

# Definições e conceitos de software e engenharia de software

A engenharia de software é uma disciplina que abrange todo o ciclo de vida do desenvolvimento de sistemas, desde a definição de requisitos até a manutenção pós-entrega. Seu objetivo é aplicar princípios de engenharia para produzir software de alta qualidade, confiável, eficiente, econômico e alinhado às expectativas do cliente.

Segundo Pressman (2010), a engenharia de software é estruturada em camadas, tendo o *processo* como base que integra todas as tecnologias envolvidas. Métodos definem as técnicas para comunicação, análise de requisitos, modelagem, construção, testes e manutenção, garantindo a aplicação eficaz das práticas de engenharia e a entrega de software com qualidade.

As ferramentas de engenharia de software oferecem suporte automatizado ou semi automatizado aos processos e métodos utilizados no desenvolvimento (PRESSMAN, 2010). Apesar de sua importância, essas técnicas ainda não são amplamente adotadas ou aplicadas corretamente.

Para compreender a engenharia de software, é essencial entender primeiro o conceito de software. Segundo Pressman (2010), o software é composto por:

1. Instruções que executam funções e determinam o desempenho desejado.
2. Estruturas de dados que permitem a manipulação da informação.
3. Documentação sobre operação e uso dos programas.

Dado seu caráter complexo, o software exige um rigoroso controle de qualidade, que deve ser incorporado desde o início do desenvolvimento até a entrega final, garantindo confiabilidade e eficiência no produto final.

## Software x hardware

Segundo Pressman (2010), comparar software e hardware ajuda a entender suas diferenças e características. O desenvolvimento de software se distingue do hardware em vários aspectos:

- **Intangibilidade:** O software pertence a um sistema lógico, enquanto o hardware é um produto físico.

- **Personalização:** O software geralmente é desenvolvido sob medida, enquanto o hardware reutiliza componentes pré-existentes.
- **Custos:** No software, os custos estão no desenvolvimento, não na manufatura.
- **Deterioração:** O software não se desgasta fisicamente, mas pode se deteriorar devido a erros introduzidos em manutenção ou evolução.

Enquanto o hardware sofre falhas iniciais que são estabilizadas com o tempo, o software pode se tornar mais instável à medida que mudanças são aplicadas. Essa peculiaridade exige abordagens específicas para garantir sua qualidade e longevidade.

Diferente do hardware, o software não envelhece fisicamente, mas sua curva de falhas não segue um padrão estável. Inicialmente, há uma alta incidência de erros (bugs), que são corrigidos até a estabilização. No entanto, a manutenção e as mudanças frequentes aumentam a chance de introduzir novos erros, gerando picos na curva de falhas.

Esse ciclo pode transformar o software em uma "colcha de retalhos", com falhas difíceis de prever e corrigir. Assim, em vez de manter indefinidamente um sistema com sucessivas correções, muitas vezes é mais eficiente desenvolver um novo software, alinhado às necessidades atuais e às tecnologias modernas.

Diferente do hardware, onde componentes são padronizados e reutilizados com alta confiabilidade, o reuso de software ainda está em evolução. Apesar de práticas antigas, como bibliotecas científicas, o conceito se expandiu para incluir estruturas de dados, interfaces gráficas e componentes orientados a objetos.

Neste módulo, exploraremos os principais processos de software, incluindo o modelo em cascata, modelos evolucionários (prototipagem, espiral e concorrente), modelos incrementais (RAD e incremental), além do processo unificado e métodos ágeis.