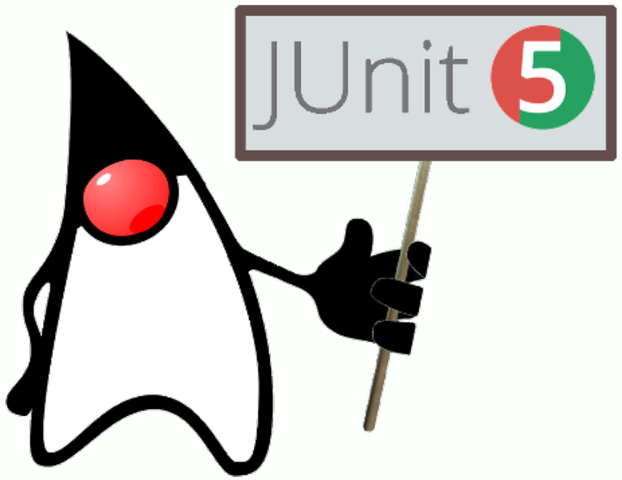
UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Introdução aos testes unitários

QXD0007 - Programação Orientada a Objetos



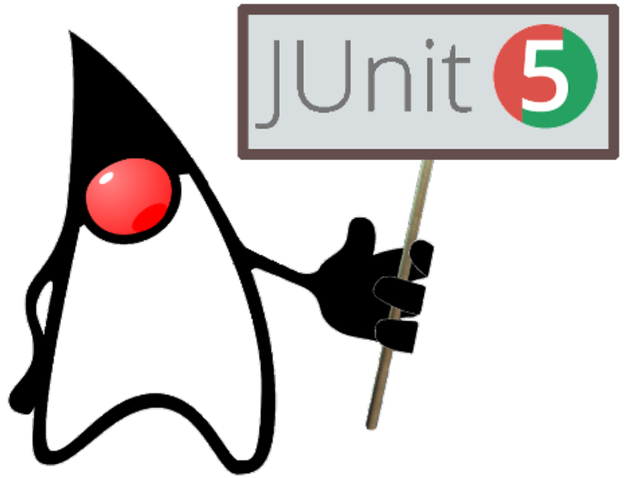
Prof. Bruno Góis Mateus (brunomateus@ufc.br)

[1]

Conteúdo

- Introdução
- Testes unitários
- Testes nas atividades

Introdução



[1]

Introdução

O que são testes de software ?

São processos que verificam se o software funciona conforme esperado.

Introdução

Qual o propósito de testes?

- Identificar erros antes que cheguem ao usuário
- Garantir que as funcionalidades implementadas atendem aos requisitos
- Melhorar a qualidade do software

