

The image features a central white circle containing text. This circle is surrounded by two concentric rings: an inner dark teal ring and an outer light gray ring. Four thick lines, two dark teal and two light gray, intersect at the center, forming a cross shape that passes through the rings. The background is white, with four light gray triangular sectors positioned between the lines, creating a star-like pattern.

# Teste de sistemas

- Gustavo Boeing -

ANO

2025

# Testes de Caixa-Preta

## Definição

Teste de caixa-preta é uma técnica de teste de software onde se avalia a funcionalidade de um sistema sem conhecer sua estrutura interna ou código-fonte. O foco está nas entradas e saídas do sistema, verificando se ele se comporta conforme o esperado.

- ☐ Baseado em requisitos e especificações funcionais.
- ☐ Ignora a lógica interna, algoritmos e estruturas do código.
- ☐ Testa o que o sistema faz, não como ele faz.
- ☐ Muito utilizado para testar interfaces, validações de entrada, fluxo de dados e comportamento do sistema.

## Conceitos

1. Foco no comportamento externo – Verifica se o sistema funciona conforme os requisitos.
2. Independente do código – Não requer conhecimento da linguagem ou arquitetura interna.
3. Baseado em especificações – Utiliza documentação (requisitos, user stories, casos de uso).
4. Aplicável em diferentes níveis – Pode ser usado em testes unitários, de integração, sistema e aceitação.

## Exemplos

### 1. Particionamento de Equivalência

Divide as entradas em grupos válidos e inválidos, testando um representante de cada partição.

- Exemplo: Um campo que aceita idade (1 a 120 anos):
  - Partição válida: 25 (deve ser aceita).
  - Partição inválida: -5 ou 150 (deve ser rejeitada).

### 2. Análise de Valor Limite (BVA)

Testa os valores nos limites das partições.

- Exemplo: Um sistema que permite 1 a 10 usuários:
  - Testar com 0, 1, 10 e 11 usuários.

### 3. Tabela de Decisão

Usada para sistemas com combinações de entradas e regras complexas.

- Exemplo: Um sistema de login com:
  - Usuário válido + senha válida = Acesso permitido.
  - Usuário válido + senha inválida = Erro.
  - Usuário inválido + qualquer senha = Erro.

## Testes Funcionais

Objetivo: Verificar se o sistema funciona conforme os requisitos especificados, validando suas funcionalidades.

Características:

- Baseado em casos de uso e regras de negócio.
- Foco em entradas e saídas esperadas.
- Pode ser manual ou automatizado.

## Testes de Aceitação

Objetivo: Garantir que o sistema atenda às expectativas do usuário final e esteja pronto para produção.

Características:

- Realizado pelo cliente ou PO (Product Owner).
- Baseado em critérios de aceitação (ex.: user stories).
- Pode incluir UAT (User Acceptance Testing).

## Testes Exploratórios

Objetivo: Descobrir defeitos não previstos por meio de investigação livre e criativa.

Características:

- Não há roteiro fixo – baseado na experiência do testador.
- Foco em cenários não óbvios e erros de usabilidade.
- Combina teste e aprendizado simultaneamente.

# Testes de Caixa-Cinza

## Definição

É uma abordagem híbrida que combina elementos dos testes de caixa-preta (sem conhecimento interno) e caixa-branca (com conhecimento do código). O testador tem acesso parcial à estrutura interna (como diagramas de arquitetura ou SQL) mas foca no comportamento externo.

## Conceitos

- Conhecimento limitado do sistema: Acesso a documentação, esquemas de banco de dados ou APIs, mas não ao código-fonte completo.
- Objetivo: Validar funcionalidades considerando fluxos internos sem analisar linha por linha.
- Técnicas comuns:
  - Teste baseado em modelos (ex.: diagramas de estados).
  - Análise de requisições/respostas de APIs.
  - Testes de banco de dados (verificar dados após uma operação).

## Exemplos

1. Teste de um sistema de login:
  - Caixa-preta: Verifica apenas se usuário/senha corretos permitem acesso.
  - Caixa-cinza: Analisa se, após o login, o sistema registra a data de acesso no banco de dados.
2. E-commerce:
  - Verificar se um desconto aplicado em um produto reflete no banco de dados e no carrinho de compras.
3. API REST:
  - Validar se uma requisição POST /users cria um registro no banco com os campos corretos.

## Quando Usar?

- ✓ Quando é necessário validar integração entre componentes.
- ✓ Para testes de segurança (ex.: SQL injection, onde o comportamento do banco é relevante).
- ✓ Em sistemas complexos onde o teste de caixa-preta puro é insuficiente.

## Teste de Regressão

Definição:

Testes realizados para garantir que alterações no código (novas funcionalidades ou correções) não introduziram bugs em funcionalidades existentes.

Conceitos-Chave:

- Foco em estabilidade: Assegurar que o que funcionava antes continua funcionando.
- Pode ser automatizado: Testes de regressão são frequentemente automatizados (ex.: com Selenium, JUnit).
- Tipos comuns:
  - Regressão completa: Testa todo o sistema após uma mudança.
  - Regressão parcial: Foca em áreas impactadas pela alteração.

Exemplos:

1. Correção de um bug em um formulário:
  - Antes da correção: Campo "CEP" não aceitava hífen.
  - Após a correção: Testar se o CEP com hífen funciona e se outros campos (como nome e e-mail) ainda estão válidos.
2. Atualização de uma API:
  - Verificar se endpoints modificados não quebraram integrações com outros sistemas.

# Teste de Cobertura

## Conceitos-Chave

### a) Cobertura Baseada em Interfaces

- Foca em APIs, bancos de dados e logs para inferir a execução interna.
- Exemplo: Monitorar queries SQL para garantir que operações CRUD sejam testadas.

