Universidade de Brasília – UnB
Faculdade UnB Gama – FGA
Nome do Curso

Título: Subtítulo do Trabalho

Autor: Nome do Autor
Orientador: Titulação Acadêmica e Nome do Orientador

Brasília, DF
2013

Nome do Autor

Título: Subtítulo do Trabalho

Monografia submetida ao curso de graduação em Nome do Curso da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Nome do Curso.

Universidade de Brasília – UnB

Faculdade UnB Gama – FGA

Orientador: Titulação Acadêmica e Nome do Orientador

Coorientador: quando houver, Titulação Acadêmica e Nome do Orientador

Brasília, DF

2013

Nome do Autor
Título: Subtítulo do Trabalho/ Nome do Autor. – Brasília, DF, 2013-
67 p. : il. (algumas color.) ; 30 cm.

Orientador: Titulação Acadêmica e Nome do Orientador

Trabalho de Conclusão de Curso – Universidade de Brasília – UnB
Faculdade UnB Gama – FGA , 2013.

1. Palavra-chave01. 2. Palavra-chave02. I. Titulação Acadêmica e Nome do Orientador. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Título: Subtítulo do Trabalho

CDU 02:141:005.6

Errata

Elemento opcional da ABNT (2011, 4.2.1.2). **Caso não deseje uma errata, deixar todo este arquivo em branco**. Exemplo:

FERRIGNO, C. R. A. **Tratamento de neoplasias ósseas apendiculares com reimplantação de enxerto ósseo autólogo autoclavado associado ao plasma rico em plaquetas**: estudo crítico na cirurgia de preservação de membro em cães. 2011. 128 f. Tese (Livre-Docência) - Faculdade de Medicina Veterinária e Zootecnia, Universidade de São Paulo, São Paulo, 2011.

Folha	**Linha**	**Onde se lê**	**Leia-se**
1	10	auto-conclavo	autoconclavo

Nome do Autor

Título: Subtítulo do Trabalho

Monografia submetida ao curso de graduação em Nome do Curso da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Nome do Curso.

Trabalho aprovado. Brasília, DF, 01 de junho de 2013 – Data da aprovação do trabalho:

Titulação Acadêmica e Nome do Orientador
Orientador

Titulação e Nome do Professor Convidado 01
Convidado 1

Titulação e Nome do Professor Convidado 02
Convidado 2

Brasília, DF
2013

A dedicatória é opcional. Caso não deseje uma, deixar todo este arquivo em branco.

Este trabalho é dedicado às crianças adultas que,
quando pequenas, sonharam em se tornar cientistas.

Agradecimentos

A inclusão desta seção de agradecimentos é opcional, portanto, sua inclusão fica a critério do(s) autor(es), que caso deseje(em) fazê-lo deverá(ão) utilizar este espaço, seguindo a formatação de *espaço simples e fonte padrão do texto (sem negritos, aspas ou itálico.*

Caso não deseje utilizar os agradecimentos, deixar toda este arquivo em branco.

A epígrafe é opcional. Caso não deseje uma, deixe todo este arquivo em branco.

"Não vos amoldeis às estruturas deste mundo,
mas transformai-vos pela renovação da mente,
a fim de distinguir qual é a vontade de Deus:
o que é bom, o que Lhe é agradável, o que é perfeito.
(Bíblia Sagrada, Romanos 12, 2)

Resumo

O resumo deve ressaltar o objetivo, o método, os resultados e as conclusões do documento. A ordem e a extensão destes itens dependem do tipo de resumo (informativo ou indicativo) e do tratamento que cada item recebe no documento original. O resumo deve ser precedido da referência do documento, com exceção do resumo inserido no próprio documento. (. . .) As palavras-chave devem figurar logo abaixo do resumo, antecedidas da expressão Palavras-chave:, separadas entre si por ponto e finalizadas também por ponto. O texto pode conter no mínimo 150 e no máximo 500 palavras, é aconselhável que sejam utilizadas 200 palavras. E não se separa o texto do resumo em parágrafos.

Palavras-chave: latex. abntex. editoração de texto.

Abstract

This is the english abstract.

Key-words: latex. abntex. text editoration.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

Fig.	Area of the i^{th} component
456	Isto é um número
123	Isto é outro número
lauro cesar	este é o meu nome

Lista de símbolos

Γ	Letra grega Gama
Λ	Lambda
ζ	Letra grega minúscula zeta
$\in$	Pertence

Sumário

1 Introdução

Este capítulo introduz os conceitos fundamentais que norteiam este trabalho: experimentação em Engenharia de Software, segurança de produtos de software e DevSecOps. Adicionalmente, são apresentados o escopo do problema, a questão de pesquisa que guia a investigação e a estrutura geral do documento e das atividades.

1.1 Contexto

Avaliar os fatores de qualidade de um software é de suma importância no desenvolvimento de software e, para isso, faz-se necessária a utilização de um modelo que guie a avaliação, de forma a sistematizar o processo e reduzir subjetividades (SIAVVAS et al., 2021). Nesse sentido, o modelo proposto por McCall (MCCALL; RICHARDS; WALTERS, 1977) foi o primeiro modelo hierárquico para analisar a qualidade, no qual os diferentes fatores eram analisados por critérios e avaliados por métricas. Subsequentemente, o modelo de Boehm (BOEHM, 1978) evoluiu as ideias de McCall, e ambos se tornaram modelos seminais, servindo de referência para os modelos subsequentes.

A partir da ISO/IEC 9126 (2001), estabeleceu-se um padrão internacional para a qualidade de software que decompunha essa qualidade em características e subcaracterísticas, em um modelo hierárquico, além de definir os termos técnicos da área. A ISO/IEC 25010 (2011) surgiu como uma evolução da ISO 9126, expandindo seus conceitos e adaptando o modelo para a nova realidade da qualidade de software moderno. Diferentemente da ISO 9126, na ISO 25010 a segurança já é definida como um dos pilares da qualidade, e não como uma subcaracterística.

Como padrão internacional de segurança da informação, tem-se a ISO/IEC 27001 (2022). Ela se difere das normas de qualidade de software na medida em que seu foco está na especificação dos requisitos necessários para estabelecer, implementar, manter e melhorar continuamente um Sistema de Gestão de Segurança da Informação (SGSI).

Contudo, tanto a ISO/IEC 25010 (2011) quanto a ISO/IEC 27001 (2022) não fornecem um método para avaliar a segurança de software de maneira quantitativa. Assim, torna-se necessário recorrer a outros modelos e ferramentas que ofereçam uma abordagem numérica para medir a segurança, por meio da definição de métricas, do seu cálculo e do estabelecimento de valores de referência para avaliá-las.

