UNIVERSIDADE FEDERAL DO PAMPA

Ester Oribes Menezes

Orquestração em Tempo Real de Dados Multissensoriais de IoT em Redes de Sensores Sem Fio

Alegrete
Julho de 2026

Ester Oribes Menezes

Orquestração em Tempo Real de Dados Multissensoriais de IoT em Redes de Sensores Sem Fio

Dissertação apresentada ao Programa de Pós-graduação Stricto Sensu em Engenharia Elétrica da Universidade Federal do Pampa, como requisito parcial para obtenção do Título de Mestre em Engenharia Elétrica.

Orientador: Fábio Diniz Rossi

Alegrete
Julho de 2026

Ficha catalográfica elaborada automaticamente com os dados fornecidos
pelo(a) autor(a) através do Módulo de Biblioteca do
Sistema GURI (Gestão Unificada de Recursos Institucionais) .

M534o Menezes, Ester
Orquestração em Tempo Real de Dados Multissensoriais de IoT em Redes de Sensores Sem Fio / Ester Menezes.
59 p.

Dissertação(Mestrado)-- Universidade Federal do Pampa, MESTRADO EM ENGENHARIA ELÉTRICA, 2026.
"Orientação: Fábio Diniz Rossi".

1. Internet das Coisas. 2. Redes de Sensores Sem Fio. 3. Processamento de Dados. 4. Sistema em Tempo Real. 5. Escalonamento de tar. I. Título.

ESTER ORIBES MENEZES

ORQUESTRAÇÃO EM TEMPO REAL DE DADOS MULTISSENSORIAIS DE IOT EM REDES DE SENSORES SEM FIO

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Pampa, como requisito parcial para obtenção do Título de Mestre em Engenharia Elétrica.

Dissertação defendida e aprovada em: 06/07/2026.

Banca examinadora:

_____

Prof. Dr. Fábio Diniz Rossi

Orientador

IFFar

_____

Prof. Dr. Marcelo Caggiani Luizelli

Unipampa

_____

Prof. Drª. Josiane Fontoura dos Anjos

IFFar

Assinado eletronicamente por **MARCELO CAGGIANI LUIZELLI**, **PROFESSOR DO MAGISTERIO SUPERIOR**, em 14/07/2026, às 15:50, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

Assinado eletronicamente por **FABIO DINIZ ROSSI**, **PESSOAL VOLUNTÁRIO**, em 14/07/2026, às 16:00, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

Assinado eletronicamente por **Josiane Fontoura dos Anjos**, **Usuário Externo**, em 14/07/2026, às 16:37, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

A autenticidade deste documento pode ser conferida no site https://sei.unipampa.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **2071572** e o código CRC **7050D03E**.

AGRADECIMENTOS

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Agradeço ao meu orientador, Prof. Dr. Fábio Diniz Rossi, pela orientação, dedicação, disponibilidade e importantes contribuições ao longo do desenvolvimento desta pesquisa. Seus conhecimentos, sugestões e incentivo foram fundamentais para a realização deste trabalho e para o meu crescimento acadêmico e profissional. Expresso minha sincera gratidão pela confiança depositada e pelo apoio durante toda esta trajetória.

RESUMO

Aplicações em tempo real da Internet das Coisas, como monitoramento de saúde e automação industrial, exigem processamento imediato dos dados, tornando inadequadas as filas assíncronas tradicionais do tipo *middleware*. Orquestrar eficientemente sensores heterogêneos — incluindo os tipos contínuos, orientados a eventos e orientados a consultas — é um desafio central devido aos diferentes níveis de prioridade e às limitações de recursos. Este trabalho propõe o RT-IMO, uma estrutura de orquestração multissensorial em tempo real que integra estratégias de escalonamento com base em prioridade, balanceamento de carga e adaptação, visando garantir baixa latência e alocação eficiente de recursos. Os resultados experimentais demonstram que o RT-IMO melhora a capacidade de resposta, a equidade e a eficiência do sistema em comparação com abordagens existentes. Pesquisas futuras explorarão o uso de escalonamento adaptativo baseado em aprendizado de máquina e sua extensão para ambientes heterogêneos de computação na borda. Os resultados mostram que o RT-IMO assegura baixa latência, uso eficiente de recursos e equidade, priorizando dados críticos enquanto descarta seletivamente tarefas de menor prioridade. Ele se adapta dinamicamente às variações da carga de trabalho, superando abordagens estáticas em termos de responsividade e estabilidade sob alta carga.

Palavras-chave: Processamento de dados, Internet das Coisas, Gerenciamento, Aplicações em tempo real, Sensores.

ABSTRACT

Real-time Internet of Things applications, such as healthcare monitoring and industrial automation, require immediate data processing, making traditional asynchronous middleware-like queues unsuitable. Efficiently orchestrating heterogeneous sensors—including continuous, event-driven, and query-driven types—is a key challenge due to varying priority levels and resource constraints. This paper proposes RT-IMO, a real-time multi-sensor orchestration framework that integrates priority-based, load-balanced, and adaptive scheduling strategies to ensure low latency and efficient resource allocation. Experimental results demonstrate that RT-IMO improves responsiveness, fairness, and system efficiency compared to existing approaches. Future research will explore machine learning-based adaptive scheduling and its extension to heterogeneous edge computing environments. The results show that RT-IMO ensures low latency, efficient resource use, and fairness, prioritizing critical data while selectively dropping lower-priority tasks. It adapts dynamically to workload variations, outperforming static approaches in responsiveness and stability under high load.

Keywords: Data processing, Internet of Things, Management, Real-time applications, Sensors.

LISTA DE ILUSTRAÇÕES

Figura 1 – Comparação entre arquiteturas de processamento de dados em IoT. À esquerda, o modelo baseado em *middleware* assíncrono com fila intermediária; à direita, o modelo de processamento direto em tempo real, no qual sensores comunicam-se diretamente com a aplicação. . . . 21
Figura 2 – Arquitetura básica da IoT com camadas de percepção, rede e aplicação. 26
Figura 3 – Comparação entre sensores contínuos, orientados a eventos e orientados a consultas com base em seus comportamentos de transmissão de dados. 28
Figura 4 – Fluxo operacional da heurística RT-IMO. 39
Figura 5 – Experimento de Latência. 47
Figura 6 – Experimento de Uso de Banda. 48
Figura 7 – Experimento de Utilização da CPU. 49
Figura 8 – Experimento de Tamanho da Fila. 49
Figura 9 – Gerenciamento de Prioridades – Parte 1. 50
Figura 10 – Gerenciamento de Prioridades – Parte 2. 50

LISTA DE TABELAS

Tabela 1 – Análise comparativa entre os trabalhos relacionados e o RT-IMO. . . . 33
Tabela 2 – Comparação entre o RT-IMO e as estratégias de escalonamento existentes. 51

SUMÁRIO

1 INTRODUÇÃO 19
1.1 Problema de Pesquisa 21
1.2 Objetivos 22

2 REFERENCIAL TEÓRICO E TRABALHOS RELACIONADOS . . 25
2.1 Referencial Teórico 25
2.1.1 Internet das Coisas 25
2.1.1.1 Tipos de Sensores 26
2.1.2 Processamento de Dados IoT em Tempo Real 28
2.2 Trabalhos Relacionados 30

3 ORQUESTRAÇÃO DE MULTISENSORES EM TEMPO REAL PARA IOT 35
3.1 Formulação do Problema 35
3.1.1 Restrições do Sistema 36
3.1.2 Prioridade Adaptativa 37
3.1.3 Função de Custo 37
3.2 RT-IMO: Heurística de Orquestração Multissensorial 37
3.2.1 Função de Prioridade Adaptativa 38
3.2.2 Algoritmo de Escalonamento 38
3.2.3 Complexidade Computacional 40
3.3 Limitações e Desafios 40

4 RESULTADOS E DISCUSSÃO 43
4.1 Metodologia 43
4.1.1 Cenários de Uso do RT-IMO em Sistemas IoT 45
4.1.1.1 Cenário de Monitoramento de Saúde em Tempo Real 45
4.1.1.2 Cenário de Infraestrutura Crítica e Redes Elétricas Inteligentes 46
4.1.1.3 Impacto nas Métricas de Desempenho 46
4.2 Resultados 47

5 CONCLUSÕES E TRABALHOS FUTUROS 53

REFERÊNCIAS 55

1 INTRODUÇÃO

A Internet das Coisas (IoT) [1] emergiu como um paradigma transformador na computação, permitindo a interconectividade contínua entre inúmeros dispositivos, sensores e sistemas. Ambientes de IoT facilitam a coleta, o processamento e a comunicação de dados em diversos domínios, incluindo cidades inteligentes, monitoramento de saúde, automação industrial e monitoramento ambiental. Esses sistemas dependem de sensores embutidos no ambiente para monitorar continuamente as condições, coletar dados e transmitir informações para infraestruturas de computação centralizadas ou distribuídas, visando análise em tempo real e tomada de decisão.

A rápida expansão da IoT tem sido impulsionada por avanços em tecnologias de comunicação sem fio, computação em nuvem e computação na borda, permitindo que aplicações em tempo real e quase tempo real processem grandes volumes de dados de forma eficiente. No entanto, embora arquiteturas gerais de IoT frequentemente se baseiem em modelos centrados na nuvem com mecanismos de *buffer*, aplicações de IoT em tempo real impõem restrições mais rigorosas que exigem arquiteturas especializadas para garantir baixa latência, alta disponibilidade e resposta imediata [2].

Um subconjunto significativo das aplicações de IoT opera em ambientes de tempo real, onde o processamento instantâneo de dados e a tomada de decisões são fundamentais [3]. Essas aplicações abrangem domínios críticos, como sistemas de monitoramento de saúde, veículos autônomos, controle de processos industriais e gestão de redes elétricas inteligentes, nos quais atrasos no processamento de dados podem causar consequências graves [4]. Diferentemente das arquiteturas tradicionais de IoT que utilizam *middlewares* assíncronos, como filas de mensagens ou *buffers* baseados em eventos, as aplicações em tempo real geralmente não podem tolerar os atrasos inerentes a esses mecanismos.

Em ambientes de tempo real, o processamento deve ocorrer assim que os dados são gerados, garantindo que nenhuma informação crítica seja atrasada ou perdida devido a ineficiências no escalonamento [5]. Essa necessidade impõe requisitos rigorosos à infraestrutura computacional subjacente, exigindo fluxos de dados diretos, modelos de processamento síncronos e mecanismos de escalonamento determinísticos para garantir um comportamento previsível do sistema [6].

Sistemas de IoT dependem de diversos tipos de sensores, cada um caracterizado por comportamentos distintos de aquisição e transmissão de dados [7]. De modo geral, esses sensores se enquadram em três categorias principais: contínuos, orientados a eventos e orientados a consultas [8]. Sensores contínuos geram e transmitem dados em intervalos fixos, fornecendo informações constantes, como monitores de ECG na área da saúde ou sensores de temperatura em ambientes industriais. Sensores orientados a eventos, por outro lado, permanecem inativos até que uma condição específica ou anomalia seja detectada. Nesse momento, transmitem dados imediatamente — por exemplo, sensores de queda em cuidados com idosos ou sistemas de alarme de incêndio em edifícios inteligentes.

Nesses casos, esses eventos precisam ser priorizados em relação aos fluxos contínuos de dados, pois atrasos no seu processamento podem comprometer decisões críticas do sistema. Por fim, sensores orientados a consultas operam sob demanda, recuperando dados apenas quando solicitados explicitamente por um sistema ou usuário externo, como aparelhos de ressonância magnética na área médica ou sensores de nutrientes do solo na agricultura de precisão. Gerenciar e orquestrar esses fluxos heterogêneos de dados dentro da mesma infraestrutura em tempo real representa um desafio significativo, pois cada tipo de sensor possui exigências distintas quanto ao tempo, à alocação de banda e à priorização computacional [9].

O problema central abordado neste trabalho é a orquestração eficiente de fluxos de dados oriundos de sensores mistos dentro de uma infraestrutura unificada de IoT em tempo real. Dada a natureza diversa dos comportamentos dos sensores, abordagens de escalonamento ingênuas, como o processamento por ordem de chegada *(FIFO)*, falham em considerar diferenças de criticidade, prioridade e temporalidade entre as fontes de dados.

