



# Técnicas de Teste



**Certified  
Developer**

The Ultimate Tech Degree

**DigitalHouse** >  
Coding School



# Índice

1

Categorias de técnicas de teste

2

Técnicas de Caixa Preta

3

Técnicas de Caixa Branca

4

Técnicas baseadas na experiência



**1**

# **Categorias de técnicas de teste**



“

O objetivo de uma técnica de teste é  
**ajudar a identificar as condições,  
os casos e os dados de teste.**

”





## Escolha de uma técnica de teste

A escolha da técnica de teste a ser utilizada depende de alguns fatores:

- Tipo e complexidade do componente ou sistema.
- Classes e níveis de risco.
- Objetivo do teste.
- Documentação disponível.
- Conhecimentos e competências do testador.
- Modelo do ciclo de vida do software.
- Tempo e orçamento.



# Classificação das técnicas de teste



## **Técnicas de caixa preta:**

São baseados no comportamento extraído da análise dos documentos que constituem a base de teste (documentos de requisitos formais, casos de uso, etc.). São aplicáveis tanto para testes funcionais como para não funcionais. Se concentram nas entradas e saídas sem levar em conta a estrutura interna.



## **Técnicas de caixa branca:**

São baseados na estrutura extraída dos documentos de arquitetura, projeto detalhado, estrutura interna ou código do sistema. Se concentram no processamento dentro do objeto de teste.



## **Técnicas baseadas na experiência:**

Aproveitam o conhecimento dos desenvolvedores, testadores e usuários para projetar, implementar e executar os testes.

## **2 | Técnicas de Caixa Preta**



## Partição de equivalência

Nesta técnica, os dados de entrada são divididos em partições conhecidas como **classe de equivalência**, onde cada elemento dessas classes ou partições é processado da mesma maneira. As características desta técnica são:

- A “partição de equivalência **válida**” contém valores que são aceitos pelo componente ou sistema.
- A “partição de equivalência **inválida**” contém valores que são rejeitados pelo componente ou sistema.
- Pode-se dividir as partições em subpartições.
- Cada valor pertence a apenas uma partição de equivalência.
- As partições de equivalência inválidas devem ser testadas individualmente para evitar o mascaramento de falhas.
- A cobertura é medida da seguinte forma:

$$\text{Cobertura} = \frac{\text{Partições testadas}}{\text{Partições existentes}}$$



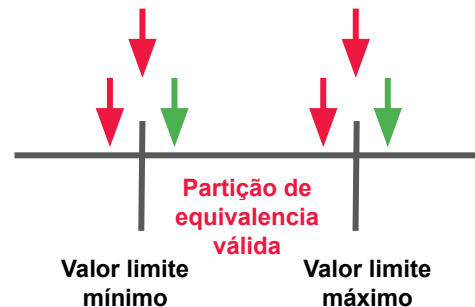


## Análise de valores limites

É uma extensão da técnica de partição de equivalência, é uma técnica utilizada em conjunto com a técnica de partição de equivalência, e essas partições são compostas por dados numéricos ou sequenciais.

- São identificados os valores limites mínimo e máximo (ou valores inicial e final).
- Pode-se utilizar 2 ou 3 valores limites:
  - Para 2 valores limites, toma-se o valor que marca o limite (como valor que corresponde à partição válida), e o valor anterior ou posterior que corresponde à partição de equivalência inválida.
  - Para 3 valores limites, toma-se o valor que marca o limite, um valor anterior e outro depois desse limite.
- A cobertura é medida da seguinte forma:

$$\text{Cobertura} = \frac{\text{Valores limites testados}}{\text{Valores limites identificados}}$$



# Tabela de decisão

Esta técnica é utilizada para testes combinatórios, compostos por regras de negócio complexas que um sistema deve implementar. As características desta técnica são:

- Deve-se identificar as condições (entradas) e as ações resultantes (saídas). Estas constituem as linhas da tabela.
- As colunas da tabela correspondem às regras de decisão. Cada coluna forma uma combinação única de condições e a execução de ações associadas a essa regra.
- Os valores das condições e ações podem ser valores booleanos, discretos, numéricos ou intervalos de números.
- Ajuda a identificar todas as combinações importantes de condições e a encontrar qualquer incompatibilidade de requisitos.
- A cobertura é medida da seguinte forma:

$$\text{Cobertura} = \frac{\text{Número de regras de decisão testadas}}{\text{Número total de regras de decisão}}$$

Condiciones	Reglas							
Condición 1	S	S	S	S	N	N	N	N
Condición 2	S	S	N	N	S	S	N	N
Condición 3	S	N	S	N	S	N	S	N
Acción 1	X	X						
Acción 2				X		X		X
Acción 3			X				X	
Acción 4					X			

**3**

## **Técnicas de Caixa Branca**



## Técnicas de Teste Caixa Branca

Nesta técnica, os aspectos de implementação são fundamentais na escolha dos casos de teste, pois os testes desta técnica é baseado na estrutura interna do sistema.