O DevOps é um paradigma que visa remover as barreiras entre os times de desenvolvimento e operações, a fim de construir um ambiente colaborativo e integrado (RAJAPAKSE et al., 2021). Seu objetivo é reduzir o ciclo de vida do desenvolvimento de

software, permitindo entregas mais frequentes. As principais práticas de DevOps são a Integração Contínua (CI), que consiste em integrar o código desenvolvido na ramificação principal com validação de build e testes de forma automática para detectar falhas, e a Entrega/Implantação Contínua (CD), que consiste em deixar o software pronto para entrar em produção e realizar o seu lançamento de forma automatizada (RAJAPAKSE et al., 2021).

Já o DevSecOps integra os princípios e práticas do DevOps, adicionando o time de segurança ao processo. Esse paradigma implementa uma abordagem de segurança chamada Shift-Left, na qual os processos de segurança são realizados desde o início do desenvolvimento, com o objetivo de evitar problemas decorrentes de uma avaliação tardia. Além disso é formado por práticas de segurança como treinamento da equipe, testes de segurança automatizados e feedback contínuo (RAJAPAKSE et al., 2021).

Nesse sentido, o modelo de maturidade de segurança OWASP DSOMM (DevSecOps Maturity Model) é uma importante ferramenta na construção e avaliação de projetos DevSecOps. Ele define atividades, métricas e tecnologias que devem ser usadas para construir um ambiente DevSecOps, além de proporcionar o acompanhamento da maturidade da segurança do projeto em cinco dimensões: Build and Deployment, Culture and Organization, Implementation, Information Gathering e Test and Verification (LANGE; KUNZ, 2024).

1.2 Problema

Medir a segurança de software representa um grande desafio(RAJAPAKSE et al., 2021). A literatura atual carece de modelos que apresentem formas sistematizadas de avaliação da segurança; frequentemente, os métodos são baseados em critérios subjetivos, como a análise manual por especialistas, e não possuem validação empírica, o que afeta a confiabilidade dos resultados (SIAVVAS et al., 2021).

No contexto DevOps, esse desafio se torna ainda mais difícil. Métodos tradicionais de análise de segurança são impraticáveis devido à velocidade das entregas (RAJAPAKSE et al., 2021). A medicao da seguranca se torna ainda mais desafiador ao lidar ciclos continuos de lançamento. A segurança é uma propriedade multifacetada, emergente e dependente do contexto, o que complica sua quantificação (KUDRIAVTSEVA; GADYATSKAYA, 2024).

1.3 Questão de Pesquisa

A definição da questão de pesquisa foi elaborada utilizando a abordagem Goal Question Metric (GQM). Essa é uma abordagem que tem como objetivo definir, de ma-

neira top-down e hierárquica, os objetivos a serem alcançados, as perguntas a serem respondidas para cumprir tais objetivos e as métricas necessárias para responder a cada pergunta de forma quantitativa, como mostra a Figura 1. Essa estrutura foi adaptada para o contexto da pesquisa, conforme a Tabela 1, resultando na seguinte questão:

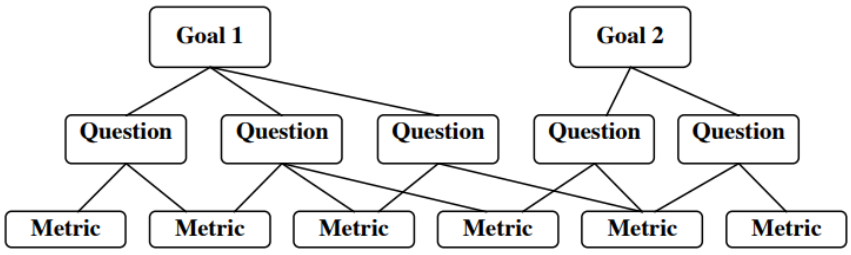
Tabela 1 – GQM Adaptado

Característica	**Valor**
Analisar	A característica de qualidade de produto de software: **segurança**
Subcaracterísticas	Confidencialidade, integridade, autenticidade, responsabilidade, etc.
Visões	Interna e externa
Com o propósito de	Caracterizar
Com respeito a	Desenvolvimento e operação de produtos de software seguros (DevSecOps)
Do ponto de vista de	Pesquisador
No contexto de	Desenvolvimento de aplicações web seguras (software livre, organizações públicas e privadas)

Fonte: Adaptado de BASILI, CALDIERA e ROMBACH (1994)

Como analisar a característica de segurança no desenvolvimento contínuo de sistemas web, considerando as visões de qualidade interna e externa?

Figura 1 – Abordagem GQM



Fonte: Adaptado de BASILI, CALDIERA e ROMBACH (1994)

1.4 Objetivos

O objetivo geral deste trabalho consiste em avaliar o impacto das práticas DevSecOps na qualidade interna e externa de um produto de software, por meio de uma análise quantitativa. Para alcançar este propósito, foram definidos os seguintes objetivos específicos:

- Fundamentar teoricamente os conceitos de DevSecOps, modelos de segurança e metodologias de desenvolvimento seguro.

- Incorporar um conjunto de práticas DevSecOps ao ciclo de desenvolvimento do produto de software sob análise.
- Planejar um estudo de caso focado na observação das práticas implementadas.
- Conduzir o estudo de caso, realizando a coleta e a análise das métricas de qualidade de software.
- Apresentar as conclusões e os insights resultantes desta investigação.

1.5 Estrutura do Trabalho

A seguir, são apresentados os capítulos que compõem a estrutura deste trabalho.

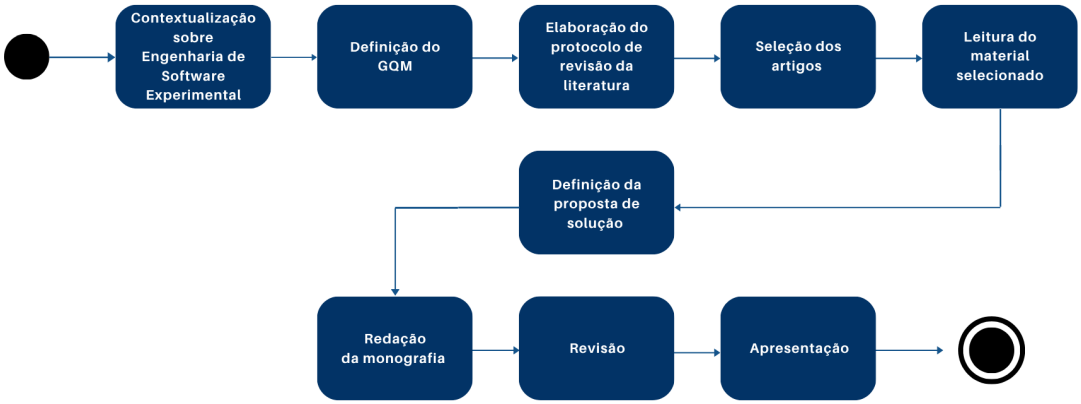
- Introdução: apresenta a contextualização do trabalho, o problema de pesquisa, a definição da questão de pesquisa e dos objetivos do trabalho. Por fim, descreve a estrutura das atividades realizadas.
- Revisão Estruturada da Literatura: explicita o processo empregado para seleção que fundamentam este trabalho incluindo o protocolo de pesquisa e filtragem dos estudos e os resultados obtidos.
- Referencial Teórico: estabelece a fundamentação teórica da monografia, abordando os tópicos centrais da pesquisa: DevSecOps, qualidade de software, segurança de software e modelos de avaliação de maturidade.
- Estudo de caso: descreve o protocolo utilizado para a condução do estudo de caso, detalhando seus objetivos, perguntas de pesquisa, atividades e resultados alcançados.
- Conclusão: consolida os achados obtidos ao final do estudo e como esses resultados respondem à questão de pesquisa principal, bem como as limitações do estudo e as possibilidades de aprofundamento de trabalhos futuros.

