

# TIPOS DE TESTE: O ALVO DO TESTE

PÓS EM DESENVOLVIMENTO DE APLICAÇÕES WEB  
MÓDULO TESTE DE SOFTWARE PARA SISTEMAS WEB

MSc. Rafael Ramos  
2024

# SUMÁRIO

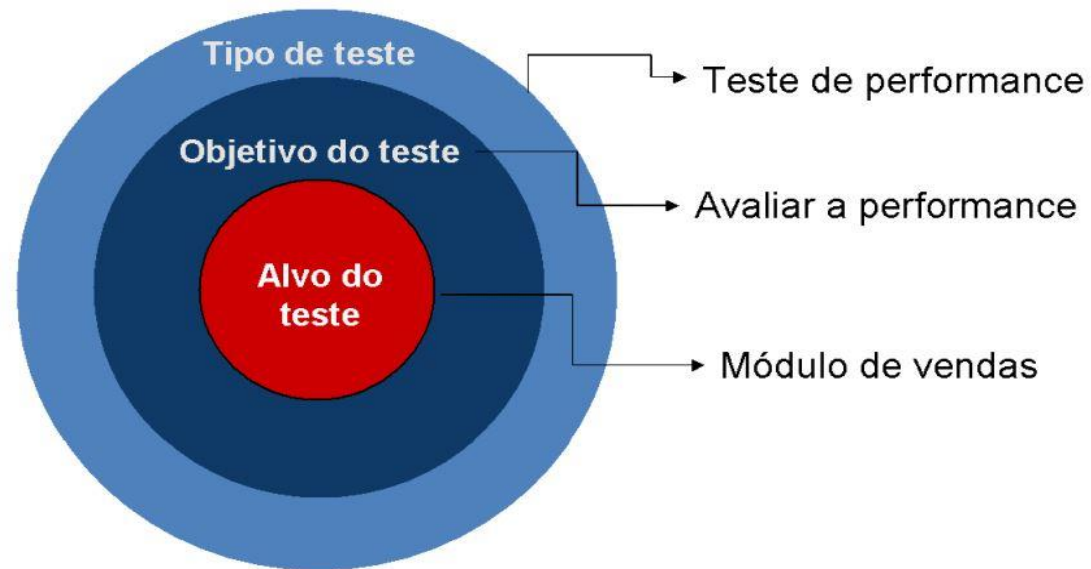
1. Definições
2. Pirâmide dos Testes Ágeis
3. Teste Funcional
4. Teste Estrutural
5. Teste Não Funcional
6. Testes Relacionados a Mudanças



# DEFINIÇÕES

*“Alvo é um Objetivo a que se dirige algum intento; fim, meta”*

- Tipo de teste: Um grupo de atividades de teste destinado a testar características específicas de um sistema de software, ou parte de um sistema, com base em um objetivo específico.



# DEFINIÇÕES

- Um tipo de teste é focado em um objetivo específico de teste, como por exemplo:

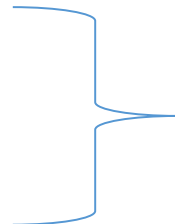
- Avaliar Funcionalidades
- Medir a confiabilidade
- Avaliar a usabilidade
- Avaliar a estrutura do sistema



Para avaliar cada objetivo, um **tipo de teste** é utilizado.

- Quatro grandes grupos de tipos de teste:

- Testes funcionais
- Testes não-funcionais
- Testes estruturais
- Testes relacionados a mudanças

