Submission date: 04/07/2025
1st Resubmission date: 16/10/2025
2nd Resubmission date: 07/12/2025
Camera ready submission date: 13/01/2026

1st round notification: 30/09/2025
2nd round notification: 27/11/2025
3rd round notification: 06/01/2026
Available online: 27/03/2026
Publication date: 27/03/2026

Section: Regular article

Tecnologias e Ferramentas Utilizadas em *DevSecOps*: Uma revisão sistemática da literatura

Technologies and Tools for *DevSecOps*: A systematic literature review

Francisco Carlos Silva Pimentel[1], **Luis Carlos Costa Fonseca**[1], **Omar Andres Carmona Cortes**[2,1]

[1]Universidade Estadual do Maranhão (UEMA)
São Luis, MA – Brasil

[2]Departamento de Computação (DComp) – Instituto Federal do Maranhão (IFMA)
São Luis, MA – Brasil

calpimentel@gmail.com, omar@ifma.edu.br, luisfonseca@professor.uema.br

Abstract. *The advent of DevOps has brought faster Agile deliveries through automation. However, the speed of continuous delivery and deployment has hurt security testing and threat detection, mainly because such tests are applied after the application is delivered. Because they are time-consuming, they cannot keep up with the speed of software delivery. To fill this gap, DevSecOps emerged, including security testing in each phase of the DevOps lifecycle. In this sense, this study aims to investigate the security technologies and tools used in DevSecOps and to provide a knowledge base on which the authors have referenced all of them in their research. Therefore, a Systematic Literature Review was adopted as the methodology for this research, delimiting the period to the first half of 2024. Of the 87 published studies, 42 were selected after a comprehensive review process focusing on three guiding questions. The first question (Q1): "What types of security tools have been reported in the most recent publications?", the second question (Q2): "What technologies and platforms are referenced in DevSecOps publications?" and the third question (Q3): "What security tools and technologies are unique and more unusual addressed in the latest publications?" As a result, the answers to question Q1 revealed thirteen categories of consistent tools referenced by different articles. As for Q2 on technologies, AI was most frequently referenced in the articles. As for Q3, only two tools were identified as unique in the publications selected for this RSL, indicating that, within the time frame of this research, there was significant consistency in the tools used in DevSecOps*

Keywords. *DevSecOps; DevOps; Continuous Delivery; Continuous Deployment; Security Tools.*

Cite as: Pimentel, F. C. S. & Fonseca, L. C. C. & Cortes, O. A. C. (2026). Tecnologias e Ferramentas Utilizadas em *DevSecOps*: Uma revisão sistemática da literatura. iSys: Revista Brasileira de Sistemas de Informação (Brazilian Journal of Information Systems), 18(1), 18:1-18:24. DOI: 10.5753/isys.2025.6207

Resumo. *Com o advento do DevOps as entregas ágeis tornaram-se mais rápidas devido a automatização. Porém, a rapidez das entregas contínuas juntamente com o deployment contínuo tem tido efeito negativo quanto aos testes de segurança e detecção de ameaças, principalmente, por que tais testes são aplicados ao final, após a entrega do aplicativo. Por serem demorados, eles não conseguem acompanhar a velocidade das entregas de software. Para cobrir esta lacuna, o DevSecOps surgiu, incluindo testes de segurança em cada fase do ciclo de vida DevOps. Neste sentido, este estudo tem como objetivo investigar as tecnologias e ferramentas de segurança utilizadas no DevSecOps, com vista a fornecer uma base de conhecimento sobre quais delas têm sido referenciadas pelos autores em suas pesquisas sobre o assunto. Posto isso, uma Revisão Sistemática de Literatura foi adotada como metodologia para esta pesquisa, delimitando o período ao primeiro semestre de 2024, no qual, dos 87 estudos publicados, 42 deles foram selecionados após um processo de revisão completo sob o foco de três perguntas norteadoras. A primeira pergunta (Q1): "quais tipos de ferramentas de segurança foram reportados nas publicações mais recentes?", a segunda pergunta (Q2): "quais tecnologias e plataformas são referenciadas nas publicações sobre DevSecOps?" e a terceira pergunta (Q3): "Que ferramentas e tecnologias de segurança são únicas e mais incomuns abordadas nas últimas publicações?". Como resultado, as respostas da pergunta Q1, revelaram treze categorias de ferramentas consistentes referenciadas por diferentes artigos. Quanto a resposta da Q2, sobre as tecnologias, o emprego de IA foi mais frequente em referências dos artigos. Já quanto a resposta da Q3, apenas duas ferramentas foram identificadas como únicas nas publicações selecionadas para esta Revisão Sistemática da Literatura, revelando que no corte de tempo desta pesquisa houve muita consistência em relação as ferramentas usadas em DevSecOps encontradas na literatura.*

Palavras-Chave. *DevSecOps; DevOps; Entrega Contínua; Implantação Contínua; Ferramentas de Segurança.*

1. Introdução

DevOps é, basicamente, a colaboração entre as culturas de times de Desenvolvimento e Operações representados pelas suas siglas Dev e Ops. O seu principal objetivo é encurtar o *Software Development Lifecycle* (SDLC, traduzido como Ciclo de Vida de Desenvolvimento de Software) durante as entregas de novas funcionalidades, correções de erros e atualizações frequentes em um alinhamento fechado com os objetivos dos negócios. Ademais, automatização, entrega contínua e ciclos rápido de *feedback* para melhorar a qualidade de software são características associados ao *DevOps* [Jammeh 2020].

A origem do *DevOps* ocorreu pela necessidade de melhorar as limitações das antigas práticas de desenvolvimento de software, as quais são conhecidas como abordagem de silos, em que as equipes de desenvolvimento e operações trabalhavam separadamente [Jammeh 2020]. Seu conceito surgiu por volta de 2008, quando a indústria de Tecnologia da Informação (TI) começou a discutir a ideia de uma abordagem integrada e colaborativa no desenvolvimento e implantação de software. O crédito do termo *DevOps* foi concedido

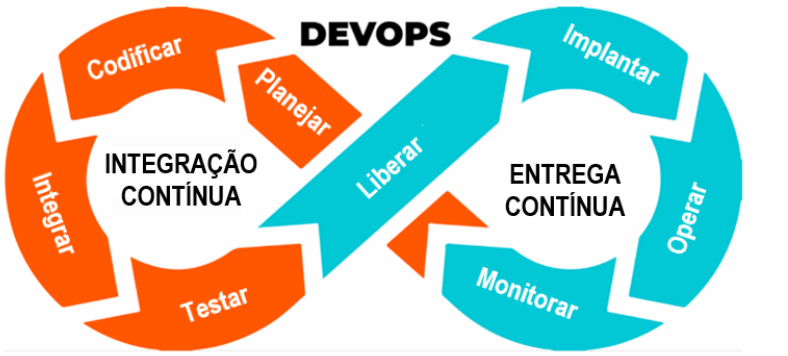

Figura 1. Ciclo de Vida *DevOps*

a Patrick Debois, um consultor de TI belga, quando ele organizou o evento ”DevOpsDay“ ocorrido na cidade de Ghent na Bélgica, em 2009. Este evento reuniu profissionais das esferas de desenvolvimento e operações para discutir maneiras de preencher a lacuna entre as duas disciplinas tradicionalmente separadas [Alok and Ziadon 2020].

As principais práticas de *DevOps* enfatizam a Integração Contínua (CI, do inglês, *Continuous Integration*) e a Implantação Contínua (CD, originalmente,*Continuos Delivery*), que são as mais importantes no ciclo *DevOps*. O ponto central do *DevOps* não está apenas na utilização das tecnologias e ferramentas sofisticadas que se utilizam atualmente, mas também na ênfase dos aspectos humanos e organizacionais da tecnologia [Jammeh 2020].

A CI apresenta várias fases, incluindo, Planejar, Codificar, Construir e Testar, representadas na cor vermelha na Figura 1. Cada uma das fases é interconectada com a saída das outras anteriores e, consecutivamente, alimentando a próxima. Como resultado, tem-se um fluxo de trabalho estabelecido para dá suporte ao desenvolvimento de software rápido e confiável. Consequentemente, auxiliando as equipes a reduzir erros manuais, melhorando a qualidade do código e acelerando o lançamento do software, beneficiando o usuário final. A CD estende os princípios da CI para os estágios posteriores do ciclo de vida de desenvolvimento de software, com foco na automação da liberação (*Release*), implantação (*Deploy*), operação (*Operation*) e monitoramento (*Monitor*) de aplicativos, conforme ilustrado, em azul, na Figura 1. Cada fase da CD é suportada por ferramentas e práticas específicas que automatizam as tarefas manuais que levariam mais tempo e recursos [Havard and Colomo-Palacios 2017].