1.6 Cronograma e Atividades

1.6.1 Primeira Etapa

Esta subseção detalha as atividades desenvolvidas na primeira etapa da monografia. A Figura 2 ilustra o fluxo das atividades, enquanto a Figura 3 apresenta o cronograma correspondente.

Figura 2 – Atividades da Primeira Etapa



Fonte: Autor

- Contextualização sobre Engenharia de Software Experimental: Estudo sobre os métodos de pesquisa empírica em Engenharia de Software, como revisão sistemática da literatura, survey, experimentos e estudo de caso (WOHLIN et al., 2024), fundamentais para a condução do trabalho.
- Definição do GQM: Aplicação da abordagem Goal Question Metric (GQM) (BASILI; CALDIERA; ROMBACH, 1994) para a construção da questão principal de pesquisa, suas subquestões e as métricas que orientarão a revisão da literatura.
- Elaboração do Protocolo de Revisão: Estruturação de um protocolo de revisão sistemática da literatura (KITCHENHAM; BRERETON, 2013) para pesquisar, selecionar e analisar os artigos. Este processo inclui a definição da string de busca (baseada no framework PICO), a elaboração de sinônimos, a definição dos critérios de inclusão e exclusão e o método para extração de dados.
- Seleção dos Artigos: Execução da filtragem dos estudos por meio da leitura de títulos, resumos e palavras-chave, com a aplicação dos critérios de inclusão e exclusão definidos.
- Análise do Material Selecionado: Leitura completa dos artigos selecionados para aprofundar o conhecimento sobre o estado da arte em métodos de avaliação de segurança de sistemas web e práticas de desenvolvimento seguro.
- Definição da Proposta de Solução: Definição os modelos de segurança, as ferramentas e as atividades do estudo de caso.

Figura 3 – Cronograma da Primeira Etapa

Fonte: Autor

- Redação da Monografia: Escrita do texto da monografia, conforme a estrutura presente na Seção 1.5.
- Revisão: Realização das correções e dos ajustes solicitados pelo orientador.
- Defesa: Preparação do material e apresentação do trabalho final para a banca examinadora.

2 Referencial Teórico

2.1 Engenharia de Software Experimental

2.2 DevSecOps

2.3 Segurança de Software

2.4 Modelos de Qualidade de Software

3 Revisão Estruturada da Literatura

Este capítulo destina-se a documentar o processo realizado para selecionar o conjunto de obras acadêmicas que compõe a bibliografia desta monografia. Para isso, a base de dados Scopus foi escolhida devido à sua característica de indexar diversos artigos da área da computação, muitos publicados nos principais meios de divulgação científica (ELSEVIER, 2025).

3.1 Protocolo

O protocolo utilizado para realizar a revisão estruturada da literatura foi baseado no modelo proposto por Kitchenham e Brereton (2013). Seu objetivo é tornar possível que outros pesquisadores, partindo do mesmo ponto, cheguem aos mesmos resultados, facilitando a replicabilidade em estudos futuros e permitindo a conferência dos resultados obtidos.

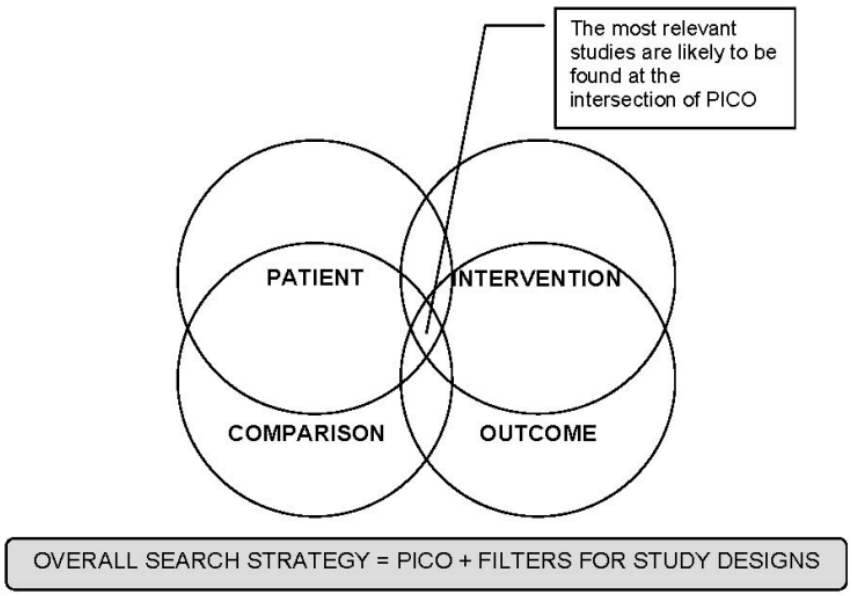
3.1.1 String de Busca

A busca em bases de dados acadêmicas requer o uso de um protocolo, pois elas indexam grande quantidade de artigos de várias áreas. Seu uso incorreto pode acarretar em um número excessivo de artigos sem relação com o tema da pesquisa ou, inversamente, retornar um volume insuficiente de estudos para responder à questão de pesquisa.

Por essa razão, o protocolo PICO foi utilizado para guiar a elaboração de uma string de busca adequada às necessidades da monografia. O protocolo, no entanto, precisou ser adaptado, pois sua origem é na medicina e nem todos os seus elementos se adequam ao nosso escopo. Uma representação visual que facilita a compreensão do protocolo pode ser vista na Figura 4 (PAI et al., 2004).

