01 A 04 DE SETEMBRO | FOZ DO IGUAÇU - PR

MINICURSOS SBSeg 2025

XXV Simpósio Brasileiro de Cibersegurança

ARAUCÁRIA · PARANÁ GOVERNO DO ESTADO SECRETARIA DA CIÊNCIA, TECNOLOGIA E ENSINO SUPERIOR · cgi.br · nic.br · Google · UNICO · CNPq · CISSA · CESAR · Optidata · ELDORADO INSPIRAÇÃO PELO NOVO

MINICURSOS DO XXV SIMPÓSIO BRASILEIRO DE CIBERSEGURANÇA

Organizadores do Livro
Diogo Menezes Ferrazani Mattos (UFF)
Cíntia Borges Margi (USP)
Rodrigo Brandão Mansilha (UNIPAMPA)
Altair Santin (PUCPR)
André Grégio (UFPR)
Eduardo Kugler Viegas (PUCPR)

Porto Alegre
Sociedade Brasileira de Computação – SBC
2025

Realização

Organização

Patrocinadores

Master

Infinity

Platinum

Diamante

Ouro

Prata

Bronze

Apoiadores

Capa: Manoela Resende Gomes da Cunha

Dados Internacionais de Catalogação na Publicação (CIP)

S612 Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (25. : 01 – 04 set. 2025 : Foz do Iguaçu, PR)

Minicursos do SBSeg 2025 [recurso eletrônico] / organização: Diogo Menezes Ferrazani Mattos ... [et al.]. – Dados eletrônicos. – Porto Alegre : Sociedade Brasileira de Computação, 2025.

200 p. : il. : PDF.

Modo de acesso: World Wide Web.
Inclui bibliografia
ISBN 978-85-7669-651-3 (e-book)

1. Computação – Brasil – Evento. 2. Segurança da informação. 3. Sistemas computacionais. 4. Cibersegurança. 5. Computação segura. 6. Computação quântica I. Mattos, Diogo Menezes Ferrazani. II. Margi, Cíntia Borges. III. Mansilha, Rodrigo Brandão. IV. Santin, Altair. V. Gregio, André. VI. Viegas, Eduardo Kugler. VII. Sociedade Brasileira de Computação. VIII. Título.

CDU 004(063)

Ficha catalográfica elaborada por Annie Casali – CRB-10/2339

Biblioteca Digital da SBC – SBC OpenLib

Índices para catálogo sistemático:

1. Ciência e tecnologia dos computadores : Informática – Publicação de conferências,congressos e simpósios etc. 004(063)

Sociedade Brasileira de Computação
Av. Bento Gonçalves, 9500
Setor 4 | Prédio 43.412 | Sala 219 | Bairro Agronomia Caixa Postal 15012 | CEP 91501-970
Porto Alegre - RS
Fone: (51) 99252-6018
sbc@sbc.org.br

Índice

Mensagem da Coordenação Geral

É com grande alegria e satisfação que damos as boas-vindas a todos ao XXV Simpósio Brasileiro em Cibersegurança (SBSeg 2025), que será realizado presencialmente de 1 a 4 de setembro de 2025, em Foz do Iguaçu, Paraná. Esta é uma edição especial do Jubileu de Prata, celebrando um quarto de século de contribuição contínua para o avanço da pesquisa, da inovação e da cooperação na área de cibersegurança no Brasil.

Neste ano, alcançamos novamente recordes de trabalhos submetidos à Trilha Principal, ao Salão de Ferramentas e ao WTICG, evidenciando o crescente interesse e engajamento da comunidade científica e profissional de CiberSegurança do país. Ao todo foram 366 submissões e 145 artigos/trabalhos aceitos, assim distribuídos: 229 registros de submissões, com 84 artigos completos e curtos aceitos nas trilhas principais, 24 registros de submissões com 12 trabalhos aceitos no SF, 15 registros de submissões com 12 trabalhos aceitos no WGID, 47 registros de submissões com 14 aceitos no WTICG, 5 registros de submissões com 3 aceitos no WTE, 7 registros de submissões com 4 aceitos no WQuSec, 23 registros de submissões com 12 aceitos na Trilha da Indústria, 16 minicursos propostos com 4 aceitos. No CTA, foram recebidos 29 trabalhos e atribuídos 52 selos. Temos ainda até 20 posters das trilhas principais que estão em processo de definição para apresentação no evento.

Gostaríamos de expressar nossos sinceros agradecimentos a todos os envolvidos no processo de organização do SBSeg pelo apoio, dedicação e comprometimento com o evento. Primeiramente, aos patrocinadores: a Fundação Araucária, a Secretaria da Ciência, Tecnologia e Ensino Superior do Estado do Paraná, ao Comitê Gestor da Internet no Brasil (CGI.br) e ao Núcleo de Informação e Coordenação do Ponto BR (NIC.br), ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), a Google, a Unico, ao Centro Integrado de Segurança em Sistemas Avançados (CISSA), ao CESAR, a Optidata e ao Eldorado. Agradecemos também aos apoiadores, que incluem a Itaipu Binacional, a Tempest, a Rede Nacional de Ensino e Pesquisa (RNP) e a Globethics, cujo suporte econômico tem sido essencial para a realização desta edição.

Agradecemos aos coordenadores do Comitê de Programa e da Trilha da Indústria, Diego Kreutz (UNIPAMPA) e Michelle Wangham (RNP/Univali). Agradecemos também aos coordenadores de Palestras e Tutoriais, Marjory da Costa Abreu (Sheffield Hallam University) e Marinho Pilla Barcellos (University of Waikato). Nos minicursos, o agradecimento vai para Diogo Menezes Ferrazani Mattos (UFF) e Cintia Borges Margi (USP). No Salão de Ferramentas, contamos com a coordenação de Wilson de Souza Melo Jr. (Inmetro) e Davidson Boccardo (Hospital Israelita Albert Einstein). O Workshop de Trabalhos de Iniciação Científica e de Graduação foi coordenado por Flavia Paiva Agostini (Inmetro) e Vinícius Fülber Garcia (UFPR). O Brazilian Women in CyberSecurity (WISE) teve como coordenadoras Marcia Tosta (CISO Advisor) e Patricia Peck Pinheiro (Peck Advogados). O Workshop de Gestão de Identidades Digitais (WGID) foi coordenado por Frederico Schardong (IFRS) e Emerson Ribeiro de Mello (IFSC). O Workshop de Tecnologia Eleitoral (WTE) contou com a coordenação de Roberto Samarone Araujo (UFPA) e Paulo Matias (UFSCar). O Workshop de Cibersegurança Quântica: Teoria, Tecnologias e Aplicações (WQuSec) teve como coordenadores Jeferson Campos Nobre (UFRGS) e Antônio Jorge Gomes Abelém (UFPA). O Workshop de Ética em CiberSegurança (WECS) foi coordenado por Maria Eugenia Barroso (Globethics), Rudolf Eduard Von Sinner (PUCPR) e Adriana Baravalle (Globethics). Por fim, mas não menos importante, agradecemos o Weverton Cordeiro (UFRGS) pela coordenação do Test of Time Award e Rildo Souza e Michelle Wangham pela coordenação do Hackathon.

Agradecemos aos nossos palestrantes internacionais, Anderson C. A. Nascimento (VISA), Christian Doerr (University of Potsdam), David Mohaisen (University of Central Florida) e Guofei Gu (Texas A&M University) e aos palestrantes nacionais, que despender precioso tempo de suas vidas pessoais e profissionais para tornarem o SBSeg 2025 ainda mais prestigioso.

Agradecemos ao Programa Hackers do Bem que com apoio técnico da Rede Nacional de Ensino e

Pesquisa (RNP) e do Itaipu Parquetec organizaram o 4º Hackathon do SBSeg e 3º Hackathon do Programa Hackers do Bem. Nossos agradecimentos também vão para os demais colaboradores da organização do SBSeg 2025,Thiago Heinrich (UFPR e MPI), Lourenço Alves Pereira Jr e Roben Castagna Lunardi (IFRS) no Comitê Técnico de Artefatos; José Lázaro F. B. Junior e Alexsandro T. Ribeiro (PPGINF UFPR) em Mídias Sociais, Amanda Gobus como ilustradora e Manoela Resende Gomes da Cunha como Designer (UNIPAMPA) e Rodrigo Brandão Mansilha (UNIPAMPA) em Publicações.

Agradecemos todo o apoio da Sociedade Brasileira de Computação (SBC) e da Coordenação e Comitê Gestor da Comissão Especial de CiberSegurança (CESeg) da SBC na pessoa de seu coordenador Marcos Antonio Simplicio Junior (USP) e Vice-coordenador Diego Kreutz (UNIPAMPA) . Um agradecimento especial a Pontifícia Universidade Católica do Paraná e a Universidade Federal do Paraná pelos muitos apoios para realizarmos o SBSeg, desde o envolvimento e trabalho árduo (e urgente) de seus técnicos administrativos até os muitos outros colaboradores que nos ajudaram a fazer o evento acontecer. Enfim, agradecemos a todos os envolvidos que trabalharam incansavelmente para proporcionar uma programação rica e diversificada, abordando temas relevantes no cenário nacional e internacional.

A contribuição da comunidade científica brasileira foi fundamental para manter a qualidade técnica dos trabalhos e fortalecer a ciência, a tecnologia e a inovação no Brasil na área de cibersegurança.

Desejamos a todos um SBSeg 2025 com ricas e inspiradoras experiências para celebrar de maneira memorável seu jubileu de prata em Foz do Iguaçu, PR.

Altair O. Santin (PUCPR), André R. A. Grégio (UFPR) e Eduardo K. Viegas (PUCPR)
Coordenadores gerais do SBSeg 2025

Mensagem da Coordenação dos Minicursos

Temos a satisfação de apresentar a seleção de capítulos deste livro, que reúne os minicursos do XXV Simpósio Brasileiro de Cibersegurança (SBSeg), anteriormente denominado Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais, realizado em Foz do Iguaçu, PR, entre os dias 1º e 4 de setembro de 2025. Foram recebidas 16 propostas de minicursos, das quais 4 foram selecionadas para apresentação durante o evento e publicação neste livro, resultando em uma taxa de aceitação de 25%. Os minicursos do SBSeg têm acompanhado as demandas do público, contemplando tanto os participantes que buscam aprofundamento técnico quanto aqueles interessados em fundamentos conceituais e nos avanços mais recentes da área de cibersegurança.

Os capítulos reunidos neste livro refletem a diversidade de enfoques, abrangendo desde aspectos teóricos fundamentais até aplicações práticas relevantes e em consolidação. Cada capítulo corresponde a um minicurso ministrado presencialmente por especialistas durante o evento, constituindo um registro técnico e didático do conteúdo apresentado. A seguir, são descritos de forma sintética os temas abordados nos capítulos.

Capítulo 1. Para além dos Perímetros da Cibersegurança com a Infraestrutura como Código (IaC)
O capítulo apresenta uma introdução abrangente à aplicação de princípios de Cibersegurança com foco em Infrastructure as Code (IaC) e no modelo de Confiança Zero (Zero Trust). Seu objetivo principal é capacitar os leitores a compreenderem e aplicarem práticas seguras de provisionamento automatizado de infraestrutura. A abordagem combina fundamentos teóricos, como automação, boas práticas de segurança e arquitetura de infraestrutura, com atividades práticas realizadas em ambiente virtualizado, permitindo a experimentação controlada sem necessidade de recursos locais. O capítulo também destaca os riscos associados à exposição inadvertida de componentes sensíveis, como credenciais e imagens de containers.

Capítulo 2. Zero Trust Architecture para Dispositivos de Internet das Coisas e Redes de Próxima Geração: Ferramentas, Tendências e Desafios
O capítulo explora a adoção da Arquitetura de Confiança Zero (Zero Trust Architecture – ZTA) como resposta à crescente vulnerabilidade de dispositivos conectados e à limitação dos modelos de segurança baseada em perímetros em redes contemporâneas. A proposta central está na aplicação de autenticação contínua, microsegmentação e controle dinâmico de acesso como fundamentos para ambientes de rede mais resilientes e adaptativos. A abordagem teórica introduz os conceitos essenciais da ZTA, detalha seus principais componentes e discute a viabilidade da arquitetura em cenários complexos, como redes corporativas e federadas. O capítulo inclui atividades práticas baseadas na plataforma OpenZiti, permitindo a experimentação de túneis seguros e serviços implementados segundo os princípios da ZTA. O capítulo busca desenvolver uma compreensão crítica sobre os desafios técnicos e operacionais da arquitetura, ao mesmo tempo em que evidencia seu potencial para a proteção de infraestruturas digitais em redes de próxima geração.

Capítulo 3. Introdução à Computação Quântica e Impactos em Criptografia
O capítulo examina os impactos da Computação Quântica sobre sistemas criptográficos clássicos, discutindo algoritmos como Shor e Grover e suas implicações para mecanismos como RSA, ECC e AES. O capítulo apresenta soluções emergentes, como a Criptografia Pós-Quântica e a Criptografia Quântica, com ênfase em fundamentos teóricos e aplicações práticas. Também são explorados casos de uso em áreas como saúde, finanças e processamento de linguagem natural, articulando riscos e oportunidades. O conteúdo combina exposição conceitual, experimentação com Qiskit e reflexão sobre desafios éticos e de formação de profissionais na área. O capítulo visa fornecer uma base sólida para a compreensão das transformações provocadas pelo paradigma quântico.

Capítulo 4. WiFi Sensing e CSI aplicados à Cibersegurança: Fundamentos, Aplicações e Prática
O capítulo explora o uso de redes WiFi como sensores contextuais de alta resolução, aproveitando o acesso ao Channel State Information (CSI) e os avanços da emenda 802.11bf. O capítulo combina fundamentos eletromagnéticos, técnicas de autenticação de proximidade e detecção de intrusão, além de estratégias de reforço para sistemas de defesa. São apresentados métodos de captura e análise de CSI em plataformas acessíveis, com demonstrações práticas utilizando ESP32 e Raspberry Pi. O capítulo inclui ainda tendências de pesquisa como fusão sensorial, aprendizado federado e defesa contra ataques adversariais.

Registramos nosso reconhecimento aos autores que submeteram propostas ao SBSeg 2025. A qualidade e a diversidade dos trabalhos recebidos contribuíram de forma decisiva para a consolidação e a relevância contínua do evento. Agradecemos, em particular, aos autores dos minicursos selecionados, que se dedicaram à elaboração de capítulos consistentes e tecnicamente robustos. Estendemos também nossos agradecimentos aos membros do Comitê de Programa, cuja avaliação criteriosa e colaboração voluntária foram fundamentais para a qualidade desta edição. Reiteramos nossa gratidão ao coordenador de publicações, professor Rodrigo Brandão Mansilha (UNIPAMPA), pela editoração e publicação deste livro. Agradecemos ainda aos coordenadores gerais do SBSeg 2025, professores Altair Santin (PUCPR), André Gregio (UFPR) e Eduardo K. Viegas (PUCPR), pelo suporte institucional, pelas orientações fornecidas e pela confiança depositada na condução desta atividade. Espera-se que a leitura deste livro seja proveitosa e estimule novas contribuições para o avanço da cibersegurança.

Diogo Menezes Ferrazani Mattos (UFF) e Cintia Borges Margi (USP)
Coordenadores dos Minicursos do SBSeg 2025

Comitê de Organização de Minicursos

Coordenação Geral
Altair Olivo Santin (PUCPR)
André R. A. Grégio (UFPR)
Eduardo Kugler Viegas (PUCPR)

Coordenação de Minicursos
Diogo Menezes Ferrazani Mattos (UFF)
Cintia Borges Margi (USP)

Comitê de Programa de Minicursos
Altair Olivo Santin (PUCPR)
Cintia Borges Margi (USP)
Dianne Scherly Varela de Medeiros (UFF)
Diego Kreutz (UNIPAMPA)
Diogo Menezes Ferrazani Mattos (UFF)
Eduardo James Pereira Souto (UFAM)
Eduardo Kugler Viegas (PUCPR)
Eduardo Luzeiro Feitosa (UFAM)
Igor Monteiro Moraes (UFF)
Luiz Fernando Rust da Costa Carmo (Inmetro)
Marcia Henke (UFSM)
Marcos Antonio Simplicio Jr (USP)
Natalia Castro Fernandes (UFF)
Raul Ceretta Nunes (UFSM)
Ricardo Dahab (UNICAMP)
Ricardo Felipe Custódio (UFSC)

Coordenação de Publicações
Rodrigo Brandão Mansilha (UNIPAMPA)

Comissão Especial de Cibersegurança

Coordenação

Marcos Simplicio (USP)	Coordenador
Diego Kreutz (UNIPAMPA)	Vice-Coordenador

Comitê Gestor

Aldri Luiz dos Santos (UFMG)
Altair Olivo Santin (PUCPR)
André Grégio (UFPR)
Diego Kreutz (UNIPAMPA)
Diogo Mattos (UFF)
Edelberto Franco (UFJF)
Igor Monteiro Moraes (UFF)
Lourenço Pereira Jr. (ITA)
Marcos Simplicio (USP)
Michelle Wangham (UNIVALI)
Rodrigo Miani (UFU)

Sociedade Brasileira de Computação

Presidência

Thais Vasconcelos Batista (UFRN)	Presidente
Cristiano Maciel (UFMT)	Vice-Presidente

Diretoria

Denis Lima do Rosário (UFPA)	Diretor de Eventos e Comissões Especiais
Michelle Silva Wangham (UNIVALI)	Diretora de Relações Profissionais
Alírio Santos de Sá (UFBA)	Diretor de Comunicação
Eunice Pereira dos Santos Nunes (UFMT)	Diretora de Secretarias Regionais
André Luís de Medeiros Santos (UFPE)	Diretor de Planejamento e Programas Especiais
José Viterbo Filho (UFF)	Diretor de Publicações
Ronaldo Alves Ferreira (UFMS)	Diretor de Cooperação com Sociedades Científicas
Rodrigo Silva Duran (IFB)	Diretor de Educação
Leila Ribeiro (UFRGS)	Diretora de Computação na Educação Básica
Renata de Matos Galante (UFRGS)	Diretora Administrativa
Flávia Maria Santoro (INTELLI)	Diretora de Inovação
Francisco Dantas Medeiros Neto (UERN)	Diretor de Finanças
Carlos Eduardo Ferreira (USP)	Diretor de Competições Científicas

Diretoria Extraordinária

Marcelo Antonio Marotta (UNB)	Diretor de Tecnologia da Informação

Conselho - Mandato 2023-2027

Altigran Soares da Silva (UFAM)
Carla Maria dal Sasso Freitas (UFRGS)
Débora Christina Muchaluat Saade (UFF)
José Carlos Maldonado (USP)
Jussara Marques de Almeida (UFMG)

Conselho - Mandato 2025-2029

Antonio Jorge Gomes Abelém (UFPA)
Fabio Kon (USP)
José Palazzo Moreira (UFRGS)
Mirella Moura Moro (UFMG)
Teresa Bernarda Ludermir (UFPE)

Suplentes - Mandato 2025-2029

Aletéia Patrícia Favacho de Araújo (UNB)
Claudia Lage Rebello da Motta (UFRJ)
Flávio Rech Wagner (UFRGS)
Guilherme Horta Travassos (UFRJ)
Roberto Pereira (UFPR)

Minicursos

Capítulo 1

Para além dos Perímetros da CiberSegurança com a Infraestrutura como Código (IaC)

Fellipe M. Veiga (PUCPR), Altair O. Santin (PUCPR), Juliano S. Langaro (PUCPR), Juarez de Oliveira (PUCPR), Eduardo K. Viegas (PUCPR)

Abstract

Traditional perimeter-based security approaches have proven insufficient in the face of increasingly sophisticated threats and the growing complexity of modern, dynamic, and automated computing environments. In this context, this short course proposes an integrated approach that combines the principles of Zero Trust Architecture with Infrastructure as Code (IaC) practices, aiming to promote more secure, resilient, and versionable infrastructures from the provisioning stage. The course includes a theoretical foundation covering infrastructure automation with IaC, microsegmentation, identity-based access control, and continuous context analysis—key elements for mitigating contemporary threats. In the practical phase, participants will be guided through the secure provisioning of an infrastructure using tools such as Terraform and Ansible. Critical security aspects will be explored, including attack surface management, configuration drift detection, the use of trusted images, and the implementation of least privilege policies. By the end of the course, participants are expected to understand the challenges of securing code-defined environments and to be able to apply Zero Trust principles in real-world provisioning scenarios, fostering infrastructures that are more reliable and resilient to lateral movements and insider threats.

Resumo

As abordagens tradicionais de segurança baseadas em perímetro têm-se mostrado insuficientes diante da crescente sofisticação das ameaças e da complexidade de ambientes computacionais modernos, dinâmicos e automatizados. Nesse cenário, este minicurso apresenta uma abordagem integrada que combina os princípios da arquitetura Zero Trust com as práticas de Infrastructure as Code (IaC), com o objetivo de promover ambientes mais seguros, resilientes e versionáveis desde o provisionamento. O ambiente de experimentação fornece uma fundamentação sobre automação de infraestrutura com IaC,

microssegmentação, controle de acesso baseado em identidade e análise contínua de contexto — elementos centrais para a mitigação de ameaças contemporâneas. Na etapa prática, os participantes serão orientados a provisionar uma infraestrutura segura utilizando ferramentas como Terraform e Ansible, nas quais serão explorados aspectos críticos de segurança, como o gerenciamento da superfície de ataque, a detecção de desvios de configuração, o uso de imagens confiáveis e a aplicação de políticas de mínimo privilégio. Ao final, espera-se que os participantes compreendam os desafios da segurança em ambientes definidos como código e estejam aptos a aplicar os princípios do modelo Zero Trust em cenários reais de provisionamento, promovendo infraestruturas mais confiáveis e robustas contra movimentações laterais e ataques internos.

1.1. Introdução

A Infraestrutura como Código (*Infrastructure as Code*, IaC) é uma técnica que viabiliza a automação, o provisionamento e a configuração de recursos computacionais de forma eficiente e reprodutível. Essa abordagem organiza o gerenciamento de *data centers* por meio da definição de arquivos legíveis por máquina (*machine-readable files*), eliminando a necessidade de configuração física de hardware ou da utilização de ferramentas interativas [Fowler 2013]. Por exemplo, utilizando Terraform, é possível definir toda a infraestrutura necessária para um cenário de aplicação — servidores, rede, bancos de dados etc. — em arquivos de configuração que podem ser versionados e reaplicados automaticamente, garantindo consistência e agilidade no ambiente de produção [Filho et al. 2025]. De forma complementar, o gerenciamento de contêineres empacotam a aplicação e suas dependências, facilitando o desenvolvimento, o teste e a implantação em ambientes isolados e reprodutíveis, usando Docker [Rodrigues et al. 2025]. No contexto da IaC, a integração entre Terraform e Docker possibilita que o Terraform automatize o provisionamento da infraestrutura subjacente, enquanto o Docker gerencia a implantação e execução das aplicações. Essa combinação promove uma orquestração completa, desde a configuração da infraestrutura até a entrega do ambiente de execução, garantindo controle, rastreabilidade e facilidade de reprodução.

A adoção da IaC otimiza processos, assegura a replicabilidade e promove ambientes computacionais mais padronizados e consistentes [Rahman et al. 2019a]. Muitas organizações que oferecem serviços em nuvem já incorporam tecnologias da IaC em seus fluxos de trabalho. Essa prática evidencia a importância do tema, pois seus benefícios facilitam o provisionamento automatizado e o gerenciamento por meio de *scripts*, reduzindo erros humanos, acelerando a entrega de recursos e garantindo maior controle sobre as configurações ao longo do ciclo de vida da infraestrutura.

Apesar das vantagens, a prática da IaC apresenta desafios significativos relacionados à segurança. A automação, embora eficiente, pode amplificar falhas existentes em configurações manuais, tornando mais evidentes problemas tradicionais, como a exposição acidental de credenciais em arquivos de configuração, permissões excessivas e a falta de controle sobre mudanças. Esses desafios, antes dispersos e difíceis de detectar, tornam-se mais críticos no contexto da IaC, onde erros se propagam rapidamente e em larga escala. Assim, a segurança do código IaC, a proteção da infraestrutura provisionada e o gerenciamento das modificações após o provisionamento são fundamentais para garantir uma implementação segura e resiliente. Os problemas tradicionais de segurança não

desaparecem com a automação, ao contrário, podem ser potencializados, mas a detecção e correção é facilitada. A facilidade proporcionada pela IaC para implementar e configurar recursos com agilidade exige atenção redobrada, mas favorece a segurança por projeto (*security by design*), a manutenção e operação do ambiente devido a programabilidade da IaC.

Por se tratar de uma implementação baseada em código, a identificação e correção de *code smells* – indicadores de problemas de manutenção ou qualidade no código-fonte, tornam-se um recurso para fortalecer a robustez do sistema [Rahman et al. 2019b]. Embora nem todo *code smell* represente uma falha funcional direta, sua presença pode revelar limitações estruturais que, por sua vez, podem gerar *security smells* — padrões de código associados a potenciais vulnerabilidades [Silveira Neto et al. 2024]. Dessa forma, o monitoramento tanto dos *code smells* quanto dos *security smells* facilita a manutenção e a legibilidade do código, além de atuar preventivamente, reduzindo riscos de segurança ao promover um código mais limpo e menos suscetível a falhas exploráveis.

Outro desafio relacionado à segurança em IaC está no tratamento de segredos e credenciais sensíveis nos *scripts* [Rahman et al. 2022]. Elementos como identificadores de usuário, senhas, *tokens* e chaves criptográficas são essenciais para a autenticação entre artefatos. Contudo, a prática comum de armazená-los em texto claro, inclusive em sistemas de controle de versão, expõe a infraestrutura a riscos significativos. Essas falhas de configuração são um fator de exposição recorrente [MITRE ATT&CK 2024], demandando a adoção de técnicas adequadas para proteger segredos dentro em IaC.

Os riscos mais frequentes na gestão de segredos incluem a falha de exposição de credenciais privilegiadas, o uso de segredos embutidos no código, a ausência de rotação periódica e a dependência de processos manuais [Rahman and Anwar 2021]. Embora o uso de cofres de senhas *(vault)* ofereça melhorias, essas soluções ainda apresentam vulnerabilidades, especialmente quando baseadas em arquiteturas com segurança por perímetro. A adoção de uma validação criteriosa e contínua para cada acesso visa reduzir a dependência do *(vault)*, garantindo que, mesmo em caso de exposição de segredos, seu uso seja restrito e monitorado, melhorando a segurança em ambientes da IaC.

O modelo de segurança perimetral tradicional pode apresentar falhas que comprometem a eficácia de mecanismos como *firewalls*, VPN *(Virtual Private Network)*, IDS *(Intrusion Detection System)* e IPS *(Intrusion Prevention System)* [Simioni et al. 2025b, Viegas et al. 2017], tornando-os insuficientes contra ameaças sofisticadas e movimentos laterais na rede. Nesse cenário, surge o modelo *Zero Trust* para, implicitamente, gerenciar por credenciais todos os acessos, seja para usuários, dispositivos ou aplicações. Na prática, esse modelo exige validação rigorosa e contínua de todas as entidades que acessam recursos, aplicando o mínimo privilégio (por padrão), autenticação, autorização e criptografia a cada requisição [Rose et al. 2020]. Essas características são especialmente relevantes em ambientes provisionados como código, onde a agilidade e a escala podem gerar exposição caso a segurança não seja incorporada por projeto *(security by design)*.

Em suma, *Zero Trust* rejeita a confiança automática e adota por padrão o mínimo privilégio, microssegmentação e autenticação multifator (*Multi-factor Authentication*, MFA) — aos fluxos de trabalho que junto com a IaC representam uma estratégia eficaz para mitigar vulnerabilidades em infraestruturas distribuídas e dinâmicas.

Conteúdo. Este minicurso tem como objetivo apresentar uma abordagem teórica e prática para o provisionamento de infraestrutura segura por meio da IaC, com ênfase na aplicação dos princípios do modelo *Zero Trust*. O minicurso abordará os fundamentos da IaC, capacitando os participantes a identificar e mitigar riscos como a exposição de segredos em código e o uso de imagens vulneráveis. Também será realizada uma análise comparativa entre o modelo de segurança perimetral tradicional, suas limitações e o risco de movimentos laterais, justificando a adoção da arquitetura *Zero Trust*. Serão detalhados os princípios desse modelo, como a verificação contínua, o mínimo privilégio e o acesso baseado em contexto. Numaa atividade prática os participantes utilizarão ferramentas como Terraform e Ansible para provisionar e configurar uma infraestrutura segura, sendo incentivados a refletir sobre a arquitetura implementada e as decisões de segurança adotadas.

Estrutura. O documento está organizada da seguinte forma. A seção 1.2 apresenta os fundamentos da IaC e da gestão de segredos. A seção 1.3 destacada as vulnerabilidades, assim como práticas de segurança aplicáveis à IaC, incluindo ferramentas de análise, verificação de imagens (binários) e prevenção de recursos órfãos. A seção 1.4 faz um comparativo entre a segurança tradicional, o modelo *Zero Trust* e os desafios específicos da IaC. A seção 1.5 detalha o ambiente de experimentação, enquanto a seção 1.6 apresenta um estudo de caso guiado por meio de experimentação de laboratório. Por fim, a seção 1.7 sintetiza as vantagens de segurança da abordagem apresentada.

1.2. Fundamentação

A abordagem tradicional de segurança foi estruturada com base em perimetros, geralmente, domínio (rede) local e externo. No domínio local, compartilhado em segmentos de rede interna, se assume que tudo é confiável, enquanto que no domínio externo nada é confiável. Portanto, o simples fato de ter acesso a rede interna já permite alcançar recursos que no cenário atual podem configurar ama superficie de ataque menos protegida. Em ambientes de computação em nuvem e práticas com o uso da IaC, esse modelo tem se mostrado cada vez mais limitado e vulnerável. A automação de infraestrutura em larga escala amplia significativamente a superfície de ataque e, na ausência de controles robustos, pode replicar configurações inseguras de forma massiva. Nesse contexto, o paradigma da *Zero Trust Architecture* (ZTA) ganha relevância ao rejeitar qualquer forma de confiança implícita e exigir validações contínuas de identidade, contexto e postura de segurança, independentemente da origem da solicitação. A integração entre IaC e os princípios do *Zero Trust* configura uma abordagem estratégica para mitigar riscos de exposição de recurso, permitindo que políticas de controle de acesso baseadas em credenciais, segmentação e autenticação sejam definidas e aplicadas diretamente no código da infraestrutura.

A seguir apresentamos a fundamentação em três abordagens principais: a automação por meio da IaC, a superação das limitações do modelo perimetral tradicional e a adoção do paradigma de *Zero Trust*. A partir dessas bases, discutimos os fundamentos técnicos e conceituais necessários para a construção de infraestruturas resilientes e seguras desde as etapas iniciais de seu provisionamento.

1.2.1. Infraestrutura como código (IaC): Conceitos e características

A prática da Infraestrutura como Código (*Infrastructure as Code*, IaC) representa uma mudança de paradigma na forma como as organizações configuram, provisionam e gerenciam seus recursos de tecnologia da informação. Tradicionalmente, esses processos exigiam intervenções manuais demoradas, sujeitas a erros e inconsistências operacionais. Com a adoção da IaC, essas atividades são automatizadas por meio de *scripts* e arquivos de configuração, promovendo ganhos significativos em eficiência, replicabilidade e versionamento, que permite auditabilidade de mudanças.

A IaC proporciona benefícios como o compartilhamento ágil de recursos, escalabilidade eficiente e redução da incidência de erros humanos [Özdoğan et al. 2023]. Essa abordagem utiliza ferramentas que interpretam arquivos de configuração escritos em linguagens declarativas ou imperativas, possibilitando a criação e o gerenciamento de servidores, redes, políticas de segurança, volumes de armazenamento e outros componentes críticos da infraestrutura computacional. A adoção da IaC reduz significativamente o tempo necessário para a implantação de ambientes, além de aumentar a previsibilidade e a gestão operacional [Wettinger et al. 2014]. Tratar a infraestrutura como código-fonte viabiliza práticas como o reuso de componentes, a colaboração entre equipes por meio de *pull requests* e a integração com *pipelines* de entregas contínuas (CI/CD) [Horchulhack et al. 2022a]. Essa integração reforça a padronização das configurações e contribui para a construção de ambientes mais seguros, versionáveis *(versionable)* e consistentes.

As ferramentas que implementam os conceitos da IaC podem ser classificadas de acordo com seus propósitos. Ferramentas de provisionamento como o Terraform[1], são responsáveis pela alocação de recursos em plataformas de computação em nuvem públicos, *on-premise* ou híbridos [Viegas et al. 2020]. Já ferramentas de configuração, como Ansible[2] e Puppet[3], concentram-se na definição de estados desejados e na aplicação de ajustes pós-provisionamento. Essas soluções permitem que a infraestrutura seja descrita por meio de arquivos versionáveis, com suporte a auditoria de mudanças e reutilizáveis, promovendo consistência entre ambientes e integrando-se de forma eficiente a fluxos de *DevOps*, por exemplo. Tipicamente, a IaC pode ser implementada segundo dois paradigmas principais [Chen et al. 2018]:

- **Modelo imperativo.** Nesse modelo, o responsável especifica detalhadamente a sequência de comandos que devem ser executados para criar e configurar os recursos de infraestrutura. A lógica de controle fica sob total responsabilidade do autor do código, que precisa definir a ordem exata das operações. Embora proporcione maior controle e flexibilidade sobre o processo, esse modelo pode tornar os *scripts* mais verbosos, difíceis de manter e propensos a erros em ambientes complexos;
- **Modelo declarativo.** Ao adotar esse modelo, o autor descreve o estado final desejado da infraestrutura, e a ferramenta encarrega-se de interpretar essa descrição e aplicar as ações necessárias para alcançar a configuração pretendida. Isso reduz a complexidade do código e facilita a manutenção, uma vez que o foco está no

[1]https://github.com/hashicorp/terraform
[2]https://github.com/ansible
[3]https://www.puppet.com/community

resultado e não na forma de alcançá-lo. Ferramentas como Terraform utilizam esse modelo, promovendo maior reprodutibilidade, controle de estado e consistência entre ambientes distintos.

Além da automação e da padronização, a IaC possibilita a criação de ambientes efêmeros, a realização de testes automatizados de infraestrutura e a reversão controlada de mudanças (*rollback*). Esses recursos ampliam significativamente a segurança operacional, a rastreabilidade das alterações e a velocidade de entrega de ambientes, ao permitir ciclos de desenvolvimento mais ágeis e confiáveis. Ambientes efêmeros, por exemplo, podem ser criados sob demanda para testes específicos e destruídos em seguida, reduzindo custos e diminuindo a superfície de ataque. Testes automatizados garantem que configurações estejam em conformidade antes da aplicação em produção, enquanto mecanismos de *rollback* permitem reverter mudanças de forma segura em caso de falhas.

No entanto, os benefícios trazidos pela IaC também introduzem novos desafios. A gestão de segredos sensíveis, como credenciais e chaves criptográficas, torna-se mais complexa quando embutida em código ou disseminada entre arquivos e repositórios. A detecção de derivações de estado (*drifts*) — situações em que a infraestrutura real diverge da definida no código — exige monitoramento constante e ferramentas que verifiquem a consistência. Dessa forma, a adoção segura da IaC requer a implementação de controles adicionais, revisão contínua e validações automatizadas ao longo de todo o ciclo de vida da infraestrutura.

1.2.1.1. Orquestração com Docker Compose

Além das ferramentas específicas da IaC para provisionamento e configuração, é comum a utilização de *containers* para estruturar ambientes modulares, isolados e portáveis. O Docker Compose[4] é uma ferramenta que permite definir e orquestrar múltiplos *containers* por meio de um único arquivo de configuração, o que facilita o controle de dependências, a padronização e a reprodutibilidade do ambiente, especialmente em fluxos de desenvolvimento e testes contínuos.

O código 1.1 apresenta um exemplo de arquivo `docker-compose.yml` que configura dois *containers*: um servidor de banco de dados MariaDB[5] e uma instância da aplicação GLPI[6], um sistema de gerenciamento de serviços de TI amplamente utilizado.

Esta configuração YAML define um ambiente Docker Compose com dois serviços: `mariadb` e `glpi`. O serviço `mariadb` utiliza a imagem oficial do MariaDB 10.5, configura variáveis de ambiente para a senha do usuário `root` e o nome do banco de dados, além de montar um volume para garantir persistência dos dados. O serviço `glpi` baseia-se em uma imagem personalizada, expõe a porta 8080 no host, ajusta o fuso horário e depende do serviço `mariadb`, assegurando que o banco de dados esteja ativo antes de sua inicialização. Ambos os serviços são conectados por meio de uma rede dedicada, e volumes para persistência são configurados para o armazenamento dos dados do banco,

[4]https://docs.docker.com/compose/

[5]https://mariadb.org/

[6]https://glpi-project.org/

Código Fonte 1.1. YAML exemplo no Docker Compose para execução do MariaDB e GLPI.

```yaml
version: "3.8" # Versão do Docker-compose
services:
  mariadb: # Serviço MariaDB
    image: mariadb:10.5 # Imagem
    container_name: mariadb_serv
    environment: # Variáveis de ambiente
      MYSQL_ROOT_PASSWORD: rootpass
      MYSQL_DATABASE: glpidb
    volumes:  # Volumes
      - db_data:/var/lib/mysql
    networks: # Redes
      - glpi_net

  glpi: # Serviço GLPI
    image: diouxx/glpi # Imagem
    container_name: glpi_serv
    ports: # Portas mapeadas
      - "8080:80"
    environment: # Variáveis de ambiente
      TIMEZONE: America/Sao_Paulo
    depends_on: # Serviço de dependência
      - mariadb
    networks: # Redes
      - glpi_net

volumes:
  db_data:

networks:
  glpi_net:
```

garantir integridade e isolamento. Na prática, este exemplo define os *containers* de forma declarativa, garantindo que o ambiente de aplicação esteja pronto para uso com um único comando: `docker-compose up -d`.

1.2.1.2. Descrevendo recursos com Terraform

O Terraform é uma ferramenta da IaC desenvolvida pela HashiCorp que permite o provisionamento, gerenciamento e atualização de recursos de forma automatizada e declarativa. Utilizando arquivos de configuração escritos em *HashiCorp Configuration Language* (HCL), o Terraform possibilita a descrição do estado desejado de uma infraestrutura, incluindo servidores, redes, balanceadores de carga, bancos de dados e outros serviços em provedores como AWS, Azure, Google Cloud, entre outros. Ao interpretar os arquivos, o Terraform realiza um plano de execução (`terraform plan`) que compara o estado atual com o estado desejado, calculando as ações necessárias para convergência. Esse mecanismo facilita a padronização de ambientes, o controle de mudanças e a rastreabilidade das operações, sendo especialmente útil em equipes que adotam práticas de DevOps e CI/CD. Além disso, o gerenciamento de estado permite detectar derivações e aplicar correções com segurança, promovendo consistência e versionamento na gestão da infraestrutura.

Código Fonte 1.2. Exemplo de configuração Terraform.

```
terraform {
  required_providers {
    local = {
      source  = "hashicorp/local"
      version = "~> 2.4"    # Versão requerida
    }
  }
}

# Provedor local (não exige configuração adicional)
provider "local" {}

# Recurso: arquivo de texto gerado pelo Terraform
resource "local_file" "exemplo" {
  filename = "exemplo.txt"
  content  = "Este é um exemplo prático com Terraform."
}
```

Para ilustrar, este exemplo prático apresenta a definição de um recurso básico de infraestrutura utilizando a ferramenta Terraform. O Código 1.2 especifica a criação de uma instância de Máquina Virtual (VM) em uma plataforma de computação em nuvem compatível com o provedor local.

Este *script* do Terraform configura o provedor local, que permite ao Terraform gerenciar os recursos do sistema local. Ele começa especificando o provedor necessário (*hashicorp/local*) com uma restrição de versão (> 2.4), garantindo a compatibilidade com uma versão estável. O bloco provider é declarado, mas não requer nenhuma configuração específica. Em seguida, um recurso do tipo *local_file* é definido, denominado exemplo. Este recurso instrui o Terraform a criar um arquivo chamado "*exemplo.txt*"contendo o texto: "*Este é um exemplo prático com Terraform*". O *script* é simples, mas demonstra como o Terraform pode ser usado para gerar arquivos locais como parte da automação de infraestrutura.

A Figura 1.1 apresenta uma organização típica de diretórios contendo arquivos do Terraform. Após a declaração dos recursos, a execução do comando `terraform plan` gera um plano de execução, conforme ilustrado na Figura 1.2. Esse plano permite ao usuário revisar as alterações antes de aplicá-las, analisando o código de configuração e comparando-o com o estado atual da infraestrutura.

O resultado é um plano detalhado que indica precisamente quais recursos serão criados (marcados com +), modificados (~) ou destruídos (−). Cada recurso é descrito com seus atributos, incluindo valores já definidos e aqueles que só serão conhecidos após a aplicação (*known after apply*). Ao final, um resumo como `Plan: 1 to add, 0 to change, 0 to destroy` fornece uma visão das mudanças previstas, garantindo controle e previsibilidade sobre as modificações no ambiente.

Esse modelo de execução contribui significativamente para a segurança e a previsibilidade da infraestrutura, ao permitir a revisão prévia das alterações, a verificação da conformidade com as políticas organizacionais e a prevenção de efeitos inesperados.

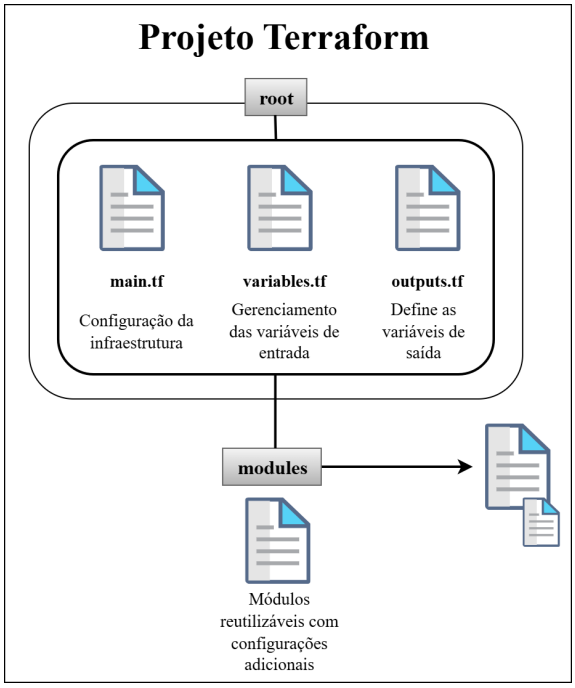


Figura 1.1. Exemplo de estrutura de diretórios de um projeto Terraform.

terraform plan

```
Terraform used the selected providers to generate the following execution plan.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_s3_bucket.example will be created
  + resource "aws_s3_bucket" "example" {
      + acl    = (known after apply)
      + arn    = (known after apply)
      + bucket = "meu-bucket-de-exemplo-12345"
      + id     = (known after apply)
      + tags   = {
          + Environment = "Dev"
          + ManagedBy   = "Terraform"
        }

      # (outros atributos omitidos para brevidade)
    }

Plan: 1 to add, 0 to change, 0 to destroy.
```

Figura 1.2. Saída do comando `terraform plan` para o exemplo apresentado no código 1.2.

Além disso, as mudanças no código torna-se auditável, reutilizável e facilmente integrável a *pipelines* automatizados, favorecendo práticas de DevOps e promovendo maior controle ao longo do ciclo de vida da infraestrutura.

Código Fonte 1.3. Playbook Ansible para instalar e iniciar o Apache.

```
- name: Instala e inicia o servidor web Apache
  hosts: servidores_web
  become: yes

  tasks:
    - name: Garante que o pacote do Apache2 está instalado
      ansible.builtin.apt:
        name: apache2
        state: present
        update_cache: yes

    - name: Garante que o serviço do Apache está iniciado
      ansible.builtin.service:
        name: apache2
        state: started
        enabled: yes
```

1.2.1.3. Configuração com Ansible

Além do provisionamento de recursos, a automação das configurações é uma etapa fundamental para assegurar consistência e segurança nos ambientes. O Ansible é uma ferramenta amplamente adotada para esse propósito, permitindo a definição do estado desejado dos sistemas por meio de arquivos YAML, que são simples, legíveis e facilmente versionáveis. Sua abordagem é baseada em um modelo declarativo, no qual são especificadas as tarefas que devem ser realizadas para alcançar uma determinada configuração, como a instalação de pacotes, a edição de arquivos, a criação de usuários ou a ativação de serviços. O Ansible opera de forma *agentless*, ou seja, não requer a instalação de agentes nos nós gerenciados. Em vez disso, utiliza conexões SSH para executar comandos remotamente, o que reduz a complexidade de manutenção e aumenta a portabilidade. Os *playbooks* — arquivos que descrevem as tarefas — são organizados em blocos reutilizáveis, o que favorece a modularização e a padronização de configurações em diferentes ambientes. Além disso, o Ansible integra-se facilmente a *pipelines* de CI/CD, permitindo aplicar configurações automaticamente após o provisionamento de infraestrutura, o que o torna uma peça chave em fluxos de trabalho DevOps baseados em IaC.

O código 1.3, apresenta um exemplo básico de *playbook* do Ansible que instala o servidor *web* Apache[7] em uma máquina Linux.

Este *playbook* do Ansible automatiza a instalação e a inicialização do servidor web Apache em um grupo de hosts denominado `servidores_web`. A execução é realizada com privilégios elevados (`become: yes`), garantindo permissões suficientes para modificar configurações e instalar pacotes em nível de sistema. O *playbook* é composto por duas tarefas principais:

- **Garantia que o Apache esteja instalado**: para tanto, utiliza o módulo nomeado `ansible.builtin.apt` para instalar o pacote `apache2`. A opção de atualiza-

[7] https://www.apache.org/

ção do cache de pacotes é ativada, assegurando que a versão mais recente disponível no repositório seja utilizada;

- **Garanta que o serviço Apache esteja ativo e habilitado**: emprega o módulo `ansible.builtin.service` para iniciar o serviço `apache2` e configurá-lo para iniciar automaticamente junto com o sistema operacional;

Com essas duas tarefas, o *playbook* garante que o servidor Apache esteja corretamente instalado, ativo e configurado para iniciar automaticamente em todas as máquinas de destino definidas no inventário. Essa abordagem simplifica o gerenciamento de servidores web e assegura consistência na configuração dos ambientes.

Docker, Terraform e Ansible são ferramentas complementares amplamente utilizadas em conjunto para provisionar, configurar e operar ambientes de infraestrutura programável, especialmente em contextos baseados em DevOps e automação. Em um fluxo típico, o Terraform é utilizado inicialmente para provisionar os recursos de infraestrutura, como máquinas virtuais, redes, volumes de armazenamento e balanceadores de carga, seja em nuvem pública ou *on-premise* (ambientes locais). Após o provisionamento, o Ansible pode ser utilizado para configurar esses recursos, instalando pacotes, ajustando serviços e aplicando políticas de segurança de forma declarativa. O Docker, por sua vez, pode ser utilizado para encapsular aplicações e suas dependências em *containers* portáveis e reprodutíveis, garantindo isolamento entre os serviços. Um cenário prático que ilustra essa integração seria o provisionamento de um cluster de servidores em uma nuvem privada (*on-premise*) com Terraform, seguido pela configuração automática de cada nó com Ansible para instalação do Docker e de serviços de orquestração (como Docker Compose ou Kubernetes). Em seguida, as aplicações são implantadas como *containers*, garantindo agilidade, consistência e escalabilidade no ambiente.

1.2.1.4. Validação contínua com Checkov

Um dos principais desafios na adoção da IaC é garantir que o código da infraestrutura esteja em conformidade com boas práticas de segurança antes de ser aplicado em ambientes de produção. A natureza automatizada e reprodutível da IaC, embora traga ganhos em eficiência, também amplifica o impacto de erros de configuração ou permissões indevidas, por exemplo. Nesse contexto, a análise estática de código se torna uma etapa importante do processo, permitindo inspecionar os arquivos de configuração sem executá-los, com o objetivo de detectar vulnerabilidades, omissões e desvios de compatibilidade de forma antecipada. Ferramentas como o Checkov[8], CloudFormation Linter[9] e kube-linter[10] aplicam regras pré-definidas para examinar arquivos Terraform, CloudFormation ou Kubernetes, sinalizando práticas inseguras como a ausência de criptografia, exposição de serviços à internet pública, uso de senhas em texto claro ou a falta de controle de identidade e acesso. A integração dessas ferramentas a fluxos de CI/CD permite que a validação de segurança

[8]https://www.checkov.io/

[9]https://github.com/aws-cloudformation/cfn-lint

[10]https://kube-linter.io/

checkov saída

```
checkov -d ./infraestrutura/terraform

By bridgecrew.io | version: 2.3.187

terraform scan results:

Passed checks: 4, Failed checks: 3, Skipped checks: 0

Check: CKV_AWS_19: "Ensure all data stored in the S3 bucket is securely encrypted at
rest"
  FAILED for resource: aws_s3_bucket.data_bucket
  File: /main.tf:10-21
  Guide: https://docs.bridgecrew.io/docs/s3_16-enable-bucket-encryption-in-rest

  Code lines: 10-21
  10 | resource "aws_s3_bucket" "data_bucket" {
  11 |   bucket = "meu-bucket-de-dados-importantes"
  ...
  18 |   # FALHA: Bloco de criptografia do servidor ausente
  20 | }

Check: CKV_AWS_21: "Ensure all data stored in the S3 bucket has versioning enabled"
  FAILED for resource: aws_s3_bucket.data_bucket
  File: /main.tf:10-21
  Guide: https://docs.bridgecrew.io/docs/s3_15-enable-versioning

Check: CKV_AWS_54: "Ensure S3 bucket has block public policy enabled"
  FAILED for resource: aws_s3_bucket.data_bucket
  File: /main.tf:10-21
  Guide: https://docs.bridgecrew.io/docs/s3_10-enable-block-public-policy
```

Figura 1.3. Resultado da validação com Checkov, identificando falhas em um recurso Terraform.

ocorra de forma automatizada e contínua, reduzindo significativamente os riscos antes do provisionamento efetivo da infraestrutura.

No exemplo a seguir, utiliza-se o comando *checkov -d ./infraestrutura/terraform* para analisar um projeto Terraform que cria um bucket S3 na AWS[11]. A Figura 1.3 apresenta a execução do Checkov nesse diretório, evidenciando as verificações realizadas e os possíveis alertas gerados. Essa ferramenta pode ser facilmente utilizada pelo usuário para identificar vulnerabilidades e garantir a conformidade das configurações antes da aplicação das mudanças na infraestrutura.

A verificação de segurança efetuada pelo Checkov realizou um total de sete checagens, das quais quatro foram aprovadas e três reprovadas. Todas as falhas estão relacionadas ao recurso `aws_s3_bucket.data_bucket`, definido entre as linhas 10 e 21 do arquivo `main.tf`. Especificamente, o bucket não possui criptografia do lado do

[11] https://aws.amazon.com

servidor para dados em repouso, o versionamento está desabilitado e falta a configuração para bloqueio de políticas públicas, o que representa riscos significativos à segurança, sendo três práticas inseguras segundo referências de conformidade como CIS Benchmarks [Center for Internet Security (CIS)] e NIST [Mell and Grance 2011]. Para cada falha são fornecidas referências a documentações detalhadas que orientam na correção dessas vulnerabilidades e na melhoria da segurança do bucket.

Outras ferramentas complementares, como tfsec[12], TFLint[13] e Terrascan[14], também podem ser empregadas em conjunto para fortalecer a qualidade e a segurança do código IaC, oferecendo diferentes níveis de análise estática, detecção de vulnerabilidades e boas práticas específicas para ambientes Terraform.

1.2.2. Ferramentas e ecossistemas da IaC

A crescente adoção da IaC estimulou o desenvolvimento de um ecossistema diversificado de ferramentas especializadas, que atuam em diferentes etapas do ciclo de vida da infraestrutura, considerando desde o provisionamento de recursos até a configuração, monitoramento, segurança, versionamento e integração com *pipelines* de entrega contínua.

Ferramentas como Terraform e OpenTofu[15] são amplamente utilizadas para o provisionamento declarativo de recursos em provedores de nuvem como AWS, Azure[16], GCP[17], além de ambientes *on-premise*. Por meio de arquivos de configuração, os administradores descrevem o estado desejado da infraestrutura, enquanto os motores dessas ferramentas aplicam as mudanças necessárias com base em planos de execução versionáveis. A *HashiCorp Configuration Language* (HCL), utilizada pelo Terraform, consolidou-se como referência nesse domínio, por seu equilíbrio entre simplicidade e expressividade [Wang 2022].

No contexto da configuração de sistemas e aplicações, ferramentas como Ansible, SaltStack[18] e Puppet possibilitam o gerenciamento automatizado de pacotes, serviços, arquivos e políticas. Essas soluções atuam de forma complementar ao provisionamento, assegurando que os recursos recém-criados sejam inicializados e configurados corretamente conforme os padrões definidos pela equipe técnica. A combinação de ferramentas de provisionamento e configuração reforça a consistência entre ambientes — desenvolvimento, testes e produção — além de aprimorar a rastreabilidade das alterações realizadas [Rahman et al. 2019a].

Além das ferramentas principais, há um conjunto crescente de soluções voltadas à verificação, segurança e conformidade da infraestrutura codificada. Ferramentas como Trivy[19], Checkov, Tfsec e Terrascan realizam análises estáticas nos arquivos de configuração, identificando problemas como recursos públicos expostos inadvertidamente, ausência de criptografia, falhas no controle de acesso e presença de credenciais em texto claro

[12]https://aquasecurity.github.io/tfsec/latest/
[13]https://github.com/terraform-linters/tflint
[14]https://runterrascan.io/
[15]https://opentofu.org
[16]https://azure.microsoft.com
[17]https://cloud.google.com
[18]https://saltproject.io/
[19]https://trivy.dev

[Reddy Konala et al. 2023]. Essas boas práticas garantem proteção contra configurações inseguras ou fralhas de configuração não intencionais.

O ecossistema da IaC inclui serviços de armazenamento seguro de segredos, como o HashiCorp Vault[20], que se integram aos *scripts* de provisionamento e configuração para fornecer chaves, *tokens* e senhas de forma segura e auditável. Combinados com políticas de controle de acesso e autenticação por provedor de identidada (IdP), esses serviços previnem a exposição de informações sensíveis e permitem a aplicação de princípios fundamentais de segurança, como mínimo privilégio e tempo de vida limitado das credenciais [Oliveira et al. 2024].

Além disso, ferramentas de integração contínua, como GitLab CI[21], GitHub Actions[22] e Jenkins[23], atuam como orquestradores no *pipeline* automatizado de infraestrutura. Elas possibilitam a execução programada ou condicional de planos de provisionamento, validações de segurança, testes e implantações, promovendo práticas de auditoria contínua, versionamento, *rollback* e rastreabilidade — aspectos essenciais para ambientes dinâmicos e baseados em microserviços.

1.2.3. Gestão de segredos e práticas de proteção

Em ambientes automatizados com IaC, a gestão de segredos constitui um aspecto crítico para assegurar a proteção de credenciais, chaves de API, certificados e *tokens* de acesso utilizados durante o provisionamento e a configuração de recursos. Como os *scripts* da IaC circulam por repositórios versionados, *pipelines* de CI/CD e múltiplos ambientes de execução, qualquer vazamento acidental de segredos representa um sério risco à integridade e à segurança da infraestrutura [Rahman et al. 2021a, Rahman et al. 2021b].

Embora ferramentas como Ansible, Terraform e Kubernetes ofereçam recursos robustos para automação, o manuseio seguro de informações sensíveis depende da adoção de boas práticas. O uso de arquivos criptografados (como o Vault do Ansible) ou integrações com sistemas externos de gerenciamento de segredos — como HashiCorp Vault, AWS Secrets Manager e Azure Key Vault — é amplamente recomendado para reduzir a superfície de exposição [Yuliarman et al. 2023, Oliveira et al. 2024, Vehent, Julien 2018].

A adoção de soluções especializadas para gerenciamento de segredos (*vaults*) viabiliza não apenas o armazenamento seguro, mas também funcionalidades como rotação automática de credenciais, controle de acesso baseado em políticas e registro detalhado de mudanças [Abreu et al. 2020]. A simples detecção de segredos em código-fonte não é suficiente sem uma estratégia efetiva de mitigação e prevenção [Rahman et al. 2022]. Nesse contexto, a aplicação de práticas como o uso de *placeholders*, a integração com ferramentas automatizadas de escaneamento de segredos em *pipelines* (como Gitleaks e TruffleHog) e a implementação do princípio do minimo privilégio tornam-se componentes essenciais de uma postura segura em ambientes com IaC.

Além disso, observa-se um movimento crescente em direção à incorporação de

[20]https://developer.hashicorp.com/vault
[21]https://docs.gitlab.com/ci
[22]https://github.com/features/actions
[23]https://www.jenkins.io

autenticação não interativa e baseada em múltiplos fatores, como *One-Time Password* (OTP) e certificados digitais, com o objetivo de automatizar o acesso a sistemas sem comprometer a segurança. Dessa forma, a gestão de segredos em ambientes com IaC requer uma abordagem DevSecOps (desenvolvimento, segurança e operação) mais desenvolvida em que segurança, automação, operação e gestão atuem de forma integrada. Essa abordagem garante que segredos sejam tratados como ativos sensíveis, com ciclos de vida bem definidos, rastreabilidade completa e mecanismos de proteção desde a definição no código até a execução em produção.

A gestão segura de credenciais é também um elemento central para a aplicação prática do modelo *Zero Trust* em ambientes automatizados com IaC. Em vez de depender de autenticações interativas (com intervenção de um ator humano) — suscetíveis a falhas operacionais e vazamentos, devido a exposição de credenciais quando o humano não pode estar presente — esse modelo privilegia mecanismos automatizados, granulares e versionáveis, que reforçam o princípio de não confiar implicitamente em nenhum ator, mesmo nos limites do perímetro da rede interna.

Uma abordagem recomendada para a gestão segura de credenciais em ambientes com IaC é a utilização de cofres de senhas (*vault*), como o HashiCorp Vault, que permitem armazenar e distribuir segredos de forma segura e programática. Contas de serviço utilizadas em *pipelines* de CI/CD ou em *scripts* de automação podem acessar os recursos por meio de *tokens* gerados sob demanda, com tempo de vida limitado e escopo específico. Essa estratégia elimina a necessidade de armazenar senhas ou chaves de acesso diretamente no código-fonte ou em repositórios versionados, reduzindo significativamente o risco de exposição e comprometimento.

As permissões associadas a essas contas de serviço seguem o princípio do mínimo privilégio (*least privilege*) e são definidas por meio de políticas de acesso declarativas, codificadas nos próprios arquivos de infraestrutura. Isso garante que os acessos sejam configurados de forma rastreável, por versionamento, e coerente com as diretrizes de segurança da organização, promovendo maior controle sobre o ciclo de vida de permissões de acesso.

A autenticação não interativa também possibilita a execução de tarefas recorrentes, atualizações automatizadas e verificações de integridade sem intervenção humana, aumentando a escalabilidade e a resiliência dos sistemas. Dessa forma, o processo de provisionamento e manutenção de infraestrutura torna-se não apenas mais eficiente, mas também mais seguro, uma vez que os mecanismos de autenticação e autorização são continuamente validados e monitorados por sistemas autoamtizados.

1.3. Segurança e riscos em ambientes automatizados com IaC

A prática da IaC transformou a forma como ambientes computacionais são provisionados e gerenciados, promovendo agilidade, padronização e escalabilidade. Contudo, essa automação intensiva também introduz novos vetores de risco à cibersegurança, exigindo atenção quanto à exposição, inclusive de informações sensíveis, à validação das configurações e à manutenção do código da IaC.

Ao tratar a infraestrutura como código-fonte, os riscos tradicionalmente associados

ao desenvolvimento de *software* passam a afetar diretamente essa camada, tornando o código IaC um ativo estratégico. Qualquer falha nesse código — como concessão de permissões excessivas, exposição indevida de portas de rede ou inclusão de segredos embutidos (*tokens*, senhas, chaves de API) — pode comprometer múltiplos ambientes simultaneamente, especialmente em cenários com replicação automatizada [Rahman et al. 2019b].

Um dos desafios recorrentes nesse contexto são os chamados *security smells*, padrões de codificação que indicam potenciais vulnerabilidades. Ferramentas amplamente adotadas, como Ansible, Terraform e Chef, podem potencializar tais falhas sistêmicas, incluindo o uso de senhas fixadas diretamente no código (*hardcoded*)[NIST 2025], a ausência de validação de entradas e o emprego de parâmetros inseguros por padrão[Basak et al. 2022, Rahman et al. 2021a]. Essas práticas tornam-se especialmente perigosas quando o código IaC é versionado em repositórios públicos ou mantido sem controles adequados, ampliando significativamente a superfície de ataque.

Outro aspecto relevante é o fenômeno conhecido como *configuration drift*, que ocorre quando o estado real da infraestrutura diverge do estado declarado no código. Essa discrepância pode ser causada por alterações manuais no código, falhas na aplicação dos *scripts* ou inconsistências na sincronização entre ambientes. O resultado é uma infraestrutura instável e propensa a erros, com impactos diretos na confiabilidade, na rastreabilidade e na segurança dos sistemas [Rahman et al. 2019a].

A segurança de ambientes automatizados por meio da IaC está intrinsecamente ligada à confiabilidade dos artefatos utilizados durante o provisionamento. Entre esses, destacam-se as imagens (binários) do sistema operacional e de *containers*, que servem como base para instanciar servidores, serviços e aplicações [Horchulhack et al. 2022b]. Quando mal configuradas, desatualizadas ou não validadas, essas imagens podem conter fragilidades conhecidas, *malwares* ou configurações inseguras, comprometendo a integridade, a estabilidade e a segurança de toda a infraestrutura.

Imagens comprometidas ampliam a superfície de ataque e facilitam a exploração por agentes maliciosos, sobretudo em ambientes de larga escala, nos quais a replicação automatizada a partir de um mesmo códgo pode propagar falhas para dezenas ou centenas de instâncias desse *template* – arquivos reutilizáveis que definem a configuração desejada da infraestrutura [Queen 2024]. Nesse contexto, a verificação contínua e automatizada de imagens, utilizando ferramentas específicas de análise de vulnerabilidades, torna-se uma prática importante para garantir arquiteturas seguras e resilientes baseadas em IaC.

Para mitigar esses riscos, é fundamental integrar ferramentas de análise estática e verificadores de configuração ao ciclo de desenvolvimento da IaC. Soluções como Trivy, Checkov, tfsec, Terrascan e Docker Content Trust[24] realizam inspeções automatizadas nos *scripts* e nos binários das imagens utilizadas, identificando vulnerabilidades conhecidas, violações de políticas organizacionais e desvios em relação às boas práticas de segurança [Reddy Konala et al. 2023, Chiari et al. 2022]. Essas ferramentas funcionam como recurso de análise preventiva no ciclo DevSecOps, promovendo boas práticas de cibersegurança desde as fases iniciais do processo de automação (programabilidade) da infraestrutura.

[24]https://docs.docker.com/engine/security/trust/

Neste caso, a segurança das imagens em IaC é garantida pelo uso de repositórios confiáveis, com autenticidade e integridade dos artefatos por meio de assinaturas digitais. Soluções como o Sigstore[25], especialmente com sua ferramenta Cosign[26], permitem assinar imagens no formato OCI e registrar essas assinaturas em *logs* que permitem rastrear as mudanças, assegurando que apenas imagens verificáveis e não adulteradas sejam implantadas [Lorenc 2021].

A adoção de assinatura digital necessita que fatores como usabilidade, integração com ambientes automatizados e suporte à *pipelines* de CI/CD estejam disponíveis a fim de garantir a eficácia e a adoção em larga escala dessas tecnologias [Kalu et al. 2025]. Esses elementos tornam-se indispensáveis para compor uma cadeia de fornecimento segura de software dentro de infraestruturas baseadas em código. Obviamente, não é necessariamente indispensável o uso de DevSecOps com IaC, mas como essa abordagem tem a automatização como objetivo e mais comumente mencionada lá.

Cosign é uma ferramenta do ecossistema Sigstore projetada para assinar, verificar e armazenar assinaturas de imagens de *containers* de forma segura, automatizada e compatível com fluxos de DevSecOps. Seu principal objetivo é garantir que apenas imagens autenticadas e provenientes de fontes confiáveis sejam utilizadas em ambientes de produção, reduzindo o risco de execução de artefatos adulterados ou maliciosos.

O fluxo de assinatura com Cosign/Sigstore caracteriza-se por três etapas principais que reforçam a integridade e a autenticidade de imagens de *containers*:

1. **Assinatura da imagem.** O desenvolvedor ou sistema automatizado utiliza o Cosign para gerar ou obter uma chave criptográfica (ou certificado, muitas vezes emitido de forma efêmera pela infraestrutura do Sigstore) e assinar a imagem de *container* com base em seu *digest* criptográfico (normalmente um hash SHA256) [Boulden 2023]. Isso garante que a assinatura esteja vinculada de forma única ao conteúdo da imagem;

2. **Armazenamento e registro.** A assinatura, juntamente com o certificado utilizado, é armazenada no próprio registro de imagens (como Docker Hub, GHCR ou Harbor), em um local separado mas associado ao *digest* da imagem. Paralelamente, os metadados dessa assinatura são registrados em um *log* de transparência, como o Rekor, o que permite auditoria pública e impede alterações silenciosas, promovendo a imutabilidade e a confiança no histórico da imagem.

3. **Validação no deployment.** Durante o processo de implantação, ferramentas de política como Kyverno[27], OPA/Gatekeeper[28] ou validações automatizadas no *pipeline* verificam a presença e a validade da assinatura digital. Imagens não assinadas ou cuja assinatura não corresponda ao certificado confiável são automaticamente rejeitadas, evitando implantações de artefatos comprometidos.

Esse fluxo garante que apenas imagens autenticadas e verificáveis sejam executadas

[25]https://www.sigstore.dev/
[26]https://github.com/sigstore/cosign
[27]https://kyverno.io/
[28]https://github.com/open-policy-agent/gatekeeper

em ambientes sensíveis, contribuindo para a construção de cadeias de suprimento de software seguras e versionáveis.

Ao incorporar esse processo no *pipeline* da IaC, obtém-se uma camada adicional de confiança automática e verificável, uma vez que somente imagens assinadas e auditadas são provisionadas, complementando a verificação estática e a validação de recursos existentes. Essas assinaturas funcionam como certificados de origem, sendo fundamentais para suportar ambientes seguros e alinhados ao modelo *Zero Trust*. Adicionalmente, o uso de repositórios oficiais ou internos validados contribui para o controle do ciclo de vida das imagens, evitando atualizações não autorizadas ou modificações maliciosas. Esse controle, aliado à inspeção contínua, fortalece a gestão dos recursos utilizados na infraestrutura automatizada.

Além disso, a incorporação de práticas de segurança em todas as etapas do *pipeline* de entrega contínua é um princípio fundamental do paradigma DevSecOps. Isso envolve a execução de testes automatizados nos *scripts* da IaC, a aplicação consistente do princípio do menor privilégio, a gerstão de mudanças e a segmentação dos ambientes de acordo com seus respectivos níveis de risco. Incorporar a segurança como parte integrante do ciclo de vida da infraestrutura, e não como uma fase posterior, é essencial para a construção de ambientes resilientes e versionáveis [Alonso et al. 2023].

Nesse cenário, é igualmente importante evitar inconsistências que levam a criação de recursos órfãos, ou *orphaned resources* — componentes que permanecem ativos mesmo após terem sido removidos do código de infraestrutura. Esses recursos não referenciados consomem recursos computacionais, violam os processos regulares de verificação, auditoria e aplicação de políticas, além de representarem potenciais vetores de ataque e gerarem custos operacionais desnecessários em grande escala [CloudOptimo 2025].

A mitigação desses riscos requer a adoção de mecanismos de verificação automatizada, como os fornecidos por serviços como AWS Config e AWS Trusted Advisor, bem como a implementação de políticas de *decommissioning* que detectem e eliminem recursos não declarados explicitamente no código da IaC. Esse tipo de controle de manutenção é importante para garantir ambientes mais seguros, eficientes e gerenciais [Mangera 2025].

Dessa forma, a segurança em ambientes automatizados com IaC demanda uma abordagem multidimensional, que abrange desde a adoção de boas práticas de codificação e controle de versões até o uso de ferramentas automatizadas para verificação, gestão segura de segredos e aplicação contínua de políticas de manutenibilidade. O foco não se limita à proteção do código em si, mas se estende à preservação da integridade, confidencialidade e disponibilidade de toda a infraestrutura como código, de maneira proativa, escalável e versionável.

1.3.1. *Code smells* e *Security smells* em IaC

No contexto da engenharia de *software*, o termo *code smell* refere-se a indícios de problemas estruturais no código que, embora não causem falhas funcionais imediatas, podem comprometer sua manutenibilidade, legibilidade e robustez. Em ambientes baseados em IaC, essas fragilidades assumem um papel ainda mais crítico, pois afetam diretamente a segurança e a confiabilidade dos ambientes computacionais provisionados. Os *code smells*

presentes em *scripts* IaC frequentemente antecedem ou facilitam o surgimento de vulnerabilidades conhecidas como *security smells* [Rahman et al. 2021a]. Estes representam padrões recorrentes de práticas inseguras ou mal estruturadas, como o uso de senhas e *tokens* em texto claro, a concessão excessivas de privilégios de acesso, a exposição de interfaces administrativas ou a ausência de controles explícitos de autenticação. Embora essas práticas possam inicialmente parecer inofensivas, elas frequentemente se tornam vetores de exploração para agentes maliciosos, comprometendo a integridade do ambiente automatizado.

A presença desses *security smells* está frequentemente ligada à pressa na entrega, à ausência de revisão por pares e testes, e à falta de políticas formais de desenvolvimento seguro. Ferramentas de análise estática específicas para IaC têm sido utilizadas para identificar essas falhas, possibilitando que as equipes corrijam os problemas antes da execução dos *scripts*. Entre os principais tipos de *security smells* identificados na literatura, destacam-se:

- **Uso de segredos embutidos**: refere-se à prática de inserir credenciais, como senhas, chaves de API ou tokens de acesso diretamente nos arquivos de configuração ou *scripts* IaC. Essa abordagem facilita a exposição acidental desses dados sensíveis em repositórios públicos ou internos, aumentando o risco de comprometimento da infraestrutura. Além disso, dificulta a rotação e o controle centralizado dos segredos, negligêncaindo boas práticas de segurança.

- **Configurações permissivas**: envolve a ausência ou inadequação de restrições de acesso a recursos, incluindo a abertura desnecessária de portas TCP (*Transmission Control Protocol*), a exposição pública de interfaces administrativas ou a definição de permissões excessivas para usuários e serviços. Essas configurações ampliam a superfície de ataque, facilitando a exploração por agentes maliciosos e aumentando a probabilidade de incidentes de segurança.

- **Ausência de validação**: refere-se à falta de mecanismos para verificar a integridade, autenticidade ou conformidade das configurações aplicadas pela IaC. Isso pode incluir a ausência de validações de entrada, testes automatizados ou checagens de políticas de segurança que garantam que o estado provisionado corresponde às diretrizes definidas. A consequência é a possível implantação de configurações incorretas, inseguras ou incompatíveis, comprometendo boas práticas de segurança dos ambientes.

- **Dependências não verificadas**: diz respeito ao uso de imagens de sistema operacional, contêineres, módulos ou outros recursos cuja origem, integridade e atualizações de segurança não são verificadas adequadamente. Essa prática pode introduzir vulnerabilidades conhecidas, *backdoors* ou componentes desatualizados na infraestrutura, ampliando riscos e dificultando a manutenção com políticas de segurança e a rastreabilidade com o versionamento.

A mitigação desses *security smells* requer a adoção de ferramentas automatizadas de análise estática e auditoria de mudanças, além da implementação de padrões de

Tabela 1.1. Principais vulnerabilidades em IaC, descrições e ferramentas de mitigação

Vulnerabilidade	**Descrição**	**Mitigação (Ferramenta)**
Uso de segredos embutidos	Credenciais, tokens ou senhas codificados diretamente no código-fonte.	Gestão de segredos: Gitleaks, HashiCorp Vault, AWS Secrets Manager
Configurações permissivas	Falta de restrições adequadas ou permissões excessivas.	Análise estática: Checkov, tfsec, OPA/Gatekeeper, Kyverno
Ausência de validação	Falta de avaliação sistêmica da integridade e conformidade das configurações.	Validação e testes: Terraform Validate, Terrascan
Dependências sem auditabilidade de mudanças	Uso de imagens ou módulos sem verificação de origem e atualizações.	Scanner de vulnerabilidades: Trivy, Docker Content Trust, Cosign/Sigstore
Versões desatualizadas de ferramentas	Ferramentas e módulos IaC desatualizados podem conter falhas.	Gerenciamento de dependências: Renovate, Dependabot
Configuração incorreta/mal configurada	Parâmetros incorretos ou inadequados que podem comprometer a segurança.	Análise estática e validação: Checkov, tfsec, terrascan; testes em ambiente isolado
Gerenciamento inadequado de segredos	Falta de controle de acesso ou armazenamento seguro dos segredos utilizados.	Cofres de segredos: HashiCorp Vault, AWS Secrets Manager
Falta de validação e testes automatizados	Ausência de processos que verifiquem a qualidade e segurança dos *scripts*.	Integração contínua: GitLab CI, GitHub Actions, Jenkins; ferramentas de teste IaC
Permissões excessivas em recursos	Concessão de privilégios sem adoção do principio do mínimo privilégio.	Políticas de acesso: OPA/Gatekeeper, Kyverno; revisão de permissões IaC
Uso de módulos ou binários de imagens não confiáveis	Inclusão de componentes de fontes desconhecidas ou sem auditoria prévia.	Assinatura e verificação de imagens: Cosign/Sigstore, Docker Content Trust
Ausência de monitoramento e *logging* adequado	Falta de registros de versionamento que permitam auditoria de mudanças e detecção de atividades suspeitas em recursos provisionados.	Ferramentas de monitoramento: AWS CloudTrail, Azure Monitor; alertas configurados
Provisionamento de recursos não seguros	Criação de recursos com configurações inseguras ou sem proteção adequada.	Análise estática, testes e políticas: tfsec, Checkov, Kyverno; revisão manual
Falta de controle de versão e mudanças	Ausência de versionamento, auditoria e *rollback*.	Sistemas de versionamento: Git e ferramentas de revisão

desenvolvimento seguro[29]. Integração de revisões contínuas e verificações de segurança nos *pipelines* de entrega permite que, em tempo, se identifique falhas e a correção seja realizada antes da aplicação das configurações. Essas práticas não apenas fortalecem a segurança da infraestrutura, mas também promovem a maturidade operacional das equipes responsáveis pela gestão de ambientes definidos como código.