Para lidar de forma eficaz com esse desafio, foi desenvolvido o RT-IMO (Orquestração Multissensorial em Tempo Real para IoT), uma estrutura de escalonamento adaptativo que otimiza o processamento de dados em tempo real em ambientes de IoT heterogêneos. O RT-IMO foi desenvolvido com o objetivo de garantir baixa latência e resposta imediata para dados críticos, ao mesmo tempo em que gerencia de forma eficiente fluxos contínuos e consultas sob demanda.

Diferentemente de estratégias tradicionais de orquestração, o RT-IMO ajusta dinamicamente as prioridades dos sensores com base na urgência dos prazos, nas taxas de geração de dados e no desempenho recente de latência, garantindo que dados críticos orientados a eventos sejam processados sem atraso, ao mesmo tempo que mantém a equidade entre fontes contínuas e orientadas a consultas. O RT-IMO melhora a responsividade e a eficiência de recursos ao integrar escalonamento baseado em prioridade para alertas sensíveis ao tempo, mecanismos de balanceamento de carga para evitar a escassez de recursos e ajustes adaptativos em tempo real conforme a carga do sistema.

Essa abordagem permite que infraestruturas de IoT mantenham um desempenho estável sob variações de carga de trabalho, reduzindo a latência e o *overhead* computacional. Avaliar o RT-IMO em comparação com estratégias de escalonamento existentes oferece *insights* sobre sua eficácia no equilíbrio entre minimização da latência, otimização de recursos e equidade do sistema, tornando-o uma solução viável para aplicações de IoT em tempo real de missão crítica.

Sistemas de IoT em tempo real operam em ambientes altamente dinâmicos, nos quais o volume de dados de sensores pode sobrecarregar redes e recursos computacionais, tornando essencial a orquestração para lidar com a heterogeneidade dos sistemas e garantir eficiência e confiabilidade no processamento dos dados.

1.1 PROBLEMA DE PESQUISA

A Figura 1 ilustra duas arquiteturas contrastantes para o processamento de dados em ambientes de IoT. No modelo tradicional baseado em *middlewares* assíncronos, os sensores transmitem dados para uma fila intermediária, a partir da qual a aplicação consome as mensagens de forma desacoplada. Esse modelo facilita a escalabilidade e a tolerância a falhas, porém introduz latência adicional, imprevisibilidade temporal e perda de informações críticas devido a congestionamento e políticas de descarte. Em contraste, o modelo de processamento direto em tempo real estabelece comunicação síncrona entre sensores e aplicação, eliminando filas intermediárias.

Embora essa abordagem reduza significativamente a latência, ela impõe desafios substanciais, uma vez que fluxos de dados com diferentes comportamentos de geração, requisitos temporais e níveis de criticidade coexistem em uma infraestrutura unificada de IoT em tempo real e precisam ser processados de forma simultânea. Nessas condições, diferentes fluxos de dados compartilham os recursos computacionais disponíveis, exigindo que suas diferentes características, requisitos temporais e níveis de criticidade sejam considerados durante a orquestração. Isso inclui a necessidade de escalonamento determinístico, controle rigoroso de recursos computacionais e priorização dinâmica de fluxos heterogêneos. Nesse cenário, sensores contínuos, orientados a eventos e orientados a consultas competem diretamente por recursos limitados, tornando a orquestração adaptativa essencial para garantir previsibilidade temporal, eficiência computacional e preservação de eventos críticos. O presente trabalho aborda esse segundo modelo, propondo uma estratégia de orquestração em tempo real capaz de lidar com a heterogeneidade e criticidade dos fluxos sensoriais sem depender de *middlewares* assíncronos.

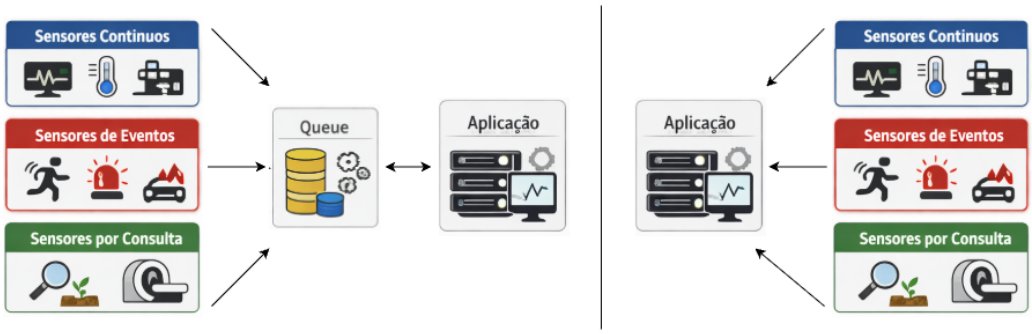

Figura 1 – Comparação entre arquiteturas de processamento de dados em IoT. À esquerda, o modelo baseado em *middleware* assíncrono com fila intermediária; à direita, o modelo de processamento direto em tempo real, no qual sensores comunicam-se diretamente com a aplicação.

Em ambientes de IoT em tempo real nos quais diferentes tipos de sensores operam simultaneamente em uma mesma infraestrutura, são gerados fluxos contínuos de dados e eventos esporádicos que precisam ser processados de forma imediata. Por exemplo, em

um sistema de monitoramento de saúde, sensores contínuos podem transmitir sinais vitais do paciente de forma periódica, enquanto sensores orientados a eventos disparam alertas críticos apenas quando uma condição específica é detectada, como a ocorrência de uma queda ou uma alteração abrupta nos batimentos cardíacos. Esses alertas orientados a eventos representam situações de maior criticidade e exigem que o sistema priorize seu processamento em relação aos dados contínuos. Nesses cenários, eventos críticos não devem competir em igualdade com fluxos rotineiros de dados, pois atrasos no seu processamento podem comprometer decisões importantes e afetar diretamente a segurança do paciente.

No entanto, em infraestruturas tradicionais de IoT, estratégias simples de escalonamento, como o processamento por ordem de chegada *(FIFO)* ou o uso de *middlewares* assíncronos, tratam todos os dados de forma semelhante, desconsiderando diferenças de prioridade e criticidade. Como consequência, dados urgentes podem ser processados tardiamente em meio a grandes volumes de informações contínuas, resultando em aumento de latência, congestionamento de recursos e degradação do desempenho do sistema.

O problema central abordado neste trabalho é a orquestração eficiente de fluxos de dados oriundos de sensores heterogêneos em uma infraestrutura unificada de IoT em tempo real. Dada a natureza diversa dos comportamentos dos sensores, abordagens de escalonamento ingênuas, como o processamento por ordem de chegada (*FIFO*), falham em considerar diferenças de criticidade, prioridade e requisitos temporais entre as fontes de dados, resultando em aumento de latência, ineficiência no uso de recursos e degradação do desempenho do sistema.

1.2 OBJETIVOS

Com o objetivo de enfrentar esse desafio, este trabalho propõe o *Real-Time IoT Multisensor Orchestration* (RT-IMO), uma estrutura adaptativa de escalonamento projetada para ambientes de IoT heterogêneos. O RT-IMO visa garantir baixa latência e resposta imediata para dados críticos, ao mesmo tempo em que gerencia eficientemente fluxos contínuos e consultas sob demanda por meio de priorização dinâmica, balanceamento de carga e controle adaptativo de recursos. Diferentemente de estratégias tradicionais de orquestração, o RT-IMO ajusta dinamicamente as prioridades dos sensores com base na urgência dos prazos, nas taxas de geração de dados e no comportamento recente de latência do sistema. Essa abordagem integra escalonamento baseado em prioridade, mecanismos de balanceamento de carga e ajustes adaptativos em tempo real conforme a carga do sistema, promovendo equidade entre sensores contínuos, orientados a eventos e orientados a consultas.

Os objetivos específicos deste trabalho são:

1. Formular o problema de orquestração de sensores heterogêneos em ambientes de IoT em tempo real.

2. Projetar uma estratégia adaptativa de escalonamento capaz de minimizar latência e otimizar o uso de recursos computacionais e de rede.

3. Avaliar o desempenho do RT-IMO em comparação com estratégias tradicionais de escalonamento, considerando métricas como latência, uso de banda, utilização de CPU, tamanho das filas e taxa de eventos críticos processados.

Os resultados experimentais demonstram que o RT-IMO supera abordagens convencionais, alcançando menor latência média, maior eficiência no uso de recursos e maior equidade entre sensores, além de preservar eventos críticos sob condições de sobrecarga. Esses resultados indicam que o RT-IMO é uma solução promissora para aplicações de IoT em tempo real de missão crítica, como monitoramento de saúde, automação industrial e sistemas de resposta a emergências. Sistemas de IoT em tempo real operam em ambientes altamente dinâmicos, nos quais o volume de dados sensoriais pode sobrecarregar redes e recursos computacionais, tornando a orquestração adaptativa um componente fundamental para garantir eficiência, previsibilidade temporal e confiabilidade no processamento dos dados.

A estrutura desta dissertação está organizada da seguinte forma: O Capítulo 2 apresenta o referencial teórico, discutindo os fundamentos da Internet das Coisas, os diferentes tipos de sensores e os desafios associados ao processamento de dados em tempo real, além de uma revisão crítica dos principais trabalhos relacionados. O Capítulo 3 introduz a proposta de orquestração multissensorial em tempo real, incluindo a formulação do problema, o modelo conceitual e matemático, bem como a descrição detalhada da heurística RT-IMO. No Capítulo 4 são apresentados os resultados experimentais e a discussão, comparando o desempenho do RT-IMO com estratégias de escalonamento tradicionais em termos de latência, uso de largura de banda, utilização de CPU e tamanho de fila. Por fim, o Capítulo 5 conclui o trabalho, destacando as principais contribuições, limitações e perspectivas para pesquisas futuras em orquestração adaptativa de sensores em ambientes IoT em tempo real.

2 REFERENCIAL TEÓRICO E TRABALHOS RELACIONADOS

Este capítulo apresenta o referencial teórico que fundamenta esta pesquisa, contextualizando os principais conceitos, modelos e abordagens relacionados à orquestração de dados sensoriais, sistemas em tempo real e escalonamento em ambientes de IoT. Além disso, são discutidos trabalhos relacionados da literatura, destacando avanços recentes, limitações existentes e lacunas de pesquisa que motivam a proposta deste trabalho. Essa fundamentação teórica e empírica fornece o suporte conceitual necessário para a definição do modelo proposto e para a análise crítica das soluções existentes.

2.1 REFERENCIAL TEÓRICO

Esta seção apresenta os conceitos fundamentais necessários para compreender os desafios da orquestração de dados em tempo real na IoT. Inicia-se com uma visão geral da IoT, discutindo sua arquitetura básica e principais componentes. Em seguida, são classificados os tipos de sensores, diferenciando sensores contínuos, orientados a eventos e orientados a consultas com base em seus comportamentos de aquisição e transmissão de dados. A discussão então se volta para o processamento de dados em tempo real na IoT, destacando as restrições que impedem o uso de *middlewares* assíncronos tradicionais, como filas de mensagens. Além disso, esta seção introduz os desafios de escalonamento associados à integração de sensores heterogêneos dentro de uma infraestrutura em tempo real, preparando o terreno para a formulação do problema e a solução proposta nas seções seguintes.

2.1.1 INTERNET DAS COISAS

A Internet das Coisas (IoT) refere-se a uma vasta rede de dispositivos, sensores e sistemas interconectados que coletam, transmitem e processam dados sem intervenção humana direta [10]. A IoT possibilita a comunicação fluida entre sistemas físicos e digitais, facilitando a automação, o monitoramento em tempo real e a tomada de decisões baseada em dados. A proliferação de sensores de baixo custo, tecnologias de comunicação sem fio e computação em nuvem acelerou a adoção da IoT em diversos domínios, incluindo cidades inteligentes, automação industrial, saúde e monitoramento ambiental.

Uma arquitetura típica de IoT consiste em três camadas interconectadas que trabalham em conjunto para permitir a coleta, transmissão e processamento contínuos de dados [11]. A *Camada de Percepção* é responsável por interagir com o ambiente físico, composta por diversos sensores e atuadores que monitoram continuamente e coletam dados como temperatura, movimento e umidade. Uma vez coletados, esses dados são transmitidos pela *Camada de Rede*, que assegura a comunicação eficiente entre os dispositivos IoT e os sistemas de processamento centrais. Essa camada utiliza protocolos como MQTT, CoAP ou 5G para facilitar a troca de dados de forma confiável e escalável. Por fim, a *Camada*

de Aplicação processa os dados recebidos, extraindo *insights* significativos e fornecendo serviços aos usuários finais. Essa camada geralmente utiliza computação em nuvem ou na borda para realizar análises em tempo real, automatizar decisões e disponibilizar uma interface de interação com o ecossistema IoT. A Figura 2 ilustra a arquitetura básica da IoT, mostrando a interação entre sensores, protocolos de comunicação e unidades de processamento.

