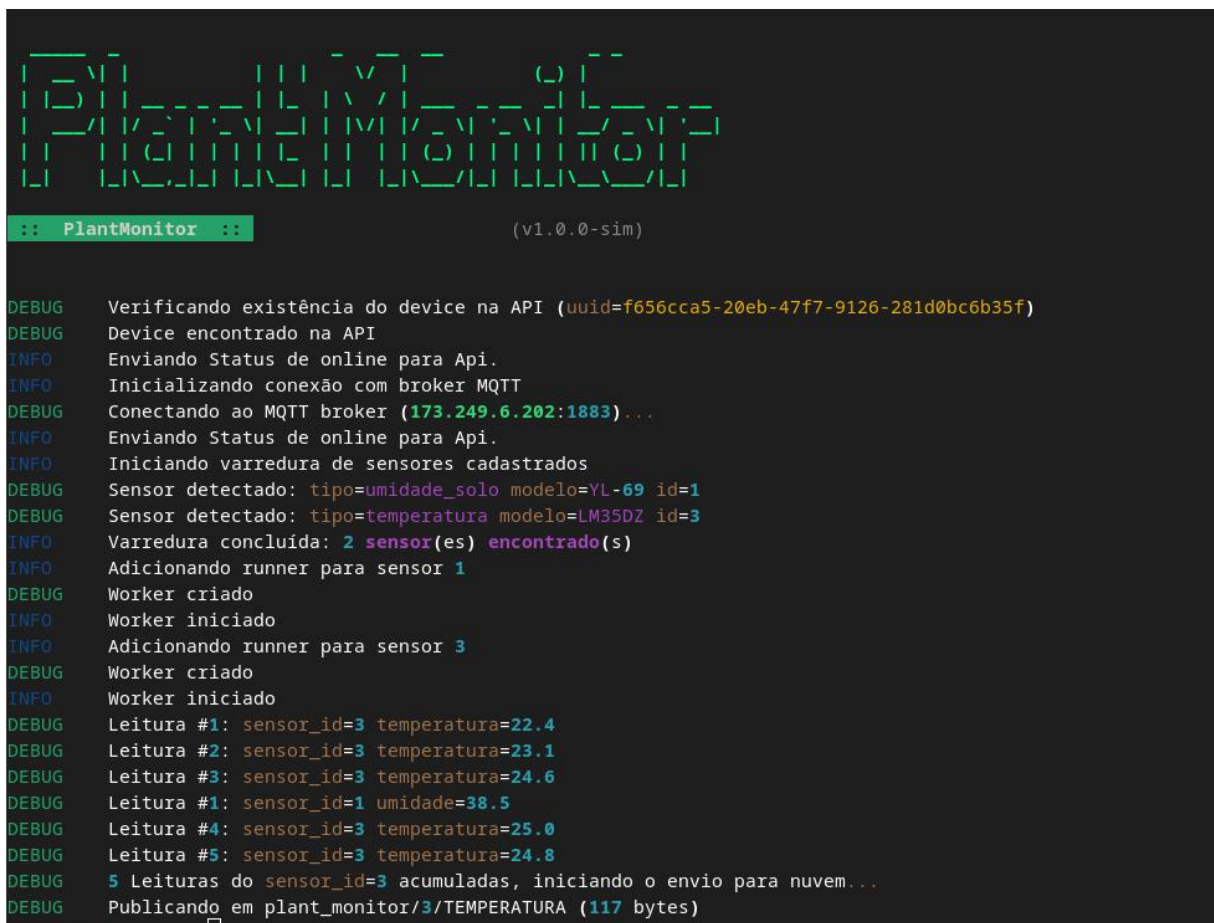


Fonte: Elaborado pelo autor (2026).

Feito isso, é o próprio dispositivo que assume a iniciativa de se fazer reconhecer: em vez de esperar uma ação do usuário, ele consulta repetidamente a API até obter confirmação de que o código foi reconhecido na interface, e só transmite seus dados definitivos e persiste o identificador localmente depois dessa confirmação. Essa ordem importa. Se o dispositivo persistisse o identificador antes da confirmação, uma falha de rede no meio do processo poderia deixá-lo com uma identidade que a plataforma ainda não reconhece.

O comportamento já muda na próxima figura. Com o identificador já persistido, as execuções seguintes não precisam de pareamento o agente verifica sua existência na API, informa que está online e parte direto para a conexão com o *broker* MQTT, intermediário da comunicação entre a borda e a nuvem.

Figura 11 - Conexão com o *broker*, busca de novos sensores e envio de leituras



```

:: PlantMonitor ::                                     (v1.0.0-sim)

DEBUG    Verificando existência do device na API (uuid=f656cca5-20eb-47f7-9126-281d0bc6b35f)
DEBUG    Device encontrado na API
INFO     Enviando Status de online para Api.
INFO     Inicializando conexão com broker MQTT
DEBUG    Conectando ao MQTT broker (173.249.6.202:1883)...
INFO     Enviando Status de online para Api.
INFO     Iniciando varredura de sensores cadastrados
DEBUG    Sensor detectado: tipo=umidade_solo modelo=YL-69 id=1
DEBUG    Sensor detectado: tipo=temperatura modelo=LM35DZ id=3
INFO     Varredura concluída: 2 sensor(es) encontrado(s)
INFO     Adicionando runner para sensor 1
DEBUG    Worker criado
INFO     Worker iniciado
INFO     Adicionando runner para sensor 3
DEBUG    Worker criado
INFO     Worker iniciado
DEBUG    Leitura #1: sensor_id=3 temperatura=22.4
DEBUG    Leitura #2: sensor_id=3 temperatura=23.1
DEBUG    Leitura #3: sensor_id=3 temperatura=24.6
DEBUG    Leitura #1: sensor_id=1 umidade=38.5
DEBUG    Leitura #4: sensor_id=3 temperatura=25.0
DEBUG    Leitura #5: sensor_id=3 temperatura=24.8
DEBUG    5 Leituras do sensor_id=3 acumuladas, iniciando o envio para nuvem...
DEBUG    Publicando em plant_monitor/3/TEMPERATURA (117 bytes)
  
```

Fonte: Elaborado pelo autor (2026).

Ao ser iniciado, o agente verifica a existência do dispositivo na API a partir do identificador já persistido, informa seu status como online e conecta-se ao *broker* MQTT descrito na Seção 3.3. Em seguida, realiza a varredura dos sensores cadastrados, identificando os dispositivos disponíveis para leitura. As leituras de temperatura e umidade realizadas nessa execução não são transmitidas uma a uma: o software as acumula e as envia em um único pacote, apresentando na prática o mecanismo de agregação de leituras fundamentado anteriormente.

Esse mesmo agente, porém, também precisa lidar com sensores que não respondem, o que a Figura 12 documenta.

Figura 12 – Tratamento de falha na leitura de sensor

```

:: PlantMonitor ::                                (v1.0.0-sim)

DEBUG    Verificando existência do device na API (uuid=f656cca5-20eb-47f7-9126-281d0bc6b35f)
DEBUG    Device encontrado na API
INFO     Enviando Status de online para Api.
INFO     Inicializando conexão com broker MQTT
DEBUG    Conectando ao MQTT broker (173.249.6.202:1883)...
INFO     Enviando Status de online para Api.
INFO     Iniciando varredura de sensores cadastrados
DEBUG    Sensor detectado: tipo=umidade_solo modelo=VL-69 id=1
DEBUG    Sensor detectado: tipo=temperatura modelo=LM35DZ id=3
INFO     Varredura concluída: 2 sensor(es) encontrado(s)
INFO     Adicionando runner para sensor 1
DEBUG    Worker criado
INFO     Worker iniciado
INFO     Adicionando runner para sensor 3
DEBUG    Worker criado
INFO     Worker iniciado
WARNING  Sensor não respondeu: sensor_id=3 tentativa=1/5
WARNING  Sensor não respondeu: sensor_id=3 tentativa=2/5
ERROR    Erro na leitura do sensor: sensor_id=3
DEBUG    Enviando erro do sensor (sensor_id=3)
WARNING  Erro do sensor reportado (sensor_id=3)
WARNING  Sensor não respondeu: sensor_id=3 tentativa=4/5
DEBUG    Leitura obtida: sensor_id=3 temperatura=24.3 tentativa=5/5

```

Fonte: Elaborado pelo autor (2026).