A Tabela 1.1 apresenta as principais vulnerabilidades encontradas em ambientes da IaC, acompanhadas de descrições e das ferramentas recomendadas para sua mitigação. A tabela destaca problemas comuns como o uso de segredos embutidos no código, configurações permissivas e mal configuradas, além da ausência de validação e testes automatizados,

[29]https://www.microsoft.com/en-us/securityengineering/sdl/practices

que podem comprometer a segurança e o funcionamento adequado da infraestrutura. Também são evidenciados riscos relacionados ao uso de versões desatualizadas, dependências decorrentes de mudanças não auditadas e imagens não confiáveis, reforçando a importância da gestão adequada de segredos, controle de privilégios e monitoramento contínuo. As ferramentas citadas, como Checkov, tfsec, Trivy, Cosign e HashiCorp Vault oferecem soluções automatizadas para detecção, validação e gestão, integrando-se aos pipelines de desenvolvimento para garantir práticas seguras e auditoria de mudanças ao longo do ciclo de vida da infraestrutura.

Na próxima seção, serão apresentadas as práticas recomendadas para a gestão segura de segredos e credenciais em ambientes IaC, com foco no uso de cofres de senhas (*vault*), políticas de rotação e controle de acesso.

1.4. Da segurança perimetral à arquitetura *Zero Trust*

Durante décadas, o modelo tradicional de cibersegurança das organizações usa o modelo baseado em proteção de perimetro. Essa abordagem assume que, uma vez autenticados, os usuários tem acesso a uma rede delimitada por um perimetro construído por meio de *firewalls*, VPNs ou *proxies* etc. Dentro desse perimetro todos (usuários e sistemas) são confiáveis e podem acessar os recursos compartilhados. No cenário atual, caracterizado pelo trabalho remoto, mobilidade corporativa, computação em nuvem, e da crescente complexidade dos ambientes distribuídos, esse paradigma tornou-se limitado porque se perde a noção de perímetro, tornando muito díficil garantir uma proteção eficaz diante dos vetores de ataque atuais [Simioni et al. 2025a, Rose et al. 2020].

O modelo de arquitetura de *Zero Trust* (*Zero Trust Architecture*, ZTA) surge como uma resposta a essas limitações, reformulando a noção de confiança na infraestrutura de TI. O modelo *Zero Trust* adota o princípio "*nunca confie, sempre verifique*" [Rose et al. 2020]. Isso isignifica que nenhum dispositivo, usuário ou aplicação deve ser considerado confiavel de forma implícita, mesmo que esteja dentro dentro de um perimetro (no domínio interno de uma organização). A validação da confiança passa a ser contínua e contextual, considerando múltiplas camadas de proteção como identidade, postura do dispositivo, localização, contexto da solicitação e nível de privilégio concedido.

Nos fundamentos técnicos da ZTA destacam-se a microsegmentação da rede, a autenticação contínua e multifatores, controle de acesso granular baseado em políticas (como ABAC ou RBAC [Abreu et al. 2017]) e a visibilidade contextual sobre os fluxos de tráfego e operações. Esses princípios são implementados com o suporte de ferramentas como *gateways* de acesso definido por software (*Software Defined Perimeter*, SDP), sistemas de identidade baseado em provedor (*Identity Provider*, IdP), cofres de segredos e serviços de verificação contínua da postura de segurança [Syed et al. 2022, Marquis 2024, Matthew Kosinski 2024].

A integração entre o modelo *Zero Trust* e a abordagem da IaC é particularmente estratégica em ambientes nativos de nuvem computacional. Ao tratar a infraestrutura como código, torna-se possível aplicar controles de autenticação, autorização e segmentação de forma padronizada e reproduzível, desde as fases iniciais do ciclo de provisionamento. Por exemplo, ao utilizar Terraform para orquestrar recursos na nuvem computacional, é viável incorporar regras que restringem o tráfego por região

geográfica, exigência de autenticação baseada em chaves rotativas armazenadas em cofres seguros e configuração de segmentações lógicas por serviço ou domínio de aplicação [Rahman et al. 2022, Oliveira et al. 2024].

Ao contrário da segurança perimetral, que frequentemente se apoia em regras estáticas e segmentações baseadas em topologia física, a integração entre IaC e o modelo *Zero Trust* viabiliza uma infraestrutura adaptável, capaz de responder dinamicamente a mudanças no contexto operacional. Essa flexibilidade é fundamental para mitigar ameaças atuais como movimentações laterais em redes internas, uso de credenciais comprometidas, ataques originados de dentro da organização e falhas de configuração — vetores amplamente documentados em repositórios de inteligência [MITRE ATT&CK 2024, de Oliveira et al. 2023].

Para superar as limitações do modelo tradicional baseado em perímetro, é preciso adotar mecanismos mais refinados de controle, como a microssegmentação. Essa técnica isola aplicações, serviços e cargas de trabalho em domínios lógicos distintos, permitindo a aplicação de políticas específicas para cada fluxo de comunicação. O controle do tráfego interno, também chamado de tráfego "*Leste-Oeste*", torna-se assim mais preciso e contextual, reduzindo drasticamente a superfície de ataque e limitando o impacto de comprometimento a abrangencia local. A microssegmentação permite:

- **Isolamento efetivo:** Segmenta aplicações, cargas de trabalho e ambientes em nível granular: por serviço, função, equipe ou sensibilidade dos dados. Isso viabiliza a aplicação de políticas de segurança específicas e contextuais. Dessa forma, mesmo que uma parte da infraestrutura seja comprometida, o impacto permanece restrito aquele segmento específico;

- **Redução da superfície de ataque:** Ao limitar a comunicação entre componentes apenas ao que é estritamente necessário, a microssegmentação restringe significativamente os caminhos disponíveis para atacantes realizarem movimentações laterais. Isso dificulta a propagação de ameaças internas e externas, reduzindo a probabilidade de exploração de vulnerabilidades em cadeia. Recursos expostos ao público podem ser rigidamente isolados de sistemas sensíveis, como bancos de dados ou backends;

- **Proteção robusta:** A separação lógica e dinâmica de ambientes permite reforçar a segurança de contêineres, máquinas virtuais, dispositivos e microsserviços. Isso impede que falhas ou acessos indevidos em um componente comprometam o restante da infraestrutura. Essa abordagem também facilita a implementação de controles de acesso baseados em identidade e contexto, além de permitir respostas mais ágeis e precisas a incidentes.

Portanto, a transição da segurança perimetral para modelos baseados em *Zero Trust* exige não apenas uma reformulação da arquitetura de TI, mas também uma mudança cultural e técnica mais profundas. Nesse cenário, a automação da infraestrutura por meio da IaC torna-se um habilitador estratégico, ao permitir que controles de segurança, políticas de acesso e práticas de conformidade sejam aplicados de forma programática, repetível e versionável. Essa abordagem facilita a gestão, aumenta a resiliência operacional

e proporciona agilidade para responder a ameaças em tempo real, revando a resposta a incidentes de segurança à velocidade compatível com à complexidade dos ambientes modernos.

1.4.1. ZTA aplicada à segurança com IaC

O modelo de ZTA surgiu como resposta às limitações dos paradigmas tradicionais de segurança baseados em perímetro, nos quais se assume que todos os elementos dentro da rede corporativa são confiáveis por padrão. Essa suposição tem se tornado cada vez mais inadequada diante da crescente sofisticação dos vetores de ataque, da adoção massiva de serviços em nuvem e da mobilidade organizacional [Viegas et al. 2020]. Conforme estabelecido pelo NIST na publicação SP 800-207[30], a filosofia *Zero Trust* sustenta que nenhuma entidade, seja interna ou externa, deve ser automaticamente confiável [Rose et al. 2020].

A implementação do modelo ZTA requer políticas de acesso restritas, fundamentadas em identidade, contexto e avaliações dinâmicas de risco, com validações contínuas para cada solicitação de acesso. Entre os elementos centrais estão a autenticação multifator, a microsegmentação da rede, o monitoramento constante de tráfego e o uso de fontes confiáveis para verificação de identidade e autorização, promovendo uma postura de segurança adaptativa e centrada na minimização da confiança implícita.

No contexto da IaC, a adoção do modelo *Zero Trust* é estratégica ao possibilitar que princípios como mínimo privilégio, verificação contínua e controle automatizado sejam incorporados desde a comcepção da infraestrutura. Utilizando arquivos versionáveis, é possível declarar e aplicar políticas de segurança durante o provisionamento, assegurando que os recursos estejam adequadamente protegidos antes mesmo da infraestrutura entrarem em operação [Syed et al. 2022]. A convergência entre ZTA e IaC facilita a adoção de práticas como microsegmentação e isolamento granular de componentes, reduzindo significativamente o risco de movimentações laterais em cenários de comprometimento. Ferramentas como o *Open Policy Agent* (OPA) podem ser integradas aos *pipelines* de CI/CD para validar automaticamente as políticas declarativas, impedindo a promoção de configurações inseguras aos ambientes produtivos.

Além disso, soluções como o Twingate[31] e o Zscaler[32], baseadas no modelo de *Zero Trust Network Access* (ZTNA), reforçam essa abordagem ao viabilizar conexões seguras com base na identidade do usuário, no contexto da solicitação e no estado do dispositivo. A integração dessas ferramentas com práticas da IaC permite a construção de arquiteturas dinâmicas, versionáveis (com suporte a auditoria de mudanças) e alinhadas aos princípios de segurança *by design*. A IaC exerce papel central na concretização dessa estratégia ao descrever, de forma declarativa, o estado desejado dos ambientes de TI — seja em nuvens computacionais públicas, privadas (*on-premise*), híbridas ou em infraestrutura local. Essa representação programável permite a aplicação sistemática de controles de segurança, viabilizando práticas como o versionamento de políticas de acesso, controle automatizado de mudanças e rastreabilidade completa das modificações realizadas.

Com a sobreposição de camadas de segurança, como autenticação multifator (MFA),

[30] https://csrc.nist.gov/pubs/sp/800/207/final
[31] https://www.twingate.com/docs/
[32] https://www.zscaler.com/br

Tabela 1.2. Comparação entre Segurança Perimetral e *Zero Trust Architecture*

Característica	Segurança Perimetral	Zero Trust Architecture
Modelo de Confiança	Confiança baseada na localização (dentro da rede igual a confiável)	Nunca confie, sempre verifique; nenhuma localização é automaticamente confiável
Autenticação e Autorização	Autenticação no ponto de entrada; acesso geralmente irrestrito após entrada	Autenticação contínua e contextual, com autorização mínima (mínimo privilégio)
Proteção de Recursos	Protege a rede como um todo, com foco no perímetro	Protege cada recurso individualmente, baseado em identidade, contexto e políticas
Comportamento em Nuvem computacional e Mobilidade	Ineficaz para nuvem, dispositivos móveis e trabalho remoto	Projetado para ambientes distribuídos, nuvem, dispositivos móveis e acesso remoto
Visibilidade e Monitoramento	Monitoramento focado no perímetro, com pouca visibilidade interna	Monitoramento contínuo e granular de usuários, dispositivos e comportamento
Controle de Privilégios	Acesso amplo após autenticação inicial	Mínimo privilégio e segmentação de acesso

controle de sessão, gestão de identidade e análise comportamental, a arquitetura passa a operar de forma sinérgica e com controle contínuo. Essa abordagem fortalece mecanismos de autenticação e autorização, ao mesmo tempo em que promove conformidade regulatória e mitiga riscos com maior eficácia. Nesse contexto, a automação não elimina a necessidade de verificação, ao contrário, amplia a segurança ao tornar políticas de mudanças auditáveis, reproduzíveis e integradas ao ciclo de vida da infraestrutura de TI.

Para facilitar a compreensão das diferenças entre os modelos de segurança discutidos, segurança perimetral tradicional e ZTA, a Tabela 1.2 apresenta uma síntese comparativa dos principais conceitos, objetivos, benefícios, riscos e boas práticas. Essa visualização evidencia como o modelo *Zero Trust*, quando aliado às práticas da IaC, representa uma evolução significativa frente às limitações do paradigma tradicional baseado em perímetro, promovendo uma estratégia de defesa mais robusta, dinâmica e alinhada aos desafios de lidar com a complexidade da cibersegurança em ambientes programáveis.

1.4.2. Considerações finais da parte teórica

A revisão teórica apresentada nesta seção estabeleceu os fundamentos para compreender a abordagem da IaC e sua importância no cenário atual da cibersegurança. Inicialmente, foi analisado como a IaC promove uma mudança significativa nas práticas de provisionamento e gerenciamento de infraestrutura, ao substituir processos manuais por mecanismos automatizados e versionáveis, em conformidade com os princípios do DevSecOps.

Na sequência, a análise contrapôs essa abordagem aos modelos tradicionais de segurança baseados em perímetro. Embora esses modelos tenham sido eficazes em contextos centralizados, mostram-se insuficientes diante da crescente complexidade, descentralização e dinamicidade dos ambientes de computação em nuvem e de infraestruturas distribuídas. As arquiteturas *Zero Trust* destacam-se como uma abordagem robusta, pautada no

princípio de "confiança zero". Ou seja, em vez de presumir confiança, o foco recai na microsegmentação, na validação contínua de identidade e na aplicação de políticas de segurança orientadas pelo contexto.

As seções também ressaltaram a importância de boas práticas na gestão de segredos, na automação segura e na validação contínua da configuração, reconhecendo esses componentes como necessários para assegurar a integridade e a confiabilidade da infraestrutura definida como código. Monstrou-se que a integração entre IaC e *Zero Trust* favorecem as necessidades atuais de segurança viabilizando a construção de ambientes computacionais mais resilientes, versionáveis e alinhados aos requisitos de *Security By Design*.

Essa base conceitual sustenta a transição para a próxima seção, que apresentará a parte prática por meio do ambiente de experimentação desenvolvido para esse minicurso, onde serão detalhados os elementos estruturantes da arquitetura proposta, bem como os mecanismos e ferramentas selecionados para sua implementação em um cenário voltado à segurança e à gestão da infraestrutura.

1.5. Ambiente de experimentação

Esta seção apresenta a experimentação com dois objetivos complementares: (1) traduzir em código os princípios teóricos da IaC e *Zero Trust* discutidos nas seções anteriores e (2) fornecer aos participantes um roteiro reproduzível que possa ser adaptado a cenários reais. Ao concluir a experimentação o participante deverá ser capaz de provisionar, segmentar e monitorar uma infraestrutura mínima, entendendo a diferença entre o modelo perimetral e ZTA no contexto da IaC.

1.5.1. Arquitetura de referência

Para cada fase da experimentação, será apresentada uma arquitetura de referência, que incluirá alguns serviços de rede típicos, selecionados por sua relevância e ampla adoção no mercado de TI.

Essa arquitetura base será mantida para as demais etapas para facilitar o entendimento e proporcionar ao participante a continuidade e aprofundamento nas práticas propostas. A figura 1.4 apresenta uma visão geral da arquitetura base.

Para tanto, os componentes escolhidos para as etapas de experimentação são:

- **GLPI (porta 8080)**: sistema de gestão de chamados e ativos.
- **MariaDB**: *backend* relacional para persistência do GLPI.
- **Servidor de Backup**: ambiente leve (utilizando uma distribuição Linux Alpine) para simulações de rotina de salvamento de dados via rede.
- **Servidor de Automação**: execução de rotinas com Ansible e Checkov para aplicação de configurações seguras.
- **Twingate Connector**: agente de acesso remoto seguro que evita exposição de portas e restringe acessos com base em identidade e postura de dispositivo.

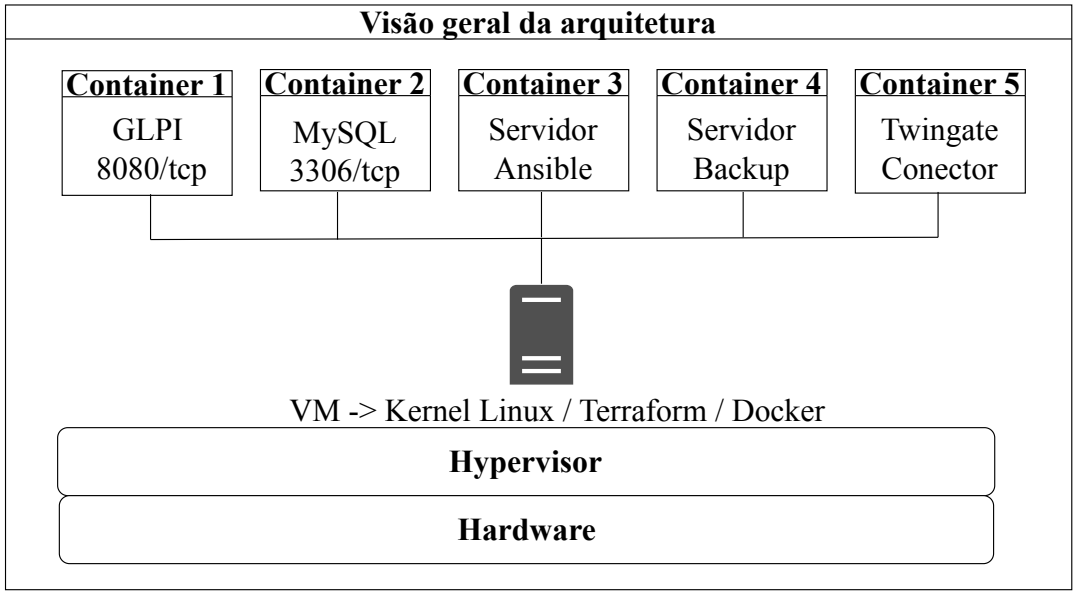


Figura 1.4. Visão geral da arquitetura do ambiente de experimentação

Para apoiar essa arquitetura, será utilizada a tecnologia de *containers*, permitindo isolamento e portabilidade entre as etapas do laboratório. Nesse contexto, destacam-se duas ferramentas fundamentais:

- Dockerfile: é um arquivo declarativo que descreve, passo a passo, como construir uma imagem Docker customizada. Nele são definidas as dependências, o ambiente de execução e os *scripts* necessários para inicializar os serviços.
- Docker Compose: é uma ferramenta que permite a orquestração de múltiplos *containers*. Utilizando um arquivo YAML, define como os serviços interagem, suas redes, volumes e demais parâmetros operacionais, simplificando a execução e manutenção de aplicações distribuídas.

Essas ferramentas serão exploradas na prática a partir da Etapa 1 da experimentação. A figura 1.5 ilustra a diferença conceitual entre Dockerfile e Docker Compose.

1.5.2. Descrição de componentes da arquitetura

- **Perímetro Virtual**: Refere-se a um agrupamento lógico de ativos críticos, como servidores de aplicação, bancos de dados, sistemas de automação (e.g., Ansible) e servidores de backup. Diferente do modelo tradicional baseado em perímetro físico, essa abordagem adota um perímetro definido por *software*, protegido por políticas de acesso baseadas em identidade, contexto e critérios explícitos de autorização.
- **Terraform/Ansible (IaC)**: O Terraform é utilizado para provisionar a infraestrutura de forma declarativa e automatizada, incluindo a criação dos *Connectors* do Twingate, assegurando rastreabilidade e padronização. De forma complementar, o Ansible atua na configuração pós-provisionamento, executando rotinas de automação

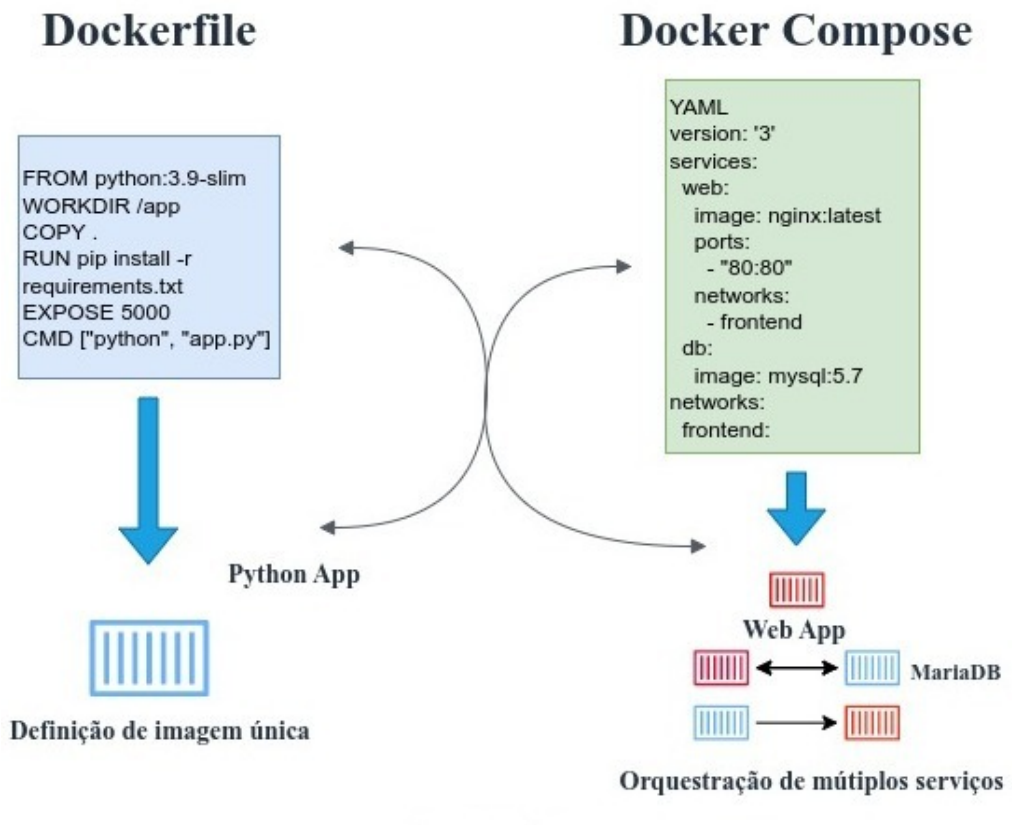


Figura 1.5. Diferença entre Dockerfile e Docker-compose

voltadas à aplicação de políticas de segurança, instalação de serviços e ajuste de parâmetros do sistema. Essa integração entre as ferramentas assegura a consistência, a conformidade e a eficiência na entrega dos ativos de infraestrutura.

- **Twingate**: Plataforma que substitui VPNs tradicionais ao oferecer acesso seguro, invisível e segmentado. Seus principais componentes são:
  - *Twingate Client*: Instalado no dispositivo do usuário, estabelece túneis criptografados para os recursos autorizados.
  - *Twingate Connector*: Componente leve implantado no Perímetro Virtual. Realiza conexões de saída para o plano de controle, sem exigir portas abertas no firewall.
  - *Twingate Control Plane*: Núcleo orquestrador disponível em nuvem, responsável por autenticação, verificação da postura dos dispositivos e aplicação de políticas, sem interferir no tráfego direto.

1.5.3. Fluxo Operacional da Arquitetura

A seguir é apresentada a sequencia de passos (*walkthrough*) para desenvolvimento da infraestrutura como código da experimentação.

1. **Definição da Infraestrutura como Código**: O administrador descreve a infraestrutura e as políticas de acesso via arquivos Terraform, especificando recursos e

conexões seguras.

2. **Provisionamento Automatizado**: Através do Terraform, os recursos são provisionados em nuvem ou *on-premises*. Pelo Ansible os *Connectors* do Twingate são implantados e configurados automaticamente.

3. **Solicitação de Acesso**: O usuário utiliza o Twingate Client para iniciar o acesso a um recurso específico.

4. **Autenticação e Autorização**: O Client comunica-se com o Control Plane, que realiza autenticação com provedores de identidade, como o Google Workspace ou Azure AD, e verifica a conformidade do dispositivo.

5. **Estabelecimento do Túnel Seguro**: Após autorização, um túnel *peer-to-peer* criptografado é estabelecido entre o *Client* e o *Connector*, garantindo privacidade e baixa latência.

6. **Acesso Específico ao Recurso**: O acesso é concedido apenas ao recurso autorizado, impedindo a visualização ou o tráfego para outros componentes da rede.

A arquitetura implementada no ambiente de experimentação combina os princípios do *Zero Trust Network Access* (ZTNA) e da *Infrastructure as Code* (IaC) com o objetivo de estabelecer um ambiente seguro, resiliente e automatizado, assegurando a proteção granular de recursos, credenciais e fluxos operacionais. Essa arquitetura é composta por dois componentes principais, que atuam de forma integrada para mitigar riscos e reduzir a superfície de exposição.

A Figura 1.6 ilustra uma arquitetura de segurança baseada no modelo de *Zero Trust* (Confiança Zero), cujo objetivo é proteger o acesso a recursos computacionais sensíveis por meio de segmentação e verificação contínua de identidade. A abordagem apresentada combina práticas de *Infrastructure as Code*, utilizando o Terraform para o provisionamento automatizado da infraestrutura, com a solução de acesso remoto seguro do Twingate, uma plataforma de *Zero Trust Network Access* (ZTNA).

O princípio central do modelo *Zero Trust* é sintetizado no princípio *"nunca confie, sempre verifique"*. Nenhum usuário ou dispositivo é confiável por padrão, independentemente de sua localização, sendo cada solicitação de acesso avaliada com base em identidade, contexto e postura de segurança.

1.5.4. Benefícios da Arquitetura

A seguir são relacionados os principais benefícios do cenário de experimentação.

- **Segurança granular**: Elimina-se a confiança implícita, com acesso fundamentado em identidade, contexto e segurança do dispositivo.

- **Superfície de ataque reduzida**: O uso exclusivo de conexões de saída e ausência de portas abertas limita significativamente as possibilidades de exploração externa.

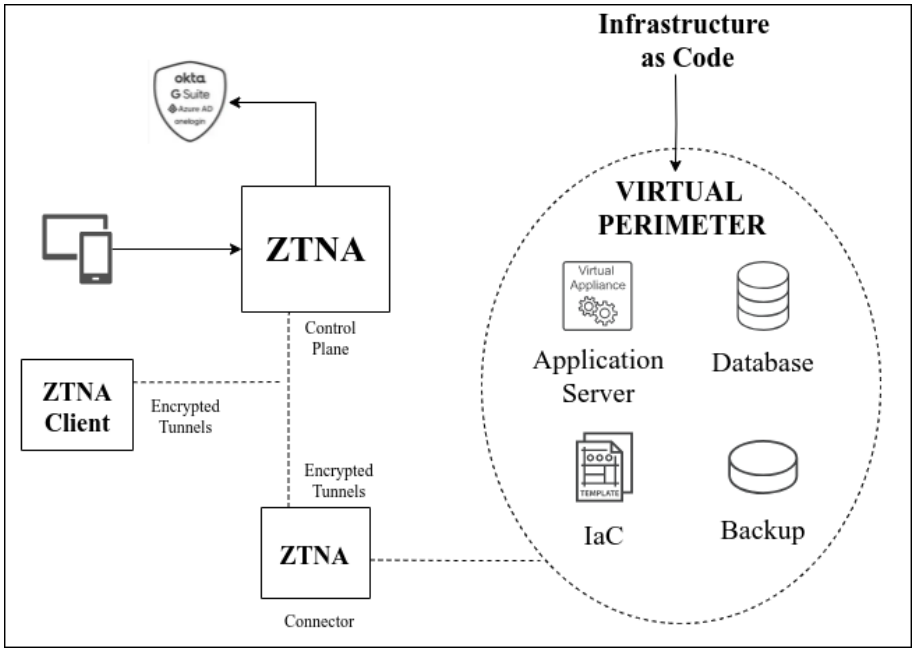


Figura 1.6. Visão geral da arquitetura do ambiente de experimentação.

- **Automação e gestão**: A configuração como código permite versionamento, auditoria de mudanças e replicação de ambientes com consistência e segurança.
- **Experiência transparente ao usuário**: O acesso é contínuo e automatizado, dispensando o uso de VPNs manuais.

1.5.5. Etapas da experimentação

Esse cenário laboratorial propõe a construção de um ambiente seguro e controlado por meio de quatro etapas evolutivas: provisionamento manual com Docker Compose, automação com Terraform, aplicação de políticas de segurança utilizando Ansible, Checkov e Twingate, e por fim, a fase de testes e validação da infraestrutura.

O objetivo é demonstrar como o modelo da IaC, aliado ao paradigma de *Zero Trust*, pode elevar substancialmente o nível de segurança e gestão em ambientes computacionais modernos.

Ao longo das atividades práticas, os participantes percorrem uma trajetória que vai desde o provisionamento tradicional com Docker Compose até a implementação de uma arquitetura automatizada e segura, baseada em segmentação lógica, validação contínua e controle de acesso rigoroso, incorporando os princípios da IaC de forma aplicada.

Etapa 1 - Provisionamento com Docker Compose

Nesta etapa introdutória, o ambiente da aplicação é configurado manualmente utilizando o Docker Compose. O objetivo é familiarizar os participantes com os componentes que serão posteriormente automatizados, fornecendo uma base prática para a compreensão da infraestrutura.

As instruções completas para a realização dessa etapa estão disponíveis no re-

positório do minicurso no GitLab[33], onde são apresentados, de forma estruturada, os procedimentos necessários para a inicialização dos *containers* e preparação do ambiente.

Durante essa fase, os participantes são orientados a analisar o conteúdo do arquivo Dockerfile, responsável pela definição e construção das imagens utilizadas no projeto, bem como a estrutura do arquivo `docker-compose.yml`, que organiza os serviços, redes e volumes empregados na composição do ambiente.

Além disso, são realizados procedimentos de inspeção e análise do ambiente por meio de comandos Docker, permitindo observar *logs*, configurações e o comportamento dos *containers* em execução.

Esta etapa tem como finalidade apresentar os fundamentos práticos da infraestrutura da aplicação e estabelecer uma base comparativa para as atividades de automação que serão desenvolvidas nas etapas seguintes.

Etapa 2 - Provisionamento automatizado com Terraform (IaC)

Nesta fase do minicurso, busca-se aplicar os princípios da *Infrastructure as Code* (IaC) por meio da ferramenta Terraform, com o objetivo de automatizar o provisionamento dos mesmos recursos anteriormente configurados manualmente.

Essa comparação permite evidenciar os benefícios operacionais em termos de rastreabilidade, padronização e reprodutibilidade.

Na sequência, introduzem-se os fundamentos da infraestrutura como código e prepara o ambiente para configurações adicionais nas etapas seguintes do minicurso.

As instruções detalhadas para a execução estão disponíveis no repositório do minicurso no GitLab[33], abrangendo desde a estruturação dos arquivos até a execução do provisionamento automatizado.

Etapa 3 - Aplicação de segurança com Ansible (IaC) e Checkov

Na terceira etapa do minicurso, são aplicadas práticas de automação e validação da infraestrutura provisionada por meio das ferramentas Ansible e Checkov, com foco em reforço da segurança e conformidade. Esta fase complementa o provisionamento realizado anteriormente, agregando camadas de configuração, inspeção e boas práticas de segurança.

Entre os tópicos abordados, destacam-se a análise dos arquivos de configuração (.tf), a configuração do `SSH` e chave pública nos *containers* e a comparação entre as abordagens manual (Docker Compose) e automatizada (Terraform).

Dentre as ações previstas nesta etapa, destacam-se:

- Aplicação de rotinas de configuração automatizada em serviços como o GLPI, utilizando *playbooks* do Ansible;
- Implementação de práticas de segurança, como a habilitação de HTTPS, a configuração de certificados digitais e ajustes nos serviços para reduzir vulnerabilidades;
- Utilização da ferramenta Checkov para validar os arquivos de infraestrutura declarativa, identificando potenciais falhas de conformidade antes da aplicação em

[33] https://projects.ppgia.pucpr.br/secplab/sbseg25-IaC-ZT

ambientes produtivos.

A integração entre Ansible e Checkov exemplifica uma abordagem DevSecOps no contexto da *Infrastructure as Code*, permitindo que políticas de segurança e requisitos operacionais sejam aplicados de forma automatizada e auditável. Esta etapa fortalece a confiabilidade da infraestrutura e prepara o ambiente para integração com mecanismos de controle de acesso e microssegmentação.

As instruções detalhadas para execução estão disponíveis no repositório do minicurso no GitLab[33], com roteiros organizados para facilitar a aplicação das tarefas automatizadas.

Etapa 4 - Diagnóstico e validação do ambiente provisionado

Na etapa final do minicurso, são realizados testes de conectividade, diagnóstico de rede e análise de exposição de serviços, para validar os mecanismos de segmentação e controle de acesso implementados ao longo do ambiente provisionado.

As instruções detalhadas para a execução dessa etapa encontram-se disponíveis no repositório do minicurso no GitLab[33], incluindo os comandos e procedimentos recomendados para simulação de acesso e varredura.

Durante esta fase, os participantes devem realizar operações básicas de diagnóstico, utilizando comandos como `ping, curl, traceroute, dig`, entre outros, com o propósito de verificar a conectividade entre os serviços e identificar possíveis falhas de comunicação.

Além disso, são conduzidos testes de acesso seguro, simulando tentativas de conexão a serviços protegidos pelo mecanismo de *Zero Trust Network Access* (ZTNA), implementado com a solução Twingate. O objetivo é observar as restrições de visibilidade impostas, validando se a política de microssegmentação está sendo aplicada de forma adequada.

Como parte da simulação de ameaças, é utilizada a ferramenta `nmap` para realizar varreduras de portas e serviços, alinhando a atividade à tática TA0007 – Discovery do *framework* MITRE ATT&CK. Essa prática permite avaliar a eficácia das medidas de proteção frente a tentativas de mapeamento da infraestrutura por agentes não autorizados.

Por fim, com base nas respostas obtidas e na visibilidade dos serviços, os participantes devem refletir sobre o grau de exposição da infraestrutura e a efetividade das medidas de segurança aplicadas ao longo do ciclo de provisionamento.

Essas etapas estão resumidas na Tabela 1.3 com as ferramentas utilizadas, atividades e abordagem de segurança aplicada.

1.5.6. Considerações Finais

A integração entre o modelo *Zero Trust* e práticas da IaC representa uma evolução na proteção de ambientes computacionais modernos. A combinação do Terraform e Ansible com o Twingate viabiliza a implementação prática de políticas de acesso segmentado e com alterações auditável, promovendo escalabilidade, segurança e conformidade regulatória. Trata-se de uma abordagem estratégica para organizações que buscam reduzir riscos,

Tabela 1.3. Resumo das Etapas do Laboratório Guiado com IaC e *Zero Trust*

Etapa	Ferramentas	Atividades Principais	Abordagem de Segurança
1	Docker Compose	Provisionamento manual dos *containers* GLPI, MariaDB, Backup e Ansible.	Sem controle de acesso ou segmentação; ambiente exposto.
2	Terraform (IaC)	Reprovisionamento automatizado dos recursos da Etapa 1.	Provisionamento automatizado com rastreabilidade e início da proteção declarada.
3	Checkov + Terraform + Ansible	Análise estática com Checkov, Reprovisionamento com Terraform, Aplicação de configuração HTTPS (GLPI), e inicialização do serviço Twingate.	Validação pré-provisionamento e aplicação de políticas de segurança automatizadas.
4	Ansible + Twingate	Testes de conectividade segura via Twingate. Validação de segmentação, visibilidade mínima e política de acesso.	Acesso remoto seguro com base em identidade e postura do dispositivo (modelo *Zero Trust*).

controlar acessos com precisão e manter a resiliência operacional em ambientes híbridos ou distribuídos.

1.6. Laboratório guiado: Provisionamento com Docker Compose, Terraform, Ansible e *Zero Trust*

Esta seção apresenta um cenário de laboratório dividido em quatro etapas evolutivas, que podem ser realizadas de forma autônoma pelos participantes.

O laboratório tem como objetivo mostrar, na prática, a transição de um ambiente manual para uma infraestrutura automatizada, segura e baseada no modelo *Zero Trust*.

Para a execução das atividades, todos os arquivos, *scripts* e instruções detalhadas estão disponíveis no repositório GitLab oficial do minicurso. Cada etapa está alinhada aos fundamentos discutidos na seção 1.5.5, promovendo o reforço conceitual por meio da prática aplicada.

Etapa 1 – Provisionamento Manual com Docker Compose

Nesta etapa inicial, os serviços GLPI, MariaDB, servidor de *backup* e servidor de automação são configurados de forma manual, utilizando arquivos declarativos do Docker Compose. A atividade permite compreender a estrutura básica do ambiente, suas dependências e a organização da rede interna entre os *containers*. Essa abordagem serve como base para comparações posteriores com processos automatizados.

Etapa 2 – Provisionamento Automatizado com Terraform (IaC)

Com o uso do Terraform, os mesmos recursos da etapa anterior são recriados mas de forma automatizada, adotando o modelo declarativo da IaC. A automação introduz benefícios como rastreabilidade, versionamento e controle de estado.

Etapa 3 – Configuração com Ansible e Validação com Checkov

Após o provisionamento, são aplicadas rotinas de configuração com Ansible, que incluem práticas de melhoria de segurança, ajustes de HTTPS, instalação e ativação do

Twingate. Também é utilizada a ferramenta Checkov para validação da infraestrutura declarada, com foco na conformidade e prevenção de falhas antes da aplicação final. Essa etapa consolida a segurança do ambiente antes de seu uso.

Etapa 4 – Integração *Zero Trust* e Diagnóstico de Rede

Na etapa final, são realizados testes de conectividade e diagnóstico para validar o isolamento e as restrições de acesso aplicadas com o Twingate. Os participantes simulam cenários de descoberta de rede e varredura de serviços, observando a aplicação das políticas de segmentação e visibilidade mínima. A atividade reforça o entendimento sobre os efeitos práticos da arquitetura *Zero Trust* no controle de acessos e na redução da superfície de ataque.

Este ensaio de laboratório propõe a construção de um ambiente seguro e controlado por meio de quatro etapas evolutivas: provisionamento manual com Docker, automação com Terraform e aplicação de políticas de segurança com Ansible, Checkov e Twingate. O objetivo é demonstrar como o modelo de *Infrastructure as Code* (IaC), quando aliado ao paradigma de Confiança Zero (*Zero Trust*), pode elevar significativamente o nível de segurança e governança em ambientes computacionais modernos.

Durante a experimentação os participantes transitam de um provisionamento tradicional (com Docker Compose) até uma arquitetura segura e automatizada, baseada em segmentação, validação contínua e controle restrito de acesso, utilizando IaC.

Ao final da experimentação, espera-se que os participantes:

- Compreendam o ciclo completo de provisionamento e proteção de infraestrutura, desde a criação de ambientes até a aplicação de políticas de segurança automatizadas;
- Reflitam sobre os benefícios da IaC na redução de falhas humanas, na padronização de ambientes e na capacidade de rastrear mudanças;
- Sejam capazes de integrar práticas modernas de segurança, análise estática de infraestrutura, e acesso remoto seguro via Twingate;
- Visualizem, por meio da aplicação prática, o uso de *Zero Trust* em ambientes controlados e reprodutíveis, mitigando riscos e fortalecendo a postura de segurança;
- Pèrcebam na experimentação práticas reais de proteção, como a execução de varreduras de rede com ferramentas como o `nmap`, relacionadas à tática *Discovery* (TA0007) do MITRE ATT&CK, ampliando a compreensão sobre possíveis vetores de ataque e medidas preventivas.
- Dessa forma, o minicurso proporciona não apenas uma abordagem conceitual, mas também uma vivência prática sobre a construição de ambientes mais resilientes, versionáveis e alinhados com os desafios atuais da cibersegurança.

As quatro etapas descritas nesta seção estruturam o percurso prático de transição entre um ambiente tradicional e uma arquitetura segura com aplicação dos princípios de

Zero Trust. A descrição fornecida tem caráter conceitual, destacando os papéis e objetivos de cada ferramenta.

Todos os recursos necessários, incluindo *scripts*, arquivos de configuração e orientações técnicas, estão organizados no repositório oficial do minicurso, cuja descrição detalhada pode ser consultada na [subsec:repositorio]Subseção 1.6.4 (Estrutura de Arquivos e Repositório GitLab).

1.6.1. Objetivos de aprendizagem

a) Visualizar a diferença entre provisionamento manual e declarativo;

b) Avaliar ferramentas distintas para construir imagens de *containers* e provisionamento de infraestrutura como Dockerfile, Docker compose e Terraform;

c) Compreender noções de funcionamento e aplicação das ferramentas da IaC (Terraform e Ansible);

d) Compreender como práticas de segurança podem ser automatizadas com IaC;

e) Aplicar *Zero Trust* para restringir acessos, reduzir exposição e proteger os ativos digitais;

f) Avaliar os benefícios de políticas de segmentação e validação de dispositivos em ambientes reais.

1.6.2. Cenário de Laboratório

A execução do cenário de laboratório pode ser realizada de forma autônoma, mesmo na ausência de tutor. Cada uma das quatro etapas descritas na seção anterior está detalhadamente documentada no repositório GitLab do minicurso, incluindo os arquivos de configuração, *scripts* e orientações de execução.

Para cada etapa, recomenda-se que o participante siga a sequência de instruções disponível nos respectivos arquivos README.md, presentes em cada diretório do repositório. Esses arquivos contêm explicações sobre o objetivo da etapa, dependências envolvidas e passos operacionais necessários. A estrutura modular adotada permite que o cenário seja reproduzido com flexibilidade em diferentes ambientes, promovendo autonomia no aprendizado.

1.6.3. Requisitos

Antes de iniciar o laboratório, é essencial que o ambiente do participante esteja preparado com as ferramentas necessárias para a execução das etapas propostas. São requeridos: Docker (versão 20.10 ou superior), Docker Compose, Terraform (versão 1.0 ou superior), Ansible, Checkov e Git (utilizado para clonar o repositório do projeto). Esses componentes são fundamentais para viabilizar a construção progressiva da infraestrutura, desde o provisionamento inicial até a aplicação automatizada de políticas de segurança.

O sistema operacional recomendado é Linux, preferencialmente com suporte a *containers* (como Ubuntu ou Debian). Caso o participante não disponha de um ambiente

Tabela 1.4. Estrutura de diretórios do repositório GitLab

Diretório	Conteúdo Principal
`/etapa1/`	Arquivos do Docker Compose e `Dockerfile` para provisionamento manual
`/etapa2/`	Arquivos `.tf` para provisionamento automatizado com Terraform
`/etapa3/`	*Playbooks* do Ansible, arquivos de configuração HTTPS e validação com Checkov
`/etapa4/`	*Scripts* de testes de conectividade e varredura, voltados à análise com ZTNA
`/docs/`	Documentação complementar com requisitos, instruções gerais e guias de instalação
`README.md`	Arquivo principal com introdução ao laboratório, instruções de uso e licenciamento

compatível, é possível utilizar uma máquina virtual com distribuição Linux ou *containers* pré-configurados.

Os *links* para instalação, versões recomendadas e instruções de configuração estão disponíveis no diretório docs/ambiente do repositório GitLab do minicurso.

1.6.4. Estrutura de Arquivos e Repositório GitLab

O repositório do minicurso no GitLab[34] foi organizado de forma modular, com diretórios separados para cada etapa do ensaio. Isso permite que os participantes identifiquem facilmente os arquivos e *scripts* relacionados a cada fase da experimentação.

A tabela 1.4 apresenta um resumo da estrutura principal do repositório. Essa estrutura foi projetada para facilitar a navegação, a execução progressiva das etapas e o reuso dos componentes em outros projetos de infraestrutura.

1.6.5. Segurança com *Zero Trust* no ambiente

A integração com o modelo *Zero Trust* não é apenas complementar, mas central para o ensaio de laboratório. Neste contexto, ele é implementado por meio do Twingate, que incorpora os seguintes princípios de segurança:

- **Microssegmentação de serviços**: o acesso é concedido por recurso, e não à rede inteira. Por exemplo, um usuário pode acessar apenas o GLPI, sem ter visibilidade dos servidores de backup ou banco de dados.
- **Eliminação da confiança implícita**: a localização na rede (interna ou externa) não é critério de confiança. Todo acesso é autenticado, autorizado e registrado.
- **Acesso baseado em identidade e contexto**: o Twingate Control Plane realiza autenticação integrada a diretórios (AD, Google Workspace etc.) e verifica postura de segurança dos dispositivos.
- **Redução da superfície de ataque**: os serviços não precisam expor portas TCP públicas nem estar acessíveis na internet. O conector Twingate inicia a conexão de forma reversa (saída), mantendo os ativos invisíveis para escaneamentos.

Essa abordagem reforça a confidencialidade, integridade e disponibilidade dos recursos, mesmo em ambientes com múltiplos usuários ou acessos remotos.

[34]Disponível em: https://projects.ppgia.pucpr.br/secplab/sbseg25-IaC-ZT

1.6.6. Integração Progressiva do *Zero Trust* no ensaio de laboratório

Ao longo das quatro etapas, os participantes acompanham a transição de um ambiente tradicional e com exposição a ataques para uma infraestrutura segura, automatizada e aderente a *Zero Trust*. O Twingate é gradualmente integrado como ferramenta de microssegmentação e controle de acesso, conforme descrito a seguir:

- **Etapa 1 (Docker Compose)**: os serviços GLPI, MariaDB, backup e Ansible são expostos diretamente à rede interna, sem qualquer controle de identidade ou segmentação. Trata-se de um ambiente vulnerável por padrão, configurado manualmente via `docker-compose.yml`.
- **Etapa 2 (Terraform)**: os mesmos serviços são provisionados de forma automatizada por meio do Terraform. Embora o ambiente ainda não esteja conectado ao Twingate, foram executados comandos que preparam a infraestrutura para a etapa seguinte.
- **Etapa 3 (Ansible + Checkov)**: nesta fase, são aplicadas configurações de melhoria de segurança no GLPI, ativação do protocolo HTTPS e validações automatizadas com o Checkov. Adicionalmente, o serviço do Twingate é instalado e iniciado, viabilizando a aplicação de políticas de acesso e garantindo que os ativos permaneçam invisíveis fora do escopo autorizado.
- **Etapa 4 (Ansible + Twingate)**: os testes finais são acessos aos serviços via Twingate, validando o funcionamento da segmentação e a aplicação das políticas configuradas. O acesso é autenticado, segmentado e limitado com base no princípio da confiança mínima.

O *Zero Trust* no ambiente de experimentação, atua como o eixo transversal de cibersegurança, oferecendo autenticação baseada em identidade, controle de postura de dispositivo, política de acesso mínimo e eliminação da confiança implícita.

1.6.7. Política de Acesso (Exemplo *Zero Trust*)

A política demonstrada no Código 1.4 exemplifica como uma abordagem *Zero Trust* define explicitamente os recursos, os perfis permitidos, a postura esperada dos dispositivos e os requisitos de autenticação adicional (como MFA e geolocalização).

1.6.8. Síntese: Benefícios da Integração IaC + *Zero Trust*

A abordagem integrada neste laboratório promove:

- **Gestão de configuração**: com Terraform e Ansible, toda a infraestrutura e suas regras são codificadas e versionáveis.
- **Segurança na origem (Shift-Left)**: com Checkov, falhas conhecidas são detectadas antes da execução da infraestrutura.
- **Resiliência e visibilidade reduzida**: com Twingate, o ambiente se torna invisível a agentes externos e acessível apenas mediante autenticação.

Código Fonte 1.4. Política Zero Trust simulada para acesso ao banco de dados.

```
policy_name: "db-prod-access"
description: "Controla o acesso ao banco de dados principal."

resources:
  - "database.prod.internal"

principals:
  - group: "database_admins"
  - user: "service_account_backup@example.com"

conditions:
  - device_posture: "compliant"
  - mfa: "required"
  - location: "BR"

action: "allow"
```

- **Eficiência operacional**: a reimplantação de ambientes seguros pode ser feita em minutos com um único comando.

Essa integração reflete as tendências mais atuais em cibersegurança, possibilitando que mesmo equipes pequenas apliquem práticas de elevado nível na proteção de seus ambientes computacionais.

1.7. Considerações sobre a experimentação

Esta seção final retoma os principais conceitos abordados ao longo do minicurso, com ênfase na integração entre *Infrastructure as Code* (IaC) e o modelo de segurança *Zero Trust*. A proposta foi conduzir os participantes por uma trilha estruturada, da fundamentação teórica à aplicação prática, promovendo uma visão crítica e aplicada aos desafios atuais da segurança em ambientes automatizados.

1.7.1. Integração entre IaC e *Zero Trust*: Conceitos Fundamentais

A combinação entre IaC e *Zero Trust* representa uma evolução em relação aos modelos tradicionais de segurança baseados em perímetro. Essa integração oferece maior resiliência operacional, adaptabilidade e proteção dinâmica, com base em três princípios:

- **Confiança mínima**: nenhum recurso ou identidade deve ser implicitamente confiável;
- **Validação contínua**: autenticações e autorizações devem ocorrer a cada nova interação;
- **Visibilidade segmentada**: controle granular sobre acessos, horários e escopos de interação.

Esses princípios foram operacionalizados na arquitetura experimental proposta ao longo das atividades práticas.

1.7.2. Automação, Versionamento e Conformidade Adaptativa

A utilização de arquivos declarativos via IaC viabiliza o provisionamento automatizado, auditável e consistente da infraestrutura. Essa abordagem reduz a incidência de erros manuais e facilita o versionamento, contribuindo para uma recuperação rápida em cenários de falha [Wang 2022].

Quando combinada ao modelo *Zero Trust Network Access* (ZTNA), essa estratégia potencializa:

- A conformidade dinâmica com políticas de segurança em constante evolução;
- A resposta proativa a ameaças emergentes;
- A reprodutibilidade de ambientes com segurança incorporada desde a concepção.

1.7.3. Microssegmentação e Redução da Superfície de Ataque

A microssegmentação, pilar estruturante do *Zero Trust*, foi aplicada para isolar recursos críticos como o GLPI e o banco de dados MariaDB, limitando os fluxos de rede a conexões estritamente autorizadas. Essa estratégia inibe movimentações laterais em caso de comprometimento inicial.

A autenticação contextual, baseada na análise contínua de identidade, localização e permissões, contribui para reduzir riscos de reutilização de credenciais e escalonamento indevido de privilégios.

1.7.4. Competências Desenvolvidas e Aplicação Prática

Ao longo do minicurso, os participantes foram desafiados a aplicar os conceitos em um ambiente realista, utilizando ferramentas modernas e práticas seguras de provisionamento. Espera-se que, ao final, tenham desenvolvido as seguintes competências:

- Provisionar ambientes seguros com ferramentas da IaC;
- Aplicar os princípios fundamentais de *Zero Trust*;
- Avaliar criticamente estratégias de segurança adotadas em infraestrutura;
- Identificar vulnerabilidades e propor melhorias com base na experimentação.

1.7.5. Conclusão

A experimentação conduzida reforça o papel da IaC e do modelo *Zero Trust* como fundamentos para a construção de ambientes seguros, versionáveis e alinhados às práticas contemporâneas de DevSecOps e gestão de cibersegurança.

A arquitetura utilizada mostrou, de forma prática, que é possível incorporar controles como verificação contínua, isolamento de recursos e autenticação contextual em

fluxos automatizados de infraestrutura. Com isso, foi evidenciada a viabilidade técnica e o potencial formativo dessa abordagem, promovendo sua aplicabilidade em contextos reais de operação e segurança organizacional.

Referências

[Abreu et al. 2020] Abreu, V., Santin, A. O., Viegas, E. K., and Cogo, V. V. (2020). *Identity and Access Management for IoT in Smart Grid*, page 1215–1226. Springer International Publishing.

[Abreu et al. 2017] Abreu, V., Santin, A. O., Viegas, E. K., and Stihler, M. (2017). A multi-domain role activation model. In *2017 IEEE International Conference on Communications (ICC)*, page 1–6. IEEE.

[Alonso et al. 2023] Alonso, J., Piliszek, R., and Cankar, M. (2023). Embracing iac through the devsecops philosophy: Concepts, challenges, and a reference framework. *IEEE Software*, 40(1):56–62.

[Basak et al. 2022] Basak, S. K., Neil, L., Reaves, B., and Williams, L. (2022). What are the practices for secret management in software artifacts? In *2022 IEEE Secure Development Conference (SecDev)*, page 69–76. IEEE.

[Boulden 2023] Boulden, S. (2023). Sigstore - signature sorcery. RXRW Blog. Accessed 2025-06-14.

[Center for Internet Security (CIS)] Center for Internet Security (CIS). Foundational cloud security with cis benchmarks. Blog Post.

[Chen et al. 2018] Chen, W., Wu, G., and Wei, J. (2018). An approach to identifying error patterns for infrastructure as code. In *2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, pages 124–129.

[Chiari et al. 2022] Chiari, M., Pascalis, M. D., and Pradella, M. (2022). Static analysis of infrastructure as code: a survey. *arXiv preprint arXiv:2206.10344*.

[CloudOptimo 2025] CloudOptimo (2025). Detecting orphaned resources using aws config rules. Acesso em: 14 jun. 2025.

[de Oliveira et al. 2023] de Oliveira, P. R., Santin, A. O., Horchulhack, P., Viegas, E. K., and de Matos, E. (2023). A dynamic network-based intrusion detection model for industrial control systems. In *2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, page 1496–1501. IEEE.

[Filho et al. 2025] Filho, A. G., Viegas, E. K., Santin, A. O., and Geremias, J. (2025). A dynamic network intrusion detection model for infrastructure as code deployed environments. *Journal of Network and Systems Management*, 33(4).

[Fowler 2013] Fowler, M. (2013). Infrastructure as code. Publicado originalmente no blog de Martin Fowler com contribuições de Kief Morris.

[Horchulhack et al. 2022a] Horchulhack, P., Viegas, E. K., and Santin, A. O. (2022a). Detection of service provider hardware over-commitment in container orchestration environments. In *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, page 6354–6359. IEEE.

[Horchulhack et al. 2022b] Horchulhack, P., Viegas, E. K., and Santin, A. O. (2022b). Detection of service provider hardware over-commitment in container orchestration environments. In *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, page 6354–6359. IEEE.

[Kalu et al. 2025] Kalu, K. G., Okorafor, S., Singla, T., Torres-Arias, S., and Davis, J. C. (2025). Why johnny signs with sigstore: Examining tooling as a factor in software signing adoption in the sigstore ecosystem. In *arXiv preprint arXiv:2503.00271*.

[Lorenc 2021] Lorenc, D. (2021). Cosign image signatures. *Sigstore Blog*. Accessed 2025-06-14.

[Mangera 2025] Mangera, V. (2025). Shedding light on orphaned cloud resources – ghosts haunting. LinkedIn. Acesso em: 14 jun. 2025.

[Marquis 2024] Marquis, Y. A. (2024). From theory to practice: Implementing effective role-based access control strategies to mitigate insider risks in diverse organizational contexts. *Journal of Engineering Research and Reports*, 26(5):138–154.

[Matthew Kosinski 2024] Matthew Kosinski, A. F. (2024). Identity and access management?

[Mell and Grance 2011] Mell, P. and Grance, T. (2011). The nist definition of cloud computing. Special Publication 800-145, National Institute of Standards and Technology (NIST).

[MITRE ATT&CK 2024] MITRE ATT&CK (2024). MITRE ATT&CK Framework. https://attack.mitre.org/. Acesso em: 12 set. 2024.

[NIST 2025] NIST (2025). How do i create a good password? Technical report, National Institute of Standards and Technology. Acesso em: 08 de maio de 2025.

[Oliveira et al. 2024] Oliveira, J., Santin, A., Viegas, E., and Horchulhack, P. (2024). A non-interactive one-time password-based method to enhance the vault security. In Barolli, L., editor, *Advanced Information Networking and Applications*, pages 201–213, Cham. Springer Nature Switzerland.

[Queen 2024] Queen, C. (2024). What is infrastructure as code security? Accessed: 2024-09-14.

[Rahman et al. 2019a] Rahman, A., Mahdavi-Hezaveh, R., and Williams, L. (2019a). A systematic mapping study of infrastructure as code research. *Information and Software Technology*, 108:65–77.

[Rahman et al. 2019b] Rahman, A., Parnin, C., and Williams, L. (2019b). The seven sins: Security smells in infrastructure as code scripts. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, pages 164–175.

[Rahman et al. 2021a] Rahman, A., Rahman, M. R., Parnin, C., and Williams, L. (2021a). Security smells in ansible and chef scripts: A replication study. *ACM Transactions on Software Engineering and Methodology*, 30(1):1–31.

[Rahman et al. 2021b] Rahman, A., Rahman, M. R., Parnin, C., and Williams, L. (2021b). Security smells in ansible and chef scripts: A replication study. *ACM Trans. Softw. Eng. Methodol.*, 30(1).

[Rahman and Anwar 2021] Rahman, M. A. and Anwar, Z. (2021). Secret management in cloud native environments. In *2021 IEEE International Conference on Cloud Engineering (IC2E)*, pages 54–62.

[Rahman et al. 2022] Rahman, M. R., Imtiaz, N., Storey, M.-A., and Williams, L. (2022). Why secret detection tools are not enough: It's not just about false positives - an industrial case study. *Empirical Software Engineering*, 27(3).

[Reddy Konala et al. 2023] Reddy Konala, P. R., Kumar, V., and Bainbridge, D. (2023). Sok: Static configuration analysis in infrastructure as code scripts. In *2023 IEEE International Conference on Cyber Security and Resilience (CSR)*, pages 281–288.

[Rodrigues et al. 2025] Rodrigues, M. G., Viegas, E. K., Santin, A. O., and Enembreck, F. (2025). A mlops architecture for near real-time distributed stream learning operation deployment. *Journal of Network and Computer Applications*, 238:104169.

[Rose et al. 2020] Rose, S., Borchert, O., Mitchell, S., and Connelly, S. (2020). Zero trust architecture. NIST Special Publication 800-207, National Institute of Standards and Technology, Gaithersburg, MD. Acesso em: 08 de maio de 2025.

[Silveira Neto et al. 2024] Silveira Neto, M., Malucelli, A., and Reinehr, S. (2024). Can personality types be blamed for code smells? pages 196–205.

[Simioni et al. 2025a] Simioni, J., Viegas, E. K., Santin, A., and Horchulhack, P. (2025a). An early exit deep neural network for fast inference intrusion detection. In *Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing*, SAC '25, page 730–737. ACM.

[Simioni et al. 2025b] Simioni, J. A., Viegas, E. K., Santin, A. O., and de Matos, E. (2025b). An energy-efficient intrusion detection offloading based on dnn for edge computing. *IEEE Internet of Things Journal*, 12(12):20326–20342.

[Syed et al. 2022] Syed, N. F., Shah, S. W., Shaghaghi, A., Anwar, A., Baig, Z., and Doss, R. (2022). Zero trust architecture (zta): A comprehensive survey. *IEEE Access*, 10:57143–57179.

[Vehent, Julien 2018] Vehent, Julien (2018). *Securing DevOps: Security in the Cloud*, page 384. Simon and Schuster.

[Viegas et al. 2020] Viegas, E., Santin, A., Bachtold, J., Segalin, D., Stihler, M., Marcon, A., and Maziero, C. (2020). Enhancing service maintainability by monitoring and auditing sla in cloud computing. *Cluster Computing*, 24(3):1659–1674.

[Viegas et al. 2017] Viegas, E., Santin, A., Neves, N., Bessani, A., and Abreu, V. (2017). A resilient stream learning intrusion detection mechanism for real-time analysis of network traffic. In *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, page 1–6. IEEE.

[Wang 2022] Wang, R. (2022). *Infrastructure as Code Patterns and Practices: With Examples in Python and Terraform*. Book Collection ITPro. Manning, Shelter Island, NY, 1st edition. Acesso em: 07 maio de 2025.

[Wettinger et al. 2014] Wettinger, J., Breitenbücher, U., and Leymann, F. (2014). Comparing and combining deployment automation approaches. In *Proceedings of the 2014 International Conference on Web Services (ICWS)*, pages 305–312. IEEE.

[Yuliarman et al. 2023] Yuliarman, S., Ridwan, S. H., and Resi, S. B. (2023). *Implementation of Hashicorp Vault as Multi-Factor Data Communication Security in Integration with Modern Infrastructure*. SNTE.

[Özdoğan et al. 2023] Özdoğan, E., Ceran, O., and ÜSTÜNDAĞ, M. (2023). Systematic analysis of infrastructure as code technologies. *Gazi University Journal of Science Part A: Engineering and Innovation*, 10.

Capítulo 2

Zero Trust Architecture para Dispositivos de Internet das Coisas e Redes de Próxima Geração: Ferramentas, Tendências e Desafios

Guilherme N. N. Barbosa (UFF), Martin Andreoni (TII),
Diogo M. F. Mattos (UFF)

Abstract

The exponential growth of connected devices and the complexity of emerging networking infrastructures increase digital security risks. The vulnerability of resource-constrained devices and the obsolescence of perimeter models hinder the adequate protection of data and services. This chapter surveys the adoption of Zero Trust Architecture (ZTA) as a solution to ensure continuous authentication, microsegmentation, and dynamic access policies in heterogeneous networks. The approach combines theoretical exposure with practical activities using the OpenZiti platform to implement services in a ZTA environment. The chapter discusses fundamentals of the architecture, identifies its main components, discusses the configuration of secure tunnels, and critically reflects on the challenges and opportunities of ZTA in next-generation networks. The chapter also explores real-world use cases in enterprise environments and federated networks.

Resumo

O crescimento exponencial de dispositivos conectados e a complexidade das infraestruturas de rede emergentes aumentam os riscos à segurança digital. A vulnerabilidade de dispositivos com recursos limitados e a obsolescência dos modelos perimetrais dificultam a proteção adequada de dados e serviços. Este capítulo examina a adoção da Arquitetura de Confiança Zero (Zero Trust Architecture - ZTA) como solução para garantir autenticação contínua, microsegmentação e políticas dinâmicas de acesso em redes heterogêneas. A abordagem combina exposição teórica com atividades práticas utilizando a plataforma OpenZiti para implementar serviços em um ambiente baseado em ZTA. O capítulo discute os fundamentos da arquitetura, identifica seus principais componentes, aborda a configuração de túneis seguros e propõe uma reflexão crítica sobre os desafios e as oportunidades da ZTA em redes de próxima geração. Também são explorados casos de uso reais em ambientes corporativos e redes federadas.

Este capítulo foi realizado com recursos do CNPq, CAPES, RNP e FAPERJ. Ferramentas de Inteligência Artificial Generativa, incluindo ChatGPT, Grammarly e Llama3.1, foram empregadas na revisão textual deste trabalho.

2.1. Introdução

A evolução das redes de computadores, impulsionada por tecnologias como Redes Definidas por *Software* (*Software Defined Networking* - SDN), Virtualização de Funções de Rede (*Network Function Virtualization* - NFV) e computação em nuvem, redefiniu o conceito de segurança, anteriormente centrado em perímetros físicos e esquemas tradicionais de autenticação e autorização. A virtualização e a descentralização tornaram obsoletas as abordagens baseadas em perímetro, impondo desafios mais complexos diante do acesso remoto e dinâmico de usuários [Jose Diaz Rivera et al., 2024]. Com a implementação da quinta geração (5G) dos sistemas de comunicação móvel, teve início uma nova era caracterizada por conectividade mais fluida, adaptável e de baixa latência. Essa transformação exigiu o fortalecimento das infraestruturas de rede, visando maior robustez e eficiência para suportar aplicações críticas com requisitos rigorosos de desempenho. Dentre os setores que mais se beneficiam dessa evolução destacam-se os veículos autônomos, a telecirurgia, as aplicações com drones e os sistemas de comunicação máquina a máquina (*Machine to Machine* - M2M), fundamentais no contexto da Internet das Coisas (*Internet of Things* - IoT), que já integra a infraestrutura digital conectando bilhões de dispositivos e promovendo avanços em áreas como saúde, mobilidade urbana e cidades inteligentes [Cunha Neto et al., 2024].

Projeções indicam que, até 2030, haverá cerca de 29 bilhões de dispositivos conectados à Internet [Gebresilassie et al., 2025]. Diante desse crescimento, torna-se imperativo o fortalecimento dos mecanismos de segurança, dado que esses dispositivos tendem a possuírem capacidades limitadas de proteção e se tornam vetores potenciais de ataques [Chen et al., 2023]. Tais vulnerabilidades expõem dados sensíveis a riscos crescentes, em um cenário regulado por legislações como a Lei Geral de Proteção de Dados (LGPD), no Brasil, e a *General Data Protection Regulation* (GDPR), na Europa. Embora amplamente difundida, a IoT impõe desafios crescentes, como privacidade, interoperabilidade, desempenho, regulamentação e confiança [Gebresilassie et al., 2025]. A heterogeneidade e distribuição dos dispositivos ampliam a superfície de ataque e dificultam a aplicação de soluções clássicas, como *firewalls* e sistemas de detecção de intrusão (*Intrusion Detection Systems* - IDS), sobretudo quando esses dispositivos compartilham redes locais com ativos críticos. Ademais, suas limitações computacionais frequentemente levam fabricantes a suprimir funcionalidades de segurança. A isso somam-se práticas como o uso corporativo de dispositivos pessoais (*Bring Your Own Device* – BYOD), o crescimento do teletrabalho e o uso de tecnologias de código aberto, ampliando os riscos e exigindo arquiteturas que tratem todos os dispositivos como potenciais ameaças.

Nesse cenário desafiador, o número de ciberataques tem aumentado significativamente em frequência e gravidade. Destacam-se os ataques de negação de serviço distribuído (*Distributed Denial of Service* - DDoS), viabilizados pela mobilização de bilhões de dispositivos comprometidos em redes *botnet*. Segundo relatório da Checkpoint[1], no primeiro trimestre de 2025, os ciberataques semanais aumentaram 47% em relação ao mesmo período de 2024. O Departamento de Ciência, Inovação e Tecnologia do Reino

[1]Disponível em: `https://blog.checkpoint.com/research/q1-2025-global-cyber-attack-report-from-check-point-software-an-almost-50-surge-in-cyber-threats-worldwide-with-a-rise-of-126-in-ransomware-attacks`.

Unido[2] relata que 43% das empresas e 30% das instituições de caridade sofreram incidentes de segurança no último ano. Paralelamente, Grandes Modelos de Linguagem (*Large Language Models* - LLMs) e Modelos Compactos (*Small Large-Language Models* - SLMs) se tornaram alvos de exploração, dada sua integração em diversas aplicações cotidianas e sua integração a agentes autônomos de inteligência artificial. Técnicas adversariais vêm sendo desenvolvidas para explorar vulnerabilidades, inclusive por meio de transferência de ataques entre modelos abertos e fechados, explorando similaridades arquiteturais e de dados de treinamento [Kumar et al., 2024]. Outra ameaça emergente é a injeção de *prompt* (*prompt injection*), capaz de subverter instruções de forma maliciosa. Para mitigar esses riscos, são empregados mecanismos de segurança conhecidos como *guardrails*, embora estes ainda apresentem limitações diante de ataques sofisticados [de Oliveira et al., 2025].

Apesar da evolução tecnológica, muitas organizações ainda operam com modelos tradicionais de segurança centrados em perímetros físicos, com mecanismos centralizados de autenticação e controle de acesso. Essa abordagem baseia-se na premissa de confiança implícita, assumindo que dispositivos conectados à rede corporativa são confiáveis e que os serviços da rede são confiáveis. Contudo, esse modelo apresenta sérias limitações frente a ataques internos e à mobilidade crescente de usuários. A adoção de dispositivos móveis, a migração para a nuvem e a incorporação de dispositivos IoT romperam as fronteiras tradicionais, inviabilizando a segmentação baseada em localização física. A diversidade de dispositivos amplia a superfície de ataque e dificulta a aplicação de políticas uniformes. A visibilidade limitada sobre comportamentos e atividades, principalmente na rede interna, compromete a detecção de ameaças e a resposta a incidentes, exigindo modelos de proteção mais dinâmicos, contextuais e granulares.

Esse cenário evidencia a necessidade urgente de reformulação das estratégias de cibersegurança, com ênfase em novas tecnologias e práticas organizacionais. Tais estratégias devem incluir o fortalecimento de políticas, adoção novas arquiteturas e investimento contínuo na capacitação de profissionais, com foco na proteção de ativos estratégicos, privacidade e infraestrutura crítica [AlDaajeh e Alrabaee, 2024]. Ainda que campanhas de conscientização sejam relevantes, elas não são suficientes para induzir mudanças comportamentais duradouras [van Steen, 2025]. Com o avanço das regulamentações e estratégias nacionais, cresce a exigência por estruturas robustas de segurança e alinhamento com melhores práticas internacionais [Srinivas et al., 2019].

O *National Institute of Standards and Technology* (NIST) propôs a Arquitetura de Confiança Zero (*Zero Trust Architecture* - ZTA) como um novo paradigma de segurança [Sheikh et al., 2021, Zivi e Doerr, 2022]. A ZTA busca eliminar a confiança implícita por meio de autenticação contínua e validação rigorosa de cada tentativa de acesso. A arquitetura assume que qualquer segmento da rede pode ser comprometido, exigindo que cada solicitação seja autenticada, avaliada e autorizada dinamicamente [Zivi e Doerr, 2022]. Seus princípios fundamentais incluem autenticação robusta, controle de acesso baseado em políticas, verificação contínua e segmentação granular dos recursos. Adicionalmente, o conceito de Perímetro Definido por *Software* (*Software Defined Perimeter* -

[2]Disponível em: `https://www.gov.uk/government/statistics/cyber-security-breaches-survey-2025/cyber-security-breaches-survey-2025`.

SDP) e a microsegmentação são estratégias recomendadas pelo NIST [Syed et al., 2022].

A Arquitetura de Confiança Zero [Stafford, 2020] aprimora a segurança ao aplicar monitoramento contínuo, políticas rigorosas e autenticação contextual em toda a infraestrutura de Tecnologia da Informação e Comunicação (TIC). A Rede de Confiança Zero (*Zero Trust Networking* - ZTN) protege o tráfego de rede por meio de criptografia e segmentação, restringindo as comunicações a fluxos autorizados e seguros. Este capítulo aborda os fundamentos da ZTA e apresenta uma atividade prática de implantação de serviços utilizando a plataforma OpenZiti. O objetivo central do capítulo é contextualizar a arquitetura de confiança zero no aprimoramento da segurança de dispositivos IoT e redes de próxima geração, como redes além do 5G (*beyond 5G* - B5G) e 6G [Andreoni et al., 2022]. São discutidos os componentes essenciais da arquitetura, bem como os principais desafios de sua implementação. O capítulo adota uma abordagem técnico-prática, guiando os participantes na construção de um ambiente funcional baseado em ZTA com OpenZiti. Sua relevância está atrelada ao crescimento da IoT, à intensificação de ameaças de dia zero (*Zero Day Threats*) e à adoção do modelo BYOD [Anderson et al., 2022].

O restante do capítulo está organizado da seguinte forma. A Seção 2.2 apresenta a Arquitetura de Confiança Zero. Os principais componentes, soluções comerciais, taxonomia de aplicações e comparações entre modelos são discutidos na Seção 2.3. A Seção 2.4 descreve a implementação da ZTA, abordando técnicas e aspectos práticos. A Seção 2.5 traz um exemplo de aplicação da ZTA com a infraestrutura OpenZiti. Casos de uso reais são apresentados na Seção 2.6, e a Seção 2.7 explora projetos atuais de pesquisa e desenvolvimento. Desafios e tendências futuras são tratados na Seção 2.8, enquanto a Seção 2.9 conclui o capítulo.

2.2. Arquitetura de Confiança Zero (*Zero Trust Architecture*)

O conceito de (*Zero Trust*) representa uma mudança de paradigma na cibersegurança, afastando-se da lógica tradicional de confiança implícita (“confiar, mas verificar”) para o princípio de confiança por transação (“nunca confiar, sempre verificar”). Embora esse conceito tenha se popularizado a partir de 2020, com a publicação do arcabouço definido pelo NIST, sua ideia central remonta às discussões sobre a desperimetrização iniciadas pelo Jericho Forum em 2004. Esse grupo criticava a dependência de perímetros de rede seguros e discutia os desafios de manter simultaneamente os protocolos de segurança tradicionais e a disponibilidade dos serviços [Nace, 2020]. A formalização do modelo, no entanto, foi realizada por John Kindervag, então analista da Forrester Research, em seu relatório de 2010 intitulado "*No More Chewy Centers: Introducing The Zero Trust Model Of Information Security*" [Kindervag et al., 2010]. Essencialmente, *Zero Trust* visa a eliminação da confiança implícita em qualquer parte da rede, independentemente de sua localização. Diferentemente dos modelos tradicionais, que assumem confiabilidade dentro do perímetro corporativo, o *Zero Trust* considera todo tráfego como potencialmente malicioso até que sua legitimidade seja verificada. Isso implica a verificação contínua de identidade, controle rigoroso de acesso, proteção persistente dos recursos e monitoramento detalhado de todas as atividades na rede. O conceito evoluiu ao ser integrado a arquiteturas como o *Secure Access Service Edge* (SASE) [Yiliyaer e Kim, 2022] e consolidado em políticas públicas, como a Ordem Executiva 14028 dos Estados Unidos, publicada em maio de 2021, que estabeleceu o *Zero Trust* como base para a modernização

da cibersegurança federal norte-americana [Alnoaimi e Alomary, 2025].

