# Estudo das Praticas de Segurança Aplicadas no Ciclo de Desenvolvimento de Software

**Davi Cândido de Almeida**[1], **Lucas Carneiro Nassau Malta**[1]

[1] Instituto de Ciências Exatas e Informática
Pontifícia Universidade Católica de Minas Gerais (PUC Minas)
Belo Horizonte – MG – Brazil

## 1. Introdução

A era moderna trouxe consigo uma ampla gama de facilitadores para as mais diversas problemáticas antes enfrentadas pela sociedade, sendo que boa parte dessas soluções foram aplicadas pelo meio digital, ou seja, a partir de sistemas de software. No entanto, o meio digital exige que grandes volumes de dados sejam manipulados e armazenados de forma praticamente constante – dados esses muitas vezes críticos e de alto risco caso sejam expostos indevidamente. Portanto, não basta somente satisfazer as necessidades para as quais os sistemas de software são desenvolvidos, mas também oferecer segurança e credibilidade aos seus usuários [Khan et al. 2021].

Nessa perspectiva, métodos de desenvolvimento de software modernos têm destacado a importância da adoção de práticas de segurança ao longo de todo o ciclo de vida do produto. Um exemplo dessa abordagem é o DevSecOps (*Development*, *Security* e *Operations*, ou seja, Desenvolvimento, Segurança e Operações), o qual foi utilizado pela primeira vez em 2012 pela Gartner, três anos após a popularização do termo DevOps (*Development* e *Operations*). Idealizado pelo belga Patrick Debois, o DevOps buscava implantar uma cultura de automação e otimização no desenvolvimento de software, o que se assemelhava às práticas dos métodos ágeis no que tange à colaboração e entrega contínua, porém com um enfoque maior para o *deploy* e a operação do sistema. O DevSecOps, por sua vez, buscava ampliar essa abordagem distribuindo a responsabilidade pela segurança por todo o ciclo de desenvolvimento, isto é, promovendo uma cultura de automação de segurança através de uma responsabilidade distribuída, garantindo a priorização da proteção do sistema desde as fases iniciais. Ao integrar segurança às abordagens de desenvolvimento ágil, a popularização do DevSecOps possibilitou a implementação de controles de segurança de forma rápida, escalável e eficaz. Dessa maneira, vulnerabilidades podem ser identificadas e corrigidas em um ritmo acelerado, reduzindo riscos e fortalecendo a segurança dos sistemas.[Myrbakken and Colomo-Palacios 2017]

A partir dessas definições, é importante destacar as principais vulnerabilidades presentes na maioria dos sistemas, bem como as estratégias adotadas para mitigá-las. A seguir, são apresentadas algumas das principais técnicas de segurança da informação e suas áreas de atuação no desenvolvimento de software: A criptografia é amplamente reconhecida como uma técnica essencial de segurança, utilizada para evitar a intercepção de dados sensíveis transmitidos entre cliente e servidor, proteger informações armazenadas contra exposição em caso de vazamento e dificultar ataques de força bruta a senhas [Buchmann and Buchamann 2004]. Outra estratégia fundamental é a autenticação e autorização, que tem como objetivo proteger o sistema contra furtos de credenciais, sequestro de sessão e escalada de privilégios, garantindo que apenas usuários devidamente

autorizados tenham acesso a determinados recursos [Idrus et al. 2013]. A proteção contra vulnerabilidades também desempenha um papel crucial na segurança do software. Essa abordagem previne ataques como injeção de código, exploração de dependências vulneráveis e comprometimento de contêineres. Em geral, as tecnologias associadas a essa categoria analisam o código desenvolvido e identificam possíveis falhas de segurança antes que possam ser exploradas por invasores [Hanif et al. 2021].

Dessa forma, diante do crescente volume de dados sensíveis manipulados por sistemas de software, torna-se essencial adotar práticas de segurança desde as fases iniciais do desenvolvimento. O DevSecOps surge como uma abordagem que integra segurança ao ciclo de vida do software, promovendo automação e distribuindo a responsabilidade entre todos os envolvidos no processo. Com isso, estratégias como criptografia, autenticação, autorização e proteção contra vulnerabilidades se consolidam como pilares fundamentais para a proteção de sistemas, o que é de extrema importância para garantir a integridade, confidencialidade e disponibilidade das informações para os usuários e clientes da plataforma [Lyra 2008].

## References


Buchmann, J. and Buchamann, J. (2004). *Introduction to cryptography*, volume 335. Springer.

Hanif, H., Nasir, M. H. N. M., Ab Razak, M. F., Firdaus, A., and Anuar, N. B. (2021). The rise of software vulnerability: Taxonomy of software vulnerabilities detection and machine learning approaches. *Journal of Network and Computer Applications*, 179:103009.

Idrus, S. Z. S., Cherrier, E., Rosenberger, C., and Schwartzmann, J.-J. (2013). A review on authentication methods. *Australian Journal of Basic and Applied Sciences*, 7(5):95–107.

Khan, R. A., Khan, S. U., Khan, H. U., and Ilyas, M. (2021). Systematic mapping study on security approaches in secure software engineering. *IEEE Access*, 9:19139–19160.

Lyra, M. R. (2008). *Segurança e Auditoria em Sistemas de Informação*. Editora Ciência Moderna, Rio de Janeiro.

Myrbakken, H. and Colomo-Palacios, R. (2017). Devsecops: a multivocal literature review. In *Software Process Improvement and Capability Determination: 17th International Conference, SPICE 2017, Palma de Mallorca, Spain, October 4–5, 2017, Proceedings*, pages 17–29. Springer.