DISSERTAÇÃO DE MESTRADO PROFISSIONAL

Proposta de Arquitetura para Melhoria do Inventário de Hardware e Software em Sistemas IoT

Welber Santos de Oliveira

Brasília, Outubro de 2025

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

DISSERTAÇÃO DE MESTRADO PROFISSIONAL

Proposta de Arquitetura para Melhoria do Inventário de Hardware e Software em Sistemas IoT

Welber Santos de Oliveira

Dissertação de Mestrado Profissional submetida ao Departamento de Engenharia Elétrica como requisito parcial para obtenção do grau de Mestre em Engenharia Elétrica

Banca Examinadora

Prof Dr. Georges Daniel Amvame Nze, ENE/UnB _____
Presidente Orientador

Prof Dr. Fábio Lúcio Lopes de Mendonça, ENE/UnB _____
Examinador Interno

Prof Dr. Jairo Simão Santana Melo, DTI/CNJ _____
Examinador Externo

Prof Dr. Daniel Alves da Silva, PPEE/UnB _____
Suplente

FICHA CATALOGRÁFICA

OLIVEIRA, WELBER
Proposta de Arquitetura para Melhoria do Inventáriode Hardware e Software em Sistemas IoT [Distrito Federal] 2025.
xvi, 82 p., 210 x 297 mm (ENE/FT/UnB, Mestre, Engenharia Elétrica, 2025).
Dissertação de Mestrado Profissional - Universidade de Brasília, Faculdade de Tecnologia.
Departamento de Engenharia Elétrica

1. Inventário de Ativos de TI	2. Segurança da Informação
3. Protocolos de Descoberta de Dispositivos	4. Internet das Coisas (IoT)
5. Software de Código Aberto	6. Gestão Automatizada de Inventário
I. ENE/FT/UnB	II. Título (série)

PUBLICAÇÃO: PPEE.MP.099

REFERÊNCIA BIBLIOGRÁFICA

OLIVEIRA, WELBER (2025). *Proposta de Arquitetura para Melhoria do Inventáriode Hardware e Software em Sistemas IoT*. Dissertação de Mestrado Profissional, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 82 p.

CESSÃO DE DIREITOS

AUTOR: Welber Santos de Oliveira
TÍTULO: Proposta de Arquitetura para Melhoria do Inventáriode Hardware e Software em Sistemas IoT.
GRAU: Mestre em Engenharia Elétrica ANO: 2025
PUBLICAÇÃO: PPEE.MP.099

Welber Santos de Oliveira
Depto. de Engenharia Elétrica (ENE) - FT
Universidade de Brasília (UnB)
Campus Darcy Ribeiro
CEP 70919-970 - Brasília - DF - Brasil

DEDICATÓRIA

Dedico este trabalho, com todo o meu amor e gratidão, à minha mãe,(in memoria) mulher forte e incansável, que foi pai e mãe em todos os momentos da minha vida. Sua dedicação, coragem e amor incondicional foram e sempre serão o alicerce das minhas conquistas.

A minha querida esposa, companheira, fiel, que durante essa jornada, esteve presente nos momentos mais desafiadores, oferecendo apoio, compreensão e incentivo para que eu nunca desistisse dos meus sonhos e ao meu filho Pedro que chegou durante a execução deste sonho.

Aos meus irmãos, que caminharam ao meu lado nos momentos difíceis e celebraram comigo cada pequena vitória, meu sincero agradecimento.

A cada um de vocês, minha eterna gratidão.

AGRADECIMENTOS

Em primeiro lugar, agradeço a Deus, cuja orientação e sabedoria me sustentaram ao longo de toda esta jornada. Foi com a Sua graça que encontrei forças para superar os desafios e concluir este trabalho.

Ao meu orientador, Professor Dr. Georges Daniel Amvame Nze, expresso minha mais sincera gratidão. Seu apoio constante, desde as primeiras ideias até a finalização deste projeto, foi fundamental. Suas orientações e disponibilidade para esclarecer minhas dúvidas foram cruciais para o sucesso deste trabalho e para meu crescimento acadêmico ao longo do curso.

Manifesto também minha profunda gratidão ao Professor Dr. Fábio Lúcio Lopes de Mendonça, cujos conselhos e ensinamentos foram valiosos desde os tempos de graduação. Sua orientação, tanto no âmbito acadêmico quanto pessoal, foi um alicerce importante para que eu pudesse alcançar esta etapa. Sou imensamente grato por todo o conhecimento e sabedoria compartilhados ao longo dos anos.

Estendo meus agradecimentos ao Professor Dr. Robson de Oliveira Albuquerque, por sua dedicação, incentivo e pelas contribuições significativas ao longo da minha formação. Seu apoio foi essencial para meu amadurecimento acadêmico e profissional.

Agradeço, de maneira especial, ao MsC. Flávio Garcia Praciano, pela ajuda inestimável ao longo da elaboração desta dissertação. Sua colaboração e insights foram fundamentais para o desenvolvimento do meu trabalho.

Aos meus familiares, expresso meu sincero agradecimento por acreditarem em minha capacidade de concluir mais esta fase da minha vida. À minha esposa, sou profundamente grato por seu apoio incondicional, compreensão nos momentos de maior dificuldade e constante incentivo para seguir em frente.

Agradeço, ainda, ao Laboratório LATITUDE/UnB, que proporcionou as condições e recursos necessários para a realização deste trabalho, e ao PPEE, que me permitiu chegar até aqui. Cada projeto e atividade em que estive envolvido ao longo do curso contribuiu de maneira significativa para a minha formação e para a concretização desta dissertação.

RESUMO

Nas últimas décadas, o inventário de hardware consolidou-se como prática essencial para o controle e a gestão de ativos de tecnologia em organizações, permitindo acompanhar a vida útil de equipamentos, realizar auditorias e planejar substituições. No entanto, esses métodos tradicionais foram concebidos em um cenário predominantemente composto por computadores, servidores e periféricos, e não acompanharam o ritmo acelerado da digitalização. Com a popularização da Internet das Coisas (IoT), emergiu um novo desafio: milhares de sensores, dispositivos embarcados e equipamentos inteligentes passaram a se conectar em redes locais heterogêneas sem, contudo, serem plenamente visíveis às ferramentas convencionais de inventário.

Essa lacuna torna o inventário não apenas um processo administrativo, mas um ponto crítico para a segurança e a governança tecnológica. Ignorar dispositivos "invisíveis" significa deixar brechas abertas para riscos cibernéticos e falhas operacionais. É nesse contexto que este trabalho se insere, propondo uma arquitetura de inventário automatizado e seguro, orientada à detecção de dispositivos IoT. A solução amplia a visibilidade sobre ativos conectados, viabiliza a coleta estruturada de dados e possibilita avaliações contínuas de conformidade e segurança.

Os resultados demonstraram um aumento expressivo na taxa de detecção de dispositivos em redes heterogêneas, evidenciando a viabilidade de um inventário mais abrangente e alinhado às demandas atuais. A pesquisa aponta ainda caminhos futuros, como a integração a protocolos industriais (ex.: Modbus, CoAP) e a incorporação a sistemas SIEM para monitoramento em tempo real, reforçando a importância de repensar o inventário não como uma simples listagem de ativos, mas como um mecanismo estratégico de resiliência cibernética.

Palavras-chave: Inventário de hardware, Inventário de software, Melhores práticas, Segurança de rede.

ABSTRACT

Over the past decades, hardware inventory has become an essential practice for managing IT assets, helping organizations track the lifecycle of equipment, plan replacements, and conduct audits. However, traditional inventory methods were designed for environments dominated by computers, servers, and peripheral devices, and they have not kept pace with the rapid digital transformation. With the rise of the Internet of Things (IoT), a new challenge has emerged: thousands of sensors, embedded systems, and smart devices now connect to heterogeneous local networks, yet remain largely invisible to conventional asset management tools.

This invisibility turns inventory management from a purely administrative task into a critical issue for both security and governance. Overlooking these devices means exposing organizations to cyber risks and operational vulnerabilities. Addressing this gap, the present work proposes a secure and automated inventory architecture specifically designed to detect IoT devices. The solution enhances asset visibility, enables structured data collection, and supports continuous security and compliance assessment.

Our results demonstrate a significant improvement in the detection rate of assets across heterogeneous networks, highlighting the feasibility of building inventories that go beyond traditional approaches and embrace a cybersecurity perspective. Furthermore, the study outlines future directions, including compatibility with industrial protocols (e.g., Modbus, CoAP) and integration with SIEM platforms for real-time monitoring. Ultimately, this research argues that inventory should no longer be seen as a simple list of assets, but rather as a strategic mechanism for strengthening cyber resilience in modern IT environments.

Keywords: Hardware Inventory, Software Inventory, Best Practices, Network Security

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SÍMBOLOS

/ - divisão

= - igual

μ - mi

- - negativo

+ - positivo

σ - sigma

LISTA DE SIGLAS

3GPP *3rd Generation Partnership Project*.

ABNT Associação Brasileira de Normas Técnicas.

BLE *Bluetooth Low Energy*.

COAP *Constrained Application Protocol*.

CPU Unidade Central de Processamento.

CRA *Cyber Resilience Act)*.

CSF *Cybersecurity Framework*.

CVE *Common Vulnerabilities and Exposures*.

DDos *Distributed Denial of Service*.

DNS *Domain Name System*.

DTLS *Datagram Transport Layer Security*.

fping *Fast Ping*.

HDD *Hard Disk Drives*.

HTTP *Hypertext Transfer Protocol*.

HTTPS *Hypertext Transfer Protocol Secure*.

IA *Inteligência Artificial*.

ICMP *Internet Control Message Protocol*.

IDS *Intrusion Detection System*.

IEEE *Institute of Electrical and Electronics Engineers*.

IIOT *Industrial Internet of Things*.

IoT *Internet das Coisas*.

IP *Internet Protocol*.

IPS *Intrusion Prevention System*.

IPv4 *Internet Protocol version 4*.

IPV6 *Internet Protocol version 6*.

ISO *International Organization for Standardization*.

LAN Rede de Área Local.

LATITUDE *Laboratório de Tecnologias da Tomada de Decisão*.

LGPD *Lei Geral de Proteção de Dados*.

LTS *Long Term Support*.

MAC *Media Access Control address*.

mDNS *Multicast Domain Name System*.

ML *Machine Learning*.

MQTT *Message Queuing Telemetry Transport*.

NIS2 *Network and Information Security Directive 2*.

NIST *National Institute of Standards and Technolog*.

Nmap *Network Mapper*.

OCS *Open Computer and Software Inventory Next Generation*.

OSI *Open Systems Interconnection*.

oVirt *Open Virtualization*.

PGFN Procuradoria Geral da Fazenda Nacional.

PPEE Programa de Pós-Graduação Profissional em Engenharia Elétrica.

RAM Memória de Acesso Aleatório.

RFID *Radio-Frequency Identification*.

SIEM *Security Information and Event Management*.

SNMP *Simple Network Management Protocol*.

SOAR *Security Orchestration, Automation and Response*.

SSD Unidade de Estado Sólido.

SSDP *Simple Service Discovery Protocol*.

SSH *Secure Shell*.

TCP *Transmission Control Protocol*.

TI Tecnologia da Informação.

TLS *Transport Layer Security*.

UDP *User Datagram Protocol*.

UIoT *UIoT–Universal Internet of Things*.

UnB Universidade de Brasília.

UPnP *Universal Plug and Play*.

WMI *Windows Management Instrumentation*.

1 INTRODUÇÃO

O processo de inventário consolidou-se, desde meados do século XX, como atividade essencial para a gestão de bens em empresas, instituições públicas e mesmo em contextos pessoais. Inicialmente restrito a práticas manuais de levantamento, registro e classificação, evoluiu na década de 1990 para sistemas informatizados capazes de alinhar registros contábeis à realidade física dos ativos, promovendo maior precisão e confiabilidade [1]. Essa evolução atendeu à necessidade de controle patrimonial em ambientes predominantemente compostos por desktops, servidores e periféricos tradicionais.

Com o avanço da digitalização a partir dos anos 2000, os inventários passaram a integrar códigos de barras, RFID e sistemas de auditoria automatizados, reforçando seu papel estratégico para o planejamento e a governança organizacional [2]. Contudo, esse modelo mostrou-se limitado diante da ascensão da Internet das Coisas (IoT) na última década, quando milhares de sensores, câmeras, atuadores e dispositivos embarcados passaram a se conectar em redes heterogêneas. A complexidade trazida por essa nova onda tecnológica ampliou a superfície de ataque e introduziu riscos de segurança que os inventários tradicionais não estavam preparados para endereçar.

No cenário contemporâneo (2020–2025), a gestão de ativos de Tecnologia da Informação (TI) e a segurança cibernética tornaram-se prioridades globais. Organizações enfrentam o desafio de proteger dados e sistemas em redes onde dispositivos IoT, muitas vezes limitados em capacidade de processamento e atualização, representam pontos vulneráveis de difícil rastreabilidade. A ausência de inventários abrangentes e atualizados compromete a visibilidade e abre espaço para ameaças explorarem lacunas de segurança ainda pouco exploradas pela literatura especializada.

Nesse contexto, emergem propostas que combinam melhores práticas de governança com softwares de código aberto, visando construir inventários mais robustos, dinâmicos e seguros. A adoção dessas práticas não se restringe ao controle administrativo, mas avança para o fortalecimento da resiliência cibernética, alinhando-se a diretrizes internacionais de segurança e conformidade. Assim, este estudo propõe uma arquitetura inovadora para o inventário de hardware e software em ambientes IoT, explorando protocolos de descoberta como mDNS e SSDP/UPnP, aliados a métodos tradicionais (ICMP e Nmap), de modo a superar as limitações identificadas no estado da arte e preencher uma lacuna científica relevante: a ausência de soluções integradas que unifiquem visibilidade, segurança e conformidade regulatória em inventários de IoT [3].

Dessa forma, a contribuição desta dissertação não se restringe ao desenvolvimento técnico de um protótipo no Laboratório *UIoT–Universal Internet of Things* (UIoT) / *Laboratório de Tecnologias da Tomada de Decisão* (LATITUDE), mas à consolidação de uma proposta científica que amplia o entendimento sobre como inventários podem evoluir de mecanismos administrativos para instrumentos estratégicos de resiliência cibernética em ecossistemas digitais contemporâneos.

1.1 MOTIVAÇÃO

A rápida expansão da *Internet das Coisas* (IoT). tem transformado a infraestrutura tecnológica das organizações, introduzindo uma grande quantidade de dispositivos conectados, como sensores, atuadores, câmeras, wearables e equipamentos inteligentes. Esses ativos, muitas vezes distribuídos e com capacidades de comunicação autônoma, representam novos desafios para o controle patrimonial e a segurança da informação. Assim, há uma necessidade urgente de desenvolver e implementar práticas mais modernas, alinhadas com os princípios da automação, integração e cibersegurança.

Neste sentido, a motivação deste estudo está alicerçada na crescente demanda por melhorias na segurança e no controle das infraestruturas de IoT de forma que com o avanço da complexidade e da sofisticação das ameaças cibernéticas no cenário global, torna-se indispensável uma gestão eficaz de ativos de hardware e software para proteger informações sensíveis e assegurar a continuidade das operações nesse tipo de ambiente.

Este trabalho, apresenta a utilização de ferramentas de código aberto para inventário e monitoramento como uma abordagem eficaz não apenas para fortalecer a segurança cibernética, mas também para otimizar a administração dos recursos tecnológicos, garantindo conformidade com políticas de segurança da informação e promovendo maior eficiência operacional. A adoção de soluções open-source destaca um enfoque estratégico que prioriza, além da segurança, a sustentabilidade e a redução de custos.

A relevância deste estudo abrange tanto os dispositivos de redes IoT quanto os softwares associados (*middleware*). A ideia principal é propõr um modelo de referência através de uma arquitetura que combina inovação tecnológica com boas práticas de segurança da informação, estabelecendo uma base sólida para mitigação de riscos e aprimoramento contínuo das infraestruturas de IoT.

Portanto, melhorar os processos de inventário nesses ambientes é essencial não apenas para manter a visibilidade e o controle operacional, mas também para garantir a segurança, a continuidade do negócio e a conformidade com padrões internacionais, promovendo uma gestão de Tecnologia da Informação (TI) mais inteligente e responsiva.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

O objetivo geral deste trabalho é desenvolver uma arquitetura inovadora para o inventário de hardware e software em ambientes de IoT, assegurando maior rastreabilidade, precisão e eficiência na gestão dos ativos.

1.2.2 Objetivos Específicos

Para alcançar o objetivo geral da proposta, foram elencados os seguintes objetivos específicos, que garantem um aprofundamento metodológico e aplicado na proposta de arquitetura apresentada.

- Mapear o estado atual dos processos de inventário de hardware e software em ambientes com dispositivos IoT.
- Identificar fragilidades e lacunas relacionadas à rastreabilidade, atualização e controle de ativos tecnológicos em sistemas IoT.
- Investigar boas práticas e frameworks adotados na gestão de inventário para ambientes inteligentes e conectados.
- Propor um modelo otimizado de inventário que incorpore práticas de automação, segurança e integração com plataformas IoT.
- Como prova de conceito, testar e avaliar a eficácia do modelo proposto por meio de um estudo de caso ou simulação prática.

1.3 PRINCIPAIS CONTRIBUIÇÕES

Esta dissertação alcançou tanto contribuições técnicas quanto científicas com base em uma abordagem aplicada e exploratória, dividida nas seguintes etapas:

- Revisão bibliográfica sobre inventário de TI, gestão de ativos, práticas em ambientes IoT e padrões internacionais (como ITIL, ISO/IEC 19770).
- Estudo de caso junto ao Laboratótio Latitude através de um ambiente simulado com uso intensivo de dispositivos IoT.
- Diagnóstico das principais falhas e oportunidades de melhoria, com base em critérios de segurança, desempenho e escalabilidade.
- Desenvolvimento e validação de um modelo ou proposta de boas práticas, utilizando ferramentas de automação de inventário, como scripts de monitoramento, plataformas IoT e dashboards de visualização.
- Análise dos resultados, com base em indicadores como tempo de atualização, taxa de erro, cobertura do inventário e feedback de usuários

1.4 TRABALHOS PUBLICADOS

Durante o período de execução do mestrado, foram aceitos um total de três trabalhos, dos quais dois são artigos como autor principal e um como coautora. A seguir, serão elencados os trabalhos publicados.

- [4] SANTOS, Welber Oliveira.; MENDONCA, Fabio L. L. ; AMVAME-NZE, Georges. HARDWARE AND SOFTWARE INVENTORY BEST PRACTICES APPLIED TO GOVERNMENT COMPUTER NETWORK AND SYSTEMS. In: The 22nd International Conference e-Society (ES 2024),

2024, Porto, Portugal. HARDWARE AND SOFTWARE INVENTORY BEST PRACTICES APPLIED TO GOVERNMENT COMPUTER NETWORK AND SYSTEMS. Porto, Portugal: E-SOCIETY, 2024. v. 1. p. 191-198.

- [5] Welber Santos de Oliveira, Heitor Gomes Pereira, Vinicius Pereira Gonçalves, Diego Oliveira Martins, Lucimar Rizzo Lopes dos Santos, Georges Daniel Amvame Nze. ARQUITETURA PARA RASTREAMENTO DE PATRIMÔNIO DE ATIVOS DE REDES COM TECNOLOGIAS SDN E IOT. In: 20ª Conferência Ibero Americana WWW/INTERNET (CIAWI), 2023.

- [6] Coelho, J., Bispo, G., Vergara, G., Saiki, G., Serrano, A., Weigang, L., Neumann, C., Martins, P., Santos de Oliveira, W., Albarello, A., Casonatto, R., Missel, P., Medeiros Junior, R., Gomes, J., Rosano-Peña, C. and F. da Costa, C. (2023). Enhancing Industrial Productivity Through AI-Driven Systematic Literature Reviews. In Proceedings of the 19th International Conference on Web Information Systems and Technologies - WEBIST; ISBN 978-989-758-672-9; ISSN 2184-3252, SciTePress, pages 472-479. DOI: 10.5220/0012235000003584

Os artigos [4] e [5] estão ligados diretamente ao tema dessa dissertação. Já os artigo [6] foi publicado no início das pesquisas e está relacionado a uma pesquisas de Artificial Intelligence (AI) e Natural Language Processing (NLP), assuntos importantes para essa dissertação.

1.5 ESTRUTURA DA DISSERTAÇÃO

A estrutura deste trabalho está organizada em mais cinco capítulos, além dessa introdução, conforme descritos a seguir.

- Capítulo 2 - Referencial Teórico: Este capítulo apresenta diversos conceitos de tecnologias utilizadas no trabalho.

- Capítulo 3 - Trabalhos Relacionados: Este capítulo apresenta diversas abordagens presentes na literatura que abordam o problema investigado nesta pesquisa, comparando as metodologias e resultados desses trabalhos com o modelo proposto nesta dissertação.

- Capítulo 4 - Solução Proposta: Este capítulo apresenta o modelo de arqitetura com os componentes principais, e as técnicas implementadas para proposta de diagnóstico das principais falhas e oportunidades de melhoria, com base em critérios de segurança.

- Capítulo 5 - Avaliação de Desempenho: Neste capítulo, são descritos o Desenvolvimento e validação de um modelo ou proposta de boas práticas, utilizando ferramentas de automação de inventário, como scripts de monitoramento, plataformas IoT e dashboards de visualização.

- Capítulo 6 - Conclusão e Trabalhos Futuros: Neste capítulo, é apresenta uma síntese dos principais achados da pesquisa, além de apresentar direções para trabalhos futuros.

2 REFERENCIAL TEÓRICO E TRABALHOS CORRELATOS

Este estudo investiga aspectos essenciais relacionados à gestão e segurança em ambientes de Tecnologia da Informação, com ênfase na proteção de redes, no papel dos sistemas operacionais, na utilização de software de código aberto e nas ferramentas voltadas ao monitoramento e inventário de ativos. A segurança de redes é um componente crítico para garantir a integridade, confidencialidade e disponibilidade das informações, com o uso de soluções como firewalls, autenticação e criptografia de dados.

No campo dos sistemas operacionais, destaca-se o uso do Linux, especialmente distribuições como o Ubuntu, amplamente reconhecidas por sua estabilidade, flexibilidade e forte apoio da comunidade open source. Esse ecossistema colaborativo também impulsiona o desenvolvimento de soluções adaptáveis, como o *Open Computer and Software Inventory Next Generation* (OCS), uma ferramenta robusta para o gerenciamento automatizado de ativos de TI.

O presente trabalho parte da implementação e modificação do código-fonte do OCS com o objetivo de ampliar sua capacidade de detecção e inventário de dispositivos periféricos IoT, tradicionalmente não reconhecidos por soluções convencionais. Essa adaptação visa preencher lacunas existentes no controle e rastreabilidade desses dispositivos em redes corporativas. Além disso, reforça-se a importância do monitoramento contínuo e da manutenção de inventários atualizados, não apenas para fins de gestão e desempenho, mas também como mecanismo de conformidade com diretrizes de segurança, como as estabelecidas pelo framework *National Institute of Standards and Technolog* (NIST).

Ao integrar práticas de segurança, automação e personalização de ferramentas open source, este estudo contribui para a construção de um modelo mais eficaz e abrangente de gerenciamento de ativos em ambientes tecnologicamente heterogêneos e cada vez mais conectados.

2.1 ESTRUTURA E FUNCIONALIDADE DO *HARDWARE* EM SISTEMAS COMPUTACIONAIS

O hardware compreende o conjunto de componentes físicos que formam a estrutura de um sistema computacional conforme demonstrado na Figura 2.1, sendo essencial para o seu funcionamento. Esses componentes incluem desde a Unidade Central de Processamento (CPU), que é o cérebro responsável pela execução das instruções de um programa, até dispositivos de armazenamento, como *Hard Disk Drives* (HDD) e Unidade de Estado Sólido (SSD), que armazenam permanentemente os dados e arquivos do sistema. Além disso, a memória Memória de Acesso Aleatório (RAM) desempenha um papel crucial ao fornecer acesso temporário e rápido aos dados que estão sendo processados em tempo real. Outros elementos importantes incluem a placa-mãe, que serve como plataforma de conexão e comunicação entre os diversos componentes internos, e dispositivos periféricos, como teclados, monitores, impressoras e mouses, que permitem a interação do usuário com o sistema.

A capacidade de processamento, armazenamento e comunicação proporcionada pelo hardware determina o desempenho global de um sistema. Processadores de maior potência, por exemplo, são capazes de realizar múltiplas operações simultâneas com maior velocidade, enquanto grandes quantidades de memória RAM permitem a execução de vários programas de forma fluida e eficiente. Dispositivos de armazenamento de alta capacidade e velocidade asseguram que grandes volumes de dados sejam manipulados de maneira ágil, o que é essencial em tarefas que demandam grande processamento, como renderização gráfica, simulações científicas e análises de dados complexos.

Outro aspecto relevante é a escalabilidade e a adaptabilidade do hardware. Sistemas modernos são frequentemente projetados para serem expansíveis, permitindo a adição de componentes ou a substituição de peças para aumentar a capacidade de processamento ou armazenamento, conforme as necessidades do usuário evoluem. Além disso, a integração de novos dispositivos periféricos, como placas de vídeo de alta performance ou dispositivos de entrada especializados, pode transformar um sistema básico em uma plataforma adequada para funções específicas, como desenvolvimento de software, design gráfico ou análise de big data.

Portanto, o hardware não só fornece a base estrutural e operacional de um sistema computacional, como também determina a sua capacidade de responder a demandas tecnológicas crescentes e a necessidade de executar tarefas complexas.[7] [8]

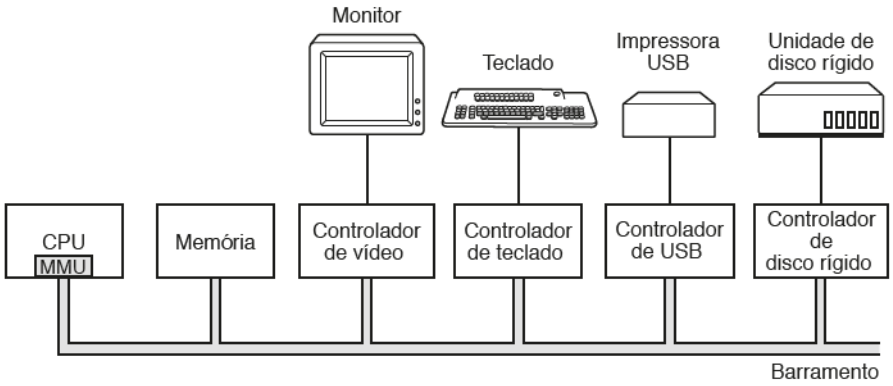

Figura 2.1: Estrutura de um sistema computacional

Fonte: TANENBAUM, A. S.(2024, p. 21) [9]

2.2 FUNÇÃO DO *SOFTWARE* EM SISTEMAS COMPUTACIONAIS

O software refere-se ao conjunto de programas, aplicativos e sistemas operacionais que instruem o hardware a realizar tarefas específicas. Diferentemente do hardware, o software é intangível e consiste em códigos e algoritmos que definem as operações a serem executadas pelos componentes físicos de um sistema computacional. Entre os exemplos de software, destacam-se sistemas operacionais, como Windows e Linux, bem como aplicativos, tais como navegadores de internet, editores de texto e programas de edição de imagem.

A função do software é servir como uma interface entre o usuário e o hardware, permitindo que os

comandos sejam interpretados e executados pela máquina. Nesse contexto, o software desempenha um papel fundamental ao traduzir as ações do usuário em operações que o hardware pode processar. O software pode ser classificado em várias categorias, sendo as mais comuns o software de sistema e o software aplicativo. O software de sistema, que inclui os sistemas operacionais, é responsável por gerenciar os recursos do computador e controlar o funcionamento básico da máquina. Já o software aplicativo abrange programas voltados para o usuário, permitindo a realização de tarefas específicas, como edição de documentos, criação de planilhas, ou navegação na internet. Conforme demonstrado na Figura 2.2.