Ao adaptar o modelo PICO para o presente trabalho, suas definições adquirem um novo significado no contexto da Engenharia de Software. Por exemplo, Patient, outrora usado para indicar o perfil do paciente, passa a representar a área de aplicação, neste caso, o desenvolvimento de software. Intervention também sofre adaptação, deixando de significar "tratamento médico"para se referir à metodologia avaliada. Não obstantemente, Outcome mantém seu sentido original, referindo-se aos efeitos ou consequências observadas. Por fim, Comparison que está relacionada a investigar como a intervenção proposta se relaciona com outras propostas de intervenção não pode ser adotada, devido a estar muito mais alinhada com os objetivos da médicina que da engenharia de software. Assim, a definição de cada um dos elementos usados do modelo estão definidos na Tabela 2.

Figura 4 – Abordagem GQM



Fonte: (PAI et al., 2004)

Desta maneira, a string de busca foi construída usando o operador lógico OR entre os termos de cada elemento, com o objetivo de englobar todos os termos da pesquisa. Já o operador AND foi usado para conectar os diferentes elementos do protocolo PICO, assim restringindo a busca apenas aos estudos que apresentam os termos necessários para responder as perguntas de pesquisa. Ademais, foram adicionados os termos própios da base de dados para a realização da consulta, resultando na seguinte string de busca:

TITLE-ABS-KEY (("software development"OR "software developments"OR "software system"OR "software systems"OR "online system"OR "online systems"OR "software application"OR "software applications"OR "system development"OR "systems development"OR "application development"OR "applications development") AND ("DevSecOps"OR "cybersecurity practice"OR "cybersecurity practices"OR "security automation"OR "security automations"OR "secure software development"OR "secure software developments"OR "CI/CD"OR "continuous integration"OR "continuous integrations"OR "continuous deployment"OR "continuous deployments"OR "continuous delivery"OR "continuous deliveries"OR "DevOps"OR "security development culture"OR "security development cultures") AND ("security quality"OR "security qualities"OR "software security"OR "software securities"OR "application security"OR "application securities"OR "security improvement"OR "security improvements"OR "security assurance"OR "security assurances"OR "vulnerability reduction"OR "vulnerability reductions"OR "protection against threat"OR "protection against threats"OR "system security"OR "system securities"OR "OWASP"OR "CWE"OR "common weakness"OR "common weaknesses"))

Tabela 2 – GQM Adaptado

Elementos	**Termo Central**	**Sinônimos e Termos Relacionados**
Population	software development	software systems, online systems, software applications, systems development, application development
Intervention	DevSecOps	cybersecurity practices, security automation, secure software development, CI/CD, continuous integration, continuous deployment, continuous delivery, DevOps, security development culture
Outcome	security quality	software security, application security, security improvement, security assurance, vulnerability reduction, protection against threats, system security, owasp, cwe, common weakness

Fonte: Autor

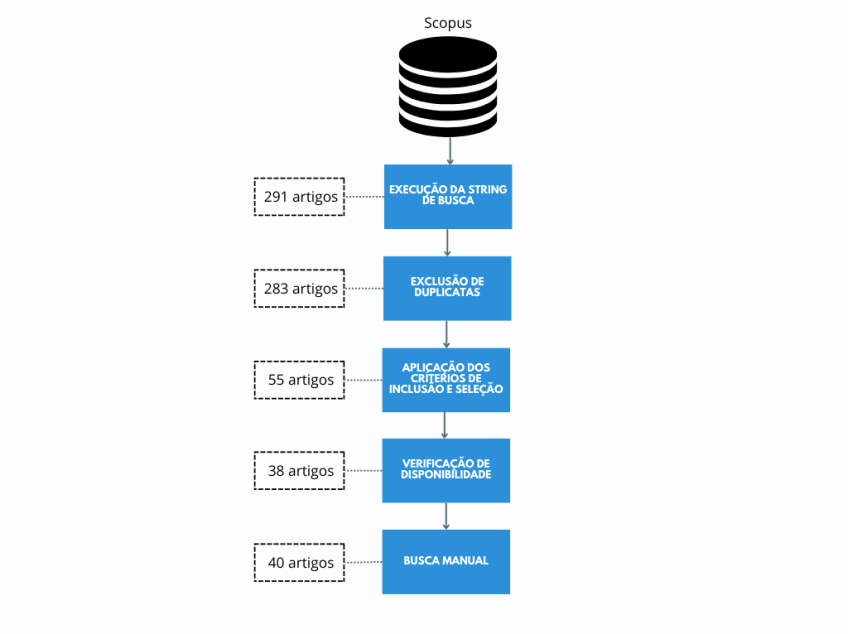
3.2 Seleção dos Artigos

Com a string de busca criada, foram estabelecidos os critérios de inclusão e exclusão visando selecionar apenas os artigos relacionados ao contexto da pesquisa. Além disso, em primeiro momento, foram lidos o título, resumo e palavras-chave de todos os artigos resultantes da execução da string de busca, isso se deu para selecionar com maior rigor aqueles estudos que por ventura não correspondessem ao objetivo do trabalho. Posteriormente, os artigos que aprovados pelos critérios de escolha foram lidos e os dados relevantes para a formulação das respostas das perguntas de pesquisa foram extraídos em um formulário. A Tabela 3 e Tabela 4 contém o protocolo completo.

Ao todo foram analisados 291 artigos resultantes da string de busca, desses 38 foram aceitos e lidos de maneira integral, sendo que esses artigos foram obtidos na base de dados Scopus (ELSEVIER, 2025) no dia 7 de maio de 2025. Afim de complementar os artigos selecionados de forma automatizada foi realizada uma busca manual com o objetivo de responder de maneira mais acertiva as perguntas de pesquisa, resultando em mais dois artigos, Accelerate State of DevOps 2024 (DORA, 2024) e Quantitative DevSecOps Metrics for Cloud-Based Web Microservices (ZHANG; ZHANG, 2024), usados para compor o referencial teórico, totalizando 40 artigos. Os artigos selecionados estão

dispostos nas Tabela 5, 6, 7 e 8. Para facilitar a compreensão do protocolo de seleção de artigos pode-se verificar a Figura 5, que ilustra todas as etapas explanadas anteriormente.

Figura 5 – Seleção dos Artigos



Fonte: Autor

3.3 Resultados

Esta seção apresentará os resultados obtidos com a leitura do material coletado. Durante o processo de leitura, com o objetivo de facilitar a elaboração das respostas da perguntas de pesquisa e para possibilitar a aferição da revisão da literatura, foi construída uma planilha [1] contendo todos os dados extraídos dos artigos, a planilha está separada em vários páginas, pois cada página está relacionada com uma pergunta de pesquisa ou um conjunto de perguntas de pesquisa semelhantes, desse modo, possibilitando que o leitor consiga enxergar como cada artigo responde as perguntas de pesquisa. Assim, com os dados extraídos dos artigos foi possível estruturar o conhecimento para responder as perguntas de pesquisa baseado no estado da arte sobre o tema.

Nas subseções a seguir são apresentadas as respostas para as perguntas de pesquisa.