Depois de duas tentativas sem resposta, o *software* reporta o problema à própria API, e não apenas a um log local. A diferença é operacional: um registro local só seria útil em uma depuração posterior, enquanto o reporte à API permite que a plataforma saiba, quando ocorre, que aquele sensor está inoperante. Ainda assim, o *software* não trata a falha como definitiva. Ele continua tentando, e o sensor volta a responder na tentativa seguinte, com uma leitura válida. Reportar o erro e insistir na leitura não são comportamentos contraditórios aqui: o primeiro comunica um estado observado naquele instante, o segundo assume que esse estado pode ser transitório. É essa distinção, entre comunicar uma falha e desistir da leitura, que evita que uma instabilidade momentânea do sensor seja tratada como uma condição permanente do sistema.

Essas três execuções, tomadas em conjunto, evidenciam que o *software* embarcado decide localmente sobre identidade, busca de

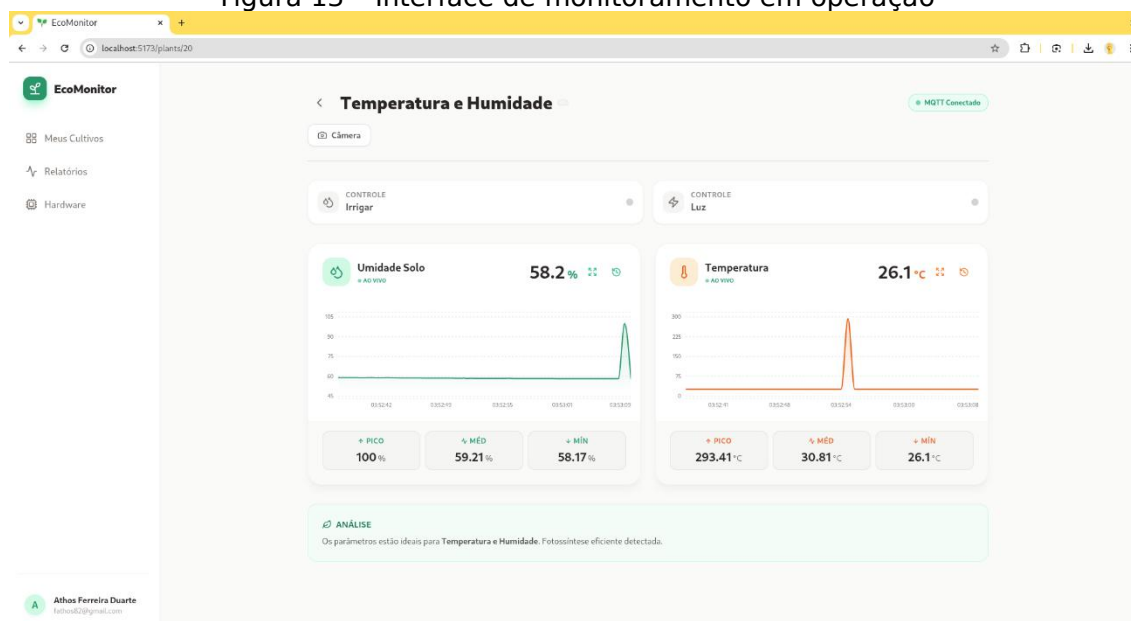
sensores, leitura e falha, antes de qualquer dado alcançar a nuvem, decisões essas que sustentam a operação do sistema, examinada a seguir.

4.2 OPERAÇÃO DO ECOSSISTEMA IMPLEMENTADO

A implementação do sistema permitiu a operação integrada entre os componentes da camada de borda, da infraestrutura em nuvem e da interface de visualização, evidenciando o funcionamento contínuo do fluxo de dados proposto na arquitetura. A partir da execução das rotinas embarcadas no *gateway*, os dados coletados pelos sensores foram processados e transmitidos para a nuvem por meio do protocolo MQTT, sendo posteriormente disponibilizados ao usuário por meio da aplicação *web*. Esse encadeamento operacional indica que as etapas de coleta, transmissão e disponibilização foram efetivamente estabelecidas no ambiente de execução.

Para evidenciar esse funcionamento, a Figura 13 apresenta a interface de monitoramento do sistema em operação, na qual se observa a atualização contínua das variáveis ambientais monitoradas. A visualização em tempo real permite acompanhar o comportamento das leituras conforme são recebidas pela aplicação, o que indica a manutenção do fluxo de dados entre os dispositivos e a interface ao longo do período observado.

Figura 13 – Interface de monitoramento em operação



Fonte: Elaborado pelo autor (2026).

A partir da interface apresentada, percebe-se que as leituras são organizadas de modo a permitir a identificação das variáveis monitoradas e de seus respectivos valores. Essa organização não se restringe à apresentação visual, mas expressa o tratamento realizado na aplicação, na qual os dados recebidos são estruturados antes de serem exibidos ao usuário. A forma como essas informações são dispostas contribui para a interpretação das condições monitoradas, ao mesmo tempo em que revela o funcionamento interno da aplicação.

A observação do sistema em operação também permite examinar a continuidade da comunicação entre as camadas. Durante o período analisado, não se verificaram interrupções na atualização das leituras, o que indica que o envio de dados ocorreu de maneira regular. Esse comportamento sugere que a integração entre a camada de borda e a infraestrutura em nuvem se manteve estável ao longo da execução, sustentando o fluxo de informações necessário para o monitoramento.

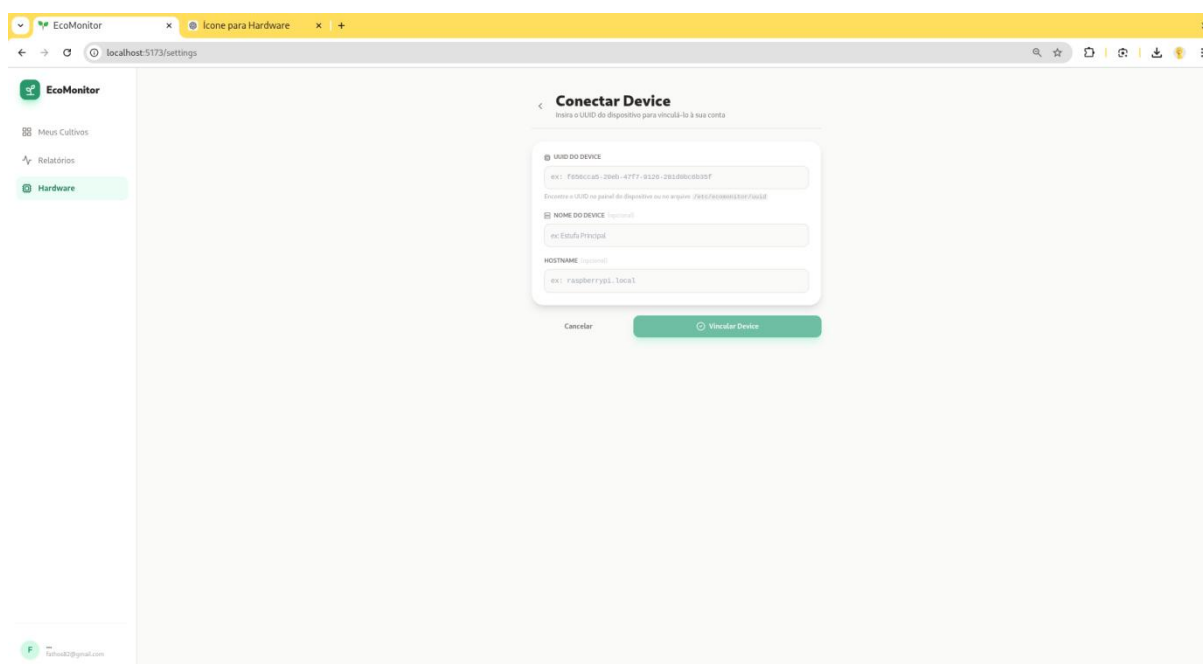
Com isso, a operação do ecossistema implementado evidencia que a arquitetura proposta foi traduzida em um sistema funcional, no qual os dados percorrem as diferentes camadas de forma contínua e organizada. Esse resultado cria as condições para avançar na análise,

direcionando a atenção para o comportamento das leituras, a persistência dos dados e os limites observados durante o funcionamento do sistema.

