

Projeto, implementação e Teste de Software

# Artefatos de Teste de Software

Prof. Esp. Dacio F. Machado



LOADING...



# Método de Avaliação II → TURMA B

- *1º Bimestre*

*1,0 - Atividade de Estudo Programada.*

*1,0 - Prova Integrada.*

***8,0 - Avaliação do Professor***

*Projeto em Grupo de Implementação e Documentação de UMA Técnicas de Teste - 4,0*

*Prática e participação em Laboratório - 2,0*

*Prova Escrita - 2,0*

*Total 8,0*





## Método de Avaliação II → TURMA B

### ***Projeto em Grupo de Implementação e Documentação de Técnicas de Teste - 4,0***

Desenvolver os artefatos, implementar os testes, executar e gerar os relatórios de teste para cada uma das técnicas de teste sorteadas através do nível e da perspectiva sorteadas.

*1x Plano de Teste completo - Todos artefatos estudados.*

*1x Relatório de Testes completo*

*Arquivos de Implementação de Teste de Software*

*Artigo Científico do Trabalho desenvolvido*





Perspectiva	Nível de Teste	Tipos de Teste mais comuns	Exemplo
<b>Caixa Branca</b>	Unidade	Estrutural, Regressão	Cobertura de código, verificação de condições e laços
<b>Caixa Branca</b>	Integração	Estrutural, Funcional	Testar chamadas entre módulos (contratos de interface)
<b>Caixa Branca</b>	Sistema	Desempenho, Estrutural	Identificar gargalos de código em execução
<b>Caixa Branca</b>	Aceitação	Pouco usual, mas possível em auditoria técnica	Cliente valida se a lógica interna atende padrões exigidos



<b>Caixa Preta</b>	Unidade	Funcional	Testar entradas/saídas de um módulo sem olhar o código
<b>Caixa Preta</b>	Integração	Funcional, Regressão	Verificar integração entre componentes de software
<b>Caixa Preta</b>	Sistema	Funcional, Não-funcional	Testar requisitos end-to-end, carga, usabilidade, segurança
<b>Caixa Preta</b>	Aceitação	Funcional, Não-funcional	Validação final pelo cliente: requisitos, desempenho mínimo



<b>Turma B</b>	<b>Integrantes</b>	<b>Perspectiva de Teste</b>	<b>Técnicas De Teste</b>	
<b>Grupo 1</b>	Julia Batistello; Beatriz L...	Caixa Preta	Aceitação	
<b>Grupo 2</b>	Bruno Bertolase; Lauro L	Caixa Branca	Unidade	
<b>Grupo 3</b>	Ana Clara Paim;Amanda L..	Caixa Branca	Unidade	
<b>Grupo 42</b>	Felipe Borreto; Vinicius R...	Caixa Preta	Unidade	
<b>Grupo 5</b>	Gabriel Sandrei; Matheus..	Caixa Branca	Integração	
<b>Grupo 6</b>	Joao Victor Marin; Gust...	Caixa Preta	Sistema	
<b>Grupo 7</b>	Lucca Rocha; Alexandre..	Caixa Branca	Sistema	
<b>Grupo 8</b>	Felipe Broetto Kaike...	Caixa Branca	Aceitação	
<b>Grupo 9</b>	thiago Nunes; Mateus Lo...	Caixa Preta	Integração	



## Níveis de Teste

- Teste de Unidade / Unitário
- Teste de Componentes
- Teste de Integração
- Teste de Sistema
- Teste de Regressão
- Teste de Aceitação

## Técnicas de Teste

- Teste Caixa-Branca / *White-Box* / Estrutural
- Teste Caixa-Preta / *Black-Box* / Funcional

## Tipos de Teste

- Teste de Usabilidade
- Teste de Desempenho / *Performance*
- Teste de Carga
- Teste de Estresse / Esforço
- Teste de Segurança





## Anteriormente em Teste de Software

- Visão de Técnicas de Teste de Software
- Testes Funcionais



# Estratégias de teste de software

## Teste de Unidade

- Teste Seletivo de caminhos de execução
- Teste de fronteira

## Teste de Integração

- Integração descendente (top-down)
- Integração ascendente (bottom-up)
- Teste de regressão
- Teste fumaça

## Teste de Validação ou Aceitação

- Testes Alfa
- Testes Beta

## Teste de Sistema

- Teste de recuperação
- Teste de segurança
- Teste por esforço
- Teste de desempenho
- Teste de disponibilização