O *DevOps* se concentra em ciclos de desenvolvimento mais curtos, conhecidos como incremento ou iterações, o que aumenta a velocidade da frequência de implantação em lançamentos mais confiáveis, estreitamente alinhados com os objetivos do negócio. Ao mesmo passo que se espera que eles estejam seguros contra as ameaças mais conhecidas. Contudo, os atuais paradigmas tradicionais de desenvolvimento de software e os testes de segurança são realizados no Ambiente de Testes, após a conclusão do desenvolvimento do código ou posteriormente no Ambiente de Homologação. Após sair da fase de testes bem-sucedidos, ele é considerado seguro e distribuído aos clientes ou implantado no Ambiente de Produção. Ainda neste contexto, as organizações estão tentando

aplicar o modelo *DevSecOps* adicionando ferramentas de segurança nos pipelines CI/CD para melhorar a segurança e automatizar a análise de segurança com o fluxo de vida do software [Aljohani and Alqahtani 2023].

O *DevSecOps* é uma junção de Desenvolvimento, Segurança e Operações que constitui uma metodologia progressiva em engenharia de software e práticas de implantação. Esta metodologia é construída de uma evolução cultural e processual com foco em estruturas colaborativas, processos automatizados e uma abordagem integrada à segurança em todo ciclo de vida de TI. Em relação à abreviatura do termo, *Dev* diz respeito ao desenvolvimento, significa o processo de planejamento, codificação, construção e teste do aplicativo. A sigla *Seg* refere-se a segurança, que significa a introdução de segurança antes e durante o SDLC. E *Ops* é referente ao processo de liberação, monitoramento e correção de erros ou problemas incertos que surgem no estágio de produção. É importante enfatizar que a inclusão da segurança como parte integrante de todo o ciclo de vida do aplicativo significa proteção integrada e não proteção funcional em torno de aplicativos de dados [RedHat 2023].

Assim, *DevSecOps* foi elaborado como uma extensão dos princípios *DevOps*, integrando segurança intercalada a cada ciclo do desenvolvimento de software. Baseando-se nos mesmos princípios, criou-se uma abordagem a qual as medidas de segurança tornaram-se uma responsabilidade compartilhada das equipes de desenvolvimento, segurança e operações, o que garantia que a segurança fosse parte dos ciclos contínuos já estabelecidos pelo *DevOps* no desenvolvimento de software. A segurança que era adicionada ao final do processo de forma separada, agora encontra-se intercalada em cada ciclo do processo. Estas mudanças para o *DevSecOps* foram impulsionadas pela crescente complexidade das ameaças à segurança e pela necessidade dos métodos de segurança serem integrados ao processo e também adequados a velocidade das entregas ágeis [Manohar et al. 2023].

Nesse contexto, o *DevSecOps* herda muitas práticas do *DevOps*, incluindo CI e CD, bem como a integração das alterações de código em um repositório central, o qual é utilizado para construir e testar o aplicativo automaticamente. Assim, o objetivo é fornecer um *feedback* rápido para detectar se um defeito foi introduzido no código. Caso possa ser identificado um problema de segurança, ele poderá ser corrigido de forma mais rápida possível. Já no contexto *DevSecOp*, as verificações de segurança são integradas ao pipeline de CI/CD, permitindo garantia de segurança antecipada e contínua [Havard and Colomo-Palacios 2017, Rajapakse et al. 2022].

A segurança do código e do software precisa ser discutida nas práticas em *DevOps*, as quais são bem definidas e usadas na indústria de forma mais ampla [Jammeh 2020]. Quando o software for desenvolvido em um ambiente não *DevSecOps*, problemas relativos à segurança podem emergir e causar atrasos no tempo de desenvolvimento. A rápida entrega de segurança do *DevSecOps* pode economizar no tempo e reduzir custos, bem como minimizar a necessidade de repetir o processo após o desenvolvimento e reconstruções desnecessárias de cenários de teste de segurança, resultando em código mais seguro. Dessa forma, *DevSecOps* visa incluir segurança dentro do processo de desenvolvimento tão profundamente quanto o *DevOps* introduz com suas operações, e

introduzindo segurança previamente no SDLC de uma aplicação, minimizando assim, objetivos de negócios [RedHat 2023].

Diante do exposto, este estudo tem como objetivo investigar as tecnologias e ferramentas de segurança utilizadas no *DevSecOps*, fornecendo uma base de conhecimento sobre as mais referenciadas em pesquisas recentes. Para tanto, adota-se uma Revisão Sistemática de Literatura (RSL) focada no primeiro semestre de 2024, analisando 42 estudos selecionados a partir de três perguntas norteadoras (Q1, Q2 e Q3). Essa abordagem preenche a lacuna de escassez de publicações sobre o tema, especialmente quanto à aplicação prática de ferramentas de segurança, e contribui para profissionais de TI ao mapear consistências e inovações na área.

Além disso, diferentemente de estudos como o de [Tomas et al. 2019], que focam em desafios gerais de integração de segurança em DevOps, esta RSL enfatiza ferramentas e tecnologias específicas reportadas em publicações recentes, priorizando sua frequência e unicidade.

Nesse contexto, este trabalho faz uma RSL, sendo estruturada da seguinte forma: na Seção 2 é apresentada a fundamentação teórica com conceitos básicos e a descrição do tema principal; na Seção 3 é detalhada a metodologia utilizada, explicitando as questões de pesquisa e como foram respondidas; a Seção 4 relata os principais resultados e discussões sobre cada questão, e por fim, a Seção 5 apresenta as conclusões finais e trabalhos futuros.

2. Fundamentação Teórica

A grande parte dos especialistas em segurança também eram organizados em silos separados e as preocupações com a segurança eram a última etapa a ser verificada quando um aplicativo já foi desenvolvido e pronto para o lançamento [Tomas et al. 2019, Myrbakken and Colomo-Palacios 2017] Semelhante, ao início do *DevOps*, o *DevSecOps* tenta promover a colaboração entre equipes de desenvolvimento, operação e segurança, estabelecendo uma abordagem proativa para limitar o ataque de superfície da aplicação [Yasar and Kontostathis 2016]. Contudo, a integração de práticas de segurança na moderna engenharia de software cria vários problemas, tais como:

- métodos de segurança tradicionais não são aplicáveis por não conseguirem acompanhar a agilidade e velocidade do *DevOps*;
- pouco se sabe sobre *DevSecOps* até agora, uma vez que apenas alguns estudos foram realizados sobre este tema [Tomas et al. 2019];
- falta de conhecimento de quando e onde usar as ferramentas existentes em automação, o que atrapalha profissionais de software de integrar a segurança em suas atividades de *DevOps*, como CI/CD [Tomas et al. 2019].

A abordagem *DevSecOps* tem por objetivo a integração de práticas de segurança a cada estágio do SDLC, para que isto possa ocorrer é necessário promover a cultura de colaboração, automação e alinhamento com segurança de software, ao unir equipes e processos de desenvolvimento, segurança e operações. Quanto a esses processos associados ao *DevSecOps*, podem-se destacar como principais princípios e práticas, os que seguem [Nikolov and Aleksieva-Petrova 2023]:

- **Segurança Shift-Left**: *DevSecOps* enfatiza o conceito de mudança de prática e consideração de segurança para esquerda no processo de desenvolvimento, integrando atividades de segurança desde o início, como durante as fases de coleta de requisitos, design e formalização, em vez de tratar a segurança de forma secundária.
- **Automação**: ela desempenha um papel importante no *DevSecOps*, a qual envolve o uso de ferramentas, *scripts* e estruturas para automatizar verificações de segurança, testes, implantação e processos de monitoramento. Para ajudar a identificar e resolver problemas importantes de segurança algumas práticas podem ser empregadas, como por exemplo, varredura de segurança automatizada, avaliações de vulnerabilidade e monitoramento contínuo.
- **CI/CD**: o *DevSecOps* incentiva o uso de pipelines de CI/CD, o que facilita o lançamento e implantação contínua de software. As organizações podem lançar softwares de forma mais rápida e segura, mantendo os controles de segurança necessários, automatizando os processos de construção, testes e implantação.
- **Infraestrutura como código (IaC)**: esta é uma prática em que os componentes da infraestrutura, como servidores, redes e configurações, são definidos e gerenciados por meio de código. As equipes de desenvolvimento possuem mais controle nas versões e automatização das alterações de infraestrutura por meio da adoção da IaC, o que torna as configurações de segurança mais consistentes, repetíveis e auditáveis.
- **Teste de segurança**: os testes de segurança, em todo SDLC, são um dos pontos defendidos pelo *DevSecOps*, nos quais se incluem a realização de avaliações de segurança, testes de penetração, revisões de código e varredura de vulnerabilidade para identificar e mitigar fragilidades de segurança desde o início do desenvolvimento.
- **Monitoramento contínuo**: este envolve o monitoramento ativo de software e infraestrutura para eventos e anomalias de segurança. Ainda neste contexto, ele inclui o registro, detecção de ameaças em tempo real e a resposta a incidentes. Além disso, permite que a detecção de violação de segurança, vulnerabilidade e atividades suspeitas.
- **Cultura de segurança**: a promoção de cultura de segurança dentro das equipes de desenvolvimento e organizações é um dos pontos a ser atingido. Este processo envolve a criação de conscientização, treinamento e educação dos membros da equipe sobre as melhores práticas de segurança.

