



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE ITAPAJÉ
CURSO DE GRADUAÇÃO EM SEGURANÇA DA INFORMAÇÃO

CARLOS ADRIEL SOUSA BASTOS

**CARACTERIZAÇÃO DE PRÁTICAS DE *APPSEC* E DA ADOÇÃO DE
FERRAMENTAS AUTOMATIZADAS DE SEGURANÇA: UM QUESTIONÁRIO
ABRANGENTE COM DESENVOLVEDORES BRASILEIROS**

ITAPAJÉ

2026

CARLOS ADRIEL SOUSA BASTOS

CARACTERIZAÇÃO DE PRÁTICAS DE *APPSEC* E DA ADOÇÃO DE
FERRAMENTAS AUTOMATIZADAS DE SEGURANÇA: UM QUESTIONÁRIO
ABRANGENTE COM DESENVOLVEDORES BRASILEIROS

Trabalho de Conclusão de Curso apresentado
ao Curso de Graduação em Segurança da Infor-
mação do Campus de Itapajé da Universidade
Federal do Ceará, como requisito parcial à
obtenção do grau de Tecnólogo em Segurança
da Informação.

Orientador: Prof. Dr. Anderson Gonçal-
ves Uchôa

Coorientadora: Profa. Dra. Carla Ilane
Moreira Bezerra

ITAPAJÉ

2026

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

B327c Bastos, Carlos Adriel Sousa.

Caracterização de Práticas de AppSec e da Adoção de Ferramentas Automatizadas de Segurança: Um Questionário Abrangente com Desenvolvedores Brasileiros / Carlos Adriel Sousa Bastos. – 2026.
160 f. : il.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Itapajé, Curso de Segurança da Informação, Fortaleza, 2026.

Orientação: Prof. Dr. Anderson Gonçalves Uchôa..

Coorientação: Prof. Dr. Carla Ilane Moreira Bezerra.

1. AppSec. 2. Segurança de Software. 3. Ferramentas Automatizadas. 4. Desenvolvedores. I. Título.
CDD 005.8

CARLOS ADRIEL SOUSA BASTOS

CARACTERIZAÇÃO DE PRÁTICAS DE *APPSEC* E DA ADOÇÃO DE
FERRAMENTAS AUTOMATIZADAS DE SEGURANÇA: UM QUESTIONÁRIO
ABRANGENTE COM DESENVOLVEDORES BRASILEIROS

Trabalho de Conclusão de Curso apresentado
ao Curso de Graduação em Segurança da Infor-
mação do Campus de Itapajé da Universidade
Federal do Ceará, como requisito parcial à
obtenção do grau de Tecnólogo em Segurança
da Informação.

Aprovada em: 09 de Janeiro de 2026

BANCA EXAMINADORA

Prof. Dr. Anderson Gonçalves Uchôa (Orientador)
Universidade Federal do Ceará (UFC)

Profa. Dra. Carla Ilane Moreira
Bezerra (Coorientadora)
Universidade Federal do Ceará (UFC)

Prof. Dr. Solon Alves Peixoto
Universidade Federal do Ceará (UFC)

Prof. Dr. Tarek Sayjari
Universidade Federal do Ceará (UFC)

José Eric Mesquita Coelho
Universidade Federal do Ceará (UFC)

Caio Barbosa Vieira da Silva
Universidade Federal de Alagoas (UFAL)

À minha família, por sua capacidade de acreditar em mim e investir em mim. Mãe, seu cuidado e dedicação foram o que me deram, em alguns momentos, a esperança para seguir. Pai, sua presença significou segurança e certeza de que não estou sozinho nessa caminhada.

AGRADECIMENTOS

Dedico este trabalho aos meus pais, André Bastos e Adriana Bastos, meus maiores incentivadores. Agradeço imensamente por todo o sacrifício e investimento que fizeram para que eu pudesse me dedicar aos estudos. O suporte emocional de vocês foi a base que me sustentou durante toda a graduação. Sou eternamente grato por me guiarem até aqui.

Ao meu orientador, professor Anderson Uchôa, expresso minha profunda gratidão pela oportunidade e pela confiança depositada em mim para a realização deste estudo. Agradeço imensamente pela paciência em ensinar e pelo nível de exigência imposto. Compreendo hoje que cada cobrança e cada momento de tensão foram reflexos de sua preocupação genuína com a excelência do meu trabalho e com a minha formação profissional. Agradeço também às professoras Carla Ilane e Joanna Cecilia da Silva Santos pelo direcionamento e contribuições feitas para a elaboração deste trabalho.

Agradeço imensamente aos meus amigos: Aline Kelly, Alexandre Lima, Danilo Soares, Helen Brandão e Moniky Mendonça. A presença e a alegria de vocês foram o refúgio necessário nos momentos de cansaço e o principal motivo para eu me manter firme e disposto a enfrentar a rotina acadêmica. Destaco minha gratidão especial: à Aryely Matos, pela parceria incansável, trocas de conhecimento e apoio mútuo nos estudos; ao Luís Carlos, pela convivência diária e pela irmandade compartilhada sob o mesmo teto durante esses três anos; ao Ruan Lima, por sempre me inspirar e manter acesa minha paixão pela tecnologia. Vocês foram uma peça fundamental em minha vida durante esses anos e se tornaram pessoas muito importantes para mim.

Agradeço ao Programa de Educação Tutorial (PET) pela oportunidade de ter sido bolsista. O suporte financeiro provido pelo programa foi decisivo para a minha permanência e dedicação integral à graduação. Estendo minha gratidão ao excelentíssimo professor João Corrêa, com quem tive a honra de trabalhar; sua orientação foi um exemplo

de excelência. Agradeço também pelas valiosas habilidades técnicas e interpessoais que desenvolvi e, principalmente, pelas pessoas incríveis que conheci e com as quais criei laços durante esse período.

À Universidade Federal do Ceará (UFC), Campus de Itapajé, agradeço pelo acolhimento e por ter me fornecido a base sólida de conhecimentos que me tornaram o profissional que sou hoje. Por fim, agradeço ao Laboratório de P&D em Ciência do Software e Integração de Dados (ResetLab), em especial aos membros Mário Patrício, Silas Eufrásio e Kevin Pontes, o suporte e a infraestrutura de vocês foram vitais, e sem o apoio do laboratório, a realização deste estudo não teria sido possível.

“Siga em frente. Recuar, jamais!”

(LeBlanc, League of Legends)

RESUMO

A adoção de práticas de segurança em aplicações (*Application Security* – AppSec) é fundamental para proteger sistemas contra vulnerabilidades e ameaças cibernéticas. Entretanto, a efetividade dessas práticas depende fortemente de sua adoção consistente e da percepção dos desenvolvedores em relação à sua facilidade de uso, efetividade, essencialidade e aplicabilidade no contexto do desenvolvimento de software. Este estudo visa caracterizar um conjunto de práticas de AppSec quanto à frequência de uso, facilidade de aplicação, efetividade percebida e essencialidade percebida no desenvolvimento de software. Além disso, busca-se analisar como as ferramentas automatizadas de segurança vêm sendo incorporadas ao processo de desenvolvimento e quais desafios os desenvolvedores brasileiros enfrentam em sua adoção. Para atingir esses objetivos, foi inicialmente conduzido um levantamento sistemático de trabalhos relacionados na literatura científica, a partir do qual foi identificado e catalogado um conjunto inicial de práticas de segurança de software. Esse conjunto passou por um processo iterativo de filtragem, refinamento e agrupamento, conduzido por especialistas em Engenharia de Software e Segurança de Software, com o objetivo de eliminar redundâncias, harmonizar terminologias e garantir relevância prática e conceitual. Como resultado, foi definido um conjunto final de 16 práticas de AppSec. Em seguida, foi elaborado e aplicado um questionário direcionado a desenvolvedores de software no Brasil, totalizando 83 participantes válidos. O questionário abordou a experiência dos participantes com segurança de software, a adoção e percepção de práticas de AppSec e o uso de ferramentas automatizadas. Os dados revelam um cenário heterogêneo de adoção, com variações significativas entre práticas quanto à frequência, facilidade, efetividade e essencialidade percebidas. Práticas como revisão de código e uso de ferramentas estáticas são amplamente utilizadas, enquanto outras, como modelagem de ameaças e testes de penetração, apresentam menor aplicabilidade. Além disso, ferramentas automatizadas são vistas como valiosas, mas desenvolvedores enfrentam desafios como curva de aprendizado, falsos positivos e integração ao fluxo

de trabalho. Os achados contribuem para o entendimento sobre o uso real de práticas e ferramentas de segurança no desenvolvimento de software no Brasil. Espera-se que os resultados subsidiem ações educacionais e estratégias organizacionais para fortalecer a cultura de segurança entre desenvolvedores e promover a adoção eficaz de práticas de AppSec.

Palavras-chave: AppSec; segurança de software; ferramentas automatizadas; desenvolvedores.

ABSTRACT

The adoption of Application Security (AppSec) practices is essential to protect software systems against vulnerabilities and cyber threats. However, the effectiveness of these practices strongly depends on their consistent adoption and on developers' perceptions regarding their ease of use, perceived effectiveness, essentiality, and applicability within the software development context. This study aims to characterize a set of AppSec practices in terms of frequency of use, ease of application, perceived effectiveness, and perceived essentiality in software development. In addition, it seeks to analyze how automated security tools are being incorporated into the development process and to identify the main challenges faced by Brazilian developers in their adoption. To achieve these objectives, we first conducted a survey of related work in the scientific literature, from which an initial and comprehensive set of software security practices was identified and cataloged. This initial set then underwent an iterative process of filtering, refinement, and grouping, carried out by experts in Software Engineering and Software Security, intending to reduce redundancy, harmonize terminology, and ensure both conceptual and practical relevance. As a result, a final set of 16 AppSec practices was defined. Subsequently, a questionnaire was designed and administered to software developers working in Brazil, yielding a total of 83 valid participants. The questionnaire collected information about participants' profiles, their experience with software security, their adoption and perceptions of AppSec practices, and their use of automated security tools. The results reveal a heterogeneous landscape of AppSec practice adoption, with significant variations across practices in terms of frequency of use, ease of application, perceived effectiveness, and perceived essentiality. Practices such as code review and the use of static analysis tools show widespread adoption, whereas others, such as threat modeling and penetration testing, exhibit lower applicability. Moreover, although automated security tools are widely perceived as valuable, developers report recurring challenges related to learning curves, false positives, and integration into

existing development workflows. The findings contribute to a deeper understanding of the actual use of security practices and tools in software development within the Brazilian context. The results are expected to inform educational initiatives and organizational strategies aimed at strengthening developers' security culture and promoting a more consistent and effective adoption of AppSec practices.

Keywords: AppSec; software security; automated tools; developers.

LISTA DE FIGURAS

Figura 1 – Integração dos <i>Security Touchpoints</i> ao ciclo de desenvolvimento de software.	30
Figura 2 – Etapas e procedimentos metodológicos.	45
Figura 3 – Distribuição dos participantes por estados brasileiros	67
Figura 4 – Tipos de sistemas	69
Figura 5 – Domínios de sistemas	70
Figura 6 – Distribuição dos participantes por setor de atuação	71
Figura 7 – Distribuição dos participantes por tamanho da empresa	71
Figura 8 – Distribuição das práticas de AppSec nas categorias	84
Figura 9 – Fases do ciclo de desenvolvimento com maior uso de ferramentas automatizadas de segurança	108
Figura 10 – Mapa de calor dos desafios na automação de ferramentas de segurança, por grupo de respondentes.	113
Figura 11 – Melhorias que os participantes gostariam de ver nas ferramentas. . .	115

LISTA DE TABELAS

Tabela 1 – Comparação da frequência de uso, facilidade de uso, efetividade e essencialidade das práticas de AppSec.	74
Tabela 2 – Correlação de Spearman entre frequência de uso, facilidade de uso, efetividade e essencialidade das práticas de AppSec.	78
Tabela 3 – Resultados da influência de experiência, função e porte organizacional nas percepções sobre práticas de AppSec	95
Tabela 4 – Ferramentas automatizadas de segurança agrupadas por categoria, frequência de uso e quantidade de menções	105

LISTA DE QUADROS

Quadro 1 – Comparação entre estudos relacionados e o presente trabalho	40
Quadro 2 – Questões do questionário	47
Quadro 3 – Hipóteses formuladas para a análise de correlação entre atributos das práticas de AppSec	77
Quadro 4 – Práticas adicionais de segurança de software aplicadas pelos partici- pantes	86
Quadro 5 – Formas de implementação das práticas de segurança de software (Q19)	88
Quadro 6 – Recomendações para adoção de ferramentas de segurança segundo os participantes	110
Quadro 7 – Razões para não utilizar ferramentas automatizadas de segurança segundo os participantes	117

LISTA DE ABREVIATURAS E SIGLAS

AppSec	<i>Application Security</i>
CVP	Ciclo de Vida do Produto
DAST	<i>Dynamic Application Security Testing</i>
DevOps	Desenvolvimento e Operação
DevSecOps	Desenvolvimento, Segurança e Operação
GT	<i>Grounded Theory</i>
LGPD	Lei Geral de Proteção de Dados
OWASP	<i>Open Worldwide Application Security Project</i>
SANS	<i>SysAdmin, Audit, Network, and Security</i>
SAST	<i>Static Application Security Testing</i>
SCA	<i>3rd Party Vulnerability Scanning</i>
SDLC	Ciclo de Vida de Desenvolvimento de Software
SGSI	Sistemas de Gestão de Segurança da Informação
TCLE	Termo de Consentimento Livre e Esclarecido

LISTA DE SÍMBOLOS

$W(i)$	É o peso total atribuído ao participante i
$E(i)$	É o nível acadêmico mais alto do participante i
$S(i)$	É o nível de senioridade relatado pelo participante i em sua posição atual
$C(i)$	Indica a quantidade de certificações válidas relacionada à segurança da informação que o participante i possui
$Y(i)$	É o número de anos de experiência relatado pelo participante i em desenvolvimento de software
$Z(i)$	É o número de anos de experiência relatado pelo participante i em segurança de software
$\text{Mediana}(Y)$	É a mediana dos anos de experiência em desenvolvimento, considerando as respostas de todos os participantes
$\text{Mediana}(Z)$	É a mediana dos anos de experiência em segurança de software, considerando as respostas de todos os participantes
$R(i)$	Representa o valor da relação entre experiência direta e especialização em segurança de software de software
$D(i)$	É o valor da relação entre experiência direta e especialização em segurança de software
$X(i)$	Denota a experiência do desenvolvedor em segurança de software, atribuindo um valor de 1 se o desenvolvedor for um especialista em segurança de software
$N(j)$	É o valor normalizado para a frequência de uso ou facilidade de uso ou efetividade ou essencialidade, de uma prática de segurança de software j ;

$T(j)$	É o valor total calculado para a prática j na Etapa 2
$W(i)$	É o valor de peso para o participante i;
n	Número total de participantes da pesquisa.
✔	Refere-se ao atributo de efetividade
🕒	Refere-se ao atributo de frequência
😊	Refere-se ao atributo de facilidade
★	Refere-se ao atributo de essencialidade
🎓	Refere-se a classificação de especialização
🏭	Refere-se a classificação do porte organizacional
🔧	Refere-se a classificação de experiência

SUMÁRIO

1	INTRODUÇÃO	21
1.1	Objetivos	23
1.1.1	<i>Objetivo Geral</i>	24
1.1.2	<i>Objetivos Específicos</i>	24
1.1.3	<i>Questões de Pesquisa</i>	25
1.2	Contribuições do Estudo	25
1.3	Organização do Trabalho	27
1.4	Resumo do Capítulo 1	27
2	FUNDAMENTAÇÃO TEÓRICA	28
2.1	Desenvolvimento de Software Seguro	28
2.2	Práticas de Segurança de Software (AppSec)	29
2.3	Ferramentas Automatizadas de Segurança	31
2.4	Resumo do Capítulo 2	33
3	TRABALHOS RELACIONADOS	34
3.1	Estudos em Contextos Organizacionais e Culturais	34
3.2	Estudos sobre Segurança de Software em Ambientes Ágeis e Iterativos	36
3.3	Estudos sobre Automação e Integração de Segurança em DevOps	37
3.4	Estudo sobre Cultura e Mensuração de Segurança de Software	39
3.5	Análise Comparativa	39
3.6	Resumo do Capítulo 3	40
4	METODOLOGIA	42
4.1	Objetivo do Estudo e Questões de Pesquisa	42
4.2	Etapas da Metodologia do Estudo	44
4.2.1	<i>Etapas da Metodologia do Estudo</i>	44

4.2.2	<i>Etapa 2: Identificação da População-alvo e Amostragem</i>	45
4.2.3	<i>Etapa 3: Elaboração do Instrumento de Pesquisa</i>	47
4.2.4	<i>Etapa 4: Validação do Instrumento de Pesquisa</i>	52
4.2.5	<i>Etapa 5: Administração do Questionário</i>	53
4.2.6	<i>Etapa 6: Condução da Análise de Dados</i>	54
4.2.6.1	<i>Análise para Responder à QP₁</i>	55
4.2.6.2	<i>Análise para Responder à QP₂</i>	61
4.2.6.3	<i>Análise para Responder à QP₃</i>	64
4.3	Resumo do Capítulo 4	64
5	RESULTADOS	66
5.1	Caracterização Geral dos Participantes	66
5.2	Percepção dos Desenvolvedores sobre Práticas de AppSec (QP1) .	73
5.3	Influência de Experiência, Função e Porte Organizacional nas Percepções sobre Práticas de AppSec (QP2)	94
5.4	Benefícios, Adoção e Desafios das Ferramentas Automatizadas de Segurança no Desenvolvimento de Software (QP3)	103
5.5	Resumo do Capítulo 5	120
6	AMEAÇAS À VALIDADE DO ESTUDO	121
6.1	Resumo do Capítulo 6	126
7	CONCLUSÕES E TRABALHOS FUTUROS	127
	REFERÊNCIAS	129
	APÊNDICES	135
	APÊNDICE A – Termo de Consentimento Livre e Esclarecido (TCLE)	135
	APÊNDICE B – Script do Questionário	138

1 INTRODUÇÃO

A crescente complexidade dos sistemas e o aumento de ameaças cibernéticas têm pressionado organizações de diferentes setores a adotar práticas robustas de segurança ao longo de todo o ciclo de desenvolvimento de seus sistemas (SCHNEIER, 2018). Exigências regulatórias, requisitos de mercado e a necessidade de proteger informações sensíveis tornam indispensável a construção de produtos mais seguros, com mecanismos que previnam, detectem e mitiguem vulnerabilidades de forma mais proativa (MALATJI, 2023; FORCE, 2020). No Brasil, e.g., além da Lei Geral de Proteção de Dados (LGPD), que reforça obrigações de proteção de dados, órgãos governamentais e setores regulados têm adotado diretrizes mais rigorosas de segurança, exigindo que empresas adotem processos formais de gestão de riscos e proteções técnicas adequadas (BRASIL, 2018).

Diante desse contexto, desenvolvedores são hoje desafiados não apenas a implementar funcionalidades, mas também a incorporar práticas de segurança como parte natural do fluxo de trabalho – uma demanda alinhada ao princípio de *Security by Design* (HOWARD; LIPNER, 2006) e ao movimento de *Shift Left* (RAJAPAKSE *et al.*, 2022). Apesar disso, estudos têm mostrado que muitos desenvolvedores ainda carecem de conhecimento adequado sobre segurança, interpretam de forma limitada princípios fundamentais e apresentam dificuldade em aplicar certas práticas consideradas essenciais, como modelagem de ameaças, análise de vulnerabilidades e codificação segura (ASSAL; CHIASSON, 2019; RAHMAN; WILLIAMS, 2016; TAHAEI; VANIEA, 2019). Essa falta de conhecimento pode levar a decisões equivocadas de projeto, implementação insuficiente de controles e adoção limitada de práticas preventivas.

Além das práticas manuais, ferramentas automatizadas de segurança, como *Static Application Security Testing* (SAST), *Dynamic Application Security Testing* (DAST), *3rd Party Vulnerability Scanning* (SCA), *Secrets Management* e *Configuration/Compliance Scanners*, tornaram-se componentes centrais no desenvolvimento de qualquer

sistema (ANGERMEIR *et al.*, 2021). No entanto, sua adoção de forma integrada no ambiente de desenvolvimento depende de fatores como facilidade de uso, nível de ruído (falsos positivos), custo de adoção, compatibilidade tecnológica e do contexto organizacional no qual os desenvolvedores estão inseridos (RAJAPAKSE *et al.*, 2022).

Do ponto de vista da pesquisa em segurança de software, há um corpo crescente de estudos que busca compreender como práticas e ferramentas de segurança são adotadas em diferentes realidades. Trabalhos conduzidos em países como Bangladesh e Palestina revelam impactos de fatores culturais e estruturais sobre a adoção da segurança (SHRESTHA *et al.*, 2023; OMAR *et al.*, 2022). Estudos em ambientes corporativos específicos, como o caso da IBM (MORRISON *et al.*, 2017), demonstram como maturidade organizacional e suporte institucional moldam o uso de práticas.

Pesquisas em escala internacional mostram que práticas amplamente difundidas podem não ser necessariamente percebidas como eficazes ou fáceis de integrar a fluxos ágeis (RINDELL *et al.*, 2021), e que barreiras humanas e organizacionais continuam determinantes (ASSAL; CHIASSON, 2019). Revisões da literatura sobre Desenvolvimento, Segurança e Operação (DevSecOps) destacam dificuldades técnicas de automação, problemas de integração de ferramentas, falsos positivos, falta de padronização e desafios culturais (RAJAPAKSE *et al.*, 2022). Por fim, estudos globais com mais de mil desenvolvedores evidenciam uma dissociação entre prática técnica e cultura organizacional (RYAN *et al.*, 2023).

Apesar dessas contribuições, observa-se uma carência de estudos que buscam compreender de forma sistemática como desenvolvedores percebem, aplicam e priorizam práticas de segurança e ferramentas automatizadas em seu cotidiano, especialmente em países em desenvolvimento, onde maturidade da organização de software, disponibilidade de recursos e cultura de segurança podem variar substancialmente. Portanto, entender essas dinâmicas é fundamental, pois a segurança não se limita à disponibilidade de recursos técnicos, mas é também influenciada por fatores individuais e organizacionais.

No contexto brasileiro, esse cenário ganha ainda mais relevância. Empresas locais enfrentam um ambiente regulatório em transformação, crescentes exigências de conformidade e aumento significativo de incidentes de segurança. Além disso, muitas organizações ainda carecem de infraestrutura adequada, políticas consolidadas e programas contínuos de capacitação, o que reforça a importância de estudos empíricos voltados especificamente à realidade brasileira. Assim como ocorreu em pesquisas internacionais sobre privacidade e segurança (SHRESTHA *et al.*, 2023; OMAR *et al.*, 2022; MORRISON *et al.*, 2017; VENSON *et al.*, 2019; ASSAL; CHIASSON, 2019; RYAN *et al.*, 2023; MATOS *et al.*, 2025), compreender a perspectiva do desenvolvedor brasileiro permite identificar desafios particulares e oportunidades para o avanço das práticas de *Application Security* (AppSec).

Portanto, este estudo busca avançar esse entendimento por meio de uma investigação empírica com desenvolvedores brasileiros com experiência profissional relacionada à segurança de software. O questionário desenvolvido examina: (i) como diferentes práticas de segurança são percebidas em termos de frequência de uso, facilidade, efetividade e essencialidade; (ii) como características individuais e organizacionais influenciam tais percepções; e (iii) como ferramentas automatizadas de segurança são utilizadas, incluindo desafios, benefícios e razões para adoção ou não adoção. Ao direcionar o questionário ao contexto brasileiro, este trabalho fornece evidências inéditas sobre o estado atual da segurança de software na prática, oferecendo evidências para a comunidade acadêmica, equipes de desenvolvimento e interessados em fortalecer a maturidade de AppSec em seus ambientes de desenvolvimento.

1.1 Objetivos

A segurança de software, embora amplamente reconhecida como essencial, ainda apresenta lacunas relacionadas à adoção sistemática de práticas seguras e ao uso eficaz de ferramentas automatizadas (AppSec). Muitas vezes, essas práticas são

aplicadas de forma reativa, pouco frequente ou restrita a fases tardias do desenvolvimento. Como consequência, esse comportamento pode aumentar riscos, retrabalho e custos. Assim, compreender como desenvolvedores brasileiros percebem e aplicam tais práticas é um passo essencial para apoiar iniciativas de capacitação, melhorar processos de desenvolvimento de software e orientar organizações na integração efetiva da segurança ao seu fluxo de desenvolvimento. Assim, nas Subseções 1.1.1 e 1.1.2 são apresentados, respectivamente, o objetivo geral e os objetivos específicos que orientaram este estudo.

1.1.1 Objetivo Geral

Caracterizar as práticas de segurança de software e avaliar como e por que ferramentas automatizadas de segurança são utilizadas, considerando frequência, facilidade de uso, efetividade e essencialidade percebidas por desenvolvedores brasileiros.

1.1.2 Objetivos Específicos

A fim de operacionalizar o objetivo geral e aprofundar a compreensão sobre diferentes aspectos das práticas de segurança de software e do uso de ferramentas automatizadas de segurança, este estudo foi conduzido a partir dos seguintes objetivos específicos:

- a) Compreender como os desenvolvedores percebem diferentes práticas de segurança de software, considerando sua frequência de uso, efetividade, facilidade de uso e essencialidade;
- b) Identificar quais ferramentas automatizadas de segurança são mais utilizadas, bem como o estágio do ciclo de desenvolvimento em que são aplicadas;
- c) Comparar práticas de segurança de software quanto às suas dimensões perceptivas, identificando lacunas e oportunidades de melhoria;
- d) Investigar diferenças nas percepções sobre práticas de segurança entre grupos

de desenvolvedores com diferentes níveis de experiência, funções profissionais e portes organizacionais; e

- e) Identificar desafios e barreiras associados ao uso (ou não uso) de ferramentas automatizadas de segurança.

1.1.3 Questões de Pesquisa

Com o propósito de atingir os objetivos desta pesquisa, propomos as seguintes Questões de Pesquisa (QP):

- a) **QP1:** Como as percepções dos desenvolvedores sobre frequência, efetividade, facilidade de uso e essencialidade diferem entre várias práticas de segurança de software?;
- b) **QP2:** Existem diferenças significativas nas percepções sobre as práticas de segurança entre grupos de desenvolvedores com diferentes níveis de experiência, funções e tipos de organização?;
- c) **QP3:** Como as ferramentas automatizadas de segurança são adotadas no desenvolvimento de software e quais desafios os desenvolvedores enfrentam em seu uso?

1.2 Contribuições do Estudo

Os dados e resultados apresentados neste Trabalho de Conclusão de Curso são derivados de um estudo inédito sobre práticas de segurança de software e adoção de ferramentas automatizadas de segurança por desenvolvedores brasileiros. Todo o conjunto de artefatos, incluindo o questionário, os *scripts* de análise, gráficos e o *codebook* utilizado na categorização das respostas abertas, foi disponibilizado em um repositório público no GitHub <<https://github.com/reset-ufc/appsec-practices-characterization-survey>>, assegurando transparência, reprodutibilidade e apoio a futuras investigações na área de

Engenharia de Software e AppSec.

As principais contribuições deste estudo incluem: (i) uma caracterização abrangente das práticas de segurança de software segundo a percepção de desenvolvedores brasileiros, considerando *facilidade de uso*, *efetividade*, *frequência de uso* e *essencialidade*; (ii) evidências empíricas sobre como fatores individuais e organizacionais, como função desempenhada, experiência em segurança e porte da organização, influenciam a maneira como tais práticas são avaliadas e aplicadas; (iii) uma análise detalhada da adoção de ferramentas automatizadas de segurança, destacando benefícios percebidos, desafios enfrentados, motivos para não adoção e sugestões de melhorias; e (iv) a consolidação de um panorama atualizado sobre percepções, barreiras e tendências relacionadas à integração de segurança no ciclo de desenvolvimento de software.

De forma geral, os resultados obtidos revelam lacunas importantes na adoção sistemática de práticas e ferramentas de segurança. As percepções coletadas evidenciam que, embora desenvolvedores reconheçam a relevância de práticas como revisão de código, codificação segura, modelagem de ameaças e gerenciamento de vulnerabilidades, sua aplicação não é uniforme e é frequentemente reativa. A adoção de ferramentas automatizadas, por sua vez, mostra-se influenciada por fatores como complexidade, integração ao fluxo de trabalho, suporte organizacional e familiaridade técnica.

Os achados deste estudo contribuem para um entendimento mais claro sobre como segurança é incorporada – ou negligenciada – no cotidiano do desenvolvimento de software no Brasil. Eles reforçam a necessidade de investimentos em capacitação, melhorias nas ferramentas de AppSec, fortalecimento da cultura organizacional de segurança e incentivo à adoção proativa e contínua de práticas que reduzam riscos desde as fases iniciais do ciclo de desenvolvimento.

1.3 Organização do Trabalho

Este trabalho está estruturado da seguinte forma: O Capítulo 2 aborda a fundamentação teórica, discutindo conceitos importantes relacionados à segurança de software e ferramentas automatizadas. No Capítulo 3 são apresentados os principais trabalhos relacionados, estabelecendo uma base comparativa e evidenciando possíveis lacunas na literatura. No Capítulo 4 são detalhados os processos metodológicos utilizados no desenvolvimento deste estudo. O Capítulo 5 reúne e analisa os principais resultados obtidos. Em seguida, no Capítulo 6 as possíveis ameaças à validade e as estratégias utilizadas para mitigá-las são apresentadas. Por fim, o Capítulo 7 traz as principais conclusões do trabalho e sugestões para estudos futuros.

1.4 Resumo do Capítulo 1

Neste capítulo é realizada uma introdução ao tema, com a devida contextualização do problema e a relevância da pesquisa. São apresentados o objetivo geral e os objetivos específicos, bem como as questões de pesquisa que orientaram o desenvolvimento do estudo. Destacam-se, ainda, as principais contribuições pretendidas, tanto sob a perspectiva teórica quanto sob a perspectiva prática, especialmente no que se refere aos aspectos relacionados às práticas de segurança de software e ferramentas automatizadas de segurança. Por fim, descreve-se a organização do trabalho de conclusão de curso, detalhando-se a estrutura dos capítulos e a lógica adotada para a condução do estudo. Na sequência, é apresentado o capítulo de fundamentação teórica, no qual são discutidos os principais trabalhos relacionados ao tema, que servem de base conceitual para a pesquisa desenvolvida.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão apresentados os conceitos que fundamentam o desenvolvimento deste trabalho. Nas Seções 2.1 é introduzido conceitos referentes à desenvolvimento de software seguro. Na Seção 2.2, conceitos referentes a práticas de segurança de software. Finalmente, na Seção 2.3 é apresentado a importância das ferramentas de segurança automatizadas.

2.1 Desenvolvimento de Software Seguro

O conceito de segurança de software diz respeito ao processo de criação e desenvolvimento de software que assegura os três pilares da segurança de informação (Confidencialidade, Integridade e Disponibilidade), e incorporar a segurança em qualquer nível do Ciclo de Vida de Desenvolvimento de Software (SDLC) tornou-se um requisito urgente (KHAN *et al.*, 2022). Essa urgência é corroborada por normas internacionais, como a ISO/IEC 27001 (MALATJI, 2023), que estabelece requisitos para Sistemas de Gestão de Segurança da Informação (SGSI), e a ISO/IEC 27034 (COMMISSION *et al.*, 2011), que fornece diretrizes específicas para a segurança de aplicações, orientando as organizações a integrarem controles de segurança ao longo de todo o ciclo de vida do software. O SDLC estabelece uma conexão entre os parâmetros de segurança, proteção e desempenho. Se comparado à definição de requisitos de segurança, o SDLC oferece uma extensão futura simples e integração direta ao processo do Ciclo de Vida do Produto (CVP) com nenhuma ou quase nenhuma interação humana (NINA *et al.*, 2021).

Além disso, dois conceitos importantes ao se falar sobre desenvolvimento de software seguro são os conceitos de Desenvolvimento e Operação (DevOps) e DevSecOps. O DevOps visa aproximar as equipes de desenvolvimento e operação, podendo assim acelerar o lançamento de novos recursos de software e aprimorar o monitoramento de sistemas em produção (BASS *et al.*, 2015). No entanto, no DevOps não há

a preocupação de tratar a segurança como algo que faz parte do processo de desenvolvimento. A necessidade de implementar recursos de segurança durante o processo de desenvolvimento levou à criação do DevSecOps (RAJAPAKSE *et al.*, 2022). O DevSecOps trata a segurança como algo intrínseco ao processo de desenvolvimento, tendo o princípio de manter a segurança como algo prioritário durante o ciclo de vida do desenvolvimento (MYRBAKKEN; COLOMO-PALACIOS, 2017). No cenário brasileiro, essa integração é impulsionada pela LGPD (BRASIL, 2018).

Acoplar a segurança no processo de DevOps tem sido um desafio, pois os métodos de segurança tradicionais não conseguiram acompanhar a agilidade e rapidez de entrega que o DevOps proporciona (MYRBAKKEN; COLOMO-PALACIOS, 2017). Uma das soluções propostas para que o DevSecOps seja possível é a implementação de ferramentas de teste de segurança de aplicativos centradas no desenvolvedor que visem a práticas contínuas no DevSecOps, junto a pesquisas sobre como práticas de segurança de software comumente manuais podem ser automatizadas para que possam se adequar ao processo de DevSecOps (RAJAPAKSE *et al.*, 2022).

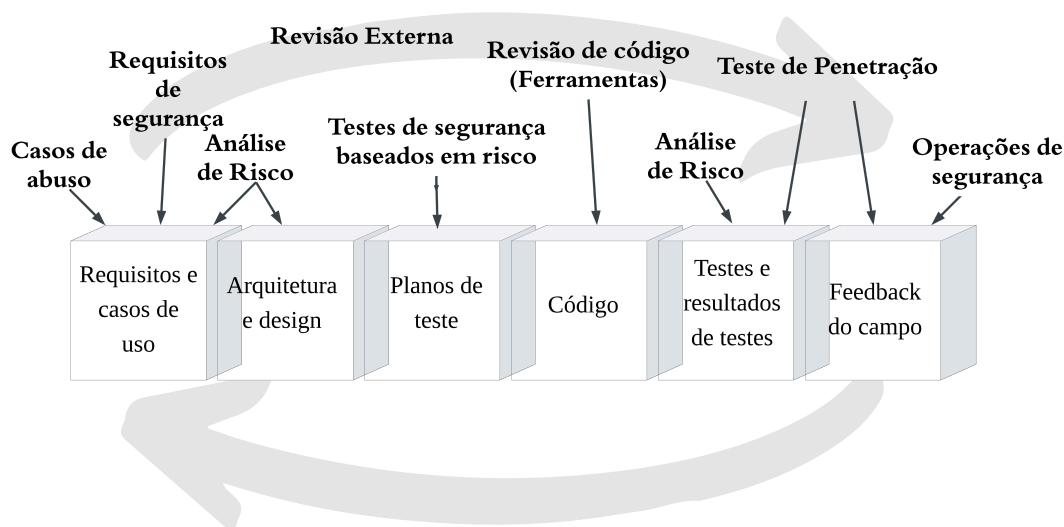
2.2 Práticas de Segurança de Software (AppSec)

As práticas de segurança de software, também conhecidas como práticas de AppSec, visam reduzir vulnerabilidades por meio da incorporação de atividades como *revisão de código*, *testes de penetração*, *uso de padrões de codificação segura* e *modelagem de ameaças*, dentre outras. Diversos autores propuseram taxonomias dessas práticas ao longo do ciclo de desenvolvimento de software (MORRISON *et al.*, 2017; ASSAL; CHIASSON, 2019; RINDELL *et al.*, 2021).

Um dos modelos mais amplamente referenciados é o proposto por McGraw (2006), que introduz os *Security Touchpoints*. Esses *touchpoints* representam pontos estratégicos onde atividades de segurança devem ser integradas aos artefatos típicos do processo de desenvolvimento, como requisitos, arquitetura, código e testes. Ao contrário

de abordagens que tratam segurança como uma etapa isolada ao final do processo, os *touchpoints* incentivam uma integração contínua e proativa da segurança desde o início do projeto. A Figura 1 ilustra como essas práticas se distribuem ao longo do ciclo de vida de desenvolvimento, indicando os artefatos aos quais cada *touchpoint* está associado.

Figura 1 – Integração dos *Security Touchpoints* ao ciclo de desenvolvimento de software.



Fonte: Adaptado de McGraw (2006)

Esse modelo teórico fornece a base conceitual para a classificação das práticas analisadas neste estudo, permitindo explorar sua aplicabilidade real e a percepção dos desenvolvedores em relação à sua adoção. Os sete *Security Touchpoints* definidos por McGraw (2006) são:

- a) **Requisitos de Segurança (*Security Requirements*)** – definição explícita de requisitos de segurança no início do projeto;
- b) **Casos de Abuso (*Abuse Cases*)** – modelagem de casos de uso malicioso a partir de requisitos e fluxos existentes;
- c) **Análise de Risco (*Risk Analysis*)** – análise de riscos em múltiplas fases (design,

- testes e operação);
- d) **Análise de Risco Arquitetural (*Architecture Risk Analysis*)** – avaliação de riscos estruturais na arquitetura do sistema;
 - e) **Revisão de Código (*Code Review*)** – análise manual ou automatizada do código para identificar vulnerabilidades;
 - f) **Testes de Segurança baseado em Risco (*Risk-Based Security Tests*)** – geração de planos de teste com base em riscos identificados; e
 - g) **Teste de Penetração e Operações de Segurança (*Penetration Testing e Security Operations*)** – testes de penetração e práticas de segurança contínuas em ambiente de produção.