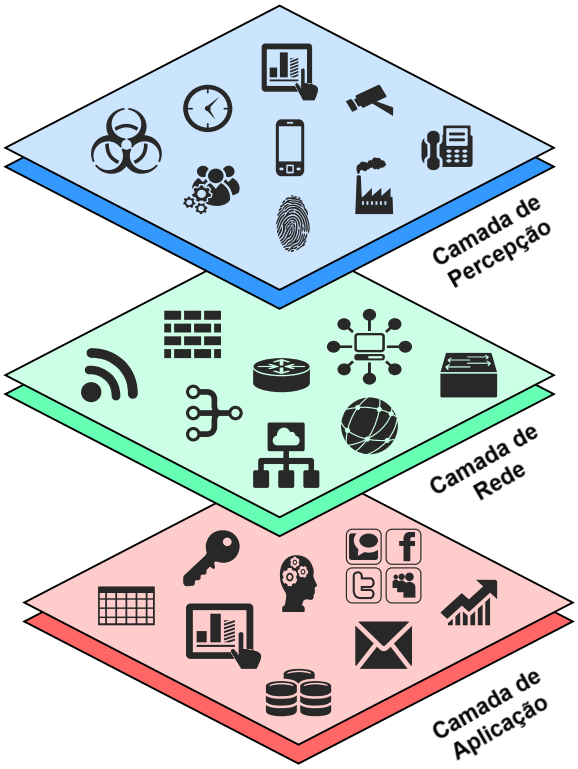

Figura 2 – Arquitetura básica da IoT com camadas de percepção, rede e aplicação.

2.1.1.1 TIPOS DE SENSORES

Sistemas de IoT dependem de uma variedade de sensores que capturam dados ambientais e sistêmicos [12]. Com base em seus comportamentos de aquisição e transmissão de dados, os sensores em ambientes IoT podem ser classificados em três categorias principais [13]:

- **Sensores Contínuos:** Esses sensores geram e transmitem dados em intervalos fixos, garantindo um fluxo constante de informações em tempo real. São particularmente úteis em aplicações que exigem monitoramento contínuo para detectar mudanças graduais, tendências ou anomalias. No entanto, podem impor demandas elevadas de largura de banda e processamento devido ao fluxo persistente de dados. Exemplos incluem:

  - *Monitores de ECG* na saúde, que acompanham continuamente a frequência cardíaca e detectam irregularidades, como arritmias. São cruciais em unidades de terapia intensiva (UTIs) e em sistemas de monitoramento remoto de pacientes.
  - *Sensores de Temperatura* em aplicações industriais, garantindo condições seguras para equipamentos sensíveis. Por exemplo, a regulação precisa da temperatura é necessária em usinas de energia e na fabricação de semicondutores.
  - *Sensores de Qualidade do Ar* em sistemas de monitoramento ambiental, medindo níveis de poluentes como CO_2, PM2.5 e compostos orgânicos voláteis (VOCs) em tempo real.
  - *Sensores de Vibração* na manutenção preditiva de máquinas industriais, monitorando continuamente vibrações estruturais para detectar desgastes ou desalinhamentos.

- **Sensores Orientados a Eventos:** Permanecem inativos até que uma condição predefinida ou anomalia seja detectada, momento em que transmitem dados imediatamente. Esse tipo de sensor ajuda a reduzir o consumo de energia e o uso de banda, ativando-se apenas quando necessário. São especialmente úteis em aplicações que exigem alertas em tempo real. Exemplos incluem:
  - *Sensores de Queda* em sistemas de cuidado com idosos, usando acelerômetros e giroscópios para detectar quedas e notificar cuidadores ou serviços de emergência.
  - *Alarmes de Incêndio* em edifícios inteligentes, que disparam ao detectar fumaça, calor ou níveis elevados de monóxido de carbono.
  - *Sensores de Intrusão* em sistemas de segurança, que emitem alertas quando há movimento não autorizado em áreas restritas.
  - *Sensores de Vazamento de Água* em residências ou indústrias, detectando vazamentos em tubulações para prevenir danos à infraestrutura.
  - *Sensores Sísmicos* em sistemas de detecção de terremotos, que captam tremores do solo e enviam alertas em tempo real.

- **Sensores Orientados a Consultas:** Não transmitem dados continuamente, mas apenas quando solicitados por um sistema ou usuário. São empregados em aplicações onde a coleta frequente de dados não é necessária, otimizando o consumo de energia e a largura de banda. Exemplos incluem:
  - *Aparelhos de Ressonância Magnética (MRI)* na medicina, que geram dados apenas durante a realização do exame.
  - *Sensores de Umidade do Solo* na agricultura de precisão, ativados por sistemas automatizados para determinar o momento ideal de irrigação.

– *Sensores de Monitoramento Estrutural* na engenharia civil, que fornecem dados sobre pontes ou túneis quando solicitados por engenheiros.

– *Sensores de Inventário Baseados em RFID* em armazéns, que transmitem dados apenas quando lidos por um leitor RFID.

– *Sensores de Medição de Energia* em redes inteligentes, que fornecem dados de consumo em horários agendados ou sob demanda.

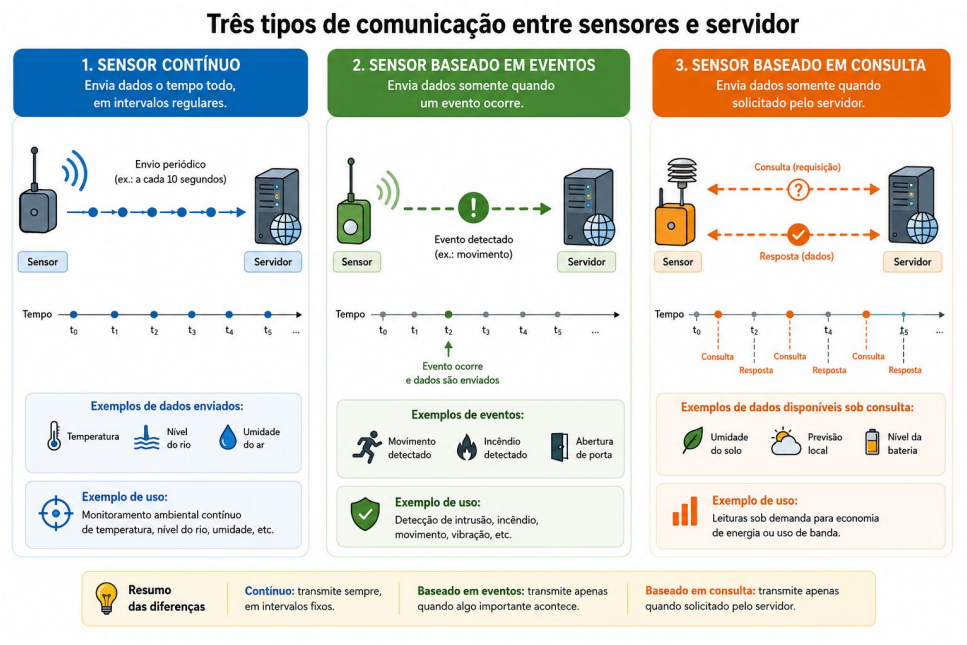

Figura 3 – Comparação entre sensores contínuos, orientados a eventos e orientados a consultas com base em seus comportamentos de transmissão de dados.

A Figura 3 fornece uma visão comparativa desses tipos de sensores, ilustrando seus padrões de transmissão de dados. O sensor contínuo, representado por uma linha azul sólida, transmite dados em intervalos regulares, proporcionando monitoramento contínuo do ambiente, porém com maior consumo de energia e largura de banda. O sensor orientado a eventos, indicado por uma linha verde tracejada, envia dados apenas quando eventos específicos ocorrem, mantendo a responsividade com maior eficiência energética. O sensor orientado a consultas, ilustrado por uma linha vermelha pontilhada, transmite dados somente sob solicitação, economizando energia, mas podendo introduzir atrasos. A visualização destaca os *trade-offs* entre disponibilidade de dados, consumo energético e eficiência da comunicação em diferentes designs de redes sensoriais.

2.1.2 PROCESSAMENTO DE DADOS IOT EM TEMPO REAL

Diferentemente das arquiteturas tradicionais de IoT que dependem de *middlewares* assíncronos como filas de mensagens ou brokers orientados a eventos, aplicações de IoT

em tempo real exigem o processamento imediato dos dados dos sensores para garantir decisões oportunas [6]. Aplicações como veículos autônomos, sistemas de controle industrial e monitoramento de saúde não podem tolerar os atrasos introduzidos por mecanismos de *buffer*, pois esses atrasos podem causar ineficiências operacionais ou até falhas catastróficas [14]. No modelo assíncrono, sensores enviam dados para um endpoint, que os coloca em uma fila de mensagens. Essa fila funciona como *buffer*, permitindo que as aplicações processem os dados no seu próprio ritmo — ideal para cenários em que a resposta imediata não é crítica, mas confiabilidade e escalabilidade são. Já o modelo em tempo real conecta diretamente os sensores ao *endpoint*, que atua como balanceador de carga, distribuindo os dados em tempo real às aplicações. Essa arquitetura assegura baixa latência e processamento imediato, fundamental para aplicações que exigem decisões instantâneas. No entanto, ela exige maior poder computacional e estabilidade da rede para evitar perdas ou atrasos.

O processamento de dados em tempo real na IoT apresenta diversos desafios críticos que devem ser superados para garantir desempenho eficiente e confiável do sistema. Um dos principais desafios é o processamento de baixa latência, pois os dados dos sensores devem ser tratados imediatamente após sua geração, com sobrecarga computacional mínima. Além disso, muitas aplicações de IoT operam em ambientes com recursos limitados, especialmente em dispositivos de borda com restrições de processamento e memória, exigindo alocação eficiente de recursos. Outro aspecto essencial é a priorização de dados críticos — alertas orientados a eventos, como emergências médicas, devem ser processados antes de fluxos contínuos rotineiros para evitar atrasos fatais. Finalmente, à medida que os ecossistemas IoT se expandem, a escalabilidade se torna crucial: o sistema deve ser capaz de lidar com um número crescente de sensores heterogêneos sem degradação de desempenho [3].

Uma limitação importante dos sistemas IoT em tempo real é a impossibilidade de depender de mecanismos tradicionais de processamento assíncrono, como filas de mensagens ou *frameworks* distribuídos orientados a eventos. Esses mecanismos introduzem latência devido ao *buffer* e aos atrasos de escalonamento, tornando-os inadequados para aplicações sensíveis ao tempo. Em seu lugar, são necessárias estratégias de escalonamento determinístico para garantir tempos de resposta previsíveis [15].

Diversas abordagens de escalonamento têm sido propostas para lidar com dados sensoriais heterogêneos em tempo real, cada uma oferecendo vantagens e desvantagens. O escalonamento baseado em prioridade garante que dados de sensores orientados a eventos sejam processados antes de dados contínuos e orientados a consultas, permitindo que alertas críticos recebam atenção imediata. Embora eficaz na redução da latência de eventos de alta prioridade, essa abordagem pode levar à inanição de fluxos de dados de menor prioridade [16]. Por outro lado, o escalonamento com balanceamento de carga busca distribuir os recursos computacionais de forma equitativa entre os diferentes tipos de sensores, evitando que uma categoria monopolize o processamento. Essa estratégia melhora a estabilidade do sistema, embora possa não garantir a resposta mais rápida para alertas críticos [17].

Uma abordagem mais dinâmica é o escalonamento adaptativo, que ajusta continuamente as prioridades com base na carga de trabalho em tempo real. Monitorando a carga do sistema e a disponibilidade de recursos, o escalonamento adaptativo garante estabilidade ao realocar o poder de processamento conforme necessário, sendo particularmente eficaz em ambientes com cargas flutuantes e atividade sensorial imprevisível [18]. Uma estratégia de escalonamento adequada otimiza o desempenho da IoT em tempo real ao equilibrar latência, equidade e eficiência computacional.