4.3 GERENCIAMENTO DE HARDWARE

O software embarcado registra-se autonomamente junto à API assim que é executado, gerando o identificador exibido no terminal, conforme demonstrado na Seção 4.1. Nesse ponto, porém, o dispositivo ainda não pertence a ninguém: falta ao usuário reivindicá-lo, o que ocorre na tela de adição de *hardware*, apresentada na Figura 14.

Figura 14 – Tela de registro de dispositivos



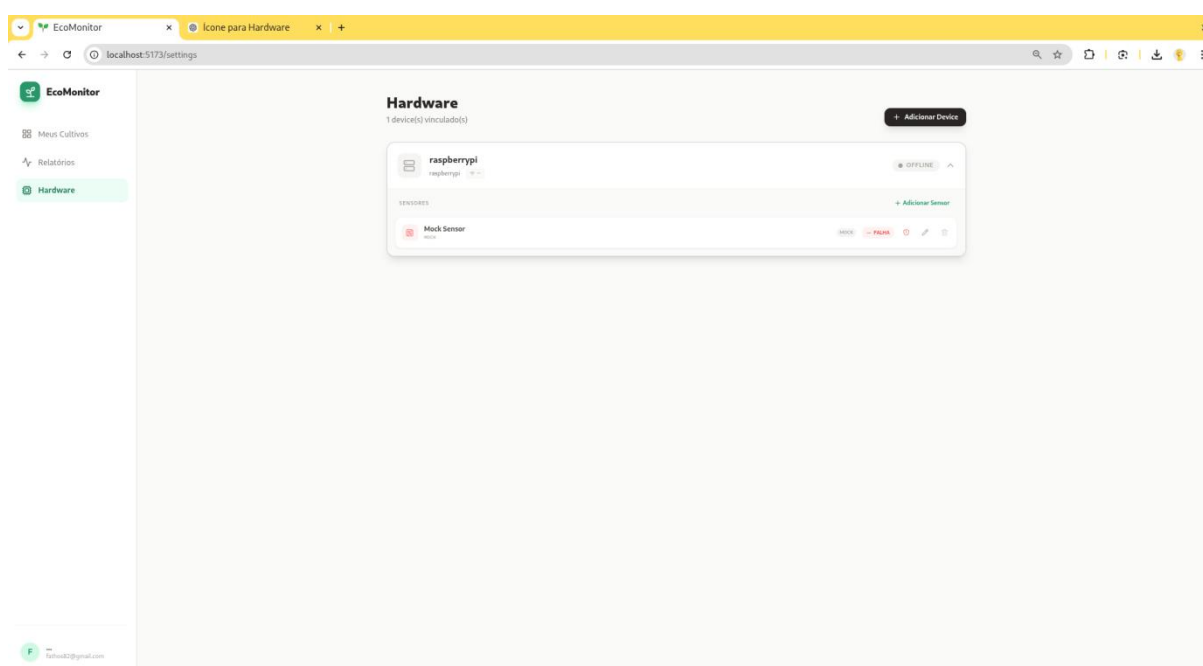
Fonte: Elaborado pelo autor (2026).

O usuário insere o identificador previamente exibido pelo dispositivo e declara, com esse gesto, a propriedade sobre o *hardware*. A separação entre os dois momentos não é incidental. A existência técnica do dispositivo decorre exclusivamente da execução do *software* embarcado; sua apropriação, por outro lado, depende de uma ação do

usuário que o sistema não pode inferir sozinho. Impedir essa inferência automática é o que garante o isolamento entre infraestruturas de diferentes usuários, discutido na modelagem do banco de dados, condição que uma vinculação automática comprometeria diretamente.

Vinculado o dispositivo, ele passa a compor a listagem de *hardware* do usuário, mostrada na Figura 15.

Figura 15 – Tela de listagem de dispositivos



Fonte: Elaborado pelo autor (2026).

A listagem reúne, em um único painel, os dispositivos vinculados ao usuário, exibindo para cada um o identificador gerado no auto registro e o instante da última comunicação recebida, a mesma informação de saúde da infraestrutura discutida na Seção 3.4. É a partir desse último dado que se torna possível perceber, ainda pela interface, um equipamento que parou de responder.

O registro de um novo dispositivo depende, portanto, de dois agentes que não se substituem: o *software* embarcado garante a existência técnica do *hardware* na plataforma; o usuário garante sua

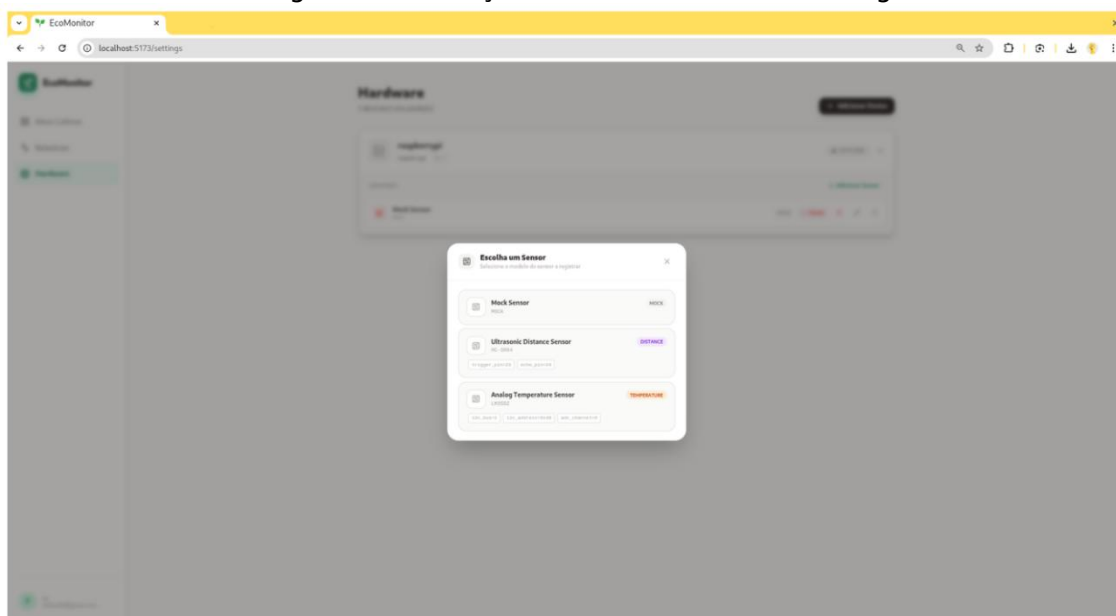
apropriação lógica. É essa divisão que sustenta o isolamento entre infraestruturas distintas.

4.4 PROVISIONAMENTO DINÂMICO DE SENSORES

Uma vez registrado o *hardware*, o passo seguinte é a incorporação dos sensores que nele realizam a coleta. Esse mecanismo de provisionamento, descrito na Figura 9, viabiliza que essa expansão ocorra sem intervenção direta na estrutura da aplicação, e sua materialização na interface evidencia como essa flexibilidade se traduz em fluxo de interação funcional para o cadastro de novos sensores.