2.3 Ferramentas Automatizadas de Segurança

As ferramentas automatizadas de segurança desempenham um papel importante na integração contínua das práticas de AppSec, permitindo que atividades antes manuais – como análises de código, varreduras por vulnerabilidades e auditorias de dependências – sejam realizadas de forma contínua, escalável e integrada ao fluxo de desenvolvimento. Esse tipo de automação é particularmente valorizado em ambientes que adotam práticas ágeis e DevOps, nos quais ciclos de entrega são curtos e a eficiência operacional é crítica. Além disso, o uso de ferramentas automatizadas auxilia na detecção precoce de vulnerabilidades, amplia a cobertura das verificações e reduz a carga manual sobre as equipes técnicas (RAJAPAKSE *et al.*, 2022; RAHMAN; WILLIAMS, 2016). Dentre os principais categorias de ferramentas (ANGERMEIR *et al.*, 2021) adotadas no contexto de AppSec, destacam-se:

- a) **3rd Party Vulnerability Scanning**: envolve ferramentas responsáveis por verificar a implementação e os testes corretos relacionados a riscos de segurança e vulnerabilidades em componentes de terceiros, e.g., Dependabot, PHP Security Checker;

- b) ***Static/Dynamic Application Security Testing***: abrange ferramentas que realizam testes de vulnerabilidades em componentes do produto e suas dependências, assegurando que os patches de segurança estejam atualizados, e.g., CodeQL, Bandit;
- c) ***Secure Configuration/Hardening***: refere-se à aplicação de medidas de configuração de segurança para minimizar a superfície de ataque e garantir a conformidade com padrões de segurança, e.g., Chef, Ansible;
- d) ***Compliance/Hardening Checks***: compreende ferramentas que verificam se as medidas de configuração e reforço de segurança estão alinhadas às práticas recomendadas ou padrões de conformidade, e.g., Ansible-Lint, InSpec; e
- e) ***Secrets Management***: engloba a gestão e a proteção de informações confidenciais, como senhas e chaves privadas, prevenindo o vazamento de dados sigilosos, e.g., GitLeaks.

O uso dessas ferramentas traz benefícios como a automação da detecção de falhas, suporte a *pipelines* de integração contínua e maior rastreabilidade de erros (RAHMAN; WILLIAMS, 2016; ANGERMEIR *et al.*, 2021; RAJAPAKSE *et al.*, 2022). Além disso, quando integradas de forma eficaz, permitem que a segurança seja incorporada desde as etapas iniciais do desenvolvimento - alinhando-se à abordagem de *security shift-left* (RAJAPAKSE *et al.*, 2022).

Contudo, a adoção dessas ferramentas também apresenta diversos desafios práticos. Dentre os mais reportados na literatura e na prática estão: geração excessiva de falsos positivos, dificuldade de interpretação técnica, curva de aprendizado elevada, custos de aquisição e manutenção, e barreiras para integração fluida nos pipelines de desenvolvimento (RAHMAN; WILLIAMS, 2016; RAJAPAKSE *et al.*, 2022). Muitos desenvolvedores relatam que, mesmo quando as ferramentas estão disponíveis, sua utilização plena é comprometida pela ausência de treinamento adequado, sobrecarga de alertas e falta de tempo para configurar regras específicas para o projeto.

Dessa forma, embora as ferramentas automatizadas sejam reconhecidas como

facilitadoras da segurança contínua em ambientes DevOps e DevSecOps, sua eficácia depende do contexto organizacional, do suporte institucional e do nível de capacitação técnica das equipes envolvidas (RAHMAN; WILLIAMS, 2016; RAJAPAKSE *et al.*, 2022). Nosso estudo busca aprofundar essa discussão a partir das percepções dos desenvolvedores brasileiros, contribuindo com evidências empíricas sobre os usos, limitações e oportunidades de melhoria dessas soluções no desenvolvimento cotidiano de software.

2.4 Resumo do Capítulo 2

Nesse capítulo foi abordada a contextualização sobre alguns temas relevantes presentes nas discussões feitas durante o decorrer do estudo. Também foram apresentados temas como o que é *Desenvolvimento de Software Seguro*, sobre práticas de AppSec, assim como algumas classificações dessas práticas. Também foram referenciados os conceitos de DevOps e DevSecOps, assim como alguns desafios por eles enfrentados junto a uma proposta de solução para que o DevSecOps possa ocorrer. E por fim, foi abordado o conceito de *Ferramentas Automatizadas de Segurança*, apresentando não apenas a sua importância ao implementar práticas de segurança durante o processo de desenvolvimento, como algumas categorias em que essas ferramentas são encaixadas.

3 TRABALHOS RELACIONADOS

A segurança de software tem ganhado destaque crescente na engenharia de software, impulsionando pesquisas voltadas à compreensão dos desafios enfrentados por desenvolvedores e das barreiras para adoção de práticas seguras. Estudos recentes investigam como tais práticas são aplicadas em diferentes contextos organizacionais e culturais, revelando fatores técnicos, estruturais e humanos que influenciam sua efetiva adoção (SHRESTHA *et al.*, 2023; OMAR *et al.*, 2022; MORRISON *et al.*, 2017; VENSON *et al.*, 2019; RINDELL *et al.*, 2021; ASSAL; CHIASSON, 2019; RAHMAN; WILLIAMS, 2016; RAJAPAKSE *et al.*, 2022; RYAN *et al.*, 2023). Apesar desses avanços, persistem lacunas em estudos quantitativos mais amplos e na análise da adoção de ferramentas automatizadas em contextos diversos. Dessa forma, nas próximas seções serão apresentados os estudos que contribuíram para o desenvolvimento deste trabalho.

3.1 Estudos em Contextos Organizacionais e Culturais

Shrestha *et al.* (2023) realizaram entrevistas com 15 desenvolvedores em Bangladesh para entender como práticas de segurança são adotadas localmente. O estudo revelou como a segurança é integrada no ciclo de desenvolvimento local e identificou desafios que inibem práticas de segurança em Bangladesh. Esse estudo é semelhante ao explorar as percepções de desenvolvedores sobre segurança e investigar práticas de segurança no desenvolvimento de software. Ambos utilizam investigação direta com desenvolvedores (este por entrevistas, nosso por questionário), mas em contextos diferentes, o estudo de Shrestha *et al.* (2023) no contexto de Bangladesh (Ásia), enquanto nosso foco é no Brasil. Além disso, o estudo de Shrestha *et al.* (2023) usam entrevistas qualitativas, com amostra pequena, ao passo que nosso estudo emprega *survey* (qualitativo e quantitativo) em escala maior.

Omar *et al.* (2022) avaliaram a maturidade em segurança no setor de TI da

Palestina por meio de um *survey* com 40 profissionais de diferentes setores (privado, público e educacional), perfis e níveis de experiência. Os resultados indicaram que a maioria das práticas de segurança não é plenamente adotada, especialmente nas fases iniciais do ciclo de desenvolvimento. Destaca-se que práticas como modelagem de ameaças e treinamentos em segurança apresentaram os maiores índices de não aplicabilidade. Os autores também identificaram fatores que influenciam negativamente a adesão, incluindo a ausência de cultura organizacional voltada à segurança, limitação de orçamento e prazos apertados, além da falta de formação em segurança por parte dos desenvolvedores. Assim como o nosso, o estudo utiliza abordagem quantitativa para mapear percepções sobre práticas de segurança via questionário estruturado. No entanto, enquanto Omar *et al.* (2022) focam especificamente na adesão ao SDLC e destacam fatores organizacionais de forma qualitativa, nosso estudo amplia a análise incluindo a percepção de facilidade de uso e efetividade das práticas, além da adoção de ferramentas automatizadas e da análise por perfis profissionais. O nosso também se diferencia pelo maior tamanho amostral, foco exclusivo no Brasil e por empregar análises comparativas e de correlação entre variáveis individuais e organizacionais.

Morrison *et al.* (2017) conduziram um estudo de caso em um projeto da IBM para avaliar a adoção de 13 práticas de segurança, utilizando o framework SP-EF (Security Practices Evaluation Framework) por meio de observações, *survey* interno e análise de artefatos (como *issues*). Os autores identificaram uma ampla aplicação das práticas, com destaque para *monitoramento de vulnerabilidades*, *codificação segura* e *ferramentas automatizadas*. O estudo também revelou a importância de apoio organizacional contínuo para manter tais práticas - incluindo ferramentas, treinamento e processos. Por outro lado, práticas como *classificação de dados* e *testes de penetração* foram menos frequentes, refletindo tanto seu escopo de aplicação quanto restrições contextuais. Comparado ao nosso estudo, que adota um *survey* com desenvolvedores de diferentes contextos no Brasil, o trabalho de Morrison *et al.* (2017) oferece uma análise profunda em um único

ambiente corporativo. Nosso estudo amplia essa visão ao considerar percepções sobre facilidade, efetividade e o uso de ferramentas em escala nacional.

Venson *et al.* (2019) aplicaram um *survey* com 110 profissionais de 29 países para avaliar o impacto das práticas de segurança sobre o esforço de desenvolvimento. O *survey* foi direcionado a membros de grupos especializados em segurança no LinkedIn. Os autores identificaram que práticas como *codificação segura* são amplamente usadas, enquanto atividades mais intensivas, como *testes de penetração* ou *revisões de segurança*, são menos frequentes e demandam maior esforço (ex.: mais de dois dias por aplicação). Os autores também evidenciaram que muitas organizações não consideram essas atividades nas estimativas iniciais de esforço, e que barreiras culturais e organizacionais dificultam sua adoção. Assim como nosso estudo, esse trabalho utiliza abordagem quantitativa baseada em percepções de desenvolvedores, mas com foco específico no custo e esforço associados à segurança. Em contraste, nosso estudo enfatiza frequência, facilidade e efetividade percebida, além do uso de ferramentas automatizadas, oferecendo uma visão complementar sobre os fatores que influenciam a adoção prática dessas atividades.

3.2 Estudos sobre Segurança de Software em Ambientes Ágeis e Iterativos

Rindell *et al.* (2021) conduziram um *survey* com 61 profissionais da indústria de software na Finlândia para investigar como 40 práticas de segurança são adotadas e percebidas em contextos ágeis. O estudo revelou que práticas frequentemente usadas nem sempre são vistas como eficazes, sugerindo dificuldades de integração ao fluxo ágil. Assim como nosso estudo, o trabalho de Rindell *et al.* (2021) adota uma abordagem quantitativa baseada em *survey* e analisa frequência e eficácia percebida das práticas. No entanto, o estudo de Rindell *et al.* (2021) foca explicitamente no contexto ágil e europeu (Finlândia), com ênfase em como a agilidade afeta a seleção e integração de práticas de segurança. Nosso estudo, por sua vez, tem escopo mais geral (não limitado a métodos

ágeis) e considera também fatores como facilidade de uso, eficácia percebida e uso de ferramentas automatizadas no contexto brasileiro.

Assal e Chiasson (2019) conduziram um *survey* com 123 desenvolvedores na América do Norte para investigar motivações, barreiras na adoção de práticas de segurança. O estudo se baseia em uma perspectiva centrada no humano, questionando o estigma de que os desenvolvedores seriam o elo fraco na segurança. O estudo mostrou que, embora a segurança nem sempre seja prioridade organizacional, os desenvolvedores demonstram motivação intrínseca para aplicá-la, enfrentando obstáculos como ausência de planos formais, falta de ferramentas e pouca cultura de segurança. Em termos de semelhança, o estudo compartilha com o nosso o uso de *survey* e o foco em atitudes e práticas de desenvolvedores frente à segurança. Ambos identificam que, apesar de reconhecerem a importância da segurança, desenvolvedores frequentemente carecem de suporte organizacional adequado. As principais diferenças residem no escopo (América do Norte vs. Brasil), na ênfase metodológica (análise fatorial de motivadores e barreiras vs. combinação de estatísticas descritivas, correlações e análise qualitativa), e no foco adicional do nosso trabalho em variáveis como facilidade de uso, eficácia percebida e uso de ferramentas automatizadas.

3.3 Estudos sobre Automação e Integração de Segurança em DevOps

Rahman e Williams (2016) investigaram a integração de práticas de segurança em ambientes DevOps por meio de análise qualitativa de 66 artefatos públicos (e.g., blogs, apresentações e vídeos) relacionados à segurança em DevOps; e *survey* com nove organizações que adotam DevOps (e.g., Google, Netflix). O estudo identificou que atividades típicas de DevOps, como *monitoramento automatizado* e *pipelines de entrega contínua*, são percebidas positivamente em relação à segurança, desde que acompanhadas por práticas como análise de requisitos de segurança, configurações seguras e forte colaboração entre equipes de desenvolvimento, operações e segurança. A

automação foi destacada como eixo central, tanto em atividades de teste e monitoramento quanto em processos como revisão de código e gerenciamento de licenças. Além disso, observou-se que a colaboração pode ser um risco se não for bem supervisionada, pois o acesso excessivo de membros pode comprometer a segurança. Em termos de semelhança, ambos os trabalhos têm como foco principal as percepções de profissionais da indústria sobre práticas de segurança, com ênfase na integração prática de tais práticas no desenvolvimento de software. Semelhante ao nosso estudo, reconhecem o papel da automação e da percepção dos profissionais, embora com foco em grandes empresas internacionais e escopo restrito ao DevOps. Nosso trabalho, por outro lado, examina práticas mais amplas e a utilização de ferramentas automatizadas com base em *survey* estruturado e análises comparativas (o que permite análises correlacionais mais refinadas) com desenvolvedores brasileiros.

Rajapakse *et al.* (2022) conduziram uma revisão sistemática de 54 estudos sobre DevSecOps, identificando 21 desafios e 31 soluções, organizados em quatro categorias: Pessoas, Práticas, Ferramentas e Infraestrutura. Dentre os principais desafios, os relacionados à categoria de Ferramentas, destacam-se problemas como dificuldade de integração com pipelines existentes, falta de padronização entre ferramentas, curva de aprendizado acentuada, geração de falsos positivos e custos de aquisição e manutenção. A categoria de Práticas também apresentou desafios, como a resistência à mudança cultural e a falta de maturidade no processo de desenvolvimento seguro. Como soluções, os autores destacam a importância de práticas como o “*security shift-left*” (antecipação da segurança para as fases iniciais do desenvolvimento), automação contínua e treinamentos. Enquanto esse estudo fornece uma visão teórica da literatura, nosso trabalho complementa com dados empíricos sobre o uso de ferramentas automatizadas e desafios práticos enfrentados por desenvolvedores brasileiros, reforçando tendências apontadas pela revisão.

3.4 Estudo sobre Cultura e Mensuração de Segurança de Software

Ryan *et al.* (2023) propuseram uma abordagem mais ampla para mensurar segurança em software, incorporando aspectos culturais e comportamentais. Em um *survey* com mais de 1.100 desenvolvedores de 59 países, combinaram o uso de 12 práticas de segurança (ex.: revisão de código, *threat modeling*, testes de penetração) com indicadores de cultura organizacional, como apoio institucional, incentivo ao uso de ferramentas de segurança e tempo dedicado à segurança. Os autores identificaram uma dissociação entre prática e cultura, ou seja, muitas organizações adotam práticas técnicas, mas carecem de conscientização e apoio gerencial. Isso reforça a ideia de que medir apenas o uso de ferramentas ou a presença de controles não é suficiente para avaliar a real maturidade em segurança de software.

Nosso estudo se aproxima dessa proposta ao incluir dimensões subjetivas como percepção de efetividade, facilidade de uso e importância das práticas, fornecendo uma visão mais humana e contextualizada sobre o uso real das práticas de segurança. Embora não tenhamos medido diretamente “cultura organizacional”, nossos resultados sugerem que fatores individuais e contextuais moldam a aplicação da segurança. Assim, ambos os estudos reforçam a importância de métricas mais humanas e contextualizadas para avaliar maturidade em segurança de software.

3.5 Análise Comparativa

O Quadro 1 apresenta um resumo comparativo dos principais trabalhos relacionados a este estudo. O quadro está dividido em: (i) a referência do estudo; (ii) o país ou contexto geográfico analisado; (iii) a metodologia empregada; (iv) o foco principal de investigação; e (v) a relação direta com o presente trabalho.

Quadro 1 – Comparação entre estudos relacionados e o presente trabalho

Estudo	Local	Método	Foco Principal	Relação com este Estudo
Shrestha <i>et al.</i> (2023)	Bangladesh	Entrevistas qualitativas	Desafios e percepção cultural	Explora percepções como nosso estudo, mas com método qualitativo e escopo local limitado
Omar <i>et al.</i> (2022)	Palestina	Survey (n=40)	Adoção de práticas no SDLC	Também usa survey; foca em maturidade e fatores organizacionais. Nosso estudo é mais amplo e analítico
Morrison <i>et al.</i> (2017)	IBM (EUA)	Estudo de caso (survey, observações, artefatos)	Aplicação de 13 práticas em projeto real	Estudo aprofundado em um único projeto; nosso estudo é mais geral e inclui variáveis subjetivas
Venson <i>et al.</i> (2019)	Internacional	Survey (n=110)	Impacto de práticas sobre custo e esforço	Ambos são surveys; Venson foca em esforço, nosso em frequência, eficácia e ferramentas
Rindell <i>et al.</i> (2021)	Finlândia	Survey (n=61)	Segurança em contextos ágeis	Ambos analisam eficácia percebida; nosso escopo é mais geral e inclui ferramentas
Assal e Chiasson (2019)	América do Norte	Survey (n=123)	Motivações e barreiras	Ambos usam survey; eles focam em motivações e barreiras, nós ampliamos para percepção e uso de ferramentas
Rahman e Williams (2016)	Empresas Globais que usam DevOps	Artefatos + survey (n=9 orgs)	Segurança no DevOps com foco em automação	Ambos tratam de integração prática; nosso escopo é mais amplo, com dados de desenvolvedores brasileiros
Rajapakse <i>et al.</i> (2022)	Global (literatura)	Revisão Sistemática (n=54 estudos)	Desafios e soluções no DevSecOps	Base teórica; nosso estudo fornece dados empíricos atuais e práticos sobre automação e ferramentas
Ryan <i>et al.</i> (2023)	Global (59 países)	Survey (n=1.100)	Práticas e cultura organizacional	Mostram dissociação entre prática e cultura; nosso estudo reforça essa visão por meio de percepção e contexto
Esse estudo	Brasil	Survey (n=83)	Frequência, efetividade, ferramentas	Integra achados anteriores em cenário brasileiro

Fonte: Elaborado pelo autor (2026).

3.6 Resumo do Capítulo 3

Os trabalhos revisados demonstram um interesse crescente na incorporação de práticas de segurança ao ciclo de vida do desenvolvimento de software, incluindo abordagens que envolvem automação, integração contínua e aspectos organizacionais e culturais. No entanto, a maioria dos estudos concentra-se em contextos específicos (por exemplo, grandes empresas, DevOps, ambientes ágeis) ou adota perspectivas teóricas

ou organizacionais, com pouca ênfase na percepção prática dos desenvolvedores em contextos diversos. Nesse sentido, o presente estudo busca contribuir ao explorar, de forma empírica e em larga escala, como desenvolvedores brasileiros percebem e aplicam práticas e ferramentas de segurança em seu cotidiano. Esta abordagem permite uma visão mais realista e contextualizada dos fatores que afetam a adoção dessas práticas na indústria. Com base nessa fundamentação, o próximo capítulo apresenta a metodologia utilizada na pesquisa.

4 METODOLOGIA

Este capítulo descreve a metodologia adotada neste estudo e está organizado da seguinte forma. A Seção 4.1 apresenta o objetivo do estudo e as questões de pesquisa. Em seguida, a Seção 4.2 detalha o método de pesquisa utilizado e as etapas utilizadas, incluindo os procedimentos realizados para seleção dos participantes válidos e para a condução das análises qualitativas e quantitativas dos dados. Por fim, a Seção 4.3 traz as considerações finais do capítulo.

O estudo foi conduzido por meio de um questionário estruturado, desenvolvido com base na literatura sobre práticas de segurança de software, automação e adoção de ferramentas de AppSec. O instrumento buscou captar percepções, experiências e comportamentos de desenvolvedores de software e profissionais de segurança da informação no contexto de práticas de desenvolvimento seguro e uso de ferramentas automatizadas.

Os dados coletados foram analisados por meio de abordagens quantitativas e qualitativas. A etapa quantitativa envolveu técnicas estatísticas descritivas e inferenciais, permitindo identificar padrões de uso, tendências de adoção e relações entre variáveis. Já a etapa qualitativa empregou procedimentos de codificação aberta e axial, possibilitando a identificação de temas emergentes e interpretações que complementam e aprofundam os resultados quantitativos.

4.1 Objetivo do Estudo e Questões de Pesquisa

Para definir o objetivo e as Questões de Pesquisa (QPs), foi seguido o modelo *Goal-Question-Metric* (GQM) (CALDIERA; ROMBACH, 1994). O objetivo é: **analisar** (i) um conjunto de práticas de segurança de software e (ii) a adoção de ferramentas automatizadas de segurança; **com o propósito de** caracterizar essas práticas e avaliar como e por que ferramentas automatizadas de segurança são utilizadas; **com respeito** à frequência de uso, facilidade de uso, efetividade e essencialidade percebidas das

práticas de segurança, bem como aos fatores que influenciam a adoção ou não adoção de ferramentas automatizadas; **da perspectiva de desenvolvedores; no contexto de desenvolvedores de software trabalhando no Brasil.** Cada QP é detalhada a seguir:

- a) **QP₁: Como as percepções dos desenvolvedores sobre frequência, efetividade, facilidade de uso e essencialidade diferem entre várias práticas de segurança de software?** – QP₁ visa identificar as práticas de segurança mais comumente utilizadas por desenvolvedores em suas atividades diárias. Além disso, busca-se compreender como essas práticas são percebidas em quatro dimensões: (i) *efetividade*, que se refere à capacidade de cada prática em prevenir, remover e/ou detectar problemas de segurança nos projetos; (ii) *facilidade de uso*, que trata das experiências dos desenvolvedores ao aplicar essas práticas no seu fluxo de desenvolvimento; (iii) *frequência de uso*, que indica o grau de adoção cotidiana; e (iv) *essencialidade*, isto é, o quão indispensáveis essas práticas são consideradas para o proteção de sistemas. Ao responder a QP₁, busca-se compreender melhor a aplicabilidade e a eficácia das práticas de segurança, permitindo identificar lacunas de conhecimento e oportunidades de melhoria.
- b) **QP₂: Existem diferenças significativas nas percepções sobre as práticas de segurança entre grupos de desenvolvedores com diferentes níveis de experiência, funções e tipos de organização?** – QP₂ busca identificar como fatores individuais e organizacionais influenciam a maneira como as práticas de segurança são percebidas em termos de efetividade, facilidade de uso, frequência e essencialidade. Enquanto a QP₁ caracterizou as práticas de forma geral, esta questão analisa diferenças entre grupos com base em três dimensões: (i) *nível de experiência em segurança* (direta ou indireta); (ii) *função desempenhada* (papéis relacionados ou não à segurança); e (iii) *porte da organização* (pequena/média ou grande). A análise visa compreender como esses fatores moldam a adoção e a importância atribuída às práticas, contribuindo para identificar padrões de comportamento e

apoiar iniciativas que fortaleçam a cultura de segurança no desenvolvimento de software.

- c) **QP₃**: *Como as ferramentas automatizadas de segurança são adotadas no desenvolvimento de software e quais desafios os desenvolvedores enfrentam em seu uso?* – **QP₃** tem como objetivo investigar a adoção de ferramentas automatizadas de segurança considerando: (i) as etapas do ciclo de desenvolvimento em que são utilizadas; (ii) os benefícios percebidos; (iii) os principais desafios que dificultam sua adoção; e (iv) sugestões de melhorias. Também se busca identificar as razões para a não utilização dessas ferramentas, quando aplicável. Ao responder a **QP₃**, espera-se fornecer evidências que apoiem recomendações práticas e melhorias relacionadas à integração e ao uso de ferramentas automatizadas de segurança.

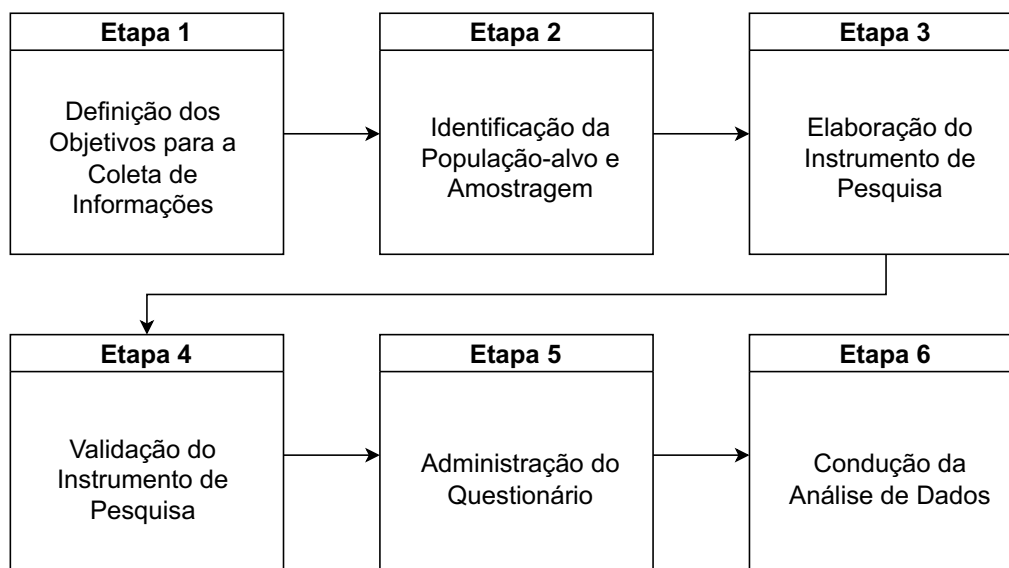
4.2 Etapas da Metodologia do Estudo

Para responder às *QPs*, adotou-se o método de pesquisa baseado em questionário. Tal método foi considerado o mais adequado para investigar as *QPs* propostas através de uma amostra relativamente grande de desenvolvedores de software. Para a elaboração e condução do questionário, foram seguidas as diretrizes propostas por Linaker *et al.* (2015) para configurar e conduzir o questionário. Figura 2 ilustra as principais etapas do estudo. As etapas do processo são descritas em detalhe nas subseções a seguir.

4.2.1 Etapa 1: Definição dos Objetivos para a Coleta de Informações

Nesta etapa, foi conduzida uma sessão de *brainstorming* com o propósito de definir os objetivos e o escopo da pesquisa. O objetivo principal consistiu em caracterizar as práticas de segurança de software em termos de efetividade, facilidade de uso, frequência de uso e essencialidade. Adicionalmente, compreender como ocorre a adoção de ferramentas automatizadas de segurança, incluindo seus benefícios, limitações

Figura 2 – Etapas e procedimentos metodológicos.



Fonte: Elaborado pelo autor (2026).

e fatores que influenciam seu uso ou não uso. Além de compreender como é a adoção de ferramentas automatizadas com aplicação na área de segurança. Esses objetivos orientaram o *design* e o foco das perguntas do questionário, garantindo a coleta de dados relevantes e abrangentes.

4.2.2 Etapa 2: Identificação da População-alvo e Amostragem

A população-alvo deste estudo consiste em indivíduos com mais de 18 anos que trabalham nas áreas de desenvolvimento de software e possuem algum nível de experiência com segurança da informação (software) no Brasil. Para garantir que os participantes se enquadravam neste perfil, foi incluída uma pergunta de controle para confirmar a familiaridade deles com práticas de segurança no processo de desenvolvimento de software (Quadro 2, Q14).

Com o objetivo de recrutar participantes qualificados, utilizamos o Linke-

dIn ¹ como principal fonte de amostragem. O LinkedIn é reconhecido como a maior rede profissional do mundo, com mais de 1 bilhão de usuários distribuídos em diversos setores, incluindo tecnologia da informação (AGUADO *et al.*, 2019). Portanto, a estratégia adotada seguiu os princípios da amostragem intencional (*purposive sampling*), que permite selecionar indivíduos com experiência pertinente ao fenômeno em investigação (TONGCO, 2007; BALTES; RALPH, 2022). Estudos anteriores demonstram que utilizar o LinkedIn para recrutar profissionais é uma fonte ideal para identificar um conjunto diversificado e experiente de candidatos para pesquisas (DAKA; FRASER, 2014; MELLO *et al.*, 2015; OZKAYA; ERATA, 2020; MATOS *et al.*, 2025).

A identificação de potenciais participantes foi realizada por meio de uma busca por profissionais que possuíam um ou mais dos seguintes termos-chave em seus perfis: “*Security Analyst*”, “*DevSecOps*”, “*Pentest*”, “*DevOps*”, “*Software Security*”, “*Cyber Security*”, “*Information Security*”, e “*AppSec*”. Para cada termo-chave, foram selecionados aleatoriamente mais de 60 profissionais, totalizando 609 indivíduos únicos ². Esse processo garantiu diversidade e representatividade entre os perfis abordados.

Para garantir que a amostra fosse composta por profissionais com experiência relevante, cada perfil identificado foi avaliado manualmente por dois pesquisadores envolvidos no estudo. Essa avaliação considerou o histórico profissional, descrição de atividades e projetos em busca de evidências de experiência prática com segurança. Perfis em que os termos-chave apareciam apenas na seção de cursos foram excluídos, pois nosso objetivo era recrutar participantes com experiência prática. Após essa triagem, os profissionais selecionados receberam via LinkedIn um convite individual contendo uma mensagem personalizada e o link de acesso ao questionário hospedado no Google Forms.

¹ <https://www.linkedin.com/>

² Devido a restrições orçamentárias, não foi possível utilizar a versão Premium do LinkedIn durante todo o processo, o que limitou a quantidade de buscas e convites que poderíamos realizar em determinados períodos.

4.2.3 Etapa 3: Elaboração do Instrumento de Pesquisa

A elaboração do questionário foi realizada de forma colaborativa; o mesmo foi aplicado por meio do *Google Forms*. O questionário consiste em 35 perguntas de diferentes tipos, ou seja, abertas, fechadas, de múltipla escolha e escala Likert (LIKERT, 1932). O questionário foi estruturado em cinco seções, conforme apresentado no Quadro 2 e detalhadamente descrito no Apêndice B. As perguntas foram organizadas e agrupadas intencionalmente em seções de modo a minimizar vieses e evitar a influência entre as respostas dos participantes. Cada seção foi disponibilizada apenas após o preenchimento completo da seção anterior.

Quadro 2 – Questões do questionário

ID	Questão	Tipo
I – Caracterização Geral do Participante		
Q1	Qual é a sua idade?	C
Q2	Em qual estado do Brasil você mora atualmente?	C
Q3	Qual é o seu gênero?	C
Q4	Qual é o seu nível de educação formal concluída?	C
Q5	Qual é o seu modelo de trabalho?	C
Q6	Qual papel que melhor descreve suas atividades atuais em projetos de desenvolvimento de software?	M
Q7	Por favor, indique a senioridade da posição que ocupa.	C
Q8	Quantos anos de experiência você tem em funções relacionadas ao desenvolvimento de software ou TI?	C
Q9	Quantos anos de experiência você tem em funções relacionadas a Segurança de Software?	C
Q10	Quais são os tipos de sistemas que você trabalha?	M
Q11	Quais são os domínios dos sistemas que você trabalha atualmente?	M
Q12	Qual é o setor da organização para a qual você trabalha atualmente?	C
Q13	Qual o tamanho da empresa para a qual você trabalha atualmente?	C
II – Caracterização da Experiência com Segurança de Software		
Q14	[Pergunta de Controle] Você tem (ou teve) alguma experiência de trabalho relacionada à segurança de software no processo de desenvolvimento de software?	C
Q15	Você possui alguma certificação na área de segurança da informação? Se sim, quais são elas? Caso não possua, selecione a opção "Não possuo nenhuma certificação.	M

Continua.

Continuação do Quadro 2.

ID	Questão	Tipo
Q16	Por favor, descreva brevemente a sua experiência com práticas de segurança de software. Em caso negativo, responda que não possui experiência.	O
III – Caracterização de Práticas de Segurança de Software		
Q17	Com que <i>frequência</i> você costuma aplicar as seguintes práticas de segurança de software em seus projetos? Considere marcar “Não Aplicável” para as práticas das quais você não utiliza.	L
Q18	Você aplica outras práticas de segurança de software além das mencionadas anteriormente? Se sim, por favor, descreva brevemente sobre essas práticas. (Opcional)	O
Q19	Como você tem implementado as práticas de segurança de software mencionadas anteriormente em seus projetos?	O
Q20	Com base em sua experiência, como você avaliaria o nível de <i>facilidade de uso</i> das seguintes práticas de segurança de software? Considere marcar “Não Aplicável” para as práticas das quais você não utiliza.	L
Q21	Com base em sua experiência, como você avalia a <i>efetividade</i> das seguintes práticas de segurança de software na prevenção, remoção ou detecção de problemas de segurança em seus projetos? Considere marcar “Não Aplicável” para as práticas das quais você não utiliza.	L
Q22	Qual a sua opinião quanto a frase a seguir aplicada em cada uma das práticas abaixo: “ <i>Essa prática é essencial para a proteção do sistema.</i> ”	L
IV – Adoção de Ferramentas de Segurança Automatizadas		
Q23	[Pergunta de Controle] Você costuma utilizar ferramentas de segurança automatizadas em seu dia a dia?.	C
Se resposta a Q23 for "Sim" ou "Uso parcialmente"		
Q24	Para cada uma das seguintes categorias de ferramentas de segurança automatizadas, indique com que <i>frequência</i> você as utiliza. Considere marcar “Não Aplicável” para as categorias de ferramentas das quais você não utiliza.	L
Q25	Quais das seguintes ferramentas específicas de segurança de software você utiliza atualmente ou já utilizou anteriormente? Caso nenhuma das opções listadas seja adequada, selecione "Outros" e especifique.	M
Q26	Em quais fases do ciclo de desenvolvimento de software você costuma utilizar as ferramentas de segurança automatizadas selecionadas na pergunta anterior? (Selecione todas as opções que se aplicam)	M
Q27	Quais recomendações você daria para um desenvolvedor sobre a adoção de ferramentas de segurança para seus projetos?	O
Q28	Quais são os principais desafios que você enfrenta ao automatizar ferramentas de segurança? (Selecione todas as opções que se aplicam)	M
Q29	Quais melhorias você gostaria de ver nas ferramentas de segurança automatizadas que utiliza atualmente? (Selecione todas as opções que se aplicam)	M

Continua.

Continuação do Quadro 2.

ID	Questão	Tipo
Q30	Caso você queira receber os resultados desse estudo e participar de uma futura entrevista sobre Práticas de Segurança de Software, por favor, informe seu e-mail. (Opcional)	O
Se resposta a Q23 for "Não"		
Q31	Quais são as razões pelas quais você opta por não utilizar ferramentas automatizadas de segurança de software? Por favor, descreva brevemente sobre.	O
Q32	Você considera utilizar ferramentas automatizadas de segurança de software em seus projetos no futuro?	C
Q33	Se você respondeu "Sim" ou "Talvez" à pergunta anterior, por favor, indique quais ferramentas utilizaria. Marque "Nenhuma" se respondeu "Não". Caso nenhuma das opções listadas seja adequada, selecione "Outros" e especifique.	M
Q34	Caso você queira receber os resultados desse estudo e participar de uma futura entrevista sobre Práticas de Segurança de Software, por favor, informe seu e-mail. (Opcional)	O
V – Validação de Participação		
Q35	Marque a imagem em que aparece um filhote de cachorro.	C

Nota. C: Fechada; M: Múltipla; O: Aberta; L: Likert.

Fonte: Elaborado pelo autor (2026).

No início do questionário, foram disponibilizadas informações gerais sobre o estudo, incluindo seu objetivo, a metodologia empregada, os procedimentos de tratamento de dados e os contatos dos pesquisadores responsáveis. Também foi apresentado um Termo de Consentimento Livre e Esclarecido (TCLE) (ver Apêndice A), descrevendo as condições e estipulações que regiam a participação dos respondentes.³ Os participantes foram informados de que sua participação era totalmente voluntária e que tinham a liberdade de recusar a participação ou retirar seu consentimento a qualquer momento, sem sofrer qualquer tipo de penalidade. O questionário foi conduzido de forma anônima, sem solicitação de informações de contato dos respondentes.

A primeira seção é composta por 13 perguntas gerais (Q1–Q13), destinadas a coletar informações sobre a formação, perfil profissional e as habilidades dos desenvolvedores. A segunda seção consiste em três perguntas (Q14–Q16) voltadas à coleta

³ As condições estavam em conformidade com os padrões éticos de privacidade previstos na Lei Geral de Proteção de Dados Pessoais (Lei nº 13.709/2018).

de dados sobre a experiência dos participantes com segurança de software, incluindo certificações, tempo de atuação e descrição de atividades relacionadas à área.

