



**INSTITUTO
FEDERAL**
Paraíba

Instituto Federal de Educação, Ciência e Tecnologia da Paraíba

Campus João Pessoa

Programa de Pós-Graduação em Tecnologia da Informação

Nível Mestrado Profissional

LEANDRO HENRIQUE BATISTA DA SILVA

**GERENCIAMENTO DE DISPOSITIVOS IOT
MULTIFUNCIONAIS CIENTE DOS RECURSOS
COMPUTACIONAIS**

DISSERTAÇÃO DE MESTRADO

JOÃO PESSOA

2025

Leandro Henrique Batista da Silva

**Gerenciamento de Dispositivos IoT Multifuncionais Ciente dos
Recursos Computacionais**

Dissertação apresentada como requisito parcial para obtenção do título de Mestre em Tecnologia da Informação, pelo Programa de Pós-Graduação em Tecnologia da Informação do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba – IFPB.

Orientador: Prof. Dr. Paulo Ditarso Maciel Jr.

João Pessoa

2025

Dados Internacionais de Catalogação na Publicação (CIP)
Biblioteca Nilo Peçanha - *Campus* João Pessoa, PB.

S586g Silva, Leandro Henrique Batista da.

Gerenciamento de dispositivos IoT multifuncionais ciente dos recursos computacionais / Leandro Henrique Batista da Silva. – 2025.

53 f. : il.

Dissertação (Mestrado em Tecnologia da Informação) – Instituto Federal de Educação da Paraíba / Programa de Pós-Graduação em Tecnologia da Informação (PPGTI), 2025.

Orientação: Profº. Dr. Paulo Ditarso Maciel Jr.

1. Internet das coisas. 2. Alocação de recursos. 3. Dispositivos multifuncionais. 4. Gerenciamento de rede. I. Título.

CDU 004.658(043)



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA

PROGRAMA DE PÓS-GRADUAÇÃO *STRICTO SENSU*
MESTRADO PROFISSIONAL EM TECNOLOGIA DA INFORMAÇÃO

LEANDRO HENRIQUE BATISTA DA SILVA

Gerenciamento de Dispositivos IoT Multifuncionais Ciente dos Recursos Computacionais

Dissertação apresentada como requisito para obtenção do título de Mestre em Tecnologia da Informação, pelo Programa de Pós- Graduação em Tecnologia da Informação do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba – IFPB - Campus João Pessoa.

Aprovado em 29 de agosto de 2025

Membros da Banca Examinadora:

Dr. Paulo Ditarso Maciel Jr.

IFPB - PPGTI

Dr. Ruan Delgado Gomes

IFPB - PPGTI

Dr. Marcelo Portela Sousa

IFPB

João Pessoa/2025

Documento assinado eletronicamente por:

- **Paulo Ditarso Maciel Junior**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 30/08/2025 10:13:22.
- **Ruan Delgado Gomes**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 30/08/2025 10:16:33.
- **Marcelo Portela Sousa**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 30/08/2025 10:36:05.
- **Leandro Henrique Batista da Silva**, DISCENTE (202319840009) DE Mestrado Profissional em Tecnologia da Informação, em 30/08/2025 12:49:33.

Este documento foi emitido pelo SUAP em 20/08/2025. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código 752206
Verificador: 27588513a4
Código de Autenticação:



Av. Primeiro de Maio, 720, Jaguaribe, JOÃO PESSOA / PB, CEP 58015-435
<http://ifpb.edu.br> - (83) 3612-1200

Este trabalho é dedicado ao futuro.

AGRADECIMENTOS

Agradeço a minha mãe e avó que apoiaram meus estudos mesmo em momentos de dificuldade e ao meu orientador pelo apoio e oportunidade.

RESUMO

Muitos cenários de Internet das Coisas (IoT) exigem a captura contínua de informações a partir de sensores, dispositivos e unidades inteligentes. Em redes IoT também é comum encontrar dispositivos projetados para executar múltiplas tarefas. No entanto, a alocação de tarefas para esses dispositivos multifuncionais não é simples, devido à urgência e à prioridade que cada tipo de coleta de dados requer. Este trabalho apresenta um esquema de gerenciamento e alocação de recursos em uma rede IoT multifuncional, a partir de políticas previamente definidas. As políticas levam em consideração as características das aplicações em execução na rede e as diferentes especificações dos dispositivos disponíveis. Foram implementados testes em um simulador de rede com o objetivo de analisar a eficiência do esquema proposto em um contexto urbano. Os resultados indicam que considerar os requisitos exigidos pelas aplicações e as distintas características dos dispositivos IoT multifuncionais traz benefícios em relação ao uso dos recursos na rede, à alocação de recursos eficiente e redução da re-execução de tarefas.

Palavras-chave: Internet das Coisas, Alocação de Recursos, Dispositivos Multifuncionais, Gerenciamento de Rede.

ABSTRACT

Many Internet of Things (IoT) scenarios require the continuous collection of information from sensors, devices, and smart units. In IoT networks, it is also common to find devices designed to perform multiple tasks. However, task allocation to such multifunctional devices is not straightforward, given the urgency and priority that different types of data collection demand. This work presents a resource management and allocation scheme in a multifunctional IoT network, based on predefined policies. These policies take into account the characteristics of the applications running in the network as well as the heterogeneous specifications of the available devices. Tests were conducted in a network simulator to evaluate the efficiency of the proposed scheme in an urban context. The results indicate that considering both the application requirements and the diverse characteristics of multifunctional IoT devices provides benefits regarding network resource utilization, improves allocation efficiency, and can reduce task re-execution.

Keywords: Internet of Things, Resource Allocation, Multifunctional Devices, Network Management.

LISTA DE FIGURAS

Figura 1 – Ambiente IoT multifuncional.	28
Figura 2 – Políticas de fluxo de trabalho.	28
Figura 3 – Interação entre módulos para seleção de nós.	30
Figura 4 – Demonstração simplificada dos módulos do código.	31
Figura 5 – Execução de aplicações por grupo de nós.	32
Figura 6 – Resultados para os cenários sem consumo de energia.	37
Figura 7 – Resultados para os cenários com 10% de consumo de energia.	38
Figura 8 – Resultados para os cenários com 30% de consumo de energia.	39
Figura 9 – Resultados para os cenários com 50% de consumo de energia.	40
Figura 10 – Resultados para a percentagem de aplicações preemptadas (Eq. 3).	42
Figura 11 – Resultados para a percentagem de aumento do tempo total de execução (Eq. 4).	43

LISTA DE TABELAS

Tabela 1 – Tabela de comparação com trabalhos relacionados	26
Tabela 2 – Parâmetros utilizados nas simulações.	33

SUMÁRIO

1	INTRODUÇÃO	12
1.1	Motivação e Definição do Problema	13
1.2	Conceitualização	15
1.2.1	Internet das Coisas	15
1.2.2	Cidades Inteligentes	16
1.2.3	Simulação de Redes	17
1.3	Objetivos	17
1.3.1	Objetivo geral	17
1.3.2	Objetivos específicos	18
1.4	Estrutura do Documento	18
2	TRABALHOS RELACIONADOS	19
3	METODOLOGIA	27
3.1	Ambiente	27
3.2	Políticas	27
3.3	Simulação	30
3.4	Parâmetros de Simulação	31
4	RESULTADOS	36
5	CONCLUSÕES E PERSPECTIVAS FUTURAS	44
5.1	Conclusões	44
5.2	Perspectivas Futuras	44
	REFERÊNCIAS BIBLIOGRÁFICAS	46
	APÊNDICE A – ESTRUTURA DO CÓDIGO DO POSITRON (BASE- ADO EM NS-3)	50
A.1	Visão Geral do Repositório	50
A.2	Organização de Diretórios e Arquivos Relevantes	50
A.3	Evolução Estrutural a partir dos <i>Commits</i> POSITRON	50
A.4	Exemplos de Código	51
A.5	Execução e Parametrização	52

1 INTRODUÇÃO

Nas últimas décadas, o aumento da população urbana, bem como o rápido desenvolvimento tecnológico, têm se mostrado uma tendência. Muitas cidades estão experimentando um crescimento exponencial à medida que as pessoas se mudam das áreas rurais em busca de melhores empregos e educação, o que tem levado os serviços e a infraestrutura das cidades ao limite em termos de escalabilidade e segurança (KHATOUN; ZEADALLY, 2016). Esse crescimento populacional acelerado implica um maior consumo energético nas cidades, bem como o aumento do tráfego de dados e maiores exigências sobre os sistemas de infraestrutura urbana, como transporte, saúde, segurança e abastecimento. Com o objetivo de melhorar a qualidade de vida dos cidadãos e atender às necessidades mencionadas, o conceito de cidades inteligentes (*Smart Cities*) vem sendo desenvolvido.

Para as cidades inteligentes, a integração da Internet das Coisas (*Internet of Things - IoT*) é essencial. Como apontado por (FIROUZI et al., 2020), a IoT é um ecossistema vasto e diversificado que não pode ser entendido como uma única tecnologia, mas como uma estrutura abrangente que engloba múltiplos protocolos, conceitos e tecnologias específicas. A integração deste conceito permite uma grande variedade de cenários, oferece vantagens como escalabilidade e flexibilidade, e possibilita a comunicação entre uma ampla gama de dispositivos com diversas características técnicas, físicas e funcionais. Com os dispositivos físicos conectados à Internet e trocando informações em tempo real, é possível tomar decisões mais assertivas. Em contrapartida, a heterogeneidade desses dispositivos dificulta a interoperabilidade, e o uso eficiente dos recursos disponíveis nesses ambientes torna-se um desafio, dado que várias tarefas são executadas ao mesmo tempo pelos mais diversos dispositivos.

Para superar os desafios mencionados, um esquema de gerenciamento de recursos eficiente se mostra necessário. A eficiência na alocação de recursos para aplicações refere-se à capacidade de distribuir de forma apropriada os recursos computacionais, de armazenamento, de comunicação e de energia entre as diversas aplicações ou serviços de IoT em execução em uma rede de dispositivos conectados. Isso é crucial para garantir o funcionamento adequado do sistema, ao mesmo tempo que entrega a funcionalidade desejada e minimiza o desperdício de recursos. A eficiência pode ser definida por diversos aspectos-chave, como: **utilização de recursos**, que mede o quão efetivamente os recursos disponíveis são usados, avaliando o percentual de recursos (por exemplo, CPU, memória, armazenamento) que estão ativamente envolvidos no processamento de tarefas em relação ao total disponível; **latência**, que se refere ao tempo necessário para que uma requisição seja enviada da origem até o destino; **eficiência energética**, que foca em quão eficientemente a energia é utilizada por dispositivos IoT que frequentemente possuem uma vida útil limitada da bateria; **balanceamento de carga**, que envolve a distribuição equilibrada de tarefas entre os recursos disponíveis, evitando sobrecarregar alguns dispositivos enquanto outros

ficam subutilizados; e **tolerância a falhas**, que representa a capacidade do sistema de continuar operando mesmo diante de falhas de hardware ou software, possivelmente realocando recursos para sistemas de backup ou redundantes. A alocação eficiente de recursos é frequentemente alcançada por meio da combinação entre a utilização de hardware conforme os requisitos das aplicações e algoritmos de software inteligentes.

Assim, para atingir a eficiência na alocação de recursos, é importante adotar estratégias de gerenciamento que se adaptem dinamicamente às necessidades da rede. Nesse contexto, a utilização de uma abordagem de gerência de redes baseada em políticas permite que os serviços oferecidos sejam configurados de acordo com regras pré-estabelecidas, não havendo a necessidade de reinicialização ou reimplantação dos sistemas (GONÇALVES; SANTOS; HARA, 2014). As políticas oferecem uma maneira de formalizar ações entre os elementos da rede em resposta a eventos específicos, proporcionando flexibilidade para ajustes caso ocorram imprevistos e permitindo implementações sem alterações estruturais no ambiente. Ao utilizar políticas, torna-se viável gerenciar os recursos da rede, levando em conta as particularidades de cada dispositivo e as demandas de cada tarefa ou serviço. Além disso, uma abordagem de gerenciamento baseada em políticas facilita a incorporação de novas regras para atender a diversos objetivos, como redução de custos, otimização computacional e acordos comerciais, entre outros.

1.1 Motivação e Definição do Problema

Nas últimas décadas, os centros urbanos passaram por um crescimento populacional sem precedentes, com a estimativa de que, até 2050, cerca de 68% da população mundial viverá em cidades (KLEIN; ANDEREGG, 2021). Uma população maior significa filas mais longas, mais tráfego, competição por empregos, a necessidade de mais hospitais e centros de saúde, mais moradias e o aumento da geração, captação e distribuição de energia e água, entre outros problemas (United Nations, 2019). A IoT se mostra como uma tecnologia essencial para enfrentar tais desafios, uma vez que possibilita a coleta contínua de dados de diferentes naturezas, como condições ambientais, mobilidade urbana, consumo energético e segurança pública. Entretanto, embora a IoT esteja consolidada e amplamente empregada em diferentes domínios, ainda persiste o problema da subutilização e da má gestão de recursos. Em muitas situações, redes IoT são implantadas para atender a apenas um serviço específico, deixando o potencial da infraestrutura multifuncional inexplorado. Essas infraestruturas dispõem de dispositivos capazes de atender a múltiplas demandas e aplicações. Além disso, os dispositivos IoT de vários fabricantes podem ter características físicas diferentes, como capacidade da bateria e alcance de transmissão, o que pode limitar os tipos de aplicações que um determinado conjunto de dispositivos pode atender (PERERA et al., 2021). Essa heterogeneidade dos dispositivos dificulta a alocação eficiente de aplicações, resultando em desperdício de recursos e limitações no atendimento às demandas urbanas.

Frente ao contexto apresentado, nota-se a necessidade de soluções que garantam o uso eficiente dos recursos em ambientes urbanos inteligentes. A heterogeneidade dos dispositivos IoT, combinada com a crescente demanda por serviços em tempo real e a limitação de recursos computacionais e energéticos, exige mecanismos de gerenciamento capazes de responder dinamicamente às condições da rede. O consumo agregado de energia de dispositivos de baixa potência nas redes pode se acumular e tornar-se muito elevado (ALZAHIRANI; EJAZ, 2018). A ausência de um controle eficaz pode levar à sobrecarga de alguns dispositivos, à subutilização de outros, ao consumo excessivo de energia e à degradação da qualidade dos serviços prestados. Dessa forma, a definição de estratégias de alocação de recursos que considerem as características individuais de cada dispositivo, bem como as exigências específicas de cada tarefa, é o principal desafio. A utilização de abordagens baseadas em políticas mostra-se uma alternativa promissora, pois permite a adaptação dos recursos de forma flexível e orientada às necessidades específicas.

Muitas soluções da literatura focam apenas em aspectos específicos, como reduzir a latência de comunicação ou economizar energia, geralmente recorrendo a abordagens continuadas com a borda ou nuvem. Contudo, essas abordagens desconsideram a potencialidade dos dispositivos multifuncionais, capazes de atender a diferentes aplicações dentro da mesma infraestrutura. Este trabalho foca justamente nessa lacuna ao propor um esquema que realiza a alocação justa de tarefas em dispositivos IoT multifuncionais. O esquema baseia-se em políticas hierárquicas de seleção que consideram, no primeiro nível, a adequação entre os requisitos das aplicações e as capacidades mínimas dos nós e, em um segundo nível, a escolha entre estratégias de balanceamento ou saturação da carga. Essa concepção amplia a eficiência da alocação de recursos ao garantir que cada aplicação seja direcionada para o dispositivo mais adequado, respeitando as especificações de hardware e as demandas do serviço.

