

RICA

v. 1 n.1 2026.
15 de Outubro

REVISTA INTERDISCIPLINAR DE COMPUTAÇÃO APLICADA

UNIVASF



Arte: Canoa no Rio São Francisco
Créditos: Marcus Vinicius Midena Ramos

UNIVASF

RICA - REVISTA INTERDISCIPLINAR DE COMPUTAÇÃO APLICADA — V. 1 N. 1 2026.

Revista acadêmica aberta, com periodicidade semestral, voltada para publicação de artigos científicos teóricos ou práticos e relatos de experiência de trabalhos que possuam a Ciência ou Engenharia da Computação aplicadas de forma direta ou como parte fundamental de ações interdisciplinares em Engenharia, Saúde, Ciências Exatas, Ciências Humanas e Sociais.

COMITÊ EDITORIAL

Prof. Jairson Barbosa Rodrigues, Dr. (editor-chefe)
Prof. Eubis Pereira Machado, Dr.
Prof. Jadsonlee da Silva Sá, Dr.
Prof. Jorge Luis Ramos Cavalcanti, Dr.
Prof. Walter Felipe dos Santos, Dr.
Ana Paula Lopes da Silva, Dra.
Renan Frank Silva Vieira, Bel.

Qualquer parte desta publicação pode ser reproduzida, desde que citada a fonte.

SUMÁRIO

1-10 | HortaTech: uma arquitetura IoT baseada em rede CAN, LoRa e supervisão em nuvem para monitoramento agrícola.

11-22 | Sistema de Monitoramento e Alerta de Temperatura para Compostagem na Agricultura Regenerativa: Integração IoT, Análise de Dados e Automação Operacional.

23-32 | Nintendo Switch como plataforma de computação embarcada: arquitetura e limites de desempenho.

33-40 | Automação do Onboarding Corporativo com Agente Conversacional Baseado em IA: Um Relato de Experiência.

41-54 | Impacto do Tamanho da Amostra de Treinamento e do Vocabulário Para Tarefas de Classificação de Texto.

55-62 | Da engenharia de prompts à mediação pedagógica: avaliação formativa em tempo real com a plataforma EducAI.

Artigo Completo

HortaTech: uma arquitetura IoT baseada em rede CAN, LoRa e supervisão em nuvem para monitoramento agrícola

HortaTech: an IoT Architecture Based on CAN Network, LoRa, and Cloud Supervision for Agricultural Monitoring

Pedro Henrique Ferreira Amorim da Silva [Universidade Federal do Vale do São Francisco | *pedrohenriquefa97@gmail.com*]
Jadsonlee da Silva Sá [Universidade Federal do Vale do São Francisco | *jadsonlee.sa@univasf.edu.br*]

Universidade Federal do Vale do São Francisco, Avenida Antônio Carlos Magalhães, 510, Bairro Santo Antônio, Juazeiro, Bahia, 48902-300, Brasil.

Resumo. Este trabalho apresenta o desenvolvimento, a implementação e a validação experimental em bancada de um sistema de monitoramento para apoio ao manejo hídrico no cultivo de pimentão, com foco em aplicações de agricultura de precisão e Internet das Coisas (IoT). O sistema foi concebido a partir de uma arquitetura distribuída e modular, composta por nós sensores baseados em microcontroladores PIC, comunicação local por rede CAN, transmissão sem fio de longa distância por LoRa e integração com uma plataforma web em nuvem. A aquisição das variáveis ambientais e edáficas foi realizada por sensores de temperatura do solo, umidade do solo, temperatura ambiente e umidade relativa do ar. Os dados coletados pelos nós foram enviados ao barramento CAN, retransmitidos por um *gateway* CAN/LoRa, processados em uma estação base com *backend* em Node.js e armazenados no *Firebase Realtime Database*, sendo posteriormente visualizados em uma interface desenvolvida em Vue.js. Os testes demonstraram a operação integrada do fluxo de dados ponta a ponta, com comunicação CAN configurada em 250 kbps, enlace LoRa validado entre 150 m e 200 m, *payload* médio de aproximadamente 32 *bytes* e taxa efetiva LoRa estimada em 5,4 kbps. O tempo médio de resposta entre a aquisição da leitura e a atualização no *dashboard* permaneceu na faixa de 1,5 s a 2,0 s, enquanto o envio periódico das leituras permitiu o acompanhamento contínuo do microclima e das condições do solo. Embora o protótipo atual tenha sido validado principalmente como plataforma de monitoramento, a arquitetura proposta mostrou-se tecnicamente viável, escalável e preparada para futura integração de válvulas solenoides, sensores de fluxo e estratégias automáticas de irrigação.

Abstract. This paper presents the development, implementation, and bench-level experimental validation of a monitoring system to support water management in bell pepper cultivation, focusing on precision agriculture and Internet of Things (IoT) applications. The system was designed as a distributed and modular architecture composed of PIC-based sensor nodes, local communication through a CAN network, long-range wireless transmission using LoRa, and integration with a cloud-based web platform. Environmental and soil-related variables were acquired using sensors for soil temperature, soil moisture, ambient temperature, and relative air humidity. The data collected by the nodes were transmitted over the CAN bus, forwarded by a CAN/LoRa gateway, processed at a base station using a Node.js backend, stored in the Firebase Realtime Database, and displayed through a Vue.js web interface. The tests demonstrated the integrated end-to-end operation of the data flow, with the CAN network configured at 250 kbps, the LoRa link validated over distances between 150 m and 200 m, an average payload of approximately 32 *bytes*, and an estimated LoRa effective data rate of 5.4 kbps. The average response time from data acquisition to dashboard update remained within the range of 1.5 s to 2.0 s, while the periodic transmission of readings enabled continuous monitoring of microclimate and soil conditions. Although the current prototype was mainly validated as a monitoring platform, the proposed architecture proved to be technically feasible, scalable, and suitable for future integration of solenoid valves, flow sensors, and automatic irrigation strategies.

Palavras-chave: Agricultura, sensores, wireless LoRa, rede CAN, sistema web, cultivo de hortaliças

Keywords: Agriculture, sensors, wireless LoRa, CAN network, web system, vegetable cultivation

Recebido/Received: 15 02 2026 • **Aceito/Accepted:** 08 07 2026 • **Publicado/Published:** 15 10 2026

Publicado sob a Licença Pública Internacional Creative Commons Atribuição 4.0 (CC BY 4.0)

1 Introdução

O agronegócio tem se consolidado na economia brasileira, sendo ele o responsável por elevar o PIB do país, que em 2023 representou 23,8%, de acordo com dados calculados pelo Centro de Estudos Avançados em Economia Aplicada(Cepea/Esalq/USP) em parceria com a Confederação da Agricultura e Pecuária do Brasil (CNA) [CEPEA/ESALQ-USP, 2024].

Dentre as áreas do agronegócio, a que mais vem chamando a atenção dos investidores é a agricultura, sendo responsável por 43,2% das exportações, segundo dados realizado pela Embrapa no ano de 2019 [Embrapa, 2019]. Os avanços tecnológicos na área da agricultura têm ajudado os agricultores no desenvolvimento, porém, tendo em vista a necessidade de altos investimentos muitos agricultores não conseguem acompanhar o mercado, isso faz com que exista um grande

desequilíbrio estrutural entre eles.

Um dos principais desafios que atingem os produtores rurais é a gestão ineficiente dos recursos hídricos, o que resulta em um desperdício significativo de água e compromete a sustentabilidade da produção. Estima-se que 50% desta água não é aproveitada pelas plantas por conta de sistemas inadequados de irrigação ou vazamentos nas tubulações, com grande desperdício de energia e trabalho [Marouelli *et al.*, 2011].

Com o avanço da tecnologia, equipamentos foram criados para minimizar esses prejuízos e maximizar o aproveitamento dos recursos disponíveis. Dentre estes equipamentos, destaca-se o sistema de irrigação, onde a técnica por gotejamento se mostra bastante benéfico para a agricultura em geral.

Diante deste cenário, a resposta para a crescente demanda por alimentos e sustentabilidade reside na adoção de tecnologias inovadoras. É aqui que surge o conceito de Agricultura 4.0 [Pereira *et al.*, 2023], que utiliza a Internet das Coisas (IoT), sensores e automação para criar sistemas de produção inteligentes e baseados em dados [Duraes *et al.*, 2022]. Ambientes controlados, como as estufas para cultivo de hortaliças, representam o campo de provas ideal para essas tecnologias, pois permitem maximizar a eficiência de recursos de forma precisa [Elijah *et al.*, 2018].

Nesse contexto, este trabalho tem como objetivo desenvolver, implementar e validar experimentalmente, em ambiente de bancada, um sistema IoT para monitoramento de variáveis ambientais e edáficas associadas ao cultivo de pimentão. A proposta busca integrar uma rede local de sensoriamento baseada em CAN, comunicação sem fio de longo alcance por LoRa, uma estação base para processamento dos dados e uma plataforma web em nuvem para armazenamento, visualização e acompanhamento das medições.

Diferentemente de soluções baseadas apenas em sensores isolados ou em comunicação sem fio direta, a arquitetura proposta combina uma rede local robusta, adequada à interligação de múltiplos nós sensores, com um enlace LoRa para transmissão dos dados até uma estação de supervisão. Essa abordagem permite separar a comunicação de campo, realizada em uma rede cabeada e tolerante a ruídos, da comunicação de maior alcance, voltada ao envio das informações para processamento e visualização.

As principais contribuições deste trabalho são:

- desenvolvimento de uma arquitetura distribuída e modular para monitoramento agrícola, integrando nós sensores, rede CAN, comunicação LoRa e plataforma em nuvem;
- implementação de nós sensores baseados em microcontroladores PIC para aquisição de variáveis de solo e ambiente, incluindo temperatura do solo, umidade do solo, temperatura ambiente e umidade relativa do ar;
- integração de um *gateway* CAN/LoRa e de uma estação base com *backend* em Node.js para recepção, validação, processamento e envio dos dados ao *Firebase Realtime Database*;
- desenvolvimento de uma aplicação web em Vue.js para autenticação, supervisão e visualização das medições por meio de cartões informativos e gráficos históricos;
- validação experimental do fluxo de dados ponta a ponta, incluindo a análise de métricas como taxa de comunicação CAN, alcance do enlace LoRa, tamanho médio do *payload* e tempo de resposta da interface.

O restante deste artigo está organizado da seguinte forma. A Seção 2 apresenta a fundamentação e as tecnologias aplicadas ao trabalho, incluindo aspectos relacionados ao cultivo, à irrigação, à Agricultura 4.0, à rede CAN, à comunicação LoRa e às plataformas de supervisão. A Seção 3 descreve a arquitetura e a implementação do sistema proposto. A Seção 4 apresenta os resultados experimentais, a discussão das métricas obtidas e as limitações do estudo. Por fim, a Seção 5 apresenta as considerações finais e indica possibilidades de trabalhos futuros.

2 Fundamentação e Tecnologias Aplicadas

Esta seção apresenta os principais conceitos agronômicos e tecnológicos que fundamentam o desenvolvimento do sistema proposto. Inicialmente, são discutidos aspectos relacionados ao ambiente de cultivo, ao solo, à irrigação por gotejamento e à cultura do pimentão, uma vez que essas características orientam a escolha das variáveis monitoradas. Em seguida, são abordadas as tecnologias habilitadoras associadas à Agricultura 4.0, incluindo redes de comunicação, sensoriamento, conectividade de longo alcance e plataformas de supervisão. Dessa forma, busca-se estabelecer a base conceitual necessária para compreender a arquitetura e a implementação apresentadas nas seções seguintes.

2.1 Ambiente de Cultivo e Solo

O sucesso da atividade agrícola é intrinsecamente dependente das características do ambiente de cultivo. Oscilações na temperatura e umidade podem comprometer o desenvolvimento da cultura, ocasionando malformações, perdas na produção e maior incidência de pragas e doenças, o que torna a análise e mitigação desses parâmetros essenciais [Furrier and Ross, 2004]. Neste estudo, utilizou-se como ambiente de cultivo recipientes agrícolas (vasos) contendo a espécie de pimentão.

Nesse contexto, a estufa agrícola destaca-se como um ambiente controlado que permite o manejo preciso de variáveis climáticas, irrigação e umidade, adequando-se às necessidades específicas de hortaliças. Além de oferecer proteção contra insetos-praga e viabilizar o plantio fora de época, o cultivo protegido reduz significativamente a perda de nutrientes nitrogenados por volatilização — que pode chegar a 50% em campo aberto, aumentando a eficiência dos fertilizantes, a lucratividade e a sustentabilidade ambiental [Pereira *et al.*, 2023].

Concomitantemente, a escolha adequada do solo é determinante para a eficiência e sanidade do crescimento vegetal [Souza, 2017]. Características físicas do solo influenciam diretamente o manejo da irrigação; o sistema de gotejamento, embora aplicável a diversos tipos de solo, é particularmente indicado para solos densos e compactos com baixa taxa de infiltração, pois permite a absorção lenta da água [Matias, 2021]. Ademais, o monitoramento da temperatura do solo é uma métrica vital, pois impacta as propriedades biológicas,

químicas e físicas, favorecendo a proliferação de microrganismos benéficos ao cultivo.

2.2 Irrigação por Gotejamento

A irrigação por gotejamento define-se pela aplicação pontual e frequente de água diretamente na zona radicular da planta, utilizando emissores denominados gotejadores dispostos estrategicamente ao longo da extensão do cultivo [Souza and Matsura, 2004]. Diferente dos métodos de superfície ou aspersão, esta técnica fundamenta-se na manutenção de um bulbo úmido constante no solo, o que permite que a planta mantenha seu potencial hídrico otimizado sem a necessidade de ciclos de estresse hídrico entre as regas.

Este método assume relevância estratégica diante do cenário de escassez hídrica global, destacando-se como um recurso de alta eficiência (podendo atingir índices superiores a 90%) que adequa a oferta hídrica às necessidades específicas da cultura. Ao reduzir as perdas por evaporação direta do solo e eliminar a deriva causada pelo vento, o sistema minimiza o desperdício e mitiga impactos ambientais negativos, como a lixiviação de nutrientes para o lençol freático [Pereira *et al*., 2023].

Além disso, ao operar com baixa vazão e alta frequência, o sistema proporciona um controle rigoroso sobre o manejo da água e da fertilidade — técnica conhecida como fertirrigação. Esse monitoramento preciso previne o alagamento e a consequente hipóxia radicular, protegendo a estrutura física do solo contra a degradação e a erosão [Souza and Matsura, 2004]. No contexto da agricultura de precisão, o gotejamento torna-se a infraestrutura ideal para a integração de sensores de umidade e temperatura do solo, permitindo uma automação baseada em dados reais que maximiza a produtividade por gota aplicada.

2.3 A Cultura do Pimentão

O pimentão destaca-se como uma das hortaliças de maior importância econômica na olericultura, sendo a espécie escolhida para este projeto devido ao seu ciclo biológico dinâmico e rápido desenvolvimento. Essa característica é fundamental para experimentações tecnológicas, pois permite observar a resposta da planta a diferentes estímulos hídricos em um curto intervalo de tempo, facilitando a coleta e análise de dados de crescimento.

Além da agilidade no desenvolvimento, a escolha do pimentão justifica-se por sua alta sensibilidade ao manejo ambiental. Por possuir um sistema radicular relativamente superficial, a planta exige uma manutenção rigorosa da umidade do solo; tanto o déficit hídrico quanto o encharcamento podem causar perdas imediatas na produtividade e o surgimento de doenças fisiológicas.

Dessa forma, o pimentão atua como um excelente modelo biológico para validar o sistema de monitoramento proposto, uma vez que o controle preciso das variáveis de umidade e temperatura do solo através de sensores é determinante para suprir suas exigências fisiológicas e garantir a qualidade final do fruto [Souza and Matsura, 2021; Lopes and Silva, 2022].

2.4 Agricultura 4.0 e Tecnologias Habilitadoras

A gestão manual de sistemas agrícolas, mesmo em ambientes controlados, permanece suscetível a ineficiências. A superação desses desafios reside na Agricultura 4.0, que integra tecnologias digitais — como *IoT*, Conectividade, *Cloud Computing*, *Big Data* e Automação — para criar ecossistemas inteligentes e transitar da Agricultura de Precisão para uma Agricultura Preditiva e Digital [Lisbinski *et al*., 2020].

A materialização desses conceitos através da automação de sistemas de irrigação proporciona benefícios que variam da economia de recursos hídricos e tempo à redução da intervenção humana. A automatização eleva a eficiência na aplicação de água e nutrientes, mitigando desperdícios e otimizando a produtividade da cultura ao controlar com precisão os insumos e evitar a sub ou superirrigação [dos Santos *et al*., 2014].

2.4.1 Arquitetura de Comunicação: Rede LoRa

No âmbito da conectividade, destaca-se a tecnologia LoRa (*Long Range*), uma técnica de modulação sem fio projetada para alcançar grandes distâncias (até 30 km em condições ideais) com baixo consumo energético e alta capacidade de penetração de sinal. Essas características a tornam ideal para aplicações de IoT em ambientes rurais e urbanos, servindo de base para o protocolo de rede LoRaWAN [Alipio *et al*., 2024]. No Brasil, a tecnologia foi regulamentada pela ANATEL em 2017 (Ato 14.448), operando no padrão de 915 MHz a 928 MHz [Ortiz *et al*., 2019].

2.4.2 Rede CAN

Controller Area Network (CAN) é um padrão de barramento serial de dois fios, desenvolvido pela *Bosch* na década de 1980, originalmente para a indústria automotiva, visando a interconexão rápida e segura entre microcontroladores com taxas de transmissão de até 1 Mbps [Robert Bosch GmbH, 1991]. O protocolo estrutura-se em camadas (objeto, transferência e física), diferenciando-se do modelo OSI por focar, em sua camada física, na temporização e sincronização dos bits [Di Natale *et al*., 2012].

A robustez da rede CAN reside em seu sistema de mensagens (dados, erro, requisição e sobrecarga) e no método de arbitragem CSMA/DCR (*Carrier Sense Multiple Access/Deterministic Collision Resolution*). Este mecanismo prioriza o acesso ao barramento com base em identificadores de mensagem, evitando colisões e garantindo que módulos prioritários transmitam dados críticos sem atrasos, o que confere ao sistema baixo custo, alta precisão e confiabilidade operacional [de Souza and Campos, 2017].

2.4.3 Plataformas de Supervisão e Controle

As plataformas de supervisão e controle desempenham papel fundamental em sistemas de agricultura de precisão, pois integram dispositivos de campo, serviços de armazenamento, mecanismos de processamento e interfaces de visualização. Em sistemas agrícolas baseados em Internet das Coisas, essas plataformas permitem que dados provenientes de sensores sejam organizados, armazenados e apresentados ao usuário de forma compreensível, apoiando o acompanhamento das condições de cultivo e a tomada de decisão [García-García

et al., 2020; Kamienski *et al*., 2019].

No contexto da irrigação inteligente, plataformas *IoT* geralmente são organizadas em camadas, envolvendo dispositivos de sensoriamento, comunicação, infraestrutura em nuvem e interfaces de supervisão. Essa estrutura permite que variáveis como temperatura, umidade relativa do ar, umidade do solo e outros parâmetros ambientais sejam monitorados em tempo real, possibilitando a análise das condições de cultivo e o planejamento de intervenções de irrigação [Morchid *et al*., 2024].

Além da visualização instantânea das medições, plataformas supervisórias podem incorporar recursos como histórico de dados, geração de gráficos, configuração de limiares operacionais, notificações e suporte à automação. Esses recursos são relevantes para a transição de sistemas meramente informativos para sistemas de apoio à decisão ou controle automático, nos quais atuadores, como bombas e válvulas de irrigação, podem ser acionados com base em regras previamente definidas ou em algoritmos de controle [Maia *et al*., 2020; Morchid *et al*., 2024].

Outro aspecto importante refere-se aos requisitos de operação dessas plataformas em ambientes agrícolas. Sistemas aplicados ao campo devem ser escaláveis, flexíveis, capazes de lidar com diferentes tipos de sensores e robustos diante de falhas de comunicação ou de dispositivos. Tais requisitos são especialmente importantes em aplicações rurais, nas quais a conectividade pode ser instável e os dispositivos estão sujeitos a condições ambientais adversas [Maia *et al*., 2020].

Nesse sentido, a camada de supervisão constitui um elemento estratégico para a adoção de tecnologias digitais no campo, pois aproxima o usuário final dos dados produzidos pelos sensores e favorece a tomada de decisão baseada em evidências. Assim, sua presença em sistemas agrícolas inteligentes contribui para a rastreabilidade das informações, para o acompanhamento das condições de cultivo e para a melhoria da eficiência no uso de recursos hídricos.

3 Arquitetura e Implementação do Sistema

A arquitetura do sistema proposto foi desenvolvida para operar como uma plataforma modular de monitoramento agrícola, integrando sensoriamento distribuído, comunicação local, transmissão sem fio, armazenamento em nuvem e visualização web. Conforme ilustrado na Figura 1, a solução é organizada em três subsistemas: o Subsistema 1, responsável pela rede local de sensoriamento baseada em CAN; o Subsistema 2, composto pelo *gateway* CAN/LoRa, pela estação base e pela aplicação de supervisão; e o Subsistema 3, associado à persistência e sincronização dos dados em nuvem.

O protótipo foi concebido para monitorar variáveis de solo e ambiente no cultivo de pimentão em laboratório, nas dependências da UNIVASF, Campus Juazeiro/BA. Embora a arquitetura tenha sido planejada para permitir a futura integração de válvulas solenoides e sensores de fluxo, a implementação validada neste trabalho concentrou-se na aquisição, transmissão, armazenamento e visualização dos dados.

3.1 Subsistema 1: rede de sensoriamento CAN

O Subsistema 1 corresponde à rede local de aquisição de dados. Sua arquitetura foi baseada em uma topologia de barramento CAN, na qual os nós sensores são conectados a um mesmo meio físico e transmitem dados provenientes dos sensores a uma taxa de 250 kbps. Cada nó foi implementado em uma placa de circuito impresso dedicada, contendo unidade de processamento, interface de comunicação e circuitos de condicionamento para os sensores.

As placas foram alimentadas por uma fonte de 12 V/3,5 A, com regulação local para 5 V, tensão utilizada pelo microcontrolador e pelos componentes eletrônicos. A unidade de processamento de cada nó foi baseada no microcontrolador PIC18F2580 [Inc., 2024], o qual possui um controlador CAN integrado. A interface com o barramento físico foi realizada por meio do transceptor MCP2551 [Inc., 2018].

Na configuração implementada, foram previstos quatro nós na rede. Os nós 2 e 3 foram instrumentados para monitoramento do solo, enquanto o nó 4 foi dedicado ao monitoramento das condições ambientais. O nó 1 foi mantido como ponto de expansão da rede, demonstrando a possibilidade de incorporar novos pontos de medição sem alterar a estrutura principal do sistema.

3.1.1 Nós de monitoramento do solo

Os nós de monitoramento do solo foram projetados para aferir variáveis edáficas relevantes ao manejo hídrico do pimentão. Cada nó de solo é composto por um microcontrolador PIC18F2580, um sensor DS18B20 [Integrated, 2015] para medição da temperatura do solo e um sensor de umidade do solo Watermark 200SS [IRROMETER Company, Inc., 2026] para medição da umidade do solo. O DS18B20 fornece leituras digitais de temperatura na região próxima à zona radicular, enquanto o sensor *Watermark 200SS* é um sensor matricial granular utilizado para estimar a tensão da água no solo. Seu princípio de funcionamento baseia-se na variação da resistência elétrica em uma matriz granular, cuja resposta está relacionada ao potencial hídrico do solo.

3.1.2 Nó de monitoramento do ambiente

O nó 4 foi implementado para fornecer informações macroclimáticas do ambiente de cultivo. Para isso, foi utilizado o sensor DHT22, responsável pela medição da temperatura ambiente e da umidade relativa do ar [ASAIR, 2021]. Essas informações complementam as leituras de solo e permitem relacionar as condições edáficas às condições gerais do ambiente.

O posicionamento do nó ambiental foi planejado para reduzir distorções causadas pela proximidade com o solo, com as plantas ou com fontes localizadas de aquecimento. Dessa forma, as medições desse nó podem servir como referência para análises futuras envolvendo evapotranspiração, conforto climático e resposta da cultura às condições ambientais.

3.1.3 Protótipo físico

A Figura 2 apresenta o protótipo físico do sistema de sensoriamento. A implementação foi dimensionada para operar com alimentação de 12 V DC, característica que favorece sua futura integração com baterias ou sistemas de energia solar

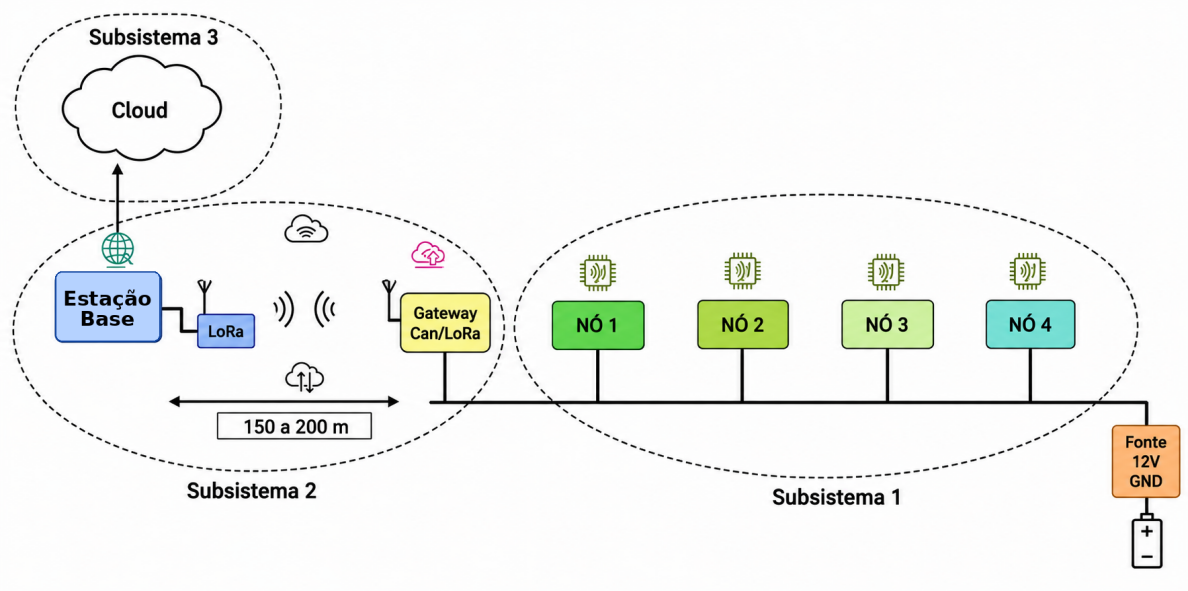


Figura 1. Arquitetura do sistema proposto.

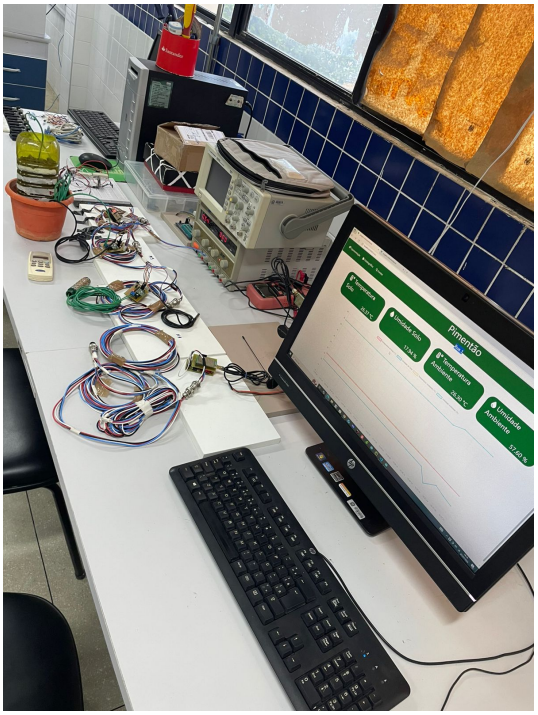


Figura 2. Protótipo físico do sistema de sensoriamento.

em aplicações de campo.

A concepção original dos nós considerou a possibilidade de evolução para um sistema de controle em malha fechada. Para a cultura do pimentão, foram adotados como referência valores de temperatura do solo entre 20°C e 30°C e umidade do solo entre 60% e 80%. A lógica prevista permitiria acionar uma válvula solenoide em situações nas quais as leituras ultrapassassem limites configurados. No entanto, a integração física da válvula e do sensor de fluxo não foi incluída nesta versão do protótipo, permanecendo como trabalho futuro.

Os *firmwares* dos nós foram desenvolvidos em linguagem C, utilizando a *IDE PIC C Compiler*, e gravados nos microcontroladores com o programador PICKit 2 [Inc., 2005]. Cada nó recebeu uma lógica específica de leitura e transmissão, de acordo com sua função na rede.

3.2 Subsistema 2: *gateway* CAN/LoRa e estação base

O Subsistema 2 realiza a integração entre a rede local de sensoriamento e a plataforma de supervisão. Ele é composto por um gateway CAN/LoRa, responsável por receber mensagens do barramento CAN e retransmiti-las via LoRa para uma estação base, responsável por receber os dados sem fio, processá-los e encaminhá-los à nuvem.

A comunicação foi organizada em três etapas. Na primeira, os nós sensores enviam suas leituras ao barramento CAN utilizando identificadores de mensagem associados a cada variável monitorada. Na segunda, o *gateway* CAN/LoRa escuta as mensagens do barramento, empacota os dados relevantes e os transmite por rádio frequência. Na terceira, a estação base recebe os pacotes LoRa por meio de um módulo conectado via USB ao computador.

3.2.1 Estação base e integração IT/OT

O *Gateway PC* representa o ponto de integração entre a tecnologia da operação (OT), composta pelos sensores, rede CAN e enlace LoRa, e a tecnologia da informação (IT), composta pelo *backend*, banco de dados em nuvem e interface web. Um módulo receptor LoRa conectado ao computador entrega os dados recebidos à porta serial, onde são tratados pela aplicação *backend*.

No computador da estação base, a aplicação desenvolvida em Node.js realiza a leitura da porta serial, valida e transforma os dados recebidos, agrupa as leituras e as envia ao *Firebase*. O mesmo ambiente computacional pode hospedar a aplicação web em Vue.js, utilizada como Interface Homem-Máquina (IHM) para visualização dos dados.

3.3 Subsistema 3: plataforma em nuvem

O Subsistema 3 foi implementado com o *Firebase Realtime Database*, responsável pela persistência e sincronização das leituras. O banco de dados foi organizado em estrutura NoSQL, no formato de árvore JSON, permitindo associar cada leitura ao respectivo nó, variável monitorada e instante de aquisição.

A organização adotada seguiu uma estrutura do tipo `UsersData/NodeID/readings/timestamp`. Essa forma de armazenamento favorece a rastreabilidade das medições e permite que a aplicação web recupere tanto valores atuais quanto registros históricos. Além disso, a natureza reativa do *Firebase* permite que alterações no banco sejam refletidas automaticamente na interface, reduzindo a necessidade de implementação manual de mecanismos de atualização por *polling* ou *WebSockets*.

3.4 *Backend* e fluxo de dados

O *backend* foi desenvolvido como um serviço em Node.js responsável pela ingestão, validação, transformação e persistência dos dados recebidos pela estação base. A comunicação com o módulo LoRa conectado ao computador foi realizada por meio da biblioteca `serialport`, com segmentação do fluxo por quebras de linha para permitir o processamento individual das mensagens.

Para reduzir a possibilidade de armazenamento de dados incorretos, foi implementado um filtro de validação baseado em expressão regular. Apenas mensagens compatíveis com o protocolo definido para identificação do nó e das grandezas medidas são aceitas pelo sistema. Após a validação, o *backend* aplica conversões específicas conforme o identificador do nó e a variável recebida, normalizando os valores antes do envio ao banco de dados.

Também foi implementada uma lógica de envio por rodadas, com armazenamento temporário da leitura mais recente de cada nó em um buffer. Quando uma rodada válida é completada, os dados são enviados em lote ao *Firebase*, e o sistema aguarda um intervalo antes de novo envio. Essa estratégia reduz escritas excessivas no banco de dados e mantém uma atualização periódica adequada ao comportamento das variáveis agrícolas monitoradas.

3.5 Aplicação web de supervisão

A aplicação web foi desenvolvida com o *framework Vue.js* e constitui a camada de visualização e interação do sistema. Sua função é permitir que o usuário acompanhe as variáveis monitoradas, visualize históricos de medições e acesse páginas de configuração preparadas para futuras funcionalidades de controle. A aplicação foi organizada como uma *Single Page Application* (SPA), proporcionando navegação fluida e atualização dinâmica dos dados sincronizados no *Firebase*.

3.5.1 Autenticação e página inicial

O acesso ao sistema ocorre por meio de uma tela de autenticação, ilustrada na Figura 3. A implementação utilizou o *Firebase Authentication* para gerenciamento de credenciais e controle de acesso, restringindo a visualização dos dados a usuários autorizados.

Após a autenticação, o usuário é direcionado à página inicial, mostrada na Figura 4. Essa página funciona como painel de navegação principal, permitindo acesso às ruas de monitoramento e às páginas de configuração.

3.5.2 Dashboard de monitoramento

Os *dashboards* de monitoramento constituem o principal recurso funcional da aplicação. Eles são acessados a partir da página inicial e representam diferentes ruas de plantio. Cada *dashboard* apresenta cartões informativos com as leituras atuais e gráficos históricos das variáveis monitoradas, como mostrado na Figura 5.

A atualização dos cartões e gráficos é realizada por meio do método `onValue` do *Firebase Realtime Database*. Esse método estabelece uma conexão reativa com os nós de dados, permitindo que a interface seja atualizada automaticamente sempre que novas leituras são inseridas no banco.

3.5.3 Página de configuração

A página de configuração foi projetada como infraestrutura inicial para a transição do sistema de monitoramento para um sistema com controle ativo. Nessa interface, são previstos campos para definição de limiares operacionais relacionados às variáveis monitoradas, como umidade e temperatura do solo e do ambiente.

Embora o protótipo validado opere em modo de monitoramento, a leitura e a escrita de configurações no *Firebase* validam o caminho para futuras implementações nas quais alterações realizadas pelo usuário possam ser interpretadas pelo *backend* e repassadas ao *hardware* de campo, permitindo o acionamento de atuadores.

3.6 Procedimento de validação experimental

A validação experimental do protótipo foi conduzida em ambiente de laboratório, com o objetivo de verificar a integração entre os nós sensores, a rede CAN, o enlace LoRa, a estação base e a plataforma *web* em nuvem. Os ensaios foram realizados utilizando recipientes com solo e mudas de pimentão, de modo a reproduzir, em escala reduzida, uma condição de cultivo monitorado.

O procedimento de teste foi organizado em quatro etapas. Na primeira etapa, verificou-se individualmente a leitura dos sensores de temperatura do solo, umidade do solo, temperatura ambiente e umidade relativa do ar. Na segunda etapa, avaliou-se a comunicação local entre os nós sensores e o *gateway* por meio da rede CAN, configurada para operar a 250 kbps. Na terceira etapa, os dados recebidos pelo *gateway* CAN/LoRa foram retransmitidos por comunicação LoRa até a estação base, sendo o enlace validado em uma distância aproximada entre 150 m e 200 m. Na quarta etapa, os dados recebidos pela porta serial do computador foram processados pela aplicação Node.js, armazenados no *Firebase Realtime Database* e exibidos no *dashboard web* desenvolvido em *Vue.js*.

Para avaliar o desempenho do sistema, foram considerados os seguintes indicadores: taxa nominal de transmissão da rede CAN, taxa efetiva estimada do enlace LoRa, tamanho médio do pacote transmitido, alcance experimental do enlace sem fio, intervalo de atualização das leituras e tempo de resposta ponta a ponta. O tempo de resposta foi definido como o intervalo aproximado entre a aquisição da leitura pelo nó sensor e a atualização visual correspondente na interface *web*.

4 Resultados e Discussões

Esta seção apresenta os resultados obtidos a partir da validação experimental do sistema proposto, considerando a operação integrada dos nós sensores, da rede CAN, do enlace LoRa, da estação base, do banco de dados em nuvem e da interface *web* de supervisão. Diferentemente da seção anterior, que

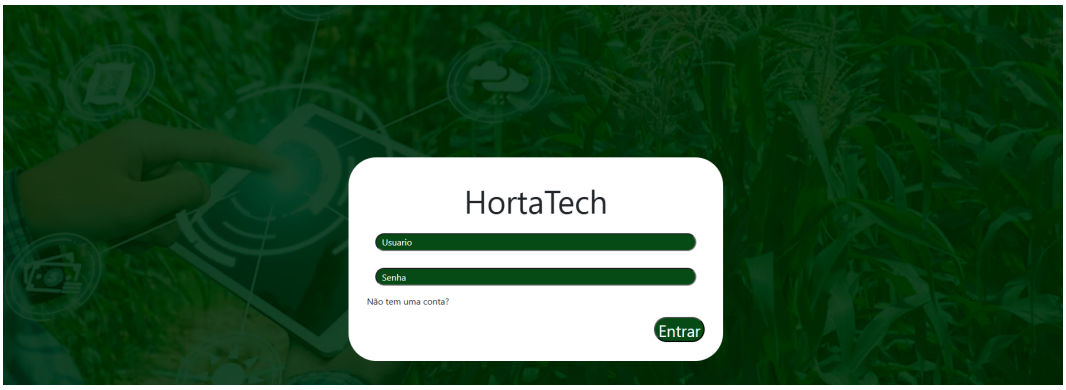


Figura 3. Interface de login do sistema.



Figura 4. Página inicial da aplicação *web*.

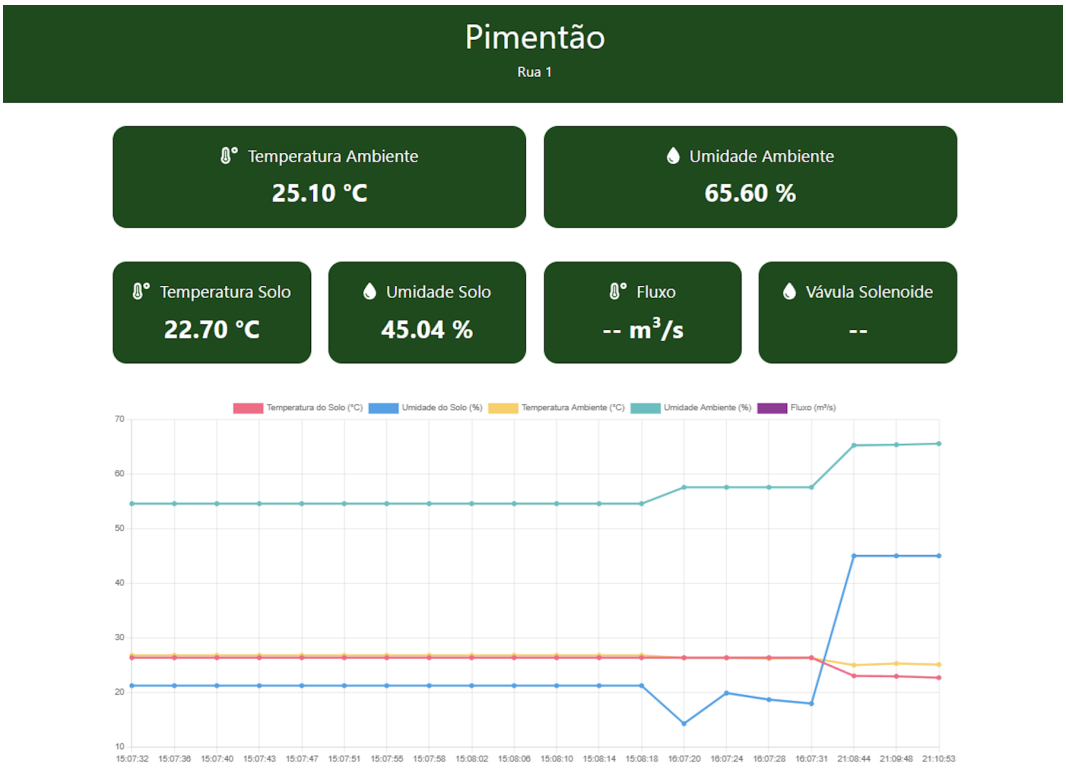


Figura 5. *Dashboard* de monitoramento das variáveis do sistema.

descreveu a arquitetura e a implementação dos subsistemas, a análise apresentada a seguir concentra-se nos aspectos mensuráveis do protótipo, na funcionalidade observada durante os testes e nas limitações identificadas.

4.1 Validação funcional do protótipo

A validação funcional demonstrou que o sistema foi capaz de realizar o fluxo completo de aquisição, transmissão, processamento, armazenamento e visualização dos dados. As leituras de temperatura do solo, umidade do solo, temperatura ambiente e umidade relativa do ar foram coletadas pelos nós sensores, transmitidas pela rede CAN até o *gateway*, retransmitidas por LoRa até a estação base, processadas pelo *backend* e disponibilizadas na interface *web*.

Durante os testes, observou-se que os dados recebidos foram organizados por nó e por variável monitorada, permitindo sua exibição em cartões informativos e em gráficos históricos no *dashboard*. Esse resultado confirma a integração entre as camadas de *hardware*, comunicação e *software*, demonstrando que a arquitetura proposta é funcional para aplicações

de monitoramento agrícola em escala de protótipo.

4.2 Desempenho da comunicação CAN–LoRa

Os testes realizados permitiram quantificar parâmetros importantes de desempenho da comunicação e da atualização dos dados. A Tabela 1 apresenta os principais indicadores observados durante a validação experimental.