A Figura 16 apresenta a etapa inicial desse processo, na qual o usuário seleciona, a partir de um catálogo de modelos homologados, o tipo de sensor a ser registrado.

Figura 16 – Seleção do modelo de sensor a registrar

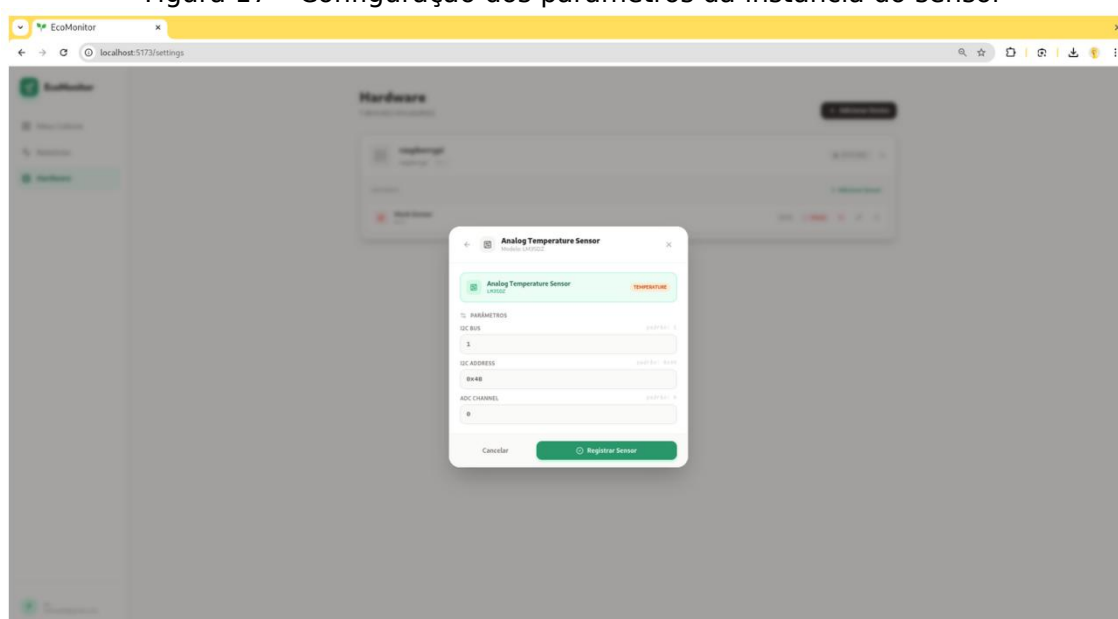


Fonte: Elaborado pelo autor (2026).

Observa-se que cada item do catálogo já apresenta, de forma resumida, os parâmetros de configuração associados ao modelo, a exemplo dos pinos de *trigger* e *echo* do sensor ultrassônico ou do

barramento e endereço I2C do sensor de temperatura. Essa exibição antecipada indica que o catálogo restringe o cadastro a combinações compatíveis com o *hardware* homologado, expressando na interface a separação entre a definição genérica de um sensor e sua instância concreta. Selecionado o modelo desejado, o usuário é conduzido à etapa seguinte do provisionamento, na qual os parâmetros específicos da instância são apresentados para confirmação ou ajuste.

Figura 17 – Configuração dos parâmetros da instância do sensor



Fonte: Elaborado pelo autor (2026).

Nota-se que os campos de configuração vêm preenchidos com os valores padrões definidos no catálogo, incumbindo ao usuário mantê-los ou sobrescrevê-los conforme a instalação física do sensor. Esse comportamento torna visível, em tempo de execução, a hierarquia de precedência concebida na modelagem do banco de dados, onde os valores do catálogo são os padrões iniciais e foram sobrescritos apenas em uma configuração explícita do usuário. Ao mesmo tempo, essa camada adicional de resolução não degrada a simplicidade que o usuário tem, mas fixa, internamente à aplicação, a lógica através da qual a flexibilidade do provisionamento é garantida.

Dessa maneira, observa-se que o provisionamento dinâmico não é apenas uma conveniência de interface, mas expressa, na camada de

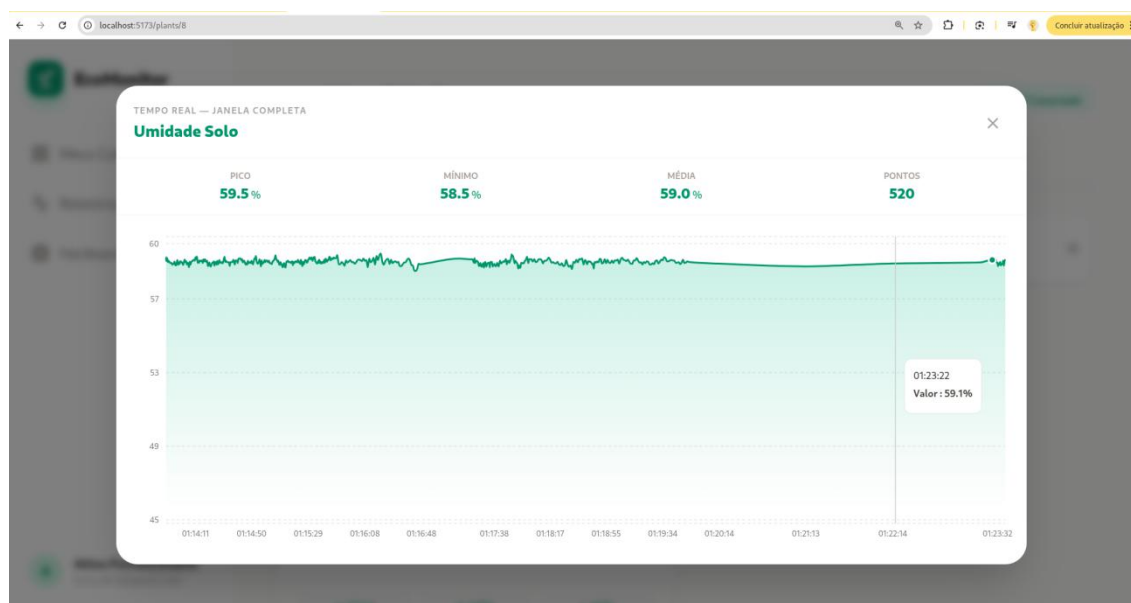
apresentação, as decisões estruturais assumidas na modelagem do sistema. Além disso, torna possível a varredura em busca dos novos sensores por parte do sistema embarcado, comportamento esse, apresentado na Figura 11.

4.5 COMPORTAMENTO DAS VARIÁVEIS MONITORADAS

Uma vez estabelecida a operação contínua do sistema, a análise volta-se ao comportamento das variáveis coletadas ao longo do tempo, considerando a forma como os dados são apresentados na interface e as variações observadas nas leituras. A partir das informações registradas, torna-se possível examinar não apenas os valores medidos, mas também a regularidade das amostragens e a presença de oscilações durante o período monitorado.

A Figura 18 apresenta uma visualização detalhada das leituras de umidade do solo em um intervalo específico, permitindo observar a distribuição dos valores ao longo do tempo. Nota-se que as medições se mantêm em uma faixa relativamente estável, com pequenas variações entre os valores mínimo, médio e máximo registrados no período. Esse comportamento indica que, nas condições observadas, o sensor operou de forma consistente, sem apresentar variações abruptas que comprometessem a continuidade das leituras.

Figura 18 – Comportamento das leituras de umidade do solo



Fonte: Elaborado pelo autor (2026).