O problema abordado nesta pesquisa pode ser definido como o desafio de realizar a alocação justa e eficiente de recursos em redes IoT multifuncionais, caracterizadas pela heterogeneidade dos dispositivos e pela diversidade de requisitos das aplicações. Em tais cenários, não basta apenas distribuir tarefas entre os nós disponíveis; é necessário assegurar que cada aplicação seja executada no dispositivo mais adequado ao seu perfil de demanda, seja em termos de processamento, memória, armazenamento, capacidade energética ou taxa de transmissão. As soluções existentes, em sua maioria, concentram-se na otimização de métricas isoladas, como latência ou consumo energético; frequentemente recorrendo ao *offloading* para ambientes de borda ou nuvem, o que desconsidera o potencial de aproveitamento da infraestrutura já presente nos dispositivos multifuncionais. Assim, o problema principal consiste em como projetar mecanismos e políticas que permitam realizar uma alocação adaptativa e equitativa dos recursos, explorando de forma plena as capacidades locais da rede IoT, reduzindo a subutilização de recursos e mitigando a necessidade de sobrecarregar plataformas externas.

1.2 Conceitualização

1.2.1 Internet das Coisas

O conceito de internet das coisas é definido por (BAHGA; MADISETTI, 2014) como uma infraestrutura de rede global dinâmica, com capacidades de autoconfiguração, baseada em protocolos de comunicação padronizados e interoperáveis, onde “coisas” físicas e virtuais possuem identidades, atributos físicos e personalidades virtuais, utilizam interfaces inteligentes e são integradas de forma transparente à rede de informação, frequentemente comunicando dados associados aos usuários e seus ambientes. Esses dispositivos ou “coisas” podem variar desde objetos comuns, como eletrodomésticos, lâmpadas inteligentes e termostatos, até equipamentos industriais avançados, como drones e veículos conectados.

A essência da IoT está na capacidade desses dispositivos de coletar informações por meio de sensores embarcados, comunicá-las pela Internet e, frequentemente, processá-las em plataformas de computação em nuvem. Essa integração possibilita benefícios como monitoramento remoto, análise de dados em tempo real e automação de processos, criando ecossistemas inteligentes que interagem de maneira automatizada. As aplicações da IoT abrangem desde casas inteligentes, agropecuária inteligente, saúde digital até automação industrial, mudando a forma como vivemos e trabalhamos.

Os componentes fundamentais da IoT são dispositivos sensores, chamada remota de serviços, redes de comunicação e o processamento de eventos sensíveis ao contexto (KHODADADI; DASTJERDI; BUYYA, 2016). Os dispositivos IoT também podem ter características distintas, dependendo da fabricação, o que pode limitar as aplicações que atendem. Como uma forma de ampliar esse limite, esta pesquisa foca em dispositivos multifuncionais, ou seja, dispositivos que podem atender múltiplas tarefas. Assim, uma única rede IoT multifuncional poderia atender a diferentes aplicações e suportar diferentes tipos de dispositivos, o que permitiria substituir várias outras redes IoT e reduzir custos. Portanto, os desenvolvedores de aplicativos aproveitariam a mesma infraestrutura existente de dispositivos IoT. Nesse contexto, dispositivos multifuncionais atuam como plataformas de computação robustas para uma ampla gama de aplicações. Priorizando a privacidade e a segurança acima de tudo, a ênfase é dada ao processamento de dados, imagens e sons no próprio dispositivo, reduzindo a dependência do armazenamento ou transmissão de dados baseados em nuvem.

No entanto, iniciativas de pesquisa desconsideram o potencial dessa infraestrutura e focam em descarregar as aplicações para algum ambiente de borda ou nuvem, como forma de alocar recursos e reduzir métricas de uso. Contrariando essa tendência, iniciativas como o antigo *SAGE: A Software-Defined Sensor Network* (CATLETT et al., 2020) e agora o *Array of Things Project* (CATLETT et al., 2022) exemplificam a implementação de um ecossistema extensível de monitoramento urbano e conscientização ambiental integrado a uma infraestrutura de IoT. No contexto agrícola, por exemplo, é mostrado em (GUEBSI; MAMI; CHOKMANI, 2024)

que drones desempenham um papel crucial em intervenções diretas nas lavouras, oferecendo soluções precisas e eficazes para diversas operações agrícolas. Algumas delas incluem capturar imagens aéreas em diferentes espectros e aplicar agrotóxicos, pesticidas e fertilizantes.

No entanto, com o rápido crescimento da IoT, notam-se desafios críticos. A interoperabilidade entre dispositivos de diferentes fabricantes é um dos principais obstáculos, uma vez que muitos deles operam em protocolos e padrões distintos. Além disso, a gestão eficiente de recursos de rede, como transmissão e armazenamento, também exige soluções para lidar com o aumento de dispositivos interconectados.

1.2.2 Cidades Inteligentes

As cidades inteligentes, ou *smart cities*, representam um conceito que busca integrar tecnologia para aprimorar a qualidade de vida urbana. Essas cidades adotam uma ampla variedade de soluções tecnológicas e sistemas de informação para enfrentar os desafios urbanos modernos, como mobilidade e segurança, devido ao crescimento populacional. De acordo com (ROSCIA; LONGO; LAZAROIU, 2013), as cidades inteligentes são compostas por estruturas avançadas de Tecnologia da Informação e Comunicação (TIC), que permitem a conexão de setores variados, considerando aspectos sociais e econômicos.

O conceito de IoT é crucial para o funcionamento desse modelo urbano. De acordo com (PEDROSO et al., 2021), tal tecnologia tornou-se básica e essencial para a operação de diversos cenários urbanos inteligentes. Além disso, a coleta e a análise de grandes quantidades de dados, por meio de sensores, dispositivos e infraestruturas conectadas à Internet, fornecem informações em tempo real sobre diversos aspectos da vida urbana, desde o monitoramento do tráfego e do transporte público até o fornecimento de energia, água e segurança. Esses dados alimentam sistemas de gestão, possibilitando uma administração mais ampla, além de oferecer serviços personalizados para a população.

Contudo, essa grande quantidade de dados gerada por sensores, dispositivos e sistemas interconectados em cidades inteligentes nos mostra outro conceito fundamental. *Big Data* é um conceito que surgiu juntamente com o crescimento da Internet, não se relacionando somente à grande quantidade de dados acumulados, mas também à busca de formas para organizá-los e analisá-los. De acordo com (TAURION, 2013), a IoT é um impulsionador poderoso para o *Big Data*, com seus objetos gerando dados a todo instante.

Adicionalmente, a gestão eficiente dos recursos e a segurança dos dados coletados são fundamentais para assegurar que as soluções tecnológicas sejam sustentáveis e benéficas em cidades inteligentes. Questões de privacidade e a necessidade de regulamentações claras também são críticas para o sucesso das *smart cities*, devido ao grande volume de dados com informações pessoais e sensíveis envolvidas. De acordo com (LEAL; SANTOS, 2025), o conceito de cidade inteligente vai além da simples digitalização de serviços públicos, implicando um novo paradigma de planejamento urbano, baseado na inovação, na participação cidadã e na

sustentabilidade ambiental. Portanto, os avanços em cidades inteligentes dependem tanto de inovações tecnológicas quanto do desenvolvimento de estruturas regulatórias e modelos de governança que permitam uma integração segura, eficiente e equitativa das tecnologias.

1.2.3 Simulação de Redes

As simulações computacionais são essenciais em pesquisas na área de redes de computadores, pois permitem a avaliação de desempenho, escalabilidade e novas tecnologias de forma econômica e controlada. Elas possibilitam a repetição de experimentos, a exploração de cenários fisicamente impraticáveis e a validação de soluções inovadoras. Além disso, são uma ferramenta segura para testar diferentes condições de rede e comparar abordagens, sem os custos e riscos associados a redes físicas reais. Existem diversas alternativas de simuladores de redes, sendo o *Network Simulator 3*¹ (NS-3) um dos mais utilizados por sua flexibilidade, precisão e ampla comunidade de suporte, tornando-o ideal para pesquisas acadêmicas e desenvolvimento de novas tecnologias de redes. O NS-3 é frequentemente utilizado para simular uma ampla gama de redes, incluindo redes sem fio, com fio, móveis e redes IoT.

De acordo com (RILEY; HENDERSON, 2010), o NS-3 é um ambiente de simulação baseado em eventos discretos, onde cada evento está associado a um tempo específico de execução, e a simulação avança ao processar os eventos na ordem temporal do tempo de simulação. Quando um evento é processado, ele pode gerar zero, um ou mais eventos subsequentes. À medida que a simulação avança, os eventos são consumidos e novos eventos podem ser gerados. A simulação termina automaticamente quando não há mais eventos na fila ou quando um evento especial de “parada” é encontrado.

Desenvolvido em C++ e com suporte para integração com Python, o NS-3 destaca-se por suas várias características notáveis. Seu modelo de rede flexível suporta uma diversidade de protocolos, permitindo simulações detalhadas e adaptáveis. Ele cobre as camadas física, de enlace, de rede e de transporte do modelo OSI, facilitando a análise de redes complexas em diferentes níveis de abstração. Além disso, o NS-3 oferece integração com ferramentas e aplicações reais, o que enriquece a simulação com dados práticos e cenários realistas. Suas avançadas ferramentas de análise de desempenho fornecem informações detalhadas sobre o comportamento das redes, permitindo uma avaliação precisa e aprofundada dos sistemas simulados. Essas características fazem do NS-3 uma ferramenta eficaz para realizar simulações detalhadas e realistas.

1.3 Objetivos

1.3.1 Objetivo geral

- Desenvolver um sistema de gerenciamento que considere as características heterogêneas dos dispositivos IoT multifuncionais e avaliar a qualidade do sistema, bem como a eficácia

¹ <http://nsnam.org/>

das políticas de alocação definidas, por meio de experimentos de simulação.

1.3.2 Objetivos específicos

- Propor um esquema para o gerenciamento da alocação de recursos em dispositivos IoT multifuncionais, levando em consideração tanto as características das aplicações quanto as especificações dos dispositivos;
- Definir políticas hierárquicas de alocação que permitam selecionar os dispositivos mais adequados para cada aplicação, considerando critérios como adequação, balanceamento de carga ou saturação;
- Implementar o esquema de alocação em um simulador de rede, como prova de conceito, estendendo o ambiente de simulação para representar características reais de dispositivos IoT multifuncionais (CPU, memória, armazenamento, bateria e taxa de transmissão);
- Construir cenários de avaliação, envolvendo diferentes perfis de aplicações (processamento intensivo, armazenamento intensivo e alta taxa de transmissão) e diferentes grupos de dispositivos com especializações correspondentes;
- Validar a eficiência do esquema proposto por meio de métricas de desempenho, como a quantidade de nós utilizados e o percentual de alocação justa, avaliando o impacto das políticas de balanceamento e saturação;
- Demonstrar os benefícios da abordagem justa de alocação, evidenciando que considerar explicitamente a heterogeneidade dos dispositivos e os requisitos das aplicações resulta em maior eficiência e aproveitamento da infraestrutura IoT.

1.4 Estrutura do Documento

Os capítulos subsequentes desta dissertação estão organizados da seguinte maneira:

- Os trabalhos relacionados são analisados no Capítulo 2, incluindo uma tabela com os principais quesitos envolvidos neste trabalho;
- No Capítulo 3 é apresentada a metodologia de pesquisa proposta nesta dissertação, descrevendo o ambiente multifuncional contemplado, as políticas elaboradas, os parâmetros de simulação e as métricas de avaliação;
- No Capítulo 4 são descritos os resultados obtidos a partir das métricas elencadas, comparando as duas políticas de alocação propostas;
- As considerações finais e as propostas de continuação do trabalho são apresentadas no Capítulo 5.

2 TRABALHOS RELACIONADOS

Embora a literatura apresente soluções para alocar recursos em redes IoT, tais soluções geralmente focam na redução de métricas específicas, como custo na comunicação (TSAI, 2018; CIMINO; DEUFEMIA, 2025), latência (ZHAO; WANG et al., 2019; SANGAIAH et al., 2020; CHELLADURAI et al., 2025) e consumo de energia (GUIM; METSCH et al., 2022; XAVIER et al., 2022; NIKPOUR et al., 2023; ANDAL; JAYAPAL, 2022; PAPAIOANNOU et al., 2025). Além disso, costumam ser voltadas para um ambiente ou solução específica, como em *Multi-access Edge Computing* (MEC) (XAVIER et al., 2020; BOLETTIERI et al., 2021; WANG et al., 2022), Zigbee (LV; HU; LV, 2020), *Software-Defined Networking* (SDN)/*Network Function Virtualization* (NFV) (MUKHERJEE et al., 2020) e *blockchain* (RAFIQUE et al., 2020). Também propõem algum tipo de prova de conceito (PoC) (CALDERONI; MAGNANI; MAIO, 2019; ALI; CALIS, 2019), otimização por meio de modelos matemáticos (LI; ZHAO et al., 2021; BASHIR et al., 2022) ou otimização por meio de rotinas (SOUSA; PINTO; PRAZERES, 2025). Adicionalmente, a maioria dessas abordagens foca no descarregamento (*offloading*) de aplicações como forma de alocar recursos.

O esquema de alocação proposto visa aprimorar a distribuição de recursos em ambientes IoT multifuncionais por meio da integração de: requisitos das aplicações, especificações de hardware dos nós e gerenciamento baseado em políticas. Essa abordagem inclui uma implementação de prova de conceito para validar a viabilidade técnica antes da implantação em larga escala. Ao considerar os requisitos das aplicações, o sistema pode adaptar a alocação de recursos para atender às necessidades específicas dos usuários de cidades inteligentes, priorizando as tarefas críticas. As especificações de hardware dos nós garantirão uma distribuição justa de recursos, melhorando o desempenho e reduzindo gargalos. O gerenciamento baseado em políticas oferece flexibilidade para ajustar dinamicamente a alocação de recursos com base na variabilidade do sistema e nas prioridades dos usuários. Em última análise, essa abordagem abrangente promete um sistema de alocação de recursos robusto e eficiente, adequado para ambientes computacionais complexos, como os de IoT.

Em (TSAI, 2018), os autores consideram várias restrições em um sistema IoT de alto desempenho para alocação de recursos, como: diferentes requisitos de usuários, diferentes tipos de dispositivos, enorme necessidade de comunicações, largura de banda da rede e poder de computação limitado. O trabalho propõe um algoritmo para resolver a questão da alocação de recursos em tal sistema, visando reduzir o custo total de comunicação entre os dispositivos. Para isso, são empregados conceitos de agrupamento de dados (*data clustering*) e meta-heurísticas, além do uso de *gateways* IoT para uma comunicação mais eficiente. No entanto, a solução não considera nós que desempenham múltiplas funções, nem utiliza políticas para o gerenciamento autônomo da rede IoT. Assim, a eficiência é medida em termos de comunicação de dados,

desconsiderando outros aspectos de recursos, como processamento e armazenamento.

Em (ZHAO; WANG et al., 2019), os autores estudam o desafio de alocar recursos de forma eficiente para servidores de borda (*edge servers*) em um cenário de crescente número de dispositivos e serviços de IoT em cidades inteligentes. Eles otimizam as aplicações hospedadas para um desempenho satisfatório e comparam algoritmos de descarregamento de carga de trabalho (*workload offloading*), assumindo o tempo de resposta como a principal métrica. Já em (SANGAIAH et al., 2020), os autores propõem um algoritmo de otimização para a alocação e o agendamento de recursos em sistemas IoT, com o objetivo de reduzir o custo total de comunicação entre os dispositivos e um *gateway* próximo. Isso torna inviável uma abordagem manual devido à heterogeneidade dos dados e à quantidade de recursos a serem alocados. A alocação eficiente proposta aqui, por outro lado, pode considerar diferentes aspectos além da comunicação para alocar melhor as aplicações em dispositivos IoT multifuncionais existentes. Além disso, implementamos uma política de balanceamento de carga e também introduzimos uma política de saturação.

Em (GUIM; METSCH et al., 2022), é proposto um gerenciamento autônomo do ciclo de vida para plataformas de borda (*edge*) de computação verde convergentes, o que permite uma orquestração da carga de trabalho com eficiência de recursos. Recursos dinâmicos e inteligentes que hospedam serviços de múltiplos usuários são configurados, garantindo os SLOs (*Service Level Objectives*) para cada serviço. Contudo, a eficiência é considerada apenas em termos de consumo de energia, e dispositivos multifuncionais são desconsiderados.

