

# Introdução

A engenharia de software está presente em diversos aspectos da vida, incluindo sistemas críticos que impactam nossa saúde e bem-estar. Diferente de outras áreas da engenharia, o desenvolvimento de software evolui rapidamente, exigindo profissionais capacitados e em constante aprendizado para garantir qualidade, cumprimento de prazos e controle de custos.

Desde os primórdios da computação, desafios como qualidade, prazos e escopo sempre estiveram presentes, tornando essencial uma base sólida em teoria e prática da engenharia de software. Essa disciplina aborda conceitos fundamentais, como a necessidade de um processo estruturado, modelos de desenvolvimento, técnicas de levantamento de requisitos, testes de software, qualidade e gerenciamento de projetos, incluindo o PMBoK.

Ao compreender esses temas, você estará preparado para os desafios da engenharia de software e sua aplicação no desenvolvimento Java, aprofundando-se nos aspectos técnicos e de gestão essenciais para a área.

## A crise do software e a engenharia de software

A expressão *engenharia de software* surgiu na década de 1960 e foi oficialmente utilizada na *NATO Conference on Software Engineering* em 1968, na Alemanha. O evento abordou a *crise do software*, buscando soluções para tornar o desenvolvimento de sistemas mais sistemático e controlado.

Sistemas de software complexos possuem múltiplos componentes interconectados e dependentes entre si. Essa complexidade pode surgir devido à evolução do sistema e sua adaptação a novas necessidades, tornando essencial a aplicação de princípios de engenharia para garantir eficiência e qualidade.

### A crise do software

A *crise do software* surgiu no final da década de 1960 para descrever os desafios no desenvolvimento de software diante do aumento da complexidade dos sistemas e da falta de técnicas consolidadas. Com computadores mais poderosos, tornou-se possível criar softwares maiores e mais sofisticados, mas a abordagem informal de desenvolvimento resultava em atrasos, altos custos e baixa confiabilidade.

A *NATO Conference on Software Engineering* de 1968 marcou um ponto de virada ao estabelecer práticas mais maduras para o desenvolvimento de software, dando origem à

engenharia de software como disciplina. Estudos, como o de Spector e Gifford (1986), comparavam o desenvolvimento de software à construção de pontes, destacando que, enquanto pontes eram concluídas dentro do prazo e orçamento, softwares frequentemente sofriam atrasos e apresentavam falhas, evidenciando a necessidade de maior rigor técnico na área.

## O Chaos Report

Em 1995, o *Chaos Report*, do *The Standish Group*, revelou que 84% dos projetos de software falhavam de alguma forma, com 31,1% sendo cancelados e 52,7% ultrapassando significativamente custos e prazos. Apenas 16,2% eram concluídos conforme o planejado. O prejuízo estimado nos EUA foi de US\$ 81 bilhões em projetos cancelados e US\$ 59 bilhões em projetos atrasados.

Apesar dos avanços na engenharia de software, desafios persistem. Em 2015, 29% dos projetos foram bem-sucedidos, mas 19% fracassaram e 52% enfrentaram problemas. Grady Booch, co-criador da UML, descreveu essa longa crise como uma verdadeira “anormalidade”, destacando que décadas de experiência ainda não garantiram qualidade e previsibilidade no desenvolvimento de software.