O Arquitetura de Confiança Zero (*Zero Trust Architecture* – ZTA), proposta pelo *National Institute of Standards and Technology* (NIST) [Stafford, 2020], constitui um paradigma baseado na premissa de que nenhuma entidade, seja usuário, dispositivo ou rede, deve ser implicitamente confiável. Essa abordagem adota um mecanismo de defesa com foco em **usuários, ativos e recursos individuais**, implementando verificações contínuas de identidade e autenticidade como critério para concessão de acesso. A transição do modelo de segurança atual, centrado em perímetros, para o modelo baseado em confiança zero é complexa, exigindo especialmente a reavaliação dos processos relacionados à segurança [Chen et al., 2023]. A autenticação e a autorização contínuas a cada solicitação aumentam a complexidade e requerem sistemas robustos de gerenciamento de identidade. Outro aspecto é a necessidade de todos os dispositivos manterem um nível mínimo de segurança, que deve ser monitorado e atualizado continuamente. A criptografia exerce papel essencial ao garantir a confidencialidade das comunicações, enquanto a microsegmentação permite dividir a rede em blocos menores, nos quais é possível aplicar políticas específicas. Essa estratégia não apenas limita os movimentos laterais de agentes maliciosos dentro da rede, mas também facilita o isolamento rápido de ameaças detectadas, especialmente em cenários envolvendo ataques do tipo *ransomware*. *Ransomware* é um tipo de *software* malicioso que, ao infectar um sistema, criptografa dados críticos e exige o pagamento de um resgate para restaurar o acesso. Esse tipo de ataque explora vulnerabilidades em dispositivos, aplicações ou credenciais, propagando-se rapidamente dentro das redes corporativas, especialmente quando há confiança implícita entre os elementos da infraestrutura. A ascensão e a popularização de *ransomware* nos últimos anos se devem à combinação de fatores como a ampliação da superfície de ataque, a interconectividade dos sistemas, o uso extensivo de dispositivos pessoais e o acesso remoto a ambientes sensíveis. Nesse contexto, a arquitetura tradicional baseada em perímetro se mostra insuficiente, pois assume confiabilidade interna e dificulta o isolamento de ameaças. A abordagem de confiança zero (*Zero Trust*), ao eliminar pressupostos de confiança e exigir autenticação e autorização contínuas para cada interação, torna-se essencial para conter movimentos laterais, limitar o escopo de comprometimento e viabilizar respostas rápidas e segmentadas a ataques de *ransomware*.

Os princípios que norteiam uma arquitetura de confiança zero são dinâmicos e adaptáveis, evoluindo conforme as ameaças cibernéticas se transformam, devendo ser agnósticos de tecnologias. São classificados como [Stafford, 2020]:

- **Todas as fontes de dados e serviços de computação são considerados recursos.** Uma rede pode incluir diversas classes de dispositivos. Uma organização pode optar por classificar dispositivos pessoais como recursos da empresa, caso estes tenham acesso a ativos corporativos.
- **Todas as comunicações são protegidas, independentemente da localização da rede.** Estar em uma rede corporativa não implica automaticamente confiança. Requisições de acesso, seja de dentro ou fora da infraestrutura, devem seguir os mesmos requisitos de segurança.
- **O acesso a recursos corporativos individuais é concedido por sessão.** A confi-

ança no solicitante é avaliada antes da concessão. O acesso deve ser concedido com o mínimo de privilégios necessários para a tarefa. Isso pode significar que o acesso foi concedido recentemente para a transação específica e não necessariamente imediatamente antes de iniciar a sessão ou transação. No entanto, a autenticação e autorização para um recurso não garantem automaticamente o acesso a outro recurso.

- **O acesso aos recursos é determinado por uma política dinâmica, podendo incluir outros atributos comportamentais e ambientais.** Uma organização protege seus recursos ao defini-los, identificando os membros envolvidos e estabelecendo os níveis de acesso necessários. No modelo de confiança zero, a identidade do usuário abrange a conta, os atributos associados e os artefatos utilizados para autenticação. O estado do dispositivo considera fatores como *software* instalado, localização geográfica, histórico de comportamento e credenciais. As políticas de acesso são baseadas em atributos do usuário, das propriedades do ativo e do ambiente, incluindo variáveis como localização e horário. As regras de acesso são alinhadas aos processos de negócios e ao nível de risco, adotando o princípio do menor privilégio para restringir visibilidade e acessibilidade de forma precisa e proporcional às necessidades operacionais.

- **A organização monitora e mede a integridade e a postura de segurança de todos os ativos próprios e associados.** Nenhum ativo é confiável por padrão. A segurança dos ativos é avaliada pela empresa durante o processamento de solicitações de acesso. Na implementação de uma arquitetura de confiança zero, torna-se essencial a adoção de um sistema contínuo de diagnóstico e mitigação (CDM) para monitorar dispositivos e aplicações, garantindo a aplicação de correções sempre que necessário. Ativos comprometidos, com vulnerabilidades conhecidas ou não gerenciados pela organização, podem ser submetidos a restrições severas, incluindo a negação de acesso a recursos corporativos. Essa abordagem também se estende a dispositivos pessoais, que podem ter acesso limitado a recursos específicos. Um sistema robusto de monitoramento e geração de relatórios é fundamental para fornecer informações precisas sobre o estado dos ativos da organização, permitindo ações proativas e baseadas em dados.

- **Toda autenticação e autorização de recursos são dinâmicas e rigorosamente aplicadas antes que o acesso seja permitido.** Um ciclo contínuo para acesso, avaliação de ameaças e adaptação constante da confiança na comunicação. Entidades que adotam a arquitetura de confiança zero devem implementar sistemas de gerenciamento de identidade, credenciais e ativos (ICAM), além de autenticação multifator (*Multi-Factor Authentication* - MFA) para acesso aos ativos. O monitoramento contínuo com reautenticação e reautorização ocorre em todas as transações, conforme definido pela política, visando equilibrar segurança, disponibilidade, usabilidade e custo.

- **Coleta do máximo de informações possível sobre o estado atual dos ativos, infraestrutura de rede e comunicações, utilizadas para aprimorar a segurança.** Uma organização deve coletar dados relacionados ao comportamento de segurança

dos ativos, como tráfego de rede e solicitações de acesso. Em seguida, processá-los e utilizar as percepções obtidas para aprimorar a criação e a aplicação de políticas de segurança. Esses dados podem ser usados para fornecer um contexto mais robusto nas avaliações de solicitações de acesso realizadas pelos usuários.

2.3. Principais Componentes da Arquitetura de Confiança Zero

O modelo de confiança zero é estruturado em diversos componentes lógicos, projetados para garantir que nenhuma entidade, interna ou externa, seja confiável por padrão. A interação entre esses componentes ocorre em um plano de controle dedicado, enquanto os dados das aplicações trafegam separadamente no plano de dados. Entre os principais componentes destacam-se o *Policy Decision Point* (PDP) e *Policy Enforcement Point* (PEP). O PDP é subdividido em dois elementos lógicos: o *Policy Engine (PE)* e o*Policy Administrator (PA)*. O *Policy Engine* é responsável por avaliar as políticas e determinar as decisões de acesso com base em atributos como identidade, contexto e regras predefinidas. Já o *Policy Administrator* implementa essas decisões, gerenciando as configurações de controle de acesso e aplicando as políticas nos componentes do plano de dados. Essa separação clara entre os planos de controle e de dados é essencial para garantir um gerenciamento seguro e eficiente dos recursos [Stafford, 2020]. Outro componente importante é o Algoritmo de Confiança (*Trust Algorithm* - TA), sendo esse utilizado para conceder ou negar acesso a um ativo. O TA recebe entradas de várias fontes, incluindo banco de dados de políticas, informações observáveis sobre os assuntos, padrões históricos de comportamento dos usuários, fontes de inteligência de ameaças e outros metadados. Essas entradas são categorizadas para avaliar a solicitação de acesso com base em critérios pré-definidos ou uma pontuação de confiança. Existem variações no TA, como a avaliação baseada em critérios versus a baseada em pontuação e a consideração singular versus contextual das solicitações [Stafford, 2020].

- ***Policy Decision Point* (PDP)** é o componente responsável por decidir se o acesso a um recurso deve ser permitido ou negado, além de estabelecer ou encerrar a comunicação entre uma entidade (usuário, dispositivo ou serviço) e o recurso solicitado [Teerakanok et al., 2021]. Em sua estrutura, o PDP é dividido em dois elementos lógicos: o **Policy Engine (PE)** e o **Policy Administrator (PA)**, que atuam para tomada de decisão e gestão de comunicação, respectivamente.

- ***Policy Engine* (PE)** é o componente central responsável pela decisão final de conceder, negar, revogar ou restringir o acesso de uma entidade a um determinado recurso. Para determinar sua decisão, este mecanismo utiliza um algoritmo que processa um conjunto de regras (políticas de segurança) baseado no princípio do privilégio mínimo. As entradas para esse algoritmo incluem as políticas pré-definidas e dados de fontes externas, como sistemas de Diagnóstico e Mitigação Contínuos (CDM), serviços de inteligência (*Threat Intelligence*) e registros de atividades. Operando em conjunto com o componente o *Policy Administrator*, o *Policy Engine* verifica explicitamente cada solicitação, toma a decisão, registra-a como aprovada ou negada, e o *Policy Administrator* executa a decisão [Stafford, 2020, Hussain et al., 2024].

- ***Policy Administrator* (PA)** é o componente responsável por estabelecer e encerrar a comunicação entre uma entidade e um recurso, atuando sob as decisões do *Policy Engine*. Quando uma sessão é autorizada, o PA gera *tokens* de autenticação ou credenciais específicas para a sessão e configura o *Policy Enforcement Point* (PEP) para iniciar a comunicação. Inversamente, caso a sessão seja negada ou uma autorização prévia seja revogada, o PA comunica o PEP para encerrar a conexão. Essa interação, que ocorre no plano de controle, assegura que o acesso aos recursos seja controlado de forma dinâmica, garantindo que todas as conexões sejam devidamente autorizadas e monitoradas. Embora algumas implementações possam unificar o PA e o PE, eles são conceitualmente definidos como componentes lógicos distintos [Stafford, 2020].

- ***Policy Enforcement Point* (PEP)** é responsável por habilitar, monitorar e, eventualmente, encerrar a conexão entre uma entidade e um recurso. Sua comunicação é feita com o *Policy Administrator* para encaminhar requisições e receber atualizações das políticas a serem aplicadas. Embora constitua um componente lógico em uma arquitetura de confiança zero (ZTA), o PEP pode ser implementado de maneira dividida, com um agente no lado do cliente e um *gateway* no lado do recurso. Esse mecanismo realiza o gerenciamento dinâmico do acesso e, caso detecte qualquer atividade suspeita durante a conexão, o PEP, por intermédio do PA, sinaliza o término imediato da sessão. Dessa forma, garante-se um controle de acesso seguro, contínuo e responsivo, sendo que a zona de confiança que contém o recurso corporativo situa-se após do PEP.

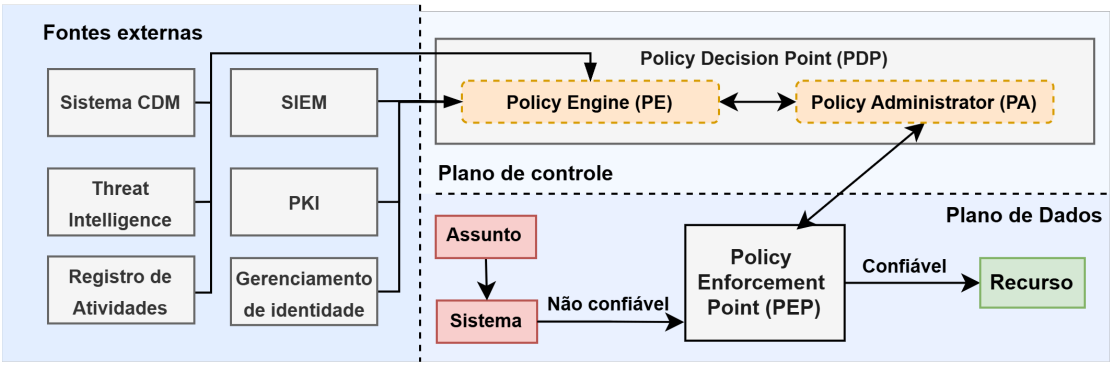


Figura 2.1. Componentes principais do modelo Zero Trust. Fontes externas como sistemas CDM, SIEM, *threat intelligence*, PKI, gerenciamento de identidade e registros de atividades fornecem informações ao *Policy Engine* (PE), que compõe, junto ao *Policy Administrator* (PA), o *Policy Decision Point* (PDP). O PE avalia as condições de acesso com base nas políticas definidas. O *Policy Enforcement Point* (PEP) aplica essas decisões, classificando o acesso do sistema como confiável ou não confiável. Apenas acessos considerados confiáveis são autorizados a interagir com o recurso no plano de dados.

Além dos componentes principais dos planos de dados e controle, outras fontes de dados podem ser utilizadas para fornecer regras de entrada e tomar decisões de acesso [Abdalla et al., 2024], tais como:

- **Sistema Contínuo de Diagnóstico e Mitigação (CDM)** desempenha um papel fundamental ao coletar informações atualizadas sobre o estado atual dos ativos. Sua principal função é implementar atualizações críticas em configurações e componentes de *software*, avaliando se os ativos estão executando componentes adequadamente, verificando a integridade dos componentes de *software* e identificar quaisquer componentes não autorizados, sinalizando possíveis vulnerabilidades.

- ***Threat intelligence feed(s)*** fornecem informações valiosas de diversas fontes, tanto internas quanto externas, auxiliando o PDP de confiança zero a tomar decisões de acesso informadas. Esses *feeds* atualizam constantemente o PDP sobre ameaças emergentes, vulnerabilidades recém-descobertas, relatórios de *malware* e ataques recentes a outros ativos. Ao aproveitar essa inteligência, o PDP pode identificar rapidamente riscos potenciais e impedir o acesso de fontes suspeitas.

- **Logs de rede e de atividades** atuam como um repositório de dados, coletando informações detalhadas sobre o tráfego de rede, acessos e eventos do sistema. Esses dados, atualizados em tempo quase real, alimentam sistemas de detecção e resposta a incidentes, permitindo a identificação rápida de ameaças e a tomada de decisões para a proteção da rede.

- ***Security Information and Event Management* (SIEM)** realiza a coleta, análise e o armazenamento de dados de eventos de toda a infraestrutura. Seu principal objetivo é fornecer uma visão completa da segurança da rede, agilizando a detecção de atividades suspeitas, incidentes cibernéticos e violações de políticas. Geralmente esses sistemas utilizam técnicas de correlação e análise comportamental, acelerando a identificação de ameaças.

- ***Public Key Infrastructure* (PKI)** é um conjunto de tecnologias, políticas e procedimentos utilizados para gerenciar o ciclo de vida de certificados digitais e chaves criptográficas. Baseia-se em criptografia assimétrica e depende de Autoridades Certificadoras (CAs) para a emissão, validação e revogação de certificados, sendo fundamental para aplicações como autenticação, assinatura digital e comunicação segura.

Enquanto o *Policy Engine* (PE) funciona como o cérebro, o *Trust Algorithm* constitui o processo utilizado pelo PE para conceder ou negar acesso a um recurso solicitado. Essa concessão baseia-se na análise de um conjunto de informações compiladas de múltiplas fontes, dentre as quais se destacam: a requisição de acesso, os base de entidades, base de ativos (abrangendo dispositivos como BYOD), as políticas de acesso, os registros de segurança e de tráfego de rede, bem como os dados de sistemas de inteligência de ameaças [Teerakanok et al., 2021]. A Figura 2.2 apresenta o processo de como as entradas podem ser categorizadas para serem utilizadas pelo *Trust Algorithm* [Stafford, 2020, Hussain et al., 2024]:

- **Requisição de acesso** refere-se ao processo pelo qual um indivíduo ou sistema explicitamente solicita acesso a um recurso específico. Essa requisição inclui principalmente informações sobre o recurso solicitado, mas também considera diversos

atributos de quem realiza a solicitação, tais como versão do sistema operacional, *software* utilizado, por exemplo, se o aplicativo está em uma lista de aplicações autorizadas, e nível de atualização de segurança. Dependendo desses fatores e do nível de segurança do ativo solicitado, o acesso pode ser restrito ou negado.

- **Base de entidades** é um repositório que contém as credenciais de todos as entidades (humanos e aplicações) e suas informações associadas, como atributos e privilégios. Este banco de dados detalha "quem" está solicitando acesso, incluindo identidades lógicas e resultados de autenticações. Atributos como hora e geolocalização são usados para avaliar a confiança. É crucial que os privilégios sejam atribuídos individualmente e não apenas por função. Todas essas informações são codificadas e armazenadas em um sistema de gerenciamento de identidade e banco de dados de políticas, sustentando a avaliação contínua da confiança em cada solicitação.

- **Base de ativos** é um repositório, ou sistema de inventário, que armazena o *status* dos ativos de uma organização (físicos, virtuais ou BYOD). Ele compara esse *status* registrado com o *status* observável de um ativo que faz uma requisição, incluindo informações como versão do sistema operacional, *softwares* e sua integridade, localização e nível de atualização. Com base nessa comparação, o acesso a outros ativos pode ser restrito ou negado.

- **Políticas de acesso** complementam a base de entidades, definindo os requisitos mínimos para acessar os recursos. As políticas podem incluir níveis de garantia como localização da autenticação multifatorial, por exemplo, negar acesso de endereços IP estrangeiros, confidencialidade dos dados e solicitações de configuração. Esses requisitos devem ser desenvolvidos tanto pelo custodiante dos dados quanto pelos responsáveis pelos processos de negócios que utilizam os dados.

- ***Threat Intelligence e Logs*** são as informações como assinaturas de ataques recém-descobertos ou *malware* operando na Internet, fornecidas por fontes internas ou externas.

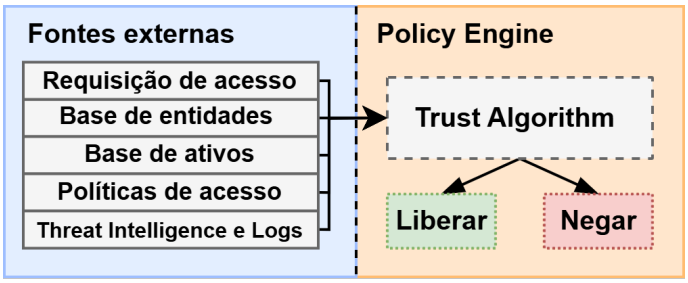


Figura 2.2. Funcionamento do *Policy Engine* a partir de fontes externas e como o algoritmo de confiança (*Trust Algorithm*) recebe dados de diversas fontes, como requisições de acesso, bases de entidades e ativos, políticas de acesso e informações de ameaças com *logs*. Com base nos dados, o algoritmo decide se a solicitação deve ser liberada ou negada.

2.3.1. Arquiteturas Alternativas e Componentes Inovadores

A evolução da *Zero Trust Architecture* (ZTA) exige a exploração de modelos e componentes alternativos que possam complementar ou até mesmo substituir a estrutura clássica formada pelo *Policy Engine* (PE), *Policy Administrator* (PA) e *Policy Enforcement Point* (PEP). Em ambientes altamente dinâmicos, móveis ou distribuídos, como sistemas industriais, redes táticas ou infraestruturas multi-cloud, os mecanismos tradicionais podem não oferecer a agilidade, visibilidade ou adaptabilidade necessárias. Nesse contexto, arquiteturas como *Single Packet Authorization* (SPA), *Software Defined Perimeter* (SDP) e modelos endógenos emergem como alternativas promissoras para estender os princípios de confiança mínima e verificação contínua da ZTA a novos domínios operacionais [Gupta et al., 2024, Dhiman et al., 2024, Gambo e Almulhem, 2025].

Uma abordagem recente e com destaque é o uso de *Single Packet Authorization* (SPA), um mecanismo de controle de acesso que se baseia na aceitação de um único pacote autenticado e criptografado como gatilho para liberar o acesso a um serviço ou porta protegida. Diferente de *firewalls* tradicionais, que estão sempre escutando, o SPA oculta completamente os serviços até que um pacote válido seja recebido, tornando os alvos efetivamente invisíveis a *scanners* ou sondas externas [Gupta et al., 2024]. Quando aplicado à ZTA, o SPA pode operar como uma extensão do PEP, adicionando uma camada de invisibilidade e controle extremamente granular sobre quais pacotes são considerados para avaliação. Isso é particularmente útil em ambientes industriais e militares, em que a redução da superfície de ataque é crítica.

Outra arquitetura complementar relevante é o *Software Defined Perimeter* (SDP), que pode ser interpretado como uma VPN de nova geração associada a políticas dinâmicas, identidade forte e microsegmentação. O SDP define quem pode ver e acessar recursos com base na verificação contextual de identidade, comportamento do dispositivo e localização, além de manter uma superfície de ataque mínima. A visibilidade é tratada como privilégio e não como pressuposto. Em ZTA, o SDP pode atuar como camada de controle entre o usuário e os recursos, aplicando autenticação mútua e verificação contínua antes de conceder acesso a qualquer *endpoint* [He et al., 2022, Nahar et al., 2024]. Essa abordagem é altamente compatível com redes 6G e IoT, em que os limites físicos de rede são efêmeros e os perímetros precisam ser definidos logicamente.

Propostas recentes de arquiteturas *endógenas* definem sistemas projetados para monitorar continuamente a si mesmos e adaptar seu comportamento com base em padrões históricos e aprendizado contínuo. Essas arquiteturas se inspiram em sistemas imunes artificiais, computação auto-defensiva e comportamentos emergentes adaptativos. Elas podem, por exemplo, identificar variações estatísticas no comportamento de processos ou fluxos de rede e ajustar automaticamente as políticas de acesso, a configuração de isolamento ou o nível de confiança atribuído a determinados dispositivos [Gambo e Almulhem, 2025]. A ZTA pode incorporar tais mecanismos como uma camada adicional de análise comportamental local, aumentando sua capacidade de resposta autônoma a ameaças em tempo real.

A adoção dessas arquiteturas alternativas e componentes não substitui diretamente a ZTA tradicional, mas a complementa. A combinação de técnicas como SPA para ocultação de superfície de ataque, SDP para controle baseado em identidade e arquiteturas

Tabela 2.1. Comparativo entre arquiteturas aplicadas à Zero Trust.

Modelo	Princípio Central	Aplicação em ZTA	Benefícios principais
Single Packet Authorization (SPA)	Liberação condicional via pacote criptografado	Acesso sigiloso a serviços e portas; invisibilidade por padrão	Superfície de ataque reduzida; difícil detecção por atacantes
Software Defined Perimeter (SDP)	Identidade como perímetro e controle de visibilidade	Camada de controle de acesso entre dispositivos e serviços	Microsegmentação; negação por padrão; controle granular
Arquitetura Endógena	Adaptação autônoma e auto-monitoramento	Reforço de políticas com base em análise comportamental interna	Detecção precoce de desvios; resposta adaptativa sem intervenção externa

endógenas para inteligência contextual posiciona a segurança como um sistema dinâmico e proativo. À medida que ambientes computacionais tornam-se mais heterogêneos, autônomos e distribuídos, a integração dessas abordagens será essencial para garantir a escalabilidade, resiliência e adaptabilidade dos mecanismos de controle de acesso.

2.3.2. Soluções Comerciais

Por se tratar de um paradigma crucial para a cibersegurança, diversas empresas já desenvolveram abordagens para implementação da arquitetura de confiança zero. Em 2009, o Google, através da solução **BeyondCorp**[3] começou uma iniciativa interna para permitir que os funcionários trabalhassem em redes não confiáveis sem o uso de uma Rede Virtual Privada (*Virtal Private Network* - VPN). Isso ocorreu após uma ação conhecida como Operação Aurora [Tankard, 2011]. Nesse incidente, os invasores obtiveram acesso à rede interna do Google, visando fontes de propriedade intelectual. Eles então usaram o sistema comprometido como um "ponto de partida" para se movimentar lateralmente dentro da rede. Consequentemente, o principal objetivo do Google era reduzir a confiança implícita que usuários e dispositivos haviam desenvolvido com base em sua presença física ou eletrônica na rede corporativa [Hosney et al., 2022]. Assim, todo acesso a sistemas internos só é liberado após três requisitos simultâneos: (i) autenticação do usuário, (ii) verificação do estado do dispositivo e (iii) criptografia de ponta a ponta. Esse processo garante que a experiência de uso seja idêntica para quem está na rede corporativa ou fora dela. Para reforçar essa lógica, o Google mantém uma "rede sem privilégios", isolada em endereços privados, que se comporta como se fosse externa. Dentro desse ambiente, listas de controle de acesso estritamente gerenciadas definem, dispositivo a dispositivo, quais recursos cada um pode alcançar, assegurando segmentação e mínimo privilégio [Kang et al., 2023].

Considerando centros de dados (*data centers*), organizações utilizam cada vez mais soluções de Rede Definida por *Software* (SDN), tais como o **VMware NSX**[4]. Esta plataforma permite a implementação de microssegmentação de forma granular, ao oferecer um modelo de segurança distribuída que atua diretamente no hipervisor. As políticas

[3]Disponível em `https://cloud.google.com/beyondcorp?hl=pt_br`.

[4]Disponível em `https://www.vmware.com/docs/vmware-zero-trust-evolution`.

de controle são aplicadas diretamente na interface virtual de cada máquina virtual, o que garante um nível elevado de isolamento entre cargas de trabalho. A microssegmentação é realizada por meio da definição de zonas de segurança lógicas entre diferentes aplicações, independentemente da sua localização física ou da estrutura de rede. As regras de segurança podem ser configuradas com base em múltiplos atributos, como identidade da VM, tipo de aplicação, sistema operacional ou regras de afinidade, dispensando a dependência exclusiva de endereços IP ou segmentações tradicionais. Com isso, é possível restringir a comunicação entre sistemas apenas ao que for estritamente necessário para o funcionamento das aplicações, reduzindo significativamente a superfície de ataque. Além disso, a capacidade de limitar interações desnecessárias entre cargas de trabalho contribui para dificultar o movimento lateral de ameaças em ambientes comprometidos, o que reforça a adoção de princípios alinhados ao modelo de confiança zero.

Para atender aos princípios da confiança zero no contexto de identidade, o **Microsoft Entra**[5] funciona como uma família de produtos de gerência de identidade e controle de acesso. Ele permite que as organizações não utilizem a confiança implícita, tratando cada identidade como um potencial ponto de violação. A solução implementa um arcabouço que verifica explicitamente todas as identidades, valida as condições de acesso, confirma permissões e monitora continuamente atividades suspeitas, garantindo que apenas usuários e dispositivos autenticados e autorizados acessem os recursos de rede de forma segura. Ainda nesse contexto, o **Okta** é um serviço de Gerenciamento de Identidade e Acesso (*Identity and Access Management* - IAM) que funciona como um intermediário para autenticação de usuários em diversas aplicações e sistemas de uma organização. Sua arquitetura é baseada em nuvem e seu objetivo é centralizar o controle de acesso. No processo de login de um serviço integrado, o Okta redireciona o usuário para sua própria plataforma para a validação das credenciais de acesso. A plataforma então utiliza protocolos padrão da indústria para confirmar a identidade do usuário, podendo aplicar requisitos como *Single Sign-On* (SSO), que permite o uso de um único login para múltiplos serviços, ou Autenticação Multifator (MFA), que exige uma segunda forma de comprovação de identidade. Após a validação, o Okta informa ao aplicativo que o usuário está autorizado, permitindo o acesso com base nas políticas que a organização configurou previamente.

O **Zscaler Zero Trust Exchange**[6] é uma plataforma em nuvem desenvolvida para implementar os princípios da arquitetura de confiança zero, com foco em segurança e conectividade. Em vez de permitir que usuários ou dispositivos acessem diretamente a rede corporativa, o Zscaler cria conexões seguras e verificadas entre usuários autenticados e as aplicações específicas que eles estão autorizados a acessar, sem que a rede como um todo seja exposta. Essa solução baseia-se em quatro pilares: identidade, dispositivo, contexto e política. A plataforma verifica a identidade do usuário, dispositivo por meio de integrações com provedores de identidade externos. Uma vez validada a identidade, o destino da conexão é avaliado para que, em seguida, sejam determinados os riscos baseados no contexto, considerando fatores como comportamento do usuário, postura

[5]Disponível em `https://learn.microsoft.com/pt-br/entra/fundamentals/what-is-entra`.

[6]Disponível em `https://www.zscaler.com/br/products-and-solutions/zero-trust-exchange-zte`.

do dispositivo, destino entre outros. Atendendo a todos os requisitos, as políticas são aplicadas por sessão e por solicitação.

O **Prisma Access** é uma plataforma de segurança em nuvem da Palo Alto[7] com objetivo de proteger o acesso a aplicações e *Software as a Service* (SaaS). Seu funcionamento baseia-se na consolidação de múltiplas funções de segurança e rede, gerenciado de forma centralizada. Após o usuário se conectar, a plataforma realiza uma verificação de confiança contínua com base no comportamento, com o objetivo de reduzir a superfície de ataque. Todo o tráfego, seja para aplicações de Internet, SaaS ou privadas, é protegido por meio de uma inspeção de segurança. A abordagem de *Zero Trust* é aplicada através do "ZTNA Connector"para segmentar usuários e redes através de um túnel seguro, garantindo que o acesso seja concedido apenas aos recursos estritamente necessários, seguindo o princípio de privilégio mínimo. A visibilidade é mantida por meio de uma política de prevenção de perda de dados (*Data Loss Prevention* - DLP) única para proteger tanto o acesso quanto os dados em toda a empresa.

2.3.3. Taxonomia de Aplicações, Tecnologias e Requisitos da ZTA

A adoção da *Zero Trust Architecture* (ZTA) vem se consolidando como uma resposta estratégica aos desafios de segurança em ambientes cada vez mais distribuídos, móveis e heterogêneos. Gambo *et al.* apresentam uma taxonomia abrangente que sistematiza os principais elementos que compõem uma arquitetura ZTA moderna e busca organizar o conhecimento existente em torno de quatro eixos fundamentais: os domínios de aplicação da ZTA, as tecnologias habilitadoras, os mecanismos de controle de acesso e os requisitos funcionais e não funcionais para sua implementação eficaz [Gambo e Almulhem, 2025].

Os domínios de aplicação identificados abrangem contextos como *Internet das Coisas* (*Internet of Things* - IoT), redes móveis emergentes (5G e 6G), ambientes *multi-cloud* e sistemas industriais (*Industrial Control Systems* - ICS/SCADA). Cada um desses cenários impõe exigências específicas de autenticação, controle de acesso e resiliência. Por exemplo, em sistemas IoT, a comunicação entre sensores deve ser autenticada continuamente em redes potencialmente maliciosas, já em redes 6G, a escalabilidade e a latência de decisões são aspectos críticos.

Gambo *et al.* apontam as tecnologias habilitadoras da ZTA como sendo autenticação contínua e multifatorial, microsegmentação, sensoriamento contextual, criptografia ponta-a-ponta, integração com *blockchain* para auditoria imutável e o uso de inteligência artificial para análise comportamental e adaptação de políticas de segurança. Essas tecnologias tendem a operar de forma coordenada para permitir decisões de acesso granulares e dinâmicas, rompendo com o paradigma de confiança estática baseada em perímetro.

Quanto aos mecanismos de controle de acesso, a taxonomia reconhece os modelos Controle de Acesso Baseado em Papéis (*Role-Based Access Control* - RBAC), Controle de Acesso Baseado em Atributos (*Attribute-Based Access Control* - ABAC) e Controle de Acesso Baseado em Políticas (*Policy-Based Access Control* - PBAC). O RBAC baseia-se na associação de usuários a papéis predefinidos, oferecendo simplicidade e facilidade de administração, mas apresenta limitações em ambientes dinâmicos por sua rigidez e falta

[7]Disponível em `https://www.paloaltonetworks.com/resources/datasheets/prisma-access`.

de contexto. O ABAC supera essas limitações ao permitir decisões com base em múltiplos atributos, como identidade, dispositivo, localização e horáriom promovendo maior flexibilidade e granularidade. No entanto, sua gestão pode se tornar complexa à medida que o número de atributos e regras cresce. Por sua vez, o PBAC formaliza o controle de acesso por meio de políticas explícitas, escritas em linguagens declarativas, facilitando auditoria, modularidade e integração com rotinas automatizadas [Oliveira et al., 2024]. Apesar disso, requer automações especializadas e conhecimento técnico mais avançado. No contexto de ZTA, ABAC e PBAC são mais alinhados aos princípios de verificação contínua e privilégio mínimo, sendo preferidos em implementações que exigem adaptabilidade e governança refinada sobre decisões de acesso. Embora o RBAC continue amplamente utilizado, ele é limitado em cenários dinâmicos. O ABAC, por sua vez, permite decisões mais refinadas ao considerar atributos de usuários, dispositivos e contexto. O PBAC é apontado como o mais aderente à filosofia *Zero Trust*, por sua capacidade de representar regras contextuais explícitas e adaptativas. A Tabela 2.2 compara as características dos diferentes modelos de controle de acesso para a aplicação em ZTA e a Tabela 2.3 apresenta exemplos de categorias e elementos da taxonomia [Gambo e Almulhem, 2025].

A taxonomia proposta enfatiza a importância de requisitos funcionais como autenticação contínua, monitoramento, resposta adaptativa e controle de acesso granular. Em paralelo, destaca requisitos não funcionais, tais como escalabilidade, interoperabilidade, resiliência, desempenho e compatibilidade com sistemas legados. A articulação desses requisitos com os demais eixos da taxonomia permite uma avaliação mais holística da maturidade de uma implementação ZTA.

A definição clara dos requisitos funcionais e não funcionais é essencial para orientar o projeto, a implementação e a avaliação de arquiteturas baseadas em *Zero Trust*. Os requisitos funcionais estabelecem os mecanismos fundamentais necessários para assegurar os princípios centrais da ZTA, como autenticação contínua, controle de acesso dinâmico e resposta adaptativa a ameaças. Já os requisitos não funcionais dizem respeito a qualidades do sistema que garantem sua viabilidade prática em ambientes reais, incluindo aspectos como escalabilidade, interoperabilidade e resiliência. A Tabela 2.4 resume os principais requisitos identificados na literatura técnica, com destaque para sua aplicabilidade em ambientes distribuídos, híbridos e dinâmicos.

De modo geral, a taxonomia de Gambo *et al.* oferece uma base sólida para compreender os fundamentos estruturantes da ZTA, permitindo seu desdobramento prático em diferentes domínios tecnológicos. Ao estabelecer relações explícitas entre aplicação, tecnologia e controle, ela contribui para a consolidação da ZTA como um paradigma capaz de responder aos desafios contemporâneos de segurança cibernética.

2.3.4. Comparação entre Modelos Convencionais de Segurança e Arquitetura de Confiança Zero

A evolução das arquiteturas de segurança em redes corporativas acompanha o aumento da complexidade e da dispersão dos ativos digitais. O modelo perimetral, ou castelo-e-fosso (*castle-and-moat*), historicamente predominante, assume confiança implícita nos elementos internos da rede, protegendo-os com firewalls, VPNs e segmentações físicas. Esse paradigma, no entanto, revela-se inadequado frente à popularização

Tabela 2.2. Comparação entre modelos de controle de acesso em diferentes domínios de aplicação.

Critério	Modelo	Descrição
Base de decisão	RBAC	Baseado em papéis atribuídos a usuários; permissões são estáticas e definidas por função
	ABAC	Baseado em múltiplos atributos de usuário, recurso e ambiente; permite decisões dinâmicas
	PBAC	Baseado em políticas declarativas codificadas; permite regras contextuais e sofisticadas
Flexibilidade	RBAC	Baixa; difícil adaptação a mudanças contextuais ou dinâmicas de acesso
	ABAC	Alta; incorpora atributos contextuais e ambientais para decisões mais precisas
	PBAC	Muito alta; políticas podem ser alteradas, versionadas e auditadas conforme necessidade
Adequação a IoT	RBAC	Limitada; requer definição prévia de papéis em ambientes com grande variabilidade
	ABAC	Alta; considera o estado do dispositivo, localização e outros atributos contextuais
	PBAC	Alta; possibilita definição de políticas específicas para dispositivos com base em risco
Aplicação em Nuvem	RBAC	Ampla adoção em sistemas IaaS, mas com limitações em granularidade
	ABAC	Amplo suporte; útil para ambientes com múltiplos *tenants* e requisitos dinâmicos
	PBAC	Ideal; permite governança e automação via "*policy-as-code*"
Redes Críticas	RBAC	Limitada; pouco adaptável a eventos de segurança em tempo real
	ABAC	Boa; permite decisões com base em atributos de missão e contexto operacional
	PBAC	Excelente; oferece rastreabilidade, controle de conformidade e resposta automatizada
Outras ferramentas	RBAC	Active Directory, FreeIPA
	ABAC	AWS IAM, XACML
	PBAC	Open Policy Agent (OPA), Rego, Keycloak Authorization Services

Tabela 2.3. Exemplos de categorias e elementos na taxonomia de ZTA.

Categoria	Elemento	Descrição
Domínio de Aplicação	IoT	Autenticação contínua entre sensores em rede maliciosa
Tecnologia Habilitadora	*Blockchain*	Auditoria imutável e controle distribuído
Mecanismo de Controle	ABAC	Acesso baseado em atributos de usuários e contexto
Requisito Funcional	Autenticação Contínua	Verificação periódica de identidade e postura do dispositivo
Requisito Não Funcional	Escalabilidade	Suporte eficiente a milhares de dispositivos simultâneos

de ambientes multi-*cloud*, à ampliação do trabalho remoto e à sofisticação dos ataques. Nesse cenário, a *Zero Trust Architecture* (ZTA) emerge como resposta, pautada no princípio de que nenhuma entidade (interna ou externa) deve ser considerada confiável por

Tabela 2.4. Requisitos funcionais e não funcionais para a implementação eficaz de uma arquitetura *Zero Trust*.

Tipo	Requisito	Descrição
Funcional	Autenticação Contínua	Verificação permanente da identidade e do estado dos dispositivos antes e durante a sessão
	Controle de Acesso Granular	Definição de permissões com base em atributos dinâmicos de usuário, recurso e contexto
	Monitoramento e Auditoria	Registro e análise contínua de eventos para detectar comportamentos anômalos e violações
	Resposta Adaptativa	Capacidade de reagir automaticamente a eventos de segurança, como revogação de acesso ou isolamento de sessão
Não Funcional	Escalabilidade	Suporte a um grande número de usuários, dispositivos e sessões simultâneas com desempenho aceitável
	Interoperabilidade	Integração com sistemas legados, múltiplos domínios e soluções heterogêneas
	Baixa Latência	Tempo de resposta reduzido para autenticação, autorização e troca de políticas
	Resiliência	Tolerância a falhas e ataques, com mecanismos de recuperação rápida e redundância

padrão [Gupta et al., 2024, Dhiman et al., 2024].

A distinção fundamental entre os dois modelos reside na forma de atribuição de confiança. No modelo tradicional, qualquer entidade inserida na rede passa a ser confiável automaticamente. Em contraste, a ZTA aplica validações contínuas, baseadas em múltiplos fatores como identidade, contexto, estado do dispositivo e comportamento recente. Enquanto o controle de acesso no modelo perimetral é centralizado em dispositivos de borda, na ZTA ele é distribuído, com autenticação e autorização aplicadas em cada tentativa de acesso [Gambo e Almulhem, 2025].

Outro ponto crítico é a gestão da sessão. No modelo legado, a autenticação inicial costuma garantir acesso prolongado, sem revalidação. Na ZTA, as sessões são curtas, monitoradas continuamente e revogadas dinamicamente diante de alterações contextuais ou comportamento anômalo. A detecção de ameaças também difere: enquanto o modelo tradicional se apoia na inspeção em pontos fixos da rede, a ZTA emprega telemetria contínua, análise comportamental e mecanismos automatizados para identificar desvios em tempo real.

A gestão de identidades e dispositivos reforça a maturidade exigida pela ZTA. No modelo tradicional, a autenticação é pontual e generaliza a confiança. Na ZTA, exige-se autenticação multifator, avaliação da postura do dispositivo e validação contextual alinhada às políticas definidas. Essa abordagem mostra-se eficaz para cenários modernos, como acesso remoto, ambientes BYOD (*bring your own device*), infraestrutura em nuvem e sistemas industriais conectados [Gupta et al., 2024, Dhiman et al., 2024].

A Tabela 2.5 resume as diferenças fundamentais entre os modelos. Em suma, a transição do modelo perimetral para o paradigma *Zero Trust* não representa apenas uma atualização tecnológica, mas uma mudança profunda de mentalidade, centrada em verificação contínua, privilégio mínimo e visibilidade completa.

Tabela 2.5. Comparação entre modelo de segurança perimetral e *Zero Trust Architecture* (ZTA).

Critério	Modelo Perimetral	Zero Trust Architecture (ZTA)
Princípio de confiança	Confiança implícita dentro da rede	Nenhuma confiança por padrão; verificação contínua
Local do controle de acesso	Na borda da rede (ex: firewalls)	Distribuído; aplicado em cada solicitação de acesso
Gestão da sessão	Sessões persistentes após login	Sessões curtas, monitoradas e revogáveis dinamicamente
Detecção de ameaças	Inspeção de tráfego na borda	Análise comportamental e telemetria contínua
Identidade e dispositivos	Autenticação pontual com baixa contextualização	Autenticação multifator e avaliação da postura do dispositivo

2.3.5. Confiança Zero em Redes 6G e Ambientes Inteligentes

A evolução para a sexta geração de redes móveis (6G) introduz uma nova era de conectividade hiperconvergente, densamente distribuída e intensiva em dados. Diferentemente das gerações anteriores, a 6G está intrinsecamente associada à integração de ambientes inteligentes (*smart environments*), como cidades autônomas, sistemas industriais avançados, veículos conectados e dispositivos de realidade imersiva. Nesse cenário, a superfície de ataque cresce exponencialmente, enquanto os modelos tradicionais de segurança, centrados em perímetros rígidos, tornam-se obsoletos. A arquitetura *Zero Trust* (ZTA) emerge como um paradigma fundamental para prover segurança contínua, contextual e verificável em redes 6G, operando com base no princípio de "nunca confiar, sempre verificar" [Nahar et al., 2024, Gupta et al., 2024].

As aplicações de ZTA em redes 6G cobrem desde o isolamento de *network slices* até a imposição de políticas dinâmicas de acesso em ambientes altamente distribuídos. Com a virtualização e a segmentação lógica de redes, cada fatia (*slice*) pode representar um domínio de missão crítica, como saúde, transporte ou manufatura. A ZTA, aplicada a esse contexto, assegura que cada entidade (dispositivo, usuário ou serviço) seja autenticada continuamente e que o acesso a qualquer recurso seja mediado por políticas contextuais. Essa arquitetura é particularmente eficaz em cenários que demandam latência ultrabaixa e alta confiabilidade, como controle de veículos autônomos, automação fabril e operações remotas de saúde [He et al., 2022].

A inteligência artificial (IA) torna-se uma aliada estratégica na implementação de motores de decisão em ZTA (*Zero Trust Decision Engines*). Com a coleta contínua de dados de contexto, comportamento de dispositivos, modelos de aprendizado de máquina podem inferir pontuações de risco e ajustar permissões dinamicamente. Técnicas como aprendizado federado, aprendizado por reforço e redes neurais leves permitem processar dados na borda, respeitando restrições de latência e privacidade. No entanto, esses mecanismos enfrentam desafios relevantes, como a redução de falsos positivos, a transparência das decisões automatizadas e a mitigação de viés em conjuntos de dados. Ainda assim, os *Zero Trust Decision Engines* baseados em IA representam um avanço essencial na direção de políticas adaptativas e autônomas.

A segmentação de redes densas, com bilhões de dispositivos interconectados, re-

Tabela 2.6. Integração entre componentes da ZTA, inteligência artificial e elementos da arquitetura 6G.

Componente ZTA	Uso de IA	Aplicação em Redes 6G
Motor de decisão	Análise comportamental e risco adaptativo	Controle de acesso dinâmico em *network slices*
Monitoramento contínuo	Aprendizado federado e detecção de anomalias	Supervisão de dispositivos em ambientes inteligentes
Segmentação lógica	Classificação automática de dispositivos	Microsegmentação em redes densas e móveis
Controle de políticas	Ajuste automatizado com base em contexto e ameaça	Orquestração em múltiplos domínios 6G

quer mecanismos refinados de microsegmentação e isolamento dinâmico. A ZTA fornece meios para classificar dispositivos por atributos de identidade, função, criticidade e confiabilidade, permitindo que comunicações ocorram apenas entre entidades autorizadas dentro de segmentos lógicos. Essa segmentação pode ser adaptada em tempo real com base em mudanças de topologia, comportamento suspeito ou modificações de contexto. Ambientes inteligentes com sensores, atuadores, câmeras, veículos e nós de computação de borda se beneficiam da capacidade da ZTA de orquestrar políticas em múltiplos domínios administrativos, com visibilidade e controle unificados.

A integração entre componentes da ZTA e as redes 6G é explicitada na Tebala 2.6. Apesar das vantagens promissoras, o uso de ZTA em redes 6G impõe desafios substanciais. A escalabilidade de mecanismos de autenticação e verificação contínua em ambientes massivos ainda requer avanços significativos. A latência na aplicação de decisões de acesso baseadas em IA precisa ser rigorosamente controlada para não comprometer requisitos de tempo real. Adicionalmente, a interoperabilidade entre diferentes domínios administrativos e fornecedores de rede ainda é uma barreira à aplicação prática de ZTA em escala global. Entre as direções futuras, destaca-se a incorporação de *blockchain* para auditoria distribuída, o uso de identidade descentralizada para autenticação federada, e a formalização de políticas como código para validação automática e verificabilidade.

A integração entre a *Zero Trust Architecture* (ZTA) e tecnologias *blockchain* surge como uma estratégia promissora para reforçar os pilares de segurança, auditabilidade e descentralização em ambientes distribuídos. Enquanto a ZTA opera sob o princípio da verificação contínua e do privilégio mínimo, *blockchain* fornece uma base imutável, confiável e auditável para registro e execução de políticas de acesso. A união dessas abordagens é particularmente relevante em contextos como Internet das Coisas (IoT), redes federadas, sistemas multi-organizacionais e infraestruturas críticas, nos quais a confiança entre entidades não pode ser presumida e a necessidade de visibilidade é essencial [Gupta et al., 2024, Pooja e Chandrakala, 2024].

No âmbito da ZTA, os componentes centrais do controle de acesso, *Policy Enforcement Point* (PEP), *Policy Decision Point* (PDP) e *Policy Administrator* (PA), dependem da correta autenticação, autorização e avaliação de contexto para decidir sobre a permissão de acesso a recursos sensíveis. A introdução da *blockchain* permite descentralizar parte dessas decisões, registrando regras e auditorias em contratos inteligentes. Com isso, é possível automatizar a aplicação de políticas de forma verificável e garantir que modifi-

cações nos critérios de acesso sejam historicamente rastreáveis. O uso de *blockchain* com suporte a contratos inteligentes, como Ethereum e Hyperledger Fabric, permite modelar regras de acesso granulares com base em atributos de identidade, localização, horário ou nível de sensibilidade do recurso [Tuler De Oliveira et al., 2022, He et al., 2022].

A tecnologia *blockchain* pode atuar como repositório confiável para identidades digitais descentralizadas (DID), *tokens* de acesso e eventos de auditoria [de Oliveira et al., 2024]. Em um sistema tradicional, essas informações ficariam armazenadas em servidores centrais, sujeitos a comprometimentos internos ou externos. Com a descentralização, cada entidade pode controlar sua identidade por meio de chaves criptográficas, enquanto a verificação de permissões ocorre por meio de consenso ou validação distribuída. Tal abordagem é compatível com mecanismos de controle de acesso acesso fino (*fine-grained*) como o ABAC (*Attribute-Based Access Control*), permitindo que decisões sejam tomadas com base em múltiplos atributos dinâmicos registrados na rede *blockchain* [Nahar et al., 2024].

Esse controle de acesso fino pode ser ampliado com contratos inteligentes que integram políticas contextuais complexas, como restrições geoespaciais (*geofencing*), análise de tempo de acesso, associação a grupos temporários ou níveis de risco atribuídos por algoritmos de aprendizado de máquina [Tuler De Oliveira et al., 2022]. Em arquiteturas ZTA integradas com *blockchain*, a decisão de acesso não depende apenas da verificação tradicional em um servidor central, mas da avaliação distribuída de múltiplos fatores, com evidência criptográfica e transparência.

A aplicação prática dessa integração já aparece em sistemas de saúde, consórcios industriais e ambientes acadêmicos, nos quais dados sensíveis precisam ser compartilhados entre instituições com diferentes níveis de autoridade e protótipos de redes B5G e 6G. Em plataformas de telemedicina, por exemplo, é possível garantir que apenas profissionais devidamente autenticados e com atributos específicos (ex: especialidade, vínculo institucional, jurisdição) acessem prontuários protegidos, sendo cada requisição registrada em *blockchain* para futura auditoria. Similarmente, em colaborações científicas, o uso de ZTA com *blockchain* permite que permissões de leitura, modificação e reuso de *datasets* sejam atribuídas de forma condicional, auditável e reversível [de Oliveira et al., 2024].

Contudo, a latência natural de redes *blockchain* pode ser um entrave para decisões em tempo real. Soluções como o uso de infraestruturas de *blockchain* permissionadas, uso de canais privados e mecanismos de cache podem mitigar esse impacto. A complexidade da governança das políticas e da atualização dos contratos inteligentes exige mecanismos robustos de versionamento, autenticação de administradores e validação formal de regras. Apesar dessas limitações, a sinergia entre ZTA e *blockchain* representa um avanço notável na construção de ecossistemas seguros, confiáveis e interoperáveis para redes 6G. Ao permitir um controle de acesso fino com rastreabilidade total, essas tecnologias reforçam os princípios de segurança por *design*, transparência e descentralização, fundamentais para os cenários modernos de computação em nuvem, IoT e cooperação federada entre organizações.

2.4. Implementação da Arquitetura de Confiança Zero

Existem múltiplas possibilidades para a implementação de uma arquitetura de confiança zero, que se diferenciam pelos componentes empregados e pela origem das políticas adotadas. Todas as abordagens contemplam os princípios descritos anteriormente, embora um ou dois deles possam assumir papel preponderante na formulação das políticas. Para obter uma solução abrangente, recomenda-se integrar elementos das três abordagens: governança de identidade aprimorada, microssegmentação lógica e segmentação baseada em rede [Stafford, 2020].

Na abordagem de **Governança de Identidade Aprimorada**, a identidade e os atributos são a base para a formulação de políticas de acesso. Cada solicitação a recursos corporativos é avaliada com base nos privilégios concedidos a uma dada identidade e fatores complementares, como o tipo de dispositivo, o estado e a rede, ajustam o nível de confiança resultante, podendo, por exemplo, restringir o acesso a subconjuntos de dados ou impor verificações adicionais. Embora o acesso à rede seja liberado a todos os ativos por padrão, o acesso a recursos críticos permanece condicionado à validação da identidade e a infraestrutura deve monitorar continuamente movimentações laterais ou tentativas de negação de serviço para conter ameaças internas. Dispositivos IoT, por exemplo, operam em condições de alta exposição, o que facilita a ação de invasores em busca de informações para viabilizar seus ataques. Para neutralizar essa ameaça, a auditoria regular da rede se torna uma solução crucial. Esse processo é um dos pilares da governança de identidade, garantindo a eficácia das políticas de autenticação e autorização e fortalecendo a segurança como um todo [Rizvi et al., 2023].

Alguns trabalhos avaliam o uso de Identidade Descentralizada para cenários *Zero Trust* em que diferentes domínios precisam compartilhar recursos com segurança. Em vez de depender de um servidor central, cada participante gera seu próprio identificador descentralizado (DID) e recebe credenciais verificáveis, guardadas em uma cadeia de blocos que funciona como registro imutável de confiança. Com isso, o controle da identidade passa a ser do utilizador (*Self-Sovereign Identity*); os atributos só são revelados quando necessário, e a autenticação/autorização é realizada com provas criptográficas que não expõem dados sensíveis [Bernabé Murcia et al., 2025]. O ciclo de vida da identidade consiste em cinco estágios essenciais [Aboukadri et al., 2024]: Provisionamento, Autenticação, Autorização, Manutenção e Desprovisionamento.

- **Provisionamento** envolve a geração de um identificador distinto, a criação de credenciais e a documentação do perfil de atributos do sujeito. Esses atributos podem incluir elementos como localização geográfica, endereço de e-mail ou características específicas de contexto.

- **Autenticação** consiste na confirmação que um dado usuário é realmente quem diz ser. Existem diversos tipos de autenticação, como baseadas em senhas, dois fatores, biométricas, federada ou *token*. Um dos requisitos para implementação da arquitetura de confiança zero, é que este processo seja feito de forma contínua.

- **Autorização** consiste em conceder ou negar permissões a um usuário, grupo de usuários ou uma entidade (como um serviço ou aplicação) para realizar ações espe-

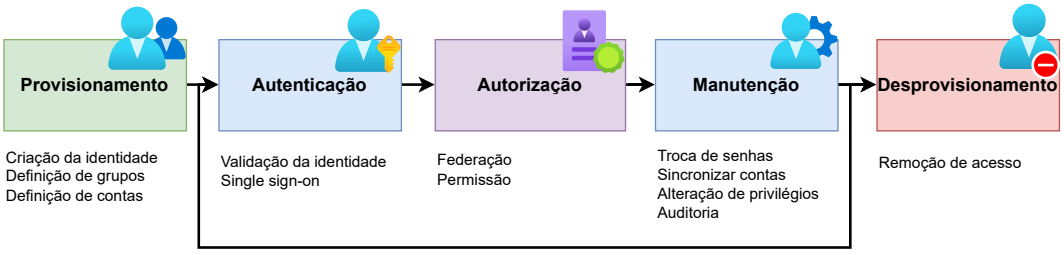


Figura 2.3. Ciclo de gerenciamento de identidade e acesso. O fluxo incia-se com o Provisionamento (criação de identidade e contas), seguido pela Autenticação (validação da identidade), Autorização (definição de permissões), Manutenção (atualização de senhas, sincronização e auditoria), e termina com o Desprovisionamento (remoção de acesso). Esse processo garante a segurança e o controle adequado sobre o acesso dos usuários ao longo de seu ciclo de vida no sistema.

cíficas. Nesse sentido, a autorização determina **o** que um usuário ou entidade pode fazer após a autenticação.

- **Manutenção** consiste no processo que lida com o fato de que as identidades são dinâmicas e podem sofrer alterações ao longo do tempo, seja devido à evolução das características do usuário ou a mudanças nas demandas de negócios, o que pode levar a transformações na identidade.

A **micro-segmentação** por sua vez, é uma abordagem de segurança que divide a rede em segmentos isolados e granulares, permitindo a aplicação de políticas de segurança específicas para cada carga de trabalho (*workload*). Por reduzir significativamente a superfície de ataque, esse conceito além de restringir o tráfego **leste-oeste** (movimentação lateral), também impõe controles sobre o tráfego **norte-sul**, limitando o tráfego entre usuários, aplicações e servidores. Essa granularidade não apenas eleva o nível de segurança, mas também aumenta a precisão de ferramentas de monitoramento, como as de detecção de anomalias, que operam com mais eficácia em ambientes menores e mais homogêneos. Impulsionada pelo avanço das Redes Definidas por *Software* (SDN) e dos centros de dados definidos por *software* (*Software Defined Data Centers* - SDDC), a micro-segmentação tornou-se peça-chave para conciliar segurança e flexibilidade [Mujib e Sari, 2020]. Em arquiteturas tradicionais, a segurança é concentrada em *firewalls* de borda. Contudo, uma vez que um atacante ultrapassa essa barreira de perímetro, ele ganha liberdade para se movimentar lateralmente pela rede interna e escalar o ataque. A micro-segmentação soluciona essa vulnerabilidade ao este para dentro do data center, controlando o tráfego leste-oeste, ou seja, a comunicação entre os componentes internos da infraestrutura [Mämmelä et al., 2016]. O modelo de serviços de uma rede micro-segmentada pode ser visualizado na Figura 2.4, sendo um componente lógico fundamental na arquitetura de segurança de confiança zero, baseando-se no princípio do “privilégio mínimo"para conceder acesso apenas quando estritamente necessário [Sheikh et al., 2021]. Essa abordagem consiste em dividir a rede em pequenas zonas isoladas, criando perímetros de segurança em torno de recursos e aplicações onde cada zona exige autorização específica para acesso. Dessa forma, a micro-segmentação permite implementar de

maneira granular e eficaz os tradicionais mecanismos de segurança, como Autenticação, Autorização e Auditoria, garantindo que apenas usuários e sistemas autenticados possam se comunicar.

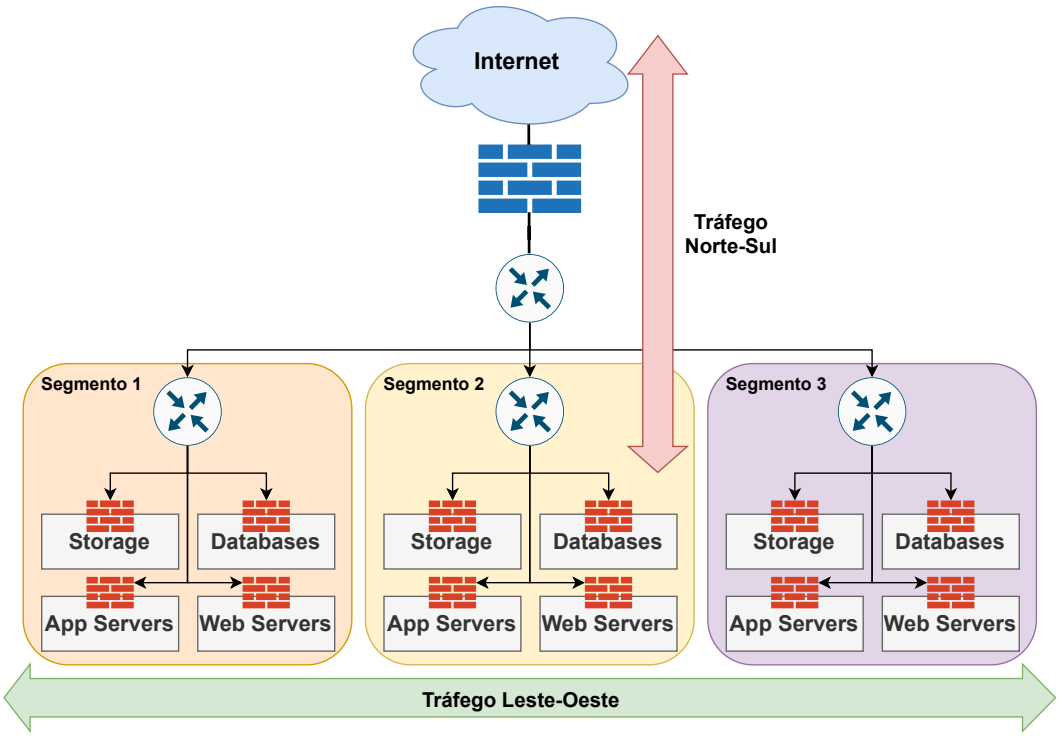


Figura 2.4. Ilustração de uma arquitetura de rede micro-segmentada, destacando a separação entre diferentes segmentos. Cada segmento é composto por componentes, como servidores de armazenamento (Storage), bancos de dados (Databases), servidores de aplicação (App Servers) e servidores web (Web Servers). Os segmentos são protegidos por *firewalls*, que controlam e limitam o tráfego interno (leste-oeste) entre os componentes do data center, bem como o tráfego externo (norte-sul) entre a rede interna e a internet. Essa abordagem de micro-segmentação visa aumentar a segurança ao isolar os diferentes segmentos e restringir a movimentação lateral de possíveis ameaças dentro da rede.

Para cargas de trabalho em contêineres, que são efêmeras e distribuídas por natureza, a micro-segmentação é crucial. Avanços foram feitos no suporte à micro-segmentação para esses ambientes, impulsionados em grande parte pela tecnologia eBPF (***extended Berkeley Packet Filter***) [Chang e Mukherjee, 2024]. O eBPF permite a execução de programas em um ambiente restrito dentro do *kernel* do Linux, fornecendo controle sobre o tráfego de rede e as chamadas de sistema, sem a necessidade de alterações no código da aplicação. Soluções de segurança nativas da nuvem aproveitam o eBPF para implementar a micro-segmentação, detre elas:

- **Cilium** utiliza o eBPF para criar políticas de rede independentes nas camadas 3 e 4 e segurança nas camadas 4 a 7 da pilha TCP/IP [Koukis et al., 2024]. O Cilium

estabelece uma **identidade estática** para cada carga de trabalho com base em seus metadados (como labels do Kubernetes), em vez de depender de endereços IP, que são variáveis em ambientes de contêineres. Com essa identidade, o Cilium impõe políticas de segurança diretamente no kernel, controlando quais serviços podem se comunicar entre si [Chang e Mukherjee, 2024].

- **Calico**, assim como o Cilium, é uma solução para contêineres que utiliza o kernel Linux e o protocolo BGP para criar uma rede de alto desempenho. Uma das suas principais características é o gerenciamento de endereços IP (IPAM) e aplicação de políticas de segurança granulares. Por padrão, o Calico emprega tunelamento de sobreposição (*overlay*) IP-em-IP [Nagendra e Hemavathy, 2023].

- **Flannel** é um plugin CNI *(Container Network Interface)* que simplifica a comunicação em rede orquestração de contêineres como o Kubernetes. O objetivo também é criar uma rede virtual sobre a infraestrutura (*overlay*), utilizando por padrão o encapsulamento VXLAN, mas também suporta alternativas como IPIP, host-gw e UDP [Koukis et al., 2024]. Para o gerenciamento da rede, o Flannel implementa um agente binário, o **flanneld**, na forma de um DaemonSet. Esse componente é responsável por alocar uma sub-rede de IP única para cada host, garantindo que não haja conflitos de endereço. A distribuição da configuração de rede e o roteamento do tráfego são coordenados através de um armazenamento central de chave-valor, como o etcd [Nagendra e Hemavathy, 2023].

Nesse sentido, a micros-segmentação substitui o modelo de segurança de confiança implícita pelo de acesso explícito, baseado em identidade e contexto. Ao criar zonas de controle menores, é possível aplicar políticas de segurança precisas, coletar métricas detalhadas e isolar incidentes rapidamente, o que limita o impacto de ameaças como o *ransomware*.

A última abordagem para a implementação da arquitetura de confiança zero consiste em considerar as redes não apenas como canais de comunicação, mas como elementos estratégicos no gerenciamento distribuído de dados. Essa perspectiva abrange atividades como acesso, processamento, transferência e armazenamento de informações diretamente na rede, reforçando seu papel central na arquitetura e na segurança das operações. Espera-se, portanto, que as estruturas de segurança de próxima geração, baseadas nesse modelo, garantam a proteção dos dados ao longo de todo o seu ciclo de vida [Ramezanpour e Jagannath, 2022]. Essa abordagem pode ser viabilizada por meio de uma rede *overlay*, geralmente operando na camada 7, mas que também pode ser configurada em níveis inferiores da pilha OSI. Tais soluções são frequentemente denominadas *Software Defined Perimeter* (SDP) e, em muitos casos, integram conceitos de *Software Defined Network* (SDN) e *Intent Based Network* (IBN) [Stafford, 2020]. Além de redefinir o papel das redes, o SDP atua como um mecanismo de segurança capaz de lidar com aspectos críticos como autenticação, controle de acesso granular e auditoria detalhada. Ele garante que apenas usuários autorizados tenham acesso aos recursos da rede, alinhando-se aos princípios fundamentais da confiança zero [Shen e Shen, 2024]. Utilizando políticas dinâmicas, o SDP estabelece túneis de comunicação seguros e isolados, ocultando a infraestrutura da rede e limitando o acesso exclusivamente aos recursos necessários, como

servidores e aplicações. Essa estratégia não apenas reforça a segurança, como também reduz a superfície de ataque, tornando a infraestrutura mais resiliente e adequada às complexidades inerentes das redes de próxima geração. Quando implementado na camada de aplicação. Existem algumas variações para iniciar a implementação [Stafford, 2020], podendo destar as baseadas em **Agente/*Gateway***, **Enclave** e **Portal**.

Na implementação baseada em **agente e *gateway***, o *Policy Enforcement Point* (PEP) é dividido em dois componentes: o agente, instalado no dispositivo (como um laptop), e o *gateway*, posicionado antes do recurso que se deseja o acesso, como uma aplicação ou banco de dados, por exemplo. O agente é responsável por interceptar e redirecionar o tráfego de rede do usuário para o *gateway* apropriado. Por sua vez, o *gateway* funciona como um *proxy* reverso, garantindo que toda a comunicação com o recurso passe obrigatoriamente por ele. O processo tem início quando uma entidade (usuário ou aplicação) tenta acessar um recurso corporativo. A solicitação é capturada pelo agente no dispositivo e enviada ao *Policy Administrator* (PA), um componente do plano de controle da arquitetura. Este por sua vez consulta o *Policy Engine* (PE), que avalia se o acesso cumpre as regras de segurança vigentes. Caso a solicitação seja aprovada, o PA instrui o agente e o *gateway* a estabelecerem um canal de comunicação seguro e direto. Para isso, são configurados parâmetros como endereços IP, portas, *tokens* de sessão e outros elementos de segurança que asseguram a confidencialidade e a integridade. Com o canal estabelecido, os dados trafegam criptografados diretamente entre o agente e o *gateway*, operando exclusivamente no plano de dados. A conexão é encerrada automaticamente ao final da sessão ou caso ocorra um evento de segurança detectado pelo PA, como a expiração do tempo de uso ou uma falha de reautenticação. Este modelo é ideal para organizações que gerenciam seus próprios dispositivos e não permitem o uso de aparelhos pessoais (BYOD). É também altamente eficaz em ambientes com recursos discretos que exigem comunicações estritamente controladas, representando uma implementação clássica do modelo cliente-servidor [Stafford, 2020]. A Figura 2.5 apresenta o fluxo dessa implementação.

A implementação baseada em **Enclave**, ilustrada na Figura 2.6, é uma variação do modelo agente/gateway. Nesse caso, o *gateway* encontra-se na fronteira de um enclave de recursos, dispensando a integração ponto a ponto. Por outro lado, forma-se uma zona implícita de confiança entre o *gateway* e os recursos, suprimindo as informações contextuais mais granulares oferecidas pelo primeiro modelo [Alevizos et al., 2022]. Este modelo pode ser útil para organizações que utilizam microsserviços baseados em nuvem para um único processo de negócios na qual toda a nuvem privada está localizada atrás de um *gateway* [Stafford, 2020]. Como desvantagem é que o *gateway* protege um conjunto de recursos, podendo não ser capaz de proteger cada recurso individualmente. Isso pode permitir que entidades acessem recursos aos quais não têm privilégios de acesso, caracterizando assim uma possível falha.

O modelo baseado em Portal, não requer a instalação de agentes nos dispositivos, sendo todas as solicitações de acesso encaminhadas a um portal [Tsai et al., 2024], conforme Figura 2.7. O principal benefício desse modelo em relação aos outros, reside no fato de não ser necessário a instalação de nenhum componente, sendo mais flexível em casos que se adote políticas de BYOD. No entanto, informações limitadas podem ser inferidas a partir dos dispositivos que solicitam acesso. Este modelo só consegue escanear e

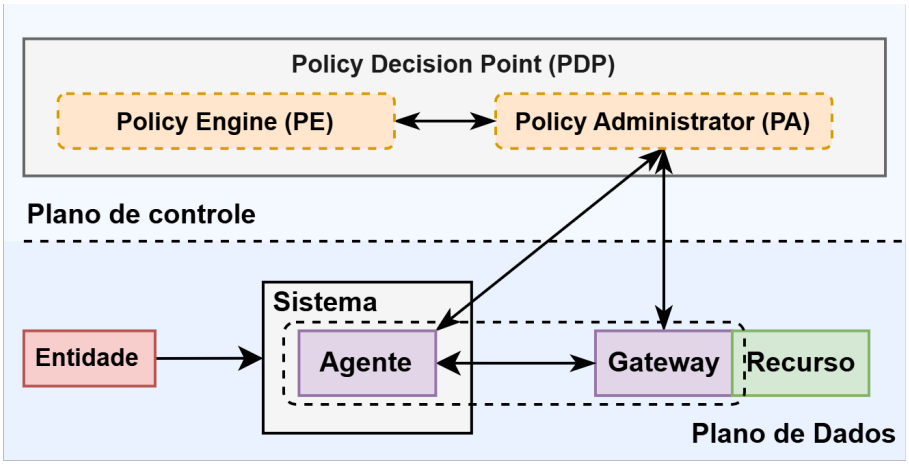


Figura 2.5. Arquitetura baseada no modelo agente-gateway. Esse modelo está dividido em dois planos: o Plano de Controle, onde se encontra o Ponto de Decisão de Política (PDP), composto pelo Policy Engine (PE) e pelo Policy Administrator (PA); e o Plano de Dados, que contém o Sistema, formado por um Agente e um Gateway, além do Recurso a ser acessado. A Entidade realiza uma solicitação, que é avaliada pelo Agente e encaminhada ao Gateway. As decisões de controle são tomadas com base nas políticas definidas no PDP.

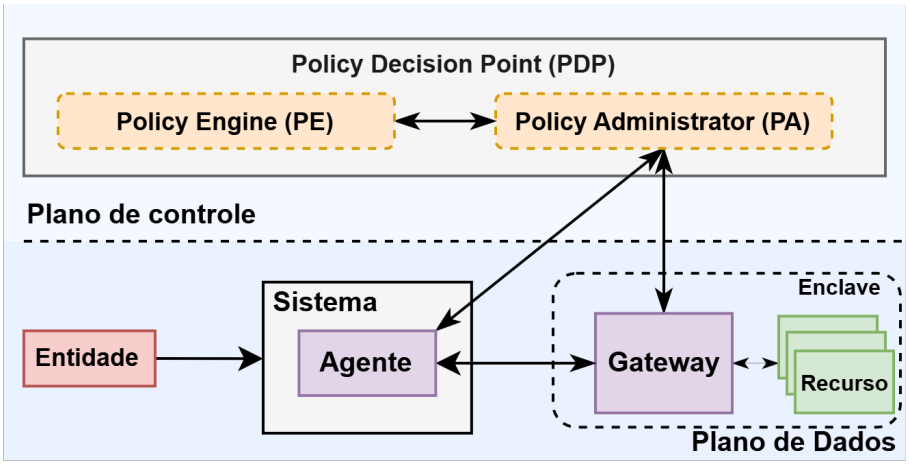


Figura 2.6. Arquitetura de controle de acesso com enclave de recursos. O Plano de Controle, onde atua o Ponto de Decisão de Política (PDP), composto pelo Policy Engine (PE) e pelo Policy Administrator (PA), e o Plano de Dados, onde a Entidade interage com o Sistema. O Sistema contém um Agente que se comunica com o Gateway, localizado em um Enclave junto aos Recursos protegidos. O PA aplica as decisões do PE, controlando o acesso por meio do Gateway conforme as políticas previamente definidas.

analisar ativos e dispositivos quando eles se conectam ao portal PEP e pode não ser capaz de monitorá-los continuamente em busca de atividades suspeitas, vulnerabilidades não corrigidas e configurações apropriadas [Stafford, 2020]. Um ponto de atenção, é o fato do portal se tornar um potencial alvo de ataques DDoS. Um único ponto de falha, pode interromper todos os serviços da organização. Nesse sentido, é mandatório que o portal seja bem provisionado para fornecer disponibilidade contra um ataque DoS ou interrupção da rede.

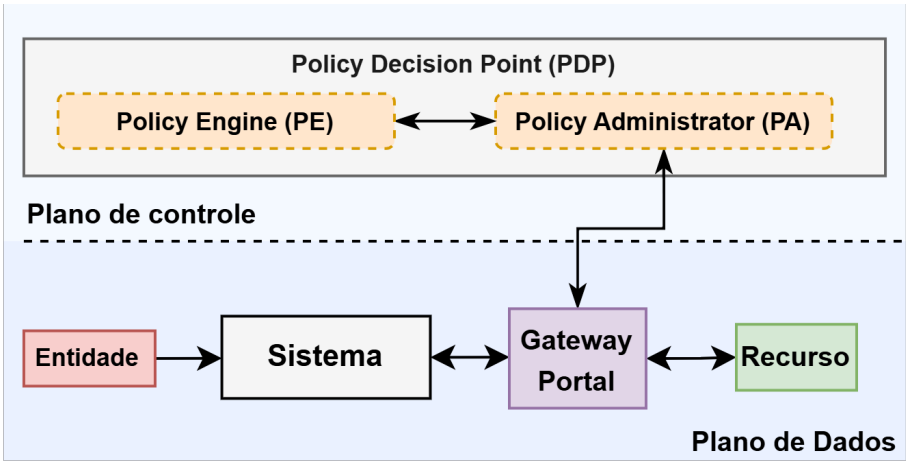


Figura 2.7. Arquitetura de controle de acesso baseado em portal. O Policy Decision Point (PDP), composto pelo Policy Engine (PE) e pelo Policy Administrator (PA), toma decisões com base em políticas definidas. A Entidade realiza uma solicitação ao Sistema, que interage com o Gateway Portal. O acesso ao Recurso é autorizado ou negado conforme as instruções do PA, que aplica decisões do PE.