Já em (XAVIER et al., 2022), os autores apresentam um algoritmo de alocação de recursos totalmente distribuído para um ambiente de IoT-borda-nuvem. Ele gerencia o uso de recursos, promovendo a colaboração entre os nós de borda, e suporta a heterogeneidade e os requisitos das aplicações, reduzindo a latência e aumentando a eficiência energética na borda. Eles também visam otimizar o gerenciamento de energia, enquanto esta proposta lida com a alocação eficiente de diversos recursos computacionais, sendo capaz de balancear a carga de trabalho para economizar bateria.

Em (NIKPOUR et al., 2023), os autores desenvolveram uma estrutura IoT pensada para o gerenciamento de energia em cidades inteligentes. Essa estrutura não só coleta e armazena dados, mas também serve como uma plataforma para o desenvolvimento de aplicações. Além disso, o estudo aprofunda-se em soluções inteligentes de gerenciamento de energia e explica como os dados coletados contribuem para o monitoramento contínuo e a melhoria da eficiência do sistema. Apesar de um contexto compartilhado que ressalta o papel vital da captura contínua de dados por meio de sensores multifuncionais e dispositivos inteligentes, é importante notar que este estudo se concentra principalmente na redução do consumo de energia. Esse trabalho, por outro lado, foca em um contexto mais amplo no processo de gerenciamento de recursos, que é facilitado por estratégias baseadas em políticas.

Similarmente, os autores em (ANDAL; JAYAPAL, 2022) apresentam um sistema de

gerenciamento de energia em tempo real integrado a uma plataforma adequada que se beneficia da IoT para monitorar e processar dados de controle em tempo real. Sensores multifuncionais detectam dados sem fio, transmitindo-os para servidores em nuvem via conectividade com a internet. O sistema demonstra a capacidade de gerenciar a produção de energia em diversas microrredes e permite o controle remoto do consumo e da geração de energia. Embora o artigo se aprofunde no gerenciamento de recursos em redes IoT, este trabalho foca na alocação equitativa de dispositivos multifuncionais, enquanto este estudo se concentra em um sistema de energia renovável.

Em (XAVIER et al., 2020), os autores propõem um algoritmo de alocação de recursos para um ambiente CoT (*Cloud of Things*). Esse ambiente combina uma arquitetura de três camadas, envolvendo dispositivos IoT, nós de borda (*edge nodes*) e a nuvem. A proposta suporta a heterogeneidade de dispositivos e aplicações, aproveitando a natureza distribuída dos nós de borda para promover a colaboração durante o processo de alocação. O algoritmo colaborativo segue uma abordagem baseada em heurística, inspirada em um modelo econômico, para resolver o problema de alocação de recursos em CoT. Os resultados simulados indicam um uso eficiente dos recursos do sistema, ao mesmo tempo em que atendem aos requisitos de latência e consideram diferentes prioridades das aplicações IoT, em comparação com uma abordagem baseada em nuvem de duas camadas. Ao contrário deste trabalho, que funciona de forma centralizada, o algoritmo colaborativo depende do uso de recursos externos para uma maior eficiência na alocação de recursos.

O estudo em (CLARINDO; CASTRO; AGUIAR, 2021) propõe uma arquitetura que integra uma camada de nuvem com uma camada de computação em névoa (*fog computing*) para a extração, transformação e carregamento de dados em um depósito de dados espaciais (*spatial data warehouse*). Ele introduz diretrizes para que gestores de cidades inteligentes implementem essa arquitetura de forma eficaz. O trabalho aborda o contexto do uso da IoT em ambientes urbanos. Enquanto este artigo se concentra na gestão de dados espaciais para análise em cidades inteligentes, este trabalho destaca a alocação de tarefas em uma rede IoT multifuncional baseada em políticas predefinidas.

Em (BOLETTIERI et al., 2021), investiga-se a questão do acesso múltiplo de borda (MEC) por aplicações executadas em dispositivos IoT. A arquitetura MEC proposta visa suportar vários provedores de serviços de IoT em uma plataforma comum. Do ponto de vista da aplicação, o posicionamento de serviço (*placement*) e o gerenciamento de dados são modelados, assumindo dependências em nível de dados entre os serviços de IoT e os recursos de sensoriamento. Contudo, essa abordagem não considera a multifuncionalidade dos dispositivos IoT nem as diferentes especificações de hardware.

Já em (WANG et al., 2022), os autores propõem um algoritmo de descarregamento de carga (*offloading*) baseado em aprendizado profundo distribuído no cenário IoT SD-MEC. Múltiplas redes neurais são invocadas em paralelo para otimizar o agendamento de recursos.

Semelhante ao trabalho aqui apresentado, as características da aplicação são consideradas na decisão de descarregamento de carga. No entanto, a estratégia MEC busca reduzir a utilização da IoT, em vez de fazer um uso eficiente dos recursos disponíveis.

O trabalho (LV; HU; LV, 2020) descreve um sistema inteligente de monitoramento do ambiente urbano que utiliza a rede sem fio ZigBee. Esse sistema é composto por uma rede de monitoramento básica e um terminal receptor remoto. Os nós da rede são organizados dinamicamente, e cada nó recebe um endereço exclusivo. Assim como este, o trabalho aborda a necessidade de captura contínua de dados em cenários de IoT. No entanto, existem distinções nas soluções propostas. Enquanto este trabalho se concentra principalmente na alocação eficiente de recursos de rede, o artigo em questão enfatiza a coleta de informações e sua transmissão para um terminal remoto.

A pesquisa dos autores de (MUKHERJEE et al., 2020) apresenta o desenvolvimento de uma rede IoT distribuída baseada em SDN (*Software-Defined Networking*) com a implementação de NFV (*Network Function Virtualization*) para cidades inteligentes. O objetivo é aprimorar o desempenho, a escalabilidade, a disponibilidade, a integridade e a segurança da rede. Múltiplos controladores distribuídos e um esquema de agrupamento são empregados para melhorar o balanceamento de carga. Apesar das similaridades, este artigo propõe uma rede IoT distribuída baseada em SDN com implementação de NFV para cidades inteligentes, enquanto este trabalho apresenta uma abordagem de alocação de recursos por meio de políticas predefinidas.

O trabalho (RAFIQUE et al., 2020) sugere o uso da tecnologia Blockchain para uma implementação segura de Sistemas de Transporte Inteligentes (ITS), fornecendo uma estrutura que integra veículos inteligentes e infraestrutura de serviços de forma segura e eficiente. Enquanto ambos os trabalhos propõem soluções para otimizar a utilização de recursos de rede e aprimorar a eficiência de aplicações em contextos de IoT, este artigo em particular enfatiza a utilização de Sistemas de Transporte Inteligentes (ITS) e as preocupações com suas vulnerabilidades de segurança. Em contraste, este trabalho foca no gerenciamento eficiente por meio da alocação de recursos através de políticas.

Em (CALDERONI; MAGNANI; MAIO, 2019), os autores propõem a utilização do *IoT Manager*, uma estrutura abrangente desenvolvida na Universidade de Bolonha. O *IoT Manager* apresenta uma solução aberta e flexível para o gerenciamento de redes de sensores, completa com uma estratégia de implementação detalhada e um cliente prontamente disponível para fins de pesquisa e educação. Embora o trabalho aborde a necessidade imperativa de gerenciamento de redes IoT, o *IoT Manager* é apresentado como uma estrutura de código aberto (*open-source framework*) voltada para o gerenciamento de redes de sensores. Em contraste, a proposta apresentada aqui é projetada para a alocação de recursos em redes IoT multifuncionais. Enquanto o artigo enfatiza a estratégia de implementação e sua acessibilidade para pesquisa e educação, este trabalho foca na alocação eficiente de recursos por meio de políticas predefinidas.

Em (ALI; CALIS, 2019), os autores descrevem uma estrutura de governança inteligente

centralizada para monitorar o desempenho de entidades governamentais em uma cidade inteligente. Milhares de dispositivos IoT se comunicam entre si, gerando uma vasta quantidade de dados. O objetivo é controlar, gerenciar, monitorar, operar e consolidar essa rede de dispositivos a partir de um local central. A estrutura emprega técnicas de classificação ótimas para analisar dados e monitorar o desempenho das entidades governamentais. Ambos os trabalhos abordam o gerenciamento de dispositivos IoT em contextos diferentes. Enquanto o artigo citado introduz uma estrutura de governança inteligente para monitorar o desempenho de entidades governamentais, este trabalho é apresentado para o gerenciamento da alocação de recursos em uma rede IoT multifuncional.

Em (LI; ZHAO et al., 2021), a disponibilidade da rede é afetada por vários tipos de aplicações e dispositivos com capacidades computacionais limitadas. Para lidar com isso, é proposto um modelo cooperativo de descarregamento de carga (*offloading*) de aplicações IoT multiusuário, baseado em uma formulação de programação não linear inteira mista. Os resultados indicam a eficiência na alocação de parte das aplicações na borda (*edge*), reduzindo o consumo de energia, a latência e a utilização da rede. No entanto, a característica multifuncional dos dispositivos não é explorada e uma implementação de referência não é fornecida. Em (BASHIR et al., 2022), de forma semelhante ao trabalho anterior, uma abordagem de *offloading* é apresentada como uma estratégia de alocação dinâmica de recursos para ambientes de nuvem e névoa, utilizando regressão logística para calcular a carga de borda necessária. Um framework para classificar os nós disponíveis leva em consideração as decisões realizadas pelo algoritmo TOPSIS. Contudo, as formulações matemáticas em ambos os trabalhos não relacionam os diferentes perfis de carga de trabalho a dispositivos multifuncionais. Além disso, eles utilizam uma abordagem inversa à nossa proposta, que realiza uma alocação mais adequada nos nós finais com base em suas especificações.

Em (CIMINO; DEUFEMIA, 2025) os autores propõem o Sigfrid, um método para detectar interferências em casas inteligentes causadas por regras conflitantes do tipo IF-THEN. À medida que os dispositivos se tornam mais multifuncionais, essas regras podem gerar comportamentos inesperados e inseguros. O Sigfrid utiliza Modelos de Linguagem de Grande Escala (LLMs) para construir grafos de interação de cenas, identificando relações causais entre eventos. Uma nova técnica de engenharia de *prompts* é usada para melhorar o raciocínio dos LLMs. Apesar de levar em consideração o aumento no uso de dispositivos multifuncionais em redes IoT, o artigo foca apenas na detecção de interferências, não considerando as demais características dos dispositivos.

Em (PAPAIIOANNOU et al., 2025), é proposto um Sistema de Gerenciamento de Recursos Hierárquico (HRMS) para cidades inteligentes com IoT, usando uma arquitetura descentralizada que vai do nível de edifícios até a escala da cidade. O sistema é baseado no Nó Adaptativo Resiliente (ARN), que gerencia energia de forma autônoma e se recupera de falhas. O HRMS permite decisões em tempo real, distribuição eficiente de recursos e alta tolerância

a falhas. Testes mostraram melhoria de até 50% na eficiência energética e redução de 30% no tempo de inatividade, destacando seu potencial para promover cidades mais sustentáveis e resilientes. O artigo foca principalmente em resiliência e otimização do consumo de energia; em contrapartida, este trabalho trata de forma mais ampla a alocação de recursos, levando em consideração características distintas dos dispositivos.

Em (CHELLADURAI et al., 2025), os autores propõem uma solução para reduzir o consumo de energia e a latência em cidades inteligentes baseadas em IoT, que enfrentam desafios relacionados à qualidade de serviço e à alocação de recursos. A proposta usa uma técnica chamada OL-GJO (aprendizado por oposição com Otimização Golden Jackal) para melhorar o agendamento de tarefas em uma arquitetura com três camadas: dispositivos IoT, UAVs (drones) na borda e a nuvem. Embora ambos os trabalhos tratem do gerenciamento de recursos em IoT, o artigo concentra-se especificamente em métricas de consumo de energia e latência, enquanto a nossa proposta adota uma perspectiva mais ampla com o uso de políticas que podem ser utilizadas para mais métricas, como processamento e armazenamento, ou ser adaptadas para regras de negócio.

Em (SOUSA; PINTO; PRAZERES, 2025), foi abordado o desafio de gerenciar dispositivos IoT dentro das infraestruturas de cidades inteligentes. Foi proposta uma solução adaptada aos requisitos específicos da gestão de dispositivos IoT, diferente da gestão tradicional de dispositivos de rede. A abordagem integra centenas de dispositivos em áreas urbanas, aproveitando as tecnologias de telecomunicação e informação (TIC) para melhorar os serviços urbanos. Foram revisadas arquiteturas e plataformas existentes e foi desenvolvido um protótipo para demonstrar a aplicação prática da solução. O protótipo garante a disponibilidade consistente de serviços e o gerenciamento eficiente de recursos. Apesar de o artigo levar em consideração o contexto de gerenciamento de redes IoT em cidades inteligentes, os autores se baseiam em rotinas operacionais que descrevem implementações e otimizações de procedimentos já existentes. No entanto, seu escopo está restrito à manutenção e otimização desses processos, enquanto o trabalho proposto atua em um nível estratégico ao introduzir mecanismos de alocação de recursos orientados por políticas que consideram não apenas a disponibilidade, mas também múltiplos fatores, como memória, armazenamento, processamento e eficiência energética.

No geral, o esquema proposto aborda a alocação de recursos em uma rede IoT multifuncional por meio de políticas predefinidas, satisfazendo todas as características importantes mencionadas anteriormente. A maioria dos trabalhos não considera características multifuncionais e políticas de gerenciamento autônomo, como (TSAI, 2018) e (CIMINO; DEUFEMIA, 2025), que enfatizam a eficiência da comunicação de dados. No entanto, alguns trabalhos lidam com a alocação de recursos para servidores de borda em cidades inteligentes, considerando algoritmos de *offloading* de carga de trabalho, como (ZHAO; WANG et al., 2019), que levam em conta os custos de comunicação, mas carecem de gerenciamento baseado em políticas e da utilização de dispositivos multifuncionais. Os trabalhos (GUIM; METSCH et al., 2022), (XAVIER et

al., 2022), (NIKPOUR et al., 2023), (ANDAL; JAYAPAL, 2022), (PAPAIOANNOU et al., 2025) e (CHELLADURAI et al., 2025) focam principalmente na gestão de energia, sem considerar as demais características e desconsiderando a multifuncionalidade. Os autores em (XAVIER et al., 2020) propõem uma plataforma para colaboração entre nós, e o trabalho (CLARINDO; CASTRO; AGUIAR, 2021) foca principalmente na gestão de dados espaciais como uma camada de serviço. Em contraste, o esquema proposto enfatiza a alocação de tarefas em uma rede IoT multifuncional e pode ser visto como um gerenciador de infraestrutura para ambos os trabalhos. Em (BOLETTIERI et al., 2021), a pesquisa também desconsidera a multifuncionalidade e as especificações de hardware. Já a pesquisa (WANG et al., 2022) considera as características das aplicações no agendamento de recursos, mas busca reduzir a utilização da IoT em vez de fazer um uso eficiente dos recursos disponíveis, como o esquema apresentado. Os autores em (LV; HU; LV, 2020) focam na captura contínua de dados, de forma semelhante ao esquema proposto, mas enfatizam a coleta e transmissão de informações em vez da alocação de recursos. O trabalho (MUKHERJEE et al., 2020) compartilha semelhanças com o escopo estudado, mas foca no aprimoramento da rede em vez da alocação de recursos através de políticas predefinidas. O artigo (RAFIQUE et al., 2020) enfatiza a segurança e a eficiência, e o artigo (CALDERONI; MAGNANI; MAIO, 2019) foca na gestão de redes de sensores; no entanto, ao contrário do esquema proposto, não se concentra na alocação de recursos baseada em políticas. O trabalho (ALI; CALIS, 2019) foca na gestão de aspectos da rede de dispositivos IoT, contrastando com a ênfase do esquema apresentado na alocação de recursos. Os trabalhos (LI; ZHAO et al., 2021) e (BASHIR et al., 2022) não exploram a multifuncionalidade, o gerenciamento baseado em políticas ou os perfis de carga de trabalho. Em particular, (SANGAIAH et al., 2020) foca na minimização dos custos de comunicação entre recursos e *gateways*, enquanto o esquema proposto visa otimizar o uso de recursos, combinando as necessidades da aplicação com as capacidades do dispositivo. Ambas as abordagens envolvem políticas de balanceamento de carga, mas diferem na implementação: o esquema aqui apresentado foi validado por meio de uma prova de conceito (PoC), enquanto em (SANGAIAH et al., 2020) foi empregada uma otimização baseada em heurísticas, utilizando algoritmos evolutivos. Por fim, (SOUSA; PINTO; PRAZERES, 2025) apresenta semelhanças no contexto; porém, aborda de forma diferente. Tem foco na organização da infraestrutura e nas rotinas operacionais, enquanto o trabalho proposto leva esse gerenciamento a um nível mais estratégico de alocação eficiente de recursos orientada por políticas.