A terceira seção inclui seis questões (Q17–Q22) voltadas à caracterização de práticas de segurança de software em termos de quatro dimensões. *Frequência de uso*, utilizando uma escala Likert de 4 pontos (LIKERT, 1932), variando de *Diariamente* a *Anualmente*, além da opção de "Não aplicável". *Facilidade de uso e efetividade*, ambas avaliadas por uma escala Likert de 5 pontos (LIKERT, 1932), variando de *Muito Baixo* a *Muito Alto*, além da opção de "Não aplicável". Adicionalmente, os participantes avaliaram a *essencialidade* de cada prática ao informar o seu nível de concordância com relação à seguinte afirmação “*Essa prática é essencial para a proteção do sistema.*”, utilizando uma escala Likert de 5 pontos (LIKERT, 1932), variando de *Concordo Totalmente* a *Discordo Totalmente*. Essas escalas permitiram mensurar as percepções dos desenvolvedores de forma consistente e comparável entre diferentes práticas de segurança.

As práticas de segurança de software incluídas no questionário foram definidas a partir de um processo iterativo de revisão e refinamento, fundamentado tanto em literatura especializada quanto na validação por especialistas da área. Inicialmente, foi identificado um conjunto abrangente de 45 práticas com base em estudos empíricos que investigam a adoção de práticas de segurança em projetos de software (RINDELL *et al.*, 2021; OMAR *et al.*, 2022; MORRISON *et al.*, 2017; VENSON *et al.*, 2019; RAHMAN; WILLIAMS, 2016; ZHOU *et al.*, 2023; RAJAPAKSE *et al.*, 2022; RYAN *et al.*, 2023; SHRESTHA *et al.*, 2023). Em seguida, esse conjunto inicial foi, então, analisado de forma colaborativa por três especialistas com experiência em Engenharia de Software e Segurança de Software. O objetivo desse processo foi reduzir a granularidade excessiva, eliminar redundâncias semânticas e selecionar práticas representativas, de modo a garantir clareza conceitual, relevância prática e viabilidade do instrumento, evitando fadiga dos respondentes do questionário.

Durante esse refinamento, as práticas foram organizadas e agrupadas se-

gundo sete categorias clássicas de segurança, fundamentadas nos *security touchpoints* da literatura de *Software Security* (MCGRAW, 2006): (i) *Code Review*, (ii) *Architectural Risk Analysis*, (iii) *Penetration Testing*, (iv) *Risk-based Security Tests*, (v) *Abuse/Misuse Cases*, (vi) *Security Requirements*, e (vii) *Security Operations*. Uma descrição detalhada dessas categorias é apresentada na Seção 2.2.

Como resultado desse processo de consolidação, foi definido um conjunto final de 16 práticas de AppSec, contemplando atividades amplamente reconhecidas tanto na indústria quanto na academia. Entre elas destacam-se: *Revisão de Código*, *Modelagem de Ameaças*, *Identificação de Casos de Abuso/Mau Uso*, *Análise de Riscos Arquiteturais*, *Testes de Penetração*, *Gerenciamento de Riscos*, *Gerenciamento de Vulnerabilidades*, *Testes de Segurança Manuais/Automatizados*, *Execução de Testes Fuzz*, *Engenharia de Requisitos de Segurança*, *Codificação Segura*, *Resposta Emergencial ao Código*, *Configurações e Gerenciamento de Segurança*, *Monitoramento de Eventos de Segurança*, *Auditorias de Segurança Externas*, e *Segurança da Cadeia de Suprimentos de Software*.

Adicionalmente, foi disponibilizada no questionário a opção de resposta “Outro”, permitindo que os participantes descrevessem práticas específicas adotadas em seus contextos organizacionais. Essa estratégia buscou ampliar a flexibilidade do instrumento e capturar práticas emergentes ou contextualmente relevantes que não estivessem explicitamente contempladas no conjunto pré-definido.

A quarta seção visa investigar a adoção de ferramentas automatizadas de segurança de software e a experiência dos desenvolvedores em seu uso. Para isso, foi incluída uma pergunta de controle (Q23): “*Você costuma utilizar ferramentas de segurança automatizadas em seu dia a dia?*”, com as opções de resposta *Sim*, *Uso parcialmente* ou *Não*. Caso a resposta seja *Sim* ou *Uso parcialmente*, o participante é direcionado a um conjunto de sete perguntas (Q24–Q30) voltadas a compreender: (i) a frequência e as categorias de ferramentas adotadas, utilizando uma escala Likert de 4 pontos (LIKERT, 1932), variando de *Diariamente* a *Anualmente*, além da opção “Não aplicável”; (ii) as

ferramentas específicas adotadas; (iii) as fases do ciclo de desenvolvimento em que essas ferramentas são aplicadas – *durante a codificação, durante a integração contínua, durante a entrega contínua, durante a revisão de código, durante os testes ou em produção*; e (iv) aspectos qualitativos relacionados à adoção, incluindo recomendações, desafios (como *custo, curva de aprendizado, falsos positivos*), e possibilidades de melhoria (como *menos falsos positivos, facilidade uso, redução do tempo de análise*).

O conjunto de categorias e ferramentas automatizadas foi definido com base em um estudo anterior que analisou atividades automatizadas de segurança em pipelines de Integração Contínua (CI) de projetos de software de código aberto orientados por empresas (*enterprise-driven OSS*) (ANGERMEIR *et al.*, 2021). No total, foram consideradas 53 ferramentas, agrupadas em cinco categorias principais: (1) *Compliance/Hardening Checks*; (2) *Static/Dynamic Application Security Testing (SAST/DAST)*; (3) *3rd Party Vulnerability Scanning*; (4) *Secrets Management*; e (5) *Secure Configuration/Hardening*. Mais detalhes sobre essas categorias podem ser vistos na Seção 2.3.

Por outro lado, se a resposta for *Não*, o participante é encaminhado a um conjunto alternativo de quatro perguntas (Q31–Q34), elaboradas para identificar as razões para a não utilização, a intenção de uso futuro e as ferramentas que poderiam ser adotadas. Por fim, a última seção inclui uma pergunta adicional de verificação (Q35), destinada a confirmar se o respondente é um participante humano.

4.2.4 Etapa 4: Validação do Instrumento de Pesquisa

Antes da aplicação do questionário, foi conduzido um teste piloto (LITWIN; FINK, 1995) com três estudantes de mestrado em Ciência da Computação e um profissional da indústria com experiência em segurança da informação. Ambos não estão envolvidos em nosso estudo. O *feedback* fornecido pelos participantes contribuiu para o refinamento do instrumento, resultando em ajustes na formulação das perguntas, na inclusão das categorias de ferramentas, na melhoria nas definições das práticas de segurança

de software, na inclusão de novas opções de resposta e na reordenação de alternativas em questões fechadas. As recomendações foram incorporadas ao questionário. Vale destacar que as respostas coletadas no teste piloto não foram incluídas nos resultados apresentados neste estudo. Com relação ao tempo de preenchimento, os respondentes do piloto levaram, em média, 15 minutos para concluir o questionário. Esse tempo médio foi informado aos participantes quando o questionário foi disponibilizado publicamente.

4.2.5 Etapa 5: Administração do Questionário

O questionário foi hospedado e administrado na plataforma *Google Forms*. A etapa de administração seguiu diretamente o processo de amostragem descrito anteriormente, no qual potenciais participantes foram identificados e filtrados manualmente com base em seus perfis profissionais no LinkedIn. Após essa filtragem, que garantiu que apenas indivíduos com experiência relevante em segurança de software fossem convidados, cada profissional selecionado recebeu uma mensagem personalizada contendo o link para participação no estudo.

A divulgação também contou com uma estratégia complementar de alcance ampliado. Além dos convites diretos enviados via LinkedIn, foram utilizados WhatsApp e e-mail para contatar profissionais da rede dos pesquisadores, bem como publicações públicas no LinkedIn visando atingir indivíduos que, embora não tenham sido identificados na etapa de amostragem, pertenciam ao mesmo ecossistema profissional de interesse. Essa combinação de convites direcionados e divulgação aberta permitiu manter rigor na seleção inicial, ao mesmo tempo em que ampliou a possibilidade de participação voluntária de perfis aderentes ao estudo. O questionário ficou disponível por 105 dias, de 20 de janeiro a 5 de maio de 2025.

4.2.6 Etapa 6: Condução da Análise de Dados

No total, foram obtidas 95 respostas, das quais 12 foram excluídas por não apresentarem qualquer experiência profissional, direta ou indireta, relacionada à segurança no processo de desenvolvimento de software. Assim, foram consideradas 83 respostas para análise. Em nossa amostra de dados, cerca de 62 (74,69%) dos participantes têm experiência profissional direta relacionada com segurança no processo de desenvolvimento de software, enquanto 21 (25,3%) têm experiência indireta.

A abordagem de análise variou de acordo com os tipos de perguntas. Para perguntas fechadas e de múltipla escolha, foram calculadas e apresentadas as porcentagens correspondentes a cada opção selecionada, complementadas por visualizações de dados. Para as perguntas da escala *Likert*, foram realizadas análises de correlação.

Por fim, para as perguntas abertas, aplicaram-se procedimentos da *Grounded Theory* (GT), com a realização de codificação aberta e axial (CORBIN; STRAUSS, 2008). GT refere-se a um método de geração indutiva de teoria a partir de dados. Tradicionalmente, a GT envolve três procedimentos de codificação: codificação aberta, codificação axial e codificação seletiva, sendo a codificação o processo de decompor, comparar e categorizar dados qualitativos (CORBIN; STRAUSS, 2008; EASTERBROOK *et al.*, 2008). Neste estudo, foram aplicados apenas dois desses procedimentos: a codificação aberta, com o objetivo de identificar tópicos recorrentes e aspectos relevantes nas respostas abertas dos participantes, e a codificação axial, empregada para agrupar tópicos inter-relacionados em categorias de nível mais elevado. A codificação seletiva não foi adotada, uma vez que o objetivo do estudo não foi a construção de uma teoria substantiva, mas a caracterização e compreensão das percepções dos desenvolvedores.

O processo iniciou-se com a leitura minuciosa e análise sistemática de todas as respostas textuais, nas quais os dados foram segmentados em unidades discretas de significado. A cada unidade foram atribuídos códigos descritivos que representavam

ideias, percepções, práticas relatadas ou desafios associados à segurança de software e à adoção de práticas e ferramentas de AppSec. Esse processo indutivo permitiu que os códigos emergissem diretamente dos dados, sem a imposição de categorias pré-definidas.

Em seguida, foi conduzida a codificação axial, na qual os códigos previamente identificados foram comparados, relacionados e organizados em categorias mais abrangentes. Essa etapa concentrou-se na identificação de padrões e relações entre os tópicos emergentes, tais como benefícios percebidos das práticas de AppSec, barreiras técnicas e organizacionais, fatores que influenciam a adoção de ferramentas automatizadas e oportunidades de melhoria apontadas pelos participantes. Por fim, as categorias resultantes foram refinadas por meio de comparações iterativas, assegurando consistência na codificação e alinhamento com as questões de pesquisa. Esse processo forneceu uma visão estruturada e coerente das percepções dos desenvolvedores, fundamentando a análise qualitativa apresentada nas Seções 5.2 e 5.4. A seguir, são apresentadas as análises conduzidas para responder às questões de pesquisa (*QPs*) formuladas.

4.2.6.1 *Análise para Responder à QP₁*

Para responder à QP₁, foram realizadas análises quantitativas e qualitativas, descritas em detalhes a seguir.

Análise de tendências das práticas de segurança de software. Para compreender a adoção e a percepção das práticas, foi conduzida uma análise descritiva com base nas respostas obtidas para as quatro dimensões investigadas: *frequência de uso* (Q17) que utiliza a escala Likert de 4 pontos, *facilidade de uso* (Q20), *efetividade* (Q21) e *essenciabilidade* (Q22), ambas avaliadas em escala Likert de 5 pontos. Essa análise visa sintetizar os padrões gerais das respostas por dimensão.

Análise de correlações sobre as percepções de frequência, facilidade de uso, efetividade e essencialidade das práticas de segurança de software. Para avaliar as percepções dos desenvolvedores sobre as práticas de segurança de software, analisamos

os dados sobre a frequência de uso (Q17), facilidade de uso (Q20), efetividade (Q21) e essencialidade (Q22) de 16 práticas, com base nas respostas de 83 desenvolvedores. Nesse contexto, com o objetivo de considerar diferentes níveis de especialização entre os desenvolvedores, adaptamos uma metodologia inspirada em estudos anteriores (DIAS-NETO *et al.*, 2017; MATOS *et al.*, 2025). Essa estratégia permitiu comparar percepções entre perfis heterogêneos e identificar possíveis padrões comportamentais ou divergências relevantes. Essa estratégia envolveu três etapas principais, detalhadas a seguir:

Etapa 1 - Atribuição de peso: A cada participante foi atribuído um peso com base em seus anos de experiência, nível de senioridade, grau acadêmico e a relação entre experiência direta e especialização em segurança de software (Eq.4.1). O peso atribuído reflete a especialização geral do participante, considerando tanto as qualificações formais quanto a experiência prática.

$$W(i) = E(i) + S(i) + C(i) + \left(\frac{Y(i)}{\text{Mediana}(Y)} \right) + \left(\frac{Z(i)}{\text{Mediana}(Z)} \right) + R(i), \text{ onde:} \quad (4.1)$$

- a) $W(i)$: é o peso total atribuído ao participante i ;
- b) $E(i)$: é o nível acadêmico mais alto do participante i , como (1) Ensino Médio, (2) Ensino Superior, (3) Especialização, (4) Estudante de Mestrado, (5) Mestrado, (6) Estudante de Doutorado e (7) Doutorado;
- c) $S(i)$: é o nível de senioridade relatado pelo participante i em sua posição atual, como (1) Estagiário/*Trainee*, (2) Júnior (até 5 anos), (3) Pleno (6 a 9 anos) e (4) Sênior (mais de 10 anos);
- d) $C(i)$: indica a quantidade de certificações válidas relacionada à segurança da informação que o participante i possui;
- e) $Y(i)$: é o número de anos de experiência relatado pelo participante i em desenvolvimento de software, como (1) Menos de 1 ano, (2) Entre 1 e 3 anos, (3) Entre 4 e 6 anos, (4) Entre 7 e 14 anos e (5) Mais de 15 anos;

- f) $Z(i)$: é o número de anos de experiência relatado pelo participante i em segurança de software, como (1) Menos de 1 ano, (2) Entre 1 e 3 anos, (3) Entre 4 e 6 anos, (4) Entre 7 e 14 anos e (5) Mais de 15 anos;
- g) $\text{Mediana}(Y)$: é a mediana dos anos de experiência em desenvolvimento, considerando as respostas de todos os participantes; e
- h) $\text{Mediana}(Z)$: é a mediana dos anos de experiência em segurança de software, considerando as respostas de todos os participantes;
- i) $R(i)$: representa o valor da relação entre experiência direta e especialização em segurança de software.

Explicamos o $R(i)$ da seguinte forma (Eq. 4.2).

$$R(i) = \frac{D(i) + X(i)}{2}, \text{ onde:} \quad (4.2)$$

- a) $R(i)$: é o valor da relação entre experiência direta e especialização em segurança de software;
- b) $D(i)$: indica experiência direta em segurança de software, com um valor de 1 se o desenvolvedor tiver experiência direta em segurança de software, caso contrário, 0;
- c) $X(i)$: denota a experiência do desenvolvedor em segurança de software, atribuindo um valor de 1 se o desenvolvedor for um especialista em segurança de software, ou seja, se sua posição atual for *Cryptographer*, *Penetration tester*, *DevOps engineer*, *Privacy engineer*, ou *Security engineer*, caso contrário, 0.

Em resumo, $R(i)$ calcula a média de dois atributos binários, $D(i)$ e $X(i)$, para refletir a relação geral do participante entre experiência direta e especialização profissional em segurança de software. A interpretação para cada caso é a seguinte:

- a) $D = 0, X = 0$: O participante não tem experiência direta nem conhecimento profissional em segurança de software. Neste caso, $R = 0$, indicando nenhuma relação;
- b) $D = 1, X = 0$: O participante tem experiência direta, mas não é considerado um especialista. Aqui, $R = 0,5$, representa um envolvimento moderado.

- c) $D = 0, X = 1$: O participante é considerado um especialista, mas não tem experiência direta. Da mesma forma, $R = 0,5$.
- d) $D = 1, X = 1$: O participante possui experiência direta e conhecimento profissional em segurança de software. Neste caso, $R = 1$, indicando o mais alto nível de relevância.

Etapa 2 - Respostas ponderadas: Após atribuir pesos, a resposta de cada participante à frequência de uso, facilidade de uso, efetividade e essencialidade das 16 práticas de segurança de software foi multiplicada pelo peso correspondente. Isso resulta em uma pontuação ponderada para cada prática. O valor total para cada prática é então calculado somando-se essas respostas ponderadas de todos os participantes, conforme definido na Eq. 4.3.

$$T(j) = \sum_{i=1}^n (\text{Resposta}(i, j) \times W(i)), \text{ onde:} \quad (4.3)$$

- a) $T(j)$: é o valor total da frequência de uso, facilidade de uso, efetividade e essencialidade para a prática de segurança de software j ;
- b) $\text{Resposta}(i, j)$: é o valor da resposta que varia de 1 a 5, relacionado à frequência de uso, facilidade de uso, efetividade e essencialidade do participante i na prática de segurança de software j ;
- c) $W(i)$: é o peso para o participante i .

Etapa 3 - Normalização: Considerando que as quatro dimensões avaliadas (*frequência de uso, facilidade de uso, efetividade e essencialidade*) foram mensuradas por meio de escalas Likert com diferentes granularidades, foi necessário aplicar um processo de normalização para viabilizar comparações diretas entre dimensões e práticas analisadas. As respostas “Não aplicável” foram tratadas como valores ausentes (*missing values*) e, portanto, excluídas das análises estatísticas, assegurando a validade das médias e proporções calculadas.

Especificamente, as dimensões foram operacionalizadas da seguinte forma: (1) *Frequência de uso*, medida por uma escala Likert de 4 pontos (LIKERT, 1932), variando de *Diariamente* a *Anualmente*, além da opção “Não aplicável”; (2) *Facilidade de uso* e *Efetividade*, ambas avaliadas por uma escala Likert de 5 pontos (LIKERT, 1932), variando de *Muito Baixo* a *Muito Alto*, também com a opção “Não aplicável”; e (3) *Essencialidade*, na qual os participantes expressaram seu grau de concordância com a afirmação “*Essa prática é essencial para a proteção do sistema.*”, utilizando uma escala Likert de 5 pontos (LIKERT, 1932), variando de *Concordo Totalmente* a *Discordo Totalmente*.

Devido a essas diferenças de escala, o valor agregado obtido na Etapa 2 para cada prática precisou ser transformado em um indicador percentual comparável. Assim, foi calculado um valor normalizado entre 0 e 100%, dividindo-se a pontuação total acumulada para cada prática pelo número máximo possível que poderia ter sido alcançado, considerando o tamanho da escala correspondente e o peso dos participantes. A normalização segue a Eq. 4.4:

$$N(j) = \frac{T(j)}{\sum_{i=1}^n (W(i)) \times PontMaxPorParticipante}, \text{ onde:} \quad (4.4)$$

- a) $N(j)$: é o valor normalizado para a frequência de uso ou facilidade de uso ou efetividade ou essencialidade, de uma prática de segurança de software j ;
- b) $T(j)$: é o valor total calculado para a prática j na Etapa 2;
- c) $W(i)$: é o valor de peso para o participante i ;
- d) $PontMaxPorParticipante$: é a pontuação máxima que um participante poderia atribuir naquela dimensão, considerando o tipo de escala (4 pontos no caso de frequência; 5 pontos nas demais).

Esse procedimento garante que todas as dimensões, mesmo derivadas de escalas diferentes, possam ser analisadas de forma consistente, possibilitando comparações significativas entre práticas e dimensões de percepção. O valor normalizado (Eq.4.4) é

usado na Tabela 1 para comparar a *frequência de uso, facilidade de uso, efetividade e essencialidade* de cada prática de segurança de software.

Para demonstrar a aplicação da fórmula, usamos as respostas de frequência percebida para a prática *Revisão de Código* (Tabela 1) como exemplo, da seguinte forma:

Calcule o valor total de frequência percebida: A frequência total percebida (T) para esta prática é calculada multiplicando o peso de cada desenvolvedor ($W(i)$) pela resposta correspondente ($Resposta(i, \text{Revisão de Código})$) e somando os resultados de todos os participantes (n). Neste caso, ($T(\text{Revisão de Código}) \approx 1074,11$):

$$T(\text{Revisão de Código}) = \sum_{i=1}^n (\text{Resposta}(i, \text{Revisão de Código}) \times W(i)) \quad (4.5)$$

Normalizar para determinar o nível de frequência percebida: O nível de frequência percebida é determinado como uma porcentagem, variando de 0% a 100%, normalizando o valor total de frequência. Isso é feito dividindo $T(\text{Revisão de Código})$ pela soma dos pesos possíveis para todos os participantes $\sum_{i=1}^n (W(i))$, multiplicada pelo valor da escala utilizada, que no caso do atributo de frequência é 4. A normalização garante que os resultados sejam comparáveis entre todas as práticas, independentemente do número de participantes ou de seus pesos.

$$N(j) = \frac{1074,11}{344,69 \times 4} \approx 0.7790$$

O resultado, $N(j) \approx 0.7790$, indica que a efetividade percebida da prática de *Revisão de Código* é de aproximadamente 77,90%.

Análise qualitativa. Foram igualmente aplicados procedimentos da GT para a análise das respostas às perguntas abertas (Q18 e Q19), sendo a Q18 uma pergunta opcional. Na Q18, perguntou-se aos participantes se eles usavam alguma prática de segurança de software adicional além das mencionadas na pergunta anterior (Q17)

e, em caso afirmativo, que as especificassem brevemente. Essa pergunta tinha como objetivo revelar práticas menos convencionais que pudessem complementar ou expandir o conjunto já predefinido. Para a Q19, perguntou-se aos participantes como eles têm implementado as práticas de segurança de software mencionadas anteriormente (Q17 e Q18) em seus projetos. Essa pergunta visa compreender como essas práticas são efetivamente aplicadas no contexto real de desenvolvimento em termos de mecanismos de integração ao fluxo de trabalho, bem como possíveis barreiras enfrentadas pelos participantes durante a implementação.

4.2.6.2 *Análise para Responder à QP₂*

Semelhante à questão anterior, para responder à QP₂, foram realizadas análises quantitativas descritas em detalhes a seguir.

Análise de correlações sobre as práticas de segurança de software considerando diferentes grupos. Para identificar se as percepções dos desenvolvedores sobre as práticas de segurança de software variam conforme fatores individuais e organizacionais, os respondentes foram divididos em diferentes grupos demográficos. Depois, procedeu-se à comparação das respostas obtidas na escala *Likert* em cada afirmação das perguntas sobre *frequência de uso* (Q17), *facilidade de uso* (Q20), *efetividade* (Q21) e *essencialidade* (Q22).

Foram definidos os seguintes grupos de análise: (a) Respondentes com experiência direta em segurança de software no processo de desenvolvimento (62 respondentes); (b) Respondentes com experiência indireta em segurança de software, ou seja, seus grupos realizaram trabalhos relacionados à segurança de software, mas os respondentes não estiveram diretamente envolvidos (21 respondentes).

No que se refere ao porte organizacional, os respondentes foram agrupados em: (c) Respondentes que trabalham em organizações relativamente maiores (65 respostas) e (d) Respondentes que trabalham em organizações relativamente menores

(16 respostas). Do total de 83 respondentes, dois participantes foram excluídos dessa análise por selecionarem a opção “Não sei” na caracterização do porte da organização em que atuam, o que impossibilitou sua classificação em qualquer um dos dois grupos. Por fim, quanto à função exercida, foram considerados: (e) Respondentes considerados especialistas com base na função que exercem relacionada à segurança (26 respostas); e (f) Respondentes considerados não especialistas com base na função que exercem não relacionada à segurança (57 respostas).

Para categorizar os participantes com base em sua experiência direta ou indireta com segurança no processo de desenvolvimento, foram consideradas as respostas à Pergunta 14, na qual os desenvolvedores indicaram uma das seguintes opções: *Experiência direta*: “Eu já trabalhei (ou trabalho) com segurança no processo de desenvolvimento de software.” *Experiência indireta*: “Meu grupo/equipe de desenvolvimento trabalha (ou trabalhou) com segurança, mas não estou diretamente envolvido em nenhuma tarefa.”; e *Sem experiência*: “Nunca tive experiência direta e nem indireta com segurança no processo de desenvolvimento de software”. Em relação ao tamanho da organização, a categorização baseou-se nas respostas à Pergunta 13: “Qual é o tamanho da empresa em que você trabalha atualmente?” Com base nas respostas, as organizações são categorizadas da seguinte forma (MATOS *et al.*, 2025): *Pequenas organizações*: até 9 funcionários, 10 a 49 funcionários e 50 a 99 funcionários; e *Grandes organizações*: 100 a 499 funcionários, 500 a 999 funcionários e mais de 1.000 funcionários. Por fim, foram considerados especialistas em segurança (Q6), quando os participantes selecionaram uma dessas funções nos projetos de desenvolvimento de software: *Cryptographer*, *Penetration tester*, *DevOps engineer*, *Privacy engineer*, ou *Security engineer*.

As práticas de software e os resultados da comparação estão resumidos na Tabela 3. Para realizar esta análise, foram utilizados o *Wilcoxon Rank Sum Test* (WHITLEY; BALL, 2002) e o *Cliff’s Delta* (d) (GRISSOM; KIM, 2005) a fim de avaliar, para cada prática, o grau de concordância entre grupos previamente definidos (experiência

direta vs. indireta, especialistas vs. não especialistas, empresas grandes vs. pequenas) em cada uma das dimensões analisadas.

O *Wilcoxon Rank Sum Test* foi utilizado para verificar se as diferenças observadas entre dois grupos são estatisticamente significativas, adotando um nível de confiança de 95% (valor $p < 0,05$). A medida *Cliff's Delta* (d) (GRISSOM; KIM, 2005), por sua vez, quantifica a magnitude da diferença, ou seja, o quanto um grupo tende a apresentar valores maiores que o outro. Por exemplo, ele permite responder perguntas como: *quão maior é a frequência de uso da prática de Modelagem de Ameaças entre desenvolvedores com experiência direta em segurança em comparação aos com experiência indireta?*. Para interpretar o tamanho do efeito *Cliff's Delta* (d), foi utilizada uma classificação bem conhecida (ROMANO *et al.*, 2006), que define quatro categorias de magnitude, representadas na Tabela 3: insignificante (sem símbolo), pequena (*), média (**) e grande (***). As magnitudes (d) positivas são representadas pelo símbolo (+) e as negativas pelo símbolo (–). Como resultado, identificamos que oito práticas apresentaram diferenças estatisticamente significativas entre os grupos analisados.

Para ilustrar a interpretação desses resultados, considere o caso da prática *Modelagem de Ameaças*, apresentado na Tabela 3. Os desenvolvedores com experiência direta em segurança indicaram maior frequência de uso (média = 2,80) do que aqueles com experiência indireta (média = 1,83). O valor $p = 0,02$ indica significância estatística, e o *Cliff's Delta* calculado é positivo e classificado como de grande magnitude. Isso significa que, de forma consistente, desenvolvedores com experiência direta utilizam (ou percebem utilizar) *Modelagem de Ameaças* com muito mais frequência do que aqueles sem experiência direta. Uma evidência importante para compreender como diferentes perfis profissionais adotam práticas de segurança.

4.2.6.3 *Análise para Responder à QP₃*

Para responder à QP₃, foram conduzidas análises qualitativas e quantitativas para compreender como desenvolvedores adotam ferramentas de segurança automatizadas, quais fatores influenciam essa adoção e quais recomendações, desafios e melhorias são percebidos pelos participantes, conforme demonstrado a seguir.

Análise quantitativa. Foram analisadas as respostas às questões fechadas e de múltipla escolha (Q24–Q26, Q28, Q29, Q32, Q33), cobrindo: (1) frequência de uso por categoria de ferramenta (Q24); (2) ferramentas específicas utilizadas (Q25); (3) fases do ciclo de desenvolvimento em que as ferramentas são aplicadas (Q26); (4) desafios percebidos na automação da segurança (Q28); e (5) melhorias desejadas (Q29). Para participantes que não utilizam ferramentas (a partir da Q23), também foram analisadas suas intenções futuras de adoção (Q32) e ferramentas de interesse (Q33). **Análise qualitativa.** As respostas abertas (Q27 e Q31), também foram analisadas por meio da aplicação dos procedimentos de GT. Em Q27, os participantes relataram recomendações para outros desenvolvedores. Já em Q31, analisaram-se as justificativas dadas pelos participantes que optam por não utilizar ferramentas automatizadas de segurança.

4.3 **Resumo do Capítulo 4**

Este capítulo apresentou os procedimentos metodológicos adotados nesta pesquisa, que utiliza uma abordagem quantitativa e qualitativa, de caráter descritivo e exploratório, para analisar um conjunto de práticas de segurança de software e a adoção de ferramentas automatizadas de segurança. O objetivo principal foi caracterizar essas práticas e compreender como e por que tais ferramentas são utilizadas no contexto do desenvolvimento de software. O estudo foi conduzido por meio de um questionário estruturado aplicado a desenvolvedores e profissionais de segurança, permitindo coletar dados sobre frequência de uso, facilidade, efetividade e essencialidade atribuídas a

diferentes práticas e ferramentas. As etapas metodológicas descritas neste capítulo fundamentam a análise dos resultados empíricos, apresentados no Capítulo 5.

5 RESULTADOS

Este capítulo está organizado da seguinte forma: A Seção 5.1 descreve a caracterização geral dos participantes. Em seguida, a Seção 5.2 apresenta os resultados relacionados à percepção dos desenvolvedores sobre práticas de AppSec. Logo após a Seção 5.3 aborda a influência de experiência, função e porte organizacional nas percepções sobre práticas de AppSec. Por fim, a Seção 5.4 descreve os benefícios, adoção e desafios das ferramentas automatizadas de segurança no desenvolvimento de software.

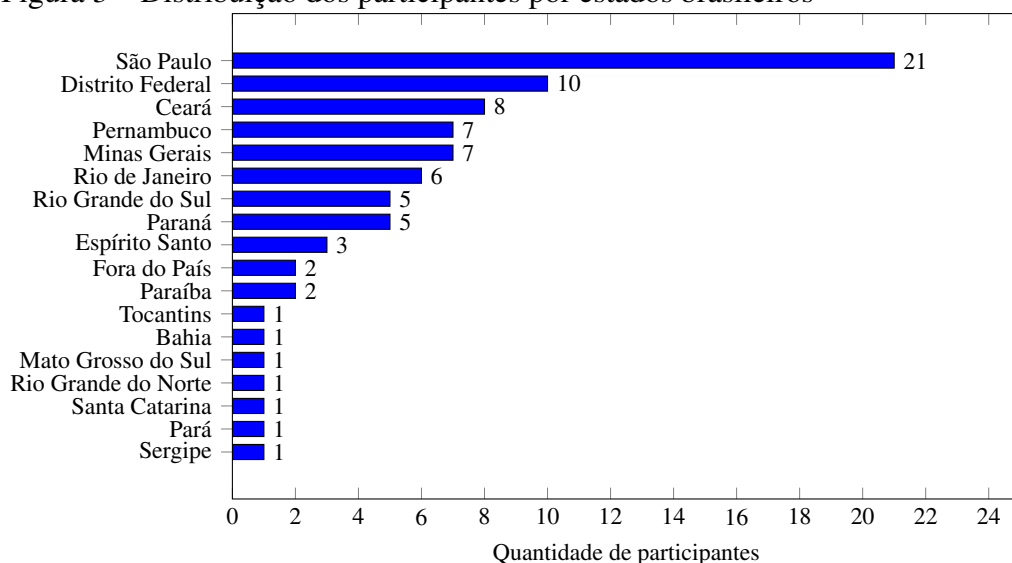
Como mencionado na Seção 4.2.6, a amostra de dados deste estudo consistiu em 83 respostas para análise. Consideramos apenas os participantes que relataram experiência de trabalho direta ou indireta relacionada à segurança no processo de desenvolvimento de software. Para simplificar, utilizaremos a notação ($n/83$) ao longo do texto para indicar um número n dentre os 83 participantes válidos.

5.1 Caracterização Geral dos Participantes

Como apresentado na Figura 3, os participantes estão distribuídos entre 17 estados brasileiros (Q2) mais duas participações de brasileiros que residem no exterior, sendo São Paulo o estado com a participação mais significativa, com 21 participantes e representando aproximadamente 25,3%.

Em seguida, o Distrito Federal com 10 participantes (12%) e o Ceará com 8 participantes (8,9%) são os estados com maior participação no estudo. Com menor participação, podem-se citar os estados Pernambuco (7 participantes), Minas Gerais (7 participantes), Rio de Janeiro (6 participantes), Rio Grande do Sul (5 participantes), Paraná (5 participantes), Espírito Santo (3 participantes), Paraíba (2 participantes). Por fim, os estados com participação menos significativa foram Tocantins, Bahia, Mato Grosso do Sul, Rio Grande do Norte, Santa Catarina, Pará e Sergipe, cada um com 1 participante. Assim, nota-se uma grande influência da região Sudeste do país, com

Figura 3 – Distribuição dos participantes por estados brasileiros



Fonte: Elaborado pelo autor (2026).

cerca de 44,57% dos participantes, e conta com a participação das regiões Nordeste, Sul, Centro-Oeste e Norte.

Quanto à questão de gênero (Q3), a maioria dos participantes se identifica como do gênero masculino 84,3% (70/83), enquanto 14,5% (12/83) se identificam como do gênero feminino e 1,2% (1/83) preferiu não se identificar com esses gêneros. Quanto à faixa etária (Q1) os participantes em sua maioria possuem entre 25 e 34 anos 53% (44/83), seguidos por aqueles com 18 a 24 anos (26,5%; 22/83). Adicionalmente, participantes com faixas etárias mais elevadas têm menor representatividade: 35 a 44 anos (16,9%; 14/83), 45 a 54 anos (2,4%; 2/83) e 55 a 64 anos (1,2%; 1/83).

Sobre a formação acadêmica (Q4), a maioria possui pelo menos uma graduação completa, onde 31 (37,3%) participantes possuem ensino superior completo, 18 (21,7%) possuem especialização, 9 (10,8%) possuem mestrado, três (3,6%) são estudantes de Doutorado, dois (2,4%) são estudantes de mestrado e um (1,2%) possui doutorado. Por fim, 19 participantes (22,9%) possuem apenas o ensino médio completo. Em seguida, foi questionado o modelo de trabalho adotado pelos participantes (Q5). Observa-se que

a maioria atua de forma remota 62,7% (5/83), seguida por aqueles em regime híbrido (31,3%; 26/83) e, por fim, presencial (6,0%; 5/83).

Em relação ao papel exercido na organização (Q6), os participantes puderam selecionar múltiplas opções para refletir a diversidade de funções desempenhadas. O papel mais frequente foi *Security Engineer*, selecionado por 40 participantes (48,2%), dos quais 18 também se identificaram como *Penetration Testers*. Em seguida, destacam-se *DevOps Engineers* (32,5%; 27/83), com 12 desses também atuando como *Security Engineers*, e *Penetration Testers* (31,3%; 26/83).

Outros papéis incluem: *Developer* (Backend, Frontend ou Fullstack) com 12 participantes (14,5%), *Solution Architect* com 8 (9,6%), *Project Lead/Manager* com 6 (7,2%), *Data Scientist* com 4 (4,8%) e *Privacy Engineer* com 2 (2,4%). Papéis menos representados (com 1 participante cada, 1,2%) foram: *Test Manager/Tester*, *Requirements Engineer*, *Product Owner*, *Cryptographer* e *UX/UI Designer*. *SOC Analyst*, *Application Security*, *Gerente de Segurança da Informação*, *AppSec* (3 menções), *DevSecOps*, *Security Analyst*, *Cybersecurity*, *Consulting*, *Security Champion*, *Sustentação/Infraestrutura*, *Arquiteto de Segurança*, *Information Security Analyst I / IAM*, e *Tech Lead/Team Leader*.

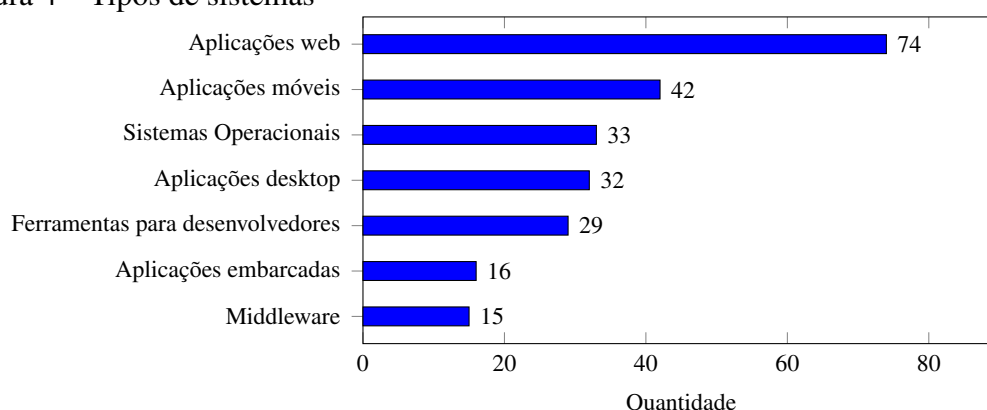
Quanto à senioridade (Q7), 41% (34/83) dos participantes são sênior, com mais de 10 anos de experiência, enquanto 19,3% (16/83) são plenos com entre 6 e 9 anos de experiência. Já 28,9% (24/83) são júnior com até 5 anos de experiência e 10,8% (9/83) são estagiários. Com relação a experiência do participante com Desenvolvimento de Software (Q8), 32 participantes (38,6%) possuem entre 7 e 14 anos de experiência, 24 participantes (28,9%) possuem entre 1 e 3 anos de experiência, 16 participantes (19,3%) possuem entre 4 e 6 anos de experiência, 8 participantes (9,6%) possuem mais de 15 anos de experiência e 3 participantes (3,6%) possuem menos de 1 ano de experiência.