Pelo fato de o *DevOps* focar em ciclos curtos e entregas rápidas a expectativa gerada por esta velocidade recair na segurança, a qual espera-se o lançamento de aplicação mais seguras e confiáveis contra ameaças e vulnerabilidades conhecidas, de forma tão rápida quanto os processos ágeis de desenvolvimento. Nos paradigmas tradicionais de desenvolvimento de software, após o desenvolvimento deles, a aplicação passa por testes de segurança em ambientes exclusivos. Uma vez que as aplicações passem nos testes, são tidas como seguras e distribuídas aos clientes ou implantadas no ambiente de produção. Contudo, as organizações estão tentando aplicar o modelo *DevSecOps* adicionando ferramentas de segurança como o fluxo de trabalho do SDLC [Jammeh 2020].

2.1. SAST e DAST

Os artigos selecionados nesta Revisão Sistemática de Literatura (RSL) relacionados ao DevSecOps examinam metodologias de testes de segurança para identificar as diferenças entre elas e estabelecer um ponto em comum no qual seu uso combinado possa fornecer um nível mais alto de proteção contra ataques cibernéticos. Outra parte destes artigos científicos fornecem informações extremamente detalhadas e sistemáticas sobre as características de *Static Application Security Testing* (SAST, traduzido como Testes de Segurança de Aplicativos Estático) e *Dynamic Application Security Testing* (DAST, em português, Testes de Segurança de Aplicativos Dinâmico) [Nikolov and Aleksieva-Petrova 2023, Yang et al. 2019].

Contudo, há referências nesta RSL que abordam ambas as metodologias. Sendo que algumas pesquisas semelhantes focam especificamente em testes de segurança de aplicativos estáticos, como por exemplo os trabalhos de [Oyetoyan et al. 2018] e [Rangnau et al. 2020], que abordam, particularmente, métodos de testes de segurança de aplicativos estáticos de uma forma mais ampla. Outra parte das publicações são focadas especificamente em testes de segurança dinâmica [Oyetoyan et al. 2018], que apresentam a metodologia de testes de software dinâmico com mais destaque.

As principais diferenças entre SAST e DAST encontram-se em seus tipos de abordagens quanto aos testes de segurança. O SAST examina o código de um aplicativo em sua fase de desenvolvimento, com o objetivo de identificar códigos inseguros que podem comprometer a segurança e implicar em riscos. Já o DAST, faz a avaliação dos testes de segurança com o aplicativo em execução e o código fonte do aplicativo não é necessário para este tipo de ferramenta [Sharma 2021, Dencheva 2022].

Logo, o DAST opera simulando um ataque do ponto de vista externo, o qual assume que o agente de teste não possui conhecimento do funcionamento interno do aplicativo, isso permite que a vulnerabilidade de segurança, não capturadas pelo SAST, possam ser identificadas. Em particular, as que somente se manifestam durante o funcionamento do software [Sharma 2021, Dencheva 2022].

2.2. Teste de Segurança

As aplicações modernas que combinam Web e Nuvem contêm testes de falhas de segurança de serviços, infraestrutura e plataforma, a níveis oferecidos pelos próprios provedores de nuvem. Os testes dinâmicos de segurança de aplicativos (DAST) têm o foco em testes para determinar como um aplicativo, em execução, responde a solicitações maliciosas. Os cenários de ataques, especificamente, são definidos como casos de testes que consistem em solicitações (elaboradas) e são enviadas ao sistema [Hsu 2019].

Neste cenário, um dos desafios do DAST é enviar as solicitações corretas de cada ataque e identificar as informações dentro da resposta que indica a presença de vulnerabilidades. Ainda neste contexto, DAST pode ser usado em configurações de caixa preta, em que o código do aplicativo não está disponível, ou em caixa branca, o qual o código pode ser acessível. Além disso, existem três técnicas de DAST que podem ser automatizadas: *Web Application Security Testing* (WAST), *Security API Scanning* (SAS) e *Behaviour Driven Security Testing* (BDST) [Hsu 2019].

A técnica WAST consiste em um teste automatizado de segurança que ataca um aplicativo da web por meio das suas próprias interfaces de usuários incluindo três etapas, que são:

- varredura da web: é mais aplicado nas URLs;
- varredura ativa: não executa solicitações maliciosas contra cada recurso identificado e avalia cada resposta do aplicativo para determinar possíveis problemas de segurança na URL de destino;
- relatórios de resultados: são gerados após as varreduras [Hsu 2019].

Além disso, a técnica WAST escaneia todo o aplicativo da web, mas pode não detectar falhas nos serviços de back-end. Portanto, é atualmente recomendável testar o serviço web por meio de suas APIs com o SAS. Esta técnica permite testes em cada *endpoint* em grandes detalhes e pode cobrir vários casos relevantes de segurança, como autenticação, validação de entrada ou tratamento de erros. No SAS, uma solicitação parametrizada é gerada e enviada para a API do serviço da web que está sendo testado por meio de um componente de solicitação. Os dados de entradas podem variar de credenciais para autenticação a cargas maliciosas, como injeção de SQL [Hsu 2019].

A BDST é uma extensão do *Test Driven Development* (TDD) e segue a ideia de integrar percepções em testes [Lenka et al. 2018]. O Desenvolvimento Orientado ao Comportamento (BDD, em inglês, *Behavior Driven Development*) usa uma abordagem de linguagem natural para definir o comportamento e o resultado esperado dos casos de testes. O BDST aplica a ideia do BDD ao domínio de testes de segurança para o benefício adicional de que profissionais não especializados em segurança possam entender tais testes, melhorando ainda mais a colaboração entre especialistas em segurança e equipes de DevOps [Hsu 2018]. Como esta técnica é executada contra o sistema como um todo, ela permite a identificação de vulnerabilidades que têm como alvo vários pontos de entrada no sistema. Além disso, o BDST combina muitas técnicas de segurança, como SAS ou WAST, para imitar cenários de ataques por um hacker, bem como encontrar problemas de segurança durante o uso normal do sistema [Hsu 2019].

2.3. Ferramentas de Segurança

Esta RSL identificou algumas categorias de ferramentas de segurança empregadas no contexto *DevSecOps*. Estas categorias por sua vez, podem ser divididas em grupos, mas as ferramentas de segurança foram ordenadas de forma diferente, como seguem-se:

1. **Para teste de segurança de aplicativos:**
    - **Hybrid Application Security Testing (HAST) e Interactive Application Security Testing (IAST)**: as ferramentas HAST unem o poder de duas ou mais técnicas de testes de segurança; já o IAST, pode ser considerado uma combinação de funcionalidades DAST e SAST, realizadas pela integração de agentes de software e sensores no ambiente de aplicativos em execução [Rajapakse et al. 2021a, Inc. b].
    - **Web Application Firewall (WAF)**: ele funciona semelhante a um proxy reverso, porém é instalado fora do aplicativo e pode proteger todo o conjunto de aplicativos de web [21].

- **Runtime Application Self-Protection (RASP)**: protege e monitora o comportamento e as solicitações de um aplicativo [Inc. b], ao rastreá-lo, em execução, na entrada e na saída, pode detectar e proteger contra ameaças, além de compatibilizar a integração [Rajapakse et al. 2021a].
- **Software Composition Analysis tools (SCA)**: checa vulnerabilidades conhecidas em dependências de aplicações, por exemplo, bibliotecas open-source de terceiros sem executar o aplicativo [Inc. b].