Tabela 1. Indicadores quantitativos observados na validação do sistema.

Indicador	Valor observado/-configurado	Interpretação
Taxa da rede CAN	250 kbps	Valor suficiente para o volume de dados dos sensores e adequado para expansão da rede local.
Taxa efetiva LoRa	$\approx$ 5,4 kbps	Compatível com o *payload* reduzido do sistema e com aplicações de telemetria agrícola.
Payload médio	$\approx$ 32 *bytes*	Pacote compacto, adequado para transmissão periódica de leituras ambientais.
Alcance LoRa validado	150 m a 200 m	Comunicação sem fio adequada para separar a área monitorada da estação base.
Tempo de resposta ponta a ponta	1,5 s a 2,0 s	Intervalo entre aquisição, transmissão, processamento, armazenamento e atualização do *dashboard*.
Intervalo de envio das leituras	configurável; testes com atualização periódica	Permite ajustar a frequência de coleta conforme a necessidade agronômica e o custo de armazenamento.
Número de nós implementados	quatro nós previstos, com nós ativos de solo e ambiente	Arquitetura modular, preparada para expansão horizontal.

A comunicação CAN apresentou desempenho estável durante os testes, sem perdas perceptíveis de dados na integração entre os nós sensores e o *gateway*. A adoção desse protocolo mostrou-se adequada para a rede local de campo, pois sua sinalização diferencial, seus mecanismos de arbitragem e sua verificação de erros favorecem a operação em ambientes sujeitos a interferências eletromagnéticas.

O enlace LoRa, por sua vez, demonstrou ser apropriado para a comunicação entre o *gateway* e a estação base. Embora a taxa de transmissão efetiva seja inferior à da rede CAN, o tamanho reduzido dos pacotes transmitidos torna essa limitação pouco relevante para a aplicação proposta. Como os dados de temperatura e umidade variam lentamente no contexto do monitoramento agrícola, a prioridade do enlace não é a alta vazão, mas sim o alcance, a robustez e o baixo consumo energético.

4.3 Desempenho da plataforma *web*

A integração com o *Firebase Realtime Database* permitiu que as leituras fossem sincronizadas com a aplicação *web* de forma reativa. Essa estratégia eliminou a necessidade de consultas periódicas tradicionais ao servidor e permitiu que o *dashboard* fosse atualizado automaticamente sempre que novos dados fossem inseridos no banco.

Nos testes realizados, o tempo médio de resposta de 1,5 s a 2,0 s mostrou-se adequado para uma aplicação de monitoramento agrícola, especialmente porque as variáveis analisadas não exigem tempos de atuação na ordem de milissegundos. Do ponto de vista funcional, o sistema permitiu a visualização das leituras atuais em cartões informativos e o acompanhamento histórico das variáveis por meio de gráficos.

Essa organização facilitou a identificação de tendências, como variações de umidade do solo, oscilações de temperatura e mudanças na umidade relativa do ar. Assim, a interface desenvolvida atua como ferramenta de apoio à decisão, permitindo que o operador acompanhe as condições do cultivo e planeje intervenções de irrigação com maior base quantitativa.

4.4 Análise crítica dos resultados

Os resultados obtidos indicam que a principal contribuição do trabalho está na integração funcional de uma arquitetura híbrida de comunicação, combinando rede CAN para a camada local de sensoriamento e LoRa para o enlace de longa distância. Essa combinação apresenta mérito técnico para aplicações agrícolas, pois separa a rede de campo, que exige robustez e organização determinística das mensagens, da rede sem fio, que exige maior alcance e menor dependência de infraestrutura de internet no local de cultivo.

A adoção da rede CAN mostrou-se especialmente adequada para a conexão entre os nós sensores, pois permite a expansão do sistema por meio da inclusão de novos nós no barramento. Além disso, o uso de identificadores de mensagem facilita a diferenciação entre leituras de diferentes sensores e diferentes ruas de plantio. Essa característica é importante para aplicações futuras em áreas maiores, nas quais múltiplos pontos de medição podem ser distribuídos ao longo da estufa ou do campo.

A comunicação LoRa complementou a arquitetura ao permitir que os dados da rede local fossem enviados a uma estação base localizada a uma distância de até 200 m durante os testes. Esse resultado é relevante para o contexto rural, no qual frequentemente há limitação de infraestrutura de rede cabeada ou Wi-Fi. A taxa efetiva estimada de 5,4 kbps foi suficiente para o envio dos pacotes de telemetria, uma vez que as leituras de temperatura e umidade possuem baixo volume de dados e não exigem transmissão contínua em alta frequência.

A interface *web* desenvolvida também se mostrou adequada ao objetivo do sistema. A organização dos dados em cartões e gráficos permite que o usuário interprete rapida-

mente o estado do cultivo. Além disso, a arquitetura baseada em *Firebase* e *Vue.js* favoreceu a atualização em tempo real da interface, com latência média compatível com a dinâmica das variáveis monitoradas.

Apesar desses resultados, é importante destacar que a validação realizada ainda se concentrou na funcionalidade do sistema de monitoramento e na comunicação ponta a ponta. A etapa de controle físico da irrigação, incluindo o acionamento da válvula solenoide e a medição do volume de água por sensor de fluxo, não foi implementada no protótipo validado. Dessa forma, ainda não foi possível quantificar experimentalmente a economia de água em comparação ao manejo manual. Essa limitação não compromete a validade da arquitetura proposta, mas delimita o alcance dos resultados apresentados neste estudo.

4.5 Limitações do estudo

Embora o sistema tenha demonstrado viabilidade técnica, algumas limitações devem ser consideradas. A primeira está relacionada ao ambiente de validação. Os testes foram realizados em bancada e em ambiente laboratorial, utilizando recipientes com solo e cultivo de pimentão em escala reduzida. Portanto, os resultados ainda não representam integralmente as condições de uma estufa agrícola em operação contínua, sujeita a variações climáticas mais intensas, interferências eletromagnéticas, umidade elevada, poeira e maior distância entre os nós.

A segunda limitação refere-se à ausência de validação agronômica de longo prazo. O sistema monitorou variáveis relevantes para o manejo hídrico, mas não foi realizado um ciclo completo de cultivo comparando produtividade, consumo de água ou consumo de energia entre o manejo manual e o manejo apoiado pelo sistema. Assim, embora a arquitetura seja compatível com estratégias de economia hídrica, os ganhos em termos de redução de consumo de água ainda precisam ser medidos experimentalmente.

A terceira limitação está associada ao subsistema de atuação. A interface *web* e a estrutura lógica foram projetadas para futura integração de válvulas solenoides e sensores de fluxo, porém o acionamento físico da irrigação não foi validado nesta etapa. Consequentemente, o sistema atual deve ser caracterizado como uma plataforma de monitoramento com suporte à decisão, e não como um sistema de irrigação totalmente automatizado.

Por fim, a calibração dos sensores deve ser aprofundada em trabalhos futuros. Embora os sensores utilizados sejam adequados para prototipagem e validação funcional, uma avaliação mais robusta deve incluir comparação com instrumentos de referência, análise de erro, repetibilidade das medições e estabilidade das leituras ao longo do tempo.

5 Considerações finais

Este trabalho apresentou o desenvolvimento e a validação experimental de um sistema de monitoramento para apoio ao manejo hídrico no cultivo de pimentão, integrando sensores, microcontroladores, rede CAN, comunicação LoRa, *backend* em Node.js, banco de dados em nuvem *Firebase* e interface *web* em *Vue.js*. A arquitetura proposta foi concebida de forma modular, permitindo a expansão do número de nós sensores e a futura inclusão de atuadores para automação da irrigação.

Os testes realizados demonstraram a operação integrada do sistema, desde a aquisição das leituras pelos nós sensores até a visualização em tempo real no *dashboard web*. A rede CAN foi configurada em 250 kbps e mostrou-se adequada para a comunicação local entre os nós. O enlace LoRa foi validado em uma distância aproximada entre 150 m e 200 m, com *payload* médio de 32 *bytes* e taxa efetiva estimada em 5,4 kbps. A atualização da interface apresentou tempo médio de resposta entre 1,5 s e 2,0 s, valor considerado satisfatório para o monitoramento de variáveis agrícolas de variação lenta, como temperatura e umidade do solo e do ambiente.

A principal contribuição do trabalho está na demonstração de uma arquitetura híbrida CAN–LoRa aplicada ao monitoramento agrícola, combinando robustez na comunicação local, alcance sem fio para transmissão dos dados e visualização remota em ambiente *web*. Essa integração torna o sistema uma base promissora para aplicações de agricultura de precisão em pequenas propriedades, estufas e ambientes de cultivo protegido.

Como limitação, destaca-se que a validação foi realizada em ambiente de laboratório e que o acionamento físico da irrigação ainda não foi implementado. Dessa forma, não foi possível quantificar, nesta etapa, a economia efetiva de água ou energia em comparação ao manejo manual. Trabalhos futuros devem incluir a implementação da válvula solenoide e do sensor de fluxo, a calibração sistemática dos sensores, a realização de testes em estufa durante um ciclo completo de cultivo e a comparação quantitativa do consumo de água antes e depois da automação.

Mesmo com essas limitações, os resultados obtidos indicam que a arquitetura proposta é tecnicamente viável, escalável e adequada para evoluir de uma plataforma de monitoramento para um sistema automatizado de irrigação baseado em dados.

Declarações complementares

Contribuições dos autores

Ambos os autores contribuíram para a concepção, metodologia e validação deste estudo, desenvolvimento e arquitetura de hardware. PHFAS desenvolveu o software da plataforma e realizou os experimentos. JSS atuou na supervisão e orientação geral do projeto. Ambos os autores realizaram conjuntamente a análise formal dos dados obtidos e a redação da conclusão. PHFAS é o principal contribuidor e escritor deste manuscrito.Todos os autores leram e aprovaram o manuscrito final.

Conflitos de interesse

Os autores declaram que não há conflitos de interesse.

Disponibilidade de artefatos de pequisa

Os dados e outros materiais criados e/ou usados neste artigo serão disponibilizados mediante solicitação.

Uso de Inteligência Artificial

Os autores utilizaram o ChatGPT para revisão textual.

Outras informações relevantes

O manuscrito não precisou de aceite em comitê de ética, visto que não foram utilizados animais e nem pessoas durante a pesquisa.

Referências

Alipio, M. *et al*. (2024). A review of lorawan performance optimization through cross-layer approaches. *Internet of Things*.

ASAIR (2021). Data sheet DHT20. `https://aqicn.org/air/sensor/spec/asair-dht20.pdf`. Acessado em: 16-06-2024.

CEPEA/ESALQ-USP (2024). Pib do agronegócio fecha 2023 com queda de 2,99%. Technical report, Centro de Estudos Avançados em Economia Aplicada, Piracicaba.

de Souza, A. G.; Campos, G. L. (2017). Rede can veicular: levantamento bibliográfico e apresentação de conceitos iniciais. *ForScience*, 5(1).

Di Natale, M., Zeng, H., Giusto, P., and Ghosal, A. (2012). *Understanding and using the controller area network communication protocol: theory and practice*. Springer Science & Business Media.

dos Santos, C. M., COSTA, B., SILVA, R., and Scalassara, P. R. (2014). Desenvolvimento de um módulo de controle de nível utilizando o kit arduino uno. In *XX Congresso Brasileiro de Automática. Belo Horizonte, Brasil*.

Duraes, W., Ferreira, F. H. I. B., and Manzan, R. (2022). *Arquitetura de soluções IoT: Desenvolva com Internet das Coisas para o mundo real*. Casa do Código.

Elijah, O., Rahman, T. A., Orikumhi, I., Leow, C. Y., and Hindia, M. N. (2018). An overview of internet of things (iot) and data analytics in agriculture: Benefits and challenges. *IEEE Internet of Things Journal*, 5(5):3758–3773. DOI: 10.1109/JIOT.2018.2844392.

Embrapa (2019). A agricultura brasileira - portal embrapa. `https://www.embrapa.br/vii-plano-diretor/a-agricultura-brasileira`. Acessado em: 30-04-2024.

Furrier, M. and Ross, J. L. S. (2004). Aplicação da técnica de plantio direto em plantação de café localizada em ambiente de alta fragilidade.

García-García, L., Parra, L., Jimenez, J. M., Lloret, J., and Lorenz, P. (2020). Iot-based smart irrigation systems: An overview on the recent trends on sensors and iot systems for irrigation in precision agriculture. *Sensors*, 20(4):1042. DOI: 10.3390/s20041042.

Inc., M. T. (2005). Pickit 2 programmer/debugger user's guide. Acesso em: 15 jun. 2024.

Inc., M. T. (2018). *MCP2551 CAN Transceiver Data Sheet*. Microchip Technology Inc., Chandler. Acesso em: 15 jun. 2024.

Inc., M. T. (2024). *PIC18F2580 Data Sheet*. Microchip Technology Inc., Chandler. Acesso em: 15 jun. 2024.

Integrated, M. (2015). Ds18b20 programmable resolution 1-wire digital thermometer, waterproof version. Acesso em: 15 jun. 2024.

IRROMETER Company, Inc. (2026). *WATERMARK 200SS Soil Moisture Sensor*. IRROMETER Company, Inc. Acesso em: 03 jun. 2026.

Kamienski, C., Soininen, J.-P., Taumberger, M., Dantas, R., Toscano, A., Cinotti, T. S., Maia, R. F., and Torre Neto, A. (2019). Smart water management platform: Iot-based precision irrigation for agriculture. *Sensors*, 19(2):276. DOI: 10.3390/s19020276.

Lisbinski, F. C., Mühl, D. D., Oliveira, L. d., and Coronel, D. A. (2020). Perspectivas e desafios da agricultura 4.0 para o setor agrícola. *Anais..[do] VIII Simpósio da Ciência do Agronegócio*.

Lopes, M. T. and Silva, R. S. (2022). Desenvolvimento e produtividade da cultura do pimentão sob diferentes regimes de irrigação. *Revista Brasileira de Engenharia Agrícola e Ambiental*, 26(4):245–252.

Maia, R. F., Ballester Lurbe, C., Agrahari Baniya, A., and Hornbuckle, J. (2020). Irrisens: An iot platform based on microservices applied in commercial-scale crops working in a multi-cloud environment. *Sensors*, 20(24):7163. DOI: 10.3390/s20247163.

Marouelli, W. A., Oliveira, S. d., Coelho, E. F., Sousa, L. C. N., and de, V. F. (2011). `https://ainfo.cnptia.embrapa.br/digital/bitstream/item/55931/1/IRRIGACAO-e-FERTIRRIGACAO-cap5.pdf`. Acessado em: 30-04-2024.

Matias, S. R. S. (2021). Cobertura do solo no cultivo da cebolinha-comum: eficiência no controle de plantas daninhas e uso da água.

Morchid, A., Jebabra, R., Khalid, H. M., El Alami, R., Qjidaa, H., and Ouazzani Jamil, M. (2024). Iot-based smart irrigation management system to enhance agricultural water security using embedded systems, telemetry data, and cloud computing. *Results in Engineering*, 23:102829. DOI: 10.1016/j.rineng.2024.102829.

Ortiz, F. M., Almeida, T., Ferreira, A. E. L. A., and Costa, L. H. M. K. (2019). Caracterizaçao de desempenho de uma rede lora em ambientes urbanos: Simulaçao vs. prática. In *Anais do III Workshop de Computação Urbana*, pages 167–180. SBC.

Pereira, S. A. *et al*. (2023). Impactos da agricultura 4.0 nos pequenos produtores rurais de uma cooperativa do agronegócio no paraná. Master's thesis, Universidade Tecnológica Federal do Paraná.

Robert Bosch GmbH (1991). CAN Specification Version 2.0. Technical report, Robert Bosch GmbH, Stuttgart, Germany.

Souza, A. C. and Matsura, J. M. (2021). Manejo hídrico e automação na cultura do pimentão em ambiente protegido. *Horticultura Brasileira*, 39:112–119.

Souza, C. A. (2017). Cultivo orgânico de cebolinha (allium fistulosum l.) sob diferentes manejos de cobertura de solo. *Anais dos Seminários de Iniciação Científica*, (21).

Souza, C. F. and Matsura, E. E. (2004). Distribuição da água no solo para o dimensionamento da irrigação por gotejamento. *Revista Brasileira de Engenharia Agrícola e Ambiental*, 8:7–15.

Artigo Completo

Sistema de Monitoramento e Alerta de Temperatura para Compostagem na Agricultura Regenerativa: Integração IoT, Análise de Dados e Automação Operacional

Temperature Monitoring and Alert System for Composting in Regenerative Agriculture: IoT Integration, Data Analysis, and Operational Automation

Maria Clara Mendes da Silva [Universidade Federal do Vale do São Francisco | *clara.mendes@discente.univasf.edu.br*]
Ana Júlia Fernandes de Oliveira Barros [Universidade Federal do Vale do São Francisco | *anajulia.oliveira@univasf.edu.br*]
Danillo de Lima Alves Xavier [Universidade Federal do Vale do São Francisco | *danillox5@hotmail.com*]
Rodrigo Fernandes Scocca [Universidade Federal do Vale do São Francisco | *rodrigoscocca@hotmail.com*]
Jadsonlee da Silva Sá [Universidade Federal do Vale do São Francisco | *jadsonlee.sa@univasf.edu.br*]

Universidade Federal do Vale do São Francisco, Avenida Antônio Carlos Magalhães, 510, Bairro Santo Antônio, Juazeiro, Bahia, 48902-300, Brasil.

Resumo. A compostagem é um processo biológico aeróbio influenciado por variáveis como temperatura, umidade, oxigenação e relação carbono/nitrogênio. Entre esses parâmetros, a temperatura se destaca como um indicador operacional relevante, pois reflete a atividade microbiana e permite acompanhar a transição entre as fases mesofílica, termofílica e de maturação. Este trabalho apresenta a estruturação e validação de uma arquitetura computacional para monitoramento e alerta de temperatura em leiras de compostagem no contexto da agricultura regenerativa, utilizando dados reais coletados no pátio de compostagem da unidade da Bayer em Petrolina-PE. A solução integra sensores Laird conectados via LoRaWAN, transmissão por *gateway*, armazenamento em BigQuery, visualização em Power BI e automação de alertas por meio do Fabric Activator e Microsoft Teams. O conjunto de dados analisado compreendeu 100 dias de monitoramento, sendo submetido a etapas de filtragem, organização, verificação de consistência temporal, avaliação de volumetria e análise descritiva. Os resultados permitiram caracterizar padrões térmicos das leiras, identificar comportamentos associados a diferentes fases do processo e estruturar indicadores visuais para acompanhamento operacional. Além disso, foram configuradas regras de alerta para notificar a entrada das leiras na faixa termofílica e realizar o acompanhamento recorrente enquanto essa condição térmica permanece ativa. A principal contribuição do trabalho está na demonstração de como uma infraestrutura IoT já implantada pode ser organizada e integrada a ferramentas analíticas e de automação para transformar medições contínuas de campo em informações úteis à tomada de decisão operacional. Embora o sistema utilize a temperatura como variável central e não substitua medições complementares de umidade, oxigenação ou composição química, os resultados indicam seu potencial para apoiar o monitoramento contínuo, reduzir a dependência de inspeções manuais e fortalecer práticas de gestão associadas à agricultura regenerativa.

Abstract. Composting is an aerobic biological process influenced by variables such as temperature, moisture, oxygen availability, and carbon-to-nitrogen ratio. Among these parameters, temperature stands out as a relevant operational indicator because it reflects microbial activity and supports the monitoring of transitions between the mesophilic, thermophilic, and maturation phases. This work presents the structuring and validation of a computational architecture for temperature monitoring and alerting in composting windrows within the context of regenerative agriculture, using real data collected at the composting yard of Bayer's unit in Petrolina-PE, Brazil. The proposed solution integrates Laird sensors connected through LoRaWAN, gateway-based data transmission, storage in BigQuery, visualization in Power BI, and alert automation using Fabric Activator and Microsoft Teams. The analyzed dataset covered 100 days of monitoring and was processed through filtering, organization, temporal consistency verification, volumetry assessment, and descriptive analysis. The results enabled the characterization of thermal patterns in the windrows, the identification of behaviors associated with different composting phases, and the development of visual indicators for operational monitoring. In addition, alert rules were configured to notify users when windrows entered the thermophilic range and to provide recurring updates while this thermal condition remained active. The main contribution of this work lies in demonstrating how an existing IoT infrastructure can be organized and integrated with analytical and automation tools to transform continuous field measurements into useful information for operational decision-making. Although the system uses temperature as its central variable and does not replace complementary measurements of moisture, oxygen availability, or chemical composition, the results indicate its potential to support continuous monitoring, reduce dependence on manual inspections, and strengthen management practices associated with regenerative agriculture.

Palavras-chave: Compostagem. Monitoramento de temperatura. IoT. LoRaWAN. Automação de alertas. Agricultura regenerativa.

Keywords: Composting. Temperature monitoring. IoT. LoRaWAN. Alert automation. Regenerative agriculture.

Recebido/Received: 31 03 2026 • **Aceito/Accepted:** 25 08 2026 • **Publicado/Published:** 15 10 2026

Publicado sob a Licença Pública Internacional Creative Commons Atribuição 4.0 (CC BY 4.0)

1 Introdução

A agricultura regenerativa tem se consolidado como uma abordagem estratégica para a sustentabilidade agrícola, ao integrar produtividade, conservação dos recursos naturais, restauração da saúde do solo e redução de impactos ambientais. Segundo Lacanne and Lundgren [2018], esse modelo busca aliar práticas produtivas rentáveis à conservação dos recursos naturais. Schreefel *et al*. [2020] destacam que a regeneração contínua do solo, o aumento da biodiversidade e a ciclagem de nutrientes são pilares centrais dessa abordagem, tornando-a relevante diante das pressões climáticas, da necessidade de redução de resíduos e da busca por sistemas agrícolas mais resilientes.

Nesse contexto, a compostagem ocupa papel essencial ao transformar resíduos orgânicos em um composto estável e nutritivo, contribuindo para o fechamento do ciclo de nutrientes no solo e para a redução do descarte inadequado de biomassa. Albino *et al*. [2021] apontam que a compostagem contribui para a melhoria da estrutura e da fertilidade do solo, além de mitigar passivos ambientais associados ao acúmulo de resíduos orgânicos. A eficiência desse processo, entretanto, depende do equilíbrio de variáveis críticas, como temperatura, umidade, oxigenação e relação carbono/nitrogênio (C/N). Entre essas variáveis, a temperatura se destaca como um indicador operacional relevante, pois reflete a atividade microbiana, permite acompanhar a transição entre as fases mesofílica, termofílica e de maturação, e auxilia na identificação de possíveis desvios no processo.

A fase termofílica, em particular, é fundamental para a degradação de compostos orgânicos mais complexos e para a higienização do material, pois temperaturas adequadas contribuem para a redução de patógenos e sementes indesejadas. Por outro lado, quedas abruptas de temperatura podem indicar redução da atividade microbiológica, enquanto temperaturas excessivamente elevadas podem comprometer o equilíbrio biológico do processo. Dessa forma, o monitoramento contínuo da temperatura não substitui análises complementares de umidade, oxigenação ou composição química, mas fornece uma referência operacional importante para apoiar decisões como inspeção da leira, revolvimento, irrigação ou verificação de anomalias térmicas.

A adoção de tecnologias digitais tem ampliado significativamente o potencial de monitoramento e otimização da compostagem. Soluções baseadas em *Internet of Things* (IoT) permitem a coleta contínua e remota de dados de campo, reduzindo a dependência de medições manuais e aumentando a rastreabilidade do processo. Em ambientes agrícolas distribuídos, tecnologias de comunicação de longo alcance e baixo consumo, como LoRaWAN, possibilitam conectar sensores instalados em locais afastados a sistemas de armazenamento, processamento e visualização de dados. Quando integradas a plataformas analíticas e mecanismos de automação, essas tecnologias permitem transformar leituras brutas em indicadores visuais e alertas capazes de orientar ações operacionais.

A unidade da Bayer em Petrolina-PE implantou, em 2022, um pátio de compostagem, ilustrado na Figura 1, composto por leiras destinadas ao processamento de resíduos provenientes do cultivo protegido, jardinagem e refeitório. Nesse ambiente, sensores Laird instalados nas leiras realizam medições de temperatura e transmitem os dados por meio de uma infraestrutura baseada em comunicação LoRaWAN. Apesar da existência dessa infraestrutura de coleta, havia a necessidade de organizar, validar e estruturar o uso computacional dos dados, de modo a facilitar sua interpretação, ampliar a rastreabilidade e apoiar a tomada de decisão operacional. A ausência de uma camada integrada de análise, visualização e alerta limitava o aproveitamento dos dados coletados continuamente em campo.

Diante desse cenário, este trabalho apresenta a estruturação e validação de uma arquitetura computacional para monitoramento e alerta de temperatura em leiras de compostagem, utilizando a infraestrutura já existente na unidade da Bayer em Petrolina-PE. A solução integra sensores Laird conectados via LoRaWAN, transmissão por *gateway*, armazenamento dos dados em BigQuery, visualização em Power BI e automação de alertas por meio do Fabric Activator e Microsoft Teams. Assim, a infraestrutura física de sensoriamento e comunicação previamente implantada é complementada por uma camada computacional voltada à organização, análise, visualização e reação a eventos térmicos.

A principal contribuição deste trabalho está na validação e estruturação de um fluxo computacional aplicado ao monitoramento térmico de leiras de compostagem em ambiente operacional real. Diferentemente de abordagens baseadas apenas em protótipos laboratoriais, o estudo demonstra como uma infraestrutura IoT já implantada pode ser integrada a ferramentas analíticas e de automação para transformar medições contínuas de campo em informações úteis ao acompanhamento do processo. Nesse sentido, a contribuição é caracterizada como uma aplicação de Engenharia da Computação voltada à integração de sistemas IoT, pipeline de dados, análise descritiva e automação baseada em eventos, com aderência a problemas reais de monitoramento no setor agroindustrial.

Este trabalho não propõe um novo algoritmo de controle nem um novo modelo de predição, mas valida uma arquitetura aplicada capaz de apoiar o monitoramento contínuo da compostagem e reduzir a dependência de inspeções exclusivamente manuais. Como resultado, busca-se fornecer uma base computacional para acompanhamento térmico, emissão de alertas, análise histórica e futura evolução para estratégias mais avançadas, como incorporação de sensores adicionais, avaliação de falhas de comunicação, análise de latência, segurança da informação e modelos preditivos para apoio à tomada de decisão.

O restante deste artigo está organizado da seguinte forma: a Seção 2 apresenta a fundamentação teórica e os trabalhos relacionados, abordando compostagem, parâmetros críticos, IoT e sistemas de monitoramento; a Seção 3 descreve a metodologia, a arquitetura computacional, a aquisição e o pré-processamento dos dados; a Seção 4 apresenta os resultados e a discussão, incluindo a análise térmica, o dashboard e a automação de alertas; por fim, a Seção 5 apresenta as conclusões, limitações e perspectivas futuras.

2 Fundamentação Teórica e Trabalhos Relacionados

A otimização do processo de compostagem em um contexto de agricultura regenerativa exige o acompanhamento de pa-



Figura 1. Fotografia do setor de compostagem na Bayer Petrolina.

râmetros capazes de indicar a eficiência microbiológica, a estabilidade do processo e a qualidade potencial do composto final. Nesta seção são apresentados os fundamentos da compostagem, seus estágios e variáveis críticas, com ênfase na temperatura como indicador operacional. Em seguida, são discutidos sistemas de monitoramento baseados em *Internet of Things* (IoT), a pilha de comunicação utilizada em aplicações distribuídas e o papel de ferramentas de visualização e automação. Por fim, são apresentados trabalhos relacionados e uma comparação com a abordagem adotada neste estudo.

2.1 Compostagem e Parâmetros Críticos

A compostagem é um processo biotecnológico de decomposição aeróbia da matéria orgânica, conduzido por microrganismos sob condições controladas de temperatura, umidade, oxigenação e relação carbono/nitrogênio (C/N). A gestão adequada desses fatores acelera a decomposição da matéria orgânica, reduz o descarte de resíduos em aterros e contribui para a produção de um composto mais estável e rico em nutrientes [Prieto, 2019; Farooq *et al*., 2022].

O processo ocorre, de modo geral, em fases sucessivas associadas à atividade microbiológica e à variação térmica da massa compostada. Inicialmente, a fase mesofílica é caracterizada por temperaturas moderadas e intensa atividade de microrganismos adaptados a essa faixa. Em seguida, a fase termofílica apresenta elevação da temperatura, geralmente entre 45 °C e 65 °C, faixa importante para a degradação de compostos orgânicos mais complexos e para a redução de patógenos e sementes indesejadas [Singh and Kulshreshtha, 2021; Rashid *et al*., 2021]. Posteriormente, ocorre a fase de resfriamento, seguida pela maturação do composto. A Tabela 1 resume as faixas de temperatura e os intervalos típicos de tempo associados às principais fases do processo.

Embora a temperatura seja a variável central analisada neste trabalho, ela está diretamente relacionada a outros parâmetros críticos da compostagem. A umidade influencia a atividade microbiana e, consequentemente, a geração de calor: valores muito baixos reduzem a decomposição, enquanto o excesso de umidade pode limitar a difusão de oxigênio e favorecer condições anaeróbias. A oxigenação, por sua vez, sustenta a respiração aeróbia dos microrganismos, sendo essencial para a manutenção da fase termofílica. A relação C/N afeta a disponibilidade de carbono como fonte de energia e de nitrogênio para o crescimento microbiano, influenciando a intensidade e a duração das fases do processo [Brietzke, 2017; Pedrosa *et al*., 2015].

Tabela 1. Estágios do processo de compostagem e faixas de controle.

Estágio	Faixa de temperatura	Faixa de tempo
Fase mesofílica	Até 40 °C	2 a 5 dias
Fase termofílica	Entre 45 °C e 65 °C	3 a 5 semanas
Fase de resfriamento	De 40 °C até a temperatura ambiente	1 a 2 meses

Fonte: Adaptado de EMBRAPA [2024].

Dessa forma, variações térmicas podem indicar indiretamente alterações na umidade, na aeração ou na atividade biológica da leira. Quedas abruptas de temperatura podem estar associadas à redução da atividade microbiológica, à baixa umidade, à compactação do material ou à deficiência de oxigenação. Temperaturas excessivamente elevadas, por outro lado, podem indicar desequilíbrios que comprometem a eficiência biológica do processo. Assim, neste trabalho, a temperatura é tratada como um indicador operacional central para o monitoramento da compostagem, mas não como critério único para avaliação completa da qualidade do composto.

2.2 Sistemas de Monitoramento Baseados em IoT

Soluções baseadas em *Internet of Things* têm ganhado espaço na agricultura por possibilitarem a coleta contínua e remota de dados ambientais com baixo consumo energético e maior rastreabilidade. Sensores embarcados, conectados a redes de comunicação de longo alcance, permitem monitorar variáveis como temperatura, umidade e gases associados ao processo

de decomposição, reduzindo a dependência de inspeções manuais e ampliando a capacidade de resposta operacional.

Rodrigues *et al*. [2021] demonstram a aplicação de microcontroladores como ESP32 em composteiras automatizadas, validando a viabilidade de sistemas de baixo custo para controle de temperatura e umidade. Entretanto, em cenários industriais ou agrícolas distribuídos, a adoção de tecnologias de maior alcance e baixo consumo torna-se especialmente relevante. Nesse contexto, o protocolo LoRaWAN é amplamente utilizado em aplicações IoT por permitir comunicação sem fio de longo alcance, baixo consumo energético e operação em ambientes com sensores geograficamente distribuídos.

Além da camada de comunicação sem fio, sistemas IoT aplicados ao monitoramento ambiental normalmente dependem de mecanismos de mensageria, processamento, armazenamento e visualização. Protocolos leves, como MQTT (*Message Queuing Telemetry Transport*), são frequentemente utilizados para transmissão de mensagens em arquiteturas distribuídas, pois seguem um modelo de publicação/assinatura adequado a dispositivos com restrições de processamento, energia e largura de banda [iso, 2016]. Após a transmissão, os dados podem ser estruturados por aplicações intermediárias, armazenados em bancos analíticos e disponibilizados para painéis de visualização e sistemas de alerta.

2.3 Pilha de Comunicação e Integração de Dados

Em sistemas distribuídos de monitoramento, a organização da pilha de comunicação é essencial para garantir rastreabilidade, manutenção e evolução da solução. No contexto deste trabalho, a arquitetura analisada pode ser compreendida como um fluxo em camadas, iniciando na medição física da temperatura e finalizando na visualização e automação de alertas. A Tabela 2 apresenta uma visão resumida dessa pilha.

Essa organização em camadas evidencia a natureza computacional do sistema, uma vez que o monitoramento não depende apenas da presença de sensores, mas da integração entre dispositivos físicos, protocolos de comunicação, serviços de armazenamento, ferramentas analíticas e mecanismos de reação a eventos.

2.4 Visualização Dinâmica e Sistemas de Alerta

O uso de ferramentas de *Business Intelligence* (BI), como Power BI, Grafana e Kibana, permite a criação de painéis dinâmicos para acompanhamento de séries temporais e indicadores operacionais. Esses ambientes facilitam a interpretação de grandes volumes de dados, permitindo que usuários identifiquem tendências, anomalias e condições críticas de forma mais rápida do que por inspeção direta de tabelas ou registros brutos. Almeida *et al*. [2021] destacam que dashboards interativos contribuem para o acompanhamento de indicadores e para o suporte à tomada de decisão em ambientes operacionais.

No caso da compostagem, a visualização dinâmica permite acompanhar a evolução da temperatura das leiras ao longo do tempo, identificar permanência na faixa termofílica e observar quedas ou picos térmicos que possam indicar necessidade de intervenção. Quando associada a sistemas de automação, essa visualização pode ser complementada por alertas baseados em regras, acionados quando determinados limites são ultrapassados.

Ferramentas como o Fabric Activator permitem configurar eventos associados a dados monitorados em plataformas analíticas, possibilitando o envio de notificações automáticas quando uma condição definida é satisfeita. No contexto deste trabalho, essa automação é utilizada para notificar o atingimento de condições térmicas de interesse e acompanhar a permanência das leiras em faixas consideradas relevantes para o processo. Assim, a integração entre visualização e alerta transforma o sistema de monitoramento em um mecanismo de apoio à decisão operacional.

2.5 Trabalhos Relacionados

Pesquisas recentes demonstram o potencial da integração entre sensores IoT e compostagem. Haque *et al*. [2019] implementaram um sistema automatizado para monitoramento de variáveis ambientais em aplicações agrícolas utilizando sensores e comunicação sem fio, evidenciando ganhos associados à automação e ao acompanhamento remoto. Rodrigues *et al*. [2021] desenvolveram uma composteira automatizada baseada em ESP32, com foco no controle de temperatura e umidade em uma solução de baixo custo. Seibel [2023] apresentou uma solução de monitoramento e controle de compostagem baseada em IoT, destacando o uso de sensores e sistemas embarcados para reduzir a dependência de interferência manual.

Apesar da relevância dessas abordagens, muitos trabalhos concentram-se em protótipos locais ou em sistemas desenvolvidos especificamente para validação experimental. Ainda há espaço para estudos que discutam a utilização de infraestruturas já existentes em ambientes operacionais reais, especialmente quando o foco está na organização dos dados, na validação da continuidade das medições, na integração com plataformas corporativas e na automação de alertas. A Tabela 3 apresenta uma comparação entre trabalhos relacionados e a abordagem adotada neste estudo.

A comparação evidencia que a principal contribuição deste estudo não está na proposição de um novo sensor, algoritmo ou protocolo, mas na estruturação e validação de uma arquitetura computacional aplicada em ambiente operacional real. No caso analisado, a infraestrutura física de sensoriamento e transmissão já estava implantada. Assim, o foco deste trabalho está na organização do fluxo de dados, na validação da consistência dos registros, na análise descritiva das séries temporais, no desenvolvimento de visualizações em Power BI e na automação de alertas por meio do Fabric Activator e Microsoft Teams.

3 Metodologia: Arquitetura e Validação Computacional

Este trabalho adota uma abordagem aplicada, quantitativa e exploratória no campo da Engenharia da Computação, com foco na validação e estruturação computacional de um sistema de monitoramento térmico para leiras de compostagem. A pesquisa foi conduzida a partir de dados reais coletados no pátio de compostagem da unidade da Bayer em Petrolina-PE, utilizando uma infraestrutura de sensoriamento e comunicação previamente implantada. Assim, o estudo não propõe uma

Tabela 2. Pilha de comunicação e integração de dados utilizada no sistema.

Camada	Tecnologia	Função
Sensoriamento	Sensores Laird	Medição da temperatura nas leiras.
Comunicação IoT	LoRaWAN	Transmissão de longo alcance e baixo consumo energético.
Borda	Gateway LoRaWAN	Recepção e encaminhamento dos pacotes de dados.
Mensageria	MQTT	Transporte das mensagens pelo modelo publicação/assinatura.
Processamento	Middleware em Python	Estruturação e preparação dos registros coletados.
Armazenamento	BigQuery	Persistência e consulta analítica dos dados.
Visualização	Power BI	Construção de painéis e indicadores operacionais.
Automação	Fabric Activator/Teams	Geração e envio de alertas aos responsáveis.

Tabela 3. Comparação entre trabalhos relacionados e a abordagem deste estudo.

Trabalho	Contexto e variáveis	Arquitetura tecnológica	Supervisão e alertas
Haque *et al*. [2019]	Ambiente agrícola experimental; monitoramento de variáveis ambientais.	Sistema sem fio com solução própria para aquisição e controle.	Dashboard não enfatizado; contempla controle e automação.
Rodrigues *et al*. [2021]	Protótipo de composteira; monitoramento de temperatura e umidade.	Sistema local baseado em ESP32 e processamento embarcado.	Dashboard não enfatizado; controle realizado localmente.
Seibel [2023]	Protótipo de compostagem; monitoramento de variáveis associadas ao processo.	Sistema embarcado com recursos de IoT e armazenamento próprio.	Apresenta dashboard e mecanismos de alerta.
Este trabalho	Ambiente operacional real; monitoramento da temperatura em leiras de compostagem.	Integração LoRaWAN/MQTT, middleware em Python, BigQuery e Power BI.	Dashboard operacional com alertas via Fabric Activator/Teams.

nova infraestrutura física de coleta, mas organiza, valida e integra os dados gerados pelo sistema existente a ferramentas de análise, visualização e automação de alertas.

A metodologia foi estruturada em quatro etapas principais: (i) caracterização da infraestrutura de coleta e transmissão de dados; (ii) extração, filtragem e pré-processamento dos registros armazenados no BigQuery; (iii) análise descritiva das séries temporais de temperatura; e (iv) desenvolvimento de mecanismos de visualização e automação por meio do Power BI, Fabric Activator e Microsoft Teams.

3.1 Caracterização do Ambiente de Estudo

O ambiente analisado corresponde ao pátio de compostagem da unidade da Bayer em Petrolina-PE, implantado em 2022 para processamento de resíduos orgânicos provenientes do cultivo protegido, jardinagem e refeitório. O ambiente operacional contempla 15 leiras de compostagem e utiliza 13 sensores Laird para o monitoramento térmico. Os sensores são instalados no interior da massa orgânica, aproximadamente a 60 cm de profundidade, quando há material em processo de decomposição.

Como o número de leiras é superior ao número de sensores disponíveis, nem todas as leiras apresentam necessariamente medições simultâneas durante todo o período de operação. A presença de registros para determinada leira depende da instalação e da disponibilidade do sensor durante o respectivo ciclo de compostagem.

A infraestrutura física de monitoramento já estava em operação antes do desenvolvimento deste estudo. Os sensores realizam medições de temperatura e transmitem os dados por meio de comunicação LoRaWAN para um *gateway*. Em seguida, os registros são encaminhados ao ambiente de armazenamento e análise, onde ficam disponíveis para consulta e processamento. Dessa forma, a contribuição metodológica deste trabalho concentra-se na validação computacional do fluxo de dados, na organização das medições, na construção de indicadores visuais e na configuração de alertas automáticos.

As análises quantitativas foram concentradas nos dispositivos e nas leiras que apresentaram registros válidos e consistentes no período considerado em cada análise. Essa distinção é importante, pois sensores podem apresentar diferenças de volumetria em função de fatores como disponibilidade da leira, manutenção, falhas de transmissão, reinicializações ou períodos sem material a ser monitorado.

3.2 Arquitetura Computacional do Sistema

A arquitetura utilizada neste estudo integra camadas de sensoriamento, comunicação, processamento, armazenamento, visualização e automação. A Figura 2 apresenta a organização geral do sistema, desde a coleta da temperatura nas leiras até o envio de notificações aos usuários responsáveis pelo acompanhamento operacional.

Na primeira camada, os sensores Laird realizam a medição da temperatura no interior das leiras. Em seguida, os dados são transmitidos por LoRaWAN, tecnologia adequada a aplicações IoT distribuídas por operar com baixo consumo energético e comunicação de longo alcance. O *gateway* LoRaWAN recebe os pacotes enviados pelos sensores e os encaminha para a camada de mensageria, na qual o protocolo MQTT (*Message Queuing Telemetry Transport*) é utilizado para publicação e transporte das mensagens. Na etapa seguinte, um *middleware* desenvolvido em Python recebe essas mensagens, estrutura os registros e os disponibiliza para armazenamento no BigQuery.