3.3.1 Which DevSecOps practices are most prevalent in organizations?

- Metodologias e Cultura
  - Shift Security Left: é o principio fundamental do DevSecOps, ele consiste em deslocar as práticas de segurança para o inicio do processo de desenvolvimento

[1] Planilha com o resultado da revisão: <https://docs.google.com/spreadsheets/d/1HdgzzRaP8YlS_08hZhIKaUKV9HKP2dTaq4xWVKFzh84/edit?usp=sharing>

do projeto, em vez de deixar no final como acontece em grande parte dos casos, essa postura eleva a prioridade da segurança no projeto, permitindo que as vulnerabilidades sejam tratadas mais cedo (RAJAPAKSE et al., 2021).

  - Continuos Vulnerability Assessment: Juntamente com Shift Left, a prática Avaliação Contínua de Vulnerabilidades forma os pilares do DevSecOps, nessa prática a segurança do software é continuamente verificada, não somente durante o desenvolvimento, mas também após a implantação do software, pois é necessário verificar a segurança do sistema durante todo o seu ciclo de vida (RAJAPAKSE et al., 2021).
  - CI/CD: Já Continuous Integration (CI) e Continuous Delivery/Deployment (CD) integram o DevSecOps, pois este se trata de uma evolução do DevOps, como esses conceitos são herdados é importante defini-los para completo entendimento da metodologia. CI se refere a prática de realizar continuamente integrações de código na branch principal do código, como essa atividade envolve a alteração da ramificação principal ela é validada por meio de verificadores de build e testes automatizados. Já CD visa a implatação automatica e contínua das atualizações do código no ambiente de produção depois de passar o novo código por verificações de qualidade e, se tudo estiver de acordo com a política de qualidade estabelecida, o código é publicado sem intervenção humana em produção através do pipeline (RAJAPAKSE et al., 2021).
  - Continuous Feedback: Obter feedback de maneira continua e rápida é vital em ambientes de entrega contínua. Nessa abordagem, os problemas são rapidamente identificados, assim, possibilitando que as informações cheguem rápido nas equipes para que as medidas necessárias sejam tomadas. Os métodos tradicionais de coleta de dados/feedback são demasiadamente lentos para a agilidade requisitada em ambientes DevSecOps, essa lentidão afeta a diretamente a velocidade de localização e resolução dos problemas (RAJAPAKSE et al., 2021).

- Testes de Segurança

  - SAST, DAST, IAST: Ferramentas SAST executam testes estáticos, ou seja, eles são realizados apenas sobre o conteúdo do código-fonte, elas podem indicar potenciais vulnerabilidades e práticas ruins, conhecidas como code smells no código. Por outro lado, as ferramentas DAST analisam o software em execução, o que possibilita testar o comportamento do sistema em uso. Ferramentas IAST, entretanto, fornecem uma abordagem híbrida incorporando características da análise estática e dinâmica são modernas e possuem boa integração em ambientes de desenvolvimento contínuo (RAJAPAKSE et al., 2021).

  - Fuzz Testing: o teste de fuzzing, como também é conhecido, se destaca entre as práticas de teste de segurança. Eles consistem em fornecer para a aplicação entradas aleatórias, inválidas e inesperada, de forma a verificar possíveis casos de borda não testados (MASOOD; JAVA, 2015).
  - BDST: os testes BDST são baseados no BDD, porém aplicado ao contexto de textes de segurança. Como os testes descrevem o seu comportamento em linguagem natural, pessoas fora da área de desenvolvimento e segurança de software conseguem compreender os testes realizados (RANGNAU et al., 2020).

- Infraestrutura
  - IaC: essa prática consiste em definir a infraestrutura do projeto como código. É altamente usada em contextos DevSecOps, pois possibilita a configuração rápida da infraestrutura. Como a infraestrutura está definida como código, pode-se realizar o seu versionamneto, teste e implantação de forma ágil e adequado a ambientes complexos que necessitam de segurança robusta (RAJAPAKSE et al., 2021).

- Acompanhamento
  - Monitoring e Logging: muitas vezes o registro e documentação dos eventos relacionados a segurança são negligenciados pelas equipes, porém isso consiste em um grande erro, registrar e monitorar fornece feedback valioso que pode ser utilizado para tomar decisões estratégicas ou realizar auditorias (RAJAPAKSE et al., 2021).

3.3.2 Which security models are most commonly employed in DevSecOps environments?

- OWASP SAMM: o Software Assurance Maturity Model é um modelo prescritivo de maturidade de segurança de software, isso quer dizer que ele contém informa quais atividades precisam ser realizadas, diferentemente de um modelo descritivo que apenas descreve as atividades a serem realizadas. Sua estrutura é adequada para diferentes tipos e tamanhos de empresas, bem como diferentes metodologias de desenvolvimento, como por exemplo cascata e ágil (LANGE; KUNZ, 2024).

- OWASP DSOMM: embora baseado no SAMM, o DevSecOps Maturity Model nasce devido a necessidade de um modelo adequado para ambientes DevOps onde a segurança é parte essencial do ciclo de vida. Também é prescritivo, mas diferentemente do SAMM, as suas atividades estão definidas em um nível mais perto do programador que da gestão, desse modo, ele fornece com detalhes tecnicos os requisitos

necessários para atingir cada um dos níveis de maturidade em suas diferentes dimensões (LANGE; KUNZ, 2024).

- BSIMM: diferentemente dos outros, esse é um modelo descritivo, ou seja, ela descreve atividades sem exigir que elas sejam implementadas. Outra diferença fundamental é que ele é um modelo proprietário, mantido pela Synopsys, que a aplicação do modelo, bem como sua metodologia de avaliação só são disponíveis mediante contratação dos serviços da empresa (LANGE; KUNZ, 2024).

3.3.3 Which measures are commonly used for security evaluation?

- Métricas de Zhang: Segundo Zhang e Zhang (2024) medir de forma eficaz característica de softwares web é fundamental, porém pouco explorado, para resolver este problema eles realizarem uma revisão sistemática da literatura e definiram doze métricas voltadas a atender as necessidades dos sistemas Web que usam DevSecOps como metodologia de desenvolvimento. Elas permitem quantificar o desempenho do serviço, segurança e eficiência da operação, assim apoiando as tomadas de decisão e melhoria contínua das práticas.
  - Non-Comment Lines of Code: tamanho do código de fonte, excluindo comentários e linhas em branco
  - Design Defect Ratio: proporção de defeito de design e linhas de código não comentadas
  - Shared or Unknown Library Ratio: proporção de bibliotecas compartilhadas ou não verificadas em um serviço
  - Technical Debt Ratio: compara o custo de resolução de um débito técnico com o custo total do código-fonte
  - Continuous Deployment Cycles Score:
  - Mean Change Lead Time: tempo médio que uma mudança leva desde o commit até a chegar em produção
  - Mean Time to Recover: tempo médio para recuperação causados por falhas nas pipelines de CI/CD
  - Mean Number of Test Cases Per Parameter: média de testes por parâmetro
  - Points of Environmental Risk: toal de riscos de segurança não resolvidos em produção
  - Time for Response: tempo médio que os times de desenvolvimento levam para solucionar os incidentes de segurança
  - Throughput: mede a capacidade de processamento de um serviço

  - Errors Per Time Unit: taxa de erros em determinada unidade de tempo

- Métricas DORA: O DevOps Research and Assessment é um dos principais programas de pesquisa do mundo na área de DevOps. Esse programa faz parte do Google e depois de anos de estudos eles chegaram em quatro métricas chave para medir os aspectos DevOps de um projeto. As suas métricas passam por uma avaliação estatística rigorosa para possibilitar entender a relação entre as métricas medidas e o sucesso das organizações (DORA, 2024).