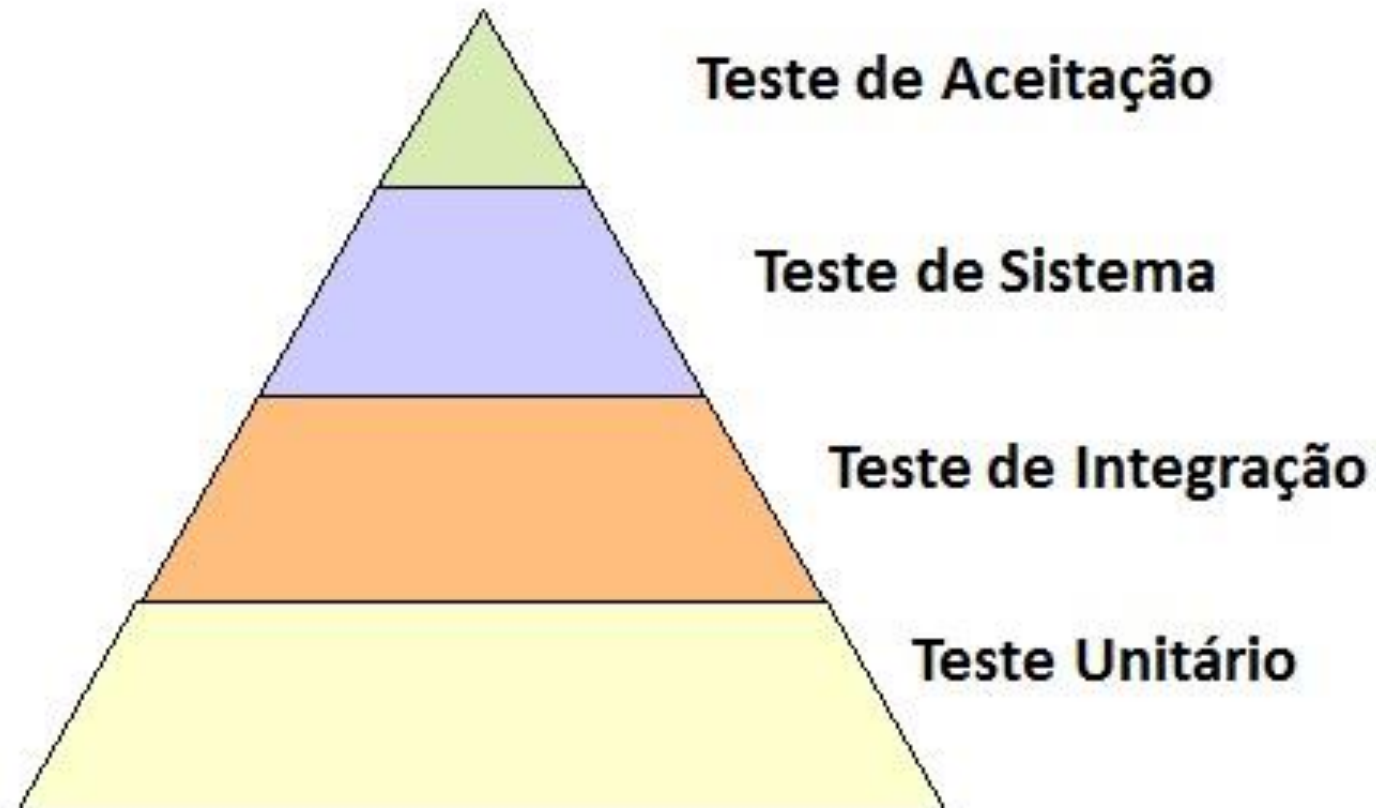


**Agrupadores dos tipos de teste**

**OBS:** Podemos ter um tipo de teste ocorrendo em um ou mais níveis/fases de teste.

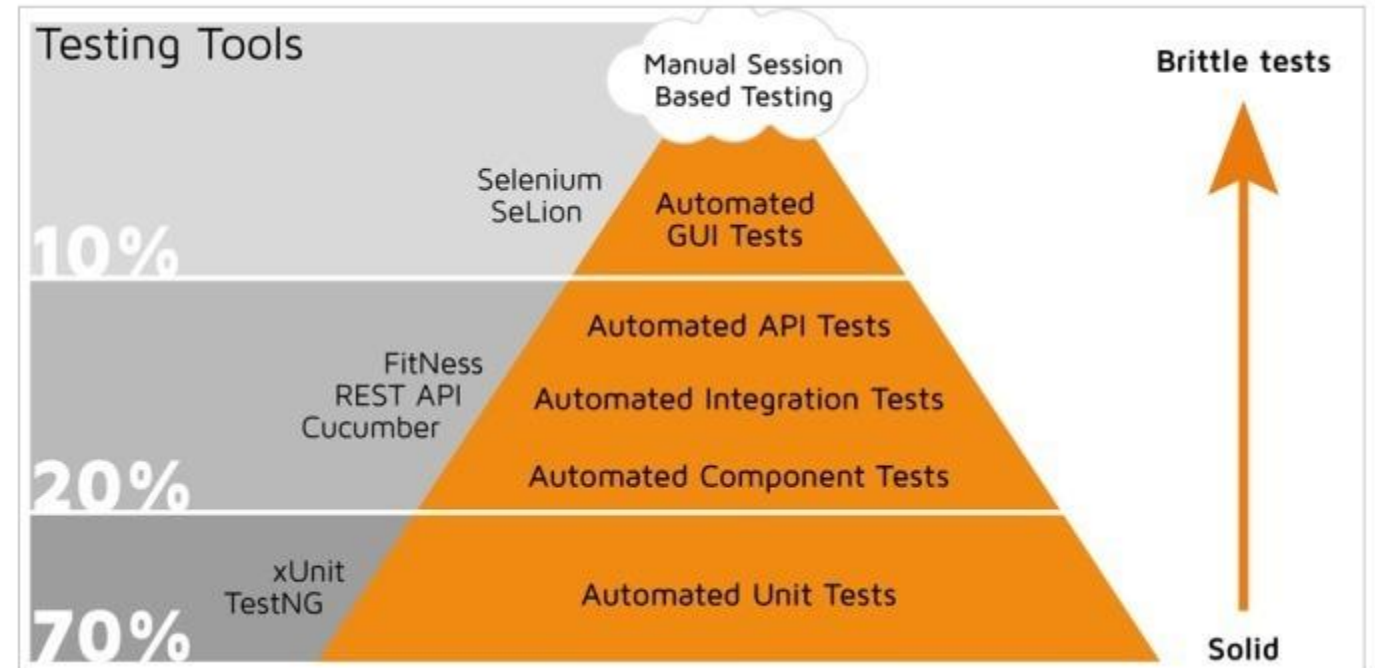
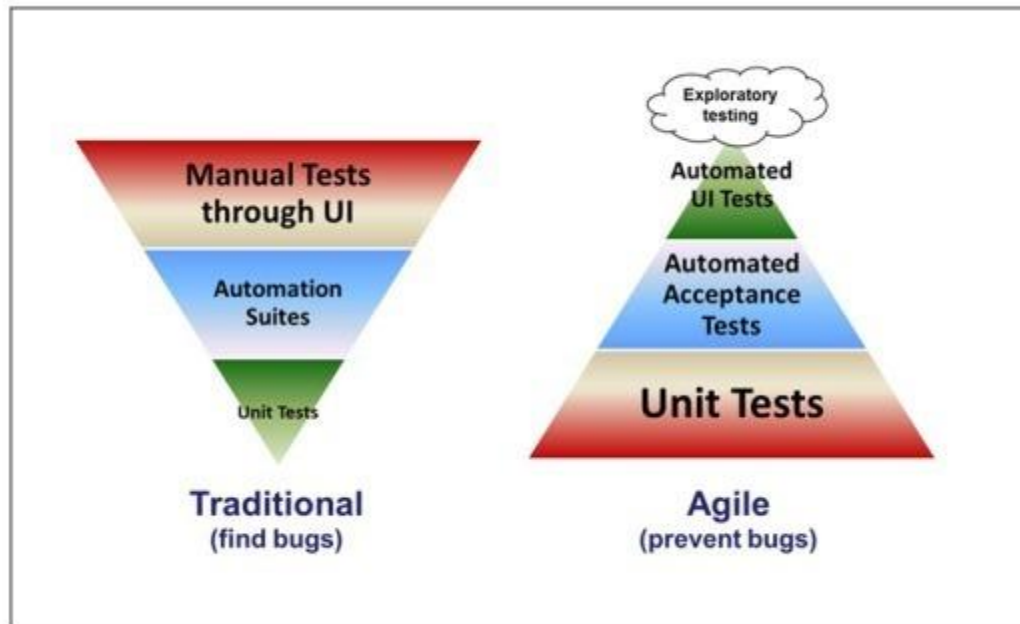
# DEFINIÇÕES

- **Níveis de Testes:**



# DEFINIÇÕES

- **Pirâmide de Testes Ágeis:**



# TESTE FUNCIONAL

- **Conceito:**

- Teste funcional (caixa-preta) analisa o item a ser testado sob o aspecto funcional, observando apenas qual foi o resultado obtido e não como o mesmo foi obtido.
- Avalia as funções do software (“o que o sistema faz”).
  - Estas funções podem estar documentadas em especificações de requisitos, casos de uso e critérios de aceitação.
- Tem como meta a verificação da aceitação dos dados, do processamento e da resposta a este processamento, e a implementação apropriada das regras de negócio.
- Para execução dos testes funcionais serão utilizados os procedimentos a serem definidos nos casos de teste.
- Um dos métodos para realizar o teste funcional é executar a função do sistema utilizando dados válidos e inválidos
- Deverá ocorrer em ambiente de testes separado do ambiente de desenvolvimento ou de produção.