2. **Para testes de segurança de infraestrutura:**
   - **Intrusion Detection and Intrusion Prevention Systems (IDS/IPS)**: monitoram e previnem ambientes de computadores e redes de intrusos, com o intuito de identificar comportamentos que podem indicar um risco de segurança ou um ataque, assim ele relata os incidentes de segurança observados. O IPS é comparado ao IDS em termos de funcionalidade e também tenta ativamente prevenir ataques ao ambiente [OWASP 2022].
   - **Container Security Scanning**: pode ser vista como um tipo de Análise e Composição de Software (SCA do inglês *Software Composition Analysis*) cujo objetivo é identificar vulnerabilidades em containers, analisando imagens do container com o propósito de verificar se ela contém vulnerabilidades conhecidas [Inc. a].
3. **Para testes de segurança envolvendo automação:**
   - **Unattended robots (bot)**: Robôs autônomos, tal qual o nome indica, trabalham sem qualquer interferência humana. Quando o bot precisa funcionar junto com pessoas em processos, em que ocorre a explícita intervenção humana é necessária, os robôs são classificados como assistidos [Bhamidipati 2022].
   - **Hybrid Bots**: Robôs híbridos trazem recursos de Inteligência Artificial (IA) para o contexto, com algumas atividades realizadas por intervenção de humanos com contato direto, em caso de bots assistidos, é substituído por uma IA fazendo-os ficar intermediariamente entre bots assistidos e não assistidos [Bhamidipati 2022].
   - **Cybersecurity AI**: Esta nova tecnologia surgiu em 2003 e ainda está em seus estágios iniciais em termos de desenvolvimento. Basicamente, a intenção é usar IA para aprimorar os processos de segurança cibernética. Entre outras características desta tecnologia é a automação das análises de ameaças, detecção de anomalias e respostas a potenciais ataques cibernéticos. Espera-se que esta nova ferramenta continue evoluindo e possa ser implementada para melhorar gradualmente os processos de detecção de ameaças, e assim, reduzir vulnerabilidades em aplicativo [Bhamidipati 2022].
4. **Para outros tipos de testes de segurança:**
   - **Threat Modelling Tools**: podem ajudar a identificar, comunicar e entender ameaças de segurança e são usadas para modelar o entendimento de ameaças, incluindo mapeamentos de um ataque de superfície em diferentes níveis, por exemplo, de redes, de infraestrutura, de aplicações e por humano [Stallings 2017].

- **Security Information and Event Management (SIEM)**: são ferramentas que encapsulam a prática de segurança por meio de monitoria, sendo usadas para agregar relatórios e alertas de segurança de várias outras ferramentas de segurança [Rajapakse et al. 2021b].

3. Metodologia

Este estudo classifica-se como descritivo, com o propósito de retratar as mais recentes pesquisas da literatura, fazendo uma comparação acerca dos trabalhos publicados sobre as tecnologias e ferramentas empregadas em *DevSecOps*. Os dados foram analisados por meio de uma RSL, com o objetivo de possibilitar que autores possam realizar a mesma pesquisa e comparar os mesmos dados.

Determinou-se a produção cientifica relativa ao tema *DevSecOps*, por ser este bastante relevante e ao mesmo tempo que não há tantos estudos com este enfoque. O horizonte de tempo desta pesquisa contempla apenas o primeiro semestre de 2024.

3.1. Objetivos da pesquisa

O objetivo principal desta pesquisa foi realizar uma RSL sobre o tema *DevSecOps* relacionada às principais tecnologias e ferramentas empregadas nos processos de aplicação de segurança e detecção de ameaças. Para conquista deste objetivo, foram elaboradas questões norteadoras, com base nas amostras existentes, sendo elas:

- (Q1) Quais tipos de ferramentas de segurança foram reportadas nas publicações mais recentes?
- (Q2) Quais tecnologias e plataformas são referenciadas nas publicações sobre *DevSecOps*?
- (Q3) Que ferramentas e tecnologias de segurança são únicas e mais incomuns abordadas nas últimas publicações?

Para a busca foram selecionados artigos de acordo com os rótulos de títulos, palavras-chaves, principalmente do idioma inglês, por falta de publicações mais atuais em português.

3.2. Coleta e Origem dos Trabalhos

Para a pesquisa dos trabalhos acadêmicos mais relevantes sobre *DevSecOps*, a busca experimental é realizada em bases de dados eletrônicas on-line individualmente, tais como: o *IEEE Explore*, *ACM Digital Library* e *Science Direct*. Optou-se apenas por um termo o "*DevSecOps*", dentro das palavras-chaves, pois o mesmo define de forma consistente e satisfatória todo o escopo da pesquisa, que no total retornou 87 publicações, das quais, 50 delas, foram obtidas por meio da base *IEEE Explore*, na base ACM foram encontradas 18 publicações referente ao tema, e, por fim, a base *Science Direct*, com 19 publicações, como pode ser visto na Tabela 1.

3.3. Critérios de Seleção e Exclusão

Para esta RSL optou-se por seguir a recomendação de [Kitchenham 2004] usando critérios de seleção para inclusão e exclusão das pesquisas relacionadas ao tema. Neste sentido, elaboraram-se dois *checklists* que foram criados durante o desenvolvimento deste trabalho alinhados às questões de pesquisa, a saber:

Tabela 1. Publicações encontradas com a palavra-chave "*DevSecOps*"

Bases	Quantidade de Artigos	Percentual
IEEE Explore	50	57,47%
ACM Digital Library	18	20,69%
Science Direct	19	21,84%
TOTAL	87	100,00%

1. **Checklist para o critério de inclusão:**
   - A pesquisa tem que ter título, resumo, introdução e conclusão relevantes para as perguntas ou o objetivo principal desta RSL;
   - O texto da pesquisa em sua totalidade deve ser relevante às perguntas ou o objetivo principal desta RSL;
   - A pesquisa deve ter sido publicada nas três bases selecionadas;
   - O período das publicações deve abranger somente o primeiro semestre do ano de 2024;
   - A área de pesquisa deve estar no campo de Engenharia de Software e Ciência da Computação.
2. ***Checklist* para o critério de exclusão:**
   - O tema principal não tem nenhum foco em *DevSecOps*;
   - O tema do artigo de pesquisa descreve apenas o processo *DevSecOps* sem menção das tecnologias e ferramentas de segurança;
   - Pesquisas que têm como foco e tema principal plataformas de nuvem ou IoT;
   - Artigos os quais o texto não tem nenhum dado relevante a ser extraído para as questões desta pesquisa.

Das 87 publicações retornadas pelo termo de pesquisa nas três bases de dados, após a aplicação dos dois *checklist* de critérios de inclusão e exclusão, foram selecionadas 42 publicações elegíveis para esta RSL, os quais, algumas das bases de dados restringiram o acesso a alguns artigos que ficaram indisponíveis e foram contabilizados a parte, a Tabela 2 apresenta mais detalhes sobre estas ocorrências. Todos os artigos selecionados estão devidamente referenciados no Apêndice A, seguindo uma identificação da letra "A" e uma numeração com vistas a não se confundir com as Referências Bibliográficas.

Tabela 2. Publicações selecionadas

Bases	Elegíveis	Excluídos	Indisponível	Selecionados
IEEE Explore	50	11	4	35
ACM Digital Library	18	5	9	4
Science Direct	19	16	0	3
Total elegíveis	87		**Total selecionados**	42

3.4. Ameaças à Validade

Este estudo está sujeito a algumas ameaças à validade que precisam ser explicitadas. Em termos de validade interna, a aplicação dos critérios de inclusão e exclusão baseia-se na interpretação dos pesquisadores, o que pode levar à seleção ou descarte indevido de alguns estudos. Para mitigar esse risco, foram adotados checklists claros, inspirados em recomendações consolidadas para RSL, além de sucessivas leituras dos artigos elegíveis, buscando consistência na decisão final.

No que se refere à validade externa, o recorte temporal restrito ao primeiro semestre de 2024 limita a generalização dos achados para períodos mais longos. Ainda assim, tal delimitação foi definida intencionalmente para capturar o estado da arte mais recente em *DevSecOps*, servindo como um "instantâneo" da literatura. Estudos futuros poderão ampliar a janela temporal e comparar diferentes períodos, de forma a verificar a evolução das tecnologias e ferramentas identificadas.

Por fim, quanto à validade de construção e de conclusão, o uso de um único termo de busca ("DevSecOps") e de três bases específicas pode ter deixado de fora estudos relevantes que utilizam combinações alternativas de termos. No entanto, essa estratégia foi considerada adequada ao objetivo de mapear ferramentas explicitamente associadas ao conceito de DevSecOps na literatura, permitindo manter foco na interseção entre desenvolvimento, operações e segurança.

4. Resultados e Discussões

É importante enfatizar que os resultados apresentados nesta seção foram extraídos e sintetizados exclusivamente da seleção de artigos, A1 até A42, os quais se encontram listados no Apêndice A.

Ao sintetizar os dados das publicações selecionadas para esta pesquisa, considerou-se o fato de excluir marcas e fornecedores, concentrando-se apenas nos respectivos tipos de ferramentas e tecnologias empregadas com vistas a não desviar o foco. Além disso, na compilação de todos os dados há certas ferramentas que são relevantes e mais comuns nas publicações.

No recorte de tempo desta RSL, optou-se por igualar os dados e ter como alvo principal as categorias e tecnologias referentes as ferramentas de segurança e detecção de ameaças.