  - Change lead time: tempo que uma alteração leva para chegar em produção
  - Deployment frequency: frequência com que as alterações chegam em produção
  - Change fail percentage: porcentagem de implantações que causam falhas em produção
  - Failed deployment recovery time: tempo que leva para se recuperar de uma implatação com falha

3.3.4 How are security practices evaluated?

É fundamental que as práticas de segurança sejam avaliadas para que seja possível avaliar a eficácia das práticas adotadas e conseguir evoluir já existentes ou adotar novas práticas que se adequem mais as necessidades da organização. Sendo assim, encontra-se na avaliação de métricas uma forma eficente e quantidativa de avaliar práticas de segurança, pois elas permitem acompanhar a evolução das atividades realizadas e verificar os resultados de cada atividade concluida. Outra forma de verificar as práticas de segurança é através de autitorias, dessa forma, verificando adequação da organização aos padrões de segurança e conformidade (RAJAPAKSE et al., 2021).

Por fim, pode-se usar modelos de avaliação da maturidade para avaliar as práticas pois eles introduzem uma linha de base para comparar com a organização avaliada. Eles permitem avaliar o estado atual da aplicação, além de identificar áreas de melhória e traçar um plano para alcançar um maior nível de segurança (LANGE; KUNZ, 2024).

3.3.5 In what ways are security measures evaluated in organizations?

As medidas de segurança são avaliadas de diversas formas dentro das organizações. Uma das principais formas de se avalia-las é analisando a repercussão das métricas de segurança nos KPIs da organização. KPIs são indicadores chave de desempenho, são as métricas centrais que medem a saúde dos projetos, é crucial criar alertas e dashboards para acompanhar o desenvolvimento das métricas e KPIs, pois assim, pode-se obter insights de como as métricas de segurança impactam nos KPIs da empresa, além de possibilitar o rastreio das métricas durante todo o periodo em que ela foi monitorada (JOSHI, 2024).

Outra maneira que as medidas de segurança são avaliadas tem a ver com o quanto aquelas medidas se adequam ao compliance da orgranização ou a aspectos regulatórios. Muitas vezes as empresas precisam seguir rigídos padrões de conformidade relacionados as métricas, como por exemplo cobertura de testes, onde dependendo da área precisa ser extremamente alta, bem como áreas financeiras que alguma falha de segurança pode gerar um prejuizo bilionário.

3.3.6 What types of vulnerabilities are mitigated by DevSecOps practices?

Existem diversos tipos de falhas de segurança que podem ocorrer durante o processo de desenvolvimento de software, entre elas, falhas que podem extrapolar o escopo do trabalho, por exemplo falhas relacionadas ao hardware utilizado. Por esse motivo, é necessário entender quais problemas de segurança são afetados pelas práticas DevSecOps, pois, desse modo será possivel analisar de forma mais assertiva a repercussão da adoção dessa metodologia na qualidade da segurança. Assim, na Tabela 9, foram elencadas as principais vulnerabilidades que são impactadas por esse paradigma.

3.3.7 What tools and technologies are used in DevSecOps?

- Monitoramento: Prometheus, Grafana, Loki
- Infraestrutura: Terraform, Kubernetes, Docker
- CI/CD: Jenkins, GitLab CI/CD, GitHub Actions, Tekton, ArgoCD
- Testes: SonarQube, FindBugs, Snyk, OWASP Dependency-Check, OWASP ZAP, Trivy, Detect Secrets, Asylo, StackHawk, JMeter, Selenium

3.3.8 What is the annual volume of DevSecOps publications from 2009 to 2025?

O volume anual de artigos está representado na Figura 6. Observa-se o crescimento das pesquisas sobre o tema principalmente após o ano de 2020, chegando em seu pico em 2024, sendo que em abril de 2025, mês em que a string de busca foi executada, a quantidade de estudos já era ao voluma do ano de 2023 inteiro, o que evidencia o crescimento e importância da área no meio acadêmico e profissional.

3.3.9 How many articles were published in academic journals?

Como indicado na Figura 7, a grande maioria dos artigos foram publicados em revistas científicas, desse modo, sabe-se que a maior parte dos artigos passaram por um

Figura 6 – Volume Anual de Artigos entre 2009 e 2025

Fonte: Autor

crítério alto de revisão e análise de qualidade, elevando o nível de confiablidade dos resultados da pesquisa.

3.3.10 How many studies have experimental validation?

Um total de dezesseis estudos contam com validação experimental de diferentes tipos, a Figura 8 contém o gráfico que ilustra a quantidade de estudos que possuem ou não validação experimental . O baixo índice de validação experimental pode estar relacionado a característica emergente da área, por ser muito recente, ainda faltam ser estabelecidos e consolidados os métodos de pesquisa comumente usados por outras áreas para validação do estudo.