### b) Cobertura de Estados

- Verifica se os testes cobrem transições entre estados (ex: "Pedido criado" → "Pago" → "Enviado").

### c) Cobertura de Caminhos

- Valida se diferentes fluxos lógicos são testados (ex: caminhos felizes (*happy paths*) e de erro).

### d) Cobertura de Dados

- Garante que os testes manipulam dados em diferentes formatos (ex: JSON válido/inválido em APIs).

## Exemplos Práticos

### Exemplo 1: API de Pagamento

- Cobertura de Interface:
  - Testar respostas HTTP (200, 400, 500) para diferentes entradas.
- Cobertura de Dados:
  - Verificar se um pagamento bem-sucedido registra a transação no banco.
- Cobertura de Estados:
  - Validar se o status do pedido muda de "Processando" para "Concluído".

### Exemplo 2: Sistema de Login

- Cobertura de Caminhos:
  - Testar credenciais válidas, senhas incorretas e campos vazios.
- Cobertura de Dados:
  - Checar se tentativas de login são logadas com timestamps.

# Testes de Caixa-Branca

## Definição

Também chamados de testes estruturais, avaliam o comportamento interno do software, analisando o código-fonte, fluxos lógicos, estruturas de dados e algoritmos.

## Conceito

- Acesso total ao código: O testador conhece a implementação.
- Foco em:
  - Cobertura de código (linhas, branches, condições).
  - Fluxos de controle (if-else, loops).
  - Tratamento de exceções.
- Técnicas comuns:
  - Análise de caminhos básicos.
  - Teste de condições e decisões.
  - Cobertura de código (ex.: 100% dos branches).

## Exemplos

1. **Verificação de um loop for:**
  - Testar se um loop que processa uma lista de 10 itens executa exatamente 10 vezes.
2. **Condicional em um sistema de login:**
  - Validar se o código trata corretamente um if (senha.length() < 8) retornando erro.
3. **Cobertura de exceções:**

- Garantir que um bloco try-catch capture um erro de divisão por zero.

### Quando Usar?

- ✓ Durante o desenvolvimento (para devs e QA).
- ✓ Em componentes críticos (ex.: algoritmos de criptografia).
- ✓ Para cumprir normas de segurança (ex.: PCI-DSS).

## Testes de Unidade

Definição:

Testam unidades individuais do código (funções, métodos, classes) de forma isolada.

Conceitos-Chave:

- Escopo pequeno: Uma função/método por vez.
- Ferramentas: Frameworks como JUnit (Java), pytest (Python), NUnit (.NET).
- Práticas:
  - Mocking: Simular dependências (ex.: banco de dados).
  - TDD (Test-Driven Development): Escrever testes antes do código.

Exemplos:

#### 1. Função de soma

python

```
def soma(a, b):
```

```
    return a + b
```

*# Teste:*

```
assert soma(2, 3) == 5
```

### Quando Usar?

- ✓ Durante a codificação (feito pelos devs).
- ✓ Para evitar regressões em refatorações.



## Testes de Integração

Definição:

Verificam a interação entre módulos ou serviços (ex.: API + banco de dados).

Conceitos-Chave:

- Foco em interfaces: Comunicação entre componentes.
- Tipos:
  - Top-down: Testa módulos de alto nível primeiro.
  - Bottom-up: Começa pelos módulos base.
  - Big Bang: Todos os módulos integrados de uma vez.
- Ferramentas: Postman (APIs), Selenium (UI), Docker (containers).

Exemplos:

1. Integração API + Banco de Dados:
  - Testar se um POST /users cria um registro no banco.
2. Microserviços:
  - Verificar se o serviço de pagamento chama corretamente o serviço de estoque.
3. Frontend + Backend:
  - Validar se uma página web exibe dados buscados de uma API.

### Quando Usar?

- ✓ Após testes de unidade.
- ✓ Em sistemas distribuídos (ex.: microsserviços).

# Comparativos

## Comparativo entre os Tipos

Critério	Caixa-Branca	Teste de Unidade	Teste de Integração
<b>Escopo</b>	Código interno.	Funções/métodos isolados.	Módulos/serviços combinados.
<b>Responsável</b>	Devs/QAs.	Devs.	QAs/Devs.
<b>Ferramentas</b>	SonarQube, Coverity.	JUnit, pytest.	Postman, Selenium.
<b>Exemplo</b>	Cobertura de if-else.	Testar calculaImposto().	Validar login → banco de dados.

## Resumo dos 3 Tipos de Testes de Software

### 1. Testes de Caixa-Branca (White-Box)

- Foco: Análise interna do código-fonte
- O que testa: Estrutura lógica, fluxos, condições e caminhos de execução
- Quando usar: Durante o desenvolvimento, para componentes críticos
- Exemplo: Verificar se todos os branches de um if-else são executados

### 2. Testes de Caixa-Preta (Black-Box)

- Foco: Funcionalidades externas sem conhecimento interno
- O que testa: Entradas e saídas conforme requisitos
- Quando usar: Validação de sistemas completos ou por usuários finais
- Exemplo: Testar um formulário de login com dados válidos e inválidos

### 3. Testes de Caixa-Cinza (Gray-Box)

- Foco: Combinação de abordagens interna e externa

- O que testa: Comportamento do sistema com conhecimento parcial da estrutura
- Quando usar: Testes de integração e validação de APIs/banco de dados
- Exemplo: Testar uma API verificando tanto as respostas quanto os registros no banco

#### Comparação Rápida:

- Caixa-Branca: "Como funciona" (código)
- Caixa-Preta: "O que faz" (funcionalidades)
- Caixa-Cinza: "Como se comporta" (sistema em operação)

Esses três tipos se complementam para garantir qualidade em diferentes níveis do desenvolvimento de software.