Ao observar a série temporal com maior atenção, é possível identificar pequenas oscilações entre leituras consecutivas. Essas variações não indicam falha no processo de aquisição, mas refletem características inerentes ao sensoriamento contínuo, especialmente quando se utilizam sensores analógicos. Fatores como ruído elétrico, variações ambientais de curto intervalo e sensibilidade do dispositivo contribuem para esse comportamento, o que exige cautela na interpretação direta dos valores individuais.

Outro aspecto relevante refere-se à densidade de amostragem, evidenciada pela quantidade de leituras registradas no intervalo analisado. O volume de dados indica que o sistema foi capaz de manter uma frequência de coleta elevada, ampliando a capacidade de observação do ambiente monitorado. Por outro lado, essa característica também impõe a necessidade de organização dos dados, uma vez que a repetição de valores próximos pode dificultar a identificação de padrões quando não há tratamento adequado das informações.

Para complementar essa análise, a Tabela 3 apresenta um consolidado estatístico das leituras obtidas em um intervalo contínuo de três minutos, permitindo observar a relação entre valores mínimos, médios

e máximos para cada variável monitorada. Essa organização contribui para uma leitura mais objetiva do comportamento das medições em curto prazo, reduzindo a dependência exclusiva da interpretação visual dos gráficos.

Tabela 3 – Consolidado estatístico de uma amostra de 3 minutos de telemetria

Instrumento	Grandeza Monitorada	Mínimo	Média	Pico (Máximo)
LM35	Temperatura do Ar (°C)	24,25	24,50	24,75
YL-69	Umidade do Solo (%)	46,51	46,80	47,06
MQ-135	Qualidade do Ar (AQI)	371,78	376,97	389,42

Fonte: Elaborado pelo autor (2026).

A partir dos valores apresentados, observa-se que a temperatura do ar se manteve em uma faixa estreita de variação, indicando estabilidade nas condições monitoradas durante o intervalo analisado. A umidade do solo apresenta variações discretas, sugerindo alterações graduais no ambiente, compatíveis com o comportamento esperado dessa variável. Já os valores relacionados à qualidade do ar apresentam uma amplitude maior entre as leituras, o que pode estar associado à sensibilidade do sensor MQ-135 e à influência de fatores externos sobre as medições.

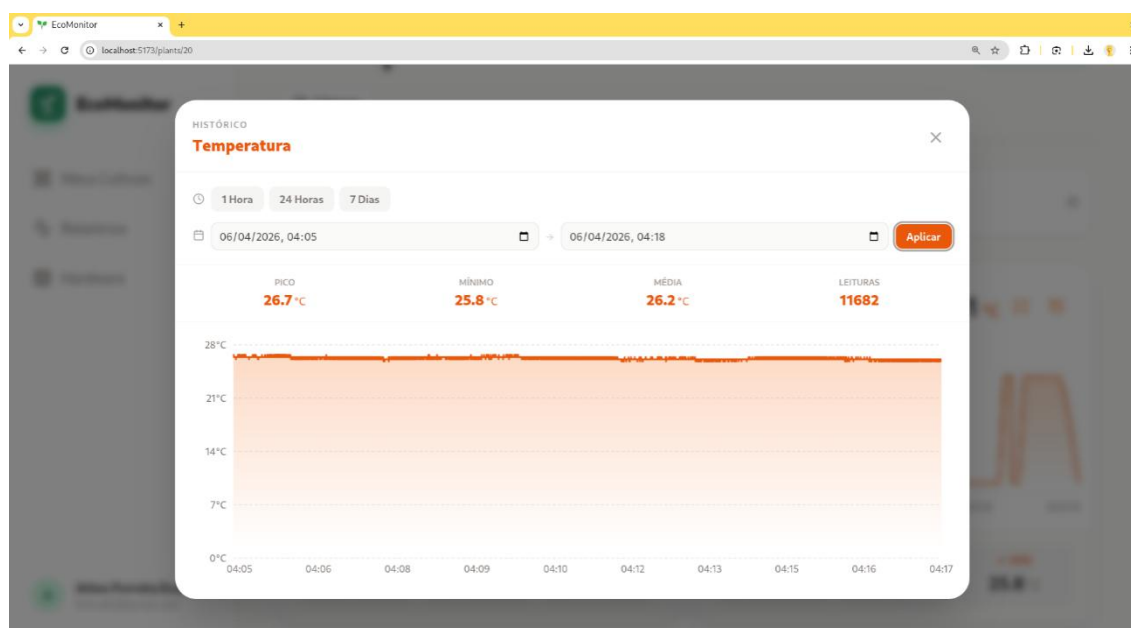
A análise conjunta entre a visualização gráfica e o consolidado estatístico permite compreender que o sistema produz dados consistentes, ao mesmo tempo em que evidencia diferenças no comportamento entre as variáveis monitoradas. Essas diferenças não indicam inconsistência, mas refletem as características específicas de cada sensor e das grandezas observadas. Esse entendimento é relevante para orientar a interpretação dos dados e delimitar o uso dessas informações em análises mais amplas.

4.6 PERSISTÊNCIA E CONSULTA HISTÓRICA DOS DADOS

A análise do sistema não se restringe ao acompanhamento das leituras em tempo imediato, mas envolve também a forma como os dados são armazenados e posteriormente disponibilizados para consulta. A persistência das informações permite reconstruir o comportamento das variáveis ao longo de intervalos maiores, o que amplia as possibilidades de interpretação e acompanhamento das condições monitoradas.

A Figura 19 apresenta a interface de consulta histórica, na qual se observa a organização das leituras ao longo do tempo. A visualização permite identificar a distribuição dos dados em uma escala temporal mais extensa, evidenciando a continuidade do processo de coleta e a manutenção do registro das informações no banco de dados.

Figura 19 – Visualização histórica das leituras monitoradas



Fonte: Elaborado pelo autor (2026).

A partir dessa representação, é possível perceber que o sistema mantém a integridade das informações ao longo do período analisado, permitindo a recuperação dos dados sem perdas aparentes. A disposição sequencial das leituras indica que o processo de armazenamento ocorreu

de forma consistente, preservando a ordem temporal e garantindo a rastreabilidade das medições realizadas.

Por outro lado, a ampliação do intervalo de observação evidencia um aumento significativo no volume de dados apresentados, o que impacta diretamente a forma de visualização. A concentração de pontos em períodos mais longos pode dificultar a leitura imediata das informações, especialmente quando se busca identificar padrões ou variações específicas. Esse aspecto revela uma característica inerente a sistemas de monitoramento contínuo, nos quais a densidade de dados cresce rapidamente ao longo do tempo.

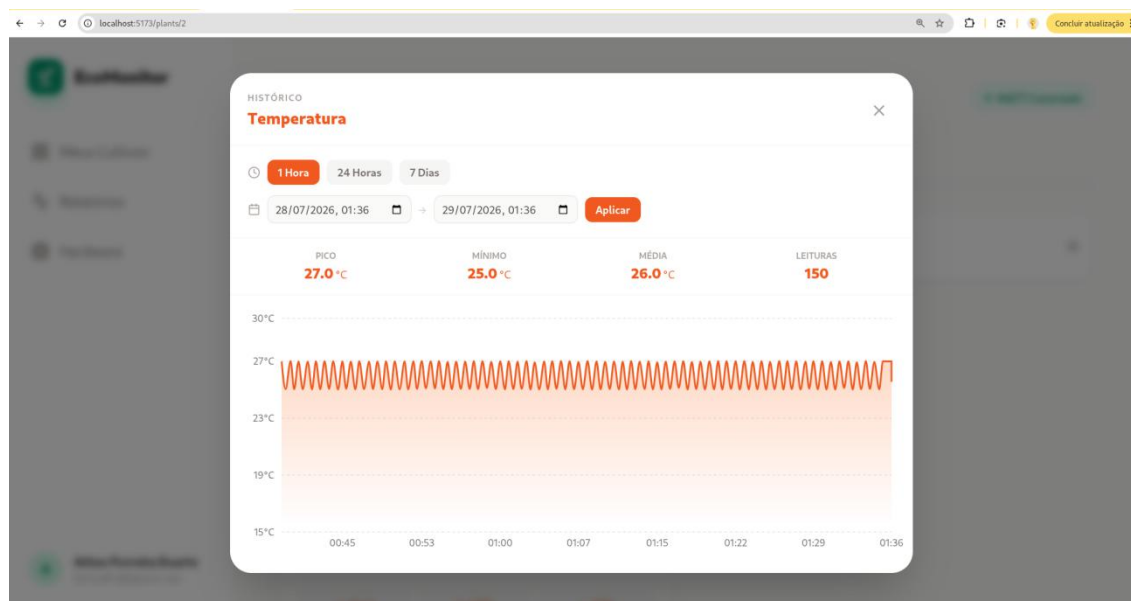
Essa condição exige o emprego de estratégias que permitam equilibrar a quantidade de informações exibidas e a legibilidade da visualização. Nesse contexto, a consulta histórica não se limita à recuperação dos dados armazenados, mas envolve também a forma como essas informações são organizadas e apresentadas ao usuário. A análise desse comportamento evidencia que a persistência dos dados foi efetivamente implementada, ao mesmo tempo em que aponta a necessidade de mecanismos que auxiliem na interpretação de grandes volumes de informações.

