

A photograph of a large, multi-story concrete building with a grid-like facade of windows. The word "SENAI" is mounted on the upper part of the building in large, red, three-dimensional block letters. To the right of the letters, a flag is visible. The building is surrounded by green trees and a clear blue sky.

SENAI

Testes de Software

Prof. Fernanda Fretes

1. Testes

1.1. Definição

1.2. Tipos

1.2.1. Funcionais

1.2.2. Não funcionais

1.3. Níveis

1.3.1. Unitário

1.3.2. De integração

1.3.3. De sistema

1.3.4. De aceitação

1.4. Técnicas

1.4.1. Regressão

1.4.2. Estresse

1.4.3. Recuperação

1.4.4. Performance

1.4.5. Segurança

1.4.6. Paralelo

2. Planejamento de testes

2.1. Análise de risco

2.2. Plano de teste

3. Execução de testes

3.1. Ambiente de teste

3.1.1. Configuração

3.2. Equipe de testes

3.3. Casos de teste

3.4. Ferramentas

3.4.1. Gestão de teste

3.4.2. Gestão de defeitos

3.5. Relatório de teste

3.6. Normalização

1.3. Níveis

1.3.1. Unitário

1.3.2. De integração

1.3.3. De sistema

1.3.4. De aceitação



Teste de Integração

Os testes de integração são uma etapa essencial do ciclo de desenvolvimento de software, focando na verificação da comunicação entre diferentes módulos ou sistemas.

Enquanto os testes unitários garantem que cada componente funcione individualmente, os testes de integração asseguram que esses componentes funcionam corretamente em conjunto.

Teste de Integração

Diferença entre Testes Unitários, de Integração e de Sistema

Testes Unitários:

Avaliam módulos individuais isoladamente.

Testes de Integração:

Avaliam a interação entre módulos.

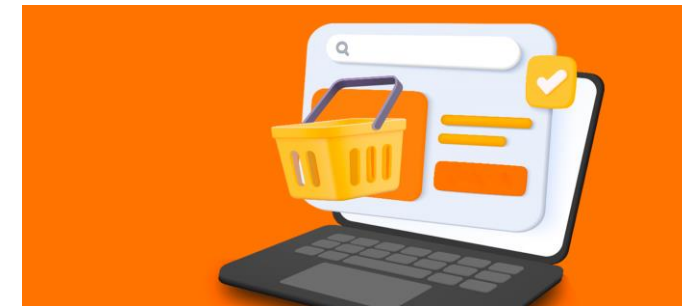
Testes de Sistema:

Testam a aplicação como um todo em um ambiente realista.

Teste de Integração

Imagine uma loja online que tem três sistemas principais:

- carrinho de compras,
- processamento de pagamento;
- sistema de estoque.



Testes de integração verificariam se, ao finalizar uma compra, o sistema:

- Deduz corretamente os itens do estoque.
- Processa o pagamento e retorna a confirmação.
- Gera um pedido e notifica o usuário.

Sem testes de integração, uma falha na comunicação entre esses módulos poderia gerar pedidos sem confirmação de pagamento ou cobranças duplicadas.

Teste de Integração

Benefícios dos Testes de Integração



Detecção antecipada de falhas:
Problemas são encontrados antes que afetem o usuário final.

Melhor qualidade do software: Redução de bugs e melhor estabilidade do sistema.

Redução de custos: Corrigir falhas antes da produção é mais barato.

Maior confiabilidade e segurança: Garante que a interação entre serviços seja segura.

Teste de Integração

Exemplo do Mercado

A Amazon investe pesadamente em testes de integração para garantir que seu sistema de recomendações funcione corretamente com seu sistema de pagamento e estoque. Isso evita que um produto fora de estoque seja sugerido erroneamente para compra.



Teste de Integração

Big bang

Todos os módulos são testados simultaneamente.

Rápido, mas pode ser difícil identificar falhas individuais.

Exemplo: Um banco lançando um novo sistema que conecta serviços de internet banking, cartão de crédito e atendimento ao cliente de uma vez.

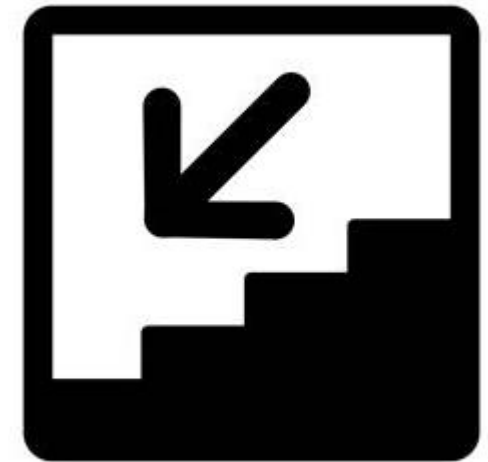


Teste de Integração

Top-Down

Testa primeiro os módulos de mais alto nível.
Usa "stubs" para substituir componentes ainda não desenvolvidos.

Exemplo: Um app móvel de delivery pode testar a interface de usuário antes da implementação final da API do restaurante.



Teste de Integração

Bottom-Up

Testa primeiro os módulos de baixo nível.
Usa "drivers" para simular entradas superiores.

Exemplo: Um sistema bancário testando cálculos financeiros antes da interface de relatórios estar pronta.



Teste de Integração

Sanduíche (Híbrido)

Combina top-down e bottom-up.
Equilibra velocidade e precisão na detecção de falhas.

Exemplo: Um e-commerce testando tanto a interface do cliente quanto os serviços de back-end simultaneamente.



Teste de Integração

Ferramentas e Estratégias de Testes de Integração

Postman: Teste de APIs.

Selenium: Testes de interface web.

JUnit/PyTest: Testes automatizados em Java e Python.

Mocking/Stubbing: Simulação de componentes não finalizados.



Teste de Integração

Escrevendo e Executando Testes de Integração

Definir cenário de teste: O que precisa ser testado?

Criar dados de teste: Simular entradas reais.

Executar testes: Usando ferramentas como Postman, Selenium ou JUnit.

Analisar resultados: Identificar e corrigir erros.



Teste de Integração

Casos Reais de Falhas por Falta de Testes de Integração

NASA Mars Climate Orbiter (1999): Falha na conversão entre unidades métricas e imperiais.

Knight Capital (2012): Bug em software de trading causou prejuízo de \$440 milhões.

Apple Maps (2012): Locações erradas devido a falhas de integração de dados.