Este trabalho não tem como objetivo comparar diferentes plataformas de *middleware*. Os mecanismos apresentados nesta seção têm a finalidade de contextualizar as diferenças entre arquiteturas assíncronas e arquiteturas de processamento direto em tempo real, cenário no qual a proposta RT-IMO está inserida. Isso ocorre porque o RT-IMO é uma heurística de orquestração para o processamento de dados em tempo real, e não como uma plataforma de *middleware*. Assim, a avaliação concentra-se na comparação entre as abordagens *(FIFO)*, escalonamento baseado em prioridade, balanceamento de carga e escalonamento dinâmico, por representarem estratégias de escalonamento descritas na literatura para a alocação de recursos e o processamento de tarefas em sistemas de tempo real [16, 17, 18], permitindo analisar o impacto da política de escalonamento sobre as métricas de desempenho avaliadas neste trabalho.

2.2 TRABALHOS RELACIONADOS

O campo do escalonamento em tempo real para IoT tem sido amplamente explorado, com diversas abordagens propostas para lidar com os desafios de redução de latência, eficiência no uso de recursos e escalonamento adaptativo para sensores heterogêneos. As estratégias existentes diferem em seus focos — seja na execução baseada em prioridades, na eficiência energética ou na tomada de decisão em tempo real. No entanto, nenhuma fornece uma estrutura de orquestração que equilibre latência, utilização de recursos e equidade como o RT-IMO proposto.

Uma das contribuições pioneiras no escalonamento em tempo real para IoT é o *Just-in-Time Scheduling (JiTS)*, introduzido por Liu et al. [19]. O JiTS foi desenvolvido para otimizar a disseminação de dados sensoriais ao alinhar os horários de transmissão às demandas da aplicação, garantindo entrega de dados com baixa latência e minimizando o consumo de energia. Essa abordagem é eficiente em ambientes sincronizados, onde os sensores operam em intervalos pré-definidos. No entanto, apesar de sua eficiência em redes estruturadas, o JiTS apresenta dificuldades em redes heterogêneas, pois não incorpora mecanismos para adaptação dinâmica a diferentes tipos de sensores e variações de carga. Essa limitação torna-se evidente em implantações escaláveis de IoT, onde sensores contínuos, orientados a eventos e orientados a consultas coexistem, exigindo uma abordagem mais flexível e adaptativa.

No contexto do monitoramento de saúde em tempo real, Nam et al. [20] propuseram

um sistema ubíquo para assistência médica domiciliar, integrando escalonamento de sensoriamento e comunicação para facilitar o monitoramento contínuo de pacientes. Seu sistema sincroniza a coleta e a transmissão de dados, garantindo a atualização em tempo real de parâmetros de saúde — crucial para aplicações médicas críticas, como detecção de quedas e monitoramento cardíaco. Apesar de destacar a importância da priorização de sensores, a solução é voltada para ambientes locais e não se mostra escalável para implantações de IoT em larga escala com infraestruturas sensoriais heterogêneas. Além disso, o sistema depende de políticas de escalonamento fixas, sendo pouco responsivo a variações dinâmicas de carga.

Uma perspectiva diferente é oferecida por Ahmad et al. [21], que desenvolveram uma ferramenta de verificação formal para escalonamento de tarefas em tempo real em ambientes de IoT. Seu trabalho enfatiza a importância da confiabilidade e da correção nas decisões de escalonamento, especialmente em aplicações críticas onde erros podem levar a falhas no sistema ou riscos à segurança. A abordagem assegura que restrições pré-definidas de escalonamento sejam cumpridas por meio de modelos rigorosos de verificação. No entanto, embora fortaleça as garantias teóricas, carece de validação prática em ambientes dinâmicos, onde a variação de carga exige soluções adaptativas.

Uma abordagem alternativa foca no escalonamento sensível ao contexto, no qual as condições ambientais e os requisitos específicos da aplicação influenciam a execução das tarefas sensoriais. Hassani et al. [20] investigaram a execução eficiente de consultas complexas em aplicações de IoT em tempo real, propondo um modelo de execução otimizado que reduz a sobrecarga de processamento e acelera a recuperação de dados críticos. Seu trabalho é aplicável a sistemas inteligentes, como monitoramento urbano e automação predial. No entanto, o estudo concentra-se na otimização de consultas em banco de dados e não aborda diretamente o escalonamento de sensores heterogêneos em ambientes com restrições computacionais.

Diversos estudos também examinaram políticas tradicionais de escalonamento em tempo real, principalmente em sistemas de automação industrial. Cayssials et al. [21] analisaram o escalonamento por taxa monotônica *(rate monotonic)*, muito utilizado em tarefas de controle em tempo real. A pesquisa visa minimizar o número de níveis de prioridade mantendo execução determinística, sendo aplicável a robótica e manufatura automatizada. Entretanto, sua aplicabilidade a sistemas baseados em sensores é limitada devido à rigidez na atribuição de prioridades, sem considerar as variações dinâmicas nas taxas de geração de dados.

Outra técnica é o escalonamento com restrições por janela de tempo, proposta por West e Poellabauer [22], que gerencia fluxos de pacotes sensíveis ao tempo e de melhor esforço, buscando melhorar o desempenho da rede em ambientes IoT. O modelo equilibra necessidades de pacotes críticos e não críticos, otimizando o uso da largura de banda. Contudo, essa abordagem é predominantemente voltada para o nível de rede e não

incorpora priorização de tarefas sensoriais, sendo menos adequada para ambientes com restrições computacionais.

Além do escalonamento em nível de sistema, algumas pesquisas focam na otimização da comunicação entre sensores. He et al. [23] desenvolveram o protocolo *SPEED*, que prioriza transmissões determinísticas entre sensores para reduzir atrasos de rede. Embora eficaz na redução de atrasos, opera exclusivamente na camada de rede e não aborda o escalonamento de tarefas no nível de CPU, essencial para o processamento eficiente em dispositivos IoT com recursos limitados.

A eficiência energética também tem sido foco central. Saifullah et al. [24] propuseram um esquema de escalonamento paralelo sensível à energia para dispositivos IoT alimentados por bateria. O método otimiza o consumo de CPU mantendo a execução em tempo real, sendo útil em dispositivos vestíveis e sensores remotos. Entretanto, não incorpora escalonamento adaptativo para sensores heterogêneos, limitando sua aplicabilidade em implantações complexas.

Mais recentemente, a integração entre aprendizado de máquina e escalonamento adaptativo tem atraído atenção. Bukhari et al. [25] introduziram o *OpenSense*, uma estrutura baseada em aprendizado incremental para o escalonamento em tempo real de sensores em dispositivos embarcados. A abordagem ajusta as prioridades com base no aprendizado contínuo, sendo eficaz em ambientes imprevisíveis. Contudo, o modelo exige recursos computacionais elevados, dificultando sua aplicação em infraestruturas IoT leves e com restrições energéticas.

Liu et al. [26] propuseram um modelo baseado em escopo espaço-temporal para orquestração de sensores. Esse modelo quantifica o valor dos dados sensoriais ao longo do tempo e do espaço, permitindo alocação inteligente de recursos com base na relevância contextual. Apesar de otimizar a utilização de rede e processamento, requer análise prévia e metadados, o que pode introduzir sobrecarga computacional significativa para dispositivos de baixa potência.

Aswin et al. [27] propuseram o framework SLICED, uma arquitetura híbrida que integra IoT, borda e nuvem para reduzir latência e melhorar a segurança na transmissão de dados. A abordagem realiza pré-processamento na borda e utiliza comunicação segura para otimizar o desempenho. No entanto, não incorpora mecanismos de escalonamento adaptativo nem priorização entre diferentes tipos de sensores, limitando sua aplicação em cenários heterogêneos.

Em arquiteturas distribuídas, Khairnar [28] propõe o EdgeFlow, baseado em computação em borda para processamento distribuído em ambientes IoT heterogêneos. A abordagem utiliza balanceamento de carga e alocação adaptativa de recursos para reduzir latência e consumo de banda, apresentando melhorias em cenários de alta demanda. No entanto, embora opere em ambientes heterogêneos, não incorpora priorização entre sensores, limitando-se à distribuição de processamento e não à orquestração em tempo

real, lacuna tratada pelo RT-IMO.

Embora cada trabalho apresente contribuições relevantes, nenhum oferece uma estrutura holística que integre, de forma conjunta, priorização adaptativa, balanceamento de carga e alocação de recursos em tempo real em ambientes com sensores heterogêneos. A Tabela 1 apresenta uma análise comparativa entre essas abordagens e o RT-IMO, evidenciando limitações em adaptabilidade, eficiência energética e suporte à heterogeneidade.

Autor	**Abordagem de Escalonamento**	**Adaptabilidade**	**Eficiência Energética**	**Suporte a Sensores Heterogêneos**
Liu et al. [19]	Just-in-Time Scheduling (JiTS)	Limitada	Moderada	Limitada
Nam et al. [20]	Sistema de Assistência Médica com Escalonamento	Moderada	Moderada	Moderada
Ahmad et al. [21]	Verificação Formal de Escalonamento em Tempo Real	Limitada	Não abordada	Limitada
Hassani et al. [20]	Execução de Consultas Sensíveis ao Contexto	Moderada	Não abordada	Moderada
Cayssials et al. [21]	Rate Monotonic Scheduling	Limitada	Não abordada	Limitada
West e Poellabauer [22]	Escalonamento com Janela de Tempo	Moderada	Não abordada	Moderada
He et al. [23]	Protocolo SPEED para Comunicação em Tempo Real	Alta	Moderada	Moderada
Saifullah et al. [24]	Escalonamento Paralelo com Consumo Energético Otimizado	Moderada	Alta	Limitada
Bukhari et al. [25]	OpenSense: Escalonamento com Aprendizado de Máquina	Moderada	Alta	Alta
Liu et al. [26]	Modelo Espaço-Temporal para Orquestração de Sensores	Alta	Alta	Moderada
Aswin et al. [27]	Framework SLICED baseado em Edge-Cloud com segurança e processamento adaptativo	Moderada	Moderada	Moderada
Khairnar. [28]	EdgeFlow: processamento distribuído com alocação adaptativa e ML no edge	Alta	Alta	Moderada
RT-IMO	**Orquestração Adaptativa em Tempo Real Multissensor**	**Alta**	**Alta**	**Alta**

Tabela 1 – Análise comparativa entre os trabalhos relacionados e o RT-IMO.

A estrutura RT-IMO integra adaptabilidade em tempo real, escalonamento sensível ao consumo de energia e priorização de sensores heterogêneos em uma estratégia unificada de orquestração. Ao empregar ajuste dinâmico de prioridades, balanceamento de carga e alocação de recursos ciente da CPU/memória, o RT-IMO supera as limitações observadas nos trabalhos anteriores.

3 ORQUESTRAÇÃO DE MULTISENSORES EM TEMPO REAL PARA IOT

Este capítulo apresenta a abordagem de orquestração multissensorial em tempo real adotada neste trabalho. Inicialmente, estabelece-se a formulação formal do problema, explicitando os elementos do sistema, as restrições operacionais e os critérios de otimização que caracterizam o cenário de IoT heterogêneo considerado. Em seguida, descreve-se o RT-IMO como uma heurística adaptativa de escalonamento capaz de tomar decisões sob limitações dinâmicas de recursos, priorizando dados críticos e preservando a equidade entre diferentes tipos de sensores. Por fim, discutem-se limitações e desafios práticos associados a implantações em larga escala, conectando a formulação teórica às condições reais de rede e processamento.

3.1 FORMULAÇÃO DO PROBLEMA

Considera-se um sistema de IoT composto por sensores heterogêneos implantados em um ambiente de tempo real. O modelo a seguir descreve formalmente os componentes do sistema, os recursos disponíveis e as restrições operacionais, servindo como base para a definição do problema de escalonamento multissensorial.

Seja $S = \{S_1, S_2, \ldots, S_N\}$ o conjunto de N sensores presentes no sistema. Esses sensores são classificados de acordo com seu comportamento de geração de dados em três categorias principais:

- T_c: sensores contínuos, que geram dados periodicamente;
- T_e: sensores orientados a eventos, que transmitem dados apenas quando uma condição crítica é detectada;
- T_q: sensores orientados a consultas, que respondem sob demanda.