O BigQuery atua como repositório analítico central, permitindo consultas, filtragens e consolidação dos dados de temperatura. A partir desse ambiente, o Power BI é utilizado para criação de painéis interativos, indicadores visuais e grá-

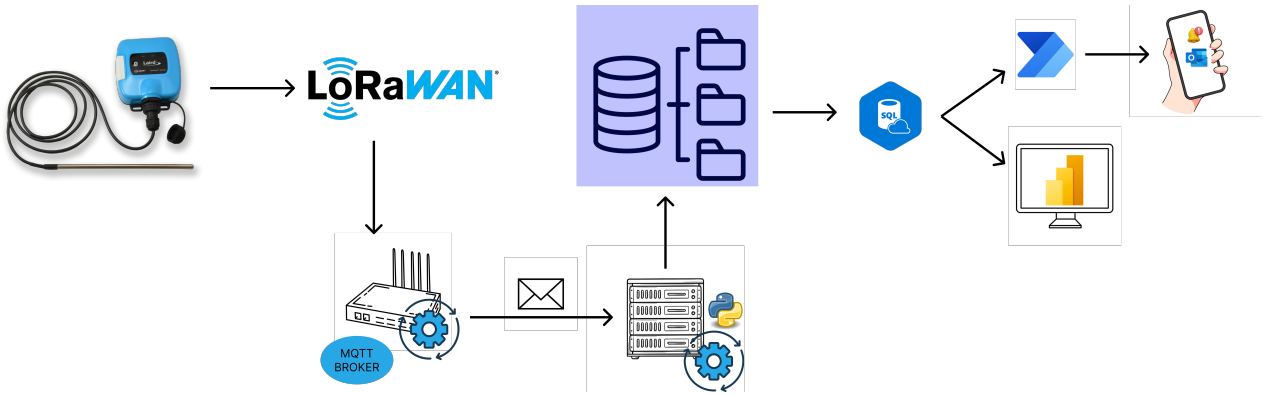


Figura 2. Arquitetura integrada do sistema, da coleta de dados à automação de alertas.

ficos históricos. Por fim, o Fabric Activator é empregado para configurar regras de automação baseadas em eventos térmicos, com envio de notificações pelo Microsoft Teams. Assim, a arquitetura não se limita à coleta de dados, mas estabelece um fluxo computacional completo entre medição, transmissão, processamento, armazenamento, análise, visualização e reação operacional.

3.3 Pilha de Comunicação e Integração

A solução analisada pode ser representada como uma pilha de comunicação e integração de dados, conforme apresentado na Tabela 2. Essa organização permite evidenciar os componentes computacionais envolvidos e a função de cada camada no processo de monitoramento.

A descrição em camadas também permite discutir características importantes da arquitetura, como escalabilidade, manutenção e tolerância a falhas. A inclusão de novos sensores ou leiras pode ser realizada mantendo o mesmo fluxo geral de ingestão e análise, desde que os novos dispositivos sejam corretamente identificados no *pipeline* de dados. Além disso, a separação entre coleta, comunicação, processamento, armazenamento e visualização facilita a identificação de falhas em pontos específicos, como ausência de transmissão, inconsistências temporais ou interrupções no recebimento dos registros.

3.4 Aquisição, Filtragem e Pré-processamento dos Dados

Os sensores foram configurados para registrar medições em intervalos de aproximadamente uma hora, resultando em até 24 leituras diárias por dispositivo em condições regulares de operação. Os dados utilizados neste estudo foram obtidos a partir da base armazenada no BigQuery, considerando um intervalo de 100 dias de monitoramento. A extração foi realizada por meio de consulta SQL, apresentada no Apêndice **??**, com o objetivo de selecionar exclusivamente os registros associados aos sensores instalados no pátio de compostagem da unidade de Petrolina-PE.

O conjunto inicial continha registros brutos provenientes da infraestrutura IoT. Após a filtragem por local, tipo de dispositivo, variável de interesse e identificação dos sensores, foram consolidados os registros válidos utilizados nas análises. Essa distinção entre dados brutos e dados válidos é necessária porque bases operacionais podem conter medições de diferentes dispositivos, variáveis distintas, duplicidades, registros incompletos ou mensagens não associadas diretamente ao escopo do estudo.

A variável de temperatura armazenada no BigQuery estava originalmente identificada como *external_temperature_degF*, indicando valores em graus Fahrenheit. Antes da análise estatística e da construção dos indicadores, os valores foram convertidos para graus Celsius, conforme a Equação 1.

$$T_C = (T_F - 32) \times \frac{5}{9} \tag{1}$$

O pré-processamento incluiu a seleção dos sensores de interesse, remoção de duplicidades quando identificadas, padronização dos campos temporais, conversão da temperatura para graus Celsius e organização dos registros por sensor e data. Também foram verificadas a consistência temporal, a volumetria por dispositivo e a presença de lacunas de transmissão. Essas etapas permitiram avaliar se o conjunto de dados era adequado para a caracterização térmica das leiras e para a construção dos painéis de monitoramento.

3.5 Validação Computacional dos Registros

A validação computacional teve como objetivo verificar a confiabilidade do fluxo de dados, considerando a disponibilidade dos registros, sua distribuição temporal e a coerência dos valores armazenados. Para isso, foram avaliados os seguintes aspectos:

- **Integridade dos registros:** verificação da presença dos campos essenciais, como identificação do sensor, instante de medição, nome da variável e valor medido.
- **Consistência temporal:** análise da distribuição dos registros ao longo dos 100 dias, observando lacunas, intervalos sem transmissão e possíveis interrupções.
- **Volumetria por sensor:** contagem de registros por dispositivo, permitindo identificar sensores com maior ou menor disponibilidade de dados.
- **Consistência física dos valores:** verificação das faixas de temperatura após conversão para graus Celsius, com atenção a valores incompatíveis com o processo de compostagem.
- **Adequação para análise operacional:** avaliação da suficiência dos registros para construção de métricas,

gráficos e regras de alerta.

Essa etapa é particularmente relevante em sistemas IoT aplicados a ambientes reais, pois variações na quantidade de dados por sensor podem ocorrer em razão de falhas de comunicação, reinicializações, indisponibilidade de rede, manutenção dos dispositivos, ausência de material em determinada leira ou diferenças na configuração de envio das mensagens. Portanto, diferenças de volumetria não foram interpretadas automaticamente como falha crítica, mas como aspectos a serem analisados no contexto da operação.

3.6 Análise Descritiva das Temperaturas

Após o pré-processamento e a validação dos registros, foram calculadas métricas descritivas para caracterizar o comportamento térmico das leiras monitoradas. Para cada sensor, foram obtidos valores de média, desvio padrão, mínimo e máximo de temperatura. Essas métricas foram utilizadas para identificar diferenças entre sensores, avaliar a variabilidade térmica e reconhecer padrões compatíveis com diferentes fases do processo de compostagem, além de subsidiar a construção dos indicadores visuais e a definição das regras de alerta.

As faixas de referência utilizadas para interpretação dos dados foram baseadas nos estágios apresentados na Tabela 1. Em particular, a faixa termofílica, entre 45 °C e 65 °C, foi considerada relevante para acompanhamento operacional, uma vez que está associada à maior atividade microbiológica e à higienização do material. Valores abaixo ou acima dessa faixa foram utilizados como referência para análise de possíveis desvios térmicos.

Sempre que disponíveis, as informações sobre o início de formação das leiras foram consideradas para correlação com as séries temporais de temperatura, uma vez que a idade da leira influencia diretamente a fase predominante do processo. Caso essas datas não estejam disponíveis de forma estruturada no conjunto de dados, essa ausência é considerada uma limitação metodológica, pois restringe a associação direta entre o perfil térmico e o estágio cronológico da compostagem.

De forma semelhante, a temperatura ambiente externa constitui uma variável de referência relevante para a interpretação do comportamento térmico das leiras, especialmente nas fases de resfriamento e maturação. Na ausência de um sensor externo dedicado à temperatura ambiente no pátio, as análises ficam restritas às temperaturas internas das leiras, sendo essa ausência considerada uma limitação para comparação entre a temperatura do material em compostagem e as condições térmicas do ambiente.

3.7 Desenvolvimento do Dashboard e Indicadores Visuais

A etapa de visualização foi desenvolvida no Power BI, utilizando o BigQuery como fonte de dados. O objetivo do *dashboard* foi transformar os registros históricos de temperatura em informações de fácil interpretação para acompanhamento operacional. Para isso, foram construídos gráficos de séries temporais, indicadores de temperatura máxima, média e mínima, comparações entre sensores e classificações visuais das faixas térmicas.

Os indicadores foram organizados para permitir que o usuário identifique rapidamente a condição térmica de cada leira e observe tendências ao longo do tempo. A utilização de filtros por período e por sensor permite a análise individual ou comparativa das leiras. Além disso, a representação gráfica auxilia na identificação de picos, quedas abruptas, períodos de estabilidade e permanência em faixas compatíveis com a fase termofílica.

Nesse contexto, o *dashboard* não é tratado apenas como elemento visual, mas como parte da arquitetura computacional de apoio à decisão. Ele permite transformar registros numéricos em indicadores operacionais, reduzindo a necessidade de inspeção manual direta sobre grandes volumes de dados.

3.8 Automação de Alertas

A automação foi implementada por meio do Fabric Activator, integrado ao ambiente de visualização e ao Microsoft Teams. Foram definidas regras baseadas nas faixas térmicas de interesse para o processo de compostagem, com ênfase na identificação da entrada das leiras na faixa termofílica, considerada neste trabalho a partir de 45 °C.

Foram consideradas duas estratégias complementares de acompanhamento. A primeira corresponde ao acionamento de uma regra quando a temperatura monitorada atinge ou ultrapassa 45 °C. Essa condição sinaliza a entrada da leira na faixa termofílica adotada como referência no estudo e permite que os responsáveis sejam notificados sem a necessidade de acompanhamento contínuo do *dashboard*.

A segunda estratégia corresponde ao acompanhamento recorrente da condição térmica, permitindo o envio periódico de notificações enquanto a leira permanece em uma faixa de interesse para o processo. Dessa forma, o sistema contempla tanto a identificação do início de uma condição térmica relevante quanto seu acompanhamento ao longo do tempo.

As notificações são encaminhadas aos usuários definidos como destinatários no Fabric Activator por meio do Microsoft Teams. As mensagens podem conter informações associadas ao evento monitorado, como identificação da leira ou do sensor e a condição térmica responsável pelo acionamento da regra.

A automação baseada em eventos contribui para reduzir a dependência de verificação manual contínua do *dashboard*. No entanto, as notificações devem ser interpretadas como mecanismos de apoio ao acompanhamento operacional, e não como substitutas da avaliação técnica do processo de compostagem. A decisão sobre revolvimento, irrigação, inspeção ou outras ações deve considerar também informações complementares, como condição visual da leira, umidade, histórico de manejo e estágio do ciclo.

3.9 Considerações sobre Aspectos Computacionais

Além da análise térmica, a metodologia considera aspectos computacionais relevantes para sistemas distribuídos e IoT. A escalabilidade é observada pela possibilidade de inclusão de novos sensores e leiras no fluxo existente, desde que a identificação dos dispositivos seja incorporada às consultas e aos painéis. A tolerância a falhas é analisada a partir da detecção de lacunas, interrupções ou diferenças de volumetria entre sensores, embora o sistema atual não implemente

mecanismos automáticos de retransmissão ou recuperação de falhas.

A latência do sistema é considerada sob uma perspectiva operacional. Como o monitoramento da compostagem não exige resposta em tempo real estrito, atrasos da ordem de minutos podem ser aceitáveis, desde que os alertas sejam emitidos em tempo compatível com as ações de manejo. Em relação à segurança, a arquitetura depende dos mecanismos próprios das tecnologias utilizadas, como autenticação e criptografia na comunicação LoRaWAN e controle de acesso nas plataformas corporativas de armazenamento, visualização e notificação. Uma avaliação formal de segurança, entretanto, não faz parte do escopo deste trabalho e é indicada como possibilidade de investigação futura.

4 Resultados e Discussão

Esta seção apresenta os principais resultados obtidos com a organização e validação dos dados armazenados no BigQuery, a análise do comportamento térmico das leiras e a implementação das camadas de visualização e automação. Os achados são discutidos tanto sob a perspectiva do monitoramento térmico da compostagem quanto sob o ponto de vista computacional, considerando a integração entre aquisição de dados, armazenamento, análise e geração de alertas.

4.1 Validação e Organização dos Dados

A primeira etapa dos resultados concentrou-se na verificação da consistência dos registros provenientes da infraestrutura IoT já instalada na unidade. A consulta SQL aplicada ao BigQuery permitiu isolar exclusivamente os dados de temperatura associados às leiras do pátio de compostagem, removendo registros não relacionados ao escopo do estudo e estruturando uma base adequada às etapas posteriores de análise.

A inspeção do conjunto de dados confirmou que o fluxo de aquisição permaneceu operacional ao longo do período analisado, possibilitando a consolidação de um volume representativo de medições. A análise temporal dos registros também evidenciou diferenças de volumetria entre sensores, comportamento esperado em aplicações de campo, nas quais fatores como reinicializações, disponibilidade de material nas leiras, manutenção e oscilações de comunicação podem influenciar a quantidade de medições por dispositivo.

Do ponto de vista metodológico, esse resultado é relevante porque mostra que a etapa de validação não se restringe à contagem de registros, mas envolve a análise de integridade, consistência temporal e adequação do conjunto para uso analítico e operacional. Em sistemas IoT aplicados a ambientes reais, a qualidade do histórico armazenado é condição necessária para que gráficos, indicadores e alertas possam ser utilizados com confiabilidade.

4.2 Análise do Comportamento Térmico das Leiras

Após o pré-processamento e a validação dos registros, foram calculadas métricas descritivas para caracterizar o comportamento térmico das leiras monitoradas. A análise estatística e gráfica evidenciou comportamentos térmicos distintos entre as leiras ao longo do período observado, compatíveis com diferentes condições e momentos do processo de compostagem.

A Figura 3 apresenta a distribuição das temperaturas para as Leiras 04 a 09, selecionadas para a análise ilustrada. Observa-se que as Leiras 04 e 06 apresentaram temperaturas centrais mais elevadas e maior amplitude térmica, com registros alcançando valores superiores a 50 °C. Esses comportamentos são compatíveis com períodos de maior atividade térmica e com condições associadas à faixa termofílica. A Leira 07 também apresentou maior dispersão dos valores e registros superiores a 50 °C. Em contraste, as Leiras 05, 08 e 09 apresentaram distribuições mais concentradas e temperaturas centrais inferiores. Entretanto, somente a temperatura não permite determinar de forma inequívoca a fase biológica correspondente a cada leira, uma vez que sua interpretação também depende de informações como idade da leira, umidade, aeração e histórico de manejo.

A análise do boxplot reforça a heterogeneidade natural do processo de compostagem. Mesmo em um mesmo pátio, as leiras podem se encontrar em etapas distintas do ciclo biológico, refletindo diferenças de idade, manejo, umidade, revolvimento e composição do material. Dessa forma, o uso de estatísticas descritivas por leira é essencial para evitar interpretações generalistas baseadas apenas em médias globais.

Outro resultado relevante foi a identificação de uma temperatura máxima de 56,84 °C no conjunto apresentado pelo painel. Esse registro encontra-se dentro da faixa termofílica adotada como referência neste estudo, entre 45 °C e 65 °C, demonstrando que pelo menos parte das leiras monitoradas atingiu condições térmicas compatíveis com essa fase. O valor máximo isolado, entretanto, não permite inferir a duração da exposição térmica nem comprovar, por si só, a higienização do material.

4.3 Dashboard para Monitoramento Operacional

Com base nos dados estruturados, foi desenvolvido um *dashboard* no Power BI para acompanhamento operacional do processo. A interface do painel, apresentada na Figura 4, foi concebida para reunir, em um único ambiente visual, informações temporais, estatísticas e operacionais relevantes ao monitoramento das leiras.

Na parte superior do painel, os filtros de intervalo temporal e seleção de leiras permitem que o usuário realize análises específicas por período e por unidade monitorada. O gráfico de linhas apresenta a evolução temporal da temperatura média, permitindo identificar ciclos térmicos, elevações, quedas bruscas e períodos de estabilidade. Esse comportamento é especialmente útil para observar transições entre fases da compostagem e inferir momentos em que houve necessidade de intervenção.

O painel também incorpora um boxplot para comparação estatística entre leiras, um indicador de temperatura máxima atingida e uma tabela de métricas descritivas contendo informações como média, mínimo, máximo, amplitude, desvio padrão, quartis e medidas derivadas. Além disso, o *dashboard* integra um quadro de atividades das leiras, registrando eventos como irrigação e revolvimento, o que amplia a capacidade interpretativa do sistema ao relacionar alterações térmicas com ações operacionais efetivamente realizadas.

Do ponto de vista prático, a principal contribuição do *dashboard* está em transformar um volume elevado de registros brutos em um conjunto de indicadores de interpretação

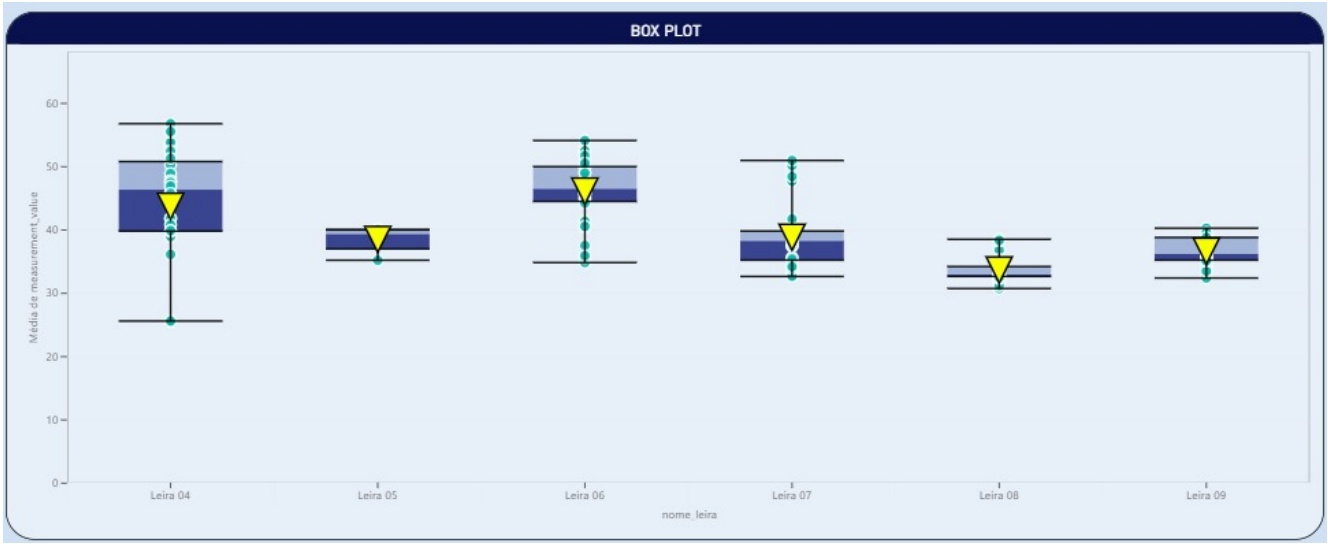


Figura 3. Distribuição das temperaturas registradas por leira no período analisado.

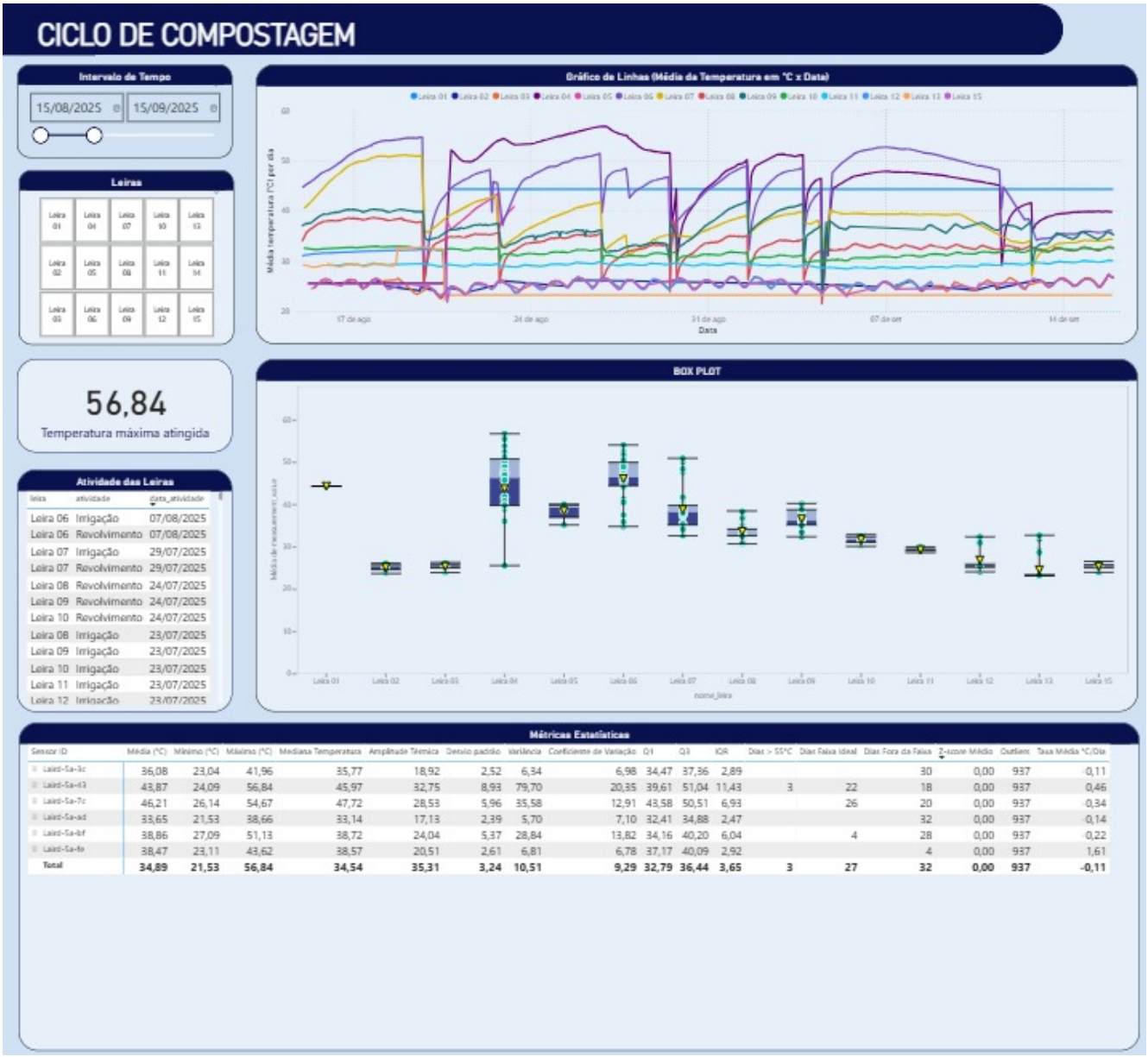


Figura 4. *Dashboard* desenvolvido no Power BI para acompanhamento das temperaturas das leiras.

imediata. Em vez de depender exclusivamente de consultas a banco de dados ou inspeção manual de séries extensas, a equipe pode identificar rapidamente padrões anormais e acompanhar a evolução do processo de forma mais clara e centralizada.

4.4 Automação de Alertas com Fabric Activator

Além da visualização dos dados, foi implementada uma camada de automação utilizando o Fabric Activator, integrada ao Power BI e ao Microsoft Teams. A Figura 5 apresenta a configuração de uma das regras efetivamente utilizadas no sistema.

Na configuração apresentada, o monitoramento está associado à métrica *last_measurement*. A condição utilizada corresponde ao acionamento da regra quando o valor monitorado atinge ou ultrapassa 45 °C. Considerando a faixa de referência adotada neste trabalho, esse limiar corresponde à entrada da leira na faixa termofílica.

Além da condição de acionamento, o Fabric Activator permite definir os usuários destinatários da notificação, o título da mensagem, seu conteúdo e o contexto associado ao evento. Dessa forma, a detecção da condição térmica ocorre automaticamente a partir dos dados disponibilizados ao ambiente de monitoramento.

A configuração demonstra que a solução desenvolvida não se limita à apresentação de informações históricas no *dashboard*. A arquitetura incorpora uma camada de reação automática a eventos, na qual condições identificadas nos dados podem gerar notificações para os responsáveis pelo acompanhamento operacional.

A Figura 6 apresenta uma ação de teste executada a partir da regra configurada no Fabric Activator e recebida no Microsoft Teams. O teste permitiu verificar o fluxo de integração entre o mecanismo de automação e o ambiente utilizado para entrega das notificações.

Na mensagem apresentada na Figura 6, observa-se a indicação de que se trata de uma ação de teste, bem como o título configurado para o alerta e a mensagem associada à condição térmica monitorada. Esse resultado confirma o funcionamento do mecanismo de entrega da notificação entre o Fabric Activator e o Microsoft Teams.

No sistema implementado, foram consideradas duas estratégias complementares de acompanhamento. A primeira corresponde à notificação associada ao atingimento do limiar de 45 °C, utilizado como referência para indicar a entrada na faixa termofílica. A segunda consiste no acompanhamento recorrente enquanto determinada condição térmica permanece ativa, permitindo que os responsáveis acompanhem sua evolução ao longo do tempo.

A combinação dessas estratégias permite utilizar a automação tanto para identificar o início de uma condição relevante quanto para acompanhar sua permanência. Dessa forma, o sistema reduz a necessidade de observação manual contínua do *dashboard*, mantendo os responsáveis informados sobre eventos térmicos de interesse.

4.5 Implicações Operacionais e Computacionais

Os resultados obtidos demonstram que a solução proposta contribui para ampliar o controle operacional do processo de compostagem. A integração entre sensoriamento distribuído, comunicação LoRaWAN, organização dos dados no BigQuery, visualização no Power BI e automação com Fabric Activator constitui um fluxo computacional completo, capaz de transformar medições contínuas em informações acionáveis.

Sob a perspectiva operacional, o sistema favorece a identificação antecipada de situações que exigem atenção, como entrada na faixa termofílica, oscilações abruptas ou comportamentos térmicos persistentes. Isso contribui para reduzir o intervalo entre a ocorrência de um evento e sua percepção pela equipe responsável, aumentando a capacidade de resposta e a rastreabilidade do processo.

Sob a perspectiva da Engenharia da Computação, a principal contribuição está na estruturação de uma arquitetura aplicada em ambiente real, integrando coleta, transmissão, tratamento, armazenamento, visualização e reação automática. O trabalho mostra que uma infraestrutura previamente instalada pode ser reorganizada e potencializada por meio de ferramentas analíticas e mecanismos de automação, sem necessidade de substituição integral da camada física de monitoramento.

Apesar dos avanços observados, os resultados também devem ser interpretados à luz das limitações do estudo. A temperatura foi tratada como variável central, mas não representa isoladamente todas as condições necessárias à avaliação completa da compostagem. Variáveis como umidade, oxigenação, composição do material e histórico detalhado de manejo influenciam o processo e podem ser incorporadas em estudos futuros para ampliar a robustez da análise.

5 Conclusão

Este trabalho apresentou a organização, validação e estruturação computacional dos dados de temperatura coletados no pátio de compostagem da unidade da Bayer em Petrolina-PE, integrando a infraestrutura IoT existente a ferramentas de armazenamento, análise, visualização e automação. A partir dos resultados obtidos, verificou-se que o fluxo de dados disponível foi capaz de sustentar análises descritivas do comportamento térmico das leiras e a construção de mecanismos de apoio ao acompanhamento operacional.

A análise dos registros armazenados no BigQuery mostrou que a base de dados possui volume e consistência suficientes para uso analítico, desde que submetida a etapas de filtragem, organização e verificação temporal. A análise térmica evidenciou diferenças relevantes entre as leiras, confirmando a heterogeneidade do processo de compostagem e a importância do monitoramento individualizado para interpretação adequada do comportamento de cada unidade.

O *dashboard* desenvolvido no Power BI demonstrou ser uma ferramenta eficaz para sintetizar grandes volumes de dados em indicadores visuais e gráficos de interpretação rápida. A integração de filtros, séries temporais, estatísticas descritivas e histórico de atividades ampliou a capacidade de acompanhamento do processo, tornando a análise mais

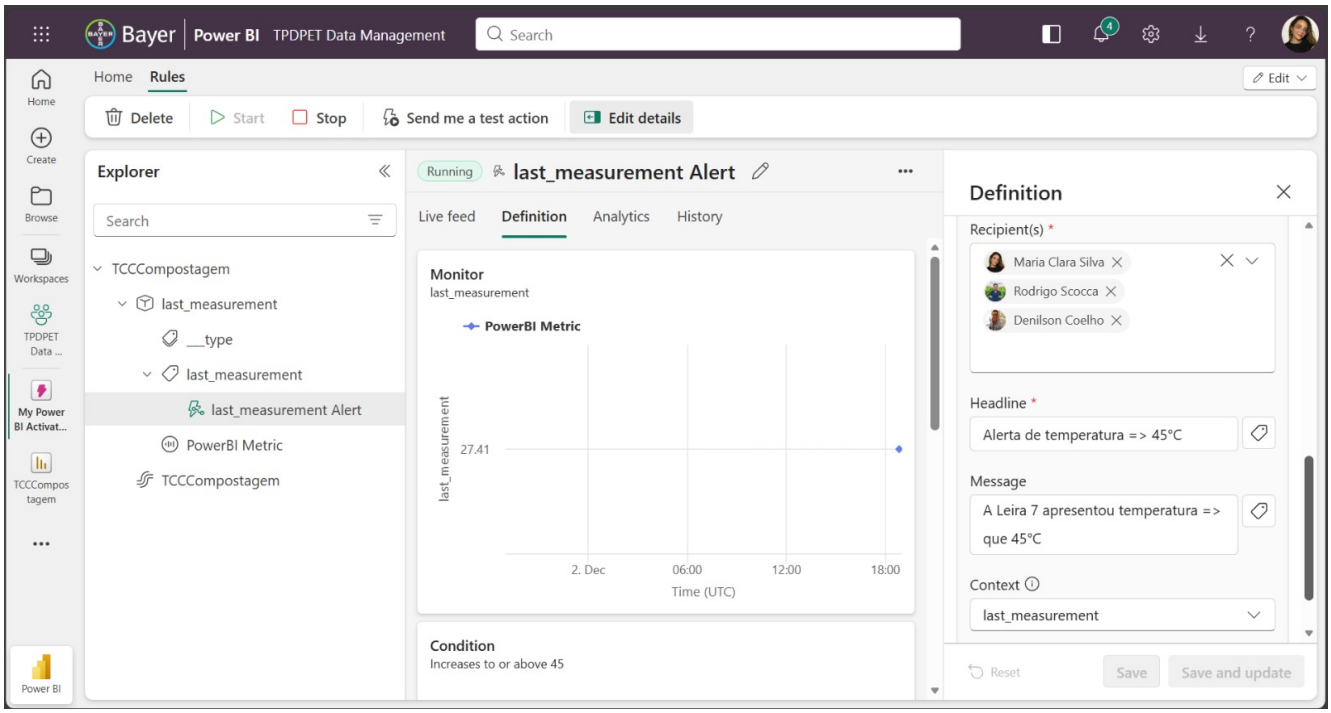


Figura 5. Configuração da regra de monitoramento de temperatura no Fabric Activator.

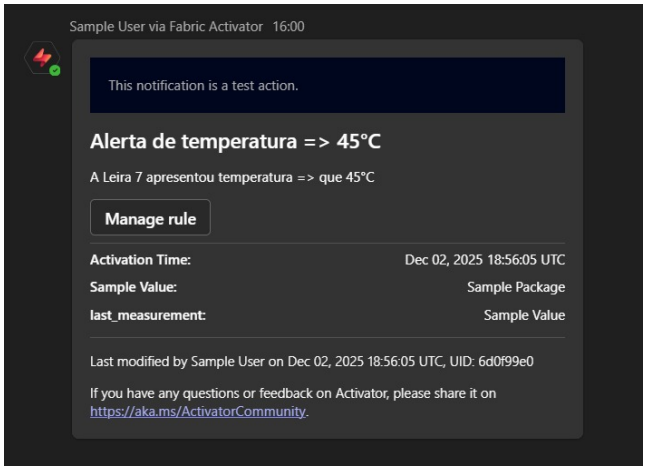


Figura 6. Notificação de teste gerada pelo Fabric Activator e recebida no Microsoft Teams.

acessível e operacionalmente útil.

A automação implementada com o Fabric Activator, por sua vez, acrescentou uma camada de reação automática aos eventos monitorados. A configuração de regras para envio de notificações via Microsoft Teams mostrou-se adequada para reduzir a dependência da inspeção manual contínua, permitindo acompanhamento mais proativo e maior agilidade na percepção de condições térmicas relevantes.

Em síntese, o trabalho demonstra que a combinação entre IoT, análise de dados e automação pode fortalecer práticas de monitoramento aplicadas à compostagem em ambiente operacional real. A principal contribuição está na validação e integração computacional de uma infraestrutura já existente, transformando medições contínuas em suporte efetivo à tomada de decisão.

Como trabalhos futuros, recomenda-se aprofundar a análise temporal dos ciclos de compostagem, incorporando métricas adicionais, como umidade e tempo de permanência em cada fase. Também é possível evoluir para modelos preditivos, capazes de antecipar desvios ou estimar o momento ideal para revolvimento, além de explorar integrações com sensores adicionais ou sistemas corporativos de gestão agrícola. Tais avanços podem aproximar o ambiente de compostagem de um sistema ciberfísico mais automatizado e inteligente.

Declarações complementares

Contribuições dos autores

MCMS trabalhou na concepção do estudo, desenvolvimento da metodologia, análise e interpretação dos dados e redação do manuscrito. JSS, AJFOB, DLAX e RFS atuaram na supervisão, revisão crítica do manuscrito e validação dos resultados.

Conflitos de interesse

Os autores declaram que não há conflitos de interesse.

Disponibilidade de artefatos de pequisa

Os dados e outros materiais criados e/ou usados neste artigo serão disponibilizados mediante solicitação.

Uso de Inteligência Artificial

Os autores utilizaram ferramentas de IA generativa durante o desenvolvimento deste trabalho, incluindo ChatGPT (versão GPT-5.1) e Google Gemini (versão 2.5 Flash). Essas ferramentas foram empregadas exclusivamente para apoiar tarefas não autorais, tais como: busca preliminar por referências e temas relacionados à compostagem e ao monitoramento ambiental; revisão textual para melhoria de clareza e coerência; auxílio na formatação do documento em LaTeX; e elaboração de descrições técnicas de figuras e diagramas presentes no trabalho. Nenhum conteúdo foi utilizado sem verificação, e todas as decisões conceituais, metodológicas e analíticas permanecem sob responsabilidade integral dos autores.

Referências

(2016). ISO/IEC 20922:2016: Information technology – Message Queuing Telemetry Transport (MQTT) v3.1.1.

Albino, J. d. S., Silva, M. A. d., Silva, P. M. M., and Ferreira, T. d. O. (2021). Impacto da compostagem na redução do lixo residencial. Curso Técnico em Meio Ambiente. Acesso em: 9 jun. 2026.

Almeida, D. P. d. *et al*. (2021). Aplicação de dashboards interativos para o monitoramento de indicadores operacionais em tempo real. *Revista Gestão & Tecnologia*, 21(1):85–101. Acesso em: 10 abr. 2025.

Brietzke, D. T. (2017). Avaliação do processo de compostagem considerando a relação carbono/nitrogênio.

EMBRAPA (2024). *Eventos Técnicos & Científicos 1*. Embrapa, Brasília, DF. Acesso em: 20 mar. 2025.

Farooq, A. *et al*. (2022). Composting: A sustainable approach for organic waste management and value-added product formation. *Journal of Cleaner Production*, 343:130833.

Haque, M. S. T., Rouf, K. A., Khan, Z. A., Emran, A., and Zishan, M. S. R. (2019). Design and implementation of an iot based automated agricultural monitoring and control system. In *2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, pages 13–16, Dhaka, Bangladesh. DOI: 10.1109/ICREST.2019.8644212.

Lacanne, C. E. and Lundgren, J. G. (2018). Regenerative agriculture: merging farming and natural resource conservation profitably. *PeerJ*, 6:e4428.

Pedrosa, T. D., Farias, C. A. S., Pereira, R. A., and Farias, E. T. d. R. (2015). Monitoramento dos parâmetros físico-químicos na compostagem de resíduos agroindustriais. *Nativa*, 1(1):1–9. Acesso em: 2 abr. 2025.

Prieto, P. M. C. (2019). Aproveitamento de resíduos sólidos orgânicos mediante a técnica da compostagem para utilização em áreas degradadas. Acesso em: 5 abr. 2025.

Rashid, M. I. *et al*. (2021). Composting process: Enhancing nutrient transformation and microbial activities. *Agronomy*, 11(3):556.

Rodrigues, L. D., Pereira, R. M., Palmeira, D. W. F., Silva, G. H. A., Leite, M. A. F., and Silva, J. S. (2021). Controle de temperatura e de umidade de uma composteira utilizando o esp32. In *Anais do 13. Congresso Brasileiro de Agroinformática (SBIAGRO)*, pages 53–61, Evento Online. Sociedade Brasileira de Computação. Acesso em: 6 abr. 2025.

Schreefel, L. *et al*. (2020). Regenerative agriculture – the soil is the base. *Global Food Security*, 26:100404.

Seibel, T. L. (2023). Sistema de monitoramento e controle de compostagem baseado em internet das coisas (iot). Acesso em: 15 abr. 2025.

Singh, M. and Kulshreshtha, S. (2021). Bioremediation of organic waste through composting: A review. *Environmental Technology & Innovation*, 23:101708.

Artigo Completo

Nintendo Switch como plataforma de computação embarcada: arquitetura e limites de desempenho

Nintendo Switch as an embedded computing platform: architecture and performance limits

Luis Otávio Vilela da Cruz [Universidade Federal de São Carlos (UFSCar) | *luisvilelajuridico@outlook.com*]

Programa de Pós-Graduação em Produção de Conteúdo Multiplataforma (PPGPCM), Centro de Educação e Ciências Humanas (CECH), Universidade Federal de São Carlos (UFSCar), Rod. Washington Luís, km 235, São Carlos, SP, CEP 13565-905, Brasil.

Resumo. Este artigo analisa o Nintendo Switch como estudo de caso de computação embarcada, com ênfase nas escolhas de arquitetura que sustentam seu desempenho em um formato híbrido e alimentado por bateria. A investigação busca compreender em que medida o console combina capacidade de processamento gráfico, eficiência energética e controle térmico, além de situar esse compromisso diante de plataformas com regimes de projeto distintos. O estudo adota abordagem qualitativa, bibliográfica e documental. O corpus reúne documentação pública da Nintendo, da NVIDIA e da Arm, literatura acadêmica sobre sistemas em chip, computação heterogênea CPU-GPU, eficiência por watt e gestão térmica, bem como estudos de benchmark conduzidos em plataformas Tegra X1 e Jetson TX1. A etapa comparativa mobiliza especificações oficiais do PlayStation 4, PlayStation 4 Pro, Xbox One X, Steam Deck e Nintendo Switch 2. A análise concentra-se na família Nintendo Switch original, com dados funcionais do modelo HAC-001(-01), e trata o Switch 2 como contexto geracional, sem transferir para o primeiro console características da arquitetura sucessora. Os resultados indicam que o valor computacional do Switch decorre menos de métricas isoladas de pico e mais do ajuste entre subsistema gráfico, memória compartilhada, orçamento energético, portabilidade e otimização de software. Os benchmarks de plataformas correlatas reforçam que cargas, frequência, acesso à memória e perfil de execução alteram de modo expressivo a relação entre desempenho e energia, o que recomenda cautela ao converter especificações nominais em expectativas práticas. A comparação com consoles estacionários mostra que o Switch trabalha sob restrições físicas e elétricas muito mais severas, enquanto o confronto com o Steam Deck evidencia duas respostas diferentes ao problema da computação portátil. Conclui-se que o Nintendo Switch constitui um exemplo relevante de engenharia embarcada orientada por compromisso: oferece desempenho gráfico suficiente e previsível dentro de limites estreitos de consumo e dissipação, ao mesmo tempo que apresenta restrições claras em cargas sustentadas intensivas em CPU, memória e largura de banda.

Abstract. This article analyzes the Nintendo Switch as a case study in embedded computing, emphasizing the architectural choices that sustain its performance in a hybrid, battery-powered form factor. The investigation seeks to understand how the console combines graphics processing capability, energy efficiency and thermal control, while placing this design compromise alongside platforms built under different engineering regimes. The study follows a qualitative, bibliographic and documentary approach. Its corpus includes public documentation from Nintendo, NVIDIA and Arm, academic literature on system-on-chip design, heterogeneous CPU-GPU computing, performance per watt and thermal management, as well as benchmark studies conducted on Tegra X1 and Jetson TX1 platforms. The comparative stage uses official specifications for PlayStation 4, PlayStation 4 Pro, Xbox One X, Steam Deck and Nintendo Switch 2. The analysis focuses on the original Nintendo Switch family, using functional data from model HAC-001(-01), and treats Switch 2 as generational context without transferring features of the successor architecture to the first console. The results indicate that the computational value of the Switch derives less from isolated peak metrics than from the fit among its graphics subsystem, shared memory, energy budget, portability and software optimization. Benchmarks on related platforms reinforce that workload, frequency, memory access and execution profile substantially affect the relationship between performance and energy, which calls for caution when converting nominal specifications into practical expectations. The comparison with stationary consoles shows that the Switch operates under much tighter physical and electrical constraints, while the contrast with Steam Deck reveals two different responses to the challenge of portable computing. The article concludes that the Nintendo Switch is a relevant example of compromise-oriented embedded engineering: it delivers sufficient and predictable graphics performance within narrow limits of power and heat dissipation, while retaining clear constraints in sustained CPU-intensive, memory-intensive and bandwidth-intensive workloads.