A partir da constatação do comportamento, tornou-se necessária a adoção de uma estratégia de tratamento sobre esses dados, apresentados na consulta histórica. Nesse contexto, foi realizado o *downsampling*, isto é, a redução dos pontos apresentados no gráfico, porém sem perda da representatividade da informação ao longo do tempo.

Esse procedimento consiste em agregar as medições originais em intervalos regulares, para que cada janela temporal se transforme em um valor sintético, isto é, média, mínimo e máximo correspondentes, em vez do conjunto total dos registros brutos. Essa estratégia garante um equilíbrio entre a quantidade das informações apresentadas e a legibilidade da visualização, atendendo à necessidade identificada na análise anterior.

A Figura 20 apresenta o resultado da aplicação dessa técnica sobre a interface de consulta histórica.

Figura 20 – Visualização histórica das leituras após aplicação de downsampling



Fonte: Elaborado pelo autor (2026).

A partir da comparação com a Figura 19, observa-se uma redução significativa na densidade de pontos exibidos, o que resulta em uma leitura mais consistente do comportamento das variáveis monitoradas ao longo do tempo. Esse ajuste não indica perda de informação, mas reflete uma reorganização na forma como os dados são apresentados, mantendo a integridade do registro original armazenado no banco de dados.

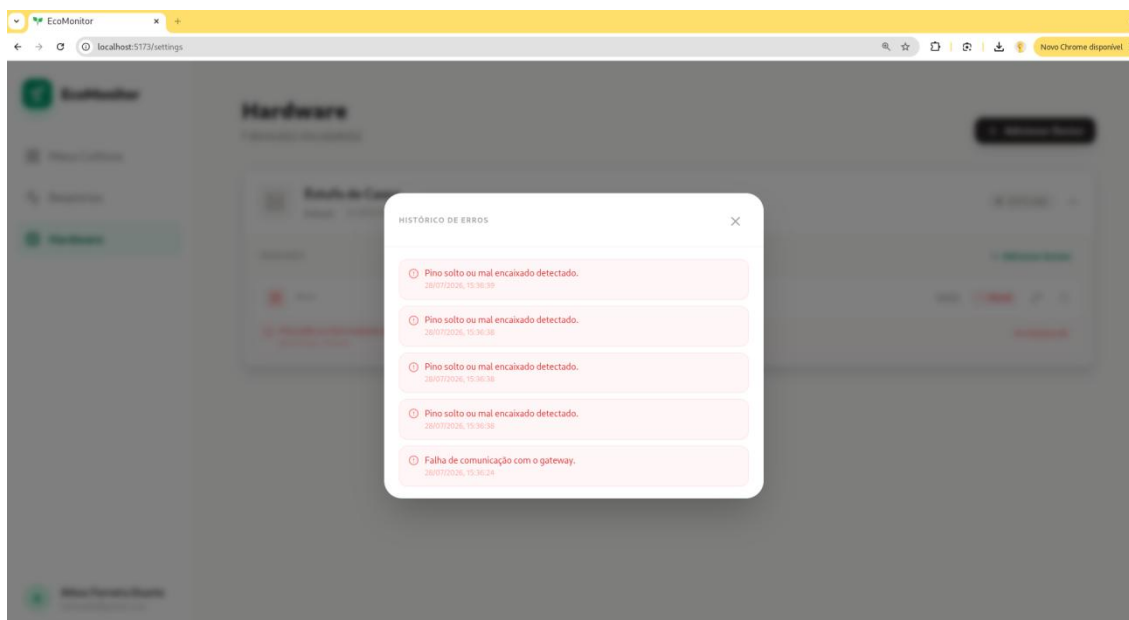
Esse resultado é relevante, pois evidencia que o tratamento adequado do volume de dados contribui diretamente para a interpretação das informações, sem comprometer a rastreabilidade das leituras. A análise desse comportamento reforça a importância de mecanismos que auxiliem na visualização de grandes volumes de dados, especialmente em sistemas de monitoramento contínuo, nos quais a densidade de informações tende a crescer ao longo do tempo.

4.7 OCORRÊNCIA DE FALHAS E LIMITES NO FUNCIONAMENTO

A análise do sistema em operação também permite identificar situações em que as leituras não se comportam conforme o esperado, evidenciando limitações inerentes tanto aos sensores utilizados quanto ao próprio processo de aquisição dos dados. A observação dessas ocorrências é relevante, pois contribui para a compreensão do funcionamento do sistema em condições reais, nas quais interferências e instabilidades podem ocorrer.

A Figura 21 apresenta a interface de visualização de erros associados ao sensores, na qual são registrados eventos que indicam comportamentos inconsistentes nas leituras. Esses registros evidenciam que, em determinados momentos, os dados coletados podem apresentar valores fora do padrão esperado, o que exige atenção quanto à sua interpretação.

Figura 21 – Registro de ocorrências de erro nos sensores



Fonte: Elaborado pelo autor (2026).

A ocorrência dessas falhas pode ser atribuída a vários fatores, sendo um dos principais o erro na montagem do equipamento, podendo

incluir desconexões parciais de cabos, má conexão dos conectores e montagem incorreta dos sensores. Outros fatores que contribuem para o problema são a sensibilidade dos sensores analógicos a interferências elétricas, as flutuações rápidas nas condições ambientais e as instabilidades transitórias na leitura. Esses fatores não estão todos presentes ao mesmo tempo, mas podem ocorrer a qualquer momento durante toda a operação, o que reforça a necessidade de se aplicar procedimentos cuidadosos de montagem e de calibração sobre o sistema.

Assim, se levadas em conta essas limitações, se verificará que a confiabilidade dos dados não está relacionada apenas com a sua capacidade de coleta, mas com a robustez da instalação física e a resposta do sistema diante de eventuais erros. O registro dessas falhas e a sua visualização representam, neste sentido, um recurso valioso, porque viabilizam a verificação do padrão das falhas, a sua contagem, e principalmente, o diagnóstico do que as acarreta, identificando alguma possível má instalação, ou interferência inesperada.

A análise dessas ocorrências contribui para uma compreensão mais realista do sistema desenvolvido, evidenciando que, embora seja capaz de operar de forma contínua, ele ainda está sujeito a limitações decorrentes dos componentes utilizados e das condições de funcionamento. Esse reconhecimento não compromete a proposta do trabalho, mas delimita seu escopo de aplicação, indicando que o sistema se encontra em um estágio de prototipagem que demanda ajustes e aprimoramentos para cenários de uso mais exigentes.