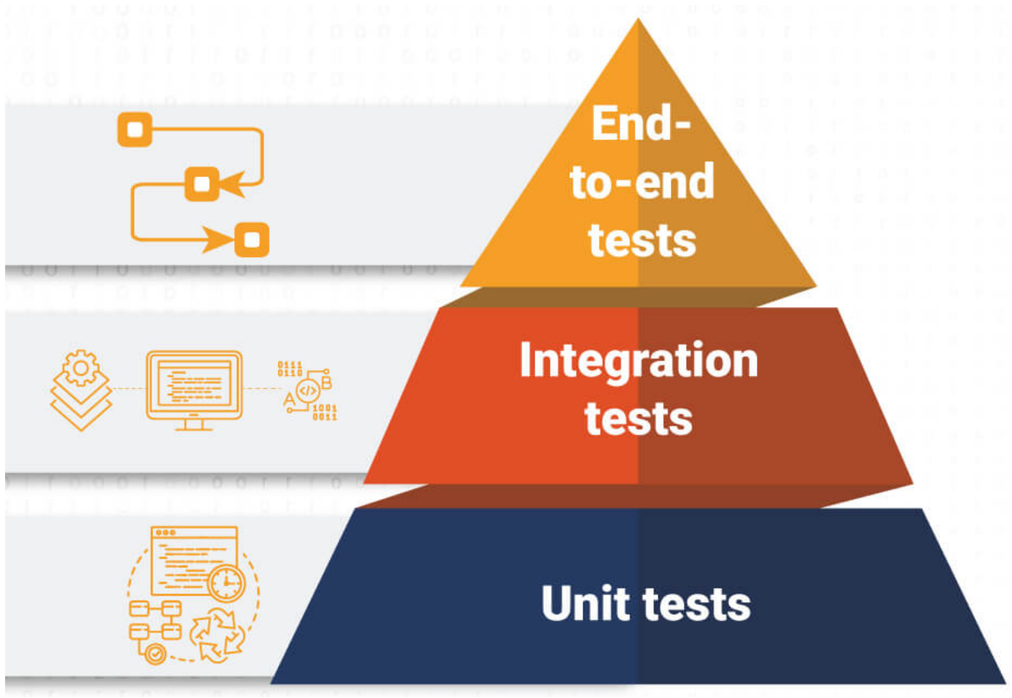
Introdução

Origem de falhas

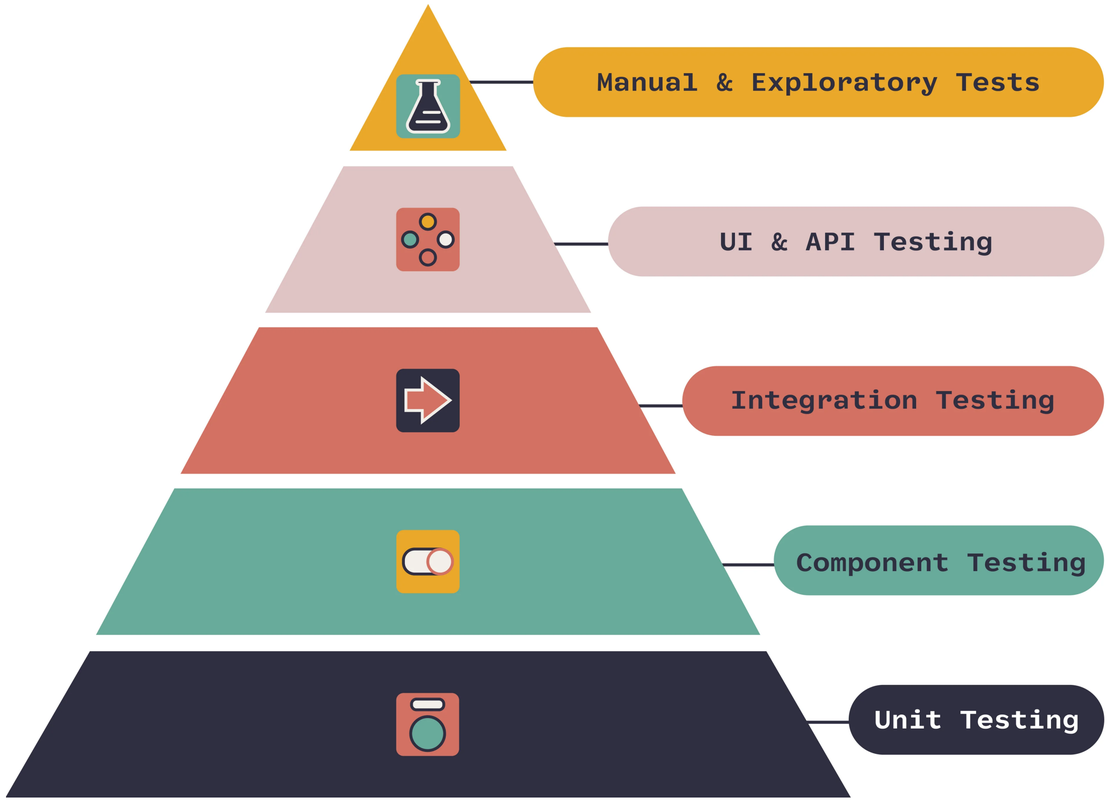
- Falhas podem ser originadas por diversos motivos
  - Especificação pode estar errada ou incompleta
  - Requisitos impossíveis de serem implementados, devido a limitações de hardware ou software
  - A implementação também pode estar errada ou incompleta, como um erro de um algoritmo