Quanto a experiência em funções relacionadas a Segurança de Software (Q9), 25 (30,1%) participantes possuem entre 1 e 3 anos de experiência, 24 (28,9%) possuem entre 4 e 6 anos de experiência, 17 (20,5%) participantes possuem entre 7 e 14 anos

de experiência, 12 (14,5%) participantes possuem menos de 1 ano de experiência, 4 (4,8%) participantes possuem nenhuma experiência, e 1 (1,2%) participante possui mais de 15 anos de experiência. Os participantes também foram questionados sobre os tipos de sistemas com os quais trabalham (Q10). Essa foi uma pergunta de múltipla escolha, permitindo que os participantes selecionassem várias opções.

A Figura 4 mostra que o tipo mais comum de sistema foi Aplicações web 89,16% (74/83), Aplicações móveis 50,6% (42/83), Sistemas Operacionais 39,76% (33/83), Aplicações desktop 38,55% (32/83), Ferramentas para desenvolvedores 34,94% (29/83), Aplicações embarcadas 19,28% (16/83), e Middleware 18,07% (15/83). Além disso, na opção "Outro", 1 participante mencionou que não atua com desenvolvimento e 1 participante falou que trabalha com todas as aplicações que estavam no questionário. Ainda no campo "Outros" foram citadas: *API*, *Ferramentas de AppSec*, *"Infraestrutura e banco de dados"*, cada uma com uma citação.

Figura 4 – Tipos de sistemas

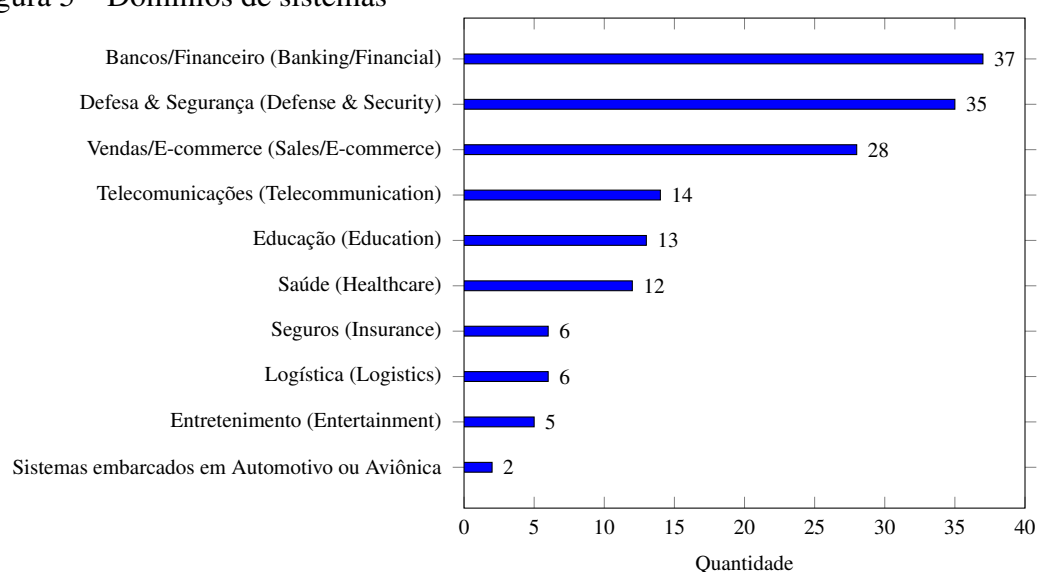


Fonte: Elaborado pelo autor (2026).

Logo em seguida, foi perguntado sobre os domínios dos sistemas em que os participantes trabalham (Q11) que também era de múltipla escolha. A Figura 5 mostra que os domínios mais comuns foram Bancos/Financeiro com 44,58% (37/83) e Defesa & Segurança com 42,17% (35/83). Em seguida vêm os domínios de Vendas/E-commerce

(28/83, 33,73%), Telecomunicações (14/83, 16,87%), Educação (13/83, 15,66%), Saúde (12/83, 14,46%), Seguros (6/83, 7,23%), Logística (6/83, 7,23%), Entretenimento (5/83, 6,02%) e Sistemas embarcados em Automotivo ou Aviônica (2/83, 2,41%). No campo "Outro" foram citados: Sistemas de Tecnologia em Geral, Transporte, Cloud, Consultoria, Educação e e-commerce, Aplicações de software para governos, segurança pública (3 citações), Indústria, Energia e Monitoramento, CLM, Inteligência Artificial (IA) e Processamento de Linguagem Natural (NLP).

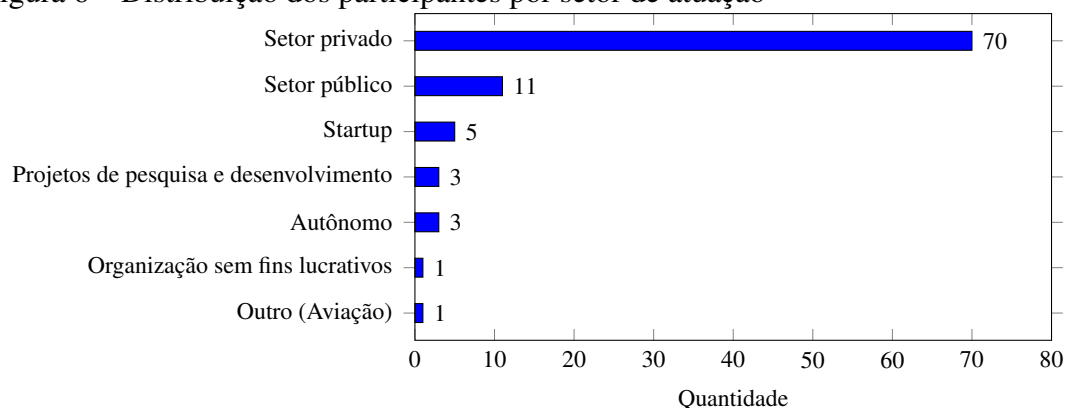
Figura 5 – Domínios de sistemas



Fonte: Elaborado pelo autor (2026).

Caracterização das organizações. Quanto ao setor de atuação das organizações (Q12) em que os participantes trabalham, como mostrado na Figura 6, 84,34% (70/83) dos participantes marcaram que trabalham no setor privado, enquanto 13,25% (11/83) alegaram trabalhar no setor público. Além disso, 6,02% (5/83) trabalham em startup, 3,61% (3/83) trabalham em projetos de pesquisa e desenvolvimento, 3,61% (3/83) são autônomos e apenas um participante, 1,2% (1/83) trabalha em uma organização sem fins lucrativos. No campo "Outro" foi citado: Aviação.

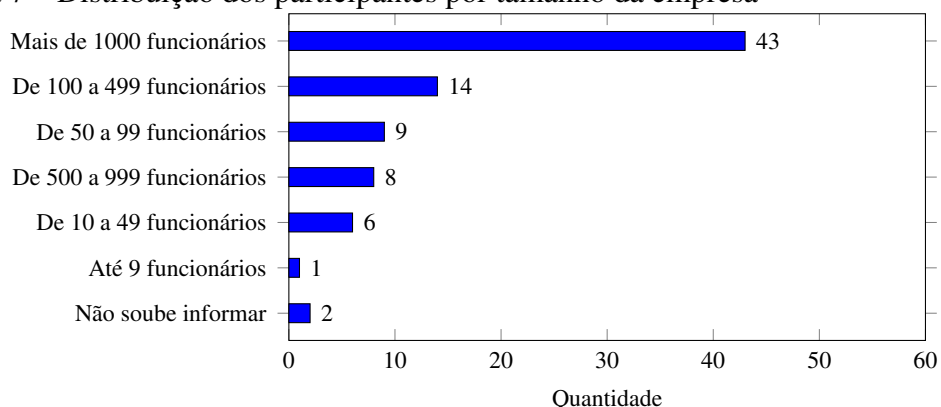
Figura 6 – Distribuição dos participantes por setor de atuação



Fonte: Elaborado pelo autor (2026).

Por fim, foi questionado sobre o tamanho da empresa em que o participante trabalha (Q13). A Figura 7 mostra que a maioria dos participantes (51,8%, 43/83) atua em grandes organizações com mais de 1000 funcionários. Em seguida, 16,9% (14/83) trabalham em empresas de 100 a 499 colaboradores, enquanto 10,8% (9/83) estão inseridos em empresas de 50 a 99 funcionários. Além disso, 9,6% (8/83) atuam em empresas de 500 a 999 funcionários, e 7,2% (6/83) em organizações de 10 a 49 colaboradores. Apenas 1,2% (1/83) trabalham em empresas de até 9 funcionários. Por fim, 2,4% (2/83) afirmaram não saber informar o tamanho da empresa na qual trabalham.

Figura 7 – Distribuição dos participantes por tamanho da empresa



Fonte: Elaborado pelo autor (2026).

Experiência específica com segurança de software no processo de desenvolvimento. O questionário incluiu uma seção dedicada a caracterizar a experiência dos participantes com práticas de Segurança de Software. Logo de início, foi questionado sobre experiência de trabalho diretamente relacionada à segurança no processo de desenvolvimento (Q14). Observou-se que a maioria possui experiência direta com a área, com um total de 62 participantes (74,7%), os quais selecionaram a opção "Eu já trabalhei (ou trabalho) com segurança no processo de desenvolvimento de software." Outros 21 participantes (25,3%) marcaram a opção "Meu grupo/equipe de desenvolvimento trabalha (ou trabalhou) com segurança, mas não estou diretamente envolvido em nenhuma tarefa."

Na sequência, perguntou-se se o participante possui alguma certificação na área de segurança da informação (Q15). A maioria, 48 (47,83%) participantes não possuem nenhuma certificação na área. Entre aqueles com certificações, destacam-se: CompTIA Security+ com 11 (13,25%) participantes; Certified Ethical Hacker (CEH) com 8 participantes (9,64%), e DCPT, ISO 27001, Certified Information Systems Security Professional (CISSP) e Offensive Security Certified Professional (OSCP), cada uma com 3 participantes (3,61%). As certificações eWPTX, Certified Information Security Manager (CISM), CASE, SC-900, AWS Cloud Practitioner e PenTest+ possuem um total de 2 participantes cada (2,42%). Por fim, outras certificações citadas por 1 participante (1,2%) cada incluem: LGPD, Certified Information Privacy Professional (CIPP), Certified Cloud Security Professional (CCSP), Certified AppSec Practitioner (CAP) e CySA+.

Por último, os participantes foram convidados a descrever brevemente sua experiência prática com segurança de software em uma pergunta aberta (Q16). A seguir, apresenta-se um exemplo representativo dos relatos coletados:

"#P24 disse: Tenho experiência com práticas de segurança de software, incluindo a implementação de criptografia para proteger dados sensíveis, validação de entradas para prevenir ataques como SQL injection e XSS, além de utilizar autenticação e autorização seguras para garantir o controle

de acesso adequado. Também tenho conhecimento em testes de segurança, como análise de vulnerabilidades e o uso de ferramentas de segurança para identificar e corrigir falhas"

5.2 Percepção dos Desenvolvedores sobre Práticas de AppSec (QP1)

Para responder a **QP1**, foram conduzidas duas análises complementares: (i) análises de tendências da percepção dos desenvolvedores em relação às quatro dimensões: *frequência de uso, facilidade de uso, efetividade e essencialidade*; (ii) Adicionalmente, foram realizadas análises de correlações entre as diferentes percepções coletadas, bem como as respostas às perguntas abertas em Q18 (práticas adicionais de segurança aplicadas pelos desenvolvedores) e Q19 (como as práticas de segurança têm sido implementadas em seus projetos). Os procedimentos de análise empregados para responder à QP1 são detalhados na Seção 4.2.6.1.

Análise da frequência de uso, facilidade de uso, efetividade e essencialidade das práticas de AppSec. A Tabela 1 mostra a comparação da percepção dos participantes sobre as práticas de AppSec em relação a quatro atributos: *frequência de uso* (Q17), *facilidade de uso* (Q20), *efetividade* (Q21) e *essenciabilidade* (Q22). Para a construção dessa tabela, foram utilizados os pesos atribuídos aos participantes, conforme detalhado previamente na Seção 4.2.6.1, garantindo que as médias refletissem adequadamente a heterogeneidade da amostra em termos de experiência, senioridade e *background* técnico.

A Tabela 1 apresenta os resultados em formato de mapa de calor, no qual tonalidades mais escuras de verde indicam valores mais elevados nos respectivos atributos, ou seja, maior percepção de facilidade, maior efetividade percebida, maior frequência de uso ou maior essencialidade. Por outro lado, tonalidades mais próximas do vermelho representam valores mais baixos, evidenciando práticas percebidas como menos frequentes, menos fáceis, menos efetivas ou menos essenciais.

Tabela 1 – Comparação da frequência de uso, facilidade de uso, efetividade e essencialidade das práticas de AppSec.

Prática de AppSec	Facilidade	Efetividade	Frequência	Essencialidade
Revisão de Código	0.711819	0.827335	0.779046	0.889946
Modelagem de Ameaças	0.644940	0.817032	0.684476	0.853050
Identificação de Casos de Abuso/Mau Uso	0.617958	0.749196	0.740710	0.808108
Análise de Riscos Arquiteturais	0.613141	0.817950	0.653920	0.854048
Testes de Penetração	0.642370	0.879404	0.591110	0.887689
Gerenciamento de Riscos	0.623149	0.840651	0.694755	0.877835
Gerenciamento de Vulnerabilidades	0.704160	0.853873	0.814044	0.899389
Testes de Segurança Manuais/Automatizados	0.682587	0.821981	0.758843	0.865436
Execução de Testes Fuzz	0.626588	0.652547	0.696384	0.670138
Engenharia de Requisitos de Segurança	0.633173	0.814941	0.670668	0.827280
Codificação Segura	0.659280	0.852279	0.814300	0.899239
Resposta Emergencial ao Código	0.602114	0.723812	0.736950	0.761906
Configurações e Gerenciamento de Segurança	0.662384	0.809853	0.745611	0.856081
Monitoramento de Eventos de Segurança	0.642345	0.870926	0.791371	0.874442
Auditorias de Segurança Externas	0.630050	0.712381	0.505349	0.747624
Segurança da Cadeia de Suprimentos de Software	0.611427	0.780249	0.695577	0.835238

Fonte: Elaborado pelo autor (2026).

Analizando os resultados da Tabela 1, nota-se que as práticas percebidas como mais fáceis de aplicar são *Revisão de Código* (71,18%), *Gerenciamento de Vulnerabilidades* (70,41%) e *Testes de Segurança Manuais/Automatizados* (68,25%). As demais práticas, em geral, apresentam um nível moderado de facilidade ou inferior, sugerindo que a adoção de parte dessas atividades ainda envolve desafios de complexidade técnica, esforço adicional ou necessidade de especialização. No que se refere à efetividade percebida, as práticas, de modo geral, apresentam valores altos, indicando forte percepção de que elas contribuem para melhorar a segurança do software. As práticas com destaque quanto à sua efetividade são: *Testes de Penetração* (87,94%), *Monitoramento de Eventos de Segurança* (87,09%), *Gerenciamento de Vulnerabilidades* (85,38%) e *Codificação Segura* (85,22%) que aparecem como as mais bem avaliadas nesse atributo.

Considerando a frequência de uso, os valores mais altos correspondem às práticas mais incorporadas ao fluxo de dia a dia do desenvolvimento: *Codificação Segura* (81,43%), *Gerenciamento de Vulnerabilidades* (81,40%), *Monitoramento de Eventos de Segurança* (79,13%) e *Revisão de Código* (77,90%) são as que obtiveram as melhores

posições no ranking. Por outro lado, práticas percebidas como mais complexas ou dependentes de recursos externos apresentam frequência muito inferior: *Auditorias Externas de Segurança* (50,53%), o valor mais baixo da tabela, *Testes de Penetração* (59,11%), *Execução de Testes Fuzz* (69,63%) e *Análise de Riscos Arquiteturais* (65,39%).

Por fim, a percepção de essencialidade é alta em quase todas as práticas, indicando consenso entre os participantes sobre sua importância para a proteção de sistemas. Entre as práticas consideradas mais essenciais estão: *Gerenciamento de Vulnerabilidades* (89,93%), *Codificação Segura* (89,92%), *Revisão de Código* (88,99%) e *Teste de Penetração* (88,76%). Adicionalmente, pode-se observar que mesmo práticas com baixa frequência de uso mantêm altos níveis de essencialidade, como: *Execução de Testes Fuzz* (67,01%), e *Auditorias de Segurança Externas* (74,76%) e *Resposta Emergencial ao Código* (76,19%). Esses resultados demonstram que, independentemente da dificuldade ou frequência de uso, quase todas as práticas são consideradas importantes para a proteção do sistema.

Finalmente, é importante destacar que a prática *Gerenciamento de Vulnerabilidades* aparece entre as três práticas com melhores resultados em todos os atributos, considerando o conjunto de práticas de AppSec analisadas. Enquanto isso, o uso de *Auditorias de Segurança Externas* (63% em facilidade, 71,23% em efetividade, 50,53% em frequência e 74,76% em essencialidade) esteve sempre entre os piores resultados. Desta forma, essa análise permite a priorização da adoção das práticas com base em sua facilidade de uso, efetividade percebida, frequência de uso e essencialidade.

Achado 1: As práticas de *Revisão de Código*, *Gerenciamento de Vulnerabilidades*, *Codificação Segura* e *Monitoramento de Eventos de Segurança* estão entre as 5 melhores práticas em pelo menos três dos quatro atributos. Por outro lado, práticas como *Auditorias Externas* e *Execução de Testes Fuzz* aparecem entre as últimas colocações em três atributos, apesar de manterem essencialidade moderada-alta. Finalmente, a prática de *Testes de Penetração* é altamente efetiva e essencial, mas pouco frequente, indicando barreiras práticas à adoção.

Análise da correlação entre frequência de uso, facilidade de uso, efetividade e essencialidade das práticas de AppSec. Com o objetivo de compreender de forma mais aprofundada como os atributos das práticas de AppSec se relacionam entre si, foi conduzida uma análise de correlação entre *frequência de uso*, *facilidade de uso*, *efetividade percebida* e *essencialidade* relatadas pelos participantes ($N = 83$). Para orientar a análise estatística, foi formulado um conjunto de seis hipóteses. A Tabela 3 sintetiza as hipóteses alternativas (H_A) e nulas (H_0), permitindo uma visão clara das relações avaliadas em cada teste de correlação. Esses pares de atributos foram examinados individualmente para cada prática de AppSec.

Foi aplicado o teste de *Shapiro-Wilk* (SHAPIRO; WILK, 1965) para avaliar a distribuição de nossos dados. Confirmamos que os dados não seguem uma distribuição normal. Diante desse resultado, optou-se pelo uso do *Spearman's rank correlation coefficient* (WOHLIN *et al.*, 2012). Foi considerado o intervalo de confiança de 95% ($p\text{-valor} < 0,05$). Como resultado, foi obtido um $p\text{-valor} < 0,01$ para todas as práticas de AppSec em todas as correlações. Portanto, as correlações calculadas apresentam significância estatística para todas as práticas de AppSec.

A Tabela 2 exibe os resultados das correlações. Os valores de correlação foram classificados de acordo com o estudo de Salkind (2012) em que: relação muito forte ($0,80 \leq |\rho| \leq 1,00$); relação forte ($0,60 \leq |\rho| < 0,80$); relação moderada ($0,40 \leq$

Quadro 3 – Hipóteses formuladas para a análise de correlação entre atributos das práticas de AppSec

ID	Hipótese Alternativa (HA)	Hipótese Nula (H0)
H1	Há forte correlação entre a frequência de uso da [prática] <i>n</i> e sua efetividade percebida.	Não há forte correlação entre frequência de uso da [prática] <i>n</i> e efetividade percebida.
H2	Há forte correlação entre a frequência de uso da [prática] <i>n</i> e sua facilidade de uso.	Não há forte correlação entre frequência de uso da [prática] <i>n</i> e facilidade de uso.
H3	Há forte correlação entre a frequência de uso da [prática] <i>n</i> e sua essencialidade percebida.	Não há forte correlação entre frequência de uso da [prática] <i>n</i> e essencialidade percebida.
H4	Há forte correlação entre a efetividade percebida da [prática] <i>n</i> e sua facilidade de uso.	Não há forte correlação entre efetividade percebida da [prática] <i>n</i> e facilidade de uso.
H5	Há forte correlação entre a efetividade percebida da [prática] <i>n</i> e sua essencialidade.	Não há forte correlação entre efetividade percebida da [prática] <i>n</i> e essencialidade.
H6	Há forte correlação entre a facilidade de uso da [prática] <i>n</i> e sua essencialidade percebida.	Não há forte correlação entre facilidade de uso da [prática] <i>n</i> e essencialidade percebida.

Fonte: Elaborado pelo autor (2026).

$|\rho| < 0,60$); relação fraca ($0,20 \leq |\rho| < 0,40$); e relação fraca ou inexistente ($0,00 \leq |\rho| < 0,20$).

Análise da correlação entre Frequência x Efetividade. Ao analisar a Tabela 2 é possível observar que a maior parte das práticas apresenta correlação forte entre frequência de uso e efetividade percebida. Mais especificamente, as práticas de *Revisão de Código* ($\rho=0,737$), *Modelagem de Ameaças* ($\rho=0,669$), *Identificação de Casos de Abuso/Mau Uso* ($\rho=0,695$), *Análise de Riscos Arquiteturais* ($\rho=0,645$), *Testes de Penetração* ($\rho=0,694$), *Gerenciamento de Riscos* ($\rho=0,615$), *Gerenciamento de Vulnerabilidades* ($\rho=0,691$), *Testes de Segurança Manuais/Automatizados* ($\rho=0,722$), *Execução de Testes Fuzz* ($\rho=0,655$), *Codificação Segura* ($\rho=0,605$), *Monitoramento de Eventos de Segurança* ($\rho=0,612$), e *Segurança da Cadeia de Suprimentos de Software* ($\rho=0,612$) possuem correlação forte, indicando que quanto maior a efetividade atribuída pelos desenvolvedores, maior tende a ser a frequência de utilização dessas práticas.

Entretanto, nem todas as práticas seguem esse padrão. Algumas apresentam correlação moderada, como *Engenharia de Requisitos de Segurança* ($\rho=0,563$), *Configurações e Gerenciamento de Segurança* ($\rho=0,572$) e *Auditorias de Segurança Externas*

Tabela 2 – Correlação de Spearman entre frequência de uso, facilidade de uso, efetividade e essencialidade das práticas de AppSec.

Práticas de AppSec	Frequência × Efetividade		Efetividade × Facilidade		Frequência × Facilidade	
	rho	Categoria	rho	Categoria	rho	Categoria
Revisão de Código	0.737	Forte	0.766	Forte	0.702	Forte
Modelagem de Ameaças	0.669	Forte	0.779	Forte	0.731	Forte
Identificação de Casos de Abuso/Mau Uso	0.695	Forte	0.703	Forte	0.725	Forte
Análise de Riscos Arquiteturais	0.645	Forte	0.755	Forte	0.730	Forte
Testes de Penetração	0.694	Forte	0.737	Forte	0.750	Forte
Gerenciamento de Riscos	0.615	Forte	0.714	Forte	0.600	Forte
Gerenciamento de Vulnerabilidades	0.691	Forte	0.729	Forte	0.739	Forte
Testes de Segurança Manuais/Automatizados	0.722	Forte	0.685	Forte	0.627	Forte
Execução de Testes Fuzz	0.655	Forte	0.732	Forte	0.812	Muito forte
Engenharia de Requisitos de Segurança	0.563	Moderada	0.701	Forte	0.698	Forte
Codificação Segura	0.605	Forte	0.711	Forte	0.740	Forte
Resposta Emergencial ao Código	0.374	Fraca	0.729	Forte	0.554	Moderada
Configurações e Gerenciamento de Segurança	0.572	Moderada	0.618	Forte	0.655	Forte
Monitoramento de Eventos de Segurança	0.612	Forte	0.612	Forte	0.732	Forte
Auditorias de Segurança Externas	0.599	Moderada	0.680	Forte	0.733	Forte
Segurança da Cadeia de Suprimentos de Software	0.612	Forte	0.779	Forte	0.741	Forte

Práticas de AppSec	Efetividade × Essencialidade		Facilidade × Essencialidade		Frequência × Essencialidade	
	rho	Categoria	rho	Categoria	rho	Categoria
Revisão de Código	0.755	Forte	0.664	Forte	0.598	Moderada
Modelagem de Ameaças	0.711	Forte	0.703	Forte	0.611	Forte
Identificação de Casos de Abuso/Mau Uso	0.680	Forte	0.648	Forte	0.623	Forte
Análise de Riscos Arquiteturais	0.732	Forte	0.739	Forte	0.652	Forte
Testes de Penetração	0.864	Muito forte	0.738	Forte	0.742	Forte
Gerenciamento de Riscos	0.768	Forte	0.674	Forte	0.575	Moderada
Gerenciamento de Vulnerabilidades	0.855	Muito forte	0.699	Forte	0.675	Forte
Testes de Segurança Manuais/Automatizados	0.847	Muito forte	0.689	Forte	0.713	Forte
Execução de Testes Fuzz	0.752	Forte	0.595	Moderada	0.572	Moderada
Engenharia de Requisitos de Segurança	0.809	Muito forte	0.663	Forte	0.526	Moderada
Codificação Segura	0.712	Forte	0.597	Moderada	0.553	Moderada
Resposta Emergencial ao Código	0.561	Moderada	0.505	Moderada	0.306	Fraca
Configurações e Gerenciamento de Segurança	0.744	Forte	0.589	Moderada	0.505	Moderada
Monitoramento de Eventos de Segurança	0.740	Forte	0.491	Moderada	0.518	Moderada
Auditorias de Segurança Externas	0.621	Forte	0.542	Moderada	0.494	Moderada
Segurança da Cadeia de Suprimentos de Software	0.702	Forte	0.610	Forte	0.541	Moderada

Fonte: Elaborado pelo autor (2026).

($\rho=0,599$). Isso sugere que para essas práticas a percepção de efetividade exerce pouca influência sobre a frequência de uso. Além disso, a prática de *Resposta Emergencial ao Código* apresenta correlação fraca ($\rho=0,374$), indicando baixa associação entre esses dois atributos. Portanto, a hipótese **H1_A** é confirmada apenas para as práticas que exibem correlação forte; para as demais, os resultados rejeitam a **H1_A** (e confirmam a **H1₀**).

Análise da correlação entre Efetividade x Facilidade. De forma geral, é possível observar que todas as práticas apresentaram correlação forte entre efetividade percebida e facilidade de uso, com destaque para as práticas de *Modelagem de Ameaças* ($\rho=0,779$), *Segurança da Cadeia de Suprimentos de Software* ($\rho=0,779$) e *Revisão de*

Código ($\rho=0,766$), que possuem os maiores coeficientes. Tais resultados indicam que quanto maior a facilidade percebida de uma prática, maior tende a ser sua efetividade percebida pelo desenvolvedor. Assim, confirma-se **H2_A** para todas as práticas analisadas.

Análise da correlação entre Frequência x Facilidade. A Tabela 2 mostra que a maioria das práticas apresenta forte correlação entre frequência de uso e facilidade percebida, indicando que práticas vistas como mais fáceis tendem a ser aplicadas com maior frequência pelos desenvolvedores. O destaque é a prática de *Execução de Testes Fuzz* ($\rho=0,812$), sendo a única prática com correlação muito forte, sugerindo que sua adoção depende fortemente da facilidade percebida pelo desenvolvedor.

Outras práticas com correlação forte incluem: *Gerenciamento de Vulnerabilidades* ($\rho=0,739$), *Codificação Segura* ($\rho=0,740$), *Segurança da Cadeia de Suprimentos de Software* ($\rho=0,741$), *Testes de Penetração* ($\rho=0,750$), *Análise de Riscos Arquiteturais* ($\rho=0,730$), *Identificação de Casos de Abuso/Mau Uso* ($\rho=0,725$), *Revisão de Código* ($\rho=0,702$), *Modelagem de Ameaças* ($\rho=0,731$), *Testes de Segurança Manuais/Automatizados* ($\rho=0,627$), *Monitoramento de Eventos de Segurança* ($\rho=0,732$) e *Configurações e Gerenciamento de Segurança* ($\rho=0,655$). Todas essas práticas confirmam **H3_A**, reforçando a relação direta entre facilidade percebida e frequência de uso. A única exceção foi para a prática de *Resposta Emergencial ao Código* que apresenta correlação moderada ($\rho=0,554$). Assim, para essa prática nega-se **H3_A** e confirma-se **H3₀**. Para todas as demais práticas, a hipótese **H3_A** é confirmada.

Análise da correlação entre Efetividade x Essencialidade. A Tabela 2 também mostra que quatro práticas se destacam com correlação muito forte: *Testes de Penetração* ($\rho=0,864$), *Gerenciamento de Vulnerabilidades* ($\rho=0,855$), *Testes de Segurança Manuais/Automatizados* ($\rho=0,847$) e *Engenharia de Requisitos de Segurança* ($\rho=0,809$). Esses resultados indicam que, para essas práticas, quanto maior a efetividade percebida pelos desenvolvedores, maior é a probabilidade de serem consideradas essenciais. As demais práticas apresentam correlação forte. A única prática que se distancia desse pa-

drão é *Resposta Emergencial ao Código*, que apresenta correlação moderada ($\rho=0,561$). Isso sugere que, nesse caso, a percepção de essencialidade não depende exclusivamente da efetividade percebida, possivelmente por se tratar de uma atividade reativa, acionada sob demanda e não integrada ao fluxo do dia a dia do desenvolvimento. Dessa forma, confirma-se **H4_A** para todas as práticas que exibem correlação forte ou muito forte; para a prática *Resposta Emergencial ao Código*, confirma-se **H4₀**, negando-se **H4_A**.

Análise da correlação entre Facilidade x Essencialidade. A Tabela 2 indica que a maior parte das práticas apresenta correlação forte entre *facilidade de uso percebida* e *essencialidade*. Entre os maiores coeficientes, destacam-se as práticas de *Análise de Riscos Arquiteturais* ($\rho=0,739$) e *Testes de Penetração* ($\rho=0,738$), seguidas por *Modelagem de Ameaças* ($\rho=0,703$), *Identificação de Casos de Abuso/Mau Uso* ($\rho=0,648$), *Gerenciamento de Vulnerabilidades* ($\rho=0,699$), *Gerenciamento de Riscos* ($\rho=0,674$), *Testes de Segurança Manuais/Automatizados* ($\rho=0,689$), *Revisão de Código* ($\rho=0,664$), *Engenharia de Requisitos de Segurança* ($\rho=0,663$) e *Segurança da Cadeia de Suprimentos de Software* ($\rho=0,610$). Para todas essas práticas, a percepção de facilidade tende a estar associada a níveis maiores de essencialidade.

Por outro lado, algumas práticas exibem correlação apenas moderada. Entre elas estão *Execução de Testes Fuzz* ($\rho=0,595$), *Codificação Segura* ($\rho=0,597$), *Configurações e Gerenciamento de Segurança* ($\rho=0,589$), *Monitoramento de Eventos de Segurança* ($\rho=0,491$), *Auditorias de Segurança Externas* ($\rho=0,542$) e *Resposta Emergencial ao Código* ($\rho=0,505$). Nesses casos, a essencialidade pode estar mais relacionada a fatores políticas internas da organização ou pela natureza reativa da prática, do que à facilidade de uso em si. Dessa forma, confirma-se **H5_A** para as práticas que apresentam correlação forte e confirma-se **H5₀** (negando **H5_A**) para aquelas com correlação moderada.

Análise da correlação entre Frequência x Essencialidade. Finalmente, a análise da Tabela 2 mostra que apenas algumas práticas apresentam correlação forte entre frequência de uso e essencialidade percebida, como *Testes de Penetração* ($\rho=0,742$),

Análise de Riscos Arquiteturais ($\rho=0,652$), *Testes de Segurança Manuais/Automatizados* ($\rho=0,713$), *Gerenciamento de Vulnerabilidades* ($\rho=0,675$), *Identificação de Casos de Abuso/Mau Uso* ($\rho=0,623$) e *Modelagem de Ameaças* ($\rho=0,611$).

A maioria das práticas, porém, apresenta correlação moderada. Entre elas estão *Revisão de Código* ($\rho=0,598$), *Gerenciamento de Riscos* ($\rho=0,575$), *Execução de Testes Fuzz* ($\rho=0,572$), *Engenharia de Requisitos de Segurança* ($\rho=0,526$), *Codificação Segura* ($\rho=0,553$), *Configurações e Gerenciamento de Segurança* ($\rho=0,505$), *Monitoramento de Eventos de Segurança* ($\rho=0,518$), *Auditorias de Segurança Externas* ($\rho=0,494$) e *Segurança da Cadeia de Suprimentos de Software* ($\rho=0,541$). Por fim, apenas a prática de *Resposta Emergencial ao Código* apresenta correlação fraca ($\rho=0,306$), indicando que, para essa prática, a percepção de essencialidade tem pouca relação com a sua adoção no cotidiano, possivelmente por ser acionada apenas em cenários emergenciais. Dessa forma, confirma-se **H6_A** apenas para as práticas com correlação forte; para todas as demais, confirma-se **H6₀** (negando-se **H6_A**).

Achado 2: As práticas de AppSec apresentaram padrões consistentes de correlação. A maioria, 75% (12/16), apresentou forte correlação entre *frequência de uso* e *efetividade percebida*. Todas as práticas (16/16) mostraram forte correlação entre *efetividade* e *facilidade de uso*. Além disso, 93,75% (15/16) apresentaram forte correlação entre *frequência* e *facilidade de uso*. Na relação entre *efetividade* e *essencialidade*, 94% (15/16) exibiram correlação forte ou muito forte. Por fim, a relação entre *frequência de uso* e *essencialidade*, apenas 38% (6/16) tiveram correlação forte, enquanto 56% (9/16) foram moderadas e 6% (1/16) fraca.

Práticas de AppSec específicas nas correlações. A maioria das práticas de AppSec apresenta correlações fortes ($0,60 \leq |\rho| < 0,80$) especialmente nas relações que envolvem efetividade, evidenciando que práticas percebidas como eficazes também são consideradas essenciais, fáceis o suficiente para serem aplicadas e adotadas com maior

frequência pelos desenvolvedores. Práticas como *Modelagem de Ameaças*, *Análise de Riscos Arquiteturais*, *Testes de Penetração*, *Gerenciamento de Vulnerabilidades*, *Testes de Segurança Manuais/Automatizados* e *Execução de Testes Fuzz* apresentam correlações fortes simultaneamente nas três primeiras dimensões (Frequência–Efetividade, Efetividade–Facilidade e Frequência–Facilidade). Além disso, a maioria delas também exibe correlações fortes entre Efetividade–Essencialidade, com destaque para *Testes de Penetração* ($p=0,864$) e *Gerenciamento de Vulnerabilidades* ($p=0,855$).

Adicionalmente, algumas práticas, como *Engenharia de Requisitos de Segurança*, apresentam correlações moderadas em Frequência–Efetividade, sugerindo menor adoção apesar da percepção positiva de efetividade e importância. Já *Resposta Emergencial ao Código* exibe as correlações mais fracas do conjunto, indicando uso menos consolidado. No geral, os resultados reforçam que percepções de efetividade e essencialidade estão fortemente associadas ao uso real das práticas de AppSec.

Achado 3: Práticas de AppSec como *Modelagem de Ameaças*, *Análise de Riscos Arquiteturais*, *Testes de Penetração* e *Gerenciamento de Vulnerabilidades* aparecem como amplamente eficazes e adotadas, apresentando forte relação entre essencialidade, efetividade, facilidade e frequência de uso, indicando a consolidação dessas práticas no fluxo de desenvolvimento seguro. Por outro lado, práticas como *Engenharia de Requisitos de Segurança* e *Resposta Emergencial ao Código* apresentam correlações mais fracas, sugerindo a necessidade de aprimoramentos em usabilidade, integração ao processo de desenvolvimento e conscientização sobre seus benefícios, de modo a incentivar uma adoção mais consistente.