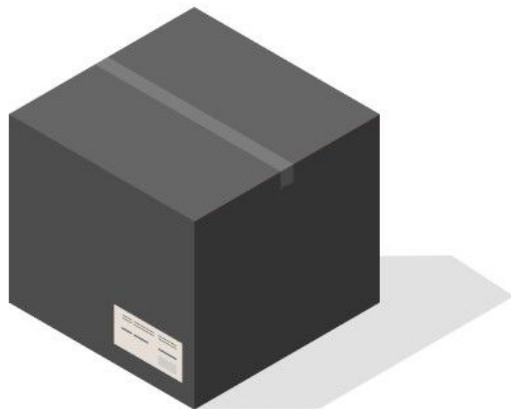
## 5. Visão interna e externa do teste

- Teste caixa-branca
- Teste caixa-preta

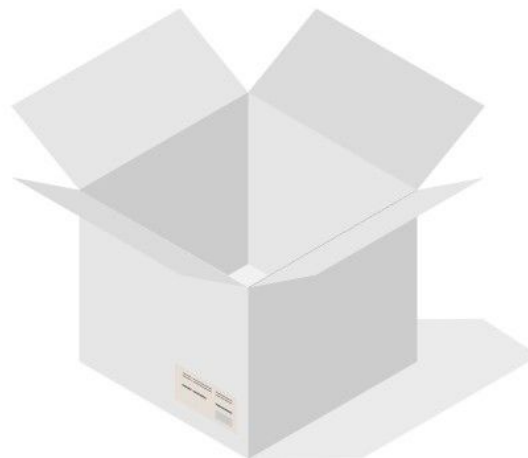


## Visão de Teste

Segundo Pressman (2011), qualquer produto de engenharia pode ser testado a partir de duas perspectivas diferentes:



**Black box - we do not  
know anything**



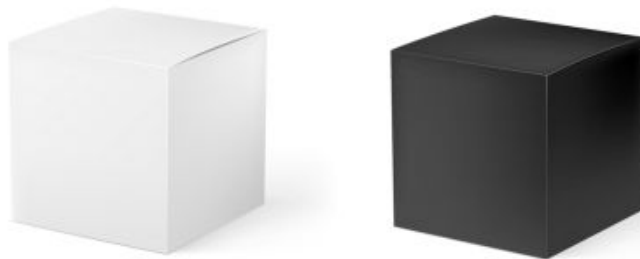
**White box - we know  
everything**



## Visão de Teste

(1) a lógica interna do programa é exercitada usando técnicas de projeto de caso de teste “caixa branca”;

(2) os requisitos de software são exercitados usando técnicas de projeto de casos de teste “caixa preta”.





## CAIXA PRETA

Verifica as operações e ações de uma aplicação.

É baseado nos requisitos do cliente.

Ajuda a melhorar o comportamento da aplicação.

O teste de caixa preta é fácil de executar manualmente.

Ele testa o que o produto faz.

O teste é baseado nos requisitos do negócio.

## CAIXA BRANCA

Verifica o comportamento de uma aplicação.

É baseado nas expectativas do cliente.

Ajuda a melhorar o desempenho da aplicação.

É difícil executar o teste de caixa branca manualmente.

Ele descreve como o produto se sai.

O teste é baseado nos requisitos de desempenho.

### Exemplos

1 - Teste de Unidade  
2 - Teste de Integração  
3 - Teste de Regressão

1 - Teste de Desempenho  
2 - Teste de Carga  
3 - Teste de Estresse

Aspecto	Caixa Branca	Caixa Preta
<b>Foco</b>	Estrutura interna, lógica e código	Funcionalidade e comportamento externo
<b>Objetivo</b>	Verificar a execução correta do código	Validar a conformidade com os requisitos
<b>Exemplos de Técnicas</b>	Cobertura de caminho, condição, loop	Testes de equivalência, valor limite
<b>Ferramentas</b>	JUnit, NUnit, PyTest	Selenium, QTP/UFT, Postman



## Teste Funcional

Teste FUNCIONAL ou CAIXA-PRETA focaliza os requisitos funcionais do sistema

- As técnicas de teste caixa-preta permitem derivar séries de condições de entrada que utilizarão completamente todos os requisitos funcionais para um programa
- O teste caixa-preta não é uma alternativa às técnicas caixa-branca.
- É uma abordagem complementar, com possibilidade de descobrir uma classe de erros diferente



# Teste Funcional

O teste caixa-preta tenta encontrar erros nas seguintes categorias:

