

# Testes de Software

Tipos de Teste

# Tipos de Teste

- Caixa Branca
  - Avaliam a lógica interna e a estrutura do código, garantindo que cada caminho de execução funcione corretamente
- Caixa Preta
  - Avaliam a funcionalidade do software com base nos requisitos e especificações, sem considerar a estrutura do código.

# Tipos de Teste

- Teste Unitário
  - Verificação de unidades individuais de código
    - Métodos Estruturas condicionais e de repetição, etc.
  - Garante que cada parte funcione corretamente de forma isolada

# Tipos de Teste

## - Teste Unitário

```
public void Somar_DoisDouble_RetornaDouble()
{
    // Arrange
    var num1 = 2.9;
    var num2 = 3.1;
    var valorEsperado = 6;

    // Act
    var soma = Calculo.Somar(num1, num2);

    //Assert
    Assert.Equal(valorEsperado, soma);
}
```

# Tipos de Teste

- Teste de Integração
  - Verificação a interação entre diferentes componentes
  - Garante que os componentes funcionem juntos corretamente
  - Exemplos:
    - Testes de interfaces entre módulos
    - Validação de chamadas de Banco de Dados
    - Validação de chamadas de API entre serviços

# Tipos de Teste

## - Teste de Integração

```
[TestFixture]
0 references | Alexandre Cordeiro Trapp, 20 minutes ago | 1 author, 1 change
public class ProgramaVendaCervejaTest
{
    [Test]
    0 references | Alexandre Cordeiro Trapp, 20 minutes ago | 1 author, 1 change
    public void deve_cadastrar_a_venda_de_10_cervejas_brahma_4_reais_cada_e_cinco_stella_6_reais_mais_3_porcento_na_base_de_dados_totalizando_uma_venda_de_73_reais()
    {
        // prepare
        var programaVenda = new ProgramaVendaCerveja(Repositories._vendaCervejaRepository);

        // action
        programaVenda.EfetuarVendaCerveja();

        // verify
        var listaCervejasVendidas = ObterVendasDeCerveja();

        const int BRAHMA = 0;
        const int STELLA = 1;

        Assert.AreEqual("brahma", listaCervejasVendidas[BRAHMA].Marca);
        Assert.AreEqual(4.00m, listaCervejasVendidas[BRAHMA].ValorUnitario);
        Assert.AreEqual(10, listaCervejasVendidas[BRAHMA].Quantidade);
        Assert.AreEqual(40.00m, listaCervejasVendidas[BRAHMA].ValorTotalVenda);

        Assert.AreEqual("stella artois", listaCervejasVendidas[STELLA].Marca);
        Assert.AreEqual(6.00m, listaCervejasVendidas[STELLA].ValorUnitario);
        Assert.AreEqual(5, listaCervejasVendidas[STELLA].Quantidade);
        Assert.AreEqual(33.00m, listaCervejasVendidas[STELLA].ValorTotalVenda);

        Assert.AreEqual(73.00m, (programaVenda.ValorTotalDasVendas));
    }
}
```

# Tipos de Teste

- Teste de Sistema
  - Avalia o sistema completo integrado
  - Foco em requisitos funcionais e não funcionais
  - Testes E2E (End to End)
    - Simulação de cenários reais de uso
    - Verificar o funcionamento do fluxo completo de uma funcionalidade

# Tipos de Teste

- Teste de Aceitação
  - Avalia se o sistema atende os critérios de aceitação
  - Foco nos requisitos especificados pelo cliente
  - Critérios de aceitação:
    - Conjunto de condições que o software deve cumprir
    - Definidos em conjunto com os stakeholders



# Tipos de Teste

- Teste de Carga
  - Avalia o desempenho do sistema sob condições normais e esperadas de carga
  - Exemplo:
    - Simular 1000 usuários acessando um site de e-commerce durante um dia normal de operação

# Tipos de Teste

- Teste de Capacidade
  - Determina a quantidade máxima de carga que o sistema pode aceitar até que fique inaceitavelmente lento ou ocorra um crash
  - Exemplo:
    - Adicionar usuários acessando um site de e-commerce de forma gradual até que o sistema rompa.