A eficiência e a funcionalidade de um sistema computacional dependem da integração entre software e hardware. Um software bem projetado, que utiliza os recursos do hardware de forma otimizada, pode melhorar significativamente o desempenho do sistema. Da mesma forma, um hardware potente, quando aliado a um software eficiente, resulta em um ambiente capaz de suportar uma ampla gama de atividades, desde operações simples até tarefas complexas e de alta demanda computacional.[10] [11]

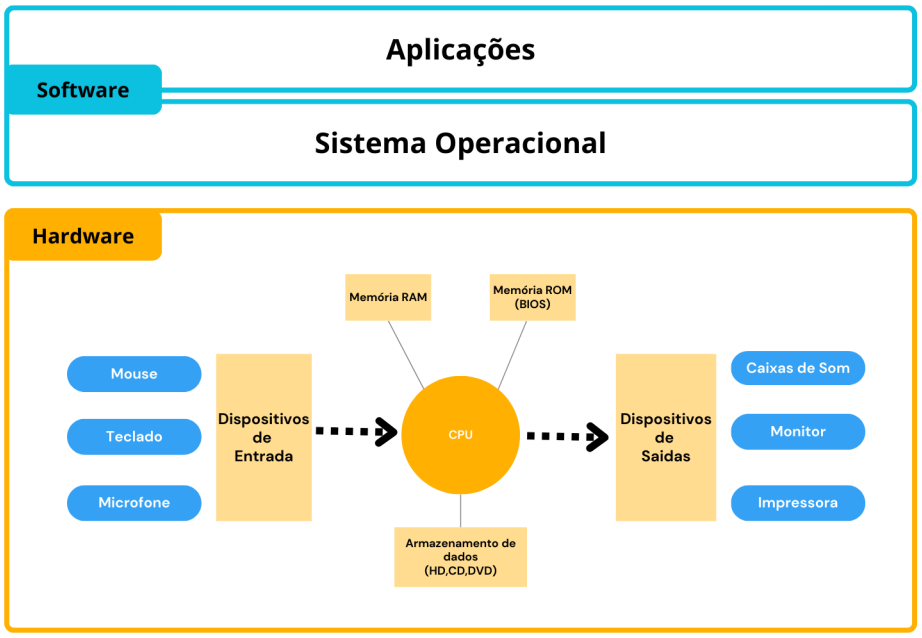

Figura 2.2: Conjuntos de programas que instruem o hardware a realizar tarefas específicas.

Fonte: Elaborado pelo autor (2024)

2.3 IMPORTÂNCIA DAS REDES DE COMPUTADORES NA INFRAESTRUTURA DIGITAL

Redes de computadores constituem sistemas complexos de dispositivos interconectados que facilitam a comunicação e o compartilhamento de recursos, dados e serviços entre diferentes pontos, conforme demonstrado na Figura 2.3. Essas redes podem variar amplamente em termos de escala e complexidade, desde pequenas Rede de Área Local (LAN), que conectam dispositivos em ambientes limitados como escritórios ou residências, até vastas redes globais, como a Internet, que interligam sistemas em todo o mundo. Para

possibilitar e manter essas conexões, são utilizados componentes de hardware, como roteadores, switches e cabos, bem como software especializado, incluindo protocolos de comunicação e ferramentas de gerenciamento de rede.

O objetivo primordial de uma rede de computadores é assegurar a troca eficiente e segura de informações entre usuários e sistemas. Redes bem projetadas e gerenciadas oferecem acesso remoto a recursos, possibilitam o compartilhamento de arquivos e suportam a comunicação em tempo real, o que é crucial para aumentar a produtividade e fomentar a colaboração em ambientes corporativos e pessoais. A integração eficiente dos dispositivos e a implementação de protocolos de segurança robustos são essenciais para garantir a integridade e a confidencialidade dos dados transmitidos.

Além de suas funções básicas de comunicação e compartilhamento, as redes desempenham um papel vital na infraestrutura digital moderna. Elas são a base para uma vasta gama de serviços e aplicações, incluindo e-mail, videoconferências, comércio eletrônico e serviços em nuvem, que são indispensáveis para o funcionamento das atividades econômicas e sociais contemporâneas. A capacidade de uma rede de adaptar-se às necessidades em constante evolução dos usuários e de integrar novas tecnologias é fundamental para a sua longevidade e eficácia. Portanto, a gestão adequada e a constante atualização das redes de computadores são indispensáveis para manter a eficiência operacional e a segurança dos sistemas interconectados.[12] [13]

Figura 2.3: Sistemas complexos de dispositivos interconectados

Fonte: Elaborado pelo autor (2024)

2.4 PROTOCOLOS DE DESCOBERTA DE DISPOSITIVOS EM REDES IOT

A identificação e o mapeamento automatizado de dispositivos em redes heterogêneas, especialmente em ambientes com presença de dispositivos IoT, exigem a utilização de protocolos específicos de descoberta. Tais protocolos têm como função principal localizar, identificar e, em alguns casos, interagir com dispositivos conectados à rede, mesmo que estes não utilizem mecanismos tradicionais de comunicação

cliente-servidor.

Entre os protocolos mais utilizados está o *mDNS (Multicast DNS)*, que permite que dispositivos resolvam nomes de host localmente, sem a necessidade de um servidor DNS central. Isso é particularmente útil em redes domésticas ou ambientes fechados, como laboratórios, onde dispositivos IoT se anunciam automaticamente, facilitando a sua detecção. Já o *SSDP (Simple Service Discovery Protocol)*, amplamente adotado em dispositivos com suporte a *UPnP (Universal Plug and Play)*, permite que serviços em rede sejam descobertos por meio de mensagens multicast enviadas a portas específicas, como a 1900/*User Datagram Protocol* (UDP).

Além desses, outros protocolos de descoberta também são relevantes, como o *ICMP (Internet Control Message Protocol)*, utilizado em comandos como o *ping*, que permite verificar a presença de dispositivos através de respostas à solicitação de eco. Em contextos mais avançados, *CoAP (Constrained Application Protocol)* e *Zigbee Device Discovery* também surgem como alternativas, embora demandem componentes adicionais para integração conforme demonstrado na Figura 2.4.

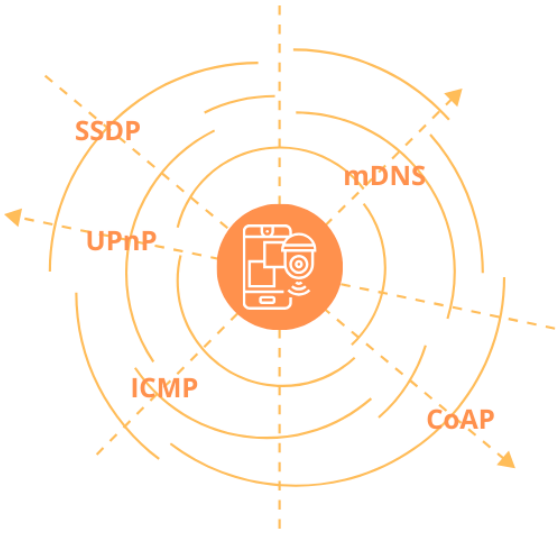

Figura 2.4: Protocolos de comunicação

Fonte: Elaborado pelo autor (2025)

A compreensão desses protocolos é fundamental para o desenvolvimento de arquiteturas de inventário orientadas a ambientes com dispositivos embarcados e recursos limitados, visto que sua implementação adequada pode aumentar significativamente a taxa de detecção e a visibilidade da infraestrutura [14, 15].

2.5 ESTRATÉGIAS E DESAFIOS NA SEGURANÇA DE REDES DE COMPUTADORES

A segurança em redes de computadores é um campo complexo e multifacetado que visa proteger a integridade, confidencialidade e disponibilidade dos dados que circulam por essas redes, conforme demons-

trado na Figura 2.5. Para garantir a segurança, é essencial implementar um conjunto robusto de práticas, políticas e tecnologias. Entre essas medidas, destacam-se a utilização de firewalls, que atuam como barreiras para impedir acessos não autorizados; a criptografia, que codifica dados para proteger informações sensíveis durante a transmissão; e sistemas de autenticação de usuários, que asseguram que apenas pessoas autorizadas possam acessar determinados recursos. O controle de acesso e o monitoramento contínuo do tráfego também são críticos para detectar e prevenir atividades suspeitas e ataques cibernéticos.

A segurança em redes não se limita apenas à proteção contra ameaças externas. É igualmente importante abordar riscos internos, que podem surgir devido a comportamentos inadequados de funcionários, falhas de configuração ou vulnerabilidades não corrigidas. A implementação de políticas de segurança rigorosas, como a atualização regular de software e a aplicação de senhas robustas, é fundamental para manter a integridade do sistema. Além disso, programas de treinamento e conscientização para funcionários ajudam a prevenir erros humanos e a aumentar a vigilância contra ameaças internas.

Figura 2.5: Integridade, confidencialidade e disponibilidade dos dados que circulam pela rede.

Fonte: Elaborado pelo autor (2024)

Com o crescimento exponencial de tecnologias emergentes, como a IoT e a computação em nuvem, a segurança em redes torna-se ainda mais crítica. Esses avanços tecnológicos introduzem novos vetores de ataque e aumentam a complexidade das infraestruturas de rede. A segurança precisa evoluir constantemente para acompanhar as mudanças no panorama de ameaças e garantir que os dados e operações permaneçam protegidos. A adoção de uma abordagem proativa, que inclui a análise regular de riscos, a realização de auditorias de segurança e a implementação de soluções de segurança avançadas, é essencial para enfrentar os desafios emergentes e proteger eficazmente os sistemas e informações digitais.

Além disso, a integração de tecnologias de segurança, como sistemas de detecção e resposta a intrusões *Intrusion Detection System* (IDS) e *Intrusion Prevention System* (IPS) para *Security Information and Event Management* (SIEM), pode proporcionar uma visibilidade mais aprofundada e uma resposta mais rápida a incidentes de segurança. Essas ferramentas são essenciais para monitorar continuamente o ambiente de rede, identificar possíveis ameaças e responder a incidentes de maneira eficaz. A segurança em redes é,

portanto, um processo contínuo e dinâmico que exige vigilância constante e adaptação às novas ameaças e desafios tecnológicos.[16] [17]

2.6 RELEVÂNCIA DA INTERNET DAS COISAS NA TRANSFORMAÇÃO DIGITAL

A IoT é um conceito tecnológico que descreve a interconexão de dispositivos físicos à internet, permitindo que esses equipamentos coletem, transmitam e processem dados de forma autônoma. Essa abordagem viabiliza uma comunicação eficiente entre sensores, atuadores, softwares e sistemas de análise, proporcionando automação e controle inteligente em diversos contextos, como residências, indústrias, cidades e ambientes corporativos.

A IoT se caracteriza por integrar dispositivos variados como câmeras de segurança, lâmpadas inteligentes, sensores ambientais e etiquetas *Radio-Frequency Identification* (RFID) a uma infraestrutura digital capaz de monitorar e reagir a condições específicas em tempo real conforme demonstrado na Figura 2.6. Essa capacidade transforma objetos convencionais em ativos inteligentes, otimizando processos, reduzindo custos operacionais e aumentando a segurança e a eficiência de sistemas.

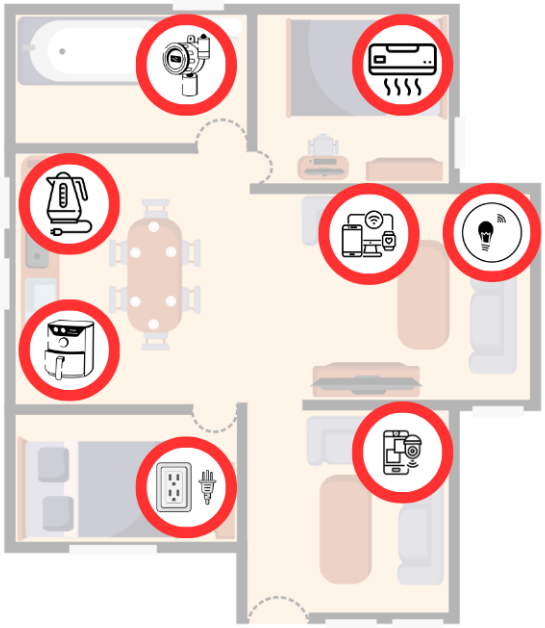

Figura 2.6: Componentes comuns em uma rede IoT residencial.

Fonte: Elaborado pelo autor (2025)

Uma das principais vantagens da IoT é sua aplicação em soluções de monitoramento e automação, possibilitando o controle remoto de equipamentos e a análise preditiva baseada em dados coletados continuamente. Esse modelo favorece decisões mais precisas e ágeis, sendo amplamente adotado em áreas como saúde, agricultura, transporte, logística e gestão de energia.

O ecossistema IoT é sustentado por tecnologias complementares, como redes sem fio, computação

em nuvem, protocolos de comunicação e inteligência artificial. A interoperabilidade entre dispositivos e plataformas é um desafio contínuo, mas também um elemento-chave para garantir a escalabilidade e a confiabilidade das soluções implementadas.

Outro fator essencial para o avanço da IoT é a segurança da informação. À medida que milhões de dispositivos passam a fazer parte de redes conectadas, torna-se imprescindível aplicar boas práticas de proteção de dados, autenticação e controle de acesso. Frameworks como o NIST fornecem diretrizes para mitigar vulnerabilidades e fortalecer a resiliência das infraestruturas IoT.

Por sua versatilidade e potencial transformador, a IoT representa uma das principais forças impulsionadoras da transformação digital, promovendo a integração entre o mundo físico e digital e abrindo caminho para inovações em diversos setores da sociedade.

2.6.1 Desafios de Inventário em Ambientes com Dispositivos IoT

A gestão de ativos tecnológicos tem passado por transformações significativas com a incorporação de dispositivos IoT em ambientes corporativos, industriais e acadêmicos. No entanto, esses avanços tecnológicos também introduziram novos desafios à rotina de inventário de hardware e software.

Uma das principais dificuldades reside na heterogeneidade dos dispositivos IoT, que podem variar amplamente em termos de sistema operacional, protocolos de comunicação, níveis de segurança e capacidade de resposta. Dispositivos como sensores passivos, etiquetas RFID ou atuadores embarcados frequentemente não possuem endereçamento IP ou não se comunicam ativamente na rede, dificultando sua detecção por ferramentas convencionais de varredura.

Outro desafio recorrente é a ausência de padronização na exposição de informações técnicas. Enquanto alguns dispositivos fornecem facilmente dados como versão de firmware, hostname e serviços ativos, outros ocultam ou restringem tais informações por padrão, seja por questões de segurança ou por limitações de hardware.

Além disso, há uma limitação estrutural das ferramentas tradicionais de inventário, que foram originalmente desenvolvidas para ambientes com desktops e servidores, e não consideram a dinâmica dos dispositivos IoT — como sua mobilidade, operação intermitente e grande volume.

Portanto, adaptar metodologias de inventário para contemplar dispositivos IoT demanda abordagens híbridas, que envolvam varreduras ativas, escuta passiva, coleta via SNMP, documentação manual e integração com gateways de comunicação. Essas soluções devem ser pensadas de forma a garantir a rastreabilidade, segurança e conformidade dos ativos em ambientes distribuídos e dinâmicos [18, 19].

2.7 IMPORTÂNCIA DO SISTEMA OPERACIONAL NA GESTÃO DE RECURSOS

Um sistema operacional é um software fundamental que desempenha um papel central na gestão do hardware e dos recursos de um computador, além de fornecer uma interface crítica entre o usuário e as aplicações. Ele coordena e controla operações essenciais, incluindo o gerenciamento da memória, a admi-

nistração de dispositivos de entrada e saída, e a execução de programas, assegurando que o sistema funcione de maneira eficiente e segura. Entre os exemplos amplamente reconhecidos de sistemas operacionais estão o Windows, o MacOS e o Linux, conforme demonstrado na Figura 2.7.

Além das funções de gerenciamento de hardware, o sistema operacional facilita a execução de aplicativos, permitindo que os usuários realizem uma ampla gama de tarefas, desde navegar na internet e editar documentos até jogar videogames e utilizar softwares especializados. Ele atua como um intermediário vital, traduzindo comandos do usuário em ações que o hardware pode processar e garantindo que os recursos do sistema sejam utilizados de maneira otimizada. Esse processo é fundamental para manter a estabilidade, a segurança e o desempenho do sistema.

Os sistemas operacionais modernos oferecem uma variedade de recursos adicionais, como multitarefa, que permite a execução simultânea de vários aplicativos, e gerenciamento de processos, que assegura a alocação eficiente de recursos entre diferentes tarefas. Eles também fornecem mecanismos de segurança, como controle de acesso e proteção contra malware, para proteger o sistema contra ameaças externas e internas.

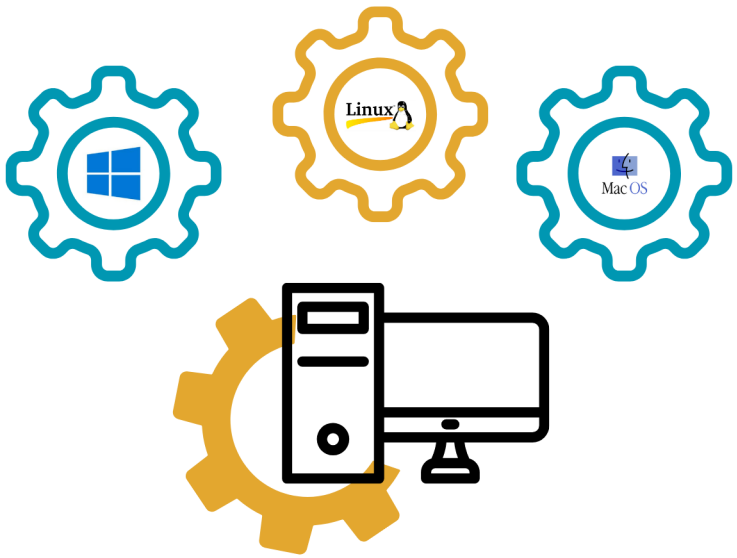

Figura 2.7: Principais sistemas operacionais: Windows, MacOS e Linux

Fonte: Elaborado pelo autor (2024)

A evolução dos sistemas operacionais tem sido impulsionada pela necessidade de suportar novas tecnologias e aplicações, como a computação em nuvem, a virtualização e os dispositivos móveis. Essas inovações exigem que os sistemas operacionais se adaptem continuamente para oferecer compatibilidade, desempenho e segurança aprimorados. Sem um sistema operacional, o hardware de um computador não poderia operar de forma funcional, limitando a capacidade dos usuários de interagir com a máquina e executar tarefas complexas. Portanto, o sistema operacional é essencial para a operação eficaz e acessível dos computadores modernos, servindo como a base sobre a qual todas as outras aplicações e serviços são

construídos.[20] [21]

2.7.1 Linux: Segurança e Customização em Sistemas Operacionais

Linux é um sistema operacional de código aberto que tem sua origem no núcleo (kernel) desenvolvido por Linus Torvalds em 1991. Desde então, Linux evoluiu significativamente, originando uma vasta gama de distribuições, como Ubuntu, Debian, Fedora e CentOS, conforme demonstrado na Figura 2.8. Essas distribuições são adaptadas para atender a diversas necessidades, que vão desde o uso em desktops pessoais até a gestão de servidores corporativos e sistemas embarcados. A popularidade do Linux pode ser atribuída a suas características notáveis de estabilidade, segurança e flexibilidade.

Uma das características distintivas do Linux é sua alta customização. O sistema operacional oferece uma ampla gama de opções de configuração, permitindo que desenvolvedores e administradores ajustem o ambiente de software para atender a requisitos específicos e variados. Essa flexibilidade é crucial para atender a diferentes cenários de uso e preferências individuais. Além disso, o fato de o Linux ser open source implica que seu código-fonte está disponível para análise e modificação por qualquer pessoa. Essa abertura não só fomenta a inovação, mas também garante que uma vasta comunidade de desenvolvedores e entusiastas possa colaborar no aprimoramento contínuo do sistema.

Figura 2.8: Distribuições Linux

Fonte: Elaborado pelo autor (2024)

A colaboração global desempenha um papel vital na evolução do Linux. A comunidade ativa contribui para a atualização constante de funcionalidades, a correção de vulnerabilidades e a oferta de suporte técnico. Essa rede colaborativa permite que o Linux se mantenha na vanguarda da tecnologia, adaptando-se rapidamente às novas demandas e desafios. Além disso, o modelo de desenvolvimento open source favo-

rece a transparência e a confiança, uma vez que o código é acessível para revisão e auditoria por qualquer pessoa interessada.

Linux também é amplamente utilizado em ambientes de alta performance e missão crítica, como supercomputadores e data centers, devido à sua capacidade de oferecer desempenho confiável e escalabilidade. Em dispositivos móveis, o sistema operacional Android, que é baseado no núcleo Linux, exemplifica a aplicação do Linux em uma ampla gama de dispositivos de consumo. A combinação de personalização, robustez e suporte da comunidade torna o Linux uma escolha atraente para empresas e usuários que buscam uma alternativa eficaz aos sistemas operacionais proprietários. A sua adoção continua a crescer, refletindo a confiança e a satisfação dos usuários em sua flexibilidade e capacidade de adaptação.[22] [23]

2.7.2 Ubuntu: Inovação em Sistemas Operacionais de Código Aberto

O Ubuntu é uma distribuição do sistema operacional Linux que se distingue pela sua interface amigável e acessível, o que o torna apropriado tanto para iniciantes quanto para usuários avançados. Desenvolvido pela Canonical Ltd., o Ubuntu é um sistema open source baseado no kernel Linux, o que significa que seu código-fonte é livremente disponível para modificação e aprimoramento por parte da comunidade global. Desde seu lançamento, o Ubuntu tem sido amplamente adotado em uma variedade de contextos, incluindo desktops pessoais, servidores e dispositivos de IoT, graças à sua confiabilidade, segurança e facilidade de uso.

Figura 2.9: Base sólida de aplicativos, Ubuntu.

Fonte: Elaborado pelo autor (2024)

Uma característica notável do Ubuntu é sua vasta gama de aplicações pré-instaladas, que inclui ferramentas essenciais para produtividade, como suítes de escritório, navegadores de internet e programas de edição de imagens e vídeo, Conforme demonstrado na Figura 2.9. Essa abordagem facilita o uso imediato do sistema e proporciona uma experiência de usuário fluida. O Ubuntu também é conhecido por seu compromisso com a atualização contínua e o suporte a *Long Term Support* (LTS), oferecendo versões que recebem atualizações de segurança e manutenção por períodos estendidos. Isso garante um ambiente

estável e seguro para os usuários, especialmente em contextos corporativos e de produção.

O suporte de uma comunidade global ativa é outro fator crucial para o sucesso do Ubuntu. A comunidade contribui com melhorias contínuas, correções de bugs e novos recursos, além de fornecer suporte técnico através de fóruns e documentação. Esse modelo de desenvolvimento colaborativo não apenas acelera a inovação, mas também fortalece a segurança do sistema, à medida que as vulnerabilidades são identificadas e corrigidas rapidamente.

Além de sua aplicabilidade em desktops e servidores, o Ubuntu é amplamente utilizado em ambientes de computação em nuvem e desenvolvimento de software. Suas versões específicas, como o Ubuntu Server, são projetadas para atender às necessidades de servidores e data centers, oferecendo funcionalidades robustas para gerenciamento e automação. A compatibilidade do Ubuntu com uma ampla gama de hardware e sua integração com tecnologias modernas, como contêineres e virtualização, reforçam sua posição como uma solução versátil e eficaz no cenário tecnológico.

Portanto, o Ubuntu não é apenas uma escolha popular para indivíduos e organizações que buscam uma alternativa viável aos sistemas operacionais proprietários, mas também uma plataforma confiável e inovadora para uma ampla variedade de aplicações e ambientes de uso. Sua combinação de acessibilidade, suporte da comunidade e atualizações contínuas faz do Ubuntu uma opção proeminente no ecossistema de sistemas operacionais baseados em Linux.[24] [25]

2.8 *OPEN SOURCE*: COLABORAÇÃO E FLEXIBILIDADE NO SOFTWARE

O software open source refere-se a programas cujo código-fonte é acessível ao público, permitindo que qualquer indivíduo examine, modifique e distribua o código conforme desejado. Em contraste com os softwares proprietários, que mantêm o código fechado e controlado por uma única entidade, o open source promove um ambiente de colaboração e transparência. Esse modelo permite que desenvolvedores de todo o mundo contribuam com melhorias, correções de bugs e inovações, criando uma rede global de colaboração. Conforme demonstrado na Figura 2.10.

Exemplos notáveis de software open source incluem o navegador Mozilla Firefox, o sistema de controle de versão Git e o sistema de gerenciamento de banco de dados MySQL. Essas ferramentas são amplamente utilizadas devido à sua flexibilidade e adaptabilidade, permitindo aos usuários ajustar as funcionalidades do software para atender a necessidades específicas, além de integrá-las com outros sistemas.

A natureza colaborativa do software open source oferece vantagens significativas, especialmente em termos de segurança e inovação. Com uma comunidade ativa de desenvolvedores, a resposta a problemas de segurança é frequentemente mais rápida, e novas funcionalidades são introduzidas com maior agilidade. Este modelo de desenvolvimento não só proporciona uma alternativa viável aos softwares comerciais fechados, mas também fomenta um ambiente onde a melhoria contínua e a inovação são impulsionadas pela contribuição coletiva de usuários e desenvolvedores ao redor do mundo.

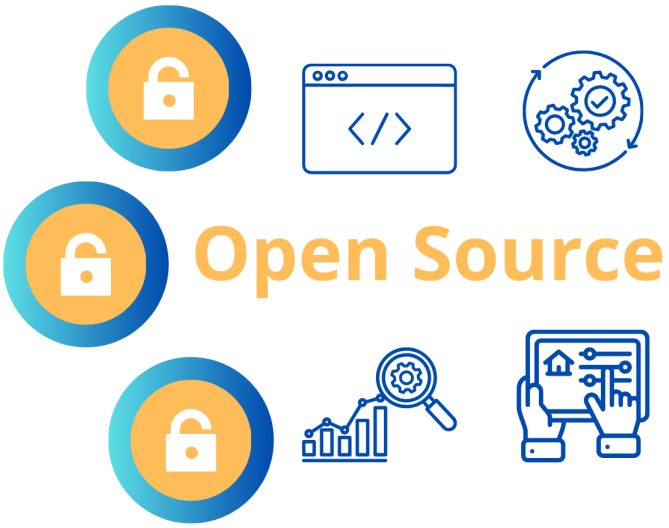

Figura 2.10: software open source.

Fonte: Elaborado pelo autor (2024)

Além disso, o software open source é frequentemente considerado uma opção econômica, pois, em muitos casos, é oferecido sem custos de licenciamento, o que pode reduzir significativamente os gastos com tecnologia para organizações e indivíduos. A transparência do código-fonte também permite que os usuários verifiquem a integridade do software e garantam que ele não contenha elementos indesejados ou maliciosos.

Portanto, o software open source representa uma abordagem dinâmica e acessível ao desenvolvimento de software, caracterizada pela colaboração global, flexibilidade e inovação contínua.[26] [27]

2.9 *OCS INVENTORY*: GERENCIAMENTO ABRANGENTE DE ATIVOS DE TI

O OCS é uma ferramenta robusta para o gerenciamento de ativos de TI, projetada para coletar e organizar informações detalhadas sobre o hardware e o software presentes em uma rede de computadores. Esta solução é amplamente adotada para centralizar a supervisão e administração dos ativos de TI, oferecendo uma visão integrada e clara dos recursos disponíveis na infraestrutura de tecnologia da informação de uma organização.

Uma das funcionalidades centrais do OCS Inventory é o inventário de hardware, que permite a coleta e o acompanhamento de dados detalhados sobre os componentes físicos dos dispositivos. Isso inclui informações sobre processadores, memória RAM, discos rígidos, placas de rede e outros periféricos. A capacidade de monitorar esses aspectos facilita a gestão do ciclo de vida dos ativos e a realização de manutenção preventiva e corretiva, contribuindo para a longevidade e eficiência dos equipamentos.

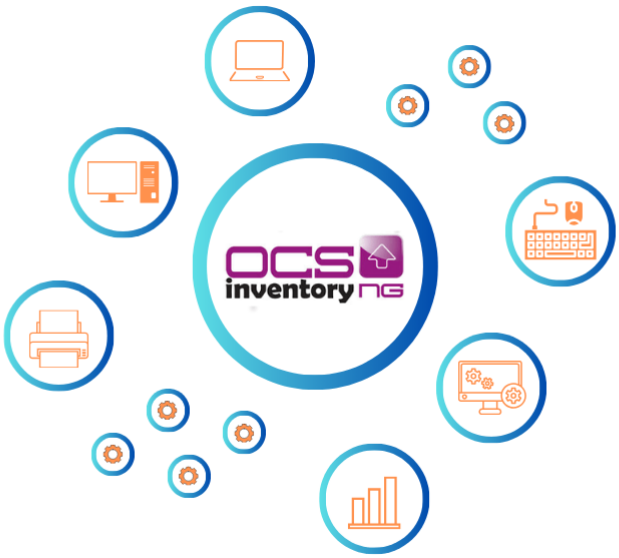

Figura 2.11: OCS Inventory (Open Computer and Software Inventory).