A Tabela 1 compara o trabalho atual com outros estudos, considerando aspectos importantes do esquema proposto. Este trabalho diferencia-se da literatura por integrar requisitos das aplicações, especificações de hardware dos nós e gerenciamento baseado em políticas em um sistema de alocação justa e adaptativa de recursos para ambientes IoT multifuncionais, validado por uma implementação de prova de conceito voltada para cenários complexos de cidades inteligentes.

Tabela 1 – Tabela de comparação com trabalhos relacionados

	Q1	Q2	Q3	Q4	Q5	Q6
(TSAI, 2018)		✓	✓			✓
(ZHAO; WANG et al., 2019)		✓	✓			✓
(CALDERONI; MAGNANI; MAIO, 2019)	✓					✓
(ALI; CALIS, 2019)	✓					
(SANGAIAH et al., 2020)					✓	✓
(XAVIER et al., 2020)		✓	✓			✓
(LV; HU; LV, 2020)	✓					
(MUKHERJEE et al., 2020)	✓					✓
(RAFIQUE et al., 2020)	✓					✓
(LI; ZHAO et al., 2021)				✓		✓
(BOLETTIERI et al., 2021)		✓				
(CLARINDO; CASTRO; AGUIAR, 2021)	✓	✓				
(WANG et al., 2022)		✓				✓
(BASHIR et al., 2022)				✓		✓
(GUIM; METSCH et al., 2022)			✓			
(XAVIER et al., 2022)		✓				✓
(ANDAL; JAYAPAL, 2022)	✓			✓		✓
(NIKPOUR et al., 2023)	✓			✓		✓
(CIMINO; DEUFEMIA, 2025)	✓			✓		
(PAPAIANOANNOU et al., 2025)	✓	✓	✓		✓	✓
(CHELLADURAI et al., 2025)	✓	✓	✓	✓		✓
(SOUSA; PINTO; PRAZERES, 2025)	✓	✓	✓	✓		✓
Este Trabalho	✓	✓	✓	✓	✓	✓

Q1: Implementa prova de conceito.

Q2: Considera os requisitos da aplicação.

Q3: Considera as especificações dos nós.

Q4: Aborda o contexto de dispositivos multifuncionais.

Q5: Utiliza gerenciamento baseado em políticas.

Q6: Aborda a alocação de recursos IoT.

3 METODOLOGIA

3.1 Ambiente

Diferentemente da arquitetura tradicional de IoT, centrada em sensores e atuadores conectados diretamente a um ponto central de coleta de dados para aplicações específicas, arquiteturas multifuncionais, tais como (CATLETT et al., 2022), adotam nós urbanos inteligentes e reconfiguráveis, capazes de coletar, processar e compartilhar dados de múltiplas fontes e finalidades em tempo real, ampliando sua aplicabilidade em contextos complexos como cidades inteligentes. Uma rede IoT multifuncional resulta dos diferentes tipos de funcionalidades em diferentes dispositivos e configurações. Esse ambiente atende a diferentes tipos de aplicações que solicitam serviços em diversos contextos (por exemplo, IA, ITS, *blockchain* etc.), como é esperado em cidades inteligentes. Além disso, essas aplicações possuem as mais variadas exigências de hardware. Pode-se supor que o monitoramento de vias públicas requer uma capacidade mínima de armazenamento disponível nos nós da rede, enquanto a realização de criptografia pode beneficiar-se de um maior poder de processamento. Certamente, aplicações que realizam o sensoramento de algum fenômeno físico precisam transmitir dados constantemente pela rede. No entanto, mesmo que os dispositivos IoT apresentem características heterogêneas, é provável que se encontre grupos de nós com configurações semelhantes. Tais semelhanças podem indicar que um determinado grupo de nós é mais adequado para a execução de um tipo específico de aplicação.

A metodologia adotada nesta dissertação baseou-se na utilização do NS-3 para modelar e simular um ambiente representativo de IoT multifuncional. O gerenciamento de dispositivos IoT envolveu a alocação justa e eficiente de recursos em um cenário de aplicações urbanas. A Figura 1 apresenta a topologia da rede estudada, na qual a camada de controle gerencia a alocação justa de recursos. O controlador serviu como interface entre as aplicações e a infraestrutura de rede, avaliando as demandas das aplicações por meio do **Módulo de Decisão** (MD), que utiliza políticas para selecionar os dispositivos mais adequados. Essa abordagem baseada em políticas permite que o sistema tome decisões adaptativas, considerando as condições da rede. O **Módulo de Configuração** (MC) comunica-se com os dispositivos para aplicar a alocação sugerida, enquanto o **Módulo de Persistência** (MP) registra e atualiza as informações dos nós na base de dados, mantendo um histórico de execução e a visão atualizada do estado da rede.

3.2 Políticas

A seleção dinâmica de nós utiliza uma hierarquia de políticas para escolher dispositivos com base no estado da rede e nas demandas das aplicações. A hierarquia de políticas permite refinamentos sucessivos para encontrar o subconjunto de nós mais adequado para cada solici-

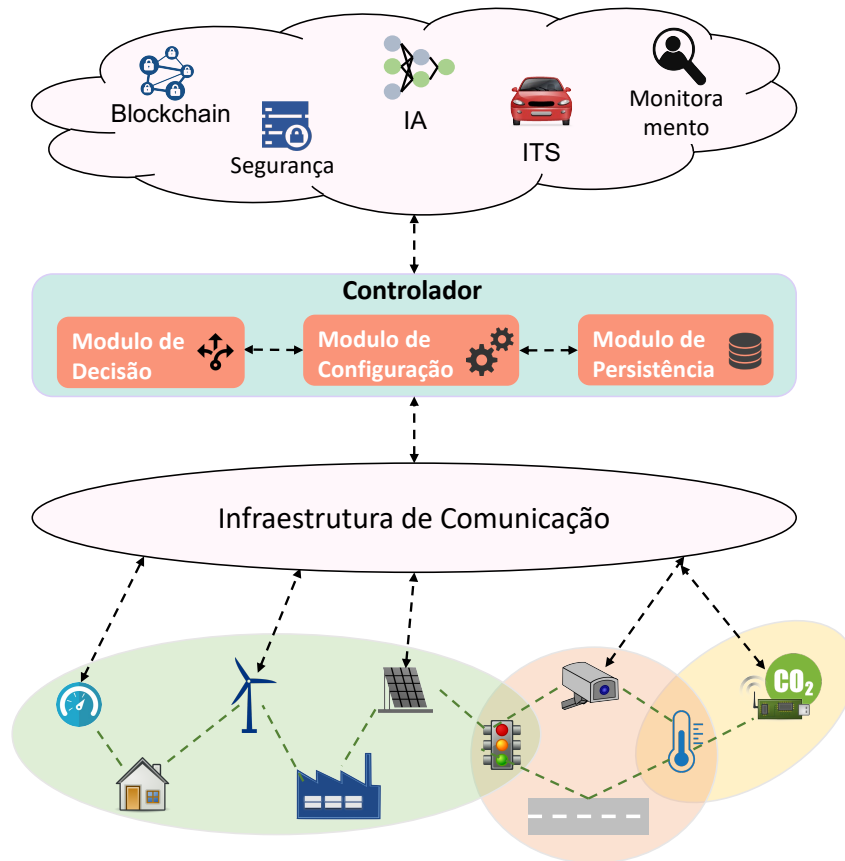


Figura 1 – Ambiente IoT multifuncional.

tação, além de flexibilizar a adição de novas políticas com diferentes objetivos, como o custo computacional.

Foram utilizadas três políticas simples em dois níveis hierárquicos para validar o sistema, conforme mostrado na Figura 2. A função `atende()` seleciona todos os nós que atendem minimamente aos requisitos de uma aplicação. A função `seleciona()` escolhe os primeiros nós de uma lista, e a função `ordena()` organiza a lista conforme um critério. Inicialmente, o sistema aplica uma política de adequação, escolhendo nós que atendem aos requisitos mínimos. No segundo nível, a seleção é refinada com base no critério da aplicação, utilizando políticas de **balanceamento** ou **saturação**. O balanceamento distribui a carga igualmente, enquanto a saturação prioriza nós mais sobrecarregados, liberando outros para futuras requisições.

```

if politica == ADEQUACAO
    then listaNos = atende(listaNos, nos.recursos, app.requisitos)
if politica == BALANCEAMENTO
    then listaNos = seleciona(n, ordenaLista(listaNos, criterio, ASC))
if politica == SATURACAO
    then listaNos = seleciona(n, ordenaLista(listaNos, criterio, DESC))
  
```

Figura 2 – Políticas de fluxo de trabalho.

O Algoritmo 1 descreve os passos para a alocação de uma aplicação em um nó da rede após sua chegada. O algoritmo começa aguardando a chegada da aplicação e extrai sua

descrição, incluindo requisitos mínimos, como processamento, memória ou bateria (linhas 2-3). Em seguida, traça um perfil da aplicação para o primeiro nível hierárquico de gerenciamento (linha 4) e consulta a base de dados para listar nós que atendem a esses requisitos (linha 5). Se a lista estiver vazia, uma mensagem de indisponibilidade é enviada (linha 7). Caso contrário, o algoritmo testa e aplica políticas de balanceamento ou saturação (linha 10), ordenando a lista de nós conforme necessário (linha 11) para, finalmente, alocar a aplicação de acordo com seus requisitos.

Algorithm 1 Processo de alocação de aplicação em nó multifuncional.

Require: AppDesc

```

1: while Verdade do
2:   AppDesc  $\leftarrow$  aguardaApp()                                ▷ linhas 2, 3 e 4 executadas pelo MD
3:   requisitos  $\leftarrow$  estimaReq(AppDesc)
4:   perfil  $\leftarrow$  definePerfil(requisitos)
5:   listaNós  $\leftarrow$  aplica política de adequação conforme perfil          ▷ MC consulta o MP
6:   if (listaNós = Vazia) then                                     ▷ MC executa deste ponto em diante
7:     ecoa("Não há nó disponível")
8:     configuraAlertaDisponibilidade()
9:   else
10:    listaNós  $\leftarrow$  aplica política de balanceamento ou saturação
11:    alocaApp(AppDesc, listaNós)
12:   end if
13: end while
  
```

A complexidade computacional do algoritmo pode ser analisada sob duas perspectivas: considerações tecnológicas e o próprio algoritmo. Questões tecnológicas, como atraso de transmissão e poder computacional, são abstraídas, assumindo que diversas técnicas de computação em borda e em nuvem podem tratá-las. Já a complexidade algorítmica envolve a busca por nós que atendem a determinados critérios e, em seguida, sua ordenação com base em uma política escolhida (balanceamento ou saturação). A busca geralmente possui complexidade $O(n)$, onde n é o número de elementos. A ordenação, para algoritmos conhecidos como o *mergesort*, costuma ter complexidade $O(n \log n)$. Combinando ambas as etapas, a complexidade total é a soma das duas. Como a ordenação normalmente domina, especialmente em listas grandes, a complexidade geral tende a ser $O(n \log n)$ em média. No entanto, é possível melhorar o tempo de processamento realizando uma pré-ordenação proativa com base na ocupação dos nós.

A Figura 3 mostra a comunicação entre os módulos, detalhando o processo desde a chegada de duas aplicações até a execução final. Destaca-se a fase crítica de seleção do dispositivo, que envolve consultar e ordenar candidatos com base no perfil da aplicação e na política de gerenciamento. O diagrama também ilustra um cenário em que um nó (nó X) fica sem bateria, levando o módulo de controle (MC) a selecionar um novo dispositivo (nó Y). Após a execução das aplicações, o MC solicita uma atualização de estado dos nós ao módulo de persistência (MP) para garantir informações precisas para futuras seleções. Mensagens de atualização e de cadastro/saída dos nós no MP foram omitidas para simplificação.

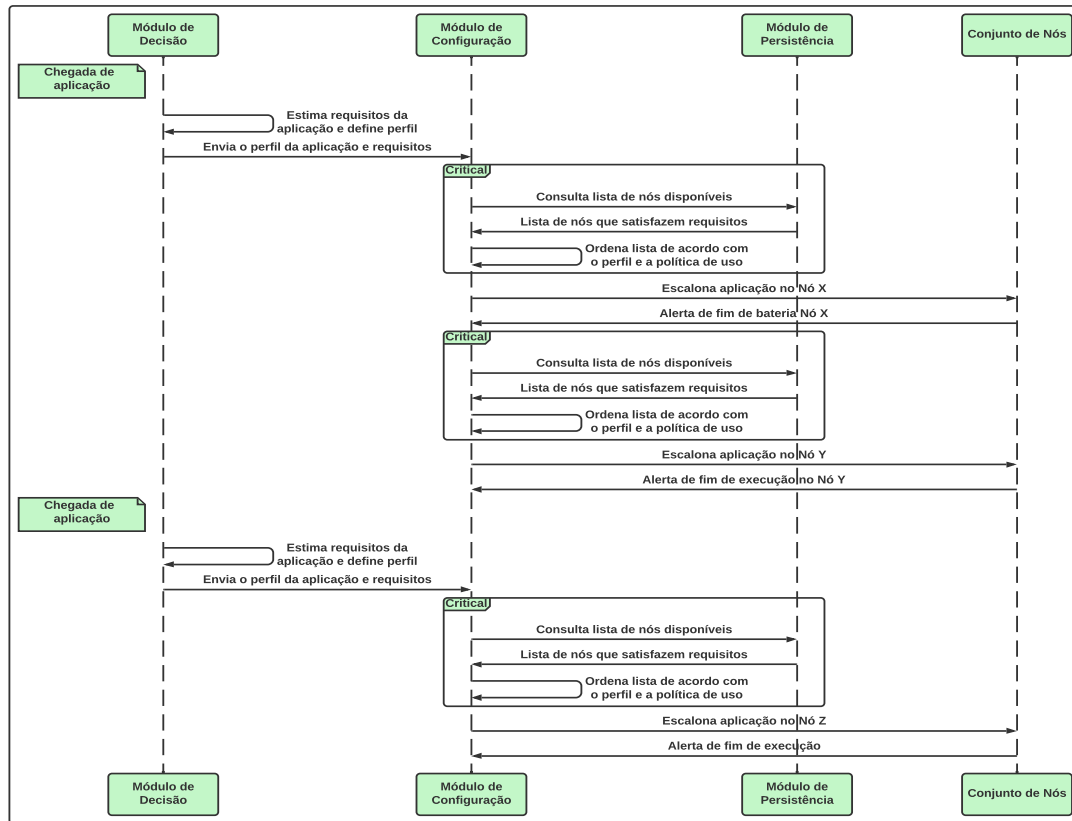


Figura 3 – Interação entre módulos para seleção de nós.

