



Caixa branca e Caixa preta



**Certified
Developer**
The Ultimate Tech Degree

DigitalHouse >
Coding School



Temas

1

Caixa branca

2

Caixa preta

3

**Caixa branca vs.
caixa preta**



1 | Caixa branca



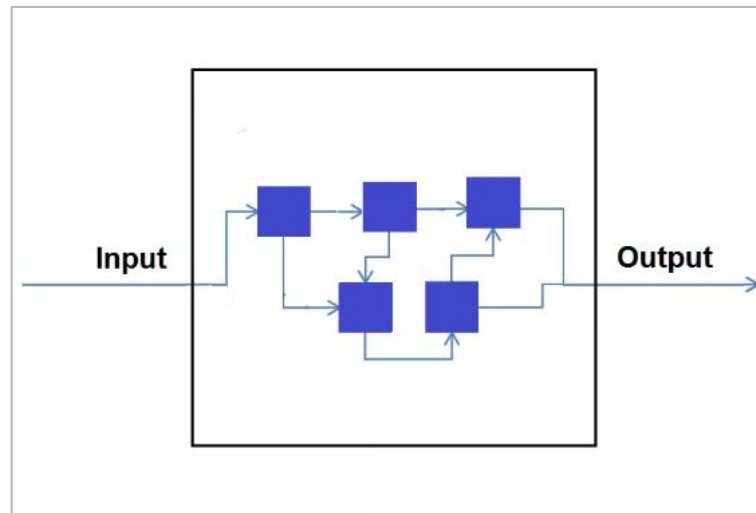
O que é caixa branca?

Nós o usamos para testar a estrutura interna, design e codificação de uma solução de software; o código é visível para o testador e se concentra principalmente em verificar o fluxo de entradas e saídas por meio do aplicativo, melhorando o design e a usabilidade, além de fortalecer a segurança. O teste de caixa branca também é conhecido como teste de caixa clara, teste de caixa aberta, teste estrutural, teste baseado em código e teste de caixa de vidro. Geralmente é feito por desenvolvedores.





Ele também é composto de uma série de camadas, cuja essência é o teste cuidadoso do aplicativo no nível do código-fonte para reduzir bugs ocultos posteriormente.





Etapas para a implementação da caixa branca

Existem duas etapas básicas que os testadores realizam ao testar um aplicativo usando a técnica de teste da caixa branca:

1) Entender o código fonte

O testador deve ter total conhecimento do código do software e da lógica usada. É recomendável que você esteja ciente das práticas de codificação seguras e, portanto, protegendo seu código de hackers.



Etapas para a implementação da caixa branca

2) Crie casos de teste e execute-os

O teste de caixa branca envolve o teste do código-fonte do aplicativo quanto ao fluxo e estrutura adequados. Uma maneira é escrever mais código para testar o código-fonte do aplicativo e desenvolver pequenos testes para cada processo do aplicativo. Esse método requer que o testador tenha um conhecimento profundo do código, como já mencionamos. Outros métodos incluem teste manual, teste de tentativa e erro e o uso de ferramentas de teste.



Ferramentas para testar com caixa branca

- Parasoft Jtest
- PyUnit
- EclEmma
- HTMLUnit
- NUnit
- CppUnit



3 níveis de teste de caixa branca

O teste unitário garante que o código funcione conforme o planejado antes que ocorra a integração com o código testado anteriormente.

Integration testing as interações das interfaces entre si são testadas, garantindo que o código funcionará adequadamente no ambiente de integração do sistema.

Regression testing as caixas usadas nos níveis anteriores são recicladas.

2 | Caixa preta



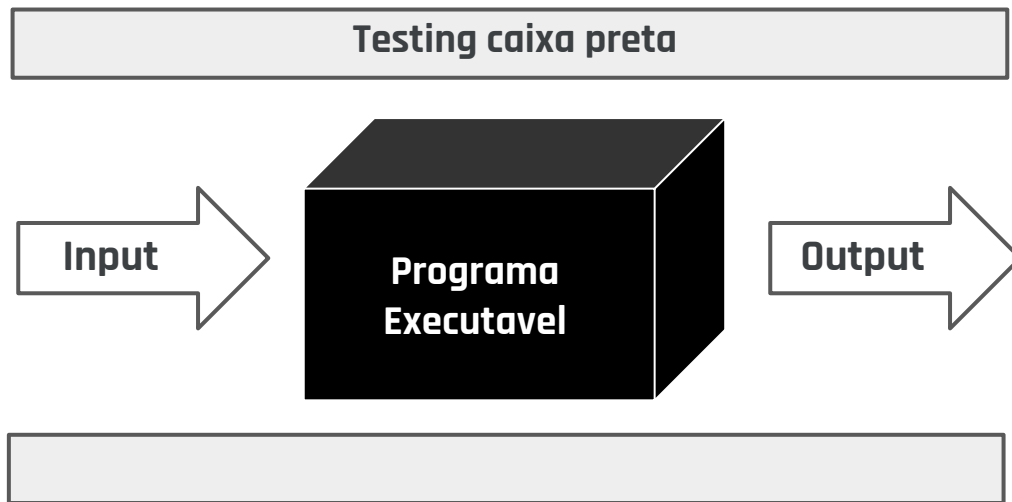
O que é caixa preta?