# TESTE ESTRUTURAL

- **Conceito:**
  - Teste estrutural (caixa-branca) analisa a estrutura interna e o comportamento a ser testado, necessitando que o DEV entenda como o componente foi codificada.
  - Nesses tipo de teste, devem ser observados todos ou quase todos os possíveis caminhos e condições de parada de um fluxo de eventos associados ao componente analisado.
  - Com esse tipo de teste conseguimos medir o quanto de código foi testado e o quanto não foi (cobertura de código).
  - Pode ser executado em todos os níveis de testes.
  - Garante que:
    - Todos os caminhos independentes dentro de um programa tenham sido exercitados pelo menos uma vez.
    - Todas as decisões lógicas sejam exercitadas.
    - Todos os laços em suas fronteiras e dentro de seus limites tenham sido executados.
    - Todas as estruturas de dados internas tenham sido exercitadas para garantir a sua validade.



# TESTE ESTRUTURAL (TESTES UNITÁRIOS)

- São testes que verificam se uma parte específica do código, normalmente a nível de função, está funcionando corretamente.
- No momento da escrita do teste não leva-se em consideração a implementação da função. Deve-se focar em verificar se a função cumpre o seu papel. Ex: Teste unitário sobre uma função de listagem de usuários (validar se uma lista de usuários é retornada).
- Exemplos de Framework para criação de Unit Tests: MSTest, NUnit, XUnit, JUnit...

# TESTE ESTRUTURAL (TESTES UNITÁRIOS)

```
it('Debe retornar uma lista de usuários', ()=>{  
  component.users = [];  
  let userList = component.listUsers();  
  expect(userList.length).toBe(0); 1  
  component.users = [{name: 'Ricardo', age: 20}, {name: 'Pedro', age: 30}];  
  userList = component.listUsers();  
  expect(userList.length).toBe(component.users.length) 2  
})
```

- Exemplo de Unit Test feito em angular para validar uma função de listagem de usuários, onde em:
  - 1 — Verifica se a lista está vazia;
  - 2 — Verifica se a listagem não está vazia.

# TESTE NÃO-FUNCIONAL

- **Conceito:**
- Teste não-funcional analisa os aspectos que são importantes ainda que não estejam relacionados diretamente às funções que o sistema desempenha.
- O termo teste não-funcional descreve que o teste é executado para medir as características que podem ser quantificadas em uma escala variável (exemplo: o tempo de resposta em um teste de performance).
- Tipos de teste não-funcional incluem, mas não se limitam a:
  - Teste de performance
  - Teste de carga
  - Teste de estresse
  - Teste de recuperação de falhas
  - Teste de instalação