A desvantagem desta técnica é que ela não analisa se a especificação está correta, ela se concentra apenas no código e não verifica a lógica da especificação.

Para utilizar esta técnica é importante termos em mente a definição de caminho. Um caminho é uma sequência de execução de comandos que se inicia em um ponto de entrada e termina em um ponto de saída do sistema em teste.

Não é viável testar todos os caminhos lógicos de um programa, então contamos com algumas técnicas para nos ajudar.



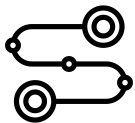


## Técnicas de Teste Caixa Branca



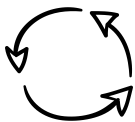
### **Teste de Caminho Básico:**

São gerados casos de teste para executar cada instrução pelo menos uma vez.



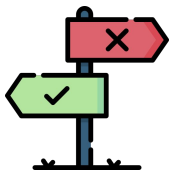
### **Teste de Condição:**

São gerados casos de teste focados nas condições booleanas do código, onde o teste valida os valores positivos e falsos para garantir que cada condição não contém erros.



### **Teste de Ciclo:**

São gerados casos de teste focados na validação da construção de ciclo.



### **Teste de Decisão:**

São gerados casos de teste focados nas decisões do código, onde o teste valida as decisões positivas e falsas de cada comando de decisão para garantir que cada decisão não contém erros.

**4**

**Técnicas baseadas na  
experiência**



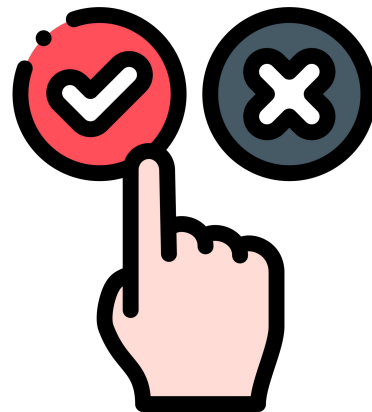
## Previsão de erros

Esta técnica é utilizada para antecipar a ocorrência de erros, defeitos e falhas com base no conhecimento do testador.

Uma lista é criada considerando:

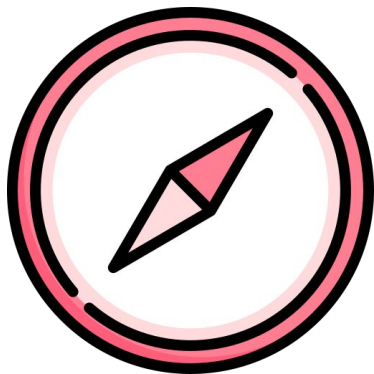
- Como a aplicação funcionou no passado.
- Erros comuns dos desenvolvedores.
- Falhas nas aplicações relacionadas.

Com base nessa lista, os testes são projetados para expor essas falhas e defeitos.





## Teste exploratório



Nesta técnica, os testes informais são **projetados, executados, registrados e avaliados** dinamicamente durante a execução do teste.

Os resultados destes testes são utilizados para aprender mais sobre o funcionamento do componente ou sistema.

Geralmente, são utilizados para complementar outras técnicas formais ou quando as especificações são escassas, inadequadas ou limitadas pelo tempo.





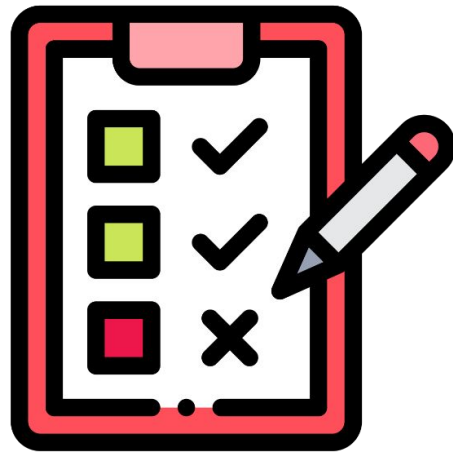
## Teste baseado em listas de verificação

Nesta técnica, são projetados, implementados e executados os casos de teste que cobrem as condições que estão em uma lista de verificação definida.

As listas são criadas com base na experiência e no conhecimento do que o testador acredita ser importante para o usuário e são usadas devido à falta de casos de teste detalhados.

Durante a execução, pode haver alguma variabilidade, dependendo de quem está executando o teste e das condições de contexto. Isso resulta em uma maior cobertura.

É utilizado em testes funcionais e em testes não funcionais.



DigitalHouse>  
Coding School