3.3 Simulação

A simulação foi implementada em C++, no simulador de redes NS-3, versão 3.38. No que diz respeito às especificações de IoT, foi herdada a classe que implementa os nós da rede para adicionar atributos comuns a dispositivos multifuncionais: nível de bateria, taxa inicial de consumo, taxa atual de consumo, CPU, memória, taxa de transmissão e capacidade de armazenamento. Para representar os módulos apresentados no NS-3, foi utilizada a lógica apresentada na Figura 4.

No início da simulação, são fornecidas as características das aplicações, que representam as requisições distribuídas ao longo do dia, bem como as dos nós, que correspondem a dispositivos IoT multifuncionais. Essas informações são utilizadas pelo módulo de decisão (`main.cc`) para classificar aplicações e nós em categorias específicas, conforme suas propriedades funcionais. Em seguida, essas configurações são encaminhadas ao módulo de configuração (`controller.cc`), responsável pela criação das estruturas que representam tanto as aplicações quanto os nós. Essas estruturas são, então, cadastradas no banco de dados por meio do módulo de persistência (`database.cc`).

Após a fase de configuração inicial, o módulo de decisão permanece em espera pela chegada de novas aplicações e suas respectivas características. Ao recebê-las, ele aciona o módulo de configuração, que realiza uma consulta ao banco de dados para identificar os nós

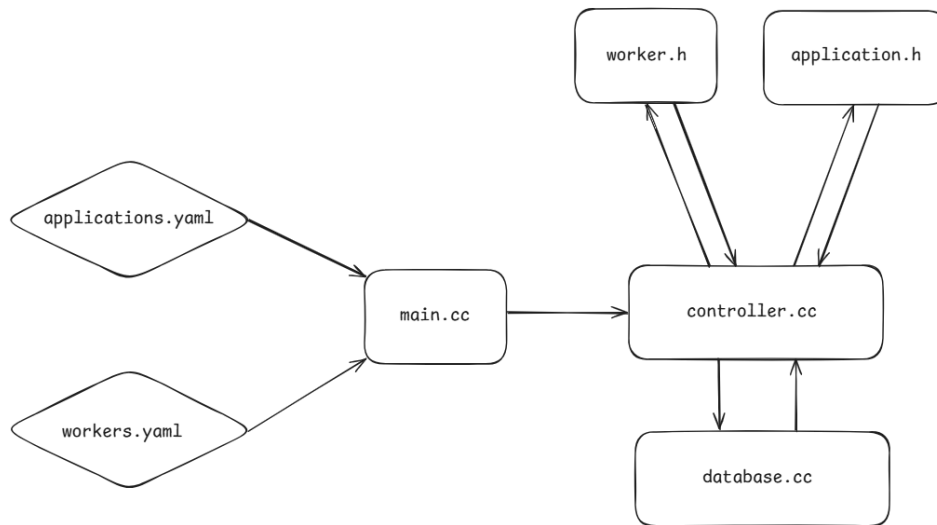


Figura 4 – Demonstração simplificada dos módulos do código.

disponíveis capazes de atender à requisição. A seguir, realiza-se a alocação do recurso de acordo com os parâmetros definidos para aquele cenário e registra-se a operação no banco de dados.

Cada nó foi caracterizado por atributos como a capacidade de CPU, memória, armazenamento e o nível inicial de consumo energético. O consumo energético é modelado com base em uma taxa de diminuição constante, calibrada de modo que, na ausência de qualquer requisição, o nó esgotaria totalmente sua bateria após 24 horas de operação contínua. Durante a simulação, essa taxa é acumulada de acordo com a quantidade de aplicações alocadas em cada nó, refletindo um aumento proporcional no consumo energético.

Cada aplicação foi caracterizada por requisitos como capacidade de CPU, memória e armazenamento, semelhantes aos dos nós, além do horário de chegada e do tempo de duração. Esses requisitos são usados para agrupá-las em perfis e determinar quais nós são mais adequados para atendê-las durante a execução.

Os parâmetros utilizados nos cenários de avaliação são apresentados na Seção 3.4 e os detalhes sobre a estrutura do código desenvolvido podem ser encontrados no Apêndice A.

3.4 Parâmetros de Simulação

Para validar o esquema, foram considerados três tipos de demandas das aplicações: alto poder de processamento, grandes capacidades de armazenamento e alta taxa de transmissão de dados. Três tipos de nós IoT foram configurados para atender a essas demandas específicas. O número total de nós varia de 30 a 180 nós no sistema, representando um fator de 10, 20, 30, 40, 50 ou 60 vezes para cada grupo. Um lote total de 600 aplicações (200 de cada tipo) é introduzido no sistema, com cerca de um terço delas chegando aleatoriamente a cada intervalo de turno de 8 horas ao longo do dia. Especificamente, as primeiras 200 chegam entre 0h e 8h da manhã, mais 200 entre 8h da manhã e 16h, e as últimas 200 após as 16h, totalizando 600 aplicações ao longo

do dia. A Figura 5 demonstra essa distribuição da execução das aplicações ao longo do tempo de simulação, em cada um dos três grupos distintos de nós, para um cenário específico. A duração de cada aplicação foi selecionada com base em uma distribuição normal, com uma média de 6 horas e um desvio padrão de 20%. O objetivo é simular um dia em uma cidade inteligente, onde diferentes aplicações podem chegar em momentos distintos e diferentes números de nós IoT estão disponíveis para executá-las.

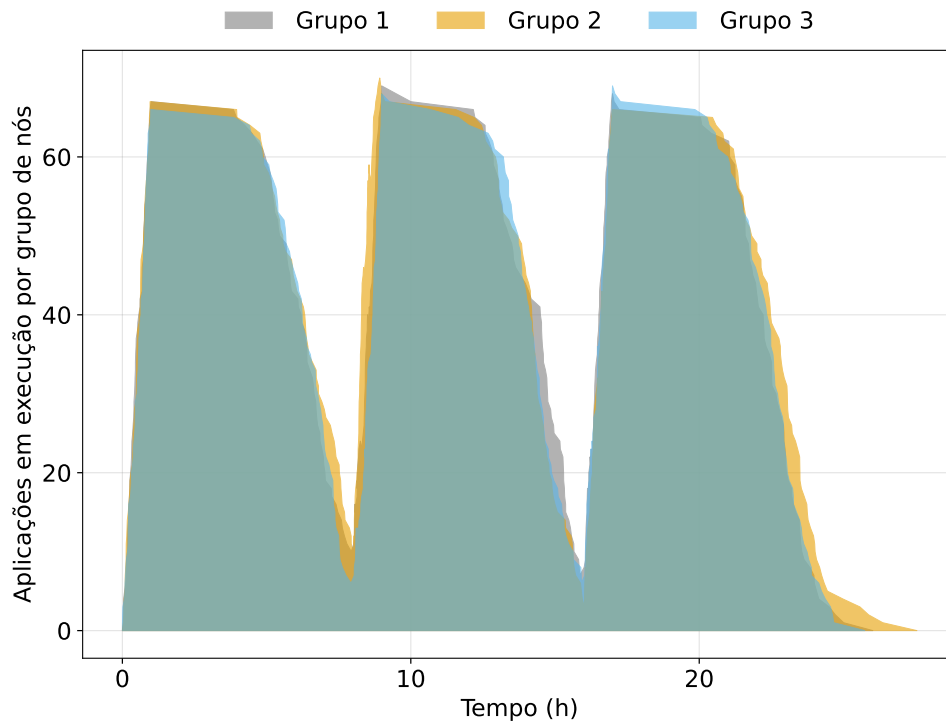


Figura 5 – Execução de aplicações por grupo de nós.

As chegadas das aplicações foram correspondidas com nós adequados para avaliar a eficiência do esquema. Cada cenário foi repetido 30 vezes para garantir uma análise estatística robusta e aumentar o nível de confiança. Em relação ao consumo de energia, o primeiro cenário foi simulado sem consumo de energia e, em seguida, os cenários em que os nós têm consumo ao longo da simulação, com 10%, 30% e 50% dos nós ficando totalmente sem energia. Dado que nem todos os nós podem estar convenientemente localizados perto de uma fonte de energia, os cenários buscam representar situações mais realistas em que podem ocorrer problemas com o executor da requisição e avaliar o impacto dessa perda sobre o esquema de alocação eficiente proposto e nas políticas investigadas. A Tabela 2 apresenta os parâmetros de simulação.

Para avaliar o efeito das políticas de alocação nos dispositivos IoT e o desempenho da alocação justa, foram definidas duas métricas. A primeira métrica mede o número de nós efetivamente utilizados, diferenciando entre políticas de balanceamento de carga e saturação. Assume-se um conjunto de perfis de aplicações P e a quantidade de aplicações em cada perfil I_p . Os nós IoT são classificados em tipos G , com a quantidade de nós em cada tipo representada por J_g . Isso permite definir o conjunto de aplicações A , o conjunto de nós IoT que chegam ao

Tabela 2 – Parâmetros utilizados nas simulações.

Parâmetros	Valores
Número de aplicações	600 (200 de cada tipo) ^f
Número de nós	$n \times 3$ (total de grupos) ^g ($n = \{10, 20, 30, 40, 50, 60\}$)
Políticas de gerenciamento	2 níveis (demanda ^h e carga ⁱ)
Duração das aplicações	distribuição normal com 6 horas de média e 20% de variância
Cenários de Bateria dos nós	(i) Sem consumo de bateria, (ii) 10% de consumo de bateria, (iii) 30% de consumo de bateria e (iv) 50% de consumo de bateria

^f Criptografia (cri), monitoramento (mon) e sensoriamento (sen).

^g Processamento (pro), armazenamento (arm) e transmissão de dados (tra).

^h Associação entre a demanda da aplicação e especificação do dispositivo.

ⁱ Saturação na utilização dos nós ou balanceamento de carga.

sistema N e o número total de nós $|N|$.

$$A = \{a_{p,i} \mid p \in P; i \in I_p\}, \quad N = \{n_{g,j} \mid g \in G; j \in J_g\} \quad \text{e} \quad |N| = \sum_{g=1}^{|G|} \sum_{j=1}^{|J_g|} 1.$$

A primeira métrica, portanto, indica o percentual de nós IoT utilizados ao longo da simulação e é representada por:

$$P_{\text{utilizados}} = \frac{100}{|N|} \sum_{g=1}^{|G|} \sum_{j=1}^{|J_g|} U(n_{g,j}), \quad (1)$$

onde uma função de utilização é definida por:

$$U(n_{g,j}) = \begin{cases} 1, & \text{se } n_{g,j} \text{ foi utilizado para executar uma aplicação qualquer;} \\ 0, & \text{caso contrário.} \end{cases}$$

Um percentual de utilização maior ou menor dos recursos físicos não indica necessariamente qualidade e está relacionado à política de alocação adotada. A segunda métrica mede a “justiça” na alocação das aplicações nos nós IoT, avaliando a adequação dos dispositivos às aplicações executadas, com foco em utilizar dispositivos com hardware mais apropriado para os perfis de aplicações.

Logo, defini-se o percentual de alocação justa em um determinado nó g como:

$$P_{\text{justa}}^g = \frac{100}{|N|} \sum_{p=1}^{|P|} \sum_{i=1}^{|I|} \sum_{j=1}^{|J|} U(n_{g,j,a_{p,i}}) \cdot J(a_{p,i},g), \quad (2)$$

onde utiliza-se uma especificação da função de utilização anterior que define,

$$U(n_{g,j}, a_{p,i}) = \begin{cases} 1, & \text{se } n_{g,j} \text{ foi utilizado para executar a aplicação específica } a_{p,i}; \\ 0, & \text{caso contrário.} \end{cases}$$

Adicionalmente, utiliza-se a seguinte função que relaciona o perfil da aplicação p ao grupo de dispositivos g verificando se o dispositivo tem as características mais desejáveis para a aplicação:

$$J(a_{p,i}, g) = \begin{cases} 1, & \text{se o grupo } g \text{ é o mais adequado para executar o perfil } p; \\ 0, & \text{caso contrário.} \end{cases}$$

Por fim, definiu-se o percentual total de alocação justa como o somatório dos percentuais de cada grupo de nós IoT:

$$P_{\text{justa}} = \sum_{g=1}^{|G|} P_{\text{justa}}^g.$$

Outras duas métricas que quantificam efetivamente a sobrecarga do sistema são a porcentagem de aplicações preemptadas, causadas principalmente por falhas nos nós; e o aumento do tempo de conclusão, que indica o crescimento percentual no tempo de resposta das aplicações. A primeira pode ser representada pela razão das aplicações que foram preemptadas pelo menos uma vez, dada por:

$$P_{\text{preemptadas}} = \frac{100}{|A|} \sum_{p=1}^{|P|} \sum_{i=1}^{|I|} R(a_{p,i}), \quad (3)$$

Em que,

$$R(a_{p,i}) = \begin{cases} 1, & \text{se } a_{p,i} \text{ foi reexecutada pelo menos uma vez;} \\ 0, & \text{caso contrário.} \end{cases}$$

Finalmente, o aumento percentual no tempo de conclusão leva em consideração tanto a duração da aplicação $D(a_{pi})$, quanto o tempo total no sistema $M(a_{pi})$; que, por sua vez, é o tempo em que a aplicação foi concluída $F(a_{pi})$ menos o tempo de chegada $A(a_{pi})$, ou seja:

$$M(a_{p,i}) = F(a_{p,i}) - A(a_{p,i})$$

representando assim, o tempo total que a aplicação leva desde o momento de sua chegada até sua conclusão. Desta forma, temos que

$$P_{\text{aumento}} = \frac{100}{|A|} \sum_{p=1}^{|P|} \sum_{i=1}^{|I|} \left(1 - \frac{D(a_{p,i})}{M(a_{p,i})} \right), \quad (4)$$

é a porcentagem de aumento do tempo de conclusão. Observa-se que

$$\frac{D(a_{p,i})}{M(a_{p,i})} \leq 1$$

e

$$D(a_{p,i}) = M(a_{p,i})$$

se e somente se $a_{p,i}$ for executada imediatamente (após a chegada) e de forma ininterrupta. Basicamente, o aumento percentual no tempo de conclusão normaliza o tempo de sistema de uma aplicação pela sua duração. Um aumento deste percentual indica um tempo de sistema proporcionalmente maior em comparação com o tempo total de execução.

De acordo com os parâmetros utilizados na simulação (Tabela 2), os conjuntos foram instanciados da seguinte maneira: $P = \{cri, mon, sen\}$, $I_p = \{1, 2, \dots, 200\} \ \forall p$, $G = \{pro, arm, tra\}$ e $J_g = \{1, \dots, n\} \ \forall g \mid n = \{10, 20, 30, 40, 50, 60\}$; este último definido como parâmetro variado em cada cenário de simulação.

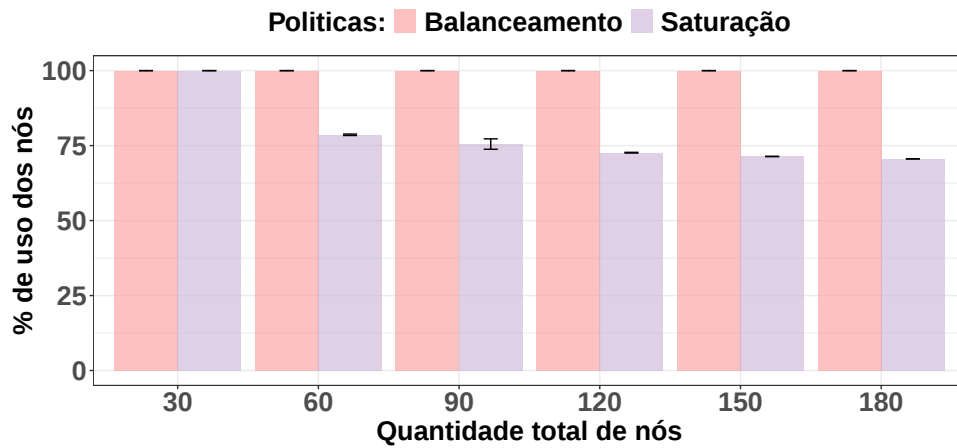
Por fim, os parâmetros de hardware utilizados na simulação foram: sistema operacional Windows 11 Pro, processador Intel Core i5-8400, placa-mãe Gigabyte H310M M.2 2.0 e 16 GB de memória RAM.