Análise das interseções entre frequência de uso, facilidade de uso, efetividade e essencialidade. Foi analisado também a ocorrência de cada prática de AppSec em quatro categorias: (1) *Mais essencial, mais eficaz, mais usado e mais fácil de usar*; (2) *Menos essencial, menos eficaz, menos usado e menos fácil de usar*; (3) *Mais essencial,*

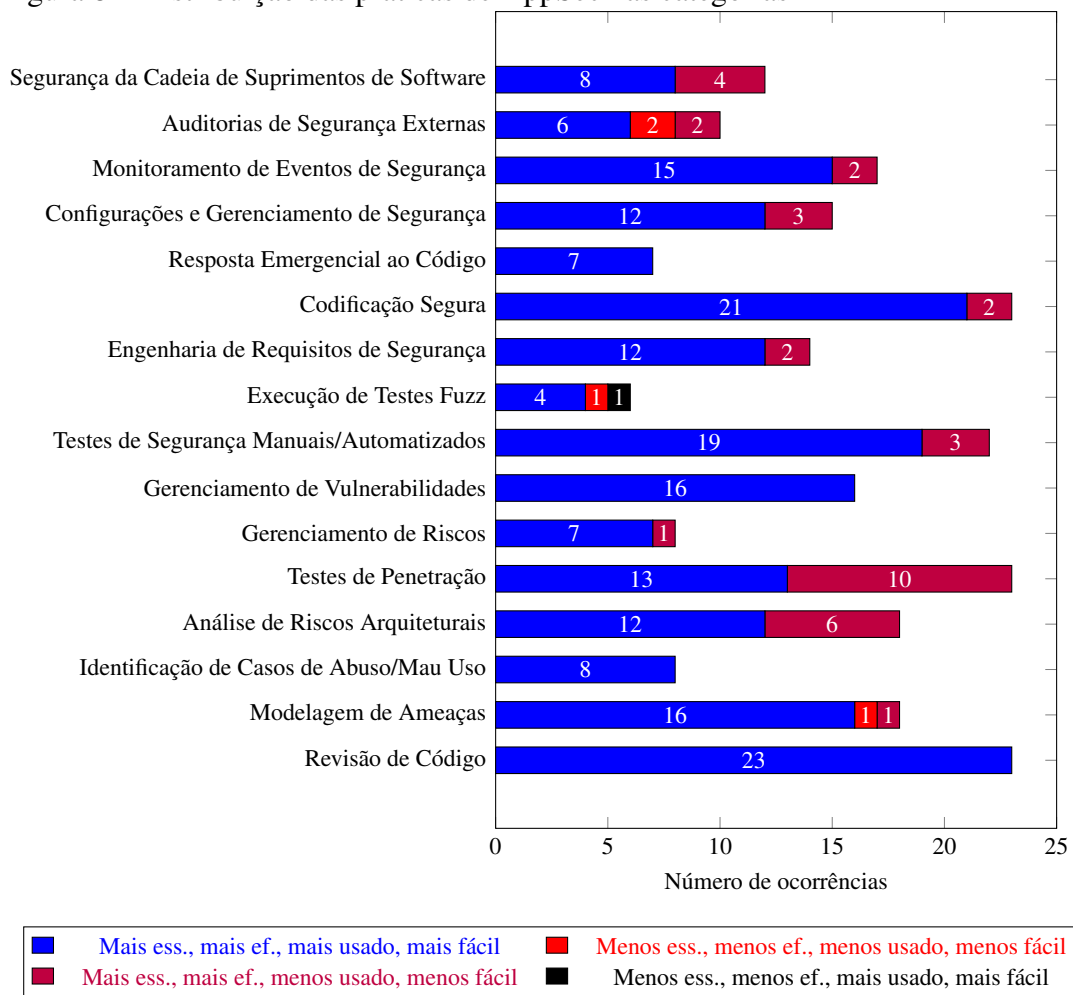
mais eficaz, menos usado e menos fácil de usar; e (4) *Menos essencial, menos eficaz, mais usado e mais fácil de usar*. A classificação foi realizada com base nas escalas de Likert utilizadas no questionário.

Para os atributos *essencialidade, efetividade e facilidade de uso*, a escala variava de 1 a 5, sendo que os valores 5 e 4 foram interpretados como indicadores de maior força do atributo (por exemplo, maior essencialidade ou maior facilidade), enquanto os valores 1 e 2 foram interpretados como indicadores de menor força. No caso específico do atributo *frequência de uso*, cuja escala variava de 1 a 4, adotou-se o seguinte critério: os valores 4 e 3 foram considerados como representação de maior frequência de uso, enquanto os valores 2 e 1 foram considerados como baixa frequência. A Figura 8 apresenta a distribuição da frequência das práticas de AppSec dentro dessas quatro categorias.

Analisando a Figura 8, percebe-se que todas as práticas de AppSec ocorrem pelo menos quatro vezes na categoria (1) *Mais essencial, mais eficaz, mais usado e mais fácil de usar*. Mais especificamente, *Revisão de Código* (23 ocorrências), *Codificação Segura* (21 ocorrências), *Testes de Segurança Manuais/Automatizados* (19 ocorrências), *Modelagem de Ameaças* (16 ocorrências) e *Gerenciamento de Vulnerabilidades* (12 ocorrências) foram as práticas com maior índice de ocorrência. Esse padrão sugere que essas práticas têm uma maturidade consolidada e ampla aceitação entre os desenvolvedores. Já na categoria (2) *Menos essencial, menos eficaz, menos usado e menos fácil de usar*, apenas três práticas aparecem: *Auditorias de Segurança Externas* (2 ocorrências), *Modelagem de Ameaças* (1 ocorrência) e *Execução de Testes Fuzz* (1 ocorrência).

Na categoria (3) *Mais essencial, mais eficaz, menos usado e menos fácil de usar*, um total de 11 práticas apareceram, indicando um grupo de práticas consideradas importantes, porém pouco utilizadas e percebidas como mais complexas. Nesse conjunto, as práticas com maiores ocorrências foram: *Testes de Penetração* (10 ocorrências), *Análise de Riscos Arquiteturais* (6 ocorrências) e *Segurança da Cadeia de Suprimentos*

Figura 8 – Distribuição das práticas de AppSec nas categorias



Fonte: Elaborado pelo autor (2026).

de Software (4 ocorrências). Por fim, na categoria (4): *Menos essencial, menos eficaz, mais usado e mais fácil de usar*, observa-se a ocorrência de apenas uma única prática, *Execução de Testes Fuzz*, que ocorre uma única vez nessa categoria.

Achado 4: Práticas como *Revisão de Código*, *Codificação Segura* e *Testes de Segurança Manuais/Automatizados* mostram forte alinhamento entre os quatro atributos avaliados. Em contraste, práticas como *Testes de Penetração*, *Análise de Riscos Arquiteturais* e *Segurança da Cadeia de Suprimentos* revelam uma lacuna entre alta essencialidade e baixa frequência de uso, indicando desafios de integração dessas práticas no fluxo de desenvolvimento. Já práticas como *Auditorias Externas* e *Testes Fuzz* aparecem nas categorias menos favoráveis, sugerindo baixa percepção de valor ou limitações de aplicabilidade.

Práticas adicionais de segurança de software reportadas pelos participantes. Também foi questionado aos participantes se eles utilizam outras práticas de segurança de software, além das mencionadas (Q18). Dada a natureza optativa da Q18, apenas 13 participantes responderam, mesmo assim, as respostas revelam um conjunto diverso de iniciativas complementares, distribuídas em sete categorias analíticas. Quadro 4 sintetiza essas categorias, subcategorias e suas frequências.

Entre as práticas mais citadas, destacam-se ações voltadas à ***Deteção, Hardening e Resposta a Incidentes***, com ênfase na configuração segura de servidores, monitoramento contínuo e uso de firewalls. Como exemplificado por #P11, tais práticas envolvem a “*configuração adequada de servidores, monitoramento contínuo e o uso de firewalls*”, o que reflete uma preocupação com a proteção da infraestrutura e com a continuidade operacional. Em um nível mais avançado, alguns participantes também mencionaram táticas de detecção proativa, como “*Threat hunting*” (P8), demonstrando atenção a ameaças que podem escapar de ferramentas automatizadas tradicionais.

Outra dimensão fortemente representada diz respeito à ***Educação, Cultura e Capacitação em Segurança***, evidenciando que muitos profissionais entendem a segurança como um processo sociotécnico. Vários participantes relataram treinamentos regulares de sensibilização, especialmente sobre *phishing* e engenharia social (#P11),

Quadro 4 – Práticas adicionais de segurança de software aplicadas pelos participantes

Categoria	Subcategoria	#
1. Engenharia de Arquitetura Segura e Ameaças	Arquitetura Segura / Design Resiliente	1
	Análise de Ameaças Direcionadas e Vetores de Fraude	1
2. Integração Contínua de Segurança (DevSecOps)	Pipeline DevSecOps e Maturidade SDL	1
	Gates de Segurança Automatizados (SAST/DAST)	1
3. Segurança de Terceiros e <i>Supply Chain</i>	Avaliação de Riscos em Aplicações de Terceiros	1
4. Educação, Cultura e Capacitação em AppSec	Treinamento Contínuo e Conscientização (<i>Awareness</i>)	1
	Formação Técnica sobre Vulnerabilidades e Boas Práticas	2
	Programas de Certificação e Formação Especializada (OWASP/SANS)	1
	Capacitação e Desenvolvimento de Competências em AppSec	1
5. IA Aplicada à Engenharia de Segurança	IA para Análise de Código e Arquitetura Segura	1
6. Gestão de Identidades e Proteção Criptográfica	Autenticação Robusta (MFA)	1
	Gestão de Identidades e Acessos (IAM)	1
	Criptografia de Dados em Trânsito e Repouso	1
7. Detecção, <i>Hardening</i> e Resposta a Incidentes	<i>Hardening</i> de Infraestrutura e Controles de Borda	1
	<i>Threat Hunting</i> e Detecção Proativa	1
8. Ausência de Práticas Complementares	Sem Adoção Adicional / Resposta Indefinida	2

Fonte: Elaborado pelo autor (2026).

bem como formações técnicas sobre vulnerabilidades e mitigação (#P51). Houve também menção a programas estruturados de capacitação contínua e desenvolvimento de competências em AppSec (#P53), além de treinamentos especializados oferecidos com base em referenciais como *Open Worldwide Application Security Project* (OWASP) e *SysAdmin, Audit, Network, and Security* (SANS) (#P82).

No campo da **Engenharia de Arquitetura Segura e Análise de Ameaças**, surgiram relatos voltados à arquitetura segura de sistemas e à identificação de mecanismos de fraude direcionada. O participante #P9 mencionou a atuação em “*arquitetura de*

segurança” e no *“levantamento de ameaças direcionadas a fraude”*, indicando práticas alinhadas ao fortalecimento estrutural e ao enfrentamento de vetores de ataque mais específicos e sofisticados. Além disso, os participantes demonstraram preocupação com aspectos de ***Gestão de Identidades e Proteção Criptográfica***, destacando o uso de criptografia para dados em trânsito e em repouso (#P79), autenticação multifator (#P11) e gestão de identidades e acessos (IAM) (#P61). Essas práticas complementam controles essenciais para garantir confidencialidade e integridade, sendo amplamente recomendadas por padrões internacionais.

A análise também destaca práticas associadas ao uso de ***Inteligência Artificial Aplicada à Engenharia de Segurança***, como relatado por #P41, que *“utiliza IA para análise de arquitetura e código”*. Adicionalmente, alguns participantes descreveram práticas relacionadas à ***Integração Contínua de Segurança (DevSecOps)***, como a *“implementação de SAST/DAST e o uso de gates de segurança em pipelines”* (#P80), além de ações de *“gestão de maturidade do SDL”* (#P53). Por fim, emergiram também práticas associadas à *Segurança de Terceiros e Supply Chain*, como mencionado pelo #P70, que realiza *“avaliação de aplicações de terceiros [...] e reprovar contratação de soluções com risco elevado”*, o que revela a preocupação com vulnerabilidades introduzidas via dependências externas ou soluções adquiridas.

Implementação das práticas de segurança de software no ciclo de desenvolvimento. Para compreender como os desenvolvedores efetivamente implementam práticas de segurança de software em seus projetos, o questionário incluiu a questão aberta Q19: *“Como você tem implementado as práticas de segurança de software mencionadas anteriormente em seus projetos?”*. O objetivo desta pergunta foi capturar estratégias, processos, ferramentas e comportamentos adotados na prática, permitindo revelar aspectos que não emergiriam apenas das questões fechadas (Q17, e Q20-Q22). Como descrito na Seção 4.2.6.1 as respostas foram analisadas por meio de codificação qualitativa aberta e axial. A análise qualitativa gerou um conjunto estruturado de oito

categorias, que variam desde modelos de segurança profundamente integrados ao ciclo de desenvolvimento até práticas mais reativas, parciais ou mesmo inexistentes. Esses padrões evidenciam diferentes graus de maturidade em AppSec entre os participantes. Quadro 5 sintetiza essas categorias, subcategorias e suas frequências.

Quadro 5 – Formas de implementação das práticas de segurança de software (Q19)

Categoria	Subcategoria	#
1. Integração da Segurança ao Ciclo de Desenvolvimento	Segurança por Design	11
	Governança de Segurança no Desenvolvimento	10
	Revisões de Código com Foco em Segurança	8
	Modelagem de Ameaças e Riscos Arquiteturais	8
	Segurança Integrada ao Fluxo Ágil	6
	Antecipação da Segurança (Shift Left)	4
2. Automação de Segurança e Ferramentas Especializadas	Scanners Automatizados (SAST, DAST, SCA)	13
	Segurança Automatizada em CI/CD	11
	Automação Customizada e Scripts de Segurança	9
3. Monitoramento Contínuo e Observabilidade de Segurança	Observabilidade e Análise de Logs	9
	Monitoramento e Detecção de Ameaças (IDS/SIEM)	5
4. Testes de Segurança e Avaliações Independentes	Testes de Penetração (Pentest)	8
	Auditorias Internas e Externas de Segurança	5
5. Conformidade, Normas e Padrões de Segurança	Adoção de Frameworks de Mercado (CIS, NIST, etc.)	5
	Conformidade Legal e Regulatória	3
	Referenciais OWASP (Top 10, ASVS, SAMM)	3
	Políticas e Diretrizes Internas de Segurança	4
6. Cultura Organizacional e Capacitação em Segurança	Treinamento e Desenvolvimento em Segurança	5
	Comunicação e Papéis de Referência (Security Champions)	5
	Consultorias e Acompanhamento Especializado	3
7. Controles Técnicos e Infraestrutura Segura	Gestão de Credenciais e Secrets	4
	Proteção Criptográfica e Controles de Autenticação	4
	Controles de Acesso e Proteção de Bordas	4
	Práticas de Codificação Segura	3
	Infraestrutura Segura em Ambientes Cloud	3
8. Adoção Parcial ou Inconsistente de Práticas de Segurança	Execução Ad Hoc de Práticas de Segurança	7
	Ausência de Implementação Formal	4
	Indefinido / Resposta Genérica	3

Fonte: Elaborado pelo autor (2026).

A categoria mais frequente emergiu em torno da ***Integração da Segurança***

ao Ciclo de Desenvolvimento, especialmente por meio de abordagens como *Security by Design* e *Shift Left*, com vários participantes relatando que introduzem requisitos, controles e preocupações de segurança já nas fases iniciais do projeto. Por exemplo, #P25 disse: “*Começo integrando princípios de segurança ao ciclo de desenvolvimento desde a fase de planejamento, adotando o conceito de security by design.*”. Adicionalmente, #P44 reforça que aplica segurança “*desde as fases iniciais do desenvolvimento, garantindo que a segurança seja integrada ao ciclo de vida do software*”.

Em ambientes mais complexos, essa integração ocorre até mesmo em sistemas legados, como mencionado pelo #P82: “*Quando o SSDLC é introduzido em um ambiente onde as aplicações já existem, seguimos um processo gradual e estruturado para incorporar segurança ao longo do ciclo de vida do software [...]*”. O movimento de antecipação (*Shift Left*) também aparece associado à redução de custos e prevenção precoce de falhas, como mencionado pelo #P20: “*a fim de minimizar custos de correções*”, que utiliza *shift left* para minimizar retrabalho, ou pelo #P47, que destaca que está “*fazendo o shift left de segurança, de forma que o pensamento seguro ocorra o quanto antes no SDLC.*”

Além disso, a integração da segurança se materializa em práticas concretas como *Modelagem de Ameaças*, *Revisões de Código com Foco em segurança*, *Governança de Segurança no Desenvolvimento* e *Segurança Integrada ao Fluxo Ágil*. Por exemplo, #P24 relata que “[...] *utilizo modelagem de ameaças para antecipar possíveis vetores de ataque. [...] faço análise de riscos arquiteturais para prevenir vulnerabilidades no design.*”. As revisões de código são frequentes e destacadas como parte natural do processo, como mencionado pelo #P2: “*Sou responsável por assegurar isso no código que entrego*”, ou #P7, que menciona “*code reviews a cada sprint*”. Há também processos formais de governança, com #P14 descrevendo “[...] *comitê contínuo para remediação de pendências e vulnerabilidades.*”, e integração à agilidade, com #P55 afirmando que “[...] *tais práticas são inseridas de acordo com a necessidade.*”.

A segunda categoria, *Automação de Segurança e Ferramentas Especializadas*, revela uma forte dependência de pipelines, scanners e automações, refletindo práticas de DevSecOps, no qual a segurança é cada vez mais operacionalizada por meio de ferramentas automatizadas, pipelines integrados e *scripts* personalizados que ampliam a cobertura e reduzem o esforço manual. Neste sentido, os desenvolvedores relataram a *Adoção de Scanners como SAST, DAST e SCA*, utilizados para detectar vulnerabilidades de forma contínua. Essa automação aparece como estratégia para garantir escala, repetibilidade e rapidez, como mencionado pelo #P57: “[...] Tenho integrado ferramentas como SAST, DAST e SCA nos pipelines CI/CD [...]”.

Outro aspecto interessante é a integração dessas *ferramentas nos pipelines CI/CD*, onde a segurança passa a ser parte das entregas contínuas. Por exemplo, #P14 mencionou “*Pipeline com validações automáticas e scans de segurança [...]*”, enquanto #P70 reforçou que a segurança está embutida na “*esteira de desenvolvimento com a inclusão de ferramentas automáticas e/ou processos incorporados em diferentes fases do processo de desenvolvimento [...]*”. Além das ferramentas, muitos desenvolvedores mencionam a criação de *scripts* próprios, uso de ferramentas open-source e adaptações customizadas, especialmente quando há restrições de orçamento. Neste caso, o #P56, que cita a utilização de “[...] *ferramentas de software livre como DefectDojo, snyk, sonarqube, Infishal para gestão de secrets, algumas automações usando tenable [...]*”.

A terceira categoria, *Monitoramento Contínuo e Observabilidade de Segurança*, evidencia que muitos participantes compreendem a segurança como um processo vivo, sustentado por métricas, logs e mecanismos de detecção ativa. Neste contexto, as citações evidenciam que o monitoramento não se restringe à coleta de eventos, mas envolve uma abordagem sistemática de observabilidade, em que logs, métricas e telemetrias são analisados continuamente para identificar anomalias, uso indevido e potenciais incidentes. Por exemplo, o #P30 descreve práticas centradas em “[...] *análise de métricas e de logs para identificar possíveis mau-usos [...]*”, enquanto que o #P39 relatou que utiliza “[...]”

ferramentas de monitoramento como Prometheus e Grafana”, ampliando a visibilidade operacional de serviços distribuídos. Outros reforçam o uso de *stacks* avançados, #P26: “[...] Também aplico detecção de ameaças com regras personalizadas no ElastAlert2 e Packetbeat para análise de tráfego.[...]”, indicando maturidade e profundidade técnica no tratamento de eventos de segurança.

Além disso, um subconjunto significativo dos participantes aponta a adoção de ferramentas especializadas de detecção e resposta, como IDS, SIEM e firewalls avançados. Neste caso o #P44 descreveu “[...] usando SIEM (ex.: Splunk, ELK Stack) para detecção de anomalias”, enquanto #P11 complementa ao detalhar sistemas de “sistemas de detecção de intrusão (IDS), logs de auditoria [...]” como parte da rotina. Há ainda casos que sugerem uma postura mais ativa, com monitoramento contínuo via WAF (#P52) e plataformas como Wazuh (#P67).

A quarta categoria, **Testes de Segurança e Avaliações Independentes**, evidencia a necessidade de validar continuamente a segurança por meio de testes sistemáticos, tanto internos quanto conduzidos por equipes especializadas ou por terceiros. Por exemplo, o #P24 afirma que “[...] Conduzo testes de penetração regulares [...]”, reforçando uma postura preventiva e recorrente. Outros descrevem *pentests* executados sob demanda ou de forma obrigatória, como #P43, ao mencionar que “a cada nova aplicação, testes são obrigatoriamente realizados”. Há também casos em que a responsabilidade recai diretamente sobre o próprio desenvolvedor como #P65, que afirma: “Sou responsável por fazer todos os testes de segurança necessários e assim comunicar aos responsáveis e solicitar suas correções [...]”.

Os participantes também mencionaram a condução de auditorias internas e externas, para analisar a postura geral de segurança, verificar conformidade com padrões e identificar fragilidades estruturais. Por exemplo, #P44 evidencia auditorias internas recorrentes “[...] para garantir que as práticas adotadas estão sendo seguidas e documentadas”, enquanto #P34 descreve “[...] auditorias externas para validação

e controle da cadeia de suprimentos para reduzir riscos com dependências.” Já #P60 relata a atuação de “[...] analistas externos que realizam tentativas de invasão no sistema e também analisam possíveis vulnerabilidades na rede [...]” como parte do processo de segurança contínua. Adicionalmente, em organizações com menos recursos, auditorias aparecem integradas ao uso de ferramentas open-source e testes esporádicos, como mencionado pelo #P56: “Somos uma startup com um orçamento limitado, então utilizamos preferencialmente ferramentas de software livre e algumas automações [...] testes de penetração esporádicos [...]”.

A quinta categoria, **Conformidade, Normas e Padrões de Segurança**, evidencia que muitos participantes estruturam suas práticas a partir de referenciais amplamente reconhecidos no mercado. Frameworks como CIS, NIST, CSF e PCI-DSS aparecem como guias centrais para definição de controles e avaliação de riscos, como exemplificado por #P23: “[...] aplicando frameworks conhecidos de mercado como CIS, NIST, CSF,[...]”. Em paralelo, surgem também referências a normas e regulamentações, especialmente LGPD e ISO 27001, indicando que parte das equipes atua em ambientes que demandam aderência a requisitos legais e processos de conformidade, como mencionado pelo #P44: “LGPD e ISO 27001: Nos projetos que lidam com dados de usuários, implemento pseudonimização e registros de consentimento para atender aos requisitos legais”. Além das normas externas, diversas citações revelam a existência de políticas e diretrizes internas, incluindo PPSIs, processos validados por times de segurança e acompanhamento sistemático via KPIs, como descrevem #P15, #P74 e #P75. Por fim, material da OWASP (Top 10, ASVS, SAMM) aparece como referência recorrente e prática para orientar desenvolvedores e priorizar controles, como mencionado pelo #P45: “[...] aplicação de controles recomendados para mitigar riscos como OWASP Top 10 e ASVS [...]”.

A sexta categoria, **Cultura Organizacional e Capacitação em Segurança**, destaca a dimensão humana e colaborativa da segurança de software, mostrando que

muitos participantes reconhecem que práticas técnicas só se sustentam quando acompanhadas de treinamentos e comunicação contínua dentro das equipes. Os participantes enfatizaram a importância da capacitação, vista como estratégia essencial para reduzir vulnerabilidades e elevar a maturidade dos times. O participante #P53 mencionou “*capacitação sobre falhas comuns e recorrências, tendências e como prevenir*”.

Além da formação, os participantes destacam a relevância da comunicação entre os desenvolvedores, frequentemente mediada por papéis de referência como *security champions*. O participante #P47 enfatiza esse aspecto ao relatar que sua atuação envolve “*uma relação próxima com os devs, fazendo o shift left de segurança, de forma que o pensamento seguro na engenharia de software ocorra o quanto antes no SDLC*”. Complementando essa perspectiva, há também menções claras ao uso de consultorias ou *advisory* internos, como mencionado por #P53 ao afirmar que realiza “*entrevistas e consultorias aos times sobre defeitos no processo, que acarretam falhas de segurança*”.

A sétima categoria, **Controles Técnicos e Infraestrutura Segura**, evidencia o conjunto de práticas que os participantes aplicam diretamente em nível técnico para proteger sistemas, dados e ambientes de execução. Por exemplo, a utilização de gestão segura de credenciais e secrets, com o #P52, que relata “*gerenciamento de secrets com vault*”. A utilização de criptografia e autenticação reforçada também foi mencionada, e.g., #P25 mencionou [...] *implemento controles de segurança robustos, como autenticação multifator, criptografia de dados sensíveis e validação rigorosa de entradas [...]*. Além disso, há forte presença de práticas de defesa em profundidade, com implementação de WAF, IAM, controle de acessos e firewalls. #P52 mencionou [...] *monitoramento ativo das aplicações [...] utilizando waf [...]*. As respostas também evidenciam atenção à codificação segura (e.g., sanitização, uso de *prepared statements* (#P44)) e ao fortalecimento de infraestrutura em nuvem, como o participante #P39, que descreve [...] *tenho utilizado as ferramentas disponibilizadas pela AWS para gerenciar toda a estrutura envolvida nos projetos, práticas de segurança recomendadas pela empresa [...]*.

Finalmente, a oitava categoria evidencia casos em que a implementação de práticas de segurança ocorre de forma pontual, superficial ou inexistente, revelando lacunas importantes de maturidade. Parte dos participantes adota medidas apenas sob demanda, de maneira reativa, como exemplificado por #P71, que menciona práticas “*realizadas de forma reativa*”. Outros indicam explicitamente a ausência de processos formais, alinhando-se ao grupo *Não Implementa ou Não Aplica*. Há ainda respostas genéricas, como simples “sim”, que não permitem identificar práticas concretas.

Achado 5: A implementação da segurança entre os participantes revela diferentes cenários de maturidade em AppSec. De um lado, há desenvolvedores que incorporam segurança de forma madura e estratégica, integrando-a desde o design, automatizando controles no CI/CD e aplicando monitoramento contínuo. De outro, surgem práticas fragmentadas, reativas ou até inexistentes, muitas vezes limitadas a ações pontuais ou genéricas. Esse cenário evidencia que, embora as práticas recomendadas sejam amplamente reconhecidas, sua execução varia drasticamente, revelando diferentes níveis de maturidade na aplicação de AppSec no ciclo de desenvolvimento.

5.3 Influência de Experiência, Função e Porte Organizacional nas Percepções sobre Práticas de AppSec (QP2)

Para responder à **QP₂**, analisou-se o nível de concordância entre os desenvolvedores quanto às 16 práticas de segurança de software, considerando diferentes grupos de participantes. A comparação foi estruturada em quatro dimensões de percepção: (1) *Facilidade de uso* (Q20), que avalia o quão fácil os desenvolvedores consideram a utilização de cada prática em seus fluxos de trabalho; (2) *Efetividade* (Q21), mensura o grau em que cada prática é percebida como capaz de prevenir, remover ou detectar problemas de segurança nos projetos; (3) *Frequência de uso* (Q17), indica com que

regularidade os participantes adotam cada prática no seu cotidiano de desenvolvimento, permitindo identificar padrões de uso real; e (4) *Essencialidade* (Q22), avalia o nível de importância atribuído a cada prática para a proteção do sistema, independentemente de sua facilidade ou frequência de adoção. Essa análise multidimensional permitiu identificar diferenças significativas nas percepções entre grupos com distintos níveis de experiência, funções desempenhadas e portes organizacionais. Os procedimentos de análise de dados empregados para responder à *QP*₂ são detalhados na Seção 4.2.6.2.

Influência de fatores individuais e organizacionais nas percepções sobre práticas de AppSec. A Tabela 3 apresenta um panorama geral dos resultados sobre o nível de concordância que os diferentes grupos de desenvolvedores possuem em relação às 16 práticas mencionadas, cada prática analisada sob as quatro dimensões de percepção descritas. Utilizamos uma codificação de cores para indicar qual grupo tinha maior probabilidade de concordar com cada afirmação: a cor **cinza escuro** indica que o primeiro grupo apresentou maior nível de concordância, enquanto a cor **cinza claro** indica que o segundo grupo teve maior probabilidade de concordância.

Tabela 3 – Resultados da influência de experiência, função e porte organizacional nas percepções sobre práticas de AppSec

Dimensão	Likert	Direta vs. Indireta				Grande vs. Pequena				Especialista vs. Não especialista			
		Média	Média	p-value	d	Média	Média	p-value	d	Média	Média	p-value	d
Análise de Riscos Arquiteturais													
facilidade	2.49	3.05	2.73	0.20	(+)*	3.07	2.79	0.17	(+)*	2.86	3.06	0.72	(-)
efetividade	3.58	4.14	4.38	0.74	(-)	4.18	4.20	0.69	(-)	4.27	4.14	0.44	(+)
frequência	1.94	2.65	2.00	0.03	(+)**	2.59	2.38	0.17	(+)*	2.52	2.57	0.52	(-)
essencialidade	4.25	4.26	4.24	0.76	(-)	4.25	4.31	0.64	(-)	4.27	4.25	0.62	(-)
Auditorias de Segurança Externas													
facilidade	2.19	3.22	2.67	0.09	(+)*	3.18	3.00	0.35	(+)	3.17	3.12	0.38	(+)
efetividade	2.67	3.69	3.33	0.16	(+)*	3.64	3.69	0.62	(-)	3.44	3.72	0.84	(-)*

Continua.

(continuação) Resultados da influência de experiência, função e porte organizacional nas percepções sobre práticas de AppSec

Dimensão	Likert	Direta vs. Indireta				Grande vs. Pequena				Especialista vs. Não especialista			
		Média	Média	p-value	d	Média	Média	p-value	d	Média	Média	p-value	d
frequência	1.18	1.91	2.00	0.53	(-)	1.77	2.36	0.95	(-)*	1.81	1.97	0.71	(-)
essencialidade	3.76	3.81	3.62	0.39	(+)	3.71	3.94	0.71	(-)	3.81	3.74	0.42	(+)
Codificação Segura													
facilidade	2.59	3.41	2.93	0.06	(+)*	3.36	3.23	0.40	(+)	3.10	3.40	0.85	(-)*
efetividade	3.48	4.35	4.50	0.59	(-)	4.37	4.42	0.75	(-)	4.05	4.52	0.99	(-)*
frequência	2.41	3.24	3.45	0.71	(-)	3.15	3.75	0.96	(-)*	3.05	3.38	0.96	(-)*
essencialidade	4.47	4.53	4.29	0.56	(-)	4.43	4.62	0.57	(-)	4.50	4.46	0.92	(-)*
Configurações e Gerenciamento de Segurança													
facilidade	2.64	3.38	3.08	0.17	(+)*	3.41	3.00	0.11	(+)*	3.38	3.29	0.40	(+)
efetividade	3.25	4.08	4.14	0.64	(-)	4.13	4.00	0.26	(+)	3.79	4.21	0.97	(-)*
frequência	2.08	2.96	2.80	0.27	(+)	2.85	3.25	0.88	(-)*	2.78	3.00	0.79	(-)
essencialidade	4.28	4.27	4.29	0.82	(-)	4.26	4.31	0.55	(-)	4.27	4.28	0.73	(-)
Engenharia de Requisitos de Segurança													
facilidade	2.42	3.26	2.80	0.08	(+)*	3.28	3.14	0.30	(+)	3.00	3.29	0.89	(-)*
efetividade	3.48	4.14	4.38	0.84	(-)*	4.24	4.06	0.41	(+)	4.00	4.28	0.96	(-)*
frequência	1.71	2.67	1.83	0.02	(+)**	2.56	2.67	0.64	(-)	2.59	2.58	0.49	(+)
essencialidade	4.12	4.16	4.00	0.73	(-)	4.09	4.19	0.47	(+)	4.19	4.09	0.60	(-)
Execução de Testes Fuzz													
facilidade	2.10	3.27	2.33	0.03	(+)**	3.25	3.00	0.19	(+)*	3.06	3.21	0.69	(-)
efetividade	2.39	3.38	3.14	0.28	(+)	3.35	3.40	0.57	(-)	3.29	3.38	0.63	(-)
frequência	1.45	2.85	2.00	0.14	(+)**	2.79	2.78	0.49	(+)	2.69	2.85	0.66	(-)
essencialidade	3.39	3.44	3.24	0.37	(+)	3.42	3.25	0.26	(+)	3.69	3.25	0.08	(+)*
Gerenciamento de Riscos													
facilidade	2.45	3.17	2.91	0.30	(+)	3.14	3.21	0.59	(-)	3.10	3.14	0.45	(+)

Continua.

(continuação) Resultados da influência de experiência, função e porte organizacional nas percepções sobre práticas de AppSec

Dimensão	Likert	Direta vs. Indireta				Grande vs. Pequena				Especialista vs. Não especialista			
		Média	Média	p-value	d	Média	Média	p-value	d	Média	Média	p-value	d
efetividade	3.54	4.29	4.15	0.10	(+)*	4.26	4.21	0.63	(-)	4.00	4.38	0.94	(-)*
frequência	1.98	2.78	2.44	0.16	(+)*	2.64	3.00	0.89	(-)*	2.71	2.74	0.53	(-)
essencialidade	4.37	4.31	4.57	0.96	(-)*	4.32	4.50	0.75	(-)	4.35	4.39	0.80	(-)
Gerenciamento de Vulnerabilidades													
facilidade	2.95	3.48	3.58	0.61	(-)	3.51	3.53	0.52	(-)	3.36	3.56	0.74	(-)
efetividade	3.78	4.29	4.36	0.41	(+)	4.30	4.21	0.35	(+)	4.18	4.35	0.82	(-)
frequência	2.64	3.30	2.62	0.00	(+)**	3.12	3.40	0.87	(-)*	3.32	3.12	0.21	(+)
essencialidade	4.49	4.50	4.48	0.85	(-)	4.48	4.56	0.60	(-)	4.42	4.53	0.90	(-)
Identificação de Casos de Abuso/Mau Uso													
facilidade	2.35	3.11	2.45	0.02	(+)**	2.96	3.14	0.71	(-)	2.89	3.04	0.69	(-)
efetividade	3.19	3.82	3.92	0.59	(-)	3.89	3.64	0.33	(+)	3.70	3.90	0.82	(-)
frequência	1.86	2.96	2.14	0.02	(+)**	2.86	2.82	0.39	(+)	3.07	2.77	0.20	(+)
essencialidade	4.05	4.06	4.00	0.69	(-)	4.05	3.94	0.29	(+)	4.15	4.00	0.31	(+)
Modelagem de Ameaças													
facilidade	2.46	3.28	3.00	0.18	(+)*	3.24	3.23	0.40	(+)	3.19	3.26	0.47	(+)
efetividade	3.24	4.18	3.90	0.22	(+)*	4.22	3.86	0.22	(+)	3.94	4.21	0.94	(-)*
frequência	1.86	2.80	1.83	0.02	(+)**	2.53	3.36	1.00	(-)**	2.94	2.61	0.09	(+)*
essencialidade	4.23	4.32	3.95	0.23	(+)	4.12	4.56	0.96	(-)*	4.19	4.25	0.78	(-)
Monitoramento de Eventos de Segurança													
facilidade	2.51	3.24	3.29	0.52	(-)	3.33	3.00	0.27	(+)	3.55	3.11	0.10	(+)*
efetividade	3.57	4.42	4.40	0.43	(+)	4.44	4.31	0.50	(+)	4.19	4.52	0.99	(-)*
frequência	2.27	3.21	2.83	0.10	(+)*	3.04	3.54	0.94	(-)*	3.05	3.17	0.76	(-)
essencialidade	4.41	4.35	4.57	0.96	(-)*	4.42	4.38	0.40	(+)	4.38	4.42	0.79	(-)
Resposta Emergencial ao Código													

Continua.

(continuação) Resultados da influência de experiência, função e porte organizacional nas percepções sobre práticas de AppSec

Dimensão	Likert	Direta vs. Indireta				Grande vs. Pequena				Especialista vs. Não especialista			
		Média	Média	p-value	d	Média	Média	p-value	d	Média	Média	p-value	d
facilidade	2.01	3.07	2.89	0.41	(+)	3.10	2.85	0.37	(+)	2.50	3.22	0.98	(-)**
efetividade	2.59	3.83	3.10	0.05	(+)*	3.72	3.67	0.47	(+)	3.36	3.82	0.90	(-)*
frequência	1.47	2.89	3.00	0.61	(-)	2.77	3.27	0.84	(-)*	2.82	2.94	0.66	(-)
essencialidade	3.84	3.89	3.71	0.55	(-)	3.86	3.75	0.19	(+)	3.96	3.79	0.47	(+)
Revisão de Código													
facilidade	3.08	3.60	3.40	0.37	(+)	3.62	3.53	0.34	(+)	3.09	3.76	0.99	(-)**
efetividade	3.82	4.20	4.33	0.67	(-)	4.19	4.33	0.78	(-)	4.00	4.33	0.97	(-)*
frequência	2.82	3.19	2.88	0.07	(+)*	3.05	3.40	0.90	(-)*	3.12	3.12	0.57	(-)
essencialidade	4.45	4.55	4.14	0.13	(+)	4.38	4.69	0.91	(-)*	4.35	4.49	0.94	(-)*
Segurança da Cadeia de Suprimentos de Software													
facilidade	2.14	3.02	3.38	0.79	(-)*	3.18	2.67	0.14	(+)*	3.00	3.10	0.65	(-)
efetividade	2.87	3.94	3.70	0.20	(+)*	3.92	3.75	0.52	(-)	3.88	3.91	0.55	(-)
frequência	1.61	2.73	2.80	0.53	(-)	2.63	3.10	0.92	(-)*	2.60	2.79	0.77	(-)
essencialidade	4.11	4.23	3.76	0.23	(+)	4.06	4.19	0.68	(-)	4.35	4.00	0.11	(+)*
Testes de Penetração													
facilidade	2.60	3.19	3.10	0.46	(+)	3.19	3.14	0.45	(+)	3.00	3.27	0.79	(-)
efetividade	3.64	4.45	4.00	0.06	(+)*	4.40	4.29	0.32	(+)	4.32	4.40	0.79	(-)
frequência	1.72	2.42	1.67	0.03	(+)**	2.30	2.29	0.45	(+)	2.38	2.27	0.35	(+)
essencialidade	4.36	4.44	4.14	0.35	(+)	4.40	4.19	0.14	(+)*	4.50	4.30	0.29	(+)
Testes de Segurança Manuais/Automatizados													
facilidade	2.87	3.38	3.50	0.64	(-)	3.52	3.00	0.07	(+)*	3.42	3.39	0.51	(-)
efetividade	3.66	4.17	4.14	0.29	(+)	4.07	4.40	0.87	(-)*	4.08	4.20	0.61	(-)
frequência	2.60	3.07	2.67	0.11	(+)*	2.94	3.19	0.78	(-)	2.92	3.04	0.64	(-)
essencialidade	4.30	4.35	4.14	0.67	(-)	4.22	4.56	0.84	(-)	4.31	4.30	0.69	(-)

Fonte: Elaborado pelo autor (2026).