Fonte: Elaborado pelo autor (2024)

O inventário de software é igualmente fundamental, pois fornece uma visão detalhada dos programas instalados em cada máquina. Este recurso identifica todas as aplicações, suas versões e informações sobre licenciamento, o que é essencial para garantir a conformidade com as políticas de software e evitar problemas relacionados a licenças vencidas ou não autorizadas. A transparência proporcionada por essa funcionalidade também ajuda a identificar e remover softwares obsoletos ou não utilizados, otimizando o desempenho dos sistemas.

O OCS também se destaca pela funcionalidade de gestão de pacotes, que permite a instalação, remoção e atualização de softwares de forma remota. Esse recurso simplifica a administração dos sistemas, reduzindo o tempo e o esforço necessários para gerenciar atualizações e garantir que todos os dispositivos estejam equipados com as versões mais recentes das aplicações. Além disso, a ferramenta possui uma funcionalidade de descoberta de dispositivos de rede, que identifica e catalogará equipamentos como impressoras e roteadores mesmo que não estejam diretamente conectados ao agente OCS, ampliando a visibilidade e o controle sobre a infraestrutura de rede.

O OCS é, portanto, uma solução completa que oferece uma ampla gama de funcionalidades para o gerenciamento eficaz de ativos de TI. Sua capacidade de coletar e centralizar informações detalhadas sobre hardware e software, juntamente com suas funcionalidades avançadas de gestão e descoberta, torna-o uma ferramenta valiosa para a administração de recursos de tecnologia da informação em ambientes corporativos. Com a implementação do OCS, as organizações podem alcançar uma maior eficiência operacional, garantir a conformidade com políticas de licenciamento e manter um controle mais rigoroso sobre sua infraestrutura de TI.[28] [29]

2.10 MONITORAMENTO DE SISTEMAS E REDES: DESEMPENHO E SEGURANÇA

O monitoramento é um processo contínuo e sistemático de observação e análise de sistemas, redes e processos, destinado a assegurar seu funcionamento adequado, detectar problemas e otimizar o desempenho. No contexto da tecnologia da informação, o monitoramento abrange a supervisão detalhada de diversos componentes, incluindo servidores, dispositivos de rede, aplicativos e outros recursos digitais. Este processo é vital para identificar falhas, gargalos de desempenho e ameaças de segurança, permitindo uma intervenção rápida e eficaz.Conforme demonstrado na Figura 2.12.

Ferramentas de monitoramento desempenham um papel fundamental ao coletar e analisar dados em tempo real. Elas fornecem descobertas cruciais sobre o estado dos sistemas e redes, permitindo a visualização de métricas essenciais como uso de CPU, consumo de memória, tráfego de rede e tempo de resposta de aplicativos. Através dessas métricas, é possível identificar padrões anômalos, detectar problemas emergentes e avaliar o desempenho geral dos sistemas.

Figura 2.12: Monitoramento de sistemas e redes de computadores.

Fonte: Elaborado pelo autor (2024)

Uma das principais funções do monitoramento é prever e evitar interrupções nos serviços. Através da análise contínua dos dados coletados, é possível identificar e corrigir problemas antes que eles causem impacto significativo, como a queda de serviços críticos ou a degradação do desempenho. Por exemplo, ao detectar um aumento anômalo no uso de memória, os administradores podem tomar medidas corretivas para evitar que isso resulte em uma falha no sistema.

Além de sua função preventiva, o monitoramento é crucial para a gestão eficiente de recursos e a otimização de processos. A coleta e análise de métricas permitem ajustes na infraestrutura para melhorar a eficiência operacional e reduzir custos. Por exemplo, a análise do tráfego de rede pode ajudar a identificar a necessidade de balanceamento de carga, enquanto o monitoramento do uso de CPU pode indicar a necessidade de atualização de hardware ou ajustes na configuração dos sistemas.

O monitoramento também desempenha um papel importante na segurança cibernética. Através da detecção de atividades suspeitas e de padrões incomuns de tráfego, as ferramentas de monitoramento podem ajudar a identificar e responder rapidamente a ameaças de segurança, como ataques de malware ou tentativas de invasão. Isso contribui para a proteção dos dados e a integridade dos sistemas.

Em resumo, o monitoramento é um componente essencial na gestão de ambientes tecnológicos modernos. Ele não apenas garante a operação contínua e eficiente dos sistemas, mas também facilita a tomada de decisões informadas para melhorar a infraestrutura e a segurança. A implementação de estratégias robustas de monitoramento é crucial para manter a estabilidade, a disponibilidade e a confiabilidade dos serviços de TI, assegurando uma operação eficaz e a continuidade dos negócios.[30] [31]

2.11 TÉCNICAS DE COLETA DE INFORMAÇÕES E FINGERPRINTING DE DISPOSITIVOS

A coleta de informações em dispositivos de rede é uma etapa essencial para qualquer processo de inventário, monitoramento e análise de segurança. Em ambientes com dispositivos IoT, essa tarefa se torna ainda mais complexa, exigindo o uso de técnicas específicas, entre elas o *fingerprinting de dispositivos*.

O *fingerprinting* consiste na identificação de características únicas de um dispositivo, que podem ser obtidas por meio de análise ativa ou passiva do tráfego. Na abordagem ativa, utiliza-se ferramentas como o *Nmap*, que realiza varreduras nas portas de rede e interpreta as respostas recebidas para identificar o sistema operacional, serviços habilitados e até mesmo a versão do firmware. Por meio do módulo (OS detection) e scripts adicionais, o Nmap é capaz de inferir perfis de dispositivos com base em bancos de dados de assinaturas.

Já na abordagem passiva, o *fingerprinting* analisa pacotes de rede em trânsito sem interação direta com os dispositivos. Técnicas como análise de TTL, banners de serviços, respostas HTTP, *DHCP fingerprinting* ou observação de pacotes multicast (como mDNS ou SSDP) permitem inferir a natureza do dispositivo sem causar impacto em seu funcionamento.

Além disso, a coleta direta por *SNMP (Simple Network Management Protocol)*, quando habilitado, pode fornecer detalhes estruturados como nome do dispositivo, *uptime*, localização, fabricante e interface de rede. Em dispositivos que oferecem APIs ou interfaces web, é possível extrair dados através da leitura de *headers* HTTP ou por meio de scripts automatizados.

A escolha da técnica adequada depende do ambiente, do nível de acesso disponível e da criticidade dos dispositivos. Em todos os casos, o objetivo é garantir a obtenção segura, precisa e atualizada das informações, para subsidiar o inventário, a classificação de risco e as ações corretivas [32, 33, 34].

2.12 GESTÃO DE INVENTÁRIO: CONTROLE EFICIENTE DE ATIVOS DE TI

O inventário é um processo crítico que envolve a criação e manutenção de um registro abrangente e organizado de todos os bens, recursos e ativos disponíveis em uma empresa, organização ou sistema conforme demonstrado na Figura 2.13. Em termos de tecnologia da informação, isso inclui uma vasta gama de itens, como hardware (servidores, computadores, dispositivos de rede), software (programas, aplicações, licenças) e outros recursos digitais essenciais. A catalogação minuciosa desses ativos permite uma gestão mais eficiente e estratégica, ajudando a monitorar o estado e a localização de cada item e facilitando tarefas importantes como a manutenção preventiva, o planejamento de aquisições futuras e a realização de auditorias.

Figura 2.13: Gestão de Inventário.

Fonte: Elaborado pelo autor (2024)

A gestão de inventário não se limita apenas ao controle físico e operacional dos ativos. Ela também desempenha um papel vital na segurança e na conformidade regulatória. Um inventário atualizado e preciso permite à organização identificar rapidamente equipamentos obsoletos ou software fora de conformidade, reduzindo riscos de segurança e evitando possíveis penalidades associadas ao uso de software não licenciado. Além disso, a prática de manter um inventário detalhado auxilia na conformidade com regulamentações e políticas internas, garantindo que todos os ativos sejam geridos de acordo com as normas estabelecidas.

O processo de inventário é também uma ferramenta estratégica para a tomada de decisões financeiras e operacionais. Ao fornecer uma visão clara e atualizada dos ativos, o inventário permite aos gestores realizar análises detalhadas sobre o desempenho dos recursos, identificar áreas onde a eficiência pode ser aprimorada e tomar decisões informadas sobre substituições, upgrades e alocações de orçamento. Essa abordagem não só contribui para a eficiência operacional, mas também ajuda a controlar e otimizar os custos relacionados à gestão de ativos.

Além disso, um inventário bem estruturado facilita a resposta rápida a incidentes e a recuperação de ativos em situações de emergência. Em caso de falhas de hardware ou software, por exemplo, a capacidade de acessar rapidamente informações detalhadas sobre os ativos afetados pode acelerar a resolução de problemas e minimizar o impacto nas operações da organização.

Portanto, a implementação de um sistema de inventário robusto e a realização de atualizações periódicas são essenciais para assegurar a integridade, a segurança e a eficiência dos ativos tecnológicos. O inventário não apenas proporciona um controle efetivo sobre os recursos, mas também desempenha um papel estratégico na gestão geral e na proteção dos ativos da organização.[35] [36]

2.12.1 Características

1 Registro Detalhado: Um sistema de inventário fornece um registro abrangente de todos os ativos de TI, incluindo hardware, software, licenças e outros recursos digitais. Cada item é catalogado com informações detalhadas, como modelo, número de série, data de aquisição e localização física.

2 Atualizações em Tempo Real: Permite a atualização contínua e em tempo real das informações sobre os ativos, refletindo alterações como novas aquisições, substituições, atualizações de software e remoções.

3 Classificação e Organização: Oferece funcionalidades para classificar e organizar os ativos em categorias, facilitando a busca e a visualização das informações. As categorias podem incluir hardware, software, dispositivos de rede e outros.

4 Relatórios e Auditorias: Gera relatórios detalhados e realiza auditorias periódicas dos ativos. Os relatórios podem incluir o status dos ativos, conformidade com licenças e análise de desempenho.

6 Integração com Outros Sistemas: Pode integrar-se com outros sistemas de gerenciamento, como sistemas de gerenciamento de manutenção, sistemas financeiros e plataformas de monitoramento de rede, para uma visão mais holística dos recursos da TI.

7 Controle de Licenciamento: Monitora o status das licenças de software, garantindo que todas as aplicações estejam em conformidade com os termos de licenciamento e ajudando a evitar o uso não autorizado.[37] [38]

2.12.2 Vantagens

1 Eficiência Operacional: Melhora a eficiência operacional ao fornecer uma visão centralizada e organizada dos ativos de TI. Isso facilita a gestão e a alocação dos recursos, reduzindo o tempo necessário para localizar e gerenciar ativos.

2 Redução de Custos: Ajuda a controlar os custos ao permitir a otimização dos recursos existentes e evitar a compra desnecessária de novos ativos. O monitoramento contínuo também identifica oportunidades para a reutilização de hardware e software.

3 Melhoria na Segurança: Contribui para a segurança da informação ao garantir que todos os ativos estejam atualizados e em conformidade com as políticas de segurança. A identificação rápida de ativos obsoletos ou vulneráveis permite a implementação de medidas corretivas.

4 Facilitação de Auditorias: Torna as auditorias mais fáceis e rápidas, proporcionando registros precisos e atualizados que podem ser facilmente acessados para verificar a conformidade e a integridade dos ativos.

5 Melhor Planejamento e Tomada de Decisões: Oferece dados valiosos para o planejamento de futuras aquisições e investimentos em TI. As análises e relatórios fornecidos ajudam na tomada de decisões estratégicas sobre upgrades, substituições e gerenciamento de ativos.[39] [40]

2.12.3 Benefícios

1 Aumento da Produtividade: Ao reduzir o tempo gasto na busca e gestão de ativos, um sistema de inventário aumenta a produtividade das equipes de TI e permite que os recursos sejam utilizados de maneira mais eficaz.

2 Maior Controle e Visibilidade: Proporciona maior controle e visibilidade sobre todos os ativos de TI, permitindo uma gestão mais proativa e informada dos recursos tecnológicos da organização.

3 Conformidade Regulamentar: Ajuda a garantir que a organização esteja em conformidade com regulamentações e políticas de licenciamento, evitando problemas legais e financeiros relacionados ao uso não autorizado de software.

4 Suporte à Recuperação de Desastres: Facilita a recuperação de desastres ao fornecer informações detalhadas sobre todos os ativos, permitindo uma recuperação rápida e eficiente após incidentes ou falhas.

5 Aprimoramento da Estratégia de TI: Contribui para o aprimoramento da estratégia de TI ao fornecer insights baseados em dados sobre a utilização dos recursos, tendências de desempenho e áreas para melhorias.[27] [41]

2.13 NORMAS E BOAS PRÁTICAS DE SEGURANÇA: ENFOQUE NO FRAMEWORK NIST

A adoção de normas e boas práticas de segurança é um componente essencial na gestão eficiente e segura de infraestruturas de Tecnologia da Informação, especialmente em ambientes que incluem dispositivos IoT. Entre os principais referenciais utilizados internacionalmente destaca-se o framework do NIST, amplamente reconhecido por estabelecer diretrizes claras e abrangentes para a proteção de ativos tecnológicos.

O NIST foi desenvolvido com o objetivo de auxiliar organizações a identificar, proteger, detectar, responder e recuperar-se de incidentes de segurança cibernética. Essas cinco funções: Identify, Protect,

Detect, Respond e Recover, formam a espinha dorsal do framework e são especialmente relevantes quando aplicadas à gestão de ativos de TI e IoT [42].

No contexto deste estudo, o eixo Identify (Identificar) é particularmente relevante, pois destaca a importância de manter um inventário atualizado de ativos, incluindo hardware, software e dispositivos conectados à rede. Essa prática permite que a organização compreenda com clareza quais dispositivos estão em operação, quais são os riscos associados e como priorizar medidas de segurança. A coleta de informações precisa e contínua é um dos fundamentos do controle eficaz, sendo diretamente abordada nas diretrizes do NIST conforme demonstrado na Figura 2.14.

Além disso, a função Protect do framework reforça a necessidade de estabelecer medidas técnicas e organizacionais para garantir a segurança dos dispositivos inventariados. Isso inclui o controle de acesso, a atualização de firmwares, a segmentação de rede e o uso de criptografia ações que se tornam ainda mais críticas quando se trata de dispositivos IoT, frequentemente expostos a vulnerabilidades devido a limitações de hardware ou ausência de atualizações automatizadas.

Figura 2.14: Framewor de segurança NIS

Fonte: Elaborado pelo autor (2024)

O uso de ferramentas de código aberto, como o OCS, aliado à aplicação das práticas propostas pelo NIST, possibilita não apenas a visibilidade dos ativos conectados, mas também o fortalecimento da postura de segurança da organização. Neste projeto, a personalização do OCS busca justamente alinhar o processo de inventário automatizado com os princípios do NIST, oferecendo uma solução que vá além do monitoramento tradicional e permita maior controle sobre dispositivos IoT, que muitas vezes escapam dos métodos convencionais de detecção.

Dessa forma, a incorporação das diretrizes do NIST no desenvolvimento da solução proposta garante não apenas a conformidade com padrões reconhecidos, mas também contribui para um ambiente mais seguro, rastreável e resiliente frente às crescentes ameaças no cenário digital.

2.14 PROTOCOLOS DE REDE EM AMBIENTES IOT

Os protocolos de rede constituem a base da comunicação entre dispositivos conectados, viabilizando a troca estruturada de informações em diferentes camadas do modelo *Open Systems Interconnection* (OSI). Em ambientes IoT, onde coexistem dispositivos heterogêneos com diferentes restrições de processamento, energia e conectividade, a escolha e aplicação adequada dos protocolos assume papel crítico para assegurar interoperabilidade, desempenho e segurança [43].

Na camada de aplicação, destacam-se protocolos como *Hypertext Transfer Protocol* (HTTP)/*Hypertext Transfer Protocol Secure* (HTTPS), amplamente utilizados para comunicação cliente-servidor, e o *Message Queuing Telemetry Transport* (MQTT), que se tornou padrão em IoT por sua leveza e suporte a conexões de baixa largura de banda [44]. Outro protocolo relevante é o *Constrained Application Protocol* (COAP), projetado especificamente para dispositivos com recursos limitados, permitindo operações semelhantes ao HTTP, mas com menor sobrecarga de rede [45].

Na camada de transporte, os protocolos *Transmission Control Protocol* (TCP) e UDP desempenham funções centrais. O TCP garante confiabilidade por meio de controle de fluxo, retransmissão de pacotes e verificação de integridade, sendo adequado para aplicações críticas [46]. Já o UDP, por não exigir confirmação de entrega, apresenta baixa latência e menor sobrecarga, sendo preferido em aplicações de streaming, comunicação em tempo real e cenários IoT sensíveis ao tempo [43].

Na camada de rede, o *Internet Protocol* (IP) é responsável pelo endereçamento e roteamento dos pacotes. Em dispositivos IoT, a transição para o *Internet Protocol version 6* (IPV6) é essencial devido à necessidade de escalabilidade e à exaustão de endereços IPV4 [47]. Protocolos auxiliares, como o *Internet Control Message Protocol* (ICMP), também desempenham papel importante no diagnóstico e monitoramento de redes, fornecendo informações sobre latência, perda de pacotes e acessibilidade.

Além disso, em camadas inferiores, protocolos como Ethernet, Wi-Fi, *Bluetooth Low Energy* (BLE), ZigBee e LoRaWAN oferecem diferentes alternativas de conectividade, cada uma com vantagens específicas em termos de alcance, consumo energético, taxa de transmissão e topologia de rede [48]. A escolha do protocolo adequado depende diretamente das características do ambiente e dos requisitos da aplicação.

A definição da pilha de protocolos em ambientes IoT deve considerar não apenas o desempenho e a eficiência energética, mas também os requisitos de segurança conforme Tabela 2.1 . A utilização de extensões como *Transport Layer Security* (TLS)/*Datagram Transport Layer Security* (DTLS) para criptografia e autenticação é fundamental para garantir a confidencialidade e a integridade dos dados, mitigando vulnerabilidades que poderiam ser exploradas por agentes maliciosos [49].

Portanto, os protocolos de rede não apenas viabilizam a conectividade entre dispositivos heterogêneos, mas também estabelecem os fundamentos sobre os quais se constroem mecanismos de interoperabilidade, resiliência e proteção.

Protocolo	Camada	Vantagens	Limitações	Aplicação em IoT
MQTT	Aplicação	Leve; opera bem com baixa largura de banda e links instáveis.	Depende de *broker* central; exige TLS/DTLS para segurança.	Telemetria, sensoriamento remoto, *smart home*.
COAP	Aplicação	Otimizado para dispositivos restritos; estilo REST; baixa sobrecarga.	Escalabilidade e segurança dependem de DTLS; menos suporte em nuvem.	Automação residencial; M2M leve.
HTTP/HTTPS	Aplicação	Onipresente; fácil integração Web/APIs.	Overhead maior; não ideal para muito baixo consumo.	APIs de IoT; dashboards e integrações Web.
TCP	Transporte	Confiável (ordem/entrega); controle de fluxo.	Overhead/latência em redes limitadas.	Inventário crítico; atualização de firmware.
UDP	Transporte	Baixa latência; menor sobrecarga.	Sem garantia de entrega/ordem.	Streaming de dados; tempo real sensível.
IP/IPV6	Rede	Escalabilidade massiva; endereçamento global.	Adoção desigual; coexistência com IPv4.	Endereçamento amplo de sensores.
ZigBee	Enlace/Física	Baixo consumo; *mesh* nativo.	Taxa de dados baixa; alcance limitado.	Iluminação/automação residencial.
LoRaWAN	Enlace/Física	Longo alcance; baixíssimo consumo.	Baixa taxa de dados; alta latência.	Agricultura/ambiental em longas distâncias.
BLE	Enlace/Física	Consumo muito baixo; integra com celulares.	Alcance curto; interferências.	Wearables; saúde conectada.
Wi-Fi	Enlace/Física	Alta taxa de dados; infraestrutura difundida.	Consumo alto; alcance médio.	Câmeras e dispositivos domésticos conectados.

Tabela 2.1: Comparação dos principais protocolos de rede utilizados em ambientes de Internet das Coisas (IoT)

Fonte: Elaborado pelo autor (2025)

2.15 TRABALHOS CORRELATOS

O inventário de dispositivos tecnológicos sempre foi considerado um requisito básico para a gestão de ativos em redes corporativas. Entretanto, com o crescimento acelerado da Internet das Coisas (IoT) e suas características específicas, como grande heterogeneidade de dispositivos, diferentes capacidades de processamento e protocolos diversos , o simples modelo de inventário tradicional tornou-se insuficiente. Essa nova realidade demanda abordagens mais avançadas, capazes não apenas de registrar os ativos, mas também de prover informações contínuas sobre sua atualização, localização e estado de segurança.

Diversos estudos recentes, publicados em conferências e periódicos de destaque (IEEE, ACM, Springer, SBRC), têm buscado enfrentar esse desafio. As propostas incluem desde métodos automatizados de descoberta de dispositivos em redes heterogêneas até a integração com plataformas de monitoramento e

sistemas de detecção de intrusões (IDS) e SIEM, que possibilitam correlação de eventos em tempo real. Essas pesquisas evidenciam que o inventário não deve ser entendido apenas como uma listagem estática de ativos, mas como um processo dinâmico, orientado à segurança e à governança em ambientes IoT.

Nesta seção, serão analisados alguns desses trabalhos correlatos, destacando suas contribuições para o avanço das práticas de inventário automatizado e sua relevância para a construção de arquiteturas mais seguras e resilientes.

2.15.1 Inventario de IoT focados em Segurança

Nesta subseção serão apresentados os principais trabalhos correlatos que abordam soluções voltadas ao inventário e à gestão de dispositivos IoT. Cada estudo destacado demonstra diferentes estratégias para a descoberta automática de ativos, registro estruturado de informações e integração com práticas de segurança.

2.15.2 Técnicas de Monitoramento e Avaliação de Vulnerabilidades para IoT

No trabalho de Guo et al. [50], é apresentado um sistema de monitoramento e identificação de vulnerabilidades em ambientes industriais, com ênfase em dispositivos IoT e *Industrial Internet of Things* (IIOT). O estudo destaca a necessidade de coletar informações detalhadas de ativos conectados, de forma a viabilizar tanto o inventário de dispositivos quanto a análise de sua superfície de ataque.

Os autores propõem técnicas capazes de detectar e classificar equipamentos industriais, associando dados como hardware, firmware, número de série e módulos de comunicação a cada dispositivo monitorado. A Figura 2.15 ilustra esse processo ao mostrar a identificação detalhada de um Controlador Lógico Programável (PLC) Siemens S7-300/ET200M, evidenciando informações fundamentais para inventários automatizados, como versão de firmware, tipo de módulo e identificação de estação.

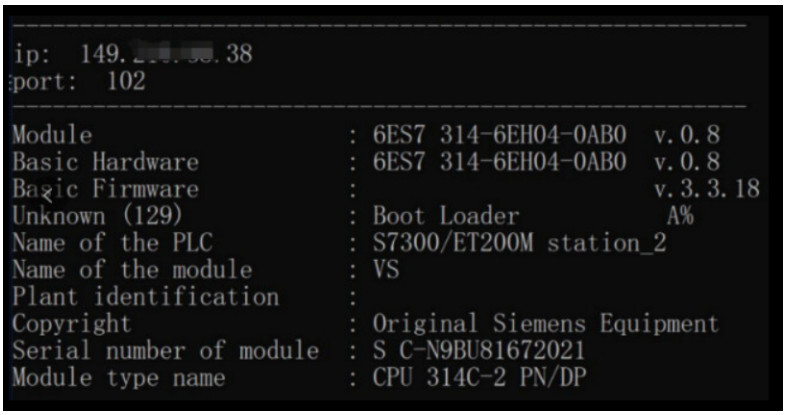

Figura 2.15: PLC Siemens S7-300/ET200M, dados de hardware, firmware, número de série e tipo de módulo.

Fonte: Guo et al.[50]

Esse tipo de coleta é essencial em projetos de inventário de IoT/IIoT, pois possibilita a rastreabilidade dos ativos, o mapeamento de vulnerabilidades específicas (por exemplo, firmware desatualizado) e a cria-

ção de uma base sólida para estratégias de defesa cibernética. Ao associar monitoramento em tempo real com inventário detalhado, a abordagem de Guo et al. reforça a importância de alinhar a segurança com a gestão de ativos industriais [50].

2.15.3 Riscos de Segurança e Desafios Emergentes em Ambientes IoT

No estudo publicado na SpringerLink [51], os autores realizaram uma análise crítica sobre as tendências de segurança em aplicações IoT, com foco em identificar desafios emergentes no contexto da expansão de dispositivos conectados. O artigo descreve que o aumento da superfície de ataque, aliado à diversidade de protocolos e ambientes de comunicação, potencializa riscos para a confidencialidade, integridade e disponibilidade dos sistemas.

Entre os pontos destacados, os autores enfatizam a necessidade de arquiteturas resilientes e escaláveis, além do desenvolvimento de mecanismos de detecção precoce de ataques e de modelos de resposta automatizados. Segundo os autores, abordagens tradicionais de segurança de redes não são suficientes para o cenário IoT, dado o baixo poder de processamento e restrições energéticas de diversos dispositivos [51].

A Figura 2.16 apresentada no artigo ilustra o fluxo de funcionamento em quatro estágios: (i) sensores, atuadores e dispositivos; (ii) sistemas de aquisição de dados; (iii) pré-processamento e análises em borda; e (iv) análises em nuvem. Esse modelo evidencia que a coleta e organização sistemática das informações desde a borda até a nuvem é essencial não apenas para segurança, mas também para a manutenção de inventários atualizados de dispositivos IoT, permitindo rastrear quais equipamentos estão conectados, quais dados produzem e como são tratados ao longo da arquitetura.

A contribuição central do artigo é oferecer uma visão consolidada sobre desafios futuros e indicar caminhos para a aplicação de técnicas emergentes, como inteligência artificial aplicada à cibersegurança, blockchain para autenticação distribuída e criptografia, como formas de mitigar vulnerabilidades sem comprometer o desempenho dos dispositivos [51].

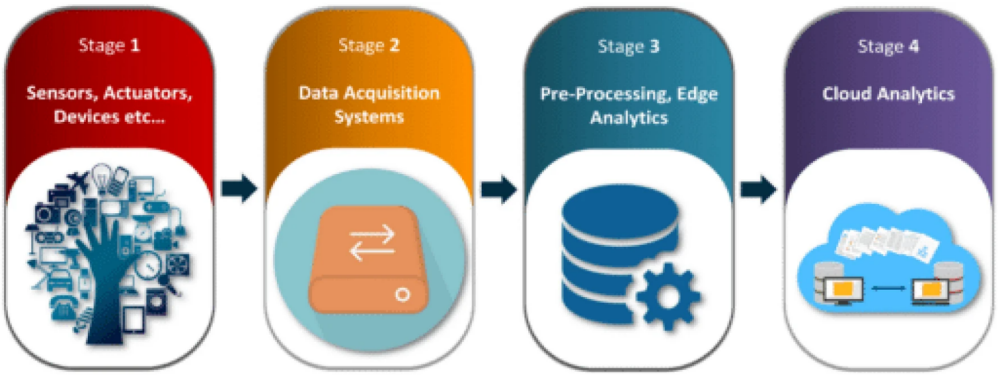

Figura 2.16: Arquitetura em quatro estágios para aplicações IoT

Fonte: SpringerLink.[51]

2.15.4 Disponibilidade e Limitações de Bases de Dados IoT para Segurança de Redes

No artigo publicado na *Institute of Electrical and Electronics Engineers* (IEEE) Communications Surveys [52], é apresentado um levantamento abrangente sobre datasets públicos de IoT voltados para pesquisas em segurança de redes. O trabalho sistematiza conjuntos de dados amplamente utilizados em experimentos de detecção de intrusão, ataques *Distributed Denial of Service* (DDos) e anomalias de tráfego em ambientes IoT, destacando suas características, limitações e aplicabilidade em diferentes contextos.