4.1. Pergunta de pesquisa Q1

No contexto geral sobre a questão da pesquisa Q1, este estudo não faz quantificação ou qualificação comparativa entre as ferramentas de segurança, muito menos ao seu uso. Portanto, ferramentas aplicadas a *DevSecOps* que foram encontradas nas publicações com mais frequência são apresentadas nesta pesquisa, da mesma forma que as menos frequentes. A adoção deste critério, deve-se ao fato da impossibilidade de coletar de forma confiável, nem extrair ou deduzir algumas ferramentas aplicadas em *DevSecOps*.

Este estudo encontrou treze tipos de ferramentas de segurança e detecção de ameaças que são aplicadas em diversas fases do *DevSecOps*. Uma apresentação destes

tipos de ferramentas e suas respectivas ocorrências nos artigos selecionados é apresentada na Figura 2. A Tabela 3 lista as publicações referentes aos tipos de ferramentas.

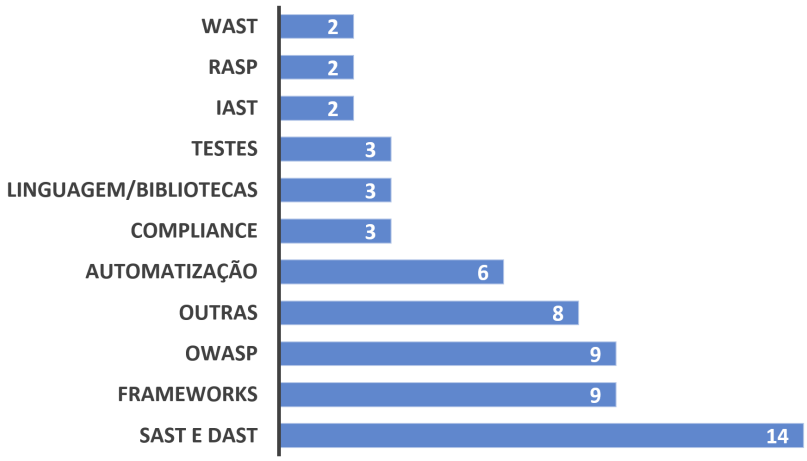

Figura 2. Tipos de Ferramentas de Segurança

Tabela 3. Tipos de ferramentas mencionados nas publicações selecionadas

Tipos de ferramentas	Artigos selecionados
Automatização	[A3, A11, A19, A28, A33, A36]
Compliance	[A11, A19, A25]
Frameworks	[A1, A3, A6, A14, A19, A31, A33, A36, A42]
Linguagens e bibliotecas	[A9, A19, A40]
Testes	[A19, A37, A35]
IAST	[A14, A16]
OWASP	[A2, A7, A12, A15, A23, A28, A37, A39, A41, A42]
RASP	[A7, A17]
WAST	[A12, A16]
SAST e DAST	[A1, A4, A6, A7, A12, A14, A16, A20, A21, A29, A32, A33, A37, A42]
Outros	[A7, A12, A14, A12, A17, A22, A27, A42]

Uma motivação para elaboração da questão de pesquisa Q1 foi a observação da pesquisa exploratória neste estudo. Foi percebido durante a pesquisa que, no recorte de tempo desta RSL, apesar de curto, levantou-se uma variedade de tecnologias e ferramentas abordadas entre os autores dos artigos selecionados. Porém um certo padrão foi identificado: uma concentração em tecnologias mais recentes e outras já consolidadas.

Outrossim, a predominância de ferramentas SAST e DAST nas publicações analisadas pode ser explicada pelo grau de maturidade e difusão dessas abordagens na indústria de software. Testes estáticos e dinâmicos de segurança se integram de forma relativamente direta aos pipelines de CI/CD, contam com amplo suporte de ferramentas comerciais e de

código aberto, e já fazem parte de boas práticas consolidadas para aplicações web e em nuvem [Rajapakse et al. 2022, Sinan et al. 2025].

Por outro lado, categorias menos frequentes, como ferramentas de modelagem de ameaça, robôs híbridos ou soluções baseadas em Inteligência Artificial para segurança, refletem tanto tendências emergentes quanto lacunas de adoção e padronização. Isso sugere que, embora o discurso sobre *DevSecOps* enfatize automação avançada e uso intensivo de IA, a prática reportada na literatura ainda se concentra em mecanismos clássicos de teste e análise de vulnerabilidades.

No final das contas, treze tipos de ferramentas de segurança foram identificados durante a compilação dos dados nesta RSL, sendo que a maior parte dos autores fizeram referencias sobre as ferramentas SAST e DAST aplicadas a testes de segurança e ameaças a aplicativos, conforme pode-se observar na Figura 2. Uma parte destas ferramentas foram identificadas como práticas de segurança, tais como: modelagem de segurança [A2, A7, A10, A13, A17, A25, A39, A41]; Gerenciamento de Segredos [A15, A31, A40]; Segurança e Gerenciamento de Infraestrutura [A1, A15, A16, A18, A23, A25, A28].

Ademais, as práticas de segurança OWASP foram mencionadas em 10 artigos (conforme Tabela 3) dentre os que foram selecionados nesta RSL. Porém houve poucas referências sobre ferramentas de testes segurança e ameaças para aplicativos Web. Sendo interessante destacar que foram identificados artigos [A9, A19, A40] que trataram soluções utilizando linguagens de programação, especialmente, *Python*. Além disso, vários *frameworks* com soluções combinadas são propostos por diferentes autores [A1, A3, A6, A14, A19, A31, A33, A36, A42].

A Figura 2 demostra que outras ferramentas foram referenciadas em um menor número de artigos [A7, A12, A14, A12, A17, A22, A27, A42], tais como: *Security API Scanning* (SAS), *Security-by-Design Development methodology* (SSDE), *Microsoft Threat Modeling Tool* (TMT). O mesmo ocorreu para as publicações que referenciaram testes *Behaviour Driven Security Testing* (BDST) e *Test Driven Development* (TDD), aplicados ao processo *DevSecOps*.

A categorização das treze famílias de ferramentas de *DevSecOps*, aqui identificadas, transcende a mera descrição técnica, posicionando-se como um pilar fundamental para a tomada de decisão gerencial em contextos de Sistemas de Informação (SI). A adoção e a implementação dessas tecnologias implicam uma reconfiguração substancial nos processos de CI/CD, onde a integração de ferramentas como SAST e DAST, não apenas introduz a segurança no processo de desenvolvimento, mas exige uma transformação cultural que promova a responsabilidade compartilhada entre as equipes de Desenvolvimento, Segurança e Operações. A resistência cultural à mudança é um dos principais desafios, especialmente quando equipes acostumadas à agilidade do *DevOps* percebem a segurança como um entrave à velocidade de entrega [Akbar et al. 2022].

Além disso, observa-se forte presença de ferramentas para testes de aplicações (por exemplo, SAST, DAST e SCA) e segurança de infraestrutura (como scanning de containers e IaC), reforçando que esses grupos constituem o núcleo das implementações de *DevSecOps* reportadas [Prates and Pereira 2025].

Do ponto de vista gerencial, a escolha estratégica e o investimento prioritário em certas categorias de ferramentas (como *Vulnerability Management* ou *Secrets Management*) devem ser justificados pela sua capacidade de mitigar riscos de negócio, reduzir custos operacionais com remediação e garantir a conformidade regulatória (*compliance*). Portanto, os achados desta RSL oferecem subsídios para que gestores de TI e CISOs possam elaborar estratégias de maturidade em *DevSecOps* que equilibrem a velocidade de entrega com a robustez da segurança organizacional.

Sob a perspectiva de sistemas de informação, a categorização das treze famílias de ferramentas de *DevSecOps* transcende a mera descrição técnica e se posiciona como um insumo para compreender *DevSecOps* como fenômeno sociotécnico. Revisões focadas em cultura e aspectos organizacionais de *DevSecOps* evidenciam que a integração de controles como SAST, DAST, SCA, IaC e monitoramento contínuo implica mudanças em processos, papéis e formas de colaboração entre equipes de desenvolvimento, operações e segurança, reforçando a ideia de que essas ferramentas atuam como mecanismos de coordenação e governança, e não apenas como componentes tecnológicos isolados [Rajapakse et al. 2022].

Em setores que lidam com serviços críticos, como governo, finanças e saúde, estudos apontam que práticas de *DevSecOps* associadas a essas ferramentas têm sido exploradas para reduzir a superfície de ataque e aumentar a resiliência organizacional frente a incidentes, ainda que nem sempre sob a mesma terminologia ou com o mesmo grau de formalização conceitual. Nesses cenários, a literatura sugere que a adoção de *DevSecOps* depende tanto da capacidade técnica de integrar ferramentas ao pipeline quanto da construção de uma cultura de segurança compartilhada, que alinhe tecnologia, processos e pessoas — perspectiva que reforça a aderência do tema à área de Sistemas de Informação [Rajapakse et al. 2022].