Além disso, cada prática foi anotada para indicar diferenças estatisticamente significativas entre grupos, conforme segue: (1) 🔧 indica diferença significativa entre participantes com experiência de trabalho **direta** e aqueles com experiência **indireta** em segurança de software; (2) 🎓 denota diferença significativa entre os grupos de função **especialista** e **não especialista**; e (3) 🏢 indica diferença significativa entre participantes que trabalham em organizações de **grande** e **pequeno** porte. Para facilitar a interpretação dos resultados, cada atributo avaliado nas práticas também foi anotado por um ícone específico: 😊 *Facilidade de uso*, ✔️ *Efetividade*, ⌚ *Frequência de uso*, e ★ *Essencialidade*.

Por exemplo, a prática [Auditorias de Segurança Externas] mostra que não há diferenças significativas entre nenhum dos grupos de participantes em relação à percepção de 😊 facilidade de uso, ✔️ efetividade, ⌚ frequência de uso e ★ essencialidade. Por outro lado, [🔧 ⌚ Análise de Riscos Arquiteturais] indica que há uma diferença significativa entre os grupos que têm experiência direta e indireta com segurança de software quanto à percepção sobre frequência de uso.

🔧 **Experiência Direta vs. Indireta em Trabalho com Segurança de Software.** De forma geral, a experiência direta com segurança mostrou-se o fator mais associado a diferenças nas percepções e, sobretudo, no uso efetivo das práticas. Em termos de **facilidade de uso** (😊), participantes com experiência direta tenderam a avaliar práticas mais avançadas como menos complexas. Essa diferença é particularmente evidente em práticas como [🔧 😊 Execução de Testes Fuzz] e [🔧 😊 Identificação de Casos de Abuso/Mau Uso], por apresentarem diferenças estatisticamente significativas entre participantes com experiência **direta** e **indireta** em segurança de software.

No caso de *Execução de Testes Fuzz*, a facilidade média passou de aproximadamente 2,10 (experiência indireta) para 3,27 (experiência direta); já em *Casos de Abuso/Mau Uso*, a média aumentou de 2,35 para 3,11, com valores de p próximos de 0,03–0,02 e efeitos positivos de pequena a moderada magnitude. Esses resultados refor-

çam que a exposição direta às atividades de segurança contribui para reduzir barreiras de entrada percebidas em práticas mais técnicas e cognitivamente exigentes, tornando-as mais acessíveis e intuitivas para esses profissionais.

No que se refere à efetividade percebida (👍), poucas práticas apresentaram diferenças estatisticamente significativas. Uma exceção relevante foi [🔧👍 **Resposta Emergencial ao Código**], em que participantes com experiência direta atribuíram maior efetividade às ações corretivas rápidas ($p \approx 0,05$). Esse resultado sugere que profissionais que já vivenciaram incidentes reais tendem a reconhecer com maior clareza o valor dessas intervenções imediatas no controle de danos e na restauração da segurança do sistema.

As maiores diferenças podem ser observadas na frequência de uso (👤). Desenvolvedores com experiência direta relatam utilizar com maior frequência um conjunto de práticas centrais em AppSec ao longo do ciclo de desenvolvimento. Foram observadas diferenças significativas em [🔧👤 **Análise de Riscos Arquiteturais**], [🔧👤 **Engenharia de Requisitos de Segurança**], [🔧👤 **Gerenciamento de Vulnerabilidades**], [🔧👤 **Identificação de Casos de Abuso/Mau Uso**], [🔧👤 **Modelagem de Ameaças**] e [🔧👤 **Testes de Penetração**]. Esses achados indicam que a experiência direta não apenas molda a percepção dos desenvolvedores sobre as práticas, mas também se reflete em uma adoção mais sistemática e consistente ao longo das etapas de desenvolvimento.

Por outro lado, para o atributo essencialidade (★), as diferenças entre experiência direta e indireta foram mínimas e estatisticamente não significativas. Ambos os grupos atribuíram valores elevados (tipicamente acima de 4,0), indicando um consenso generalizado sobre a importância dessas práticas de segurança, independentemente do nível de envolvimento prévio com atividades de AppSec. Esses resultados sugerem que, embora a experiência direta influencie substancialmente 😊 *como as práticas são percebidas* (especialmente em facilidade) e 👤 *com que intensidade são utilizadas* (frequência), ela não altera de forma relevante o reconhecimento compartilhado de que essas práticas são fundamentais para a proteção dos sistemas.

Organizações Grandes vs. Pequenas. No agrupamento por porte organizacional, nenhum dos quatro atributos, facilidade (☺), efetividade (☑), frequência de uso (☹) ou essencialidade (★), apresentou diferenças estatisticamente significativas ($p \geq 0,05$). As médias permaneceram próximas entre organizações grandes e pequenas, com variações leves que não configuraram efeitos robustos. Em algumas práticas, organizações maiores apresentaram ligeiramente maior facilidade ou frequência, enquanto em outras organizações menores exibiram valores um pouco superiores. Ainda assim, a ausência de diferenças significativas indica que o porte não se mostra um fator determinante na percepção ou adoção das práticas de AppSec na amostra analisada.

Esse padrão repete-se especialmente na **frequência de uso** (☹), sugerindo que a adoção das práticas pode estar mais fortemente associada à cultura organizacional, aos processos internos ou ao nível de maturidade em segurança, em vez do tamanho formal da organização. Por fim, o atributo **essencialidade** (★) manteve-se elevado e semelhante entre os portes, reforçando um entendimento amplamente compartilhado sobre a importância dessas práticas independentemente do contexto organizacional.

Funções de Especialista vs. Não-Especialista. A comparação entre especialistas e não especialistas revelou que para os quatro atributos, facilidade (☺), efetividade (☑), frequência (☹) e essencialidade (★), as médias foram bastante semelhantes, sem diferenças estatisticamente significativas ($p \geq 0,05$). Embora algumas práticas tenham apresentado tendências não significativas. Por exemplo, especialistas perceberam maior efetividade em *Segurança da Cadeia de Suprimentos* e *Revisão de Código*, ou maior essencialidade em *Testes de Penetração*, tais variações não atingiram o limiar de significância estatística.

No atributo facilidade de uso (☺), especialistas tenderam a avaliar algumas práticas tecnicamente mais complexas como ligeiramente menos fáceis, possivelmente refletindo uma visão mais crítica associada ao conhecimento técnico aprofundado. Entretanto, essas diferenças permaneceram dentro da variabilidade esperada. Por fim, assim

como observado nos outros agrupamentos, o atributo **essencialidade** (★) manteve-se elevado e estável entre especialistas e não especialistas, indicando um consenso generalizado sobre o papel central das práticas de AppSec em processos de desenvolvimento de software com segurança.

Comparações entre grupos. Quando múltiplos agrupamentos (experiência, função e porte organizacional) e múltiplos atributos (facilidade, efetividade, frequência e essencialidade) são analisados de forma conjunta, é possível observar alguns padrões.

Entre os três agrupamentos, a experiência direta vs. indireta com segurança é o fator que produz os padrões mais consistentes quando os atributos são analisados em conjunto. Mesmo quando determinadas práticas não apresentam diferenças significativas em um único atributo, o conjunto dos atributos revela diferenças claras em várias práticas, incluindo: *Análise de Riscos Arquiteturais*, *Engenharia de Requisitos de Segurança*, *Gerenciamento de Vulnerabilidades*, *Identificação de Casos de Abuso/Mau Uso*, *Modelagem de Ameaças*, *Execução de Testes Fuzz*, *Testes de Penetração*, e *Gerenciamento de Riscos*. Nessas práticas, observam-se que os participantes tendem a percebê-las como: (1) maiores níveis de facilidade percebida, (2) maior efetividade atribuída em algumas delas, e, sobretudo, (3) maior frequência de uso entre participantes com experiência direta. Essa observação sugere um efeito multidimensional da experiência prática sobre a adoção e compreensão das práticas de segurança.

No caso do porte organizacional, a análise por atributo sugere poucas diferenças significativas entre organizações de grande e pequeno porte. Contudo, quando práticas e atributos são considerados simultaneamente, surgem alguns padrões sutis, especialmente em práticas relacionadas à operação contínua e detecção, tais como: *Monitoramento de Eventos de Segurança*, *Codificação Segura*, *Gerenciamento de Vulnerabilidades*, e *Validações e Configurações de Segurança*. Neste sentido, foi possível observar que organizações de grandes tendem a apresentar avaliações mais homogêneas nesses casos, enquanto organizações menores exibem maior dispersão nas percepções, su-

gerindo que fatores internos (maturidade dos processos, políticas e cultura de segurança) podem exercer mais influência que o porte isoladamente.

Por fim, o agrupamento por função (especialistas vs. não especialistas) apresenta médias semelhantes quando cada atributo é analisado isoladamente. Contudo, a análise em conjunto tende a revelar um padrão mais complexo, especialmente com as seguintes práticas: *Segurança da Cadeia de Suprimentos*, *Revisão de Código*, *Testes de Segurança Manuais/Automatizados*, *Modelagem de Ameaças*, *Gerenciamento de Riscos*, e *Testes de Penetração*. Neste sentido, foi possível observar uma tendência dos especialistas a atribuírem: menor facilidade, demonstrando uma visão mais crítica baseada em conhecimento aprofundado; e uma maior efetividade e essencialidade, sobretudo em práticas com impacto direto na mitigação de riscos.

Achado 6: A experiência direta (🔑) é o principal fator que diferencia percepções, especialmente em frequência de uso (🕒) e facilidade (😊). O Porte organizacional (🏢) não apresenta impacto significativo nos quatro atributos. Especialização (🎓) influencia pouco as percepções, com diferenças discretas e majoritariamente não significativas. A percepção de essencialidade (★) é consistentemente alta em todos os grupos, indicando forte consenso sobre a importância das práticas.

5.4 Benefícios, Adoção e Desafios das Ferramentas Automatizadas de Segurança no Desenvolvimento de Software (QP3)

Para analisar a adoção de ferramentas automáticas de segurança de software, inicialmente aplicou-se a questão de controle (Q23), que permitiu dividir os participantes em dois grupos distintos: (1) participantes que utilizam ferramentas automáticas de segurança (49/83), ou utilizam parcialmente (20/83) – totalizando 69 participantes com algum nível de adoção; e (2) participantes que não utilizam ferramentas automáticas de segurança (14/83). Essa separação fornece o contexto necessário para interpretar a

frequência de uso, os benefícios percebidos e os desafios enfrentados.

Frequência e contexto de adoção das ferramentas automatizadas de segurança. Para compreender não apenas quais ferramentas automatizadas de segurança são utilizadas, mas também como elas são incorporadas ao fluxo de desenvolvimento, realizamos uma análise cruzada entre as questões Q24 e Q25. A Q24 capturou a frequência de uso com relação a seis categorias de ferramentas automatizadas de segurança – *3rd Party Vulnerability Scanning*, *Static Application Security Testing*, *Dynamic Application Security Testing*, *Secure Configuration/Hardening*, *Compliance/Hardening Checks* e *Secrets Management*. Já a Q25 solicitou aos participantes que indicassem, dentre um conjunto de 53 ferramentas mapeadas em um estudo anterior (ANGERMEIR *et al.*, 2021), quais ferramentas específicas eles utilizam atualmente ou já utilizaram. A Q25 também possuía um campo para "Outros" onde os participantes podiam escrever ferramentas que não estavam presentes dentre as opções de seleção da questão. Porém, nenhuma nova ferramenta foi mencionada.

Esse cruzamento permitiu identificar padrões de adoção tanto dentro quanto entre categorias, revelando, por exemplo, quais ferramentas são utilizadas diariamente e semanalmente e quais são usadas apenas de forma eventual ou não se aplicam ao contexto dos desenvolvedores. A Tabela 4 sumariza as ferramentas automatizadas de segurança agrupadas por categoria, frequência de uso e quantidade de menções.

Para as análises descritas a seguir, foram consideradas apenas as respostas do grupo (1). Na categoria **3rd Party Vulnerability Scanning**, 27 dos 69 participantes do grupo (1) indicaram utilizar esse tipo de ferramenta *Diariamente*, 9 *Semanalmente*, 12 *Mensalmente*, 3 *Anualmente* e 18 marcaram *Não Aplicável*. Ao cruzar esses dados com as ferramentas específicas reportadas na Q25, observa-se que, entre os que declararam uso diário, destacam-se *Snyk* (18 menções), *Trivy* (15) e *Dependabot* (15). Por outro lado, mesmo entre os participantes que marcaram *Não Aplicável* para a categoria, surgem ferramentas como *Sonatype*, *PHP Security Checker*, *Dependabot*, *Node Security Platform*

Tabela 4 – Ferramentas automatizadas de segurança agrupadas por categoria, frequência de uso e quantidade de menções

Categoria da Ferramenta	Diariamente	Semanalmente	Mensalmente	Anualmente	Não aplicável
3rd Party Vulnerability Scanning	Snyk (18) Trivy (15) Dependabot (15) Sonatype (9) Whitesource (4) RetireJS (4) Node Sec. Platform (3) PHP Sec. Checker (3) Clair (1) SRCCLR (1) Blackduck (1)	Snyk (7) Dependabot (6) Trivy (4) RetireJS (1) Anchore (1)	Snyk (6) Trivy (5) Dependabot (2) Blackduck (1) PHP Sec. Checker (1) Clair (1) Sonatype (1)	Snyk (1) PHP Sec. Checker (1) Trivy (1)	Sonatype (2) PHP Sec. Checker (2) Dependabot (2) Node Sec. Platform (2) Trivy (2)
Static Application Security Testing	SonarQube (29) CodeQL (12) ESLint Security (9) Bandit (4) Brakeman (3) Dart Analyzer (3) LGTM (1) CheckStyle (1)	SonarQube (11) CheckStyle (1) ESLint Security (1) CodeQL (1) Bandit (1)	SonarQube (5)	SonarQube (1) CodeQL (1) ESLint Security (1)	SonarQube (8) ESLint Security (1) CodeQL (1)
Dynamic Application Security Testing	Checkmarx (5) Qark (1)	Checkmarx (5) Qark (1)	Checkmarx (5)	Checkmarx (2)	Checkmarx (3) Qark (1)
Secure Configuration / Hardening	Ansible (9) Terraform (8) Puppet (4) Seccomp (2) CloudCustodian (1) Chef (1)	Terraform (5) Ansible (3)	Terraform (8) Ansible (5) Chef (1) Puppet (1)	Terraform (4) Ansible (3) Phan (1)	Terraform (13) Ansible (8) Puppet (3) Chef (1)
Compliance / Hardening Checks	Kube Audit (2) Blackbox (1) InSpec (1)		Kube Audit (3) Blackbox (2)	Puppet-Lint (1)	Puppet-Lint (1) InSpec (1) Foodcritic (1) Kube Audit (1)
Secrets Management	GitLeaks (16) Docker Secrets (10) Trufflehog (10) Trufflehog Conjur (5)	Docker Secrets (2) CyberArk Conjur (2) GitLeaks (1) Trufflehog (1)	Docker Secrets (1) GitLeaks (1)	GitLeaks (2)	Docker Secrets (3) GitLeaks (1)

Fonte: Elaborado pelo autor (2026).

e *Trivy* (duas menções cada), sugerindo que alguns desenvolvedores podem utilizar ferramentas da categoria sem necessariamente percebê-la como aplicável ao seu contexto ou sem associá-la claramente a uma ferramenta de segurança automatizada.

Na categoria **Static Application Security Testing**, 36 participantes indicaram uso *Diariamente*, 12 *Semanalmente*, 6 *Mensalmente*, 2 *Anualmente* e 13 marcaram *Não Aplicável*. Entre os que reportaram uso diário, destacam-se as ferramentas: *SonarQube* (29 menções), *CodeQL* (12) e *ESLint Security* (9). Mesmo entre os participantes que marcaram *Não Aplicável*, foram citados usos de *SonarQube* (8 menções), *ESLint Security* (1) e *CodeQL* (1), reforçando novamente a existência de possíveis lacunas na categorização ou na compreensão do escopo dessas ferramentas. Para a categoria **Dynamic Application Security Testing**, 16 participantes declararam que utilizam *Diariamente*, 18 *Semanalmente*, 13 *Mensalmente*, 4 *Anualmente* e 18 indicaram *Não Aplicável*. Entre os que utilizam essas ferramentas diariamente, destacam-se *Checkmarx* (5 menções) e *Qark* (1). As mesmas ferramentas também aparecem entre os participantes que marcaram *Não Aplicável*, com 3 menções para *Checkmarx* e 1 para *Qark*.

Na categoria **Secure Configuration/Hardening**, 18 participantes indicaram

que utilizam *Diariamente*, 11 *Semanalmente*, 10 *Mensalmente*, 5 *Anualmente* e 25 marcaram *Não Aplicável*. Entre os participantes que usam diariamente, sobressaem duas ferramentas *Ansible* (9 menções) e *Terraform* (8). Curiosamente, entre os que marcaram *Não Aplicável*, *Terraform* e *Ansible* também aparecem com alta frequência (13 e 8 menções, respectivamente), evidenciando que parte dos desenvolvedores utiliza essas ferramentas de infraestrutura e configuração sem necessariamente reconhecê-las como mecanismos diretos de segurança ou *hardening*.

Na categoria **Compliance/Hardening Checks**, 14 participantes indicaram uso *Diário*, 9 *Semanal*, 15 *Mensal*, 4 *Anual* e 27 marcaram *Não Aplicável*. Entre os usuários diários, foram mencionadas ferramentas como *Kube Audit* (2 menções), *Blackbox* (1) e *InSpec* (1). Dentre aqueles que selecionaram *Não Aplicável*, surgem ferramentas como *Puppet-Lint*, *InSpec*, *Foodcritic* e a própria *Kube Audit* (uma menção cada), o que reforça o padrão observado de uso efetivo sem associação direta à categoria conceitual de *compliance* ou *hardening checks*.

Por fim, na categoria **Secrets Management Tools**, 40 participantes indicaram que utilizam *Diariamente*, 12 *Semanalmente*, 3 *Mensalmente*, 4 *Anualmente* e 12 marcaram *Não Aplicável*. Entre os que utilizam essas ferramentas diariamente, destacam-se: *GitLeaks* (16 menções), *Docker Secrets* (10) e *Trufflehog* (10). Mesmo entre os participantes que marcaram *Não Aplicável*, ainda há menções a *Docker Secrets* (3) e *GitLeaks* (1), reforçando que o gerenciamento de segredos está presente em diferentes níveis de maturidade de uso, mas nem sempre é percebido explicitamente como uma categoria específica de ferramenta de segurança.

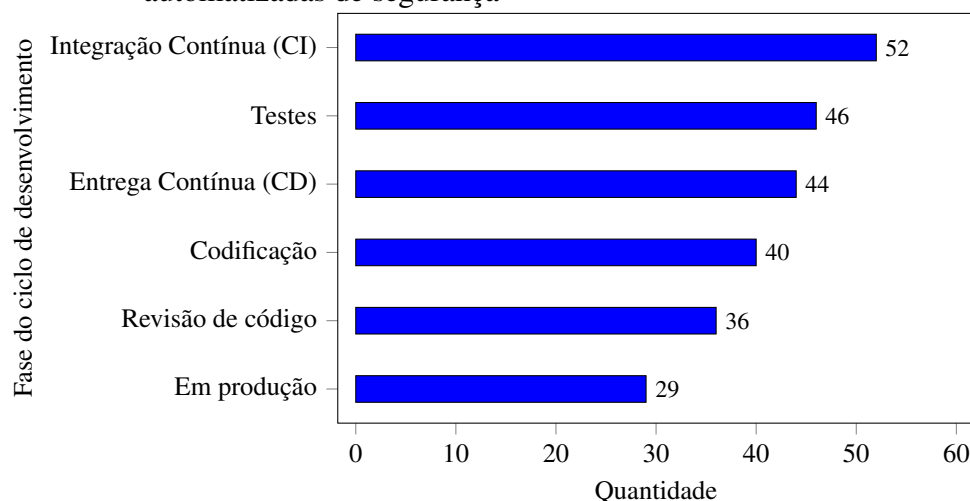
Achado 7: Embora desenvolvedores utilizem ativamente diversas ferramentas de segurança, muitas delas diariamente, como *Snyk*, *Trivy*, *SonarQube*, *CodeQL*, *Terraform* ou *GitLeaks*, parte significativa desses mesmos desenvolvedores marca tais categorias como “*Não Aplicável*”. Esse padrão indica que os desenvolvedores, na prática, incorporam mecanismos de segurança automatizados ao seu fluxo de trabalho, mas nem sempre reconhecem claramente a qual categoria cada ferramenta pertence ou como elas se conectam às práticas formais de AppSec e DevSecOps. Em outras palavras, há uso, mas há também um desalinhamento conceitual que fragmenta a percepção sobre o próprio ecossistema de segurança adotado.

Do ponto de vista prático, esse desalinhamento sugere que a maturidade em segurança não depende apenas da presença de ferramentas, mas também da compreensão do seu papel dentro do ciclo de desenvolvimento. Isso implica que treinamentos, guias internos e documentação técnica precisam enfatizar não apenas como utilizar as ferramentas, mas para quê elas servem e como se relacionam às diferentes classes ou categorias de mecanismos de segurança. Além disso, tornar essas categorias mais explícitas em pipelines, wikis ou processos internos pode melhorar a comunicação entre times e evitar que lacunas importantes passem despercebidas. Por exemplo, equipes que acreditam não realizar *3rd Party Vulnerability Scanning* mesmo utilizando Dependabot diariamente. Logo, a integração efetiva das ferramentas requer não só adoção técnica, mas alinhamento conceitual para fortalecer práticas consistentes de DevSecOps.

Etapas do desenvolvimento com maior adoção de ferramentas automatizadas de segurança. A questão Q26 perguntou diretamente aos desenvolvedores: “*Em quais fases do ciclo de desenvolvimento de software você costuma utilizar as ferramentas de segurança automatizadas selecionadas na pergunta anterior?*” (*múltipla escolha*). O objetivo era mapear em que momentos do processo de desenvolvimento essas ferramentas são efetivamente empregadas na prática. As respostas revelam que as ferramentas

de segurança automatizadas são mais utilizadas nas etapas diretamente vinculadas às práticas de DevOps/DevSecOps, especialmente em atividades contínuas e integradas ao fluxo de entrega. A Figura 9 sintetiza esses resultados.

Figura 9 – Fases do ciclo de desenvolvimento com maior uso de ferramentas automatizadas de segurança



Fonte: Elaborado pelo autor (2026).

Como é possível observar, a Integração Contínua aparece como a fase mais citada (52), reforçando o papel central do CI nas práticas modernas de DevSecOps. Em seguida, destacam-se Testes (46) e Entrega Contínua (CD) (44), reforçando o alinhamento entre segurança e automação. O uso também é expressivo em atividades diretamente ligadas ao desenvolvimento, como Codificação (40 menções) e Revisão de Código (36 menções), evidenciando um movimento de “*shift left*”. Por fim, embora menos frequente, a presença de ferramentas em Produção (29 menções) mostra que parte dos participantes também realiza verificações contínuas ou monitoramento pós-implantação.

Benefícios percebidos e recomendações dos desenvolvedores sobre uso de ferramentas automatizadas de segurança. Para compreender como os desenvolvedores orientam a adoção de práticas automatizadas de AppSec, os participantes foram questionados sobre: “*Quais recomendações você daria para um desenvolvedor sobre a adoção*

de ferramentas de segurança para seus projetos?”. As respostas da Q27 revelaram um conjunto diversificado de recomendações, organizadas em categorias e subcategorias que refletem diferentes perspectivas práticas sobre o uso dessas ferramentas. O Quadro 6 sintetiza essas recomendações fornecidas pelos participantes, desde orientações centradas na automação e integração contínua, até aspectos mais humanos, como capacitação, colaboração com times de segurança e uso criterioso das ferramentas.

Ao analisar o Quadro 6, é possível perceber que as recomendações mais recorrentes são relacionado à categoria de adoção de práticas de **Automação, DevSecOps e Integração em Pipelines**, especialmente por meio da utilização de *ferramentas específicas de SAST, DAST, SCA e plugins integrados ao ambiente de desenvolvimento* (13 menções). Os participantes enfatizam que a automação contínua constitui a base de uma estratégia moderna de desenvolvimento, permitindo identificar vulnerabilidades de forma precoce e sistemática. Como mencionado pelo #P66: *“Tenho pelo menos uma ferramenta para identificação em código (Secret Scanning) [...] SAST e SCA [...] como mínimo obrigatório”*. Isso não apenas fortalece o SDLC, mas reduz significativamente o retrabalho, alinhando-se à literatura que reforça o papel da automação na detecção preventiva de vulnerabilidades (SINAN *et al.*, 2025; TOH *et al.*, 2019).

A importância de integrar essas ferramentas diretamente ao pipeline de CI/CD também é mencionada por vários participantes – *“Configure ferramentas de segurança para rodar automaticamente no pipeline de integração contínua [...]”* e *“Incorpore segurança ao pipeline de CI/CD [...] para detectar e corrigir problemas em fases iniciais.”* (#P44 e #P25, respectivamente). Essa recomendação evidencia que os desenvolvedores reconhecem o valor dos *security gates* ao longo do fluxo DevSecOps.

Outra categoria envolve recomendações para a **Integração Precoce e Contínua da Segurança ao SDLC**, com 12 menções que reforçam a necessidade de iniciar a proteção desde o início das etapas do desenvolvimento. Por exemplo, #P33 mencionou: *“Implementar as melhores práticas de segurança desde o começo pode evitar muitos*

Quadro 6 – Recomendações para adoção de ferramentas de segurança segundo os participantes

Categoria	Subcategoria	#
1. Automação, DevSecOps e Integração em Pipelines	Ferramentas específicas (SAST/DAST/SCA, plugins, IDE)	13
	CI/CD, esteiras e gates de segurança	8
	Monitoramento contínuo e logging de segurança	2
	Gestão automatizada de credenciais e configuração segura	1
2. Complemento Humano: Testes Manuais, Pentests e Bom Senso	Bom senso, pensamento crítico e OWASP Top 10	6
	Testes manuais e pentests	4
3. Cultura, Colaboração e Desenvolvimento Contínuo	Capacitação, aprendizado contínuo e mudança de cultura	6
	Trabalho conjunto e comunicação com time de segurança	6
	Documentação e processos de segurança	1
4. Ferramentas são essenciais, mas não suficientes	Ferramentas como parte da qualidade e não como fim em si	3
5. Integração Precoce e Contínua da Segurança ao SDLC	Adoção desde o início / em todas as fases	10
	Ferramentas como parte essencial do SDLC	2
6. Priorização Baseada em Risco e Impacto	Focar ameaças mais relevantes e custo de correção	5
7. Seleção, Configuração e Limitações das Ferramentas	Esforço de interpretação, suas limitações e falsos positivos	4
	Escolha adequada ao contexto e requisitos do negócio	6
	Ferramentas como apoio, não bala de prata	5
	Manutenção e atualização contínua das ferramentas	2
8. Ausência de recomendação ou não uso	Sem recomendação específica / não uso declarado	3
9. Resposta vaga ou não informativa	Resposta vaga ou não informativa	1

Fonte: Elaborado pelo autor (2026).

problemas futuros [...]” e #P41 recomendou que “*em todo o ciclo de software você deve utilizar as ferramentas de segurança*”. Essas recomendações evidenciam um alinhamento com o paradigma de *shift-left security* (ASSAL; CHIASSON, 2019; SHAHIN *et al.*, 2017). Para esses participantes, inserir segurança apenas ao final do ciclo produz riscos que poderiam ser evitados e aumenta os custos operacionais, enquanto tratá-la

como parte essencial do processo de desenvolvimento amplia a segurança do sistema.

Recomendações relacionadas a fatores humanos também foram mencionadas na categoria ***Cultura, Colaboração e Desenvolvimento Contínuo*** (13 menções). Para muitos participantes, a adoção efetiva de ferramentas depende tanto de aspectos técnicos quanto de fatores sociotécnicos – como treinamento, comunicação e disposição para receber feedback. Recomendações como: “*O maior desafio do desenvolvimento seguro é mudar a cultura do time [...] continue sempre aprendendo.*” (#P72) e “*converse sempre com o time de segurança. O seu trabalho não é ruim por isso, ele só precisa ser lapidado. [...]*” (#P65) ilustram uma percepção de que valores coletivos e práticas colaborativas desempenham no fortalecimento de processos de segurança, em consonância com achados da literatura sobre comportamento de desenvolvedores (ASSAL; CHIASSON, 2019).

Ainda que a maioria veja valor claro na adoção de ferramentas, um conjunto significativo de recomendações ressalta que a automação é um complemento ao julgamento humano. Participantes também destacaram o papel de ***Testes Manuais, Pentests e do Bom Senso*** na compreensão crítica das vulnerabilidades, como mencionado pelo #P47: “*[...] Mas a maior recomendação é a prática de bom senso e pensamento crítico sobre o que está sendo desenvolvido. Olhando o OWASP top 10 [...] quase todas as [vulnerabilidades] são mitigadas com bom senso, validação de inputs e sensibilidade nos dados [...]*”, e pelo #P19: “*As ferramentas podem agilizar bastante [...] porém é necessário que testes manuais continuem sendo realizados para garantir a segurança da aplicação*”. Essas recomendações sugerem que vulnerabilidades relacionadas ao fluxo de trabalho raramente são identificadas automaticamente, exigindo análise humana contextualizada (JOHNSON *et al.*, 2013; DACREMA *et al.*, 2019).

Finalmente, alguns participantes destacam recomendações relacionadas à ***Seleção, Configuração e Limitações das Ferramentas***, especialmente relacionadas a falsos positivos, esforço de interpretação e necessidade de configuração adequada. Por exemplo, comentários como o realizado pelo #P69: “*Nenhuma ferramenta é bala de*

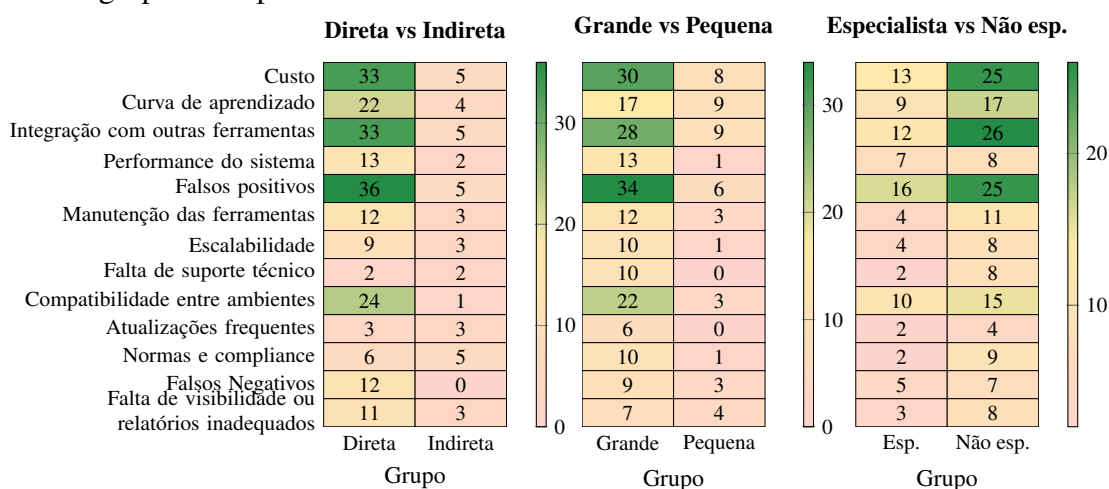
prata. Não adianta rodar infinitas ferramentas e assumir que o produto está seguro.”, reforçam que, para os desenvolvedores, ferramentas são essenciais, mas não constituem solução completa. Essa percepção é consistente com estudos que apontam a importância de calibrar expectativas, até mesmo para especialistas, e compreender o funcionamento interno das ferramentas (KROMBHOLZ *et al.*, 2017).

Achado 8: As recomendações mostram que, embora a automação seja amplamente valorizada e vista como um requisito mínimo para um SDLC moderno, os desenvolvedores reconhecem que seu real impacto depende de integração precoce, uso contínuo e seleção adequada das ferramentas. Paralelamente, enfatizam que a segurança não é alcançada somente por meio de *scanners*, mas também por fatores humanos, tais como pensamento crítico, testes manuais, colaboração com times de segurança e cultura organizacional madura. Essas observações reforçam que a efetividade das ferramentas está fortemente ligada ao quanto são compreendidas, contextualizadas e complementadas por práticas humanas.

Principais desafios mencionados pelos desenvolvedores. Os desenvolvedores foram diretamente questionados sobre quais desafios encontram ao automatizar ferramentas de segurança em seus projetos (Q38). Para compreender como essas percepções variam entre diferentes perfis de desenvolvedores, aplicamos novamente os três agrupamentos utilizados anteriormente: desenvolvedores com *experiência direta* (62) vs. *experiência indireta* (21) com segurança; *especialistas* (26) vs. *não especialistas* (57); e desenvolvedores de *empresas grandes* (65) vs. *empresas pequenas* (16). Esse agrupamento permite observar como grupos distintos interpretam e vivenciam a automação da segurança. A Figura 10 apresenta um mapa de calor que sintetiza os desafios e destaca como cada grupo percebe, em maior ou menor grau, cada um dos desafios avaliados.

A segmentação dos participantes permite observar como diferentes perfis interpretam e vivenciam a automação de ferramentas de segurança. É possível perceber

Figura 10 – Mapa de calor dos desafios na automação de ferramentas de segurança, por grupo de respondentes.



Fonte: Elaborado pelo autor (2026).

que, os participantes com *experiência direta* tendem a perceber com mais frequência que os com *experiência indireta* a maioria dos desafios abordados. Com uma grande diferença entre as respostas dos dois grupos para os seguintes desafios: *Custo* (33 vs. 5 citações), *Curva de aprendizado* (22 vs. 4), *Integração com outras ferramentas* (33 vs. 5), *Falsos positivos* (36 vs. 5) e *Compatibilidade com diferentes ambientes de desenvolvimento* (24 vs. 1). Essa observação sugere que quem lida diretamente com automação vivencia de forma mais clara seus gargalos técnicos e operacionais.

Em relação aos participantes de *empresas grandes* vs *empresas pequenas*, o mesmo padrão pode ser observado, organizações maiores relatam desafios com frequência significativamente mais alta. Mais especificamente, *Custo* (30 vs. 8), *Integração com outras ferramentas* (28 vs. 9), *Falsos positivos* (34 vs. 6) e *Compatibilidade entre ambientes* (22 vs. 3). Por fim, ao comparar as respostas dos participantes *especialistas* e *não especialistas*, observa-se um padrão oposto, em que os *não especialistas* relatam mais desafios no geral. Alguns desafios em que se percebe esse padrão, incluem: *Custo* (13 vs. 25), *Curva de aprendizado* (9 vs. 17), *Integração com outras ferramentas* (12 vs. 26) e *Falsos positivos* (16 vs. 25). Essa observação indica que, embora especialistas

enfrentem barreiras técnicas, os não especialistas percebem mais dificuldades iniciais, desde a compreensão das ferramentas até sua integração no fluxo de trabalho.

Além disso, no campo “Outros”, participantes pertencentes aos grupos *experiência direta, empresa pequena e não especialistas* mencionaram desafios adicionais, como a *ausência de um processo completo de gestão das vulnerabilidades*, isso reforçando que, em alguns contextos, a automação falha não por limitações técnicas da ferramenta, mas devido à ausência de uma infraestrutura organizacional que a sustente.

Achado 9: Desenvolvedores com *experiência direta* relatam níveis muito mais elevados de dificuldades relacionadas a *custo, curva de aprendizado, integração com outras ferramentas, falsos positivos e compatibilidade entre ambientes*. Em empresas maiores, esses desafios tornam-se ainda mais evidente, sugerindo que a complexidade organizacional influencia os desafios. Por outro lado, *não especialistas* relatam mais desafios iniciais do que especialistas, sugerindo que falta de familiaridade agrava problemas percebidos.