Um dos principais desafios na implementação desse paradigma pelas organizações encontra-se na dificuldade de criar processos e políticas que atendam aos requisitos da arquitetura de confiança zero. Para que a implementação seja eficaz, todos os recursos que se deseja proteger precisam estar previamente mapeados o que exige um grande esforço dependendo do tamanho da organização. Outro aspecto desafiador na implementação é o gerenciamento de identidades. Em modelos tradicionais, esse gerenciamento enfrenta um problema central: dependência de uma terceira parte confiável. Esses sistemas foram desenvolvidos para facilitar a administração das bases de usuários pelas empresas, mas acabam limitando o poder dos próprios indivíduos sobre seus dados. Os usuários muitas vezes não sabem quais informações estão sendo coletadas, como são utilizadas ou com quem são compartilhadas. Esse desequilíbrio compromete a privacidade, a segurança e a autonomia digital das pessoas. Nesse contexto, a Identidade Autossoberana (*Self-Sovereign Identity* - SSI) surge como uma abordagem alternativa [Satybaldy et al., 2024]. Baseada em tecnologias descentralizadas, como *blockchain*, a SSI permite que os próprios usuários gerenciem suas identidades digitais. Ao inverter a lógica tradicional, a SSI devolve aos indivíduos a posse e o controle de suas informações, promovendo um ambiente digital mais seguro, transparente e centrado no usuário.

Mukta *et al.* propõem uma arquitetura de controle de acesso descentralizada e orientada por princípios de confiança zero, que integra identidades autossoberanas (*Self-Sovereign Identity* - SSI), identificadores descentralizados (*Decentralized Identifier* - DID), e controle de acesso baseado em capacidades (CapBAC). A proposta visa resolver as limitações de modelos tradicionais de controle de acesso, como a centralização da gestão de identidades e a ausência de verificação contínua de confiança, especialmente em ambientes dinâmicos e distribuídos. Com SSI e DIDs, os usuários mantêm controle direto sobre suas identidades e podem comprovar atributos sem revelar informações desnecessárias, enquanto o uso de CapBAC permite a delegação de acessos de forma segura, rastreável e com granularidade [Mukta et al., 2025].

2.5. Plataforma OpenZiti para Arquitetura de Confiança Zero: Exemplo Prático

OpenZiti[8] é um projeto de código aberto que permite incorporar redes de confiança zero diretamente em aplicações. Sua principal proposta consiste na simplificação da gestão de redes, ao elevar a identidade do usuário ou da aplicação, em vez do endereço IP, ao papel central da infraestrutura de rede. Essa abordagem permite reduzir complexidades inerentes das redes, como IPs estáticos, sub-redes, NAT e firewalls, que frequentemente se tornam obstáculos, pois necessitam de conhecimento especializado. A arquitetura do OpenZiti é composta por um controlador, roteadores de borda (*edge routers*) e roteadores *fabric* (*fabric routers*), que trabalham em conjunto para formar a estrutura de confiança zero. O projeto oferece SDKs que permitem a integração direta em aplicações, além de disponibilizar aplicativos de tunelamento que estendem o acesso a aplicações que não podem ser modificadas para incorporar os SDKs. Os principais componentes do OpenZiti a nível de aplicação:

- **Controlador Ziti** (*Ziti Controller*) funciona como o componente central, responsável pelo plano de controle da rede. É o primeiro elemento a ser configurado e iniciado, pois armazena e gerencia todas as informações de configuração dos roteadores, serviços, políticas e identidades. Uma das funções do controlador é a capacidade de estabelecer e atualizar dinamicamente as rotas para os serviços em nome dos clientes e roteadores, adaptando-se a mudanças na topologia da rede ou otimizando para maior eficiência.

- **Roteador de borda** (*edge router)* é o principal ponto de entrada e saída de tráfego para clientes, como aplicações com SDKs ou *tunnelers*, permitindo que acessem ou disponibilizem serviços de forma segura. Cada roteador possui identidade criptográfica própria e é previamente registrado no *controller*, e, quando designado como Edge Router, passa a autenticar, criptografar e autorizar todo o tráfego conforme políticas definidas, além de participar do roteamento interno, interceptar fluxos TCP/UDP provenientes de interfaces LAN ou WAN, e encaminhar tráfego para hosts sem *tunnelers*.

Esses são os principais componentes necessários para estabelecer uma rede com aplicações baseadas no modelo de confiança zero. Uma vez implementados, passa-se à próxima etapa: a configuração lógica do ambiente, para a qual é necessário compreender os conceitos de Identidades, Serviços e Políticas[9]. As **Identidades** representam *endpoints* individuais com a capacidade de estabelecer conexões. A segurança dessas interações é garantida através de autenticação mútua, empregando certificados X.509. Cada identidade é intrinsecamente vinculada ao seu certificado, tipicamente por meio da sua impressão digital. Ao iniciar uma conexão, o cliente na borda apresenta seu certificado. A infraestrutura de rede, então, realiza a validação do certificado. Este processo não apenas autentica a identidade, confirmando sua veracidade, mas também a autoriza, definindo os limites de

[8]Disponível em `https://openziti.io/`.

[9]Disponível em `https://openziti.io/docs/learn/introduction/components`.

seus privilégios. Somente após essa validação e autorização, os serviços específicos para os quais a identidade possui permissão são disponibilizados.

O **Serviço** é a unidade lógica que encapsula qualquer recurso acessível por clientes, definido por uma identidade, caracterizada por um nome e/ou certificado, ao invés de depender de conceitos de infraestrutura como DNS ou endereços IP; esse modelo proporciona um *namespace* praticamente ilimitado para identificação e abstrai do usuário final a complexidade do encaminhamento, pois o serviço apenas declara o roteador de borda a utilizar para a saída do tráfego da rede, podendo esse roteador coincidir ou não com o de entrada: se forem distintos, a própria rede realiza o roteamento interno necessário, de modo que basta ao usuário referir-se ao serviço para que todo o restante seja tratado automaticamente.

As **Políticas** definem as regras que controlam como identidades, serviços e roteadores de borda interagem entre si. Para que uma identidade possa acessar um determinado serviço, é necessário que haja uma associação explícita entre ambos. Como todo o tráfego destinado a um serviço é encaminhado por meio de um ou mais roteadores de borda, tanto a identidade quanto o serviço precisam ter permissão para utilizar o mesmo roteador, ou um conjunto comum de roteadores, garantindo, assim, que a comunicação ocorra de forma segura e controlada. Na plataforma OpenZiti existem as seguintes políticas:

- As **Políticas de Serviço** (*Service Policies*) definem a relação entre identidades e serviços dentro do OpenZiti. Nesse sentido, pode ser composta por um conjunto de serviços (aplicações) e um grupo de identidades. Quando um serviço é adicionado a uma Política de Serviço, todas as identidades associadas àquela política passam a ter acesso a esse serviço. Da mesma forma, ao incluir uma identidade em uma Política de Serviço, essa identidade adquire permissão para acessar os serviços definidos na política. Essas políticas são ainda responsáveis por determinar quais identidades podem acessar (*dial*) um serviço quanto quais podem fornecer ou hospedar (*bind*) esse serviço. No entanto, cada Política de Serviço pode permitir apenas um desses tipos de acesso por vez ou *dial* ou *bind*, nunca ambos simultaneamente.
- As **Políticas de Roteador de Borda** (*Edge Router Policies*) são responsáveis pelo mapeamento entre identidades e roteadores de borda. Cada política pode ser composta por um grupo de roteadores de borda e um grupo de identidades. Ao adicionar um roteador a uma Política de Roteador de Borda, todas as identidades associadas a essa política passam a ter acesso ao roteador em questão. Da mesma forma, ao incluir uma identidade em uma Política de Roteador de Borda, essa identidade adquire permissão para acessar os roteadores definidos na política.
- As **Políticas de roteador de borda de serviço** (*Service Edge Router Policie*s) atuam para mapear grupos de serviços existentes com um ou mais roteadores de borda. Ao adicionar um roteador de borda a esta classe de política, concederá os serviços associados a essa mesma política, ao roteador de borda e vice versa.

Para atender à diversidade de necessidades organizacionais e possibilitar uma transição gradual para o modelo de confiança zero, o OpenZiti oferece três modelos distintos de acesso:

- ***Zero Trust Application Access* (ZTAA)** é o modelo mais abrangente, como foco na segurança das comunicações entre aplicações. É projetado para simplificar implementações em ambientes multi-cloud , eliminando a confiança implícita na rede subjacente, incluindo a rede do host.

- ***Zero Trust Host Access* (ZTHA)** é o modelo que foca em proteger as comunicações a nível de *host*. A integração com soluções legadas ou existentes é viabilizada por meio do OpenZiti Tunneler, que encapsula aplicações tradicionais sem exigir modificações em seu código. Essa abordagem elimina a necessidade de confiar na rede subjacente, ao mesmo tempo em que garante que os firewalls — tanto de rede quanto do sistema operacional — operem em modo de negação por padrão, permitindo apenas o tráfego explicitamente autorizado. No entanto, neste modelo, apenas a rede do host é considerada uma zona de rede confiável.

- ***Zero Trust Network Access* (ZTNA)** tem como objetivo proteger o acesso a aplicações e serviços dentro de uma zona de rede segura. Seu funcionamento ocorre utilizando um OpenZiti Router. Embora o *firewall* de rede opere em modo de negação por padrão, os *firewalls* do sistema operacional ainda requerem regras de porta de entrada por serviço. Esse modelo permite o acesso de confiança zero em dispositivos que não podem instalar um OpenZiti Tunneler. Isso o torna especialmente adequado para organizações que estão no início de sua jornada rumo à confiança zero, mas que já enfrentam demandas urgentes por segurança reforçada e controle de acesso granular.

Com os conceitos previamente explorados, é possível iniciar a implementação da infraestrutura OpenZiti. No presente exemplo prático, é abordada a configuração utilizando o modelo ZTNA. Para isso, foram provisionados servidores em nuvens públicas utilizado o sistema operacional Debian 12, que desempenharão as funções de controlador e roteador de borda.

2.5.1. Configuração do Controlador

O primeiro passo consiste em instalar o binário do controlador OpenZiti, conforme o Código Fonte 2.1. Uma vez instalados é necessário configurar o arquivo com variáveis de ambiente boostrap.env e incluir as informações conforme o Código Fonte 2.2. É importante ressaltar que as configurações dos registros de DNS e regras de *firewall* dos servidores já estejam previamente criados.

Código Fonte 2.1. Instalação do controlador OpenZiti.

```
1 # Instalação do Controlador OpenZiti
2 apt update && apt upgrade -y
3 curl -sS https://get.openziti.io/install.bash | sudo bash -s
   ↪ openziti-controller
4 vim /opt/openziti/etc/controller/bootstrap.env
```

Código Fonte 2.2. Configuração do bootstrap do Controlador OpenZiti.

```
# the controller's permanent FQDN (required)
ZITI_CTRL_ADVERTISED_ADDRESS='controller.zetin.uff.br'

# the controller's advertised and listening port (default: 1280)
ZITI_CTRL_ADVERTISED_PORT='1280'

# name of the default user (default: admin)
ZITI_USER='admin'

# password will be scrubbed from this file after
# creating default admin during database initialization
ZITI_PWD='PASSWORD'

# additional arguments to: ziti create config controller
ZITI_BOOTSTRAP_CONFIG_ARGS=''
```

Para configurar as variáveis de ambiente do controlador é necessário gerar o arquivo completo e iniciar o serviço. O Código Fonte 2.3 apresenta os comandos para executar essas tarefas.

Código Fonte 2.3. Inicialização do Controlador OpenZiti.

```
/opt/openziti/etc/controller/bootstrap.bash
systemctl enable --now ziti-controller.service
```

Inicializado o controlador, todo o gerenciamento pode ser realizado por meio da interface de linha de comando (*Command-Line Interface* - CLI). No entanto, também é possível instalar uma interface gráfica, que torna o processo de administração mais intuitivo e acessível. O Código Fonte 2.4 apresenta os comandos necessários para execução desta tarefa.

Código Fonte 2.4. Configuração da interface gráfica.

```
mkdir -p /var/lib/ziti-controller/
wget https://github.com/openziti/ziti-console/releases/latest/download
 ↪  /ziti-console.zip
unzip -d /var/lib/ziti-controller/zac ./ziti-console.zip
```

Após a instalação, é necessário adicionar o caminho para o código no arquivo de configuração. Edite o arquivo `/var/lib/ziti-controller/config.yml` e adicione as seguintes linhas ao final do bloco `web` como ilustrado no Código Fonte 2.5. Com o controlador e a interface gráfica configurados, é possível realizar a autenticação na plataforma e iniciar todo processo de configuração de identidades, serviços e do roteadores de borda, conforme ilustrado na Figura 2.8.

Código Fonte 2.5. Arquivo de configuração da interface gráfica.

```
      - binding: zac
        options:
          location: /var/lib/ziti-controller/zac
          indexFile: index.html
```

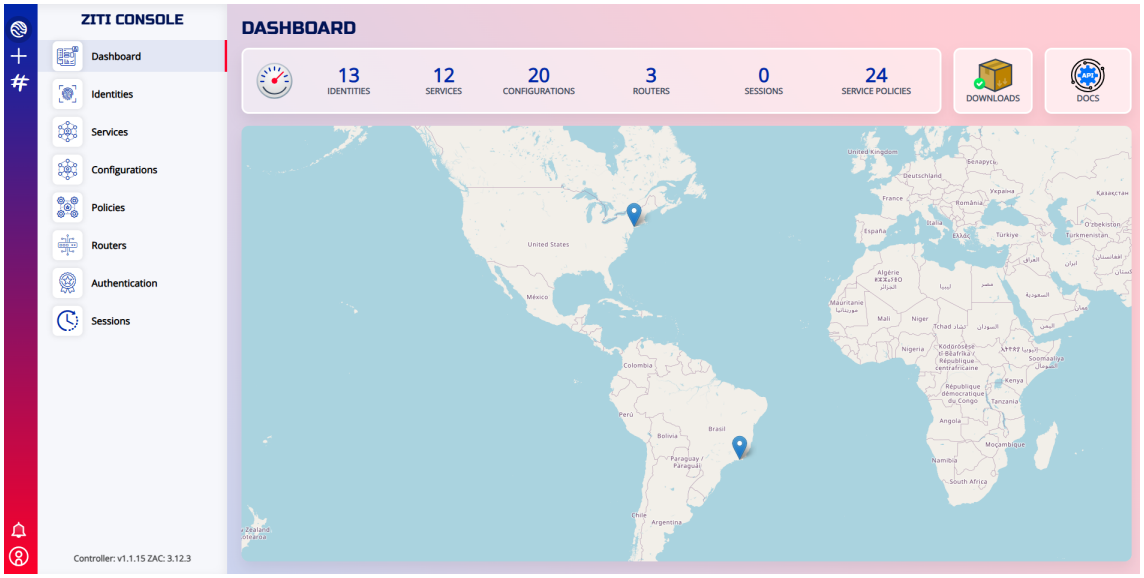


Figura 2.8. Tela inicial de gerenciamento da plataforma OpenZiti.

2.5.2. Configuração do Roteador de Borda

Esta etapa consiste na configuração de, no mínimo, um roteador de borda. Vale ressaltar que, dependendo da estratégia de implementação adotada, pode ser necessário configurar múltiplos roteadores, a fim de atender a requisitos específicos, como localização geográfica ou segmentação por serviços. Neste exemplo, será configurado um único roteador de borda em um servidor Linux Debian 12. Porém, é necessário criar logicamente o roteador para obter seu *token* de autenticação. A Figura 2.9 ilustra o processo para se criar o roteador. Uma vez concluída esta tarefa, é possível obter o *token* no formato jwt conforme Figura 2.10

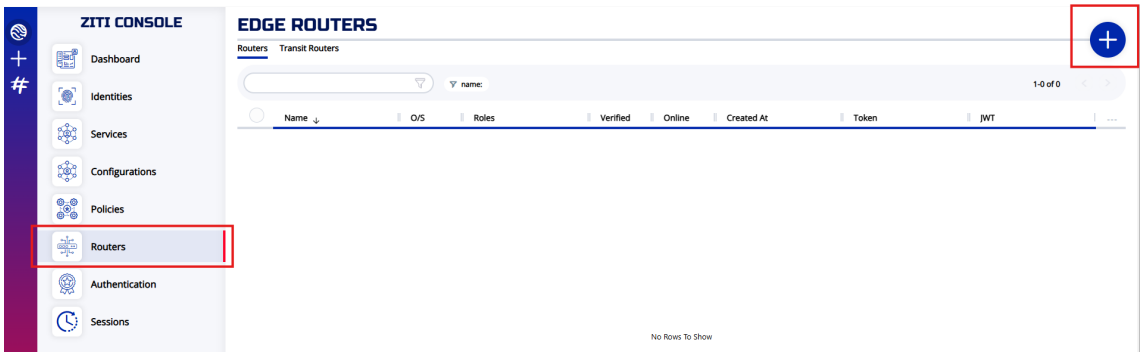


Figura 2.9. Configuração do roteador de borda.

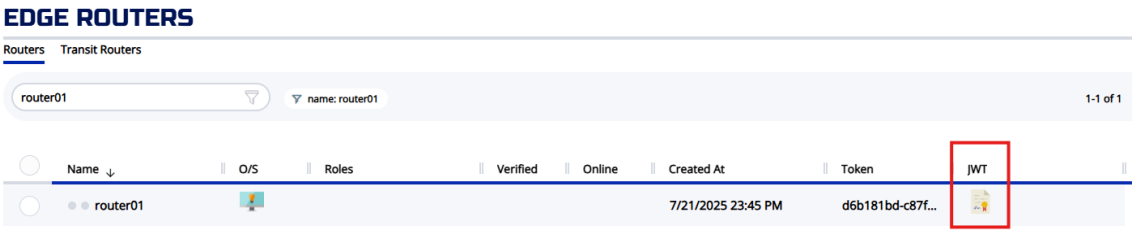


Figura 2.10. Obtenção do token para *enroll*.

Após a configuração lógica, é inicia-se a implementação do roteador de borda. Esse processo é semelhante ao da configuração do controlador, conforme ilustrado no Código Fonte 2.6. Um ponto de atenção fundamental é o uso do *token* gerado durante a etapa de configuração lógica. Esse *token* deve ser atribuído à variável ZITI_ENROLL_TOKEN e só pode ser utilizado uma única vez. Por isso, é essencial garantir que ele seja corretamente aplicado no momento do provisionamento, evitando erros ou a necessidade de gerar um novo *token*. Após configurado, é possível iniciar o serviço do roteador conforme o Código Fonte 2.7.

Código Fonte 2.6. Instalação do Roteador de Borda OpenZiti.

```
# Instalação do Controlador OpenZiti
apt update && apt upgrade -y
curl -sS https://get.openziti.io/install.bash | sudo bash -s
 ↪  openziti-router
vim /opt/openziti/etc/router/bootstrap.env

#Edite o arquivo boostrap.env e insira as seguintes informações

# the controller's DNS name (required)
ZITI_CTRL_ADVERTISED_ADDRESS='FQDN'

# the controller's port (default: 1280)
ZITI_CTRL_ADVERTISED_PORT='1280'

# this router's DNS name or IP address (default: localhost)
ZITI_ROUTER_ADVERTISED_ADDRESS='router01.zetin.uff.br'

# this router's port (default: 3022), if <= 1024,
# then grant the NET_BIND_SERVICE ambient capability in
# /etc/systemd/system/ziti-router.service.d/override.conf
ZITI_ROUTER_PORT='3022'

# token will be scrubbed from this file after enrollment
ZITI_ENROLL_TOKEN='TOKEN'

```

Código Fonte 2.7. Inicialização do roteador de borda OpenZiti.

```
/opt/openziti/etc/router/bootstrap.bash
systemctl enable --now ziti-router.service
```

2.5.3. Configuração das Identidades

A próxima etapa consiste na criação das identidades. A identidade é uma peça fundamental do OpenZiti, uma vez que é utilizada tanto pelos clientes (que realizam o *dial*) quanto pelos servidores (que hospedam as aplicações). Nesse exemplo, serão criadas duas identidades, Servidor1 e Cliente1, conforme Figura 2.11.

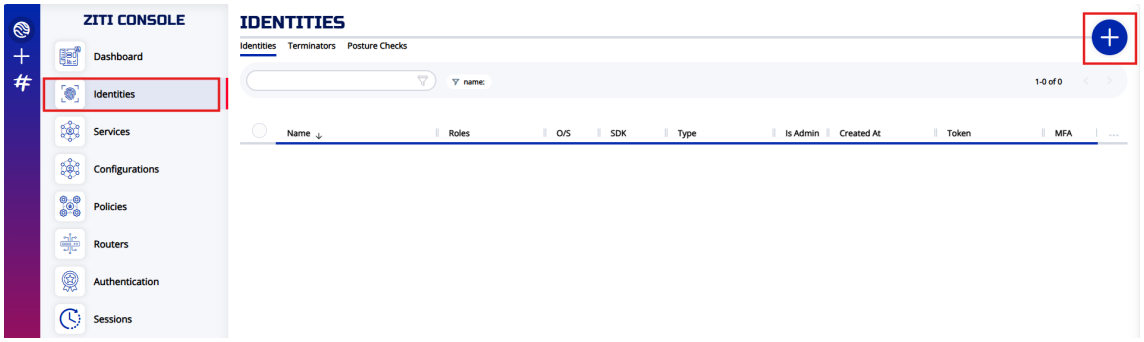


Figura 2.11. Criação de identidades no OpenZiti.

Ambas as identidades devem ser salvas conforme Figura 2.12 para serem configuradas nos respectivos ambientes.

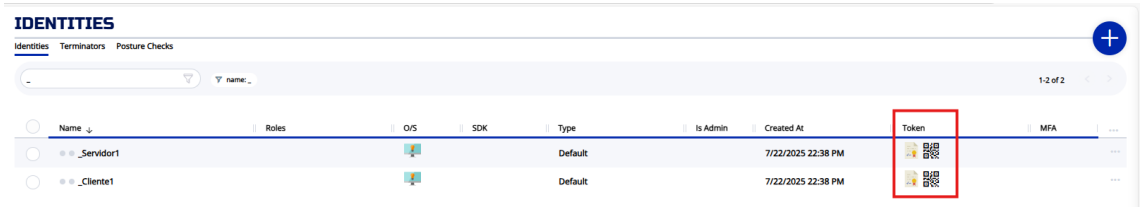
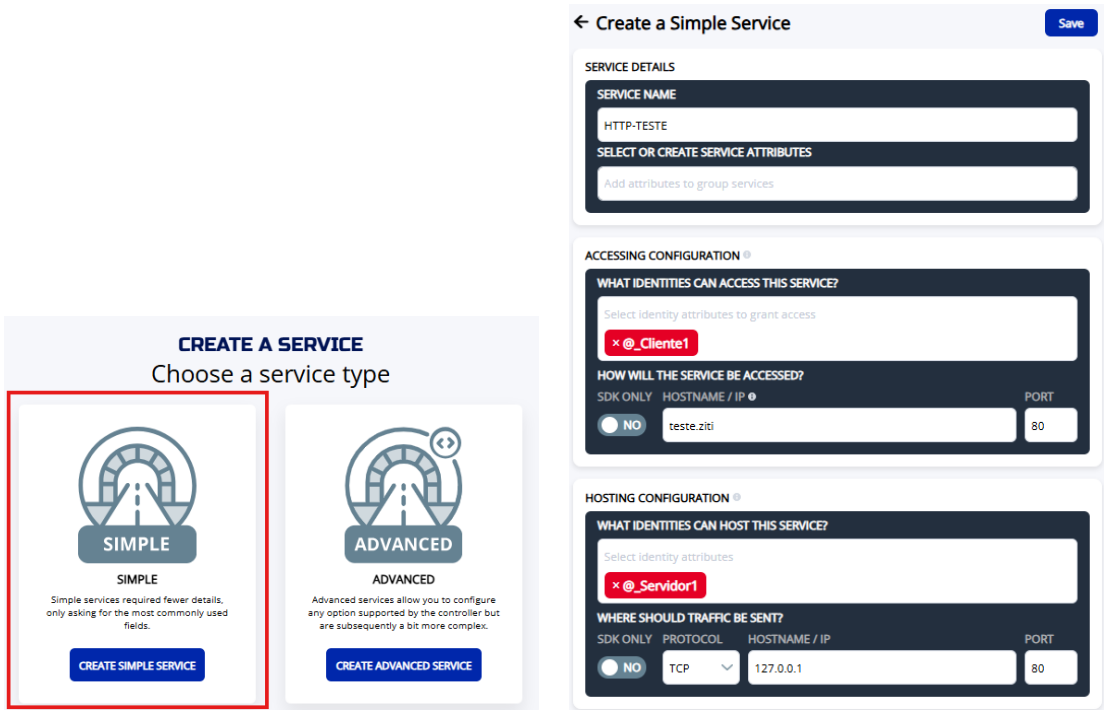


Figura 2.12. Identidades do servidor e cliente.

2.5.4. Configuração dos Serviços

Os serviços podem ser configurados através do CREATE SIMPLE SERVICE ilustrado pela Figura 2.13(a) acessado através do menu **Services**. Com está configuração, as **políticas de serviços** também serão criadas, tornando o processo mais intuitivo. O serviço depende de duas configurações essenciais para operar corretamente. A primeira é o acesso, que define quais identidades tem permissão para usá-lo, conforme Figura 2.13(b).Esta configuração é utilizada pelo cliente, indicando qual será o hostname e porta que serão interceptados para encaminhamento ao host que hospeda o serviço. A outra configuração é relacionado ao host, também ilustrado na Figura 2.13(b). Nessa configuração, o hostname ou IP deverão ser preenchidos com base em como o host consegue

se comunicar com a aplicação. Neste exemplo, como a aplicação é executada localmente, basta inserir o endereço IP da interface do host.

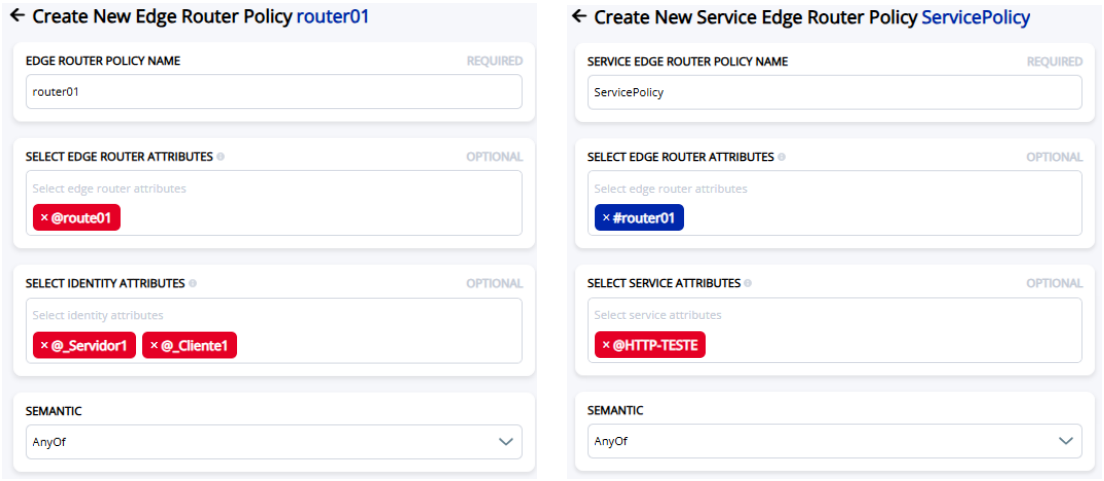


(a) Modos disponíveis para criação de um serviço. (b) Configuração de atributos de um serviço.

Figura 2.13. Criação de um serviço básico dentro do controlador OpenZiti.A Figura 2.13(a) ilustra os modos para criar os serviços, sendo eles o Simples e Avançado. A Figura 2.13(b) apresenta as configurações necessárias para mapear os atributos de acesso e hospedagem do serviços.

2.5.5. Configuração das Políticas

A última etapa da configuração no controlador OpenZiti consiste na criação das políticas do roteador de borda. Elas definem quais identidades (usuários, dispositivos, serviços) estão autorizadas a se conectar a determinados *edge routers* da rede Ziti. Essa definição é essencial, pois, para que uma identidade possa utilizar ou oferecer um serviço, é necessário que se conecte à rede por meio de um roteador autorizado. Para isso, acesse o menu *Policies* e, em seguida, ***Router Policies***. Após clicar no símbolo +, uma janela será exibida para o preenchimento dos dados, conforme a Figura 2.14(a). Neste momento, serão selecionados os atributos que compõem a política do *router01*. Neste exemplo, as identidades *Servidor1* e *Cliente1* terão permissão para acessar o roteador de borda *router01*. Com isso, o tráfego entre cliente e servidor, necessário para acessar ou hospedar um serviço, será encaminhado por esse roteador. A última política a ser configurada é ***Service Router Policies***, que determina quais roteadores podem hospedar e acessar serviços específicos, permitindo controlar e otimizar o caminho do tráfego com base em critérios como proximidade geográfica, desempenho ou disponibilidade.



(a) Política do roteador de borda.

(b) Política de serviço do roteador de borda.

Figura 2.14. Configuração das políticas do OpenZiti. A Figura 2.14(a) ilustra a configuração para criar as políticas do roteador de borda, relacionando quais identidades podem se conectar a um determinado roteador. Enquanto a Figura 2.14(b) relaciona quais serviços podem se associar a um roteador.

2.5.6. Configuração do Cliente

O OpenZiti oferece suporte à configuração de clientes em diversas plataformas. No caso do Windows, é necessário instalar o Ziti Desktop Edge, disponível no site oficial. Após a instalação, a identidade gerada deverá ser importada por meio da interface gráfica do aplicativo. Neste exemplo, foi utilizada uma identidade no formato .jwt, com o nome Cliente1. Uma vez autenticado, o aplicativo passa a interceptar e redirecionar o tráfego de rede para os serviços definidos na configuração, garantindo um acesso seguro e direto, sem a necessidade de VPNs tradicionais. Todo o acesso

2.5.7. Indicadores e Métricas Práticas

A adoção de uma Arquitetura de Confiança Zero (ZTA) impõe desafios operacionais e de desempenho, especialmente em ambientes de produção com restrições de recursos ou requisitos de latência. Para garantir sua viabilidade técnica e justificar sua implantação, é fundamental o uso de métricas objetivas que permitam comparar diferentes implementações, otimizar componentes críticos e avaliar sua resiliência. Indicadores funcionais estão relacionados à eficácia dos mecanismos centrais de controle e verificação da ZTA:

- **Tempo de Autenticação (T_{auth}).** Tempo médio entre a solicitação de acesso e a validação de identidade e dispositivo.
- **Latência de Autorização (T_{policy}).** Tempo de resposta do mecanismo de *enforcement* de políticas (Policy Decision/Enforcement Points).
- **Taxa de Acesso Permitido / Negado.** Razão entre decisões de permissão e bloqueio, útil para calibrar políticas e evitar falsos positivos.

- **Tempo de Detecção de Comportamento Anômalo (T_{anom}).** Tempo entre o início de uma atividade suspeita e sua detecção pelo mecanismo de monitoramento.

A aplicação da ZTA em dispositivos IoT e redes de borda exige atenção especial ao consumo de recursos locais:

- **Uso Médio de CPU (%CPU).** Percentual médio de utilização do processador durante sessões autenticadas.
- **Consumo de Memória (MB).** Memória média ocupada pelos agentes de confiança e mecanismos de autorização locais.
- **Overhead de Tráfego (ΔBW).** Aumento percentual no tráfego de rede causado por trocas de autenticação, verificação de contexto e atualizações de políticas.

A capacidade da arquitetura em manter sua funcionalidade frente a falhas ou ataques é avaliada por métricas de disponibilidade e recuperação:

- **Tempo Médio de Recuperação (MTTR).** Tempo necessário para restaurar o controle de acesso após falha ou comprometimento de um componente.
- **Taxa de Falsos Positivos/Negativos.** Incidência de decisões incorretas de bloqueio ou permissão frente a atividades legítimas ou maliciosas.
- **Disponibilidade dos Serviços de Autorização.** Percentual de tempo em que os mecanismos de autorização estão acessíveis e operacionais.

Essas métricas permitem uma abordagem baseada em evidência para decisões de projeto e ajuste de ZTA, suportando a definição de SLAs (Service Level Agreements), auditoria de conformidade e benchmarking entre soluções concorrentes. A sistematização de tais indicadores deve fazer parte de um processo contínuo de avaliação e melhoria da postura de segurança.

2.6. Estudos de Caso em Ambientes Reais

No setor corporativo, a adoção de ZTA vem sendo impulsionada pela necessidade de proteger acessos remotos, dados sensíveis e aplicações em nuvem contra acesso não autorizado e movimento lateral. Um exemplo recorrente é o uso de autenticação federada integrada a controles baseados em atributos (ABAC), em que políticas de acesso são definidas não apenas pela função do usuário, mas também por seu local de origem, horário, tipo de dispositivo e postura de segurança. Empresas que migraram para ambientes SaaS têm utilizado motores de decisão contextual e segmentação de acesso por carga de trabalho. Mecanismos de telemetria contínua são empregados para ajustar dinamicamente permissões, bloquear sessões anômalas e mitigar ataques de elevação de privilégio [Gupta et al., 2024, He et al., 2022].

Tabela 2.7. Estudos de caso com aplicação de *Zero Trust Architecture*

Domínio	Cenário de Aplicação	Técnicas ZTA Empregadas	Benefícios Observados
Corporativo	Acesso remoto e SaaS com verificação de contexto	ABAC, autenticação federada, avaliação de postura de dispositivo	Redução de superfícies de ataque, controle granular, conformidade regulatória
Rede federada	Compartilhamento de dados entre universidades e centros médicos	Identidade auto-soberana, *blockchain*, contratos inteligentes	Controle fine-grained, rastreabilidade, interoperabilidade entre domínios
Infraestrutura crítica	Segmentação de redes industriais e proteção de UAVs	Segmentação lógica, autenticação contínua, controle baseado em atributos	Resiliência a ataques, disponibilidade operacional, detecção de anomalias

Um segundo estudo relevante refere-se à aplicação de ZTA em redes federadas para colaboração científica. Ambientes como consórcios de pesquisa médica, universidades interligadas e plataformas de compartilhamento de dados sensíveis exigem um modelo de confiança distribuída. Nestes casos, a arquitetura *Zero Trust* é complementada pelo uso de *blockchain* e identidade digital auto-soberana (SSI), permitindo que múltiplas instituições compartilhem dados e recursos sem depender de uma autoridade centralizada [Tuler De Oliveira et al., 2022, de Oliveira et al., 2024, Pooja e Chandrakala, 2024]. Contratos inteligentes são utilizados para registrar e auditar acessos, enquanto decisões baseadas em atributos controlam permissões granulares. A rastreabilidade e a transparência se tornam elementos fundamentais para garantir confiança inter-organizacional, especialmente quando há exigências regulatórias como GDPR ou LGPD.

Em contextos de infraestrutura crítica, como redes de energia, hospitais, sistemas industriais (ICS/SCADA) e operações militares, os desafios de segurança são amplificados por fatores como conectividade intermitente, requisitos de tempo real e criticidade operacional. Nestes domínios, a ZTA é aplicada para segmentar logicamente redes industriais, autenticar continuamente sensores e atuadores, e aplicar políticas condicionais baseadas em localização, tempo e comportamento do dispositivo. Em sistemas de controle de tráfego aéreo e redes de UAVs, por exemplo, a ZTA tem sido usada para isolar domínios operacionais, aplicar autenticação baseada em certificados e monitorar desvios em rotinas de comunicação [Nahar et al., 2024, Dhiman et al., 2024, Gambo e Almulhem, 2025]. A resiliência a ataques persistentes é obtida por meio de microsegmentação combinada com motores de decisão assistidos por IA e detecção de anomalias.

Esses estudos de caso demonstram que a adoção da ZTA é viável e benéfica em diversos cenários, embora apresente desafios específicos conforme o domínio. Em ambientes corporativos, os ganhos em controle de acesso e visibilidade são evidentes, mas exigem integração com ferramentas modernas de identidade e autenticação. Em redes federadas, a descentralização da confiança amplia a escalabilidade da colaboração, mas impõe desafios de padronização e interoperabilidade. Já em infraestruturas críticas, a aplicação de ZTA exige compatibilidade com protocolos legados, operação em tempo real e alta confiabilidade.

A maturidade das soluções ZTA aplicadas a esses cenários ainda está em evolução. A integração com inteligência artificial para decisões de acesso adaptativas, a auditoria

baseada em *blockchain* e a aplicação de modelos de reputação estão entre os próximos passos para fortalecer a resiliência desses ambientes. O desenvolvimento de plataformas que ofereçam ZTA como serviço e ferramentas *open source* como OpenZiti favorecem a democratização dessa arquitetura em organizações de diferentes portes. À medida que surgem novas ameaças e regulamentações, a capacidade de adaptar e observar o comportamento da rede em tempo real será um diferencial estratégico na aplicação prática da *Zero Trust Architecture*.

2.6.1. Zero Trust em Ambientes Críticos: UAVs e IoBT

Ambientes críticos como operações militares, sistemas de defesa cibernética e redes de sensores táticos exigem altos níveis de confiabilidade, disponibilidade e segurança. A *Internet of Battlefield Things* (IoBT) e os veículos aéreos não tripulados (*Unmanned Aerial Vehicles* - UAVs) representam elementos essenciais nesses contextos, sendo amplamente utilizados para reconhecimento, vigilância, entrega de cargas e apoio à decisão. Contudo, essas plataformas estão expostas a ciberataques altamente sofisticados, tornando o modelo tradicional de segurança baseado em perímetro insuficiente. A *Zero Trust Architecture* (ZTA) se destaca como abordagem promissora para garantir segurança adaptativa e contínua nesses cenários, eliminando suposições implícitas de confiança em redes críticas e operando com base em verificação constante, privilégio mínimo e segmentação dinâmica [Alquwayzani e Albuali, 2024, Gupta et al., 2024].

No contexto militar, UAVs são utilizados para missões de *Intelligence, Surveillance and Reconnaissance* (ISR), frequentemente operando em redes sem infraestrutura fixa e com dispositivos móveis, intermitentes e heterogêneos. A ZTA permite isolar logicamente diferentes funções embarcadas, restringindo comunicações apenas ao necessário e aplicando autenticação contínua entre os módulos de controle, navegação, *payload* e telemetria. A segmentação fina possibilita impedir, por exemplo, que falhas em módulos de visão comprometam os canais de comando. Além disso, mecanismos como ABAC (controle baseado em atributos) ou PBAC (controle baseado em políticas) permitem decisões contextuais baseadas na missão, horário, tipo de sensor ou localização geográfica. Assim, apenas entidades que cumpram múltiplos critérios verificados em tempo real podem acessar fluxos críticos de controle, mitigando ataques como *hijacking*, *spoofing* e manipulação de carga útil [Alquwayzani e Albuali, 2024].

A IoBT estende esses desafios à escala de batalhões inteiros. Soldados, veículos, sensores, UAVs e dispositivos de comunicação formam redes móveis e intermitentes que devem ser gerenciadas de forma segura e adaptativa. A ZTA fornece as bases para criar domínios de confiança efêmeros e descentralizados. Microsegmentação dinâmica baseada em função e localização permite isolar fluxos de comando e controle dos fluxos de monitoramento ambiental. A aplicação de *machine learning* embarcado, combinada à telemetria contínua, permite detectar comportamentos anômalos e adaptar permissões automaticamente. Isso é viável mesmo em redes degradadas, por meio de modelos leves de aprendizado e uso de evidências parciais para classificação de risco [He et al., 2022].

Tecnologias de controle como SDN (*Software Defined Networking*) e a integração com redes 5G/6G aumentam a flexibilidade da orquestração de políticas em tempo real. Exemplos de frameworks operacionais como o TENA (*Test and Training Enabling Ar-*

Tabela 2.8. Comparação entre o uso da ZTA em domínios críticos e em domínios empresariais.

Critério	Domínios Críticos (UAVs, IoBT, Defesa)	Domínios Empresariais
Ambiente Operacional	Hostil, degradado, com conectividade intermitente e requisitos de missão	Controlado, com infraestrutura confiável e conectividade estável
Tipo de Dispositivo	Sensores embarcados, UAVs, dispositivos móveis militares	Estações de trabalho, servidores, laptops e dispositivos móveis corporativos
Requisitos de Segurança	Autenticação contínua, integridade de missão, controle sob ataque, operação sob falha	Conformidade regulatória, proteção de dados, disponibilidade de serviços
Topologia de Rede	Redes mesh dinâmicas, MANETs, infraestrutura distribuída e efêmera	Redes estáticas, híbridas ou em nuvem com perímetros conhecidos
Modelos de Controle de Acesso	ABAC/PBAC com verificação contextual e restrições baseadas em missão e localização	RBAC ou ABAC baseado em políticas organizacionais e funções fixas
Desafios Técnicos	Restrições energéticas, latência, confiabilidade da telemetria, interoperabilidade militar	Legado, resistência à mudança, integração com sistemas existentes
Automação e IA	Detecção autônoma de ameaças, análise embarcada e resposta local adaptativa	SIEM, automação de políticas, análise de logs e resposta centralizada
Padrões e Referências	DoD ZTA Reference Architecture, TENA, STIGs	NIST SP 800-207, ISO/IEC 27001, CIS Controls

chitecture) e a *DoD Zero Trust Reference Architecture* reforçam a viabilidade de soluções ZTA modulares e interoperáveis em contextos de defesa [Gupta et al., 2024].

Além dos ganhos operacionais, a ZTA em ambientes críticos enfrenta desafios técnicos significativos. A limitação de energia e processamento embarcado impõe restrições à aplicação de criptografia forte e algoritmos de aprendizado. A confiabilidade da coleta de contexto (*telemetria*) em ambientes hostis também é um fator limitante, já que sensores podem ser corrompidos ou operarem com latência. Em operações multinacionais, a interoperabilidade de políticas de acesso *Zero Trust* ainda requer padronizações robustas. Como caminhos futuros, propõe-se a adoção de inteligência federada para decisões de acesso distribuídas, bem como a integração de criptografia pós-quântica com autenticação contínua e verificação de integridade resiliente.

A aplicação de *Zero Trust* em UAVs e redes IoBT representa um avanço fundamental para a segurança de infraestruturas críticas e ambientes militares. A segmentação lógica, a verificação contextual e a capacidade de resposta adaptativa posicionam a ZTA como o novo paradigma para redes táticas confiáveis em tempos de guerra cibernética. A Tabela 2.8 apresenta uma comparação entre a aplicação da arquitetura *Zero Trust* em domínios críticos, como defesa e IoBT, e em contextos empresariais convencionais, destacando as diferenças em requisitos operacionais, tecnologias utilizadas, desafios e padrões de referência.

2.6.2. Colaboração Científica Descentralizada

A adoção de práticas colaborativas em ciência tem ampliado a complexidade dos ecossistemas de pesquisa, impondo novos requisitos à proteção de dados sensíveis. Em

ambientes compostos por múltiplas instituições e diferentes domínios de confiança, a *Zero Trust Architecture* (ZTA) emerge como alternativa robusta para assegurar confidencialidade, rastreabilidade e controle granular de acesso em atividades como revisão por pares e intercâmbio de dados experimentais. Soluções convencionais de revisão e compartilhamento operam com suposições de confiança implícita e autenticação insuficiente, o que expõe manuscritos e dados inéditos a riscos de vazamento, uso indevido e conflitos de interesse. A ausência de padronização entre instituições e jurisdições distintas agrava esse cenário, dificultando a harmonização das políticas de acesso [Pooja e Chandrakala, 2024].

Pooja e Chandrakala propõem um modelo baseado na combinação de ZTA, *blockchain* e contratos inteligentes para mitigar essas fragilidades [Pooja e Chandrakala, 2024]. As políticas de acesso consideram atributos contextuais, como afiliação, histórico de atuação e janela temporal de acesso, sendo avaliadas automaticamente por contratos inteligentes. O acesso a documentos é concedido apenas se a pontuação de confiança do revisor atingir o limiar mínimo, e os dados são protegidos com chaves derivadas temporalmente (HKDF), limitando sua exposição.

No compartilhamento de dados sob múltiplas restrições, como os clínicos ou envolvendo seres humanos, o sistema verifica se o requisitante atende aos critérios definidos por todos os controladores envolvidos. A decisão é tomada com base em interpolação de Lagrange entre políticas divergentes, permitindo decisões consensuais, auditáveis e flexíveis. Essa abordagem reduz o fenômeno de *data lock-in* e incentiva a reutilização responsável [Pooja e Chandrakala, 2024]. A arquitetura proposta é compatível com sistemas federados, como o *Federated Research Data Infrastructure* e com tecnologias como o *Hyperledger Fabric*. Sua integração reforça princípios como auditabilidade, exposição mínima de dados e responsabilização. O modelo permite ainda o uso de avaliações dos autores sobre os revisores, incorporando critérios reputacionais aos ciclos de seleção.

2.7. Projetos de Pesquisa e Desenvolvimento

Em um cenário cada vez mais dinâmico e conectado, o modelo de confiança zero está redefinindo a segurança cibernética, especialmente no contexto da Internet das Coisas (IoT) e das redes de próxima geração. Nesse sentido, abordagens de segurança proativa e adaptável tornam-se imperativas. Diversos projetos e iniciativas têm como objetivo expandir os horizontes desse modelo, explorando sua aplicação em áreas como saúde, redes 5G, privacidade e outros campos emergentes. O projeto NextSec propõe uma arquitetura para segurança em sistemas de próxima geração, para conectividade dispositivos IoT e pessoas, além de possibilitar comunicações máquina-a-máquina e fornecer recursos computacionais e de armazenamento com baixa latência. Os sistemas são essenciais para aplicações críticas como veículos autônomos e telecirurgia[10]. O projeto *Proactive End-to-End Zero Trust-Based Security Intelligence for Resilient Non-cooperative 5G Networks* visa desenvolver uma solução de segurança de ponta a ponta para aprimorar comunicações militares e de missão crítica através de redes 5G não confiáveis utilizando o princípio *Zero Trust*. O projeto também inclui o desenvolvimento de um currículo para treinamento em segurança 5G e a promoção da diversidade na pesquisa e desenvolvimento nesta área [11].

[10]Disponível em `https://www.nsf.gov/awardsearch/showAward?AWD_ID=2148374`.
[11]Disponível em `https://www.nsf.gov/awardsearch/showAward?AWD_ID=2226232`.

Por sua vez, o projeto *Zero-trust and Traceable Data Infrastructure for Health IoT Data Storage and Sharing* tem por objetivo desenvolver um ecossistema de compartilhamento de dados rastreável e de confiança zero onde a propriedade e o controle dos dados são devolvidos aos indivíduos que geram os dados. Nesse sentido, os proprietários dos dados poderão definir a política de acesso e compartilhamento que serão aplicados no modelo de confiança zero, ou seja, sem depender de coordenadores ou infraestrutura centrais[12].

O projeto Zero Trust X (ZTX) é uma iniciativa do Departamento de Defesa dos EUA para desenvolver uma solução de segurança de ponta a ponta, chamada *Zero Trust Chain* (ZTC), para o uso seguro e confiável de redes 5G em operações militares. Dada a defasagem dos EUA na fabricação de equipamentos de rede 5G e as vulnerabilidades de segurança das redes comerciais, o ZTX propõe um *software* que integra monitoramento de ameaças via Open-RAN (O-RAN) e 5G core, complementado por aprimoramentos de segurança nos dispositivos. Essa abordagem permite que as equipes militares compartilhem informações de forma segura em redes 5G de alto desempenho, mas potencialmente não confiáveis, detectando entidades maliciosas em tempo real e protegendo o tráfego do DoD, sem a necessidade de grandes modificações nas redes 5G existentes[13]

2.8. Desafios e Tendências Futuras

A *Zero Trust Architecture* (ZTA) consolida-se como um novo paradigma de segurança cibernética, desafiando o modelo perimetral tradicional ao adotar o princípio de "nunca confiar, sempre verificar". Embora sua adoção venha crescendo em diversos setores, desde redes corporativas até ambientes militares e infraestruturas críticas, a implementação prática da ZTA ainda encontra uma série de barreiras técnicas, operacionais e organizacionais. Esta seção discute os principais desafios que limitam a difusão ampla e eficaz da ZTA e aponta direções futuras promissoras para sua evolução.

2.8.1. Desafios de Implementação

A implementação eficaz da arquitetura de confiança zero enfrenta diversos desafios. A falta de padrões de comunicação entre dispositivos, arcabouços e plataformas utilizados podem são significativos [Bertino, 2021]. A necessidade de combinar grandes volumes de dados heterogêneos de várias fontes pode ser complexa e demorada. A micro-segmentação é crucial, porém enfrenta dificuldades em redes extensas devido à complexidade dos fluxos de trabalho e à tradução desafiadora de requisitos de acesso em políticas de controle de acesso aplicáveis tanto em nível de rede quanto de aplicativo. Os principais desafios na implementação da arquitetura de confiança zero incluem [Syed et al., 2022, Stafford, 2020]: (i) a integração com infraestruturas existentes, que demanda adaptação e migração complexas de sistemas legados; (ii) o gerenciamento robusto de identidades para autenticação e autorização contínuas também é essencial; (iii) a necessidade de monitoramento contínuo, requerendo tecnologias avançadas para detecção de ameaças em tempo real; e (iv) a adaptação é outra necessidade, permitindo que a arquitetura de confiança zero atenda às especificidades variadas de diferentes ambientes e casos de uso. A dificuldade de padronização e interoperabilidade entre soluções de diferentes

[12]Disponível em `https://www.nsf.gov/awardsearch/showAward?AWD_ID=2312973`.
[13]Disponível em `https://www.nsf.gov/awardsearch/showAward?AWD_ID=2515378`.

fornecedores, o que pode levar ao aprisionamento tecnológico, a ausência de formatos de dados comuns para troca de informações entre componentes de segurança; a necessidade de evitar interrupções na experiência dos usuários durante o processo de migração e a complexidade de definir níveis apropriados de confiança e classificar os recursos adequadamente podem ser obstáculos significativos nesta jornada. Além disso, lidar com dispositivos não gerenciados, como os oriundos de políticas BYOD, implica riscos adicionais devido à ausência de controle direto da organização. Outro ponto crítico é o aprimoramento contínuo do algoritmo de confiança, que deve incorporar informações de múltiplas fontes, como credenciais, localização, comportamento de rede e inteligência de ameaças, exigindo constante calibragem para evitar falsos positivos e negativos [Teerakanok et al., 2021].

A implementação da ZTA representa um paradigma de transformação profunda na forma como as organizações protegem seus ativos, usuários e dados. Um dos principais obstáculos é o **custo**, já que os investimentos necessários para modernizar infraestruturas legadas, adquirir novas ferramentas e treinar equipes podem ser substanciais. Embora os orçamentos destinados para aprimoramento da cibersegurança estejam crescendo, as restrições orçamentárias continuam sendo uma preocupação crítica. Um desafio recorrente são as **lacunas tecnológica**s. Muitas organizações ainda não dispõem de soluções adequadas para implementar políticas de acesso dinâmico, autenticação ou segmentação de rede baseada em identidade. A complexidade para integração entre sistemas legados e novas tecnologias também contribui para esse cenário, tornando difícil a orquestração de uma arquitetura coerente e segura. Outro aspecto importante são os **obstáculos regulatórios e legais**, como a necessidade de atender a exigências de privacidade e proteção de dados impostas por diferentes jurisdições. Isso exige atenção cuidadosa para garantir que as práticas de segurança estejam em conformidade com regulamentações como GDPR ou LGPD, dependendo do setor e da localização da empresa. Esse contexto regulatório pode tornar a implementação mais lenta e burocrática. A **escassez de profissionais qualificados** é outro fator que compromete o avanço dessas iniciativas. A demanda por especialistas em identidade, governança de acesso e segurança baseada em risco supera a oferta disponível no mercado, o que dificulta a formação de equipes capacitadas para conduzir o processo de adoção desse modelo. Esse déficit é especialmente crítico em organizações menores ou em setores que estejam iniciando sua jornada em infraestrutura digital.

Há ainda desafios de conscientização e adesão organizacional. Muitas vezes, os líderes ou equipes não compreendem completamente os benefícios desse modelo ou resistem à sua adoção por receio de impactos negativos. Diante desses desafios, é fundamental que as organizações adotem uma abordagem estratégica, baseada nas necessidades e com planejamento estruturado.

2.8.2. Desafios de Evolução da Arquitetura e Tendências Futuras

Um dos maiores obstáculos enfrentados na evolução e aplicação de ZTA em larga escala está relacionado à escalabilidade. Em arquiteturas que envolvem milhares ou milhões de dispositivos, como em ambientes IoT, redes 6G ou sistemas federados, o gerenciamento eficiente de identidades, políticas de acesso e contexto dinâmico torna-se uma tarefa complexa [Gambo e Almulhem, 2025]. A manutenção de políticas de acesso

granulares, associadas a múltiplos atributos, exige estruturas de governança robustas, sincronização entre domínios e mecanismos de distribuição de contexto em tempo real.

O desempenho também constitui um gargalo relevante. O modelo ZTA impõe verificações contínuas de identidade, postura do dispositivo, localização e tempo de acesso a cada solicitação. Esse processo, se não otimizado, pode introduzir latência excessiva e sobrecarga em sistemas sensíveis ao tempo, como controle industrial, comunicações táticas e serviços médicos [He et al., 2022, Barbosa et al., 2025]. Estratégias como cache de decisões de acesso, balanceamento entre política e contexto e uso de mecanismos locais de decisão vêm sendo exploradas para mitigar esse impacto.

A interoperabilidade entre domínios administrativos distintos representa outro desafio. A ausência de padronizações amplamente adotadas para troca de contexto, formatos de políticas e autenticação federada dificulta a integração de ZTA em ambientes colaborativos, como redes interinstitucionais, cadeias logísticas globais ou consórcios científicos [de Oliveira et al., 2024]. Embora existam iniciativas como o NIST SP 800-207 e modelos baseados em ABAC e PBAC, a heterogeneidade dos ecossistemas e a resistência a alterações estruturais dificultam a adesão ampla. Soluções emergentes têm buscado mitigar essas barreiras por meio da padronização de políticas e identidades. Um exemplo é a abordagem *Policy as Code*, cuja implementação mais conhecida é o *Open Policy Agent* (OPA)[14] em conjunto com a linguagem Rego, que permite expressar políticas de acesso de forma auditável, versionável e portável entre sistemas heterogêneos. No âmbito da padronização internacional, o grupo IETF propôs o *Security Automation and Continuous Monitoring* (SACM), voltado à automação do monitoramento e da verificação contínua de postura de segurança em redes federadas [Cam-Winget e Lorenzin, 2017]. Adicionalmente, frameworks de identidade federada baseados em *Identificadores Descentralizados* (DIDs) e *Credenciais Verificáveis* (VCs), como os definidos pelo W3C[15] e aplicados em sistemas como Trustbloc[16] ou Trust Over IP[17], oferecem mecanismos compatíveis com os princípios da ZTA ao eliminar dependência de autoridades centrais e permitir verificações criptográficas baseadas em contexto [de Oliveira et al., 2024]. A combinação dessas tecnologias possibilita a criação de ambientes ZTA interoperáveis, auditáveis e alinhados com requisitos de conformidade regulatória em múltiplos domínios.

A coleta, verificação e sincronização de informações contextuais, como comportamento de segurança do *endpoint*, geolocalização, horário e análise comportamental, é uma tarefa crítica e propensa a erros. A confiabilidade das fontes de contexto e a integridade dos dados coletados são determinantes para a eficácia das decisões de acesso [Gupta et al., 2024]. A presença de falsos positivos ou contextos desatualizados pode levar a decisões de bloqueio indevido, impactando a disponibilidade do sistema e a experiência do usuário.

Paralelamente, novos vetores de ataque como manipulação de IA, ataques à cadeia de suprimentos de *software*, exfiltração lateral em ambientes segmentados e exploração de vulnerabilidades em ferramentas de autenticação desafiam a resiliência das soluções

[14]Disponível em `https://www.openpolicyagent.org/`.
[15]Disponível em `https://www.w3.org/TR/did-core/`.
[16]Disponível em `https://trustbloc.readthedocs.io/`.
[17]Disponível em `https://trustoverip.org/`.

ZTA [Nahar et al., 2024]. O modelo *Zero Trust*, ao depender fortemente da identidade e da verificação contextual, torna-se sensível à qualidade e integridade dos sinais de confiança.

Diversas tendências futuras vêm sendo apontadas como complementares à consolidação da ZTA. Uma delas é a construção de arquiteturas *Zero Trust* autoadaptativas, baseadas em inteligência artificial embarcada, modelos de aprendizado contínuo e análise comportamental em tempo real. Tais sistemas serão capazes de ajustar políticas dinamicamente de acordo com padrões emergentes e variações do ambiente [He et al., 2022, Dhiman et al., 2024]. Modelos híbridos que combinam aprendizado supervisionado, não supervisionado e por reforço, aliados a computação de borda (*edge computing*), são promissores nesse cenário.

A integração de criptografia pós-quântica (*Post-Quantum Cryptography* - PQC) com identidade auto-soberana (*Self-Sovereign Identity* - SSI) representa outra fronteira. Sistemas baseados em *blockchain*, DID e contratos inteligentes permitirão registros auditáveis de decisões de acesso, confiança descentralizada e independência de autoridades centralizadas [Tuler De Oliveira et al., 2022, de Oliveira et al., 2024]. Essa abordagem oferece proteção de longo prazo e resistência a futuras quebras criptográficas, sendo particularmente relevante em ambientes regulados e de missão crítica.

A orquestração multidomínio e a computação federada também são tendências promissoras. Com a proliferação de computação em borda (*edge computing*) e redes 6G, a ZTA será exigida a operar em ambientes de baixa latência, alta mobilidade e autonomia local [Nahar et al., 2024]. Modelos federados de decisão, descentralização da verificação de identidade e sincronização segura de contexto são requisitos-chave para a efetividade da segurança nesse cenário.

No campo da observabilidade, destaca-se a segurança auditável e verificável. A instrumentação de sistemas ZTA com registros de atividades (*logs*) imutáveis, trilhas criptográficas e telemetria contínua permitirá não apenas detectar violações, mas também fornecer garantias formais de conformidade e accountability [Gupta et al., 2024, de Oliveira et al., 2024]. Isso reforça a transparência e viabiliza auditorias automatizadas, fundamentais para ambientes regulados.

O avanço de modelos como ZTA-as-a-Service também expressivo. Nesse novo modelo de serviço, componentes como motor de decisão, controle de acesso e contexto são oferecidos via APIs sob demanda. Essa tendência acelera a adoção de ZTA por organizações menores e possibilita a customização de políticas por perfil de risco e vertical de aplicação. Em paralelo, padrões técnicos estão em constante evolução, com contribuições do NIST, IETF, ETSI e IEEE propondo modelos referenciais, protocolos interoperáveis e linguagens declarativas de política.

2.9. Considerações Finais

O aumento da conectividade digital, impulsionado pela expansão de ambientes distribuídos e pela adoção de dispositivos heterogêneos, tem acentuado a fragilidade dos modelos tradicionais de segurança. Essa realidade, fundamenta a necessidade de repensar os mecanismos de proteção baseados em perímetro, cuja eficácia se torna limitada diante da fluidez das fronteiras organizacionais. A Arquitetura de Confiança Zero (*Zero Trust*

Tabela 2.9. Desafios e tendências emergentes na implementação de ZTA.

Desafio	Descrição	Tendência/Abordagem Promissora
Escalabilidade	Gestão de políticas e contexto em larga escala	Modelos federados e aprendizado distribuído
Performance	Verificação contínua introduz latência	Decisão local assistida por IA, caching adaptativo
Interoperabilidade	Incompatibilidade entre domínios e legados	Padrões abertos, autenticação federada, ABAC
Contexto	Confiabilidade e sincronização do ambiente	Telemetria contínua, validação cruzada, edge intelligence
Ameaças emergentes	Ataques à cadeia de suprimentos e exploração de IA	Certificação formal de componentes, observabilidade criptográfica

Architecture - ZTA) surge, nesse cenário, como uma abordagem orientada a minimizar a superfície de ataque por meio de validação contínua, microsegmentação e controle contextualizado de acesso. A proposta central do capítulo consistiu em apresentar os fundamentos teóricos, os componentes estruturais e as implicações práticas da ZTA, com foco especial na aplicação em redes de próxima geração, sistemas embarcados e ambientes corporativos descentralizados.

O capítulo percorreu os principais pilares da arquitetura, com ênfase nos modelos de autenticação adaptativa, nos mecanismos de autorização baseados em atributos e na importância da visibilidade contínua sobre o comportamento das entidades. A plataforma OpenZiti foi utilizada como referência de código aberto para demonstrar a viabilidade técnica da implementação de redes seguras baseadas em ZTA, evidenciando os ganhos em termos de resiliência, auditabilidade e governança distribuída. A análise foi complementada por uma taxonomia detalhada de aplicações, tecnologias e requisitos funcionais, permitindo sistematizar as diversas camadas envolvidas na adoção do modelo. A comparação entre a abordagem perimetral e a ZTA evidenciou a disrupção conceitual entre confiar implicitamente na infraestrutura e condicionar o acesso a fatores dinâmicos e verificáveis. Esses elementos sustentam a adoção crescente da ZTA como fundamento para arquiteturas resilientes em cenários como IoT, redes 5G/6G e ambientes BYOD.

Além dos benefícios imediatos, a ZTA também se projeta para cenários futuros marcados pela ascensão da computação quântica e pela obsolescência de algoritmos criptográficos tradicionais. A recente seleção de algoritmos resistentes à computação quântica pelo NIST reforça a necessidade de adaptação proativa das estratégias de segurança. A integração da ZTA com tecnologias emergentes, como criptografia pós-quântica e inteligência artificial aplicada à segurança adaptativa, aponta para um horizonte em que as organizações poderão responder automaticamente a ameaças complexas com alta precisão. Assim, perspectivas futuras para a evolução da ZTA envolvem desafios importantes, tais como escalabilidade técnica, maturidade organizacional e interoperabilidade interorganizacional. Para assegurar a viabilidade operacional desse modelo em larga escala, é necessária a adoção coordenada de ferramentas avançadas, como aprendizado federado, auditoria imutável em plataformas *blockchain* permissionadas e modelos descentralizados

de gestão de identidades auto-soberanas (SSI). O fortalecimento dessas práticas permitirá que a ZTA avance para além de um modelo conceitual, tornando-se um padrão normativo de segurança aplicável a ecossistemas digitais complexos.

A consolidação efetiva da ZTA depende de esforços colaborativos que envolvam diferentes setores da indústria, academia e governo. É mandatório o desenvolvimento de padrões técnicos unificados, políticas consistentes de segurança digital e novas soluções que promovam a delegação segura de confiança entre múltiplas partes. Dessa forma, a ZTA poderá estabelecer-se como um pilar fundamental na construção de infraestruturas digitais robustas, seguras e preparadas para enfrentar os desafios tecnológicos emergentes das próximas décadas.

Referências

[Abdalla et al., 2024] Abdalla, A. S., Moore, J., Adhikari, N. e Marojevic, V. (2024). Ztran: Prototyping zero trust security xapps for open radio access network deployments. *IEEE Wireless Communications*, 31(2):66–73.

[Aboukadri et al., 2024] Aboukadri, S., Ouaddah, A. e Mezrioui, A. (2024). Machine learning in identity and access management systems: Survey and deep dive. *Computers & Security*, p. 103729.

[AlDaajeh e Alrabaee, 2024] AlDaajeh, S. e Alrabaee, S. (2024). Strategic cybersecurity. *Computers & Security*, 141:103845.

[Alevizos et al., 2022] Alevizos, L., Ta, V. T. e Hashem Eiza, M. (2022). Augmenting zero trust architecture to endpoints using blockchain: A state-of-the-art review. *Security and privacy*, 5(1):e191.

[Alnoaimi e Alomary, 2025] Alnoaimi, S. e Alomary, A. (2025). Zero trust security: A comprehensive comparative analysis of zero trust maturity models. Em *2024 International Conference on IT Innovation and Knowledge Discovery (ITIKD)*, p. 1–8.

[Alquwayzani e Albuali, 2024] Alquwayzani, A. A. e Albuali, A. A. (2024). A systematic literature review of zero trust architecture for military uav security systems. *IEEE Access*, 12:176033–176056.

[Anderson et al., 2022] Anderson, J., Huang, Q., Cheng, L. e Hu, H. (2022). Byoz: Protecting byod through zero trust network security. Em *2022 IEEE International Conference on Networking, Architecture and Storage (NAS)*, p. 1–8. IEEE.

[Andreoni et al., 2022] Andreoni, M., Barbosa, G. N. N. e Mattos, D. M. F. (2022). New barriers on 6G networking: An exploratory study on the security, privacy and opportunities for aerial networks. Em *2022 1st International Conference on 6G Networking (6GNet)*, p. 1–6.

[Barbosa et al., 2025] Barbosa, G. N. N., Andreoni, M. e Mattos, D. M. F. (2025). Leveraging zero trust for enhanced security and connectivity in ad-hoc mesh networks. Em *2025 12th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, p. 229–237.

[Bernabé Murcia et al., 2025] Bernabé Murcia, J. M., Cánovas, E., García-Rodríguez, J., M. Zarca, A. e Skarmeta, A. (2025). Decentralised identity management solution for zero-trust multi-domain computing continuum frameworks. *Future Generation Computer Systems*, 162:107479.

[Bertino, 2021] Bertino, E. (2021). Zero trust architecture: Does it help? *IEEE Security & Privacy*, 19(05):95–96.

[Cam-Winget e Lorenzin, 2017] Cam-Winget, N. e Lorenzin, L. (2017). Security Automation and Continuous Monitoring (SACM) Requirements. RFC 1654, RFC Editor.

[Chang e Mukherjee, 2024] Chang, H. e Mukherjee, S. (2024). Zeta: Transparent zero-trust security add-on for rdma. Em *IEEE INFOCOM 2024 - IEEE Conference on Computer Communications*, p. 1041–1050.

[Chen et al., 2023] Chen, X., Feng, W., Ge, N. e Zhang, Y. (2023). Zero trust architecture for 6G security. *IEEE Network*, p. 1–1.

[Cunha Neto et al., 2024] Cunha Neto, H. N., Hribar, J., Dusparic, I., Fernandes, N. C. e Mattos, D. M. (2024). FedSBS: Federated-learning participant-selection method for intrusion detection systems. *Computer Networks*, 244:110351.

[de Oliveira et al., 2024] de Oliveira, N. R., dos Santos, Y. d. R., Barbosa, G. N. N., Reis, L. H. A., Mendes, A. C. R., de Oliveira, M. T., de Medeiros, D. S. V. e Mattos, D. M. F. (2024). Distributed data security in digital health: Self-sovereign identity, access control, and blockchain-based log records. Em *2024 6th International Conference on Blockchain Computing and Applications (BCCA)*, p. 558–565.

[de Oliveira et al., 2025] de Oliveira, N. R., Silva, J. V. V., de Medeiros, G. N. N. B. D. S. V. e Mattos, D. M. F. (2025). Incorporação de modelos de linguagem em larga escala em dispositivos móveis: Otimização, personalização e desafios. *Jornada de Atualização em Informática 2025; SBC; Maceio*, p. 147–196.

[Dhiman et al., 2024] Dhiman, P., Saini, N., Gulzar, Y., Turaev, S., Kaur, A., Nisa, K. U. e Hamid, Y. (2024). A review and comparative analysis of relevant approaches of zero trust network model. *Sensors*, 24(4).

[Gambo e Almulhem, 2025] Gambo, M. L. e Almulhem, A. (2025). Zero trust architecture: A systematic literature review.

[Gebresilassie et al., 2025] Gebresilassie, S. K., Rafferty, J., Abu-Tair, M., Ali, A., Chen, L. e Cui, Z. (2025). Shield: Secure holistic iot environment with ledger-based defense. *Internet of Things*, 30:101473.

[Gupta et al., 2024] Gupta, A., Gupta, P., Pandey, U. P., Kushwaha, P., Lohani, B. P. e Bhati, K. (2024). ZTSA: Zero trust security architecture a comprehensive survey. Em *2024 International Conference on Communication, Computer Sciences and Engineering (IC3SE)*, p. 378–383.

[He et al., 2022] He, Y., Huang, D., Chen, L., Ni, Y. e Ma, X. (2022). A survey on zero trust architecture: Challenges and future trends. *Wireless Communications and Mobile Computing*, 2022(1):6476274.

[Hosney et al., 2022] Hosney, E. S., Halim, I. T. A. e Yousef, A. H. (2022). An artificial intelligence approach for deploying zero trust architecture (ZTA). Em *2022 5th International Conference on Computing and Informatics (ICCI)*, p. 343–350.

[Hussain et al., 2024] Hussain, M., Pal, S., Jadidi, Z., Foo, E. e Kanhere, S. (2024). Federated zero trust architecture using artificial intelligence. *IEEE Wireless Communications*, 31(2):30–35.

[Jose Diaz Rivera et al., 2024] Jose Diaz Rivera, J., Muhammad, A. e Song, W.-C. (2024). Securing digital identity in the zero trust architecture: A blockchain approach to privacy-focused multi-factor authentication. *IEEE Open Journal of the Communications Society*, 5:2792–2814.

[Kang et al., 2023] Kang, H., Liu, G., Wang, Q., Meng, L. e Liu, J. (2023). Theory and application of zero trust security: A brief survey. *Entropy*, 25(12).

[Kindervag et al., 2010] Kindervag, J., Balaouras, S. et al. (2010). No more chewy centers: Introducing the zero trust model of information security. *Forrester Research*, 3(1):1–16.

[Koukis et al., 2024] Koukis, G., Skaperas, S., Kapetanidou, I. A., Mamatas, L. e Tsaoussidis, V. (2024). Evaluating cni plugins features and tradeoffs for edge cloud applications. Em *2024 IEEE Symposium on Computers and Communications (ISCC)*, p. 1–6.

[Kumar et al., 2024] Kumar, S. S., Cummings, M. e Stimpson, A. (2024). Strengthening llm trust boundaries: a survey of prompt injection attacks. Em *2024 IEEE 4th International Conference on Human-Machine Systems (ICHMS)*, p. 1–6.

[Mämmelä et al., 2016] Mämmelä, O., Hiltunen, J., Suomalainen, J., Ahola, K., Mannersalo, P. e Vehkaperä, J. (2016). Towards micro-segmentation in 5G network security. Em *European Conference on Networks and Communications (EuCNC 2016) Workshop on Network Management, Quality of Service and Security for 5G Networks*.

[Mujib e Sari, 2020] Mujib, M. e Sari, R. F. (2020). Performance evaluation of data center network with network micro-segmentation. Em *2020 12th International Conference on Information Technology and Electrical Engineering (ICITEE)*, p. 27–32. IEEE.

[Mukta et al., 2025] Mukta, R., Pal, S., Chowdhury, K., Hitchens, M., young Paik, H. e Kanhere, S. S. (2025). Zero trust driven access control delegation using blockchain. *Blockchain: Research and Applications*, p. 100319.

[Nace, 2020] Nace, L. (2020). Securing trajectory based operations through a zero trust framework in the nas. Em *2020 Integrated Communications Navigation and Surveillance Conference (ICNS)*, p. 1B1–1–1B1–8.

[Nagendra e Hemavathy, 2023] Nagendra, T. e Hemavathy, R. (2023). Unlocking kubernetes networking efficiency: Exploring data processing units for offloading and enhancing container network interfaces. Em *2023 4th IEEE Global Conference for Advancement in Technology (GCAT)*, p. 1–7.

[Nahar et al., 2024] Nahar, N., Andersson, K., Schelén, O. e Saguna, S. (2024). A survey on zero trust architecture: Applications and challenges of 6G networks. *IEEE Access*, 12:94753–94764.

[Oliveira et al., 2024] Oliveira, N. R. d., Santos, Y. d. R. d., Mendes, A. C. R., Barbosa, G. N. N., Oliveira, M. T. d., Valle, R., Medeiros, D. S. V. e Mattos, D. M. F. (2024). Storage standards and solutions, data storage, sharing, and structuring in digital health: A brazilian case study. *Information*, 15(1).

[Pooja e Chandrakala, 2024] Pooja, S. e Chandrakala, C. B. (2024). Secure reviewing and data sharing in scientific collaboration: Leveraging blockchain and zero trust architecture. *IEEE Access*, 12:92386–92399.

[Ramezanpour e Jagannath, 2022] Ramezanpour, K. e Jagannath, J. (2022). Intelligent zero trust architecture for 5G/6G networks: Principles, challenges, and the role of machine learning in the context of o-ran. *Computer Networks*, 217:109358.

[Rizvi et al., 2023] Rizvi, S., Zwerling, T., Thompson, B., Faiola, S., Campbell, S., Fisanick, S. e Hutnick, C. (2023). A modular framework for auditing iot devices and networks. *Computers & Security*, 132:103327.

[Satybaldy et al., 2024] Satybaldy, A., Ferdous, M. S. e Nowostawski, M. (2024). A taxonomy of challenges for self-sovereign identity systems. *IEEE Access*, 12:16151–16177.

[Sheikh et al., 2021] Sheikh, N., Pawar, M. e Lawrence, V. (2021). Zero trust using network micro segmentation. Em *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, p. 1–6.

[Shen e Shen, 2024] Shen, Q. e Shen, Y. (2024). Endpoint security reinforcement via integrated zero-trust systems: A collaborative approach. *Computers & Security*, 136:103537.

[Srinivas et al., 2019] Srinivas, J., Das, A. K. e Kumar, N. (2019). Government regulations in cyber security: Framework, standards and recommendations. *Future Generation Computer Systems*, 92:178–188.

[Stafford, 2020] Stafford, V. (2020). Zero trust architecture. *NIST special publication*, 800:207.

[Syed et al., 2022] Syed, N. F., Shah, S. W., Shaghaghi, A., Anwar, A., Baig, Z. e Doss, R. (2022). Zero trust architecture (zta): A comprehensive survey. *IEEE Access*, 10:57143–57179.