4 RESULTADOS

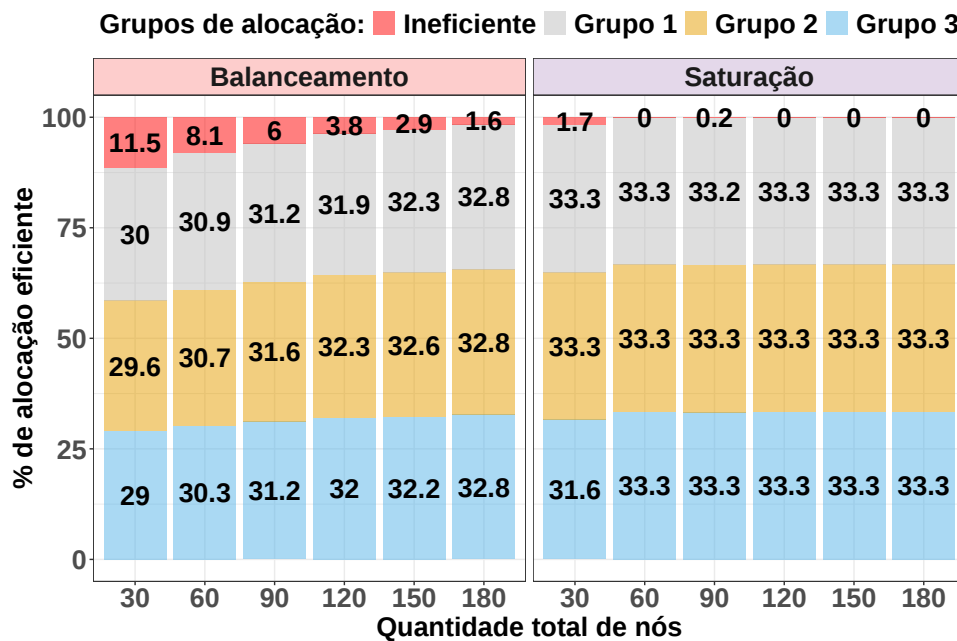
A Figura 6 ilustra os resultados da execução de um cenário simulado no qual os nós não tem consumo energético. Primeiramente, como mostrado na Figura 6(a), há uma diferença no número de recursos utilizados de acordo com a política aplicada, seja ela de balanceamento de carga ou não. Como esperado, uma política de balanceamento de recursos utiliza um número maior de nós disponíveis e pode ser aplicada em situações em que os vários nós da rede têm capacidade energética suficiente para executar as aplicações. Por outro lado, em uma situação em que é mais vantajoso usar um número menor de nós, como em um cenário de economia de consumo de energia, a política de saturação sempre utiliza uma porcentagem menor de nós. Ambos os gráficos na Figura 6(b) demonstram a capacidade de alocar adequadamente as aplicações, considerando as demandas específicas e as especificações dos nós existentes. No caso de uma política de saturação de dispositivos, as aplicações são alocadas da maneira mais “eficiente” possível, ou seja, para os nós mais adequados às suas execuções, validando assim o esquema de gerenciamento. Quando uma política de balanceamento de carga é usada, também há um alto percentual de alocação “eficiente”, com menos de 12% das aplicações alocadas de forma “ineficiente”. Em outras palavras, mesmo utilizando uma política que busca equilibrar o uso de recursos e, dessa forma, poupar outros, é possível alcançar uma alta eficiência na alocação. Assim, o custo de usar o balanceamento não causa grande prejuízo à “eficiência” exercida durante a alocação.

Vale destacar que uma alocação “ineficiente” para um determinado nó não significa que a aplicação não tenha sido executada. Essa situação representa uma alocação em que a aplicação seria mais apropriada para outro tipo de dispositivo, mas não foi desconsiderada. Por exemplo, nos resultados obtidos na Figura 6, a média de aplicações alocadas “ineficientemente” não foi superior a 12% no cenário balanceado com 30 nós. No geral, a taxa de execução das aplicações foi praticamente de 100% considerando todos os cenários.

A Figura 7 apresenta os resultados para a execução de um cenário simulado com 10% dos nós com consumo de energia. Conforme esperado, a Figura 7(a) mostra uma porcentagem maior de nós utilizados em comparação com a Figura 6(a), já que 10% dos nós ficam sem energia e saem da simulação. Assim, as aplicações rodando nesses nós são preemptadas e reexecutadas após a seleção de novos nós. Ainda há uma diferença entre as políticas de balanceamento de carga e de saturação, que podem representar padrões de utilização de acordo com as condições da rede. Na Figura 7(b), pode-se observar uma alocação quase totalmente eficiente das aplicações, independentemente do número de nós que deixam a rede. No entanto, um impacto evidente no processo de alocação é notado, uma vez que uma porcentagem desigual de aplicações alocadas é alcançada. Em particular, para a política de saturação, a porcentagem de aplicações “eficientemente” alocadas para o Grupo 1 de nós aumenta à medida que o número total de nós



(a) Porcentagem de uso de nós (Eq. 1).

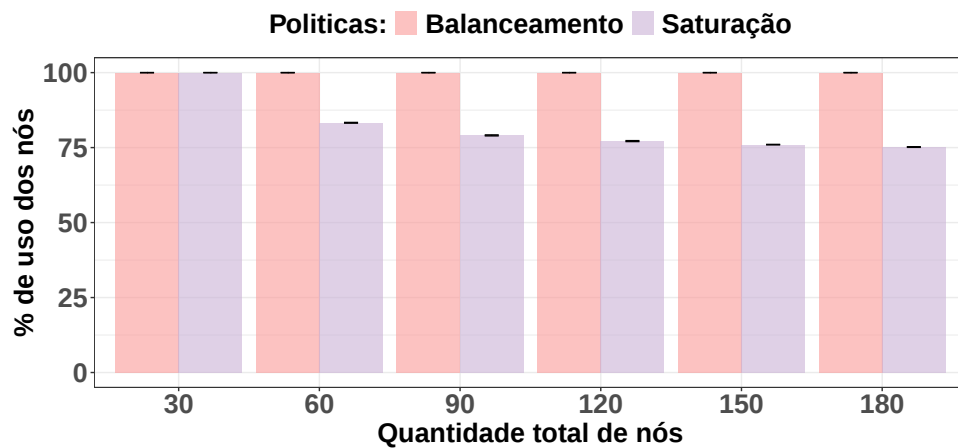


(b) Porcentagem de alocação eficiente (Eq. 2).

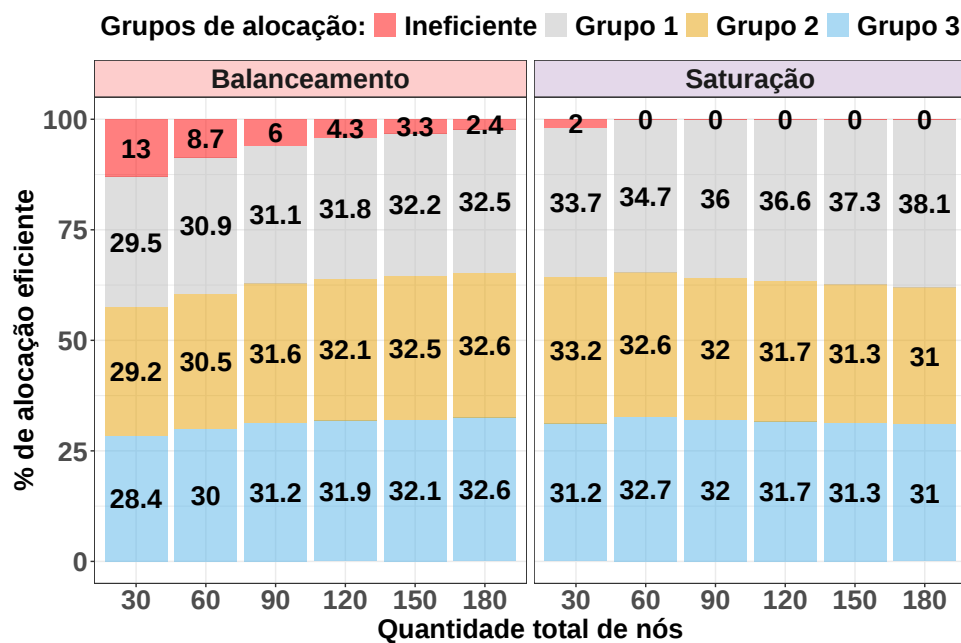
Figura 6 – Resultados para os cenários sem consumo de energia.

também aumenta. Isso ocorre quando um nó saturado perde sua bateria e um grande número de aplicações deve ser preemptado. Como as aplicações adequadas para o tipo de nó do Grupo 1 são mais restritivas em termos de requisitos (CPU e memória), foi observado um maior número de preemptões e, consequentemente, mais aplicações são realocadas nesses nós.

A Figura 8 apresenta os resultados para a execução de um cenário simulado com 30% dos nós com consumo de energia. Assim como no resultado do cenário com 10%, a Figura 8(a) mostra uma diferença cada vez menor entre as políticas de balanceamento de carga e de saturação. Além disso, o percentual de alocação eficiente também é fortemente impactado, como se pode observar na Figura 8(b). Em relação à política de balanceamento de carga, ainda há um certo equilíbrio, considerando apenas as aplicações que foram alocadas de forma eficiente,



(a) Porcentagem de uso dos nós (Eq. 1).

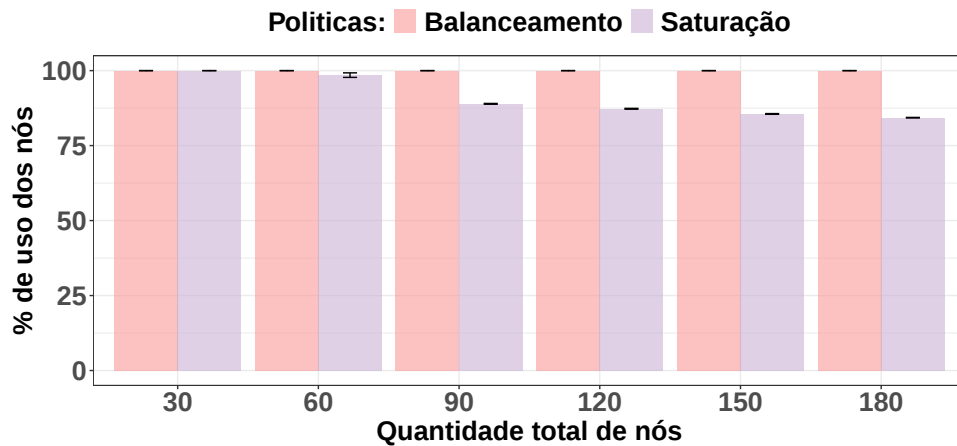


(b) Porcentagem de alocação eficiente (Eq. 2).

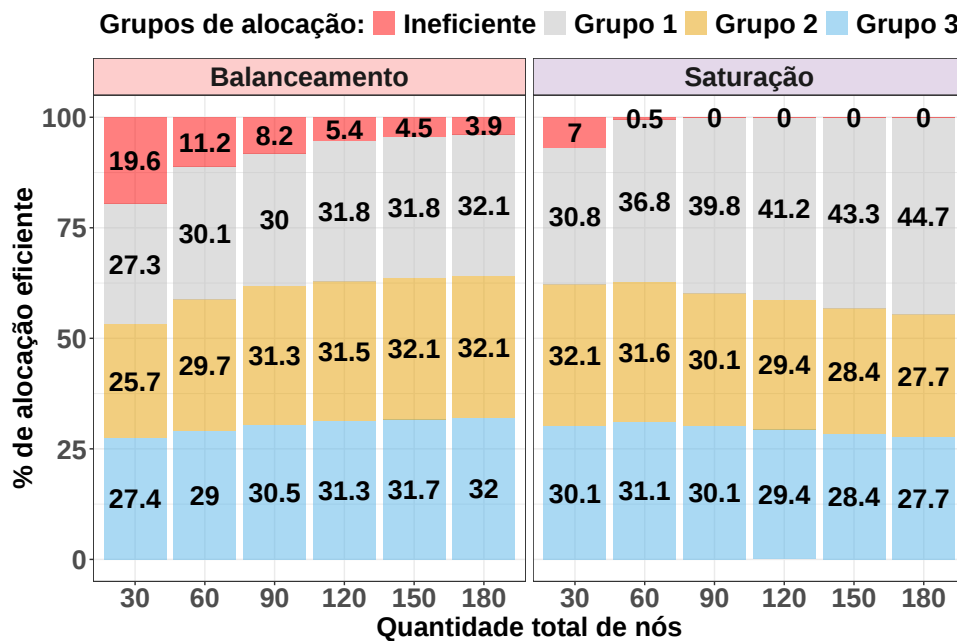
Figura 7 – Resultados para os cenários com 10% de consumo de energia.

especialmente à medida que o número de nós aumenta. No entanto, o número de aplicações alocadas “ineficientemente” aumentou significativamente, especialmente nos cenários com menor quantidade de nós. Os resultados com a política de saturação de recursos são ainda piores, devido à abordagem mais restritiva de tentar alocar o menor número possível de nós. Novamente, nota-se que a política de saturação aloca as aplicações de forma mais eficiente, mesmo que isso signifique realocar a mesma aplicação várias vezes.

Os resultados com um consumo de energia em 50% dos nós confirmam que o esquema de gerenciamento é fortemente impactado na alocação de recursos, como observado na Figura 9. A Figura 9(a) mostra que praticamente todos os nós são utilizados para executar as aplicações, como consequência da saída dos nós da simulação devido à falta de energia. Um consumo de



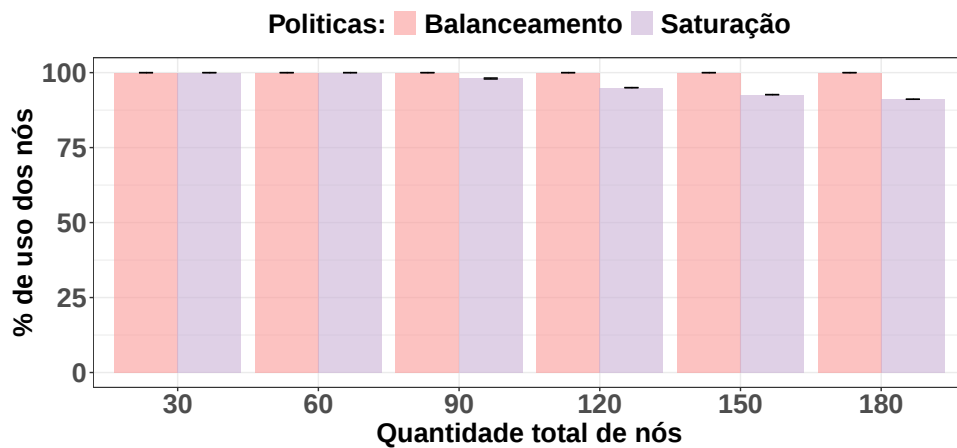
(a) Porcentagem de uso de nós (Eq. 1).