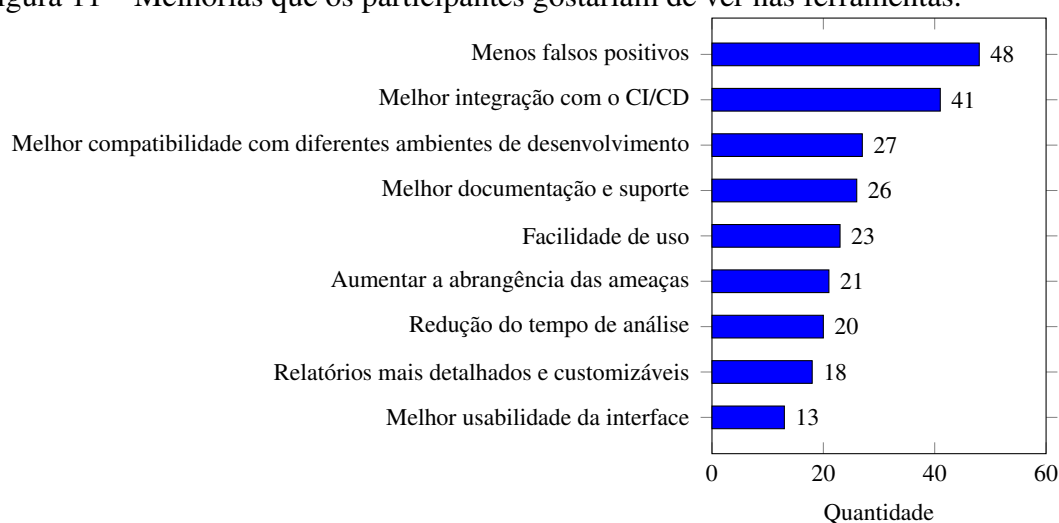
O ponto de vista prático, tais resultados indicam que: (i) ferramentas automatizadas de AppSec devem adaptar suporte e treinamentos conforme o perfil dos desenvolvedores, oferecendo abordagens diferenciadas para especialistas e não especialistas; (ii) organizações maiores precisam investir em integração entre ferramentas e padronização de ambientes para reduzir a fricção operacional entre seus pares; (iii) iniciativas de redução de falsos positivos e simplificação das ferramentas têm impacto especialmente forte nos grupos mais afetados; e (iv) fortalecer processos de gestão de vulnerabilidades é essencial para que a automação opere sobre bases organizacionais mais sólidas.

Principais sugestões de melhorias mencionadas pelos desenvolvedores.

Os desenvolvedores também foram questionados sobre quais melhorias gostariam de ver nas ferramentas automatizadas de segurança que utilizam atualmente (Q29). As

respostas revelaram um conjunto consistente de sugestões voltadas à precisão, integração com o fluxo de desenvolvimento e usabilidade. Como apresentado na Figura 11, as três melhorias mais desejadas foram: *redução de falsos positivos* (48 citações), *melhor integração com pipelines de CI/CD* (41 citações) e *compatibilidade aprimorada com diferentes ambientes de desenvolvimento* (27 citações). Esses resultados reforçam que ferramentas de AppSec precisam ser menos disruptivas e mais plugáveis ao ecossistema DevSecOps para que sua adoção seja considerada sustentável.

Figura 11 – Melhorias que os participantes gostariam de ver nas ferramentas.



Fonte: Elaborado pelo autor (2026).

Além das sugestões mencionadas anteriormente, os participantes também destacaram a necessidade de *melhor documentação e suporte* (26 citações), facilidade de uso (23), *maior abrangência de ameaças detectadas* (21), *redução do tempo de análise* (20) e *relatórios mais detalhados ou customizáveis* (18). Em conjunto, tais sugestões de melhorias refletem a busca por ferramentas que reduzam esforço operacional, retornem resultados mais interpretáveis visando uma maior clareza durante a triagem e mitigação de vulnerabilidades. Os participantes também forneceram sugestões no campo “Outros”. Entre elas aparecem sugestões de melhorias como *feedback e aprendizado contínuo*,

redução de custos, suporte a padrões de reporte como SARIF, customização avançada de regras de detecção, mecanismos de classificação ou priorização baseados em risco, e funis de triagem para findings, cada uma citada uma vez. Essas sugestões indicam um interesse crescente por ferramentas mais ajustáveis ao contexto organizacional e mais alinhadas a processos estruturados de DevSecOps.

Achado 10: As principais melhorias desejadas pelos desenvolvedores concentram-se em: *redução de falsos positivos, integração mais fluida com CI/CD e melhor compatibilidade com diferentes ambientes de desenvolvimento*. Isso sugere que o principal obstáculo não é a falta de ferramentas, mas o custo operacional que elas impõem ao fluxo de desenvolvimento. Outras sugestões incluem, *melhor documentação, maior facilidade de uso e relatórios mais úteis*, reforçam que a automação ainda demanda esforço significativo de interpretação.

Em outras palavras, do ponto de vista prático, tais resultados complementam o achado 10, e reforçam que as ferramentas de AppSec precisam priorizar precisão, integração nativa ao pipeline de CI/CD e suporte à diferentes tecnologias. Desse modo, as equipes de desenvolvimento, por sua vez, podem aumentar a efetividade adotando ferramentas com menor taxa de ruído, promovendo padronização, e incorporando práticas de priorização baseada em risco para reduzir sobrecarga operacional. Sem essas melhorias, a automação tende a permanecer subutilizada ou gerando conflitos no ciclo DevSecOps.

Razões reportadas para a não adoção de ferramentas automatizadas de segurança. Para compreender o porquê da não adoção de ferramentas automatizadas, os participantes foram convidados a responder à questão: “*Quais são as razões pelas quais você opta por não utilizar ferramentas automatizadas de segurança de software? Por favor, descreva brevemente sobre.*”. As respostas da Q31 revelam diferentes justificativas desde fatores organizacionais, limitações técnicas, ausência de conhecimento e

percepções individuais. O Quadro 7 sintetiza as categorias identificadas.

Quadro 7 – Razões para não utilizar ferramentas automatizadas de segurança segundo os participantes

Categoria	Subcategoria	#
1. Percepção Positiva e Ausência de Barreiras	Percepção positiva e ausência de obstáculos	1
	Ênfase nos benefícios das ferramentas	1
2. Barreiras Organizacionais e de Processo	Falta de processo estruturado de segurança (Security Last)	1
	Governança centralizada e dependência do time de segurança	1
3. Escopo de Atuação e Responsabilidade	Função restrita à auditoria	1
	Ferramentas alheias às atividades do cargo	1
	Ausência de atribuição relacionada a ferramentas automatizadas	1
4. Falta de Conhecimento ou Exposição	Baixa exposição ou treinamento limitado	1
	Desconhecimento técnico	1
5. Limitações das Ferramentas	Sobrecarga com falsos positivos e alto custo de interpretação	1
6. Percepção de Não Necessidade	Falta de rotina ou hábito de utilização	1
	Inadequação ao contexto atual da empresa	1
	Não utiliza, sem justificativa específica	1
	Percebida como não essencial para a equipe ou fluxo atual	1

Fonte: Elaborado pelo autor (2026).

Um primeiro grupo de participantes demonstrou **Percepção Positiva e Ausência de Barreiras** sobre a adoção de ferramentas automatizadas de segurança. Como mencionado pelo #P46: “*Não vejo razão pela qual não utilizar, visto que é algo que vai contribuir com a segurança*”. Outro participante reforça essa perspectiva apontando benefícios diretos como “*aumento de produtividade pela automação [e] eficiência no processo de segurança de software*” (#P78). Isso indica que, nesses casos, a não adoção não decorre de resistência pelo não uso, mas possivelmente por fatores externos.

Os participantes também mencionaram **Barreiras Organizacionais**. Por exemplo, o participante #P1 mencionou: “*Não adoção de um processo formal de*

desenvolvimento seguro [...] SegInfo é lembrada ao final da cadeia". Isso evidencia que alguns desenvolvedores trabalham em ambientes onde a segurança é tratada tardiamente ou sem um processo estruturado. Outro participante (#P76) mencionou a dependência de decisões centralizadas: *"O time de segurança que determina as utilizações"*, refletindo limitações impostas pelo modelo organizacional.

Também surgiram razões relacionadas ao ***Escopo de Atuação e Responsabilidade***. Há participantes que não utilizam ferramentas porque suas funções não incluem desenvolvimento ou operação de controles automatizados. Por exemplo, o #P36 afirmou que *"Não utilizo, apenas audito"*, e o #P39 mencionou que as ferramentas *"não estão diretamente ligadas com as minhas tarefas atualmente"*. Essa condição também aparece com o #P61: *"Não atuo com essas ferramentas"*, mostrando que a não adoção, nesses casos, reflete o alinhamento com responsabilidades do cargo. Adicionalmente, surpreendentemente outros dois participantes indicaram uma ***Falta de Conhecimento ou Exposição*** com ferramentas automatizadas, como em #P62: *"Não tenho conhecimento e nunca precisei"* e #P74, que simplesmente declarou *"Desconhecimento"*. Esses casos sugerem lacunas formativas e indicam oportunidades de capacitação.

Do ponto de vista técnico, também foi mencionada uma ***Limitação das Ferramentas***, relacionada ao custo de interpretação dos resultados. Como mencionado pelo #P50: *"Falsos positivos e tempo de uso vs maturidade sobre resultados"*, indicando que o esforço para lidar com alertas pode ser um desincentivo importante em equipes menos maduras. Por fim, razões baseadas na ***Percepção de Não Necessidade*** também foram mencionadas. Alguns participantes relatam não usar ferramentas por falta de hábito, #P17: *"Não uso no dia a dia"*; por inadequação ao contexto, #P34: *"Não se aplica no contexto atual da empresa"*; ou simplesmente por não considerarem essencial, #P48: *"Não uso"*. Adicionalmente, um participante (#P63) descreveu o uso pontual limitado apenas à equipe de Dev, de algumas ferramentas básicas como Sonar, Husky e Dependabot: *"No momento atual não está sendo necessário para a equipe DevOps. O*

máximo que fazemos é colocar alguma ferramenta para análise de vulnerabilidade [...] como Sonar, Husky e Dependabot”.

Intenção futura de adoção de ferramentas automatizadas de segurança.

A análise das questões Q32 e Q33 fornece uma visão importante sobre a abertura dos desenvolvedores, especialmente aqueles que hoje não utilizam ferramentas automatizadas de segurança (grupo identificado na Q23), para adotá-las em projetos futuros. Entre os 14 participantes desse grupo, observou-se que seis afirmaram que pretendem utilizar tais ferramentas, seis responderam “Talvez” e apenas dois declararam que não planejam adotá-las. Esse resultado sugere que, embora exista uma resistência no uso das ferramentas, a maior parte demonstra ao menos uma predisposição à adoção futura, indicando espaço para incentivos quanto ao uso das ferramentas. Ao detalhar quais ferramentas seriam potencialmente utilizadas (Q33), as ferramentas mais citadas entre os participantes foram: *RetireJS*, *SonarQube*, *Blackbox*, *GitLeaks* e *CloudCustodian*, cada uma com quatro citações. Em seguida, aparecem *PHP Security Checker*, *AppScan*, *Terraform*, *Docker Secrets* e *Ansible*, com três menções cada.

Um aspecto interessante é a inconsistência entre intenção declarada e seleção efetiva das ferramentas. Neste contexto, embora apenas dois participantes tenham marcado “Não” na Q32, três marcaram “Nenhuma” na Q33. Isso sugere que parte dos respondentes pode não ter clareza sobre as ferramentas listadas, ou ainda expressar indecisão mesmo após indicar intenção inicial de adoção. Isso reforça o raciocínio da existência de barreiras cognitivas, como desconhecimento das ferramentas, dúvidas sobre aplicabilidade ou falta de confiança nas categorias existentes de ferramentas. Tais barreiras podem influenciar tanto a adoção atual quanto a intenção futura. Em resumo, tais resultados mostram que mesmo entre quem não utiliza ferramentas hoje, existe um potencial considerável de expansão da adoção. Contudo, esse potencial depende diretamente de ações que reduzam incertezas, ampliem familiaridade e tornem as ferramentas mais acessíveis e integradas ao cotidiano dos desenvolvedores.

Achado 11: As razões para não adotar ferramentas automatizadas revelam barreiras sobretudo organizacionais, tais como falta de processos, decisões centralizadas e pouca clareza de responsabilidades, além de lacunas de conhecimento e percepção limitada de necessidade. Isso sugere que iniciativas de AppSec devem priorizar: (i) integrar a segurança ao processo organizacional de forma clara e distribuída; (ii) ampliar o acesso ao conhecimento técnico e reduzir a dependência de especialistas; (iii) ajustar responsabilidades para que a segurança seja parte efetiva do trabalho de desenvolvimento; e (iv) mitigar limitações usuais das ferramentas, como falsos positivos, que reduzem confiança e adesão. Sem essas ações, a adoção continuará fragmentada e condicionada ao contexto de cada equipe.

5.5 Resumo do Capítulo 5

Este capítulo apresentou os principais resultados sobre como desenvolvedores percebem e aplicam práticas e ferramentas de segurança de software. Foi identificado que a experiência direta em segurança, o porte organizacional e o nível de especialização influenciam significativamente as percepções de frequência, facilidade, efetividade e essencialidade das práticas de AppSec. Também foi observado que a adoção de ferramentas automatizadas de segurança varia entre os participantes. De forma geral, apesar do reconhecimento da importância da segurança, tanto práticas quanto ferramentas ainda não se configuram como elementos plenamente integrados e sistemáticos no desenvolvimento cotidiano. O próximo capítulo discute as ameaças à validade e as estratégias adotadas para mitigá-las.

6 AMEAÇAS À VALIDADE DO ESTUDO

Com o intuito de assegurar a validade dos achados, são discutidas a seguir as potenciais ameaças à validade do estudo de acordo com o *framework* proposto por Wohlin *et al.* (2012), contemplando validade de construto, validade de conclusão, validade interna e validade externa. Embora seja impossível eliminar todas as fontes de viés, diversas estratégias foram aplicadas para fortalecer a robustez e a credibilidade dos achados.

Validade de construto. A validade de construto diz respeito ao grau em que os instrumentos e procedimentos utilizados representam adequadamente os conceitos que a pesquisa pretende investigar. Uma primeira ameaça decorre da possível interpretação divergente dos termos e descrições das práticas de segurança de software e das ferramentas automatizadas apresentadas no questionário. Ainda que o instrumento tenha sido cuidadosamente estruturado, diferenças de experiência, contexto organizacional ou *background* técnico podem ter influenciado como cada participante compreendeu essas práticas. Para mitigar essa ameaça, foram fornecidas definições claras, descrições padronizadas e exemplos contextualizados. Outra ameaça diz respeito à utilização de autoavaliações (e.g., *frequência*, *facilidade*, *efetividade* e *essencialidade*).

Esse tipo de métrica pode ser influenciado por vieses cognitivos, como memória seletiva, excesso de confiança ou interpretação subjetiva das escalas Likert (LIKERT, 1932). Para mitigar, optou-se por utilizar escalas simétricas, descrições detalhadas dos níveis e uma análise estatística comparativa entre grupos. Uma outra ameaça à validade de construto está relacionada ao tratamento da opção de resposta “Não aplicável”, disponibilizada em todas as dimensões avaliadas. Essa opção foi incluída com o objetivo de evitar respostas forçadas ou especulativas, especialmente em casos nos quais os participantes não utilizavam ou não tinham conhecimento suficiente sobre determinada prática ou ferramenta. No entanto, a presença dessa opção pode resultar em dados ausentes e em variações no tamanho efetivo da amostra entre práticas e dimensões,

afetando a comparabilidade direta dos resultados. Para mitigar essa ameaça, as respostas “Não aplicável” foram tratadas como valores ausentes (*missing values*) e excluídas dos cálculos estatísticos, conforme descrito na etapa de normalização (Seção 4.2.6.1). Esse procedimento assegura que médias, proporções e comparações reflitam apenas percepções fundamentadas na experiência real dos participantes, preservando a validade interpretativa dos construtos avaliados.

Adicionalmente, uma ameaça está associada à seleção das práticas de segurança incluídas no questionário. Embora a lista tenha sido construída com base em estudos empíricos e literatura consolidada no domínio de AppSec (RINDELL *et al.*, 2021; OMAR *et al.*, 2022; MORRISON *et al.*, 2017; VENSON *et al.*, 2019; RAHMAN; WILLIAMS, 2016; ZHOU *et al.*, 2023; RAJAPAKSE *et al.*, 2022; RYAN *et al.*, 2023; SHRESTHA *et al.*, 2023), é possível que outras práticas igualmente relevantes não tenham sido contempladas. Para mitigar essa possível ameaça adotamos as seguintes estratégias: (i) comparação com estudos empíricos prévios; (ii) curadoria (filtragem, refinamento e agrupamento) das práticas selecionadas por pesquisadores especialistas. Esse processo de curadoria buscou reduzir redundâncias, harmonizar terminologias e assegurar que o conjunto final de práticas representasse adequadamente os principais *security touchpoints* descritos na literatura clássica de *Software Security*; e (iii) inclusão de descrições claras e neutras no questionário. Ainda assim, reconhece-se que, dada a evolução constante do domínio de AppSec, outras práticas relevantes podem emergir e não estarem contempladas no conjunto analisado.

Outra ameaça de construto refere-se à seleção das ferramentas automatizadas avaliadas. Embora a lista utilizada tenha sido construída a partir de um estudo recente (ANGERMEIR *et al.*, 2021), as ferramentas de segurança evoluem rapidamente, e a taxonomia fornecida por (ANGERMEIR *et al.*, 2021) pode não refletir com precisão a prática industrial, especialmente em ambientes DevSecOps altamente integrados. Para reduzir essas ameaças, foram adotadas as seguintes medidas: (i) utilização de categorias

baseadas em literatura consolidada; (ii) revisão das listas por especialistas; (iii) inclusão de exemplos e descrições curtas; e (iv) disponibilização da opção de resposta “Outros”, permitindo que participantes indicassem informações adicionais.

Validade de conclusão. A validade de conclusão trata da precisão das inferências estatísticas realizadas. Uma ameaça é com relação ao tamanho da amostra ($n=83$), que, embora adequado para análises descritivas e comparações, pode limitar a identificação de efeitos muito pequenos. Para reduzir esse impacto, foram aplicados testes estatísticos robustos para amostras independentes *Wilcoxon Rank-Sum* (WHITLEY; BALL, 2002) e medidas de tamanho de efeito *Cliff's Delta* (GRISSOM; KIM, 2005), seguindo boas práticas recomendadas na literatura. Outra ameaça diz respeito ao uso de múltiplas comparações, que pode aumentar o risco de erros de interpretação. Como medida de mitigação, a interpretação dos resultados considerou cuidadosamente o contexto, a consistência entre dimensões e a magnitude dos efeitos, evitando conclusões baseadas apenas em significância estatística isolada. Tais interpretações e análises foram conduzidas por dois pesquisadores envolvidos neste estudo.

Adicionalmente, quanto ao processo de análise de dados descrito na Seção 4.2.6.1, no qual se examinam as percepções dos participantes sobre as práticas de AppSec, considerando suas características individuais. Essa análise foi baseada em estudos anteriores (DIAS-NETO *et al.*, 2017; MATOS *et al.*, 2025) e emprega uma abordagem de média ponderada, na qual a ponderação é informada por atributos qualitativos de cada participante. Embora esse método forneça uma visão detalhada, ele pode resultar na perda de dados extremos. Quanto à análise de dados qualitativos, foram empregados parcialmente procedimentos de GT (CORBIN; STRAUSS, 2008) para definir nosso protocolo de análise de dados. O principal objetivo foi reduzir a subjetividade inerente à codificação das respostas às questões abertas. Todos os dados foram analisados em pares, com o intuito de minimizar vieses individuais e alcançar consenso quanto às categorias e subcategorias identificadas.

Além disso, a exclusão das respostas classificadas como “Não aplicável” pode ter implicado variações no tamanho amostral efetivo entre práticas, dimensões e grupos analisados. Embora esse procedimento possa reduzir o poder estatístico em análises específicas, ele foi adotado de forma consistente ao longo do estudo e priorizou a validade das inferências, evitando a inclusão de avaliações não fundamentadas. Dessa forma, as conclusões estatísticas apresentadas refletem comparações baseadas em dados válidos e semanticamente consistentes, reduzindo o risco de interpretações associadas a dados ausentes.

Validade interna. A validade interna diz respeito ao grau em que os resultados observados podem ser atribuídos às variáveis analisadas, e não a fatores externos. Uma primeira ameaça está relacionada à experiência prévia dos participantes com segurança de software, seja por formação, responsabilidade profissional ou interesse pessoal, o que pode ter influenciado a forma como avaliaram as práticas e ferramentas apresentadas. Para mitigar esse risco, o questionário incluiu um conjunto de perguntas destinadas a caracterizar o perfil dos respondentes, tais como: nível de experiência em segurança, tipo de função desempenhada, porte da organização e frequência com que lidam com atividades de AppSec. Essas variáveis possibilitaram análises comparativas entre grupos (experiência direta vs. indireta, especialistas vs. não especialistas, organizações grandes vs. pequenas), permitindo identificar padrões e isolar parcialmente o efeito de características individuais na percepção de cada prática e ferramenta.

Outra ameaça de validade decorre da possível influência da ordem das perguntas e da fadiga cognitiva ao longo do questionário. Como o instrumento incluía múltiplas práticas avaliadas em diferentes dimensões (frequência, facilidade, efetividade, essencialidade), e categorias de ferramentas automatizadas, havia o risco de respostas automáticas, inconsistentes ou influenciadas pelas perguntas imediatamente anteriores. Para reduzir esse impacto, o questionário foi estruturado em blocos temáticos independentes, evitando alternância frequente entre tópicos e permitindo que o participante mantivesse

um foco dentro de cada seção. Além disso, para as ferramentas automatizadas, foi inserida uma pergunta de controle específica – quanto ao uso ou não de ferramentas automatizadas – com o objetivo de validar a consistência das respostas e aumentar a confiabilidade das inferências.

Validade externa. A validade externa diz respeito à capacidade de generalizar os resultados para outros contextos. A principal ameaça está relacionada à estratégia de amostragem utilizada, baseada predominantemente em amostragem intencional via LinkedIn, complementada por divulgação aberta em redes sociais e contatos profissionais (TONGCO, 2007; BALTES; RALPH, 2022; LINÅKER *et al.*, 2015). Embora esse método seja adequado para alcançar profissionais especializados em uma determinada área de atuação, ele pode introduzir vieses de seleção. Por exemplo, a sobrerrepresentação de indivíduos com maior presença digital, atuação em ambientes corporativos mais estruturados ou maior interesse prévio pelo tema.

Outra limitação decorre da natureza voluntária da participação, que tende a atrair profissionais mais engajados ou com opiniões mais formadas sobre práticas e ferramentas de AppSec. Isso pode limitar a representatividade de desenvolvedores que atuam em organizações com baixa maturidade em segurança ou que têm pouca familiaridade com ferramentas automatizadas. Adicionalmente, a distribuição geográfica e setorial dos participantes, embora variada, pode não refletir completamente a heterogeneidade do ecossistema brasileiro de desenvolvimento de software. Apesar dessas limitações, a amostra obtida apresenta diversidade quanto a papéis profissionais, níveis de experiência, modelos de trabalho, tamanhos organizacionais, setores de atuação e exposição a diferentes culturas de segurança. Tais características contribuem para ampliar a aplicabilidade dos resultados em múltiplos cenários de engenharia de software e segurança de software no país, reforçando a validade externa dos resultados.

6.1 Resumo do Capítulo 6

Este capítulo apresentou as principais ameaças à validade relacionadas ao estudo, organizadas segundo o *framework* proposto por Wohlin *et al.* (2012). Foram discutidas potenciais limitações associadas à interpretação das práticas e ferramentas de segurança, à seleção dos itens avaliados, ao uso de autoavaliações e à estrutura do questionário. Também foram abordadas questões metodológicas, como o tamanho da amostra, possíveis vieses estatísticos, subjetividades inerentes à análise qualitativa e limitações decorrentes do estudo. Apesar dessas ameaças, diversas estratégias de mitigação foram adotadas, incluindo definições claras, revisão por especialistas, uso de métodos estatísticos, análises estratificadas e codificação em pares. O próximo capítulo apresenta as conclusões do estudo e direções para trabalhos futuros.

7 CONCLUSÕES E TRABALHOS FUTUROS

Este estudo apresentou uma investigação sobre como desenvolvedores de software brasileiros percebem e aplicam práticas de segurança de software, bem como os desafios e motivações relacionados ao uso de ferramentas automatizadas de segurança. Para isso, foi conduzido um questionário com 83 desenvolvedores, com o objetivo de explorar: (i) as percepções sobre *frequência de uso, facilidade, efetividade e essencialidade* de 16 práticas de segurança de software; (ii) as diferenças dessas percepções entre grupos de desenvolvedores com distintos níveis de experiência, funções e portes organizacionais; e (iii) a adoção atual e o interesse futuro em ferramentas automatizadas de segurança, bem como os desafios enfrentados por equipes de desenvolvimento. Os resultados obtidos fornecem evidências sobre o contexto brasileiro de AppSec, destacando como os desenvolvedores interpretam o papel das AppSec, quais barreiras enfrentam na adoção de ferramentas e como fatores organizacionais moldam essas percepções.

Nosso estudo revelou que a maioria dos desenvolvedores reconhece a importância das práticas de segurança analisadas, atribuindo altos níveis de essencialidade e efetividade a atividades como Revisão de Código, Codificação Segura, Gerenciamento de Vulnerabilidades e Monitoramento de Eventos de Segurança. No entanto, observou-se uma lacuna entre importância percebida e frequência de uso, especialmente em práticas mais avançadas, como Modelagem de Ameaças e Testes Fuzz, sugerindo limitações práticas relacionadas a custo, conhecimento técnico ou maturidade organizacional.

As análises comparativas entre grupos demonstraram diferenças significativas. Desenvolvedores com experiência direta em segurança tendem a utilizar práticas de forma mais frequente e possuem percepções mais positivas sobre facilidade e efetividade. Por outro lado, profissionais sem especialização formal frequentemente percebem algumas práticas como complexas ou de difícil aplicação. Diferenças também foram observadas entre organizações de grande e pequeno porte, indicando que maturidade,

processos internos e regulamentações podem influenciar práticas de AppSec.

Além disso, a investigação sobre ferramentas automatizadas revelou que, embora algumas sejam amplamente conhecidas, como SonarQube, GitLeaks, Retire.js e Checkov, sua adoção ainda é limitada por desafios técnicos e organizacionais. Entre os principais obstáculos relatados estão: grande volume de falsos positivos, dificuldades de integração com outros sistemas, necessidade de alto esforço de manutenção e problemas de compatibilidade. Ainda assim, a maioria dos participantes demonstrou interesse em adotar mais ferramentas automatizadas no futuro, reforçando a percepção de que automação pode apoiar equipes e reduzir riscos de segurança.

Os resultados obtidos revelam a necessidade de: (1) capacitação contínua em práticas essenciais de AppSec, especialmente para profissionais sem experiência direta em segurança; (2) processos organizacionais mais estruturados, que permitam incorporar práticas avançadas ao ciclo de desenvolvimento; (3) infraestrutura de automação mais madura, reduzindo barreiras à adoção de ferramentas; e (4) curadorias internas e pipelines integrados, para mitigar problemas como falsos positivos e inconsistências entre ambientes. Tais resultados podem orientar pesquisadores, gestores e desenvolvedores interessados em aprimorar estratégias de segurança em equipes de software no Brasil.

Em conclusão, este estudo destaca que, embora os desenvolvedores reconheçam a importância das práticas de segurança de software, a adoção sistemática ainda enfrenta barreiras técnicas, culturais e organizacionais. A percepção positiva sobre ferramentas automatizadas indica um cenário favorável para a expansão de práticas DevSecOps, desde que desafios comuns sejam abordados. Esperamos que os resultados apresentados contribuam para o entendimento do estado atual de AppSec no Brasil. Como trabalhos futuros, propõe-se a execução de entrevistas, grupos focais e estudos de caso, para investigar como fatores organizacionais, culturais e técnicos afetam a adoção prática das ferramentas e a aplicação consistente das práticas de segurança.

REFERÊNCIAS

- AGUADO, D.; ANDRÉS, J. C.; GARCÍA-IZQUIERDO, A. L.; RODRÍGUEZ, J. LinkedIn “big four”: Job performance validation in the ict sector. **Revista de Psicología del Trabajo y de las Organizaciones**, Colegio Oficial de la Psicología de Madrid, v. 35, n. 2, p. 53–64, 2019. Disponível em: <https://doi.org/10.5093/jwop2019a7>. Acesso em: 15 fev. 2025.
- ANGERMEIR, F.; VOGGENREITER, M.; MOYÓN, F.; MENDEZ, D. Enterprise-driven open source software: A case study on security automation. In: IEEE. **Proceedings of the 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)**. [S.l.], 2021. p. 278–287. Disponível em: <https://doi.org/10.1109/ICSE-SEIP52600.2021.00037>. Acesso em: 15 mar. 2025.
- ASSAL, H.; CHIASSON, S. ‘think secure from the beginning’ a survey with software developers. In: **Proceedings of the 2019 CHI conference on human factors in computing systems**. [S.l.: s.n.], 2019. p. 1–13. Disponível em: <https://dl.acm.org/doi/10.1145/3290605.3300519>. Acesso em: 27 fev. 2025.
- BALTES, S.; RALPH, P. Sampling in software engineering research: A critical review and guidelines. **Empirical Software Engineering**, Springer, v. 27, n. 4, p. 94, 2022. Disponível em: <https://doi.org/10.1007/s10664-021-10072-8>. Acesso em: 5 mar. 2025.
- BASS, L.; WEBER, I.; ZHU, L. **DevOps: A software architect’s perspective**. [S.l.]: Addison-Wesley Professional, 2015. Disponível em: <https://books.google.com.br/books?hl=pt-BR&id=fcwkcQAAQBAJoi=fndpg=PT12dq=DevOps:+A+software+architect>
- BRASIL, G. d. Lei nº 13.709, de 14 de agosto de 2018. lei geral de proteção de dados pessoais (lgpd). **Diário oficial da União**, ICP-Brasil São Paulo, SP, Brazil, v. 155, n. 157 seção 1, p. 59–64, 2018. Disponível em: <https://www2.camara.leg.br/legin/fed/lei/2018/lei-13709-14-agosto-2018-787077-publicacaooriginal-156212-pl.html>. Acesso em: 9 mar. 2025.
- CALDIERA, V. R. B.-G.; ROMBACH, H. D. Goal question metric paradigm. **Encyclopedia of software engineering**, v. 1, n. 528-532, p. 6, 1994. Disponível em: <https://www.cs.umd.edu/mvz/handouts/gqm.pdf>. Acesso em: 27 fev. 2025.
- COMMISSION, I. O. for S. E. *et al.* **ISO/IEC 27034-1: 2011–Information technology–Security techniques–Application security–Part 1: Overview and concepts**. 2011. Disponível em: <https://www.iso.org/standard/44378.html>. Acesso em: 15 mar. 2025.

CORBIN, J.; STRAUSS, A. **Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory**. 3. ed. Thousand Oaks, CA: SAGE Publications, Inc., 2008. Disponível em: <https://methods.sagepub.com/book/mono/basics-of-qualitative-research/toc>. Acesso em: 15 fev. 2025.

DACREMA, M. F.; CREMONESI, P.; JANNACH, D. Are we really making much progress? a worrying analysis of recent neural recommendation approaches. In: **Proceedings of the 13th ACM conference on recommender systems**. [S.l.: s.n.], 2019. p. 101–109. Disponível em: <https://dl.acm.org/doi/10.1145/3298689.3347058>. Acesso em: 25 fev. 2025.

DAKA, E.; FRASER, G. A survey on unit testing practices and problems. In: IEEE. **Proceedings of the 25th International Symposium on Software Reliability Engineering**. [S.l.], 2014. p. 201–211. Disponível em: <https://doi.org/10.1109/ISSRE.2014.11>. Acesso em: 5 mar. 2025.

DIAS-NETO, A. C.; MATALONGA, S.; SOLARI, M.; ROBILOLO, G.; TRAVASSOS, G. H. Toward the characterization of software testing practices in south america: looking at brazil and uruguay. **Software Quality Journal**, Springer, v. 25, p. 1145–1183, 2017. Disponível em: <https://doi.org/10.1007/s11219-016-9329-3>. Acesso em: 21 jan. 2025.

EASTERBROOK, S.; SINGER, J.; STOREY, M.-A.; DAMIAN, D. Selecting empirical methods for software engineering research. In: **Guide to Advanced Empirical Software Engineering**. [S.l.]: Springer, 2008. p. 285–311. Disponível em: https://doi.org/10.1007/978-1-84800-044-5_11. Acesso em: 27 fev. 2025.

FORCE, J. T. **Security and privacy controls for information systems and organizations**. [S.l.], 2020. Disponível em: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5-draft.pdf>. Acesso em: 9 mar. 2025.

GRISSOM, R. J.; KIM, J. J. **Effect sizes for research: A broad practical approach**. [S.l.]: Lawrence Erlbaum Associates Publishers, 2005. Disponível em: <https://psycnet.apa.org/record/2005-04135-000>. Acesso em: 2 mar. 2025.

HOWARD, M.; LIPNER, S. **The security development lifecycle**. [S.l.]: Microsoft Press Redmond, 2006. v. 8. Disponível em: <https://doi.org/10.1007/s11623-010-0021-7>. Acesso em: 9 mar. 2025.

JOHNSON, B.; SONG, Y.; MURPHY-HILL, E.; BOWDIDGE, R. Why don't software developers use static analysis tools to find bugs? In: IEEE. **Proceedings of the 35th International Conference on Software Engineering (ICSE)**. [S.l.], 2013. p. 672–681. Disponível em: <https://doi.org/10.1109/ICSE.2013.6606613>. Acesso em: 27 fev. 2025.

KHAN, R. A.; KHAN, S. U.; KHAN, H. U.; ILYAS, M. Systematic literature review on security risks and its practices in secure software development. **IEEE Access**, IEEE, v. 10, p. 5456–5481, 2022. Disponível em: <https://doi.org/10.1109/ACCESS.2022.3140181>. Acesso em: 15 mar. 2025.

KROMBHOLZ, K.; MAYER, W.; SCHMIEDECKER, M.; WEIPPL, E. "i have no idea what i'm doing-on the usability of deploying {HTTPS}". In: **Proceedings of the 26th USENIX Security Symposium (USENIX Security 17)**. [S.l.: s.n.], 2017. p. 1339–1356. Disponível em: <https://www.usenix.org/system/files/conference/usenixsecurity17/sec17-krombholz.pdf>. Acesso em: 2 mar. 2025.

LIKERT, R. **A Technique for the Measurement of Attitudes**. [S.l.]: Archives of Psychology, 1932. (A Technique for the Measurement of Attitudes, N° 136-165). Disponível em: <https://archive.org/details/likert-1932>. Acesso em: 2 mar. 2025.

LINÅKER, J.; SULAMAN, S. M.; MELLO, R. M. de; HÖST, M. Guidelines for conducting surveys in software engineering. **Technical report**, Department of Computer Science, Lund University, 2015. Disponível em: <https://portal.research.lu.se/en/publications/8ac54dbe-b7ac-4244-9c43-0f0d157efa26>. Acesso em: 15 mar. 2025.

LITWIN, M. S.; FINK, A. **How to measure survey reliability and validity**. [S.l.]: Sage, 1995. v. 7. Disponível em: <https://doi.org/10.4135/9781483348957>. Acesso em: 2 mar. 2025.

MALATJI, M. Management of enterprise cyber security: A review of iso/iec 27001: 2022. In: IEEE. **Proceedings of the 2023 International conference on cyber management and engineering (CyMaEn)**. [S.l.], 2023. p. 117–122. Disponível em: <https://doi.org/10.1109/CyMaEn57228.2023.10051114>. Acesso em: 29 jan. 2025.

MATOS, A.; PATRÍCIO, M.; NICOLAU, M. I.; CANEDO, E. D.; PEREIRA, J. A.; UCHÔA, A. Data privacy in software practice: Brazilian developers' perspectives. **Journal of Internet Services and Applications**, v. 16, n. 1, p. 299–319, 2025. Disponível em: <https://doi.org/10.5753/jisa.2025.5302>. Acesso em: 15 fev. 2025.

MCGRAW, G. Software security. **Building security in**, 2006. Disponível em: <https://doi.org/10.1109/ISSRE.2006.43>. Acesso em: 24 fev. 2025.

MELLO, R. M. de; STOLEE, K. T.; TRAVASSOS, G. H. Investigating samples representativeness for an online experiment in java code search. In: IEEE. **Proceedings of the 2015 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)**. [S.l.], 2015. p. 1–10. Disponível em: <https://doi.org/10.1109/ESEM.2015.7321205>. Acesso em: 29 jan. 2025.

MORRISON, P.; SMITH, B. H.; WILLIAMS, L. Measuring security practice use: A case study at ibm. In: IEEE. **Proceedings of the IEEE/ACM 5th International Workshop on Conducting Empirical Studies in Industry (CESI)**. [S.l.], 2017. p. 16–22. Disponível em: <https://doi.org/10.1109/CESI.2017.4>. Acesso em: 21 jan. 2025.