[Tankard, 2011] Tankard, C. (2011). Advanced persistent threats and how to monitor and deter them. *Network Security*, 2011(8):16–19.

[Teerakanok et al., 2021] Teerakanok, S., Uehara, T. e Inomata, A. (2021). Migrating to zero trust architecture: Reviews and challenges. *Security and Communication Networks*, 2021(1):9947347.

[Tsai et al., 2024] Tsai, M., Lee, S. e Shieh, S. W. (2024). Strategy for implementing of zero trust architecture. *IEEE Transactions on Reliability*, 73(1):93–100.

[Tuler De Oliveira et al., 2022] Tuler De Oliveira, M., Reis, L. H. A., Verginadis, Y., Mattos, D. M. F. e Olabarriaga, S. D. (2022). Smartaccess: Attribute-based access control system for medical records based on smart contracts. *IEEE Access*, 10:117836–117854.

[van Steen, 2025] van Steen, T. (2025). Developing a behavioural cybersecurity strategy: A five-step approach for organisations. *Computer Standards & Interfaces*, 92:103939.

[Yiliyaer e Kim, 2022] Yiliyaer, S. e Kim, Y. (2022). Secure access service edge: A zero trust based framework for accessing data securely. Em *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, p. 0586–0591.

[Zivi e Doerr, 2022] Zivi, A. e Doerr, C. (2022). Adding zero trust in BYOD environments through network inspection. Em *2022 IEEE Conference on Communications and Network Security (CNS)*, p. 1–6.

Capítulo 3

Introdução à Computação Quântica e Impactos em Criptografia

Victor Takashi Hayashi (USP), Bryan Kano Ferreira (USP), Reginaldo Arakaki (USP), Jonatas Faria Rossetti (Bradesco), Routo Terada (USP), Ever Costa (Inteli), Wildisley Filho (Inteli), Giovanna Vieira (Inteli), Luiza Petenazzi (Inteli), Priscila Falcão (Inteli)

Abstract

Quantum Computing is an emerging technology with the potential to solve some complex problems considered intractable by classical computers. However, this potential poses risks to information security, since quantum algorithms can break encryption methods widely used in current systems. This short course aims to introduce the fundamentals of quantum computing and examine its impacts on cryptography, supporting a deeper understanding of the reasons for these impacts based on an understanding of its mathematical foundations, in addition to providing a more comprehensive view of possible solutions beyond post-quantum cryptography. Given the relevance of the scenario of breaking asymmetric algorithms such as RSA, Quantum Computing is one of the main tools for understanding the need and importance of post-quantum cryptography and its standardization by NIST.

Resumo

A Computação Quântica é uma tecnologia emergente com potencial para resolver alguns problemas complexos considerados intratáveis por computadores clássicos. No entanto, esse potencial representa riscos à segurança da informação, uma vez que algoritmos quânticos podem quebrar métodos de criptografia amplamente utilizados em sistemas atuais. Este minicurso tem como objetivo introduzir os fundamentos da computação quântica e examinar seus impactos na criptografia, apoiando uma compreensão mais profunda das razões para esses impactos com base na compreensão de seus fundamentos matemáticos, além de fornecer uma visão mais abrangente de possíveis soluções além da criptografia pós-quântica. Dada a relevância do cenário de quebra de algoritmos de criptografia assimétrica como o RSA, a Computação Quântica é uma das principais ferramentas para a compreensão da necessidade e da importância da criptografia pós-quântica e de sua padronização pelo NIST.

3.1. Introdução

A Computação Quântica é uma tecnologia emergente com potencial para resolver alguns problemas complexos considerados intratáveis por computadores clássicos. Entretanto, esse potencial traz riscos à segurança da informação, já que algoritmos quânticos podem quebrar métodos de criptografia amplamente utilizados nos sistemas atuais [Gamble 2019, Khan et al. 2024].

Este minicurso tem como objetivo geral introduzir os fundamentos da computação quântica e examinar seus impactos em criptografia, suportando um aprofundamento maior nas razões para esses impactos a partir do entendimento de seus fundamentos matemáticos, além de fornecer uma visão mais abrangente sobre possíveis soluções além da criptografia pós-quântica, que são os principais diferenciais em relação aos minicursos apresentados em edições anteriores do SBSeg [Barreto et al. 2013, Paiva et al. 2023].

Além disso, o minicurso está alinhado com propósito similar ao workshop internacional ACM QSec[1], para fomentar a sinergia entre comunidades de pesquisa em Computação Quântica (principalmente aquelas relacionadas à Criptografia Quântica com *Quantum Key Distribution*) e de Segurança da Informação (que vêm pesquisando por muitos anos abordagens de Criptografia Pós-Quântica).

O minicurso destina-se a estudantes de graduação e pós-graduação, pesquisadores e profissionais de Computação, Engenharias e áreas afins, especialmente aqueles interessados em Segurança da Informação e em Computação Quântica. Dado o caráter interdisciplinar do tema de Impactos da Computação Quântica em Criptografia, que combina fundamentos de Mecânica Quântica, Matemática e Ciência da Computação, o conteúdo foi planejado para ser acessível a participantes com conhecimentos básicos de Ciência da Computação e Criptografia, não exigindo formação aprofundada em Mecânica Quântica. O público esperado inclui pessoas que desejam se atualizar sobre o impacto dos computadores quânticos na segurança da informação e conhecer as estratégias para proteger sistemas frente a essa tecnologia emergente.

Dada a relevância do cenário de quebra de algoritmos de criptografia assimétrica como o RSA e a padronização do NIST sobre criptografia pós-quântica [Alagic et al. 2022], o curso abordará desde os fundamentos da criptografia clássica na Seção 3.2, passando pelos princípios da computação quântica na Seção 3.3, até os potenciais ataques quânticos sobre algoritmos criptográficos atuais e as soluções em desenvolvimento para mitigar esses riscos na Seção 3.4. O tema será tratado de forma didática e acessível, conectando teoria e prática, de modo a ressaltar tanto a necessidade de atualizar os mecanismos de segurança diante da era quântica quanto as novas oportunidades tecnológicas que emergem desse contexto na Seção 3.5. As considerações finais com reflexões, aspectos éticos tangenciais e exemplos de capacitação na área são apresentadas na Seção 3.6.

3.2. Fundamentos de Criptografia

Nesta seção são apresentados os fundamentos de criptografia, trazendo os principais requisitos de segurança da informação que devem ser suportados pelos mecanismos cripto-

[1] `https://acm-qsec.com/`

gráficos: Confidencialidade, Integridade, Autenticidade e Irretratabilidade. Também são apresentados os fundamentos matemáticos de funções *hash* criptográficas, criptografia simétrica e criptografia assimétrica [Menezes et al. 2018].

3.2.1. Requisitos de Segurança

A Segurança da Informação é uma área multidisciplinar que tem como objetivo a proteção da informação e, consequentemente, dos sistemas que a processam contra ameaças que possam vir a comprometer seu valor. Uma definição usualmente adotada consta na legislação norte-americana, utilizada por órgãos como o *National Institute of Standards and Technology* (NIST), o *Department of Homeland Security* (DHS), o *Office of Management and Budget* (OMB) e a *Federal Information Security Modernization Act* (FISMA) como base para suas diretrizes técnicas:

> Segurança da Informação significa proteger informações e sistemas de informação contra acesso, uso, divulgação, interrupção, modificação ou destruição não autorizados, com o objetivo de assegurar:
>
> (A) **Integridade**, que significa proteger contra modificação ou destruição indevida da informação, incluindo a garantia de *irretratabilidade* e *autenticidade*;
>
> (B) **Confidencialidade**, que significa preservar as restrições autorizadas de acesso e divulgação, incluindo mecanismos de proteção da privacidade pessoal e de informações proprietárias;
>
> (C) **Disponibilidade**, que significa assegurar o acesso e uso oportunos e confiáveis da informação.
>
> [44 U.S. Code § 3542 – Definitions 2013]

Temos dois requisitos complementares à integridade. A *irretratabilidade*, também conhecida pelo termo, *não-repúdio*, pode ser definida como a propriedade que um emissor legítimo não possa negar a autoria de uma ação. A autenticidade, por sua vez, é a propriedade que garante que uma parte é, de fato, quem afirma ser [Menezes et al. 2018].

O esforço para assegurar os requisitos objetivados depende de ações coordenadas multidisciplinares que ultrapassam soluções técnicas isoladas [Stallings 2013]. De acordo com Anexo A da ISO/IEC 27001:2022, os controles necessários estão organizados em quatro categorias principais, sendo estes: controles organizacionais, relacionados a pessoas, físicos e tecnológicos. Em resumo, os controles de natureza organizacional concentram-se na formulação e condução de diretrizes e políticas de caráter estratégico e tático. Os controles voltados às pessoas abrangem dimensões relacionadas ao comportamento humano e à atribuição de responsabilidades. Já os controles físicos têm como foco a segurança dos espaços e dos ativos materiais. Por fim, os controles tecnológicos tratam de instrumentos técnicos empregados no contexto digital. Nesta última categoria, encontra-se a Criptografia.

3.2.2. Introdução à Criptografia

A Criptografia é a área do conhecimento que estuda as técnicas matemáticas que asseguram requisitos da Segurança da Informação [Menezes et al. 2018]. Tal definição pode ser

melhor especificada quando incluímos um aspecto central que a distingue de outras abordagens com o mesmo objetivo, que é o estudo de proteções que atuam diretamente sobre os dados, através de transformações na própria informação [Stallings 2013, Terada 2008]. Isto significa que a criptografia, como ferramenta, é um subconjunto de mecanismos técnicos que atuam na camada de dados, diferenciado-se de outros controles que protegem o contexto em que a informação circula, como *firewalls*, *antimalwares* e sistemas de autenticação de usuários.

A camada de Segurança de Dados por meio do uso da criptografia pode ser entendida como uma configuração primária em qualquer ecossistema de mecanismos técnicos de segurança. Essa visão é corroborada pelo conceito de defesa em profundidade, amplamente conhecido na indústria e referenciado pelo NIST *Cybersecurity Framework* [Pascoe 2023], onde essa atuação no nível mais baixo da pilha de segurança confere à criptografia um papel fundacional, em relação aos demais controles técnicos.

Usualmente, os requisitos de segurança tidos como objetivos a serem fornecidos por mecanismos criptográficos estão associados à confidencialidade, integridade, autenticação e irretratabilidade da informação [Menezes et al. 2018]. Embora, tradicionalmente a disponibilidade não fosse considerada uma propriedade diretamente assegurada pela criptografia, arquiteturas distribuídas têm ampliado esse escopo, revelando o papel de mecanismos criptográficos na manutenção da disponibilidade confiável de dados e serviços [Bonneau et al. 2015].

Os cenários adversariais considerados pelo estudo da criptografia tomam cada fase do ciclo de vida da informação como objeto de análise. Durante a transmissão (Figura 3.1), onde os dados são enviados de um ponto a outro através de um canal de comunicação, os principais problemas envolvem a interceptação da mensagem por terceiros (quebra da confidencialidade), a modificação do conteúdo em trânsito (quebra da integridade), a falsificação da identidade do remetente (quebra da autenticidade) e a negação do envio após a entrega (quebra da irretratabilidade). Com exceção da negação do envio após a entrega, que toma uma ação de má índole de uma parte legítima da comunicaçaõ, todos os outros problemas, se tomados como intencionais, podem ser englobados no cenário adversarial clássico em criptografia, onde uma parte terceira e desautorizada *C*, denominada na literatura como Carlos, tem acesso ao canal de comunicação [Terada 2008].

Tomando a fase do ciclo de vida da informação, onde a informação está em re-

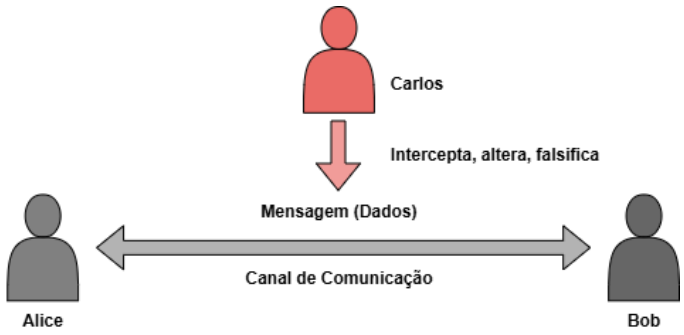


Figura 3.1. Cenário de Transmissão de Dados. Adaptado de [Terada 2008]

pouso, também chamada de armazenamento (Figura 3.2), esse cenário pode ser entendido como uma simplificação estática do problema da transmissão, na medida em que os dados permanecem parados, mas ainda sujeitos a ameaças semelhantes. Temos como problemas e cenários adversariais o acesso indevido por terceiros não autorizados à base de dados (quebra da confidencialidade) e a modificação ilegítima dos dados (quebra da integridade). Adicionalmente, para fins de auditabilidade e perícia, temos problemas relacionados à atribuição incorreta da autoria ou origem dos dados (quebra da autenticidade) e a tentativa de negar a autoria de um conteúdo previamente salvo (quebra da irretratabilidade). Uma modelagem clássica adversarial para o cenário está presente na Figura 3.2, onde o atacante deseja conseguir acesso à base de dados para perpetrar ações que podem impactar todos os requisitos de segurança considerados [Terada 2008].

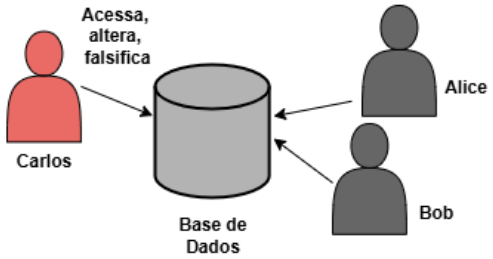


Figura 3.2. Cenário de Dados em Repouso. Adaptado de [Terada 2008]

Para fins de ciência, a etapa de processamento dos dados, embora inviável até o momento, também está sendo explorada pela comunidade acadêmica. Isso significa que os problemas considerados passam não só a considerar atacantes externos, mas mudam o paradigma de confiança no processador. Dessa forma, os dados permanecem encriptados em processamento. Esse conjunto de algoritmos criptográficos é conhecido como criptografia homomórfica [Gentry 2009].

Para assegurar todos os requisitos elencados nos múltiplos cenários adversariais, torna-se evidente a necessidade da composição de múltiplos mecanismos criptográficos. Em uma comunicação segura, por exemplo, utiliza-se criptografia assimétrica para troca de chaves, criptografia simétrica para proteger os dados, autenticação e assinaturas digitais para garantir a identidade. No armazenamento, combina-se criptografia de disco, funções *hash* criptográficas e assinaturas.

Esses mecanismos podem ser classificados, de forma simplificada, em três tipos de primitivas criptográficas (Figura 3.3). A primitiva sem o uso de chaves trata de mecanismos que não dependem de nenhum tipo de chave secreta para funcionar, como funções *hash* e geradores pseudoaleatórios. A primtiva de chave simétrica trata de mecanismos que utilizam a mesma chave secreta para encriptar e desencriptar. Nesta primitiva, destacam-se os encriptadores de dados como cifras de bloco e cifras de fluxo, além de autenticadores do tipo MACs (*Message Authentication Codes*). Por fim, a primitiva de chaves assimétricas utiliza um par de chaves, sendo uma pública e uma privada. Mecanismos assimétricos são geralmente empregados para o encapsulamento de chaves simétricas (*Key Encapsulation Mechanism - KEM*) e assinaturas digitais [Menezes et al. 2018].

Nas próximas seções exploraremos os fundamentos dos mecanismos mais usuais

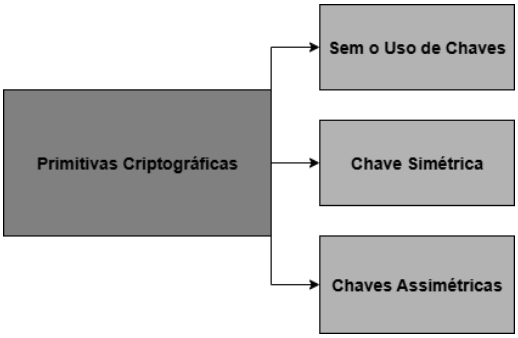


Figura 3.3. Primitivas Criptográficas de Segurança. Adaptado de [Menezes et al. 2018]

associados a cada tipo de primitiva.

3.2.3. Fundamentos Matemáticos de Funções Hash Criptográficas

Funções *hash* são os mecanismos mais usuais da primitiva de segurança sem o uso de chaves criptográficas. Originalmente foram concebidas dentro da área de Estrutura de Dados como estruturas associativas de chave e valor, utilizadas para otimizar o acesso eficiente a informações [Knuth 1997]. Formalmente, uma função *hash* pode ser definida como uma função determinística que produz uma saída de comprimento fixo.

$$H : \{0,1\}^* \rightarrow \{0,1\}^n \tag{1}$$

Onde H é a função *hash* propriamente dita, $\{0,1\}^*$ representa o conjunto de todas as cadeias binárias de comprimento arbitrário de entrada e $\{0,1\}^n$ representa o conjunto de todas as cadeias binárias de comprimento fixo de saída.

Nesse contexto primário, o uso dessas funções têm o fim de permitir operações rápidas de busca, inserção e remoção em bancos de dados. Nesse uso, não há preocupações com adversários ou segurança formal, colisões são aceitáveis e tratadas de forma eficiente. A transição do uso das funções para uso como mecanismos criptográficos surge em trabalhos seminais nas décadas de 70 e 80 [Merkle 1979, Lamport 1981]. Essa mudança de paradigma, inicialmente, deslocou o foco da indexação eficiente de dados para a verificação da integridade de dados, criando o tipo de função *hash* que pode ser classificado como MDCs (*Modification Detection Codes*) [Stallings 2013].

Em termos objetivos, se determinado conjunto de dados x é alterado resultando em x', então a saída da função *hash* $H(x')$ também será alterada de forma significativa em relação à saída original de $H(x)$, mesmo que a modificação em x seja mínima [Menezes et al. 2018]. Essa propriedade é chamada de efeito avalanche e é essencial para detectar alterações acidentais ou maliciosas [Terada 2008].

Tratando-se, agora, de uma ferramenta de segurança, tornou-se necessário uma formalização e consolidação dos requisitos de segurança que uma função *hash* criptográfica deve possuir [Menezes et al. 2018]:

1. **Resistência à pré-imagem**: dado um valor de saída $h \in \{0,1\}^n$, é difícil encontrar qualquer entrada $x \in \{0,1\}^*$ tal que $H(x) = h$.

2. **Resistência à segunda pré-imagem**: dado um valor de entrada $x \in \{0,1\}^*$, é difícil encontrar outro valor $x' \neq x$ tal que $H(x') = H(x)$.

3. **Resistência a colisões**: é difícil encontrar quaisquer dois valores distintos $x, x' \in \{0,1\}^*$ tais que $H(x) = H(x')$.

Esses requisitos elencados não são apenas teóricos, mas estão associados à proteção contra ataques reais. A resistência à pré-imagem é essencial para defesa contra ataques de inversão, como nos casos de armazenamento de senhas, onde mesmo que os *hashes* sejam expostos, não deve ser possível recuperar as senhas originais. Na quebra da segunda pré-imagem, o atacante reage a uma mensagem já existente; na quebra de colisão, ele atua proativamente, criando duas mensagens distintas que compartilham o valor *hash*. A quebra de ambas implicam em uma miríade de potenciais ações maliciosas, como falsificação de documentos, manipulação de registros e evasão de verificação de integridade. Ataques viáveis a esses dois requisitos de funções *hash* criptográficas já foram demonstrados na prática contra funções amplamente utilizadas como MD5 e SHA-1 [Wang et al. 2004, Wang et al. 2005], descartadas posteriormente.

A família de algoritmos SHA-3 é o padrão mais recente de funções *hash* criptográficas aprovado pelo NIST [FIPS PUB 202 2015]. Enquanto funções como SHA-1 e SHA-2 têm como base a construção de Merkle–Damgård, o SHA-3 utiliza a construção esponja do algoritmo Keccak [Katz and Lindell 2014]. Essa construção é diferente das abordagens anteriores por operar sobre um estado interno fixo dividido em duas partes, a *rate* (r) e a *capacity* (c), de forma que $r + c = b$, onde b representa o tamanho total do estado, usualmente 1600 *bits*. Na fase de absorção, a operação XOR é utilizada para combinar os blocos da entrada com os *r bits* do estado interno, posteriormente ocorrendo uma permutação não linear. Após toda a entrada ser processada, a fase de extração (em inglês, *squeezing*) gera o valor de saída a partir dos mesmos *r bits*. Na prática, esse funcionamento confere propriedades de segurança adequadas. Por conta da estrutura modular, a construção esponja ainda permite a adição de outros mecanismos como funções de autenticação do tipo KMAC (*Keccak Message Authentication Code*), consolidando sua versatilidade e segurança.

A capacidade de representar grandes volumes de dados usando saídas compactas aliada a sua simplicidade conceitual e eficácia computacional, consolida esse tipo de mecanismo como ferramenta essencial para assegurar a integridade dos dados em um ecossistema de Segurança da Informação.

3.2.4. Fundamentos Matemáticos da Criptografia Simétrica

O conceito principal que permeia os algoritmos de criptografia simétrica é a utilização de uma mesma chave para o processo de encriptação e desencriptação de dados [Terada 2008]. Apesar da primitiva de chave simétrica embarcar não só encriptadores, como também funções *hash* de tamanho arbitráriocomo os MACs (*Message Authentication Codes*), sistemas de asssinatura, sequências pseudoaleatórias e primitivas de identificação [Menezes et al. 2018], o foco desta subseção é exclusivamente didático e visa apresentar os fundamentos dos encriptadores simétricos, que constituem a principal aplicação prática dessa classe de algoritmos.

A criptografia simétrica foi historicamente formulada para assegurar a confidencialidade de dados em cenários adversariais, tanto para o uso durante a transmissão de informações quanto no armazenamento de dados sensíveis [Stallings 2013]. Nesse modelo, uma única chave primária é compartilhada entre as partes autorizadas, permitindo que todas possam encriptar e desencriptar os dados enviados ou guardados. O pressuposto de segurança deste tipo de mecanismo está em assegurar o segredo da chave compartilhada. A quebra ou acesso à chave implica na quebra total da segurança do sistema. Um sistema de criptografia simétrica pode ser denotado conceitualmente como uma tríade de algoritmos:

$$\mathsf{Gen}(1^n),\ \mathsf{Enc}_k(m),\ \mathsf{Dec}_k(c),$$

onde $\mathsf{Gen}(1^n)$ é o algoritmo de geração de chave, que recebe um parâmetro de segurança n e retorna uma chave secreta k a ser compartilhada entre as partes. $\mathsf{Enc}_k(m)$ é o algoritmo de encriptação, que recebe a chave k e uma mensagem m, e tem como saída um encriptado c. $\mathsf{Dec}_k(c)$ é o algoritmo de desencriptação, que têm como entrada a chave secreta k e o encriptado c, retornando a mensagem original m [Katz and Lindell 2014, Terada 2008].

A Figura 3.4 mostra a aplicação do modelo simétrico em um cenário adversarial clássico, onde após a combinação da chave secreta, Alice realiza a encriptação da mensagem m que deseja enviar com o algoritmo Enc e a chave secreta k, gerando o encriptado c que é enviado para Bob através de um canal comprometido pelo atacante Carlos. Bob, ao receber c, utiliza o algoritmo de desencriptação Dec junto com a chave secreta k para recuperar a mensagem às claras m e ter acessso aos dados originais. Note que neste modelo, Carlos só possui acesso ao encriptado c, não tendo o acesso à informação verdadeira.

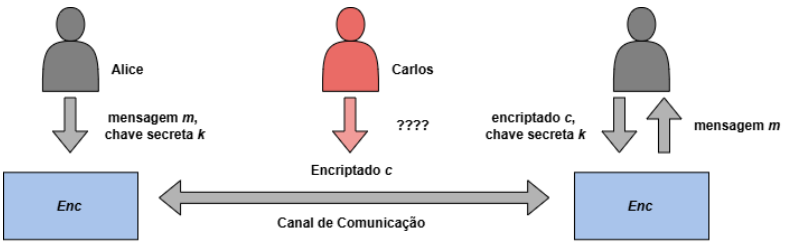


Figura 3.4. Modelo Simétrico de Criptografia. Adaptado de [Terada 2008]

Do ponto de vista de segurança, os algoritmos de criptografia simétrica buscam assegurar que a relação entre a chave secreta k, a mensagem m e o encriptado c seja suficientemente obscura matematicamente, de modo a inviabilizar qualquer ataque de reversão sem a posse da chave [Menezes et al. 2018]. Para isso, duas propriedades fundamentais postuladas por Claude Shannon são objetivadas: confusão, tem como objetivo tornar a relação entre a chave secreta e o encriptado o mais não-linear possível; difusão, por sua vez, busca espalhar a influência de cada *bit* da mensagem por vários *bits* do encriptado, viabilizando o efeito avalanche e aumentando a entropia [Terada 2008].

Essas duas propriedades são usualmente implementadas por meio de duas técnicas elementares em sistemas simétricos utilizadas desde o início da história da criptografia [Singh 1999], a substituição e a transposição. A substituição é uma operação criptográfica

em que os elementos da mensagem original, como *bits* ou *bytes* são trocados por outros segundo uma regra definida. A transposição é uma técnica que reorganiza a ordem dos elementos da mensagem ou de intermediários no processamento. Os *bits* ou blocos permanecem os mesmos, mas suas posições são alteradas [Menezes et al. 2018]. Algoritmos modernos contam com uma composição inteligente desses dois tipos de operações.

O AES (*Advanced Encryption Standard*) padronizado como encriptador simétrico de dados pelo NIST [of Standards et al. 2023], e amplamente adotado pela indústria, trabalha através de rodadas (em inglês, *rounds*) de procedimentos eficientes de substituição e transposição. O método SubBytes tem como objetivo aplicar uma transformação linear sobre cada *byte* do bloco de dados. O principal componente utilizado nesta etapa é a $\mathsf{S-box}$ que é uma tabela de substituição baseada em inverso multiplicativo e transformação afim, que envolvem aritmética modular em corpo finito e operações booleanas com XOR, sendo extremamente eficientes computacionalmente [Menezes et al. 2018]. O ShiftRows baseia-se em transposição, rotacionando ciclicamente cada linha da matriz de estado. O método de MixColumns realiza uma mistura linear dos *bytes* de cada coluna, ao invés das linhas, da matriz. Por fim, a operação AddRoundKey baseia-se em aplicar a operação XOR entre o bloco de dados e uma subchave derivada da chave secreta, preparando o novo *round*.

Combinando essas transformações em múltiplas rodadas, o AES representa uma aplicação segura dos princípios de Shannon na prática da criptografia simétrica moderna. Em resumo, esquemas simétricos tem sua principal vantagem na simplicidade de suas operações e na alta performance computacional, sendo ideais para a encriptação de grande volume de dados [Menezes et al. 2018]. Porém, seu modelo pressupõe o compartilhamento seguro da chave secreta que torna sua aplicação exclusiva inviável em contextos com o estabelecimento frequente e dinâmico de conexões, como a Internet. Esse é o problema conhecido como compartilhamento de chaves e para endereçá-lo foi criada a primitiva de chaves assimétricas que é discutida na próxima subseção.

3.2.5. Fundamentos Matemáticos da Criptografia Assimétrica

Como discutido na subseção 3.2.4, apesar de encriptadores simétricos serem altamente eficientes, o modelo pressupõe e tem como limitação que todas as partes envolvidas na comunicação compartilhem previamente a chave secreta k, idealmente, por algum tipo de canal seguro. Em ambientes militares ou para uso em armazenamento, isso pode ser viável. Porém com a democratização da tecnologia e com o crescimento da Internet e outros tipos de redes distribuídas, tornou-se inviável estabelecer esses canais prévios de forma segura. O problema do compartilhamento de chaves tornou-se o desafio a ser superado. Nesse cenário, a criptografia de chave pública surge como uma solução [Menezes et al. 2018].

Introduzida seminalmente por meio de um protocolo de combinação de chave que permitia que duas partes estabelecessem um segredo comum [Diffie and Hellman 2022], e posteriormente, com a apresentação do algoritmo RSA [Rivest et al. 1978], que ampliou as funcionalidades e aplicações permitindo a encriptação direta de mensagens e a criação de assinaturas digitais, essa abordagem modificou o rumo da criptografia.

A criptografia assimétrica se baseia no uso de duas ou mais chaves com ações

diferentes. No caso de encriptadores com o fim de confidencialidade, temos o uso da chave pública *pk* para encriptação e o uso de uma chave privada *sk* para desencriptação. Um sistema de encriptação assimétrica pode ser denotado de forma generalista como uma tríade de algoritmos,

$$\mathsf{Gen}(1^n),\ \mathsf{Enc}_{pk}(m),\ \mathsf{Dec}_{sk}(c),$$

onde $\mathsf{Gen}(1^n)$ é o algoritmo que, a partir de um parâmetro de segurança n, gera um par de chaves, pública e privada (pk, sk). O algoritmo $\mathsf{Enc}_{pk}(m)$ é utilizado junto à chave pública para encriptar uma mensagem m, gerando o encriptado c. Ao usar o algoritmo $\mathsf{Dec}_{sk}(c)$ com a chave privada sk no encriptado c, a mensagem original m é recuperada [Katz and Lindell 2014, Terada 2008].

A Figura 3.5 exibe a aplicação do modelo assimétrico no modelo adversarial de confidencialidade. Alice, ao enviar uma mensagem m, utiliza a chave pública pk de Bob e o algoritmo de encriptação Enc para gerar um encriptado c, que é enviado através do canal de comunicação. Bob, ao receber c, realiza a desencriptação com uso de sua chave privada sk no algoritmo de desencriptação Dec, conseguindo a mensagem original m. Nesse esquema, Bob e Alice não precisaram combinar uma chave previamente: Alice usa a chave pública de Bob. Carlos, ao tentar interceptar a mensagem, consegue apenas o encriptado c que não possui nenhuma informação útil. Note que ao contrário de um esquema simétrico, a encriptação assimétrica não funciona para ambos os lados de forma direta. Se Bob deseja mandar uma mensagem para Alice, ele terá que obter a chave pública dela.

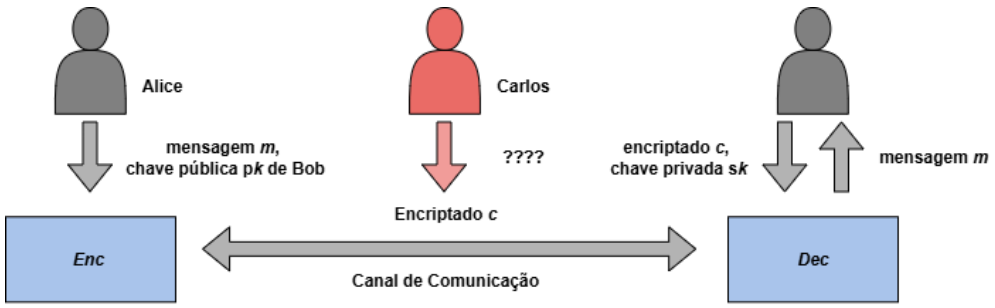


Figura 3.5. Modelo Assimétrico de Criptografia. Adaptado de [Terada 2008]

Invertendo o fluxo de encriptação, é possível derivar outro uso da criptografia assimétrica, a construção de assinaturas digitais, que tem como objetivo garantir autenticidade de dados. Nesse tipo de mecanismo, o remetente utiliza sua própria chave privada (sk) para assinar a mensagem, e qualquer terceiro, como Bob, pode verificar a autenticidade dessa assinatura utilizando a chave pública (pk) disponibilizada. Adicionalmente, a criptografia assimétrica é empregada em protocolos de acordo de chaves, que não envolvem encapsulamento via encriptação, além de esquemas de desafio-resposta, como os protocolos de Conhecimento Zero (ZK, do inglês *Zero Knowledge*) [Katz and Lindell 2014].

A ideia para construção de esquemas com base na primitiva assimétrica é o uso de uma função que é fácil de calcular em uma direção, encriptação por exemplo, mas difícil de inverter (desencriptação) sem uma informação necessária, a chave secreta *sk* [Menezes et al. 2018]. Essa assimetria de complexidade motivou a busca por problemas matemáticos computacionalmente difíceis, mas que permitissem uma solução eficiente quando a informação secreta está disponível. Diversas classes de problemas foram exploradas pela comunidade, as mais adotadas contemplam três tipos de abordagens.

O problema da fatoração de inteiros pode ser definido como: dado um número grande $N = pq$, onde p e q são primos grandes, o objetivo é encontrar p e q sem conhecer previamente um deles [Stallings 2013, Katz and Lindell 2014]. Isso era tido como computacionalmente inviável até a chegada do paradigma quântico. Porém, se um dos fatores é conhecido, a operação é trivial [Bernstein et al. 2009]. O problema do logaritmo discreto toma que dado um grupo finito G, um elemento $g \in G$ e outro elemento $h \in G$, o objetivo é encontrar um inteiro x tal que $g^x = h$. Uma variante do mesmo problema é o logaritmo discreto em curvas elípticas, que é mais eficiente para certas aplicações.

O problema fundador da fatoração dos inteiros é a base que sustenta o RSA [Rivest et al. 1978]. Para fins didáticos, o funcionamento simplificado e sem padronização do criptossistema é descrito abaixo [Terada 2008]. O processo é iniciado através da geração do par de chaves pública-privada (*pk* e *sk*):

1. Dois primos grandes p e q são escolhidos.
2. O módulo N comum das operações é calculado por $N = pq$.
3. A função totiente de Euler é calculada por $\phi(N) = (p-1)(q-1)$.
4. O expoente público e é escolhido sendo um inteiro tal que $1 < e < \phi(N)$ e $\gcd(e, \phi(N)) = 1$.
5. O expoente privado d é calculado tal que $d \equiv e^{-1} \ (mod\ \phi(N))$, isto é, o inverso multiplicativo de e módulo $\phi(N)$.

A chave pública resultante é o par (N, e) e a chave privada é (N, d). Para realizar a encriptação de uma mensagem $m \in \mathbb{Z}_N$, basta o remetente utilizar o expoente público e para calcular $c = m^e \bmod N$. É de trivial entendimento que a mesagem m pode ser recuperada com o uso do expoente secreto d através de $m = c^d \bmod N$.

O mesmo paper seminal [Rivest et al. 1978] mostra, ainda, como um esquema de assinatura digital pode ser derivado. A assinatura de um valor *hash* h de uma mensagem ou documento m pode ser feita utilizando a chave privada $sk = d$ por meio do cálculo $\sigma = h^d \ mod\ N$. Para verificação, basta o receptor receber m e σ, calcular $h = \sigma^e \ mod\ N$ e $h' = H(m)$, e fazer a comparação $h \stackrel{?}{=} h'$ para decidir se a assinatura é válida ou não. Note que o sistema é seguro porque fatorar N é difícil, mas se já se conhece p e q, todo o restante se torna trivial.

Diversos outros esquemas assimétricos foram desenvolvidos a partir das outras abordagens de escolha do problema. A abordagem do logaritmo resultou em algoritmos como ElGamal e DSA. Sua variação em curvas elípticas (ECC - *Elliptic Curve Cryptography*), fundamentou esquemas como o ECDSA (*Elliptic Curve Digital Signature Algorithm*) e ECDH (*Elliptic Curve Diffie–Hellman*), que são utilizados em protocolos

modernos como TLS (*Transport Layer Security*). Seus fundamentos são similares, mas apresentam diferenças de eficiência em cenários diferentes.

Usualmente, sistemas assimétricos são mais complexos em suas formulações quando comparados com sistemas simétricos. Enquanto os últimos apresentam operações eficientes (como XOR, substituições e permutações), sistemas assimétricos precisam garantir o artifício seguro de correlação entre as chaves, utilizando operações como exponenciação, custosas computacionalmente [Terada 2008]. Dessa forma, em termos práticos, a criptografia assimétrica não é utilizada para processar grandes volumes de dados. Esquemas assimétricos são combinados com sistemas simétricos, assumindo o papel de Mecanismos de Encapsulamento de Chave (KEM - *Key Encapsulation Mechanism*), responsáveis pela entrega da chave simétrica entre as partes [Katz and Lindell 2014].

A segurança dos algoritmos assimétricos está ligada de forma fundamental à dificuldade desses problemas em condições adversariais, ou seja, sem a posse da chave secreta. Isso significa que a segurança dos sistemas assimétricos toma a inviabilidade computacional da resolução dos problemas em tempo hábil como pressuposto. Se o problema computacional é resolvido em tempo hábil, o sistema é quebrado em sua premissa fundamental [Menezes et al. 2018].

3.3. Introdução à Computação Quântica

Esta seção apresenta uma visão geral da computação quântica, noções básicas de mecânica quântica, análise de algoritmos e complexidade computacional, além de conceitos sobre *qubits* e seus estados, portas quânticas, circuitos quânticos e como a informação pode ser representada e manipulada através dessas estruturas [Nielsen and Chuang 2010]. Serão exercitados circuitos e algoritmos quânticos utilizando ferramentas da IBM (exemplos disponíveis em `https://github.com/vthayashi/quantum-crypto`).

3.3.1. Conceitos Básicos

Na computação tradicional, agora chamada de clássica para a diferenciar da quântica, utilizamos algoritmos clássicos para resolver problemas computacionais. Esses algoritmos, ao serem implementados como programas e suportados por diversas camadas de abstração, modificarão os estados dos *bits* até obtermos uma possível solução para o problema. O *bit* é a unidade básica de informação clássica, podendo representar em determinado instante apenas um de dois valores possíveis, usualmente denotados como 0 e 1. Isso constitui uma abstração, pois naturalmente o *bit* não possui existência física: ele precisa ser implementado por meio de algum sistema físico específico. A forma predominante de se fazer isso na computação clássica é utilizando o transistor em conjunto com outros dispositivos eletrônicos. O transistor, apesar de ser construído a partir de semicondutores cuja descrição e entendimento completo só é possível com a mecânica quântica, opera sob um regime clássico e podemos descrevê-lo e utilizá-lo valendo-se de fenômenos clássicos como corrente elétrica, tensão elétrica, resistência elétrica, entre outros, conforme estabelecidos pela teoria eletromagnética.

Por outro lado, a computação quântica constitui-se através da combinação da ciência da computação com a mecânica quântica, tendo por objetivo construir sistemas computacionais que possam explorar fenômenos quânticos como superposição, emara-

nhamento e interferência para realizar computações úteis. Abaixo, temos uma breve intuição sobre cada um desses fenômenos, considerados principais dentro da computação quântica. Entretanto, como nosso foco será na parte computacional, uma introdução técnica à mecânica quântica pode ser encontrada em [Susskind and Friedman 2014] e uma exposição mais avançada em [Sakurai and Napolitano 2020]:

- **Superposição:** possibilidade de um sistema quântico ser descrito através da combinação de múltiplos estados (ou configurações) durante sua operação. Assim, para um sistema de dois níveis, além de poder assumir estados bem definidos, como 0 ou 1, pode apresentar configurações que são combinações desses dois estados de acordo com uma certa distribuição de probabilidades.

- **Emaranhamento:** um tipo especial de superposição envolvendo os estados de dois ou mais sistemas quânticos, de forma a não ser mais possível descrever cada sistema de forma independente dos demais, criando correlações. Assim, o sistema completo não pode ser totalmente descrito em termos de cada parte individual.

- **Interferência:** quando o estado de dois ou mais sistemas quânticos são combinados para formar um novo estado composto, podemos ter interações entre eles, com cada sistema influenciando o outro, que amplificam a probabilidade associada a certos estados, chamada de interferência construtiva, ou diminuem/eliminam a de outros, chamada de interferência destrutiva.

Para fazer a exploração desses fenômenos para computar, construímos dispositivos baseados em sistemas quânticos diversos (e.g., átomos, íons, elétrons, fótons), além de outros componentes, com comportamentos distintos do transistor (há diversas abordagens para se construir computadores quânticos, porém este tópico está além do escopo deste trabalho; informações podem ser obtidas em [Kasirajan 2021]). A partir desses dispositivos, operando como um sistema de dois níveis, emerge uma abstração chamada *qubit (quantum bit)*, que é a unidade básica de informação quântica, podendo representar em determinado instante tanto estados bem definidos como 0 e 1 quanto estados que são superposições deles. Com a computação quântica podemos projetar novos algoritmos que são fundamentalmente distintos dos algoritmos clássicos. Os *qubits* serão então manipulados pelos algoritmos quânticos de forma a resolver problemas computacionais de uma maneira diferente. Ao mudarmos os fenômenos físicos que alicerçam a computação, abrimos novas possibilidades, o que pesquisas já realizadas nas últimas décadas nos mostraram que permite à computação quântica resolver certos problemas computacionais de forma mais eficiente do que a computação clássica [Shafique et al. 2024].

3.3.2. Análise de Algoritmos e Complexidade Computacional

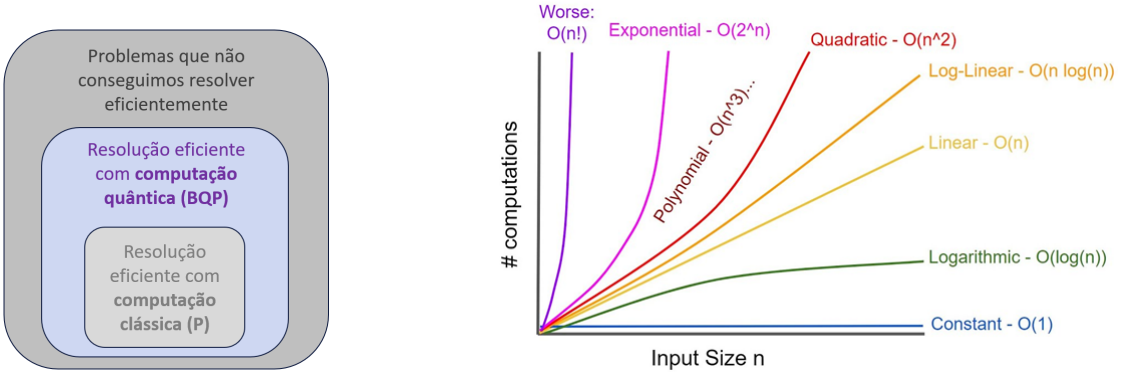
Ao falarmos sobre eficiência de algoritmos, estamos nos referindo ao seu padrão de consumo de recursos computacionais em função do tamanho do problema que se deseja resolver, uma análise matemática que é possível realizar e que nos permite classificá-los em termos de seu comportamento. Importante frisar a diferença dessa ideia para a de desempenho de programas, ou seja, das implementações de algoritmos utilizando determinadas

linguagens de programação, compiladores, sistemas operacionais, até sua conversão completa para a linguagem de máquina a ser executada pelo hardware (ou uma implementação diretamente em hardware), fatores estes que podem impactar negativamente o comportamento de um algoritmo na prática e serem mitigadores de desempenho. Ao longo da discussão sobre o potencial da computação quântica, seja neste texto ou na literatura em geral, muito se fala em termos da eficiência dos algoritmos quânticos conhecidos mesmo que ainda não existam computadores quânticos de larga escala e tolerantes a falhas para implementá-los e concretizar esse potencial.

Dessa forma, um dos objetivos durante a análise de algoritmos é estimar a quantidade de recursos necessários para resolver determinado problema conforme o tamanho da entrada varia. Também de interesse é a análise comparativa de algoritmos, onde dados ao menos dois algoritmos diferentes, identificamos qual deles é mais eficiente para resolver determinado problema. Esse estudo dos recursos necessários para executar um algoritmo pode empregar como métricas recursos temporais (e.g., número de passos computacionais ou de operações básicas necessárias) e espaciais (e.g., a quantidade de memória). A medida exata do tempo de processamento ou de utilização de memória só será possível após a implementação e execução do algoritmo em um computador real. A partir daqui, o texto irá priorizar a métrica de tempo, chamada de complexidade temporal, e por meio de uma análise matemática formal do algoritmo, tentaremos encontrar uma função matemática $f(n)$ que represente seu consumo de recursos em relação ao tamanho n da entrada.

A análise assintótica é uma ferramenta matemática que nos permite determinar essa função e tirar conclusões sobre o comportamento do algoritmo, contando o número de passos computacionais a serem executados conforme n varia. Essa análise não depende de detalhes de implementação, nos dando um indicativo sobre o comportamento desse algoritmo quando n se torna muito grande. Existem 3 notações básicas para análise assintótica de funções: Big-O, Big-Ω e Big-Θ. Vamos defini-las brevemente, porém nosso uso neste minicurso se limitará à notação Big-O. Outras notações e aprofundamentos teóricos podem ser encontrados em [Sipser 2012]:

- **Big-O**: representa o limite superior para o consumo de recursos de um algoritmo. Geralmente utilizada para estudar o comportamento do algoritmo no pior caso. Formalmente, dizemos que $f(n) = O(g(n))$ se tivermos constantes $c > 0$ e $n_0 > 0$ tais que:
$$f(n) \leq cg(n) \quad \forall n \geq n_0 \tag{2}$$
- **Big-Ω**: representa o limite inferior para o consumo de recursos de um algoritmo. Geralmente utilizada para estudar o comportamento do algoritmo no melhor caso. Formalmente, dizemos que $f(n) = \Omega(g(n))$ se tivermos constantes $c > 0$ e $n_0 > 0$ tais que:
$$f(n) \geq cg(n) \quad \forall n \geq n_0 \tag{3}$$
- **Big-Θ**: representa simultaneamente os limites inferior e superior para o consumo de recursos de um algoritmo. Geralmente utilizada para estudar o comportamento no caso médio. Formalmente, dizemos que $f(x) = \Theta(g(n))$ se tivermos constantes $c > 0$, $d > 0$ e $n_0 > 0$ tais que:
$$cg(n) \leq f(x) \leq dg(n) \quad \forall n \geq n_0 \tag{4}$$



(a) Classificação simplificada dos problemas computacionais.

(b) Exemplos de funções em notação Big-O. Fonte: [Salvi 2023]

Figura 3.6. Classificação de problemas e complexidade computacional

Uma das propriedades interessantes da notação Big-O é que ao considerar duas funções tais que $f(n) = O(g(n))$ e $x(n) = O(y(n))$, então teremos que $f(n) + x(n) = O(max\{g(n), y(n)\})$. A mesma ideia se aplica para um número arbitrário de funções. Além disso, essa notação nos indica a ordem de magnitude dessa função, ou seja, o termo dominante, para representar o que acontece com a função $f(n)$ quando n cresce de forma arbitrária. Por exemplo, se para determinado algoritmo A_1 tivermos $f_1(n) = n+1$ diremos que esse algoritmo é $O(n)$ ou também que $f_1(n) = O(n)$. No caso de um algoritmo A_2 ter $f_2(n) = n^2 + 4n + 3$ diremos que $f_2(n) = O(n^2)$. Comparando os dois, concluímos que o algoritmo A_1 é o mais eficiente, pois sua função de consumo de recursos cresce mais lentamente (é possível verificar isso através do cálculo de limites e derivadas). Como a notação Big-O denota um limite superior, não importa qual o computador real utilizado, o consumo de recursos de um determinado algoritmo nunca será melhor do que esse limite para valores de n muito grandes. A mesma ideia vale para a comparação de algoritmos: não importa o contexto prático de implementação, dado que ambos são executados sob as mesmas condições, o algoritmo A_1 será mais rápido do que A_2 para n suficientemente grande, que é o caso assintótico. Naturalmente, para valores pequenos podem até ocorrer variações.

Sabemos que a computação clássica pode resolver certos tipos de problemas de forma eficiente (classe de problemas P), enquanto a computação quântica expande essa possibilidade (classe de problemas BQP) abarcando alguns problemas considerados insolúveis por computadores clássicos, como a fatoração de números inteiros [Shor 1997]. Entretanto, vale frisar que a computação quântica não vai resolver todos os problemas considerados intratáveis atualmente. A Figura 3.6(a) representa de maneira simplificada as relações citadas acima, sendo uma interpretação lúdica dos diagramas de classes de complexidade computacional e a Figura 3.6(b) mostra alguns dos principais tipos de funções que representam o padrão de consumo de recursos, em notação Big-O, utilizadas na análise de algoritmos.

Por fim, quando falamos sobre resolução eficiente de um problema computacional, estamos nos referindo a algoritmos cuja complexidade computacional é polinomial ou melhor (o consumo de recursos cresce mais lentamente), enquanto as demais, como a

exponencial, consideramos ineficientes [Sipser 2012]. Naturalmente, um polinômio com grau muito alto não pode ser considerado eficiente na prática. Assim, mesmo um algoritmo com complexidade polinomial pode resultar num desempenho inadequado após ser implementado dependendo do tamanho do problema, sendo preferidos aqueles que apresentem complexidades constante, logarítmica ou linear. Porém, restringiremos nossa discussão à distinção entre eficientes e ineficientes em termos de complexidade computacional. Além disso, qualquer problema computacional para o qual se conhece um algoritmo pode ser fácil de resolver se o tamanho da entrada for suficientemente pequeno. As diferenças se tornam mais significativas conforme n cresce rumo ao regime assintótico, revelando a verdadeira natureza do comportamento do algoritmo.

3.3.3. Estados, Portas e Circuitos Quânticos

Nesta subseção, apresentamos uma breve introdução ao formalismo matemático e conceitual da computação quântica. Para aprofundamentos e exposições mais gerais, consultar os trabalhos de [Nielsen and Chuang 2010, Watrous 2025].

Utilizamos a notação de Dirac [Watrous 2025] para representar matematicamente os estados quânticos e seus operadores na computação quântica. Um vetor ψ escrito como $|\psi\rangle$ é chamado de *ket* e escrito como $\langle\psi|$ é chamado de *bra* nessa notação. Podemos representar um vetor na forma de coluna, correspondente ao *ket*, ou na forma de linha, corresponde ao *bra*, conforme exemplificado abaixo para dois vetores **v** e **u**:

$$|v\rangle = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} \qquad \langle u| = \begin{pmatrix} u_1 & u_2 & \cdots & u_n \end{pmatrix} \tag{5}$$

Na computação quântica, usualmente temos dois estados que representam quanticamente nossos conhecidos valores 0 e 1, agora chamados de $|0\rangle$ e $|1\rangle$ (dois estados ortogonais e normalizamos que juntos formam o que chamamos de base computacional, que será nossa referência para descrevermos os estados e resultados durante a execução de circuitos quânticos). Na maior parte do tempo, vamos utilizar os *kets*, com os *bras* aparecendo apenas quando necessários em determinadas operações. Dessa forma, os vetores coluna que representam esses estados da base computacional são definidos como:

$$|0\rangle := \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle := \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{6}$$

Representaremos os estados quânticos utilizando vetores e os operadores, que modificam tais estados, por meio de matrizes. Essa formulação é suficiente para compreendermos as bases da área e a maior parte dos algoritmos quânticos. Vale notar que há uma descrição mais geral do que essa, utilizada na teoria quântica da informação, que faz uso de matrizes densidade para descrever os estados quânticos, ferramental necessário quando desejamos modelar o efeito do ruído das interações dos nossos sistemas com o ambiente a sua volta [Watrous 2025]. Porém, essa abordagem generalizada não faz parte do escopo deste texto, pois não é necessária para nossos propósitos. Assim, o estado quântico de

um *qubit* pode ser então representado de maneira geral como uma combinação linear (ou superposição) dos estados de uma base, neste exemplo a base computacional:

$$|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha|0\rangle + \beta|1\rangle \quad , \quad \alpha, \beta \in \mathbb{C} \tag{7}$$

Os números complexos α e β são chamados de amplitudes de probabilidade e é a partir deles que conseguimos construir superposições, explorar interferências e calcular as probabilidades de se obter cada estado da base após o processo de medição. Esse processo nos permite extrair o resultado de uma computação a partir da interação entre um equipamento de medida e o computador quântico de maneira a determinar em qual configuração ele se encontra e associá-la a um determinado estado quântico de referência. A partir desse resultado, podemos associar o estado identificado com uma sequência de *bits* que representam em binário a saída do computador quântico para determinada computação. Dessa forma, o resultado obtido após a medição sempre será um valor específico, nunca obteremos superposições, e a probabilidade de o obtermos estará de acordo com a respectiva amplitude de probabilidade. Naturalmente, só conseguiremos visualizar experimentalmente tal probabilidade após repetir a execução diversas vezes. Após a medição, o estado do qubit se torna bem determinado e será o mesmo para medições subsequentes, a não ser que o modifiquemos deliberadamente, que alteremos a configuração do aparato de medida ou que ruídos no sistema causem alguma mudança inadvertida. Para obter a probabilidade de medir cada dos estados da base de referência, calcula-se o módulo de cada número complexo e eleva-se o resultado ao quadrado. Pela Regra de Born [Watrous 2025], que nos fornece uma condição de normalização, devemos ter todas essas probabilidades somando para a unidade conforme a relação $|\alpha|^2 + |\beta|^2 = 1$.

As mesmas ideias se aplicam para sistemas formados por múltiplos *qubits*, com seu estado e condição de normalização sendo representados como (para *n qubits* teremos no máximo 2^n estados na superposição):

$$|\psi\rangle = \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{2^n-1} \end{pmatrix} = \sum_{j=0}^{2^n-1} c_j|j\rangle \quad , \quad \sum_{j=0}^{2^n-1} |c_j|^2 = 1 \quad , \quad c_j \in \mathbb{C} \tag{8}$$

Em sistemas de múltiplos *qubits*, precisamos definir a ordenação que será utilizada para eles. No nosso caso, os *qubits* menos significativos serão representados por índices menores. Vamos também ordená-los da esquerda para a direita em ordem decrescente, ou seja, se tivermos um estado $|011\rangle$ fica subentendido que ele representa o estado composto dos *qubits* $q_2q_1q_0$, com q_2 sendo o qubit mais significativo e q_0 sendo o menos significativo. Disso decorre uma outra possibilidade de representação, conforme utilizado na Equação 8, que é substituir a notação binária pela decimal dentro dos *kets*. Assim, o estado $|011\rangle$ se torna $|3\rangle$ e o estado $|111\rangle$ se torna $|7\rangle$. Para evitar confusões, caso não esteja claro pelo contexto do uso, um subscrito pode ser utilizado para informar a quantos *qubits* correspondem essa notação decimal, ou seja, $|7\rangle_3$ nos indica o estado $|111\rangle$ e $|7\rangle_5$ o estado $|00111\rangle$.

Cada amplitude de probabilidade complexa pode também ser escrita como $c_j = r_j e^{i\theta_j}$, onde r_j é uma amplitude real, com $r_j > 0$, e θ_j é uma fase com $0 \leq \theta_j < 2\pi$. Assim, podemos também representar o estado de um sistema de múltiplos *qubits* como:

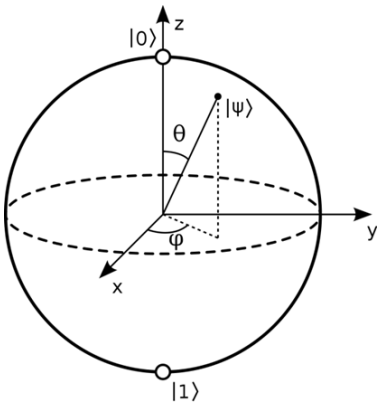
$$|\psi\rangle = \sum_{j=0}^{2^n-1} r_j e^{i\theta_j} |j\rangle \quad , \quad \sum_{j=0}^{2^n-1} r_j^2 = 1 \quad , \quad r_j, \theta_j \in \mathbb{R} \tag{9}$$

A partir dessas descrições e através de algumas manipulações algébricas, podemos representar o estado de um qubit por meio de coordenadas polares usando números reais e visualizá-lo geometricamente na chamada esfera de Bloch [Zhang et al. 2011]:

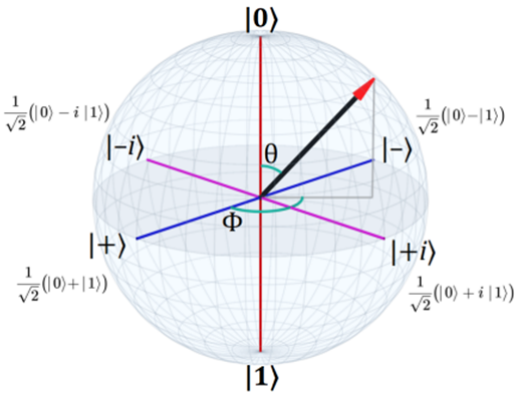
$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle \quad , \quad \theta, \phi \in \mathbb{R} \tag{10}$$

A Figura 3.7(a) mostra a esfera de Bloch, uma representação geométrica útil para o estado de um qubit, com $0 \leq \theta \leq \pi$ e $0 \leq \phi < 2\pi$, que nos permite mapear um vetor de $\mathbb{C}^2$ para $\mathbb{R}^3$. As operações codificadas pelas portas quânticas podem ser vistas como rotações em relação aos eixos dessa esfera [Kosmann-Schwarzbach and Singer 2010]. Nos seus polos temos os estados $|0\rangle$ e $|1\rangle$ da base computacional (por se localizarem sobre o eixo z, essa base muitas vezes também é chamada de base Z), sendo uma escolha arbitrária, porém de uso comum. Os dois ângulos θ e ϕ determinam a localização do vetor de estado. Para estados apresentados nos próximos exemplos, eles vão se localizar na superfície da esfera, porém é possível construir estados que se localizem em qualquer região interna dela. O ângulo ϕ é chamado de fase relativa e contém a diferença de fase entre os números complexos β e α. O fator $e^{i\phi}$ pode ser relacionado com senos e cossenos através da Fórmula de Euler, sendo i a unidade imaginária:

$$e^{i\phi} = cos(\phi) + isen(\phi) \tag{11}$$



(a) Esfera de Bloch. Fonte: [Smite-Meister 2023]



(b) Estados notáveis na Esfera de Bloch. Fonte: [Smythe 2021]

Figura 3.7. Representação de estados na Esfera de Bloch

Em relação aos estados da base computacional, a não ser pelos polos, qualquer outro ponto da superfície da esfera irá representar uma superposição deles. No equador

teremos as chamadas superposições uniformes, nas quais as probabilidades associadas a cada um desses estados será a mesma. Alguns estados notáveis que representam superposições uniformes de estados da base computacional e recebem nomes especiais são mostrados na Figura 3.7(b) e na equação abaixo:

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix} \qquad |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}$$
$$|+i\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle) = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\i\end{pmatrix} \qquad |-i\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle) = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-i\end{pmatrix} \tag{12}$$

A partir da expressão para o estado quântico utilizando coordenadas polares, conseguimos representar tais estados na esfera com a fase relativa determinando a posição do vetor de estado em relação ao eixo x (além do uso de seu nome).

Para modificar os estados dos *qubits* ao longo do tempo precisamos de operadores. Aqueles de interesse para a computação quântica são os chamados unitários, que preservam a norma unitária dos vetores de estado, de acordo com a Regra de Born. Por serem aplicados aos estados dos *qubits*, essas matrizes serão quadradas e com dimensão $2^n \times 2^n$ com n representando o número de *qubits*:

$$U = \begin{pmatrix} u_{11} & \cdots & u_{12^n} \\ \vdots & \ddots & \vdots \\ u_{2^n 1} & \cdots & u_{2^n 2^n} \end{pmatrix} \tag{13}$$

Para aplicar o operador a um determinado estado quântico, basta realizar a multiplicação padrão de matrizes, obtendo assim um novo estado:

$$|\psi'\rangle = U|\psi\rangle \tag{14}$$

Operadores unitários apresentam propriedades importantes. Por exemplo, sua matriz inversa, quando existe, é igual a sua matriz transposta conjugada ($U^{-1} = U^\dagger$). Ou seja, para inverter o efeito de um operador, ao invés de precisar calcular matrizes inversas, algo que se torna mais custoso conforme as dimensões da matriz aumentam, basta aplicar o operador transposto conjugado. A relação abaixo mostra essa ideia, onde um operador multiplicado pelo seu inverso resulta na matriz identidade.

$$UU^{-1} = U^{-1}U = I \quad \rightarrow \quad UU^\dagger = U^\dagger U = I \tag{15}$$

Dentro do conjunto dos operadores unitários temos aqueles que também são hermitianos, ou seja, que são iguais a sua conjugada transposta. Nesse caso, para realizar a operação inversa de um operador hermitiano, basta aplicar o operador novamente. Acumulando as duas propriedades, temos $U = U^\dagger = U^{-1}$. Os operadores serão representados a partir daqui como portas lógicas quânticas (que são a base para computadores

quânticos de propósito geral; para outras abordagens de propósito específico, consultar [Kasirajan 2021]) e a implicação de ter um determinado operador unitário e hermitiano é que para reverter o efeito de uma determinada porta desse tipo basta aplicá-la novamente, enquanto para as demais portas quânticas que não são hermitianas, precisamos aplicar portas quânticas diferentes que representarão os respectivos operadores transpostos conjugados.

Para representarmos estados e combinarmos operadores envolvendo múltiplos *qubits*, precisamos de estruturas matemáticas chamadas tensores, que são generalizações de estruturas como escalares, vetores e matrizes. Um escalar (real ou complexo) é entendido como um tensor de ordem 0, um vetor como um tensor de ordem 1 e uma matriz como um tensor de ordem 2. Para representarmos o produto tensorial entre dois tensores A e B, utilizaremos a notação $A \otimes B$. De forma geral, nos restringindo a vetores e matrizes, para dois deles de dimensões arbitrárias ($n \times m$ e $r \times s$) seu produto tensorial será dado pela seguinte expressão:

$$A \otimes B = \begin{pmatrix} A_{11}\begin{pmatrix} B_{11} & \cdots & B_{1s} \\ \vdots & \ddots & \vdots \\ B_{r1} & \cdots & B_{rs} \end{pmatrix} & \cdots & A_{1m}\begin{pmatrix} B_{11} & \cdots & B_{1s} \\ \vdots & \ddots & \vdots \\ B_{r1} & \cdots & B_{rs} \end{pmatrix} \\ \vdots & \ddots & \vdots \\ A_{n1}\begin{pmatrix} B_{11} & \cdots & B_{1s} \\ \vdots & \ddots & \vdots \\ B_{r1} & \cdots & B_{rs} \end{pmatrix} & \cdots & A_{nm}\begin{pmatrix} B_{11} & \cdots & B_{1s} \\ \vdots & \ddots & \vdots \\ B_{r1} & \cdots & B_{rs} \end{pmatrix} \end{pmatrix} \tag{16}$$

Anteriormente, definimos a base computacional para um qubit como sendo composta pelos estados $|0\rangle$ e $|1\rangle$. Podemos obter a base computacional para sistemas com 2 ou mais *qubits* utilizando o produto tensorial. Para o caso de 2 *qubits*, teremos $2^2 = 4$ estados possíveis de serem representados e podemos obtê-los através da combinação dos estados da base computacional de um qubit conforme expressões abaixo:

$$|00\rangle = |0\rangle \otimes |0\rangle = \begin{pmatrix} 1\begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ 0\begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad |01\rangle = |0\rangle \otimes |1\rangle = \begin{pmatrix} 1\begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ 0\begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$|10\rangle = |1\rangle \otimes |0\rangle = \begin{pmatrix} 0\begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ 1\begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \qquad |11\rangle = |1\rangle \otimes |1\rangle = \begin{pmatrix} 0\begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ 1\begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{17}$$

A representação de circuitos quânticos segue o formato representado na Figura 3.8. As linhas simples representam *qubits*, nomeados q_i, enquanto as linhas duplas re-

presentam *bits*, no caso mostrados de forma coletiva como um registrador de três *bits*, nomeado *c*, que guardam os resultados das medições que são indicadas pelo símbolo de um medidor. As demais caixas com símbolos representam portas lógicas quânticas, responsáveis por modificar os estados dos *qubits* e implementar as operações desejadas (as diferentes cores não são relevantes para nossa análise). As portas mais simples geralmente atuam sobre um ou dois *qubits*, mas podemos ter outras portas mais complexas construídas a partir das mais simples e que operam sobre um maior número deles. A seguir, veremos alguns exemplos, frisando apenas que há diversas outras portas bastante conhecidas, mas que não farão parte da exposição.

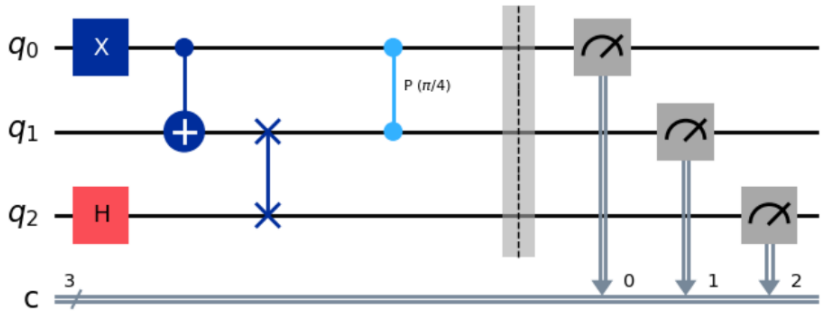


Figura 3.8. Exemplo de Circuito Quântico

Uma das portas quânticas mais importantes é a Hadamard, representada pela letra H. Ela tem por efeito criar superposições uniformes se for aplicada a estados bem definidos como os da base computacional e modificar superposições se for aplicada a estados quaisquer. Sua matriz correspondente é apresentada na Equação 18 e os resultados de sua aplicação aos estados $|0\rangle$ e $|1\rangle$ nas Equações 19 e 20, respectivamente:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{18}$$

$$H|0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle \tag{19}$$

$$H|1\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle \tag{20}$$

Equivalentemente, podemos visualizar essa mudança através da representação do estado do qubit na esfera de Bloch conforme Figura 3.9.

As próximas três portas que veremos são chamadas de portas de Pauli e são representadas pelas letras X, Y e Z, aplicando aos estados quânticos rotações de π radianos em torno dos eixos x, y e z da esfera de Bloch, respectivamente. Em relação à base computacional, a porta X tem por efeito trocar os estados da base computacional entre si ($|0\rangle \rightarrow |1\rangle$, $|1\rangle \rightarrow |0\rangle$), também chamado de *bit-flip*. A porta Z tem por efeito aplicar uma troca de sinal na fase do estado $|1\rangle$, também chamado de *phase-flip*. Por fim, a porta Y

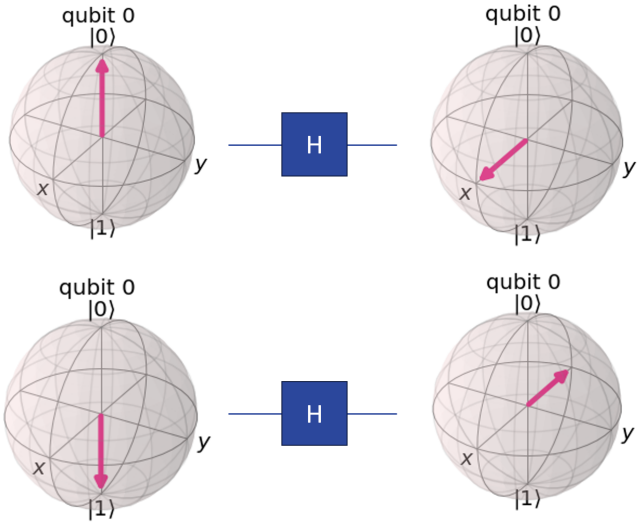


Figura 3.9. Porta H e seu efeito visualizado na esfera de Bloch

equivale a combinar os dois efeitos anteriores, realizando um *bit-flip* e um *phase-flip*. As matrizes correspondentes a essas portas são apresentadas na Equação 21 e suas representações gráficas na Figura 3.10, mostrando os resultados de sua aplicação aos estados $|0\rangle$ e $|1\rangle$.

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad , \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad , \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{21}$$

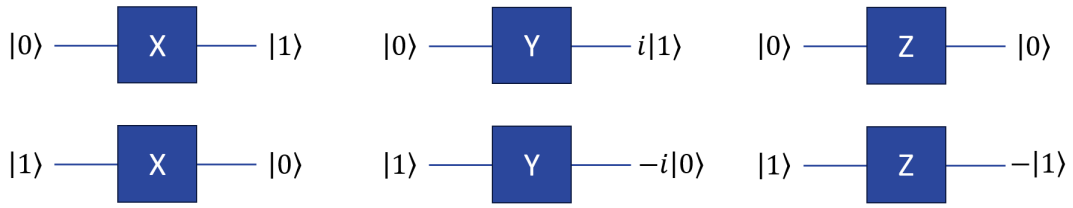


Figura 3.10. Portas de Pauli e seus efeitos sobre a base computacional

O efeito da porta X pode ser visto como o equivalente quântico da porta clássica NOT, sendo também representado como $X|j\rangle = |j \oplus 1\rangle$, com o símbolo $\oplus$ denotando a operação XOR ou adição módulo 2. Disso decorre que a porta X também é representada em muitos locais com esse símbolo. Ao ser aplicada a uma superposição, por linearidade, basta aplicá-la a cada um dos estados constituintes (mesma ideia para as demais portas). Nesse caso, a operação então se torna equivalente a trocarmos as amplitudes de probabilidade:

$$X(\alpha|0\rangle + \beta|1\rangle) = \alpha X|0\rangle + \beta X|1\rangle = \alpha|1\rangle + \beta|0\rangle = \beta|0\rangle + \alpha|1\rangle \tag{22}$$

As próximas três portas quânticas (SWAP, CX e CP) são bastante presentes em circuitos quânticos e operam sobre dois *qubits*. Suas matrizes correspondentes são:

$$SWAP = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, CX_{q_0,q_1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, CP(\phi) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\phi} \end{pmatrix} \quad (23)$$

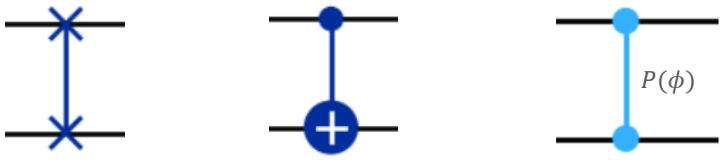
A porta SWAP tem por efeito trocar os estados dos dois *qubits* envolvidos na operação. Sua representação gráfica é apresentada na Figura 3.11(a) e os resultados de sua aplicação aos estados da base computacional de 2 *qubits* são mostrados abaixo:

$$SWAP|00\rangle = |00\rangle \;\; , \;\; SWAP|01\rangle = |10\rangle \;\; , \;\; SWAP|10\rangle = |01\rangle \;\; , \;\; SWAP|11\rangle = |11\rangle \quad (24)$$

Na Figura 3.11(b) temos a porta CX, que possui um qubit de controle (círculo menor) e um qubit alvo (círculo maior, representando a porta X). Quando o controle estiver no estado $|1\rangle$, a porta é ativada e o alvo receberá a aplicação da porta X, caso contrário nada acontece, ou seja, é a aplicação condicional da porta X ao alvo. O seu efeito nos estados da base computacional de 2 *qubits* são mostrados na Equação 25. O subscrito indica qual qubit controla (q_0) e qual é alvo (q_1). Dependendo da escolha, a matriz será diferente, porém com mesmo funcionamento geral.

$$CX_{q_0,q_1}|00\rangle = |00\rangle \;\; , \;\; CX_{q_0,q_1}|01\rangle = |11\rangle \;\; , \;\; CX_{q_0,q_1}|10\rangle = |10\rangle \;\; , \;\; CX_{q_0,q_1}|11\rangle = |01\rangle \quad (25)$$

A combinação de portas CX, H e X nos permite construir estados emaranhados, especialmente aqueles chamados de estados de Bell [Watrous 2025], essenciais em diversas tarefas na computação quântica e na comunicação quântica.



(a) Porta SWAP (b) Porta CX (c) Porta CP

Figura 3.11. Representação gráfica das portas SWAP, CX e CP

Por fim, a porta CP permite rotacionar o estado de um qubit de forma parametrizada em torno do eixo z, aplicando uma fase ao estado do qubit alvo. Quando o controle estiver no estado $|1\rangle$, a porta é ativada e o alvo receberá a fase, caso contrário nada acontece (a Figura 3.11(c) mostra a representação gráfica dessa porta). Porém, por motivos que ficarão claros na subseção 3.4.1.4, seu símbolo não faz uma distinção clara entre controle e alvo. Ao lado dele, aparece a parametrização indicada como $P(\phi)$ visto que a porta CP é a versão controlada da porta P (de único qubit cuja aplicação da fase sempre acontece;

não detalhada aqui). Um exemplo de aplicação da porta é mostrado abaixo para o estado $|q_1 q_0\rangle = |+1\rangle = |+\rangle \otimes |1\rangle$:

$$CP(\pi/4)|+1\rangle = CP(\pi/4)(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |1\rangle) = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\pi/4}|1\rangle) \otimes |1\rangle \tag{26}$$

Agora que aprendemos algumas portas quânticas fundamentais, vamos exercitar a aplicação desses conceitos para identificar qual o estado final do circuito da Figura 3.8 antes da medição. Por convenção, temos o estado inicial $|q_2 q_1 q_0\rangle = |000\rangle$ e por comodidade vamos usar índices em todas as portas para nos dizer sobre quais *qubits* elas estão aplicadas:

$$\begin{aligned} |000\rangle \xrightarrow{X_{q_0}} |001\rangle \xrightarrow{H_{q_2}} \frac{1}{\sqrt{2}}(|001\rangle + |101\rangle) \xrightarrow{CX_{q_0,q_1}} \frac{1}{\sqrt{2}}(|011\rangle + |111\rangle) \\ \xrightarrow{SWAP_{q_1,q_2}} \frac{1}{\sqrt{2}}(|101\rangle + |111\rangle) \xrightarrow{CP(\pi/4)_{q_0,q_1}} \frac{1}{\sqrt{2}}(|101\rangle + e^{i\pi/4}|111\rangle) \end{aligned} \tag{27}$$

Se realizarmos a medição desse estado, na ausência de erros quaisquer no processo, obteremos o estado $|101\rangle$ com probabilidade $1/2$ (basta usar a Regra de Born) e podemos associá-lo à cadeia de *bits* 101, guardando esse valor no registrador clássico. Equivalentemente, poderíamos obter o estado $|111\rangle$ também com probabilidade $1/2$ e associá-lo à cadeia de *bits* 111 (o fator $e^{i\pi/4}$ não influencia essa probabilidade visto que $|e^{i\pi/4}| = 1$). Podemos então reconfigurar nosso circuito e executá-lo novamente para que tenhamos amostras suficientes para reconstruir a distribuição de probabilidades associada ao estado do sistema. Perceba que mesmo exemplos simples podem ser desafiadores quando se fala de computação quântica, pois a lógica subjacente a esse modelo de computação é diferente da que estamos acostumados na computação clássica. Mesmo o entendimento de circuitos básicos e dos algoritmos existentes constitui tarefa complexa e pensar em novas soluções, melhorando ou até mesmo criando novos algoritmos, é um desafio não trivial.