Segundo os autores, um dos principais desafios enfrentados pela comunidade científica está na escassez de datasets representativos e atualizados, o que compromete a avaliação de algoritmos de detecção baseados em aprendizado de máquina [52]. Nesse sentido, o estudo organiza os datasets em categorias conforme o tipo de tráfego capturado, protocolos envolvidos (IoT-specific ou IP-based) e cenário experimental utilizado.

A Figura 2.17 apresenta a distribuição dos protocolos em datasets IoT analisados, evidenciando que a maioria está concentrada em protocolos IP, enquanto protocolos específicos de IoT, como Zigbee, Bluetooth/LE e LoRa, aparecem em menor número. Esse resultado é particularmente relevante para estudos de inventário de dispositivos IoT, pois demonstra que a maior parte das bases disponíveis ainda não reflete de forma equilibrada a diversidade de protocolos empregados em ambientes reais, limitando a rastreabilidade e o mapeamento completo dos ativos conectados.

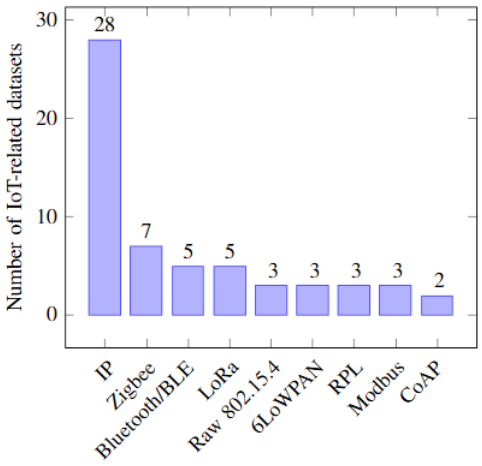

Figura 2.17: Distribuição de protocolos em datasets públicos de IoT.

Fonte: Keersmaeker et al.[52]

Como resultado, o trabalho fornece uma referência consolidada para pesquisadores, auxiliando na escolha do dataset mais adequado de acordo com a natureza do estudo. Além disso, evidencia-se a necessidade de esforços colaborativos para criação de novos repositórios públicos, capazes de refletir o crescimento e a complexidade dos ambientes IoT atuais [52].

2.15.5 Inventário Automatizado de Dispositivos IoT

Autores como Zhao et al. [53] apresentam uma revisão sistemática sobre o uso de tecnologias IoT para digitalizar e automatizar processos de inventário de ativos. O estudo evidencia que a gestão manual é insuficiente em ambientes com alta heterogeneidade de dispositivos conectados, o que demanda novas metodologias para garantir rastreabilidade, precisão e atualização contínua. Os principais objetivos identificados no trabalho são:

- Propor metodologias de inventário que permitam superar limitações do registro manual, reduzindo inconsistências nos dados.
- Minimizar o tempo necessário para a coleta de informações sobre dispositivos, com foco em eficiência operacional.
- Reduzir erros associados à falta de padronização de dados, integrando diferentes protocolos de rede e formatos de comunicação.
- Garantir a rastreabilidade de dispositivos durante todo o ciclo de vida, com atualização contínua e possibilidade de integração com sistemas de monitoramento.

O modelo de inventário automatizado descrito é geralmente estruturado em três fases: descoberta de dispositivos (*Device Discovery*), coleta de atributos (*Attribute Collection*) e registro centralizado (*Inventory Registration*), conforme detalhado a seguir.

1. **Device Discovery:** A primeira fase corresponde à detecção de dispositivos conectados na rede, utilizando protocolos como *mDNS*, *SSDP* e varreduras ativas de IP. Essa etapa é responsável por identificar a presença de dispositivos e mapear endereços IP e MAC, constituindo a base inicial do inventário.
2. **Attribute Collection:** Na segunda fase, ocorre a coleta de atributos relevantes dos dispositivos detectados, tais como fabricante, modelo, versão de firmware e localização na rede. Essa coleta pode ser feita por consultas diretas, análise de tráfego ou protocolos de gerenciamento.
3. **Inventory Registration:** Por fim, as informações coletadas são organizadas em um banco de dados estruturado, permitindo a padronização, a integração com sistemas de segurança e a geração de relatórios. Essa fase garante que o inventário se torne dinâmico, podendo ser atualizado automaticamente conforme novos dispositivos são adicionados ou removidos.

O estudo reforça que a precisão e a automação no inventário de dispositivos IoT são pré-requisitos essenciais para políticas de segurança, visto que apenas com visibilidade adequada é possível aplicar técnicas de monitoramento, detecção de anomalias e resposta a incidentes.

2.15.6 Inventário Rápido em IoT Ambiental com Restrições de Energia

Wu et al. [54] apresentam uma abordagem voltada ao inventário de dispositivos no contexto da Ambient IoT (A-IoT), padronizada pelo *3rd Generation Partnership Project* (3GPP), considerando a indisponibilidade periódica de dispositivos causada pela dependência de energia colhida do ambiente (energy harvesting). Os autores destacam que, devido à natureza intermitente desses dispositivos, os métodos tradicionais de inventário não são adequados, exigindo novas estratégias para garantir a completude do processo de descoberta. Os principais objetivos do trabalho são:

- Desenvolver um método eficiente de inventário para redes Ambient IoT, mesmo em cenários de indisponibilidade de dispositivos.
- Reduzir o tempo total de execução do inventário, explorando técnicas de paralelismo e agrupamento de dispositivos.
- Minimizar a perda de informações causada por falhas temporárias de energia nos nós IoT.
- Melhorar a escalabilidade do processo de inventário para grandes quantidades de dispositivos heterogêneos.

A solução proposta é estruturada em três fases principais: coleta de disponibilidade (*Availability Sensing*), agrupamento e priorização (*Clustering and Prioritization*) e execução do inventário rápido (*Fast Inventory Execution*), detalhadas a seguir.

1. **Availability Sensing:** consiste em identificar quais dispositivos estão ativos em um dado intervalo de tempo, considerando o comportamento intermitente causado pelo *energy harvesting*. Essa etapa estabelece a base para definir a janela temporal em que o inventário pode ser realizado.
2. **Clustering and Prioritization:** fase responsável por organizar os dispositivos detectados em grupos, de acordo com critérios como intensidade de sinal, perfil de consumo energético e padrões de resposta. Esse agrupamento permite otimizar o processo, priorizando dispositivos mais estáveis e com maior probabilidade de disponibilidade.
3. **Fast Inventory Execution:** etapa final em que o inventário é executado de forma acelerada, utilizando transmissões paralelas e estratégias de agendamento para reduzir o tempo total. Dessa forma, é possível alcançar uma taxa de inventário entre 50% e 83% mais rápida em comparação com métodos tradicionais, mesmo em ambientes com dispositivos intermitentes.

O estudo de Wu et al. [54] demonstra que o inventário em redes Ambient IoT precisa considerar explicitamente a natureza energética dos dispositivos, o que reforça a relevância de abordagens automatizadas e resilientes, alinhadas ao objetivo deste trabalho de propor uma arquitetura de inventário segura e eficiente para IoT

2.15.7 Conclusões Sobre os Trabalhos Relacionados

Os trabalhos analisados variam em aspectos como metodologias de inventário, protocolos utilizados, bases de dados de apoio, mecanismos de rastreabilidade e níveis de automação empregados. A seguir, apresenta-se uma síntese comparativa entre os trabalhos discutidos nesta seção e a proposta deste estudo, destacando contribuições, lacunas e oportunidades de avanço.

No trabalho de Guo et al. [50], é apresentado um modelo detalhado de identificação de vulnerabilidades em dispositivos industriais IoT/IIoT. Apesar de fornecer informações ricas sobre hardware e firmware, a proposta não contempla mecanismos de integração contínua do inventário com sistemas de monitoramento em tempo real, o que limita sua aplicação em ambientes dinâmicos.

Em contrapartida, o estudo de SpringerLink [51] amplia a discussão ao mapear desafios emergentes de segurança em IoT, destacando riscos associados à diversidade de protocolos e à baixa capacidade de dispositivos. Contudo, o artigo não propõe um modelo prático de inventário automatizado, permanecendo no nível conceitual.

Já Keersmaeker et al. [52] organizam os principais datasets públicos de IoT disponíveis, evidenciando lacunas na representatividade de protocolos específicos, como Zigbee, LoRa e Bluetooth LE. Embora seja uma contribuição relevante para a comunidade científica, a ausência de dados reais sobre inventários corporativos ou híbridos limita o uso direto desses datasets na construção de soluções de gestão de ativos.

Autores como Zhao et al. [53] propõem metodologias de inventário automatizado com base em fases de descoberta, coleta de atributos e registro centralizado. Apesar de avançarem na padronização, os trabalhos analisados não tratam da integração com normas de conformidade (como NIST, NIS2 e LGPD), deixando uma lacuna em termos de alinhamento regulatório.

Wu et al. [54], por sua vez, exploram o inventário em cenários de Ambient IoT (A-IoT) com restrições de energia. Embora inovador, o estudo restringe-se ao contexto de energy harvesting, sem abordar ambientes corporativos heterogêneos ou a necessidade de inventários híbridos que combinem IoT, IIoT e dispositivos convencionais.

Os trabalhos revisados não contemplam de forma completa os seguintes pontos, que se configuram como diferenciais da proposta desta dissertação:

- Um framework unificado de inventário que permita a descoberta, coleta e atualização contínua de dispositivos IoT em tempo real.
- Adoção de protocolos múltiplos de descoberta (mDNS, SSDP, varreduras ativas e registros manuais) para superar limitações de métodos únicos.
- Integração do inventário com normas de conformidade (NIST CSF, NIS2, CRA e LGPD), garantindo alinhamento regulatório.

A Tabela 2.2 apresenta uma síntese comparativa dos trabalhos analisados, destacando suas propostas, limitações e a relevância para esta dissertação. Por meio dela, evidencia-se que, embora cada estudo avance em aspectos importantes, nenhum contempla de forma integrada os requisitos de rastreabilidade, atualização contínua e conformidade regulatória em inventários de IoT/IIoT.

Autor/An	Proposta	Limitações	Relevância para este trabalho
[50] 2024	Modelo detalhado de identificação de vulnerabilidades em dispositivos IIoT, incluindo dados de hardware e firmware.	Não contempla integração contínua do inventário com sistemas de monitoramento em tempo real, reduzindo aplicabilidade em ambientes dinâmicos.	Evidencia a importância de informações ricas de dispositivos, mas reforça a lacuna de atualização contínua e integração com monitoramento.
[51] 2024	Mapeamento de desafios emergentes de segurança em IoT, destacando diversidade de protocolos e restrições de dispositivos.	Permanece em nível conceitual, sem propor modelo prático de inventário automatizado.	Fornece visão abrangente sobre riscos de IoT, mas carece de proposta prática de inventário.
[52] 2023	Organização de datasets públicos de IoT, evidenciando lacunas em protocolos como Zigbee, LoRa e BLE.	Ausência de dados reais corporativos ou híbridos, limitando aplicabilidade em inventários reais.	Relevante como fonte de dados para pesquisa, mas insuficiente para gestão prática de ativos em IoT/IIoT.
[53] 2024	Metodologias de inventário automatizado em fases (descoberta, coleta e registro centralizado).	Não trata de integração com normas de conformidade (NIST, NIS2, LGPD).	Avança em padronização do inventário, mas abre lacuna regulatória que este trabalho busca preencher.
[54] 2025	Inventário em cenários de Ambient IoT (A-IoT) com foco em dispositivos de baixo consumo e energy harvesting.	Restringe-se a A-IoT e não contempla ambientes corporativos heterogêneos ou inventários híbridos.	Inspira abordagens inovadoras, mas não cobre a necessidade de inventários híbridos e corporativos.

Tabela 2.2: Panorama de estudos relacionados ao inventário e monitoramento de dispositivos IoT

Fonte: Elaborado pelo autor (2025)

3 METODOLOGIA

Esta seção apresenta uma descrição detalhada dos métodos de coleta de informações empregados no processo dentro do UIoT. Conforme ilustrado na Figura 3.1, a análise está organizada em quatro etapas funcionais distintas, que garantem uma abordagem sistemática e eficiente na gestão de ativos de tecnologia da informação.

A metodologia adotada baseia-se na coleta automatizada de dados, centralizando informações de inventário em um servidor principal. Com o uso de agentes instalados nos dispositivos, a ferramenta capta dados sobre componentes de hardware, como aceleração, memória e armazenamento, além de informações de software, incluindo sistemas operacionais e aplicações. Esses dados são transferidos periodicamente para o servidor, onde são organizados e armazenados em um banco de dados central. A arquitetura modular permite a adição de recursos adicionais por meio de plugins e agentes personalizados, expandindo as funcionalidades da ferramenta para, por exemplo, descoberta de novos dispositivos na rede e execução de scripts automatizados de manutenção.

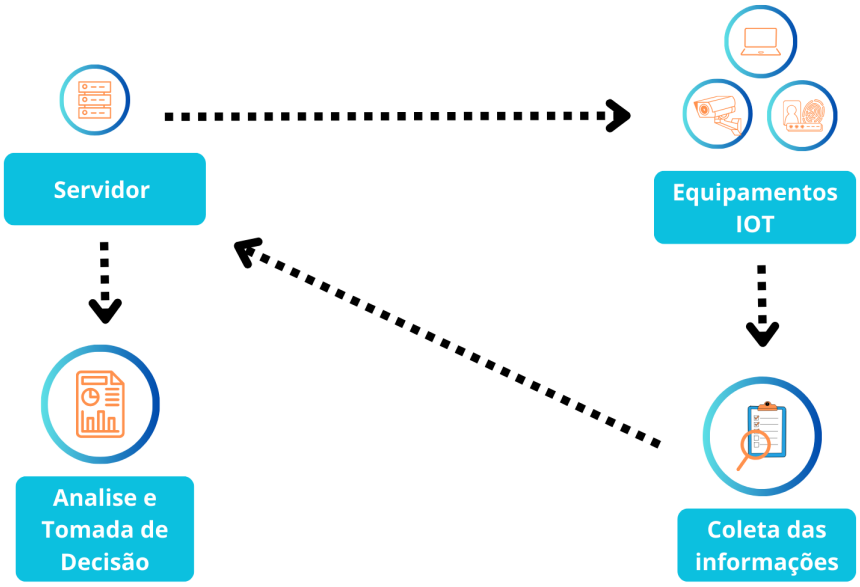

Figura 3.1: Processo de coleta de informações.

Fonte: Elaborado pelo autor (2024)

Como uma solução de código aberto voltada para a gestão de inventário de ativos de TI, o OCS permite o monitoramento e a administração eficiente dos recursos tecnológicos de uma organização. No entanto, a ferramenta apresenta limitações ao não abranger determinados hosts, especialmente dispositivos IoT. Para contornar essa limitação, foi desenvolvida uma solução personalizada em código, capaz de identificar, coletar e registrar automaticamente as informações dos dispositivos IoT na base de dados do inventário. Essa integração amplia significativamente o alcance do monitoramento, proporcionando uma visão mais completa da infraestrutura tecnológica da organização.

Durante o desenvolvimento da solução, foram adotadas as boas práticas de segurança recomendadas pelo framework NIST, garantindo que a coleta e o tratamento das informações sejam orientados com padrões reconhecidos de confidencialidade, integridade e disponibilidade. Essa abordagem contribui para um inventário mais seguro, confiável e alinhado às exigências de conformidade.

3.1 O MODELO DO SISTEMA

O modelo implementado e baseado na arquitetura Cliente/Servidor. conforme ilustrado na Figura 3.2, onde a comunicação ocorre por meio de requisições e respostas. Nesse modelo, há dois processos distintos: um executado no dispositivo Iot Cliente e outro na máquina servidor. O cliente, ao ser acionado, envia uma solicitação ao servidor por meio da rede, aguardando uma resposta. O servidor, ao receber essa requisição, processa a demanda, realiza as operações necessárias, como a coleta de informações sobre hardware e software dos dispositivos monitorados, e retorna os dados ao servidor. Esse funcionamento possibilita a centralização do gerenciamento de ativos de TI, permitindo maior controle e automação no monitoramento da infraestrutura.

Figura 3.2: Processo Cliente e Servidor

Fonte: Elaborado pelo autor (2024)

Ao integrar-se ao ambiente de monitoramento, o sistema viabiliza a automação da coleta de informações sobre hardware e software, reduzindo a necessidade de auditorias manuais e otimizando a identificação de ativos. Essa abordagem permite que os administradores de rede realizem análises preditivas, identifiquem vulnerabilidades e adotem medidas preventivas para manter a conformidade com as melhores práticas de governança de TI.

Além disso, a implementação desse modelo fortalece a adesão às diretrizes da NIST, que estabelece requisitos para o controle e segurança da informação em ambientes governamentais. A centralização do gerenciamento de ativos possibilita uma visão ampla e integrada da infraestrutura, permitindo a detecção de alterações indevidas, a gestão do ciclo de vida dos equipamentos e a aplicação de políticas de segurança com maior eficácia.

A conformidade com a NIST é reforçada pelo uso de um inventário atualizado e automatizado, garantindo que todos os dispositivos Iot estejam devidamente catalogados e monitorados em tempo real. Dessa forma, contribuindo diretamente para a governança da TI, auxiliando na mitigação de riscos, na conformidade regulatória e na implementação de controles internos alinhados às normas de segurança da informação

estabelecidas para o ambiente UIoT.

3.2 INTEGRAÇÃO MODULAR PARA COLETA DE DADOS EM AMBIENTES IOT

O ambiente estruturado no Laboratório UIoT da Universidade de Brasília foi concebido com base em uma arquitetura funcional organizada em três camadas principais: Camada de Dispositivos, Camada de Coleta e Camada de Ação. Essa divisão lógica visa garantir a modularidade, a escalabilidade e a eficiência na comunicação e no gerenciamento de ativos em redes com dispositivos de Internet das Coisas conforme ilustrado na Figura 3.3.

A Camada de Dispositivos representa a base da arquitetura, composta por diversos equipamentos conectados à rede, como sensores ambientais, câmeras IP, etiquetas RFID, atuadores, computadores e dispositivos móveis. São elementos heterogêneos que, em sua maioria, operam com capacidades limitadas de processamento e comunicação, o que dificulta a aplicação de agentes tradicionais de inventário.

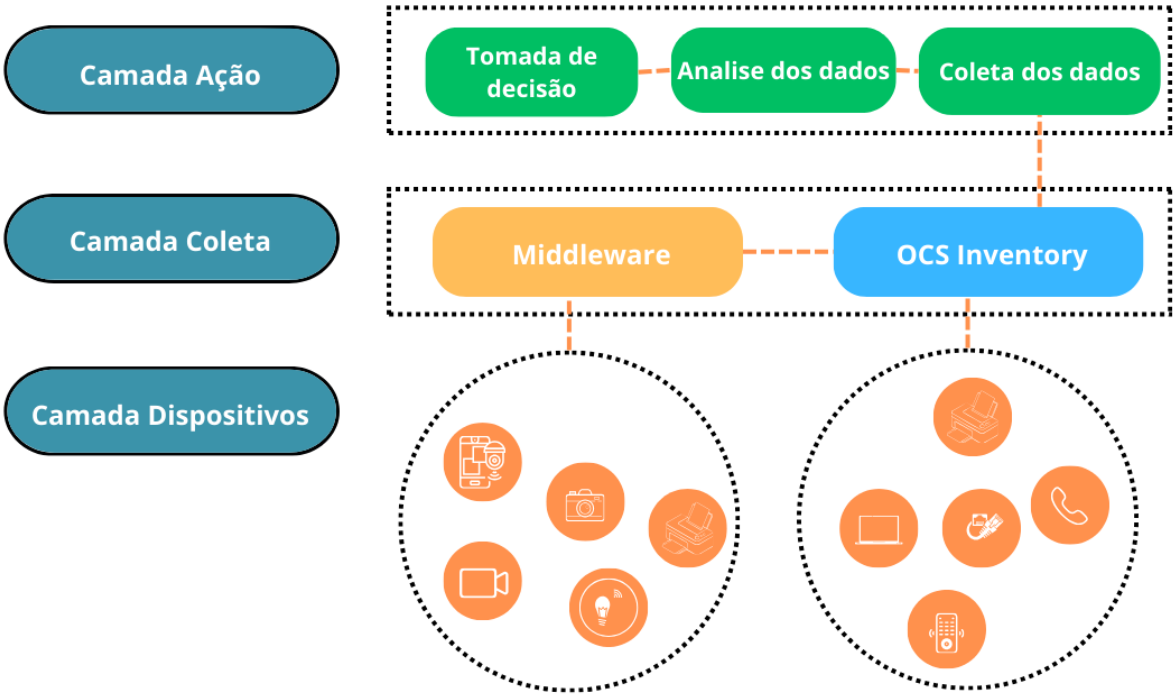

Figura 3.3: Arquitetura funcional do ambiente UIoT com integração em camadas para coleta e análise de dados IoT

Fonte: Elaborado pelo autor (2025)

Acima dela, encontra-se a Camada de Coleta, cujo papel é intermediar a comunicação entre os dispositivos físicos e as aplicações analíticas. Essa camada é composta pelo Middleware, que atua como integrador e gerenciador de protocolos e fluxos de dados, e pela ferramenta OCS Inventory, que foi adaptada e inserida estrategicamente nessa mesma camada. O OCS, originalmente projetado para inventariar computadores e servidores convencionais, foi modificado para se comunicar com dispositivos IoT utilizando protocolos

como mDNS e SSDP, passando assim a desempenhar um papel complementar ao middleware — contribuindo diretamente para a coleta e estruturação das informações de dispositivos que, até então, não eram monitorados.

Já a Camada de Ação é responsável pelo tratamento das informações extraídas, incluindo a análise dos dados, geração de relatórios e apoio à tomada de decisão. Os dados coletados pelo middleware e pelo OCS são direcionados a sistemas de análise, permitindo que gestores e pesquisadores realizem diagnósticos precisos e implementem ações corretivas com base em evidências.

Essa arquitetura modular permite não apenas uma visualização clara do fluxo de dados entre os diferentes componentes da rede, mas também promove a flexibilidade necessária para expandir ou substituir tecnologias sem comprometer o funcionamento do sistema. A integração entre middleware e OCS Inventory na camada intermediária constitui um diferencial deste projeto, ao viabilizar a coleta de inventário de forma automatizada, abrangente e alinhada com boas práticas de segurança.

3.1.1 Conformidade com as Normas

A conformidade com normas e regulamentos de segurança cibernética constitui um elemento fundamental para a gestão de ativos tecnológicos em ambientes corporativos, especialmente diante do crescimento dos dispositivos IoT . Diversos referenciais internacionais e nacionais estabelecem diretrizes que orientam a proteção, rastreabilidade e a governança de informações críticas.

O NIST Cybersecurity Framework (2018; 2024), amplamente adotado em escala global, estrutura a segurança em cinco funções centrais (Identify, Protect, Detect, Respond e Recover), destacando a importância da manutenção de inventários atualizados e da gestão contínua de riscos, conforme ilustrado na Figura 3.4.

Figura 3.4: Estrutura de funções NIST Cybersecurity Framework 2024

Fonte: NIST (2024, p. 05)[55]

Identificar: Cria um inventário detalhado dos ativos de TI, permitindo uma gestão eficiente e segura.

Proteger: Apoia na gestão de acessos e controle de configuração para mitigar riscos.

Detectar: Facilita a detecção de dispositivos e softwares não autorizados na rede.

Responder: Auxilia na resposta a incidentes por meio de registros detalhados e relatórios precisos.

Recuperar: Oferece suporte à recuperação de informações críticas ao garantir que os ativos estejam bem documentados.

A implementação das diretrizes do NIST permite uma abordagem mais proativa na gestão da segurança da informação, facilitando auditorias e garantindo conformidade com padrões de segurança reconhecidos internacionalmente.[55]

No contexto europeu, a *Network and Information Security Directive 2* (NIS2) amplia os requisitos de segurança para operadores de serviços essenciais e provedores digitais, impondo maior responsabilidade às organizações quanto à resiliência cibernética, incluindo monitoramento de ativos e incidentes.[56] Complementarmente, o *Cyber Resilience Act)* (CRA) introduz requisitos obrigatórios de segurança para produtos digitais e dispositivos conectados, garantindo que fabricantes e fornecedores incorporem segurança "by design" e mantenham atualizações ao longo do ciclo de vida do produto.[57]

No cenário brasileiro, a *Lei Geral de Proteção de Dados* (LGPD) reforça a necessidade de um inventário robusto de ativos, já que o tratamento de dados pessoais exige medidas técnicas e administrativas capazes de assegurar confidencialidade, integridade e disponibilidade das informações.[58]

Assim, a convergência entre esses marcos regulatórios NIST, NIS2, CRA e LGPD. Evidencia que o inventário de hardware, software e dispositivos IoT não deve ser entendido apenas como um processo operacional, mas como um requisito estratégico de conformidade e resiliência cibernética.

3.3 ANÁLISE DE DADOS

No estudo, serão analisadas métricas fundamentais para avaliar a eficácia da implementação do processo de inventario em conformidade com as diretrizes do NIST. Entre os principais indicadores, serão considerados a capacidade do sistema em detectar e monitorar ativos de TI, a eficiência na identificação e mitigação de softwares não autorizados, a redução do tempo médio de resposta a incidentes e a melhoria progressiva na conformidade com auditorias de segurança. Esses parâmetros fornecerão uma visão detalhada sobre o impacto do modelo adotado na gestão da segurança da informação, permitindo avaliar seu papel na redução de vulnerabilidades, no fortalecimento da governança de TI e no aprimoramento das práticas de monitoramento e controle dentro do UIoT .

- Capacidade de detecção e monitoramento de ativos de TI: Mede a eficiência em mapear, catalogar e acompanhar dispositivos conectados à rede. Essa métrica é essencial para garantir a visibilidade completa dos ativos tecnológicos e reduzir riscos associados a equipamentos não documentados.
- Identificação e mitigação de softwares não autorizados: Avalia a capacidade do sistema em detectar aplicações instaladas sem aprovação ou que representem vulnerabilidades à infraestrutura de TI. Essa métrica está diretamente ligada à segurança da informação, pois softwares não autorizados podem ser vetores de ataques e brechas de segurança.
- Tempo médio de resposta a incidentes: Indica a eficiência operacional na identificação e resolução

de incidentes relacionados a falhas de segurança, dispositivos não conformes ou tentativas de acesso indevido. Uma redução nesse tempo significa maior capacidade de mitigação de riscos e resposta rápida a ameaças emergentes.

- Conformidade com auditorias de segurança: Mede a aderência às regulamentações e normas de segurança da informação. O modelo implementado permite a rastreabilidade dos ativos e facilita a conformidade com exigências normativas, aumentando os índices de conformidade durante auditorias internas e externas.

3.4 ESTRUTURA DA PROPOSTA DO CENÁRIO DO MONITORAMENTO DO INVENTÁRIO

A primeira fase realizada no UIoT corresponde ao mapeamento do estado atual dos processos de inventário de hardware e software. Esta etapa teve como objetivo identificar as práticas adotadas pelo setor, com foco nos aspectos de rastreabilidade, atualização dos ativos e controle de dispositivos tecnológicos conectados à infraestrutura de rede conforme ilustrado na Figura 3.5.