Palavras-chave: Nintendo Switch, computação embarcada, arquitetura de hardware, sistema em chip, eficiência energética

Keywords: Nintendo Switch, embedded computing, hardware architecture, system-on-chip, energy efficiency

Recebido/Received: 06 06 2026 • **Aceito/Accepted:** 11 08 2026 • **Publicado/Published:** 15 10 2026

Publicado sob a Licença Pública Internacional Creative Commons Atribuição 4.0 (CC BY 4.0)

1 Introdução

O desempenho de um sistema computacional resulta da relação entre arquitetura, software e condições de operação. Frequência, quantidade de núcleos e capacidade teórica de ponto flutuante oferecem sinais úteis, porém ganham sentido apenas quando lidos ao lado da hierarquia de memória, do paralelismo disponível, do consumo de energia e do tipo de carga executada. Em plataformas móveis e embarcadas, esses fatores se tornam ainda mais interdependentes porque o projeto precisa transformar um orçamento elétrico pequeno em trabalho computacional estável, sem ultrapassar os limites de bateria, temperatura, área e custo [Hennessy and Patterson, 2017; Patterson and Hennessy, 2016; Suárez *et al*., 2024].

O Nintendo Switch oferece um caso particularmente claro desse problema. O produto reúne funções de console doméstico e dispositivo portátil em uma plataforma baseada em *System-on-Chip* (SoC), ou sistema em chip. Na página técnica do modelo HAC-001(-01), a Nintendo informa processador NVIDIA Custom Tegra, tela de 6,2 polegadas com resolução de 1280×720, saída de vídeo de até 1920×1080 em modo TV, 32 GB de armazenamento interno, bateria de 4310 mAh e autonomia estimada entre 4,5 e 9 horas. A fabricante também registra consumo aproximado de 4 W durante jogos nos modos portátil e de mesa e de 7 W em modo TV [Nintendo, sdc,s]. O conjunto aponta para uma plataforma cujo valor técnico depende da convivência entre mobilidade, computação gráfica e disciplina energética.

O recorte temporal deste trabalho abrange a família Nintendo Switch original, lançada em 2017, com atenção aos dados públicos do modelo HAC-001(-01), disponibilizado a partir de 2019 com menor consumo e maior autonomia. A própria Nintendo atribui essa revisão ao emprego de tecnologia de semicondutores mais eficiente no SoC, mantendo a proposta híbrida do console [Nintendo, 2021, sda]. Essa delimitação evita misturar a primeira versão HAC-001, a revisão HAC-001(-01), o Switch Lite e o Switch OLED como se fossem configurações idênticas. Sempre que o texto apresenta valores de autonomia e consumo, a referência é o HAC-001(-01), pois é esse o modelo detalhado na documentação oficial consultada.

A existência do Nintendo Switch 2 também precisa ser reconhecida. O sucessor chegou ao mercado em 5 de junho de 2025 e utiliza outro processador customizado pela NVIDIA, tela de 7,9 polegadas com resolução de 1920×1080, armazenamento UFS de 256 GB e saída de vídeo de até 4K a 60 quadros por segundo em modo TV [Nintendo, 2025, sdb]. Esses dados indicam uma transição geracional relevante, mas a documentação pública ainda não expõe todos os parâmetros internos necessários para uma comparação microarquitetural completa. Por essa razão, o Switch 2 aparece neste artigo como referência de contexto e como direção para pesquisa futura. O objeto principal permanece o Switch da primeira geração.

A pergunta que orienta a investigação é a seguinte: em que medida a arquitetura do Nintendo Switch permite desempenho computacional relevante sob restrições de portabilidade, consumo energético e dissipação térmica? A tese defendida sustenta que o interesse técnico do console está no equilíbrio de projeto. Seu hardware favorece computação visual paralela, integração de componentes e operação previsível em um envelope energético estreito. Esse arranjo reduz a margem para cargas sustentadas intensivas em CPU e memória, mas possibilita uma experiência gráfica coerente com a finalidade do produto [Mittal and Vetter, 2015a,b; Wang *et al*., 2020].

O objetivo geral consiste em analisar o hardware do Nintendo Switch sob a perspectiva da computação embarcada. Como objetivos específicos, o estudo busca: descrever os componentes confirmados em documentação pública; interpretar o papel da colaboração entre CPU e GPU; examinar o vínculo entre desempenho, energia e temperatura; discutir as limitações decorrentes da memória compartilhada e do perfil móvel; e comparar o console com plataformas estacionárias e portáteis que adotam outras prioridades de engenharia. A análise comparativa inclui PlayStation 4, PlayStation 4 Pro, Xbox One X, Steam Deck e, em caráter contextual, Nintendo Switch 2 [Sony Interactive Entertainment, sd; Microsoft, sd; Valve, sd; Nintendo, sdb].

A contribuição do artigo está em reunir fontes que normalmente aparecem separadas. A documentação da Nintendo descreve o produto em termos funcionais; os materiais da NVIDIA e da Arm ajudam a interpretar a família tecnológica associada ao SoC; os estudos acadêmicos de benchmark em Tegra X1 e Jetson TX1 mostram como energia, memória e carga alteram o desempenho observado. O cruzamento dessas evidências permite tratar o Switch com maior rigor do que uma leitura baseada apenas em números promocionais ou em comparações lineares de Teraflops [NVIDIA, 2015, 2016; Nikolskiy *et al*., 2016; Bianco *et al*., 2018; Rodrigues *et al*., 2017].

2 Referencial teórico

2.1 Sistemas em chip e computação heterogênea

Um SoC reúne, no mesmo circuito integrado, unidades de processamento, controladores de memória, interfaces de entrada e saída e outros blocos especializados. Essa integração reduz distâncias físicas, simplifica o consumo do conjunto e favorece produtos compactos. O ganho vem acompanhado de limites: CPU, GPU e demais unidades disputam recursos térmicos, elétricos e, muitas vezes, a mesma memória principal. O comportamento do sistema depende de como esses componentes compartilham o orçamento disponível [Patterson and Hennessy, 2016; Wang *et al*., 2020].

A computação heterogênea parte da ideia de que unidades diferentes devem assumir tarefas compatíveis com suas características. CPUs generalistas respondem bem a fluxos de controle, dependências seriais e operações de baixa latência. GPUs oferecem alto *throughput* quando a carga pode ser dividida em muitas operações semelhantes. O aproveitamento conjunto dessas unidades tende a produzir resultados melhores do que a busca por um único processador capaz de executar todo tipo de trabalho com a mesma eficiência [Mittal and Vetter, 2015a].

No ambiente móvel, a heterogeneidade também aparece na organização dos núcleos de CPU. A tecnologia big.LITTLE combina núcleos orientados a desempenho e núcleos orientados a economia, ajustando o perfil de execução às necessi-

dades do momento. A proposta atende a uma característica comum de dispositivos portáteis: picos de processamento alternados com períodos de baixa demanda. A eficiência resulta da capacidade de entregar potência quando a carga exige e reduzir consumo quando a intensidade cai [Arm, 2013, 2016, 2018].

O Tegra X1 foi apresentado pela NVIDIA como um SoC móvel voltado a gráficos, visão computacional e processamento paralelo. A documentação pública descreve CPU Arm de 64 bits, GPU Maxwell com 256 núcleos CUDA e organização orientada a desempenho por watt [NVIDIA, 2015]. O Jetson TX1, baseado na mesma família tecnológica, expõe uma configuração com CPU Cortex-A57, GPU Maxwell, 4 GB de LPDDR4 e largura de banda de 25,6 GB/s [NVIDIA, 2016]. Esses documentos ajudam a compreender o espaço arquitetural do Switch, mas a customização do console impede a equivalência automática entre o kit Jetson e o produto da Nintendo.

2.2 Desempenho por watt e limites térmicos

Em sistemas de mesa, o aumento de desempenho pode ser acompanhado por fontes maiores, dissipadores robustos e ventilação mais ampla. Produtos portáteis trabalham com outra margem. A energia armazenada na bateria precisa sustentar processamento, tela, memória, conectividade e demais subsistemas. O calor gerado permanece concentrado em um volume pequeno e precisa ser removido sem comprometer conforto, ruído ou vida útil [Ghadim *et al*., 2025; Suárez *et al*., 2024].

A eficiência energética de GPUs envolve mais do que reduzir frequência. Arquiteturas recentes recorrem a técnicas como *clock gating*, compressão de dados, reorganização do escalonamento e redução de acessos à memória externa. Cada acesso evitado pode economizar energia e liberar largura de banda para trabalho útil. Em plataformas integradas, esse efeito se propaga por todo o sistema porque GPU e CPU compartilham memória e envelope térmico [Mittal and Vetter, 2015b; NVIDIA, sd, 2015].

A temperatura atua como limite operacional. Quando a densidade de potência aumenta, o sistema pode reduzir frequência para permanecer dentro da faixa segura. Esse comportamento, conhecido como limitação térmica, afeta principalmente cargas longas, nas quais o pico inicial de desempenho perde importância diante da capacidade de sustentação. A avaliação de um console portátil deve considerar, portanto, o que o hardware consegue manter ao longo do tempo, e não apenas o que atinge em condições breves [Ghadim *et al*., 2025; Hennessy and Patterson, 2017].

A literatura de arquitetura também recomenda cautela com métricas de pico. Teraflops medem uma capacidade teórica específica de operações de ponto flutuante, sem representar diretamente taxa de quadros, qualidade visual ou desempenho geral. Diferenças de microarquitetura, compiladores, APIs, memória e otimização alteram o aproveitamento prático. A comparação entre plataformas precisa preservar o contexto de cada número e evitar que uma métrica isolada se transforme em explicação total do sistema [Hennessy and Patterson, 2017; Mittal and Vetter, 2015a].

2.3 Benchmarks em plataformas embarcadas correlatas

Benchmarks servem para observar a arquitetura sob cargas definidas. Em sistemas embarcados, um desenho experimental útil mede tempo, consumo, largura de banda e comportamento térmico ao lado da qualidade do resultado. O mesmo dispositivo pode apresentar respostas distintas quando a carga muda de computação intensiva para memória intensiva, ou quando o perfil energético altera frequência e disponibilidade dos núcleos [Rodrigues *et al*., 2017; Bianco *et al*., 2018; Dolz *et al*., 2023].

Nikolskiy *et al*. [2016] aplicaram o modelo Roofline e algoritmos de dinâmica molecular a plataformas Tegra K1 e X1. O estudo avaliou desempenho, utilização de pico, consumo e eficiência energética em CPU Arm e GPU NVIDIA. Os resultados mostraram que a eficiência depende do ponto em que a carga se encontra entre os limites de computação e de memória, reforçando a necessidade de interpretar o SoC a partir do perfil do algoritmo.

Bianco *et al*. [2018] compararam redes neurais em uma estação com GPU dedicada e em um Jetson TX1. O trabalho considerou precisão, complexidade, uso de memória e tempo de inferência. A variação entre modelos revelou que a arquitetura mais compacta pode ser adequada a determinadas redes, enquanto outras sofrem com custos de memória e operação. Rodrigues *et al*. [2017] chegaram a conclusão compatível ao medir energia em camadas de redes convolucionais no Jetson TX1: a observação fina do consumo expõe diferenças que métricas agregadas escondem.

Esses benchmarks não reproduzem jogos do Nintendo Switch. Sua utilidade está em oferecer evidência experimental sobre uma plataforma Tegra X1 aberta a instrumentação, próxima da família tecnológica discutida. O uso é analógico e controlado: os estudos ajudam a entender tendências de energia, memória e paralelismo, enquanto as conclusões sobre o console permanecem limitadas ao que a Nintendo confirma e ao que pode ser inferido com prudência [NVIDIA, 2015, 2016; Nikolskiy *et al*., 2016; Bianco *et al*., 2018].

3 Metodologia

3.1 Desenho da pesquisa e composição do corpus

A pesquisa é qualitativa, bibliográfica e documental. A estratégia foi escolhida porque o objeto combina documentação comercial, manuais de arquitetura e literatura experimental produzida em plataformas relacionadas. O corpus foi organizado em cinco grupos: documentação oficial do Nintendo Switch; documentação técnica da NVIDIA e da Arm; artigos sobre computação heterogênea, energia e gestão térmica; benchmarks de Tegra X1 e Jetson TX1; e páginas oficiais dos consoles empregados na comparação [Nintendo, sdc; NVIDIA, 2015; Arm, 2013; Mittal and Vetter, 2015a; Nikolskiy *et al*., 2016; Sony Interactive Entertainment, sd].

A coleta privilegiou fontes primárias. Especificações de produto foram obtidas em páginas dos fabricantes. Características de arquitetura foram retiradas de manuais e relatórios técnicos da NVIDIA e da Arm. A discussão conceitual e os resultados experimentais vieram de livros e trabalhos revisados por pares. Fontes jornalísticas, fóruns, desmontagens e

bancos de dados comunitários foram consultados apenas na etapa exploratória e ficaram fora da base probatória central. Essa decisão reduz o risco de incorporar frequências, revisões de chip e configurações de memória cuja confirmação oficial permanece incompleta.

O processo analítico ocorreu em quatro etapas. Primeiro, foram extraídos os dados que a Nintendo associa diretamente ao modelo HAC-001(-01), como tela, armazenamento, bateria, autonomia e consumo. Em seguida, a documentação da família Tegra foi lida para identificar princípios de integração, paralelismo e eficiência. A terceira etapa confrontou essas características com resultados de benchmarks em plataformas correlatas. Por fim, os achados foram comparados com consoles estacionários e portáteis, sempre preservando a categoria de projeto de cada dispositivo.

3.2 Sigilo industrial e assimetria documental

A documentação pública do Switch apresenta uma assimetria importante. A Nintendo divulga informações suficientes para uso, suporte e comparação comercial, mas resume CPU e GPU à expressão "processador NVIDIA Custom Tegra" [Nintendo, sdc]. Parâmetros microarquiteturais, frequências por modo, organização completa da memória e detalhes de firmware ficam fora da ficha pública. Esse padrão é coerente com a proteção de conhecimento técnico proprietário em setores de alta intensidade tecnológica.

Segredos industriais podem abranger desenhos, processos, configurações e informações técnicas cuja confidencialidade preserva vantagem competitiva. A Organização Mundial da Propriedade Intelectual inclui projetos, parâmetros de desempenho e conhecimento de fabricação entre os conteúdos passíveis de proteção [World Intellectual Property Organization, 2024]. Estudos sobre divulgação corporativa também mostram que empresas expõem informações de caráter geral enquanto restringem dados proprietários que poderiam reduzir sua vantagem estratégica [Glaeser, 2018]. No caso do Switch, o uso de um SoC customizado e a integração estreita entre fabricante do console e fornecedora do chip tornam plausível essa seletividade documental, sem que seja necessário atribuir à Nintendo uma motivação específica além da proteção usual de tecnologia proprietária.

Esse contexto justifica o cruzamento com manuais da NVIDIA e da Arm. A escolha procura preencher lacunas conceituais, não reconstruir um projeto confidencial. Quando o artigo descreve a família Tegra X1, o objetivo é entender a lógica do ecossistema tecnológico: CPU Arm, GPU Maxwell, memória compartilhada e orientação a baixo consumo. O texto evita converter dados do Jetson TX1 em especificações exatas do Switch, pois produto comercial, placa de desenvolvimento e revisão customizada podem operar com blocos habilitados, frequências e limites diferentes [NVIDIA, 2015, 2016].

3.3 Uso de benchmarks e hierarquia de evidências

Os benchmarks incluídos foram selecionados por três critérios: uso de Tegra X1 ou Jetson TX1; medição conjunta de desempenho e energia ou memória; e descrição metodológica suficiente para interpretar o resultado. Nikolskiy *et al*. [2016]; Rodrigues *et al*. [2017]; Bianco *et al*. [2018] atendem a esses requisitos e cobrem cargas distintas, como dinâmica molecular e redes neurais. Essa diversidade é útil porque mostra o quanto a resposta de um SoC varia conforme o perfil de execução.

Benchmarks alternativos do próprio console existem em análises técnicas e ambientes de *homebrew*, mas apresentam obstáculos para este desenho de pesquisa. Parte deles depende de desbloqueio, firmware modificado, medição indireta ou configurações sem documentação estável. A comparabilidade também sofre quando cada teste utiliza jogos, cenas e versões de software diferentes. O artigo opta por uma base menor e mais controlada, construída com documentação oficial e estudos acadêmicos reprodutíveis em plataformas correlatas. Essa decisão favorece validade documental, embora limite a proximidade com o comportamento de jogos comerciais.

A hierarquia de evidências adotada segue a seguinte ordem: dados confirmados pela fabricante do produto; documentação técnica da fornecedora do SoC e da arquitetura Arm; resultados experimentais revisados por pares; e inferências comparativas derivadas do conjunto. Toda inferência é apresentada como interpretação, sem receber o mesmo estatuto de uma especificação oficial. Essa separação é necessária porque o sigilo industrial impede uma descrição completa do hardware e porque a customização do Switch reduz a transferibilidade direta dos resultados obtidos em Jetson TX1.

3.4 Critérios de comparação e limites do método

A comparação entre consoles utiliza seis critérios: categoria de uso, organização de CPU e GPU, memória divulgada, resolução de saída, orçamento energético e finalidade do projeto. Teraflops aparecem apenas quando o fabricante os publica e funcionam como indicador parcial de capacidade gráfica. O artigo evita construir uma pontuação única, pois combinar dispositivos híbridos, portáteis e estacionários em um ranking apagaria diferenças fundamentais de alimentação, dissipação e software [Hennessy and Patterson, 2017; Mittal and Vetter, 2015a].

A análise não mede diretamente temperatura, taxa de quadros, latência ou consumo do Switch. Também não executa cargas padronizadas nos consoles comparados. As conclusões têm alcance interpretativo: explicam relações arquiteturais apoiadas em documentos e benchmarks correlatos, mas não substituem ensaio experimental no hardware. A ausência de testes próprios é tratada como limite de validade, e não como dado a ser preenchido por estimativas.

4 Análise e discussão

4.1 O Nintendo Switch como sistema embarcado híbrido

O Switch foi concebido para alternar entre uso portátil, modo de mesa e conexão a um televisor. Essa flexibilidade impõe um requisito incomum: o mesmo núcleo computacional precisa funcionar com bateria e com alimentação externa, mantendo compatibilidade de software e experiência reconhecível. A variação entre 720p na tela integrada e até 1080p na saída para TV mostra como resolução e modo de uso participam do gerenciamento da carga gráfica [Nintendo, sdc].

A revisão HAC-001(-01) reforça a centralidade da eficiência. A Nintendo manteve a proposta funcional e ampliou a autonomia por meio de um SoC fabricado com tecnologia mais eficiente. O ganho ocorreu sem transformar o produto em uma nova geração, o que sugere que redução de consumo foi tratada como melhoria suficiente para alterar o comportamento prático do console [Nintendo, 2021, sda]. A plataforma oferece, assim, um exemplo de como processo de fabricação e integração podem gerar valor mesmo quando a arquitetura visível ao usuário muda pouco.

O consumo aproximado informado pela Nintendo coloca o sistema em um regime distante dos consoles domésticos tradicionais. Quatro watts durante jogos em modo portátil precisam sustentar processamento, memória, tela, áudio e interfaces. O número deve ser lido como estimativa de uso, sujeita a variação por jogo e função, mas serve para dimensionar a restrição sob a qual o projeto opera [Nintendo, sdc]. A experiência final depende de selecionar frequências, resolução e carga gráfica compatíveis com esse teto.

4.2 Organização heterogênea do processamento

A documentação do Tegra X1 descreve uma combinação de CPU Arm e GPU Maxwell voltada a cargas móveis e visuais [NVIDIA, 2015]. Essa organização oferece um caminho natural para jogos. A CPU administra lógica, sistema, entrada e tarefas com dependências; a GPU assume a maior parte do trabalho paralelo de renderização. A eficiência surge quando o software evita deslocar para a CPU operações que a GPU executa melhor e reduz sincronizações desnecessárias entre os dois lados [Mittal and Vetter, 2015a].

O valor da heterogeneidade depende da distribuição real da carga. Um jogo com física, inteligência artificial e simulação intensas pode pressionar a CPU mesmo que a GPU ainda tenha margem. Outro título pode limitar-se pela largura de banda de memória ou pelo custo de efeitos visuais. Por isso, a mesma plataforma apresenta desempenho diferente entre gêneros, motores e cenas. O hardware estabelece o espaço de possibilidades; a implementação decide como esse espaço será ocupado [Wang *et al*., 2020; Dolz *et al*., 2023].

A estabilidade de uma plataforma fechada favorece otimizações dirigidas. Desenvolvedores trabalham com um conjunto conhecido de recursos, APIs e limites, o que permite ajustar resolução dinâmica, complexidade geométrica, qualidade de sombras e frequência de atualização. Esse processo não elimina os limites físicos, mas melhora o uso do orçamento. Em consoles, a longevidade comercial costuma produzir ganhos de software que seriam menos previsíveis em ambientes com ampla variação de hardware [Hennessy and Patterson, 2017; Mittal, 2019].

4.3 GPU Maxwell e computação visual

A GPU Maxwell ocupa posição central na interpretação do Switch. A NVIDIA associa essa arquitetura a melhorias de escalonamento, eficiência por núcleo e hierarquia de memória, além de técnicas destinadas a reduzir tráfego externo [NVIDIA, 2015, sd]. Em um dispositivo portátil, cada acesso evitado à DRAM reduz custo energético e pode aliviar um gargalo compartilhado com a CPU.

O paralelismo gráfico atende bem a operações de rasterização, sombreamento e composição de quadros. Isso explica por que o Switch consegue produzir cenas tridimensionais complexas com consumo contido, ainda que a resolução e a qualidade dos efeitos precisem ser ajustadas. A força da plataforma aparece na repetição de operações visuais em larga escala, não em uma capacidade generalista uniforme [Mittal and Vetter, 2015a,b].

Os benchmarks de Tegra X1 e Jetson TX1 ajudam a qualificar essa leitura. O estudo de dinâmica molecular mostra que a GPU integrada pode alcançar boa eficiência quando a carga explora paralelismo e permanece próxima do regime computacional adequado [Nikolskiy *et al*., 2016]. Os testes de redes neurais indicam que modelos leves e bem adaptados obtêm resultados competitivos em plataforma embarcada, enquanto arquiteturas pesadas ampliam custos de memória e tempo [Bianco *et al*., 2018; Rodrigues *et al*., 2017]. O padrão comum é a sensibilidade ao perfil de trabalho.

Para jogos, essa sensibilidade se traduz em escolhas de desenvolvimento. Técnicas de resolução dinâmica, redução de amostras, simplificação de materiais e reconstrução temporal podem preservar a estabilidade visual sem exigir o mesmo volume de operações de um console fixo. O resultado não decorre apenas da GPU, mas do encaixe entre motor, ferramentas e restrição energética. A plataforma recompensa projetos capazes de planejar o custo de cada quadro.

4.4 Memória compartilhada e largura de banda

Sistemas integrados costumam compartilhar a memória principal entre CPU e GPU. A solução reduz componentes e consumo, mas cria concorrência por largura de banda. A documentação do Jetson TX1 informa LPDDR4 com 25,6 GB/s, valor coerente com um módulo embarcado da família Tegra X1 [NVIDIA, 2016]. A Nintendo não detalha na página pública consultada a capacidade e a largura de banda da memória do Switch, razão pela qual este artigo evita fixar esses números como especificação oficial [Nintendo, sdc].

O compartilhamento favorece transferência direta de dados, pois CPU e GPU acessam um espaço comum. Ao mesmo tempo, cenas com muitas texturas, leituras e escritas podem saturar o subsistema. O desempenho passa a depender de cache, compressão e organização dos acessos. Esse limite ajuda a explicar por que largura de banda e eficiência de memória são tão importantes quanto a capacidade aritmética nominal [Mittal and Vetter, 2015b; NVIDIA, sd].

A pressão de memória também afeta a CPU. Quando a GPU consome grande parte da largura de banda, tarefas de lógica e sistema podem sofrer maior latência. O problema se intensifica em cargas que movimentam muitos dados e realizam poucas operações por byte. O modelo Roofline utilizado nos benchmarks de Tegra X1 torna essa diferença visível ao separar regiões limitadas por computação e por memória [Nikolskiy *et al*., 2016].

4.5 Energia, temperatura e desempenho sustentado

A eficiência energética do Switch precisa ser lida como propriedade do conjunto. Processador, tela, bateria, refrigeração e firmware formam um sistema. A revisão de 2019 mostra que uma mudança no SoC pode ampliar autonomia sem alterar a

proposta do produto [Nintendo, 2021]. Esse resultado seria pouco perceptível em uma comparação baseada somente em teraflops, mas é decisivo para o uso portátil.

O calor limita a frequência que o sistema consegue sustentar. Em um chassi compacto, a dissipação precisa manter componentes e superfície em faixas seguras. A literatura de gestão térmica destaca que densidade de potência, resistência térmica e duração da carga determinam a formação de pontos quentes [Ghadim *et al*., 2025]. O desempenho observado durante poucos segundos pode diferir do comportamento após longos períodos de jogo.

A operação em múltiplos modos permite ajustar o equilíbrio. No modo TV, a alimentação externa e a ausência de dependência imediata da bateria ampliam a margem, enquanto o modo portátil prioriza autonomia e conforto térmico. A Nintendo divulga consumos aproximados diferentes para esses cenários [Nintendo, sdc]. A variação confirma que o Switch não possui um único estado de desempenho, mas um conjunto de perfis coerentes com a forma de uso.

4.6 Comparação com outros consoles

A comparação a seguir organiza dados públicos dos fabricantes. A tabela reúne categorias distintas e, por isso, funciona como mapa de regimes de projeto. Os números não formam um ranking direto. A leitura deve considerar que consoles fixos recebem energia contínua e dispõem de maior volume para refrigeração, enquanto dispositivos portáteis distribuem o orçamento entre processamento, tela e bateria [Hennessy and Patterson, 2017; Ghadim *et al*., 2025].

4.6.1 Switch, PlayStation 4 e PlayStation 4 Pro

O PlayStation 4 utiliza CPU x86-64 AMD Jaguar de oito núcleos, GPU Radeon de 1,84 TFLOPS e 8 GB de GDDR5. O PlayStation 4 Pro mantém a família de CPU, amplia a GPU para 4,20 TFLOPS e trabalha com meta de saída 4K/HDR [Sony Interactive Entertainment, sd]. Esses consoles foram projetados para uso estacionário e aceitam consumos máximos muito superiores aos valores divulgados para o Switch.

A diferença de potência disponível muda todo o espaço de decisão. O PS4 pode dedicar mais energia à GPU, empregar memória GDDR5 e utilizar refrigeração maior. O Switch precisa reservar parte do orçamento para tela, bateria e operação portátil. A distância de desempenho gráfico bruto é esperada e coerente com as categorias. O ponto analítico está em entender o que cada sistema entrega por unidade de energia e dentro de seu volume físico.

A memória ilustra essa divergência. A Sony publica 8 GB de GDDR5 e utiliza esse dado como elemento técnico central [Sony Interactive Entertainment, sd]. A Nintendo concentra a ficha do Switch em tela, armazenamento, bateria, portas e modos de uso [Nintendo, sdc]. A assimetria documental reflete prioridades de produto e reforça a cautela metodológica: a comparação deve empregar apenas o que cada fabricante confirma, sem preencher lacunas com estimativas de origem incerta.

4.6.2 Switch e Xbox One X

O Xbox One X foi promovido como console para jogos em 4K, com *frame buffers* de 2160p e 6 TFLOPS de processamento gráfico [Microsoft, sd]. Documentação oficial da plataforma também registra 12 GB de GDDR5 [Microsoft, 2017]. A combinação de memória, GPU e operação estacionária atende a uma meta de alta resolução sustentada.

Diante desse regime, o Switch ocupa outro ponto do espaço de projeto. Sua tela integrada de 720p reduz a quantidade de pixels por quadro e torna viável um orçamento menor. A escolha não representa apenas perda de qualidade; funciona como mecanismo de compatibilização entre carga gráfica, autonomia e temperatura. Elevar a resolução portátil sem alterar o restante do sistema aumentaria pressão sobre GPU, memória e bateria.

A comparação também mostra por que TFLOPS não pode ser tratado como medida universal. A arquitetura Maxwell do Switch, a Radeon do Xbox One X e suas respectivas memórias operam com características diferentes. O número teórico ajuda a dimensionar a distância, mas não prevê sozinho taxa de quadros ou qualidade de imagem. Software, largura de banda, cache e otimização definem quanto da capacidade nominal chega à aplicação [Hennessy and Patterson, 2017; Mittal and Vetter, 2015a].

4.6.3 Switch e Steam Deck

O Steam Deck oferece o contraste portátil mais produtivo. A Valve informa APU AMD de 6 nm, CPU Zen 2 de quatro núcleos e oito *threads*, GPU RDNA 2 com oito unidades de computação e 1,6 TFLOPS, além de 16 GB de LPDDR5. A APU opera entre 4 e 15 W, e a tela possui resolução de 1280 $\times$ 800 [Valve, sd].

As duas plataformas resolvem o problema da portabilidade de formas diferentes. O Steam Deck aproxima-se de um computador pessoal compacto: executa biblioteca de PC, aceita maior variação de software e oferece mais recursos declarados. O Switch mantém um ambiente mais fechado, com menor consumo total divulgado e forte integração entre hardware, sistema e catálogo. O primeiro amplia flexibilidade; o segundo prioriza previsibilidade.

Essa diferença aparece no teto energético. O intervalo de 4 a 15 W da APU do Steam Deck permite maior densidade computacional, mas exige bateria e refrigeração compatíveis [Valve, sd]. O Switch informa aproximadamente 4 W para o sistema em jogo portátil [Nintendo, sdc]. Os números não são equivalentes em ponto de medição, mas revelam escalas distintas. A leitura correta reconhece a vantagem de desempenho do Steam Deck e, ao mesmo tempo, a coerência do Switch dentro de um orçamento menor.

4.7 O Nintendo Switch 2 como contexto geracional

O Switch 2 preserva a forma híbrida e amplia vários parâmetros visíveis: tela de 1080p, armazenamento UFS de 256 GB, saída de até 4K a 60 quadros por segundo e suporte a 120 quadros em resoluções menores. A Nintendo descreve CPU e GPU apenas como processador customizado pela NVIDIA [Nintendo, sdb]. A continuidade da linguagem confirma que a política de divulgação permanece centrada em funções do produto, com detalhes internos mantidos em nível restrito.

A nova geração também altera o equilíbrio energético. A bateria cresce para 5220 mAh, mas a autonomia oficial varia de 2 a 6,5 horas [Nintendo, sdb]. O dado sugere que o aumento de capacidade computacional e de resolução consome parte do ganho físico da bateria. A comparação é ilustrativa, pois

Tabela 1. Síntese comparativa de plataformas e regimes de projeto

Plataforma	Categoria	CPU/GPU divulgadas	Memória divulgada	Saída ou tela	Energia e mobilidade
Nintendo Switch HAC-001(-01)	Híbrido	Processador NVIDIA Custom Tegra	Não detalhada na página pública consultada	Tela 1280 × 720; TV até 1920 × 1080	Bateria de 4310 mAh; cerca de 4 W portátil e 7 W em TV
PlayStation 4	Mesa	CPU AMD Jaguar de 8 núcleos; GPU Radeon de 1,84 TFLOPS	8 GB GDDR5	Foco em saída para TV; suporte a HDR	Consumo máximo divulgado de 165 W
PlayStation 4 Pro	Mesa	CPU AMD Jaguar de 8 núcleos; GPU Radeon de 4,20 TFLOPS	8 GB GDDR5	Suporte a 4K e HDR	Consumo máximo divulgado de 310 W
Xbox One X	Mesa	GPU com 6 TFLOPS; plataforma voltada a 4K	12 GB GDDR5 em documentação técnica da plataforma	*Frame buffers* de 2160p e saída 4K	Alimentação contínua; sem operação portátil
Steam Deck	Portátil de uso geral	APU AMD de 6 nm; CPU Zen 2 de 4 núcleos/8 *threads*; GPU RDNA 2 de 1,6 TFLOPS	16 GB LPDDR5	Tela 1280 × 800	APU configurável entre 4 e 15 W
Nintendo Switch 2	Híbrido, contexto geracional	Processador customizado pela NVIDIA	Não detalhada na página pública consultada	Tela 1920 × 1080; TV até 4K/60; 120 fps em 1080p ou 1440p	Bateria de 5220 mAh; autonomia estimada entre 2 e 6,5 h

Fonte: elaboração própria a partir de documentação oficial [Nintendo, sdc; Sony Interactive Entertainment, sd; Microsoft, sd, 2017; Valve, sd; Nintendo, sdb]. Valores de potência possuem naturezas distintas: consumo aproximado em jogo, potência configurável da APU ou consumo máximo declarado.

autonomia depende de jogo, brilho, conectividade e modo de uso. Ainda assim, ela mostra que avanços de desempenho voltam a pressionar o orçamento móvel.

Uma análise microarquitetural entre Switch e Switch 2 exigiria dados adicionais sobre CPU, GPU, memória, frequências, perfis de energia e comportamento térmico. A documentação oficial consultada ainda não fornece esse conjunto. Este artigo, portanto, registra o sucessor e descreve seus parâmetros públicos sem incorporá-lo ao núcleo inferencial dedicado ao Tegra X1. Estudos futuros podem aplicar cargas equivalentes aos dois consoles e observar consumo, temperatura, taxa de quadros e largura de banda efetiva.

4.8 Síntese dos limites de desempenho

O Switch apresenta três limites principais. O primeiro é a CPU, cuja margem para simulação pesada, física complexa e grande quantidade de entidades é menor do que em consoles estacionários ou portáteis mais recentes. O segundo é a memória compartilhada, que obriga CPU e GPU a administrar o mesmo subsistema. O terceiro é térmico: cargas longas precisam permanecer dentro de uma capacidade de dissipação pequena [NVIDIA, 2015, 2016; Ghadim *et al.*, 2025].

Esses limites não anulam a relevância do sistema. Eles definem o tipo de trabalho em que o console se mostra competente. Aplicações visuais com paralelismo, resolução controlada e orçamento previsível ajustam-se bem à plataforma. Cargas com grande movimentação de dados, dependências seriais ou exigência contínua de alta potência encontram restrições mais cedo [Nikolskiy *et al.*, 2016; Bianco *et al.*, 2018; Mittal and Vetter, 2015a].

A resposta prática envolve coadaptação entre software e hardware. Desenvolvedores reduzem custo de renderização, escolhem técnicas compatíveis com a memória e distribuem tarefas conforme CPU e GPU disponíveis. Essa adaptação explica por que jogos visualmente ambiciosos podem funcionar em um dispositivo com potência nominal inferior. O resultado é fruto de engenharia dirigida, não de equivalência com consoles maiores.

5 Implicações para a avaliação de consoles híbridos

5.1 Métricas de pico e experiência sustentada

A comparação entre consoles costuma começar por números de pico porque eles são fáceis de comunicar. Teraflops, frequência e quantidade de memória condensam parte da capacidade do hardware em indicadores curtos. O problema surge quando essas medidas passam a representar toda a experiência. Um console executa uma sequência de tarefas interdependentes: simulação, animação, áudio, entrada, descompressionamento, transferência de dados e renderização. O desempenho final acompanha o componente mais pressionado em cada momento, não necessariamente a unidade com maior valor nominal [Hennessy and Patterson, 2017; Patterson and Hennessy, 2016].

No Switch, a resolução da tela portátil ajuda a controlar a demanda. Renderizar 1280 × 720 envolve menos pixels do que produzir uma imagem 4K. Essa diferença abre espaço para efeitos, geometria ou estabilidade de quadros dentro do mesmo orçamento. Em consoles fixos, a energia disponível permite perseguir resoluções maiores e manter frequências mais elevadas. A escolha da meta visual, portanto, faz parte da arquitetura do sistema e precisa aparecer na análise ao lado de CPU e GPU [Nintendo, sdc; Sony Interactive Entertainment, sd; Microsoft, sd].

O desempenho sustentado é ainda mais importante em jogos, pois uma sessão pode durar horas. Um pico breve de frequência tem pouco valor quando o sistema precisa reduzir o ritmo após aquecer. Dispositivos portáteis equilibram temperatura interna, conforto das mãos, ruído da ventoinha e duração da bateria. A literatura térmica mostra que a resposta depende do tempo de carga e do caminho de dissipação, o que torna medições instantâneas insuficientes [Ghadim *et al.*, 2025]. Essa observação também explica por que benchmarks de poucos segundos e testes longos podem produzir conclusões diferentes.

A relação entre pico e sustentação sugere uma prática metodológica para pesquisas futuras. Um ensaio experimental

deveria combinar ao menos três janelas: inicial, após estabilização térmica e ao final de uma sessão prolongada. Consumo, frequência e taxa de quadros precisariam ser registrados de forma sincronizada. A comparação ganharia maior validade se cada console executasse cargas equivalentes na mesma resolução de saída ou se os resultados fossem normalizados por pixel processado e energia consumida.

5.2 O software como parte da plataforma

A arquitetura de um console inclui o ambiente de software que organiza o acesso ao hardware. APIs, compiladores, ferramentas de análise e padrões de desenvolvimento influenciam a quantidade de trabalho útil extraída do SoC. Uma plataforma fechada oferece menor diversidade de configuração, o que facilita otimizações específicas ao longo do ciclo comercial. Esse benefício não transforma hardware modesto em equipamento de outra categoria, mas reduz desperdícios e torna o comportamento mais previsível [Mittal and Vetter, 2015a; Mittal, 2019].

No caso do Switch, o desenvolvedor conhece a resolução da tela, os modos de uso e o conjunto de recursos disponível. Essa estabilidade permite ajustar distância de visão, qualidade de sombras, densidade de objetos e resolução dinâmica com objetivos claros. O projeto pode reservar tempo de CPU para lógica e reduzir etapas de GPU quando a cena se torna mais pesada. O desempenho percebido passa a resultar de pequenas decisões distribuídas pelo motor, em vez de depender apenas de uma única otimização.

O Steam Deck trabalha com lógica diferente. Sua proximidade com o ecossistema de PC amplia a biblioteca e a flexibilidade, mas também aumenta a variedade de motores, configurações e expectativas. A Valve compensa parte dessa diversidade com um hardware mais recente e uma APU configurável em faixa energética maior [Valve, sd]. A comparação mostra que portabilidade pode ser alcançada por caminhos distintos: integração rígida e orçamento reduzido, como no Switch, ou maior capacidade generalista com consumo mais amplo, como no Steam Deck.

A chegada do Switch 2 preserva a estratégia de plataforma integrada, embora eleve resolução, armazenamento e capacidade gráfica pública [Nintendo, sdb]. A continuidade sugere que a Nintendo considera o vínculo entre formato híbrido, software e hardware mais importante do que a adoção de uma arquitetura aberta. Para a pesquisa, essa permanência oferece uma oportunidade: comparar duas gerações que mantêm a mesma ideia de uso, mas operam com recursos e limites diferentes.

5.3 Eficiência energética como resultado de projeto

Eficiência por watt não é uma propriedade isolada do processador. Ela depende da carga, do processo de fabricação, do subsistema de memória, da tela e do software. A revisão HAC-001(-01) ilustra esse ponto. A Nintendo ampliou a autonomia com um SoC mais eficiente, enquanto a função do aparelho permaneceu praticamente a mesma [Nintendo, 2021, sda]. O ganho veio da redução do custo energético do conjunto, e não de uma mudança visível na proposta de uso.

Os benchmarks em Tegra X1 e Jetson TX1 reforçam a natureza contextual da eficiência. Cargas com alta intensidade aritmética podem aproveitar melhor a GPU, enquanto tarefas dominadas por memória atingem o limite de largura de banda mais cedo [Nikolskiy *et al.*, 2016]. Redes neurais com tamanhos e estruturas diferentes também apresentam relações distintas entre precisão, tempo e energia [Bianco *et al.*, 2018; Rodrigues *et al.*, 2017]. Assim, afirmar que um SoC é eficiente exige indicar para qual trabalho, em qual configuração e sob qual método de medição.

Essa cautela vale para a tabela comparativa. O consumo máximo do PS4 e do PS4 Pro, a faixa de potência da APU do Steam Deck e o consumo aproximado em jogo do Switch descrevem pontos diferentes do sistema. Ainda assim, o contraste de escala ajuda a compreender as prioridades. Consoles fixos reservam mais energia para processamento e dissipação; portáteis distribuem o orçamento entre componentes e autonomia. A engenharia escolhe onde aceitar perdas e onde preservar desempenho.

O Switch responde a essa escolha com resolução portátil controlada, GPU integrada e software dirigido. O resultado é uma plataforma capaz de sustentar jogos complexos em condições que seriam restritivas para um desenho de desktop reduzido. Ao mesmo tempo, o console depende mais de otimização e enfrenta maior dificuldade quando a carga exige CPU moderna, muita memória ou alto volume de dados. A eficiência observada é real, porém vinculada ao domínio para o qual o produto foi concebido.

5.4 Critérios para uma comparação experimental futura

Uma extensão experimental deste estudo deveria começar por cargas reproduzíveis. Jogos comerciais são relevantes, mas variam em motor, versão e cena. Uma estratégia possível seria combinar jogos disponíveis em mais de uma plataforma com testes sintéticos permitidos pelos sistemas. As cenas precisariam ser gravadas ou automatizadas para reduzir variação entre execuções. Também seria importante separar resolução interna, resolução de saída e técnicas de reconstrução.