Cada sensor S_i gera um conjunto de pacotes de dados $P_i = \{p_{i,1}, p_{i,2}, \ldots, p_{i,M}\}$ ao longo do tempo, com características dependentes do tipo de sensor e da aplicação.

Cada sensor S_i é caracterizado pelos seguintes parâmetros:

- λ_i: taxa de geração de dados (pacotes por segundo);
- D_i: tamanho do pacote de dados (em bytes);
- δ_i: prazo máximo permitido para o processamento de cada pacote;
- $P_i(t)$: prioridade atribuída ao sensor no instante de tempo t;
- C_i: custo computacional para processar um pacote;
- B_i: largura de banda requerida pelo sensor.

Além disso, o sistema dispõe de recursos globais limitados:

- R: capacidade total de processamento (CPU e memória);
- B: largura de banda total disponível na rede;
- Q: capacidade total do buffer para armazenamento temporário de pacotes.

O objetivo do problema é definir uma função de escalonamento ótima $\Phi : P_i \times T \times R \times B \to \mathbb{R}^+$ que aloque os recursos disponíveis aos pacotes sensoriais, minimizando latência, evitando congestionamento, garantindo equidade entre sensores e adaptando-se dinamicamente às variações da carga de trabalho.

3.1.1 RESTRIÇÕES DO SISTEMA

Em sistemas de tempo real, cada pacote de dados deve ser processado antes do vencimento de seu prazo. Formalmente, impõe-se a restrição de latência:

$$t_{proc}(p_{i,j}) - t_{arrival}(p_{i,j}) \leq \delta_i, \quad \forall i, j, \tag{3.1}$$

onde $t_{arrival}(p_{i,j})$ representa o instante de chegada do pacote ao sistema e $t_{proc}(p_{i,j})$ o instante de conclusão do processamento. Essa restrição garante previsibilidade temporal, fundamental para aplicações críticas.

A carga computacional total não pode exceder os recursos disponíveis. Seja $x_i \in \{0, 1\}$ uma variável indicadora que assume valor 1 quando o sensor S_i é escalonado para processamento. A restrição de recursos computacionais é dada por:

$$\sum_{i=1}^{N} C_i \cdot x_i \leq R. \tag{3.2}$$

De forma análoga, o consumo total de largura de banda não deve ultrapassar a capacidade da rede:

$$\sum_{i=1}^{N} B_i \cdot x_i \leq B. \tag{3.3}$$

Para evitar perda de dados devido a congestionamento, a ocupação total do buffer não pode exceder sua capacidade:

$$\sum_{i=1}^{N} P_i(t) \leq Q, \tag{3.4}$$

onde $P_i(t)$ representa o número de pacotes pendentes do sensor S_i no instante t.

Além das restrições de desempenho, é desejável garantir equidade na alocação de recursos entre sensores heterogêneos. Para isso, define-se o índice de equidade de Jain:

$$F(t) = \frac{\left(\sum_{i=1}^{N} P_i(t) \cdot x_i\right)^2}{N \sum_{i=1}^{N} \left(P_i(t) \cdot x_i\right)^2}, \tag{3.5}$$

onde valores próximos de 1 indicam uma alocação justa de recursos entre sensores.

3.1.2 PRIORIDADE ADAPTATIVA

Dada a criticidade variável dos dados sensoriais, define-se uma função de prioridade adaptativa que combina requisitos temporais, taxa de geração de dados e histórico de latência:

$$P_i(t) = \alpha \cdot \frac{1}{\delta_i} + \beta \cdot \frac{\lambda_i}{\sum_{j=1}^{N} \lambda_j} + \gamma \cdot \left(1 - \frac{L_i}{\delta_i}\right), \tag{3.6}$$

onde α, β e γ são fatores de ponderação e L_i representa o atraso previamente experimentado pelo sensor S_i. Essa função permite priorizar sensores críticos, especialmente aqueles orientados a eventos.

3.1.3 FUNÇÃO DE CUSTO

Finalmente, o problema de escalonamento é formulado como a minimização de uma função de custo global:

$$J = \sum_{i=1}^{N} \left(w_1 \cdot (t_{proc}(p_{i,j}) - t_{arrival}(p_{i,j})) + w_2 \cdot x_i + w_3 \cdot (1 - F(t)) + w_4 \cdot \frac{\sum_{i=1}^{N} B_i \cdot x_i}{B} \right), \tag{3.7}$$

onde w_1, w_2, w_3 e w_4 são pesos que controlam o compromisso entre latência, eficiência computacional, equidade e controle de congestionamento de rede.

A formulação acima define um problema de otimização ideal. No entanto, a obtenção da solução ótima em tempo real é computacionalmente intratável em implantações IoT em larga escala, uma vez que o problema pode ser mapeado para escalonamento multi-recurso com restrições, conhecido por ser NP-difícil. Assim, este trabalho propõe uma heurística adaptativa de escalonamento, denominada RT-IMO, que aproxima a solução ótima mantendo baixa sobrecarga computacional.

3.2 RT-IMO: HEURÍSTICA DE ORQUESTRAÇÃO MULTISSENSORIAL

Com base na formulação apresentada na seção anterior, propõe-se o RT-IMO como uma heurística adaptativa para o problema de escalonamento multissensorial em tempo real. Em vez de buscar a solução ótima do problema definido, o RT-IMO fornece uma aproximação eficiente que atende às restrições de latência, recursos computacionais, largura de banda e buffer, mantendo baixa sobrecarga computacional, requisito fundamental em ambientes IoT.

O RT-IMO foi concebido para cenários nos quais *middlewares* assíncronos tradicionais, como filas de mensagens, são inadequados devido aos requisitos de latência estrita. O mecanismo combina alocação adaptativa de recursos e gerenciamento dinâmico de prioridades para suportar fluxos heterogêneos de dados provenientes de sensores contínuos, orientados a eventos e orientados a consultas.

Diferentemente de abordagens clássicas de escalonamento, como *FIFO* e políticas estáticas baseadas em prioridade, o RT-IMO toma decisões considerando restrições dinâmicas do sistema e condições variáveis da rede. Em particular, o mecanismo incorpora:

- **Priorização adaptativa**, garantindo atendimento imediato a dados críticos orientados a eventos;
- **Balanceamento de carga entre sensores heterogêneos**, evitando monopolização de recursos computacionais e de comunicação;
- **Consciência de congestionamento de rede**, ajustando taxas de transmissão e adiando ou descartando dados não críticos;
- **Gerenciamento adaptativo de buffers**, prevenindo saturação de filas e perda de pacotes críticos.

O RT-IMO foi projetado para satisfazer as restrições definidas na formulação do problema, incluindo limites de latência (Eq. 3.1), recursos computacionais (Eq. 3.2), largura de banda (Eq. 3.3), capacidade de buffer (Eq. 3.4) e equidade entre sensores (Eq. 3.5). Ao respeitar essas restrições, o mecanismo busca uma aproximação prática da solução ótima do problema de escalonamento.

3.2.1 FUNÇÃO DE PRIORIDADE ADAPTATIVA

Para orientar as decisões de escalonamento, o RT-IMO utiliza a função de prioridade adaptativa definida na Eq. 3.6, que combina requisitos temporais, taxa de geração de dados e histórico de latência. Essa função permite aumentar a prioridade de sensores com prazos curtos, altas taxas de geração ou latência acumulada, favorecendo dados críticos.

Além disso, o RT-IMO incorpora consciência de congestionamento por meio de uma função penalizadora que reduz a prioridade de sensores com alto consumo de largura de banda:

$$P'_i(t) = P_i(t) - w_4 \cdot \frac{B_i}{B}, \tag{3.8}$$

onde B_i representa a largura de banda requerida pelo sensor S_i, B a largura de banda total disponível e w_4 o fator de penalização. Essa extensão evita que sensores com tráfego intensivo monopolizem os recursos de rede.

3.2.2 ALGORITMO DE ESCALONAMENTO

O processo de escalonamento do RT-IMO é descrito no Algoritmo 1. Em cada ciclo de execução, as prioridades dos sensores são recalculadas dinamicamente, os sensores são ordenados de acordo com a prioridade consciente da rede e as decisões de escalonamento são tomadas considerando simultaneamente restrições computacionais, de comunicação e de buffer.

Algorithm 1 Algoritmo de Escalonamento Adaptativo RT-IMO

1: **Entrada:** Conjunto de sensores S, recursos computacionais R, largura de banda B, buffer Q
2: **Saída:** Escalonamento aproximado para processamento IoT em tempo real
3: Inicialize as prioridades $P_i(0)$
4: **while** O sistema estiver ativo **do**
5: Calcule $P_i(t)$ usando Eq. 3.6
6: Calcule $P'_i(t)$ usando Eq. 3.8
7: Ordene sensores por $P'_i(t)$ em ordem decrescente
8: **for** cada sensor S_i **do**
9: **if** $C_i \leq R$ **and** $B_i \leq B$ **and** $\sum P_i(t) \leq Q$ **then**
10: Escalone S_i para processamento
11: Atualize $R \leftarrow R - C_i$ e $B \leftarrow B - B_i$
12: **else**
13: **if** $B_i > B$ **then**
14: Reduza a taxa de transmissão de S_i
15: **else if** $\sum P_i(t) > Q$ **then**
16: Descarte o pacote de menor prioridade
17: **end if**
18: **end if**
19: **end for**
20: **end while**

Após a apresentação do Algoritmo 1, a Figura 4 ilustra o fluxo operacional da heurística RT-IMO, destacando as principais etapas executadas em cada ciclo de escalonamento, desde o recálculo das prioridades até o processamento dos sensores ou o tratamento de situações de sobrecarga.

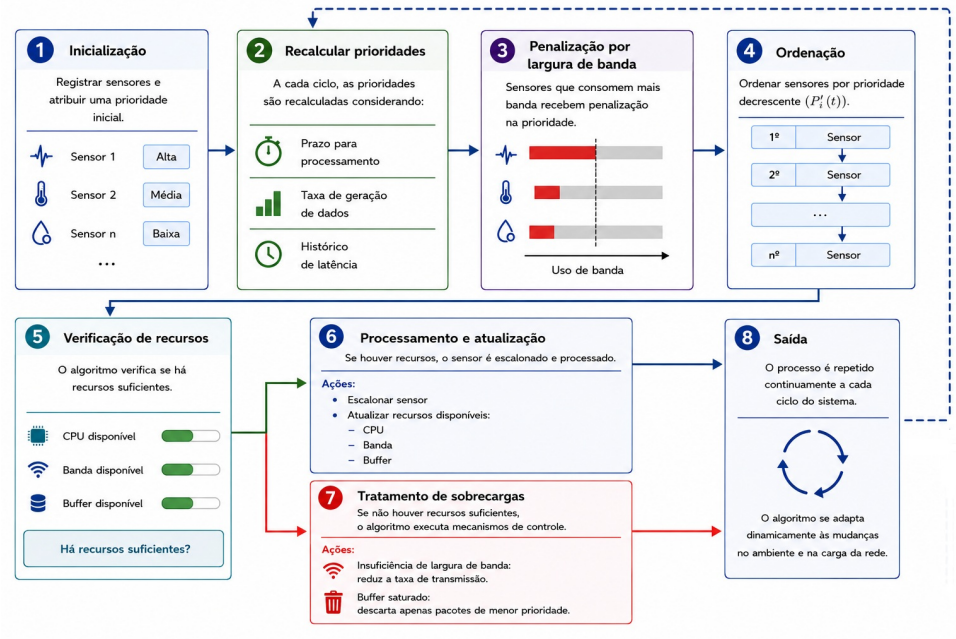

Figura 4 – Fluxo operacional da heurística RT-IMO.

A Figura 4 apresenta o fluxo operacional da heurística RT-IMO. Inicialmente, são atribuídas as prioridades aos sensores. Durante cada ciclo de execução, essas prioridades são recalculadas, é aplicada a penalização associada ao consumo de largura de banda e os sensores são ordenados de acordo com a prioridade resultante. Em seguida, o algoritmo verifica a disponibilidade de recursos computacionais, largura de banda e capacidade do *buffer* antes de decidir pelo escalonamento do sensor. Quando os recursos são insuficientes, mecanismos de tratamento de sobrecarga são acionados para reduzir a taxa de transmissão ou descartar apenas pacotes de menor prioridade.