3.4. Impactos em Criptografia e Soluções

Esta seção busca expor como algoritmos quânticos afetam a segurança dos sistemas criptográficos atuais (RSA, ECC, AES), destacando as justificativas para os diferentes níveis de impactos em mecanismos de criptografia simétrica e assimétrica. Esta seção também apresenta a criptografia pós-quântica (PQC) [Alagic et al. 2022] e quântica (QKD) como soluções em desenvolvimento [Mosca 2018], e um estudo de caso de impactos da computação quântica ao Bitcoin. Serão explorados exemplos práticos de uso de uma biblioteca de Criptografia Pós-Quântica (exemplos disponíveis em `https://github.com/vthayashi/quantum-crypto`).

3.4.1. Impactos dos Algoritmos de Grover e Shor

Os dois algoritmos quânticos mais conhecidos são os de Shor [Shor 1994] e de Grover [Grover 1996]. O algoritmo de Shor foi desenvolvido pelo matemático Peter Shor (na realidade ele propôs um conjunto de algoritmos, porém é de uso comum se referir a esse conjunto apenas como algoritmo de Shor) em 1994 e pode ser utilizado para resolver eficientemente problemas como a fatoração de números inteiros e o cálculo de logaritmos discretos, apresentando uma aceleração exponencial se comparado ao melhor algoritmo clássico conhecido para o mesmo problema. Lembrando que, para falarmos sobre aceleração em termos computacionais, comparamos a complexidade computacional para o mesmo problema entre ao menos dois algoritmos e analisamos suas diferenças.

O algoritmo de Grover foi proposto pelo cientista da computação Lov Grover em 1996 e pode ser utilizado para realizar buscas num conjunto de dados não estruturado, ou seja, um conjunto que não está ordenado ou possui qualquer tipo de atalho para encontrar o item desejado. Nesse caso, o algoritmo apresenta uma aceleração quadrática se comparado ao melhor algoritmo clássico para o mesmo problema. Ao combinar um computador quântico de larga escala e tolerante a falhas, também chamado de Computador Quântico Criptograficamente Relevante (CRQC - *Cryptographically Relevant Quantum Computer*) [NSA 2021] neste contexto de criptografia, com os algoritmos de Shor e de Grover, pode-se causar impactos substanciais em boa parte da criptografia utilizada atualmente, conforme descrito na próxima subseção.

3.4.1.1. Impactos do Algoritmo de Grover na Criptografia

Suponha que temos uma lista com $N = 2^n$ itens com N sendo um número grande e n o número de *bits* necessário para representar todos os itens. Se dentre eles existe um que possui uma propriedade única e que queremos localizar (por exemplo, uma chave criptográfica), vamos representar esse item pela letra m. Os N itens podem ser representados equivalentemente por n *qubits*, com cada item correspondendo a um estado da base computacional que podemos representar. Para encontrar o item m usando computação clássica, precisaríamos checar na média $N/2$ itens e, no pior caso, todas os N itens da lista ($O(N)$ operações). Por outro lado, em um computador quântico, podemos encontrar m com cerca de ($O(\sqrt{N})$) operações utilizando o algoritmo de Grover [Grover 1996]. O circuito quântico para executar o algoritmo tem a estrutura apresentada na Figura 3.12. A ideia principal é iniciar o circuito com todos os estados possíveis em superposição uniforme dado um certo número n de *qubits* e aplicar transformações dependentes do problema para que as amplitudes correspondentes aos estados desejados sejam amplificadas às custas das amplitudes dos estados não desejados. Vamos agora explicar em linhas gerais o que cada parte realiza.

Começamos com um registrador de n *qubits* inicializados no estado $|0\rangle$ e podendo ser representados coletivamente como $|0\rangle^{\otimes n}$. Essa notação indica o produto tensorial de n *qubits* configurados no estado quântico $|0\rangle$. Na primeira parte das operações, aplicamos portas Hadamard a todos esses *qubits*, resultando no estado $|s\rangle = H^{\otimes n}|0\rangle^{\otimes n} = |+\rangle^{\otimes n}$ que é uma superposição uniforme de todos os estados da base computacional de n *qubits*. Em seguida, repetiremos t vezes a execução da dupla de transformações chamadas de Orá-

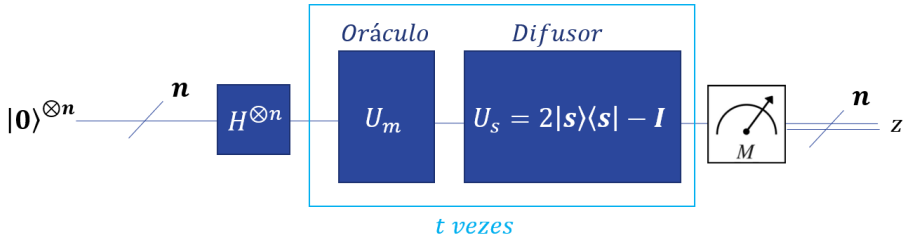


Figura 3.12. Circuito geral para executar o Algoritmo de Grover

culo U_m e Difusor U_s. A estrutura do bloco oráculo dependerá do problema em questão e vai variar de acordo com o que queremos encontrar, enquanto a estrutura do difusor pode ser fixa como $2|s\rangle\langle s| - I$. O papel do oráculo é identificar quais estados são candidatos a soluções e os marcar utilizando uma alteração de fase ($U_m|x\rangle = -|x\rangle$ se $x = m$), mantendo os demais inalterados ($U_m|x\rangle = |x\rangle$ se $x \neq m$). O papel do difusor é amplificar as amplitudes de probabilidade dos estados candidatos em detrimento dos demais de maneira que as respectivas probabilidades desses estados serem medidos aumentem. Essa sequência de operações é repetida um determinado número de vezes até que o estado resultante represente o item (ou itens) desejado com alta probabilidade. No caso de mais de 1 solução, teremos um estado que será uma superposição das soluções possíveis, porém sempre medindo apenas uma delas a cada execução.

O operador resultante a cada iteração da sequência oráculo/difusor é dado por $G = U_s U_m$, sendo G chamado comumente de operador (ou iterador) de Grover. O número t determina uma estimativa da quantidade de repetições desse bloco para obter o resultado desejado com alta probabilidade. Ao fazer essa aplicação sucessiva, a expressão final para o estado quântico do circuito será $|\psi_t\rangle = (U_s U_m)^t |s\rangle$. Para os casos de solução única m ou múltiplas soluções $M = \{m_1, m_2, ...\}$ [Nielsen and Chuang 2010], teremos:

$$t_{única} \approx \frac{\pi}{4}\sqrt{N} \rightarrow O(\sqrt{N}) \quad , \quad t_{múltiplas} \approx \frac{\pi}{4}\sqrt{N/M} \rightarrow O(\sqrt{N/M}) \tag{28}$$

Por exemplo, suponha que estamos utilizando o algoritmo AES e temos uma determinada quantidade de pares mensagem-encriptado (m_i, c_i) de forma a minimizar a probabilidade de encontrarmos chaves falsas durante a busca [Menezes et al. 2018]. Construíndo o bloco Oráculo para implementar o AES, com o algoritmo de Grover conseguimos marcar a solução desejada (a chave criptográfica k) de forma que $U_m|k\rangle = -|k\rangle$ se $Enc(k, m_i) = c_i$. Após $O(\sqrt{N})$ iterações, sendo N o tamanho do espaço de chaves, teremos o estado correspondente à chave k com alta probabilidade na saída do circuito.

Na criptografia simétrica, o nível de segurança estabelece diretamente o esforço necessário (usualmente medido em *bits*) em termos de passos computacionais para descobrirmos a chave secreta utilizada em alguma cifra simétrica ou para calcular pré-imagens de resumos criptográficos para uma determinada função de *hash*. Ou seja, dizer que um algoritmo tem 128 *bits* de nível de segurança significa que a melhor abordagem para quebrá-lo demanda um esforço da ordem de 2^{128} passos computacionais. Na ausência de outros tipos de ataques mais eficientes, podemos associar esse nível de segurança ao

esforço computacional de realizar uma busca exaustiva. No contexto criptográfico, esse tipo de busca nada mais é que uma busca num conjunto de dados não estruturado, tipo de problema onde o algoritmo de Grover pode ser aplicado. No pior caso, analisando apenas a diferença na complexidade dos melhores algoritmos clássico ($O(N) = O(2^n)$) e quântico ($O(\sqrt{N}) = O(2^{n/2})$) para esse mesmo problema, com N representando o número de itens no conjunto e n sendo o número de *bits* necessários para representar todos os itens, obtemos uma aceleração quadrática de eficiência. Neste caso, os algoritmos criptográficos podem ser impactados, mas não de forma crítica, podendo apenas demandar ajustes no tamanho da chave ou da saída da função de *hash* para retornar ao mesmo nível de segurança. Entretanto, vamos analisar em mais detalhes esse potencial impacto em funções de *hash* e nos demais tipos de algoritmos criptográficos simétricos.

De forma geral, para funções de *hash* criptográficas consideradas seguras podemos identificar dois tipos de ataque: o cálculo de pré-imagens e o cálculo de colisões. No primeiro caso, a melhor abordagem acaba sendo a busca exaustiva com complexidade temporal $O(2^n)$ sendo n o tamanho do resumo criptográfico. Um ataque de busca exaustiva, também chamado de força bruta, pode ter sua solução acelerada pelo algoritmo de Grover para $O(2^{n/2})$. No segundo caso, já existe um algoritmo clássico conhecido, explorando um ataque de aniversário, para o cálculo de colisões com complexidade temporal $O(2^{n/2})$ [Katz and Lindell 2014], então o algoritmo de Grover acaba não sendo melhor do que isso. Dessa forma, uma função de *hash* cuja saída possui 256 *bits* (por exemplo SHA-256, BLAKE-256, SHA3-256) acaba tendo sua segurança efetiva para ataques de pré-imagem reduzida para 128 *bits*, o que ainda é considerado seguro em termos dos padrões atuais de nível de segurança, que recomendam um mínimo de 112 *bits* [Barker et al. 2020]. Ou seja, em termos práticos, o algoritmo de Grover não fornece uma ameaça substancial nesse caso. O mesmo tipo de análise pode ser feito para funções de *hash* com saídas de 224 *bits*, o que nos leva a níveis de segurança efetivos de 112 *bits*, no limiar da margem recomendada. Assim, deve-se analisar cada função de *hash* utilizada e verificar se o seu nível de segurança para ataques de pré-imagem, ao ser verificado frente ao algoritmo de Grover, cai abaixo do padrão recomendado (independente de qual for). Além disso, há estudos combinando ideias do ataque de aniversário com o algoritmo de Grover para tentar melhorar o esforço computacional para o cálculo de colisões [Brassard et al. 1998, Hosoyamada and Sasaki 2020]. Dadas todas essas considerações, se o nível de segurança resultante da análise frente ao algoritmo de Grover cair abaixo do adequado, se for desejado um nível de segurança com maior expectativa de vida ou se objetiva-se manter exatamente o mesmo nível de segurança atual para ataques de pré-imagem, bastaria dobrar o tamanho do resumo criptográfico.

No caso de algoritmos criptográficos simétricos que utilizam chaves, como as cifras de bloco, cifras de fluxo e códigos de autenticação de mensagens, se não conhecermos abordagens analíticas mais eficientes, nossa melhor estratégia para recuperar a chave simétrica é a busca exaustiva no espaço de chaves, equivalente a um problema de busca num conjunto de dados não estruturado. Dessa forma, o algoritmo de Grover pode também ser utilizado nesse caso para acelerar essa tarefa para $O(2^{n/2})$. Assim, algoritmos como o AES utilizando chaves de 128 e 192 *bits* teriam seu nível de segurança efetivo reduzido para 64 e 96 *bits*, respectivamente, abaixo dos 112 *bits* recomendado atualmente. Por outro lado, algoritmos como o AES-256 e ChaCha20, que utilizam chaves de 256

bits, teriam sua segurança teoricamente reduzida para 128 *bits*, ainda considerada segura. Assim, para responder a esse potencial impacto, o ideal é que o tamanho das chaves criptográficas simétricas utilizadas no caso dos algoritmos impactados seja dobrada caso seu nível de segurança frente ao algoritmo de Grover caia abaixo do recomendado.

Importante notar que há trabalhos que analisam as dificuldades que a implementação do algoritmo de Grover traria e argumentam que seria inviável concretizar na prática essa aceleração quadrática [Sarah and Peter 2024]. Em todo caso, para teoricamente impactar a segurança de um algoritmo simétrico, a aceleração não necessariamente precisa ser quadrática, basta que o nível de segurança efetivo seja reduzido abaixo do recomendado. Por fim, a análise do impacto do algoritmo de Grover é feita em termos de análise de algoritmos sob o regime assintótico, onde não estamos preocupados com questões de implementação, nos dando um limite superior para o desempenho que pode ser obtido e, nesse caso, a recomendação de dobrar o tamanho das chaves simétricas deve ser adotada.

3.4.1.2. Impactos do Algoritmo de Shor na Criptografia

O algoritmo de Shor pode ser utilizado para fatorar números inteiros (dado um número inteiro N, encontrar seus fatores primos p e q tal que $N = pq$) e calcular logaritmos discretos (dados um número primo p e outros números inteiros g e y, encontrar x em $y = g^x \ mod \ p$), ambos de forma eficiente e com acelerações exponenciais se comparados aos melhores algoritmos clássicos para os respectivos problemas [Shor 1997]. Ou seja, mostrou que os computadores quânticos poderiam ser mais eficientes do que os computadores clássicos para problemas de interesse prático, até então um desafio para os pesquisadores da área.

A ideia principal é transformar o problema da fatoração ou do cálculo de logaritmos discretos em um problema de encontrar o período de uma função em tempo polinomial utilizando um computador quântico. Dessa forma, um algoritmo que resolva esse problema de forma eficiente pode ser utilizado para resolver esses outros problemas de forma eficiente também [Nielsen and Chuang 2010]. Por exemplo, o algoritmo de Shor para fatorar números inteiros apresenta complexidade temporal $O(poly(n))$, onde $poly(n)$ denota uma função polinomial (veremos com mais detalhes algumas possibilidades para essa função na Subseção 3.4.1.4). O melhor algoritmo clássico para resolver o mesmo problema quando N é grande é o *General Number Field Sieve* (GNFS) [Boudot et al. 2020a], que possui complexidade subexponencial dada por $O(e^{(c)(n)^{1/3}(log\ n)^{2/3}})$, onde c é uma constante, em função do tamanho n do número N. Podemos também ignorar a constante e representar de forma mais geral essa complexidade subexponencial como $O(e^{(n)^{\alpha}(log\ n)^{1-\alpha}})$, sendo $0 < \alpha < 1$.

Na criptografia assimétrica, para conseguirmos recuperar a chave privada a partir da chave pública, precisamos resolver determinado problema matemático considerado intratável para computadores clássicos. Três desses principais problemas cuja dificuldade clássica é utilizada como pilar de segurança são o problema da fatoração de números inteiros (IFP - *Integer Factorization Problem*), o problema do logaritmo discreto (DLP - *Discrete Logarithm Problem*) e o problema do logaritmo discreto elíptico (ECDLP - *Elliptic Curve Discrete Logarithm Problem*) [Aumasson 2017]. Eles são considerados intratáveis, pois os melhores algoritmos clássicos apresentam complexidade temporal su-

bexponencial ou exponencial. Entretanto, todos esses problemas podem ser endereçados de forma eficiente pelo algoritmo de Shor e resolvidos em tempo polinomial. Com isso, nossa premissa de que são problemas intratáveis acaba sendo invalidada e não podemos mais utilizar esses algoritmos criptográficos de forma segura, devendo ser substituídos por novos algoritmos cujo modelo de atacante tem como uma de suas premissas a posse de um computador quântico criptograficamente relevante, coletivamente chamados de criptografia pós-quântica (PQC - *Post-Quantum Cryptography*).

A segurança atual de algoritmos como o RSA reside no fato de que p e q, os dois números primos utilizados por cada usuário para criação do seu par de chaves, estão escondidos no valor N, que é público, e fatorar esse número (que na prática é grande, com centenas/milhares de dígitos decimais) é considerado um problema difícil classicamente. O último número fatorado utilizando computação clássica foi o RSA-250 com 250 dígitos decimais, equivalente a um tamanho de 829 *bits*, em 2020 [Boudot et al. 2020b]. A tarefa foi completada em alguns meses e o algoritmo clássico utilizado foi o GNFS. Com a ajuda do algoritmo de Shor, podemos fatorar N e obter p e q de forma eficiente. Com isso, conseguiríamos realizar os seguintes cálculos (e pode ser facilmente obtido e d pode ser calculado de forma eficiente com computação clássica):

$$N = pq \quad , \quad \phi(N) = (p-1)(q-1) \quad , \quad d = e^{-1} mod\ \phi \tag{29}$$

A partir disso, poderíamos forjar assinaturas digitais e decifrar mensagens (por exemplo uma chave que tenha sido encapsulada). Um ataque conhecido que tem por objetivo preparar as informações necessárias para se valer dessa segunda possibilidade é o chamado *Store Now Decrypt Later (SNDL)* ou *Harvest Now Decrypt Later (HNDL)* [Joseph et al. 2022]. Dessa forma, com o uso do algoritmo de Shor, o IFP se torna tratável em tempo polinomial, o que implica que nossa premissa de segurança se torna inválida e faz com que esse tipo de algoritmo criptográfico, que faz uso desse problema, deva ser abandonado.

Para o caso de algoritmos como o DH e o DSA, que utilizam instâncias do problema do logaritmo discreto, suas chaves públicas são geradas como $y = g^x\ mod\ p$ com chave privada x que é o logaritmo discreto. Dados g, p e y, desejamos encontrar x, e este problema também é considerado intratável para computadores clássicos. Porém, novamente Shor propôs um algoritmo quântico que pode resolver esse problema com eficiência polinomial $O(poly(n))$ enquanto classicamente temos complexidade subexponencial dada por $O(e^{(n)^{\alpha}(log\ n)^{1-\alpha}})$, ignorando constantes e com $0 < \alpha < 1$ [Boudot et al. 2020a]. Assim, para o caso do DH, poderíamos recuperar a chave privada de um dos participantes da comunicação e, ao combinar essa informação com a chave pública do outro participante, que pode ser recuperada facilmente, seria possível calcular o segredo compartilhado. Para o caso de assinaturas digitais, como o DSA, resolver esse problema nos permite recuperar a chave privada de assinatura.

Por fim, o algoritmo de Shor para o problema do logaritmo discreto pode ser adaptado para o caso elíptico (ECDLP). Nesse problema, dado dois pontos P e Q na curva elíptica utilizada que satisfazem $Q = xP$ sendo x um escalar e representando a chave privada, o algoritmo pode calculá-lo com complexidade temporal $O(poly(n))$, enquanto as

melhores alternativas clássicas possuem complexidade temporal exponencial $O(e^{poly(n)})$ [Roetteler et al. 2017]. Dessa forma, algoritmos como ECDH e ECDSA também devem ser abandonados dada a existência do algoritmo de Shor. Para trabalhos que discutem em profundidade a aplicação e melhorias do algoritmo de Shor para o DLP e para o ECDLP, consultar [Roetteler et al. 2017, Häner et al. 2020, Aono et al. 2022, Hhan et al. 2023].

A partir dessa discussão, as implicações são que os melhores algoritmos clássicos para resolver instâncias dos três problemas comentados para os tamanhos de parâmetros utilizados atualmente acabariam resultando em escalas de tempo muito grandes e impraticáveis, enquanto o algoritmo de Shor estaria em escalas menores e factíveis. Naturalmente, o tempo exato de execução só seria possível determinar levando-se em conta a implementação específica adotada em computadores quânticos reais de larga escala, ainda não existentes. Em suma, frente ao algoritmo de Shor, todos os criptossistemas que baseiam sua segurança na dificuldade desses três problemas devem ser abandonados e substituídos pelos novos algoritmos pós-quânticos.

3.4.1.3. Resumo dos Impactos em Criptografia

A Tabela 3.1 sumariza o que foi discutido nas subseções anteriores. Lembrando que todo e qualquer sistema atual será impactado em maior ou menor grau, independente de fabricantes e de aplicações, e que os esforços de migração, sejam para impactos mais modestos como no caso simétrico ou para impactos críticos como no caso assimétrico, constituem tarefa desafiadora. Além disso, métricas como tempo de processamento, consumo de energia e uso de memória também são afetadas, criando desafios diversos para o processo de migração como um todo.

Tabela 3.1. Resumo dos Impactos em Criptografia.

Primitiva Criptográfica	Exemplos de Algoritmos	Problema Subjacente	Complexidade Temporal Clássica	Complexidade Temporal Quântica	Contramedida
Hash	SHA-2, SHA-3	Cálculo de pré-imagem	$O(2^n)$	$O(2^{n/2})$	Ajustar o tamanho do *hash* se necessário
Simétrica	AES, ChaCha20	Busca de chave	$O(2^n)$	$O(2^{n/2})$	Ajustar o tamanho da chave se necessário
Assimétrica	RSA	IFP	$O(e^{(n)^{\alpha}(log\ n)^{1-\alpha}})$ $0<\alpha<1$	$O(poly(n))$	Substituir algoritmos
	DH, DSA, ElGamal	DLP			
	ECDH, ECDSA, EdDSA	ECDLP	$O(e^{poly(n)})$		

3.4.1.4. Exemplo de Execução do Algoritmo de Shor

Vamos mostrar a aplicação prática do algoritmo de Shor para o problema da fatoração de números inteiros. Veremos um exemplo completo de como utilizar o algoritmo para fatorar o número $N = 15$. Naturalmente, esse número é pequeno e muitos dos detalhes técnicos serão omitidos visto que seu pleno entendimento demandaria uma exposição mais longa e aprofundada sobre o assunto, o que está além do escopo deste minicurso. O exemplo completo, incluindo o código associado, está disponível em: `https://github.com/vthayashi/quantum-crypto`.

3.4.2. Fundamentos de Criptografia Pós-Quântica

A principal diferença entre a criptografia moderna tradicional e a pós-quântica está em seus modelos adversariais. Enquanto uma assume a capacidade limitada de computadores convencionais, a PQC considera um adversário com acesso a computadores quânticos capazes de executar, de forma eficiente, os algoritmos apresentados na Seção 3.3.

A criptografia pós-quântica, ao contrário da criptografia quântica, não depende e não almeja o uso de sistemas quânticos para sua implementação, pois seus algoritmos são executáveis em computadores clássicos. Em resumo, a criptografia pós-quântica trata de mitigar as vulnerabilidades inseridas pela computação quântica [Bernstein et al. 2009].

O centro das preocupações atuais e do desenvolvimento da área da criptografia pós-quântica são os algortimos que tomam a primitiva do uso de chaves assimétricas para o seu funcionamento. Isso se deve pois, de forma diferente do impacto quadrático do algoritmo de Grover em funções *hash* criptográficas e algoritmos simétricos, o algoritmo de Shor resolve os problemas matemáticos subjacentes (fatoração de inteiros e logaritmo discreto) de algoritmos assimétricos em tempo polinomial, como discutido na Subseção 3.4.1. Isso implica que quando um computador quântico criptograficamente relevante vier a existir, os criptossistemas baseados nesses problemas estarão completamente quebrados [Mosca 2018].

Ao contrário dos casos das mitigações à ameaça quântica das outras primitivas, aumentar o tamanho da chave significa uma ineficiência crescente do uso de recursos computacionais sem ganho de segurança significativo no caso assimétrico, já que o algoritmo de Shor continuará a resolver os problemas fundamentais em tempo polinomial. Para isso, diferentes classes de problemas matemáticos têm sido estudadas para substituir a segurança baseada na fatoração de inteiros e no logaritmo discreto [Beullens et al. 2021].

O processo de criação de algoritmos pós-quânticos começa com a seleção de uma classe de problemas matemáticos que sejam computacionalmente intratáveis em computadores clássicos e para os quais não existam algoritmos quânticos eficientes. Sobre essa base teórica escolhida, pesquisadores desenvolvem e propõem diversos algoritmos concretos dentro de cada família. As principais classes de problemas computacionais e algoritmos considerados para PQC podem ser encontrados na Tabela 3.2.

A dificuldade da criação de certos problemas computacionais em reticulados é a

Tabela 3.2. Principais classes de problemas em criptografia pós-quântica, suas dificuldades matemáticas e exemplos de algoritmos. Adaptado de [Beullens et al. 2021]

Classe	Fonte da Dificuldade	Exemplos de Algoritmos
Baseada em Reticulados	Problemas de reticulados euclidianos, como LWE e Module-LWE	Kyber, Dilithium, Falcon, SABER
Baseada em Códigos	Problema de Decodificação de Erros	Classic McEliece, BIKE, HQC
Baseada em Hashes	Colisões e pré-imagens em funções *hash* seguras (ex. SHA-3)	SPHINCS+
Baseada em Isogenias	Problema da Isogenia Supersingular	SIKE
Baseada em Sistemas Multivariados	Problema MQ (Multivariate Quadratic) sobre $\mathbb{F}_q$	Rainbow, GeMSS

base de segurança para a construção de alguns esquemas PQC relevantes. Um reticulado em $\mathbb{R}^n$ é o conjunto de todas as combinações lineares com coeficientes inteiros de n vetores linearmente independentes em $\mathbb{R}^n$ [Ajtai 1996]. Trata-se de uma malha regular de pontos no espaço n-dimensional (Figura 3.13). Problemas clássicos construídos são de simples entendimento, como o *Shortest Vector Problem* (SVP), que se baseia em encontrar o vetor não nulo mais curto em um reticulado, ou o *Closest Vector Problem* (CVP) que tem como objetivo encontrar o vetor do reticulado mais próximo de um ponto arbitrário definido.

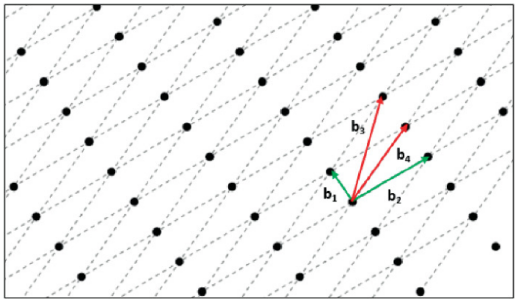


Figura 3.13. Representação de um reticulado (*lattice*) no plano com duas bases distintas. Os vetores $\mathbf{b}_1$ e $\mathbf{b}_2$ (em verde) formam uma base do reticulado, enquanto os vetores $\mathbf{b}_3$ e $\mathbf{b}_4$ (em vermelho) representam uma base alternativa. Todos os pontos pretos no plano representam os pontos do reticulado gerado por combinações lineares inteiras dessas bases. Adaptado de [Shah et al. 2025].

Na classe baseada em reticulados (*lattice-based*), o problema *Learning With Errors* (LWE), juntamente com variações como o *Module-LWE* e *Ring-LWE*, ganham destaque na construção de algoritmos PQC. Embora o problema LWE não seja, em formulação direta, um problema de reticulados como o SVP e o CVP, ele pode ser reduzido para problemas fundamentais desse conjunto. Foi demonstrado formalmente que resolver instâncias aleatórias do problema LWE é tão difícil quanto resolver problemas clássicos de reticulados no pior caso, como o *Gap Shortest Vector Problem* (GapSVP) e o *Shortest Independent Vectors Problem* (SIVP) [Regev 2005]. O LWE estabelece que dado um sistema linear com um ruído, tentar recuperar o vetor original é computacionalmente difícil. Suas variações, como o *Ring-LWE* e o *Module-LWE*, foram desenvolvidas para otimizar desempenho e reduzir o tamanho das chaves na construção de criptossistemas [Lyubashevsky et al. 2010]. Esquemas baseados em reticulados, como o CRYSTALS-KYBER e o CRYSTALS-DILITHIUM, comentados na Subseção 3.4.3 são tidos como eficientes e seguros.

Enquanto em reticulados o ruído está num espaço geométrico, na classe baseada em Códigos (*code-based*) ele é adicionado a palavras codificadas em espaço finito (por exemplo, *bits*). Essa abordagem remonta à uma proposta clássica [McEliece 1978], que utilizou códigos de Goppa para construir um criptossistema assimétrico que competiu por adoção em sua época. A dificuldade do problema principal toma que dado um código linear aleatório e uma palavra corrompida por erros, determinar a mensagem original é um problema considerado intratável por algoritmos conhecidos [Bernstein et al. 2009]. Esquemas baseados em códigos geralmente apresentam tamanhos de chave pública significativamente maiores do que outras abordagens, sendo uma limitação em comparações

para adoção prática.

Na criptografia baseada em *hashes* (*hash-based*) a ideia é usar as dificuldades provenientes dos requisitos de segurança de funções *hash* criptográficas, apresentadas na Subseção 3.2.3, como problema computacional para assegurar criptossistemas assimétricos. O principal problema utilizado é que dado somente o valor de uma função *hash*, encontrar uma pré-imagem primária, secundária ou colisão é computacionalmente inviável. Em vez de confiar em uma estrutura matemática complexa, essa abordagem parte da premissa de que funções como SHA-3 são seguras contra ataques quânticos, portanto constituem um ponto para construção de outros mecanismos [Beullens et al. 2021].

A pesquisa sobre a classe de algoritmos PQC baseados em Isogenia (*isogeny-based*) tornou-se desafiadora após a quebra pública do principal algoritmo (SIKE) da família [Castryck and Decru 2023]. Essa abordagem explora a dificuldade de encontrar isogenias entre curvas elípticas supersingulares. Uma isogenia é, de forma imprecisa e simples, um morfismo entre duas curvas elípticas que preserva a estrutura algébrica dos grupos dessas curvas. Curvas elípticas supersingulares demonstram propriedades específicas em relação ao conjunto genérico, quando definidas sobre corpos finitos. O problema pode ser formulado dado que econtrar uma isogenia de grau pequeno entre duas curvas elípticas supersingulares E e E' é computacionalmente intratável, mesmo para computadores quânticos [Jao and De Feo 2011].

A criptografia multivariada (*multivariate*) têm como fundamento a dificuldade de resolver sistemas de equações polinomiais multivariadas sobre corpos finitos, problema conhecido como problema MQ (*Multivariate Quadratic problem*), NP-difícil em geral. O problema pode ser formulado dado que para uma função polinomial $P : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$, composta por equações quadráticas em variáveis sobre um corpo finito $\mathbb{F}_q$, determinar uma pré-imagem x tal que $P(x) = y$ é difícil [Ding et al. 2006]. Um exemplo destaque dessa classe foi o esquema Rainbow que foi quebrado antes de sua padronização [Beullens et al. 2021].

Como discutido na próxima subseção, as padronizações indicam que a classe de algoritmos baseados em reticulados será adotada como principal mecanismo PQC.

3.4.3. Padronização NIST de Criptografia Pós-Quântica

Dada a relevância do cenário de quebra de algoritmos de criptografia assimétrica baseados nos problemas de logaritmo discreto e de fatoração de números inteiros, além dos impactos em algoritmos simétricos citados anteriormente, esforços de padronização de algoritmos criptográficos pós-quânticos estão em andamento [Alagic et al. 2025].

O estabelecimento de padrões criptográficos é um processo relevante para suportar o correto uso dos algoritmos. O NIST (*National Institute of Standards and Technology*) é uma agência do departamento de Comércio dos Estados Unidos que vêm apoiando a padronização de algoritmos criptográficos (e.g., SHA-3), e que conta com uma reputação relevante na comunidade de segurança da informação.

O processo de padronização do NIST envolve uma análise de segurança e considerações sobre eficiência computacional dos algoritmos candidatos. Em relação à criptografia pós-quântica, há uma chamada pública para mecanismos de encapsulamento de

chaves (KEM - *Key Encapsulation Mechanism*) e assinatura digital resilientes aos ataques potenciais de computadores quânticos. No decorrer das rodadas de avaliação, a resiliência a ataques clássicos e quânticos foi avaliada pela comunidade, em um esforço conjunto para a seleção dos melhores candidatos [Paar et al. 2024].

A iniciativa do NIST em PQC teve início em 2012 com a criação de um grupo de trabalho. O primeiro workshop sobre o tema ocorreu em 2015, o primeiro relatório veio em 2016, e a chamada pública inicial se encerrou em Novembro de 2017 [Chen et al. 2017]. Desde então, outras três novas rodadas ocorreram [Alagic et al. 2025]. A Tabela 3.3 mostra a evolução das submissões e seleção de finalistas no decorrer das rodadas de padronização.

Tabela 3.3. Resumo da evolução dos candidatos no processo de padronização NIST para criptografia pós-quântica (PQC). Em negrito, o número de algoritmos finalistas; entre parênteses, o número de algoritmos com avaliação pendente. Adaptado de [Paar et al. 2024] e [Alagic et al. 2025].

Etapa	Data de Anúncio	# KEM	# Assinatura Digital
Submissões Iniciais	Dez 2017	40	29
Após 1ª Rodada	Jan 2019	17	9
Após 2ª Rodada	Jul 2020	4 + (5)	3 + (3)
Após 3ª Rodada	Set 2022	**4** + (5)	**3** + (3)

Dentre os finalistas, há 5 algoritmos selecionados para padronização: 2 para encapsulamento de chaves e 3 para assinatura digital [Alagic et al. 2025]. CRYSTALS-KYBER é um mecanismo baseado em reticulados que foi selecionado devido ao seu compromisso entre segurança e eficiência quando comparado aos mecanismos de criptografia assimétrica existentes, e se tornou o padrão FIPS 203 adotando o nome *Module-Lattice-Based Key-Encapsulation Mechanism* (ML-KEM). CRYSTALS-DILITHIUM é um mecanismo utilizado para assinatura digital que também é baseado em reticulados, e compõe o padrão FIPS 204 com o nome *Module-Lattice-Based Digital Signature Algorithm* (ML-DSA). SPHINCS+ é um mecanismo utilizado para assinatura digital baseado em *hash*, e que se tornou o padrão FIPS 205 sob o nome *Stateless Hash-Based Digital Signature Algorithm* (SLH-DSA) [Paar et al. 2024, Alagic et al. 2025].

Os algoritmos Falcon (baseado em reticulados) e HQC (baseado em códigos corretores de erros) também foram selecionados, e estão atualmente aguardando a publicação de seus padrões associados. Como a maioria dos algoritmos finalistas na rodada 3 são baseados em reticulados, a rodada 4 obteve 3 candidatos baseados em outros problemas matemáticos (desconsiderando na contagem o candidato SIKE, que foi provado inseguro). Considerando que a segurança dos mecanismos criptográficos é baseada em problemas matemáticos complexos com resolução inviável em computadores clássicos, e que computadores quânticos tem o potencial de resolver novas classes de problemas, abordagens complementares baseadas em outros tipos de problemas matemáticos são desejáveis [Alagic et al. 2025].

Em relação à transição dos algoritmos criptográficos, o NIST não autoriza mais o uso de ECDSA, EdDSA (*Edwards-curve Digital Signature Algorithm*) e RSA do padrão FIPS 186 para assinatura digital após 2035, sendo que RSA e ECDSA são considera-

dos depreciados após 2030 (i.e., podem ser usados, mas há risco potencial associado) [Moody et al. 2024]. Para assinatura digital com níveis de segurança de 128, 192 e 256 *bits*, o NIST passa a recomendar o uso de ML-DSA (i.e., CRYSTALS-DILITHIUM) nas configurações ML-DSA-44, ML-DSA-65 e ML-DSA-87 da FIPS 204, além de padrões baseados em *hash* [Cooper et al. 2020]. Para o estabelecimento seguro de chaves, algoritmos como Diffie-Hellman e RSA não são mais autorizados para uso pelo NIST a partir de 2035, e o padrão recomendado passa a ser o ML-KEM (i.e., CRYSTALS-KYBER) nas configurações ML-KEM-512, ML-KEM-768, e ML-KEM-1024 para níveis de segurança de 128, 192 e 256 *bits*, respectivamente [Moody et al. 2024].

Conforme os novos padrões são adotados pelas organizações com a transição dos mecanismos criptográficos, se atentar a aspectos de implementação passa a ser relevante, uma vez que os algoritmos podem suportar um nível adequado de segurança nas aplicações, mas uma configuração inadequada e erros intencionais/inadvertidos podem resultar em vulnerabilidades. Por exemplo, ataques de canal lateral (*side channel attacks*) podem impactar a segurança de mecanismos PQC de assinatura digital e encapsulamento de chaves [Jedlicka et al. 2022, Ravi et al. 2024]. A análise da temporização, sinais eletromagnéticos, e níveis de consumo de energia podem levar ao comprometimento de mecanismos de criptografia pós-quântica baseados em reticulados, inclusive com o uso de redes neurais para execução dos ataques [Wang et al. 2023, Huang et al. 2024]. Neste cenário, o uso de métodos de avaliação (e.g., FIPS 140-3) é um processo relevante [Saarinen 2022]. Contramedidas como adição de ruído, introdução de atrasos aleatórios, e outras medidas de mascaramento podem contribuir para mitigar estas ameaças, tanto em software quanto em hardware [Zhao et al. 2023, Dobias et al. 2025].

3.4.4. Criptografia Quântica

A criptografia quântica surge como uma alternativa aos métodos clássicos de proteção de dados, pois transfere a segurança da matemática para as leis fundamentais da mecânica quântica [Duim and Portácio 2023]. Em vez de confiar na dificuldade de calcular fatores numéricos ou logaritmos discretos, ela explora fenômenos como superposição, emaranhamento e o princípio da incerteza de Heisenberg para garantir que qualquer tentativa de espionagem seja detectável. Essa abordagem desvincula-se das vulnerabilidades expostas pelos algoritmos de Shor e Grover, pois mesmo um adversário com poder computacional ilimitado não consegue clonar estados quânticos sem introduzir ruídos mensuráveis no canal de comunicação. O componente central da criptografia quântica é a Distribuição Quântica de Chaves (QKD - *Quantum Key Distribution*), na qual duas partes compartilham uma sequência de *bits* aleatórios e secretos codificada na polarização de fótons [Duim and Portácio 2023]. O processo completo pode ser visualizado na Figura 3.14, que apresenta um esquema geral dessa comunicação segura.

Protocolos consagrados, como o BB84, combinam bases distintas (por exemplo as bases X e Z) para codificar 0s e 1s em estados quânticos distintos [Marquezino and Helayel-Neto 2003]. Durante a troca, o receptor seleciona aleatoriamente uma base para medir cada fóton; só quando a base coincide com a do emissor é possível recuperar corretamente o *bit* original. Ao comparar, por um canal clássico, as bases usadas (sem revelar os valores dos *bits*), ambas as partes conseguem eliminar medições incongruentes, reduzindo o tamanho da chave mas assegurando sua inviolabilidade

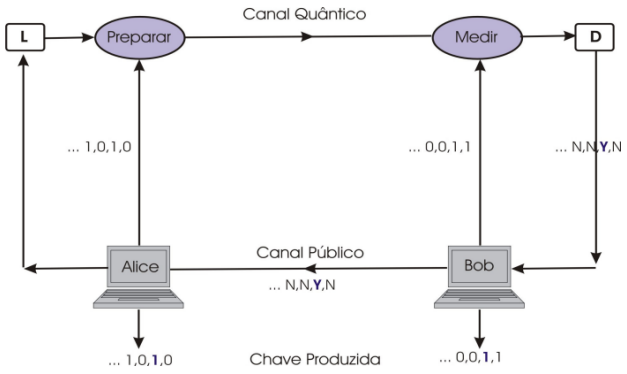


Figura 3.14. Processo completo do QKD. Adaptado de [Takagi 2003]

[Mendes et al. 2011]. Se a taxa de erro quântico (QBER - *Quantum Bit Error Rate*) exceder um limiar pré-definido, indica-se a presença de um intruso e todo o processo deve ser reiniciado para preservar a confidencialidade da comunicação.

Além do BB84, outras implementações, como o protocolo E91 (baseado em emaranhamento quântico), reforçam a robustez da QKD ao permitir que correlações quânticas garantam a geração de chaves idênticas mesmo à distância. Em aplicações práticas, combina-se a QKD com cifradores de uso único (*one-time pad*), obtendo-se segurança teórica perfeita: a interceptação administrativa de uma chave quântica deixa vestígios irrefutáveis, fazendo da criptografia quântica uma barreira efetiva contra ataques futuros que utilizem computador quântico [Mendes et al. 2011].

3.4.5. Estudo de Caso: Impactos no Bitcoin

O Bitcoin, protocolo proposto por [Nakamoto 2008], fundamenta-se em princípios de criptografia clássicos, como o algoritmo de assinatura digital ECDSA e as funções de *hash* SHA-256 e RIPEMD-160, que buscam garantir autenticidade, integridade e irretratabilidade das transações. Embora tais mecanismos sejam considerados seguros frente a ataques por computadores clássicos, estudos têm demonstrado que algoritmos quânticos, como os de Shor e Grover, podem comprometer sua segurança, evidenciando a necessidade de reavaliação da resiliência do protocolo Bitcoin em um cenário pós-quântico.

No contexto de assinaturas digitais, o Bitcoin emprega ECDSA sobre a curva elíptica secp256k1 como mecanismo padrão desde sua criação. A proposta de melhoria Taproot, ativada em 2021, introduziu o suporte a assinaturas Schnorr, que, embora baseadas na mesma curva, possuem características formais distintas, como linearidade e eficiência em agregação de chaves [Maxwell et al. 2020]. Apesar das sutis diferenças, ambos os esquemas se fundamentam na dificuldade do ECDLP, considerado computacionalmente intratável sob o paradigma clássico [Menezes et al. 2018].

Contudo, o algoritmo de Shor permite resolver o ECDLP em tempo polinomial em um computador quântico com *qubits* e correção de erros suficientes [Shor 1997], conforme vimos anterioremente. Dessa forma, qualquer chave pública que se torne visível na blockchain pode ser utilizada por um agente adversário, com acesso a um computador quântico suficientemente avançado, para derivar a chave privada associada e realizar uma

transação maliciosa [Aggarwal et al. 2017].

Além das assinaturas, o protocolo Bitcoin também utiliza funções de *hash*, sendo o algoritmo SHA-256 empregado em diversas camadas, incluindo a construção do cabeçalho de blocos no mecanismo de prova de trabalho (PoW - *Proof-of-Work*), a geração dos identificadores de transações (TXIDs) e a verificação de integridade de dados [Narayanan et al. 2016].

Diante deste cenário, algumas pesquisas têm investigado as vulnerabilidades do Bitcoin e explorado possíveis alternativas de adaptação do protocolo. Em um relatório da Chaincode Labs, foi avaliado os impactos de computadores quânticos criptograficamente relevantes sobre o ecossistema, estimando que aproximadamente 6,26 milhões de BTC encontram-se vulneráveis à extração de chaves privadas por meio do algoritmo de Shor [Milton and Shikhelman 2025]. O estudo categoriza os scripts de saída quanto à sua suscetibilidade, alinhando-se à análise de que P2PK, P2MS e P2TR são vulneráveis, enquanto P2PKH e P2WPKH se tornam vulneráveis durante o processo de gasto.

Entre as soluções propostas para mitigar esses riscos, destaca-se o `BIP-360` [Beast 2024], que introduz o conceito de *Pay-to-Quantum-Resistant-Hash* (P2QRH). Essa proposta visa permitir a utilização de assinaturas pós-quânticas diretamente em scripts de gasto por meio de algoritmos como Falcon, CRYSTALS-DILITHIUM e SPHINCS+, padronizados pelo NIST [National Institute of Standards and Technology (NIST) 2024]. A proposta prevê uma combinação de assinatura Schnorr com assinaturas pós-quânticas em um esquema híbrido e a adoção de novos formatos de endereço (com prefixo "bc1r").

Outra proposta é o `BIP-347`, que sugere a reintrodução do opcode `OP_CAT` com o objetivo de viabilizar construções baseadas em assinaturas de Lamport, uma classe de esquemas baseados em *hash* considerados resistentes a ataques quânticos. Ao possibilitar a concatenação de elementos na pilha de execução, o opcode permitiria a criação de scripts Taproot com caminhos de gasto alternativos baseados em assinaturas de Lamport [Heilman and Sabouri 2023]. De forma complementar, [Corallo 2024] propôs uma abordagem por meio da introdução de um opcode dedicado à verificação de assinaturas SPHINCS+ (`OP_SPHINCS`). A inclusão de tal verificação permitiria que carteiras adotassem, de maneira compatível, saídas com gastos criptograficamente seguros mesmo diante de CRQCs, ainda que com um aumento significativo no tamanho das transações.

3.5. Oportunidades em Computação Quântica

Esta seção explora aplicações da Computação Quântica em áreas como Processamento de Linguagem Natural (PLN) utilizando Quantum Machine Learning (QML), saúde e finanças, como por exemplo otimização de portfólios de investimento [Biamonte et al. 2017]. Esta seção possui como objetivo trazer uma visão complementar de oportunidades desta tecnologia emergente, além dos desafios e riscos associados aos impactos na criptografia.

3.5.1. QNLP

O PLN é o campo de estudo que desenvolve soluções computacionais para problemas e tarefas que envolvam o uso da linguagem, seja ela escrita ou falada, sendo dividido em duas vertentes: NLU (*Natural Language Understanding*), voltada à análise e interpreta-

ção de texto, e NLG (*Natural Language Generation*), dedicada à geração de linguagem coerente [Caseli and Nunes 2024].

Segundo [Nausheen et al. 2025], o uso da computação quântica se dá em aproveitar as características destes sistemas, como a superposição e o emaranhamento, para otimizar tanto a representação de características de um texto quanto a própria inferência de acordo com o tipo de tarefa adotada. Assim, otimizando seja a representação e extração de características, seja o próprio custo computacional, como demonstrado por [Correia et al. 2022], a incorporação do algoritmo de Grover ao procedimento de *Question Answering* reduziu a complexidade da busca de $O(N)$ para $O(\sqrt{N})$, proporcionando, portanto, uma aceleração quadrática na identificação da resposta correta. Ainda segundo [Nausheen et al. 2025], que mapearam o uso da computação quântica em tarefas de PLN, identificam-se três principais aplicações desta tecnologia no campo:

1. *Encoding*: é a codificação dos dados clássicos — no caso de PLN, o texto — em informação quântica, mapeando-os para estados quânticos. Essas técnicas de encoding facilitam uma representação eficiente dos dados e ajudam a capturar relações entre eles.

2. Modelagem: a utilização da computação quântica em nível de modelagem pode variar conforme a tarefa de linguagem natural adotada, mas, de modo geral, os modelos classificam-se em:

    - *Categorical QNLP models*: utilizam uma representação estrutural e gramatical da língua, combinando essa estrutura com o significado das palavras por meio de conceitos matemáticos como *compact closed categories*, *tensor products* e *diagrammatic calculus*. Subdivide-se em modelos Categóricos (DisCoCat) e modelos de Circuito (DisCoCirc).
    - *Probabilistic models*: exploram a representação geométrica do modelo de probabilidade quântica para maior flexibilidade na tomada de decisão, permitindo transições entre diferentes bases do espaço vetorial.
    - *Quantum circuit models*: usam circuitos parametrizados para realizar inferências. Incluem os VQA (*Variational Quantum Algorithms*), adequados aos dispositivos da era NISQ (*Noise Intermediate-Scale Quantum*), nos quais um estimador quântico avalia a função de custo do problema enquanto um otimizador clássico ajusta os parâmetros; e as QNNs (*Quantum Neural Networks*), variantes dos VQA que codificam dados via *feature map*, aplicam um modelo variacional para otimizar a *loss function* e, por fim, pós-processam classicamente os resultados de medida.
    - *Quantum Kernel models*: mapeiam dados clássicos em espaços de alta dimensão via *kernel* quântico, extraindo mais características e reduzindo ruído. Um exemplo é o QSVM (*Quantum Support Vector Machine*), que reformula a SVM (*Support Vector Machine*) em mínimos quadrados para inverter matrizes eficientemente e acelerar treinamento e classificação em *big data*.
    - *Quantum language models*: usam princípios quânticos para representar palavras como estados quânticos, capturando relações semânticas de forma mais

rica; variantes de inspiração quântica (*quantum-inspired*) simplificam isso mapeando cada palavra para um dos vetores da base ortogonal de referência e modelos como o SQLM (*Session-based Quantum Language Model*) aplicam certas transformações para rastrear a evolução das buscas dos usuários.

- *Hybrid models*: combinam processamento clássico e quântico, distribuindo etapas entre ambos, de modo a aproveitar os benefícios da computação quântica mesmo com as limitações de hardware da era NISQ.

3. *Hyperparameter tuning*: utiliza as características dos sistemas quânticos para melhorar a eficiência e a eficácia em tarefas de otimização. Dentre os principais métodos estão o *Quantum-Accelerated Hyperparameter Tuning*, o *Quantum Annealing* e abordagens híbridas.

A computação quântica já possui pesquisas de PLN em andamento com provas de conceito práticas [Correia et al. 2022, Omar and El-Hafeez 2023, Nausheen et al. 2025]. Por exemplo, [Omar and El-Hafeez 2023] desenvolveram um classificador de sentimentos para textos de redes sociais em árabe, utilizando QSVM para inferência em comparação a um modelo clássico de *Random Forest*, avaliando o desempenho de ambos em diversos conjuntos de dados. De qualquer forma, por estar em estágio inicial, ainda demanda pesquisas adicionais para consolidar métodos e ampliar seu impacto.

3.5.2. Aplicações em Saúde

Os recentes avanços da computação clássica, com o desenvolvimento de algoritmos e aumento no poder computacional amplamente disponível, permitiram a análise de dados no contexto das ciências da saúde e medicina de forma inédita. Com a computação quântica, diversas técnicas vêm sendo testadas e aprimoradas para acelerar algoritmos, reduzir o consumo de energia e flexibilizar os requisitos de dados para análise [Flöther 2023]. Nesta seção, delineiam-se os principais usos da computação quântica na área da saúde, destacando os algoritmos empregados e perspectivas futuras.

Inicialmente, como apontado em [Flöther 2023], os primeiros usos da computação quântica nas ciências biológicas se concentraram na resolução de problemas da bioquímica e biologia computacional, sobretudo por meio de técnicas de simulação quântica de moléculas, como proteínas. Esse caso de uso atraiu grande interesse por representar um dos principais gargalos em plataformas clássicas, em razão de sua elevada complexidade. Além disso, a viabilidade de execução em hardware quântico ruidoso atualmente disponível, somado ao potencial de escalabilidade com o desenvolvimento de dispositivos mais poderosos, motivou a busca de soluções quânticas para a superação dos impasses correntes [Robert et al. 2021]. Nesse cenário, esperam-se aplicações ainda mais avançadas, por exemplo, na predição de estruturas de proteínas com aminoácidos sintéticos, situação na qual mesmo técnicas de *machine learning* falham devido a falta de dados para treinamento, o que reduziria a dependência de experimentos laboratoriais caros e demorados.

Embora o maior destaque recaia sobre a simulação de compostos, as aplicações quânticas na saúde não se limitam a isso. Nesse contexto, a genômica logo passou a explorar o poder de processamento dessas máquinas trabalhar com grandes volumes de dados com estruturas complexas. Tanto algoritmos quânticos consolidados quanto

técnicas de QML foram avaliadas para acelerar tarefas genômicas. Nesse âmbito, em [Sarkar et al. 2021] propõe-se a utilização do algoritmo de Grover para acelerar o alinhamento de sequências de DNA. Já em [Prousalis and Konofaos 2019] elabora-se sobre a utilização de um algoritmo de *Quantum Pattern Recognition* (QPR) para pareamento de sequências de material genético. Curiosamente, o QPR utiliza como subrotina a Transformada Quântica de Fourier (QFT - *Quantum Fourier Transform*), também empregada no algoritmo de Shor, evidenciando a versatilidade dessas técnicas.

Por fim, quando lançamos o olhar para a utilização de algoritmos de *quantum machine learning*, de maneira semelhante ao que ocorre com suas contrapartes clássicas, observa-se ampla variedade de aplicações, principalmente em diagnóstico. *Quantum Neural Networks* (QNNs), *Quantum K-Nearest Neighbors*, *Quantum Support Vector Classifiers* (QSVCs), *Quantum Random Forest*, entre outras abordagens, têm sido usadas para detectar diversos distúrbios de saúde. Destacam-se os trabalhos que utilizam QNNs em exames de imagem para o diagnóstico de Alzheimer [Shahwar et al. 2022] e que fazem um comparativo de técnicas de machine learning clássico e quântico para a predição de insuficiência cardíaca a partir de indicadores clínicos [Kumar et al. 2021].

É interessante notar que, desde o seu surgimento, a medicina vem se apropriando das tecnologias mais recentes de cada momento histórico [Flöther 2023]. Assim, o atual entusiasmo na criação de aplicações médicas com computação quântica, seja para aprimorar métodos existentes ou para fomentar descobertas inéditas, revela o potencial transformador dessa tecnologia, prometendo alterar profundamente o cuidado à saúde e contribuir para a melhora na qualidade de vida da população.

3.5.3. Aplicações em Finanças

A computação quântica tem emergido como uma tecnologia com grande potencial de aplicação em diversos setores, especialmente nas finanças, onde a complexidade dos modelos matemáticos e a demanda por cálculos intensivos representam desafios significativos para a computação clássica [Chang et al. 2023]. Vale destacar, algoritmos quânticos têm sido propostos para o cálculo de medidas de risco, como o valor em risco (VaR) e o valor condicional em risco (CVaR), tendo assim forte atuação na gestão de carteiras de crédito e na tomada de decisão sob incerteza [Miyamoto 2022]. Essas inovações sugerem uma computação quântica extensiva, podendo impactar não só um mercado local, mas em escala de economia global, trazendo melhorias na precisão e na eficiência do processamento de grandes volumes de dados financeiros.

Dentro da diversidade de aplicações possíveis com a tecnologia, vale detalhar aqui a aplicação na análise de risco de portfólios de investimentos. O trabalho [Woerner and Egger 2019] apresenta um estudo que utiliza utiliza a plataforma de computação quântica da IBM para precificar um título do tesouro americano sob cenários de aumento de juros, além de simular o cálculo de risco financeiro para uma carteira composta por dois ativos de dívida pública com diferentes vencimentos. Os resultados apresentaram, para os problemas de escala reduzida que foram estudados, uma taxa de convergência superior aos métodos clássicos (Monte Carlo) mesmo com interferência de erros quânticos. Já em [Rebentrost and Lloyd 2024] é proposto algoritmo quântico para otimização de carteiras que permitem determinar a curva de *trade-off* entre risco e retorno

e amostrar a partir do portfólio ideal. Esse algoritmo utiliza dados históricos de retornos dos ativos e afirmam alcançar um tempo de execução de $log(N)$, superando os algoritmos clássicos com tempo de execução polinomial N.

A aplicação de técnicas de computação quântica para a otimização de portfólios também tem se expandido por meio de abordagens híbridas, como o *Variational Quantum Eigensolver* (VQE). Em [Buonaiuto et al. 2023] é apresentada uma formulação geral do problema de otimização quadrática restrita, transformado em um problema de Otimização Binária Quadrática Sem Restrições (QUBO - *Quadratic Unconstrained Binary Optimization*). Essa abordagem permite lidar com a complexidade que existe na seleção de carteiras sob restrições financeiras. Em outro trabalho [Veselỳ 2022], é explorado uma estratégia híbrida clássico-quântica usando recozimento quântico no computador D-Wave 2000Q, onde soluções iniciais geradas por métodos clássicos são refinadas por meio de recozimento quântico reverso, o que permite obter melhores tempos de convergência conforme o número de variáveis aumenta. Além disso, em [Lang et al. 2022], é proposto um fluxo de trabalho que combina pré-processamento clássico com uma nova formulação do modelo QUBO, permitindo flexibilidade quanto ao número de ativos e ao valor investido em cada um. Esse modelo foi testado em diferentes plataformas de recozimento, incluindo simulação clássica (*simulated annealing*), recozimento digital (*Fujitsu Digital Annealer*) e recozimento quântico (*D-Wave Advantage*), utilizando dados reais da bolsa de Nova York.

Diante desse cenário, é possível observar pesquisas recentes que demonstram avanços na aplicação de algoritmos quânticos híbridos e modelos de otimização adaptados a hardwares reais, indicando que as finanças podem ser uma das primeiras áreas a colher benefícios tangíveis dessa tecnologia [Khang et al. 2025]. No entanto, nota-se também que será necessário superar barreiras como os erros quânticos, os custos elevados de implementação e ainda pensar em questões éticas no uso da computação quântica. Embora os computadores quânticos ainda estejam em fase de desenvolvimento , seu potencial para transformar o setor financeiro é evidente, especialmente no que diz respeito à otimização de portfólios, precificação de ativos e gestão de risco [Ciacco et al. 2025].

3.6. Considerações Finais

Diante das ameaças da tecnologia emergente de Computação Quântica aos mecanismos criptográficos atuais, sobretudo aqueles baseados nos problemas de logaritmo discreto e de fatoração de números inteiros (e.g., cifras assimétricas como o RSA), este minicurso teve como objetivo introduzir os fundamentos da computação quântica e examinar seus impactos em criptografia, suportando um aprofundamento maior nas razões para esses impactos a partir do entendimento de seus fundamentos matemáticos, além de fornecer uma visão mais abrangente sobre possíveis soluções além da criptografia pós-quântica, que são os principais diferenciais em relação aos minicursos apresentados em edições anteriores do SBSeg [Barreto et al. 2013, Paiva et al. 2023].

Este material abordou fundamentos da criptografia clássica na Seção 3.2, passando pelos princípios da computação quântica na Seção 3.3, até os potenciais ataques quânticos sobre algoritmos criptográficos atuais e as soluções em desenvolvimento para mitigar esses riscos na Seção 3.4, buscando suportar uma compreensão não só dos impac-

tos associados, mas também de suas justificativas. Caso a Computação Quântica venha a impactar outras classes de problemas atuais considerados intratáveis em computadores clássicos, este entendimento detalhado pode fomentar novas perspectivas de exploração de outros problemas matemáticos em novos algoritmos de criptografia pós-quântica, ou até novas abordagens como a criptografia quântica. De forma complementar, a Seção 3.5 traz algumas oportunidades de aplicações de Computação Quântica para mostrar que esta tecnologia também fomenta inovação.

Um aspecto não menos importante e que vale para reflexão é sobre a ética associada à Computação Quântica, considerando que devemos endereçar o uso de recursos como hardware, software, infraestrutura e sistemas de informação em ciência, engenharia e tecnologia para as pessoas físicas e jurídicas. Além das táticas de resistir apresentadas neste texto, o NIST também recomenda as táticas de Detectar e Recuperar [Johnson 2016, Pascoe 2023]. Os ataques são elaborados e executados com sofisticação por parte daqueles que não tem nenhum compromisso ético, e que fazem parte do nosso convívio. Podem estar próximos como membros de equipes internas ou fornecedores atacando ou até cooptando colaboradores por engenharia social para obter dados (e.g., credenciais) que os aproximam dos ativos (e.g., ativos criptográficos[2]).

O minicurso é resultado de esforços de capacitação em Computação Quântica que ocorrem no Banco Bradesco, como disciplina de Pós-Graduação na Escola Politécnica da Universidade de São Paulo (USP), e em cursos optativos para a Graduação em Ciência da Computação, e de especialização no Instituto de Tecnologia e Liderança (Inteli). Para o uso desta tecnologia emergente, a formação de mão-de-obra especializada é algo essencial, e iniciativas de educação como este minicurso se tornam relevantes, sobretudo considerando que 2025 foi eleito como o Ano Internacional da Ciência e Tecnologia Quânticas pelas Nações Unidas [Bongs 2025]. Espera-se que este material possa fomentar novos cursos, workshops e outros formatos de capacitação nesta área.

Referências

[44 U.S. Code § 3542 – Definitions 2013] 44 U.S. Code § 3542 – Definitions (2013). United states code. U.S. Code. Disponível em: `https://www.govinfo.gov/app/details/USCODE-2013-title44/USCODE-2013-title44-chap35-subchapIII-sec3542`. Acesso em: 15 jul. 2025.

[Aggarwal et al. 2017] Aggarwal, D., Brennen, G. K., Lee, T., Santha, M., and Tomamichel, M. (2017). Quantum attacks on bitcoin, and how to protect against them. Technical report, arXiv preprint arXiv:1710.10377. `https://arxiv.org/abs/1710.10377`.

[Ajtai 1996] Ajtai, M. (1996). Generating hard instances of lattice problems. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 99–108.

[2] `https://github.com/IBM/CBOM`

[Alagic et al. 2022] Alagic, G., Alagic, G., Apon, D., Cooper, D., Dang, Q., Dang, T., Kelsey, J., Lichtinger, J., Liu, Y.-K., Miller, C., et al. (2022). Status report on the third round of the nist post-quantum cryptography standardization process.

[Alagic et al. 2025] Alagic, G., Bros, M., Ciadoux, P., Cooper, D., Dang, Q., Dang, T., Kelsey, J., Lichtinger, J., Liu, Y.-K., Miller, C., et al. (2025). *Status report on the fourth round of the nist post-quantum cryptography standardization process*. US Department of Commerce, National Institute of Standards and Technology.

[Aono et al. 2022] Aono, Y., Liu, S., Tanaka, T., Uno, S., Meter, R. V., Shinohara, N., and Nojima, R. (2022). The present and future of discrete logarithm problems on noisy quantum computers. *IEEE Transactions on Quantum Engineering*, 3:1–21.

[Aumasson 2017] Aumasson, J.-P. (2017). *Serious Cryptography: A Practical Introduction to Modern Encryption*. No Starch Press, USA.

[Barker et al. 2020] Barker, E., Roginsky, A., and National Institute of Standards and Technology (NIST) (2020). Recommendation for key management: Part 1 – general (revision 5). Technical Report NIST SP 800-57pt1r5, National Institute of Standards and Technology. Supersedes NIST SP 800-57 Part 1 Rev. 4 (2016).

[Barreto et al. 2013] Barreto, P., Biasi, F. P., Dahab, R., César, J., Pereira, G., and Ricardini, J. E. (2013). Introdução à criptografia pós-quântica. *Minicursos do XIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais, SBSeg*.

[Beast 2024] Beast, A. (2024). BIP-360: Pay to Quantum Resistant Hash (P2QRH). `https://github.com/bitcoin/bips/pull/1670`.

[Bernstein et al. 2009] Bernstein, D. J., Buchmann, J., and Dahmen, E., editors (2009). *Post-Quantum Cryptography*. Mathematics and Statistics. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 1 edition. eBook ISBN: 978-3-540-88702-7.

[Beullens et al. 2021] Beullens, W., D'Anvers, J.-P., Hülsing, A. T., Lange, T., Panny, L., de Saint Guilhem, C., and Smart, N. P. (2021). Post-quantum cryptography: Current state and quantum mitigation. Technical report, ENISA, Attiki, Greece.

[Biamonte et al. 2017] Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., and Lloyd, S. (2017). Quantum machine learning. *Nature*, 549(7671):195–202.

[Bongs 2025] Bongs, K. (2025). Celebrating the international year of quantum science and technology.

[Bonneau et al. 2015] Bonneau, J., Miller, A., Clark, J., Narayanan, A., Kroll, J. A., and Felten, E. W. (2015). Sok: Research perspectives and challenges for bitcoin and cryptocurrencies. In *2015 IEEE Symposium on Security and Privacy*, pages 104–121.

[Boudot et al. 2020a] Boudot, F., Gaudry, P., Guillevic, A., Heninger, N., Thomé, E., and Zimmermann, P. (2020a). Comparing the difficulty of factorization and discrete logarithm: a 240-digit experiment. Cryptology ePrint Archive, Paper 2020/697.

[Boudot et al. 2020b] Boudot, F., Gaudry, P., Guillevic, A., Heninger, N., Thomé, E., and Zimmermann, P. (2020b). Factorization of rsa-250.

[Brassard et al. 1998] Brassard, G., HØyer, P., and Tapp, A. (1998). *Quantum cryptanalysis of hash and claw-free functions: Invited paper*, page 163–169. Springer Berlin Heidelberg.

[Buonaiuto et al. 2023] Buonaiuto, G., Gargiulo, F., De Pietro, G., Esposito, M., and Pota, M. (2023). Best practices for portfolio optimization by quantum computing, experimented on real quantum devices. *Scientific Reports*, 13(1):19434.

[Caseli and Nunes 2024] Caseli, H. d. M. and Nunes, M. d. G. V. (2024). *Processamento de Linguagem Natural: Conceitos, Técnicas e Aplicações em Português*. Brasileiras em PLN (BPLN).

[Castryck and Decru 2023] Castryck, W. and Decru, T. (2023). An efficient key recovery attack on sidh. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 423–447. Springer.

[Chang et al. 2023] Chang, Y.-J., Sie, M.-F., Liao, S.-W., and Chang, C.-R. (2023). The prospects of quantum computing for quantitative finance and beyond. *IEEE Nanotechnology Magazine*, 17(2):31–37.

[Chen et al. 2017] Chen, L., Moody, D., and Liu, Y. (2017). Nist post-quantum cryptography standardization. *Transition*, 800(131A):164.

[Ciacco et al. 2025] Ciacco, A., Guerriero, F., and Macrina, G. (2025). Review of quantum algorithms for medicine, finance and logistics. *Soft Computing*, 29(4):2129–2170.

[Cooper et al. 2020] Cooper, D. A., Apon, D. C., Dang, Q. H., Davidson, M. S., Dworkin, M. J., Miller, C. A., et al. (2020). Recommendation for stateful hash-based signature schemes. *NIST Special Publication*, 800(208):800–208.

[Corallo 2024] Corallo, M. (2024). Proposal for OP_SPHINCS as Post-Quantum Signature Opcode. Bitcoin-dev mailing list.

[Correia et al. 2022] Correia, A. D., Moortgat, M., and Stoof, H. T. C. (2022). Quantum computations for disambiguation and question answering. *arXiv preprint arXiv:2106.05299*.

[Diffie and Hellman 2022] Diffie, W. and Hellman, M. E. (2022). New directions in cryptography. In *Democratizing cryptography: the work of Whitfield Diffie and Martin Hellman*, pages 365–390.

[Ding et al. 2006] Ding, J., Gower, J. E., and Schmidt, D. S. (2006). *Multivariate public key cryptosystems*. Springer.

[Dobias et al. 2025] Dobias, P., Rezaeezade, A., Chmielewski, Ł., Malina, L., and Batina, L. (2025). Sok: Reassessing side-channel vulnerabilities and countermeasures in pqc implementations. *Cryptology ePrint Archive*.

[Duim and Portácio 2023] Duim, J. L. and Portácio, R. G. (2023). Segurança criptográfica: Combinando métodos clássicos e quânticos. *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, 10(1).

[FIPS PUB 202 2015] FIPS PUB 202 (2015). SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. `https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf`. FIPS PUB 202.

[Flöther 2023] Flöther, F. F. (2023). The state of quantum computing applications in health and medicine. *Research Directions: Quantum Technologies*, 1:e10.

[Gamble 2019] Gamble, S. (2019). Quantum computing: What it is, why we want it, and how we're trying to get it. In *Frontiers of Engineering: Reports on Leading-Edge Engineering from the 2018 Symposium*. National Academies Press (US).

[Gentry 2009] Gentry, C. (2009). Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 169–178.

[Grover 1996] Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, page 212–219, New York, NY, USA. Association for Computing Machinery.

[Heilman and Sabouri 2023] Heilman, E. and Sabouri, A. (2023). BIP-347: Reintroducing OP_CAT for Lamport Signatures. https://github.com/bitcoin/bips/blob/master/bip-0347.mediawiki.

[Hhan et al. 2023] Hhan, M., Yamakawa, T., and Yun, A. (2023). Quantum complexity for discrete logarithms and related problems. Cryptology ePrint Archive, Paper 2023/1054.

[Hosoyamada and Sasaki 2020] Hosoyamada, A. and Sasaki, Y. (2020). Finding hash collisions with quantum computers by using differential trails with smaller probability than birthday bound. Cryptology ePrint Archive, Paper 2020/213.

[Huang et al. 2024] Huang, Z., Wang, H., Cao, B., He, D., and Wang, J. (2024). A comprehensive side-channel leakage assessment of crystals-kyber in iiot. *Internet of Things*, 27:101331.

[Häner et al. 2020] Häner, T., Jaques, S., Naehrig, M., Roetteler, M., and Soeken, M. (2020). Improved quantum circuits for elliptic curve discrete logarithms. Cryptology ePrint Archive, Paper 2020/077.

[Jao and De Feo 2011] Jao, D. and De Feo, L. (2011). Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *International workshop on post-quantum cryptography*, pages 19–34. Springer.

[Jedlicka et al. 2022] Jedlicka, P., Malina, L., Socha, P., Gerlich, T., Martinasek, Z., and Hajny, J. (2022). On secure and side-channel resistant hardware implementations of post-quantum cryptography. In *Proceedings of the 17th International Conference on Availability, Reliability and Security*, pages 1–9.

[Johnson 2016] Johnson, C. (2016). Guide to cyber threat information sharing. *NIST Special Publication*, pages 800–150.

[Joseph et al. 2022] Joseph, D., Misoczki, R., and Manzano, M. e. a. (2022). Transitioning organizations to post-quantum cryptography. *Nature*, 605:237–243.

[Kasirajan 2021] Kasirajan, V. (2021). *Fundamentals of Quantum Computing: Theory and Practice*. Springer Cham.

[Katz and Lindell 2014] Katz, J. and Lindell, Y. (2014). *Introduction to Modern Cryptography, Second Edition*. Chapman & Hall/CRC, 2nd edition.

[Khan et al. 2024] Khan, S., Krishnamoorthy, P., Goswami, M., Rakhimjonovna, F. M., Mohammed, S. A., and Menaga, D. (2024). Quantum computing and its implications for cybersecurity: A comprehensive review of emerging threats and defenses. *Nanotechnology Perceptions*, 20:S13.

[Khang et al. 2025] Khang, A., Rath, K. C., Madapana, K., Rao, J., Panda, L. P., and Das, S. (2025). Quantum computing and portfolio optimization in finance services. In *Shaping Cutting-Edge Technologies and Applications for Digital Banking and Financial Services*, pages 27–45. Productivity Press.

[Knuth 1997] Knuth, D. E. (1997). *The art of computer programming, volume 1 (3rd ed.): fundamental algorithms*. Addison Wesley Longman Publishing Co., Inc., USA.

[Kosmann-Schwarzbach and Singer 2010] Kosmann-Schwarzbach, P. Y. and Singer, S. F. (2010). *Lie Groups SU(2) and SO(3)*, pages 71–80. Springer New York, New York, NY.

[Kumar et al. 2021] Kumar, Y., Koul, A., Sisodia, P. S., Shafi, J., Verma, K., Gheisari, M., and Davoodi, M. B. (2021). Heart failure detection using quantum-enhanced machine learning and traditional machine learning techniques for internet of artificially intelligent medical things. *Wireless Communications and Mobile Computing*, 2021(1):1616725.

[Lamport 1981] Lamport, L. (1981). Password authentication with insecure communication. *Communications of the ACM*, 24(11):770–772.

[Lang et al. 2022] Lang, J., Zielinski, S., and Feld, S. (2022). Strategic portfolio optimization using simulated, digital, and quantum annealing. *Applied Sciences*, 12(23):12288.

[Lyubashevsky et al. 2010] Lyubashevsky, V., Peikert, C., and Regev, O. (2010). On ideal lattices and learning with errors over rings. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 1–23. Springer.

[Marquezino and Helayel-Neto 2003] Marquezino, F. and Helayel-Neto, J. (2003). Estudo introdutório do protocolo quântico bb84 para troca segura de chaves. *Centro Brasileiro de Pesquisas Fısicas, Série Monografias*.

[Maxwell et al. 2020] Maxwell, G., Poelstra, A., Seurin, Y., and Wuille, P. (2020). BIP-340: Schnorr Signatures for secp256k1. `https://github.com/bitcoin/bips/blob/master/bip-0340.mediawiki`.

[McEliece 1978] McEliece, R. J. (1978). A public-key cryptosystem based on algebraic. *Coding Thv*, 4244(1978):114–116.

[Mendes et al. 2011] Mendes, Á. J. B., Paulicena, E. H., and Souza, W. A. R. d. (2011). Criptografia quântica: uma abordagem direta. *Revista de Sistema de Informação da FSMA*, (7):39–48.

[Menezes et al. 2018] Menezes, A. J., Van Oorschot, P. C., and Vanstone, S. A. (2018). *Handbook of applied cryptography*. CRC press.

[Merkle 1979] Merkle, R. C. (1979). *Secrecy, authentication, and public key systems.* Stanford university.

[Milton and Shikhelman 2025] Milton, A. and Shikhelman, C. (2025). Bitcoin and the quantum threat: A comprehensive risk analysis. Technical report, Chaincode Labs Research Report. `https://chaincode.com/bitcoin-post-quantum.pdf`.

[Miyamoto 2022] Miyamoto, K. (2022). Quantum algorithm for calculating risk contributions in a credit portfolio. *EPJ Quantum Technology*, 9(1):1–16.

[Moody et al. 2024] Moody, D., Perlner, R., Regenscheid, A., Robinson, A., and Cooper, D. (2024). Transition to post-quantum cryptography standards. Technical report, National Institute of Standards and Technology.