3.3.11 If the study has experimental validation, what type?

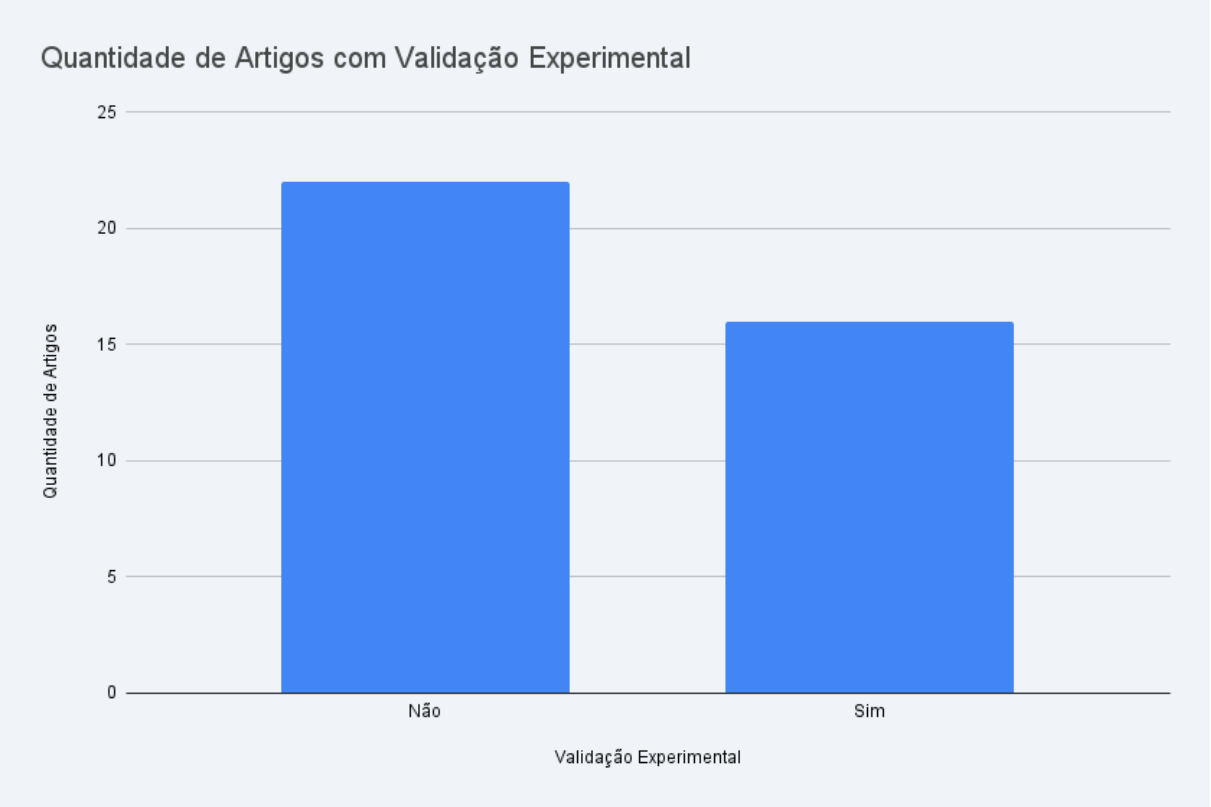
Existem diferentes tipos de validação experimental, dentre eles o estudo de caso, experimento, entrevistas e pesquisas, sendo que esses foram os métodos utilizados na elaboração dos 16 artigos que contém validação experimental, a quantidade de cada tipo de validação experimental está representado no gráfico da Figura 9, dele depreende-se que o a validação usando estudo de caso é, por uma grande margem, o método mais utilizado. Isso se deve principalmente a característica do DevSecOps de estar ligados a muitos projetos reais, tanto na academia quanto na industria, o que cria um ambiente

Figura 7 – Artigos Publicados em Revistas

Fonte: Autor

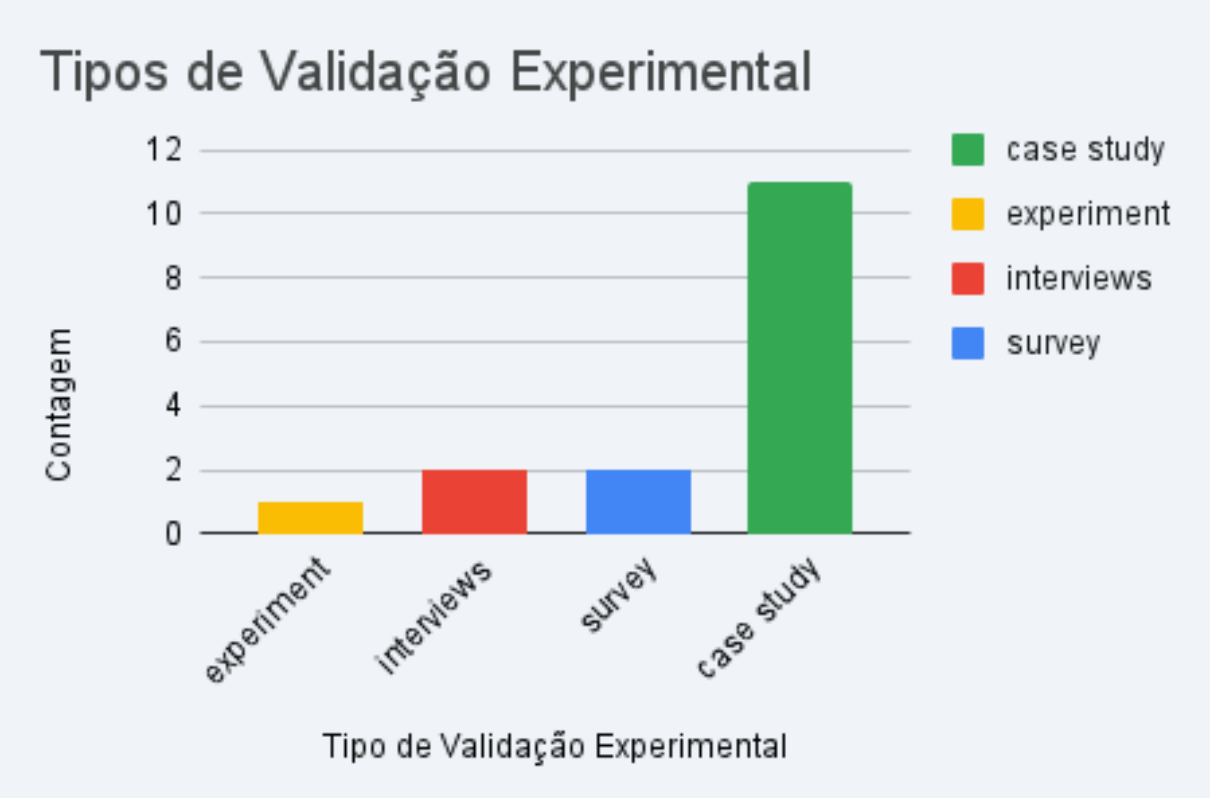
propicio para a aplicação de estudos de caso, pois eles possibilitam analisar e obter insights ao estudar o desenvolvimento realizado durante a construção de um produto.