# Tipos de Teste

- Teste de Stress
  - Avalia o comportamento do sistema sob condições extremas além da capacidade normal de operação
  - Exemplo:
    - Simular 20000 usuários acessando um site de e-commerce durante uma promoção relâmpago para ver como o sistema lida com essa carga extrema.

# Tipos de Teste

- Teste de Fumaça (Smoke Test)
  - O smoke test ou build verification test é uma abordagem de teste inicial aplicado a um novo build com foco em verificar se às **funcionalidades básicas** funcionam corretamente.
  - Tem como objetivo principal garantir que a build é estável o suficiente para continuar com testes mais detalhados.

# Tipos de Teste

- Teste de Fumaça (Smoke Test)
  - Características:
    - Testes básicos e essenciais
    - Rápido e abrangente
    - Automatizado ou Manual
    - Critério de aprovação/rejeição

# Tipos de Teste

- Teste de Fumaça (Smoke Test)
  - Benefícios:
    - Identificação precoce de problemas
    - Qualidade contínua
    - Confiança no build
    - Eficiência

# Tipos de Teste

- Teste de Fumaça (Smoke Test)
  - Exemplos:
    - Login e Logout
    - Navegação básica
    - Principais Funcionalidades
      - Exemplo: CRUD
    - Interface de usuário

# Tipos de Teste

- Teste de Fumaça (Smoke Test)
  - Quando executar?
    - Após cada novo build
    - Antes de testes de regressão
    - Após implementação de correções críticas



# Tipos de Teste

- Teste de Regressão
  - Foca na **estabilidade geral** da aplicação, verificando se novas alterações no código não introduziram novos defeitos em partes que já funcionavam corretamente.
  - Pode incluir a reexecução de todos testes existentes

# Tipos de Teste

- Teste de Regressão
  - Características:
    - Foco na estabilidade
    - Extensivo e Abrangente
    - Idealmente Automatizado
    - Iterativo

# Tipos de Teste

- Teste de Regressão
  - Benefícios:
    - Detecção precoce de defeitos
    - Manutenção da qualidade
    - Confiança no desenvolvimento
    - Redução de riscos

# Tipos de Teste

- Teste de Regressão
  - Quando executar?
    - Correções de bugs
    - Novas funcionalidades
    - Refatoração de código

# Tipos de Teste

- Teste de Sanidade (Sanity)
  - É um subconjunto dos testes de regressão
  - Tem como objetivo verificar rapidamente se uma parte específica do software que foi recentemente modificada, funciona corretamente.
  - Foco em áreas específicas

# Tipos de Teste

- Teste de Sanidade (Sanity)
  - Características:
    - Foco Restrito
    - Rapidez
    - Executados após modificações
    - Manuais ou Automatizados

# Tipos de Teste

- Teste de Sanidade (Sanity)
  - Benefícios:
    - Feedback rápido
    - Menor custo
    - Foco em áreas impactadas
    - Complemento aos testes de regressão

# Tipos de Teste

- Teste de Sanidade (Sanity)
  - Quando realizar?
    - Correções de bugs
    - Pequenas atualizações
    - Modificações específicas



# Tipos de Teste

- Teste Exploratório
  - Características
    - Não estruturado
    - Baseado em conhecimento
    - Interativo

# Tipos de Teste

- Teste Exploratório
  - Benefícios
    - Adaptabilidade
    - Defeitos inesperados
    - Foco na UX

# Tipos de Teste

- Teste Exploratório
  - Quando utilizar?
    - Funcionalidades novas
    - Fase inicial do projeto
    - Complemento a testes estruturados
    - Pouca documentação