4.8 OTIMIZAÇÕES NA TRANSMISSÃO DE DADOS

A operação contínua do sistema evidenciou limitações relacionadas ao volume de dados transmitidos entre o *gateway* e a infraestrutura em nuvem, especialmente em cenários com maior frequência de coleta. Na configuração inicial, as informações eram enviadas em formato JSON, no qual cada leitura incluía, de forma

redundante, identificadores e metadados necessários à sua interpretação. Embora esse modelo tenha permitido o funcionamento do sistema, o volume gerado por múltiplas leituras sucessivas tornou-se um fator relevante, sobretudo quando considerado o envio contínuo de dados.

Diante dessa limitação, foram implementadas modificações voltadas à reorganização e à redução das informações transmitidas. Uma das primeiras medidas consistiu na utilização de identificadores mais compactos, associados à estrutura de tópicos do protocolo MQTT, o que possibilitou eliminar a repetição de metadados em cada mensagem. Em seguida, o formato JSON foi substituído por uma estrutura binária baseada em Protocol Buffers, reduzindo a necessidade de envio de elementos textuais e tornando os pacotes mais compactos.

Além dessas alterações, foi adotada uma estratégia de agrupamento de múltiplas leituras em um único envio, diminuindo a frequência de comunicação entre o *gateway* e a nuvem. Em complemento, a representação temporal passou a ser realizada com base em diferenças relativas entre leituras consecutivas, reduzindo a quantidade de informação necessária para descrever a sequência dos dados ao longo do tempo. Essas modificações foram implementadas de forma incremental, permitindo observar o impacto de cada decisão sobre o comportamento da transmissão.

Para evidenciar esses efeitos, a Tabela 4 apresenta a evolução do tamanho dos pacotes de dados ao longo das etapas de otimização, considerando o envio de dez leituras em cada cenário analisado.

Tabela 4 – Evolução do tamanho do pacote para envio de 10 leituras

Etapa de Otimização	Custo de Rede	Custo de Dados	Tamanho Total	Redução Acumulada
1. Cenário Original (JSON Puro)	420 bytes	900 bytes	1.320 bytes	–
2. Topic Alias (Remoção de metadados)	420 bytes	470 bytes	890 bytes	~32%
3. Protobuf (Serialização Binária)	420 bytes	150 bytes	570 bytes	~56%
4. Agregação de leituras	42 bytes	150 bytes	192 bytes	~85%
5. Delta Encoding (Diferença Temporal)	42 bytes	100 bytes	142 bytes	~89%

Fonte: Elaborado pelo autor (2026).

A partir dos valores apresentados, observa-se que o modelo inicial demandava 1.320 bytes para a transmissão de dez leituras. Com a aplicação das estratégias adotadas, esse valor foi progressivamente reduzido, atingindo aproximadamente 142 bytes na configuração final. Essa redução não está associada a uma única modificação isolada, mas ao efeito combinado da eliminação de redundâncias, da adoção de um formato binário e da reorganização do envio dos dados.

A análise desses resultados evidencia que a eficiência do sistema não depende apenas da escolha dos protocolos de comunicação, mas também da forma como os dados são estruturados e preparados para transmissão. A redução do volume de dados transmitidos altera de maneira significativa o custo de comunicação, especialmente em cenários com maior número de sensores ou maior frequência de coleta.

Nesse sentido, as otimizações implementadas não apenas melhoraram o desempenho do sistema, mas também contribuíram para a viabilidade da solução no cenário testado, sugerindo potencial de aplicação em contextos similares de restrição de rede e de processamento.

4.9 ESTABILIDADE OPERACIONAL, CONSUMO DE RECURSOS E VIABILIDADE DO SISTEMA

A avaliação do sistema em execução contínua permitiu examinar seu comportamento sob diferentes condições de operação, considerando tanto cenários de maior carga quanto situações próximas ao uso esperado. Para essa análise, foram realizados testes prolongados com o objetivo de observar a estabilidade do processamento, o consumo de recursos computacionais e a capacidade de manter o funcionamento sem interrupções ao longo do tempo. Os principais indicadores de desempenho obtidos nessas avaliações estão sintetizados na Tabela 5.

Tabela 5 – Consumo de recursos e latência do sistema durante a operação

Métrica	Alta Exigência (Múltiplos Sensores)	Uso Prático (Leituras Regulares)
Uso de CPU	2% a 10% (média de 8%)	< 2% (média de 1%)
Consumo de memória	6% a 10% (sem vazamento)	~5% a 6% (alocação base)
Latência	1 a 2 segundos (pico)	< 500 milissegundos

Fonte: Elaborado pelo autor (2026).

Em um cenário de maior exigência, o sistema foi submetido a uma execução contínua com múltiplos sensores virtuais operando simultaneamente. Durante o período analisado, não foram identificadas interrupções no funcionamento nem degradações progressivas no desempenho.

O uso da unidade de processamento manteve-se em níveis reduzidos, com o consumo da CPU oscilando entre 2% e 10%, se mantendo majoritariamente em uma média de 8% ao longo da execução, enquanto o consumo de memória permaneceu estável, entre 6% e 10%, sem indícios de acúmulo ao longo da execução. Esses resultados indicam

que a aplicação foi capaz de sustentar sua operação dentro das limitações do *hardware* adotado, mesmo sob condições ampliadas de uso.

Quando observadas condições mais próximas do uso prático, com menor número de sensores e intervalos regulares de leitura, o sistema apresentou consumo ainda mais reduzido de recursos. A utilização da CPU permaneceu em uma fração da capacidade disponível (inferior a 2%), enquanto a memória utilizada manteve-se em torno de 5% a 6%, compatível com o ambiente embarcado. Esse comportamento reforça a adequação da solução para dispositivos de baixo custo, sem comprometer a execução das rotinas de coleta e transmissão de dados.

A análise do tempo de resposta indica que a disponibilização das leituras ocorre com baixa latência entre a coleta e o envio para a infraestrutura, se mantendo na média abaixo de 500 milissegundos, com a latência máxima de 1 segundo ou de 2 segundos nos piores casos.

A interface, por sua vez, introduz um pequeno atraso intencional relacionado à organização dos dados antes da exibição, através de uma aplicação simples chamada *buffering*, que acumula os dados dentro de um intervalo de tempo, com o objetivo de preservar a legibilidade das informações apresentadas. Essa decisão evidencia um equilíbrio entre atualização contínua e clareza na visualização, ainda que implique uma diferença entre o instante de coleta e o momento de exibição ao usuário.

Ao considerar o funcionamento do sistema de forma integrada, observa-se que a arquitetura implementada atende à proposta de monitoramento ambiental com base em dispositivos de baixo custo, mantendo operação contínua e consumo reduzido de recursos. Ao mesmo tempo, os resultados evidenciam limitações associadas ao estágio atual de desenvolvimento, especialmente no que se refere à distribuição física dos sensores e à adaptação do sistema para cenários com maior número de pontos de coleta.

A partir dessa análise, compreende-se que a viabilidade do sistema está diretamente vinculada ao contexto de aplicação. Em ambientes controlados, com número reduzido de sensores, a solução demonstra comportamento estável e consistente. Em cenários mais

amplos, sua utilização demanda ajustes estruturais relacionados à organização da rede e à forma de coleta dos dados. Esses aspectos não invalidam a proposta desenvolvida, mas delimitam as condições sob as quais o sistema mantém seu desempenho adequado, indicando possibilidades concretas de evolução em trabalhos futuros.

Esse comportamento está alinhado com estudos que apontam a viabilidade de arquiteturas IoT de baixo custo para aplicações de monitoramento ambiental, desde que consideradas as limitações associadas à escalabilidade, à qualidade dos sensores e à infraestrutura de comunicação (Faid; Sadik; Sabir, 2020; Singh *et al.*, 2025).