[Mosca 2018] Mosca, M. (2018). Cybersecurity in an era with quantum computers: Will we be ready? *IEEE Security & Privacy*, 16(5):38–41.

[Nakamoto 2008] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. `https://bitcoin.org/bitcoin.pdf`.

[Narayanan et al. 2016] Narayanan, A., Bonneau, J., Felten, E., Miller, A., and Goldfeder, S. (2016). *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press.

[National Institute of Standards and Technology (NIST) 2024] National Institute of Standards and Technology (NIST) (2024). Post-quantum cryptography standardization project – finalist algorithms. Technical report, US Department of Commerce. `https://csrc.nist.gov/projects/post-quantum-cryptography`.

[Nausheen et al. 2025] Nausheen, F., Ahmed, K., and Khan, M. I. (2025). Quantum natural language processing: A comprehensive review of models, methods, and applications. *arXiv preprint arXiv:2504.09909*. Preprint.

[Nielsen and Chuang 2010] Nielsen, M. A. and Chuang, I. L. (2010). *Quantum computation and quantum information*. Cambridge university press.

[NSA 2021] NSA (2021). *Frequently Asked Questions: Quantum Computing and Post-Quantum Cryptography*. National Security Agency.

[of Standards et al. 2023] of Standards, N. I., (NIST), T., Dworkin, M. J., Turan, M. S., and Mouha, N. (2023). Advanced encryption standard (aes).

[Omar and El-Hafeez 2023] Omar, A. and El-Hafeez, T. A. (2023). Quantum computing and machine learning for arabic language sentiment classification in social media. *Scientific Reports*, 13(1):17305.

[Paar et al. 2024] Paar, C., Pelzl, J., and Güneysu, T. (2024). *Understanding cryptography: from established symmetric and asymmetric ciphers to post-quantum algorithms*. Springer Nature.

[Paiva et al. 2023] Paiva, T. B., Ponciano, V., Moreira, E., Oliveira, R., Rufino, V., Lima, C., López, J., Ueda, E., and Terada, R. (2023). Explorando esquemas criptográficos pós-quânticos considerados pelo nist com implementação em sage. *Anais*.

[Pascoe 2023] Pascoe, C. E. (2023). Public draft: The nist cybersecurity framework 2.0. *National Institute of Standards and Technology*.

[Prousalis and Konofaos 2019] Prousalis, K. and Konofaos, N. (2019). A quantum pattern recognition method for improving pairwise sequence alignment. *Scientific Reports*, 9(1):7226.

[Ravi et al. 2024] Ravi, P., Chattopadhyay, A., D'Anvers, J. P., and Baksi, A. (2024). Side-channel and fault-injection attacks over lattice-based post-quantum schemes (kyber, dilithium): Survey and new results. *ACM Transactions on Embedded Computing Systems*, 23(2):1–54.

[Rebentrost and Lloyd 2024] Rebentrost, P. and Lloyd, S. (2024). Quantum computational finance: quantum algorithm for portfolio optimization. *KI-Künstliche Intelligenz*, pages 1–12.

[Regev 2005] Regev, O. (2005). On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '05, page 84–93, New York, NY, USA. Association for Computing Machinery.

[Rivest et al. 1978] Rivest, R. L., Shamir, A., and Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126.

[Robert et al. 2021] Robert, A., Barkoutsos, P. K., Woerner, S., et al. (2021). Resource-efficient quantum algorithm for protein folding. *npj Quantum Information*, 7(1):38.

[Roetteler et al. 2017] Roetteler, M., Naehrig, M., Svore, K. M., and Lauter, K. (2017). Quantum resource estimates for computing elliptic curve discrete logarithms. Cryptology ePrint Archive, Paper 2017/598.

[Saarinen 2022] Saarinen, M.-J. O. (2022). Wip: Applicability of iso standard side-channel leakage tests to nist post-quantum cryptography. In *2022 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pages 69–72. IEEE.

[Sakurai and Napolitano 2020] Sakurai, J. J. and Napolitano, J. (2020). *Modern Quantum Mechanics (3rd ed.)*. Cambridge University Press.

[Salvi 2023] Salvi, P. (2023). How to calculate big o notation time complexity.

[Sarah and Peter 2024] Sarah, D. and Peter, C. (2024). *On the practical cost of Grover for AES key recovery*. UK National Cyber Security Centre.

[Sarkar et al. 2021] Sarkar, A., Al-Ars, Z., Almudever, C. G., and Bertels, K. L. M. (2021). Qibam: Approximate sub-string index search on quantum accelerators applied to dna read alignment. *Electronics*, 10(19).

[Shafique et al. 2024] Shafique, M. A., Munir, A., and Latif, I. (2024). Quantum computing: Circuits, algorithms, and applications. *IEEE Access*, 12:22296–22314.

[Shah et al. 2025] Shah, P., Prajapati, P., and Patel, D. (2025). Lattice-based post quantum cryptography using variations of learning with error (lwe). In Patel, K. K., Santosh, K., Gomes de Oliveira, G., Patel, A., and Ghosh, A., editors, *Soft Computing and Its Engineering Applications*, pages 58–72, Cham. Springer Nature Switzerland.

[Shahwar et al. 2022] Shahwar, T., Zafar, J., Almogren, A., Zafar, H., Rehman, A. U., Shafiq, M., and Hamam, H. (2022). Automated detection of alzheimer's via hybrid classical quantum neural networks. *Electronics*, 11(5).

[Shor 1994] Shor, P. (1994). Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134.

[Shor 1997] Shor, P. W. (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509.

[Singh 1999] Singh, S. (1999). *The Code Book: The Evolution of Secrecy from Mary, Queen of Scots, to Quantum Cryptography*. Doubleday, USA, 1st edition.

[Sipser 2012] Sipser, M. (2012). *Introduction to the Theory of Computation, 3rd Edition*. Thomson Course Technology.

[Smite-Meister 2023] Smite-Meister (2023). Bloch sphere.

[Smythe 2021] Smythe, W. (2021). Qm 101: Bloch sphere.

[Stallings 2013] Stallings, W. (2013). *Cryptography and Network Security: Principles and Practice*. Prentice Hall Press, USA, 6th edition.

[Susskind and Friedman 2014] Susskind, L. and Friedman, A. (2014). *Quantum Mechanics: The Theoretical Minimum*. Basic Books.

[Takagi 2003] Takagi, N. H. (2003). Fundamentos matemáticos da criptografia quântica.

[Terada 2008] Terada, R. (2008). *Segurança de dados: criptografia em redes de computador*. Edgard Blücher, São Paulo, 1 edition. 1ª reimpressão: 2011.

[Veselỳ 2022] Veselỳ, M. (2022). Application of quantum computers in foreign exchange reserves management. *arXiv preprint arXiv:2203.15716*.

[Wang et al. 2023] Wang, R., Ngo, K., Gärtner, J., and Dubrova, E. (2023). Single-trace side-channel attacks on crystals-dilithium: Myth or reality? *Cryptology ePrint Archive*.

[Wang et al. 2004] Wang, X., Feng, D., Lai, X., and Yu, H. (2004). Collisions for hash functions md4, md5, haval-128 and ripemd. *Cryptology ePrint Archive*.

[Wang et al. 2005] Wang, X., Yin, Y. L., and Yu, H. (2005). Finding collisions in the full sha-1. In *Annual international cryptology conference*, pages 17–36. Springer.

[Watrous 2025] Watrous, J. (2025). Understanding quantum information and computation.

[Woerner and Egger 2019] Woerner, S. and Egger, D. J. (2019). Quantum risk analysis. *npj Quantum Information*, 5(1):15.

[Zhang et al. 2011] Zhang, M., Xi, Z., and Wei, J.-H. (2011). Manipulating quantum information on the controllable systems or subspaces.

[Zhao et al. 2023] Zhao, Y., Pan, S., Ma, H., Gao, Y., Song, X., He, J., and Jin, Y. (2023). Side channel security oriented evaluation and protection on hardware implementations of kyber. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 70(12):5025–5035.

Capítulo 4

Wi-Fi Sensing e CSI aplicados à Cibersegurança: Fundamentos, Aplicações e Prática

Felipe Silveira de Almeida (ITA e Exército Brasileiro), Eduardo Fabrício Gomes Trindade (ITA e e Exército Brasileiro), Gioliano de Oliveira Braga (ITA), Ágney Lopes Roth Ferraz (ITA), Giovani Hoff da Costa (ITA), Gustavo Cavalcanti Morais (ITA), Lourenço Alves Pereira Júnior (ITA)

Abstract

Digital authentication faces increasing challenges due to the limitations of traditional methods, such as passwords and conventional biometrics, in the face of sophisticated attacks like spoofing and relay. In this context, leveraging Channel State Information (CSI) from Wi-Fi networks emerges as a promising solution for secure, non-intrusive authentication. This chapter covers the theoretical foundations of Wi-Fi CSI, describes methods for capturing and preprocessing data, and presents practical demonstrations of authentication and intrusion detection using low-cost devices such as ESP32 and Raspberry Pi. Finally, current challenges and future trends are discussed, highlighting opportunities for developing robust and adaptable CSI-based cybersecurity systems.

Resumo

A autenticação digital enfrenta desafios crescentes devido às limitações de métodos tradicionais, como senhas e biometria convencional, diante de ataques sofisticados como spoofing e relay. Neste contexto, a utilização de Informações do Estado do Canal (Channel State Information – CSI) de redes Wi-Fi emerge como uma solução promissora para autenticação segura e não intrusiva. Este capítulo aborda os fundamentos teóricos do Wi-Fi CSI, detalha métodos de captura e pré-processamento dos dados, e apresenta casos práticos de autenticação e detecção de intrusão usando dispositivos de baixo custo, como ESP32 e Raspberry Pi. Ao final, discutem-se desafios atuais e tendências futuras, destacando oportunidades para o desenvolvimento de sistemas robustos e adaptáveis de cibersegurança baseados em CSI.

4.1. Introdução e Motivação

A infraestrutura Wi-Fi tornou-se onipresente na vida moderna. De residências e escritórios a aeroportos, fábricas e instituições militares, redes sem fio baseadas no padrão IEEE 802.11 passaram de soluções convenientes para componentes críticos de comunicação, controle e automação. Estima-se que, em 2025, mais de 30 bilhões de dispositivos estarão conectados por Wi-Fi em escala global, impulsionando aplicações que vão desde entretenimento doméstico até processos industriais de missão crítica[Cisco 2020]. Essa penetração massiva consolidou o Wi-Fi como infraestrutura de conectividade essencial — mas também como vetor privilegiado de ameaças.

Historicamente, o Wi-Fi foi projetado com foco na mobilidade e facilidade de implantação, o que trouxe implicações de segurança desde sua origem. Apesar da evolução dos protocolos de criptografia (como WPA3), diversos ataques exploram características subjacentes da camada física e do protocolo MAC, contornando os mecanismos tradicionais de defesa. A vulnerabilidade a ataques como *deauthentication*, *evil twin*, jamming seletivo e exploração de *management frames* é explorada por atacantes com conhecimento moderado, armados com ferramentas de baixo custo. Esse cenário impõe novos desafios para pesquisadores e profissionais de cibersegurança: como proteger um canal compartilhado, sem fio, ubíquo e inerentemente exposto?

Nos últimos anos, a comunidade científica passou a enxergar o Wi-Fi sob uma nova ótica: para além de apenas um meio de comunicação, um canal sensível às perturbações do ambiente. Essa mudança paradigmática deu origem ao conceito de *Wi-Fi Sensing*, em que a própria infraestrutura de comunicação é utilizada como sensor passivo de eventos físicos. O princípio é simples e poderoso: ao transmitir sinais OFDM em múltiplas subportadoras, os dispositivos Wi-Fi sofrem influência direta de obstáculos, movimentos e alterações no entorno. Essa influência é captada na forma de variações sutis no *Channel State Information* (CSI), um conjunto de coeficientes complexos que descreve a resposta do canal para cada subportadora e cada par de antenas.

O acesso a dados de CSI em placas comerciais foi inicialmente limitado, mas iniciativas como o Intel 5300 CSI Tool [Halperin et al. 2011], Nexmon [Schulz et al. 2017], ESP32-CSI [Systems 2022], entre outras, abriram caminho para a captura de CSI em tempo real com granularidade suficiente para análise de sinais ambiente. A sensibilidade do CSI é tamanha que pequenas variações são detectáveis e mensuráveis, tais como: o movimento da palma da mão, a respiração de uma pessoa ou a vibração de um objeto. Isso habilita uma nova categoria de aplicações, incluindo monitoramento de sinais vitais[Liu et al. 2015], reconhecimento de gestos[Abdelnasser et al. 2015], detecção de intrusão física[Li et al. 2017] e autenticação por presença[Liu et al. 2014].

A consolidação desta área veio com a criação do grupo de trabalho IEEE 802.11bf, que adiciona formalmente suporte a funcionalidades de *sensing* na pilha Wi-Fi[IEEE 802.11 Working Group 2024]. Essa padronização viabiliza, por exemplo, que um ponto de acesso informe a variação de fase entre subportadoras como parte das medições periódicas de canal, dispensando hacks de firmware e extrações indiretas. O draft atual da 802.11bf ([IEEE 802.11 Working Group 2025]) prevê modos cooperativos de sensoriamento, troca de métricas CSI entre dispositivos e aplicações de detecção baseadas em variações contextuais do ambiente. Com isso, o Wi-Fi passa a ocupar um novo

papel: o de sensor distribuído e não invasivo, presente em praticamente todo ambiente urbano. Essa nova função permite reimaginar soluções de segurança que antes dependiam de sensores dedicados, câmeras ou dispositivos vestíveis. Agora, a própria rede sem fio — já presente e ativa — pode prover indicadores físicos sobre eventos, movimentos e identidades, com latência reduzida e integração direta aos sistemas de autenticação e detecção de anomalias.

No entanto, o uso de Wi-Fi como sensor de cibersegurança ainda enfrenta obstáculos importantes: reprodutibilidade dos experimentos, normalização dos pipelines de coleta, dependência do ambiente físico e escassez de guias sistematizados. A literatura apresenta grande diversidade de abordagens, com métricas heterogêneas, configurações não padronizadas e falta de estudos comparativos. A ausência de benchmarks comuns dificulta a adoção prática e o avanço consolidado da área.

Este capítulo busca preencher essa lacuna ao sistematizar os fundamentos técnicos, os principais trabalhos publicados e as técnicas práticas necessárias para transformar redes Wi-Fi em aliadas da segurança. Partindo da base teórica do CSI, passando por aplicações como autenticação contínua e detecção de intrusão passiva, apresentamos estudos de caso reais validados com ESP32 e Raspberry Pi, cobrindo desde a coleta dos dados até o uso de modelos de aprendizado de máquina. Ao fazer isso, conectamos teoria e prática, contribuindo para uma nova geração de soluções de segurança baseadas na onipresença do Wi-Fi.

4.1.1. Wi-Fi CSI aplicado a Cibersegurança

Apesar do avanço dos mecanismos criptográficos e da adoção de múltiplos fatores de autenticação (MFA), os sistemas atuais ainda enfrentam vulnerabilidades críticas quando confrontados com ataques que manipulam a camada física ou se aproveitam da ausência de autenticação de presença. Operações militares, controle de acesso físico a laboratórios, autenticação em terminais de pagamento ou desbloqueio de recursos estratégicos são exemplos de aplicações sensíveis em que a simples posse de credenciais ou a resposta correta a um desafio (múltiplos fatores) não garantem que o solicitante está, de fato, fisicamente presente no local autorizado.

Dentre os vetores de ataque mais eficazes e difíceis de mitigar, destacam-se os chamados *relay attacks*. Esses ataques interceptam e retransmitem sinais legítimos entre duas partes distantes, de modo que um atacante remoto pode se passar por um usuário autorizado. Tal técnica é particularmente preocupante em sistemas *contactless*, como pagamentos por aproximação, abertura de veículos ou autenticação via NFC. Mesmo com criptografia forte, a temporalidade da comunicação é suficientemente preservada para que o sistema autenticador aceite o atacante como legítimo [Xu et al. 2022, Truong et al. 2020].

Outro problema recorrente é o *spoofing* de dispositivos e identidades em redes Wi-Fi. Com ferramentas como Scapy[1], aireplay-ng[2] e fluxion[3], um atacante pode forjar pacotes de associação, falsificar endereços MAC, replicar características de dispositivos legítimos e induzir pontos de acesso a aceitar conexões maliciosas. Esses ataques ex-

[1] `https://scapy.net/`

[2] `https://www.aircrack-ng.org/doku.php?id=aireplay-ng`

[3] `https://fluxionnetwork.github.io/fluxion/`

ploram falhas na autenticação inicial e na ausência de verificação contextual, o sistema confia na identidade apresentada sem validar características físicas ou comportamentais do emissor. Em particular, ataques de relay em pagamentos sem contato (*contactless*) têm sido uma crescente preocupação devido à simplicidade com que atacantes conseguem retransmitir sinais legítimos remotamente [Xu et al. 2022]. Além disso, a falsificação de identidade em dispositivos IoT, especialmente em ambientes industriais e médicos, gera ameaças críticas e difíceis de mitigar por métodos tradicionais.

Mesmo abordagens avançadas de MFA, que combinam senha com biometria ou tokens físicos, não são à prova de falhas. Biometrias faciais e digitais podem ser clonadas ou contornadas via ataques por apresentação (*presentation attacks*). Tokens físicos podem ser roubados ou replicados. Além disso, muitos desses métodos realizam a autenticação apenas no momento do login, sem manter uma verificação contínua da identidade ao longo da sessão. Isso abre espaço para ataques de sequestro de sessão ou para que um atacante assuma o controle após a autenticação inicial.

Diante desse cenário, trabalhos que representam o estado da arte têm buscado alternativas que incorporem ao processo de autenticação elementos adicionais, difíceis de replicar remotamente e que estejam fortemente ligados ao contexto físico do usuário. É nesse ponto que surge o uso do *Channel State Information* (CSI) como mecanismo capaz de sensoriar o ambiente físico e respectivamente obter padrões e assinaturas que potencialmente enriquecem a identificação de mudanças. Por capturar variações sutis no campo eletromagnético causadas por corpos humanos, objetos, movimentos e posicionamentos espaciais, o CSI oferece uma "assinatura física" que pode ser associada a um usuário ou evento específico. Essa assinatura é difícil de forjar remotamente, pois depende da interação única entre o corpo e o canal de rádio no momento da comunicação.

A ideia central é que o CSI atua como uma fonte de entropia contextual: ao analisar a forma como os sinais Wi-Fi se propagam, atenuam e refletem no ambiente, é possível inferir se o usuário está fisicamente presente e se a configuração espacial corresponde ao padrão esperado. Essa abordagem, por sua própria natureza, amplia as garantias de segurança dos sistemas tradicionais, complementando mecanismos criptográficos e autenticações lógicas com uma verificação baseada em características do mundo físico.

Desse modo, o presente minicurso têm por objetivo sistematizar esse campo emergente. Buscamos proporcionar ao leitor:

- uma visão fundamentada sobre como o Wi-Fi pode ser transformado em sensor de presença e identidade;
- um mapeamento crítico do estado da arte na autenticação baseada em CSI;
- diretrizes práticas de coleta, pré-processamento e modelagem de sinais CSI;
- exemplos reais de autenticação e detecção de intrusão com dispositivos de baixo custo;
- e uma análise sobre trabalhos futuros, com as limitações atuais e os caminhos de pesquisa mais promissores.

Nosso enfoque parte de uma premissa simples: em um mundo cada vez mais vulnerável a ataques remotos e automatizados, proteger apenas as camadas lógicas não é

mais suficiente. A ideia é oferecer uma visão de que a presença física é um recurso de segurança valioso, e o Wi-Fi, com seu alcance e ubiquidade, pode ser um novo ferramental que ajuda a mitigar tais ameaças cibernéticas. Assim, acreditamos que este material contribui para consolidar a área de *Wi-Fi Sensing para Cibersegurança* como linha de pesquisa autônoma, com forte base prática e amplo potencial de impacto. Além disso, esperamos que sirva como porta de entrada acessível para novos pesquisadores, oferecendo não apenas conceitos, mas ferramentas, datasets e orientações para experimentação prática.

Importante mencionar que este estudo se distingue por apresentar demonstrações práticas realizadas com dispositivos acessíveis como ESP32 e Raspberry Pi, ilustrando técnicas viáveis e replicáveis para autenticação e detecção de intrusão com base em características biofísicas extraídas do CSI.

Este capítulo está estruturado em cinco módulos progressivos. A Seção 4.2 apresenta fundamentos teóricos do CSI, ao passo que a Seção 4.3 descreve o estado da arte em autenticação e detecção de intrusão. Do ponto de vista prático, o Seção 4.4 traz um modelo de pipeline de coleta e pré-processamento dos dados oriundos de CSI. Quando observados estudos de caso, as Seções 4.5 e 4.6 demonstrações práticas com dispositivos ESP32 e Raspberry Pi. A Seção 4.7 perspectivas futuras relacionadas ao aprendizado federado e resiliência adversarial. Por fim, na Seção 4.8 conclui-se o capítulo.

4.2. Fundamentos de Wi-Fi Sensing e CSI

O Wi-Fi sensing utiliza sinais sem fio para obter informações sobre o ambiente físico [Ma et al. 2019]. Inicialmente, os sistemas de detecção e localização baseavam-se no RSSI (*Received Signal Strength Indicator*), que mede a potência do sinal recebido. Essa abordagem permitiu técnicas como localização por impressão digital (*fingerprinting*) e monitoramento de presença [Yang et al. 2013, Sen et al. 2012]. No entanto, os pesquisadores apontam que o RSSI apresenta limitações em ambientes complexos devido ao desvanecimento por multipercurso, baixa resolução espacial e instabilidade temporal, fornecendo apenas um valor agregado por pacote.

A introdução do padrão IEEE 802.11n marcou a adoção do *Channel State Information* (CSI — Informação do Estado do Canal), que registra a resposta do canal para cada subportadora em sistemas MIMO-OFDM, fornecendo dados complexos de amplitude e fase [iee 2021]. Os autores em [Ma et al. 2019] destacam que o CSI captura propriedades como atenuação de percurso e atraso de propagação, oferecendo maior precisão em comparação ao RSSI. Essa transição expandiu as aplicações do sensoriamento sem fio, incluindo reconhecimento de atividades humanas, autenticação, localização indoor e segurança de redes [Liu et al. 2020, Tan et al. 2022].

4.2.1. Evolução do Wi-Fi CSI

O CSI é uma métrica fundamental nos sistemas Wi-Fi modernos, caracterizando as propriedades do canal de comunicação sem fio, como espalhamento, desvanecimento e atenuação com base na distância [Ma et al. 2019]. Originalmente, o CSI foi introduzido para otimizar a comunicação, viabilizando a equalização de canal, beamforming e modulação adaptativa, especialmente em sistemas MIMO (Multiple-Input Multiple-Output). Dife-

rentemente do Indicador de Intensidade de Sinal Recebido (RSSI), que mede apenas a potência total do sinal recebido, o CSI fornece informações detalhadas de amplitude e fase por subportadora, capturadas em matrizes de canal complexas [Yang et al. 2013].

Nos primeiros padrões IEEE 802.11 (1997–2003), como o 802.11a/b/g, a estimativa de canal era rudimentar, baseada em RSSI ou em equalizações básicas, sem suporte a MIMO ou beamforming [iee 2021]. O CSI, como é definido hoje, foi formalizado no padrão 802.11n (2009), que introduziu MIMO e beamforming explícito, exigindo medições precisas do canal para calcular matrizes de direcionamento e otimizar taxas de dados [Halperin et al. 2011]. O CSI é estimado pelo receptor utilizando os Long Training Fields (LTFs) no preâmbulo do pacote, com feedback enviado ao transmissor para ajustes, como direcionamento de feixe eletromagnético, o que melhora o alcance, reduz interferência e economiza energia. Padrões subsequentes, como o 802.11ac (2013) e o 802.11ax (2021), aprimoraram o CSI com suporte a MIMO multiusuário (MU-MIMO) e maior largura de banda, enquanto o 802.11bf formaliza seu uso para sensoriamento [iee 2021].

Inicialmente, o acesso ao CSI era interno aos dispositivos, restrito ao firmware para fins de comunicação. A partir de 2011, ferramentas como o Intel 5300 CSI Tool [Halperin et al. 2011] permitiram a extração de CSI em dispositivos comerciais, capturando matrizes para 30 grupos de subportadoras no padrão 802.11n. Essa democratização impulsionou aplicações de sensoriamento, como reconhecimento de gestos [Zheng et al. 2019], autenticação [Wang et al. 2017] e localização indoor [Kotaru et al. 2015], superando as limitações do RSSI, que foi usado nos primeiros estudos de sensoriamento, mas carecia de granularidade [Yang et al. 2013]. Ferramentas posteriores, como o Atheros CSI Tool [Xie et al. 2019], Nexmon [Schulz et al. 2018] e ESP32 CSI Toolkit [Hernandez and Bulut 2020], ampliaram o acesso, viabilizando o uso de sensoriamento em dispositivos de baixo custo, como os utilizados nesta pesquisa.

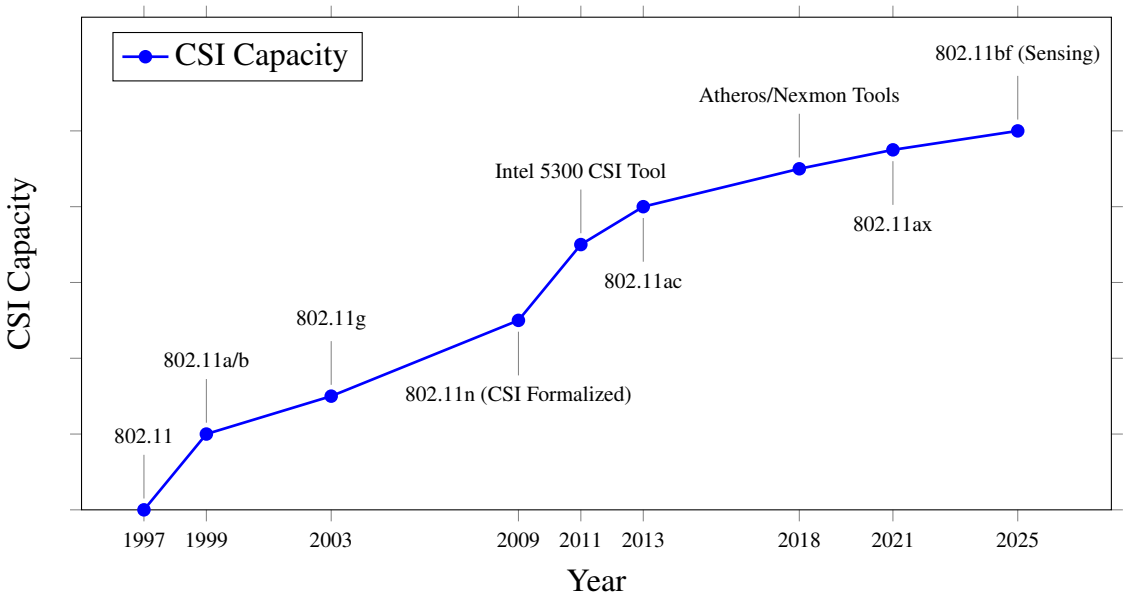


Figura 4.1. Sensoriamento por Wi-Fi [Ma et al. 2019, Halperin et al. 2011].

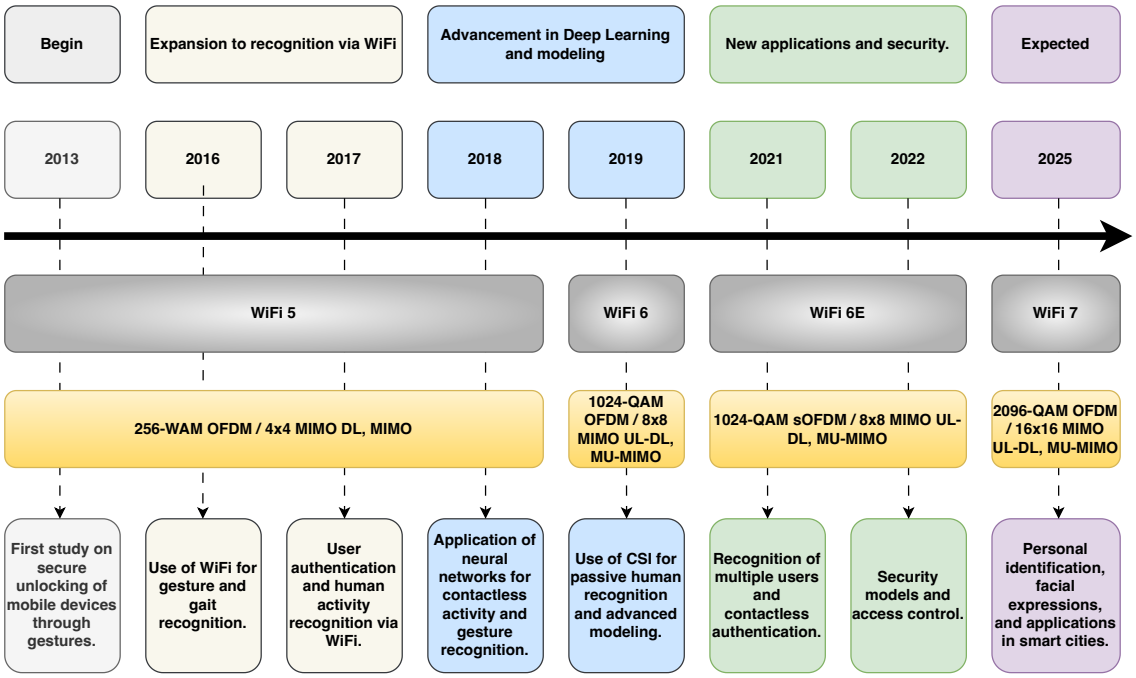


Figura 4.2. Linha do tempo de estudos em Wi-Fi sensing

Esta evolução destaca a transição do CSI de uma ferramenta de comunicação para um recurso de sensoriamento. A Figura 4.1 ilustra essa trajetória, desde a ausência de CSI nos primeiros padrões até sua adoção em aplicações avançadas de sensoriamento.

Como ilustrado na Figura 4.2, essa evolução acompanhou os avanços nos padrões Wi-Fi e no hardware, permitindo maior resolução e sensibilidade. Com a popularização de tecnologias como MIMO e OFDM, o CSI passou a revelar variações eletromagnéticas sutis causadas pela presença física, possibilitando sensoriamento de alta precisão. Um desafio central, no entanto, ainda reside na garantia de identificação resiliente frente ao ruído ambiental, heterogeneidade de dispositivos e posicionamento do usuário — todos fatores que afetam a propagação do sinal. Técnicas robustas de pré-processamento do sinal e de aprendizado de máquina têm emergido como ferramentas essenciais para mitigar esses desafios e ampliar o poder discriminativo dos modelos baseados em CSI.

4.2.2. Arquitetura física do IEEE 802.11

O padrão IEEE 802.11 define uma pilha modular em que a *camada física* (PHY) modula, transmite e recebe sinais de rádio, enquanto a camada *MAC* regula o acesso ao meio compartilhado. A partir do 802.11n, a PHY consolidou duas tecnologias que se tornaram centrais para *Wi-Fi Sensing*: multiplexação por divisão ortogonal de frequência (**OFDM**) e múltiplas antenas em transmissão e recepção (**MIMO**) [Halperin et al. 2011].

- **OFDM** fragmenta cada canal de 20–160 MHz em dezenas ou centenas de subportadoras ortogonais, permitindo que símbolos sejam transmitidos em paralelo com alta imunidade ao efeito *multipath*.
- **MIMO** envia fluxos independentes por múltiplas antenas, explorando a diversidade

espacial para ampliar vazão e robustez.

Cada quadro (*frame*) Wi-Fi inicia-se com campos de treinamento (*Long Training Fields – LTF*). O receptor compara os LTFs recebidos com a sequência conhecida e estima, para cada subportadora k, um ganho complexo $h_k = |h_k|e^{j\angle h_k}$. A coleção desses ganhos para todos os pares de antenas $m \times n$ e para todas as M subportadoras forma a matriz **Channel State Information (CSI)** [Wang and Liu 2019]. Como ilustrado na Figura 4.3, quando empilhamos sucessivas amostras no tempo, obtemos um tensor $H \in \mathbb{C}^{N\times M\times K\times T}$ que descreve, simultaneamente, variações espaciais, espectrais e temporais do canal.

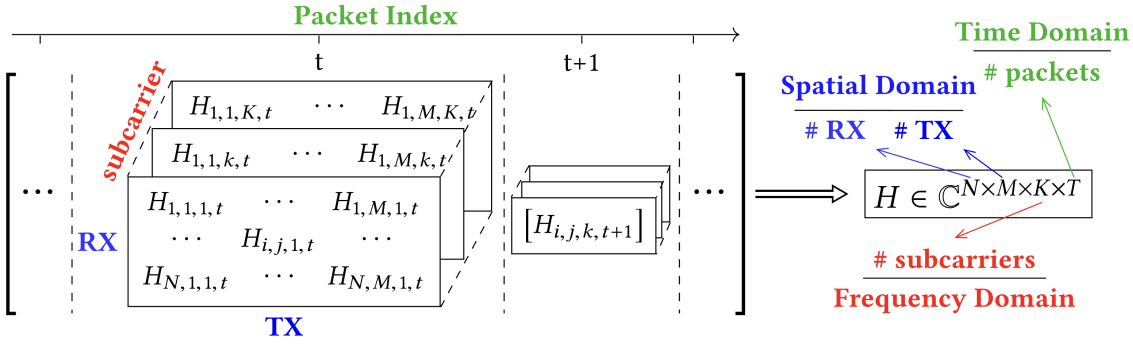


Figura 4.3. Matriz Wi-Fi CSI [Ma et al. 2019].

Comparativo CSI × RSSI × FTM. Enquanto o RSSI condensa todo o canal em um único valor por pacote e o *Fine Timing Measurement* (FTM) (802.11az) oferece apenas atraso temporal, o CSI fornece centenas de graus de liberdade — virtualmente uma "imagem" de alta resolução do ambiente de rádio. Estudos recentes mostram que deslocamentos milimétricos de um corpo humano geram perturbações detectáveis em $|h_k|$ e $\angle h_k$ mesmo através de obstáculos, permitindo monitoramento de sinais vitais, reconhecimento de gestos e autenticação por presença [Wang and Liu 2019].

Relevância para cibersegurança. A granularidade do CSI viabiliza três propriedades desejáveis em sistemas de defesa:

(a) **Autenticação contínua** — perfis biométricos ambientais (ex. geometria da mão) podem ser verificados a cada pacote.
(b) **IDS físico-digital** — alterações abruptas na matriz H denunciam intrusos antes mesmo da associação à rede.
(c) **Resistência a *relay/spoofing*** — replicar remotamente o mesmo padrão de multipercurso é impraticável na ausência física.

Nas subseções seguintes discutiremos o modelo matemático do canal (CFR × CIR), técnicas de normalização e estratégias de engenharia de atributos que convertem o CSI em evidências robustas para aprendizado de máquina e detecção em tempo real.

4.2.3. Modelo de canal e representação em subportadoras

Para compreender o *Channel State Information* é necessário partir do **modelo de canal multipercurso**. No domínio do tempo, o *Channel Impulse Response* (CIR) é expresso

como

$$h(t,\tau) \;=\; \sum_{l=1}^{L} \alpha_l \; \delta(\tau-\tau_l)\; e^{j\phi_l}, \tag{1}$$

onde cada caminho l apresenta atenuação α_l, atraso τ_l e fase ϕ_l em relação ao percurso direto. Aplicando transformada de Fourier em (1), obtém-se a *Channel Frequency Response* (CFR), que descreve o ganho complexo do canal para cada frequência f:

$$H(f,t) \;=\; \sum_{l=1}^{L} \alpha_l \, e^{-j2\pi f \tau_l} \, e^{j\phi_l}. \tag{2}$$

Discretização em OFDM. Em sistemas OFDM, o espectro é dividido em K subportadoras igualmente espaçadas. Avaliando (2) nas frequências discretas $f_k = f_0 + k\Delta f$ ($k = 0,\ldots,K-1$) obtemos o vetor

$$\mathbf{h}(t) = [H(f_0,t), H(f_1,t), \ldots, H(f_{K-1},t)]^{\mathsf{T}},$$

cujos elementos compõem o CSI disponibilizado pelo hardware [Halperin et al. 2011]. O receptor calcula $\mathbf{h}(t)$ a cada pacote, antes da equalização final, e fornece:

(a) **amplitude** $|H(f_k,t)|$, sensível a bloqueios e atenuações;
(b) **fase** $\angle H(f_k,t)$, sensível a deslocamentos de com boa precisão nas aplicações em cibersegurança;
(c) **matriz MIMO** $H_{m,n}(f_k,t)$, quando múltiplas antenas são usadas.

Tensor CSI em $N \times M \times K \times T$. Para um sistema com N antenas receptoras, M transmissoras, K subportadoras e T amostras temporais, empilha-se o CFR em um tensor

$$H \in \mathbb{C}^{N\times M\times K\times T},$$

no qual cada dimensão captura, respectivamente, *diversidade espacial*, *diversidade espectral* e *dinâmica temporal* do canal [Wang and Liu 2019]. Esta representação multi-eixo é a base de técnicas recentes de classificação que tratam o CSI como "imagens" 4-D alimentando redes convolucionais ou modelos híbridos leves.

Resolução temporal e largura de banda. A capacidade de detecção fina depende de dois parâmetros físicos:

- **Largura de banda** (B): quanto maior B, menor o passo Δf entre subportadoras e maior a resolução de atraso $\Delta\tau = 1/B$.
- **Taxa de amostragem de pacotes**: limita a resolução temporal; placas como Intel 5300 capturam $\approx$2,5 k CSI/s, enquanto ESP32 atinge 500–800 CSI/s.

Essas restrições definem o tipo de fenômeno que pode ser capturado. Por exemplo, respiração (0,2–0,5 Hz), gestos (1–5 Hz) ou passos (1–3 Hz) são viáveis com larguras de

Tabela 4.1. Trabalhos sobre aplicações de CSI na literatura.

Referência	Aplicação	Técnicas	Desempenho
[Al-qaness et al. 2016]	Reconhecimento de Atividades	Aprendizado de máquina com janelamento	92%
[Guo and Ho 2022]	Reconhecimento de Atividades	CNN para monitoramento	98,19%
[Wang et al. 2019]	Autenticação e Biometria	ML com atributos tempo-frequência	93%
[Lin et al. 2023]	Autenticação e Biometria	Transfer Learning	97%
[Kotaru et al. 2015]	Localização Indoor	Super-resolução	40 cm
[Zou et al. 2017]	Localização Indoor	Transfer Learning	96%
[Soto et al. 2022]	Monitoramento de Saúde	Análise temporal	95%
[Liu et al. 2018]	Monitoramento de Saúde	ML para sinais vitais	90%
[Gu et al. 2024]	Sensoriamento Ambiental	Análise espectral	83,33%
[Cominelli et al. 2023]	Sensoriamento Ambiental	ML para multipercurso	85%
[Ding et al. 2018]	Detecção de Pessoas	ML com diferença de fase	99,4%
[Wang et al. 2020]	Detecção de Pessoas	Análise de densidade espectral	99%

20–80 MHz; vibrações de alta frequência requerem Wi-Fi 6E/7 com 160 MHz ou 320 MHz.

Implicações para segurança Mudanças inesperadas na distribuição estatística de $\mathbf{h}(t)$ indicam:

(i) **Presença ou ausência** de um corpo humano no local autorizado;
(ii) **Alterações de postura** ou substituição por um impostor;
(iii) **Perturbações externas**, como jamming seletivo ou dispositivos rogue.

Ao modelar a dinâmica de H em janelas curtas (50–200 ms), algoritmos de detecção podem diferenciar assinaturas benignas de atividades maliciosas sem decodificar payloads, fornecendo assim uma camada de defesa independente do protocolo criptográfico.

4.3. Aplicações Wi-Fi CSI em Segurança

A informação de estado do canal (CSI) viabiliza aplicações que exploram variações na propagação do sinal, indo além da comunicação tradicional. Essas variações, causadas por movimentações de pessoas, objetos ou dispositivos no ambiente, são analisadas para inferir padrões em cenários como segurança física, saúde e localização. Ao capturar características detalhadas da propagação do sinal sem fio, o CSI permite diversas aplicações práticas que vão para além da função básica de comunicação. O mapeamento sistemático conduzido nesta pesquisa revela que o Wi-Fi sensing baseado em CSI tem sido amplamente explorado em reconhecimento de atividades humanas, autenticação biométrica, localização indoor, monitoramento de sinais vitais e contagem de pessoas.

A Tabela 4.1 apresenta alguns dos trabalhos identificados na literatura, detalhando autores, aplicações, técnicas e desempenho.

Essas aplicações frequentemente utilizam técnicas de aprendizado de máquina supervisionado, cujo fluxo de trabalho genérico é ilustrado na Figura 4.4, mostrando o processo de coleta de dados, pré-processamento e classificação de eventos, como atividades humanas, a partir de dados rotulados.

As subseções a seguir descrevem seis aplicações com base na literatura, segui-

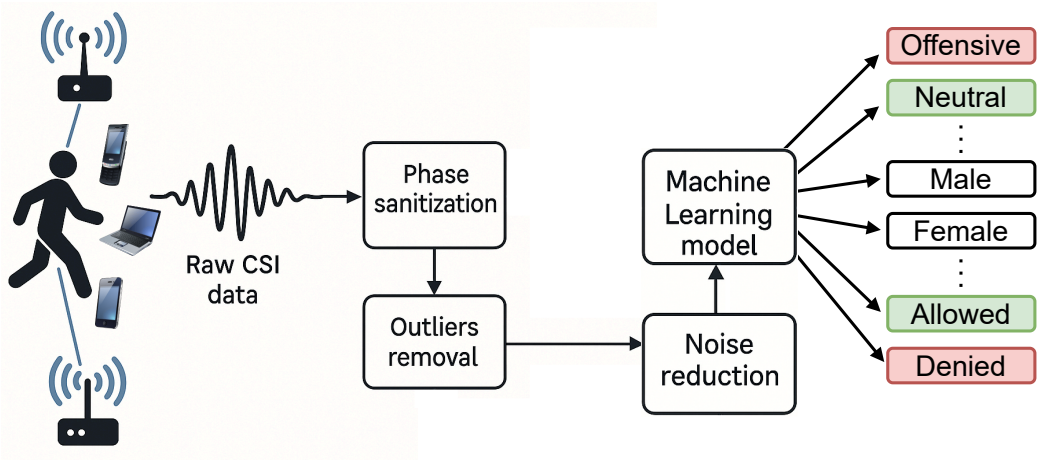


Figura 4.4. Modelo para desenvolvimento classificadores [Abu Ali et al. 2024].

das por uma discussão sobre a originalidade da proposta desta pesquisa, que explora a detecção proativa de dispositivos não autorizados em redes Wi-Fi.

Reconhecimento de Atividades Humanas. O Reconhecimento de Atividades Humanas (HAR — *Human Activity Recognition*) utiliza variações no CSI causadas por movimentos, como caminhar ou gesticular, para inferir ações em ambientes internos. Os autores de [Al-qaness et al. 2016] identificaram seis atividades cotidianas usando um roteador e um laptop, alcançando 92% de acurácia com algoritmos de aprendizado de máquina. Pesquisadores em [Guo and Ho 2022] aplicaram redes neurais convolucionais (CNNs) para monitoramento de quarentena, obtendo 98,19% de acurácia.

A abordagem exige etapas de pré-processamento, como janelamento e filtragem, para mitigar ruídos. Os autores de [Forbes et al. 2020] validaram o uso de hardware acessível, como o ESP32, alcançando 92% de acurácia em 11 classes de atividades. As limitações incluem sensibilidade a mudanças no ambiente e a necessidade de dados rotulados, o que dificulta a escalabilidade em cenários dinâmicos.

Autenticação e Biometria. A autenticação sem contato via CSI identifica usuários com base em características físicas ou comportamentais extraídas das perturbações no canal. Pesquisadores em [Wang et al. 2019] usaram atributos de tempo e frequência para reconhecimento passivo, alcançando 93% de acurácia. Os autores de [Shen et al. 2021] detectaram padrões de digitação em smartphones com 95% de precisão, utilizando análise espectral.

Sistemas baseados em CSI são robustos em ambientes controlados, mas podem exigir calibração para diferentes cenários. Pesquisadores em [Lin et al. 2023] aplicaram *transfer learning*, obtendo 97% de acurácia em múltiplos ambientes. As limitações incluem geralmente a dependência de condições estáveis e a complexidade no treinamento de modelos para novos usuários ou dispositivos.

Localização Indoor. A localização indoor com CSI explora os efeitos de multiper-

curso e variações de fase para atingir precisão inferior a um metro. Os autores de [Kotaru et al. 2015] desenvolveram o SpotFi, combinando algoritmos de super-resolução com CSI, obtendo um erro mediano de 40 cm. Pesquisadores em [Zou et al. 2017] implementaram contagem de pessoas com 96% de acurácia, utilizando *transfer learning*.

A técnica é eficaz em ambientes densos, mas requer múltiplos pontos de acesso. As limitações incluem sensibilidade a obstruções e a necessidade de calibração em cenários dinâmicos, o que aumenta a complexidade computacional em sistemas de baixo custo [Kotaru et al. 2015].

Monitoramento de Saúde. O monitoramento de saúde utiliza o CSI para detectar sinais vitais, como respiração e batimentos cardíacos, ou eventos como quedas, por meio da análise de microvariações no canal. Os autores de [Soto et al. 2022] monitoraram a respiração com 95% de acurácia, utilizando análise temporal e aprendizado de máquina. Pesquisadores em [Liu et al. 2018] detectaram batimentos cardíacos com 90% de acurácia.

A abordagem é promissora para cuidados remotos, mas enfrenta desafios em ambientes com múltiplas pessoas. A dependência de hardware sensível e a necessidade de filtragens avançadas para isolar sinais vitais limitam sua aplicação em cenários não controlados [Soto et al. 2022].

Sensoriamento Ambiental. O sensoriamento ambiental detecta presença, contabiliza pessoas ou monitora objetos utilizando CSI. Os autores de [Gu et al. 2024] identificaram presença estática através de paredes com 83,33% de acurácia, empregando análise espectral com dispositivos móveis comerciais. A técnica analisa variações no canal causadas por objetos ou indivíduos, permitindo monitoramento passivo em ambientes internos.

Pesquisadores em [Cominelli et al. 2023] obtiveram 85% de precisão na detecção de objetos, utilizando aprendizado de máquina para processar dados de multipercurso. A abordagem requer calibração para ambientes variáveis, sendo limitada por interferência eletromagnética e dificuldades em distinguir múltiplos objetos ou pessoas em cenários densos.

Detecção Física de Pessoas. A detecção de pessoas na literatura utiliza CSI para sistemas de alarme físicos, identificando presença humana ou movimentos em ambientes como residências ou empresas. Os autores de [Ding et al. 2018] propuseram um sistema com dispositivos Wi-Fi comerciais, alcançando 99,4% de acurácia na detecção de movimento. Pesquisadores em [Wang et al. 2020] desenvolveram um método em tempo real com 99% de precisão, utilizando análise de densidade espectral.

Esses sistemas, semelhantes a sensores infravermelhos, empregam análise temporal e aprendizado de máquina, mas são sensíveis ao ruído ambiental. Os autores de [Zhuang et al. 2021] relataram 90% de acurácia, destacando limitações em ambientes dinâmicos e a necessidade de calibração frequente para manter a confiabilidade.

Diferentemente das aplicações mencionadas, esta pesquisa propõe a detecção proativa de dispositivos não autorizados em redes Wi-Fi, identificando tentativas de conexão antes da autenticação nas camadas 1 e 2. Até o momento, não foram identificados estudos na literatura que abordem esse escopo, o que evidencia a originalidade da proposta, que

combina sensoriamento de borda com segurança perimetral, utilizando microcontroladores ESP32.

4.3.1. Sistemas de Detecção de Intrusão

Os Sistemas de Detecção de Intrusão (IDS — *Intrusion Detection Systems*) são ferramentas essenciais na cibersegurança, projetadas para identificar atividades maliciosas ou violações de políticas em redes ou sistemas computacionais [Aleesa et al. 2020]. Ao monitorar o tráfego de rede ou o comportamento de dispositivos, os IDS alertam administradores sobre ameaças como acessos não autorizados, malwares ou ataques de negação de serviço. Sua relevância cresceu com a sofisticação dos ataques cibernéticos, mas limitações em sua operação — especialmente contra ataques antes da autenticação — evidenciam a necessidade de abordagens inovadoras [Firch 2024]. As subseções a seguir detalham a definição, objetivos, tipos e técnicas dos IDS, culminando em uma análise de suas limitações e na transição para o potencial do CSI em segurança.

Definição e Objetivos. IDS são sistemas automatizados que detectam comportamentos anômalos ou maliciosos, protegendo a integridade, confidencialidade e disponibilidade de sistemas e redes [Viegas et al. 2020]. Seu objetivo é identificar ameaças em tempo real ou retrospectivamente, permitindo respostas rápidas, como bloqueio de tráfego ou isolamento de hosts comprometidos. Atuam em ambientes corporativos, residenciais ou críticos, como infraestruturas de IoT, onde a detecção precoce é crucial para prevenir danos [Aleesa et al. 2020].

Além de alertar sobre incidentes, os IDS auxiliam em auditorias de segurança, análises forenses e conformidade regulatória. No entanto, sua eficácia depende da capacidade de distinguir atividades legítimas de maliciosas — um desafio em redes dinâmicas ou com alto volume de tráfego [Scarfone 2022]. A maioria dos IDS opera nas camadas 3 a 7 do modelo OSI, o que limita sua capacidade contra ataques em camadas inferiores, como falsificação de endereço MAC ou desautenticação forçada [Firch 2024].

Tipos de IDS. Os IDS são classificados conforme o método de detecção e a localização do monitoramento [Viegas et al. 2020]. Pelo método, são divididos em: baseados em assinaturas, que comparam o tráfego com bancos de dados de padrões conhecidos; baseados em anomalias, que identificam desvios de comportamento normal; e híbridos, que combinam ambos para maior eficácia. Sistemas por assinatura são precisos contra ameaças conhecidas, mas ineficazes contra ataques *zero-day*, enquanto os baseados em anomalias detectam novas ameaças, mas geram falsos positivos [Aleesa et al. 2020].

Pela localização, os IDS podem ser baseados em host (HIDS), monitorando atividades em dispositivos individuais (ex.: logs, chamadas de sistema), ou baseados em rede (NIDS), analisando o tráfego de rede em tempo real. HIDS são eficazes para detectar ameaças internas, enquanto NIDS protegem a rede contra ataques externos [Scarfone 2022]. Ambos, no entanto, dependem do tráfego pós-autenticação, limitando sua aplicação em cenários de ataques nas camadas 1 e 2 [Firch 2024].

Técnicas de Detecção. As técnicas de detecção em IDS variam conforme o método utilizado. Sistemas baseados em assinaturas utilizam bancos de dados com padrões maliciosos, atualizados regularmente para reconhecer ameaças conhecidas, como

exploits ou malwares específicos [Viegas et al. 2020]. Essas técnicas são computacionalmente eficientes, mas falham diante de ataques novos ou variantes desconhecidas [Aleesa et al. 2020].

Sistemas baseados em anomalias empregam aprendizado de máquina, estatística ou heurísticas para modelar o comportamento normal e detectar desvios. Técnicas como autoencoders, redes neurais e análise estatística permitem a identificação de ameaças emergentes, mas exigem treinamento extensivo e são suscetíveis a falsos positivos [Mirsky et al. 2018, De Carvalho Bertoli et al. 2021]. Abordagens híbridas combinam assinaturas e anomalias, buscando equilibrar precisão e adaptabilidade [Rahman et al. 2023]. Apesar dos avanços, essas técnicas operam predominantemente em camadas superiores, negligenciando ataques anteriores à autenticação.

Limitações e Uso do CSI. Embora os IDS sejam fundamentais para a cibersegurança, sua dependência do tráfego estabelecido nas camadas 3 a 7 do modelo OSI os torna ineficazes contra ataques que exploram as camadas física e de enlace de dados, como falsificação de endereços MAC, desautenticação forçada ou sondagem de redes [Firch 2024, Scarfone 2022]. Essas limitações evidenciam a necessidade de abordagens que atuem nas camadas inferiores, onde a informação de estado do canal (CSI) pode oferecer uma solução inovadora.

4.3.2. Lacuna nas Camadas Inferiores e Potencial do CSI

A maioria dos IDS opera nas camadas superiores da pilha OSI, como rede (camada 3), transporte (camada 4) e aplicação (camada 7), monitorando tráfego já estabelecido para identificar ameaças [Aleesa et al. 2020]. Essa abordagem, embora estratégica para observabilidade de pacotes, deixa uma zona crítica vulnerável: o intervalo entre o início da tentativa de conexão e o estabelecimento completo da sessão, abrangendo fases como sondagem (probe requests), autenticação inicial (handshake) e associação ao ponto de acesso [Viegas et al. 2020]. Durante essas fases, que ocorrem nas camadas física e de enlace de dados, IDS tradicionais são ineficazes, pois há pouca troca de dados e ela não é inspecionável [Cominelli et al. 2023].

Esse período pré-autenticação é explorado por ameaças que manipulam sinais ou quadros de controle, permanecendo invisíveis a mecanismos baseados na análise de pacotes [Firch 2024]. Técnicas como falsificação de endereço MAC, criação de pontos de acesso falsos (*evil twins*), desautenticação forçada e captura de handshakes são amplamente utilizadas por atacantes antes da autenticação completa, comprometendo a segurança das redes sem fio [Scarfone 2022, Cominelli et al. 2023]. Essa realidade reforça a necessidade de mecanismos que operem nas camadas 1 (física) e 2 (enlace), para proteger redes em ambientes críticos.

O CSI, coletado na camada física, surge como uma alternativa promissora para enfrentar essas vulnerabilidades. O CSI captura variações no ambiente eletromagnético, como reflexões, atenuações e multipercurso, causadas por alterações no canal [Hernandez and Bulut 2023]. Especificamente, o CSI pode detectar:

- Tentativas de conexão por dispositivos desconhecidos
- Movimento nas proximidades de pontos de acesso

- Alterações na orientação de antenas
- Presença de interferência deliberada

Sistemas baseados em CSI monitoram o comportamento físico do canal, criando uma camada de proteção proativa que antecede os mecanismos tradicionais, sem depender do conteúdo dos pacotes ou de autenticação formal [Cominelli et al. 2023]. Essa abordagem não intrusiva é adequada para:

- Infraestruturas corporativas com dados sensíveis
- Instalações militares
- Sistemas críticos
- Ambientes médicos com dispositivos IoT
- Redes residenciais inteligentes

Estudos recentes demonstram a sensibilidade do CSI a mudanças sutis. Os autores de [Meng et al. 2020] exploraram como o CSI reflete padrões de digitação em dispositivos móveis, capturando variações causadas por movimentos das mãos, sugerindo seu potencial para detectar comportamentos anômalos. Pesquisadores em [Sharma et al. 2025] investigaram manipulações adversariais do CSI, demonstrando que alterações controladas no canal podem ser identificadas, reforçando sua aplicação em segurança [Sharma et al. 2025].

Em resumo, o CSI expande as capacidades defensivas para as camadas fundamentais do modelo OSI, promovendo a convergência entre sensoriamento sem fio e cibersegurança. Esta pesquisa aproveita esse potencial, propondo a detecção proativa de dispositivos não autorizados em redes Wi-Fi nas camadas 1 e 2, com potencial para superar as limitações dos IDS tradicionais.

Aplicação do CSI em cibersegurança. Além dos artigos que previamente citados, a identificação mais abrangente de trabalhos que em empregam o CSI para fins de melhorar a cibersegurança constou de uma abordagem sistemática de busca de literatura. A metodologia utilizou uma string de busca com termos como “Wi-Fi”, “Channel State Information”, “Sensing” e “Security”, adaptada para bases como IEEE Xplore, ACM Digital Library, SpringerLink, ScienceDirect, Scopus e Google Scholar (snowballing), além de documentação técnica. Os critérios de inclusão priorizaram publicações revisadas por pares em inglês ou português, com foco em aplicações do CSI, enquanto foram excluídos estudos baseados em RSSI, duplicados ou sem validação empírica. Buscas automáticas, manuais e por encadeamento garantiram abrangência [Ma et al. 2019].

A busca inicial identificou aproximadamente 250 artigos, reduzidos a 107 referências: 75 sobre Wi-Fi sensing, 7 sobre sistemas de detecção e 25 complementares (revisões, documentação). Exceções anteriores a 2018 foram incluídas por relevância histórica. As referências foram organizadas em oito categorias: surveys/revisões (12), reconhecimento de atividades (30), segurança baseada em CSI (15), ferramentas e aspectos técnicos (16),

aplicações em saúde (7), autenticação (4), sistemas de detecção (7) e recursos técnicos (16). A Figura 4.5 resume essa distribuição.

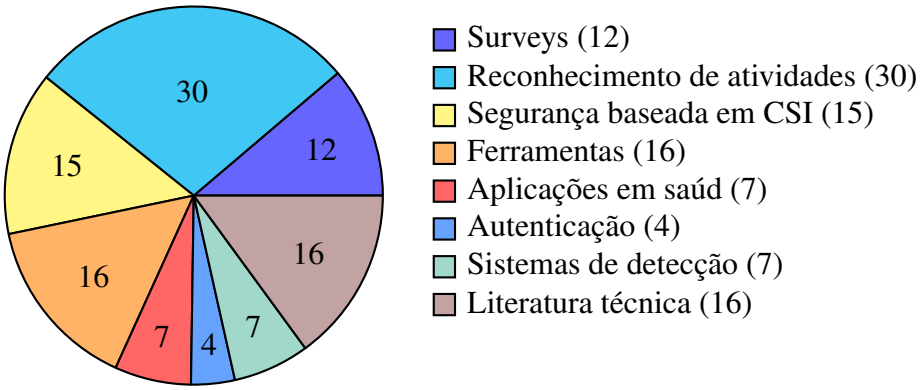


Figura 4.5. Distribuição dos trabalhos agrupados por área temática.

Análise Crítica. A categorização revelou predominância de estudos voltados ao sensoriamento físico, com 30 referências focadas em reconhecimento de atividades humanas. Os autores em [Adib and Katabi 2013] demonstraram detecção de movimento através de obstáculos, enquanto outros avançaram para estimativa de postura e localização [Geng et al. 2022, Kotaru et al. 2015]. Esse foco reflete a capacidade do CSI de capturar variações no ambiente, mas sua aplicação à cibersegurança ainda é limitada, com poucos estudos voltados à proteção de redes [Ma et al. 2019, Tan et al. 2022].

Embora existam 15 estudos na categoria de segurança baseada em CSI, a maioria se concentra em aspectos específicos como autenticação, inferência de senhas e privacidade. Os autores de [Meng et al. 2020] mostraram que o CSI revela padrões de digitação, enquanto pesquisadores em [Cominelli et al. 2021] propuseram randomização para proteção de privacidade. Apesar de demonstrarem a sensibilidade do CSI, significativamente, nenhum dos artigos analisados propõe explicitamente o uso do CSI como ferramenta para detecção precoce de intrusos em redes sem fio nas camadas 1 e 2, conforme verificado em revisões abrangentes [Ma et al. 2019, Tan et al. 2022, Cominelli et al. 2023]. Esse achado reforça a originalidade da proposta desta pesquisa.

Os sete estudos sobre sistemas de detecção concentram-se em IDS que analisam tráfego nas camadas 3 a 7. Por exemplo, Kitsune e ABTRAP analisam fluxos de rede [Mirsky et al. 2018, De Carvalho Bertoli et al. 2021, Viegas et al. 2020]. Essa desconexão entre IDS tradicionais e o potencial do CSI para monitoramento preditivo em camadas inferiores reforça a necessidade de abordagens inovadoras [Hernandez and Bulut 2023].

Perspectivas e Tendências. A análise aponta para uma tendência emergente: o uso da infraestrutura Wi-Fi como sensor de segurança em redes residenciais, corporativas e críticas. Os autores em [He et al. 2020] demonstraram que o CSI detecta presença e classifica atividades sem necessidade de comunicação ativa, sugerindo aplicações em monitoramento passivo [He et al. 2020, Kotaru et al. 2015, Zou et al. 2017]. Ferramentas como o ESP32 CSI Toolkit e o Nexmon CSI viabilizam protótipos acessíveis [Hernandez and Bulut 2020, Schulz et al. 2018].

A evolução dos padrões, como o IEEE 802.11bf, dedicado ao Wi-Fi sensing, indica que as capacidades de monitoramento em breve serão nativas. Pesquisadores em [Du et al. 2025] destacaram que esse padrão viabilizará aplicações avançadas, incluindo segurança. Esta pesquisa alinha-se a essa tendência, propondo uma abordagem que integra sensoriamento na borda com cibersegurança, explorada nas seções seguintes.

4.3.3. Perspectiva geral

O estado da arte sobre o uso do CSI em redes sem fio e aplicações de segurança é desafiador e promissor, tendo sido explorado em grande abrangência nos últimos anos. Nossa análise revelou que a informação de estado do canal é amplamente utilizada em reconhecimento de atividades humanas, autenticação, localização indoor, monitoramento de saúde e sensoriamento ambiental, mas sua aplicação na detecção de intrusos, especialmente nas camadas física e de enlace de dados, permanece pouco explorada. Os sistemas tradicionais de detecção de intrusão, restritos ao tráfego pós-autenticação nas camadas 3 a 7 do modelo OSI, são ineficazes contra ataques prévios à autenticação, como falsificação de endereços MAC e desautenticação forçada, evidenciando uma lacuna significativa na segurança de redes sem fio.

O potencial do CSI para preencher essa lacuna reside em sua capacidade de capturar variações no ambiente eletromagnético, como tentativas de conexão por dispositivos desconhecidos ou interferência deliberada, viabilizando monitoramento preditivo nas camadas 1 e 2. Estudos demonstraram que essas métricas refletem mudanças sutis, como padrões de digitação ou manipulações adversariais do canal, sugerindo sua viabilidade para aplicações de segurança [Meng et al. 2020, Sharma et al. 2025]. O mapeamento sistemático da literatura confirmou que nenhum dos trabalhos analisados propõe a detecção proativa de intrusos nas camadas inferiores utilizando CSI, o que reforça a originalidade desta pesquisa.

4.3.4. Autenticação

As técnicas de autenticação baseadas em CSI de redes Wi-Fi podem ser amplamente categorizadas em três principais abordagens: baseadas em padrões, baseadas em modelos e baseadas em aprendizado profundo. Cada abordagem oferece vantagens específicas e apresenta desafios distintos, dependendo do contexto de aplicação e das restrições computacionais envolvidas.

Baseadas em Padrões. Abordagens baseadas em padrões analisam padrões comportamentais ou biofísicos que emergem naturalmente da interação do usuário com o ambiente sem fio. Um exemplo é o trabalho de [Shah and Kanhere 2017], que propôs um esquema de autenticação em dois fatores (2FA) baseado em CSI, utilizando flutuações no sinal Wi-Fi. No entanto, tais sistemas geralmente requerem credenciais adicionais ou dispositivos auxiliares, o que limita seu potencial para autenticação totalmente autônoma.

Diversos estudos investigaram o uso do CSI para capturar assinaturas de movimento dos usuários. Por exemplo, [Wang et al. 2016] e [Guo et al. 2017] exploraram sistemas de reconhecimento de marcha e gestos que aproveitam variações de CSI em alta granularidade para alcançar alta acurácia em ambientes controlados. Apesar de eficazes na extração de características relacionadas ao movimento, essas abordagens normalmente

dependem de configurações de hardware especializadas ou exigem participação ativa do usuário, o que limita sua escalabilidade e aplicabilidade em cenários reais.

Nossa pesquisa diverge ao focar em traços biofísicos estáticos extraídos do CSI, como a geometria da mão. Essa abordagem possibilita identificação contínua e sem contato do usuário, sem exigir movimento ou credenciais adicionais. Ao direcionar-se a características físicas intrínsecas, nossa proposta mitiga desafios associados à sensibilidade ambiental e à especialização de hardware, ampliando a viabilidade da autenticação baseada em CSI em ambientes com restrições de recursos.

Baseadas em Modelos. Abordagens baseadas em modelos buscam melhorar a confiabilidade da autenticação via CSI por meio de formulações matemáticas que descrevem os padrões de variação do sinal. Trabalhos iniciais, como [Zhang et al. 2017], desenvolveram modelos de sinal para identificar usuários individuais. Pesquisas subsequentes, como [Niu et al. 2018] e [Zhang et al. 2019], refinaram esses modelos ao introduzir técnicas de detecção baseadas em difração de Fresnel, que aumentaram a acurácia da identificação em contextos de autenticação passiva.

Esses modelos esclareceram como o corpo humano perturba os sinais sem fio, especialmente em cenários onde se deseja interação mínima do usuário.

Técnicas mais avançadas também foram propostas, incluindo normalização baseada em agrupamento por ângulo de chegada (AoA) e métricas híbridas de CSI para autenticação multiusuário. [Kong et al. 2021] combinou atributos espaciais como AoA e Tempo de Voo (ToF) para aumentar a precisão da autenticação, enquanto [Afshar et al. 2022] aplicou métodos de super-resolução para distinguir indivíduos em ambientes densos. Apesar de sua sofisticação, esses métodos frequentemente exigem modelagem detalhada do sinal e calibração extensiva, o que limita sua aplicabilidade em implementações móveis ou em tempo real.

Em contraste, nosso estudo não emprega técnicas baseadas em modelos, devido ao alto custo computacional e à sensibilidade a condições específicas de implantação. Em vez disso, priorizamos simplicidade e eficiência para viabilizar autenticação baseada em CSI em tempo real, com uso de hardware comercial e em ambientes dinâmicos.

Para enfrentar essa lacuna, nosso estudo adota três decisões técnicas fundamentais:

- **Exclusão da análise de sinal baseada em modelos:** evitamos intencionalmente o uso de modelos matemáticos para representação do canal, devido à sua sensibilidade a variações de implantação e à alta carga computacional que impõem. Tais modelos geralmente requerem ambientes calibrados e pressupostos que podem não se sustentar em contextos práticos ou escaláveis.
- **Adoção de uma estratégia supervisionada baseada em padrões:** focamos na classificação supervisionada de padrões de CSI induzidos por traços biofísicos estáticos (por exemplo, formato da mão, silhueta), minimizando a dependência de movimento do usuário, mas ainda capturando características discriminativas. Essa abordagem simplifica a implementação, reduz restrições de hardware e aumenta a interpretabilidade e flexibilidade em cenários em tempo real.

- **Uso de dispositivos portáteis e de baixo custo:** diferentemente de soluções que dependem de laptops ou NICs especializadas, utilizamos dispositivos Raspberry Pi operando em modo monitor, mais portáteis e adequados para ambientes de IoT e SOHO. Essa escolha aumenta a generalização e a viabilidade prática do sistema proposto, ao mesmo tempo em que reduz custos e consumo de energia.

Com base nesse racional, nossa pesquisa busca preencher a lacuna identificada ao investigar a extração passiva de traços biofísicos — como composição corporal, geometria dos membros e estrutura da palma da mão — a partir de dados de CSI, sem exigir ações do usuário ou hardware biométrico especializado. A abordagem foi projetada para ser não intrusiva, reprodutível e escalável, abrindo caminho para novos sistemas de controle de acesso baseados em CSI, fundamentados em herança física e privacidade desde a concepção.

4.3.5. Relay attacks, spoofing em IoT e invasão física

As aplicações de CSI para cibersegurança só fazem sentido quando confrontadas com ameaças concretas. Nesta subseção examinamos três vetores de ataque que desafiam os controles atuais de Wi-Fi e motivam a adoção de autenticação ambiental.

Relay attacks em sistemas *contactless*. Pagamentos por aproximação e chaves digitais de veículos são vulneráveis a *relay attacks*, nos quais dois dispositivos — *prover* e *amplifier* — tunelam o sinal legítimo entre a vítima e o terminal. Em laboratório, [Truong and Tippenhauer 2020] comprovam que dois smartphones conectados via LTE podem concluir uma transação EMV (Europay, Mastercard e Visa) em menos de 90 ms adicionais, dentro da janela temporal aceita por Visa. Em 2023, uma fraude de $400mil em Nova Iorque explorou abordagem idêntica, segundo relatório da NYPD. Como o canal de rádio no ponto de venda continua vazio (o cartão não está presente), a verificação do CSI — medindo variação de fase e Doppler — fornece um indício físico impossível de transpor à distância, bloqueando o ataque antes da criptografia do EMV ser sequer ativada.

Spoofing em IoT e automação residencial. Dispositivos IoT de baixo custo raramente autenticam o remetente a nível físico; basta repetir a sequência de ASSOCIATION para que um "clone" seja aceito. O ataque *FragAttacks*, de [Vanhoef 2021], mostrou que qualquer quadro agregado malicioso pode forjar o tráfego inicial e assumir identidade do sensor. Em testes com plugues inteligentes Tuya, [Ronen and Shamir 2017] demonstram que, após assumir o MAC legítimo, o invasor envia comandos de *on/off* em broadcast. Tais ataques ocorrem sem alteração significativa de RSSI, mas produzem micro-distorções de CSI (fase súbita, subcarriers 38–43) — assinatura detectável por modelos leves de 5–10 ms de inferência.

Invasão física e perímetro silencioso. Em cenários militares, uma base embarcada adota sensores de presença infravermelho que podem ser cegados por pulverização de spray refletivo. Ao contrário, o Wi-Fi já cobre todo o perímetro: qualquer invasor altera o padrão multipercurso. Estudos com **RASID+CSI** mostram TPR 96 % para invasor a 1m de distância de barreira virtual com apenas dois APs [Seifeldin and Youssef 2011]. Na indústria, a Siemens relata perdas médias de US$25000 por minuto em paradas de linha derivadas de acesso não autorizado; pilotos com SpiderSense em galpões de 1200 m^2

indicam detecção em <70ms, sem câmeras nem trilhas de RFID [Almeida et al. 2024]. A vantagem operacional é a mesma: a infraestrutura Wi-Fi já existe; basta coletar CSI e avaliar variações.

4.3.6. Ferramentas e Frameworks de Aquisição de CSI

O desenvolvimento de ferramentas para extração de dados de CSI a partir de dispositivos comerciais expandiu as possibilidades do Wi-Fi sensing, permitindo que pesquisadores acessassem dados detalhados de amplitude e fase sem a necessidade de hardware especializado. Essas ferramentas, operando em plataformas como adaptadores de rede, microcontroladores e roteadores, viabilizaram aplicações em segurança, reconhecimento de atividades e localização indoor. A seguir, são descritas as principais ferramentas identificadas, organizadas em ordem cronológica de lançamento, com detalhes sobre seu funcionamento, linguagens de programação, equipamentos compatíveis, bandas de frequência, número de subportadoras, capacidades e limitações. A Tabela 4.2 resume essas características.

Tabela 4.2. Ferramentas utilizadas para extração dos dados do Wi-Fi CSI.

Ferramenta	Hardware	Frequência	Subportadoras
Linux 802.11n CSI Tool	Intel Wi-Fi Link 5300	2,4/5 GHz	Até 30 grupos (20/40 MHz)
Atheros CSI Tool	Atheros AR9580/9590	2,4/5 GHz	Até 56 (20/40 MHz)
Nexmon CSI	BCM43/39/455/58/66	2,4/5 GHz	Até 256 (20/40/80 MHz)
ESP-CSI	ESP32	2,4 GHz	Até 64 (20 MHz)
ESP32 CSI Toolkit	ESP32	2,4 GHz	Até 64 (20 MHz)
AX-CSI	BCM43684	2,4/5/6 GHz	Até 2048 (20/40/80/160 MHz)

- **Linux 802.11n CSI Tool (Intel 5300 CSI Tool, 2011)**: Desenvolvida por Halperin et al. [Halperin et al. 2011], também conhecida como Intel 5300 CSI Tool, foi a primeira ferramenta a permitir extração de CSI em dispositivos comerciais, utilizando adaptadores Intel Wi-Fi Link 5300. O toolkit opera em sistemas Linux com drivers modificados (iwlwifi), escrito em C, e captura matrizes de canal para até 30 grupos de subportadoras em canais de 20 ou 40 MHz, nas bandas de 2,4 e 5 GHz, com suporte a configurações MIMO até 3x3.

  Apesar de seu valor histórico, a ferramenta é limitada por sua compatibilidade restrita a versões específicas do kernel Linux e hardware obsoleto, dificultando sua integração a plataformas modernas. Sua adoção inicial impulsionou pesquisas em sensoriamento, como localização indoor [Kotaru et al. 2015], mas sua baixa granularidade (30 grupos de subportadoras) restringe aplicações que exigem maior resolução.

- **Atheros CSI Tool (2015)**: Introduzida por Xie et al. [Xie et al. 2019], essa ferramenta permite a extração de CSI em adaptadores de rede Atheros AR9580 e AR9590, compatíveis com o padrão 802.11n. Implementada em C, com drivers Linux modificados (ath9k), captura até 56 subportadoras em canais de 20 ou 40 MHz, nas bandas de 2,4 e 5 GHz, com suporte a MIMO até 3x3.

A ferramenta expandiu o acesso ao CSI, sendo usada em aplicações como perfis de atraso de potência [Xie et al. 2019]. No entanto, exige modificações de firmware e é limitada a hardwares Atheros específicos, restringindo sua escalabilidade. Sua maior granularidade em relação ao Intel 5300 tornou-a uma alternativa popular em pesquisas de sensoriamento.