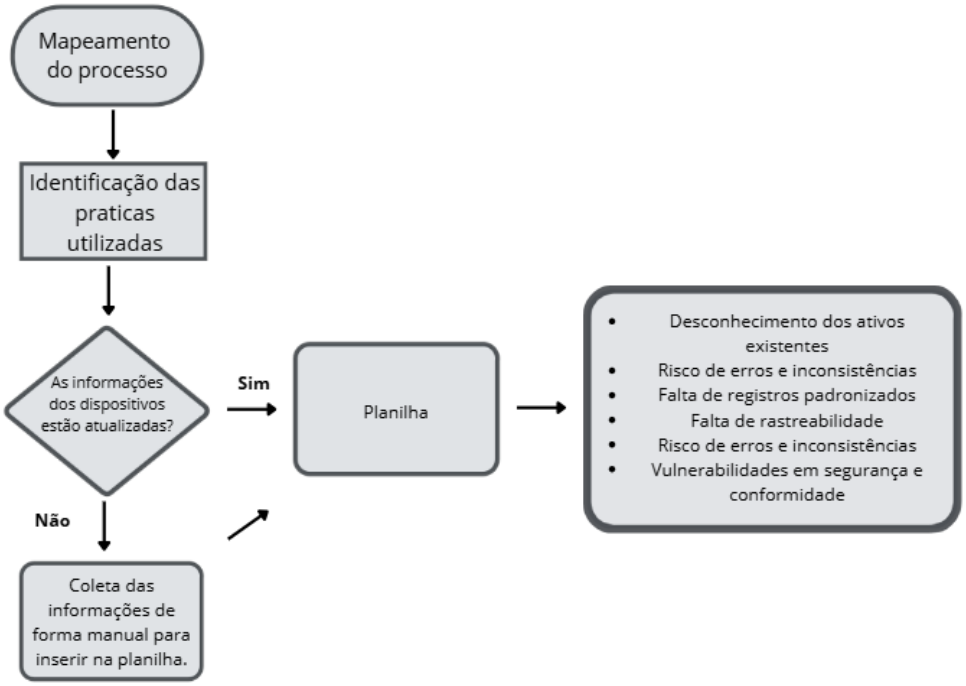

Figura 3.5: Diagrama do mapeamento do estado atual do processo de inventario

Fonte: Elaborado pelo autor (2025)

No entanto, constatou-se que não havia um inventário formalizado ou registros padronizados sobre os equipamentos presentes no ambiente. Diante dessa ausência de informações estruturadas, foi necessário iniciar o levantamento do zero, por meio de visitas in loco, identificação manual dos dispositivos conectados, e coleta direta de dados como fabricante, modelo, endereço MAC e localização física dos

equipamentos. Essa lacuna evidenciou a necessidade urgente de implementação da gestão de ativos no ambiente. Para isso, foram realizadas coletas de informações diretamente no setor analisado, com o levantamento dos equipamentos presentes, conforme Tabela 3.2 foram incluidos os dados como dispositivos, função principal, informações referentes ao Firmware e data da ultima atualização.

Dispositivo	Função Principal	Observações sobre o Firmware
Câmeras IP (Wi-Fi)	Monitoramento por vídeo e transmissão de imagens	Atualizações frequentes; vulnerável a ataques se desatualizado
Sensores ambientais	Monitoramento de fumaça e temperatura	Firmware básico, normalmente com atualizações via gateway central
Etiquetas RFID	Rastreio e identificação de ativos.	Firmware fixo; baixa necessidade de atualização.
Raspberry Pi	Controle, processamento e interface com sensores/atuadores.	Firmware flexível (bootloader) e dependente do sistema operacional
Lâmpadas inteligentes	Iluminação automatizada e controle via app .	Atualizações via nuvem; podem conter falhas de segurança conhecidas
Fechaduras eletrônicas	Controle de acesso físico .	Firmware crítico; requer autenticação segura e atualização constante
Controladores de acesso	Liberação de entrada com base em autenticação (cartão, biometria).	Firmware geralmente embarcado; foco em segurança e integridade.

Tabela 3.2: Referência da função e informações do Firmware dos equipamentos presentes no UIoT

Fonte: Elaborado pelo autor (2025)

Nessa mesma etapa foi realizado a coleta das informações mais detalhadas contendo informações como fabricante, modelo, número de série, endereço *Media Access Control address* (MAC), versão de firmware, localização física e data da ultima atualização conforme ilustrado na Figura 3.6.

Durante a coleta, observou-se a presença de diversos dispositivos IoT, como câmeras IP, sensores ambientais etiquetas RFID, microcomputadores (como Raspberry Pi), lâmpadas inteligentes, fechaduras eletrônicas e controladores de acesso. Esses dados permitiram uma análise crítica quanto à precisão das informações e à frequência das atualizações evidenciando tanto boas práticas quanto fragilidades nos processos atuais de inventário.

	Fabricante	Modelo	Número de Série	Endereço MAC	Versão de Firmware	Localização Física	Data da Última Atualizaçã
2	Raspberry Pi Foundation	Raspberry Pi 3	SN-517310nJ	2C:5B:4F:79:77:5F	3.3.45	Sala 102	13/04/2025
3	Honeywell	Sensepoint XCL	SN-396957Tn	15:3A:55:61:58:22	2.8.35	Sala 101	28/03/2024
4	Honeywell	Sensepoint XCL	SN-044325Vc	08:67:6A:7A:11:57	2.5.15	Sala TI	13/04/2025
5	HID Global	iCLASS SE	SN-808106mt	1B:11:3A:36:1E:0C	2.2.53	Auditório	30/09/2024
6	Yale	Linus Smart Lock	SN-331314JN	25:74:72:7F:3B:06	2.2.65	Auditório	11/02/2025
7	Xiaomi	Yeelight Color	SN-861456Ch	19:67:19:37:72:09	2.3.63	Laboratório 2	11/01/2024
8	HID Global	iCLASS SE	SN-221493KN	0F:39:27:20:5E:60	2.0.35	Sala 101	12/04/2024
9	Xiaomi	Yeelight Color	SN-695192yF	56:5A:0D:76:34:4C	3.4.89	Sala 101	09/01/2024
10	Honeywell	Sensepoint XCL	SN-602337Qn	50:06:18:69:0D:7E	2.0.22	Sala 205	02/01/2025
11	Honeywell	Sensepoint XCL	SN-061719nq	2E:6F:3B:3A:0C:27	3.7.35	Sala 101	20/07/2023
12	Raspberry Pi Foundation	Raspberry Pi 3	SN-783881Ga	16:69:24:4F:46:49	3.6.12	Sala TI	24/01/2025
13	Intelbras	Câmera IP VIP 5450 Z	SN-927847tS	6D:17:08:3F:62:33	3.1.50	Recepção	26/07/2024
14	Intelbras	Câmera IP VIP 5450 Z	SN-644155JE	26:7A:14:60:1E:21	2.6.33	Laboratório 1	26/03/2024

Figura 3.6: Tabela de controle de atualização UIot

Fonte: Elaborado pelo autor (2024)

A consolidação das informações foi essencial para entender o cenário real de controle dos dispositivos IoT. Com a base de dados criada, foi possível localizar fisicamente os equipamentos, acessá-los via rede para verificar dados em tempo real e registrar as versões de firmware por meio das interfaces de gerenciamento. As informações foram complementadas manualmente em planilhas e validadas com dados dos sites oficiais dos fabricantes, possibilitando a criação da base que sustentou os resultados apresentados na Tabela 3.4.

Localização dos equipamentos	Identificar fisicamente o equipamento no ambiente ou setor correspondente
Acesso aos equipamentos	Conectar-se ao equipamento pela rede corporativa utilizando IP ou nome de host.
Coleta dasinformações do Firmware	Verificar e registrar a versão atual do firmware via interface de gerenciamento.
Preenchimento manual da planilha	Consultar o site do fabricante para validar a versão do firmware e atualizações disponíveis.
Verificação das atualizações site oficial dos dispositivos Iot	Inserir os dados coletados na planilha de controle de inventário, mantendo registros atualizados.

Tabela 3.4: Fluxo de coleta de dados UIoT.

Fonte: Elaborado pelo autor (2025)

A partir do levantamento inicial realizado no setor UIoT , foram selecionados cinco dispositivos IoT distintos para a realização de testes práticos na rede local. A escolha dos equipamentos levou em consideração a diversidade de funcionalidades e fabricantes, de modo a representar diferentes perfis de operação e níveis de criticidade. Os dispositivos escolhidos foram: uma câmera IP da Intelbras, uma lâmpada inteligente da Xiaomi, dois microcomputador Raspberry Pi 3 com sistemas operacionais diferente e uma etiqueta RFID da Alien Technology conforme Tabela 3.6. Essa variedade permitiu avaliar como diferentes tipos de dispositivos se comportam em termos de comunicação, detecção na rede, exposição de informações e resposta aos métodos de coleta de inventário.

Com base na seleção dos dispositivos, deu-se início à aplicação prática dos testes no ambiente con-

Dispositivo	Fabricante	Modelo	Atualização	Sistema Operacional
Câmera IP	Intelbras	VIP 5450 Z	2022	NA
Lâmpada inteligente	Xiaomi	Yeelight Color	2022	NA
Microcomputador	Raspberry Pi Foundation	Raspberry Pi 3	2021	Windows 7
Microcomputador	Raspberry Pi Foundation	Raspberry Pi 3	2019	Ubuntu Server 20.04
Etiqueta RFID	Alien Technology	Higgs-EC	2017	NA

Tabela 3.6: Tabela 5 dispositivos coletados

Fonte: Elaborado pelo autor (2025)

trolado do setor UIoT. Para essa etapa, foi utilizado o software OCS, uma solução amplamente adotada para a gestão de inventário de ativos de TI, capaz de realizar varreduras automatizadas na rede local para identificar computadores, servidores e outros dispositivos conectados.

3.4.1 Análise de Eficiência da Coleta Manual

O ambiente de testes do OCS Inventory foi hospedado em uma instância virtual configurada com recursos suficientes para suportar operações intensivas de coleta e varredura de dispositivos. A infraestrutura utilizada contou com 4 vCPUs, 8 GB de memória RAM e 100 GB de armazenamento SSD, executando o sistema operacional Ubuntu Server 22.04 LTS.

Esse ambiente foi submetido a diferentes cenários de carga, sendo o principal deles o teste de descoberta simultânea de 50 máquinas conectadas, considerado o de maior gargalo do experimento. Nessa condição, observou-se o comportamento do sistema em termos de latência, utilização de recursos e tempo total de varredura, permitindo avaliar a eficiência e a escalabilidade da arquitetura proposta para inventários em larga escala.

A execução dos testes foi estruturada de forma incremental, em três etapas sucessivas, inicialmente com um único computador, em seguida com cinco dispositivos heterogêneos e, por fim, com dez equipamentos. Essa abordagem foi escolhida para garantir controle experimental, reprodutibilidade dos resultados e avaliação progressiva da complexidade operacional.

O primeiro cenário, envolvendo apenas um computador, teve como objetivo validar o procedimento básico de coleta e registro de informações. Essa etapa inicial permitiu confirmar o funcionamento do processo manual de inventário, assegurando que as ferramentas e comandos empregados fossem capazes de identificar corretamente os parâmetros essenciais (sistema operacional, memória, armazenamento e

tempo de coleta) conforme a Tabela 3.7.

Dispositivo	Características	Tempo Total
Notebook Dell	Ubuntu 22.04, 8 GB RAM, SSD 256 GB	7 minutos

Tabela 3.7: Teste Manual – 1 Máquina

Fonte: Elaborado pelo autor (2025)

Na segunda fase, com cinco dispositivos, buscou-se avaliar o comportamento do método diante de um ambiente mais heterogêneo, contendo equipamentos com diferentes sistemas embarcados e protocolos de comunicação (como Raspbian OS, Tizen OS e firmware proprietário). Esse estágio intermediário permitiu analisar o impacto da diversidade tecnológica no tempo de execução e na precisão dos dados coletados conforme a Tabela 3.8.

Dispositivo	Características Principais	Tempo Total
Câmera IP Genérica	Firmware próprio, acesso via navegador	**52 minutos**
Raspberry Pi 3 – 1	Raspbian OS, 1 GB RAM, microSD 32 GB	
Raspberry Pi 3 – 2	Raspbian OS, 1 GB RAM, microSD 32 GB	
Impressora Epson XP-2100	Conectividade via IP e LPD	
Smart TV Samsung	Tizen OS, conectada via Wi-Fi	

Tabela 3.8: Teste Manual – 5 Dispositivos

Fonte: Elaborado pelo autor (2025)

Por fim, o terceiro cenário, composto por dez dispositivos, foi projetado para simular uma condição mais próxima de um ambiente real de rede IoT, incluindo sensores, microcontroladores e equipamentos de diferentes fabricantes. Essa ampliação incremental proporcionou uma avaliação da escalabilidade e da eficiência do processo manual frente ao aumento do número de ativos monitorados, além de evidenciar as limitações práticas desse modelo conforme a Tabela 3.9.

De forma geral, a adoção dessa metodologia escalonada permitiu avaliar a correlação entre o número de dispositivos e o tempo total de coleta (conforme mostrado na Tabela 3.10), fundamentando a necessidade de automação do processo de inventário proposta nesta pesquisa.

Cenário	Quantidade de Dispositivos	Tempo Total	Tempo Médio por Dispositivo
Teste 1	1 (Notebook Dell)	7 minutos	7 min/dispositivo
Teste 2	5 (IoT + periféricos)	52 minutos	10,4 min/dispositivo
Teste 3	10 (IoT + PCs + sensores)	138 minutos	13,8 min/dispositivo

Tabela 3.10: Comparativo entre os três testes manuais de descoberta de dispositivos.

Dispositivo	Características Principais	Tempo Total
Raspberry Pi 3 – 1	Raspbian OS, 1 GB RAM, microSD 32 GB	**138 minutos**
Raspberry Pi 3 – 2	Raspbian OS, 1 GB RAM, microSD 32 GB	
Smart TV Samsung	Tizen OS, conectada via Wi-Fi	
Smart TV LG	WebOS, conectada via Wi-Fi	
Câmera IP Intelbras	Interface web, acesso via IP local	
Impressora Epson XP-2100	LPD/IPP ativo, detecção via rede	
Sensor Ambiental DHT22	Acesso via GPIO em microcontrolador	
Etiqueta RFID conectada	Sem sistema operacional, responde via serial	
Microcontrolador de automação	Controle via protocolo proprietário	
Notebook Dell com Ubuntu	Acesso SSH e coleta tradicional	

Tabela 3.9: Teste Manual – 10 Dispositivos

Fonte: Elaborado pelo autor (2025)

3.4.2 Escaneamento da Rede UIoT com a Estrutura Padrão do OCS Inventory

Inicialmente, OCS foi configurado para escanear o segmento de rede onde os cinco dispositivos IoT estavam conectados. O sistema conseguiu detectar parcialmente a presença desses dispositivos na rede, identificando alguns endereços IP e MAC Address, contudo, não foi capaz de coletar informações detalhadas dos equipamentos Conforme ilustrado na Figura 3.2, como versão do firmware, fabricante, serviços em execução ou status do sistema.

IP address	MAC address	Netmask	Date
172.16.4.128	00:D7:6D:5A:9F:C4 (Intel Corporate)	255.255.255.0	2025-03-31 15:08:42
172.16.4.62	00:D7:6D:83:A9:5F (Intel Corporate)	255.255.255.0	2025-03-31 17:05:17
172.16.4.141	00:D7:6D:83:A9:69 (Intel Corporate)	255.255.255.0	2025-04-26 04:31:34
172.16.4.123	00:E0:4C:6A:B8:6B (REALTEK SEMICONDUCTOR CORP)	255.255.255.0	2025-03-28 14:24:15
172.16.4.2	14:18:77:4E:98:37 (Dell Inc)	255.255.255.0	2025-04-26 04:31:34
172.16.4.56	28:F1:0E:FC:99:E3 (Dell Inc)	255.255.255.0	2025-03-19 14:36:54
172.16.4.27	50:B3:63:00:C8:46 (Digitron da Amazonia S/A)	255.255.255.0	2025-04-26 04:31:34
172.16.4.4	56:6F:2A:4D:00:20	255.255.255.0	2025-04-26 04:31:34
172.16.4.7	56:6F:2A:4D:00:35	255.255.255.0	2025-04-26 04:31:34
172.16.4.38	56:6F:2A:4D:00:D9	255.255.255.0	2025-04-26 04:31:34

Figura 3.7: Processo de varredura rede UIoT

Fonte: Elaborado pelo autor (2025)

Essa limitação ocorre principalmente devido à arquitetura dos dispositivos IoT, que muitas vezes não possuem sistemas operacionais compatíveis com os agentes do OCS, tampouco oferecem suporte a protocolos padrão de gerenciamento como *Simple Network Management Protocol* (SNMP), *Secure Shell* (SSH) ou *Windows Management Instrumentation* (WMI). Além disso, muitos desses dispositivos são projetados com foco em funcionalidade específica e segurança restrita, o que reduz significativamente sua capacidade de resposta a solicitações externas de inventário.

Durante os testes, também foram observadas variações no comportamento de resposta entre os dispositivos, especialmente naqueles com firmware fechado (como a câmera IP e a lâmpada inteligente), em contraste com os microcomputadores Raspberry Pi, que apresentaram maior abertura para coleta por utilizarem sistemas operacionais baseados em Linux e Windows.

Uma vez concluída a varredura no ambiente experimental, foi possível identificar os dispositivos que responderam aos critérios de detecção definidos pelo sistema. A Tabela 3.12 apresenta os resultados obtidos, destacando os dispositivos que foram efetivamente encontrados. Dos cinco dispositivos IoT previamente mapeados, apenas dois foram detectados com sucesso: um microcomputador Raspberry Pi 3 com sistema operacional Windows e outro com distribuição Linux.

Dispositivo	Status	Modelo	IP	S. O.
Câmera IP	N/A	VIP 5450 Z	N/A	NA
Lâmpada inteligente	N/A	Yeelight Color	N/A	NA
Microcomputador	Encontrado	Raspberry Pi 3	172.16.4.128	Windows
Microcomputador	Encontrado	Raspberry Pi 3	172.16.4.38	Ubuntu
Etiqueta RFID	N/A	Higgs-EC	N/A	NA

Tabela 3.12: Tabela dispositivos que conseguiram ser coletados

Fonte: Elaborado pelo autor (2025)

Os demais dispositivos não responderam às tentativas de comunicação, o que pode estar relacionado a limitações de protocolo, ausência de agentes compatíveis ou configurações específicas de rede.

Antes da implementação das modificações necessárias para a varredura de dispositivos IoT, é importante compreender a estrutura padrão utilizada pelo OCS. O trecho de código apresentado a seguir representa o funcionamento original da ferramenta, responsável pela varredura e coleta de informações dos equipamentos conectados à rede. Essa estrutura base serve como ponto de partida para a adaptação do código, que posteriormente será ajustado com o objetivo de ampliar sua capacidade de detecção, passando a identificar também dispositivos IoT, os quais, em sua forma padrão, não são contemplados integralmente pelo sistema.

Dentro dessa estrutura original, a varredura de rede é realizada por meio de dois métodos principais: *Network Mapper* (Nmap) e *Fast Ping* (fping) conforme ilustrado nos Algoritmos 1 e 2. Esses mecanismos estão implementados no script ipdiscover, que tem como finalidade localizar dispositivos ativos em subredes especificadas e registrar os dados correspondentes no banco de dados do sistema.

Algoritmo 1: Pseudocódigo referente à varredura Nmap e inserção no banco de dados

```
Input: scan_type, ip_range
Output: Registros de dispositivos inseridos em netmap
if scan_type = "nmap" then
    Executar comando: nmap -sn ip_range;
    foreach bloco no resultado do scan do
        ip ← extrair_ip(bloco);
        name ← extrair_nome(bloco) (se existir);
        mac ← extrair_mac(bloco) (se existir);
        mask ← calcular_máscara(ip);
        subnet ← calcular_subrede(ip, mask);
        Remover espaços de ip, mac, name, mask, subnet;
        Exibir mensagem: "Adicionando ip";
        if name e tag existem then
            Inserir em netmap (IP, MAC, MASK, NETID, NAME, TAG);
        else
            if name existe then
                Inserir em netmap (IP, MAC, MASK, NETID, NAME);
            else
                if tag existe then
                    Inserir em netmap (IP, MAC, MASK, NETID, TAG);
                else
                    Inserir em netmap (IP, MAC, MASK, NETID);
```

O método baseado em Nmap utiliza o parâmetro -sn, realizando uma varredura do tipo ping scan, que identifica hosts acessíveis na rede sem efetuar varreduras de portas. A partir dos resultados obtidos, o sistema extrai e armazena dados como endereço IP, nome da máquina (quando disponível) e endereço MAC na tabela netmap.

De forma alternativa, o fping é utilizado para varreduras ICMP rápidas e eficientes. Através do comando fping -g, o script percorre uma faixa de endereços IP, registrando todos os dispositivos que respondem às requisições ICMP. Assim como no método anterior, as informações obtidas são inseridas na base de dados para posterior análise.

Algoritmo 2: Pseudocódigo referente à varredura com fping

```
Input: Faixa de endereços IP
Output: Dispositivos ativos inseridos na base de dados
if fping está instalado then
    Executar comando: fping -g -quiet -a ip_range;
    foreach endereço IP respondendo ao fping do
        ip ← endereço retornado;
        macAddr ← ip ;              // fping não retorna MAC, usa o IP como
         referência
        Remover espaços de ip, macAddr, mask;
        Exibir mensagem: "Adicionando ip";
        if tag existe then
            Inserir em netmap (IP, MAC, MASK, NETID, TAG);
        else
            Inserir em netmap (IP, MAC, MASK, NETID);
else
    Exibir mensagem de erro: "Instale o fping ou use -scantype=nmap";
```

Essas abordagens compõem o processo de descoberta ativa padrão do OCS, sendo fundamentais para detectar máquinas conectadas à rede, mesmo na ausência de um agente instalado. No entanto, tais mecanismos ainda são limitados quanto à detecção específica de dispositivos IoT, exigindo, portanto, adaptações futuras que serão descritas nas próximas seções.

3.4.3 Ampliação da Descoberta de Dispositivos por mDNS no OCS Inventory

Com o objetivo de ampliar a capacidade do OCS em identificar dispositivos IoT na rede, foi implementada uma rotina de varredura utilizando o protocolo *Multicast Domain Name System* (mDNS), amplamente utilizado por dispositivos domésticos e industriais para anúncio de serviços em redes locais conforme ilustrado nos Algoritmos 3 . Esse protocolo permite descobrir dispositivos que publicam seus nomes e serviços sem depender de um servidor *Domain Name System* (DNS) tradicional, sendo muito comum em dispositivos da Apple, Google, impressoras de rede, câmeras IP, smart TVs, entre outros.

Algoritmo 3: Pseudocódigo referente à coleta de dispositivos IoT via mDNS

Input: Rede local com suporte a mDNS

Output: Dispositivos IoT descobertos e inseridos no banco de dados

Executar comando: `avahi-browse -a -t | grep IPv4`;

foreach *linha do resultado* **do**

  if *linha contém padrão de IPv4 e nome do serviço* **then**

    iot_name ← nome do dispositivo extraído;

    iot_ip ← endereço IPv4 extraído;

    Exibir mensagem: "`Dispositivo IoT encontrado: iot_name em iot_ip`";

    Inserir no banco de dados (IP, NAME, NETID, TAG) com tag = `IoT-mDNS`;

A varredura via mDNS foi incorporada diretamente no script ipdiscover, dentro da estrutura de varredura ativa localizada no trecho correspondente ao método Nmap. A inserção ocorreu após o bloco de código responsável pelo tratamento da variável scantype eq "nmap", de forma a complementar o processo de descoberta já existente, sem interferir na lógica original da ferramenta.

Esse código faz uso da ferramenta avahi-browse, amplamente disponível em distribuições Linux, para listar serviços anunciados por dispositivos mDNS . Após processar a saída do comando, os dispositivos identificados são inseridos na tabela netmap, associando o IP , nome do dispositivo e a tag indicativa de que a descoberta foi feita via mDNS.

A inclusão desta rotina permitiu identificar novos dispositivos na rede que antes passavam despercebidos pelas abordagens convencionais baseadas em ICMP ou Nmap . Câmeras IP, impressoras inteligentes e dispositivos de automação residencial foram detectados com maior precisão, representando um avanço significativo na abrangência da coleta de dados pelo OCS conforme ilustrado nos Algoritmos 3.

IP address	MAC address	Netmask	Date
172.16.4.128	00:D7:6D:5A:9F:C4 (Intel Corporate)	255.255.255.0	2025-03-31 15:08:42
172.16.4.62	00:D7:6D:83:A9:5F (Intel Corporate)	255.255.255.0	2025-03-31 17:05:17
172.16.4.141	00:D7:6D:83:A9:69 (Intel Corporate)	255.255.255.0	2025-04-26 04:31:34
172.16.4.123	00:E0:4C:6A:B8:6B (REALTEK SEMICONDUCTOR CORP)	255.255.255.0	2025-03-28 14:24:15
172.16.4.2	14:18:77:4E:98:37 (Dell Inc)	255.255.255.0	2025-04-26 04:31:34
172.16.4.56	28:F1:0E:FC:99:E3 (Dell Inc)	255.255.255.0	2025-03-19 14:36:54
172.16.4.27	50:B3:63:00:C8:46 (Digitron da Amazonia S/A)	255.255.255.0	2025-04-26 04:31:34
172.16.4.4	56:6F:2A:4D:00:20	255.255.255.0	2025-04-26 04:31:34
172.16.4.7	56:6F:2A:4D:00:35	255.255.255.0	2025-04-26 04:31:34
172.16.4.38	56:6F:2A:4D:00:D9	255.255.255.0	2025-04-26 04:31:34

Figura 3.8: Varredura da rede UIoT com mDNS

Fonte: Elaborado pelo autor (2025)

No entanto, embora a solução baseada em mDNS tenha expandido a visibilidade sobre dispositivos IoT, foi observado que nem todos os equipamentos utilizam esse protocolo para se anunciar na rede. Muitos dispositivos industriais, sensores embarcados ou equipamentos com foco em baixo consumo operam de forma passiva ou utilizam outros padrões de descoberta, como SSDP/UPnP, CoAP, Zeroconf ou protocolos proprietários baseados em gateways.

Essa limitação evidencia que a coleta de dados ainda está em evolução e que, para obter um inventário realmente abrangente e confiável, será necessário incorporar técnicas complementares de varredura nas etapas seguintes desse projeto.

3.4.4 Extensão da Varredura com SSDP para Descoberta de Dispositivos IoT

Com o objetivo de complementar a varredura baseada em mDNS e ampliar a cobertura de dispositivos IoT detectáveis na rede, foi implementada uma rotina de descoberta utilizando o *Simple Service Discovery Protocol* (SSDP) protocolo base do padrão *Universal Plug and Play* (UPnP) conforme ilustrado nos Algoritmos 4.

O SSDP permite que dispositivos em uma rede local anunciem seus serviços ou respondam a buscas ativas feitas por outros dispositivos. Essa abordagem é especialmente eficaz para identificar equipamentos como câmeras IP, smart TVs, impressoras em rede, roteadores domésticos, caixas de som inteligentes e diversos dispositivos de automação residencial.

A coleta foi implementada no mesmo script de varredura ipdiscover, adicionando-se uma rotina específica para envio de requisições SSDP via multicast e leitura de suas respostas. O conteúdo recebido é processado para extrair o endereço IP do dispositivo e algumas informações descritivas, como a localização do serviço ou modelo anunciado. Dispositivos detectados são então inseridos na tabela netmap do banco de dados, identificados com a tag IoT-SSDP.

Algoritmo 4: Pseudocódigo referente à coleta de dispositivos IoT via SSDP (UPnP)

Input: Endereço multicast padrão SSDP (239.255.255.250:1900)
Output: Dispositivos IoT descobertos e registrados no banco de dados
Inicializar socket multicast UDP na porta 1900;
Entrar no grupo multicast SSDP;
Construir mensagem M-SEARCH:

- HOST: 239.255.255.250:1900
- MAN: "ssdp:discover"
- MX: 3
- ST: ssdp:all

Enviar mensagem M-SEARCH para endereço multicast;
Definir tempo limite de espera (timeout);
while *tempo atual* < *timeout* **do**
    Escutar por respostas no socket;
    if *resposta recebida contém campo LOCATION* **then**
        Extrair IP e descrição do dispositivo;
        Exibir mensagem: "Dispositivo SSDP encontrado: ip";
        Inserir no banco de dados (IP, NAME, NETID, TAG) com tag = IoT-SSDP;

Essa integração permitiu ampliar a visibilidade de dispositivos IoT que não utilizam mDNS, avançando significativamente na composição de um inventário mais completo conforme ilustrado na Figura 3.9.