A energia pode ser medida na tomada para consoles fixos e na entrada USB-C ou na bateria para portáteis, desde que o método seja documentado. Valores de parede incluem perdas da fonte e acessórios, enquanto medições no SoC isolam apenas parte do sistema. A comparação deve escolher uma fronteira e mantê-la. Para um estudo de experiência portátil, o consumo total do aparelho é mais informativo porque inclui tela, memória, armazenamento e refrigeração.

Temperatura ambiente, brilho da tela, conectividade e estado da bateria também afetam o resultado. O protocolo deve registrar essas condições, repetir execuções e apresentar dispersão. Uma única média esconderia variações relevantes. A literatura de benchmark em plataformas embarcadas mostra que medidas por camada, por fase ou por intervalo de tempo oferecem explicações mais úteis do que um valor agregado [Rodrigues *et al.*, 2017; Dolz *et al.*, 2023].

Por fim, a análise deveria distinguir eficiência técnica e qualidade percebida. Quadros por joule podem favorecer configurações de resolução baixa, mas o usuário também percebe nitidez, estabilidade e latência. Um conjunto equilibrado de métricas incluiria taxa média de quadros, percentis de tempo de quadro, energia por minuto, temperatura estabilizada e resolução efetiva. Esse desenho permitiria avaliar o compro-

misso de cada console sem reduzir a comparação a potência bruta.

6 Limitações e validade da análise

A principal limitação é a ausência de ensaio experimental próprio. O estudo não mede diretamente consumo, temperatura, frequência ou taxa de quadros do Nintendo Switch. Também não executa o mesmo programa nos consoles comparados. Por esse motivo, as conclusões descrevem relações de arquitetura e regimes de projeto, sem afirmar desempenho quantitativo para jogos específicos.

A segunda limitação decorre do sigilo industrial. A ficha da Nintendo não expõe toda a organização do SoC, e os documentos da NVIDIA descrevem a família Tegra X1 em nível mais amplo [Nintendo, sdc; NVIDIA, 2015]. A inferência foi contida por uma hierarquia de evidências, mas parte do comportamento interno permanece fora do domínio público. Novos documentos oficiais podem exigir revisão de trechos interpretativos.

A terceira limitação envolve os benchmarks. Jetson TX1 e Nintendo Switch compartilham vínculos tecnológicos, porém operam com sistemas, firmware, clocks e objetivos diferentes. Os estudos experimentais citados servem para mostrar tendências de memória, energia e perfil de carga, sem representar uma medição substituta do console [Nikolskiy *et al*., 2016; Bianco *et al*., 2018; Rodrigues *et al*., 2017]. Essa distinção reduz o alcance, mas preserva a validade do argumento.

A comparação entre plataformas também mistura métricas de natureza distinta. Consumo máximo, consumo aproximado em jogo e potência configurável da APU não podem ser convertidos diretamente em uma mesma escala. A Tabela 1 mantém as definições originais e explicita a diferença. Um estudo experimental futuro deve utilizar medidores externos, cenas controladas, temperatura ambiente registrada e duração suficiente para observar desempenho sustentado.

7 Considerações finais

A análise confirma que o Nintendo Switch constitui um caso relevante de computação embarcada. O console combina SoC integrado, processamento heterogêneo, GPU orientada a paralelismo e perfis de energia compatíveis com uso portátil. Seu desempenho nasce da articulação entre esses elementos, e não de uma métrica isolada. A revisão HAC-001(-01) evidencia essa lógica ao ampliar autonomia por meio de maior eficiência do SoC, preservando a proposta funcional [Nintendo, 2021, sdc].

Em resposta à pergunta de pesquisa, a arquitetura permite desempenho computacional relevante quando a carga se ajusta ao domínio visual, ao paralelismo disponível e ao limite de energia. A plataforma perde margem em tarefas sustentadas de CPU, em cenários de alta pressão de memória e em metas gráficas próximas às de consoles fixos. Esses limites são parte do compromisso que torna possível reunir tela, bateria, controles e processamento em um equipamento híbrido.

Os benchmarks de Tegra X1 e Jetson TX1 reforçam a interpretação. Eles mostram que desempenho e energia variam com intensidade aritmética, acesso à memória e estrutura do algoritmo [Nikolskiy *et al*., 2016; Bianco *et al*., 2018; Rodrigues *et al*., 2017]. A transferência para o Switch precisa ser cautelosa, mas a evidência experimental sustenta a ideia de que arquiteturas embarcadas ganham valor quando software e hardware são avaliados em conjunto.

A comparação com PlayStation 4, PlayStation 4 Pro e Xbox One X revela a distância entre um sistema alimentado continuamente e um dispositivo portátil. O Steam Deck mostra outra solução móvel, com maior orçamento de APU e orientação próxima ao PC. O Switch ocupa uma posição própria: oferece menos potência bruta, porém organiza consumo, integração e software em torno de uma experiência previsível. O Switch 2 amplia o horizonte geracional e confirma a permanência do formato híbrido, mas demanda estudo separado devido à arquitetura diferente e à documentação ainda parcial [Nintendo, sdb].

Trabalhos futuros podem medir os dois consoles da Nintendo sob cargas comuns, com instrumentação externa de energia e temperatura. Também seria útil comparar cenas de jogos equivalentes, registrar estabilidade de frequência e modelar o efeito da resolução sobre memória e consumo. Esse passo permitiria transformar a interpretação documental em validação experimental, preservando a distinção entre dados públicos, resultados medidos e inferências de arquitetura.

Declarações complementares

Financiamento

O autor declara que esta pesquisa não recebeu financiamento específico de agência pública, comercial ou sem fins lucrativos.

Contribuições dos autores

LOVC foi responsável pela concepção do estudo, definição do problema de pesquisa, metodologia, levantamento e conferência das fontes, análise, redação do manuscrito, revisão crítica e aprovação da versão final para publicação.

Conflitos de interesse

O autor declara que não há conflitos de interesse.

Disponibilidade de artefatos de pequisa

Os dados e materiais utilizados correspondem à documentação técnica pública e à literatura acadêmica citada nas referências. Não foi produzido conjunto de dados proprietário.

Uso de Inteligência Artificial

O autor utilizou ferramenta de IA generativa de modo subsidiário, exclusivamente como apoio operacional na transição do manuscrito da versão em PDF para o código-fonte LaTeX e na conferência pontual da formatação do arquivo. A delimitação do problema, a seleção e a conferência das fontes, a análise, a redação substantiva e a responsabilidade intelectual pelo conteúdo permaneceram sob responsabilidade integral do autor.

Outras informações relevantes

Este estudo é bibliográfico e documental, baseado em fontes públicas, sem coleta de dados com seres humanos, intervenção experimental ou necessidade de apreciação por comitê de ética.

Referências

Arm (2013). big.little technology: The future of mobile. Technical report, Arm, Cambridge. White paper. Acesso em: 2 jun. 2026.

Arm (2016). *ARM Cortex-A57 MPCore Processor Technical Reference Manual*. Arm, Cambridge. Acesso em: 2 jun. 2026.

Arm (2018). *ARM Cortex-A53 MPCore Processor Technical Reference Manual*. Arm, Cambridge. Acesso em: 2 jun. 2026.

Bianco, S., Cadene, R., Celona, L., and Napoletano, P. (2018). Benchmark analysis of representative deep neural network architectures. *IEEE Access*, 6:64270–64277. DOI: 10.1109/ACCESS.2018.2877890.

Dolz, M. F., Barrachina, S., Martínez, H., Castelló, A., Maciá, A., Fabregat, G., and Tomás, A. E. (2023). Performance-energy trade-offs of deep learning convolution algorithms on arm processors. *The Journal of Supercomputing*, 79:9819–9836. DOI: 10.1007/s11227-023-05050-4.

Ghadim, H. B., Godin, A., Veillere, A., Duquesne, M., and Haillot, D. (2025). Review of thermal management of electronics and phase change materials. *Renewable and Sustainable Energy Reviews*, 208:115039. DOI: 10.1016/j.rser.2024.115039.

Glaeser, S. (2018). The effects of proprietary information on corporate disclosure and transparency: Evidence from trade secrets. *Journal of Accounting and Economics*, 66(1):163–193. DOI: 10.1016/j.jacceco.2018.04.002.

Hennessy, J. L. and Patterson, D. A. (2017). *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann, Cambridge, MA, 6 edition.

Microsoft (2017). Your invitation to feel the true power of xbox one x. `https://news.xbox.com/en-us/2017/09/22/play-xbox-one-x-retail-locations/`. Xbox Wire. Acesso em: 28 jul. 2026.

Microsoft (s.d.). Xbox one x. `https://www.xbox.com/en-US/consoles/xbox-one-x`. Página oficial do produto. Acesso em: 6 jun. 2026.

Mittal, S. (2019). A survey on optimized implementation of deep learning models on the nvidia jetson platform. *Journal of Systems Architecture*, 97:428–442. DOI: 10.1016/j.sysarc.2019.01.011.

Mittal, S. and Vetter, J. S. (2015a). A survey of cpu-gpu heterogeneous computing techniques. *ACM Computing Surveys*, 47(4):69:1–69:35. DOI: 10.1145/2788396.

Mittal, S. and Vetter, J. S. (2015b). A survey of methods for analyzing and improving gpu energy efficiency. *ACM Computing Surveys*, 47(2):19:1–19:23. DOI: 10.1145/2636342.

Nikolskiy, V. P., Stegailov, V. V., and Vecher, V. S. (2016). Efficiency of the tegra k1 and x1 systems-on-chip for classical molecular dynamics. In *2016 International Conference on High Performance Computing & Simulation (HPCS)*, pages 682–689, Innsbruck. IEEE. DOI: 10.1109/HPCSim.2016.7568401.

Nintendo (2021). Ask the developer vol. 2, nintendo switch – oled model. `https://www.nintendo.com/en-gb/News/2021/October/Ask-the-Developer-Vol-2-Nintendo-Switch-OLED-Model-2055360.html`. Entrevista técnica institucional. Acesso em: 28 jul. 2026.

Nintendo (2025). Nintendo switch 2 launches june 5, bringing new forms of game communication to life. `https://www.nintendo.com/us/whatsnew/nintendo-switch-2-launches-june-5-bringing-new-forms-of-game-communication-to-life/`. Notícia institucional publicada em 2 abr. 2025. Acesso em: 28 jul. 2026.

Nintendo (s.d.a). How long does the battery charge last on nintendo switch? `https://en-americas-support.nintendo.com/app/answers/detail/a_id/46835/`. Documento de suporte. Acesso em: 28 jul. 2026.

Nintendo (s.d.b). Nintendo switch 2 tech specs. `https://www.nintendo.com/us/gaming-systems/switch-2/tech-specs/`. Acesso em: 28 jul. 2026.

Nintendo (s.d.c). Technical specs – nintendo switch console (hac-001(-01)). `https://www.nintendo.com/en-ca/switch/tech-specs/`. Acesso em: 2 jun. 2026.

NVIDIA (2015). Nvidia tegra x1: Nvidia's new mobile superchip. Technical report, NVIDIA, Santa Clara, CA. White paper. Acesso em: 2 jun. 2026.

NVIDIA (2016). Jetson tx1 module product sheet. Technical report, NVIDIA, Santa Clara, CA. Acesso em: 2 jun. 2026.

NVIDIA (s.d.). Maxwell tuning guide. Technical report, NVIDIA. Acesso em: 2 jun. 2026.

Patterson, D. A. and Hennessy, J. L. (2016). *Computer Organization and Design ARM Edition: The Hardware/Software Interface*. Morgan Kaufmann, Cambridge, MA.

Rodrigues, C., Riley, G., and Luján, M. (2017). Fine-grained energy profiling for deep convolutional neural networks on the jetson tx1. In *2017 IEEE International Symposium on Workload Characterization (IISWC)*, pages 114–115. IEEE. DOI: 10.1109/IISWC.2017.8167764.

Sony Interactive Entertainment (s.d.). Especificações técnicas do playstation 4 e do playstation 4 pro. `https://www.playstation.com/pt-br/ps4/tech-specs/`. Acesso em: 6 jun. 2026.

Suárez, D., Almeida, F., and Blanco, V. (2024). Comprehensive analysis of energy efficiency and performance of arm and risc-v socs. *The Journal of Supercomputing*, 80:12771–12789. DOI: 10.1007/s11227-024-05946-9.

Valve (s.d.). Especificações do steam deck. `https://www.steamdeck.com/pt-br/tech`. Acesso em: 6 jun. 2026.

Wang, S., Pathania, A., and Mitra, T. (2020). Neural network inference on mobile socs. *IEEE Design & Test*, 37(5):50–57. DOI: 10.1109/MDAT.2020.2968258.

World Intellectual Property Organization (2024). *WIPO Guide to Trade Secrets and Innovation*. WIPO, Geneva.

Relato de Experiência

Automação do Onboarding Corporativo com Agente Conversacional Baseado em IA: Um Relato de Experiência

Corporate Onboarding Automation with an AI-Based Conversational Agent: An Experience Report

Vinicius de Castro Coutinho [Pesquisador independente | *viniciuscoutinho_@hotmail.com*]

Pesquisador independente, Fortaleza, Ceará, Brasil.

Resumo. O *onboarding*, processo de integração de novos colaboradores, costuma gerar um volume grande de dúvidas repetidas que acabam caindo sobre a equipe de Recursos Humanos (RH). Este artigo relata o desenvolvimento de um agente conversacional baseado em Inteligência Artificial (IA), integrado ao *Slack*, para responder a essas dúvidas. O agente foi construído e testado no cenário de uma empresa hipotética, chamada aqui de Empresa E: uma organização de tecnologia com 85 colaboradores, nove departamentos e trabalho híbrido, cujos dados foram todos criados para este estudo. A solução roda na plataforma de automação *n8n* e usa o modelo de linguagem *Gemini 3.5 Flash*, do Google, uma base de conhecimento estruturada com os departamentos da empresa e uma memória que guarda o histórico recente de cada conversa. A arquitetura segue uma versão simplificada do padrão RAG (*Retrieval-Augmented Generation*): em vez de busca vetorial, o conhecimento da empresa fica disponível ao agente como uma ferramenta que ele consulta quando precisa. Nos testes, feitos em ambiente de desenvolvimento, o agente respondeu corretamente a perguntas sobre responsáveis por setor, processos, prazos e contatos, e manteve o contexto em conversas com várias trocas de mensagens. As principais limitações são a base de conhecimento mantida manualmente e a falta de escalada automática, isto é, de um mecanismo que encaminhe a uma pessoa do RH as perguntas que o agente não consegue responder.

Abstract. Employee *onboarding* usually generates a large volume of repetitive questions that end up on the Human Resources (HR) team. This paper reports the development of an Artificial Intelligence (AI) conversational agent integrated with *Slack* to answer these questions. The agent was built and tested in the scenario of a hypothetical company, referred to here as Company E: a technology organization with 85 employees, nine departments and hybrid work, whose data was entirely created for this study. The solution runs on the *n8n* automation platform and uses Google's *Gemini 3.5 Flash* language model, a structured knowledge base describing the company's departments, and a memory that keeps the recent history of each conversation. The architecture follows a simplified version of the RAG (Retrieval-Augmented Generation) pattern: instead of vector search, company knowledge is exposed to the agent as a tool it calls when needed. In tests run in a development environment, the agent correctly answered questions about department heads, processes, deadlines and contacts, and kept context across multi-turn conversations. The main limitations are the manually maintained knowledge base and the lack of automatic escalation, that is, a mechanism that forwards to an HR person the questions the agent cannot answer.

Palavras-chave: onboarding; agente conversacional; inteligência artificial; automação de processos; slack; n8n

Keywords: onboarding; conversational agent; artificial intelligence; process automation; slack; n8n

Recebido/Received: 13 07 2026 • **Aceito/Accepted:** 07 10 2026 • **Publicado/Published:** 15 10 2026

Publicado sob a Licença Pública Internacional Creative Commons Atribuição 4.0 (CC BY 4.0)

1 Introdução

Quem entra numa empresa nova passa as primeiras semanas perguntando. Com quem falo para pedir férias? Quando sai o contracheque? Quem libera o acesso a tal sistema? Essas dúvidas são normais e fazem parte do *onboarding*, o processo pelo qual a organização ajuda o novo colaborador a se ajustar ao trabalho, às pessoas e às regras da casa [Bauer, 2010]. A literatura sobre socialização organizacional mostra que esse ajuste tem efeito concreto: colaboradores que entendem com clareza o seu papel e sabem a quem recorrer tendem a ter mais desempenho e satisfação, e menos intenção de sair [Bauer *et al*., 2007].

O problema é que boa parte dessas perguntas se repete a cada nova contratação, e quase todas acabam no RH. A equipe responde a mesma coisa várias vezes, perde tempo que poderia ir para atividades mais estratégicas, e o novo colaborador fica esperando uma resposta que já estava escrita em algum lugar [Wandhe, 2026].

Os *chatbots* não são uma ideia nova para esse tipo de demanda [Adamopoulou and Moussiades, 2020], mas durante muito tempo dependeram de regras e respostas fixas, que quebravam quando a pergunta vinha formulada de um jeito diferente. Os Modelos de Linguagem de Grande Escala (*Large Language Models*, LLMs) mudaram esse quadro, porque conseguem entender perguntas em linguagem natural e produzir respostas contextualizadas [Brown *et al*., 2020; Zhao

Tabela 1. Resumo da base de conhecimento fictícia da Empresa E.

Departamento	Exemplos de informação cadastrada
RH	Contracheque até o 5º dia útil; férias pedidas com 30 dias de antecedência; alterações cadastrais feitas pelo RH
Financeiro	Reembolso pago em até 5 dias úteis após aprovação; despesas acima de R$ 1.000 exigem dupla aprovação
Compras	Fluxo: formulário interno, aprovação da liderança, aprovação do Financeiro, envio ao fornecedor
TI	Pedidos via *Slack* ou sistema interno; resposta em até 4 horas úteis
Demais setores	Comercial, Marketing, Operações, Jurídico e Suporte: responsável, e-mail, atribuições e dúvidas comuns

et al., 2023]. No RH, as revisões mais recentes apontam o atendimento a dúvidas rotineiras como uma das aplicações mais promissoras desses modelos [Sakib *et al*., 2024], e uma pesquisa da IBM indica que 38% dos líderes de RH já estavam implementando IA generativa nas suas operações [IBM, 2025].

Usar um LLM sozinho, porém, não resolve. O modelo não conhece os processos internos de uma empresa específica e, quando não sabe a resposta, pode inventar uma que soa plausível, fenômeno chamado de alucinação [Ji *et al*., 2023]. A técnica mais usada para contornar isso é a geração aumentada por recuperação (*Retrieval-Augmented Generation*, RAG), em que o sistema busca informações numa base de conhecimento da própria organização e as entrega ao modelo junto com a pergunta [Lewis *et al*., 2020; Gao *et al*., 2024]. Assim, o modelo responde a partir de dados confiáveis em vez de depender só do que aprendeu no treinamento. Neste trabalho foi adotada uma versão simplificada desse padrão, explicada em detalhe nas Seções 3.2 e 4.

Este artigo relata a construção de um agente conversacional de *onboarding* integrado ao *Slack*, ferramenta de comunicação corporativa usada por muitas empresas no dia a dia [Slack Technologies, 2025]. O agente foi montado na plataforma de automação *n8n* [n8n GmbH, 2025], usa um modelo da família *Gemini*, do Google [Gemini Team, Google, 2023], e foi testado num cenário simulado, descrito a seguir.

1.1 O cenário: a Empresa E

Para desenvolver e testar a solução sem expor dados de uma organização real, foi criada uma empresa hipotética, chamada aqui de **Empresa E**. O nome foi escolhido de propósito por ser genérico, para evitar qualquer confusão com empresas existentes. Todos os dados descritos nesta seção foram inventados para o estudo.

A Empresa E atua com tecnologia, automação e soluções digitais, foi fundada em 2018, tem 85 colaboradores e trabalha em modelo híbrido, com parte da equipe presencial e parte remota. Sua estrutura tem nove departamentos com responsáveis definidos: Recursos Humanos, Financeiro, Compras, Tecnologia da Informação (TI), Comercial, Marketing, Operações, Jurídico e Atendimento ao Cliente, além da Diretoria. A comunicação interna acontece no *Slack*.

Para cada departamento foram cadastrados, com dados fictícios, o nome do responsável, um e-mail de contato, as atribuições do setor, informações importantes (prazos e regras internas) e exemplos de dúvidas comuns. Os e-mails usam o domínio `empresa-e.example`, que é reservado para exemplos e não pertence a ninguém. A Tabela 1 resume o conteúdo dessa base.

A situação simulada é a de um colaborador recém-contratado que, nos primeiros dias, recorre ao *Slack* para tirar dúvidas que normalmente iriam para o RH. Em vez de procurar alguém da equipe, ele escreve para o agente, que responde no mesmo canal.

1.2 Objetivos

O objetivo geral deste trabalho é relatar o desenvolvimento de um agente conversacional baseado em IA, integrado ao *Slack*, capaz de responder de forma autônoma às dúvidas rotineiras de novos colaboradores durante o *onboarding*.

Os objetivos específicos são:

- estruturar uma base de conhecimento com as informações de cada departamento da empresa;
- construir, em uma plataforma *low-code*, um fluxo que receba as mensagens do *Slack*, consulte essa base e devolva a resposta ao colaborador;
- manter o contexto de cada conversa, para que o colaborador possa fazer perguntas de continuação;
- testar o agente com perguntas típicas de *onboarding* e discutir o que funcionou, as limitações e os próximos passos.

As contribuições do trabalho são a descrição detalhada de uma arquitetura *low-code* reproduzível para esse tipo de agente, o relato das decisões de implementação e uma discussão honesta sobre o alcance e as limitações da abordagem.

1.3 Estrutura do artigo

A Seção 2 discute trabalhos relacionados e posiciona esta proposta em relação a eles. A Seção 3 apresenta os conceitos usados na solução. A Seção 4 descreve como o agente foi construído e testado. A Seção 5 traz os resultados e as limitações, e a Seção 6 fecha o artigo com as conclusões e os trabalhos futuros.

2 Trabalhos Relacionados

Wandhe [2026] analisa o uso de *chatbots* com IA no *onboarding* e argumenta que eles reduzem a carga administrativa do RH e tornam a integração mais acessível e personalizada para o novo colaborador. O trabalho é conceitual: discute benefícios e cuidados, mas não descreve uma implementação.

Sakib *et al*. [2024] fazem uma revisão crítica sobre *chatbots* baseados em LLMs na gestão de pessoas. Os autores apontam ganhos de eficiência, principalmente no atendimento a perguntas frequentes, e chamam atenção para questões de privacidade, viés e confiabilidade das respostas. Também é um trabalho de revisão, sem um sistema construído.

No lado técnico, Gao *et al*. [2024] reúnem as principais arquiteturas de RAG, desde a versão mais simples, que recupera trechos de documentos e os passa ao modelo, até variações com etapas de reescrita da pergunta e reordenação dos resultados. Esse *survey* é a base conceitual para a escolha feita aqui, de uma recuperação estruturada no lugar da busca vetorial.

Na indústria, o caso mais conhecido é o AskHR, da IBM [IBM, 2024]. Segundo a empresa, o sistema processou mais de 11,5 milhões de interações em 2024, resolveu 94% das perguntas sem intervenção humana e reduziu em 75% os chamados abertos para o RH desde 2016. É uma solução de

grande porte, construída com uma equipe e uma infraestrutura que não estão ao alcance de empresas menores.

Sobre a plataforma usada, Amir e Atif [2026] avaliaram o *n8n* num estudo de caso de pequena empresa. Os autores implementaram um fluxo de processamento de *leads*, que armazena os dados recebidos, envia um e-mail de confirmação e dispara notificações em tempo real, e compararam 20 execuções manuais com 25 execuções automatizadas em condições controladas. O tempo médio caiu de 185,35 para 1,23 segundos, e a taxa de erro foi de 5% no processo manual para zero no automatizado. É importante deixar claro que esse estudo não envolve LLMs nem agentes conversacionais: ele serve aqui como evidência de que o *n8n* é uma plataforma viável para automatizar processos em organizações pequenas, e não como validação do agente proposto.

A Tabela 2 compara esses trabalhos com a solução deste artigo.

Olhando o conjunto, os trabalhos de RH discutem o potencial dos *chatbots*, mas não mostram como construir um. O caso da IBM mostra que funciona, mas numa escala distante da realidade da maioria das empresas. E o estudo com *n8n* mostra que a plataforma serve para automação, mas sem IA. Este trabalho fica no meio desse caminho: descreve, passo a passo, um agente de *onboarding* com LLM que pode ser montado por uma pessoa só, em uma ferramenta *low-code*, sem infraestrutura própria. O que ele não oferece é uma avaliação quantitativa em produção, ponto discutido na Seção 5.

3 Referencial Teórico

Esta seção apresenta os quatro conceitos em que a solução se apoia. Primeiro, os LLMs, que são o motor do agente. Depois, o padrão RAG, que define como o agente acessa o conhecimento da empresa. Em seguida, as plataformas *low-code*, onde o fluxo foi construído. Por fim, a memória conversacional, que permite ao agente lembrar do que já foi dito.

3.1 Modelos de Linguagem de Grande Escala (LLMs)

Os LLMs são redes neurais treinadas com enormes volumes de texto para prever a próxima palavra de uma sequência. Quase todos se baseiam na arquitetura *Transformer*, proposta por Vaswani *et al*. [2017], que usa um mecanismo de atenção para relacionar cada palavra com todas as outras do texto, o que permite capturar contexto de longo alcance.

Quando esses modelos passaram a ter bilhões de parâmetros, surgiu uma característica importante para este trabalho: eles conseguem executar tarefas novas apenas a partir de instruções e exemplos escritos no próprio pedido, sem precisar de um novo treinamento [Brown *et al*., 2020]. Na prática, isso significa que basta descrever ao modelo o seu papel, por exemplo "você é um assistente de *onboarding*", e fornecer as informações necessárias para que ele passe a atuar nessa função. O texto que define esse papel é chamado de *prompt* de sistema. Zhao *et al*. [2023] apresentam um panorama completo da evolução desses modelos.

O *Gemini* é uma família de modelos multimodais desenvolvida pelo Google, capaz de lidar com texto, imagem, áudio e código [Gemini Team, Google, 2023]. Neste trabalho foi usada a versão *Gemini 3.5 Flash*, uma variante mais leve e rápida da família, escolhida principalmente por ter uma API com camada gratuita durante o desenvolvimento e por responder bem em português.

Os LLMs têm limitações conhecidas. A principal, para um assistente corporativo, é a alucinação: o modelo pode gerar uma resposta fluente e convincente, mas incorreta [Ji *et al*., 2023]. Ele também não sabe nada sobre a empresa além do que for informado. As duas limitações levam diretamente ao próximo conceito.

3.2 Arquitetura RAG (*Retrieval-Augmented Generation*)

O RAG foi proposto por Lewis *et al*. [2020] como forma de combinar o conhecimento que o modelo aprendeu no treinamento com informações buscadas em uma fonte externa no momento da pergunta. O funcionamento básico tem três passos: o sistema recebe a pergunta, recupera da base de conhecimento os trechos mais relevantes e entrega esses trechos ao modelo junto com a pergunta, para que a resposta seja gerada a partir deles [Gao *et al*., 2024].

Na versão mais comum, os documentos são divididos em pedaços e convertidos em vetores numéricos (*embeddings*), e a recuperação é feita por similaridade entre o vetor da pergunta e os vetores dos pedaços. Gao *et al*. [2024] chamam essa abordagem de RAG ingênuo (*naive RAG*) e descrevem variantes mais elaboradas. As vantagens em relação a usar o modelo sozinho são respostas mais fiéis aos dados da organização, menos alucinação e a possibilidade de atualizar o conhecimento sem treinar o modelo de novo.

Neste trabalho foi usada uma variante mais simples. A base da Empresa E é pequena e já está organizada por departamento, então não há necessidade de busca vetorial. Ela foi escrita como um objeto estruturado e oferecida ao agente como uma ferramenta: quando precisa de uma informação, o agente chama a ferramenta e recebe a base. A ideia de que o modelo decida sozinho quando usar ferramentas externas foi explorada por Schick *et al*. [2023] e, combinada com etapas de raciocínio, por Yao *et al*. [2023], no padrão conhecido como ReAct. O componente de agente do *n8n* segue essa linha. A troca é clara: o desenho fica mais simples e previsível, mas não escala para bases grandes ou documentos sem estrutura, como discutido na Seção 5.

3.3 Automação de Processos com Plataformas *Low-Code*

Plataformas *low-code* são ambientes em que se constrói *software* principalmente por meio de interfaces visuais e configuração, com pouca ou nenhuma programação [Bock and Frank, 2021]. A proposta é permitir que pessoas sem formação aprofundada em desenvolvimento criem aplicações e automações, e que equipes técnicas entreguem mais rápido. Sahay *et al*. [2020] compararam várias dessas plataformas e mostram que elas variam bastante em recursos, mas compartilham a ideia de montar a lógica com componentes prontos ligados entre si. Di Ruscio *et al*. [2022] relacionam essa abordagem com a engenharia dirigida por modelos, já conhecida na área de *software*.

O *n8n* é uma plataforma desse tipo, voltada para auto-

Tabela 2. Comparação entre os trabalhos relacionados e a solução proposta.

Trabalho	Tipo	Usa LLM	Foco
Wandhe [2026]	Conceitual	Sim (discussão)	*Chatbots* no *onboarding*
Sakib *et al*. [2024]	Revisão	Sim (discussão)	LLMs na gestão de pessoas
Gao *et al*. [2024]	*Survey*	Sim	Arquiteturas RAG
IBM AskHR [IBM, 2024]	Caso industrial	Sim	Atendimento de RH em larga escala
Amir e Atif [2026]	Estudo de caso	Não	Automação de fluxos com *n8n*
Este trabalho	Relato de experiência	Sim	Agente de *onboarding low-code* no *Slack*, montado por uma pessoa e sem infraestrutura própria

mação de fluxos de trabalho [n8n GmbH, 2025]. Cada fluxo é formado por nós, e cada nó executa uma ação: receber um evento, aplicar uma condição, chamar uma API, rodar um trecho de código. A plataforma tem conectores prontos para centenas de ferramentas, incluindo o *Slack*, e oferece nós de IA construídos sobre a biblioteca *LangChain* [Chase, 2022], que facilitam montar agentes com modelo de linguagem, memória e ferramentas. Foi essa combinação que tornou possível construir o agente deste trabalho sem escrever uma aplicação do zero.

3.4 Memória Conversacional em Agentes de IA

Um LLM, por si só, não guarda nada entre uma chamada e outra. Cada mensagem é processada isoladamente, então, sem algum mecanismo de memória, o agente não saberia a que se refere uma pergunta como "e qual o e-mail dele?". A memória é considerada um dos componentes centrais de agentes baseados em LLMs, ao lado do planejamento e do uso de ferramentas [Wang *et al*., 2024; Zhang *et al*., 2024].

A forma mais simples de memória é guardar as últimas mensagens da conversa e reenviá-las ao modelo a cada nova pergunta. Essa estratégia é chamada de memória de janela deslizante (*window buffer memory*), porque só os últimos N turnos ficam guardados e os mais antigos vão sendo descartados. Ela é barata e fácil de implementar, mas tem um limite: o que sai da janela é esquecido. Existem abordagens mais sofisticadas, como a de Packer *et al*. [2023], que gerencia diferentes níveis de memória de forma parecida com um sistema operacional. Há ainda um detalhe que justifica não usar janelas muito grandes: Liu *et al*. [2024] mostraram que os modelos tendem a aproveitar pior as informações que ficam no meio de contextos longos. Para conversas curtas de *onboarding*, a janela deslizante foi considerada suficiente.

4 Metodologia

Este trabalho é um relato de experiência de pesquisa aplicada. O problema que motivou a solução, a sobrecarga do RH com dúvidas repetitivas no *onboarding*, é real e bem documentado na literatura [Wandhe, 2026; Sakib *et al*., 2024]. Para estudá-lo sem depender dos dados de uma organização específica, o agente foi construído e testado numa simulação: a Empresa E, apresentada na Seção 1.1, com dados inteiramente fictícios. Esse tipo de abordagem, em que se constrói um artefato para um problema real e se avalia o artefato num ambiente controlado antes de levá-lo a uma organização, é comum na pesquisa em sistemas de informação [Hevner *et al*., 2004].

O trabalho foi feito em quatro etapas: (1) montagem da base de conhecimento da Empresa E; (2) construção do fluxo no *n8n*; (3) escrita do *prompt* de sistema; e (4) testes no *Slack*. As subseções a seguir descrevem cada uma.

4.1 Arquitetura e componentes do agente

O agente é um fluxo do *n8n* com sete nós, mostrado na Figura 1. A linha principal tem quatro nós, que levam a mensagem do *Slack* até o agente e devolvem a resposta. Os outros três se ligam ao nó central de agente e fornecem a ele o modelo de linguagem, a memória e a base de conhecimento. A Figura 2 mostra o mesmo fluxo como ele aparece na interface do *n8n*, já com os nós configurados.

O caminho de uma mensagem, do envio pelo colaborador até a resposta, passa pelos seguintes nós.

***Slack Trigger*.** É o ponto de entrada. Fica escutando os eventos do *workspace* do *Slack* e, a cada mensagem nova, captura o texto, o identificador do canal e a referência da *thread*, se houver.

Filtro de identidade (*If*). Verifica quem enviou a mensagem. Como o agente também escreve no *Slack*, as respostas dele geram novos eventos que voltariam para o próprio fluxo e criariam um laço infinito de respostas. O filtro compara o identificador do remetente com o do bot e só deixa passar mensagens enviadas por pessoas.

***AI Agent*.** É o centro da solução. Usa o componente de agente do *n8n*, construído sobre o *LangChain* [Chase, 2022]. Recebe o texto da mensagem, segue as instruções do *prompt* de sistema (Seção 4.2) e decide, a cada pergunta, se precisa consultar a base de conhecimento. Trabalha com três recursos conectados a ele:

- **modelo de linguagem**: o *Gemini 3.5 Flash*, que interpreta a pergunta e escreve a resposta;
- ***Simple Memory***: a memória de janela deslizante, configurada para guardar os últimos 10 turnos. A chave da memória é o identificador do canal do *Slack*, então cada canal tem o seu próprio histórico e uma conversa não se mistura com outra;
- ***Code Tool***: a ferramenta que devolve a base de conhecimento da Empresa E, escrita como um objeto JavaScript (Seção 4.2).

Enviar mensagem. Publica a resposta gerada pelo agente no mesmo canal de onde veio a pergunta e, quando a pergunta foi feita numa *thread*, responde dentro dela, para manter a conversa organizada.

Um exemplo ajuda a visualizar o funcionamento. Um colaborador novo escreve no canal: "Como faço para pedir férias?". O *Slack Trigger* captura a mensagem, o filtro confirma que ela veio de uma pessoa e o agente a recebe. O modelo percebe que precisa de informação da empresa, chama a *Code Tool*, encontra no departamento de RH a regra das férias e

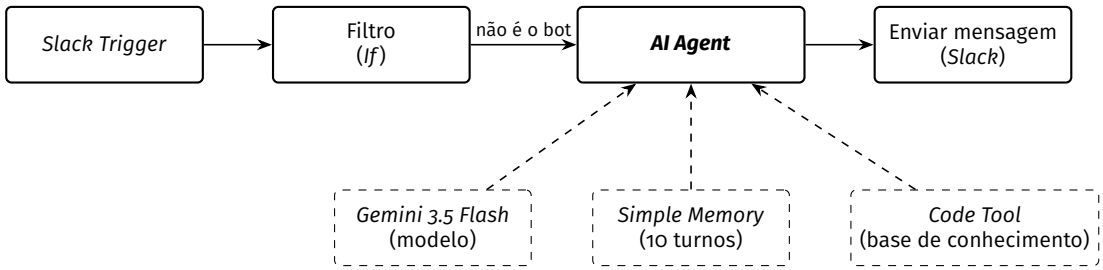


Figura 1. Fluxo do agente no *n8n*. As setas contínuas mostram o caminho da mensagem; as tracejadas, os recursos conectados ao agente.

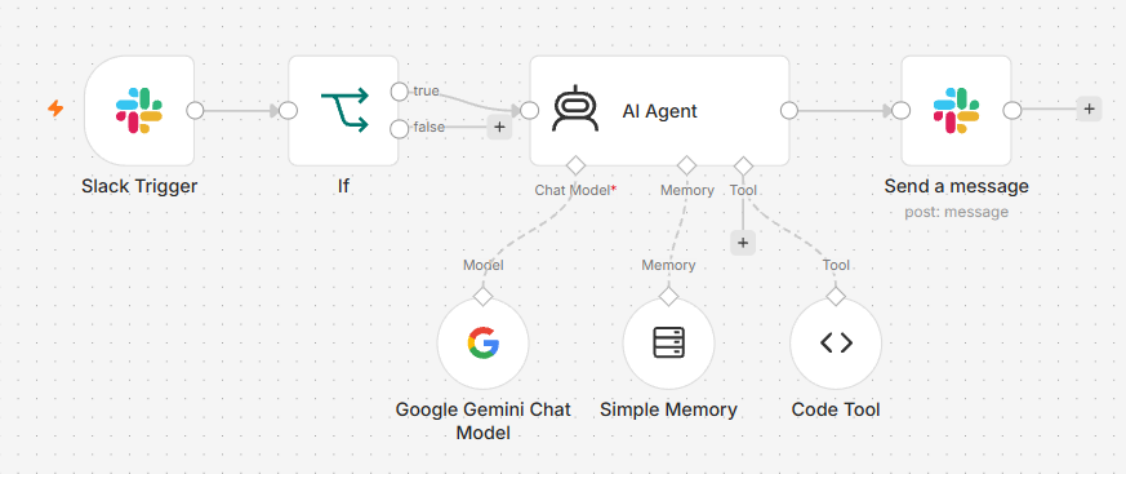


Figura 2. O fluxo implementado, como aparece no editor do *n8n*. Os nós de modelo, memória e ferramenta se conectam ao *AI Agent* pelas entradas *Chat Model*, *Memory* e *Tool*.

Código 1: *Prompt* de sistema do agente.

```
Voc  um assistente interno responsvel por apoiar o onboarding de
     novos colaboradores. Seu papel  esclarecer dvidas sobre
     setores, processos, contatos e rotinas da empresa. Sempre
     que for perguntado a pessoa responsvel por algum setor,
     informe a partir das informaes abaixo. Para responder as
     dvidas utilize as informaes encontradas no n de Code Tool.
```

escreve a resposta, que o último nó publica no canal. A pergunta e a resposta ficam guardadas na memória daquele canal. Se, em seguida, o colaborador perguntar "e com quem eu falo sobre isso?", o agente usa o histórico para entender que "isso" se refere às férias e informa o responsável pelo RH.

4.2 *Prompt* de sistema e base de conhecimento

O comportamento do agente é definido por dois elementos: o *prompt* de sistema, que diz ao modelo qual é o seu papel, e a base de conhecimento, que diz o que ele sabe sobre a empresa.

O *prompt* de sistema é o texto que o modelo recebe antes de qualquer mensagem do usuário. Ele funciona como uma descrição de função: diz quem o agente é, o que deve fazer e onde buscar as informações. O *prompt* usado está no Código 1. Ele foi mantido curto de propósito e tem três partes: o papel (assistente interno de *onboarding*), o escopo (setores, processos, contatos e rotinas) e uma instrução explícita para buscar as respostas na *Code Tool*. Essa última instrução é a mais importante, porque é ela que leva o modelo a responder com base nos dados da empresa e não no que ele "acha" que sabe.

A base de conhecimento foi escrita como um objeto JavaScript dentro da *Code Tool*. A escolha por um objeto estruturado, e não por um texto corrido, tem dois motivos. O primeiro é que cada informação fica num lugar previsível (departamento, responsável, e-mail, atribuições), o que facilita tanto a consulta pelo modelo quanto a manutenção por quem atualiza a base. O segundo é que, com uma base desse tamanho, cabe enviar o conteúdo inteiro ao modelo de uma vez, sem precisar dividir em pedaços nem gerar *embeddings*.

O objeto tem quatro blocos: dados gerais da empresa; missão, visão e valores; a lista de departamentos; e o detalhamento de cada departamento. O Código 2 mostra um trecho com os dados gerais e o departamento de RH. Os outros oito departamentos seguem a mesma estrutura, com campos extras quando faz sentido, como o fluxo de aprovação em Compras e as regras de reembolso no Financeiro (Tabela 1).

O campo `duvidasComuns` merece um comentário. Ele não é usado para comparar perguntas de forma literal. Serve como exemplo para o modelo de que tipo de pergunta cada departamento resolve, o que ajuda o agente a escolher o setor certo quando a pergunta vem escrita de outro jeito.

4.3 Procedimento de teste

Os testes foram feitos em ambiente de desenvolvimento, num *workspace* do *Slack* criado para o estudo e conectado ao fluxo do *n8n*. O autor fez o papel de um colaborador recém-contratado da Empresa E e enviou ao agente perguntas de quatro tipos:

1. **responsáveis e contatos**: quem cuida de cada setor e qual o e-mail;
2. **processos**: como pedir férias, como solicitar uma compra, como pedir reembolso;

Tabela 3. Exemplos representativos das perguntas de teste e da informação esperada na resposta, segundo a base de conhecimento.