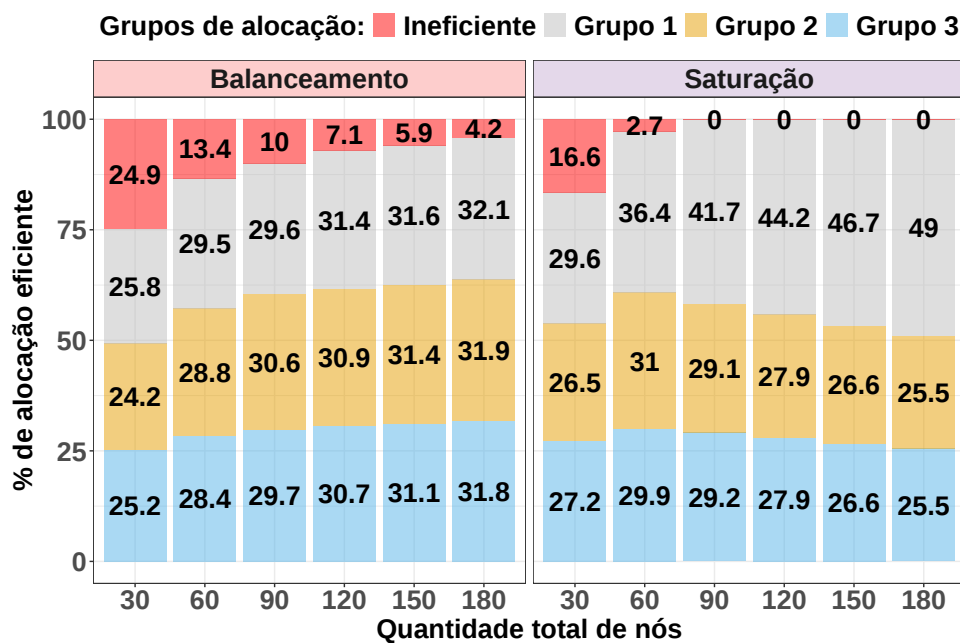
(b) Porcentagem de alocação eficiente (Eq. 2).

Figura 8 – Resultados para os cenários com 30% de consumo de energia.

energia de 50% dos nós é um cenário no qual a maioria dos nós é necessária para a simulação. Em outras palavras, a partir dessa quantidade de nós com consumo de energia, torna-se difícil distinguir entre a utilização balanceada de recursos e a saturação. Da mesma forma, a Figura 9(b) ilustra que a eficiência na alocação de recursos é ainda mais afetada, confirmando resultados insatisfatórios com uma alta taxa de perda de nós. À medida que o número de nós com consumo energético aumenta e sai da simulação, fica claro que os nós remanescentes em cada categoria são insuficientes para executar as aplicações “adequadamente”. Ou seja, aplicações preemptadas devem ser executadas por qualquer nó disponível e não pelos mais apropriados. Isso é mais evidente para um número menor de nós. Em resumo, à medida que o nível de consumo de energia aumenta, alocar recursos de forma eficiente torna-se cada vez mais desafiador.



(a) Porcentagem de uso dos nós (Eq. 1).



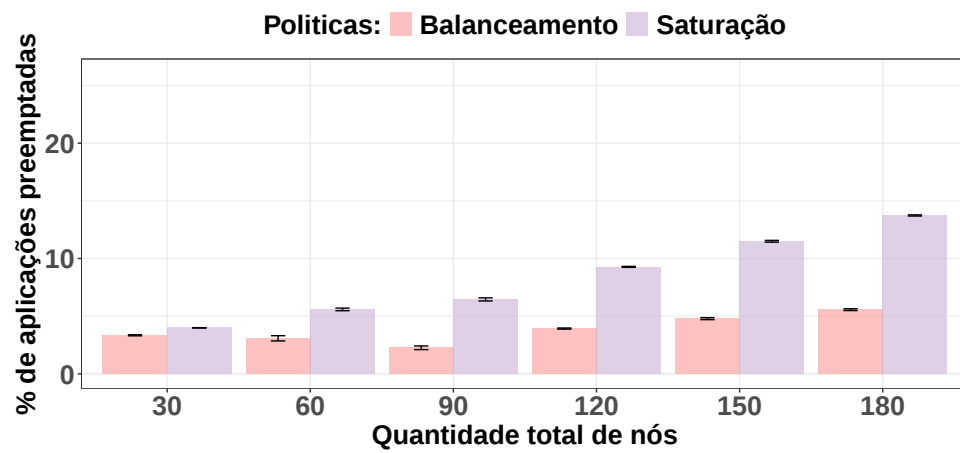
(b) Porcentagem de alocação eficiente (Eq. 2).

Figura 9 – Resultados para os cenários com 50% de consumo de energia.

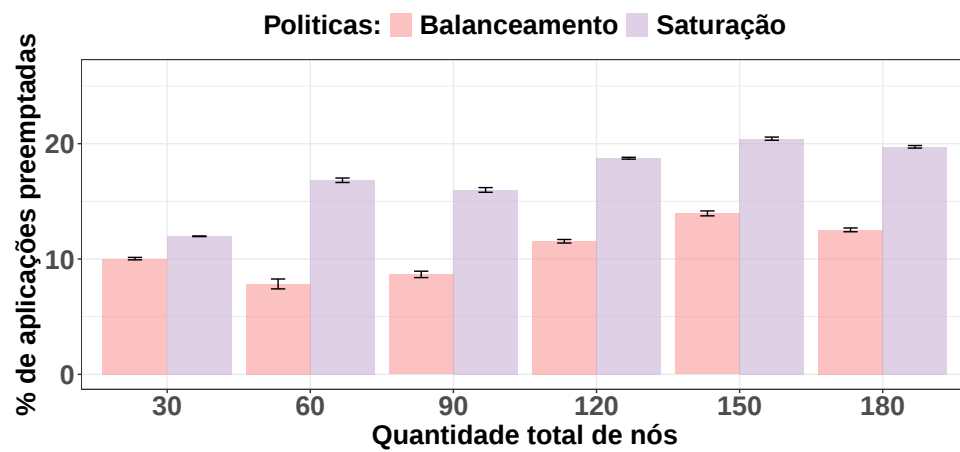
A Figura 10 corrobora que a política de saturação apresenta um percentual maior de aplicações preemptadas, ou seja, aquelas iniciadas em mais de um nó. Para todos os níveis de consumo de energia, 10% (a), 30% (b) e 50% (c), a política de saturação de recursos apresenta uma média maior de aplicações preemptadas do que a política de balanceamento de carga, variando na faixa de aproximadamente 4 a 25%. Como a primeira é mais restritiva e busca sobrecarregar o mínimo de nós possível, esses nós consomem energia mais rapidamente e precisam preemptar as aplicações novamente. A segunda também apresenta um número significativo de aplicações preemptadas, com médias na faixa de 2 a 20%.

Um resultado significativo é o aumento percentual no tempo total de execução, conforme mostrado na Figura 11. A política de balanceamento de carga apresenta consistentemente

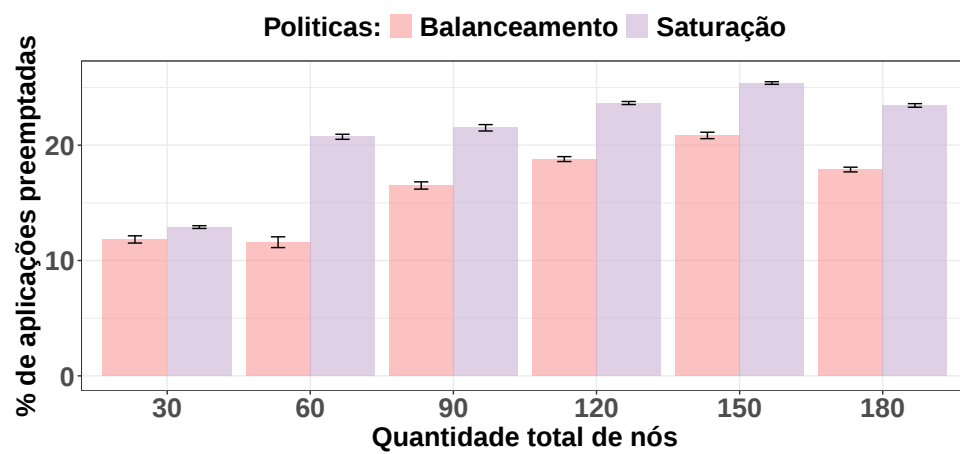
aumentos menores, resultando em tempos totais de execução mais próximos daqueles observados em cenários sem preempção de aplicações. Para esta política, o aumento médio do tempo total de execução varia de aproximadamente 1 a 7%. Esses valores traduzem-se em um aumento médio do tempo total de execução de 1% com 10% de consumo de energia, 3,2% com 30% de consumo e 4,7% com 50% de consumo. Por outro lado, os resultados com a política de saturação variam de aproximadamente 1% a 11%. Esses valores traduzem-se em um aumento médio do tempo total de execução de 2% com 10% de nós com consumo de energia, 5,8% com 30% de nós com consumo de energia e 8,1% com 50% de nós com consumo de energia. Atribuímos esses resultados à sobrecarga introduzida no sistema devido ao consumo de energia por parte dos nós, levando à perda de execuções e à necessidade de reexecução das aplicações.



(a) 10% de consumo de energia.

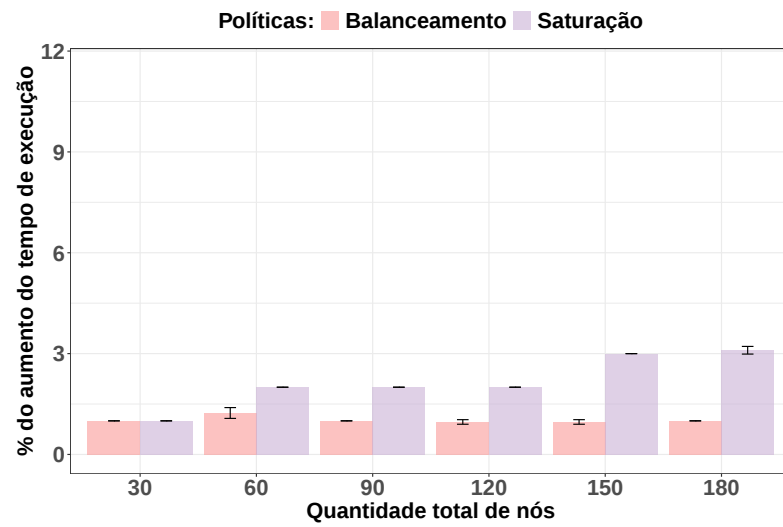


(b) 30% de consumo de energia.

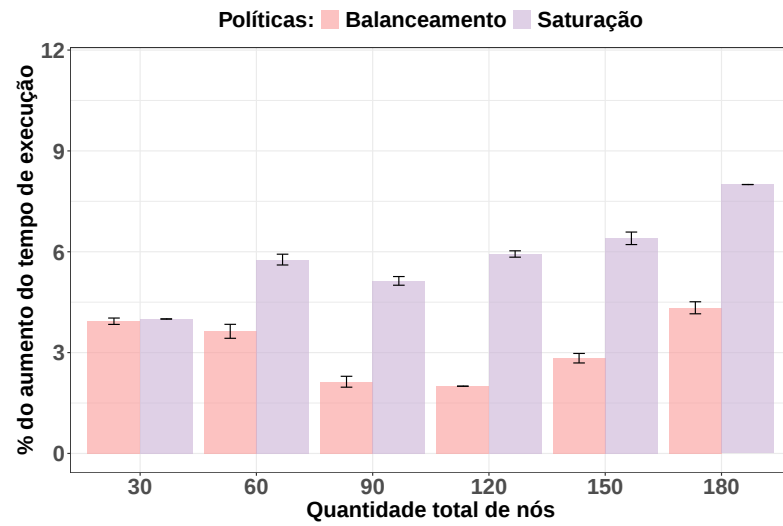


(c) 50% de consumo de energia.

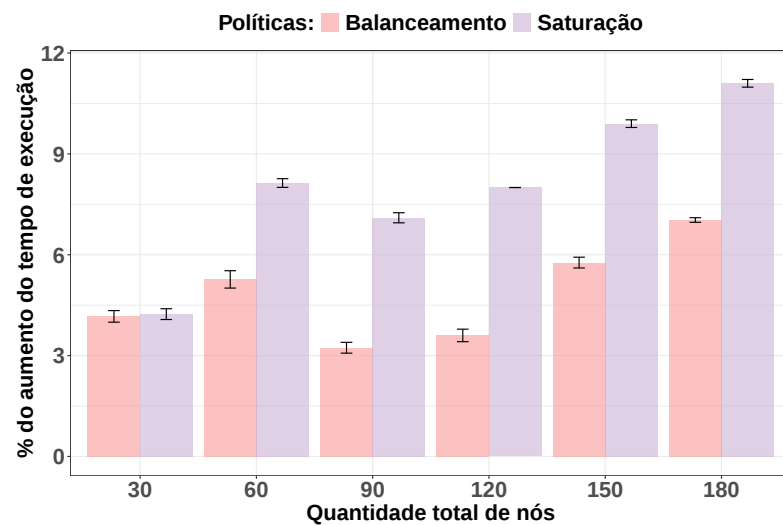
Figura 10 – Resultados para a porcentagem de aplicações preemptadas (Eq. 3).



(a) 10% de consumo de energia.



(b) 30% de consumo de energia.



(c) 50% de consumo de energia.

Figura 11 – Resultados para a percentagem de aumento do tempo total de execução (Eq. 4).

5 CONCLUSÕES E PERSPECTIVAS FUTURAS

5.1 Conclusões

Este trabalho apresentou um esquema de gerenciamento e alocação de recursos em ambientes IoT multifuncionais, fundamentado em políticas hierárquicas de decisão. Diferentemente de abordagens tradicionais que consideram apenas métricas isoladas, como latência ou consumo energético, a solução proposta integrou simultaneamente os requisitos das aplicações, as especificações de hardware dos nós e a adoção de políticas de gerenciamento. A implementação em simulador e a validação por meio de diferentes cenários comprovaram a viabilidade técnica do esquema e evidenciaram ganhos relevantes em termos de alocação justa e aproveitamento da infraestrutura heterogênea.

Os resultados mostraram que a abordagem proposta garante um elevado percentual de alocação eficiente, mesmo em cenários adversos de perda de energia. As políticas de *balanceamento* e *saturação*, embora distintas em seus objetivos, revelaram-se complementares: a primeira maximizando a utilização global da rede, enquanto a segunda priorizou a adequação das aplicações aos dispositivos mais apropriados. O impacto da falha de nós por esgotamento da bateria também foi analisado, demonstrando que a resiliência do sistema depende de mecanismos adicionais de realocação e de estratégias energéticas mais sofisticadas.

Apesar das contribuições alcançadas, algumas limitações foram identificadas. A simulação abstraiu aspectos de mobilidade, topologias dinâmicas e cenários de larga escala que são comuns em cidades inteligentes. Além disso, o mecanismo de decisão foi avaliado com um conjunto restrito de políticas, não incorporando critérios mais complexos, como custo financeiro, privacidade dos dados ou impacto ambiental.

5.2 Perspectivas Futuras

Diante dos resultados e limitações observados, destacam-se as seguintes direções para a continuidade da pesquisa:

- **Políticas híbridas e adaptativas:** explorar combinações dinâmicas entre balanceamento e saturação, ajustando-se de forma autônoma às condições da rede e ao perfil das aplicações;
- **Integração com energias renováveis e recarga dinâmica:** modelar nós capazes de recarregar suas baterias (por exemplo, via energia solar), de modo a reduzir os impactos da perda de dispositivos;

- **Escalabilidade e mobilidade:** validar o esquema em cenários de larga escala com milhares de dispositivos, incorporando também a mobilidade (veículos, drones, usuários móveis);
- **Segurança e privacidade:** estender o modelo de políticas para considerar requisitos de segurança, criptografia e confidencialidade de dados, cada vez mais relevantes em ambientes urbanos inteligentes;
- **Integração com *edge/fog computing*:** investigar como o esquema proposto pode interagir de forma cooperativa com arquiteturas de borda e névoa, equilibrando a utilização da infraestrutura local com o descarregamento seletivo de tarefas;
- **Implementação prática em protótipos reais:** além da simulação, aplicar o esquema proposto em ambientes de teste com dispositivos reais, avaliando sua eficácia frente às restrições físicas, à heterogeneidade de hardware e a falhas imprevisíveis.

Em síntese, o trabalho demonstrou que políticas orientadas às características das aplicações e às especificações dos dispositivos oferecem um caminho promissor para o gerenciamento eficiente de recursos em IoT multifuncional. Os desdobramentos futuros aqui indicados têm potencial para consolidar ainda mais a proposta, ampliando sua aplicabilidade em cenários reais de cidades inteligentes.