IP address	MAC address	Netmask	Date
172.16.5.172	A0:59:50:80:62:4A (Intel Corporate)	255.255.255.0	2025-04-25 19:44:16
172.16.5.137	A2:7A:39:34:1B:AE	255.255.255.0	2025-04-14 15:19:51
172.16.5.104	A2:A4:F1:E1:35:68	255.255.255.0	2025-04-25 13:20:28
172.16.5.104	A4:4E:31:7A:78:B4 (Intel Corporate)	255.255.255.0	2025-04-01 11:42:10
172.16.5.194	A4:4E:31:7A:83:98 (Intel Corporate)	255.255.255.0	2025-04-25 19:44:16
172.16.5.150	A6:0C:3B:BC:66:70	255.255.255.0	2025-03-31 19:36:31
172.16.5.109	A8:B1:3B:A9:80:98 (HP Inc)	255.255.255.0	2025-04-29 21:19:19
172.16.5.251	AA:BB:CC:00:01:20	255.255.255.0	2025-04-23 13:52:55
172.16.5.145	AC:5A:FC:D0:0C:C8 (Intel Corporate)	255.255.255.0	2025-04-10 13:38:18
172.16.5.158	AE:2D:D7:7D:9B:53	255.255.255.0	2025-04-16 13:16:12

Figura 3.9: Varredura da rede UIoT com mDNS e SSDP

Fonte: Elaborado pelo autor (2025)

No entanto, a natureza heterogênea das redes IoT ainda exige o uso combinado de múltiplos protocolos para garantir a abrangência da descoberta.

3.4.5 Impacto da Integração de Protocolos de Descoberta na Coleta de Inventário

A implementação das rotinas de varredura baseadas nos protocolos mDNS e SSDP representou um avanço significativo no processo de identificação de dispositivos IoT no ambiente de rede monitorado pelo OCS. Ambas as abordagens foram incorporadas ao script de varredura ipdiscover, complementando os mecanismos tradicionais baseados em Nmap e fping e permitindo a detecção de dispositivos que não respondem a requisições ICMP ou que não expõem serviços em portas comuns.

O protocolo mDNS mostrou-se eficaz na descoberta de dispositivos que anunciam seus serviços em redes locais por meio do padrão Zeroconf, como câmeras IP, dispositivos Apple, hubs de automação residencial e sensores compatíveis com HomeKit ou Google Home. Já o SSDP possibilitou identificar dispositivos que operam sob o padrão UPnP, como TVs inteligentes, impressoras de rede, roteadores domésticos e reprodutores multimídia, os quais não apareciam em varreduras anteriores.

A inserção dessas técnicas no fluxo de varredura permitiu aumentar significativamente o número de dispositivos coletados durante o inventário, especialmente em redes heterogêneas como a UIoT, onde a presença de dispositivos embarcados e soluções conectadas é elevada. Testes realizados após a implementação indicaram que dispositivos anteriormente invisíveis passaram a ser detectados e registrados corretamente na base de dados do sistema, com identificação de IP, nome de serviço e protocolo de origem da descoberta.

Apesar dos avanços obtidos, os resultados também evidenciaram que nenhuma das abordagens isoladamente é capaz de oferecer uma cobertura total do ecossistema IoT, devido à diversidade de protocolos utilizados por fabricantes e à existência de dispositivos que não respondem ativamente a nenhum tipo de anúncio de rede. Assim, embora a cobertura tenha sido consideravelmente ampliada, a taxa de detecção ainda não alcança 100 dos ativos conectados.

3.4.6 Avaliação da Varredura Após Ampliação Para Dez Dispositivos IoT

Com o propósito de ampliar o escopo de avaliação e assegurar que a arquitetura proposta fosse testada em um ambiente mais representativo de redes heterogêneas, esta etapa metodológica compreendeu a expansão controlada do conjunto de dispositivos IoT utilizados nas varreduras experimentais. O objetivo central foi avaliar o comportamento das rotinas de descoberta diante de diferentes categorias de equipamentos, protocolos de comunicação e perfis de operação em rede.

Inicialmente, foram definidos cinco dispositivos para compor a base de testes preliminares, de modo a validar as rotinas básicas de coleta e integração com o servidor do *OCS Inventory*. A partir da consolidação das rotinas baseadas em *Multicast DNS* (mDNS) e *Simple Service Discovery Protocol* (SSDP), o conjunto foi ampliado para um total de dez dispositivos distintos, abrangendo uma maior diversidade de fabricantes e sistemas embarcados.

A seleção dos dispositivos considerou a representatividade de diferentes classes tecnológicas presentes em redes IoT contemporâneas, incluindo sensores ambientais, atuadores, câmeras inteligentes, microcontroladores embarcados, lâmpadas conectadas, roteadores domésticos e impressoras de rede com suporte a protocolos de descoberta. Essa variedade assegurou a criação de um cenário experimental que reproduz, de forma controlada, a heterogeneidade de um ambiente real de Internet das Coisas.

A configuração da varredura foi executada mediante a integração das quatro técnicas previamente validadas — **ICMP** (*fping*), **Nmap**, **mDNS** e **SSDP/UPnP** — compondo um mecanismo híbrido de descoberta capaz de identificar dispositivos por múltiplos métodos de sondagem. Cada técnica foi parametrizada para operar de maneira complementar, possibilitando a observação de diferentes camadas e respostas de rede.

Dispositivo	Status	Modelo	Endereço IP	Sistema Operacional
Lâmpada inteligente	Ativo	Yeelight Color	172.16.4.115	N/A
Microcomputador	Ativo	Raspberry Pi 3	172.16.4.38	Ubuntu
Sensor de gás	Inativo	MQ-2	N/A	N/A
Impressora UPnP	Ativo	Epson L3150	172.16.4.91	Não identificado
Smart TV	Ativo	Samsung UN43	172.16.4.67	Tizen
Sensor Zigbee	Inativo	Aqara Temp	N/A	N/A
Microcomputador	Ativo	Raspberry Pi 3	172.16.4.128	Windows
Câmera IP	Inativo	VIP 5450 Z	N/A	N/A
Fechadura inteligente	Ativo	Yale Linus	172.16.4.53	N/A
Etiqueta RFID	Inativo	Alien Higgs-EC	N/A	N/A

Tabela 3.14: Dispositivos analisados na fase de ampliação experimental do ambiente de testes

Fonte: Elaborado pelo autor (2025)

A relação completa dos dispositivos incluídos nesta fase experimental está apresentada na Tabela 3.14, que sintetiza os principais equipamentos considerados, juntamente com suas características e parâmetros de identificação. Essa configuração serviu de base para as etapas subsequentes de análise, que avaliaram a eficiência, cobertura e integração dos mecanismos automatizados de inventário.

Durante os experimentos iniciais, constatou-se que a câmera IP não foi localizada pelas rotinas de varredura do OCS, embora estivesse conectada à rede de testes. A análise revelou configurações incorretas de rede, com endereço IP e gateway fora do intervalo definido, o que impedia a comunicação com os protocolos de descoberta (ICMP, mDNS e SSDP). Após o reajuste desses parâmetros, o dispositivo passou a responder corretamente, sendo identificado na etapa final da varredura. Esse incidente evidenciou a importância da validação prévia das configurações de rede em dispositivos IoT para garantir a eficácia dos mecanismos automatizados de descoberta e inventário.

3.4.7 Ampliação do Inventário: Análise de Detecção em um Cenário com 50 Dispositivos IoT

Dando continuidade ao desenvolvimento da arquitetura de inventário automatizado de ativos, esta etapa concentrou-se na estruturação e ampliação do ambiente experimental no Laboratório UIoT, com o objetivo de avaliar a escalabilidade das técnicas de varredura e detecção em redes heterogêneas. O ambiente foi projetado para reproduzir condições próximas às encontradas em ecossistemas IoT reais, caracterizados pela coexistência de dispositivos com diferentes capacidades de processamento, sistemas operacionais e protocolos de comunicação.

O processo de construção do ambiente envolveu o escalonamento gradual do número de dispositivos conectados, partindo dos cenários iniciais de testes manuais (1, 5 e 10 dispositivos) até atingir um total de 50 unidades interconectadas. Essa progressão permitiu estabelecer um modelo de avaliação controlado, no qual o aumento de complexidade foi acompanhado pela validação incremental dos procedimentos de descoberta, coleta e normalização.

A configuração buscou deliberadamente a heterogeneidade tecnológica, contemplando múltiplas categorias de equipamentos:

- *Sensores e atuadores*: sensores ambientais (p. ex., MQ-2) e dispositivos Zigbee (Aqara);
- *Dispositivos de vídeo e automação residencial*: câmeras IP (Intelbras), lâmpadas inteligentes (Xiaomi) e fechaduras inteligentes;
- *Dispositivos de conectividade e middleware*: roteadores domésticos e impressoras de rede compatíveis com UPnP;
- *Dispositivos embarcados e multimídia*: microcomputadores Raspberry Pi (Linux e Windows), smart TVs;
- *Identificação e logística*: etiquetas RFID (Alien Technology).

A infraestrutura de varredura foi implementada a partir de quatro técnicas complementares de descoberta, previamente integradas e validadas nas fases iniciais do estudo:

1. **ICMP** (*fping*) para sonoridade básica de hosts em sub-redes definidas;
2. **Nmap** para sondagem ativa de portas/serviços e fingerprint de pilha TCP/IP;
3. **mDNS** para resolução e anúncio em multicast, típico de serviços locais;
4. **SSDP/UPnP** para descoberta de dispositivos e serviços em redes domésticas e corporativas.

Cada técnica foi parametrizada de modo a operar de forma coordenada, compondo um mecanismo de coleta híbrido que abrange abordagens ativas e passivas de identificação. Os metadados obtidos (endereçamento, serviços anunciados e origem do protocolo de detecção) são consolidados no repositório do *OCS Inventory* por meio de rotinas de normalização e correlação, assegurando unicidade e rastreabilidade dos registros.

Por fim, o ambiente e os procedimentos foram documentados segundo critérios de reprodutibilidade e controle experimental, incluindo topologias de rede, versões de firmware, parâmetros de execução e configurações específicas de cada conjunto de testes. Essa documentação garante a consistência das análises subsequentes e oferece subsídios para a interpretação dos resultados reportados em seções posteriores.

Para fins de documentação e rastreabilidade do escopo, a relação de dispositivos que, nesta configuração, não se enquadram nas técnicas de descoberta consideradas (por uso de protocolos não suportados, comportamento passivo ou requisitos de autenticação), encontra-se sintetizada na Tabela 3.16

Dispositivo	Modelo	Status	IP	S. O.
Sensor industrial	Siemens S7 Modbus	N/A	N/A	NA
Sensor RFID	Motorola FX9500	N/A	N/A	NA
Controle de energia	Shelly EM	N/A	N/A	NA
Placa embarcada	BeagleBone Black	N/A	N/A	NA
Sensor de corrente	ACS712	N/A	N/A	NA
Dispositivo LoRa	Heltec WiFi LoRa 32	N/A	N/A	NA

Tabela 3.16: Dispositivos não detectados durante a varredura no Laboratório UIoT

Fonte: Elaborado pelo autor (2025)

4 RESULTADOS

4.0.1 Evolução dos Testes e Estratégias de Descoberta

No presente estudo, antes da implementação das estratégias automatizadas, foram conduzidos testes manuais para a coleta de informações de dispositivos em rede no ambiente do Laboratório UIoT. Inicialmente, a coleta foi realizada em uma única máquina, utilizando comandos manuais e scripts locais para registrar dados como processador, memória, sistema operacional e softwares instalados. Na sequência, o mesmo procedimento foi repetido com cinco e, posteriormente, com dez máquinas. Em cada etapa, observou-se um crescimento considerável no tempo necessário para a coleta, revelando as limitações do método manual em termos de escalabilidade, precisão e esforço operacional. Essa abordagem inicial não apenas evidenciou os desafios de um inventário não automatizado, mas também serviu como linha de base comparativa para avaliar a eficiência das soluções desenvolvidas ao longo do projeto.

Diante dessas dificuldades, iniciou-se um processo contínuo de evolução na estratégia de descoberta e inventário de dispositivos IoT no ambiente do Laboratório UIoT. Conforme descrito nas seções anteriores, a primeira iniciativa consistiu na implementação do mecanismo padrão de varredura do *OCS Inventory*, baseado nas ferramentas ICMP (*fping*) e Nmap. Essa solução se mostrou eficaz para a detecção de hosts convencionais, mas limitada em relação a dispositivos embarcados e de automação residencial. Para superar essa limitação, foram integradas duas novas rotinas de descoberta, baseadas nos protocolos mDNS e SSDP (*UPnP*), com o objetivo de ampliar a visibilidade sobre dispositivos IoT heterogêneos que não respondem a métodos tradicionais.

Essas técnicas foram cuidadosamente integradas ao fluxo de varredura da ferramenta e validadas por meio de ciclos experimentais sucessivos. Cada fase representou um avanço incremental na complexidade e abrangência do ambiente de testes, desde a coleta manual em máquinas individuais até a integração de múltiplos protocolos em um cenário composto por cinquenta dispositivos heterogêneos. A Figura 4.1 sintetiza essa progressão metodológica e destaca a relação entre o número de dispositivos analisados, os métodos empregados e o grau de automação alcançado em cada etapa.

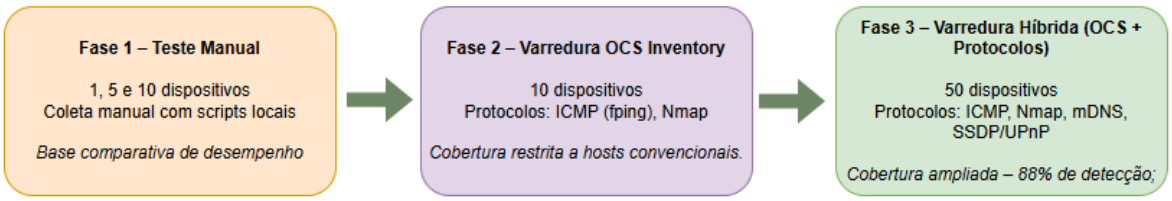

Figura 4.1: Evolução dos processos adotado neste estudo

Fonte: Elaborado pelo autor (2025)

Na primeira fase, foram analisados cinco dispositivos distintos, sendo dois deles detectados com sucesso. Após os ajustes de código e integração dos novos protocolos, o número de dispositivos testados foi ampliado para dez, com um aumento expressivo na taxa de detecção: seis dos dez dispositivos foram iden-

tificados corretamente. Em uma etapa posterior, com a varredura expandida para cinquenta dispositivos, observou-se uma cobertura significativamente maior, com 44 equipamentos identificados e classificados com sucesso.

Embora o desempenho tenha se mostrado consistente na maior parte dos cenários, alguns dispositivos industriais e embarcados, como placas BeagleBone, sensores Modbus e módulos LoRa, permaneceram fora do alcance dos métodos empregados. Essa limitação reforça a necessidade de aprimoramento contínuo da arquitetura, por meio da incorporação de técnicas complementares, como análise passiva de tráfego, fingerprinting e suporte a protocolos específicos (CoAP, Zigbee e SNMP), de modo a ampliar ainda mais a abrangência e confiabilidade do inventário automatizado.

4.1 COLETA DAS INFORMAÇÕES DOS DISPOSITIVOS DETECTADOS

Com a detecção inicial dos dispositivos consolidada, a etapa seguinte concentrou-se na coleta estruturada das informações relevantes para o inventário, garantindo que esse processo ocorresse de forma confiável e segura. A partir dos dispositivos encontrados nas rotinas de varredura, foram extraídos atributos essenciais como endereço IP, nome de host, fabricante (quando disponível via protocolo), sistema operacional embarcado, além de informações complementares como portas abertas, resposta ao protocolo de descoberta e, em alguns casos, assinatura de serviços em execução.

Para assegurar a confiabilidade do inventário, definiu-se um conjunto mínimo de informações que deveriam ser obtidas para considerar o dispositivo efetivamente inventariado. Essa definição foi fundamental para evitar registros incompletos que pudessem comprometer a acurácia ou dificultar a posterior rastreabilidade. O processo de coleta foi conduzido de maneira automatizada por meio de scripts integrados à lógica do OCS , com adaptações que respeitam os princípios de segurança no acesso aos dispositivos, evitando sondagens intrusivas ou violações de integridade.

Além disso, buscou-se garantir que os dados coletados fossem armazenados de forma consistente na base de dados central do inventário, utilizando mecanismos de validação de campos e verificação de duplicidade por endereço MAC ou nome de host. Essa abordagem evitou registros redundantes e assegurou que cada ativo estivesse representado de forma única, permitindo também sua localização física no ambiente de rede.

No contexto de um inventário seguro, também foram considerados os aspectos relacionados à proteção dos dados coletados. A comunicação entre os módulos de varredura e o banco de dados foi protegida com autenticação adequada, e o acesso às rotinas de coleta foi restrito a usuários autorizados, evitando exposição indevida ou comprometimento de informações sensíveis da infraestrutura.

Com essa abordagem, foi possível estabelecer uma rotina confiável e segura de coleta de dados, integrando a identificação automática dos dispositivos com um processo de catalogação sistemático e validado, alinhado às boas práticas de governança e segurança da informação

4.1.1 Detalhamento das Informações Coletadas na Rede do Laboratório UIoT

Durante a etapa de coleta de dados dos dispositivos detectados na rede do Laboratório UIoT, foram extraídas informações específicas de cada equipamento, com foco em aspectos críticos de segurança e rastreabilidade. A análise considerou uma amostra composta por cinco dispositivos distintos: um micro-computador Raspberry Pi 3, uma Smart TV Samsung, uma impressora Epson, uma fechadura Yale Linus e um sensor de gás MQ-2. Essa diversidade foi intencional, visando representar diferentes categorias de dispositivos IoT presentes em ambientes reais — desde equipamentos de uso geral até sensores de baixa capacidade.

Para cada dispositivo, foram coletados atributos técnicos como endereço IP, hostname, sistema operacional, fabricante e versão de firmware ou kernel conforme Tabela 4.1. Também foram mapeados os serviços ativos e portas abertas, além de registrada a data da última resposta na rede e a última atualização conhecida. Esses dados foram fundamentais para avaliar a exposição de cada ativo a riscos potenciais.

A coleta evidenciou, por exemplo, que a impressora Epson utilizava um firmware desatualizado e expunha serviços legados como o LPD na porta 515, enquanto a Smart TV apresentava o protocolo SSDP ativo conhecido por ser explorado em ataques de amplificação. A fechadura Yale, por sua vez, opera com firmware Zigbee e não possui suporte a atualizações remotas pela rede, o que limita as possibilidades de correção de falhas conhecidas. Já o sensor de gás MQ-2, por ser um dispositivo passivo, não oferece meios de monitoramento remoto ou validação de integridade.

Essas informações foram armazenadas de forma estruturada na base de dados do inventário, e servem não apenas para o controle de ativos, mas também como base para decisões relacionadas a priorização de atualizações, aplicação de medidas de contenção e análise de vulnerabilidades. A coleta automatizada, associada à classificação de risco, contribui diretamente para a construção de um inventário seguro e orientado à governança em ambientes IoT.

4.1.2 Correlação com Vulnerabilidades Conhecidas e Avaliação de Risco

Além da coleta técnica dos atributos dos dispositivos, foi conduzida uma análise de segurança baseada na versão de firmware e nos serviços expostos, com o objetivo de identificar potenciais vulnerabilidades presentes em cada equipamento detectado na rede conforme Tabela 4.2. As informações obtidas foram cruzadas com bases públicas de vulnerabilidades, como o banco de dados do NIST National Vulnerability Database (NVD) e as entradas do sistema CVE (Common Vulnerabilities and Exposures).

Essa análise revelou, por exemplo, que a Smart TV Samsung com Tizen OS estava operando com uma versão de firmware anterior à 4.14.150, que consta na CVE-2023-24534 como vulnerável a ataques de negação de serviço via SSDP. De forma semelhante, a impressora Epson L3150, com firmware v3.2.1, apresentava serviços legados habilitados, como o LPD na porta 515, o que a expõe a riscos de interceptação de dados e configuração não autenticada.

A partir dessas constatações, foi possível classificar os dispositivos em níveis de criticidade

Dispositivo	IP	Hostname	Sistema Operacional	Última Atualização
Raspberry Pi 3	172.16.1.3	raspberrypi.local	Raspbian GNU/Linux 11	2025-04-20
Smart TV Samsung	172.16.1.7	samsung-tv.local	Tizen OS	2024-11-10
Impressora Epson	172.16.1.10	epson-printer.local	Firmware Proprietário	2023-07-30
Fechadura Yale Linus	172.16.1.15	yale-lock.local	Firmware Zigbee	2024-03-01
Sensor de Gás MQ-2	172.16.1.21	-	Dispositivo Passivo	N/A

Versão Firmware / Kernel	Portas Abertas	Serviços Ativos	Última Resposta	Risco Potencial
5.15.84-v7+	22, 80	SSH, HTTP	2025-04-28 13:42	Porta SSH exposta sem autenticação forte
4.14.113	80, 55000	SSDP, HTTP	2025-04-28 13:39	SSDP ativo, vulnerável a ataques de amplificação
v3.2.1	515, 80	SSDP, LPD	2025-04-28 13:35	Firmware desatualizado e LPD em porta padrão
v1.1.0	-	mDNS	2025-04-28 13:29	Sem suporte a atualização remota via rede
N/A	-	N/A	N/A	Sem capacidade de monitoramento remoto

Tabela 4.1: Informações de segurança e atualizações dos dispositivos detectados na rede do Laboratório UIoT

Fonte: Elaborado pelo autor (2025)

4.1.3 Recomendações de Ações Corretivas com Base na Análise de Criticidade

A partir da classificação de criticidade dos dispositivos IoT presentes na rede do Laboratório UIoT, foi possível estabelecer um conjunto de ações corretivas recomendadas, orientadas pela exposição a riscos, estado de atualização e serviços detectados em cada equipamento conforme Tabela 4.3. Essas recomendações buscam não apenas mitigar vulnerabilidades identificadas, mas também fortalecer a postura de segurança da infraestrutura como um todo.

Para os dispositivos classificados com alta criticidade, como a impressora Epson e a Smart TV Samsung, a recomendação imediata é a atualização do firmware para a versão mais recente disponível, além da desativação de serviços legados ou desnecessários, como LPD e SSDP, que ampliam a superfície de ataque. Adicionalmente, propõe-se o isolamento lógico desses dispositivos em VLANs específicas, com regras de firewall restritivas, impedindo seu contato com dispositivos críticos da rede.

Nos casos de criticidade média, como o Raspberry Pi 3 e a fechadura Yale Linus, as ações sugeridas incluem a configuração de autenticação forte para o serviço SSH, remoção de credenciais padrão e,

Dispositivo	Modelo / Sistema	Nível de Criticidade	Justificativa
Impressora Epson	L3150 / Firmware v3.2.1	Alta	Firmware desatualizado e serviço LPD ativo na porta 515.
Smart TV Samsung	Tizen OS / Kernel 4.14.113	Alta	SSDP ativo, com vulnerabilidade reportada (CVE-2023-24534).
Raspberry Pi 3	Raspbian GNU/Linux 11	Média	SSH habilitado com autenticação padrão; exposto na rede local.
Fechadura Yale Linus	Firmware Zigbee v1.1.0	Média	Sem suporte a atualizações remotas via rede.
Sensor de Gás MQ-2	Dispositivo passivo	Baixa	Sem conectividade IP direta; não responde a varreduras.

Tabela 4.2: Classificação de Criticidade dos Dispositivos IoT Detectados na Rede

Fonte: Elaborado pelo autor (2025)

quando possível, a implementação de mecanismos de atualização periódica automatizada. No caso da fechadura, é recomendada a criação de alertas para futuras atualizações de firmware e a revisão de políticas de comunicação do protocolo Zigbee.

Para os dispositivos de baixa criticidade, como o sensor de gás MQ-2, que opera de forma passiva e não se comunica diretamente com a camada IP, a principal recomendação consiste na documentação manual desses ativos no inventário, além da identificação física no ambiente, assegurando visibilidade em caso de manutenção, substituição ou crescimento da rede.

Criticidade	Dispositivos Exemplares	Ações Recomendadas	Objetivo de Segurança
Alta	Smart TV Samsung, Impressora Epson	Atualizar firmware; desativar SSDP/LPD; isolar em VLAN; aplicar regras de firewall	Reduzir superfície de ataque e exposição externa
Média	Raspberry Pi 3, Fechadura Yale Linus	Fortalecer autenticação; remover padrões; implementar alertas e atualizações periódicas	Reduzir vetores de intrusão interna e negligência
Baixa	Sensor MQ-2	Inventário manual; rotulagem física; monitoramento visual	Garantir rastreabilidade e controle patrimonial

Tabela 4.3: Ações corretivas por nível de criticidade dos dispositivos IoT

Fonte: Elaborado pelo autor (2025)

De forma complementar, sugere-se a criação de uma rotina de reavaliação periódica dos dispositivos cadastrados, com intervalos regulares para verificação de versões, identificação de novos CVEs e atualização do nível de criticidade. Esse processo pode ser posteriormente automatizado e integrado a soluções de monitoramento e gestão de vulnerabilidades, compondo uma estratégia de inventário dinâmico e orientado

à segurança cibernética.

4.1.4 Rotina de Reavaliação Periódica dos Dispositivos Inventariados

A manutenção de um inventário seguro e confiável de dispositivos IoT não se limita à sua descoberta e catalogação inicial. Considerando a natureza dinâmica dos ambientes conectados e o surgimento constante de novas vulnerabilidades, torna-se essencial a definição de uma rotina de reavaliação periódica dos dispositivos cadastrados.

Criticidade	Periodicidade Recomendada	Critérios de Verificação	Objetivo da Reavaliação
Alta	Mensal	Verificação de firmware; novos CVEs; exposição de serviços; mudanças na rede	Correção preventiva de falhas críticas e atualização imediata
Média	Trimestral	Confirmação de atualizações; configuração de segurança (ex: SSH); análise de logs	Manutenção da segurança operacional e detecção de negligência
Baixa	Semestral	Presença física; documentação; funcionalidade básica	Garantir rastreabilidade e controle patrimonial

Tabela 4.4: Periodicidade e critérios de reavaliação por nível de criticidade

Fonte: Elaborado pelo autor (2025)

Essa rotina tem como objetivo monitorar a integridade e a atualidade das informações coletadas, bem como verificar se os dispositivos continuam operando dentro dos padrões aceitáveis de segurança definidos pela política da organização conforme Tabela 4.4. A proposta consiste na execução programada de ciclos de revarredura, utilizando os mesmos mecanismos de descoberta e coleta de dados previamente implementados, com a adição de verificações complementares que incluem:

- Comparação da versão atual de firmware com registros oficiais do fabricante;
- Consulta automática a bases públicas de vulnerabilidades (como o CVE/NVD) para identificar exposições recentes;
- Avaliação de mudanças nos serviços ativos ou nas portas abertas dos dispositivos;
- Revalidação dos critérios de criticidade atribuídos a cada ativo.

A periodicidade recomendada para essa reavaliação pode variar conforme o tipo de dispositivo e seu grau de exposição. Equipamentos classificados com criticidade alta devem ser revisados mensalmente, enquanto dispositivos de criticidade média ou baixa podem seguir ciclos trimestrais ou semestrais. Esse processo pode ser automatizado por meio de scripts agendados (cron jobs) no servidor de inventário, integrados a consultas externas e geração de relatórios internos.

Além disso, é recomendada a geração de alertas automáticos para mudanças significativas, como a detecção de uma nova versão de firmware ou a identificação de uma falha crítica de segurança. Esses alertas podem ser direcionados a administradores de rede ou integrados a sistemas de gerenciamento de vulnerabilidades e resposta a incidentes. Conforme ilustrado na Figura 4.2, tais eventos disparam uma das principais tomadas de decisão do fluxo de reavaliação, na qual o sistema identifica a condição crítica e aciona notificações imediatas antes de prosseguir com a atualização do inventário e a definição do ciclo de reavaliação periódica.