Tipo	Pergunta enviada	Informação esperada na resposta
Contato	Quem é o responsável pelo RH?	Vitor Frota, rh@empresa-e.example
Contato	Com quem eu falo sobre LGPD?	Jurídico, Fernanda Almeida
Processo	Preciso comprar um monitor, como faço?	Formulário interno, aprovação da liderança, aprovação do Financeiro, envio ao fornecedor
Processo	Como peço reembolso de uma despesa?	Formulário com comprovantes enviado ao Financeiro
Prazo	Quando sai meu contracheque?	Até o 5º dia útil
Prazo	Abri um chamado na TI, quanto tempo demora?	Até 4 horas úteis
Continuação	(logo após a anterior) E qual o e-mail deles?	ti@empresa-e.example

Código 2: Trecho da base de conhecimento (*Code Tool*).

```
const onboarding = {
  empresa: {
    nome: "Empresa E",
    segmento: "Tecnologia, Automao e Solues Digitais",
    numeroColaboradores: 85,
    modeloTrabalho: "Hbrido (presencial e remoto)"
  },
  departamentos: {
    rh: {
      nome: "Recursos Humanos",
      responsavel: "Vitor Frota",
      email: "rh@empresa-e.example",
      atribuicoes: ["Admisso e desligamento",
        "Emisso de contracheque", "Gesto de frias",
        "Administrao de benefcios"],
      informacoesImportantes: [
        "Contracheque disponibilizado at o 5 dia til",
        "Frias solicitadas com 30 dias de antecedncia"],
      duvidasComuns: [
        "Com quem falo para solicitar meu contracheque?",
        "Como fao para pedir frias?"]
    }
    // demais departamentos seguem a mesma estrutura
  }
};
return onboarding;
```

3. **prazos e regras**: quando sai o contracheque, prazo de reembolso, tempo de resposta da TI;
4. **continuação**: perguntas que só fazem sentido com o contexto anterior, como "e qual o e-mail dele?", para verificar a memória.

As perguntas foram escritas de formas variadas, e não apenas copiadas do campo `duvidasComuns`, para observar se o agente lidava com formulações diferentes. Cada resposta foi conferida manualmente contra a base de conhecimento. Como se trata de um relato de experiência, a avaliação foi qualitativa: o critério foi verificar se a resposta estava correta e de acordo com a base, se o agente usava o contexto da conversa e se a resposta fazia sentido para quem perguntou. A Tabela 3 mostra exemplos representativos das perguntas de cada tipo.

5 Resultados

Nos testes, o agente respondeu corretamente às perguntas sobre responsáveis por setor, processos de solicitação, prazos e contatos de e-mail, sempre de acordo com o que estava cadastrado na base. Perguntas escritas de forma diferente da prevista no campo `duvidasComuns`, como "preciso comprar um monitor, como faço?", também foram encaminhadas ao setor certo, o que mostra a vantagem de usar um LLM em vez de regras fixas.

A memória funcionou como esperado. Em conversas com várias trocas, o agente entendeu perguntas de continuação, como pedir o e-mail "deles" logo depois de uma pergunta sobre a TI, sem que o colaborador precisasse repetir o assunto. Como a memória é separada por canal, conversas em canais diferentes não interferiram entre si.

É preciso ser claro sobre o alcance desses resultados. Os testes mostram que a solução funciona tecnicamente: o fluxo recebe a pergunta, consulta a base, responde corretamente e mantém o contexto. Eles não medem o impacto num ambiente real, como redução de chamados ao RH ou satisfação dos colaboradores, porque não houve implantação numa empresa com usuários reais. Os benefícios a seguir são, portanto, os que o desenho da solução permite esperar, apoiados no que a literatura relata para sistemas parecidos:

- **disponibilidade contínua**: o agente responde a qualquer hora, sem depender da agenda da equipe de RH;
- **respostas padronizadas**: como a resposta vem sempre da mesma base, a informação não muda de acordo com quem responde;
- **menos interrupções ao RH**: dúvidas rotineiras podem ser resolvidas sem envolver a equipe, como relatam Sakib *et al*. [2024] e, em escala bem maior, o caso AskHR [IBM, 2024];
- **manutenção simples**: para mudar uma informação, basta editar o objeto na *Code Tool*, sem alterar o resto do fluxo.

Quanto ao custo, a solução usa a camada gratuita da API do *Gemini* e o *Slack*, que muitas empresas já têm. O *n8n* pode ser instalado em servidor próprio. Para uma empresa do porte da Empresa E, isso torna o custo de operação baixo, embora ele cresça com o volume de mensagens se for necessário passar para um plano pago do modelo.

5.1 Limitações

O trabalho tem limitações que precisam ser consideradas.

(i) **Base mantida à mão.** Toda a informação está num objeto escrito manualmente. Se ninguém atualizar quando um responsável muda de cargo ou uma regra muda, o agente vai passar informação errada com toda a segurança.

(ii) **Sem escalada automática.** Quando o agente não encontra a resposta na base, não há um caminho para levar a dúvida a uma pessoa do RH. O ideal seria que ele reconhecesse que não sabe e abrisse um chamado ou mencionasse alguém da equipe na conversa.

(iii) **Memória curta.** A janela de 10 turnos é suficiente para perguntas rápidas, mas conversas mais longas perdem o começo.

(iv) **Não escala para bases grandes.** Enviar a base inteira ao modelo funciona com nove departamentos, mas não com

centenas de documentos, manuais ou políticas em texto corrido. Nesse caso seria necessário um RAG completo, com busca vetorial [Gao *et al*., 2024].

(v) **Avaliação limitada.** Os testes foram feitos por uma única pessoa, num ambiente simulado e sem métricas quantitativas.

6 Conclusões e Trabalhos Futuros

Este trabalho relatou o desenvolvimento de um agente conversacional baseado em IA para apoiar o *onboarding* de novos colaboradores, integrado ao *Slack* e construído no *n8n*. No cenário simulado da Empresa E, o agente respondeu corretamente a dúvidas sobre responsáveis, processos, prazos e contatos e manteve o contexto ao longo das conversas. Os testes indicam que a solução é tecnicamente viável, de baixo custo e simples de manter, e que pode ser montada por uma pessoa só, sem desenvolver uma aplicação do zero.

A principal lição do processo foi que, para uma base de conhecimento pequena e bem organizada, a versão simplificada de RAG resolve o problema sem a complexidade da busca vetorial. A estrutura da base, separada por departamento e com exemplos de dúvidas, contribuiu tanto quanto o modelo para a qualidade das respostas. Por outro lado, a experiência deixou claro que um agente que não sabe dizer "não sei" e não tem para quem passar a pergunta é um risco num ambiente corporativo.

Como trabalhos futuros, pretende-se: (i) implementar a escalada automática, para que perguntas sem resposta na base sejam encaminhadas ao RH; (ii) incorporar busca vetorial (RAG completo), para lidar com bases maiores e documentos não estruturados; e (iii) implantar a solução numa organização real e medir o impacto com métricas de uso, como volume de perguntas respondidas sem intervenção humana e redução de chamados ao RH.

Declarações complementares

Contribuições dos autores

Este é um trabalho de autor único. VCC contribuiu em todos os aspectos do trabalho: Conceituação, Metodologia, Software, Validação, Análise Formal, Redação (rascunho original) e Redação (revisão e edição), conforme a Taxonomia CRediT (https://credit.niso.org/).

Conflitos de interesse

O autor declara que não há conflitos de interesse.

Disponibilidade de artefatos de pequisa

O arquivo JSON do fluxo n8n e a base de conhecimento utilizada neste estudo serão disponibilizados mediante solicitação. Todos os dados utilizados são fictícios e foram criados exclusivamente para fins acadêmicos.

Uso de Inteligência Artificial

A plataforma n8n e o modelo Gemini 3.5 Flash fazem parte do sistema descrito neste artigo. O Claude (Anthropic), usado por meio do Claude Code em outubro de 2026, auxiliou na redação e na revisão do texto, na verificação das referências em bases bibliográficas e na formatação em LaTeX. O autor revisou todo o conteúdo e é responsável por ele.

Outras informações relevantes

Este estudo não envolveu participantes humanos, dados pessoais reais ou informações sensíveis. Todos os perfis de colaboradores e dados organizacionais utilizados, incluindo o nome Empresa E, são inteiramente fictícios e foram criados exclusivamente para fins acadêmicos.

Referências

Adamopoulou, E. and Moussiades, L. (2020). Chatbots: History, technology, and applications. *Machine Learning with Applications*, 2:100006. DOI: 10.1016/j.mlwa.2020.100006.

Amir, A. R. and Atif, S. M. (2026). Evaluating workflow automation efficiency using n8n: A small-scale business case study. arXiv preprint arXiv:2602.01311.

Bauer, T. N. (2010). *Onboarding New Employees: Maximizing Success*. SHRM Foundation, Alexandria, VA.

Bauer, T. N., Bodner, T., Erdogan, B., Truxillo, D. M., and Tucker, J. S. (2007). Newcomer adjustment during organizational socialization: A meta-analytic review of antecedents, outcomes, and methods. *Journal of Applied Psychology*, 92(3):707–721. DOI: 10.1037/0021-9010.92.3.707.

Bock, A. C. and Frank, U. (2021). Low-code platform. *Business & Information Systems Engineering*, 63(6):733–740. DOI: 10.1007/s12599-021-00726-8.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., *et al*. (2020). Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Chase, H. (2022). LangChain. `https://github.com/langchain-ai/langchain`.

Di Ruscio, D., Kolovos, D., de Lara, J., Pierantonio, A., Tisi, M., and Wimmer, M. (2022). Low-code development and model-driven engineering: Two sides of the same coin? *Software and Systems Modeling*, 21(2):437–446. DOI: 10.1007/s10270-021-00970-2.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., and Wang, H. (2024). Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997.

Gemini Team, Google (2023). Gemini: A family of highly capable multimodal models. arXiv preprint arXiv:2312.11805.

Hevner, A. R., March, S. T., Park, J., and Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1):75–105. DOI: 10.2307/25148625.

IBM (2024). Transforming HR support with agentic AI: IBM AskHR. IBM Case Studies.

IBM (2025). What are chatbots for HR? IBM Think Topics.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., and Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38. DOI: 10.1145/3571730.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., and Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474.

Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., and Liang, P. (2024). Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173. DOI: 10.1162/tacl_a_00638.

n8n GmbH (2025). n8n documentation. `https://docs.n8n.io`.

Packer, C., Wooders, S., Lin, K., Fang, V., Patil, S. G., Stoica, I., and Gonzalez, J. E. (2023). MemGPT: Towards LLMs as operating systems. arXiv preprint arXiv:2310.08560.

Sahay, A., Indamutsa, A., Di Ruscio, D., and Pierantonio, A. (2020). Supporting the understanding and comparison of low-code development platforms. In *46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 171–178. DOI: 10.1109/SEAA51224.2020.00036.

Sakib, M. N. *et al*. (2024). The ChatGPT and the future of HR: A critical review on the benefits and challenges of AI chatbots in human resource management. *Multidisciplinary Reviews*, 7(8). DOI: 10.31893/multire.2024136.

Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., and Scialom, T. (2023). Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, volume 36, pages 68539–68551.

Slack Technologies (2025). Slack developer docs. `https://docs.slack.dev`.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.

Wandhe, P. (2026). AI chatbots as enablers of effective employee onboarding. *SSRN Electronic Journal*. DOI: 10.2139/ssrn.6099951.

Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., *et al*. (2024). A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345. DOI: 10.1007/s11704-024-40231-1.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. (2023). ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Zhang, Z., Bo, X., Ma, C., Li, R., Chen, X., *et al*. (2024). A survey on the memory mechanism of large language model based agents. arXiv preprint arXiv:2404.13501.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., *et al*. (2023). A survey of large language models. arXiv preprint arXiv:2303.18223.

Artigo Completo

Impacto do Tamanho da Amostra de Treinamento e do Vocabulário Para Tarefas de Classificação de Texto

Impact of Training Sample and Vocabulary Size on Text Classification Tasks

Ítalo Alves Tenório de Almeida [Universidade Federal do Vale do São Francisco | *italo.tenorio@discente.univasf.edu.br*]
Jorge Luís Cavalcanti Ramos [Universidade Federal do Vale do São Francisco | *jorge.cavalcanti@univasf.edu.br*]
Brauliro Gonçalves Leal [Universidade Federal do Vale do São Francisco | *brauliro.leal@univasf.edu.br*]
Rosalvo Ferreira de Oliveira Neto [Universidade Federal do Vale do São Francisco | *rosalvo.oliveira@univasf.edu.br*]

Universidade Federal do Vale do São Francisco, Av. Antônio C. Magalhães, 510 - Santo Antônio, Juazeiro, Bahia, 48902-300, Brasil.

Resumo. O crescimento dos modelos de linguagem baseados em arquiteturas Transformer tem ampliado significativamente as possibilidades de aplicação do Processamento de Linguagem Natural em tarefas de classificação de texto. Entretanto, aspectos fundamentais do treinamento desses modelos, como o tamanho do vocabulário e a quantidade de amostras utilizadas, ainda contam com poucas orientações práticas para sua configuração em tarefas de classificação de texto. Este trabalho analisa o impacto desses dois fatores no desempenho de modelos de classificação de texto utilizando o DistilRoBERTa. Foram realizados experimentos com dois conjuntos de dados públicos disponibilizados pela plataforma Hugging Face: um conjunto de resumos de artigos médicos, binarizado para neoplasmas contra as demais classes, e um conjunto de avaliações de filmes do IMDb. O estudo empregou validação cruzada *k-fold* (k = 5), planejamento fatorial 2^k e a métrica AUC-ROC para avaliar o desempenho dos modelos. Comparou-se o DistilRoBERTa com seu vocabulário original (genérico) e com um vocabulário reduzido e especializado, gerado a partir do corpus de treinamento, além de um modelo de propósito geral baseado em LLM, o GPT-3.5-turbo. Em amostras de 1000 a 5000 exemplos, o vocabulário original apresentou o melhor desempenho global entre as variantes do DistilRoBERTa em ambas as bases, e o vocabulário reduzido teve desempenho inferior, com maior variabilidade entre os *folds* em amostras menores. O vocabulário reduzido, porém, foi o que mais se beneficiou do aumento da amostra e, na base médica, superou o GPT-3.5-turbo para 2500 e 5000 amostras. A influência relativa do vocabulário e da amostra depende do domínio e do conjunto de dados: o vocabulário explicou a maior parte da variação total, em maior grau no IMDb do que na base médica. Os resultados contribuem para uma melhor compreensão dos fatores que influenciam o desempenho de modelos de classificação de texto e fornecem subsídios para o planejamento mais eficiente de experimentos em Processamento de Linguagem Natural.

Abstract. The rapid development of Transformer-based language models has significantly expanded the applicability of Natural Language Processing techniques to text classification tasks. However, fundamental training parameters, such as vocabulary size and training sample size, still have few practical guidelines for their configuration in text classification tasks. This work analyzes the impact of these two factors on the performance of text classification models based on DistilRoBERTa. Experiments were conducted using two publicly available datasets from the Hugging Face platform: a collection of medical research abstracts, binarized as neoplasms versus other classes, and the IMDb movie review dataset. The study employed 5-fold cross-validation, a 2^k factorial design, and the AUC-ROC metric to evaluate model performance. DistilRoBERTa with its original (generic) vocabulary was compared with a reduced, domain-specific vocabulary generated from the training corpus, as well as with the general-purpose large language model GPT-3.5-turbo. With samples of 1,000 to 5,000 examples, the original vocabulary achieved the best overall performance among the DistilRoBERTa variants on both datasets, whereas the reduced vocabulary performed worse, with higher variability across folds for smaller samples. The reduced vocabulary, however, benefited the most from larger samples and, on the medical dataset, outperformed GPT-3.5-turbo with 2,500 and 5,000 samples. The relative influence of vocabulary and sample size depends on the domain and dataset: vocabulary accounted for most of the total variation, to a greater extent on IMDb than on the medical dataset. These findings contribute to a better understanding of the factors that influence text classification performance and provide guidance for the design of more efficient Natural Language Processing experiments.

Palavras-chave: Processamento de Linguagem Natural, Classificação de Texto, Modelos de Linguagem, Tokenização, DistilRoBERTa

Keywords: Natural Language Processing, Text Classification, Language Models, Tokenization, DistilRoBERTa

Recebido/Received: 13 08 2026 • **Aceito/Accepted:** 07 10 2026 • **Publicado/Published:** 15 10 2026

Publicado sob a Licença Pública Internacional Creative Commons Atribuição 4.0 (CC BY 4.0)

1 Introdução

O Processamento de Linguagem Natural (do inglês *Natural Language Processing*, NLP) constitui um ramo da ciência da computação, da inteligência artificial e da linguística, dedicado ao estudo da interação complexa entre computadores e a linguagem humana. O campo abrange a modelagem matemática e computacional dos diversos aspectos da linguagem, visando desenvolver uma ampla gama de sistemas. O objetivo central do NLP é permitir que máquinas compreendam e manipulem textos ou fala em linguagem natural de forma útil, o que envolve a extração de estrutura gramatical e significado a partir dos dados de entrada. A relevância do NLP é atestada pela vasta aplicação de suas técnicas em diversas áreas, incluindo a tradução automática, a sumarização de textos, interfaces de usuário mais interativas, sistemas de recuperação de informação multilingues e, fundamentalmente, como componente chave na construção de sistemas de IA (Inteligência Artificial) [Liddy, 2001].

Nos últimos anos, a evolução do NLP foi acelerada pelo surgimento dos LLMs (do inglês *Large language Models*), que têm capturado grande atenção devido ao seu desempenho robusto em uma vasta gama de tarefas de linguagem natural. Os LLMs são modelos estatísticos pré-treinados em grande escala, baseados em arquiteturas de redes neurais, notavelmente os Transformers. Eles se distinguem por conterem dezenas a centenas de bilhões de parâmetros, treinados em quantidades massivas de dados de texto web. Esta escalabilidade não apenas melhorou a compreensão e a geração de linguagem, mas também resultou em habilidades emergentes que não se manifestavam em modelos menores, como o aprendizado em contexto (*in-context learning*), o seguimento de instruções e a capacidade de raciocínio em múltiplas etapas. Essa capacidade de resolver tarefas de propósito geral posiciona os LLMs como o bloco construtor básico para o desenvolvimento de agentes de IA mais sofisticados. Dentro deste cenário, a classificação de texto consolidou-se como uma tarefa fundamental e perene do NLP, consistindo na categorização automática de documentos em rótulos ou classes predefinidas com base em seu conteúdo, tarefa crucial para a descoberta de conhecimento e para a tomada de decisões em um contexto de crescimento exponencial de dados digitais. Com o advento da era dos LLMs, a performance destes sistemas deixou de depender exclusivamente de grandes volumes de dados rotulados, abrindo espaço para abordagens de aprendizado *zero-shot* e *few-shot*, além da utilização de modelos para a geração de dados sintéticos. No entanto, a escolha da arquitetura permanece crítica; enquanto modelos de grande porte como o Llama3 e o GPT-4 mostram excelência em tarefas complexas, modelos menores e otimizados como o RoBERTa oferecem um equilíbrio superior entre desempenho e tempo de inferência, mantendo-se competitivos especialmente em cenários de classificação binária ou quando a eficiência computacional é uma prioridade [Kostina *et al.*, 2025; Vajjala and Shimangaud, 2025].

No entanto, a definição do tamanho ideal do vocabulário (o número de tokens) é uma questão aberta, dependendo da tarefa, do domínio e da linguagem, e afeta diretamente a eficiência computacional (o comprimento das sequências) e a capacidade de generalização do modelo. Similarmente, o tamanho das amostras de treinamento do modelo, bem como seu impacto no desempenho do mesmo, permanece um problema com solução altamente dependente da tarefa que o modelo irá executar. Embora existam trabalhos sobre tokenização [Mielke *et al.*, 2021] e sobre classificação de texto com modelos de linguagem [Kostina *et al.*, 2025; Vajjala and Shimangaud, 2025], na literatura consultada pelos autores são escassas as orientações práticas que relacionem conjuntamente o tamanho do vocabulário e o tamanho da amostra de treinamento ao desempenho de modelos de menor porte em classificação de texto.

Diante disso, este trabalho se propõe a investigar o impacto do tamanho do vocabulário e da amostra de treinamento na eficácia e nas características de desempenho em tarefas de classificação de texto.

2 Referencial Teórico

2.1 Processamento de Linguagem Natural

O NLP, também conhecido como Linguística Computacional, constitui um subcampo fundamental da Ciência da Computação. Sua essência reside na aplicação de técnicas computacionais para aprender, compreender e produzir conteúdo em linguagem humana. O NLP surgiu na década de 1950, situando-se na intersecção entre a inteligência artificial e a linguística. O objetivo principal do NLP é atingir o "processamento de linguagem semelhante ao humano", sendo uma disciplina enquadrada dentro da IA. [Liddy, 2001; Reshamwala *et al.*, 2013; Hirschberg and Manning, 2015].

Embora o termo ”processamento” seja escolhido deliberadamente, e não ”compreensão” (*Natural Language Understanding* – NLU), a NLU continua sendo a meta final do NLP. Um sistema de compreensão da linguagem humana completo seria capaz de parafrasear um texto de entrada, traduzir o texto para outra língua, responder a perguntas sobre seu conteúdo e tirar inferências a partir dele [Liddy, 2001].

O NLP é a base para uma vasta gama de aplicações, muitas das quais já fazem parte do cotidiano, como os assistentes de diálogo falado [Hirschberg and Manning, 2015].

2.2 Large Language Models

Os LLMs representam a quarta e mais recente geração de Modelos de Linguagem (LMs), evoluindo a partir dos Modelos de Linguagem Pré-treinados (do inglês *Pre-trained Language Models*, PLM) que, por sua vez, sucederam os modelos de linguagem neurais e estatísticos [Zhao *et al.*, 2023]. O desenvolvimento da modelagem de linguagem, que data da aplicação da teoria da informação por Shannon na década de 1950 [Minaee *et al.*, 2024], atingiu um ponto de inflexão com o surgimento dos LLMs. Tecnicamente, os LLMs são modelos de linguagem baseados na arquitetura Transformer que contêm dezenas a centenas de bilhões (ou mais) de parâmetros e são pré-treinados em dados de texto massivos [Zhao *et al.*, 2023; Minaee *et al.*, 2024]. Eles exibem fortes capacidades de compreensão da linguagem natural e são capazes de resolver tarefas complexas através da geração de texto [Zhao *et al.*, 2023].

2.2.1 LLMs Representativos

A pesquisa em LLMs tem sido impulsionada por modelos chave, que são frequentemente agrupados em famílias. A Figura 1 reúne algumas das principais famílias de LLMs atualmente, descritas a seguir, da esquerda para a direita:

- Família GPT (OpenAI): Os Generative Pre-trained Transformers são modelos *decoder-only*. O GPT-3 (175B parâmetros) é notável por formalmente introduzir o ICL (do inglês *In-Context Learning*) [Brown *et al*., 2020; apud Minaee *et al*., 2023]. O ChatGPT, um chatbot de IA poderoso, é desenvolvido com base em LLMs da série GPT [Zhao et al., 2023]. Mais recentemente, o GPT-4 (1.76T parâmetros) elevou o nível de desempenho, incluindo capacidades multimodais [Minaee *et al*., 2024].
- Família LLaMA (Meta AI): Esta família, que inclui LLaMA (até 65B parâmetros) e LLaMA 2 (até 70B parâmetros), atraiu atenção significativa por ser de código aberto (open-source). O LLaMA 2, em particular, explorou o RLHF (do inglês *Reinforcement Learning From Human Feedback*) e possui uma versão otimizada para chat (LLaMA-chat) [Touvron *et al*., 2023; apud Zhao *et al*., 2023].
- Família PaLM (Google): O PaLM (até 540B parâmetros) foi treinado em um *dataset* de 780B tokens, e o PaLM 2 utilizou 3.6T tokens e incorpora o objetivo de pré-treinamento unificado MoD (*Mixture-of-Denoisers*) [Minaee *et al*., 2024].

2.3 Tokenização e Vocabulário

A tokenização é uma etapa fundamental e crítica no pré-processamento de texto, essencial para preparar os dados de entrada para modelos de linguagem profunda, como os LLMs. Essencialmente, a tokenização lida com a questão fundamental de quais são as unidades de texto que queremos modelar. A premissa básica é que o texto é fatiado em pequenas peças que são alimentadas no computador, eventualmente como sequências de números inteiros, e essas peças são chamadas de tokens [Mielke *et al*., 2021]. O vocabulário, por sua vez, consiste no conjunto de tokens reconhecidos pelo modelo, onde cada token e representado concretamente na forma de um vetor de alta dimensão [Nakash *et al*., 2025].

2.3.1 Metodologias de Segmentação de Subpalavras

As estratégias de tokenização moderna se dividem em abordagens de subpalavra (impulsionadas por dados), abordagens linguísticas/morfológicas, e abordagens de decomposição máxima (caractere/byte).

Os algoritmos de segmentação de subpalavra Dirigida por Dados (*Data-Driven*) buscam unidades de subpalavra que sejam frequentes ou que maximizem a verossimilhança do modelo de linguagem. BPE e WordPiece são os métodos de facto empregados pelos LLMs mais importantes, como BERT e GPT [Mielke *et al*., 2021].

2.4 Arquitetura Transformer

A arquitetura Transformer representa um marco fundamental na área de modelagem de sequências e transdução, sendo introduzida em 2017 com a proposta de que "atenção é tudo o que você precisa" . O modelo Transformer foi proposto como uma nova arquitetura de rede neural baseada somente em mecanismos de atenção, dispensando completamente a recorrência e as convoluções [Vaswani *et al*., 2017].

Historicamente, os modelos dominantes para problemas de transdução de sequência, como modelagem de linguagem e tradução automática, eram baseados em redes neurais recorrentes (do inglês *Recurrent Neural Networks*, RNNs), em particular LSTMs (*Long Short-Term Memory*) e redes neurais recorrentes com (GRUs) (*gated recurrent units*). Estes modelos recorrentes tipicamente levavam em conta a computação ao longo das posições dos símbolos na sequência de entrada e saída. Ao alinhar as posições com os passos no tempo de computação, geravam uma sequência de estados ocultos, uma função do estado oculto anterior e da entrada da posição atual . Essa natureza inerentemente sequencial impedia a paralelização dentro dos exemplos de treinamento, um obstáculo crítico para lidar com sequências mais longas [Vaswani *et al*., 2017].

O Transformer superou essa restrição fundamental ao adotar mecanismos de atenção, permitindo um grau significativamente maior de paralelização e alcançando novos resultados de ponta em tarefas de tradução com tempos de treinamento reduzidos . O mecanismo de atenção permite modelar dependências sem considerar sua distância nas sequências de entrada ou saída [Vaswani *et al*., 2017].

2.4.1 Estrutura Geral do Modelo

O Transformer segue a arquitetura geral de *encoder-decoder*, ilustrada na Figura 2. O *encoder* mapeia uma sequência de entrada de representações de símbolos

$$x = (x_1, ..., x_n)$$

para uma sequência de representações contínuas

$$z = (z_1, ..., z_n).$$

Dado z, o *decoder* então gera uma sequência de saída

$$y = (y_1, ..., y_m),$$

um elemento de cada vez, operando de forma autoregressiva e consumindo os símbolos gerados previamente como entrada adicional para gerar o próximo [Vaswani *et al*., 2017].

2.4.2 Atenção

A atenção é a base da arquitetura Transformer. Uma função de atenção pode ser descrita como o mapeamento de uma *query* e um conjunto de pares *key-value* para uma saída [Vaswani *et al*., 2017]. A saída é calculada como uma soma ponderada dos valores, onde o peso atribuído a cada valor é determinado por uma função de compatibilidade da *query* com a *key* correspondente [Vaswani *et al*., 2017].

O tipo específico de atenção utilizado no Transformer é denominado *Scaled Dot-Product Attention* (Atenção por Produto Escalar Escalonado). O processo envolve [Vaswani *et al*., 2017]:

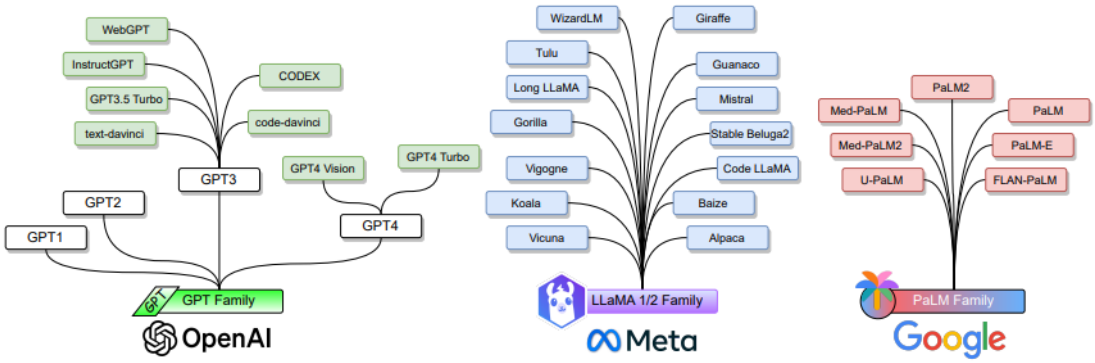


Figura 1. Famílias de LLMs [Minaee *et al.*, 2024].

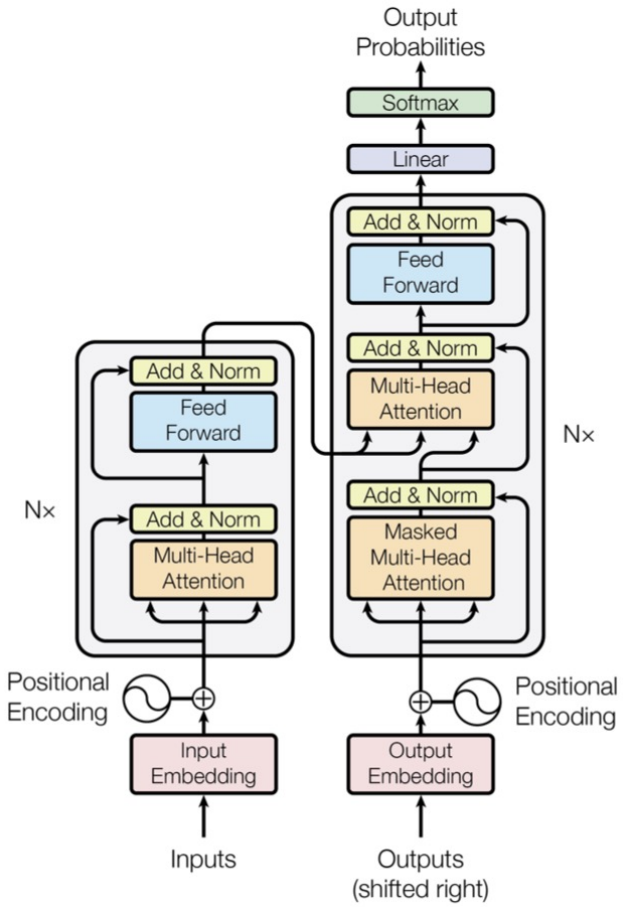


Figura 2. Diagrama da arquitetura Transformer [Vaswani *et al.*, 2017].

- Calcular os produtos escalares da *query* com todas as *keys*.
- Dividir cada resultado por d, onde d é a dimensão das *keys*.
- Aplicar a função softmax para obter os pesos sobre os *values* [Vaswani *et al.*, 2017].

Matematicamente, quando processado simultaneamente em um conjunto de *queries* Q, *keys* K e *values* V (empacotados em matrizes), a atenção é calculada como:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d}})V$$

Em vez de realizar uma única função de atenção com keys, values e queries de dimensão d_{model}, o Transformer utiliza MHA (do inglês *Multi-Head Attention*). O MHA projeta linearmente as queries, keys e values h vezes, usando projeções diferentes e aprendidas, para dimensões d, respectivamente. A função de atenção é então aplicada em paralelo a cada uma dessas versões projetadas, resultando em h saídas dimensionais. Estas são então concatenadas e projetadas novamente linearmente para produzir o valor final [Vaswani *et al.*, 2017]. O MHA permite que o modelo atenda conjuntamente a informações de diferentes subespaços de representação e em diferentes posições. A utilização de múltiplos *heads* (cabeças de atenção) mitiga o problema de que uma única cabeça de atenção, ao calcular a média, inibiria essa capacidade . Na arquitetura base, $h = 8$ cabeças paralelas são empregadas, com $d = 64$. Apesar de usar múltiplas cabeças, o custo computacional total permanece similar ao de uma única cabeça com dimensionalidade total, devido à redução das dimensões em cada head [Vaswani *et al.*, 2017].

2.5 Classificação de Texto

A classificação de texto é uma tarefa fundamental e perene no campo do NLP, consistindo na categorização automática de documentos ou fragmentos de texto em rótulos ou classes predefinidas com base em seu conteúdo. Este processo é crucial para a gestão da informação e descoberta de conhecimento, permitindo a organização eficiente de grandes volumes de dados digitais. Tradicionalmente, o fluxo de trabalho de classificação segue etapas padronizadas: coleta de dados, pré-processamento,

extração de características, redução de dimensionalidade e, por fim, a aplicação de técnicas de classificação seguidas de avaliação de desempenho [Kadhim, 2019].

2.5.1 Análise de Sentimento

A análise de sentimento, frequentemente referida como mineração de opinião, é uma subárea do NLP e da Classificação de Texto, dedicada à identificação e extração de informações subjetivas a partir de textos. O objetivo central desta tarefa é processar e analisar computacionalmente opiniões, pensamentos e impressões humanas sobre diversos tópicos, produtos ou serviços, permitindo que empresas e indivíduos tomem decisões fundamentadas com base na polaridade expressa pelos usuários [Wankhade *et al*., 2022]. A análise de sentimento é geralmente classificada em diferentes níveis de granularidade: nível de documento (atribui uma polaridade única ao texto completo), nível de frase (analisa cada sentença individualmente), nível de expressão (foca em unidades linguísticas básicas) e o nível de aspecto. Este último é considerado um dos ramos mais promissores, pois permite identificar sentimentos específicos sobre atributos distintos de um produto, como o "sabor" de uma comida ou a "lentidão" de um serviço.

3 Metodologia

3.1 Bases de Dados

Foram utilizadas duas bases de dados no experimento, ambas disponíveis publicamente no site HuggingFace (`https://huggingface.co/`).

3.1.1 Resumos de Artigos Médicos

Essa base, com tamanho de 14443 linhas, consiste em resumos de diversos artigos da área medica. Foi criada originalmente durante a escrita do trabalho de Schopf *et al*. [2022]. A base possui duas colunas: uma contendo o texto do resumo do artigo, e a outra contendo o rótulo da classe associado à área de pesquisa abrangida. São 5 rótulos: Neoplasmas, doenças cardiovasculares, doenças do sistema nervoso, doenças do sistema digestivo e condições patológicas gerais [Schopf *et al*., 2022].
Para os propósitos do experimento, a coluna de rótulos foi alterada de forma a representar um problema de classificação binária, ao invés de um problema multi-classe. Devido à quantidade de exemplos, a classe Neoplasmas foi escolhida como referência. Assim, os abstratos passaram a ser classificados em "neoplasmas" (rótulo 1), caso essa seja a área de pesquisa abrangida, e não-neoplasmas (rótulo 0) caso não seja. Essa binarização simplifica o cenário clínico original, que é multiclasse, e foi adotada para viabilizar o uso do AUC-ROC. Os resultados, portanto, não se estendem diretamente ao problema multiclasse.
As amostras de cada tamanho foram construídas com proporção equilibrada entre as classes, selecionando um número igual de exemplos positivos e negativos (proporção 50/50).

3.1.2 IMDb

Essa base, construída para classificação de texto (especificamente análise de sentimento), possui 100000 linhas e consiste em resenhas críticas de diversos filmes, extraídas do site IMDb.com. Possui duas colunas : uma contendo o texto da resenha, e a outra contendo um rótulo que classifica a resenha como positiva (1) ou negativa (0). Foi escolhida devido à grande quantidade de exemplos e ao fato de já estar estruturada como um problema de classificação binária, além de possuir distribuição 50/50 entre classes positiva e negativa [Maas *et al*., 2011].

3.2 Modelo de Linguagem

3.2.1 DistilRoBERTa

O modelo de linguagem escolhido para o treinamento foi o DistilRoBERTa, devido ao seu tamanho leve, disponibilidade gratuita e alto desempenho em tarefas de classificação de texto, além do fato de poder ser usado para *fine-tuning* em *datasets* específicos sem o requisito de recursos computacionais extensos [Shah *et al*., 2024]. O DistilRoBERTa destaca-se como uma excelente escolha para classificação de texto e análise de sentimento devido ao seu desempenho superior em comparação com outros modelos da mesma categoria. Além do alto desempenho, a principal vantagem técnica do DistilRoBERTa é a sua eficiência e tamanho reduzido. Como uma versão simplificada da arquitetura RoBERTa, ele foi projetado para utilizar menos camadas e parâmetros do que o modelo BERT original, visando diminuir significativamente o peso do modelo. Essa redução é complementada por um tokenizador especializado que utiliza um vocabulário menor, resultando em representações de tokens mais compactas que otimizam o processamento de texto sem comprometer a qualidade dos resultados. Dessa forma, o DistilRoBERTa oferece uma solução poderosa e rápida para o processamento de grandes volumes de dados, mesmo em ambientes com recursos computacionais limitados [Kyritsis *et al*., 2024].

3.2.2 GPT-3.5 turbo

Desde o lançamento inicial do ChatGPT em novembro de 2022, a OpenAI estabeleceu os padrões para modelos de linguagem de última geração. O GPT-3.5-turbo foi desenvolvido para equilibrar desempenho e eficiência, estima-se que possua cerca de 20 bilhões de parâmetros e o comprimento máximo de suas saídas atinge 4096 tokens. O modelo está disponível para ajuste fino por meio da API da OpenAI e, atualmente, seu tamanho relativamente menor é provavelmente o que o torna uma solução tão acessível em comparação com os muito maiores GPT-4 e GPT-4-turbo [Mazzullo *et al*., 2025].
O modelo GPT-3.5 turbo será utilizado para realizar uma classificação do conjunto de dados de validação nos seus rótulos correspondentes, por meio de um *prompt*.
Para realizar a classificação no conjunto de resumos médicos, o seguinte *prompt* foi utilizado: "*Você é um classificador altamente preciso de abstratos médicos. Você receberá um abstrato de um artigo medico e deve classificá-lo como 1, caso o mesmo trate da área de neoplasmas, ou 0 caso contrário. Responda apenas com*

'0' ou '1'."
Já para a classificação dos *reviews* de filmes contidos na base do IMDb, utilizou-se o seguinte *prompt*: "*Você é um classificador altamente preciso de reviews de filmes. Você receberá uma review e deve classificá-la como 1, caso a mesma seja positiva, ou 0 caso seja negativa. Responda apenas com '0' ou '1'.*"
O objetivo é comparar o desempenho, na classificação de texto, de um modelo de propósito geral amplamente usado com o de um modelo ajustado para a tarefa específica com um *fine-tuning* (o DistilRoBERTa, no caso deste experimento). O modelo foi escolhido levando em consideração as limitações computacionais e financeiras inerentes a este experimento. A conexão com o modelo será feita a partir da API da OpenAI e realiza um total de 3400 requisições (1700 para cada *dataset*).

3.3 Ferramentas

3.3.1 Python

A linguagem de programação Python (versão 3.11.9) será utilizada para a implementação computacional do experimento, junto com um conjunto de bibliotecas pertinentes às áreas abordadas no estudo. Python possui um ecossistema vasto e maduro de bibliotecas e frameworks, como NumPy, Pandas, TensorFlow, PyTorch e scikit-learn, que oferecem ferramentas otimizadas para processamento de dados, aprendizado de máquina e redes neurais profundas. Sua integração eficiente com linguagens de baixo nível (como C e C++) permite combinar facilidade de uso com alto desempenho computacional. Outro fator decisivo é a forte comunidade científica e de desenvolvedores, que garante ampla documentação, suporte contínuo e atualização constante das tecnologias, consolidando Python como a principal linguagem na pesquisa e no desenvolvimento de soluções baseadas em IA e aprendizado de máquina.

3.3.2 HuggingFace

A Hugging Face é uma plataforma e comunidade de código aberto que se tornou referência no campo do NLP e da inteligência artificial aplicada a textos. Seu principal destaque é a biblioteca Transformers, que disponibiliza uma ampla coleção de modelos pré-treinados baseados em arquiteturas modernas, como BERT, RoBERTa, GPT, T5 e DistilBERT, entre outros. Essa biblioteca simplifica a aplicação de técnicas avançadas de aprendizado profundo em tarefas como classificação de texto, tradução automática, sumarização, análise de sentimentos e reconhecimento de entidades nomeadas, com poucas linhas de código. Além disso, o ecossistema da Hugging Face inclui ferramentas para treinamento, avaliação e implantação de modelos, bem como uma comunidade ativa que compartilha pesos, *datasets* e experimentos. Assim, a Hugging Face e sua biblioteca Transformers democratizaram o acesso às tecnologias de ponta em NLP, permitindo que pesquisadores e desenvolvedores implementem soluções de IA sofisticadas de forma prática e eficiente. A biblioteca Transformers será utilizada para o carregamento dos modelos pré- treinados a serem utilizados.

3.3.3 SciKit-learn

A biblioteca scikit-learn será usada para a avaliação dos experimentos. A biblioteca Scikit-learn é uma das ferramentas mais utilizadas em projetos de aprendizado de máquina (machine learning) desenvolvidos em Python. Ela oferece uma ampla gama de algoritmos prontos para tarefas como classificação, regressão, clusterização, redução de dimensionalidade e validação de modelos. Além de sua facilidade de uso, o Scikit-learn é construído sobre bibliotecas robustas como NumPy, SciPy e Matplotlib, o que garante desempenho e integração com outras etapas do fluxo de ciência de dados. Sua API padronizada torna o processo de treinamento, teste e avaliação de modelos mais intuitivo, permitindo que tanto iniciantes quanto profissionais avancem rapidamente na implementação de soluções de IA.