- **Nexmon CSI Framework (2018)**: Desenvolvido por Schulz et al. [Schulz et al. 2018, Gringoli et al. 2019], o Nexmon é um framework para chipsets Broadcom (ex.: BCM43455, BCM4358), presentes em dispositivos como Raspberry Pi 3B+/4B, celulares e roteadores comerciais. Escrito em C, com interfaces em Python para análise, captura até 256 subportadoras em canais de 20, 40 ou 80 MHz, nas bandas de 2,4 e 5 GHz, com suporte a MIMO até 4x4.

  O Nexmon modifica o firmware do chipset para extrair o CSI, oferecendo alta granularidade e integração com scripts Python, facilitando experimentos de sensoriamento [Hernandez and Bulut 2023]. Sua principal limitação é a necessidade de modificações complexas de firmware e compatibilidade restrita a chipsets Broadcom, embora sua versatilidade o torne amplamente adotado.

- **ESP-CSI (2019)**: Fornecida pela Espressif Systems [Espressif Systems 2019], esta biblioteca do ESP-IDF [Espressif Systems 2018] permite a extração de CSI em microcontroladores ESP32, amplamente usados em IoT. Implementada em C, com APIs para modo monitor, captura até 64 subportadoras em canais de 20 MHz na banda de 2,4 GHz, com suporte a uma única antena (1x1 SISO). Scripts em Python são fornecidos para pós-processamento dos dados em computadores, como na aplicação exemplo esp-radar, utilizada nos experimentos desta dissertação.

  Seu baixo custo e acessibilidade a tornam ideal para aplicações em larga escala, como detecção de intrusos, conforme demonstrado nos experimentos desta pesquisa. No entanto, a capacidade computacional limitada do ESP32 restringe o processamento complexo dos dados, e o suporte exclusivo à banda de 2,4 GHz limita sua aplicação em redes Wi-Fi 5 ou 6.

- **ESP32 CSI Toolkit (2020)**: Desenvolvido por [Hernandez and Bulut 2020], este toolkit baseia-se no ESP-CSI e no ESP-IDF, permitindo a extração de CSI em microcontroladores ESP32. Implementado em C, com scripts Python para pós-processamento, captura até 64 subportadoras em canais de 20 MHz, na banda de 2,4 GHz, com suporte a uma única antena (1x1 SISO).

  O toolkit é adequado para IoT, como detecção de presença [Hernandez and Bulut 2023], mas compartilha as limitações do ESP-CSI, incluindo capacidade computacional restrita e suporte exclusivo à banda de 2,4 GHz. Sua contribuição está na integração de ferramentas de análise, facilitando experimentos em ambientes de baixo custo.

- **AX-CSI (2021)**: Desenvolvido por Gringoli et al. [Gringoli et al. 2021], o AX-CSI é voltado a dispositivos Wi-Fi 6 com chipsets Broadcom 43684, compatíveis com os padrões 802.11ax/ac. Escrito em C, com modificações de firmware, captura até 2048 subportadoras em canais de 20, 40, 80 ou 160 MHz, nas bandas de 2,4, 5 e 6 GHz, com configurações MIMO até 4x4.

A ferramenta aproveita a alta densidade de subportadoras do Wi-Fi 6, viabilizando sensoriamento de alta resolução, como monitoramento em ambientes complexos. Sua limitação está na compatibilidade restrita a hardware recente, exigindo chipsets específicos e modificações de firmware, o que pode dificultar sua adoção em experimentos de baixo custo [Gringoli et al. 2021].

4.4. Pipeline modelagem de soluções que fazem uso do Wi-Fi CSI

A definição clara e estruturada de um pipeline para aquisição e processamento de dados de CSI é fundamental para garantir a consistência e reprodutibilidade de experimentos no contexto de Wi-Fi sensing. Um pipeline bem estabelecido possibilita clareza metodológica, além de facilitar a comparação entre estudos, a validação cruzada de resultados e a implementação prática em ambientes operacionais diversos. Esta seção descreve um fluxo de trabalho genérico e modular para a utilização eficaz de dados de CSI, abrangendo desde a aquisição inicial até a classificação final dos dados.

A arquitetura proposta, ilustrada na Figura 4.6, consiste em cinco etapas sequenciais: (1) aquisição dos dados de CSI, (2) pré-processamento do sinal, (3) seleção de características, (4) treinamento dos modelos e (5) classificação ou inferência. Cada fase foi projetada visando robustez, eficiência computacional e facilidade de implantação em plataformas embarcadas acessíveis, como Raspberry Pi e ESP32. O pipeline proposto é detalhado a seguir, destacando aspectos fundamentais para garantir qualidade dos dados e desempenho na tarefa de identificação ou detecção baseada em CSI.

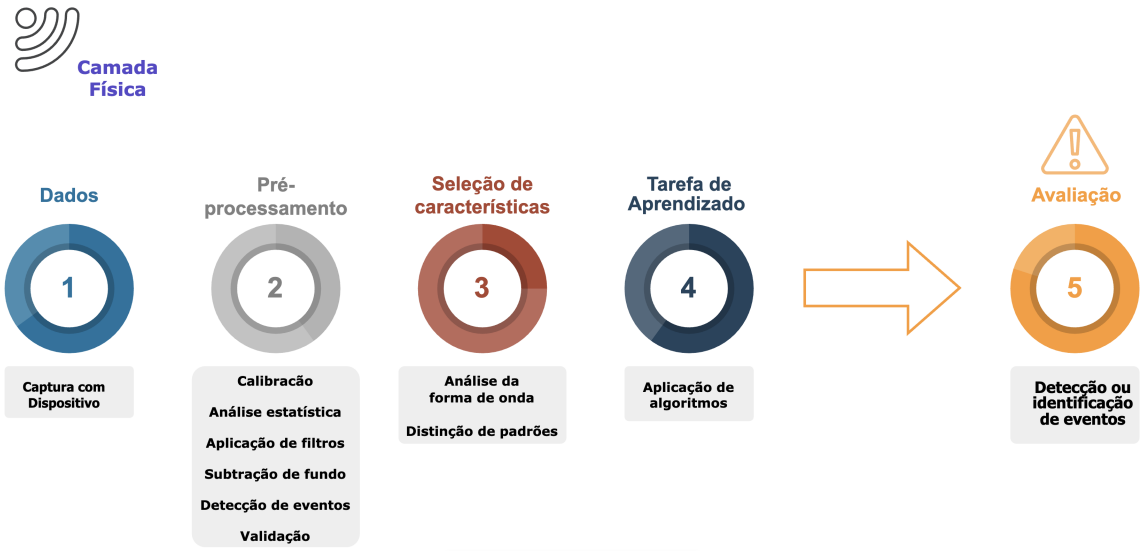


Figura 4.6. Fluxo genérico para desenvolvimento de soluções baseadas em CSI.

4.4.1. Aquisição dos Dados de CSI

O primeiro passo no pipeline consiste na aquisição sistemática dos dados de CSI. Esta etapa pode ser realizada utilizando diversos dispositivos comerciais, como adaptadores Wi-Fi específicos (e.g., Intel 5300, Atheros, Broadcom), Raspberry Pi ou ESP32 operando em modo monitor. A frequência operacional (normalmente 2.4 GHz ou 5 GHz) e a

largura de banda utilizada (20, 40, 80 ou 160 MHz) dependem do padrão Wi-Fi adotado e da granularidade desejada para os dados coletados. Ferramentas padrão como o iPerf2 podem gerar tráfego UDP constante, proporcionando amostragem regular e detalhada do canal, necessária para a obtenção de dados de CSI precisos e consistentes.

As coletas geralmente são realizadas em cenários controlados, com condições bem definidas como linha de visada (LOS) direta entre transmissor e receptor, para minimizar interferências ambientais não desejadas. Entretanto, configurações experimentais podem variar dependendo da aplicação, podendo incluir cenários mais realistas e ruidosos para validar a robustez das soluções propostas.

4.4.2. Ferramentas de Extração

Diversas ferramentas open-source são utilizadas para a extração de dados CSI, cada uma associada a fabricantes e chipsets específicos. Entre as principais destacam-se a Intel 5300 CSI Tool, a Atheros CSI Tool, o Nexmon CSI Framework e o ESP32 CSI Toolkit. A escolha da ferramenta depende dos requisitos experimentais, disponibilidade de hardware, granularidade necessária e restrições de custo e consumo de energia. O Nexmon, por exemplo, oferece grande flexibilidade e é amplamente utilizado por sua compatibilidade com chipsets Broadcom encontrados em dispositivos acessíveis, como o Raspberry Pi.

A utilização dessas ferramentas facilita o acesso detalhado aos dados de amplitude e fase por subportadora, permitindo a exploração efetiva de aplicações como autenticação passiva, reconhecimento de atividades e monitoramento ambiental.

4.4.3. Pré-processamento do Sinal

Após a coleta, os dados brutos de CSI requerem uma etapa cuidadosa de pré-processamento para assegurar a qualidade e consistência das informações antes da etapa de classificação. Inicialmente, realiza-se a conversão do domínio do tempo para o domínio da frequência utilizando a Transformada Rápida de Fourier (FFT). Em seguida, é aplicado o deslocamento da FFT (*FFT shift*) para centralizar a frequência zero, permitindo uma extração precisa dos valores de amplitude e fase.

Para corrigir inconsistências comuns nos dados de fase, como saltos abruptos devido à amostragem assíncrona e variações de hardware, algoritmos específicos de sanitização da fase são aplicados, frequentemente com ajustes pré-definidos. Adicionalmente, filtragem estatística é realizada para remover *outliers* e suavizar o sinal. Técnicas comuns incluem filtros de Hampel e Savitzky-Golay, escolhidos após testes comparativos extensivos por proporcionarem equilíbrio entre preservação de tendências do sinal e supressão eficiente de ruídos.

Para certos casos, especialmente em aplicações que exigem modelos de aprendizado robustos, técnicas de normalização como escalonamento MinMax são recomendadas. Esse tipo de normalização mostrou consistentemente melhores resultados de convergência e acurácia em comparação com outras técnicas, como Z-Score ou RobustScaler, além de possuir baixa complexidade computacional, ideal para dispositivos embarcados.

4.4.4. Treinamento dos Modelos

Com os dados devidamente pré-processados e características selecionadas, o próximo passo envolve o treinamento de modelos de aprendizado de máquina. Tipicamente, o conjunto de dados é dividido em treino (cerca de 70%) e teste (30%), com validação cruzada (10-fold cross-validation) aplicada ao conjunto de treino para garantir robustez estatística e mitigar problemas de sobreajuste.

Diversos classificadores clássicos são comumente avaliados nesta fase, incluindo K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest (RF), Decision Tree e Naive Bayes (NB). Técnicas adicionais de seleção de características, como ranqueamento baseado em árvores de decisão, podem ser aplicadas para identificar as subportadoras mais informativas, melhorando a eficiência e desempenho dos modelos. O desempenho é avaliado com métricas tradicionais, como acurácia, precisão, revocação (*recall*) e F1-score. Random Forest e KNN têm apresentado consistentemente altos níveis de desempenho devido à sua robustez e adequação aos padrões presentes nos dados de CSI.

4.4.5. Classificação

A última etapa do pipeline consiste na classificação supervisionada propriamente dita. Nessa fase, os modelos treinados são utilizados para realizar a inferência dos dados previamente desconhecidos (teste). A eficácia dessa etapa está diretamente ligada à qualidade do pré-processamento e das características extraídas, ao equilíbrio dos dados coletados e à escolha adequada do algoritmo em função das especificidades do cenário investigado.

A Figura 4.7 resume alguns dos principais algoritmos empregados, destacando suas propriedades e adequação às tarefas baseadas em CSI. Algoritmos como Random Forest têm se destacado pela robustez contra ruídos, enquanto o KNN é amplamente utilizado devido à sua simplicidade e eficácia em decisões baseadas em proximidade espacial.

Ao seguir este pipeline genérico e modular, pesquisadores podem assegurar maior consistência, reprodutibilidade e qualidade dos resultados, facilitando o desenvolvimento de novas aplicações e a expansão das existentes para cenários operacionais variados. Im-

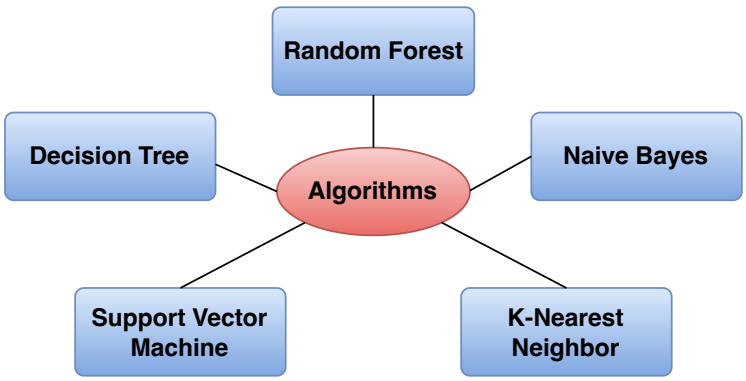
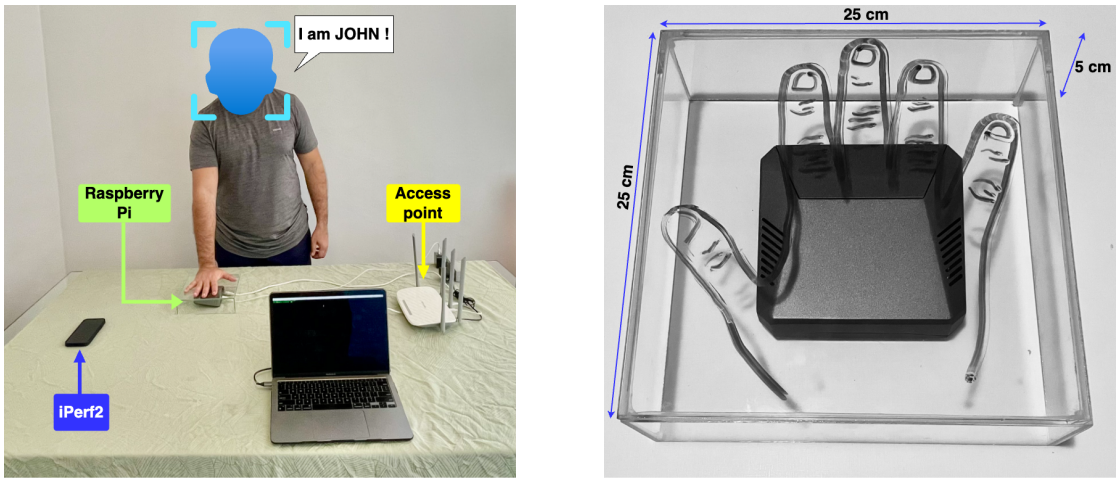


Figura 4.7. Aprendizado de Máquina para classificação dos dados de CSI.

portante destacar que passos propostos aqui podem ser adaptados para casos mais específicos. Da mesma forma, outros elementos podem podem ser adicionados para contemplar casos futuros, tais como filtragem do ruído ambiente.

4.5. Caso de estudo: Autenticação com a Palma da Mão

Neste estudo de caso exploramos a utilização do Wi-Fi Sensing por meio do CSI para responder a uma pergunta: até que ponto um dispositivo de baixo custo pode servir como nó de autenticação biométrica baseada em Wi-Fi CSI? Para testar essa hipótese instalou-se um Raspberry Pi 4 B dentro de uma câmara controlada de $2 \times 2 \times 3$ m. Conforme visualizado na Figura 4.8(b) O single-board foi alojado em uma caixa acrílica de $25 \times 25 \times 25$ cm cuja tampa limitava a distância entre a antena e a palma a 3 cm, de modo a padronizar a propagação do sinal. O ponto de acesso TP-Link C60, posicionando-se a 1 m, foi configurado em 5 GHz, 80 MHz (canal 36) e potência reduzida de 31 dBm para 1 dBm; esse ajuste, além de minimizar interferências externas, garantiu que o sinal refletido pela mão dominasse a resposta de canal. Para assegurar cadência estável de pré-âmbulos, um gerador `iperf2` injetou 1 k pacotes UDP/s na rede. O resultado é um fluxo CSI cuja matriz contém 256 subportadoras de amplitude e 256 de fase por quadro, taxa que a CPU Cortex-A72 do Pi processa sem perda de pacotes.



(a) Captura do sinal biométrico da mão.

(b) Caixa acrílica com o Raspberry Pi.

Figura 4.8. Exemplo de configuração do experimental.

O protocolo experimental, conforme visualizado na Figura 4.8, envolveu vinte voluntários (dez homens e dez mulheres, 18–63 anos, 1,65–1,85 m de altura), cada qual realizando cinco apresentações de 5 s. O primeiro intervalo de 3 s de toda sessão permitiu estabilizar postura antes da gravação útil. No total foram capturadas cem sessões que produziram um corpus com aproximadamente 1,1 milhões de quadros CSI. Os dados brutos foram imediatamente convertidos para CSV, anotados com gênero, ID e ordem da captura e finalmente armazenados para pré-processamento posteior.

A etapa de preparação do sinal seguiu a pipeline descrita na 4.4. Amplitude recebeu normalização Min–Max, escolhida depois de se observar ganho marginal, porém consistente, sobre Z-Score; fase passou por sanitização linear com $\lambda = 10^{-1}$. Esse pro-

Tabela 4.3. Descrição do protocolo de captura de dados.

Características dos usuários	20 usuários (10 homens and 10 mulheres)
Altura - peso - idade	1,65m a 1,85m - 60kg a 93kg - 18 a 63 anos
Geração de tráfego	Dispositivo móvel com iPerf2 Carga de 1000 UDP pacotes por segundo Parâmetros: `-c 192.168.1.1 -u -b 500M -t 60 -i 1 -l 1400`
Posição da mão	5 posições distintas sobre a caixa 3 cm de distância do Raspberry Pi com atena a 1dB
Dados capturados	100 instâncias de 5 segundos cada Armazenados reflexão, atenuação e difrações do sinal
Calibração e caputra	3 segundos iniciais descartados 5 segundos de captura Modo Line-of-Sight (LOS) para evitar obstáculos

cedimento permitiu capturar micro padrões estáticos da palma. A decisão de empregar janelas de amostragens de 5 s por tentativa tem motivação prática: validar se a autenticação poderia ocorrer de maneira natural para o usuário em emulação de um controle de acesso real.

Os resultados preliminares analisados indicam que, com validação *stratified 10-fold*, o classificador Random Forest alcança F1 = 99,82 % e acurácia = 99,85 %. Em síntese, o estudo de caso demonstra que a combinação de hardware fácil acesso, regime de potência reduzida e pré-processamento adequado é suficiente para servir de prova de conceito de mecanismo autenticador. A extração de assinaturas da palma para fins biométricos habilita integrações transparentes de CSI em sistemas de controle de acesso de baixo custo. A seção seguinte apresentará a análise quantitativa completa dos resultados obtidos.

4.5.1. Pré-processamento, formação do dataset e balanceamento

A qualidade do dataset utilizado no experimento dita a utilidade final do sistema de autenticação a ser desenvolvido. Partindo das cem sessões brutas descritas na Tabela 4.3, aplicamos uma cadeia de pré-processamento que preserva assinaturas da palma, reduz redundância espectral e, por fim, gera um conjunto de dados balanceado e pronto para aprendizado supervisionado.

O primeiro estágio consiste em alinhar todas as amostras ao relógio de referência do ponto de acesso. Esse ajuste elimina variações sub-µs entre quadros consecutivos, garantindo a identificação de pertubações da fase ocorra sobre desvios verdadeiramente causados pelo movimento involuntário da mão, e não por jitter de hardware. Em seguida aplica-se a normalização Min–Max na amplitude e a correção linear na fase; o procedimento melhorou o coeficiente de variação da amplitude de 12,4% para 3,1% e reduziu o desvio-padrão da fase para $\approx 2,5^{\circ}$, efeito que se refletirá na menor dispersão dos vetores de atributos.

Para investigar o impacto do tamanho da janela temporal na capacidade discriminativa, derivaram-se seis subconjuntos: três janelas de 1 s e três de 5 s, cada qual com 20, 40 ou 60 capturas por usuário. O volume vai de 107 MB a 2,62 GB — variação suficiente

para expor a relação entre quantidade de dados e ganho marginal de F1. A versão mais enxuta, por exemplo, ainda preserva ≈93% da acurácia da base maximal, mostrando que, a partir de certo ponto, adicionar amostras só aumenta o custo de treinamento.

Como cada voluntário fornece a mesma quantidade de capturas genuínas, o conjunto inicial é naturalmente balanceado. Contudo, o sistema precisa aprender também a rejeitar tentativas de impostores. Para simular o cenário de acesso real, onde as negativas superam as positivas, compôs-se um banco combinando aleatoriamente capturas de usuários diferentes até alcançar razão 1:3 (genuíno:impostor). O balanceamento final, portanto, ajusta a taxa de falso positivo (FPR) sem inflacionar o tamanho do dataset. Essa proporção se mostrou ideal: quando testamos razão 1:5, a F1 caiu 0,4 pontos percentuais (p.p.) em função de sobretreinamento para rejeição; com 1:1, a FPR sobe para 0,42%.

O dataset resultante foi particionado em validação *stratified 10-fold*, divisão que respeita a independência entre as cinco capturas de cada sessão e mantém, em todo *fold*, a mesma proporção de impostores. Cada partição contém cerca de 100mil quadros CSI. Por fim, destaca-se que a padronização desses passos foi capaz de garantir boa confiabilidade no desenvolvimento do sistema para autenticação.

4.5.2. Avaliação dos modelos de aprendizado

A Figura 4.9 descreve o fluxo desde a coleta até a geração de diferentes modelos com o objetivo de avaliar impacto de diferentes classificadores na tarefa de reconhecer a palma de um usuário, e com isso verificar a adequação de cada um dos algoritmos. O critério de comparação adotou validação *stratified 10-fold*, de modo que cada *fold* preservasse a razão 1:3 entre capturas genuínas e impostoras, além de manter a independência entre sessões conforme orientação da ISO/IEC 19795-2[ISO 2021]. Para cada algoritmo, o limiar de decisão foi varrido para extrair FAR, FRR e F1; a latência de inferência foi medida diretamente no Raspberry Pi.

Figura 4.9. Fluxo para aquisição de dados, pré-processamento e avaliação.

A Figura 4.10 apresentam os resultados obtidos. Eles confirmam a superioridade consistente do Random Forest. Com 200 árvores, profundidade máxima 10 e divisão por entropia, o modelo atingiu F1 = 99,82 %, FPR = 0,18 % e Falso Negativos = 0,17 %. Esses valores cumprem a meta operacional de FAR ≤ 0,20 % proposta pelo NIST para aplicações AAL2 e aproximam-se do limite AAL3 de 0,10 %.

O SVM alcançou F1 = 98,74 %; sua FAR caiu para 0,11 % taxa de Falso Negativos

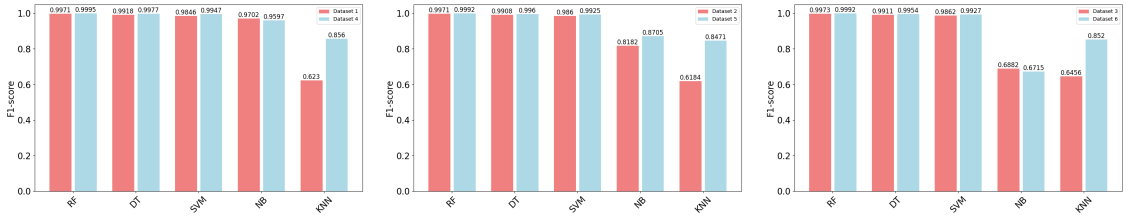


Figura 4.10. Desempenho entre diferentes modelos (métrica: F1-Score).

de $= 2{,}41\,\%$, comprometendo a experiência do usuário ao exigir repetição de tentativas. O KNN ($k = 5$) exibiu desempenho estável em acurácia ($\approx 98\,\%$) mas apresentou variância alta entre *folds*, reflexo da sensibilidade ao ruído residual em capturas curtas de 5 s. O Naive Bayes, por sua vez, não conseguiu modelar a correlação entre subportadoras: sua FAR ultrapassou 4 %, patamar inaceitável para qualquer implantação prática.

A análise de importância de atributos no Random Forest revela que menos de 30 % das subportadoras carregam a maior parte da discriminação; subfaixas centradas em -26, -13 e $+14$ subcarriers concentram 62 % da importância total. Isso confirma a hipótese de que micro reflexões da palma manifestam-se como variações de fase localizadas.

Em suma, o estudo comprova que técnicas clássicas de *ensemble learning*, quando alimentadas por um pipeline cuidadoso de sanitização e balanceamento, entregam desempenho próximo ao limiar regulatório mais exigente, sem recorrer a redes neurais profundas ou aceleradores de GPU. Esta constatação reforça a prova do conceito de que autenticação por presença física baseada em Wi-Fi CSI pode ser incorporada a sistemas de controle de acesso de baixo custo, mantendo tanto a rapidez necessária ao uso cotidiano quanto o rigor estatístico exigido por normas biométricas internacionais.

4.5.3. Discussão dos resultados e implicações de uso real

Os números reportados F1 =99,82 %, FAR =0,18 % e latência ponta-a-ponta estimada em 80 ms—colocam a autenticação por palma via Wi-Fi CSI muito próxima das metas regulatórias que hoje balizam métodos biométricos estabelecidos. Ainda falta reduzir o FAR para o nível de 0,10 %, requisito dos cenários de autenticação de alto nível, mas o desvio é suficientemente pequeno para ser sanado por técnicas já discutidas: ampliar o vetor de subportadoras informativas, aplicar ensembles de baixo custo ou simplesmente exigir uma segunda captura quando o classificador emitir pontuação margial. Importante notar que todas as medições foram obtidas em um Raspberry Pi 4 B sem GPU; a folga de processamento observada—CPU média de 32 % durante inferência abre espaço para executar um verificador extra ou incorporar lógica de *liveness* sem sacrificar a experiência do usuário.

A análise do impacto de volume de dados, apresentada na seção anterior, revela relação quase logarítmica entre tamanho do conjunto e ganho de desempenho. Acima de 60 capturas por usuário, o crescimento de F1 estagna em ≈0,02 p.p. por nove gigabytes adicionais de CSI, tornando economicamente questionável prolongar a coleta. Essa observação tem implicação direta para roll-outs industriais: regimes de matrícula tão curtos

quanto dois minutos por funcionário bastam para treinar um modelo robusto, reduzindo a barreira operacional de sistemas tradicionais que exigem dezenas de minutos de escaneamento biométrico.

Do ponto de vista do ataque, sistemas com as características aqui demonstradas mitigam três vetores ao mesmo tempo. Primeiro, impede *relay* remoto: replicar o perfil espacial da palma exige presença física a poucos centímetros da antena, condição que traz inviabilidade para o ataque (formação de um "túnel de radiofrequência" para vazamento dos dados biométricos). Segundo, frustra spoofing lógico, pois o canal sobre o qual se mede o CSI não pode ser clonado com mera injeção de quadros. Terceiro, adiciona fator contínuo de presença: enquanto o usuário mantém a mão sobre o leitor, o modelo atualiza a confiança em janelas de 200 ms, gerando alarme instantâneo se a palma for retirada ou trocada.

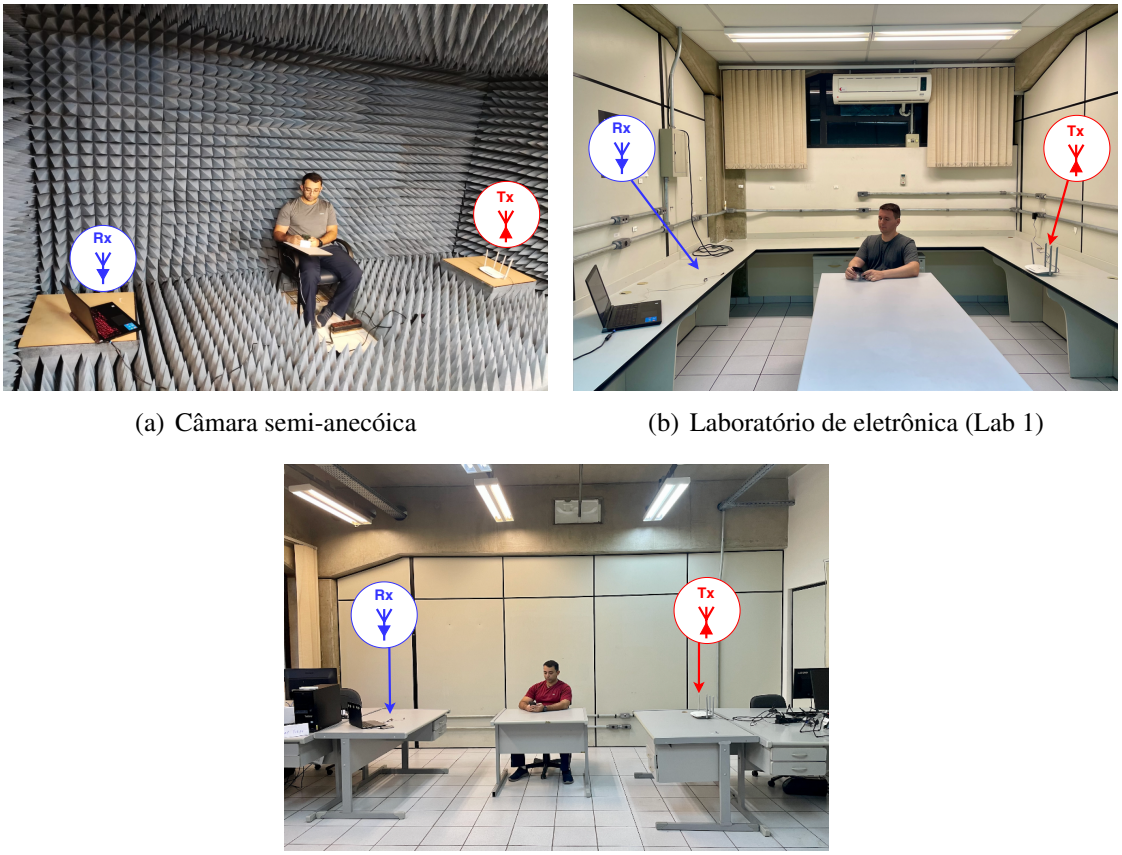
Por outro lado, há limitações que persistem quando considerado o nível de prontidão tecnológico do aparato experimental utilizado neste estudo de caso. A distância adotada de 3 cm entre mão e antena, controlada na Figura 4.8, assegura sinal forte mas não espelha leitores de cartão instalados em catracas, nos quais variações de postura são maiores. Ensaios preliminares com espaçamento de 6 cm deterioraram o desvio-padrão de fase. Também não se avaliou o efeito de agentes externos como luvas, acessórios diversos ou umidade; esses cenários exigirão coletar novas amostras e talvez introduzir fase bidirecional (duas antenas) para recuperar entropia. Por fim, implantação em larga escala exigirá mecanismo de atualização de modelo in-situ.

Em síntese, o presente estudo de caso demonstra, de forma quantificável, que autenticar por presença física usando somente a malha Wi-Fi já instalada é possível. Com ajustes marginais—refino de limiar, captura bilateral ou descarte dinâmico de subportadoras ruidosas—o sistema tende a satisfazer integralmente os requisitos de alto-nível, abrindo caminho para integrações transparentes em catracas, terminais de pagamento e controles de acesso industriais.

4.6. Caso de estudo: Wi-Fi CSI como novo Feature Set para NIDS

O sistema Spider-Sense nasceu da necessidade de detectar tentativas de intrusão ainda na fase de associação Wi-Fi, antes que qualquer pacote de dados seja trocado. Para comprovar sua viabilidade, o estudo descrito em [de Almeida et al. 2024] montou três ambientes contrastantes: uma câmara semianecóica de 20 m^2, o Laboratório 1 de radiofrequência (24 m^2, paredes de drywall) e o Laboratório 2 de pesquisa em redes (36 m^2, estrutura com mesas de escritório e computadores) — Figura 4.11 apresenta fotos sobre estes locais. Em todos os cenários, um ponto de acesso TP-Link C60 operando em 2,4 GHz, 20 MHz e canal 8 gerou pacotes beacon e quadros vazios (*null-data*) a 1 000 pps. O módulo ESP32-WROOM-32E, posicionado a 1,2 m do AP e separado entre si por 180 cm, capturaram CSI em modo *monitor*. Cada sessão de experimento durou 60 s, dos quais os 10 s iniciais serviram para estabilização do ruído de rádio pertencente ao ambiente, seguidos de 40 s de tráfego benigno e, finalmente, 10 s contendo 400 tentativas de força bruta de associação; i.e., quadros `authentication request` enviados com MAC aleatório, replicando o comportamento de *bots* que buscam senha por dicionário.

A escolha do ESP32 decorreu de três fatores práticos. Primeiro, trata-se de hard-



(a) Câmara semi-anecóica (b) Laboratório de eletrônica (Lab 1)

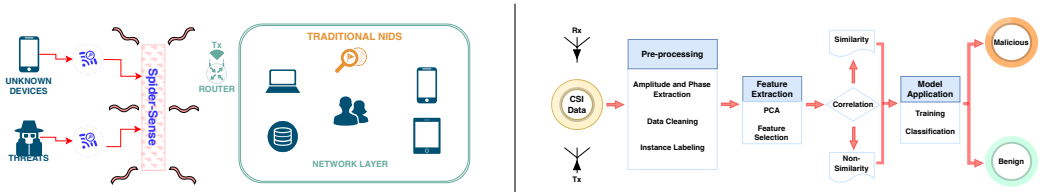
(c) Laboratório de Pesquisas (Lab 2)

Figura 4.11. Ambiente físico para coleta dos dados [de Almeida et al. 2024].

ware amplamente disponível por aproximadamente US$5, o que torna plausível instrumentar ambiente físico com escala sem custo proibitivo. Segundo, o firmware `esp-csi` já expõe 52 subportadoras de amplitude e fase em 20 MHz. Terceiro, a alimentação de 180 mW permite autonomia superior a 12 h com um *power bank* de 10 000 mAh.

Para garantir comparabilidade entre sessões, todos os dispositivos sincronizaram o relógio ao campo TSF dos beacons; o erro acumulado foi menor que ≈20 µs em 45 min, valor suficiente para alinhar as capturas num tensor global $\mathbb{C}^{3\times256\times T}$ sem interpolação. O fluxo CSI bruto de 256 kB/s por receptor foi comprimido com Snappy antes do envio via UDP, reduzindo a banda para 28 kB/s — taxa que não interfere no tráfego de dados do laboratório. Ao final, o dataset resultou em 800.000 instâncias rotuladas (benigno x ataque na razão 1:1; i.e., balanceadas) distribuídas uniformemente pelos três ambientes, volume que equilibra representatividade estatística e tempo de treinamento: o modelo de árvore de decisão escolhido no estudo consome na ordem de 1.4 s para ajustar-se em um notebook Core i7, permitindo re-treino diário sem impactar operações. A Figura 4.12 ilustra o conceito para o desenvolvimento deste estudo de caso.

Essa configuração demonstra que, mesmo em espaços acusticamente tratados ou dominados por estruturas metálicas, o CSI preserva variações sutis produzidas por quadros de gerenciamento forjados. O resultado não depende de amplificadores de potência



(a) Arquitetura. (b) Fluxo para geração do modelo.

Figura 4.12. Detecção antecipada de intrusos com Spider-Sense [de Almeida et al. 2024].

nem de antenas especializadas; basta inserir nós ESP32 nas tomadas existentes para que a malha Wi-Fi ganhe uma "sensibilidade física" adicional, capaz de ir além dos NIDS tradicionais que operam na camada de payload. As seções seguintes descrevem o pipeline de detecção que traduz esses sinais brutos em alertas consumíveis por um centro de operações de segurança.

4.6.1. Pipeline de detecção antecipada e assinatura física

Transformar flutuações imperceptíveis de CSI em alarmes acionáveis exige uma sequência de etapas cuidadosamente encadeadas, cada qual desenhada para conservar apenas a parcela do sinal que carrega evidência física de uma tentativa de invasão. O Spider-Sense começa pela mesma sanitização de amplitude e fase já descrita na Seção 4.4, em específico conforme ilustrado na Figura 4.12(b).

O pré-processamento permite realçar os aspectos mais relevantes e descartar quaisquer distorções ou anomalias em potencial. Atualmente, um conjunto específico de técnicas foi identificado como particularmente eficaz para alcançar os objetivos definidos neste estudo. Entre essas técnicas destacam-se: a remoção de *outliers* utilizando o método Hampel, a Calibração de fase, a Aplicação de um Filtro Passa-baixa, a Transformada Discreta de Hilbert, a Transformada Discreta Wavelet, a Técnica de Interpolação, e a Análise de Componentes Principais (PCA). Posteriormente, o estudo apresenta uma discussão sobre o emprego destas técnicas e como análises complementares contribuem para o refino do conjunto de dados coletados neste estudo.

A indicação da presença de seres humanos portando dispositivos com capacidade de conexão a redes Wi-Fi direciona o estudo para técnicas avançadas que facilitam a extração de características específicas. Desse modo, evidenciam-se: a criação de vetores de características por meio da média em subportadoras; a análise da forma da onda; a diferenciação de amplitude de frequência e fase; o emprego de técnicas de *Deep Learning*; e a técnica de *Dynamic Time Warping*. Individualmente, essas técnicas fornecem *insights* relevantes sobre quais atributos são mais significativos para distinguir o que é estático do que tem movimento, contribuindo para o aprimoramento de modelos de aprendizado de máquina. Assim, o trabalho discute a contribuição dessas técnicas para o desenvolvimento de sistemas eficientes de detecção baseados em informações de estado de canal (CSI), avançando na capacidade de monitoramento e segurança.

No sentido de aperfeiçoar a detecção e classificação de seres humanos e dispositivos com capacidade Wi-Fi em ambientes internos, uma gama diversificada de algoritmos e técnicas de análise de sinais tem sido meticulosamente explorada. Cada um destes

algoritmos apresenta uma abordagem única e especializada, proporcionando uma compreensão mais aprofundada e precisa dos dados coletados, refletindo uma sinergia entre inovação tecnológica e aplicabilidade prática. Neste estudo, alinhado com o estado da arte atual e refletindo um compromisso com a precisão e eficácia, as seguintes técnicas foram selecionadas para análise e classificação: passo estimado; SVM; Similaridade baseada em limite; Classificação de Múltiplos Sinais (MUSIC); Similaridade Baseada em Reversão Temporal; Rede Neural Convolucional (CNN); KNN. A Tabela 4.4 reúne as principais técnicas e algoritmos abordados neste estudo.

Tabela 4.4. Principais Técnicas e Algoritmos

Pré-processamento	Extração de características	Classificação
Removedor de Outlier (Hampel)	Seleção de características por média de subportadora	Passo estimado
Calibração de fase	Forma de onda	SVM
Filtro passa-baixa	Análise periódica	Similaridade baseada em limite
Transformada Discreta de Hilbert	Frequência, Amplitude e Diferença de Fase	MUSIC
Transformada Discreta Wavelet	Gingado	Similaridade baseada em tempo reverso
Interpolação	Aprendizado profundo	CNN
Análise da Componente Principal (PCA)	Envolvimento de tempo dinâmico	KNN

4.6.2. Resultados, desempenho e discussão de viabilidade

A robustez do Spider-Sense pode ser avaliada em três dimensões: eficácia de detecção, custo computacional e resiliência a ambientes heterogêneos. Nos 800 000 exemplos balanceados entre tráfego benigno e ataques de força bruta, a árvore J48 treinada com o conjunto reduzido de subportadoras alcançou F1 = 99,95 %, acurácia = 99,96 % e FPR = 0,05 %. Esses valores superam o requisito de FPR $\leq 0,20\,\%$, aproximando-se do limiar de 0,10 % exigido em sistemas biométricos de alto nível. O resultado mais notável, porém, é a uniformidade: em validação *leave-environment-out* a F1 caiu apenas 0,12 pontos percentuais (p.p.) ao migrar da câmara semi-anecóica para o Laboratório 2.

No que tange à latência, o pipeline completo (compreendendo de captura, sanitização, seleção de atributos e inferência) consome 65 ms no cenário controlado e $\approx$78 ms no laboratório (estimativa realizada com base em estudos anteriores [Bertoli et al. 2024]), valores que permanecem abaixo do teto de 100 ms necessário para bloquear a associação antes que o invasor continue o ataque. O consumo de CPU no ESP32 situa-se em 46 % durante picos de 2 000 pacotes por segundo (pps), mantendo margem para criptografar e enviar logs sem degradação da QoS. Em termos energéticos, o nó tem capacidade para operar 12 h com bateria de 10 000 mAh, viabilizando deploy temporário em locais sem estruturada para fornecimento de energia (tomadas).

Como pode ser observado na Tabela 4.5 a maioria dos classificadores obtiveram desempenho ótimo. Uma característica a se observar neste caso é o tempo de construção dos modelos e posteriormente sua capacidade de futuramente serem portados para dispositivos menores e com baixa capacidade computacional. Quando observada a distribuição dos valores constantes no valor amplitude das portadoras (Figura 4.13), a tarefa de distinção entre as ações maliciosas e benígnas se resume a diferenciar entre duas distribuições. Uma delas com tendência a ser uniforme (maliciosas) e outra com *bell-shape* e cauda (benígnas). Uma investigação mais aprofundada nesse sentido pode ser interessante para revelar características das distribuições com confirmação e protocolo estatístico robusto,

algo que está fora do escopo deste estudo de caso.

Tabela 4.5. Desempenho dos classificadores considerados [de Almeida et al. 2024].

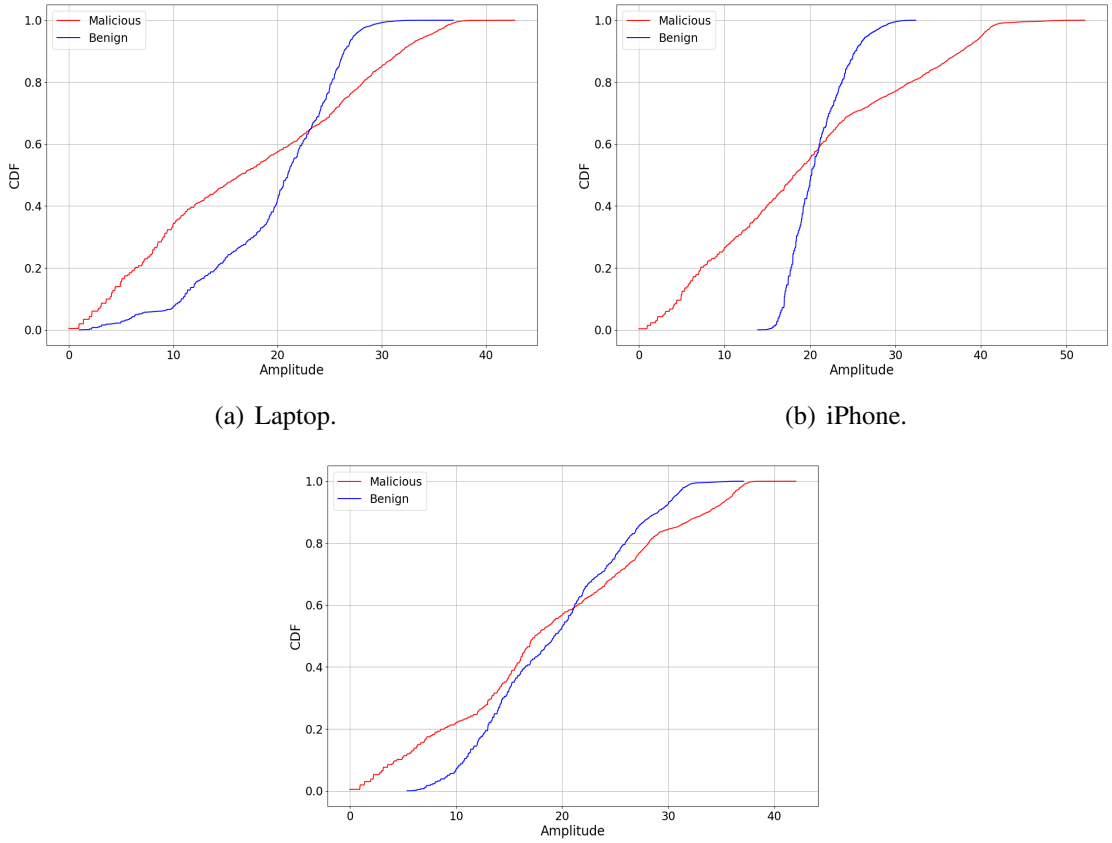
Classificador	Ambiente de teste	Precisão (%)	F1-Score	Construção em (s)
SVM	Câmara semi-anecoica	100,00	1,000	23,35
	Lab 1	99,99	1,000	43,41
	Lab 2	99,98	1,000	116,99
Random Forest	Câmara semi-anecoica	100,00	1,000	9,68
	Lab 1	99,99	1,000	27,31
	Lab 2	99,97	1,000	19,07
KNN	Câmara semi-anecoica	100,00	1,000	0,04
	Lab 1	99,99	1,000	0,03
	Lab 2	99,98	1,000	0,05
Decision Tree	Câmara semi-anecoica	100,00	1,000	1,63
	Lab 1	99,97	1,000	1,93
	Lab 2	99,95	1,000	1,29
Naive Bayes	Câmara semi-anecoica	100,00	1,000	2,91
	Lab 1	96,06	0,921	3,17
	Lab 2	94,76	0,948	2,15

Do ponto de vista operacional, o estudo indica que inserir de dois a três ESP32[2] por sala oferece cobertura angular satisfatória: o que vai permitir uma cobertura maior e potencialmente possibilitar a triangulação e localização física aproximada do atacante. Os presentes resultados demonstram que, diferentemente de NIDS baseados em payload, a distribuição espacial dos sensores melhora a qualidade do "rastro eletromagnético" usado como assinatura física.

Em síntese, os resultados provam o conceito que o Spider-Sense cumpre os limites de latência e precisão para proteção de SSIDs corporativos, sem exigir hardware especializado ou alterações na infraestrutura Wi-Fi. A seção seguinte discute como esses nós podem integrar-se a um centro de operações de segurança, enviando alertas em MQTT ou syslog, e quais extensões são necessárias para cobrir ataques em bandas de 6 GHz [de Almeida et al. 2024].

4.6.3. Integração com SOC e usos industriais

Resultados promissores de laboratório só se convertem em valor quando se encaixam no ecossistema de defesa de uma organização. O Spider-Sense foi projetado para dialogar com um Security Operations Center (SOC) moderno sem exigir alterações na malha Wi-Fi existente. Cada nó ESP32[2] encapsula a decisão da árvore J48 em um registro JSON com cinco campos: timestamp, ID do nó, canal e rótulo (benigno ou ataque). O pacote viaja por MQTT criptografado até um broker local; e finalmente, um conector é usado para preparar os dados para inserção no ambiente SIEM. Toda a conversão leva $\approx$4 ms, de modo que o alarme completo—medido da chegada do quadro `authentication request` ao disparo do evento no dashboard—fica abaixo de 85 ms, ainda dentro do or-



(a) Laptop.

(b) iPhone.

(c) Android.

Figura 4.13. CDFs das amplitudes do dataset [de Almeida et al. 2024].

çamento de 100 ms. Esse valor supera a velocidade de correlação de sondas DPI baseadas em payload, que precisam primeiro decodificar a pilha TLS antes de inferir anomalias.

Por fim, a chegada do IEEE 802.11bf simplificará ainda mais a integração. O draft atual já permite que pontos de acesso publiquem relatórios CSI em moldes similares aos usados pelo Spider-Sense, eliminando a necessidade de firmware customizado. Dessa forma, a viabilidade econômica e técnica do sistema só tende a melhorar, abrindo caminho para que a assinatura física do canal se torne elemento indispensável nos playbooks de resposta a incidentes.

4.7. Tendências Futuras e Fronteiras de Pesquisa

Os estudos de caso de autenticação (Seção4.5) e Spider-Sense (Seção4.6) demonstram que o êxito de um sistema baseado em CSI depende menos do algoritmo escolhido e mais da forma como os dados são obtidos e mantidos ao longo do tempo. Como também apresentado por [Chen et al. 2023], três aprendizados se destacam.

Primeiramente, a coleta deve ser breve e precisa. Sessões prolongadas atingem um ponto assintótico de tal forma que pode-se cessar a coleta naquele instante. Como demonstrado anteriormente, as capturas de 15 s não necessariamente trazem ganho de informação quando comparadas com capturas de menor duração (e.g., 5 s). Em segundo,

a rotulagem exige gatilhos externos. No caso da autenticação, o clique de início de captura, alinhado ao preâmbulo TSF, permite boa precisão. No Spider-Sense, os pacotes de ataque foram rotulados pelo MAC, permitindo casar cada quadro transmitido com o CSI recebido. Ensaios que dispensaram esse sincronismo acumularam desvios de até 600 ms, suficientes para misturar instâncias benignas e maliciosas e prejudicar o desempenho das soluções. A lição operacional é clara: sem rótulo confiável, o modelo aprende a classificar ruído.

Por fim, há uma tendência de presença de *concept drift*. Medições semanais em três salas apontaram degradação total na acurácia do Random Forest quando móveis foram rearranjados ou outras características exógenas (percentual de água no corpo da pessoa). Re-treinar o modelo completo consome 15 s de CPU no Raspberry Pi, mas exige novas amostras rotuladas.

Em conjunto, esses três pontos (coleta adequada, rótulo bem definido e atualização incremental) formam a base de uma implantação sustentável. Sem eles, o desempenho se dilui na variabilidade do ambiente. Ao passo que ao adotar tais cuidados, houve uma efetividade maior dos sistemas analisados e propostos neste minicurso. Um sentido para conseguir o aumento da prontidão tecnológica seria automatizar estes processos e desenvolver sistemas que sejam auto-adaptáveis (não supervisionado, auto-supervisionado, aprendizado por reforço) e consigam filtrar os diferentes ruídos presentes no sistema de modo mais generalizável (e.g., autoencoders).

4.7.1. Limitações atuais e barreiras técnicas

Do ponto de vista de consolidação de mercado, ainda há dificuldades que com a padronização do 802.11bf, apesar de tenderem a melhorar ao longo dos próximos anos. Em especial, atualmente citam-se: heterogeneidade de hardware, lacunas de padronização e desafios de privacidade.

O primeiro obstáculo é a diversidade de chipsets. Estudos concentram-se em um bem conhecido e limitado de hardware, o Intel 5300, o ESP32 e o Raspberry PI. Há implementações específicas com Rádio Definido por Software, o que pode trazer alguma flexibilidade para testes que envolvam Wi-Fi 6 com 512 subportadoras. No entanto, cada geração impõe velocidade de amostragem, quantização e filtros analógicos distintos.

A segunda barreira é o processo de padronização. O draft 802.11bf define a exposição CSI e que pode ser obtido a partir de rádios com frequências mais altas (tais como 6GHz e 60GHz[802.11ad/ay]). Porém, ainda requer a implementação pelos fabricantes para utilização. Nesse sentido, a ausência de uma API comum também também é um fator complicante. Em aplicações militares, por exemplo, processos de auditoria podem exigir a verificação de compliance e averiguação de cumprimento estrito das funcionalidades. Isso permite assegurar que sondas indesejadas sejam postas por meio de ataques em cadeia de distribuição/suprimentos (*supply-chain*).

O terceiro eixo envolve privacidade e regulação. O mesmo sinal que identifica uma intrusão pode revelar batimentos cardíacos ou padrão de respiração. Organizações que coletam CSI em larga escala precisam argumentar que a métrica não constitui dado pessoal sensível. A título de exemplo, a legislação europeia sugere enquadrar tais sinais

como "biometria soft", exigindo consentimento expresso. Já a LGPD brasileira ainda não define categoria específica. Um caminho técnico é executar inferência no próprio ponto de acesso e transmitir apenas hashes e rótulos, mas isso implica colocar modelos nos APs, aumentando custo e exigindo atualizações remotas seguras.

Somam-se ainda problemas operacionais já discutidos: concept drift acelerado por reconfiguração de ambiente e escassez de datasets de longo prazo. Sem bases públicas que cubram seis meses ou mais, permanece incerto se o filtro Hampel ou o re-treino incremental são adequados a cenários de produção onde o ambiente operacional é dinâmico.

Em síntese, a tecnologia atravessou o estágio de prova de conceito. Apesar de se apresentarem como dificuldades, há fortes desafios de pesquisas e oportunidades. No sentido de aumentar o nível de prontidão tecnológica: (i) convergência de medição CSI em 802.11bf, (ii) desenvolvimento de hardware e softwares que sejam mais suscetíveis a regulação e (iii) ferramental de validação cruzada que mantenha desempenho mesmo com hardware heterogêneo. Assim como oportunidades futuras, os protótipos avaliados colocam CSI no patamar de viabilidade técnica, mas o salto para adoção massiva dependerá de uma convergência entre novas tecnologias de rede, técnicas de aprendizado distribuído e modelos de governança de identidade.

A chegada do Wi-Fi 7 (IEEE 802.11be) multiplicará por quatro a largura de banda espectral e aumentará para 4096 o número de subportadoras capturáveis em modo sensing. Esse salto se traduz em vetor de atributos quase dezesseis vezes maior que o usado pelo ESP32 atual, abrindo margem para distinguir uma gama maior de eventos, como respiração de duas pessoas lado a lado. Contudo, processar tal volume em tempo real exigirá chipsets com unidades MAC/PHY dedicadas a exportar CSI sem obstruir a pilha de dados.

Ao mesmo tempo, a dispersão de pontos de captura demanda *Edge AI*. Executar inferência no ponto de acesso elimina salto de rede e garante que dados brutos de canal não saiam do domínio físico, mitigando riscos de privacidade. O Random Forest de 16 kB cabe na RAM de um AP Wi-Fi 6 de classe corporativa, mas redes neurais leves (TinyML) podem explorar as não linearidades extras do Wi-Fi 7 sem exceder 100 ms.

Aprendizado federado (FL) surge como estratégia para treinar esses modelos sem centralizar CSI sensível. Cada nó coleta, ajusta gradientes locais e envia apenas atualizações criptografadas para o agregador. Nesse sentido, uma lacuna a ser explorada envolve a utilização de múltiplos pontos de coleta (por exemplo, dez Raspberry Pi) em ambiente FL para reduzir o concept drift em um ambiente ou tipo de aplicação em específico. Como a integração com o ambiente físico pode trazer diferenciações na taxa com que os eventos de atualização ocorrem, o tipo de eventos que disparam mudanças e estratégias diferenciadas para agregação dos modelos.

4.8. Considerações Finais

Este capítulo apresentou uma visão abrangente sobre a autenticação baseada em informações da camada física, com foco no uso do Channel State Information (CSI) extraído de dispositivos Wi-Fi comerciais. Foram discutidas as limitações dos métodos tradicionais diante de ataques sofisticados como revezamento e spoofing, apresentando o CSI como

alternativa viável para incorporar consciência ambiental no processo de autenticação. A partir da análise de duas implementações distintas, conduzidas em contextos acadêmico e operacional, foi possível validar o potencial dessa abordagem em cenários com restrições reais de hardware, mobilidade e interferência. A combinação de pré-processamento adequado, extração eficiente de características e uso de modelos leves permitiu alcançar resultados competitivos mesmo sem infraestrutura especializada.

Em suma, dentro do contexto de cibersegurança a assinatura física do canal Wi-Fi não substitui criptografia nem autenticação tradicional; ela adiciona uma camada invisível que dificulta ataques de *relay*, *spoofing* e intrusão silenciosa. Oportunidades futuras que contemplem novas tecnologias e abordagens inovadoras tais como Wi-Fi 7, Edge AI e aprendizado federado possuem o potencial de se transformar em tendência para utilização em cibersegurança. Isso fortalecerá toda gama de dispositivos que instrumentam o mundo físico como é o caso da Internet das Coisas.

Referências

[iee 2021] (2021). Ieee standard for information technology–telecommunications and information exchange between systems - local and metropolitan area networks–specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016)*, pages 1–4379.

[ISO 2021] (2021). ISO/IEC 19795-2: 2021 Biometric performance testing and reporting — Part 2: Testing methodologies for technology and scenario evaluation.

[Abdelnasser et al. 2015] Abdelnasser, H., Youssef, M., and Harras, K. A. (2015). Wigest: A ubiquitous wifi-based gesture recognition system. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 1472–1480.

[Abu Ali et al. 2024] Abu Ali, N. A., Rehman, M., Mumtaz, S., Khan, M. B., Hayajneh, M., Ullah, F., and Shah, R. A. (2024). Contactless diseases diagnoses using wireless communication sensing: Methods and challenges survey. *ACM Comput. Surv.*, 56(9).

[Adib and Katabi 2013] Adib, F. and Katabi, D. (2013). See through walls with wifi! In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, SIGCOMM '13, page 75–86, New York, NY, USA. Association for Computing Machinery.

[Afshar et al. 2022] Afshar, A., Vakili, V. T., and Daei, S. (2022). Active user detection and channel estimation for spatial-based random access in crowded massive mimo systems via blind super-resolution. *IEEE Signal Processing Letters*, 29:1072–1076.

[Al-qaness et al. 2016] Al-qaness, M. A. A., Li, F., Ma, X., Zhang, Y., and Liu, G. (2016). Device-free indoor activity recognition system. *Applied Sciences*, 6(11).

[Aleesa et al. 2020] Aleesa, A. M., Zaidan, B. B., Zaidan, A. A., and Sahar, N. M. (2020). Review of intrusion detection systems based on deep learning techniques: coherent taxonomy, challenges, motivations, recommendations, substantial analysis and future directions. *Neural Computing and Applications*, 32(14):9827–9858.

[Almeida et al. 2024] Almeida, F. S., Trindade, E. F. G., Pettersson, M., Machado, R., and Jr., L. A. P. (2024). Spidersense: Early intruder detection in wi-fi networks using channel state information. In *Proc. IEEE Global Communications Conference (GLOBECOM)*, pages 1–6.

[Bertoli et al. 2024] Bertoli, G. d. C., Fernandes, G. V. C., Monici, P. H. B., Guibo, C. H. d. A., Santos, A. L. d., and Pereira Júnior, L. A. (2024). Design and implementation of intelligent packet filtering in iot microcontroller-based devices. *Journal of Internet Services and Applications*, 15(1):289–301.

[Chen et al. 2023] Chen, C., Zhou, G., and Lin, Y. (2023). Cross-domain wifi sensing with channel state information: A survey. *ACM Comput. Surv.*, 55(11).

[Cisco 2020] Cisco (2020). Cisco Annual Internet Report (2018–2023). White Paper.

[Cominelli et al. 2023] Cominelli, M., Gringoli, F., and Restuccia, F. (2023). Exposing the csi: A systematic investigation of csi-based wi-fi sensing capabilities and limitations. In *2023 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 81–90.

[Cominelli et al. 2021] Cominelli, M., Kosterhon, F., Gringoli, F., Lo Cigno, R., and Asadi, A. (2021). Ieee 802.11 csi randomization to preserve location privacy: An empirical evaluation in different scenarios. *Computer Networks*, 191:107970.

[de Almeida et al. 2024] de Almeida, F. S., Trindade, E. F. G., Pettersson, M. I., Machado, R., and Júnior, L. A. P. (2024). Spider-sense: Wi-fi csi as a sixth sense for early detection in network intrusion detection systems. In *GLOBECOM 2024 - 2024 IEEE Global Communications Conference*, pages 2437–2442.

[De Carvalho Bertoli et al. 2021] De Carvalho Bertoli, G., Pereira Júnior, L. A., Saotome, O., Dos Santos, A. L., Verri, F. A. N., Marcondes, C. A. C., Barbieri, S., Rodrigues, M. S., and Parente De Oliveira, J. M. (2021). An end-to-end framework for machine learning-based network intrusion detection system. *IEEE Access*, 9:106790–106805.

[Ding et al. 2018] Ding, E., Li, X., Zhao, T., Zhang, L., and Hu, Y. (2018). A robust passive intrusion detection system with commodity wifi devices. *Journal of Sensors*, 2018(1):8243905.

[Du et al. 2025] Du, R., Hua, H., Xie, H., Song, X., Lyu, Z., Hu, M., Narengerile, Xin, Y., McCann, S., Montemurro, M., Han, T. X., and Xu, J. (2025). An overview on ieee 802.11bf: Wlan sensing. *IEEE Communications Surveys& Tutorials*, 27(1):184–217.

[Espressif Systems 2018] Espressif Systems (2018). Csi documentation – esp-idf programming guide. `https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/wifi.html#wi-fi-csi`. Acessado em 2024-01-20.

[Espressif Systems 2019] Espressif Systems (2019). esp-csi: Channel state information extraction for esp32. `https://github.com/espressif/esp-csi`. Acessado em 2024-01-20.

[Firch 2024] Firch, J. (2024). Wireless network attacks: Learn the common types of wireless attacks. `https://purplesec.us/learn/wireless-network-attack/`. Acessado em 2024-03-22.

[Forbes et al. 2020] Forbes, G., Massie, S., and Craw, S. (2020). Wifi-based human activity recognition using raspberry pi. In *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 722–730.

[Geng et al. 2022] Geng, J., Huang, D., and la Torre, F. D. (2022). Densepose from wifi.

[Gringoli et al. 2021] Gringoli, F., Cominelli, M., Blanco, A., and Widmer, J. (2021). Ax-csi: Enabling csi extraction on commercial 802.11ax wi-fi platforms. In *Proceedings of the 15th ACM Workshop on Wireless Network Testbeds, Experimental Evaluation & CHaracterization*, WiNTECH '21, page 46–53, New York, NY, USA. Association for Computing Machinery.

[Gringoli et al. 2019] Gringoli, F., Schulz, M., Link, J., and Hollick, M. (2019). Free your csi: A channel state information extraction platform for modern wi-fi chipsets. In *Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, WiNTECH '19, page 21–28, New York, NY, USA. Association for Computing Machinery.

[Gu et al. 2024] Gu, Y., Chen, J., He, K., Wu, C., Zhao, Z., and Du, R. (2024). WiFiLeaks: Exposing Stationary Human Presence Through a Wall With Commodity Mobile Devices . *IEEE Transactions on Mobile Computing*, 23(06):6997–7011.

[Guo and Ho 2022] Guo, J. and Ho, I. W.-H. (2022). Csi-based efficient self-quarantine monitoring system using branchy convolution neural network. In *2022 IEEE 8th World Forum on Internet of Things (WF-IoT)*, pages 1–6.

[Guo et al. 2017] Guo, L., Wang, L., Liu, J., Zhou, W., Liu, B. L. T., Li, G., and Li, C. (2017). A novel benchmark on human activity recognition using wifi signals. In *2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pages 1–6.

[Halperin et al. 2011] Halperin, D., Hu, W., Sheth, A., Wetherall, D., Anderson, T., and Padmanabhan, V. (2011). Tool release: Gathering 802.11n traces with channel state information. In *Proc. ACM SIGCOMM*. `http://dhalperi.github.io/linux-80211n-csitool/`.

[He et al. 2020] He, Y., Chen, Y., Hu, Y., and Zeng, B. (2020). Wifi vision: Sensing, recognition, and detection with commodity mimo-ofdm wifi. *IEEE Internet of Things Journal*, 7(9):8296–8317.

[Hernandez and Bulut 2020] Hernandez, S. M. and Bulut, E. (2020). Lightweight and standalone iot based wifi sensing for active repositioning and mobility. In *2020 IEEE 21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*, pages 277–286.

[Hernandez and Bulut 2023] Hernandez, S. M. and Bulut, E. (2023). Wifi sensing on the edge: Signal processing techniques and challenges for real-world systems. *IEEE Communications Surveys& Tutorials*, 25(1):46–76.

[IEEE 802.11 Working Group 2024] IEEE 802.11 Working Group (2024). IEEE Draft Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications – Amendment: WLAN Sensing. `https://mentor.ieee.org/802.11/dcn/21/11-21-0915-12-0sensing-802-11bf-draft-specification.docx`. Draft Specification IEEE 802.11bf, v12.0, March 2024.

[IEEE 802.11 Working Group 2025] IEEE 802.11 Working Group (2025). Ieee p802.11bf/d1.4: Wireless lan sensing amendment (draft). Available from IEEE Standards Association.

[Kong et al. 2021] Kong, H., Lu, L., Yu, J., Chen, Y., Xu, X., Tang, F., and Chen, Y.-C. (2021). Multiauth: Enable multi-user authentication with single commodity wifi device. In *Proceedings of the Twenty-Second International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, MobiHoc '21, pages 31–40, New York, NY, USA. Association for Computing Machinery.

[Kotaru et al. 2015] Kotaru, M., Joshi, K., Bharadia, D., and Katti, S. (2015). Spotfi: Decimeter level localization using wifi. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM '15, page 269–282, New York, NY, USA. Association for Computing Machinery.

[Li et al. 2017] Li, S., Li, X., Niu, K., Wang, H., Zhang, Y., and Zhang, D. (2017). Ar-alarm: An adaptive and robust intrusion detection system leveraging csi from commodity wi-fi. In Mokhtari, M., Abdulrazak, B., and Aloulou, H., editors, *Enhanced Quality of Life and Smart Living*, pages 211–223, Cham. Springer International Publishing.

[Lin et al. 2023] Lin, C., Wang, P., Ji, C., Obaidat, M. S., Wang, L., Wu, G., and Zhang, Q. (2023). A contactless authentication system based on wifi csi. *ACM Trans. Sen. Netw.*, 19(2).

[Liu et al. 2014] Liu, H., Wang, Y., Liu, J., Yang, J., and Chen, Y. (2014). Practical user authentication leveraging channel state information (csi). In *Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security*, ASIA CCS '14, page 389–400, New York, NY, USA. Association for Computing Machinery.

[Liu et al. 2018] Liu, J., Chen, Y., Wang, Y., Chen, X., Cheng, J., and Yang, J. (2018). Monitoring vital signs and postures during sleep using wifi signals. *IEEE Internet of Things Journal*, 5(3):2071–2084.

[Liu et al. 2020] Liu, J., Liu, H., Chen, Y., Wang, Y., and Wang, C. (2020). Wireless sensing for human activity: A survey. *IEEE Communications Surveys& Tutorials*, 22(3):1629–1645.

[Liu et al. 2015] Liu, J., Wang, Y., Chen, Y., Yang, J., Chen, X., and Cheng, J. (2015). Tracking vital signs during sleep leveraging off-the-shelf wifi. In *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '15, page 267–276, New York, NY, USA. Association for Computing Machinery.

[Ma et al. 2019] Ma, Y., Zhou, G., and Wang, S. (2019). Wifi sensing with channel state information: A survey. *ACM Comput. Surv.*, 52(3):46.

[Meng et al. 2020] Meng, Y., Li, J., Zhu, H., Liang, X., Liu, Y., and Ruan, N. (2020). Revealing your mobile password via wifi signals: Attacks and countermeasures. *IEEE Transactions on Mobile Computing*, 19(2):432–449.

[Mirsky et al. 2018] Mirsky, Y., Doitshman, T., Elovici, Y., and Shabtai, A. (2018). Kitsune: An ensemble of autoencoders for online network intrusion detection. *Network and Distributed System Security Symposium (NDSS)*.

[Niu et al. 2018] Niu, K., Zhang, F., Chang, Z., and Zhang, D. (2018). A fresnel diffraction model based human respiration detection system using cots wi-fi devices. In *Proceedings of the 2018 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '18. Association for Computing Machinery.

[Rahman et al. 2023] Rahman, M. A., Shahriar, H., Clincy, V., Hossain, M. F., and Rahman, M. (2023). A quantum generative adversarial network-based intrusion detection system. In *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 1810–1815.

[Ronen and Shamir 2017] Ronen, E. and Shamir, A. (2017). Extended functionality attacks on iot devices: The case of smart lights. *Proceedings of the IEEE*, 105(8):1495–1510.

[Scarfone 2022] Scarfone, K. (2022). A list of wireless network attacks. `https://www.techtarget.com/searchsecurity/feature/A-list-of-wireless-network-attacks`. Acessado em 2023-08-19.

[Schulz et al. 2017] Schulz, M., Wegemer, D., and Hollick, M. (2017). Nexmon: The c-based firmware patching framework. `https://seemoo.de/nexmon/`.

[Schulz et al. 2018] Schulz, M., Wegemer, D., and Hollick, M. (2018). The nexmon firmware analysis and modification framework: Empowering researchers to enhance wi-fi devices. *Computer Communications*, 129:269–285.

[Seifeldin and Youssef 2011] Seifeldin, M. A. and Youssef, M. (2011). RASID: A robust wlan device-free passive motion detection system. In *Proc. IEEE Int. Conf. on Pervasive Computing and Communications (PerCom)*, pages 180–189.

[Sen et al. 2012] Sen, S., Radunovic, B., Choudhury, R. R., and Minka, T. (2012). You are facing the mona lisa: spot localization using phy layer information. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, MobiSys '12, page 183–196, New York, NY, USA. Association for Computing Machinery.

[Shah and Kanhere 2017] Shah, S. W. and Kanhere, S. S. (2017). Wi-auth: Wifi based second factor user authentication. In *Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, MobiQuitous 2017, pages 393–402, New York, NY, USA. Association for Computing Machinery.

[Sharma et al. 2025] Sharma, A., Mishra, D., Jha, S., and Seneviratne, A. (2025). Wispoof: Generating adversarial wireless signals to deceive wi-fi sensing systems. *Journal of Information Security and Applications*, 91:104052.

[Shen et al. 2021] Shen, X., Ni, Z., Liu, L., Yang, J., and Ahmed, K. (2021). Wipass: 1d-cnn-based smartphone keystroke recognition using wifi signals. *Pervasive and Mobile Computing*, 73:101393.

[Soto et al. 2022] Soto, J. C., Galdino, I., Caballero, E., Ferreira, V., Muchaluat-Saade, D., and Albuquerque, C. (2022). A survey on vital signs monitoring based on wi-fi csi data. *Computer Communications*, 195:99–110.

[Systems 2022] Systems, E. (2022). ESP32 CSI Tool. `https://github.com/espressif/esp32-csi-tool`.

[Tan et al. 2022] Tan, S., Ren, Y., Yang, J., and Chen, Y. (2022). Commodity wifi sensing in ten years: Status, challenges, and opportunities. *IEEE Internet of Things Journal*, 9(18):17832–17843.

[Truong et al. 2020] Truong, H., Trovato, B., and Smith, G. W. (2020). Practical replay attacks on nfc payments: From threats to mitigations. In *Proceedings of the 2020 ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pages 83–93. ACM.

[Truong and Tippenhauer 2020] Truong, H. T. and Tippenhauer, N. O. (2020). Practical relay attack on contactless payments—by smartphones. In *Proc. IEEE S&P*, pages 1132–1148.

[Vanhoef 2021] Vanhoef, M. (2021). Fragment and forge: Breaking wi-fi through frame aggregation and fragmentation. In *Proc. USENIX Security*, pages 1–18.

[Viegas et al. 2020] Viegas, E., Santin, A., Santos, R., and Abreu, V. (2020). Sistema de detecção de intrusão confiável baseado em aprendizagem por fluxo. In *Anais do*

XX Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais, pages 382–394, Porto Alegre, RS, Brasil. SBC.

[Wang et al. 2019] Wang, F., Han, J., Lin, F., and Ren, K. (2019). Wipin: Operation-free passive person identification using wi-fi signals. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6.

[Wang et al. 2016] Wang, W., Liu, A. X., and Shahzad, M. (2016). Gait recognition using wifi signals. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '16, pages 363–373, New York, NY, USA. Association for Computing Machinery.

[Wang et al. 2017] Wang, X., Gao, L., Mao, S., and Pandey, S. (2017). Csi-based fingerprinting for indoor localization: A deep learning approach. *IEEE Transactions on Vehicular Technology*, 66(1):763–776.

[Wang et al. 2020] Wang, X., Wang, Y., and Wang, D. (2020). A real-time csi-based passive intrusion detection method. In *2020 IEEE Intl Conf on Parallel& Distributed Processing with Applications, Big Data& Cloud Computing, Sustainable Computing& Communications, Social Computing& Networking (ISPA/BDCloud/SocialCom/SustainCom)*, pages 1091–1098.

[Wang and Liu 2019] Wang, Y. and Liu, Y. (2019). A survey on wi-fi based sensing: Applications, challenges, and opportunities. *Computer Networks*, 153:95–113.

[Xie et al. 2019] Xie, Y., Li, Z., and Li, M. (2019). Precise power delay profiling with commodity wi-fi. *IEEE Transactions on Mobile Computing*, 18(6):1342–1355.

[Xu et al. 2022] Xu, X., Zhang, Y., and Li, J. (2022). A survey on relay attacks in contactless payments: Current trends and countermeasures. *IEEE Access*, 10:88234–88252.

[Yang et al. 2013] Yang, Z., Zhou, Z., and Liu, Y. (2013). From rssi to csi: Indoor localization via channel response. *ACM Comput. Surv.*, 46(2).

[Zhang et al. 2017] Zhang, D., Wang, H., and Wu, D. (2017). Toward centimeter-scale human activity sensing with wi-fi signals. *Computer Society*, 50(1):48–57.

[Zhang et al. 2019] Zhang, F., Niu, K., Xiong, J., Jin, B., Gu, T., Jiang, Y., and Zhang, D. (2019). Towards a diffraction-based sensing approach on human activity recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 3(1).

[Zheng et al. 2019] Zheng, Y., Zhang, Y., Qian, K., Zhang, G., Liu, Y., Wu, C., and Yang, Z. (2019). Zero-effort cross-domain gesture recognition with wi-fi. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '19, page 313–325, New York, NY, USA. Association for Computing Machinery.

[Zhuang et al. 2021] Zhuang, W., Shen, Y., Li, L., Gao, C., and Dai, D. (2021). Develop an adaptive real-time indoor intrusion detection system based on empirical analysis of ofdm subcarriers. *Sensors*, 21(7).

[Zou et al. 2017] Zou, H., Zhou, Y., Yang, J., Gu, W., Xie, L., and Spanos, C. (2017). Freecount: Device-free crowd counting with commodity wifi. In *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pages 1–6.