A cada ciclo, o RT-IMO adapta as prioridades com base no estado atual do sistema, garantindo baixa latência para sensores orientados a eventos e evitando inanição de sensores contínuos e orientados a consultas. Em situações de sobrecarga, o mecanismo de gerenciamento de filas controla adiamento e descarte seletivo de pacotes, preservando dados críticos. O descarte seletivo é realizado apenas quando a capacidade do *buffer* é excedida. Nesses casos, manter todos os pacotes no *buffer* aumentaria o tempo de espera para processamento e poderia comprometer os requisitos de baixa latência das aplicações de IoT em tempo real. Assim, o RT-IMO descarta apenas os pacotes de menor prioridade, preservando o processamento dos dados críticos e evitando que a sobrecarga comprometa o desempenho global do sistema.

3.2.3 COMPLEXIDADE COMPUTACIONAL

O processo de decisão do RT-IMO foi projetado para manter baixa sobrecarga computacional. Em cada ciclo, o cálculo de prioridades e a verificação de restrições apresentam complexidade linear, enquanto a ordenação dos sensores domina o custo total, resultando em complexidade assintótica $O(N \log N)$, onde N é o número de sensores ativos. Essa complexidade é compatível com ambientes IoT em tempo real e não introduz atrasos perceptíveis mesmo em dispositivos com recursos limitados.

3.3 LIMITAÇÕES E DESAFIOS

Apesar da eficácia do mecanismo proposto, diversas limitações práticas devem ser consideradas em implantações reais. Em cenários de larga escala, o volume massivo de dados gerado por sensores pode sobrecarregar nós de processamento, saturar enlaces de comunicação e introduzir atrasos em filas, comprometendo garantias de latência. A escalabilidade do escalonador torna-se crítica à medida que o número de sensores cresce, aumentando a competição por recursos.

Ambientes IoT sem fio também apresentam variações imprevisíveis de latência e perdas de pacotes, causadas por interferências, mobilidade e limitações físicas do meio. Essas condições podem violar as hipóteses do modelo, exigindo mecanismos adaptativos conscientes do estado da rede.

Embora o RT-IMO incorpore priorização dinâmica, controle de congestionamento e gerenciamento de filas, sua aplicabilidade prática depende de fatores externos, como restrições energéticas de sensores, falhas de enlace e variabilidade abrupta de carga. Assim, as seções seguintes avaliam o desempenho do RT-IMO sob diferentes níveis de carga e discutem o impacto dessas limitações nas garantias de latência, equidade e eficiência em cenários IoT em larga escala.

4 RESULTADOS E DISCUSSÃO

Este capítulo descreve o ambiente experimental, as características das estratégias de escalonamento analisadas, as métricas de desempenho utilizadas para comparação e o procedimento adotado para garantir uma análise rigorosa e reprodutível. Ao avaliar o RT-IMO, buscamos compreender seus pontos fortes, limitações e adequação a aplicações de IoT em tempo real. A avaliação considera indicadores-chave de desempenho como taxa de transmissão, uso de banda, latência, utilização de recursos e taxa de eventos processados, oferecendo uma perspectiva abrangente sobre como cada abordagem gerencia dados de sensores heterogêneos. Além disso, os resultados obtidos nos experimentos são analisados e discutimos o impacto das decisões de escalonamento na eficiência, responsividade e escalabilidade do sistema em diferentes cenários de carga de trabalho da IoT.

4.1 METODOLOGIA

Para avaliar a eficácia de diferentes estratégias de orquestração no escalonamento de sensores IoT em tempo real, conduzimos uma avaliação utilizando um ambiente de simulação controlado com o simulador SimPy[1]. O objetivo é comparar o desempenho do RT-IMO com quatro estratégias de escalonamento — *FIFO*, baseada em prioridade, balanceamento de carga e dinâmica — sob condições idênticas, medindo seu impacto em métricas como taxa de transmissão, uso de banda, latência, uso de recursos e taxa de eventos processados. Esta seção detalha a configuração experimental, explica as características de cada estratégia, define as métricas de avaliação e descreve o procedimento para obter resultados significativos.

A seção a seguir apresenta avaliações experimentais para medir a eficácia do RT-IMO em diversos cenários de carga. A comparação com outras estratégias — *FIFO*, Prioridade, Balanceamento de Carga e Dinâmica — revela *insights* importantes sobre seu comportamento nas diferentes métricas. O experimento envolve uma rede heterogênea de sensores com variação entre 50 e 500 sensores, incluindo sensores contínuos, orientados a eventos e orientados a consultas, com taxas de geração de dados ajustadas conforme a classificação. O sistema opera em uma arquitetura híbrida nuvem-borda, na qual nós de borda lidam com tarefas sensíveis à latência e a nuvem agrega e armazena dados de menor prioridade. A avaliação compara cinco estratégias de escalonamento e mede seus impactos em latência, uso de banda, utilização de CPU e taxa de eventos processados. A infraestrutura computacional conta com uma máquina de 4 núcleos e 16GB de RAM, com restrição de banda de 100 Mbps, simulando limitações realistas de IoT. O número de sensores ativos cresce progressivamente de 50 a 500, avaliando escalabilidade e situações de estresse, e é introduzido um congestionamento artificial de rede em alguns cenários. O RT-IMO prioriza dinamicamente os dados dos sensores com um mecanismo adaptativo que

[1] https://simpy.readthedocs.io/en/latest/

ajusta as prioridades com base na urgência dos prazos, variações de carga e restrições do sistema. Os resultados demonstram que o RT-IMO equilibra redução de latência, equidade e uso eficiente de recursos, superando os métodos tradicionais ao garantir que dados críticos sejam processados sem atrasos excessivos, mantendo a estabilidade do sistema sob diferentes cargas.

Todas as estratégias são testadas sob condições iniciais idênticas para garantir uma comparação justa. A carga de trabalho inclui um número pré-definido de sensores gerando dados em taxas específicas, com recursos computacionais e de rede mantidos constantes ao longo da avaliação. Diferentes intensidades de carga são simuladas com o aumento gradual do número de sensores ativos e a variação na frequência de ativação de sensores orientados a eventos. Além disso, o congestionamento artificial de rede é introduzida para analisar o desempenho das estratégias sob condições degradadas de largura de banda. As quatro estratégias representam diferentes paradigmas de escalonamento de dados sensoriais em ambientes em tempo real. Cada uma gerencia a forma como os dados são processados e priorizados, influenciando a eficiência, responsividade e equidade do sistema.

Os experimentos foram conduzidos em uma simulação determinística, na qual o gerador de números aleatórios utiliza uma semente fixa (seed = 1). Nessas condições, execuções realizadas com a mesma topologia, conjunto de sensores e parâmetros operacionais produzem resultados idênticos, fazendo com que a repetição dos experimentos não introduza variação estatística nos resultados obtidos. Assim, métricas como média e desvio-padrão permanecem invariáveis, caracterizando uma limitação inerente ao modelo de simulação, mas garantindo reprodutibilidade e consistência na análise comparativa das estratégias avaliadas.

A primeira estratégia avaliada é *FIFO (First-In-First-Out)*, um método básico que processa os dados exatamente na ordem em que chegam. *FIFO* garante tratamento igualitário para todos os dados, sem qualquer mecanismo de prioridade. Isso resulta em escalonamento previsível, porém ineficiente, especialmente quando dados urgentes precisam aguardar na fila atrás de dados contínuos. Essa abordagem é comum em sistemas simples de registro de dados, mas em aplicações IoT em tempo real, tende a gerar alta latência para dados críticos.

A segunda estratégia é o escalonamento baseado em prioridade, que dá maior prioridade a sensores orientados a eventos em detrimento dos contínuos e orientados a consultas. Essa abordagem garante o processamento imediato de eventos críticos como alertas médicos ou falhas industriais. No entanto, pode causar inanição de sensores de menor prioridade em cenários com alta carga, sendo problemática para aplicações que exigem tratamento equilibrado dos dados.

A terceira estratégia é o balanceamento de carga, que distribui os recursos igualmente entre os diferentes tipos de sensores, evitando que uma categoria monopolize os recursos. Embora garanta justiça e previna a inanição, essa abordagem não diferencia

dados urgentes e não urgentes, podendo falhar em atender requisitos estritos de tempo real.

A quarta abordagem, o escalonamento dinâmico, ajusta o processamento com base na carga do sistema em tempo real. Essa abordagem monitora o uso de CPU e memória e ignora dados não críticos quando os recursos se tornam escassos. Apesar de prevenir atrasos em cenários de estresse, pode resultar na perda ou atraso indefinido de dados de baixa prioridade.

Por fim, a estratégia RT-IMO propõe uma orquestração adaptativa que equilibra dinamicamente CPU, memória e prioridades dos sensores para manter a estabilidade do sistema e otimizar a responsividade. Diferente de métodos estáticos, o RT-IMO ajusta constantemente as prioridades com base no tipo de sensor, urgência dos prazos, desempenho recente e disponibilidade de recursos. Ele previne a inanição de sensores não prioritários mesmo sob carga elevada.

Cinco métricas-chave são utilizadas para avaliar o desempenho:

- **Taxa de transmissão:** número de mensagens processadas por segundo. Quanto maior, mais eficiente o escalonamento.
- **Uso de banda:** quantidade média de dados transmitidos por unidade de tempo. Menor uso ajuda a prevenir congestionamentos.
- **Latência:** tempo entre geração e processamento dos dados.
- **Uso de CPU e memória:** avalia a sobrecarga computacional de cada estratégia.
- **Taxa de eventos:** número de eventos críticos processados por segundo. Essencial para aplicações de missão crítica.

4.1.1 CENÁRIOS DE USO DO RT-IMO EM SISTEMAS IOT

Para compreender como o RT-IMO melhora as métricas avaliadas, é importante descrever de forma operacional como o algoritmo, embora simulado, atua em cenários reais ao longo do tempo. Em cada ciclo de escalonamento, o RT-IMO executa uma sequência de decisões adaptativas, que são ilustradas nos cenários a seguir.

4.1.1.1 CENÁRIO DE MONITORAMENTO DE SAÚDE EM TEMPO REAL

Em um ambiente de monitoramento de saúde, o sistema recebe continuamente dados de sensores vitais, como frequência cardíaca e saturação de oxigênio, ao mesmo tempo em que sensores orientados a eventos podem gerar alertas críticos, como quedas ou arritmias. Sensores orientados a consultas são ativados sob demanda por profissionais de saúde.

Primeiramente, o RT-IMO coleta as informações de estado do sistema, incluindo taxas de geração de dados, latência observada, consumo de largura de banda e ocupação das filas. Em seguida, o algoritmo calcula a prioridade adaptativa de cada sensor, atribuindo valores mais elevados a sensores orientados a eventos com prazos rígidos e latência acumulada.

Após o cálculo das prioridades, o RT-IMO ordena os sensores e inicia o escalonamento, processando primeiro alertas críticos. Caso a rede esteja congestionada, o mecanismo de penalização reduz a prioridade de fluxos contínuos menos críticos, liberando largura de banda para eventos de emergência. Paralelamente, o gerenciamento de filas evita que dados contínuos ocupem todo o buffer, descartando ou adiando pacotes de menor prioridade.

Esse processo garante que alertas críticos sejam processados com latência mínima, enquanto dados contínuos são processados de forma equilibrada, evitando inanição e degradação do sistema. Assim, o RT-IMO contribui para respostas médicas mais rápidas e confiáveis, reduzindo o risco de atrasos que poderiam comprometer a segurança do paciente.

4.1.1.2 CENÁRIO DE INFRAESTRUTURA CRÍTICA E REDES ELÉTRICAS INTELIGENTES

Em redes elétricas inteligentes e sistemas industriais, sensores contínuos monitoram parâmetros operacionais, sensores orientados a eventos detectam falhas ou sobrecargas, e sensores orientados a consultas são utilizados para diagnósticos sob demanda.

Nesse cenário, o RT-IMO inicialmente avalia o estado da rede e dos recursos computacionais, identificando situações de pico de carga ou congestionamento. Quando um evento crítico de falha é detectado, o algoritmo recalcula as prioridades e promove automaticamente esses eventos para o topo da fila de processamento.

Em seguida, o RT-IMO redistribui recursos computacionais e largura de banda, reduzindo temporariamente o processamento de dados rotineiros para priorizar eventos críticos. O mecanismo de balanceamento de carga evita que um único nó de processamento seja saturado, enquanto o controle de filas impede o acúmulo de pacotes que poderia atrasar ações de controle.