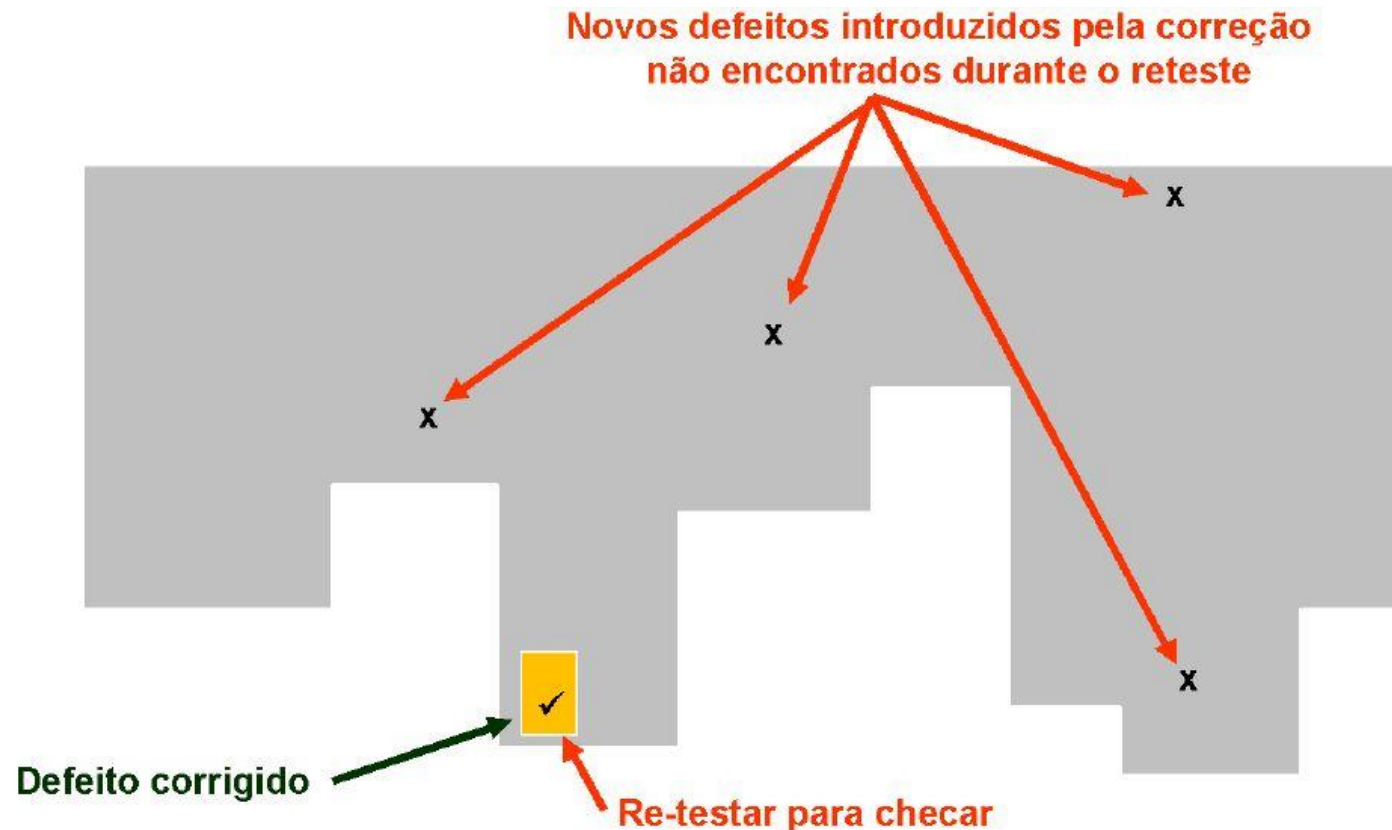
# TESTE RELACIONADO A MUDANÇAS

- **Conceito:**
- **Teste de confirmação:** teste que executa casos de teste reprovados durante sua última execução. Este procedimento é feito para verificar o sucesso das ações corretivas.
- **Teste de regressão:** teste realizado em partes do software que não sofrem alterações, mas que podem ter sofrido impacto negativo após algumas modificação feita em outras regiões correlatas do software. Este teste é realizado quando o software ou seu ambiente for alterado.



# TESTE DE CONFIRMAÇÃO

- O desenho em cinza representa todo o software e nele podemos ver que um defeito foi encontrado e já foi corrigido. Devemos então aplicar o reteste para checar se o defeito foi realmente corrigido.



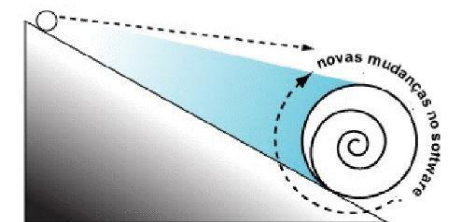
# TESTE DE REGRESSÃO

- Depois dos defeitos iniciais terem sido resolvidos, a solução de um defeito pode revelar ou criar outro defeito escondido no código. Isso também pode ocorrer em caso de uma mudança no software, não necessariamente proveniente de um defeito. A proposta do teste de regressão é determinar se o sistema ou a qualidade do sistema regrediu após uma mudança.
- Na ilustração abaixo a parte destacada em verde simula a parte alterada do software:



# TESTE DE REGRESSÃO

- **Conceito:**
- Uma situação que durante anos tem sido considerada uma praga para os profissionais de TI é o efeito bola de neve ou cascata de alterações feitas nos sistemas de software.
- As mudanças incorretamente implementadas causam novos problemas ou introduzem novos dados ou parâmetros que causam problemas em partes do software já testadas.
- Testes de regressão retestam partes do software previamente testadas para assegurar que elas funcionam corretamente depois que foram feitas mudanças em outras partes da aplicação.
- O sucesso do teste de regressão dependerá muito da qualidade da análise de impacto realizada.
- Teste e regressão devem ser executados se houver mudanças no software.
- Teste de regressão podem ser realizado em todos os níveis de teste.
- As funções do software que precisam ser frequentemente regredidas são fortes candidatas à automatização dos testes.



# AUTOMAÇÃO DE TESTES

- **Conceito:**
- A automação do teste envolve utilizar ferramentas de execução e armazenamento de teste. Tais ferramentas permitem re-executar testes que já foram executados de forma muito mais rápida e a um custo mais baixo. Contudo, o esforço para automatizar e manter o teste atualizado é alto.





# OBRIGADO!

## RAFAEL RAMOS

Software Quality Assurance  
rafael.anderson@assert.ifpb.edu.br  
+55 83 994053557