REFERÊNCIAS BIBLIOGRÁFICAS

ALI, U.; CALIS, C. Centralized Smart Governance Framework Based on IoT Smart City Using TTG-Classified Technique. In: *the 16th IEEE International Conference on Smart Cities*. [S.l.: s.n.], 2019. p. 157–160. Citado 4 vezes nas páginas 19, 22, 25 e 26.

ALZAHIRANI, B.; EJAZ, W. Resource management for cognitive iot systems with rf energy harvesting in smart cities. *IEEE Access*, v. 6, p. 62717–62727, 2018. Citado na página 14.

ANDAL, C. K.; JAYAPAL, R. Design and implementation of IoT based intelligent energy management controller for PV/wind/battery system with cost minimization. *Renewable Energy Focus*, v. 43, p. 255–262, 2022. ISSN 1755-0084. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1755008422000849>>. Citado 4 vezes nas páginas 19, 20, 25 e 26.

BAHGA, A.; MADISETTI, V. *Internet of Things: A hands-on approach*. [S.l.]: Vpt, 2014. Citado na página 15.

BASHIR, H. et al. Resource allocation through logistic regression and multicriteria decision making method in IoT fog computing. *Trans. on Emerging Telecom. Technologies*, v. 33, n. 2, 2022. Citado 4 vezes nas páginas 19, 23, 25 e 26.

BOLETTIERI, S. et al. Application-aware resource allocation and data management for MEC-assisted IoT service providers. *Journal of Network and Computer Applications*, v. 181, p. 103020, 2021. ISSN 1084-8045. Citado 4 vezes nas páginas 19, 21, 25 e 26.

CALDERONI, L.; MAGNANI, A.; MAIO, D. Iot manager: An open-source iot framework for smart cities. *Journal of Systems Architecture*, v. 98, p. 413–423, 2019. ISSN 1383-7621. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1383762118306520>>. Citado 4 vezes nas páginas 19, 22, 25 e 26.

CATLETT, C. et al. Measuring cities with software-defined sensors. *Journal of Social Computing*, v. 1, n. 1, p. 14–27, 2020. Citado na página 15.

CATLETT, C. et al. Hands-on computer science: The array of things experimental urban instrument. *Computing in Science & Engineering*, v. 24, n. 1, p. 57–63, 2022. Citado 2 vezes nas páginas 15 e 27.

CHELLADURAI, A. et al. Multi-joint symmetric optimization approach for unmanned aerial vehicle assisted edge computing resources in internet of things-based smart cities. *Symmetry*, v. 17, n. 4, 2025. ISSN 2073-8994. Disponível em: <<https://www.mdpi.com/2073-8994/17/4/574>>. Citado 4 vezes nas páginas 19, 24, 25 e 26.

CIMINO, G.; DEUFEMIA, V. Sigfrid: Unsupervised, platform-agnostic interference detection in iot automation rules. *ACM Trans. Internet Things*, Association for Computing Machinery, New York, NY, USA, v. 6, n. 2, abr. 2025. Disponível em: <<https://doi.org/10.1145/3722231>>. Citado 4 vezes nas páginas 19, 23, 24 e 26.

CLARINDO, J. P.; CASTRO, J. P. C.; AGUIAR, C. D. Combining Fog and Cloud Computing to Support Spatial Analytics in Smart Cities. *Journal of Information and Data Management*, v. 12,

n. 4, Oct. 2021. Disponível em: <<https://sol.sbc.org.br/journals/index.php/jidm/article/view/1798>>. Citado 3 vezes nas páginas 21, 25 e 26.

FIROUZI, F. et al. Iot fundamentals: Definitions, architectures, challenges, and promises. In: _____. *Intelligent Internet of Things: From Device to Fog and Cloud*. Cham: Springer International Publishing, 2020. p. 3–50. ISBN 978-3-030-30367-9. Disponível em: <https://doi.org/10.1007/978-3-030-30367-9_1>. Citado na página 12.

GONÇALVES, N. M. F.; SANTOS, A. L. d.; HARA, C. S. A policy-based storage model for sensor networks. In: *2014 IEEE Network Operations and Management Symposium (NOMS)*. [S.l.: s.n.], 2014. p. 1–8. Citado na página 13.

GUEBSI, R.; MAMI, S.; CHOKMANI, K. Drones in precision agriculture: A comprehensive review of applications, technologies, and challenges. *Drones*, v. 8, n. 11, 2024. ISSN 2504-446X. Disponível em: <<https://www.mdpi.com/2504-446X/8/11/686>>. Citado na página 15.

GUIM, F.; METSCH, T. et al. Autonomous Lifecycle Management for Resource-Efficient Workload Orchestration for Green Edge Computing. *IEEE Transactions on Green Communications and Networking*, v. 6, n. 1, p. 571–582, 2022. Citado 4 vezes nas páginas 19, 20, 24 e 26.

KHATOUN, R.; ZEADALLY, S. Smart cities: Concepts, architectures, research opportunities. *Communications of the ACM*, v. 59, p. 46–57, 07 2016. Citado na página 12.

KHODADADI, F.; DASTJERDI, A.; BUYYA, R. Chapter 1 - internet of things: an overview. In: BUYYA, R.; Vahid Dastjerdi, A. (Ed.). *Internet of Things*. Morgan Kaufmann, 2016. p. 3–27. ISBN 978-0-12-805395-9. Disponível em: <<https://www.sciencedirect.com/science/article/pii/B9780128053959000010>>. Citado na página 15.

KLEIN, T.; ANDEREGG, W. R. A vast increase in heat exposure in the 21st century is driven by global warming and urban population growth. *Sustainable Cities and Society*, v. 73, p. 103098, 2021. ISSN 2210-6707. Citado na página 13.

LEAL, S. T.; SANTOS, R. D. d. A. M. e. Cidades inteligentes: inovação e sustentabilidade como respostas aos desafios da urbanização contemporânea. *Caderno Pedagógico*, v. 22, n. 9, p. e18221, jul. 2025. Disponível em: <<https://ojs.studiespublicacoes.com.br/ojs/index.php/cadped/article/view/18221>>. Citado na página 16.

LI, X.; ZHAO, L. et al. A cooperative resource allocation model for IoT applications in mobile edge computing. *Computer Communications*, v. 173, p. 183–191, 2021. ISSN 0140-3664. Citado 4 vezes nas páginas 19, 23, 25 e 26.

LV, Z.; HU, B.; LV, H. Infrastructure monitoring and operation for smart cities based on iot system. *IEEE Transactions on Industrial Informatics*, v. 16, n. 3, p. 1957–1962, 2020. Citado 4 vezes nas páginas 19, 22, 25 e 26.

MUKHERJEE, B. K. et al. An SDN Based Distributed IoT Network with NFV Implementation for Smart Cities. In: *Cyber Security and Computer Science*. Cham: Springer International Publishing, 2020. p. 539–552. ISBN 978-3-030-52856-0. Citado 4 vezes nas páginas 19, 22, 25 e 26.

NIKPOUR, M. et al. *Intelligent Energy Management with IoT Framework in Smart Cities Using Intelligent Analysis: An Application of Machine Learning Methods for Complex Networks and Systems*. 2023. <<https://arxiv.org/abs/2306.05567>>. Citado 4 vezes nas páginas 19, 20, 25 e 26.

PAPAIIOANNOU, C. et al. Hierarchical resources management system for internet of things-enabled smart cities. *Sensors*, v. 25, n. 3, 2025. ISSN 1424-8220. Disponível em: <<https://www.mdpi.com/1424-8220/25/3/616>>. Citado 4 vezes nas páginas 19, 23, 25 e 26.

PEDROSO, C. et al. Relational Consensus-Based Cooperative Task Allocation Management for IIoT-Health Networks. In: *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. [S.l.: s.n.], 2021. p. 579–585. Citado na página 16.

PERERA, A. et al. Light-based Internet of Things: Implementation of an Optically Connected Energy-autonomous Node. In: *2021 IEEE Wireless Communications and Networking Conference (WCNC)*. [S.l.: s.n.], 2021. p. 1–7. Citado na página 13.

RAFIQUE, W. et al. A blockchain-based framework for information security in intelligent transportation systems. In: *Intelligent Technologies and Applications*. Singapore: Springer Singapore, 2020. p. 53–66. ISBN 978-981-15-5232-8. Citado 4 vezes nas páginas 19, 22, 25 e 26.

RILEY, G. F.; HENDERSON, T. R. The ns-3 network simulator. In: _____. *Modeling and Tools for Network Simulation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 15–34. ISBN 978-3-642-12331-3. Disponível em: <https://doi.org/10.1007/978-3-642-12331-3_2>. Citado na página 17.

ROSCIA, M.; LONGO, M.; LAZAROIU, G. C. Smart city by multi-agent systems. In: *2013 International Conference on Renewable Energy Research and Applications (ICRERA)*. [S.l.: s.n.], 2013. p. 371–376. Citado na página 16.

SANGAIAH, A. K. et al. IoT Resource Allocation and Optimization Based on Heuristic Algorithm. *Sensors*, v. 20, n. 2, 2020. ISSN 1424-8220. Citado 4 vezes nas páginas 19, 20, 25 e 26.

SOUSA, N. R.; PINTO, G. P.; PRAZERES, C. V. S. Internet of things devices management for smart cities. In: *INSTICC. Proceedings of the 10th International Conference on Internet of Things, Big Data and Security - IoTBDS*. [S.l.]: SciTePress, 2025. p. 15–26. ISBN 978-989-758-750-4. ISSN 2184-4976. Citado 4 vezes nas páginas 19, 24, 25 e 26.

TAURION, C. *Big data*. [S.l.]: Brasport, 2013. Citado na página 16.

TSAI, C.-W. SEIRA: An effective algorithm for IoT resource allocation problem. *Computer Communications*, v. 119, p. 156–166, 2018. ISSN 0140-3664. Citado 3 vezes nas páginas 19, 24 e 26.

United Nations. *Department of Economic and Social Affairs, Population Division (2019)*. United Nations, New York, USA, 2019. Citado na página 13.

WANG, Z. et al. Computation offloading and resource allocation based on distributed deep learning and software defined mobile edge computing. *Computer Networks*, v. 205, p. 108732, 2022. ISSN 1389-1286. Citado 4 vezes nas páginas 19, 21, 25 e 26.

XAVIER, T. C. et al. Collaborative resource allocation for Cloud of Things systems. *Journal of Network and Computer Applications*, v. 159, p. 102592, 2020. ISSN 1084-8045. Citado 4 vezes nas páginas 19, 21, 25 e 26.

XAVIER, T. C. et al. Managing Heterogeneous and Time-Sensitive IoT Applications through Collaborative and Energy-Aware Resource Allocation. *ACM Trans. Internet Things*, Association for Computing Machinery, New York, NY, USA, v. 3, n. 2, feb 2022. ISSN 2691-1914. Citado 4 vezes nas páginas 19, 20, 25 e 26.

ZHAO, L.; WANG, J. et al. Optimal Edge Resource Allocation in IoT-Based Smart Cities. *IEEE Network*, v. 33, n. 2, p. 30–35, 2019. Citado 4 vezes nas páginas 19, 20, 24 e 26.

APÊNDICE A – ESTRUTURA DO CÓDIGO DO POSITRON (BASEADO EM NS-3)

A.1 Visão Geral do Repositório

O repositório `ifpb/positron` contém uma versão do simulador *NS-3* estendida com o **POSITRON – Efficient Allocation Scheme for Multifunctional IoT Devices**. A implementação foi validada sobre o *NS-3 v3.38*, em ambiente *Ubuntu 22.04*, com suporte às bibliotecas *SQLite3* e *yaml-cpp*, utilizadas para o armazenamento dos dados e a parametrização dos experimentos.

Após a compilação, a execução é realizada a partir do diretório `scratch`, com cenários descritos no arquivo `main.cc` e parametrizados por meio do arquivo de configuração `input.yaml`, por exemplo:

```
./ns3 run main
```

A.2 Organização de Diretórios e Arquivos Relevantes

- `scratch/`: contém o ponto de entrada principal da simulação (`main.cc`) e cenários organizados por número de nós (`30nodes/main.cc`, `60nodes/main.cc`, etc.). Inclui também o arquivo de configuração `input.yaml`.
- `src/applications/helper/control-layer-helper.cc`: módulo que implementa a lógica de controle do POSITRON, incluindo consultas ao banco *SQLite*, seleção de aplicações e inserção de atribuições (*assignments*).
- `src/network/model/node.{h,cc}`: arquivos do modelo *Node* do *NS-3* alterados para contabilização e integração das rotinas de alocação do POSITRON.
- `scripts/dependencies.sh`: script para instalação de dependências (*libyaml-cpp-dev*, *sqlite3*, *toolchain* de compilação), garantindo a reprodutibilidade do ambiente.
- `README.md`: documentação com instruções de instalação, compilação e execução dos cenários POSITRON.

A.3 Evolução Estrutural a partir dos *Commits* POSITRON

A seguir, os principais *commits* iniciados com **POSITRON** e suas contribuições:

1. **POSITRON: initial commit** Introduz a base do POSITRON, incluindo o `control-layer-helper.cc`, alterações em `node.{h,cc}` e integração com SQLite.
2. **POSITRON: updating .gitignore scratch folder** Ajusta o versionamento de arquivos em `scratch/`, mantendo apenas fontes essenciais.
3. **POSITRON: scripts folder and README update** Atualiza o README com instruções de uso do script `dependencies.sh`.
4. **POSITRON: final migration to ns-3.38** Adapta o código à versão 3.38 do NS-3, consolidando cenários em `scratch/`.
5. **POSITRON: code update and adding new results** Atualiza código e adiciona resultados experimentais.
6. **POSITRON: changes for JISA submission** Ajustes de código e cenários para submissão ao periódico JISA.
7. **POSITRON: updating README.md / english version of README.md** Atualização da documentação em português e inglês.
8. **POSITRON: updating dependencies.sh** Iterações no script de dependências, incluindo pacotes adicionais e ajustes.

A.4 Exemplos de Código

A seguir, um trecho simplificado do arquivo `scratch/main.cc` é apresentado, responsável por iniciar os parâmetros de simulação:

Listing A.1 – Trecho do `scratch/main.cc`

```

1  int main (int argc, char *argv[])
2  {
3      CommandLine cmd;
4      cmd.AddValue("seed", "Random seed", seed);
5      cmd.AddValue("logging", "Enable logging", logging);
6      cmd.AddValue("tracing", "Enable tracing", tracing);
7      cmd.Parse(argc, argv);
8
9      // Carrega parametros a partir de input.yaml
10     SimulationParameters params("scratch/input.yaml");
11
12     // Execucao do POSITRON
13     Ptr<SimulationController> controller = CreateObject<SimulationController>
        ();

```

```
14     controller->Run(params);
15
16     Simulator::Run ();
17     Simulator::Destroy ();
18     return 0;
19 }
```

Outro exemplo a seguir, ilustrando a integração com SQLite no arquivo `control-layer-helper.cc`.

Listing A.2 – Trecho do `control-layer-helper.cc`

```
1 void ControlLayerHelper::InsertAssignment(int workerId, int appId) {
2     std::string sql = "INSERT INTO worker_application (worker, app) VALUES (?,
3         ?)";
4     sqlite3_stmt *stmt;
5     sqlite3_prepare_v2(db, sql.c_str(), -1, &stmt, nullptr);
6     sqlite3_bind_int(stmt, 1, workerId);
7     sqlite3_bind_int(stmt, 2, appId);
8     sqlite3_step(stmt);
9     sqlite3_finalize(stmt);
10 }
```

A.5 Execução e Parametrização

1. Instalar dependências:

```
bash scripts/dependencies.sh
```

2. Configurar e compilar:

```
./ns3 configure --enable-examples --enable-tests
./ns3 build
```

3. Executar um cenário:

```
./ns3 run scratch/main.cc
```

Alterando parâmetros no arquivo `input.yaml`.