3.3.4 Google Colab

O ambiente do Google Colab será utilizado para o treinamento em nuvem dos modelos. O Google Colab é uma plataforma gratuita baseada em nuvem que permite executar código Python diretamente em um ambiente interativo, sem a necessidade de instalação local. Amplamente utilizada na pesquisa e no ensino de inteligência artificial, o Colab oferece acesso gratuito a GPUs e TPUs, o que acelera significativamente o treinamento de modelos complexos. Além disso, sua integração com o Google Drive facilita o armazenamento e o compartilhamento de notebooks, promovendo a colaboração entre pesquisadores e desenvolvedores. Por ser baseado no Jupyter Notebook, o Colab combina código, texto explicativo, imagens e resultados em um único documento, tornando o processo de experimentação e aprendizado mais dinâmico e acessível.

3.4 Avaliação e Validação

3.4.1 K-fold Cross Validation

O *k-fold cross validation* é uma técnica de validação usada para avaliar o desempenho de um modelo de aprendizado de máquina de forma mais confiável. A estratégia consiste em dividir o conjunto de dados em k partes (ou "*folds*") de tamanhos iguais. O modelo é treinado k vezes, cada vez utilizando (k - 1) partes para treino e 1 parte diferente para teste, e em seguida as métricas de desempenho são calculadas em cada iteração. Finalmente, calcula-se a média dessas métricas, obtendo uma estimativa mais estável do desempenho do modelo. Nesse experimento, será utilizado k = 5, portanto o conjunto de dados será dividido em 5 partes iguais, onde a cada iteração uma das partes servirá como teste e as outras 4 como treinamento. Ao aproveitar esse método, maximizamos o uso dos dados disponíveis para treinamento e teste, evitando vazamento de dados entre os conjuntos, o que o torna particularmente adequado para cenários com conjuntos de dados anotados limitados [Rhoden, 2025]. Os *folds* foram estratificados, de modo a preservar em cada partição a proporção original entre as classes.

3.4.2 Planejamento fatorial 2^k

O planejamento fatorial 2^k é uma técnica experimental usada para estudar como k fatores (variáveis independentes) influenciam um resultado (variável dependente). Nesse método, cada variável é analisada em dois níveis, máximo e mínimo, e suas combinações são agrupadas em uma tabela na qual as colunas representam os fatores e combinações de fatores. Nesta, uma posição com entrada com valor igual a -1 denota o valor mínimo do parâmetro, enquanto 1, seu valor máximo. Assim, o experimento envolve 2^k combinações possíveis entre os níveis dos fatores [Prado and Leal, 2016]. Por exemplo, se houver 2 fatores ($k = 2$), serão realizadas 4 execuções experimentais, cobrindo todas as combinações possíveis. Essa metodologia permite Identificar efeitos principais (influência individual de cada fator), detectar interações entre fatores (como a combinação de dois fatores afeta o resultado) e planejar experimentos de forma eficiente, com o menor número possível de testes para obter informações relevantes. Neste experimento, serão analisados 2 fatores ($k = 2$) : o tamanho da amostra de treinamento e o tamanho do vocabulário do modelo.

3.4.3 Métricas de avaliação

A métrica utilizada nesse trabalho será a AUC-ROC (*Area Under the Receiver Operating Characteristic Curve*), uma das principais estratégias para avaliar modelos de classificação binária. Ela combina dois conceitos fundamentais:

- ROC Curve (*Receiver Operating Characteristic Curve*): é uma curva que mostra a relação entre a Taxa de Verdadeiros Positivos (TPR ou Sensibilidade) e a Taxa de Falsos Positivos (FPR = 1 − Especificidade), variando o limiar de decisão do classificador.
- AUC (*Area Under the Curve*): é a área sob a curva ROC, que resume em um único valor o desempenho do modelo para todos os limiares possíveis.

De acordo com a interpretação da métrica, um valor AUC = 1,0 implica um desempenho perfeito — o modelo separa completamente as classes positivas e negativas. um valor de AUC = 0,5 informa que o modelo não tem poder de discriminação — equivale a um palpite aleatório. Finalmente, um valor de AUC menor que 0,5 significa que o modelo está invertendo as classes (predizendo o oposto). Quanto maior o AUC, melhor o modelo em distinguir entre as duas classes.
A AUC tem sido tradicionalmente tratada como um critério importante para medir o desempenho de classificação de um modelo. Em geral, a maximização da AUC funciona bem quando existe uma distribuição de rótulos desbalanceada, ou a pontuação da AUC é a métrica padrão para avaliar e comparar diferentes métodos [Nguyen *et al*., 2023].

3.5 Parâmetros do Treinamento do modelo

Será analisado o impacto de dois fatores no desempenho do modelo escolhido : o tamanho do vocabulário e o tamanho da base de dados (em número de amostras). A base de dados terá três variações de tamanho : grande (5000) amostras, médio (2500 amostras) e pequeno (1000 amostras).
O vocabulário, por outro lado, será testado em duas modalidades:

- Vocabulário Original : vocabulário padrão usado pelo tokenizador que acompanha o modelo distilroberta-base. Possui 50265 tokens e foi gerado utilizando uma base de 160GB de texto, extraído de fontes como wikipedia, artigos de notícia em inglês e diversas outras páginas web [Liu *et al*., 2019]. Dessa forma, tem caráter genérico, contendo tokens relevantes para diversas áreas e assuntos.
- Vocabulário Reduzido : vocabulário criado para o experimento. Possui 5026 tokens (aproximadamente 10% do tamanho do vocabulário original) e foi gerado utilizando um corpus composto pelos extratos de texto contidos nas bases utilizadas. Dessa forma, o vocabulário utilizado no experimento com a base de abstratos médicos foi gerado usando esses mesmos abstratos como corpus. De forma análoga, o experimento com a base do IMDb usou um vocabulário gerado a partir dos textos de resenhas críticas de filmes, contidas na própria base. Dessa forma, temos um vocabulário reduzido e especializado, gerado a partir de textos relevantes para a área de classificação. Os *embeddings* do modelo foram reajustados para o novo tamanho do vocabulário.

Já que o vocabulário foi analisado em dois níveis (original e reduzido), para propósitos de realização do experimento fatorial 2^k, serão considerados apenas os tamanhos pequeno e grande para cada uma das bases de dados, representando os níveis mínimos e máximos, respectivamente. Dessa forma, possibilita-se que cada fator seja analisado igualmente em dois níveis, como é próprio do experimento 2^k.
Os hiperparâmetros (número de épocas, taxa de aprendizagem) foram ajustados separadamente para cada conjunto de dados utilizando o otimizador Optuna (biblioteca em python), que ajusta os parâmetros utilizando otimização bayesiana, mais especificamente TPE (do inglês *Tree Structured Parzen Estimator*). A métrica alvo da otimização foi o erro de validação (*validation loss*), sendo este minimizado, garantindo assim que o AUC-ROC obtido pelo treinamento fosse representativo do modelo com o melhor poder de generalização. O ajuste foi feito separadamente em cada *fold*, com 5 testes (*trials*) em cada um. O intervalo de busca para a taxa de aprendizagem foi de $1 * 10^{-6}$ a $1 * 10^{-4}$, e o número de épocas foi testado entre 3 e 7.

3.6 Diagrama do experimento

O experimento pode ser decomposto em 3 subexperimentos independentes, cujos resultados serão comparados ao final do processo. O diagrama dos 3 subexperimentos está ilustrado na Figura 3.

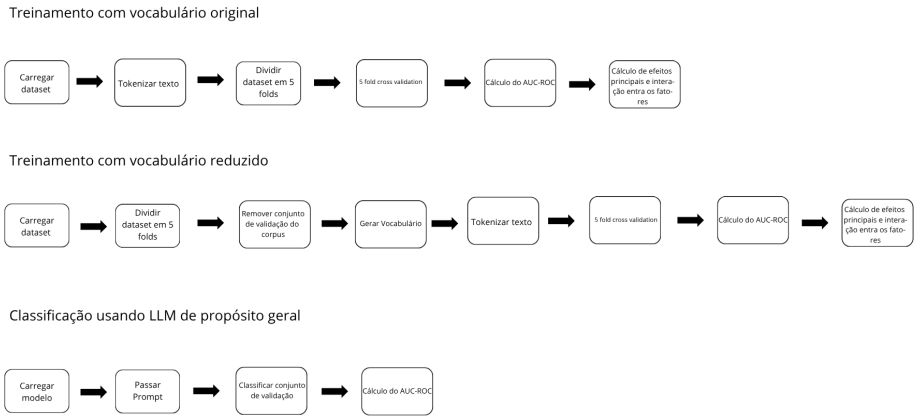


Figura 3. Diagrama do experimento.

Cada experimento é realizado para cada um dos 3 tamanhos de amostra selecionados, em cada um dos dois *datasets*. Os 3 sub-experimentos são explicados a seguir:

- Treinamento com vocabulário original : de acordo com o *k-fold* (k = 5), o texto de entrada é tokenizado com o vocabulário original do modelo e dividido em 5 partes. O treinamento é então realizado cinco vezes, alternando cada vez qual parte é usada como conjunto de validação, enquanto as outras 4 são usadas para o treinamento em si. Finalmente, os valores de AUC-ROC resultantes são usados para calcular os efeitos individuais e de interação entre os fatores (vocabulário e tamanho da base)
- Treinamento com vocabulário reduzido : o texto de entrada é dividido em 5 partes primeiro, e a tokenização é feita durante cada ciclo do *k-fold* (5 ciclos): durante cada ciclo, os textos pertencentes ao conjunto de validação são removidos do corpus que treina o tokenizador antes da geração do vocabulário, garantindo assim que não haja vazamento de dados. Feito isso, o vocabulário é gerado e os conjuntos de treinamento e validação são tokenizados, e o *k-fold* prossegue normalmente. Ao final, os valores de AUC-ROC resultantes são usados para calcular os efeitos individuais e de interação entre os fatores.
- Classificação usando LLM de propósito geral: usando uma LLM de propósito geral (GPT-3.5 turbo), classificamos o conjunto de validação, através de um *prompt* passado para o modelo. Dessa forma, comparamos o desempenho dos dois treinamentos realizados com o de um modelo popular genérico, sem qualquer *fine-tuning* para a tarefa de classificação desejada.

4 Resultados

4.1 Treinamento

As matrizes de confusão agregadas correspondentes para cada treinamento estão ilustradas nas Figuras 4 e 5 para o *dataset* médico, e nas Figuras 6 e 7 para o *dataset* do IMDb. Os quadrantes representam, da esquerda para a direita, de cima pra baixo: negativos-verdadeiros, falso-positivos, falso-negativos e positivos-verdadeiros. Com relação à acurácia (capacidade de classificar exemplos corretamente), precisão (capacidade de classificar exemplos positivos corretamente, evitando "alarmes falsos", ou falsos positivos) e *recall* (capacidade de encontrar todos os casos positivos), exibidos nas tabelas 1 (médico) e 2 (IMDb) observamos que as métricas precisão e acurácia tendem a aumentar de forma geral com o tamanho do *dataset*, independente da modalidade de vocabulário, embora os modelos com vocabulários reduzidos obtenham ganhos proporcionalmente mais expressivos que os vocabulários originais para a mesma base (ainda que o desempenho desses últimos seja maior). O recall, por outro lado, tem comportamento errático entre aumentos no tamanho da base de dados, tendo seu valor decrescido em alguns casos. Como as classes estão balanceadas e os *folds* são estratificados, esse comportamento não decorre de desbalanceamento, sendo na verdade causado apenas pelo aumento na fração de casos positivos que o modelo falha em classificar (falsos negativos).

O desempenho médio dos três modelos, para cada tamanho de *dataset*, pode ser visto na Figura 8. É possível observar que, para a classificação do *dataset* de abstratos médicos, o modelo DistilRoBERTa com o vocabulário original obteve o melhor desempenho, superando tanto o mesmo modelo com o vocabulário reduzido quanto até mesmo um modelo de propósito geral popular como o GPT-3.5 turbo. Além disso, nota-se que o DistilRoBERTa com vocabulário reduzido apresentou um desempenho inconsistente ao longo dos 5 *folds*, para os tamanhos de 1000 e 2500, com um desvio padrão (representado pelas linhas verticais) mais alto que as outras duas modalidades. Observa-se também que essa variante supera o GPT-3.5 turbo para os tamanhos médio e grande. Já na classificação do *dataset* de *reviews* do IMDb, o GPT-3.5 turbo e o modelo com vocabulário original apresentaram desempenhos próximos entre si (com o

GPT-3.5 turbo sendo ligeiramente melhor) e vastamente superiores ao modelo com vocabulário reduzido.

É interessante notar que a variante com vocabulário reduzido, apesar de apresentar desempenho inferior para todos os tamanhos de *dataset*, tem o seu próprio desempenho aumentado consistentemente com o aumento do tamanho do *dataset* para ambas as bases de dados, enquanto as outras duas modalidades mantêm seus desempenhos relativamente constantes para todos os três tamanhos escolhidos. É possível que, para uma quantidade suficientemente grande de dados, o vocabulário reduzido gere um desempenho próximo ou até mesmo superior às outras duas modalidades.

Os valores de AUC-ROC das 3 modalidades no *dataset* médico, ao longo dos 5 *folds*, são exibidos na Figura 9. Observa-se que, para todas as variações de tamanho da base, o vocabulário original prevalece em desempenho sobre o vocabulário reduzido. Para o *dataset* do IMDb, cujo desempenho por *fold* está exibido na Figura 10, o vocabulário genérico mostrou desempenho próximo ao GPT-3.5 turbo (equivalente ou ligeiramente inferior para o tamanho de 5000 (Figura 10c) e 2500 amostras (Figura 10b), e ambas as modalidades superam o vocabulário reduzido em todos os *folds*, para todos os tamanhos testados.

4.2 2^k fatorial e interação entre fatores

Os resultados para o experimento fatorial, de acordo com cada combinação de níveis para os dois fatores (vocabulário e tamanho da base), podem ser vistos para o *dataset* médico na Tabela 3, e para o *dataset* IMDb na Tabela 4. Com base nesses resultados, é possível calcular os efeitos individuais de cada fator e o efeito de interação entre os mesmos, mostrados na Tabela 5. Para o *dataset* médico, nota-se que há uma interação relativamente pequena entre os dois fatores, quando comparada aos seus efeitos individuais. O aumento no tamanho do *dataset* do nível mínimo para o nível máximo resulta em um aumento médio de 0.0668 no valor do AUC-ROC. Já a mudança do tamanho do vocabulário do mínimo (reduzido) para o máximo (genérico) resulta em um aumento médio de 0.1078 no valor do AUC-ROC.

Quanto ao *dataset* do IMDb, notamos um impacto mais de duas vezes maior no desempenho ao mudar o tipo de vocabulário, sendo essa também a contribuição predominante. O efeito de interação entre os fatores é semelhante em valor absoluto, porém tem contribuição proporcionalmente menor que na base médica. A tabela 6 mostra as parcelas individuais de cada fator com relação à soma dos quadrados total, indicando assim o papel relativo de cada fator na variação total dos dados. Para o *dataset* médico, vemos que 56.03% da variação total pode ser atribuída ao tamanho do vocabulário, enquanto 21.2% decorre do tamanho do *dataset*, e 22.7% da interação entre os dois. Por outro lado, no *dataset* do IMDb o tamanho do vocabulário contribui 79.90%, enquanto o tamanho do *dataset* tem um impacto muito menor (12.32%), e a interação entre os dois fatores tem uma contribuição relativamente menor (7.78%).

Tabela 1. Métricas das matrizes de confusão (*dataset* médico)

Vocab. / Tamanho	Original			Reduzido		
	Acurácia	Precisão	Recall	Acurácia	Precisão	Recall
1000	0.8730	0.8414	0.9148	0.6890	0.6750	0.7120
2500	0.8760	0.8619	0.8938	0.7264	0.7855	0.6187
5000	0.8728	0.8687	0.8784	0.8312	0.8275	0.8368

Tabela 2. Métricas das matrizes de confusão (*dataset* IMDb)

Vocab. / Tamanho	Original			Reduzido		
	Acurácia	Precisão	Recall	Acurácia	Precisão	Recall
1000	0.8410	0.8163	0.8742	0.5750	0.5895	0.4544
2500	0.8528	0.8196	0.9027	0.6168	0.6670	0.4578
5000	0.8532	0.8538	0.8524	0.7012	0.7135	0.6724

Tabela 3. Níveis de parâmetros máximo (+) e mínimo (-) e resultados correspondentes para o *dataset* de abstratos médicos

Vocabulário	Tamanho da Base	AUC-ROC
-	-	0.7568
-	+	0.8926
+	-	0.9336
+	+	0.9315

Tabela 4. Níveis de parâmetros máximo (+) e mínimo (-) e resultados correspondentes para o *dataset* de *reviews* de filmes

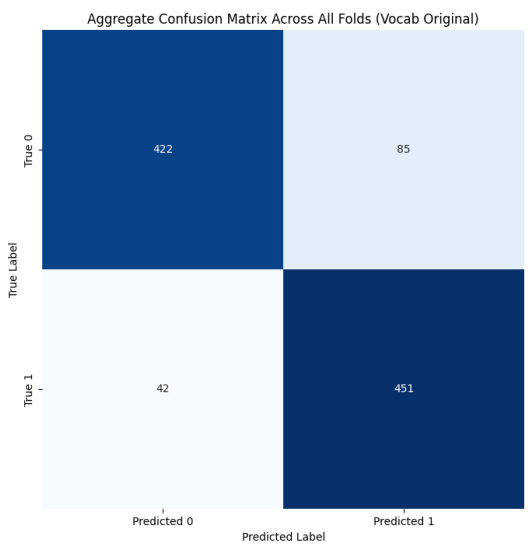
Vocabulário	Tamanho da Base	AUC-ROC
-	-	0.6193
-	+	0.7763
+	-	0.9109
+	+	0.9289

Tabela 5. Efeitos principais e interação entre os fatores

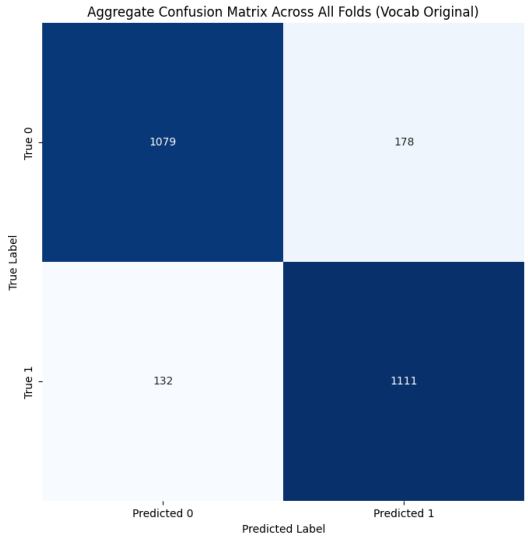
Base de Dados	Efeito Vocabulário	Efeito *Dataset*	Efeito de Interação
Base Médica	0.1078	0.0668	- 0.0689
Base IMDb	0.2221	0.0875	-0.0695

Tabela 6. parcelas dos efeitos (soma dos quadrados)

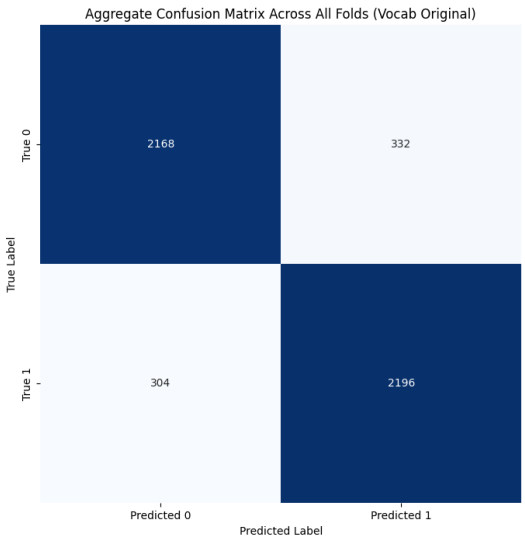
Base	Vocab.	*Dataset*	Interação	Total (soma)
Médica	0.0116	0.0044	0.0047	0.0207
IMDb	0.0493	0.0076	0.0048	0.0617



(a) matriz de confusão (1000 amostras)

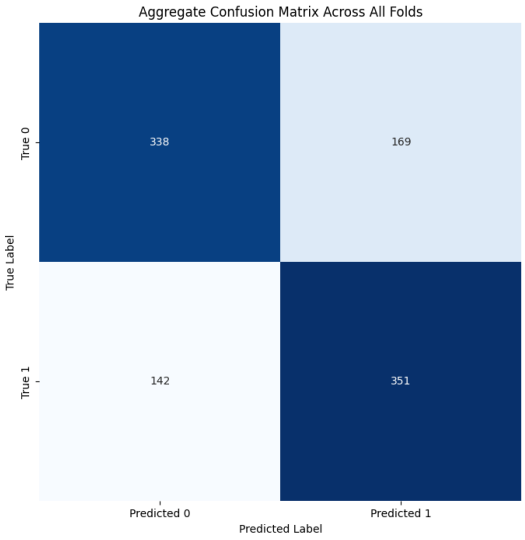


(b) matriz de confusão (2500 amostras)

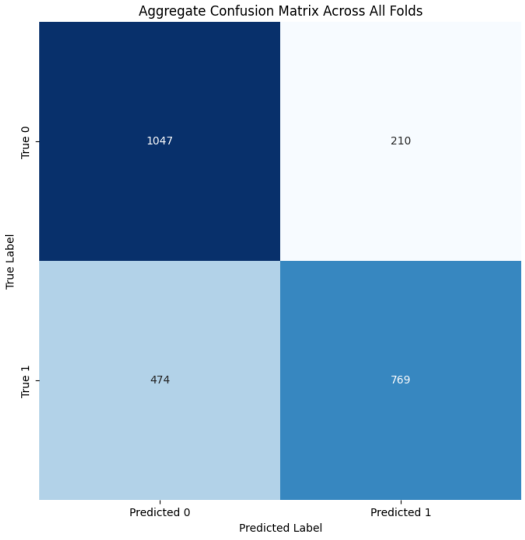


(c) matriz de confusão (5000 amostras)

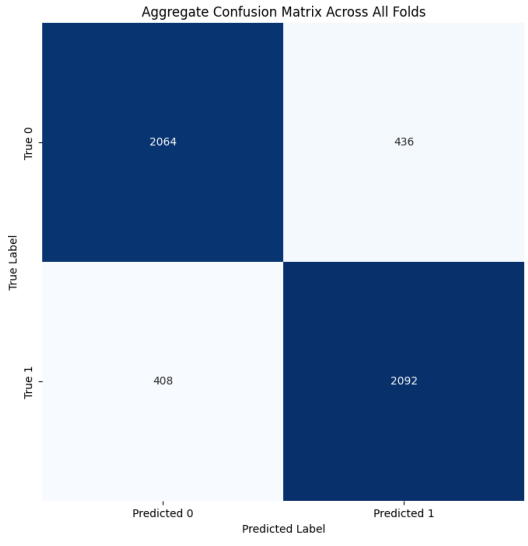
Figura 4. Matrizes de confusão agregadas (*dataset* medico c vocab original)



(a) matriz de confusão (1000 amostras)

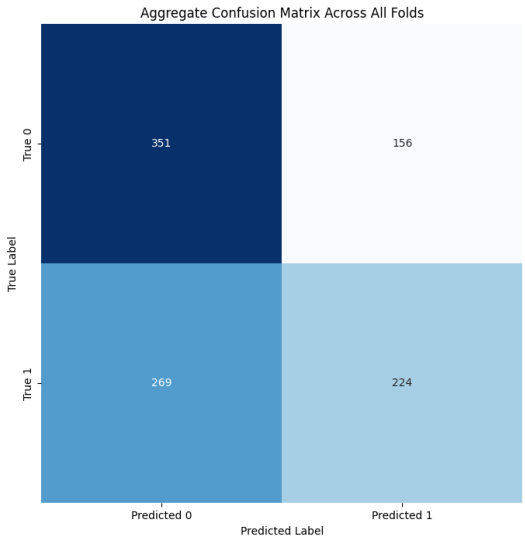


(b) matriz de confusão (2500 amostras)

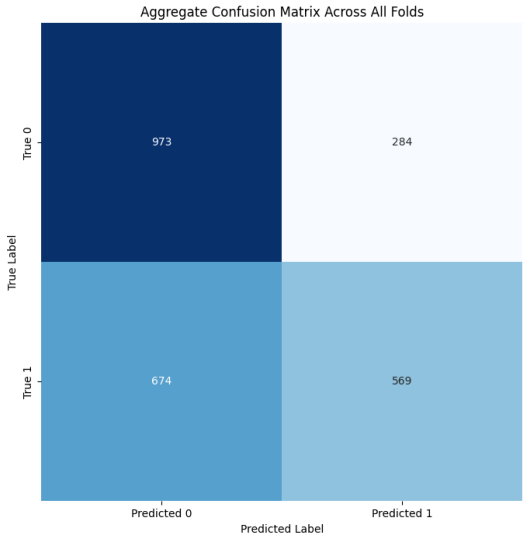


(c) matriz de confusão (5000 amostras)

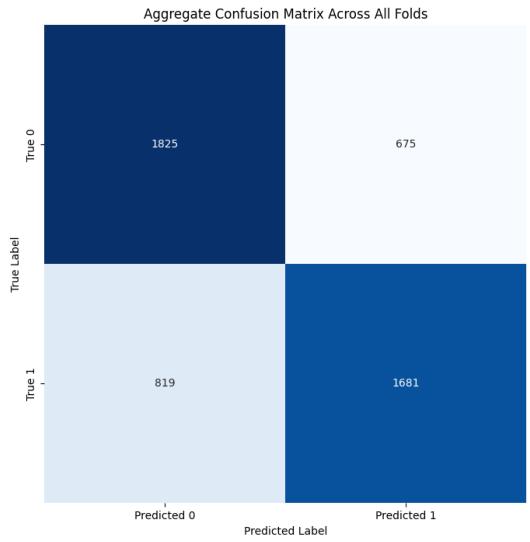
Figura 5. Matrizes de confusão agregadas (*dataset* médico c/ vocab. reduzido)



(a) matriz de confusão (1000 amostras)

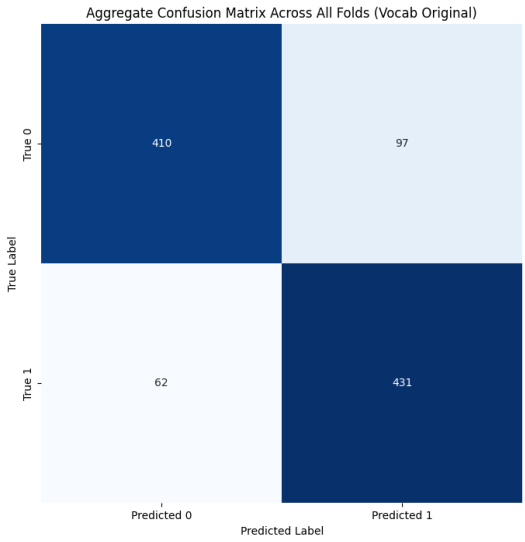


(b) matriz de confusão (2500 amostras)

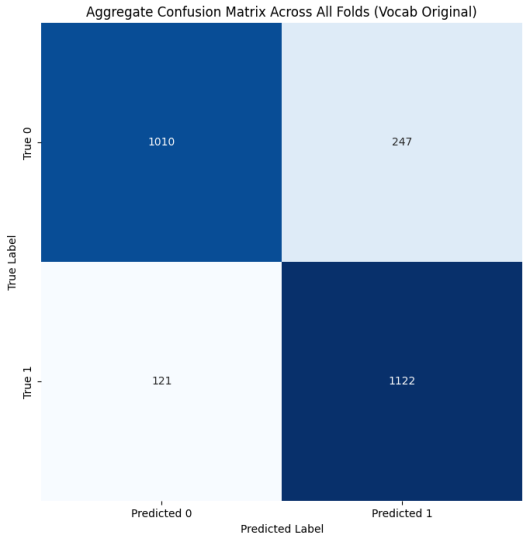


(c) matriz de confusão (5000 amostras)

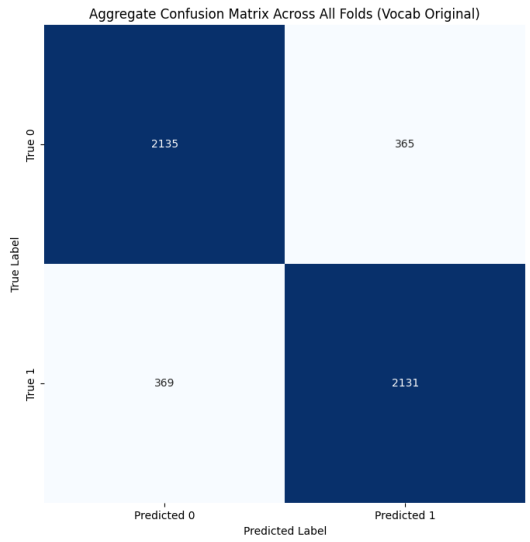
Figura 6. Matrizes de confusão agregadas (*dataset* IMDb c/ vocab. reduzido)



(a) matriz de confusão (1000 amostras)

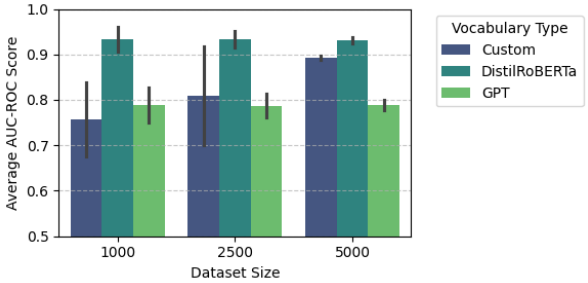


(b) matriz de confusão (2500 amostras)

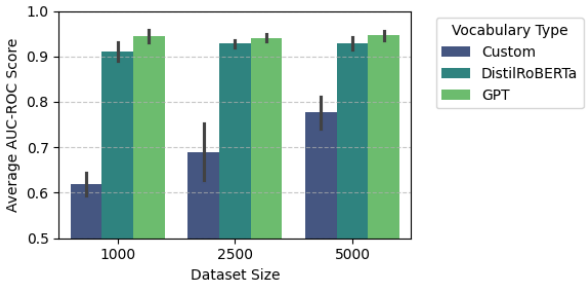


(c) matriz de confusão (5000 amostras)

Figura 7. Matrizes de confusão agregadas (*dataset* IMDb c/ vocab. original)

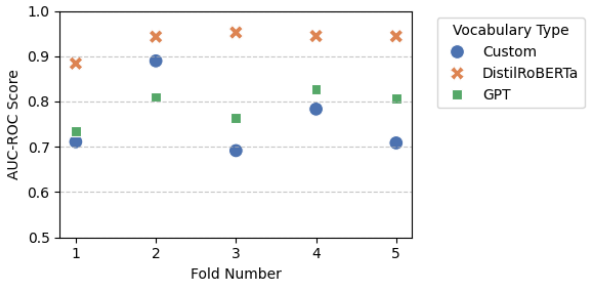


(a) Valor AUC-ROC médio (*dataset* médico)

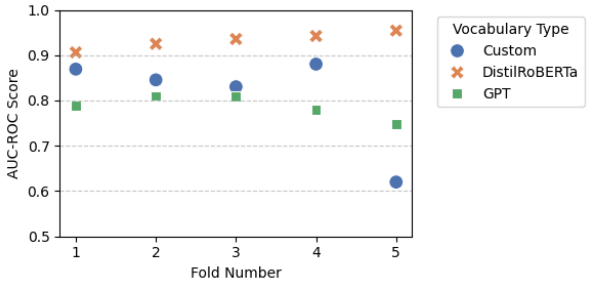


(b) Valor AUC-ROC médio (*dataset* do IMDb)

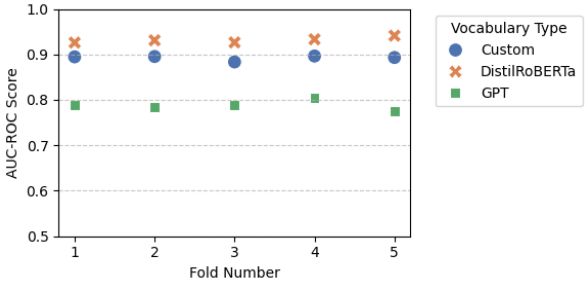
Figura 8. Valor AUC-ROC para ambos os *datasets* (média dos 5 *folds*)



(a) Valores AUC-ROC por *fold* (1000 amostras)

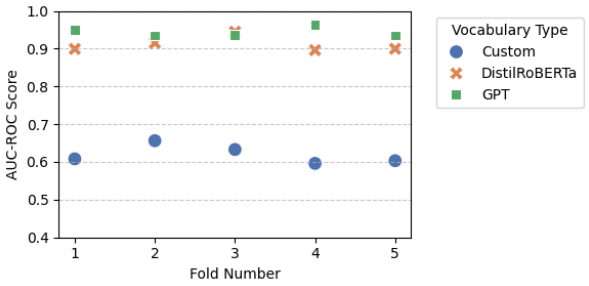


(b) Valores AUC-ROC por *fold* (2500 amostras)

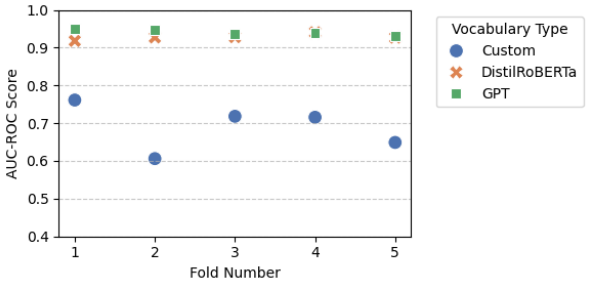


(c) Valores AUC-ROC por *fold* (5000 amostras)

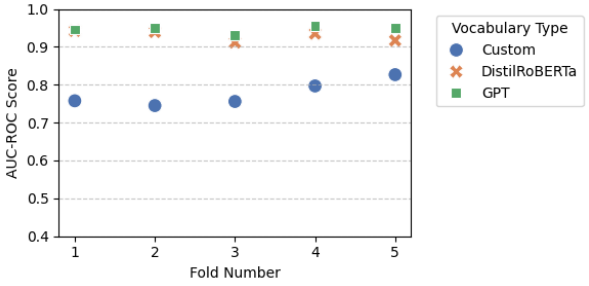
Figura 9. Valores AUC-ROC por *fold* para cada tamanho do *dataset* médico. O termo "Custom"(legenda) se refere ao vocabulário reduzido



(a) Valores AUC-ROC por *fold* (1000 amostras)



(b) Valores AUC-ROC por *fold* (2500 amostras)



(c) Valores AUC-ROC por *fold* (5000 amostras)

Figura 10. Valores AUC-ROC por *fold* para cada tamanho do *dataset* do IMDb. O termo "Custom"(legenda) se refere ao vocabulário reduzido

5 Conclusão

Este trabalho investigou o impacto do tamanho da base de treinamento e do tamanho do vocabulário no desempenho de modelos de classificação de texto, utilizando o DistilRoBERTa como modelo-base e comparando seus resultados com os obtidos por uma LLM de propósito geral, o GPT-3.5-turbo. Para isso, foram conduzidos experimentos em dois conjuntos de dados distintos, um composto por resumos de artigos médicos e outro por avaliações de filmes do IMDb, avaliados por meio da métrica AUC-ROC e analisados através de validação cruzada *k-fold* e planejamento fatorial 2^k. Os experimentos foram realizados para três tamanhos de *dataset* e duas modalidades de vocabulário.

Os resultados indicam que a generalização do vocabulário pode ter influência expressiva no desempenho em domínios específicos. No conjunto de resumos médicos, o DistilRoBERTa com vocabulário genérico obteve AUC-ROC médio em torno de 0.93 (em todos os tamanhos de base), superior ao obtido com o vocabulário reduzido (0.76 com 1000 amostras a 0.89 com 5000 amostras) e ao do GPT-3.5-turbo (aproximadamente 0.79), além de menor variabilidade entre os *folds*. Na análise fatorial (Seção 4.2), a troca do vocabulário reduzido pelo original elevou o AUC-ROC em 0.1078 em média, contra 0.0668 associado ao aumento da amostra de 1000 para 5000, e o vocabulário respondeu por 56.03% da variação total, contra 21.2% do tamanho da amostra e 22.7% da interação.

No IMDb, o tamanho do vocabulário teve influência muito maior do que a amostra: o aumento de 1000 para 5000 amostras elevou o AUC-ROC em 0.0875 em média, contra 0.2221 associado à troca para o vocabulário original, e o tamanho do vocabulário respondeu por 79.90% da variação total, contra 12.32% da amostra e 7.78% da interação (Tabela 6).

Os valores de AUC-ROC dos modelos ajustados mostram que o melhor desempenho foi o do modelo com o vocabulário genérico e maior, tendo este desempenho pouco se alterado com a alteração no tamanho da amostra e sido próximo ao modelo GPT-3.5 turbo para a base do IMDb. A modalidade com vocabulário reduzido obteve ganhos expressivos no AUC-ROC à medida que se aumentou o tamanho da amostra, tendo até mesmo superado eventualmente o GPT-3.5 turbo para a base médica, embora seu desempenho tenha se mantido menor do que os demais modelos na base do IMDb.

De forma geral, os resultados indicam que a simples utilização de vocabulários extensos e genéricos, quando em posse de um conjunto de dados relativamente pequeno, pode conduzir ao melhor desempenho em tarefas de classificação de texto em domínios específicos (como o de classificação médica). Os resultados também evidenciam que o impacto relativo dos fatores analisados depende fortemente das características do conjunto de dados, do domínio da aplicação e da otimização dos hiper-parâmetros para cada domínio.

Como limitações, a base médica foi binarizada (neoplasmas contra as demais classes), o que simplifica o problema clínico original, e a análise fatorial foi realizada sem teste estatístico.

Declarações complementares

Contribuições dos autores

IATA foi responsável pela realização integral do estudo, incluindo o desenho experimental, a execução dos experimentos, a análise dos resultados e a redação do manuscrito. Os demais autores participaram da concepção do trabalho e da revisão do texto. Todos os autores leram e aprovaram a versão final.

Conflitos de interesse

Os autores declaram que não há conflitos de interesse.

Disponibilidade de artefatos de pequisa

Os dados utilizados neste estudo foram disponibilizados em formato CSV sob o nome de arquivo ***rica-v1-n1-tenorio-et-al-2026-data.csv*** *juntamente com o código-fonte dos experimentos conduzidos, em formanto Interactive Python Notebook sob o nome de arquivo* ***rica-v1-n1-tenorio-et-al-2026-experimento.ipynb****. Ambos os arquivos encontram-se no repositório da revista.*

Uso de Inteligência Artificial

Ferramentas de inteligência artificial generativa (TeXGPT, através do editor LaTeX Overleaf) foram utilizadas apenas como apoio à revisão textual e à formatação do manuscrito. Os autores revisaram todo o conteúdo e assumem total responsabilidade pelo trabalho apresentado.

Referências

Hirschberg, J. and Manning, C. D. (2015). Advances in natural language processing. *Science*, 349(6245):261–266.

Kadhim, A. I. (2019). Survey on supervised machine learning techniques for automatic text classification. *Artif. Intell. Rev.*, 52(1):273–292.

Kostina, A., Dikaiakos, M. D., Stefanidis, D., and Pallis, G. (2025). Large language models for text classification: Case study and comprehensive review.

Kyritsis, K., Spatiotis, N., Perikos, I., and Paraskevas, M. (2024). A comparative performance evaluation of algorithms for the analysis and recognition of emotional content. In *Artificial Intelligence*. IntechOpen.

Liddy, E. D. (2001). Natural language processing. In *Encyclopedia of Library and Information Science*, volume 2, pages 2126–2136. Marcel Dekker, Inc., New York, 2nd edition.

Liu, Y. *et al*. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Mazzullo, E., Bulut, O., Walsh, C., Sitarenios, G., and MacIntosh, A. (2025). Fine-tuning GPT-3.5-turbo for automatic feedback generation. In *Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing*, pages 40–47, New York, NY, USA. ACM.

Mielke, S. J. *et al*. (2021). Between words and characters: A brief history of open-vocabulary modeling and tokenization in nlp. arXiv preprint.

Minaee, S. *et al*. (2024). Large language models: A survey. arXiv preprint.

Nakash, I., Calderon, N., Ben David, E., Hoffer, E., and Reichart, R. (2025). AdaptiVocab: Enhancing LLM efficiency in focused domains through lightweight vocabulary adaptation.

Nguyen, N. D. *et al*. (2023). Low-resource named entity recognition: Can one-vs-all auc maximization help? In *2023 IEEE International Conference on Data Mining (ICDM). IEEE*, pages 1241–1246.

Prado, L. C. and Leal, B. G. (2016). Meteorological variables influence analysis over petrolina-juazeiro pole monthly evapotranspiration. *IEEE Latin America Transactions*, 14(10):4373–4376.

Reshamwala, A., Mishra, D., and Pawar, P. (2013). Review on natural language processing. *IRACST Engineering Science and Technology: An International Journal (ESTIJ)*, 3(1):113–116.

Rhoden, E. B. (2025). Exploring active sampling strategies for self-training in named entity recognition.

Schopf, T., Braun, D., and Matthes, F. (2022). Evaluating unsupervised text classification: Zero-shot and similarity-based approaches. In *Proceedings of the 2022 6th International Conference on Natural Language Processing and Information Retrieval*, pages 6–15, New York, NY, USA. ACM.