4.2. Pergunta de pesquisa Q2

A pergunta da pesquisa Q2, investiga sobre as tecnologias e plataformas referenciadas nos artigos selecionados. No que tange *DevSecOps*, percebeu-se uma tendência por inovações em IA juntamente com processos de automatização de testes de segurança. Na Tabela 4, foram relacionadas as principais publicações com tecnologias e plataformas de tais tendências, nas quais pode-se observar a presença de infraestrutura, ambiente em nuvem e de Internet das coisas (IoT).

Um ponto levantado durante a sintetização desta RSL, foi a referência do emprego de IA abordando algoritmos de Aprendizagem de Máquina e Aprendizagem Profunda, conforme a Tabela 4. Neste sentido, foi possível identificar o emprego frequente de IA para detectar possíveis ataques ou ameaças mais frequentes, analisar logs e trafego de rede, nos processos *DevSecOp*s. Geralmente estes artigos, descritos na referida tabela, estão associados as ferramentas de automação de segurança, que aplicam monitoramento contínuo para detectar anomalias e reconhecimentos de vulnerabilidades conhecidas.

Ainda neste contexto, pode-se citar alguns destaques entre os artigos selecionados. A publicação mais expressiva sobre este tema, foi o artigo [A19] que aborda a automatização de testes de segurança e detecção de ameaças com auxílio de IA em-

Tabela 4. Tecnologias e plataformas mencionadas nas publicações selecionadas

Tecnologias	**Artigos selecionados**
AI (Aprendizagem de Máquina e Aprendizagem Profunda)	[A1, A3, A14, A18, A19, A30, A33]
Banco de dados de vulnerabilidades	[A4, A24, A26, A28]
Plataformas	**Artigos selecionados**
Infraestrutura e nuvem	[A1, A6, A8, A10, A13, A15, A16, A17, A18, A20, A23, A25, A28, A30, A32, A33, A34, A39, A40, A42]
IoT	[A1, A8]
Containers e Kubernetes	[A1, A6, A13, A15, A16, A17, A18, A20, A24, A25, A34, A42]

pregando Algoritmos Genéticos, bem como, processos automatizados utilizando linguagem Python, o que consiste em uma mescla de diferentes tecnologias em uma mesma publicação.

Percebe-se que o emprego de IA nos processos *DevSecOps* consistem em uma opção para lidar como analise de um grande volume de dados, oriundos de aplicações, impossíveis de serem interpretados em tempo de execução. Ozkok et al. [A36] explorou a aplicação do chat GPT para avaliação de logs, o que consolida a tendência do emprego das tecnologias referentes a IA inseridas em pipelines *DevSecOps*.

Um segundo ponto relevante, observado durante a sintetização dos dados, foi a presença de testes de segurança e detecção de ameaças em plataformas de infraestrutura e ambiente em nuvem (especialmente AWS e *Microsoft Azure*), foram vinte artigos [A1, A6, A8, A10, A13, A15, A16, A17, A18, A20, A23, A25, A28, A30, A32, A33, A34, A39, A40, A42], nos quais os autores referenciaram ferramentas e serviços dos provedores de nuvem para testes de segurança, ameaças e detecção de ataques. Uma parte destes artigos [A1, A6, A13, A15, A16, A17, A18, A20, A24, A25, A34, A42] tratam também de ferramentas de segurança para escaneamento de *container* e a aplicação de ferramentas de segurança em serviços de orquestração de *containers*, como o *Kubernetes*. No que diz respeito a outras plataformas, pode-se observar ainda na Tabela 4, que apenas dois artigos [A1, A8] trataram o tema de segurança em dispositivos *edge computer*, ou dispositivos IoT.

4.3. Pergunta de pesquisa Q3

A pergunta de pesquisa Q3, questiona as tecnologias e ferramentas que não são tão frequentes ou com publicações únicas nos artigos relacionados. Tais ferramentas e tecnologias são listadas na Tabela 5, na qual observa-se apenas um artigo [A7], cuja referência de um serviço de segurança em ambiente de nuvem é o *Web Application Firewall* (WAF).

Tabela 5. Tecnologias e ferramentas únicas nas publicações selecionadas

Ferramentas	**Artigos selecionados**
WAF	[A7]
SecLab	[A8]

A outra referência solitária, a *SecLab* [A8], consiste em uma ferramenta de estrutura hierarquizada que controla e gerencia vários agentes em uma rede composta de

vários nós incorporando subsistemas de testes.

O primeiro artigo [A7], trata da ferramenta *Web Application Firewall* (WAF), uma ferramenta muito utilizada, especialmente, em ambiente em nuvem para filtrar e monitorar tráfego do protocolo HTTP, sendo tipicamente empregada em aplicativos para web. No mesmo artigo o autor ainda mencionou o uso de outras ferramentas de segurança como RASP, SAST, DAST e OWASP. Apesar de ser o único artigo que menciona WAF diretamente, o autor ainda recomenda ferramentas RASP porque elas têm um bom entendimento da comunicação de dados entre microsserviços e porque funcionam como um nível extra de proteção.

Coincidentemente, o segundo artigo [A8] foi um dos poucos a referenciar segurança em dispositivos IoT, e também o único a propor uma implementação do SecLab IoT, uma ferramenta habilitada de vários cenários realísticos de ataques cibernéticos em um ambiente laboratorial flexível de IoT.

Fora estas duas publicações, os outros artigos abordaram ferramentas e tecnologias mescladas em diferentes plataformas e infraestruturas as quais foram referenciadas por mais de um autor entre os artigos selecionadas para esta RSL.

5. Conclusões

O objetivo desta RSL foi contribuir para o conhecimento da existência das principais ferramentas e tecnologias empregadas em *DevSecOps*, por meio da compilação das principais publicações mais atualizadas, as quais fazem referências às que foram empregadas nos processos e pipelines de *DevSecOps*.

Na fase inicial desta pesquisa, iniciada pela exploração do termo *DevSecOp* em bases de dados consolidadas (*IEEE Explore*, *ACM* e *Science Direct*), identificou-se a frequência de referências em artigos das tecnologias e ferramentas referentes a testes e detecção de ameaças no recorte de tempo desta pesquisa.

Ademais, motivando-se por relacionar as atuais tecnologias e ferramentas de testes e detecção de segurança aplicadas a *DevSecOps*, a presente RSL investigou as correlações mais a fundo obedecendo um direcionamento que conduziu todo o estudo por meio de três perguntas de pesquisa, que foram:

- Q1: Quais tipos de ferramentas de segurança foram reportados nas publicações mais recentes?
- Q2: Quais tecnologias e plataformas são referenciadas nas publicações sobre *DevSecOps*?
- Q3: Que ferramentas e tecnologias de segurança são únicas e mais incomuns abordadas nas últimas publicações?

Assim, com as questões norteadoras, a condução da RSL possibilitou sintetizar os dados referentes aos artigos (A1-A42), apresentados no decorrer deste trabalho, os quais reuniram os resultados e discussões sobre as perguntas da pesquisa.

Referente à pergunta Q1, pode-se observar que houve uma tendência de artigos sobre ferramentas SAST e DAST, ao passo que, em menor número outras ferramentas são referenciadas em outras ou nas mesmas publicações.

Quanto à pergunta da pesquisa Q2, IA foi outra tendência quanto a tecnologias emergentes aplicadas a processos e pipelines *DevSecOps*. Ao passo que, há uma gama de artigos fazendo referência a infraestrutura e ambiente em nuvem, bem como a facilidade das ferramentas de segurança nativas oferecidas como serviços pelos próprios provedores de ambiente em nuvem. Foi possível notar certa correlação entre alguns artigos cruzando referências entre IA, infraestrutura e ambiente em nuvem [A1, A18, A30].

As plataformas sobre *containers* e orquestração deles, como o *Kubernetes*, por exemplo, tiveram um número razoável de publicações selecionadas. Entre eles alguns artigos foram relacionados juntamente com IA, infraestrutura e ambiente em nuvem [A1, A6, A13, A15, A16, A17, A18, A20, A25, A33].

Esta pesquisa identificou duas publicações que atendem à pergunta da pesquisa Q3, referente a ferramentas mencionadas em apenas uma publicação. A primeira é sobre a ferramenta de nuvem WAF [A7] e uma solução de segurança para dispositivos IoT, o SecLab [A8]. A resposta da Q3 é uma surpresa, por revelar apenas duas ferramentas com apenas uma referência, esta pesquisa conclui que isto possa ter acontecido devido ao corte de tempo (seis meses).

Além disso, percebe-se que há uma constância nas referências das tecnologias e ferramentas utilizadas em *DevSecOps*. Contudo, identificou-se que duas publicações tiveram referências únicas, em se tratando de segurança e detecção de ameaças.