Figura 8 – Quantidade de Artigos com Validação Experimental



Fonte: Autor

Figura 9 – Tipos de Validação Experimental dos Artigos



Fonte: Autor

Tabela 3 – Protocolo de Busca

Perguntas de Pesquisa	1. How can the security aspect in the continuous development of web systems be analyzed considering internal and external quality perspectives? 2. Which DevSecOps practices are most prevalent in organizations? 3. Which security models are most commonly employed in DevSecOps environments? 4. Which measures are commonly used for security evaluation? 5. How are security practices evaluated? 6. In what ways are security measures evaluated in organizations? 7. What types of vulnerabilities are mitigated by DevSecOps practices? 8. What tools and technologies are used in DevSecOps? 9. What is the annual volume of DevSecOps publications from 2009 to 2025? 10. How many articles were published in academic journals? 11. How many studies have experimental validation? 12. If the study has experimental validation, what type?
String de Busca	Tabela 2
Critérios de Inclusão	1. Addresses the use of secure development practices 2. Emphasizes the security quality of web systems 3. Evaluates quality models with a focus on security 4. Focuses on software products 5. Includes experimental validation

Fonte: Autor

Tabela 4 – Continuação do Protocolo de Busca

Perguntas de Exclusão	1. Articles in languages other than English or Portuguese 2. Duplicate publication 3. Publications with a release date prior to 2009 4. Studies focusing on hardware, mobile, IoT security, or other topics unrelated to web systems.
Formulário de Extração	1. Title 2. Abstract 3. Publication Year 4. Publication Source 5. Authors 6. Keywords 7. Prevalent DevSecOps Practices 8. Security Models Employed 9. Security Evaluation Measures 10. Security Practices Evaluation 11. Security Models Analysis 12. Organizational Security Evaluation 13. Mitigated Vulnerabilities 14. Tools and Technologies 15. Published in Academic Journal 16. Experimental Validation 17. Type of Experimental Validation 18. Secondary Research

Fonte: Autor

Tabela 5 – Artigos Selecionados

N°	**Título**	**Publicado em Revista**	**Validação Experimental**
1	Development of Secure Software Based on the New DevSecOps Technology	Sim	Não
2	Automating Security in a Continuous Integration Pipeline	Não	Não
3	Extensive Review of Threat Models for DevSecOps	Sim	Não
4	Implementing and Automating Security Scanning to a DevSecOps CI/CD Pipeline	Sim	Não
5	Automating Static Code Analysis Through CI/CD Pipeline Integration	Sim	Sim
6	Design and Practice of Security Architecture via DevSecOps Technology	Sim	Sim
7	Implementation of DevSecOps by Integrating Static and Dynamic Security Testing in CI/CD Pipelines	Sim	Não
8	Research of Static Application Security Testing Technique Problems and Methods for Solving Them	Sim	Não
9	A Large-scale Fine-grained Empirical Study on Security Concerns in Open-source Software	Sim	Sim
10	Evolution of secure development lifecycles and maturity models in the context of hosted solutions	Não	Não

Fonte: Autor

Tabela 6 – Continuação dos Artigos Selecionados

N°	**Título**	**Publicado em Revista**	**Validação Experimental**
11	Automation and DevSecOps: Streamlining Security Measures in Financial System	Sim	Não
12	Securing the development and delivery of modern applications	Sim	Não
13	You cannot improve what you do not measure: A triangulation study of software security metrics	Sim	Sim
14	On DevSecOps and Risk Management in Critical Infrastructures: Practitioners' Insights on Needs and Goals	Sim	Sim
15	Container Security in Cloud Environments: A Comprehensive Analysis and Future Directions for DevSecOps	Não	Sim
16	Microservices-based DevSecOps Platform using Pipeline and Open Source Software	Não	Não
17	Securing the Digital Frontier: A Proactive Approach to Software Development	Sim	Não
18	A Secure Software Development Methodology for Enterprise Business Applications	Sim	Sim
19	Building Resilient CICD Pipelines: A DevOps Security-First Framework	Sim	Não
20	Review of Techniques for Integrating Security in Software Development Lifecycle	Não	Não

Fonte: Autor

Tabela 7 – Continuação dos Artigos Selecionados (cont.)

N°	**Título**	**Publicado em Revista**	**Validação Experimental**
21	A hierarchical model for quantifying software security based on static analysis alerts and software metrics	Sim	Sim
22	A preventive secure software development model for a software factory: A case study	Sim	Sim
23	Security impacts of sub-optimal DevSecOps implementations in a highly regulated environment	Sim	Sim
24	A survey and comparison of secure software development standards	Sim	Sim
25	Continuous Security Testing: A Case Study on Integrating Dynamic Security Testing Tools in CI/CD Pipelines	Sim	Sim
26	Infiltrating Security into Development: Exploring the World's Largest Software Security Study	Não	Sim
27	Challenges and solutions when adopting DevSecOps: A systematic review	Sim	Não
28	Systematic Mapping Study on Security Approaches in Secure Software Engineering	Sim	Não
29	Systematic Literature Review on Security Risks and its Practices in Secure Software Development	Sim	Não
30	BP: Security concerns and best practices for automation of software deployment processes: An industrial case study	Sim	Sim

Fonte: Autor

Tabela 8 – Continuação dos Artigos Selecionados (cont.)

N°	**Título**	**Publicado em Revista**	**Validação Experimental**
31	Static analysis for web service security - Tools & techniques for a secure development life cycle	Sim	Não
32	Security characterization for evaluation of software architectures using ATAM	Sim	Sim
33	Software security	Sim	Não
34	Using the ISO/IEC 27034 as reference to develop an application security control library	Sim	Sim
35	Hunting for aardvarks: Can software security be measured?	Não	Não
36	Francois Raynaud on DevSecOps	Sim	Não
37	Integrating application security into software development	Sim	Não
38	Busting a myth: Review of agile security engineering methods	Sim	Não

Fonte: Autor

Tabela 9 – Vulnerabilidades Reduzidas

Vulnerabilidade	**Referência**
SQL Injection	(SAEED et al., 2025)
Command Injection	(RAMIREZ; AIELLO; LINCKE, 2020)
XSS	(SAEED et al., 2025)
XXE	(NOCERA et al., 2023)
Buffer Overflow	(RAMIREZ; AIELLO; LINCKE, 2020)
CSRF	(KUSHWAHA; DAVID; SUSEELA, 2024)
DDoS	(SAEED et al., 2025)
MITM	(NOCERA et al., 2023)
Broken Authentication	(SAEED et al., 2025)
Broken Access Control	(SAEED et al., 2025)
Security Misconfiguration	(SAEED et al., 2025)
Session Hijacking	(KUSHWAHA; DAVID; SUSEELA, 2024)
SSRF	(NOCERA et al., 2023)

Fonte: Autor

4 Planejamento do Estudo de Caso

Este capitulo apresenta

4.1 Protocolo

Referências

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *NBR 14724*: Informação e documentação — trabalhos acadêmicos — apresentação. Rio de Janeiro, 2011. 15 p. Citado na página 3.

Apêndices

APÊNDICE A – Primeiro Apêndice

Texto do primeiro apêndice.

APÊNDICE B – Segundo Apêndice

Texto do segundo apêndice.

Anexos

ANEXO A – Primeiro Anexo

Texto do primeiro anexo.

ANEXO B – Segundo Anexo

Texto do segundo anexo.