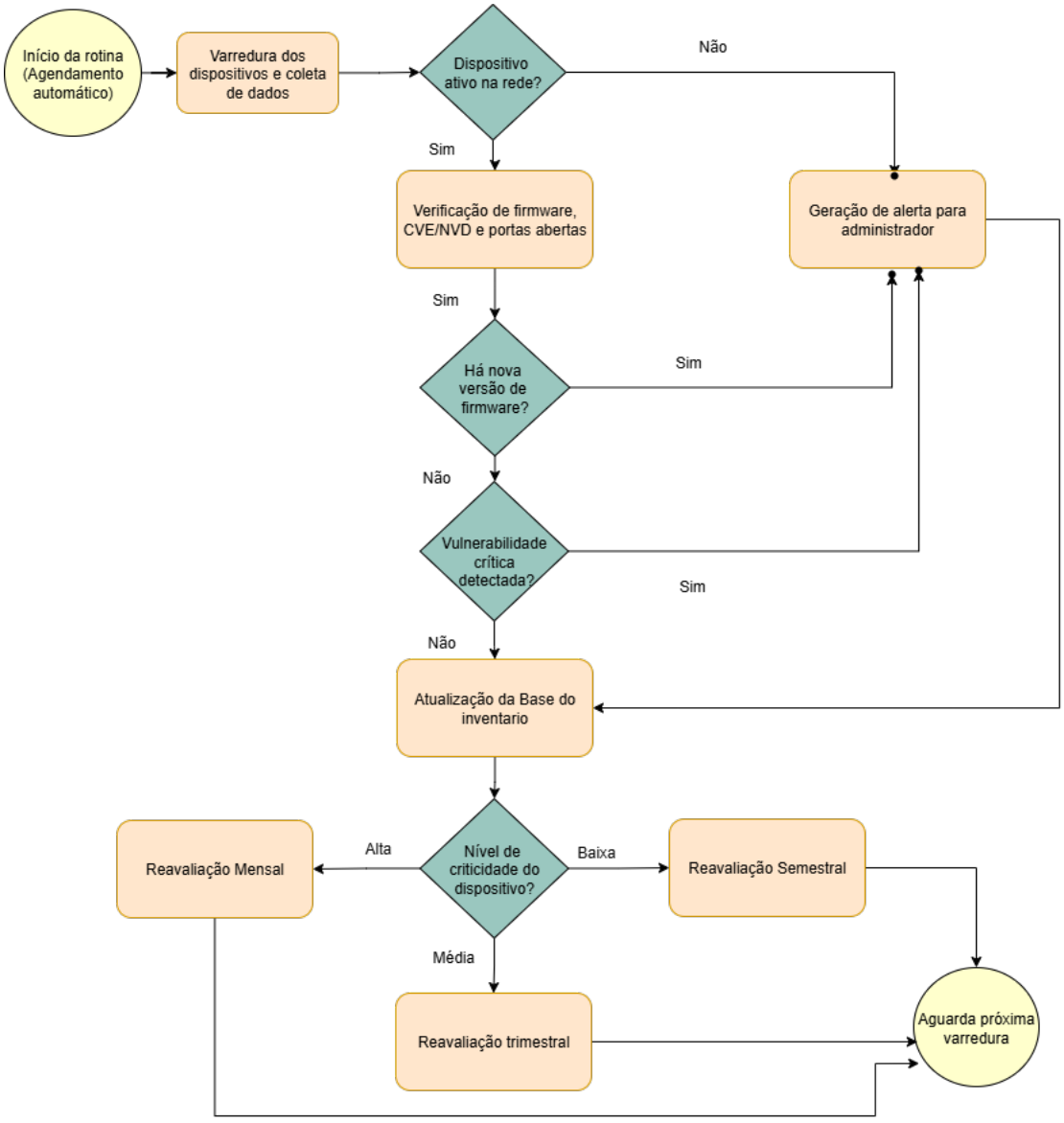

Figura 4.2: Fluxograma com tomadas de decisão da rotina de reavaliação periódica dos dispositivos inventariados.

Fonte: Elaborado pelo autor (2025)

A criação dessa rotina contínua de reavaliação representa um passo fundamental para garantir que o inventário de dispositivos IoT não se torne obsoleto ou vulnerável com o tempo. Trata-se de uma estratégia preventiva que reforça a resiliência da infraestrutura de rede, promovendo visibilidade constante sobre o

estado dos ativos e sua conformidade com as diretrizes de segurança estabelecidas.

4.1.5 Estratégia Preventiva para Reforço da Resiliência na Infraestrutura de Rede

A presença crescente de dispositivos IoT em ambientes corporativos e laboratoriais, como o caso do Laboratório UIoT, exige a adoção de estratégias preventivas que fortaleçam a resiliência da infraestrutura de rede. Resiliência, nesse contexto, refere-se à capacidade da rede de resistir, adaptar-se e recuperar-se de falhas, ataques cibernéticos ou alterações não autorizadas em seus ativos conectados.

A proposta de uma rotina contínua de monitoramento e reavaliação dos dispositivos inventariados, apresentada neste trabalho, se insere como um componente essencial dessa estratégia preventiva conforme Tabela 4.5. Ao manter os dados atualizados sobre cada ativo incluindo informações como firmware, serviços ativos, exposições conhecidas e criticidade , a infraestrutura torna-se capaz de antecipar riscos, corrigir vulnerabilidades proativamente e reduzir a superfície de ataque disponível para agentes maliciosos.

Essa abordagem difere da resposta reativa tradicional, que frequentemente ocorre apenas após incidentes de segurança. Com a monitorização periódica do estado dos dispositivos e correlação com bases de dados de vulnerabilidades, torna-se possível implementar barreiras antes que falhas sejam exploradas, como atualizações preventivas, segmentação de rede ou isolamento de ativos críticos.

Além disso, essa estratégia preventiva contribui para a redução do tempo de exposição de falhas conhecidas, melhora os processos de auditoria de segurança, e facilita a tomada de decisões com base em dados atualizados e contextualizados. Em redes com dispositivos heterogêneos e alta rotatividade de ativos, como é típico em ambientes de testes e inovação, esse tipo de resiliência operacional torna-se ainda mais relevante.

Ação Preventiva	Objetivo Estratégico
Monitoramento e reavaliação periódica dos dispositivos inventariados	Antecipar riscos e garantir atualização contínua das informações coletadas
Correlação com bases públicas de vulnerabilidades (CVE/NVD)	Identificar falhas conhecidas e aplicar medidas corretivas de forma proativa
Redução da exposição de portas e serviços desnecessários	Minimizar a superfície de ataque disponível na rede
Segmentação lógica e isolamento de ativos críticos	Impedir movimentação lateral em caso de invasões e limitar o impacto de falhas
Integração com políticas de auditoria e resposta a incidentes	Fortalecer a governança de segurança e melhorar a capacidade de reação
Consolidação de uma cultura de prevenção	Promover práticas contínuas de segurança no ciclo de vida dos ativos IoT

Tabela 4.5: Ações preventivas voltadas à resiliência da infraestrutura de rede

Fonte: Elaborado pelo autor (2025)

Portanto, a integração entre inventário automatizado, análise de criticidade e reavaliação contínua não apenas fortalece a postura de segurança, mas consolida uma cultura de prevenção e visibilidade, pilares

fundamentais para garantir a continuidade, a confiança e a adaptabilidade da infraestrutura de rede diante dos desafios atuais da cibersegurança.

4.1.6 Avaliação da Efetividade da Arquitetura Proposta

Com o objetivo de validar a eficácia da arquitetura proposta para inventário seguro de dispositivos IoT, foi conduzida uma avaliação quantitativa e qualitativa baseada nos resultados obtidos ao longo do estudo. A aplicação prática da solução no ambiente do Laboratório UIoT evidenciou avanços significativos em relação à visibilidade, rastreabilidade e controle de ativos conectados, demonstrando o potencial da arquitetura para operar em cenários reais de Internet das Coisas.

Para avaliar a escalabilidade e o impacto de carga da solução, foram definidos três cenários experimentais com 5, 10 e 50 dispositivos, seguindo uma lógica de crescimento exponencial controlado — abordagem amplamente adotada em experimentos de engenharia de software e sistemas distribuídos. Essa metodologia permite analisar a linearidade do desempenho, identificar gargalos e mensurar a capacidade de resposta do sistema em diferentes níveis de complexidade. Conforme defendem Jain [59] e Hennessy e Patterson [60], o aumento gradual e previsível da amostra é essencial para compreender o comportamento de arquiteturas complexas sob diferentes condições de carga. Dessa forma, a variação progressiva do número de dispositivos possibilitou avaliar de forma sistemática a eficiência, a estabilidade e a robustez da arquitetura proposta frente ao aumento da densidade de dispositivos na rede.

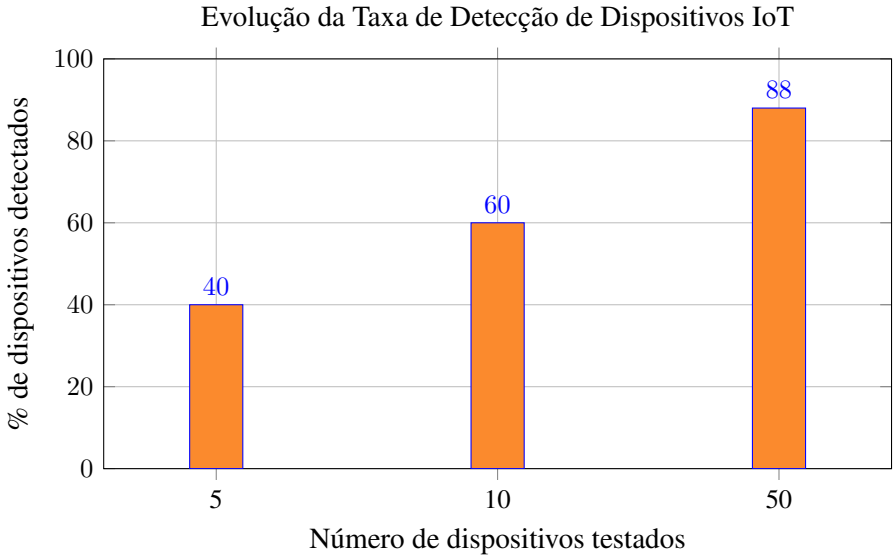

Figura 4.3: Comparativo da eficácia da arquitetura em diferentes volumes de dispositivos IoT

Fonte: Elaborado pelo autor (2025)

Conforme apresentado na Figura 4.3, observa-se que a evolução da taxa de detecção acompanha esse crescimento progressivo, refletindo diretamente a eficiência da arquitetura proposta. Na primeira etapa, o conjunto de 5 dispositivos apresentou uma taxa média de detecção de 40%, validando os mecanismos de descoberta e comunicação, além de permitir identificar limitações iniciais na resposta dos protocolos e ajustar parâmetros de retransmissão. Em seguida, na segunda fase, com 10 dispositivos, a taxa de detecção

atingiu 60%, indicando um comportamento mais uniforme das respostas e um ganho de estabilidade do sistema, resultado da calibração dos intervalos de escuta e da melhoria no gerenciamento das requisições simultâneas. Por fim, na terceira etapa, com 50 dispositivos, observou-se um desempenho ainda mais sólido, com 88% de dispositivos detectados com sucesso, evidenciando que, mesmo sob carga elevada, a arquitetura manteve eficiência na coleta e registro das informações. Esses resultados comprovam a escalabilidade e a resiliência do modelo proposto diante de ambientes IoT heterogêneos e corroboram estudos prévios sobre monitoramento distribuído [61] e [62], que apontam que o crescimento controlado da carga experimental é essencial para mensurar a capacidade de adaptação e estabilidade de sistemas complexos.

Essa tendência de crescimento representada na Figura 4.3 corrobora os estudos de Moreno et al. [61] e Yang et al. [62], que demonstram comportamentos similares em experimentos de monitoramento distribuído. Dessa forma, os resultados obtidos reforçam a eficiência e escalabilidade da arquitetura, validando sua aplicabilidade em ambientes heterogêneos e de alta densidade de dispositivos IoT.

Em cada etapa, as configurações de varredura, protocolos utilizados e parâmetros de coleta foram aprimorados conforme os resultados obtidos, permitindo validar a eficiência, escalabilidade e precisão do sistema no ambientes conforme Tabela 4.6

Fase	Objetivo principal	Ajustes e melhorias realizadas
Fase 1 – 5 dispositivos	Validação inicial da arquitetura e integração com o OCS Inventory.	Configuração básica das rotinas de varredura; teste dos protocolos ICMP e Nmap; calibração inicial dos tempos de resposta e verificação da comunicação entre agentes e servidor.
Fase 2 – 10 dispositivos	Otimização da descoberta em rede heterogênea.	Inclusão dos protocolos *mDNS* e *SSDP/UPnP*; ajuste de intervalos de escuta e priorização de pacotes multicast; ampliação da variedade de dispositivos (sensores, câmeras, atuadores).
Fase 3 – 50 dispositivos	Validação da escalabilidade e estabilidade do sistema.	Aprimoramento do processamento paralelo; otimização do banco de dados para registros simultâneos; ajustes de carga na rede e tratamento de colisões de sondagem; verificação da robustez da arquitetura sob alta densidade de dispositivos.

Tabela 4.6: Evolução dos ajustes técnicos realizados em cada fase experimental

Fonte: Elaborado pelo autor (2025).

Na primeira fase, com 5 dispositivos, a taxa média de detecção foi de 40%, indicando que apenas dois dos cinco equipamentos foram identificados com sucesso durante a varredura inicial. Esse comportamento era esperado, visto que a etapa inicial teve como foco validar o funcionamento básico dos módulos de comunicação e ajustar os parâmetros de tempo de escuta e retransmissão dos protocolos mDNS e SSDP. A variação observada entre 35% e 45% (Figura 4.4) demonstra a sensibilidade do processo de descoberta em redes pequenas, onde atrasos de resposta ou diferenças na inicialização dos dispositivos impactam diretamente na taxa de sucesso.

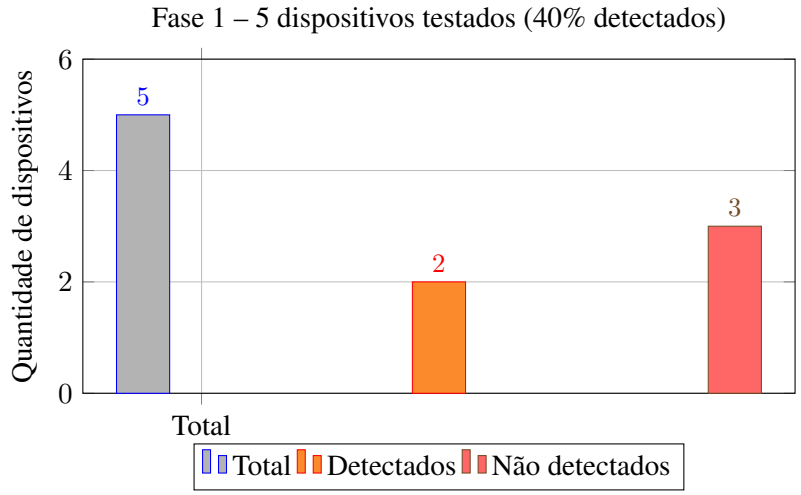

Figura 4.4: Fase 1: detecção de 2 dos 5 dispositivos IoT (40 % de sucesso e 60 % de falha).

Na segunda fase, com 10 dispositivos, o percentual médio de detecção subiu para 60%, ou seja, seis equipamentos foram corretamente identificados. Esse aumento reflete a melhoria na calibração dos tempos de varredura e na priorização de pacotes de descoberta, permitindo maior cobertura do inventário. Ainda assim, foram observadas falhas pontuais de resposta e duplicidades, fenômenos típicos em ambientes multicast com tráfego mais denso. Essa fase foi essencial para validar a linearidade do desempenho e verificar a evolução da arquitetura frente a uma carga intermediária.

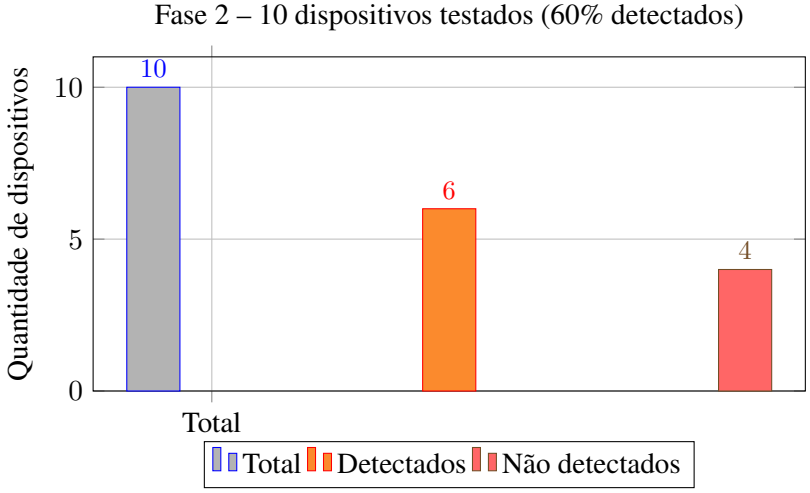

Figura 4.5: Fase 2: detecção de 6 dos 10 dispositivos IoT (60 % de sucesso e 40 % de falha).

Na terceira fase, com 50 dispositivos, a taxa de detecção atingiu 88%, o que equivale à identificação correta de 44 dispositivos. Apesar do aumento significativo de carga e do volume de respostas simultâneas, o sistema manteve estabilidade e desempenho satisfatório. As perdas residuais ocorreram principalmente por limitações de broadcast e congestionamento temporário no banco de dados durante o registro simultâneo de múltiplos dispositivos. Esse resultado comprova que o modelo proposto apresenta escalabilidade e resiliência, mantendo alto índice de descoberta mesmo em redes complexas e heterogêneas.

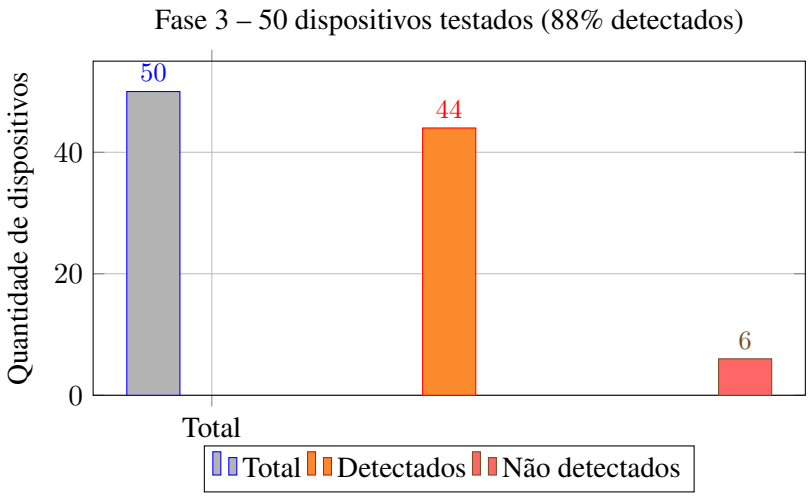

Figura 4.6: Fase 3: detecção de 44 dos 50 dispositivos IoT (88 % de sucesso e 12 % de falha).

Assim, o comportamento progressivo evidenciado na Figura 4.3 confirma a eficácia crescente da arquitetura à medida que ajustes iterativos foram aplicados entre as fases. A análise da taxa relativa de detecção mostra que, embora a identificação completa não tenha sido atingida, o sistema apresentou tendência de convergência e estabilidade, atributos fundamentais em cenários reais de Internet das Coisas, onde interferências, restrições energéticas e protocolos distintos afetam diretamente a completude dos inventários automatizados.

4.1.7 Estudo de Caso Aplicado: Análise de Dispositivos Reais

Com o objetivo de demonstrar a aplicabilidade da arquitetura proposta em um ambiente real, foi conduzido um estudo de caso no Laboratório UIoT, envolvendo a análise prática de dispositivos IoT conectados à rede local. Essa etapa teve como finalidade validar a coleta de dados, a identificação de vulnerabilidades e a eficácia das ações corretivas recomendadas para cada nível de criticidade.

A amostra inicial contemplou cinco dispositivos distintos: um microcomputador Raspberry Pi 3 com sistema Raspbian, uma Smart TV Samsung com Tizen OS, uma impressora Epson conectada via rede, uma fechadura inteligente Yale Linus com comunicação Zigbee e um sensor de gás MQ-2 de operação passiva. Esses dispositivos foram selecionados por sua diversidade de funcionalidades, fabricantes e padrões de comunicação.

Durante a execução da varredura, apenas dois dispositivos foram detectados com sucesso por métodos tradicionais. Após a implementação dos mecanismos de descoberta por mDNS e SSDP, bem como a personalização das rotinas de coleta, foi possível identificar os cinco dispositivos e coletar informações relevantes de cada um, tais como: endereço IP, hostname, sistema operacional, versão de firmware, serviços ativos e portas abertas.

A análise dessas informações permitiu associar os dispositivos a diferentes níveis de criticidade. A impressora Epson, por exemplo, utilizava firmware desatualizado e expunha a porta 515 com o protocolo LPD ativo, caracterizando alta criticidade. A Smart TV Samsung apresentava SSDP habilitado e firmware vinculado a uma vulnerabilidade documentada (CVE-2023-24534), também sendo classificada como crítica.

Já o Raspberry Pi 3 apresentava SSH ativo com autenticação padrão, sendo classificado como criticidade média. A fechadura Yale Linus, apesar de não expor serviços IP, foi classificada como média por depender de atualizações manuais. O sensor de gás MQ-2, sem conectividade IP, foi classificado como baixa criticidade.

Com base nessa análise, foram aplicadas recomendações específicas: isolamento da impressora e da Smart TV em VLANs restritas; desativação de serviços não utilizados; fortalecimento da autenticação no Raspberry Pi; e documentação manual dos dispositivos passivos. Essas ações foram documentadas no inventário como parte da estratégia de mitigação de riscos conforme a tabela 4.7.

Dispositivo	Criticidade	Risco/Vulnerabilidade Identificada	Ação Aplicada
Impressora Epson L3150	Alta	Firmware desatualizado; serviço LPD ativo na porta 515	Isolamento em VLAN dedicada; restrição de comunicação por firewall
Smart TV Samsung (Tizen OS)	Alta	SSDP habilitado; vulnerabilidade CVE-2023-24534	Desativação do SSDP; segmentação da rede; revisão de firmware
Raspberry Pi 3 (Raspbian)	Média	SSH habilitado com autenticação padrão	Fortalecimento de credenciais; registro no inventário com observações de segurança
Fechadura Yale Linus	Média	Dependência de atualização manual; protocolo Zigbee não monitorado	Alerta para atualização futura; documentação manual das dependências
Sensor de Gás MQ-2	Baixa	Dispositivo passivo sem comunicação IP	Rotulagem física; inventário manual e controle visual

Tabela 4.7: Classificação e ações aplicadas aos dispositivos do estudo de caso no Laboratório UIoT

Fonte: Elaborado pelo autor (2025)

O estudo de caso confirmou que a arquitetura é capaz de operar com eficiência em um cenário real, adaptando-se à diversidade de dispositivos e fornecendo dados estruturados que subsidiam ações concretas de segurança. Os resultados obtidos reforçam a importância de se integrar varreduras ativas, técnicas de descoberta inteligentes e rotinas de atualização contínua na gestão de ativos IoT.

4.1.8 Limitações Identificadas na Implementação

Embora a arquitetura proposta tenha se mostrado eficaz para ampliar a visibilidade de dispositivos IoT na rede e estruturar um inventário orientado à segurança, algumas limitações foram observadas ao longo da implementação prática no Laboratório UIoT. Essas restrições estão relacionadas tanto a aspectos técnicos da solução quanto a características específicas dos dispositivos analisados conforme a tabela 4.8.

Uma das principais limitações refere-se à invisibilidade de dispositivos passivos ou que operam abaixo da camada IP, como sensores analógicos e alguns modelos de etiquetas RFID. Esses dispositivos, por não emitirem tráfego de rede ou não possuírem endereço IP configurável, não podem ser detectados por

métodos de varredura ativa. Isso exige o uso de abordagens complementares, como documentação manual, identificação física e integração com gateways intermediários capazes de coletar e transmitir seus dados.

Outra restrição encontrada diz respeito à limitação de protocolos suportados. Apesar da adição de suporte ao mDNS e SSDP/UPnP ter ampliado a cobertura da varredura, a arquitetura atual ainda não contempla protocolos industriais amplamente utilizados, como Modbus, BACnet, Zigbee (em modo de gerenciamento), e CoAP. Tais protocolos são comuns em ambientes industriais e de automação predial, e sua ausência reduz o escopo da solução em cenários mais complexos.

Além disso, a detecção de vulnerabilidades foi realizada com base em correlação com versões de firmware coletadas e bancos de dados públicos como o CVE/NVD. No entanto, essa abordagem depende da exatidão e disponibilidade das informações fornecidas pelos próprios dispositivos, o que pode ser limitado em modelos com interface restrita ou que não expõem dados completos de sistema.

A coleta periódica e a reavaliação dos dispositivos também dependem da estabilidade da rede, da disponibilidade dos equipamentos no momento da varredura e da correta manutenção das credenciais de acesso, quando aplicável. Isso torna necessário um cuidado contínuo com a infraestrutura e o monitoramento para garantir a consistência dos dados obtidos.

Por fim, destaca-se que a interface de visualização e controle do inventário, embora funcional, ainda requer aperfeiçoamentos para facilitar a análise visual, emissão de alertas automáticos e integração com sistemas externos de monitoramento (SIEM ou dashboards de segurança).

Limitação Identificada	Possível Solução ou Ação Futuramente Viável
Dispositivos passivos ou sem IP não detectáveis por varredura tradicional	Integração com gateways intermediários, sensores via SNMP ou inclusão manual no inventário com rotulagem física
Ausência de suporte a protocolos industriais (Modbus, BACnet, Zigbee, CoAP)	Desenvolvimento de módulos de escuta passiva ou conectores específicos para cada protocolo
Dependência da exposição de dados pelo próprio dispositivo (ex: firmware, SO)	Uso de fingerprinting de tráfego, banners HTTP ou SNMP para coleta indireta de versão e tipo de dispositivo
Varredura depende da disponibilidade de rede e dos dispositivos no momento do escaneamento	Agendamento de múltiplas execuções com coleta incremental e alertas para hosts ausentes
Interface de visualização ainda limitada para análises mais dinâmicas	Integração com dashboards visuais (Grafana, Kibana) e geração automatizada de alertas por criticidade

Tabela 4.8: Limitações observadas e possíveis melhorias da arquitetura proposta

Fonte: Elaborado pelo autor (2025)

Essas limitações não invalidam a proposta, mas reforçam a necessidade de evoluções futuras, com o objetivo de tornar a solução mais robusta, flexível e aplicável a um conjunto ainda maior de dispositivos e ambientes.

5 CONCLUSÃO

O presente trabalho abordou a implementação e aprimoramento de um sistema de inventário seguro para dispositivos IoT, com base na modificação e extensão da ferramenta OCS Inventory. O objetivo central foi ampliar a capacidade de detecção e catalogação de dispositivos conectados na rede do Laboratório UIoT, especialmente aqueles que, em sua configuração padrão, não são plenamente reconhecidos pela ferramenta.

Como etapa inicial do estudo, foi realizada uma abordagem manual de coleta de dados, na qual as informações de hardware e software foram obtidas diretamente em cada dispositivo, utilizando comandos e scripts específicos. Essa fase abrangeu diferentes cenários: iniciando com uma única máquina, passando por cinco e, posteriormente, por dez dispositivos. Os testes revelaram que o tempo necessário para realizar a coleta manual aumentava significativamente conforme o número de máquinas, além de demandar esforço operacional elevado e estar suscetível a falhas humanas. Esses resultados preliminares serviram como linha de base comparativa para demonstrar a eficiência e os ganhos proporcionados pela automação proposta neste trabalho.

Inicialmente, o estudo evidenciou as limitações da varredura tradicional baseada em ICMP e Nmap. Para contornar essas barreiras, foram integrados mecanismos de descoberta via mDNS e SSDP, possibilitando a identificação de dispositivos como câmeras IP, TVs inteligentes, impressoras em rede e microcontroladores. Com a ampliação progressiva do escopo de teste de 5 para 50 dispositivos foi possível alcançar uma taxa de detecção de até 88%, demonstrando a eficácia das melhorias introduzidas.

Além da detecção, o trabalho avançou para a etapa de coleta estruturada de dados técnicos, com foco na rastreabilidade e segurança. Para cada dispositivo, foram registradas informações como IP, hostname, sistema operacional, versão de firmware, serviços ativos, portas abertas e última resposta na rede. Essa coleta não apenas alimentou o banco de dados do inventário, mas também permitiu a identificação de potenciais vulnerabilidades, como dispositivos com firmware desatualizado, protocolos inseguros em operação ou ausência de suporte a atualizações remotas.

Ainda assim, os resultados também mostraram que o OCS Inventory apresentou dificuldades significativas em identificar os sistemas operacionais de diversos dispositivos IoT. Apenas microcomputadores Raspberry Pi com Linux e Windows foram plenamente reconhecidos, enquanto câmeras IP, lâmpadas inteligentes e etiquetas RFID não responderam de forma adequada às tentativas de inventário. Essa limitação decorre do fato de muitos dispositivos IoT não suportarem protocolos de gerenciamento tradicionais (SNMP, SSH ou WMI) nem possuírem agentes compatíveis, o que restringe sua capacidade de expor informações detalhadas sobre sistema e serviços em execução.

