# **Engenharia de Software II – Apresentação**

Os testes unitários são uma prática de verificação do software em que partes isoladas do código, como funções, métodos ou classes, são testadas separadamente para garantir que cada uma funciona corretamente (conforme o esperado). O objetivo é detectar os erros mais cedo, nas fases iniciais do desenvolvimento, para que a manutenção do sistema seja mais fácil e segura (SOMMERVILLE, 2011).

Conforme Jorgensen (2013), os testes unitários são normalmente realizados pelos desenvolvedores durante a codificação e fazem parte das práticas de desenvolvimento orientadas a testes (TDD – Test-Driven Development).

Já o **caso de uso descritivo** é uma técnica de modelagem de requisitos que detalha, em linguagem natural estruturada, os fluxos de interação entre o usuário e o sistema. Ele serve como base para o entendimento funcional do sistema, ajudando no alinhamento entre desenvolvedores e stakeholders (PRESSMAN; MAXIM, 2016).

A **documentação de interface** é outro aspecto essencial, especialmente quando aliada aos princípios de **usabilidade**. Uma interface bem projetada deve ser intuitiva, eficiente e acessível, proporcionando uma boa experiência ao usuário. A usabilidade é avaliada com base em critérios como consistência visual, feedback claro e facilidade de navegação (NIELSEN, 2000).

Esses três pilares — teste unitário, modelagem por caso de uso e usabilidade na interface — formam a base de um desenvolvimento de software centrado na qualidade, contribuindo para sistemas mais robustos, funcionais e adequados às necessidades dos usuários.

## **Referências**

* NIELSEN, Jakob. *Projetando a usabilidade*. Rio de Janeiro: Campus, 2000.
* PRESSMAN, Roger S.; MAXIM, Bruce R. *Engenharia de Software*. 8. ed. Porto Alegre: AMGH, 2016.
* SOMMERVILLE, Ian. *Engenharia de Software*. 9. ed. São Paulo: Pearson Prentice Hall, 2011.
* JORGENSEN, Paul C. Software Testing: A Craftsman’s Approach. 4th ed. Boca Raton: CRC Press, 2013.