Introdução

Tipos de testes



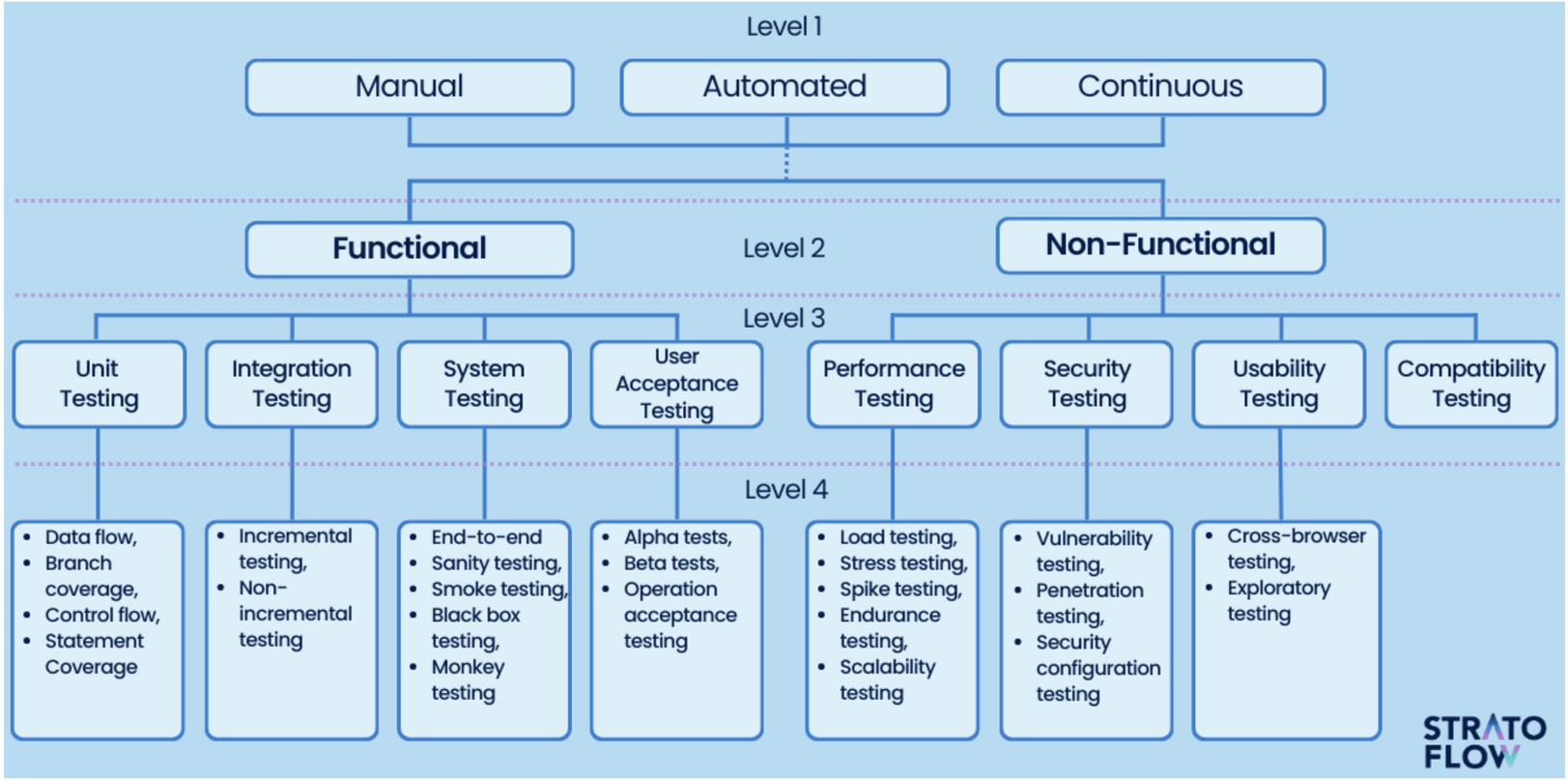
[2]



[3]

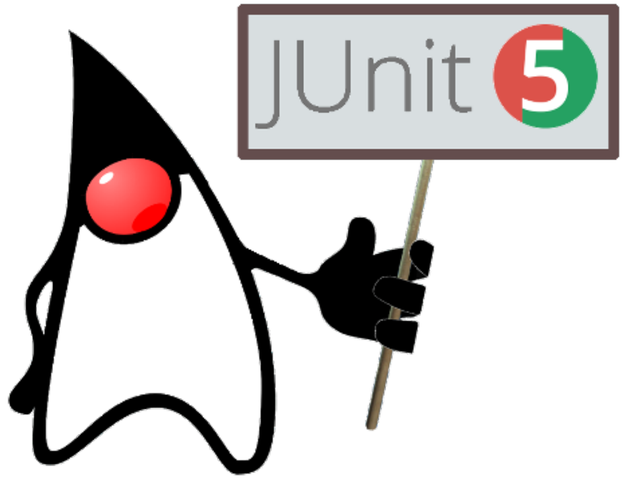
Introdução

Tipos de testes



[4]