Dessa forma, o RT-IMO permite respostas rápidas a falhas, mitigando riscos de interrupções em cascata e melhorando a resiliência operacional do sistema.

4.1.1.3 IMPACTO NAS MÉTRICAS DE DESEMPENHO

Nos cenários descritos, cada componente do RT-IMO influencia diretamente as métricas avaliadas na simulação. A priorização adaptativa reduz a latência de eventos críticos; o controle de congestionamento diminui o uso excessivo de largura de banda; o balanceamento de carga reduz a utilização de CPU em nós sobrecarregados; e o gerencia-

mento de filas reduz o tamanho do backlog, prevenindo perdas de dados. Esses mecanismos permitem que o RT-IMO mantenha desempenho previsível mesmo sob cargas variáveis, refletindo requisitos de aplicações IoT de missão crítica.

4.2 RESULTADOS

Esta seção apresenta o RT-IMO, uma estratégia de escalonamento adaptativa que equilibra latência, uso de recursos e equidade em ambientes IoT em tempo real. A análise considera os resultados experimentais obtidos, permitindo observar o desempenho do RT-IMO sob diferentes condições de carga. Diferentemente das abordagens tradicionais, o RT-IMO ajusta dinamicamente as prioridades dos sensores com base nas condições do sistema, assegurando eficiência sem sobrecarregar os recursos e influenciando o funcionamento global do sistema ao longo da execução.

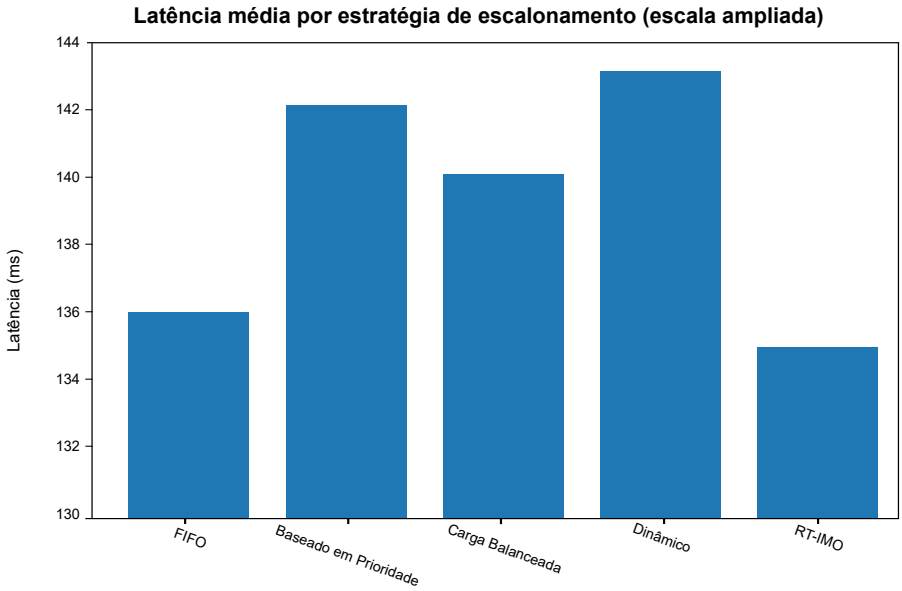

Figura 5 – Experimento de Latência.

A Figura 5 apresenta os resultados referentes à latência média. Observa-se que o RT-IMO alcança a menor latência entre as estratégias avaliadas, superando os métodos *FIFO*, por Prioridade, Balanceamento de Carga e Dinâmica. Esse comportamento decorre diretamente de sua política de priorização adaptativa, que leva em conta a proximidade dos prazos, a criticidade dos sensores e o comportamento recente do sistema. Enquanto abordagens como *FIFO* e Balanceamento de Carga tratam as tarefas de forma homogênea, ignorando a urgência temporal dos dados, o RT-IMO antecipa situações de congestionamento e reorganiza o processamento antes da formação de filas extensas. Embora as diferenças absolutas de latência não sejam expressivas, o resultado indica maior previsibilidade temporal, característica essencial para aplicações IoT em tempo real.

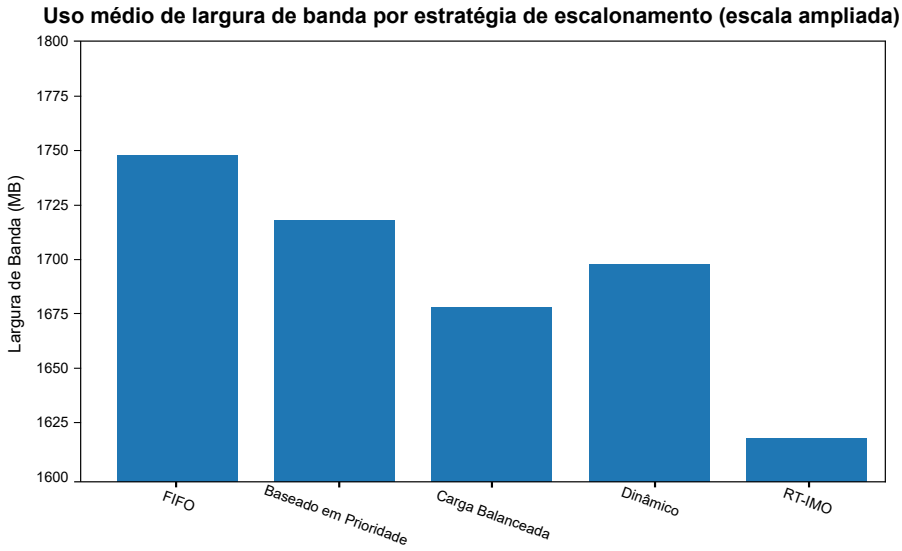

Figura 6 – Experimento de Uso de Banda.

A Figura 6 indica que o RT-IMO apresenta o menor uso de banda entre as estratégias analisadas, resultado associado à sua capacidade de modular o fluxo de dados de acordo com o estado da rede e a relevância das informações transmitidas. Estratégias tradicionais tendem a encaminhar dados de forma contínua e indiferenciada, o que pode gerar transmissões redundantes e maior ocupação da rede. No RT-IMO, a integração entre priorização e consciência de recursos permite reduzir o envio de dados de baixa relevância em momentos de maior carga, contribuindo para uma utilização mais eficiente da rede e para a estabilidade do sistema.

A Figura 7 A utilização de CPU inferior observada no RT-IMO evidencia que a estratégia consegue tomar decisões adaptativas sem introduzir sobrecarga computacional significativa. Embora o RT-IMO realize cálculos adicionais para atualização dinâmica de prioridades, esses custos são compensados pela redução do processamento de tarefas pouco relevantes ou redundantes. Em abordagens como *FIFO*, o sistema processa indiscriminadamente todas as tarefas que chegam, o que leva a altos níveis de uso de CPU mesmo quando parte desse processamento não é essencial. Já o RT-IMO atua de forma seletiva, evitando a execução de tarefas que, dadas as condições atuais do sistema, não agregam valor imediato. O resultado é um uso mais racional do processador, alinhado com os requisitos de dispositivos de borda e *gateways* IoT, nos quais a capacidade computacional é limitada e deve ser preservada para eventos críticos.

A Figura 8 apresenta os resultados relacionados ao tamanho médio das filas. O RT-IMO mantém filas mais controladas ao longo da execução, resultado direto de sua política ativa de gerenciamento de congestionamento. Em vez de permitir o crescimento indiscriminado das filas até o esgotamento dos recursos, o RT-IMO ajusta preventivamente as taxas de aceitação e descarta tarefas de baixa prioridade quando necessário. Filas

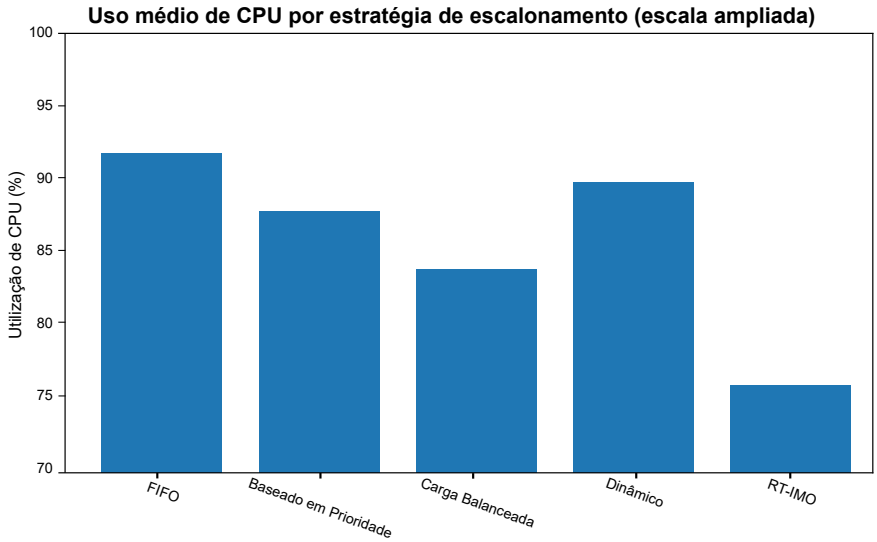

Figura 7 – Experimento de Utilização da CPU.

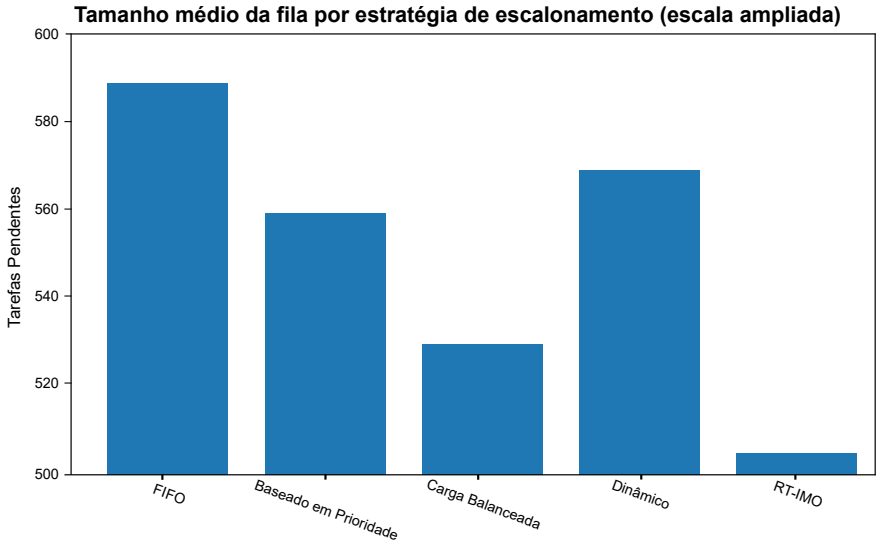

Figura 8 – Experimento de Tamanho da Fila.

menores implicam menor tempo médio de espera e maior previsibilidade no processamento, reforçando o comportamento de tempo real do sistema mesmo sob condições de carga elevada.

As Figuras 9 e 10 evidenciam o mecanismo de gerenciamento de prioridades do RT-IMO em cenários de sobrecarga extrema. Os resultados demonstram que tarefas críticas são majoritariamente preservadas, enquanto tarefas de prioridade média são parcialmente atendidas e tarefas de baixa prioridade apresentam maior taxa de descarte. Esse comportamento reflete uma política de degradação graciosa, na qual a perda controlada de dados menos relevantes é preferível à degradação global do desempenho. Tal estratégia

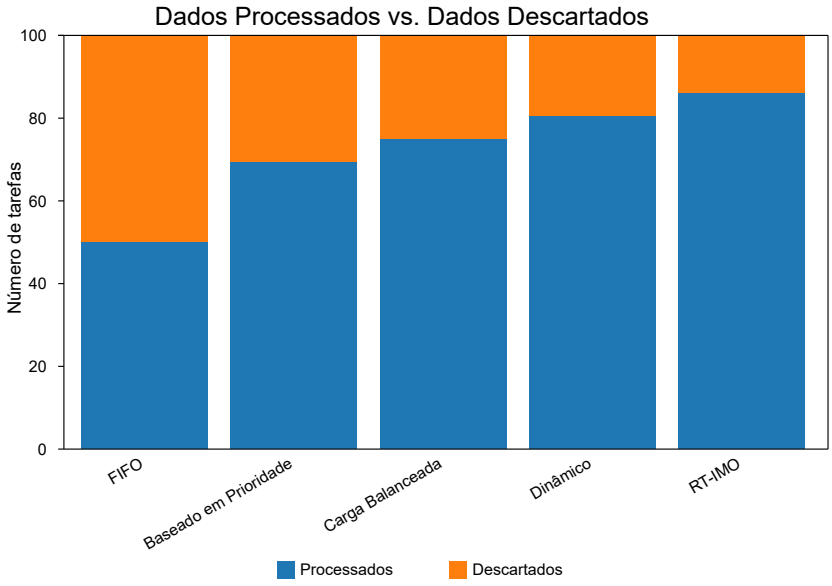

Figura 9 – Gerenciamento de Prioridades – Parte 1.

