

Engenharia de Software

08 - Testes de Software

Prof. Msc. Francisco de Assis de Souza Rodrigues

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

"Procura apresentar-te a Deus aprovado, como obreiro que não tem de que se envergonhar, que maneja bem a palavra da verdade." **2 Timóteo 2:15**

Estudar com dedicação nos torna competentes e confiáveis, não só nos estudos, mas na vida e no trabalho.

🎯 Objetivos

- Compreender o papel dos testes no processo de desenvolvimento de software.
- Identificar e aplicar diferentes tipos de teste (de componentes, de integração, de sistema, de aceitação).
- Entender e praticar o Desenvolvimento Orientado a Testes (TDD).
- Diferenciar testes de liberação e testes com usuários.
- Aplicar testes unitários reais nas funcionalidades do projeto da turma.

UniFAP

Conceito e Objetivo dos Testes

- Teste é o processo de executar um programa com a intenção de encontrar erros.
- Objetivo não é provar que o sistema funciona, mas tentar fazê-lo falhar.
- Complementa a verificação estática (revisões, inspeções, análise de código).

Sommerville destaca que o teste deve ser planejado sistematicamente, e não improvisado após a implementação.

Tipos de Teste segundo Sommerville

Tipo de Teste	Objetivo	Nível	Exemplo
Teste de Componentes (Unitário)	Verificar o funcionamento isolado de classes/métodos.	Baixo	Testar um método `calcularTotal()`
Teste de Integração	Avaliar a interação entre módulos.	Médio	Testar comunicação entre módulos de login e banco de dados
Teste de Sistema	Verificar o sistema completo conforme os requisitos.	Alto	Testar todo o fluxo de compra de um e-commerce
Teste de Aceitação (ou com Usuário)	Confirmar se o sistema atende às necessidades do cliente.	Final	Usuário executa cenários reais

UniFAP

Test-Driven Development (TDD)

- Técnica onde os testes são escritos **antes** do código.
- Ciclo proposto por Sommerville:

1. **Escreva um teste** para uma nova funcionalidade.
2. **Execute o teste** – ele deve falhar (porque a funcionalidade ainda não existe).
3. **Implemente o código** mínimo necessário para passar o teste.
4. **Reexecute o teste** – ele deve passar.
5. **Refatore** o código mantendo os testes verdes.

Sommerville ressalta que o TDD melhora o design, reduz defeitos e gera uma base de regressão automatizada.

UniFAP

Testes de Liberação e Testes com Usuários

- **Testes de Liberação (Release Testing):**
  - Executados em um sistema integrado antes da entrega.
  - Enfatizam **confiabilidade, desempenho e validação dos requisitos**.
  - São responsabilidade da equipe de QA (não apenas dos desenvolvedores).

UniFAP

Testes de Liberação e Testes com Usuários

- **Testes com Usuário (User Testing):**
  - Feitos por **usuários reais ou representantes**.
  - Incluem **testes alfa, beta e de aceitação**.
  - Foco na **usabilidade e adequação ao contexto de uso.**

Sommerville reforça que testes com usuário são fundamentais em sistemas críticos ou interativos.

UniFAP

Atividade – Testes Unitários

Objetivo: Garantir que funções isoladas do sistema funcionam corretamente.

1. Escolha uma funcionalidade, implemente (ex: cálculo de desconto, login, validação de dados).
2. Escreva testes unitários com base em casos típicos e de exceção.
3. Execute os testes e registre os resultados.

Ferramentas sugeridas:

- **Python:** pytest
- **JavaScript/React:** Jest
- **GoLang:** testing
- **PHP:** PHPUnit

UniFAP

Atividade – Ciclo TDD

Objetivo: Aplicar o ciclo *Red → Green → Refactor* em uma nova funcionalidade simples.

Exemplo de roteiro (para qualquer linguagem):

1. **Escreva um teste** para uma função que valide e-mails.
2. **Execute o teste** – deve falhar.
3. **Implemente a função mínima** que faz o teste passar.
4. **Execute novamente** – agora deve passar.
5. **Refatore** o código mantendo o teste verde.

UniFAP

Atividade – Teste de Aceitação Simulado

Objetivo: Reproduzir o processo de teste com usuário (conforme Sommerville).

1. Formem duplas: um aluno faz o papel de **usuário**, outro de **testador**.
2. O “usuário” executa um fluxo real no sistema (ex: cadastro, compra, envio).
3. O “testador” registra os resultados e comentários do “usuário”.
4. Discutam as observações de usabilidade e adequação.

UniFAP

Bibliografia

SOMMERVILLE, Ian. Engenharia de software. 10. ed. São Paulo, SP: Pearson, 2018. E-book. Disponível em: https://plataforma.bvirtual.com.br. Acesso em: 15 ago 2025.

https://plataforma.bvirtual.com.br/Acervo/Publicacao/168127



Referências

CHATGPT. Expansão conceitual sobre integração entre COBIT, ITIL e PMBOK. OpenAI, 2025. Disponível em: https://chat.openai.com/. Acesso em: 11 ago. 2025.


UniFAP


OBRIGADO
Arigato
Spasibo
Obrigado
Dankon
Maake
Multumesc
Ua Tsaug Rau Koj
Mochchakkeram
Kiitos
Merci
Raibh Maith Agat
Chokrane
Grazie
Terma Kasih
Asante
Mamana
Kia Ora
THANK YOU
Jusparxar
Welalin
Kiitos
Grazie
Obrigado
Arigato
Spasibo
Merci
Chokrane
Grazie
Kiitos
Multumesc
Terma Kasih
Maake
Mochchakkeram
Kiitos
Grazie
Obrigado
Merci
Raibh Maith Agat
Kiitos
Multumesc
Kia Ora
Grazie
Kiitos
Obrigado
Grazie
Chokrane
Merci
Mochchakkeram
Grazie
Kiitos
Obrigado
Grazie
Multumesc
Merci
Mochchakkeram
Chokrane
Grazie