MYRBAKKEN, H.; COLOMO-PALACIOS, R. Devsecops: a multivocal literature review. In: SPRINGER. **Proceedings of the International Conference on Software Process Improvement and Capability Determination**. [S.l.], 2017. p. 17–29. Disponível em: https://doi.org/10.1007/978-3-319-67383-7_2. Acesso em: 15 mar. 2025.

NINA, H.; POW-SANG, J. A.; VILLAVICENCIO, M. Systematic mapping of the literature on secure software development. **IEEE Access**, IEEE, v. 9, p. 36852–36867, 2021. Disponível em: <https://doi.org/10.1109/ACCESS.2021.3062388>. Acesso em: 15 mar. 2025.

OMAR, A.; ALSADEH, A.; NAWAHDAH, M. Adherence to secure software development lifecycle. In: INSTICC. **Proceedings of the 17th International Conference on Software Technologies**. [S.l.]: SciTePress, 2022. p. 410–417. ISBN 978-989-758-588-3. ISSN 2184-2833. Disponível em: <https://doi.org/10.5220/0011263700003266>. Acesso em: 26 fev. 2025.

OZKAYA, M.; ERATA, F. A survey on the practical use of uml for different software architecture viewpoints. **Information and Software Technology**, Elsevier, v. 121, p. 106275, 2020. Disponível em: <https://doi.org/10.1016/j.infsof.2020.106275>. Acesso em: 29 jan. 2025.

RAHMAN, A. A. U.; WILLIAMS, L. Software security in devops: synthesizing practitioners' perceptions and practices. In: **Proceedings of the International Workshop on Continuous Software Evolution and Delivery**. [S.l.: s.n.], 2016. p. 70–76. Disponível em: <https://dl.acm.org/doi/10.1145/2896941.2896946>. Acesso em: 18 fev. 2025.

RAJAPAKSE, R. N.; ZAHEDI, M.; BABAR, M. A.; SHEN, H. Challenges and solutions when adopting devsecops: A systematic review. **Information and Software Technology**, Elsevier, v. 141, p. 106700, 2022. Disponível em: <https://doi.org/10.1016/j.infsof.2021.106700>. Acesso em: 18 fev. 2025.

RINDELL, K.; RUOHONEN, J.; HOLVITIE, J.; HYRYNSALMI, S.; LEPPÄNEN, V. Security in agile software development: A practitioner survey. **Information and Software Technology**, Elsevier, v. 131, p. 106488, 2021. Disponível em: <https://doi.org/10.1016/j.infsof.2020.106488>. Acesso em: 23 fev. 2025.

ROMANO, J.; KROMREY, J. D.; CORAGGIO, J.; SKOWRONEK, J.; DEVINE, L. Exploring methods for evaluating group differences on the nsse

and other surveys: Are the t-test and cohen'sd indices the most appropriate choices. In: CITESEER. **Proceedings of the annual meeting of the Southern Association for Institutional Research**. [S.l.], 2006. p. 1–51. Disponível em: <https://www.bibsonomy.org/bibtex/29fb28c468a7f6dc911f2389e76d9f7b6/becke>. Acesso em: 2 mar. 2025.

RYAN, I.; ROEDIG, U.; STOL, K.-J. Measuring secure coding practice and culture: A finger pointing at the moon is not the moon. In: IEEE. **Proceedings of the 45th International Conference on Software Engineering (ICSE)**. [S.l.], 2023. p. 1622–1634. Disponível em: <https://doi.org/10.1109/ICSE48619.2023.00140>. Acesso em: 26 fev. 2025.

SALKIND, N. **Exploring Research**. 8th. ed. [S.l.]: Pearson Education, 2012. Disponível em: <https://books.google.com.br/books?id=G0DdtgAACAAJ>. Acesso em: 15 mar. 2025. ISBN 9780205093816.

SCHNEIER, B. **Click here to kill everybody: Security and survival in a hyper-connected world**. [S.l.]: WW Norton & Company, 2018. Disponível em: <https://dl.acm.org/doi/10.5555/3288772>. Acesso em: 5 mar. 2025.

SHAHIN, M.; BABAR, M. A.; ZHU, L. Continuous integration, delivery and deployment: a systematic review on approaches, tools, challenges and practices. **IEEE access**, IEEE, v. 5, p. 3909–3943, 2017. Disponível em: <https://doi.org/10.1109/ACCESS.2017.2685629>. Acesso em: 26 fev. 2025.

SHAPIRO, S.; WILK, M. An analysis of variance test for normality (complete samples). **Biometrika**, v. 52, n. 3/4, p. 591–611, 1965. Disponível em: <https://www.jstor.org/stable/2333709>. Acesso em: 2 mar. 2025.

SHRESTHA, A.; SHARMA, T.; SAHA, P.; AHMED, S. I.; AL-AMEEN, M. N. A first look into software security practices in bangladesh. **ACM Journal on Computing and Sustainable Societies**, ACM New York, NY, v. 1, n. 1, p. 1–24, 2023. Disponível em: <https://dl.acm.org/doi/10.1145/3616383>. Acesso em: 15 fev. 2025.

SINAN, M.; SHAHIN, M.; GONDAL, I. Integrating security controls in devsecops: Challenges, solutions, and future research directions. **Journal of Software: Evolution and Process**, Wiley Online Library, v. 37, n. 6, p. e70029, 2025. Disponível em: <https://doi.org/10.1002/smr.70029>. Acesso em: 23 fev. 2025.

TAHAEI, M.; VANIEA, K. A survey on developer-centred security. In: IEEE. **Proceedings of the 2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)**. [S.l.], 2019. p. 129–138. Disponível em: <https://doi.org/10.1109/EuroSPW.2019.00021>. Acesso em: 9 mar. 2025.

TOH, M. Z.; SAHIBUDDIN, S.; MAHRIN, M. N. Adoption issues in devops from the perspective of continuous delivery pipeline. In: **Proceedings of the 8th international conference on software and computer applications**. [S.l.: s.n.], 2019. p. 173–177. Disponível em: <https://dl.acm.org/doi/10.1145/3316615.3316619>. Acesso em: 24 fev. 2025.

TONGCO, M. D. C. Purposive sampling as a tool for informant selection. University of Hawaii at Manoa, 2007. Disponível em: <https://scholarspace.manoa.hawaii.edu/server/api/core/bitstreams/5bdc92a7-99fe-4134-aaeb-6d0970562559/content>. Acesso em: 5 mar. 2025.

VENSON, E.; ALFAYEZ, R.; GOMES, M. M.; FIGUEIREDO, R. M.; BOEHM, B. The impact of software security practices on development effort: An initial survey. In: IEEE. **Proceedings of the 2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)**. [S.l.], 2019. p. 1–12. Disponível em: <https://doi.org/10.1109/ESEM.2019.8870153>. Acesso em: 21 jan. 2025.

WHITLEY, E.; BALL, J. Statistics review 6: Nonparametric methods. **Critical care**, Springer, v. 6, p. 1–5, 2002. Disponível em: <https://doi.org/10.1186/cc1820>. Acesso em: 2 mar. 2025.

WOHLIN, C.; RUNESON, P.; HÖST, M.; OHLSSON, M. C.; REGNELL, B.; WESSLÉN, A. **Experimentation in Software Engineering**. [S.l.]: Springer, 2012. Disponível em: <https://doi.org/10.1007/978-3-642-29044-2>. Acesso em: 27 fev. 2025. ISBN 978-3-642-29043-5.

ZHOU, X.; MAO, R.; ZHANG, H.; DAI, Q.; HUANG, H.; SHEN, H.; LI, J.; RONG, G. Revisit security in the era of devops: An evidence-based inquiry into devsecops industry. **IET software**, Wiley Online Library, v. 17, n. 4, p. 435–454, 2023. Disponível em: <https://doi.org/10.1049/sfw2.12132> Acesso em: 18 fev. 2025.

APÊNDICE A – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO (TCLE)

Este documento, denominado Termo de Consentimento Livre e Esclarecido (TCLE), visa assegurar seus direitos como participante e será disponibilizado por meio eletrônico. Por favor, leia com atenção e calma, buscando entender completamente a proposta da pesquisa. Não haverá qualquer tipo de penalização ou prejuízo se você não quiser participar ou retirar sua autorização em qualquer momento.

INFORMAÇÕES SOBRE ESTA PESQUISA:

Objetivos: O objetivo desta pesquisa é explorar a caracterização de práticas de segurança de software comumente utilizadas por desenvolvedores. Nós também estamos interessados em compreender como os desenvolvedores lidam com o uso de ferramentas automatizadas de segurança e quais desafios os desenvolvedores enfrentam em sua utilização.

Importância do estudo: Este estudo visa obter *insights* valiosos sobre diferentes práticas de segurança de software, assim como fatores relacionados à adoção de ferramentas de segurança automatizadas.

Procedimentos e metodologias: Para participar desta pesquisa, você precisa **ter mais de 18 anos e atuar na área de desenvolvimento de software ou segurança da informação**. Sua participação consiste em responder a um questionário online sobre práticas de segurança de software e ferramentas automatizadas de segurança, que levará aproximadamente 15 minutos para ser concluído. A confidencialidade e o anonimato dos participantes serão rigorosamente mantidos, incentivando respostas francas e honestas para fornecer *insights* valiosos sobre a temática em questão.

Tratamento dos dados: Durante todo o processo, será assegurado o respeito às normas de privacidade e ética, conforme preconizado pela Lei Geral de Proteção de Dados (Lei nº 13.709/2018), garantindo que as informações dos participantes sejam tratadas com confidencialidade. Os dados coletados por meio do questionário online serão anonimizados, removendo qualquer informação que possa identificar diretamente os participantes. Os dados serão organizados e codificados para facilitar a análise. Os resultados serão apresentados de forma agregada e não identificáveis individualmente, preservando assim a privacidade dos participantes de acordo com as disposições legais aplicáveis.

Forma de contato com os pesquisadores: Sempre que desejar, você poderá entrar em contato para obter informações sobre este projeto de pesquisa, sobre sua participação ou outros assuntos relacionados à pesquisa. Os pesquisadores responsáveis são: Carlos Adriel (<adrielbastos@alu.ufc.br>), Anderson Uchôa (<andersonuchoa@ufc.br>), Carla Bezerra (<carlailane@ufc.br>) e Joanna Cecilia (<jdasilv2@nd.edu>).

INFORMAÇÕES SOBRE ESTA PESQUISA:

Direito de recusa a participar: A qualquer momento, você pode se recusar a participar e se retirar da pesquisa, sem constrangimentos, penalidades ou qualquer prejuízo. As informações e materiais obtidos nesta pesquisa não serão utilizados para outras finalidades que não sejam dentro do contexto desta.

Sigilo e privacidade: Garantimos a total confidencialidade e o anonimato de suas respostas. Todas as informações sensíveis, como nomes de empresas ou de outras pessoas, serão completamente anonimizadas.

Responsabilidade do(a) Pesquisador(a): Asseguramos fornecer este documento ao

participante da pesquisa e utilizar os dados obtidos exclusivamente para as finalidades descritas neste documento ou conforme o consentimento dado pelo(a) participante.

APÊNDICE B – *SCRIPT DO QUESTIONÁRIO*

Você está sendo convidado a participar, como voluntário, da pesquisa "*Caracterização de Práticas de Segurança de Software e Ferramentas Automatizadas*" realizada pela UFC e pela *University of Notre Dame*.

Essa pesquisa tem como objetivo explorar a caracterização de práticas de segurança de software comumente utilizadas por desenvolvedores. Nós também estamos interessados em compreender como os desenvolvedores lidam com o uso de ferramentas automatizadas de segurança. Ao explorar esses objetivos, esperamos poder fornecer insights valiosos sobre diferentes práticas de segurança de software, assim como, fatores relacionados a adoção ferramentas de segurança automatizadas.

Para participar desta pesquisa, é necessário ser maior de 18 anos e ter experiência na área de Desenvolvimento de Software ou Segurança da Informação. Sua participação consiste em responder a um questionário online sobre práticas de segurança de software e ferramentas automatizadas de segurança, que levará aproximadamente 15 minutos para ser concluído. Garantimos o sigilo e a confidencialidade de todas as informações fornecidas durante a pesquisa.

O Termo de Consentimento Livre e Esclarecido (TCLE) visa assegurar seus direitos como participante e está disponível neste link. Por favor, leia-o com atenção para entender completamente a proposta da pesquisa. Sua participação é voluntária, e você pode optar por não participar ou retirar seu consentimento a qualquer momento, sem qualquer penalização ou prejuízo.

Após ter recebido esclarecimentos sobre a natureza da pesquisa, seus objetivos e métodos, eu declaro que sou maior de 18 anos e aceito participar por meio do formulário de participação, conforme as disposições da Lei Geral de Proteção de Dados (Lei nº 13.709/2018). Em caso de resposta negativa, o formulário será encerrado, garantindo assim o respeito à autonomia e privacidade dos participantes, conforme

estabelecido pela legislação vigente. (Obrigatória)

- ☐ Aceito participar
- ☐ Não aceito participar

CARACTERIZAÇÃO GERAL DO PARTICIPANTE

Nesta seção recolheremos informações básicas e dados pessoais para entender um pouco mais sobre o seu perfil.

Questão 1. Qual é a sua idade? (Obrigatória)

- ☐ 18-24 anos
- ☐ 25-34 anos
- ☐ 35-44 anos
- ☐ 45-54 anos
- ☐ 55-64 anos
- ☐ 65 anos ou mais
- ☐ Prefiro não dizer

Questão 2. Em qual estado você reside atualmente? (Obrigatória)

- ☐ Acre (AC)
- ☐ Alagoas (AL)
- ☐ Amapá (AP)
- ☐ Amazonas (AM)
- ☐ Bahia (BA)
- ☐ Ceará (CE)
- ☐ Distrito Federal (DF)
- ☐ Espírito Santo (ES)
- ☐ Goiás (GO)

- ☐ Maranhão (MA)
- ☐ Mato Grosso (MT)
- ☐ Mato Grosso do Sul (MS)
- ☐ Minas Gerais (MG)
- ☐ Pará (PA)
- ☐ Paraíba (PB)
- ☐ Paraná (PR)
- ☐ Pernambuco (PE)
- ☐ Piauí (PI)
- ☐ Rio de Janeiro (RJ)
- ☐ Rio Grande do Norte (RN)
- ☐ Rio Grande do Sul (RS)
- ☐ Rondônia (RO)
- ☐ Roraima (RR)
- ☐ Santa Catarina (SC)
- ☐ São Paulo (SP)
- ☐ Sergipe (SE)
- ☐ Tocantins (TO)
- ☐ Fora do país

Questão 3. Qual o seu gênero? (Obrigatória)

- ☐ Homem
- ☐ Mulher
- ☐ Não-binário, gênero queer ou não-conformidade de gênero
- ☐ Prefiro não responder
- ☐ Outros: _____

Questão 4. Qual é o seu nível de educação formal concluída? (Obrigatória)

- ☐ Ensino Médio

- ☐ Ensino Superior
- ☐ Especialização
- ☐ Estudante de mestrado
- ☐ Mestrado
- ☐ Estudante de doutorado
- ☐ Doutorado

Questão 5. Qual seu modelo de trabalho? (Obrigatória)

- ☐ Remoto
- ☐ Híbrido
- ☐ Presencial

Questão 6. Qual papel que melhor descreve suas atividades atuais em projetos de desenvolvimento de software? (Obrigatória)

- ☐ Cryptographer
- ☐ Data Scientist
- ☐ Developer (Backend, frontend, fullstack)
- ☐ Penetration tester
- ☐ DevOps engineer
- ☐ Privacy engineer
- ☐ Product owner
- ☐ Project Lead / Project Manager
- ☐ Requirements Engineer
- ☐ Security engineer
- ☐ Solution Architect
- ☐ Test Manager / Tester
- ☐ UX/UI designer
- ☐ Outros: _____

Questão 7. Por favor, indique a senioridade da posição que ocupa. (Obrigatória)

- ☐ Estagiário
- ☐ Júnior (até 5 anos)
- ☐ Pleno (6 a 9 anos)
- ☐ Sênior (10+ anos)

Questão 8. Quantos anos de experiência você tem em funções relacionadas ao desenvolvimento de software ou TI? (Obrigatória)

- ☐ Menos de 1 ano
- ☐ Entre 1 e 3 anos
- ☐ Entre 4 e 5 anos
- ☐ Entre 7 e 14 anos
- ☐ Mais de 15 anos

Questão 9. Quantos anos de experiência você tem em funções relacionadas a Segurança de Software? (Obrigatória)

- ☐ Nenhuma
- ☐ Menos de 1 ano
- ☐ Entre 1 e 3 anos
- ☐ Entre 4 e 5 anos
- ☐ Entre 7 e 14 anos
- ☐ Mais de 15 anos

Questão 10. Quais são os tipos de sistemas que você trabalha? (múltiplas escolhas possíveis) (Obrigatória)

- ☐ Aplicações embarcadas (Embedded applications)
- ☐ Sistemas Operacionais (Operating Systems)
- ☐ Aplicações desktop (Desktop applications)
- ☐ Aplicações web (Web applications)
- ☐ Aplicações móveis (Mobile applications)
- ☐ Ferramentas para desenvolvedores (Developer tools)

☐ Middleware (Middleware)

☐ Outros: _____

Questão 11. Quais são os domínios dos sistemas que você trabalha atualmente? (múltiplas escolhas possíveis) (Obrigatória)

☐ Bancos/Financeiro (Banking/Financial)

☐ Defesa & Segurança (Defense & Security)

☐ Educação (Education)

☐ Sistemas embarcados em Automotivo ou Aviônica (Embedded systems in Automotive or Avionics)

☐ Entreterimento (Entertainment)

☐ Saúde (Healthcare)

☐ Seguros (Insurance)

☐ Logística (Logistics)

☐ Petróleo & Gás (Oil & Gas)

☐ Vendas/E-commerce (Sales/E-commerce)

☐ Telecommunication

☐ Outros: _____

Questão 12. Qual o setor da organização para a qual você trabalha atualmente? (Obrigatória)

☐ Autônomo

☐ Setor privado

☐ Setor público

☐ Startup

☐ Organização sem fins lucrativos

☐ Organização não governamental

☐ Projetos de pesquisa e desenvolvimento

☐ Outros: _____

Questão 13. Qual o tamanho da empresa para a qual você trabalha atualmente? (Obrigatória)

- ☐ Até 9 funcionários
- ☐ De 10 a 49 funcionários
- ☐ De 50 a 99 funcionários
- ☐ De 100 a 499 funcionários
- ☐ De 500 a 999 funcionários
- ☐ Mais de 1000 funcionários
- ☐ Não sei

CARACTERIZAÇÃO DA EXPERIÊNCIA COM SEGURANÇA DE SOFTWARE

Nesta seção, gostaríamos de saber sua experiência de trabalho com segurança de software.

Questão 14. Você tem (ou teve) alguma experiência de trabalho relacionada à segurança de software no processo de desenvolvimento de software? (Obrigatória)

- ☐ Eu já trabalhei (ou trabalho) com segurança no processo de desenvolvimento de software.
- ☐ Meu grupo/equipe de desenvolvimento trabalha (ou trabalhou) com segurança, mas não estou diretamente envolvido em nenhuma tarefa.
- ☐ Nunca tive experiência direta e nem indireta com segurança no processo de desenvolvimento de software.

Questão 15. Você possui alguma certificação na área de segurança da informação? Se sim, quais são elas? Caso não possua, selecione a opção "Não possuo nenhuma certificação." (múltiplas escolhas possíveis) (Obrigatória)

- ☐ Certified Information Systems Security Professional (CISSP)
- ☐ Certified Information Security Manager (CISM)
- ☐ Certified Information Systems Auditor (CISA)

- ☐ Certified Ethical Hacker (CEH)
- ☐ CompTIA Security+
- ☐ Certified Cloud Security Professional (CCSP)
- ☐ Offensive Security Certified Professional (OSCP)
- ☐ GIAC Security Essentials (GSEC)
- ☐ Certified Incident Handler (GCIH)
- ☐ Certified Information Privacy Professional (CIPP)
- ☐ Não possui nenhuma certificação
- ☐ Outros: _____

Questão 16. Por favor, descreva brevemente a sua experiência com práticas de segurança de software. Em caso negativo, responda que não possui experiência. (Obrigatória)

CARACTERIZAÇÃO DE PRÁTICAS DE SEGURANÇA DE SOFTWARE

Nesta seção, iremos abordar algumas práticas de segurança de software que são utilizadas durante o processo de desenvolvimento de software e recolheremos a sua percepção sobre essas práticas.

Para responder às próximas perguntas, considere a descrição das seguintes **práticas de segurança de software**:

- **Revisão de Código (Code Reviews):** Analisar manual ou automaticamente o código e/ou as alterações de código (commits) para identificar falhas de segurança, violações de padrões de codificação segura e mitigar vulnerabilidades identificadas.
- **Modelagem de Ameaças (Threat Modeling):** Identificar ameaças e, em seguida, definir medidas de mitigação para prevenir ou mitigar os efeitos dessas ameaças ao sistema.

- **Identificação de Casos de Abuso/Mau Uso (Identifying Abuse/Misuse Cases):** Identificar cenários potenciais de abuso ou mau uso que possam levar a vulnerabilidades de segurança.
- **Análise de Riscos Arquiteturais (Architectural Risk Analysis):** Avaliar e revisar a arquitetura de software para identificar possíveis riscos de segurança e mitigá-los antes que se tornem vulnerabilidades.
- **Testes de Penetração (Penetration Testing):** Utilizar ferramentas especializadas para simular ataques à aplicação, ajudando a identificar e corrigir fraquezas de segurança.
- **Gerenciamento de Riscos (Risk Management):** Definir os principais objetivos e avaliar a criticidade da aplicação para priorizar os esforços de segurança. Avaliar o impacto potencial de falhas de segurança nas operações comerciais para orientar as prioridades de segurança.
- **Gerenciamento de Vulnerabilidades (Vulnerability Management):** Identificar, avaliar e mitigar continuamente vulnerabilidades de segurança ao longo do ciclo de vida da aplicação.
- **Testes de Segurança Manuais/Automatizados (Manual/Automated Security Testing):** Criar e executar casos de teste focados especificamente na avaliação dos aspectos de segurança da aplicação.
- **Execução de Testes Fuzz (Performing Fuzz Testing):** Utilizar técnicas de fuzz testing para introduzir entradas inesperadas na aplicação e identificar vulnerabilidades que possam causar problemas de segurança.
- **Engenharia de Requisitos de Segurança (Security Requirements Engineering):** Identificar e analisar as necessidades de segurança com base em ameaças potenciais, requisitos regulatórios e metas de negócios, além de especificar as medidas e protocolos de segurança a serem implementados no sistema.
- **Codificação Segura (Secure Coding):** Seguir padrões de codificação segura para

minimizar vulnerabilidades no código, como a descontinuação de funções inseguras, o uso de algoritmos de hash robustos, etc.

- **Resposta Emergencial ao Código (Emergency Codebase Response):** Responder rapidamente e corrigir vulnerabilidades de segurança no código para evitar exploração.
- **Configurações e Gerenciamento de Segurança (Security Configurations & Management):** Garantir que sistemas, aplicações e redes estejam configurados para atender aos requisitos de segurança e minimizar vulnerabilidades. Verificar regularmente se as configurações de segurança estão corretamente aplicadas e mantidas.
- **Monitoramento de Eventos de Segurança (Monitoring for Security Events):** Utilizar ferramentas e processos automatizados para monitorar continuamente eventos de segurança e detectar ameaças potenciais em tempo real.
- **Auditorias de Segurança Externas (Undergoing External Security Audits):** Engajar auditores independentes de terceiros para avaliar a segurança da aplicação e recomendar melhorias.
- **Segurança da Cadeia de Suprimentos de Software (Software Supply Chain Security):** Identificar vulnerabilidades em software/bibliotecas/APIs de terceiros reutilizados e abordar/corrigir vulnerabilidades neles. Publicar e consumir a Software Bill of Materials (SBOM).

Questão 17. Com que **frequência** você costuma aplicar as seguintes práticas de segurança de software em seus projetos? (Obrigatória)

Práticas	Diariamente	Semanalmente	Mensalmente	Anualmente	Não aplicável
Revisão de Código					
Modelagem de Ameaças					
Identificação de Casos de Abuso/Mau Uso					
Análise de Riscos Arquiteturais					
Testes de Penetração					
Gerenciamento de Riscos					
Gerenciamento de Vulnerabilidades					
Testes de Segurança Manuais/Automatizados					
Execução de Testes Fuzz					
Engenharia de Requisitos de Segurança					
Codificação Segura					
Resposta Emergencial ao Código					
Configurações e Gerenciamento de Segurança					
Monitoramento de Eventos de Segurança					
Auditorias de Segurança Externas					
Segurança da Cadeia de Suprimentos de Software					

Questão 18. Você aplica outras práticas de segurança de software além das mencionadas anteriormente? Se sim, por favor, descreva brevemente sobre essas práticas. (Opcional)

Questão 19. Como você tem implementado as práticas de segurança de software mencionadas anteriormente em seus projetos? (Opcional)

Questão 20. Com base em sua experiência, como você avaliaria o nível de **facilidade de uso** das seguintes práticas de segurança de software? Considere marcar “Não Aplicável” para as práticas das quais você não utiliza. (Obrigatória)

Práticas	Muito Baixo	Baixo	Moderado	Alto	Muito Alto	Não aplicável
Revisão de Código						
Modelagem de Ameaças						
Identificação de Casos de Abuso/Mau Uso						
Análise de Riscos Arquiteturais						
Testes de Penetração						
Gerenciamento de Riscos						
Gerenciamento de Vulnerabilidades						
Testes de Segurança Manuais/Automatizados						
Execução de Testes Fuzz						
Engenharia de Requisitos de Segurança						
Codificação Segura						
Resposta Emergencial ao Código						
Configurações e Gerenciamento de Segurança						
Monitoramento de Eventos de Segurança						
Auditorias de Segurança Externas						
Segurança da Cadeia de Suprimentos de Software						

Questão 21. Com base em sua experiência, como você avalia a **efetividade** das seguintes práticas de segurança de software na prevenção, remoção ou detecção de problemas de segurança em seus projetos? Considere marcar “Não Aplicável” para as práticas das quais você não utiliza. (Obrigatória)

Práticas	Muito Baixo	Baixo	Moderado	Alto	Muito Alto	Não aplicável
Revisão de Código						
Modelagem de Ameaças						
Identificação de Casos de Abuso/Mau Uso						
Análise de Riscos Arquiteturais						
Testes de Penetração						
Gerenciamento de Riscos						
Gerenciamento de Vulnerabilidades						
Testes de Segurança Manuais/Automatizados						
Execução de Testes Fuzz						
Engenharia de Requisitos de Segurança						
Codificação Segura						
Resposta Emergencial ao Código						
Configurações e Gerenciamento de Segurança						
Monitoramento de Eventos de Segurança						
Auditorias de Segurança Externas						
Segurança da Cadeia de Suprimentos de Software						

Questão 22. Qual a sua opinião quanto à frase a seguir aplicada em cada uma das práticas abaixo: **“Essa prática é essencial para a proteção do sistema.”** (Obrigatória)

Práticas	Concordo Totalmente	Concordo	Neutro	Discordo	Discordo Totalmente
Revisão de Código					
Modelagem de Ameaças					
Identificação de Casos de Abuso/Mau Uso					
Análise de Riscos Arquiteturais					
Testes de Penetração					
Gerenciamento de Riscos					
Gerenciamento de Vulnerabilidades					
Testes de Segurança Manuais/Automatizados					
Execução de Testes Fuzz					
Engenharia de Requisitos de Segurança					
Codificação Segura					
Resposta Emergencial ao Código					
Configurações e Gerenciamento de Segurança					
Monitoramento de Eventos de Segurança					
Auditorias de Segurança Externas					
Segurança da Cadeia de Suprimentos de Software					

ADOÇÃO DE FERRAMENTAS DE SEGURANÇA AUTOMATIZADAS

Nesta seção, exploraremos a adoção de ferramentas automatizadas de segurança de software e sua experiência em utilizá-las.

Exemplo de algumas categorias de ferramentas de segurança automatizadas:

- **3rd Party Vulnerability Scanning:** é a verificação da implementação e dos testes corretos em relação a riscos de segurança e vulnerabilidades em componentes externos
- **Static/Dynamic Application Security Testing Tools:** refere-se a testes de vulnerabilidades de segurança do produto componentes e suas dependências para manter um nível de patch de segurança atualizado.
- **Secure Configuration/Hardening Tools:** descreve o processo de aplicação de

medidas de configuração de segurança para minimizar a superfície de ataque ou até mesmo garantir a segurança conformidade padrão.

- **Compliance/Hardening Checks Tools:** verificar se as medidas de configuração/reforço de segurança estão alinhadas com as práticas recomendadas.
- **Secrets Management Tools:** contém a gestão de informações confidenciais, como senhas ou chaves privadas, como bem como medidas de proteção para evitar o vazamento de informações secretas Informação.

Questão 23. Você costuma utilizar ferramentas de segurança automatizadas em seu dia a dia? (Obrigatória)

- ☐ Sim
- ☐ Uso parcialmente
- ☐ Não costumo utilizar

ADOÇÃO DE FERRAMENTAS DE SEGURANÇA AUTOMATIZADAS

Nesta seção, exploraremos a adoção de ferramentas automatizadas de segurança de software e sua experiência em utilizá-las.

Para responder a questão a seguir, considere as categorias de ferramentas abaixo de segurança automatizadas:

- **3rd Party Vulnerability Scanning:** é a verificação da implementação e dos testes corretos em relação a riscos de segurança e vulnerabilidades em componentes externos
- **Static/Dynamic Application Security Testing Tools:** refere-se a testes de vulnerabilidades de segurança do produto componentes e suas dependências para manter um nível de patch de segurança atualizado.
- **Secure Configuration/Hardening Tools:** descreve o processo de aplicação de medidas de configuração de segurança para minimizar a superfície de ataque ou até

mesmo garantir a segurança conformidade padrão.

- **Compliance/Hardening Checks Tools:** verificar se as medidas de configuração/reforço de segurança estão alinhadas com as práticas recomendadas.
- **Secrets Management Tools:** contém a gestão de informações confidenciais, como senhas ou chaves privadas, como bem como medidas de proteção para evitar o vazamento de informações secretas Informação.

Questão 24. Para cada uma das seguintes categorias de ferramentas de segurança automatizadas, indique com que **frequência** você as utiliza. Considere marcar “Não Aplicável” para as categorias de ferramentas das quais você não utiliza. (Obrigatória)

Categorias	Diariamente	Semanalmente	Mensalmente	Anualmente	Não Aplicável
3rd Party Vulnerability Scanning Tools					
Static Application Security Testing Tools					
Dynamic Application Security Testing Tools					
Secure Configuration/Hardening Tools					
Compliance/Hardening Checks Tools					
Secrets Management Tools					

Questão 25. Quais das seguintes ferramentas específicas de segurança de software você utiliza atualmente ou já utilizou anteriormente? Caso nenhuma das opções listadas seja adequada, selecione "Outros" e especifique (múltiplas escolhas possíveis) (Obrigatória)

- ☐ Dependabot
- ☐ PHP Security Checker
- ☐ Snyk
- ☐ Clair
- ☐ Blackduck
- ☐ RetireJS
- ☐ SRC:CLR
- ☐ Anchore

- [] Node Security Platform
- [] Trivy
- [] Sonatype
- [] Whitesource
- [] CodeQL
- [] CheckStyle
- [] Coverity
- [] SonarQube
- [] Bandit
- [] PHPStan
- [] LGTM
- [] Brakeman
- [] Qark
- [] ESLint Security
- [] Dart Analyzer
- [] Checkmarx
- [] DevSkim
- [] CFN NAG
- [] SpotBugs
- [] Chef
- [] Ansible
- [] Terraform
- [] Puppet
- [] Phan
- [] Seccomp
- [] CloudCustodian
- [] Turbot

- ☐ Bane
- ☐ Ansible-Lint
- ☐ Blackbox
- ☐ Puppet-Lint
- ☐ InSpec
- ☐ Kube Audit
- ☐ Foodcritic
- ☐ Cookstyle
- ☐ conftest
- ☐ Pinterest Knox
- ☐ GitLeaks
- ☐ CyberArk Conjur
- ☐ ThoughtWorks Talisman
- ☐ Berglas
- ☐ Docker Secrets
- ☐ Trufflehog
- ☐ AppScan
- ☐ Fortify
- ☐ Outros: _____

Questão 26. Em quais fases do ciclo de desenvolvimento de software você costuma utilizar as ferramentas de segurança automatizadas selecionadas na pergunta anterior?

(Selecione todas as opções que se aplicam) (Obrigatória)

- ☐ Durante a codificação
- ☐ Durante a Integração Contínua (ID)
- ☐ Durante a Entrega Contínua (ED)
- ☐ Durante a revisão de código
- ☐ Durante os testes

☐ Em produção

Questão 27. Quais recomendações você daria para um desenvolvedor sobre a adoção de ferramentas de segurança para seus projetos? (Obrigatória)

Questão 28. Quais são os principais desafios que você enfrenta ao automatizar ferramentas de segurança? (Selecione todas as opções que se aplicam) (Obrigatória)

- ☐ Custo
- ☐ Curva de aprendizado
- ☐ Integração com outras ferramentas
- ☐ Performance do sistema
- ☐ Falsos positivos
- ☐ Manutenção das ferramentas automatizadas
- ☐ Escalabilidade
- ☐ Falta de suporte técnico
- ☐ Compatibilidade com diferentes ambientes de desenvolvimento
- ☐ Necessidade de atualizações frequentes
- ☐ Aderência a normas e compliance
- ☐ Falta de visibilidade ou relatórios inadequados
- ☐ Outros: _____

Questão 29. Quais melhorias você gostaria de ver nas ferramentas de segurança automatizadas que utiliza atualmente? (Selecione todas as opções que se aplicam) (Obrigatória)

- ☐ Aumentar a abrangência das ameaças
- ☐ Menos falsos positivos
- ☐ Melhor integração com o CI/CD
- ☐ Melhor documentação e suporte
- ☐ Facilidade de uso

- ☐ Melhor usabilidade da interface
- ☐ Redução do tempo de análise
- ☐ Melhor compatibilidade com diferentes ambientes de desenvolvimento
- ☐ Relatórios mais detalhados e customizáveis
- ☐ Outros: _____

Questão 30. Caso você queira receber os resultados desse estudo e participar de uma futura entrevista sobre Práticas de Segurança de Software, por favor, informe seu Email. (Opcional)

ADOÇÃO DE FERRAMENTAS DE SEGURANÇA AUTOMATIZADAS

Nesta seção, exploraremos a adoção de ferramentas automatizadas de segurança de software e sua experiência em utilizá-las.

Questão 31. Quais são as razões pelas quais você opta por não utilizar ferramentas automatizadas de segurança de software? Por favor, descreva brevemente sobre. (Obrigatória)

Questão 32. Você considera utilizar ferramentas automatizadas de segurança de software em seus projetos no futuro? (Obrigatória)

- ☐ Sim
- ☐ Não
- ☐ Talvez

Questão 33. Se você respondeu "Sim" ou "Talvez" à pergunta anterior, por favor, indique quais ferramentas utilizaria. Marque "Nenhuma" se respondeu "Não". Caso nenhuma das opções listadas seja adequada, selecione "Outros" e especifique. (múltiplas escolhas possíveis) (Obrigatória)

- [] Nenhuma
- [] Dependabot
- [] PHP Security Checker
- [] Snyk
- [] Clair
- [] Blackduck
- [] Retire.JS
- [] SRC:CLR
- [] Anchore
- [] Node Security Platform
- [] Trivy
- [] Sonatype
- [] Whitesource
- [] CodeQL
- [] CheckStyle
- [] Coverity
- [] SonarQube
- [] Bandit
- [] PHPStan
- [] LGTM
- [] Brakeman
- [] Qark
- [] ESLint Security
- [] Dart Analyzer
- [] Checkmarx
- [] DevSkim
- [] CFN NAG

- ☐ SpotBugs
- ☐ Chef
- ☐ Ansible
- ☐ Terraform
- ☐ Puppet
- ☐ Phan
- ☐ Seccomp
- ☐ CloudCustodian
- ☐ Turbot
- ☐ Bane
- ☐ Ansible-Lint
- ☐ Blackbox
- ☐ Puppet-Lint
- ☐ InSpec
- ☐ Kube Audit
- ☐ Foodcritic
- ☐ Cookstyle
- ☐ conftest
- ☐ Pinterest Knox
- ☐ GitLeaks
- ☐ CyberArk Conjur
- ☐ ThoughtWorks Talisman
- ☐ Berglas
- ☐ Docker Secrets
- ☐ Trufflehog
- ☐ AppScan
- ☐ Fortify

[] Outros: _____

Questão 34. Caso você queira receber os resultados desse estudo e participar de uma futura entrevista sobre Práticas de Segurança de Software, por favor, informe seu Email.
(Opcional)

VALIDAÇÃO DE PARTICIPAÇÃO

Seção focada em confirmar que você é um participante humano.

Questão 35. Marque a imagem em que aparece um filhote de cachorro: (Obrigatória)

() Opção 1



() Opção 2



() Opção 3



() Opção 4