- (1) **funções** incorretas ou faltando;
- (2) erros de **interface**;
- (3) erros em **estruturas de dados** ou **acesso** a **bases de dados externas**;
- (4) erros de **comportamento** ou de desempenho;
- (5) erros de **inicialização** e **término**



## Características do Teste Funcional

- O teste funcional tem foco no comportamento do sistema
- Os testes funcionais são projetados para avaliar o software a partir da perspectiva do usuário
- NÃO estão preocupados com a implementação interna, apenas com o comportamento externo do sistema
- Os testes funcionais normalmente envolvem a criação de cenários de uso realista
- Simular as ações dos usuários, para verificar se o software executa corretamente nessas situações





# FRAMEWORKS PARA TESTE FUNCIONAL

- JUnit (Java): Ele suporta testes de unidade, testes de integração e testes funcionais.
- TestNG (Java): TestNG é uma alternativa ao JUnit para testes de unidade e funcionais em Java.
- pytest (Python): Ele é fácil de usar e oferece recursos avançados de descoberta automática de testes e geração de relatórios.
- NUnit (C#): Ele oferece suporte a parametrização de testes e outras funcionalidades avançadas.





# ATIVIDADE TESTE FUNCIONAL





# FRAMEWORKS PARA TESTE FUNCIONAL

- Selenium (Web Applications): O Selenium é uma ferramenta popular para testar aplicativos da web. Ele permite a automação de testes de interface do usuário em navegadores



+

Se



# ATIVIDADE TESTE FUNCIONAL

## Atividade prática de teste funcional Passos:

1. Projetar **casos de teste funcionais** para avaliar as quatro plataformas de teste dos itens listados abaixo, com divisão de responsabilidades.
2. Preencher os ARTEFATOS de teste abaixo para os testes projetados.
3. Configurar ambiente de testes Selenium (<https://selenium-python.readthedocs.io/> ).
4. Implementar o **mesmo fluxo de teste** nas duas ferramentas.
5. Executar os testes funcionais projetados e implementados



# ATIVIDADE TESTE FUNCIONAL

## ITENS A SEREM TESTADOS:

### 1. SauceDemo – <https://www.saucedemo.com>

- o Feito para treino de automação de testes.
- o Usuário: standard\_user
- o Senha: secret\_sauce
- o Tem login, inventário, carrinho, etc.
- o É o mais usado em exemplos de Selenium.

### 2. TheInternet(Herokuapp) – <https://the-internet.herokuapp.com/login>

- o Login simples com validação de usuário/senha.
- o Usuário: tomsmith
- o Senha: SuperSecretPassword!

### 3. PracticeTestAutomation – <https://practicetestautomation.com/practice-test-login/>

- o Usuário: student
- o Senha: Password123
- o Login de demonstração com feedback claro de sucesso/erro.

### 4. OrangeHRMDemo – <https://opensource-demo.orangehrmlive.com/> o Sistema real de RH, mas em versão demo pública.

- o Usuário: Admin
- o Senha: admin123
- o Permite explorar login e fluxo interno mais robusto.



**PLANOS DE TESTE A SER DESCRITO :**

**ITENS A TESTAR / ABORDAGEM:**

Nº	Item	Especificação	ABORDAGEM:
1			
2			
3			

**CRONOGRAMA DE TESTES**

ID	Tarefa	Início	Fim	Esforço	Pré	Pessoa	Obs
01							
02							
03							
04							

**AMBIENTE DE TESTE**

Ambiente	Descrição
Hardware	
Software	
Ferramental	

**IDENTIFICAÇÃO DE CASO DE TESTE / IDENTIFICAÇÃO DE PROCEDIMENTO DE TESTE**

Nº	Caso de Teste	Identificação do Caso de Teste	Procedimento	Identificação do Procedimento de Teste
1				
2				
3				
4				
5				



#### CASO DE TESTE

Identificação		
Itens a Testar		
Entradas	Campo	Valor
Saídas Esperadas	Campo	Valor
Ambiente		
Procedimento		
Dependência		

#### PROCEDIMENTO DE TESTE

Identificação	
Objetivo	
Requisitos	
Fluxo	

#### PARTICIONAMENTO EM CLASSES DE EQUIVALÊNCIA

VARIÁVEIS DE ENTRADA	CLASSE DE EQUIVALÊNCIA VÁLIDA	CLASSE DE EQUIVALÊNCIA INVÁLIDA



**OBRIGADO**

[dacio.francisco@unicesumar.edu.br](mailto:dacio.francisco@unicesumar.edu.br)