Testes unitários



[1]

Testes unitários

- Verificam o comportamento de unidades individuais do código
  - Geralmente métodos ou funções
- Características:
  - São pequenos e específicos
  - Testam uma única funcionalidade por vez
  - Executam de forma rápida e isolada

Testes unitários

Benefícios

- Identificação precoce de erros:
  - Erros são corrigidos antes de integrar diferentes partes do sistema
- Facilidade de manutenção:
  - Alterações no código podem ser verificadas rapidamente
- Documentação do comportamento esperado:
  - Testes servem como exemplos de uso do código
- Maior confiança no software:
  - Reduz o risco de que mudanças introduzam bugs

Testes unitários

Um bom teste unitário

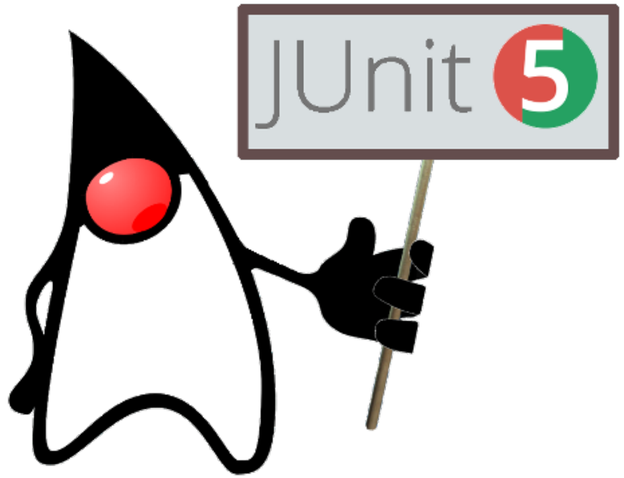
- Deve ser possível automatizar
- Pode ser executado em qualquer ordem
- São executados em memória (sem acesso ao banco de dados ou arquivos)
- Devem ser consistente (retornar o mesmo resultado)
- Devem ser de rápida execução
- Devem ser legíveis
- Devem ser de fácil manutenção

Testes unitários

Como funcionam?

- Entradas e Saídas:
  - Um teste unitário fornece entradas específicas e verifica se a saída está correta
- Assertivas:
  - São condições que devem ser verdadeiras para o teste passar
- Automação:
  - Testes são executados automaticamente por ferramentas como JUnit, Pytest, Mocha, etc

Testes nas atividades



[1]

Testes nas atividades

- Os alunos não precisarão escrever os testes.
  - Os testes já estarão implementados e serão usados para:
    - Avaliar o código
    - Validar se o comportamento está correto

Testes nas atividades

Exemplo

```
@Test
public void testDeposito() {
    ContaBancaria conta = new ContaBancaria();
    conta.depositar(100);
    assertEquals(100, conta.getSaldo());
}
```

Testes nas atividades

Vantagens

- Feedback imediato:
  - Testes falham imediatamente ao detectar problemas
- Redução de retrabalho:
  - Encontrar erros cedo economiza tempo
- Apoio ao aprendizado:
  - Testes ajudam a entender o comportamento esperado do código

Testes nas atividades

Limitações dos testes unitários

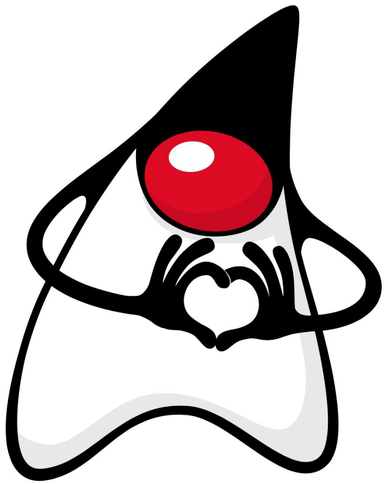
- Não substituem outros tipos de testes:
  - Eles não verificam integração ou requisitos do sistema completo
- Cobertura limitada:
  - Se mal projetados, podem não cobrir todos os casos relevantes

Testes nas atividades

Conclusão

- Testes unitários são fundamentais para o desenvolvimento moderno:
  - Garantem qualidade
  - Facilitam manutenção
  - Reduzem erros no software
- Durante as atividades:
  - Aproveitem os testes fornecidos como uma ferramenta para orientar seu trabalho
  - Corrijam os erros apontados e usem os testes como um guia

Por hoje é só



Mão na massa

NOT VISIBLE