Shah, M. S. S., Abuaieta, A. M., and Almazrouei, S. S. (2024). Safeguarding online communications using DistilRoBERTa for detection of terrorism and offensive chats. *J. Inf. Secur. Cybercrimes Res.*, 7(1):93–107.

Vajjala, S. and Shimangaud, S. (2025). Text classification in the LLM era – where do we stand?

Vaswani, A. *et al*. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Wankhade, M., Rao, A. C. S., and Kulkarni, C. (2022). A survey on sentiment analysis methods, applications, and challenges. *Artif. Intell. Rev.*, 55(7):5731–5780.

Zhao, W. X. *et al*. (2023). A survey of large language models. arXiv preprint v. 1, n. 2.

Relato de Experiência

Da engenharia de prompts à mediação pedagógica: concepção e avaliação formativa em tempo real com a plataforma EducAI

From prompt engineering to pedagogical mediation: design and real-time formative assessment with the EducAI platform

Rogério Fabiano Gonçalves [Universidade de Pernambuco, Campus Petrolina | *rogerio.goncalves@upe.br*]
Tarcísio Fulgêncio Alves da Silva [Universidade de Pernambuco, Campus Petrolina | *tarcisio.silva@upe.br*]
Emanuelle Francine Detogni Schmit [Universidade de Pernambuco, Campus Petrolina | *emanuelle.schmit@upe.br*]
Nayara Priscila Dantas de Oliveira [Universidade de Pernambuco, Campus Petrolina | *nayara.oliveira@upe.br*]
Rejane Cristiany Lins de França Pereira [Universidade de Pernambuco, Campus Ouricuri | *rejane.lins@upe.br*]
Paulo Adriano Schwingel [Universidade de Pernambuco, Campus Petrolina | *paulo.schwingel@upe.br*]

Colegiado de Fisioterapia, Universidade de Pernambuco, Campus Petrolina, Rodovia BR 203, Km 2, s/n, Vila Eduardo, Petrolina, Pernambuco, 56328-900, Brasil.

Resumo. A elaboração de avaliações diversificadas, contextualizadas e coerentes com os objetivos de aprendizagem é uma das atividades mais trabalhosas do ensino superior, pois exige itens fiéis ao conteúdo, com distratores plausíveis e diferentes níveis cognitivos. A difusão dos modelos de linguagem de grande porte abriu a possibilidade de automatizar parte dessa tarefa, mas trouxe desafios relativos à fidelidade factual, à qualidade pedagógica dos itens e ao risco de deslocamento do docente de sua função de curadoria. Este artigo objetiva relatar a concepção, a evolução e a aplicação da plataforma EducAI, ferramenta web que utiliza inteligência artificial generativa para produzir questões inéditas e comentadas a partir de materiais didáticos anexados em PDF ou de temas definidos pelo usuário. Trata-se de relato de caráter exploratório, centrado no processo, e não na mensuração de eficácia ou na validação da plataforma. São descritas as decisões de arquitetura: aplicação reunida em contêiner único, com *backend* em *Python* e interface de página única, geração delegada a um modelo acessado por API externa e organização em três perfis de uso — estudante, monitor e docente —, com recursos e permissões distintos. A extração de conteúdo é híbrida, combinando leitura estruturada de documentos nativos e reconhecimento óptico de caracteres para slides e digitalizações; o sistema oferece quatro modos de geração, diferentes formatos de item, estilos de exames brasileiros e associação entre dificuldade e Taxonomia de Bloom revisada, com possibilidade de fixar o nível cognitivo de forma independente. Detalha-se também a evolução da engenharia de prompts, tratada como artefato de software versionável e testável: das primeiras versões, com regras textuais diretas para evitar vícios recorrentes da geração automática, à incorporação de rubricas analíticas para questões abertas e, posteriormente, a uma arquitetura com persona fixa de elaborador de itens, separação entre dificuldade e nível cognitivo, distratores concebidos como diagnósticos e bases de conhecimento estruturadas acionadas de forma seletiva. Complementam esse percurso decisões voltadas à robustez e ao desempenho, como *parser* tolerante a saídas malformadas, mecanismos de autocorreção, balanceamento da posição do gabarito e paralelização de chamadas. A experiência ocorreu entre 2025 e 2026, no âmbito de um edital de inovação pedagógica da Universidade de Pernambuco, em componentes curriculares e na monitoria de um curso da área da saúde, com encontros presenciais integrados às atividades regulares e sessões em que as questões eram projetadas para discussão coletiva antes da apresentação do gabarito comentado. Nos contextos acompanhados, a geração de questões em tempo real integrou-se às aulas como apoio à discussão, enquanto o uso extraclasse permaneceu pontual e variou conforme a preferência dos docentes; como não houve registro sistemático de frequência por modalidade, essa diferença é tratada como aprendizado contextual, e não como comparação de efetividade. Ao longo do desenvolvimento, percepções exploratórias de docentes e estudantes orientaram ajustes sucessivos na ferramenta, apontaram distratores ambíguos e limites na gradação de dificuldade e reforçaram a importância da revisão docente, sem finalidade de mensurar aprendizagem, eficácia ou usabilidade validada. Conclui-se que, nos contextos acompanhados, a geração automatizada de questões adquiriu sentido formativo quando associada à mediação e à validação docente, e que tratar a engenharia de prompts como componente de software sujeito a versionamento e a teste torna as decisões mais rastreáveis, sem garantir, por si só, a qualidade pedagógica.

Abstract. Designing diversified, contextualised assessment items that are consistent with learning objectives is one of the most labour-intensive activities in higher education, as it requires items faithful to the source content, with plausible distractors and varied cognitive levels. The spread of large language models has made it possible to automate part of this task, but it has also raised concerns regarding factual fidelity, the pedagogical quality of the items and the risk of displacing instructors from their curatorial role. This article reports the design, evolution and use of the EducAI platform, a web tool that employs generative artificial intelligence to produce original, annotated assessment items from instructional materials uploaded as PDF files or from topics defined by the user. This is an exploratory report centred on the process rather than on measuring effectiveness or validating the platform. The architectural decisions are described: the application is bundled in a single container, with a Python backend and a single-page interface, generation delegated to a model accessed through an external API, and organisation into three user profiles — student, teaching assistant and instructor —, each with distinct features and permissions. Content extraction is hybrid, combining structured parsing of native documents with optical

character recognition for slides and scanned material; the system offers four generation modes, different item formats, styles drawn from Brazilian national examinations, and an association between difficulty and the revised Bloom's taxonomy, with the option of fixing the cognitive level independently. The article also details the evolution of prompt engineering, treated as a versionable and testable software artefact: from early versions relying on direct textual rules to avoid recurrent flaws of automatic generation, to the incorporation of analytical rubrics for open-ended items and, subsequently, to an architecture combining a fixed item-writer persona, the separation of difficulty from cognitive level, distractors designed as diagnostic, and structured knowledge bases retrieved selectively. This trajectory is complemented by decisions aimed at robustness and performance, such as a parser tolerant of malformed outputs, self-healing mechanisms, balancing of answer-key position and parallelised API calls. The experience took place between 2025 and 2026, within an institutional call for pedagogical innovation at the University of Pernambuco, Brazil, in curricular components and in the tutoring activities of a health sciences programme, with in-person meetings integrated into regular activities and sessions in which items were projected for collective discussion before the annotated answer key was shown. In the observed contexts, real-time item generation was integrated into classes as support for discussion, whereas out-of-class use remained occasional and varied according to instructors' preferences; since no systematic record of frequency by modality was kept, this difference is treated as contextual learning rather than as a comparison of effectiveness. Throughout development, exploratory perceptions from instructors and students guided successive adjustments to the tool, pointed out ambiguous distractors and limitations in difficulty gradation, and reinforced the importance of teacher review, with no intention of measuring learning, effectiveness or validated usability. It is concluded that, in the observed contexts, automated item generation acquired formative meaning when combined with teacher mediation and validation, and that treating prompt engineering as a software component subject to versioning and testing makes decisions more traceable, without in itself guaranteeing pedagogical quality.

Palavras-chave: Inteligência Artificial Generativa, Engenharia de Prompts, Geração Automática de Questões, Avaliação Formativa, Ensino Superior

Keywords: Generative Artificial Intelligence, Prompt Engineering, Automatic Question Generation, Formative Assessment, Higher Education

Recebido/Received: 14 08 2026 • **Aceito/Accepted:** 23 09 2026 • **Publicado/Published:** 15 10 2026

Publicado sob a Licença Pública Internacional Creative Commons Atribuição 4.0 (CC BY 4.0)

1 Introdução

A elaboração de avaliações diversificadas, contextualizadas e cognitivamente exigentes é uma das atividades mais trabalhosas do ensino superior. Construir itens fiéis ao conteúdo, com distratores plausíveis e diferentes níveis cognitivos, consome tempo que poderia ser destinado à mentoria, ao *feedback* e à mediação da aprendizagem. A difusão dos modelos de linguagem de grande porte (*Large Language Models*, LLMs) abriu a possibilidade de automatizar parte dessa tarefa, mas também trouxe desafios: fidelidade factual, qualidade pedagógica dos itens e risco de deslocamento do docente de sua função de curadoria [Zhai and Nehm, 2023; Xia *et al*., 2024].

Estudos recentes têm explorado LLMs para gerar questões a partir de textos, objetivos de aprendizagem e níveis da Taxonomia de Bloom revisada. Os resultados demonstram a viabilidade técnica da geração automática, mas também mostram que a qualidade depende das instruções fornecidas ao modelo e da revisão humana [Lee *et al*., 2024; Scaria *et al*., 2024]. Assim, o desafio não se limita a gerar questões: envolve integrá-las a uma prática avaliativa pedagogicamente orientada.

A plataforma EducAI foi concebida e desenvolvida pelos próprios autores nesse contexto, no âmbito do Edital de Inovação Pedagógica PROGRAD/PFA/UPE nº 05/2025, da Universidade de Pernambuco (UPE). Trata-se de uma aplicação web que emprega inteligência artificial (IA) generativa para criar questões inéditas e comentadas a partir de materiais didáticos anexados em PDF ou de temas especificados pelo usuário. Sua concepção procura conciliar fidelidade ao conteúdo, qualidade pedagógica e possibilidade de uso em tempo real, sem retirar do docente a responsabilidade pela avaliação. Cabe registrar que os autores acumulam as posições de desenvolvedores e de docentes usuários da ferramenta, condição explicitada aqui por seu efeito sobre o alcance do relato: as observações reunidas resultam de um olhar interno ao processo, e não de avaliação independente.

Este artigo objetiva relatar o processo de concepção, evolução e aplicação do EducAI em componentes curriculares da área da saúde. O relato articula três dimensões: as decisões de engenharia de software; a evolução da engenharia de prompts, tratada como artefato versionável e testável; e os aprendizados decorrentes do uso em sala e na monitoria. Trata-se de um relato de caráter exploratório, centrado no processo — concepção, evolução e aplicação — e não na mensuração de eficácia ou na validação da plataforma; seu propósito é sistematizar um percurso de implementação que possa subsidiar iniciativas semelhantes na computação aplicada à educação.

2 Fundamentação Teórica e Trabalhos Relacionados

As metodologias ativas partem do princípio de que a aprendizagem se fortalece quando o estudante ocupa posição central. Tecnologias digitais, quando integradas a um propósito pedagógico, podem favorecer o engajamento e a mobilização de habilidades cognitivas complexas [Freeman *et al*., 2014; Bond *et al*., 2020]. A IA insere-se nessa fronteira como tecnologia potencialmente transformadora, desde que empregada como meio, e não como fim em si [Popenici and Kerr, 2017].

A geração automática de questões (*Automatic Question Generation*, AQG) constitui um campo de pesquisa na interface entre processamento de linguagem natural e educação. Antes da disseminação dos LLMs, os sistemas produziam sobretudo itens de baixo nível cognitivo, frequentemente limitados ao reconhecimento factual [Kurdi *et al*., 2020]. Os LLMs ampliaram essas possibilidades e passaram a produzir itens contextualizados e de níveis cognitivos mais altos, especialmente quando orientados por estratégias de *prompting* alinhadas à Taxonomia de Bloom revisada [Krathwohl, 2002; Scaria *et al*., 2024]. Estratégias com poucos exemplos (*few-shot*) também têm apresentado resultados promissores na geração de questões educacionais [Lee *et al*., 2024].

No campo da avaliação, a IA pode apoiar processos formativos contínuos e oferecer *feedback* oportuno [Zhai and Nehm, 2023; Dönmez, 2024]. Entretanto, revisões recentes destacam problemas de integridade acadêmica, confiabilidade do conteúdo e necessidade de supervisão humana [Xia *et al*., 2024]. No ensino das profissões da saúde, a IA generativa apresenta potencial para apoiar o raciocínio clínico, mas ainda são escassas as evidências robustas de impacto, o que reforça a necessidade de uma integração pedagógica intencional [Pham *et al*., 2025]. A experiência com o EducAI situa-se nessa convergência entre geração automática de itens, avaliação formativa e mediação docente.

3 A Plataforma EducAI

O EducAI é uma plataforma acadêmica de geração e simulação de questões avaliativas, concebida para apoiar professores, monitores e estudantes na elaboração e na prática de avaliações ancoradas em material didático ou em temas indicados pelo usuário.

No plano da arquitetura, a aplicação reúne, em um único contêiner *Docker*, um *backend* em *Python*, desenvolvido com o *framework FastAPI*, e uma interface de página única (SPA) em *React* com *TypeScript*. A interface e a Interface de Programação de Aplicações (API) são servidas na mesma porta, o que simplifica a hospedagem em nuvem e o escalonamento sob demanda. A geração é delegada a um modelo de IA acessado por API externa, da família *DeepSeek*. Registros de uso, avaliações e auditoria são persistidos em planilhas eletrônicas (*Google Sheets*). A sessão do usuário é controlada por token assinado (JWT), que registra o perfil e a área de conhecimento.

A plataforma organiza-se em três perfis. O estudante pratica em modo simulado a partir de conteúdo próprio, com cronômetro opcional e *feedback* configurável. O monitor prepara questões e conduz sessões ao vivo, com projeção do item e opção de revelar ou ocultar o gabarito. O docente cria avaliações, refina questões por IA ou manualmente, audita inconsistências, exporta documentos editáveis e monta provas; as ferramentas de montagem ficam restritas a quem prepara a avaliação (Figura 1).

O usuário pode fornecer arquivos em PDF, como artigos, slides e capítulos digitalizados, ou indicar apenas um tema. A extração é híbrida: documentos nativos são processados por uma biblioteca de extração estruturada, enquanto slides e digitalizações passam por reconhecimento óptico de caracteres (OCR) e pré-processamento de imagem. Em ambos os

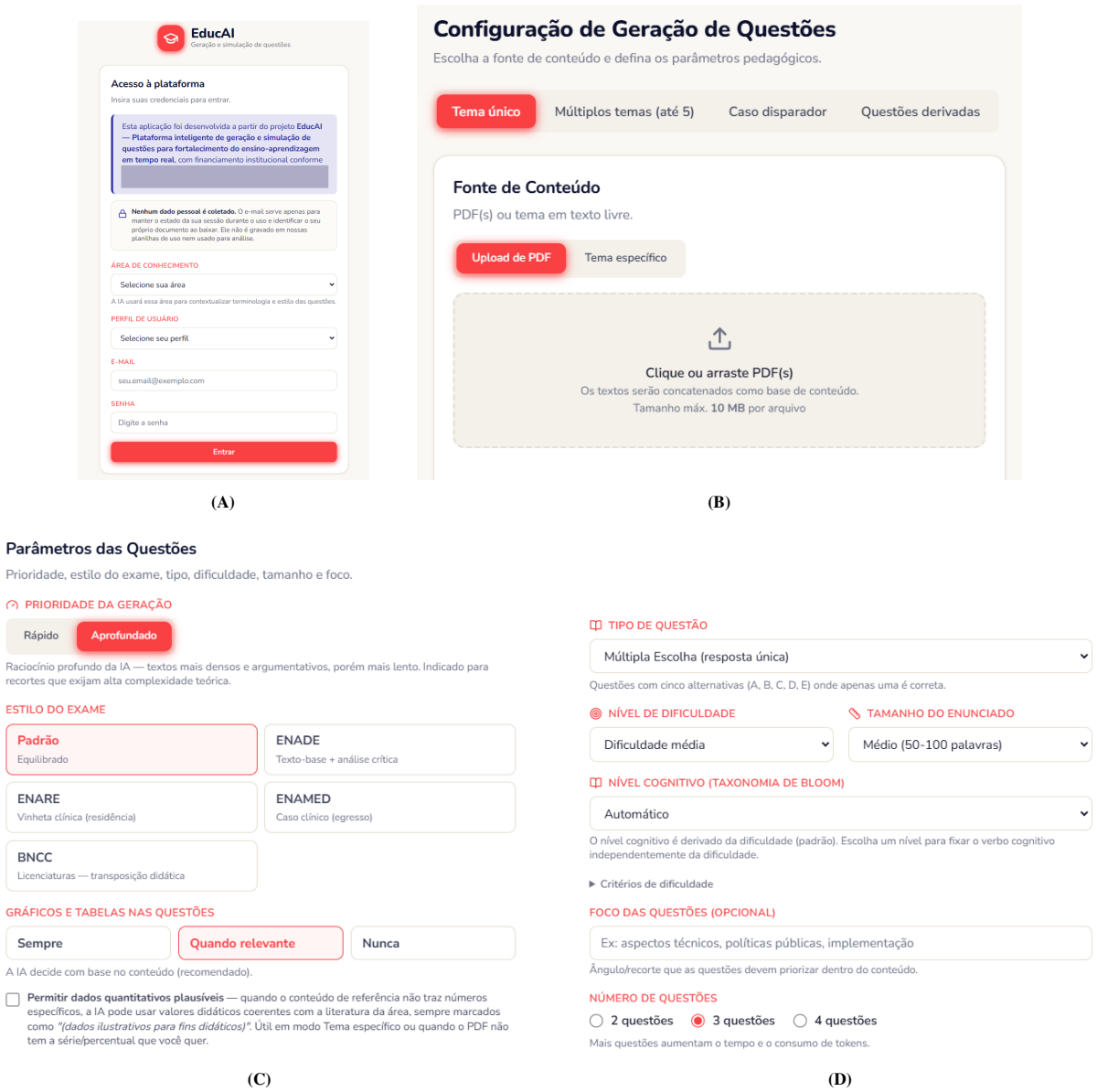


Figura 1. Perfis e configuração do EducAI: (A) tela de acesso, com seleção de área de conhecimento e perfil de usuário; (B) definição do modo de geração e da fonte de conteúdo; (C) parâmetros de geração — prioridade, estilo do exame e uso de gráficos; e (D) definição do tipo, da dificuldade e do nível cognitivo das questões. Fonte: elaboração dos autores (2026).

modos, o prompt orienta o modelo a manter fidelidade ao conteúdo e a não criar fatos, números, autores ou definições ausentes na fonte.

O sistema oferece quatro modos de geração: tema único; múltiplos temas; caso disparador, com cenário-base e questões derivadas; e questões construídas a partir de um item previamente validado. Estão disponíveis questões de múltipla escolha com resposta única, múltipla escolha contextualizada, asserção-razão e dissertativas. O usuário também pode selecionar estilos de exames brasileiros, como ENADE, ENARE e ENAMED. A dificuldade é relacionada à Taxonomia de Bloom revisada, mas o nível cognitivo pode ser definido de forma independente. O sistema ainda produz gráficos, tabelas, *boxplots* e pirâmides etárias, convertidos em imagens nas exportações.

A fidelidade é reforçada por bases de conhecimento estruturadas. Conforme o foco da geração, o sistema acrescenta ao prompt informações de fontes oficiais, como indicadores da Rede Interagencial de Informações para a Saúde (RIPSA) e metas dos Objetivos de Desenvolvimento Sustentável (ODS) [Rede Interagencial de Informações para a Saúde, 2026; Nações Unidas Brasil, 2026]. A recuperação ocorre de forma seletiva, por sobreposição de palavras-chave com limiar mínimo, evitando o acréscimo de contexto sem pertinência temática.

A confiabilidade da saída também exigiu decisões de engenharia de software. Como o modelo devolve texto em formato JSON, um *parser* tolerante recupera a estrutura mesmo diante de saídas malformadas. Mecanismos de autocorreção (*self-healing*) convertem respostas fora do padrão para a estrutura esperada. Um procedimento de balanceamento reduz o viés de posição do gabarito, frequente nos LLMs, e uma persona fixa de elaborador experiente de itens, acompanhada de padrões de qualidade, é utilizada em todas as gerações.

Na aplicação pedagógica, o simulador oferece cronômetro, três níveis de *feedback* e gabarito comentado (Figura 2). A sessão ao vivo projeta o item e mantém, em painel privado,

o gabarito, as explicações e as orientações de estudo. As questões podem ser exportadas em DOCX e HTML.

A plataforma adota medidas de privacidade desde a concepção. As planilhas não armazenam endereços de e-mail em texto claro; o e-mail institucional foi coletado apenas para comunicação e controle da participação; e os registros foram tratados de forma agregada, sem identificação individual. Um Manifesto de Uso Responsável da IA, acessível na interface, explicita que a ferramenta não substitui o docente.

4 Evolução da Engenharia de Prompts

O núcleo intelectual do EducAI é o módulo que constrói os prompts enviados ao modelo. Esse componente foi tratado como artefato de software e evoluiu, de forma iterativa, em resposta às falhas observadas na prática.

Nas primeiras versões, o prompt reunia regras textuais diretas para evitar vícios recorrentes da IA, como alternativas genéricas do tipo "todas as anteriores". A dificuldade era ajustada por instruções básicas, ainda com predomínio de itens de memorização. Também foram observadas questões puramente matemáticas em situações que exigiam análise conceitual. Para corrigir esse desvio, incluiu-se uma regra que subordina a forma da questão ao objetivo pedagógico declarado.

Com a ampliação do uso formativo, o módulo passou a incorporar rubricas analíticas para questões abertas, considerando argumentação, estrutura, fundamentação e aplicação prática. Também foi incluído um mecanismo para adequar o retorno à proficiência informada pelo estudante. O foco, portanto, deslocou-se da simples geração de itens para a produção de *feedback* capaz de apoiar a aprendizagem.

Na etapa seguinte, uma mensagem de sistema estável passou a estabelecer a persona de um elaborador experiente de itens, orientado por princípios de construção de questões (*item-writing*) e pela Taxonomia de Bloom revisada. O nível cognitivo foi separado da dificuldade, e os distratores passaram a ser concebidos como diagnósticos, de modo que cada alternativa incorreta correspondesse a um erro de raciocínio verossímil. Acrescentaram-se padrões para evitar pistas involuntárias, regras para recursos visuais, instruções para efetuar os cálculos quando o foco é quantitativo e as bases de conhecimento descritas na seção anterior.

5 Contexto da Experiência e Sistematização do Relato

A experiência foi desenvolvida no âmbito do Edital de Inovação Pedagógica PROGRAD/PFA/UPE nº 05/2025, do Programa de Fortalecimento Acadêmico da Pró-Reitoria de Graduação da Universidade de Pernambuco, no Campus Petrolina, entre 2025 e 2026. A iniciativa foi articulada aos ODS relativos à educação de qualidade, ao consumo e à produção responsáveis e ao fortalecimento de parcerias [Nações Unidas Brasil, 2026]. A aplicação ocorreu em componentes da graduação em Fisioterapia, com ênfase em Saúde Coletiva: Epidemiologia e Políticas de Saúde, além da monitoria de Epidemiologia.

A seleção dos componentes curriculares decorreu do próprio desenho do edital de inovação pedagógica, que prevê que o docente proponente desenvolva a aplicação nos componentes sob sua responsabilidade, podendo ampliar o escopo a outros componentes cujos docentes manifestem interesse em testar a ferramenta. Não houve, portanto, amostragem probabilística. Os componentes envolvidos pertencem ao curso de Bacharelado em Fisioterapia. A ênfase em Saúde Coletiva — Epidemiologia e Políticas de Saúde — resulta de serem componentes ministrados pelo docente proponente do projeto, condição que permitiu integrar as sessões às atividades regulares sem prejuízo do plano de ensino. Fisioterapia na Atenção Básica e Fisioterapia Musculoesquelética foram incorporados por adesão voluntária de docentes que consentiram em participar: nesses casos, o uso ocorreu sem sessões estruturadas, de modo que não integrou a descrição das aplicações sistematizadas, embora as percepções desses docentes tenham subsidiado o aprimoramento da ferramenta. Docentes de outro curso da área da saúde também utilizaram a plataforma sem vínculo com componente curricular, apenas para conhecê-la e testá-la.

As atividades de 2025 fizeram parte do desenvolvimento da plataforma. O registro das percepções dos participantes ocorreu em 2026, após a aprovação ética. A experiência compreendeu dois períodos: setembro a novembro de 2025, em Epidemiologia, e abril a maio de 2026, em Políticas de Saúde. Foram realizados seis encontros presenciais, três em cada componente curricular, com periodicidade mensal e integrados às atividades regulares. Participaram duas turmas de aproximadamente 20 estudantes cada. Também foram realizadas seis sessões de monitoria de Epidemiologia com os mesmos 20 estudantes dessa turma. Em cada sessão, com duração média de uma hora, eram geradas de três a seis questões.

Quanto aos participantes, não houve seleção individual nem amostragem: as sessões foram conduzidas como atividade coletiva integrada à aula, das quais tomaram parte os estudantes regularmente matriculados, cerca de vinte por turma. As sessões de monitoria foram conduzidas por monitores do próprio componente de Epidemiologia. O registro posterior de percepções foi voluntário e sem contrapartida: quinze estudantes, todos matriculados nos componentes ministrados pelo docente proponente, optaram por registrar suas impressões, assim como oito docentes — cinco do Bacharelado em Fisioterapia e três do Bacharelado em Enfermagem. O critério de inclusão foi ter utilizado a ferramenta durante o semestre letivo; não foram aplicados critérios de exclusão, de modo que foram considerados todos os registros de quem manifestou interesse em opinar sobre a plataforma. Todos os participantes assinaram Termo de Consentimento Livre e Esclarecido.

O módulo interativo era projetado para visualização coletiva. Nas questões de múltipla escolha, os estudantes discutiam as alternativas antes da apresentação do gabarito. Nas questões de cálculo, registravam as respostas em folha própria; nas abertas, debatiam os caminhos possíveis. Após a definição das respostas, o gabarito comentado era exibido e o docente conduzia a discussão.

A sistematização reuniu registros das versões da plataforma, anotações sobre os problemas técnicos e pedagógicos encontrados e a descrição das sessões realizadas. Ao longo desse processo, oito docentes da área da saúde contribuíram



(A) **(B)**

Figura 2. Saídas do EducAI: (A) questão exibida no simulador, com cronômetro; (B) painel com gabarito e justificativas. Fonte: elaboração dos autores (2026).

com percepções exploratórias sobre o percurso de uso, a confiança no conteúdo, a necessidade de edições e eventuais vieses. Quinze estudantes também registraram suas impressões após o uso das questões em debates, monitoria e atividades sobre o Sistema Único de Saúde e as Redes de Atenção à Saúde. Essas percepções subsidiaram os ajustes sucessivos da ferramenta.

Essas contribuições, assim como as observações em sala e os diálogos espontâneos, têm caráter formativo e contextual: orientaram o desenvolvimento da ferramenta até o estágio atual e não constituem investigação sistemática, nem se destinam a estimar eficácia ou a produzir resultados generalizáveis.

Em cumprimento à Resolução nº 466/2012 do Conselho Nacional de Saúde, o projeto foi submetido ao Comitê de Ética em Pesquisa com Seres Humanos do Centro Universitário Integrado de Saúde Amaury de Medeiros (CISAM/UPE) e aprovado em 19 de dezembro de 2025 (CAAE 93622225.9.0000.5191; Parecer nº 8.077.808) [Brasil, Conselho Nacional de Saúde, 2012]. A participação nesta etapa exploratória ocorreu mediante consentimento dos participantes. O projeto de pesquisa dele derivado foi concebido e aprovado pelo Comitê de Ética em Pesquisa; uma coleta de dados abrangente, com instrumentos adequados, constitui etapa posterior, ainda não realizada.

6 Aplicação em Sala e Lições Aprendidas

Nos contextos acompanhados, a aplicação em tempo real mostrou integração mais imediata às aulas e à monitoria. O uso extraclasse ocorreu de forma pontual e variou entre os docentes: alguns se interessaram principalmente pela preparação de avaliações, enquanto outros preferiram explorar a ferramenta fora da sala. Como não houve indicadores sistemáticos de frequência por modalidade, essa diferença deve ser compreendida como aprendizado contextual, e não como comparação de efetividade.

A projeção das questões, seguida de escolha das respostas e debate, converteu a geração automatizada em apoio à mediação pedagógica. As situações produzidas durante a aula permitiram discutir casos não preparados previamente e mobilizar diferentes níveis cognitivos. O valor da atividade, contudo, não decorreu apenas do item gerado, mas da possibilidade de interromper, questionar e contextualizar a resposta com a turma.

O gabarito comentado foi um dos recursos mais valorizados. As justificativas das alternativas ajudaram a esclarecer dúvidas e explicitar equívocos conceituais. Ao mesmo tempo, a mediação do professor permitiu identificar imprecisões, ajustar o vocabulário e discutir por que determinados distratores poderiam induzir interpretações ambíguas.

As percepções reunidas nesta etapa foram coerentes com essa função mediada: a confiança nas questões associou-se à revisão do professor, mais do que ao uso direto da saída automatizada, e os estudantes destacaram os cenários contextualizados, o debate coletivo e as explicações do gabarito. Mais do que avaliar a ferramenta, esses registros ajudaram a identificar o que ajustar e orientaram parte das decisões descritas nas seções anteriores.

Os registros críticos também orientaram mudanças. Alguns distratores foram percebidos como ambíguos; a gradação de dificuldade nem sempre permitiu ao estudante reconhecer seu nível; e um dos cenários sobre Redes de Atenção à Saúde concentrou-se em vocabulário técnico, sem exigir análise suficiente da organização dos serviços. Esses registros reforçaram a necessidade de revisão docente e de testes contínuos dos prompts.

Por definição, este relato não comporta um desenho destinado a medir aprendizagem, desempenho ou mudança na prática docente. As percepções reunidas tiveram finalidade exploratória e informaram o desenvolvimento, sem constituir evidência de eficácia ou de usabilidade validada; tampouco

houve medida sistemática da adesão a cada modalidade. O alcance do artigo restringe-se, portanto, à descrição crítica do processo de concepção, evolução e aplicação da plataforma e aos aprendizados que dele decorreram.

7 Desafios e Soluções

A implementação reuniu desafios relacionados à confiabilidade do conteúdo, à estabilidade da saída, ao desempenho, à recuperação de informações e à adoção pelos usuários.

O risco central de sistemas baseados em LLM é a produção de conteúdo fluente, mas incorreto [Sahoo *et al.*, 2024]. A resposta combinou regras de fidelidade no prompt, ancoragem em fontes oficiais e manutenção do docente como revisor final. Mudanças entre gerações de modelos podem melhorar a adesão às instruções, mas também alteram o comportamento da ferramenta; por isso, a atualização do modelo exige novos testes e não elimina a validação humana.

Os testes iniciais e as avaliações docentes identificaram inconsistências na geração. Foram adotados refinamento iterativo dos prompts, auditoria com registro do estado original e do editado, *parser* tolerante e mecanismos de autocorreção. O viés de posição do gabarito, que concentrava respostas corretas em determinadas letras, foi tratado por balanceamento em lote, preservando o conteúdo e a alternativa correta de cada item.

O tempo de resposta foi um gargalo para o uso em sala. Em algumas configurações, a geração podia levar vários minutos. Duas decisões reduziram essa espera: priorizar respostas mais rápidas, sem acionar por padrão modos ampliados de raciocínio (*thinking*), e paralelizar as chamadas de API em solicitações mais complexas. Em teste interno com três questões, o tempo médio observado passou de aproximadamente 150 para 24 segundos. Trata-se de um indicador técnico de processo, condicionado à configuração utilizada, e não de uma avaliação formal de desempenho.

Parte das falhas inicialmente atribuídas ao LLM decorria da recuperação da informação ou da qualidade dos arquivos de entrada. Foram observadas dificuldades na leitura de um PDF e na extração de dados quantitativos de tabelas. Para enfrentá-las, aprimorou-se o fluxo híbrido de extração, com OCR e pré-processamento de imagens, e implementou-se uma estratégia simplificada de recuperação aumentada por geração (RAG). Documentos extensos são segmentados em trechos (*chunks*), classificados por relevância temática, e apenas os segmentos mais pertinentes compõem o contexto enviado ao modelo.

Quanto à adoção, as preferências dos docentes não foram uniformes. O uso extraclasse mostrou utilidade para exploração e preparação de avaliações, mas sua incorporação à rotina permaneceu ocasional no período acompanhado. O uso em tempo real apresentou encaixe mais direto nas atividades conduzidas pelos autores. As duas modalidades, portanto, devem ser tratadas como complementares, e não como alternativas concorrentes.

Na infraestrutura, a geração é limitada principalmente pela espera de rede. Uma hospedagem econômica, com hibernação sob demanda, atende ao uso acadêmico esporádico. Em situações de maior demanda, como períodos de prova, prevê-se escalonamento temporário de memória e réplicas, seguido de retorno à configuração regular para controle dos custos.

8 Considerações Finais e Perspectivas

A principal contribuição da experiência com o EducAI não está em utilizar inteligência artificial para gerar questões, mas em mostrar como essa geração pode ser integrada à avaliação formativa sem retirar do docente a responsabilidade pelas decisões pedagógicas. A plataforma atuou como suporte dinâmico à discussão, e sua evolução técnica respondeu a necessidades observadas durante o uso.

Nos contextos acompanhados, a aplicação em tempo real apresentou integração mais direta às aulas e à monitoria, enquanto o uso extraclasse dependeu das preferências e das rotinas dos docentes. Essa observação não estabelece superioridade entre modalidades. Indica, antes, que o valor da ferramenta depende de sua adequação à atividade pedagógica e da presença de mecanismos de revisão, contextualização e *feedback*.

Os próximos ciclos concentram-se no amadurecimento da plataforma: o aprimoramento do uso extraclasse, com armazenamento de questões favoritas, simulado programado e painel de desempenho da turma; a expansão para outros campos, incluindo licenciaturas e recursos relacionados à Base Nacional Comum Curricular [Brasil, Ministério da Educação, 2018]; e a produção de documentação que favoreça a transferência tecnológica.

Do ponto de vista computacional, a experiência mostrou a utilidade de tratar a engenharia de prompts como componente de software sujeito a versionamento, testes e aprimoramento contínuo. Essa abordagem não garante, por si só, a qualidade pedagógica, mas torna as decisões mais rastreáveis e facilita a correção de falhas observadas na prática.

Declarações complementares

Agradecimentos

Os autores agradecem à Pró-Reitoria de Graduação da Universidade de Pernambuco (PROGRAD/UPE), pelo apoio concedido por meio do Programa de Fortalecimento Acadêmico (PFA); aos monitores do componente curricular de Epidemiologia, pela condução das sessões de monitoria; e aos docentes e estudantes que participaram das atividades e compartilharam suas percepções sobre a plataforma.

Financiamento

Esta experiência foi desenvolvida com apoio do Edital de Inovação Pedagógica PROGRAD/PFA/UPE nº 05/2025, do Programa de Fortalecimento Acadêmico (PFA) da Pró-Reitoria de Graduação da Universidade de Pernambuco (UPE), que também concedeu bolsa BIA vinculada ao projeto.

Contribuições dos autores

RFG é o principal contribuidor e escritor deste manuscrito e participou de todas as etapas do trabalho, incluindo administração do projeto e supervisão. Todos os autores contribuíram para a conceituação do estudo. RFG foi responsável pelo desenvolvimento do software. RFG, NPDO, EFDS e RCLFP realizaram a investigação. RFG redigiu o rascunho original; RFG, TFAS, EFDS e NPDO participaram da escrita – revisão e edição. Todos os autores leram e aprovaram o manuscrito final.

Conflitos de interesse

Os autores declaram que não há conflitos de interesse.

Disponibilidade de artefatos de pesquisa

Os dados e outros materiais criados e/ou usados neste relato serão disponibilizados mediante solicitação aos autores.

Uso de Inteligência Artificial

As ferramentas ChatGPT (OpenAI; modelos da linha Pro, com raciocínio estendido) e Claude (Anthropic; modelos de nível avançado, com raciocínio estendido), nas versões disponibilizadas entre julho e setembro de 2026, foram utilizadas para apoiar a revisão gramatical e estilística, o aprimoramento da clareza e a organização textual deste manuscrito, bem como a formatação do texto no modelo LaTeX do periódico. Não foram empregadas na coleta ou na análise dos dados. Todas as sugestões incorporadas foram criticamente revisadas e validadas pelos autores, que assumem integral responsabilidade pelo conteúdo final. Não foi atribuída autoria às ferramentas de inteligência artificial. Registra-se, adicionalmente, que a plataforma relatada neste artigo emprega um modelo de linguagem de grande porte da família DeepSeek como componente do objeto de estudo, e não como recurso de escrita do manuscrito.

Outras informações relevantes

Em cumprimento à Resolução nº 466/2012 do Conselho Nacional de Saúde, o projeto foi submetido ao Comitê de Ética em Pesquisa com Seres Humanos do Centro Universitário Integrado de Saúde Amaury de Medeiros (CISAM/UPE) e aprovado em 19 de dezembro de 2025 (CAAE 93622225.9.0000.5191; Parecer nº 8.077.808). A participação de docentes e estudantes ocorreu mediante assinatura de Termo de Consentimento Livre e Esclarecido. Nenhum dado pessoal identificável foi armazenado ou analisado.

Referências

Bond, M., Buntins, K., Bedenlier, S., Zawacki-Richter, O., and Kerres, M. (2020). Mapping research in student engagement and educational technology in higher education: a systematic evidence map. *International Journal of Educational Technology in Higher Education*, 17(1):2. DOI: 10.1186/s41239-019-0176-8.

Brasil, Conselho Nacional de Saúde (2012). Resolução nº 466, de 12 de dezembro de 2012. `https://conselho.saude.gov.br/resolucoes/2012/Reso466.pdf`.

Brasil, Ministério da Educação (2018). Base nacional comum curricular. `https://cdn.mec.gov.br/basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_versaofinal_site.pdf`. Acesso em: 27 jul. 2026.

Dönmez, M. (2024). AI-based feedback tools in education: a comprehensive bibliometric analysis study. *International Journal of Assessment Tools in Education*, 11(4):622–646. DOI: 10.21449/ijate.1467476.

Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., and Wenderoth, M. P. (2014). Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences*, 111(23):8410–8415. DOI: 10.1073/pnas.1319030111.

Krathwohl, D. R. (2002). A revision of Bloom's taxonomy: an overview. *Theory Into Practice*, 41(4):212–218. DOI: 10.1207/s15430421tip4104_2.

Kurdi, G., Leo, J., Parsia, B., Sattler, U., and Al-Emari, S. (2020). A systematic review of automatic question generation for educational purposes. *International Journal of Artificial Intelligence in Education*, 30:121–204. DOI: 10.1007/s40593-019-00186-y.

Lee, U., Jung, H., Jeon, Y., Sohn, Y., Hwang, W., Moon, J., and Kim, H. (2024). Few-shot is enough: exploring ChatGPT prompt engineering method for automatic question generation in English education. *Education and Information Technologies*, 29(9):11483–11515. DOI: 10.1007/s10639-023-12249-8.

Nações Unidas Brasil (2026). Objetivos de desenvolvimento sustentável. `https://brasil.un.org/pt-br/sdgs`. Acesso em: 29 jul. 2026.

Pham, T. D., Karunaratne, N., Exintaris, B., Liu, D., Lay, T., Yuriev, E., and Lim, A. (2025). The impact of generative AI on health professional education: a systematic review in the context of student learning. *Medical Education*, 59(12):1280–1289. DOI: 10.1111/medu.15746.

Popenici, S. A. D. and Kerr, S. (2017). Exploring the impact of artificial intelligence on teaching and learning in higher education. *Research and Practice in Technology Enhanced Learning*, 12(22). DOI: 10.1186/s41039-017-0062-8.

Rede Interagencial de Informações para a Saúde (2026). *Indicadores básicos para a saúde no Brasil: conceitos e aplicações*. Organização Pan-Americana da Saúde, Brasília, DF, 3 edition. Disponível em: `https://www.ripsa.org.br/publicacoes-ripsa/`. Acesso em: 29 jul. 2026.

Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., and Chadha, A. (2024). A systematic survey of prompt engineering in large language models: techniques and applications. *arXiv preprint arXiv:2402.07927*. DOI: 10.48550/arXiv.2402.07927.

Scaria, N., Chenna, S. D., and Subramani, D. (2024). Automated educational question generation at different Bloom's skill levels using large language models: strategies and evaluation. In *Artificial Intelligence in Education (AIED 2024)*, volume 14830 of *Lecture Notes in Computer Science*, pages 165–179. Springer, Cham. DOI: 10.48550/arXiv.2408.04394.

Xia, Q., Weng, X., Ouyang, F., Lin, T. J., and Chiu, T. K. F. (2024). A scoping review on how generative artificial intelligence transforms assessment in higher education. *International Journal of Educational Technology in Higher Education*, 21(1):40. DOI: 10.1186/s41239-024-00468-z.

Zhai, X. and Nehm, R. H. (2023). AI and formative assessment: the train has left the station. *Journal of Research in Science Teaching*, 60(6):1390–1398. DOI: 10.1002/tea.21885.