Por fim, pode-se inferir que o universo de ferramentas e tecnologias empregadas em *DevSecOps* é diversificado e segmentado. Assim, a presente pesquisa atingiu o seu objetivo de apresentar, de forma clara, o que há de mais recorrente na utilização de ferramentas para segurança cibernética, sendo possível classificar e categorizar as mais modernas. Portanto, fornece-se um estudo esclarecedor com o intuito de divulgar uma área pouco conhecida, especialmente pelos próprios profissionais em TI.

Os resultados obtidos indicam que a adoção estruturada das categorias de ferramentas identificadas pode gerar ganhos organizacionais e sociais relevantes. Ao reforçar práticas de teste, monitoramento e resposta a incidentes, as organizações tendem a reduzir o tempo de detecção de vulnerabilidades, diminuir o impacto de falhas de segurança e aumentar a confiança de clientes, cidadãos e parceiros em serviços digitais críticos. Esses efeitos se traduzem em melhor governança de TI, maior aderência a requisitos regulatórios e suporte mais robusto à tomada de decisão orientada a risco.

Sob a ótica da área de Sistemas de Informação, o mapeamento realizado neste estudo oferece um referencial para gestores e profissionais definirem arquiteturas de SI, políticas de segurança e arranjos de cooperação entre equipes técnicas. Ao explicitar quais tecnologias e ferramentas têm sido priorizadas na literatura recente, a RSL contribui para que organizações planejem investimentos, capacitações e iniciativas de mudança organizacional compatíveis com a realidade dos seus contextos de negócio.

Para esta RSL os resultados e discussões atingiram os objetivos propostos, porém como trabalhos futuros pretende-se: (i) estender o horizonte temporal, (ii) catalogar exemplos de ferramentas ao invés de somente tipos e tecnologias; e, (iii) comparar e correlacionar tecnologias, processos e ferramentas no universo *DevSecOps*.

Agradecimentos

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

Referências

[Akbar et al. 2022] Akbar, M. A., Smolander, K., Mahmood, S., and Alsanad, A. (2022). Toward successful devsecops in software development organizations: A decision-making framework. *Information and Software Technology*, 147:106894.

[Aljohani and Alqahtani 2023] Aljohani, M. A. and Alqahtani, S. S. (2023). A unified framework for automating software security analysis in devsecops. In *2023 International Conference on Smart Computing and Application (ICSCA)*, pages 1–6. IEEE.

[Alok and Ziadon 2020] Alok, M. and Ziadon, O. (2020). Devops and software quality: A systematic mapping. *Computer Science Review*, 38:100308.

[Bhamidipati 2022] Bhamidipati, V. S. (2022). A holistic approach to ensure security and compliance while using robotic process automation. In *2022 Seventh International Conference on Parallel, Distributed and Grid Computing (PDGC)*, pages 192–197. IEEE.

[Dencheva 2022] Dencheva, L. (2022). Comparative analysis of static application security testing (sast) and dynamic application security testing (dast) by using open-source web application penetration testing tools. Master's thesis, National College of Ireland, Dublin.

[Havard and Colomo-Palacios 2017] Havard, M. and Colomo-Palacios, R. (2017). Devsecops: a multivocal literature review. In *Software Process Improvement and Capability Determination: 17th International Conference*, pages 17–29, Palma de Mallorca, Spain.

[Hsu 2018] Hsu, T. (2018). *Hands-On Security in DevOps: Ensure Continuous Security, Deployment, and Delivery with DevSecOps*. Packt Publishing.

[Hsu 2019] Hsu, T. H. C. (2019). *Practical security automation and testing: tools and techniques for automated security scanning and testing in devsecops*. Packt Publishing Ltd.

[Inc. a] Inc., G. Container scanning. `https://docs.gitlab.com/ee/user/application_security/container_scanning/`. Acesso em: 15 Jun 2024.

[Inc. b] Inc., G. Secret detection. `https://docs.gitlab.com/ee/user/application_security/secret_detection/`. Acesso em: Jun 15 2024.

[Jammeh 2020] Jammeh, B. (2020). Devsecops: Security expertise a key to automated testing in ci/cd pipeline.

[Kitchenham 2004] Kitchenham (2004). Procedures for performing systematic reviews. Technical Report TR/SE-0401, Keele University, Keele, Eng. Disponível em: `https://www.inf.ufsc.br/~aldo.vw/kitchenham.pdf`. Acesso em: 15 Jul 2024.

[Lenka et al. 2018] Lenka, R. K., Kumar, S., and Mamgain, S. (2018). Behavior driven development: Tools and challenges. In *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, pages 1032–1037.

[Manohar et al. 2023] Manohar, M., Bertia, A., and Salaja, S. (2023). Implementing and automating security scanning. In *2023 World Conference on Communication & Computing (WCONF)*, pages 1–6. IEEE.

[Myrbakken and Colomo-Palacios 2017] Myrbakken, H. and Colomo-Palacios, R. (2017). Devsecops: a multivocal literature review. In *Software Process Improvement and Capability Determination: 17th International Conference*, pages 17–29, Palma de Mallorca, Spain.

[Nikolov and Aleksieva-Petrova 2023] Nikolov, L. A. and Aleksieva-Petrova, A. P. (2023). Action research on the devsecops pipeline. In *2023 International Scientific Conference on Computer Science (COMSCI)*, pages 1–6.

[OWASP 2022] OWASP (2022). Intrusion detection. `https://owasp.org/www-community/controls/Intrusion_Detection`. Acesso em: 15 Jun 2024.

[Oyetoyan et al. 2018] Oyetoyan, D. T. et al. (2018). Myths and facts about static application security testing tools: an action research at telenor digital. In *Agile Processes in Software Engineering and Extreme Programming: 19th International Conference, XP 2018*, pages 86–103, Porto, Portugal.

[Prates and Pereira 2025] Prates, L. and Pereira, R. (2025). Devsecops practices and tools. *International Journal of Information Security*, 24(1):11.

[Rajapakse et al. 2022] Rajapakse, R. N. et al. (2022). Challenges and solutions when adopting devsecops: A systematic review. *Information and software technology*, 141:106700.

[Rajapakse et al. 2021a] Rajapakse, R. N., Zahedi, M., and Babar, M. A. (2021a). An empirical analysis of practitioners' perspectives on security tool integration into devops. In *Proc. 15th ACM / IEEE Inter. Sym, ESEM*. ACM.

[Rajapakse et al. 2021b] Rajapakse, R. N., Zahedi, M., and Babar, M. A. (2021b). An empirical analysis of practitioners' perspectives on security tool integration into devops. In *Proc. 15th ACM / IEEE Inter. Sym, ESEM*. ACM.

[Rangnau et al. 2020] Rangnau, T., Buijtenen, R. V., and Fransen, F. (2020). Continuous security testing: A case study on integrating dynamic security testing tools in ci/cd pipelines. In *2020 IEEE 24th International Enterprise Distributed Object Computing Conference (EDOC)*, pages 145–154, Eindhoven, Netherlands.

[RedHat 2023] RedHat (2023). What is devsecops? `https://www.redhat.com/en/topics/devops/what-is-devsecops`. Acesso em: 15 Jul 2024.

[Sharma 2021] Sharma, M. (2021). Review of the benefits of dast (dynamic application security testing) versus sast. *INTERNATIONAL JOURNAL OF MANAGEMENT AND ENGINEERING RESEARCH*, 1(1).

[Sinan et al. 2025] Sinan, M., Shahin, M., and Gondal, I. (2025). Integrating security controls in devsecops: Challenges, solutions, and future research directions. *Journal of Software: Evolution and Process*.

[Stallings 2017] Stallings, W. (2017). *Attack Surfaces and Attack Trees*. Pearson, 6th edition.

[Tomas et al. 2019] Tomas, N., Li, J., and Huang, H. (2019). An empirical study on culture, automation, measurement, and sharing of devsecops. In *International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, pages 1–8. IEEE.

[Yang et al. 2019] Yang, J. et al. (2019). Towards better utilizing static application security testing. In *IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, Montreal, QC, Canada.

[Yasar and Kontostathis 2016] Yasar, H. and Kontostathis, K. (2016). Where to integrate security practices on devops platform. *International Journal of Secure Software Engineering*, 7:39–50.

A. Artigos Selecionados

A1. KHAMDAMOV, U., USMAN, M., KIM, J. Cost-effective High-throughput Testbed for Supporting AI-enabled DevSecOps Services. In *2023 10th International Conference on Future Internet of Things and Cloud (FiCloud)*. IEEE, 2023, pp. 95–102.

A2. RAMAJ, X. DevSecOps-enabled framework for risk management of critical infrastructures. In *Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Companion Proceedings*. 2022.