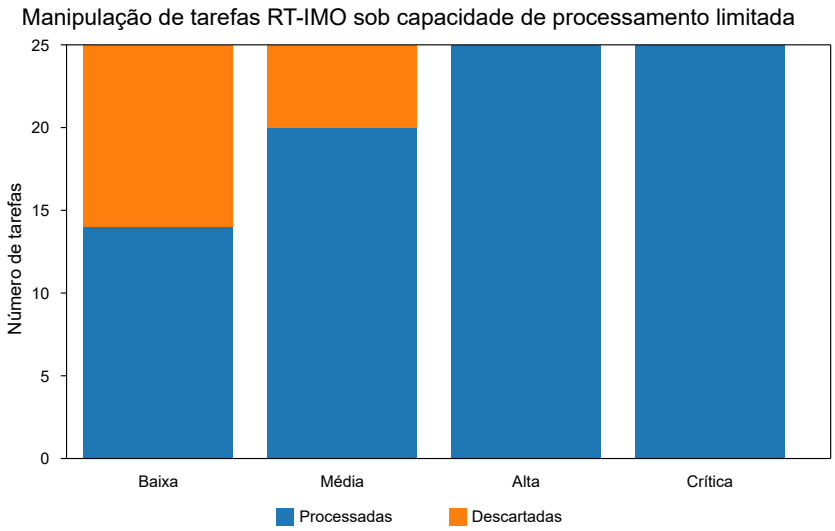

Figura 10 – Gerenciamento de Prioridades – Parte 2.

está alinhada aos princípios de sistemas de tempo real suaves, nos quais a manutenção da funcionalidade essencial tem precedência sobre a completude absoluta dos dados.

O RT-IMO demonstra uma abordagem equilibrada que integra responsividade em tempo real, eficiência de banda, otimização computacional e controle de filas. Embora apresente variações em métricas como latência e tamanho de fila, o RT-IMO se destaca por maior equidade, adaptabilidade e sustentabilidade do sistema. A Tabela 2 resume as diferenças entre o RT-IMO e as estratégias comparadas.

Estratégia	**Otimização de Latência**	**Adaptação de Recursos**	**Equidade**
FIFO	Parcial	Não	Não
Prioridade	Não	Parcial	Não
Balanceamento de Carga	Parcial	Sim	Sim
Dinâmica	Não	Parcial	Não
RT-IMO	Sim	Sim	Sim

Tabela 2 – Comparação entre o RT-IMO e as estratégias de escalonamento existentes.

5 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho abordou o desafio do escalonamento em tempo real de dados provenientes de sensores IoT heterogêneos por meio da avaliação de estratégias de orquestração existentes e da proposição do RT-IMO (Orquestração Multissensorial em Tempo Real para IoT) como uma abordagem inovadora para equilibrar latência, eficiência de recursos e equidade em ambientes dinâmicos. O estudo analisou as limitações dos métodos *FIFO*, Baseado em Prioridade, Balanceado por Carga e Dinâmico, destacando os *trade-offs* enfrentados na gestão de sensores orientados a eventos, contínuos e orientados a consultas. Por meio de experimentação extensiva, demonstramos que o RT-IMO supera as abordagens tradicionais ao adaptar dinamicamente as prioridades dos sensores, otimizar a alocação de recursos computacionais e manter a estabilidade do sistema sob cargas de trabalho variáveis.

Os resultados experimentais confirmaram que o RT-IMO alcança menor latência, maior taxa de eventos processados e melhor eficiência no uso de CPU e memória em comparação com os métodos de base. A estratégia mostrou-se especialmente adequada para aplicações de missão crítica, como monitoramento de saúde, automação industrial e sistemas de resposta a emergências. Diferentemente do escalonamento rígido baseado em prioridade, o RT-IMO garante que eventos de alta prioridade sejam processados pontualmente, sem causar inanição de recursos para sensores de menor prioridade — uma limitação comum em políticas tradicionais de escalonamento.

Diante dos resultados positivos obtidos com o RT-IMO, ainda existem oportunidades de aprimoramento que podem ser exploradas. Os principais desafios incluem a sobrecarga computacional do escalonamento dinâmico, o tratamento de flutuações extremas na carga de trabalho, a otimização do consumo energético, a garantia de escalabilidade, o fortalecimento da segurança e a validação do desempenho em ambientes reais. Melhorias futuras poderão integrar aprendizado de máquina para escalonamento preditivo, arquiteturas descentralizadas para escalabilidade, *offloading* energético de tarefas e mecanismos de cibersegurança para escalonamento seguro. Ao abordar essas limitações, o RT-IMO pode ser refinado como uma solução mais adaptativa, escalável e resiliente para sistemas IoT em tempo real da próxima geração.

REFERÊNCIAS

1 SAMUEL, A.; SIPES, C. Making internet of things real. *IEEE Internet of Things Magazine*, v. 2, n. 1, p. 10–12, 2019. Citado na página 19.

2 KUSHWAH, R.; BATRA, P. K.; JAIN, A. Internet of things architectural elements, challenges and future directions. In: *2020 6th International Conference on Signal Processing and Communication (ICSC)*. [S.l.: s.n.], 2020. p. 1–5. Citado na página 19.

3 RAJASEKAR, P. et al. Real-time stream processing in iot environments. In: *2024 Ninth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)*. [S.l.: s.n.], 2024. p. 1–5. Citado 2 vezes nas páginas 19 e 29.

4 BEDI, K. et al. Effectiveness of iot (internet of things) in real-time supply chain monitoring and management. In: *2024 Second International Conference Computational and Characterization Techniques in Engineering and Sciences (IC3TES)*. [S.l.: s.n.], 2024. p. 1–5. Citado na página 19.

5 ABDUALWHAB, M. et al. Leveraging iot data for real-time business decision-making. In: *2024 8th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*. [S.l.: s.n.], 2024. p. 1–8. Citado na página 19.

6 MARRIPUDUGALA, M. Real-time iot data analytics using advanced large language model techniques. In: *2024 Global Conference on Communications and Information Technologies (GCCIT)*. [S.l.: s.n.], 2024. p. 1–6. Citado 2 vezes nas páginas 19 e 29.

7 SEHRAWAT, D.; GILL, N. S. Smart sensors: Analysis of different types of iot sensors. In: *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*. [S.l.: s.n.], 2019. p. 523–528. Citado na página 19.

8 SAJJA, K.; BHOWMIK, B. Sensor classifications and their applications in iot systems. In: *2023 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*. [S.l.: s.n.], 2023. p. 293–298. Citado na página 19.

9 KHARE, S.; TOTARO, M. Big data in iot. In: *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. [S.l.: s.n.], 2019. p. 1–7. Citado na página 20.

10 VORAKULPIPAT, C. et al. Recent challenges, trends, and concerns related to iot security: An evolutionary study. In: *2018 20th International Conference on Advanced Communication Technology (ICACT)*. [S.l.: s.n.], 2018. p. 405–410. Citado na página 25.

11 IEEE Draft Standard for an Architectural Framework for the Internet of Things (IoT). *IEEE P2413/D0.4.6, March 2019*, p. 1–265, 2019. Citado na página 25.

12 LIU, X.; BAIOCCHI, O. A comparison of the definitions for smart sensors, smart objects and things in iot. In: *2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. [S.l.: s.n.], 2016. p. 1–4. Citado na página 26.

13 S., B. S.; M., V. R. S. Deployment of sensor nodes in iot network - a survey. In: *2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*. [S.l.: s.n.], 2021. p. 267–274. Citado na página 26.

14 D'SILVA, G. M. et al. Real-time processing of iot events with historic data using apache kafka and apache spark with dashing framework. In: *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology (RTEICT)*. [S.l.: s.n.], 2017. p. 1804–1809. Citado na página 29.

15 PADMAPRIYA, S. et al. Investigating scope of real-time data processing for iot in 5g revolution. In: *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. [S.l.: s.n.], 2024. p. 1–6. Citado na página 29.

16 NGUYEN, L. T. et al. Adaptive collision avoidance scheduling based on traffic and priority for iot sensor networks. In: *2021 International Conference on Advanced Technologies for Communications (ATC)*. [S.l.: s.n.], 2021. p. 73–78. Citado 2 vezes nas páginas 29 e 30.

17 STAVRINIDES, G. L.; KARATZA, H. D. Resource allocation and scheduling of real-time workflow applications in an iot-fog-cloud environment. In: *2022 Seventh International Conference on Fog and Mobile Edge Computing (FMEC)*. [S.l.: s.n.], 2022. p. 1–8. Citado 2 vezes nas páginas 29 e 30.

18 KHAN, M. N. et al. Energy-efficient dynamic and adaptive state-based scheduling (edass) scheme for wireless sensor networks. *IEEE Sensors Journal*, v. 22, n. 12, p. 12386–12403, 2022. Citado na página 30.

19 LIU, K.; ABU-GHAZALEH, N.; KANG, K.-D. Jits: just-in-time scheduling for real-time sensor data dissemination. In: *Fourth Annual IEEE International Conference on Pervasive Computing and Communications (PERCOM'06)*. [S.l.: s.n.], 2006. p. 5 pp.–46. Citado 2 vezes nas páginas 30 e 33.

20 NAM, M. et al. A real-time ubiquitous system for assisted living: Combined scheduling of sensing and communication for real-time tracking. *IEEE Transactions on Computers*, v. 57, n. 6, p. 795–808, 2008. Citado 3 vezes nas páginas 30, 31 e 33.

21 AHMAD, S. et al. Towards the design of a formal verification and evaluation tool of real-time tasks scheduling of iot applications. *Sustainability*, v. 11, n. 1, 2019. ISSN 2071-1050. Disponível em: <https://www.mdpi.com/2071-1050/11/1/204>. Citado 2 vezes nas páginas 31 e 33.

22 WEST, R.; POELLABAUER, C. Analysis of a window-constrained scheduler for real-time and best-effort packet streams. In: *Proceedings 21st IEEE Real-Time Systems Symposium*. [S.l.: s.n.], 2000. p. 239–248. Citado 2 vezes nas páginas 31 e 33.

23 HE, T. et al. Speed: a stateless protocol for real-time communication in sensor networks. In: *23rd International Conference on Distributed Computing Systems, 2003. Proceedings.* [S.l.: s.n.], 2003. p. 46–55. Citado 2 vezes nas páginas 32 e 33.

24 SAIFULLAH, A. et al. Near optimal rate selection for wireless control systems. In: *2012 IEEE 18th Real Time and Embedded Technology and Applications Symposium*. [S.l.: s.n.], 2012. p. 231–240. Citado 2 vezes nas páginas 32 e 33.

25 BUKHARI, A.; HOSSEINIMOTLAGH, S.; KIM, H. Opensense: An open-world sensing framework for incremental learning and dynamic sensor scheduling on embedded edge devices. *IEEE Internet of Things Journal*, v. 11, n. 15, p. 25880–25894, 2024. Citado 2 vezes nas páginas 32 e 33.

26 LIU, Y. et al. Efficient sensor scheduling strategy based on spatio-temporal scope information model. *Sensors*, v. 23, n. 12, 2023. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/23/12/5437>. Citado 2 vezes nas páginas 32 e 33.

27 ASWIN, K.; SHANMUGAPRIYA, N.; GOPI, R. Sliced: A secure and adaptive cloud–iot framework for low-latency e-learning environments. *Scientific Reports*, v. 16, n. 1, p. 1522, 2026. Citado 2 vezes nas páginas 32 e 33.

28 KHAIRNAR, S. Edge computing for iot devices: A comprehensive framework for distributed data processing and real-time analytics. *MetaVision Journal of Multidisciplinary Studies*, v. 1, n. 2, p. 1–11, 2024. Citado 2 vezes nas páginas 32 e 33.