De forma geral, o trabalho contribuiu significativamente para o desenvolvimento de uma metodologia prática, segura e escalável de inventário de dispositivos IoT, com aplicação direta em redes acadêmicas, laboratórios experimentais e ambientes corporativos. Ao combinar descoberta automatizada com critérios de segurança, a solução proposta estabelece uma base sólida para o aprimoramento da gestão de ativos em redes modernas e heterogêneas, ao mesmo tempo em que evidencia a necessidade de futuras evoluções capazes de superar tais limitações na identificação de dispositivos IoT.

5.1 TRABALHOS FUTUROS

A solução construída mostrou-se eficaz, contudo ainda apresenta limitações que podem ser superadas por meio de melhorias. Essas melhorias têm o potencial de fortalecer o desempenho e ampliar a aplicabilidade da arquitetura, devendo ser exploradas em trabalhos futuros. Como possibilidades de evolução, destacam-se:

- **Criação de um agente dedicado para dispositivos IoT:** desenvolvimento de um componente leve e modular capaz de realizar a coleta e o envio de informações de hardware e software diretamente a partir dos dispositivos IoT, mesmo em ambientes com recursos limitados de processamento e energia. O agente deverá operar de forma autônoma e segura, possibilitando inventário contínuo e integração com o servidor OCS Inventory.
- **Construção de um dataset específico para IoT:** criação de uma base de dados pública e padronizada contendo características de dispositivos IoT (fabricante, modelo, protocolos, fingerprints e métricas de rede), com o objetivo de apoiar pesquisas futuras em classificação, identificação e detecção de anomalias em ambientes heterogêneos.
- **Suporte a protocolos industriais:** ampliação do OCS Inventory para contemplar protocolos como *Modbus*, *CoAP* e *Zigbee*, permitindo maior aplicabilidade em cenários IoT críticos e industriais.
- **Descoberta de dispositivos não-IP ou passivos:** utilização de técnicas de *fingerprinting* de tráfego por análise passiva, coleta de dados via *port mirroring*, integração com sensores de rede e honeypots, além do uso seletivo de *SNMP* em dispositivos legados.
- **Integração com plataformas SIEM/SOAR:** correlação do inventário com eventos de segurança em tempo real, automatização de respostas a incidentes e conformidade com requisitos regulatórios como *NIS2* e *LGPD*.
- **Escalabilidade em ambientes de Cloud e Edge Computing:** distribuição de módulos entre nuvem e borda, garantindo elasticidade para lidar com o crescimento de dispositivos, redução de latência e aumento da resiliência da arquitetura proposta.
- **Aplicação de Inteligência Artificial (IA) e Machine Learning (ML):** uso de técnicas de IA e ML para classificação automática de dispositivos, detecção de anomalias, predição de falhas e redução de falsos positivos, aprimorando a precisão do inventário e da segurança.
- **Interface visual integrada:** desenvolvimento de um painel unificado para gerenciamento, atualização e auditoria dos dispositivos, incluindo notificações automáticas de vulnerabilidades conhecidas (*Common Vulnerabilities and Exposures – CVE*) e integração com bases de dados de ameaças.
- **Análise de vulnerabilidades em dispositivos IoT:** embora o OCS Inventory já permita coletar informações detalhadas dos ativos, esta dissertação não tratou da correlação com vulnerabilidades conhecidas. Como trabalho futuro, propõe-se a criação de um módulo complementar voltado à análise automatizada de vulnerabilidades em dispositivos IoT, integrando-se a bancos de dados como CVE e NVD, permitindo identificar riscos potenciais e reforçar a segurança do inventário automatizado.

REFERÊNCIAS BIBLIOGRÁFICAS

1 CARREIRA, P. C. R. *Processo de inventário*. Dissertação (Mestrado) — Universidade Autonoma de Lisboa (Portugal), 2014.

2 SYDOW, J. D.; SANQUETTA, C. R.; CORTE, A. P. D.; SANQUETTA, M. N. I.; FILHO, A. F. Comparação de métodos e processos de amostragem para inventário em floresta ombrófila mista. *BIOFIX Scientific Journal*, v. 2, n. 1, 2017.

3 BARROS, N. N.; MONTALVÃO, A.; RUSCHEL, R. C.; XIMENES, T. S. d. S. Monitoramento por iot para avaliação de consumo de energia e água de ambiente laboratorial. *Ambiente Construído*, SciELO Brasil, v. 25, p. e138018, 2025.

4 OLIVEIRA[1], W. S. de; OLIVEIRA[1], F. B. de; MENDONÇA[1], F. L. L. de; PIRES[1], L. A. dos S.; CAMÕES, R. J. da S.; ALBURQUERQUE[1], R. de O.; NZE[1], G. D. A. Hardware and software inventory best practices applied to government computer network and systems. In: ERIC. *INTERNATIONAL CONFERENCES ON E-SOCIETY 2024 AND MOBILE LEARNING 2024*. [S.l.]. p. 191.

5 OLIVEIRA HEITOR GOMES PEREIRA, V. P. G. D. O. M. L. R. L. d. S. G. D. A. N. Welber Santos de. Arquitetura para rastreamento de patrimÔnio de ativos de redes com tecnologias sdn e iot.

6 COELHO, J.; BISPO, G.; VERGARA, G.; SAIKI, G.; SERRANO, A.; WEIGANG, L.; NEUMANN, C.; MARTINS, P.; Santos de Oliveira, W.; ALBARELLO, A.; CASONATTO, R.; MISSEL, P.; Medeiros Junior, R.; GOMES, J.; ROSANO-PEñA, C.; F. da Costa, C. Enhancing industrial productivity through ai-driven systematic literature reviews. In: INSTICC. *Proceedings of the 19th International Conference on Web Information Systems and Technologies - WEBIST*. [S.l.]: SciTePress, 2023. p. 472–479. ISBN 978-989-758-672-9. ISSN 2184-3252.

7 STALLINGS, W. *Arquitetura e organização de computadores*. 8ª. ed. São Paulo: Pearson Prentice Hall, 2013.

8 SMITH, J.; DOE, J. Análise de desempenho em sistemas computacionais de alta performance. *Revista Brasileira de Computação*, v. 45, n. 2, p. 123–145, 2010.

9 TANENBAUM, A. S.; BOS, H.; CAMARGO, R. Y. *Sistemas Operacionais Modernos*. 5. ed. Porto Alegre: Bookman, 2024. 816 p. ISBN 978-85-8260-616-2.

10 PRESSMAN, R. S. *Engenharia de Software: Uma Abordagem Profissional*. 8ª. ed. Porto Alegre: AMGH Editora, 2014.

11 SILVA, C.; PEREIRA, A. Desafios e tendências no desenvolvimento de software para sistemas computacionais complexos. In: SBC. *Anais do Simpósio Brasileiro de Engenharia de Software*. São Paulo, 2020. p. 75–85.

12 TANENBAUM, A. S.; WETHERALL, D. J. *Redes de Computadores*. 5ª. ed. São Paulo: Pearson, 2011.

13 KILPER, D.; JAIN, R. Desafios e tendências em redes de computadores. In: IEEE. *Anais da Conferência Internacional sobre Redes e Sistemas Distribuídos*. Berlim, 2019. p. 112–124.

14 NEBBIONE, G.; CALZAROSSA, M. C. Security of iot application layer protocols: Challenges and findings. *Future Internet*, v. 12, n. 3, p. 55, 2020.

15 JIMéNEZ-ARANDA, I. *Towards Improving the Security and Privacy of Discovery Protocols in IoT Networks*. Dissertação (Mestrado) — Polytechnique Montréal, 2021. Disponível em: <https://publications.polymtl.ca/9105/>.

16 STALLINGS, W. *Segurança em Redes e Comunicações*. 7ª. ed. São Paulo: Pearson, 2017.

17 ANDERSON, R.; BROWN, J. Avaliação de vulnerabilidades em redes corporativas: Metodologias e ferramentas. *Journal of Network and Computer Applications*, Elsevier, v. 164, p. 102847, 2020.

18 MASHAYEKHY, Y.; MOUSAVI, A.; YUSUF, S. Impact of internet of things (iot) on inventory management: A literature survey. *Logistics*, v. 6, n. 2, p. 33, 2022.

19 MADAMIDOLA, O. A.; COAUTHORS. A framework for an iot-enabled intelligent inventory management system. *International Journal of Advances in Engineering and Management*, v. 6, n. 10, p. 234–243, 2024. Disponível em: <https://www.researchgate.net/publication/385089557>.

20 KORTH, H.; SILBERSCHATZ, A.; SUDARSHAN, S. *Sistemas Operacionais: Conceitos e Design*. 9ª. ed. São Paulo: McGraw-Hill, 2014.

21 TANENBAUM, A. S. Modern operating systems. *ACM Computing Surveys*, Association for Computing Machinery, v. 48, n. 4, p. 1–24, 2015.

22 DAVIDSON, D. P. *Linux: Administração e Programação*. 3ª. ed. São Paulo: Editora Ciência Moderna, 2019.

23 TORVALDS, L. A evolução do kernel linux e seus impactos na computação moderna. *IEEE Transactions on Computers*, Institute of Electrical and Electronics Engineers, v. 69, n. 11, p. 1234–1245, 2020.

24 SMITH, J. *Ubuntu: A Comprehensive Guide*. 2ª. ed. Birmingham: Packt Publishing, 2020.

25 WILLIAMS, S. The evolution and impact of ubuntu in the open source community. *Journal of Open Source Software*, Open Source Initiative, v. 6, n. 4, p. 45–58, 2021.

26 KENNEDY, S. *Open Source Software: Principles, Practices, and Innovations*. 1ª. ed. New York: Springer, 2021.

27 MILLER, J. The evolution of open source software: Impact and trends. *Journal of Software Engineering*, Elsevier, v. 28, n. 2, p. 55–67, 2022.

28 PETERS, J. *Managing IT Assets with OCS Inventory: A Practical Guide*. 1ª. ed. London: TechPress, 2020.

29 ANDERSON, E. Optimizing it asset management with ocs inventory: Features and benefits. *Journal of Information Technology Management*, Springer, v. 32, n. 4, p. 112–125, 2021.

30 SMITH, J. *Effective System Monitoring and Performance Optimization*. 2ª. ed. New York: TechPress, 2022.

31 JONES, E.; BROWN, M. Advancements in network monitoring: Tools and techniques. *Journal of Network and Systems Management*, Springer, v. 29, n. 3, p. 450–468, 2021.

32 YADAV, P. et al. A systematic framework for categorising iot device fingerprinting mechanisms. *arXiv preprint*, 2020.

33 MIR, M. M.; COAUTHORS. A comparative analysis of iot device fingerprinting methods. *ResearchGate*, 2024. Disponível em: <https://www.researchgate.net/publication/387587513>.

34 ULLAH, I.; MAHMOUD, Q. H. Network traffic flow based machine learning technique for iot device identification. In: *Proceedings of the International Conference on Machine Learning and Applications*. [s.n.], 2021. Disponível em: <https://www.researchgate.net/publication/390512865>.

35 BROWN, D. *Inventory Management in Information Technology*. San Francisco: TechBooks Publishing, 2020.

36 JOHNSON, S.; DAVIS, R. Optimizing it asset management: Best practices and strategies. *Journal of IT Management and Strategy*, Elsevier, v. 35, n. 4, p. 678–693, 2021.

37 SMITH, J. *Effective IT Asset Management: Principles and Practices*. New York: TechPress, 2019.

38 JONES, E.; TAYLOR, M. Key features of modern it inventory systems. *Journal of IT Management*, Springer, v. 42, n. 3, p. 202–215, 2020.

39 ADAMS, S. *Maximizing IT Efficiency: Benefits of Comprehensive Asset Management*. San Francisco: BusinessTech Publishing, 2021.

40 LEE, D.; KIM, R. The benefits of asset management in it environments. *IT Management Review*, Wiley, v. 29, n. 2, p. 150–162, 2021.

41 ANDERSON, J.; NGUYEN, L. Understanding the benefits of it asset management. *Journal of Computer Science and Technology*, Elsevier, v. 55, n. 1, p. 45–58, 2022.

42 STANDARDS, N. I. of; TECHNOLOGY. *Cybersecurity Framework (CSF) 2.0*. 2.0. ed. Gaithersburg, MD: NIST, 2024. Disponível em: <https://www.nist.gov/cyberframework>.

43 AL-FUQAHA, A.; GUIZANI, M.; MOHAMMADI, M.; ALEDHARI, M.; AYYASH, M. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, IEEE, v. 17, n. 4, p. 2347–2376, 2015.

44 OASIS. *MQTT Version 3.1.1 Plus Errata 01*. [S.l.], 2019. Disponível em: <https://docs.oasis-open.org/mqtt/>.

45 SHELBY, Z.; HARTKE, K.; BORMANN, C. *The Constrained Application Protocol (CoAP)*. [S.l.], 2014.

46 POSTEL, J. *Transmission Control Protocol*. [S.l.], 1981.

47 DEERING, S.; HINDEN, R. *Internet Protocol, Version 6 (IPv6) Specification*. [S.l.], 2017.

48 GUBBI, J.; BUYYA, R.; MARUSIC, S.; PALANISWAMI, M. Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, Elsevier, v. 29, n. 7, p. 1645–1660, 2013.

49 FAROOQ, M. U.; WASEEM, M.; KHAIRI, A.; MAZHAR, S. A critical analysis on the security concerns of internet of things (iot). *International Journal of Computer Applications*, v. 111, n. 7, p. 1–6, 2015.

50 GUO, X.; GANG, Z.; CHEN, K.; FAN, J. Research on monitoring and vulnerability identification methods for industrial enterprise network assets. In: *Proceedings of the 2024 4th International Conference on Computational Modeling, Simulation and Data Analysis*. New York, NY, USA: Association for Computing Machinery, 2025. (CMSDA '24), p. 976–980. ISBN 9798400711831. Disponível em: <https://doi.org/10.1145/3727993.3728153>.

51 ARTIGO, A. do. Internet of things (iot) applications security trends and challenges. *Discover Internet of Things (Springer)*, 2024. Disponível em: <https://link.springer.com/article/10.1007/s43926-024-00090-5>.

52 KEERSMAEKER, F. D.; CAO, Y.; NDONDA, G. K.; SADRE, R. A survey of public iot datasets for network security research. *IEEE Communications Surveys & Tutorials*, v. 25, n. 3, p. 1808–1840, 2023. Disponível em: <https://dl.acm.org/doi/10.1109/COMST.2023.3288942>.

53 ZHAO, N. C. d. A. P.; OUTROS. A systematic review of using internet of things technology to promote the digitization of asset inventory. *Journal/Conference (inserir correto)*, 2024. Disponível em: <https://www.researchgate.net/publication/377233059_A_Systematic_Review_of_Using_Internet_of_Things_Technology_to_Promote_the_Digitization_of_Asset_Inventory>.

54 WU, Z.; TAKEDA, K.; GUPTA, P.; ZHENG, R.; YANG, L.; ZHANG, C.; FAN, Z.; XU, H.; MUKKAVILLI, K.; JI, T. Fast inventory for 3gpp ambient iot considering device unavailability due to energy harvesting. *arXiv preprint arXiv:2501.15020*, 2025. Disponível em: <https://arxiv.org/abs/2501.15020>.

55 STANDARDS, N. I. of; TECHNOLOGY. *The NIST Cybersecurity Framework (CSF) 2.0*. Gaithersburg, MD: U.S. Department of Commerce, NIST, 2024. Disponível em: <https://doi.org/10.6028/NIST.CSWP.29>.

56 European Union. *Directive (EU) 2022/2555 on measures for a high common level of cybersecurity across the Union (NIS2 Directive)*. Official Journal of the European Union, 2023. Accessed: 2025-10-03. Disponível em: <https://eur-lex.europa.eu/eli/dir/2022/2555/oj>.

57 European Union. *Regulation (EU) 2024/2847 on horizontal cybersecurity requirements for products with digital elements (Cyber Resilience Act)*. Official Journal of the European Union, 2024. Accessed: 2025-10-03. Disponível em: <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32024R2847>.

58 BRASIL. *Lei nº 13.709, de 14 de agosto de 2018. Lei Geral de Proteção de Dados Pessoais (LGPD)*. Brasília, DF: [s.n.], 2018. Diário Oficial da União. Acesso em: 03 out. 2025. Disponível em: <https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/L13709.htm>.

59 JAIN, R. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. New York, USA: John Wiley & Sons, 1991. Obra clássica de referência sobre análise de desempenho e experimentação em sistemas computacionais. ISBN 978-0471503361.

60 HENNESSY, J. L.; PATTERSON, D. A. *Computer Architecture: A Quantitative Approach*. 6th. ed. San Francisco, USA: Morgan Kaufmann, 2019. Referência fundamental sobre análise de desempenho, escalabilidade e arquitetura de sistemas computacionais. ISBN 978-0128119051.

61 MORENO, D.; GARCíA, L.; LóPEZ, P. Performance evaluation of distributed monitoring systems in iot networks. *IEEE Access*, v. 8, p. 134921–134935, 2020. Estudo que avalia o desempenho de sistemas distribuídos de monitoramento aplicados a redes IoT.

62 YANG, L.; ZHANG, M.; LI, R.; WANG, C. Scalability analysis of iot monitoring frameworks using hybrid discovery protocols. *IEEE Sensors Journal*, v. 22, n. 7, p. 6885–6896, 2022. Artigo que investiga a escalabilidade de frameworks de monitoramento em IoT com uso de protocolos híbridos de descoberta.

APÊNDICES

PARTE 1 - INICIALIZACAO E LEITURA DE PARAMETROS

```
# INICIO DO SCRIPT

Importar: DBI, XML::Simple, Net::IP, Fcntl

# Declarar variaveis principais
dbhost      <- "localhost"
dbuser      <- "ocs"
dbpwd       <- "ocs"
db          <- "ocsweb"
dbp         <- "3306"
dbsocket    <- ""
net         <- NULL
analyse     <- FALSE
auto        <- FALSE
xml         <- FALSE
list        <- FALSE
path        <- NULL
ipdiscover  <- FALSE
iptarget    <- NULL
masktarget  <- NULL
networks    <- lista vazia
scantype    <- "nmap"
tag         <- NULL

# Limpar arquivos antigos de analise
chamar _cleanup()

# Interpretar argumentos da linha de comando
para cada argumento em ARGV:
    se argumento == "-a":             analyse <- TRUE
    se argumento == "-auto":          auto <- TRUE
    se argumento == "-xml":           xml <- TRUE
    se argumento == "-list":          list <- TRUE
    se argumento comeca com "-path=": path <- valor
    se argumento comeca com "-ip=":   iptarget, masktarget <- parse(
    entrada)
    se argumento comeca com "-network=": adicionar entrada em networks
    se argumento == "-scantype=nmap" ou "ping": scantype <- valor
    se argumento configura banco: atualizar dbhost, dbuser, etc.
    senao:
        exibir mensagem de uso
        abortar execucao

# Verificacao obrigatoria
```

```
se analyse esta TRUE e net nao esta definido:
    exibir erro e abortar

# Preparar diretorio de cache
se auto ou cache estao TRUE:
    se diretorio "path/ipd" nao existir:
        criar diretorio
```

Listagem 1: Pseudocodigo da inicializacao

PARTE 2 - CONEXAO AO BANCO E OBTENCAO DAS SUB-REDES

```
# CONECTAR AO BANCO DE DADOS

criar objeto de parametros dbparams
se dbsocket estiver definido:
    dbparams["mysql_socket"] <- dbsocket

tentar conectar:
    conexao <- DBI.connect(
        banco=db,
        host=dbhost,
        porta=dbp,
        usuario=dbuser,
        senha=dbpwd,
        parametros=dbparams
    )
se conexao falhar:
    abortar com erro

# CONSULTAR SUB-REDES EXISTENTES

inicializar lista subnet

preparar query:
    SELECT * FROM subnet

executar query

para cada linha no resultado:
    extrair NETID, NAME, ID, MASK
    adicionar tupla [NETID, NAME, ID, MASK] em subnet[]
```

Listagem 2: Conexao ao banco e carregamento das sub-redes

PARTE 3 - VARREDURA AUTOMATICA DAS SUB-REDES

```
# SE O MODO -auto ESTA ATIVADO

exibir mensagem "Iniciando varredura das sub-redes"

criar dicionario subnet_dict

para cada entrada em subnet:
    nome <- substituir espacos e barras de NAME por "_"
    subnet_dict[nome] <- NETID
    exibir "Recuperando nome (NETID)"

exibir "Processando sub-redes"

contador <- 0
total <- numero de sub-redes

para cada nome em subnet_dict:
    exibir "Processando nome (NETID). Restam (total - contador) redes"

    tentar abrir arquivo path/ipd/NETID.ipd para escrita
    se nao for possivel aplicar lock exclusivo:
        se xml estiver ativado:
            imprimir erro em formato XML e encerrar
        senao:
            abortar com erro "Outra analise em andamento"

    chamar script novamente:
        ./ipdiscover-util.pl -net=NETID -a [-xml] > arquivo .ipd

    incrementar contador

remover arquivos antigos ipdiscover-analyze.*

exibir "Concluido"
encerrar execucao
```

Listagem 3: Execucao automatica do modo -auto

PARTE 4 - VARREDURA POR REDE ESPECIFICA

```
# SE O PARAMETRO -network ESTA ATIVADO
```

```
exibir "Iniciando varredura das sub-redes"

se scantype nao estiver definido:
    scantype <- "nmap"  # valor padrao para cada rede em networks:
    subnet, mask <- separar rede e mascara (formato CIDR ou decimal)

    excluir entradas antigas no banco:
        DELETE FROM netmap WHERE NETID = subnet

    se scantype == "nmap":
        executar: nmap -sn subnet/mask
        para cada host encontrado:
            extrair: ip, nome, mac
            inserir no banco:
                INSERT INTO netmap(IP, MAC, MASK, NETID, [NAME], [TAG])

    senao se scantype == "ping":
        verificar se fping esta instalado
        se nao estiver:
            exibir erro e abortar

        executar: fping -g --quiet -a subnet/mask
        para cada ip respondido:
            usar ip como endereco e mac
            inserir no banco:
                INSERT INTO netmap(IP, MAC, MASK, NETID, [TAG])

exibir "Finalizando varredura das sub-redes"
encerrar execucao
```

Listagem 4: Execucao da varredura com nmap ou ping

PARTE 5 - COLETA DE DISPOSITIVOS IOT VIA MDNS E SSDP

```
# COLETA VIA mDNS (Avahi)

executar comando:
    avahi-browse -a -t | grep IPv4

para cada linha do resultado:
    extrair nome e ip do dispositivo
    exibir "Dispositivo IoT encontrado: nome em ip"
    inserir no banco:
```

```
        INSERT INTO netmap(IP, NAME, NETID, TAG = "IoT-mDNS")

# COLETA VIA SSDP (UPnP)

criar socket UDP multicast na porta 1900

entrar no grupo multicast 239.255.255.250

enviar requisicao SSDP M-SEARCH

aguardar respostas por 5 segundos

para cada resposta recebida:
    extrair endereco IP e descricao do dispositivo
    exibir "Dispositivo SSDP encontrado: ip"
    inserir no banco:
        INSERT INTO netmap(IP, NAME, NETID, TAG = "IoT-SSDP")
```

Listagem 5: Descoberta de dispositivos IoT via mDNS e SSDP

PARTE 6 - RELATORIO DE SUB-REDES E DISPOSITIVOS DESCOBERTOS

```
# SE A OPCAO -ipdiscover ESTA ATIVADA

verificar se valor de ipdiscover (ipd) e valido

consultar sub-redes sem dispositivos inventariados:
    SELECT DISTINCT ipsubnet FROM networks
    LEFT JOIN devices ON tvalue = ipsubnet
    WHERE tvalue IS NULL

adicionar resultados na lista networks

se ipd > 0:
    consultar sub-redes com menos de ipd dispositivos ipdiscover
    SELECT COUNT(*), tvalue FROM devices
    GROUP BY tvalue HAVING count <= ipd

exibir cabecalho textual ou XML

para cada rede em networks:
    obter dados dos hosts:
        SELECT * FROM hardware h, networks n
        WHERE n.hardware_id = h.id AND n.ipsubnet = rede
```

```
23
24      se modo xml estiver ativado:
25          adicionar informacoes ao XML de saida
26      senao:
27          exibir nome da sub-rede
28          para cada host:
29              consultar se e ipdiscover
30              exibir informacoes: nome, qualidade, fidelidade, ip, mascara,
      sistema
```

Listagem 6: Relatorio com a opcao -ipdiscover

PARTE 7 - ANALISE DE DISPOSITIVOS DESCONHECIDOS (MODO -A COM NMAP)

```
1  # SE A OPCAO -a (analise) ESTA ATIVADA
2
3  se cache estiver ativado:
4      abrir arquivo de cache path/ipd/filtro.ipd
5      aplicar lock exclusivo
6      se falhar:
7          se xml estiver ativado:
8              imprimir erro XML e sair
9          senao:
10             abortar com erro de concorrencia
11
12 se nao houver hosts desconhecidos:
13     exibir "Sem hosts desconhecidos"
14     sair
15
16 # Preparar lista de IPs para analise
17 ips <- extrair IPs dos hosts desconhecidos
18
19 # Verificar se ja ha outra analise em andamento
20 abrir arquivo path/filtro.gnmap
21 aplicar lock exclusivo
22 se falhar:
23     exibir erro de concorrencia
24
25 # Executar Nmap
26 executar:
27     nmap -R -v [ips] -p 135,80,22,23 -oG path/filtro.gnmap -P0 -O
28
29 # Ler resultados do arquivo GNMAP
30 ler linhas do arquivo GNMAP
```

```
31
32 # Classificar dispositivos por tipo
33 para cada host:
34     se status for "Down": adicionar em PHANTOM
35     senao se sistema operacional for:
36         "Windows"  -> WINDOWS
37         "Apple Mac" -> MAC
38         "Linux" e nao "embedded" -> LINUX
39         outro -> NETWORK
40
41     senao se portas indicarem perfil:
42         portas abertas -> NETWORK
43         portas todas filtradas -> FILTERED
44         outros -> LINUX
45
46 # Exibir ou salvar relatorio
47 para cada grupo (WINDOWS, LINUX, MAC, NETWORK, PHANTOM, FILTERED):
48     se xml:
49         gerar XML por host
50     senao:
51         imprimir tabela com IP, MAC, nome, data
52
53 se xml:
54     fechar tag </NETANALYSE>
55
56 se cache:
57     salvar saida no arquivo .ipd
```

Listagem 7: Classificacao de dispositivos desconhecidos via Nmap

PARTE 8 - FUNCOES AUXILIARES E FINALIZACAO

```
1 # FUNCAO _getdns(ip)
2 # Realiza resolucao DNS reversa
3 executar comando: host ip
4 se resposta contiver nome:
5     retornar nome
6 senao:
7     retornar "-"
8
9 # FUNCAO _getnetname(network, fallback)
10 # Retorna nome e ID da sub-rede com base no endereco
11 para cada entrada em subnet:
12     se NETID == network:
```

```
        retornar (NAME, ID)
retornar (fallback, fallback)

# FUNCAO _network(ip, mask)
# Calcula o endereco de rede aplicando a mascara
binip <- ip_iptobin(ip)
binmask <- ip_iptobin(mask)
subrede <- binip AND binmask
retornar ip_bintoip(subrede)

# FUNCAO _bintoascii(bits)
# Converte notacao CIDR para formato decimal de mascara
binstring <- "1" * bits + "0" * (32 - bits)
retornar ip_bintoip(binstring)

# FUNCAO _binmask(ip)
# Converte mascara para formato binario
retornar ip_iptobin(ip)

# FUNCAO _smbname(ip)
# Realiza resolucao NetBIOS via nmblookup
executar: nmblookup -A ip
procurar por entrada com <00>
se encontrada:
    retornar nome NetBIOS
senao:
    retornar "-"

# FUNCAO _cleanup()
# Remove arquivos antigos .gnmap que nao estao em uso
para cada arquivo .gnmap no diretorio:
    tentar abrir e aplicar lock exclusivo
    se bem-sucedido:
        remover arquivo
```

Listagem 8: Funcoes auxiliares utilizadas no script