5 CONCLUSÃO

Este estudo buscou responder como desenvolver um sistema IoT capaz de monitorar condições ambientais relevantes ao cultivo de olerícolas, aplicável a ambientes controlados como estufas, considerando restrições de custo, limitações de *hardware* e a necessidade de comunicação contínua entre os componentes. O objetivo foi atendido por meio da concepção e implementação de uma arquitetura baseada em dispositivos embarcados, integrando sensores analógicos, *gateway* de borda, infraestrutura em nuvem e interface de visualização, traduzida em um sistema funcional e não apenas em uma proposta teórica.

Os resultados confirmam a viabilidade da solução nos cinco eixos que orientaram o trabalho. Quanto ao monitoramento contínuo, o sistema manteve fluxo constante de leituras e atualização da interface sem interrupções, com latência média abaixo de 500 ms. Quanto ao baixo custo, o protótipo foi viabilizado com componentes acessíveis, com custo total estimado entre R\$ 270,00 e R\$ 400,00, evidenciando que a arquitetura dispensa *hardware* industrial de alto valor. Quanto ao desempenho computacional, o uso de CPU se manteve entre 2% e 10% sob carga ampliada (e abaixo de 2% em condições próximas do uso prático), enquanto a memória permaneceu estável entre 6% e 10%, sem indícios de acúmulo, demonstrando que o sistema opera dentro das limitações do *hardware* embarcado mesmo sob maior carga. Quanto à otimização da transmissão de dados, a combinação de eliminação de redundâncias, serialização binária e reorganização do envio reduziu o volume transmitido de 1.320 para aproximadamente 142 bytes a cada dez leituras, uma redução de 89%, com impacto direto sobre custo de comunicação e escalabilidade da solução. Por fim, quanto ao provisionamento dinâmico, a possibilidade de configurar novos sensores diretamente pela interface *web*, sem necessidade de intervenção

estrutural no sistema, reforça a simplicidade de manutenção e expansão da solução.

Apesar desses resultados, o sistema permanece em estágio de prototipagem e foi validado apenas em condições controladas. Limitações como a suscetibilidade dos sensores analógicos a ruídos, a necessidade de aprimorar a distribuição física dos pontos de coleta e a ausência de validação em maior escala delimitam o alcance atual da proposta.

Conclui-se, portanto, que o sistema desenvolvido é uma solução viável de monitoramento ambiental em contextos controlados, articulando sensoramento, processamento e comunicação de forma consistente e com recursos acessíveis. Como desdobramentos, sugere-se a expansão da arquitetura para cenários com maior distribuição física dos sensores, o aprimoramento dos mecanismos de tratamento de dados e a validação em diferentes culturas e sistemas de cultivo.

6 TRABALHOS FUTUROS

Como desdobramento, a continuidade da pesquisa pode avançar em três direções principais: a expansão da arquitetura para cenários com maior distribuição física dos sensores, o aprimoramento dos mecanismos de tratamento e interpretação dos dados, e a validação da arquitetura em diferentes culturas e sistemas de cultivo, como plantio a céu aberto ou outras espécies vegetais além das olerícolas. Esses caminhos não representam correções externas ao trabalho realizado, mas o aprofundamento natural de uma proposta que já demonstrou consistência no recorte em que foi desenvolvida.

REFERÊNCIAS

- AHOA, Emmanuel; KASSAHUN, Ayalew; VERDOUW, Cor et al. Challenges and solution directions for the integration of smart information systems in the agri-food sector. *Sensors*, v. 25, n. 2362, 2025.
- ATZORI, Luigi; IERA, Antonio; MORABITO, Giacomo. The Internet of Things: A survey. *Computer Networks*, v. 54, n. 15, p. 2787–2805, 2010.
- CHAVARE, Sagar M. et al. Smart plant monitoring and automated irrigation system using IoT. *International Journal of Advanced Research in Science, Communication and Technology*, v. 5, n. 4, 2025.
- CHONG, J. L. et al. Internet of Things (IoT)-based environmental monitoring and control system for home-based mushroom cultivation. *Biosensors*, v. 13, n. 1, p. 98, 2023. DOI: 10.3390/bios13010098.
- CHOUDHARY, Vikas. Internet of Things applications in agriculture and environmental monitoring. *Advances in IoT-Based Smart Agriculture*. [S. l.: s. n.], 2024.
- FISSORE, Virginia Isabella; ARCAMONE, Giuseppina; ASTOLFI, Arianna et al. Multi-sensor device for traceable monitoring of indoor environmental quality. *Sensors*, v. 24, n. 2893, 2024. DOI: 10.3390/s24092893.
- FAID, Amine; SADIK, Mohamed; SABIR, Essaid. IoT-based low cost architecture for smart farming. In: IEEE WIRELESS COMMUNICATIONS AND NETWORKING CONFERENCE WORKSHOPS (WCNCW), 2020. *Anais [...]*. [S. l.: s. n.], 2020.
- GAMAL, Y.; SOLTAN, A.; SAID, L. A. et al. Smart irrigation systems: overview. *IEEE Access*, 2023. Disponível em: <https://ieeexplore.ieee.org/document/10057388>. Acesso em: 12 ago. 2026.
- GIL, Antonio Carlos. *Como elaborar projetos de pesquisa*. 4. ed. São Paulo: Atlas, 2008.
- GOOGLE. Protocol Buffers Documentation. Disponível em: <https://protobuf.dev/>. Acesso em: 29 jul. 2026.

GUBBI, Jayavardhana 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.

HERNÁNDEZ-MORALES, Carlos Alberto et al. IoT-based spatial monitoring and environment prediction system for smart greenhouses. *IEEE Latin America Transactions*, v. 21, n. 4, p. 602–611, 2023.

INTERNET ENGINEERING TASK FORCE. The JavaScript Object Notation (JSON) Data Interchange Format. RFC 8259. Edição de: T. Bray. [S. l.], dez. 2017. Disponível em: <https://datatracker.ietf.org/doc/html/rfc8259>. Acesso em: 29 jul. 2026.

LAKATOS, Eva Maria; MARCONI, Marina de Andrade. *Fundamentos de metodologia científica*. 8. ed. São Paulo: Atlas, 2017.

LEA, Perry. *IoT and Edge Computing for Architects: implementing edge and IoT systems from sensors to clouds with communication systems, analytics, and security*. 2. ed. Birmingham: Packt Publishing, 2020.

PELKONEN, T.; FRANKLIN, S.; TELLER, J. et al. Gorilla: A Fast, Scalable, In-Memory Time Series Database. *Proceedings of the VLDB Endowment*, [S. l.], v. 8, n. 12, p. 1816, 2015. Disponível em: <https://www.vldb.org/pvldb/vol8/p1816-teller.pdf>. Acesso em: 29 jul. 2026.

POPIĆ, S.; PEZER, D.; MRAZOVAC, B. et al. Performance evaluation of using protocol buffers in the Internet of Things communication. In: INTERNATIONAL CONFERENCE ON SMART SYSTEMS AND TECHNOLOGIES (SST), 2016, Osijek. *Anais [...]*. [S. l.]: IEEE, 2016. p. 261–265. DOI: 10.1109/SST.2016.7765670. Disponível em: <https://ieeexplore.ieee.org/document/7765670>. Acesso em: 29 jul. 2026.

RAY, Partha Pratim. A survey on Internet of Things architectures. *Journal of King Saud University – Computer and Information Sciences*, v. 30, n. 3, p. 291–319, 2018.

SINGH, Tanya et al. IoT-enabled agronomic systems: real-time field intelligence for smarter farming. *International Journal of Research in Agronomy*, v. 8, n. 6, p. 182–191, 2025.

SOOKHAK, Mehdi et al. Security and privacy of smart cities: a survey. *Future Generation Computer Systems*, v. 99, p. 676–690, 2019.

WAN, Shiqi et al. A modularized IoT monitoring system with edge-computing for aquaponics. *Sensors*, v. 22, n. 23, p. 9260, 2022. DOI: 10.3390/s22239260.