A3. BHAMIDIPATI, V. S. A Holistic Approach to Ensure Security and Compliance while using Robotic Process Automation. In *2022 Seventh International Conference on Parallel, Distributed and Grid Computing (PDGC)*. IEEE, 2022, pp. 192–197.

A4. NOCERA, S. et al. A large-scale fine-grained empirical study on security concerns in cloud computing. In *2023 10th International Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE, 2023, pp. 418–425.

A5. VADILAMUDI, S., SAM, J. A Novel Approach to Onboarding Secure Cloud-Native Acquisitions into Enterprise Solutions. In *2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON)*. IEEE, 2021, pp. 228–233.

A6. ALJOHANI, M. A., ALOHAITANI, S. S. A unified framework for automating software security analysis in DevSecOps. In *2023 International Conference on Smart Computing and Application (ICSCA)*. IEEE, 2023, pp. 1–6.

A7. NIKOLOV, L. A., ALEKSEYEVA-PETROVA, A. P. Action Research in the DevSecOps Domain. In *2023 International Scientific Conference on Computer Science (COMSCI)*. IEEE, 2023, pp. 1–6.

A8. SCHWAIGER, P., SIMOPOULOS, D., WOLF, A. Automated IoT security testing with SecLab. In *NOMS 2022-2025 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2022, pp. 1–6.

A9. JIANG, L. et al. Binary AI, Binary Software Composition Analysis via Intelligent Binary Source Code Matching. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*. 2024, pp. 1–13.

A10. BOLISH, S. et al. Building Cyber Resilient Systems from Day 1. In *2024 IEEE Aerospace Conference*. IEEE, 2024, pp. 1–8.

A11. ABRAHAMS, M. Z., LANGERMAN, J. J. Compliance at velocity within a DevOps environment. In *2018 Thirteenth International Conference on Digital Information Management (ICDIM)*. IEEE, 2018, pp. 94–101.

A12. RANGNAU, T. et al. Continuous security testing: A case study on integrating dynamic security testing tools in CI/CD pipelines. In *2020 IEEE 24th International Enterprise Distributed Object Computing Conference (EDOC)*. IEEE, 2020, pp. 145–154.

A13. CHEN, T., SOU, H. Design and Practice of Security Architecture via DevSecOps Technology. In *2022 13th International Conference on Software Engineering and Service Science (ICSESS)*. IEEE, 2022, pp. 310–313.

A14. GRIGOREVA, N. M., PETRENKO, A. S., PETRENKO, S. A. Development of Secure Software Based on the New DevSecOps Technology. In *2024 Conference*

of Young Researchers in Electrical and Electronic Engineering (EICon). IEEE, 2024, pp. 158–161.

A15. IBRAHIM, A., YOUSSEF, A. H., MEDHAT, W. DevSecOps: A Security model for infrastructure as code over the cloud. In *2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*. IEEE, 2022, pp. 284–288.

A16. DAVID, P., KUSHWAHA, M. K., SUSEELA, G. DevSecOps in Finance: Strengthening the Security Model of Applications. In *2022 4th International Conference on Data Engineering and Communication Systems (ICDECS)*. IEEE, 2022, pp. 1–6.

A17. FLORA, J., ANTUNES, N. Doing more with less: A Study on Models for Intrusion Detection in Microservices. In *2024 9th European Dependable Computing Conference (EDCC)*. IEEE, 2024, pp. 9–56.

A18. ALAWNEH, M., ABBAD, I. M. Expanding DevSecOps Practices and Clarifying the Concepts within Kubernetes Ecosystem. In *2022 9th International Conference on Software Defined Systems (SDS)*. IEEE, 2022, pp. 1–7.

A19. THANTHARATE, P., ANDURAG, T. GeneticSecOps: Harnessing Heuristic Genetic HyperSecOps for the Model of the Security. In *2023 IEEE International Conference on Cyber Security and Resilience (CSR)*. IEEE, 2023, pp. 281–288.

A20. MARANDI, M., BERTIA, A., SILAS, A. Implementing and Automating Security Scanning to a DevSecOps CI/CD Pipeline. In *2023 World Conference on Communication & Computing (WCONF)*. IEEE, 2023. 1-6.

A21. AISHWARYA, V. et al. January. Incorporating of Security Methods into the Software Development Lifecycle Process (SDLC). In *2023 International Conference on Computer Communication and Informatics (ICCCI)*. IEEE, 2023. 1-4.

A22. POPENŢIU-VLĂDICESCU, F., ALBEANU, G. Increasing SoS dependability by DevSecOps. In 2022 *International Conference on Emerging Technologies in Electronics, Computing and Communication (ICETECC)*. IEEE, 2022. 1-6.

A23. DO AMARAL, T. M. S.; GONDIM, J. J. C. Integrating Zero Trust in the cyber supply chain security. In *2021 Workshop on Communication Networks and Power Systems (WCNPS)*. IEEE, 2021. 1-6

A24. SOJAN, A., RAJAN, R., KUVAJA, P. Monitoring solution for cloud-native DevSecOps. In *2021 IEEE 6th International Conference on Smart Cloud (SmartCloud)*. IEEE, 2021, pp. 125–131.

A25. CARTURAN, C., GOYA, D. Major Challenges of Systems-of-Systems with Cloud and DevOps: A financial experience report. In *2019 IEEE/ACM 7th International Workshop on Software Engineering for Systems-of-Systems (SESoS) and 13th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems (WDES)*. IEEE, 2019, pp. 10–17.

A26. GOTTEL, C. et al. Qualitative Analysis for Validating IEC 62443-4-2 Requirements in DevSecOps. In *2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2023, pp. 1–8.

A27. NIGMATULLIN, I. et al. ROCODE: Towards Object-Oriented Requirements in the Software Security Domain. In *2022 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. IEEE, 2022, pp. 2–6.

A28. ACHEAMPONG, R. et al. Security scenarios automation and deployment in cloud environments using machine. In *2021 14th International Conference on Communications (COMM)*. IEEE, 2022, pp. 1–7.
A29. BAPAL, P., LEWIS, A. Secure Development Workflows in CI/CD Pipelines. In *2022 IEEE Secure Development Conference (SecDev)*. IEEE, 2022, pp. 59–66.
A30. KONALA, P. R. R., KUMAR, V., BAINBRIDGE, D. SoK: Static Configuration Analysis in Infrastructure as Code Scripts. In *2023 IEEE International Conference on Cyber Security and Resilience (CSR)*. IEEE, 2023, pp. 281–288.
A31. LEE, J. S. The DevSecOps and agency theory. In *2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*. IEEE, 2018, pp. 243–244.
A32. CHEN, M., LIANG, B., LU, X. Practice and Application of a Novel DevSecOps Platform on Security. In *2024 5th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT)*. IEEE, 2024, pp. 558–562.
A33. SOLOMON, A., CRAWFORD, Z. Transitioning from legacy air traffic management to airspace management through secure, cloud-native automation solutions. In *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*. IEEE, 2021, pp. 1–8.
A34. BOSE, D. B., RAHMAN, A., SHAMIM, S. Under-reported security defects in Kubernetes manifests. In *2021 IEEE/ACM 2nd International Workshop on Engineering and Cybersecurity of Critical Systems (EnCyCris)*. IEEE, 2021, pp. 9–12.
A35. BURKARD, E. C. Usability testing within a DevSecOps environment. In *2020 Integrated Communications Navigation and Surveillance Conference (ICNS)*. IEEE, 2020, pp. 1C1-1.
A36. OZKOOK, M. B. et al. Honeypot's Best Friend? Investigating ChatGPT's Ability to Evaluate Honeypot Logs. In *European Interdisciplinary Cybersecurity Conference*. 2024, pp. 128–135.
A37. THOOL, A. J., BROWN, C. Securing Agile: Assessing the Impact of Security Activities on Agile Development. In *Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering*. 2024, pp. 68–78.
A38. MOYON, F., ANGERMEIER, F., MENDEZ, D. Industrial Challenges in Secure Continuous Development. In *Proceedings of the 46th International Conference on Software Engineering: Software Engineering in Practice*. 2024, pp. 30–41.
A39. AKBAR, M. A. et al. Towards People Maturity for Secure Development and Operations: A Vision. In *Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering*. 2024, pp. 228–333.
A40. LYKOUSAS, N., PTSIAKIS, C. Decoding developer password patterns: A comparative analysis of password extraction and selection practices. *Computers & Security*, 2024, 103974.
A41. FUCCI, D. et al. Evaluating software security maturity using OWASP SAMM: A model-based vision and stakeholders perceptions. *Journal of Systems and Software*, 214, 2024, 12026.
A42. CASOLA, V. et al. Secure software development and testing: A model-based methodology. *Computers & Security*, 137, 2024, 103639.