Neste modelo de teste, testamos a funcionalidade do aplicativo sem olhar para a estrutura do código interno e nos concentramos apenas nas entradas e saídas do sistema de software. Este tipo de teste é totalmente baseado nas especificações e requisitos do software. Também é conhecido como teste comportamental, caixa opaca, caixa fechada, baseado em especificações ou olho no olho.





Este método de teste também é conhecido como teste comportamental e teste funcional e é crítico durante os estágios do ciclo de vida do teste de software, como regressão, aceitação, unidade, sistema, integração e teste de desenvolvimento de software.





Passos para implementação de caixa preta

1. Reunindo os requisitos e especificações do software, os testadores verificam se há entradas positivas e negativas no software e as detectam.
2. Teste com casos de teste selecionados.
3. Todos os casos de teste são executados.
4. Os testadores verificam todas as saídas para as entradas fornecidas.
5. Defeitos, se houver, são reparados e retestados.



Teste funcional e não funcional com caixa preta

Teste funcional

O teste funcional lida com os requisitos ou especificações funcionais de um aplicativo. Aqui, diferentes funções do sistema serão testadas fornecendo a entrada e comparando a saída real com a saída esperada.

Teste não funcional

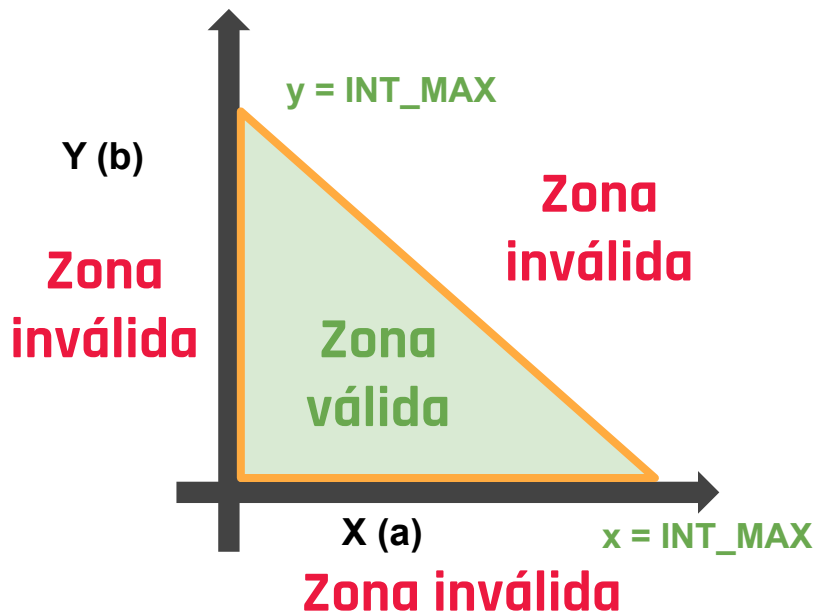
Os testes não funcionais tratam da qualidade e do desempenho do aplicativo.



Técnicas de testes em caixa preta

1) Partição de equivalência

Os valores de entrada para o sistema ou aplicativo são divididos em diferentes classes ou grupos com base em sua semelhança no resultado. Portanto, em vez de usar todos os valores de entrada, agora podemos usar qualquer valor da classe para testar o resultado, para que possamos manter a cobertura do teste enquanto podemos reduzir muito retrabalho e, o mais importante, o tempo investido.





2) Análise de valor limite

Verifica o sistema nos limites onde a saída do sistema muda. Por exemplo: se um sistema tem uma taxa de transferência de 50 segundos, como será o desempenho em 49 ou 51 segundos, e como muda sua saída?

Inválido	Válido	Inválido	Inválido
0	1 10	11 99	100
Parte 1	Parte 2	Parte 3	Parte 4

A análise de valor limite verificará valores como 0, 1, 10, 11, 99, 100.



3) Teste de mesa de decisão

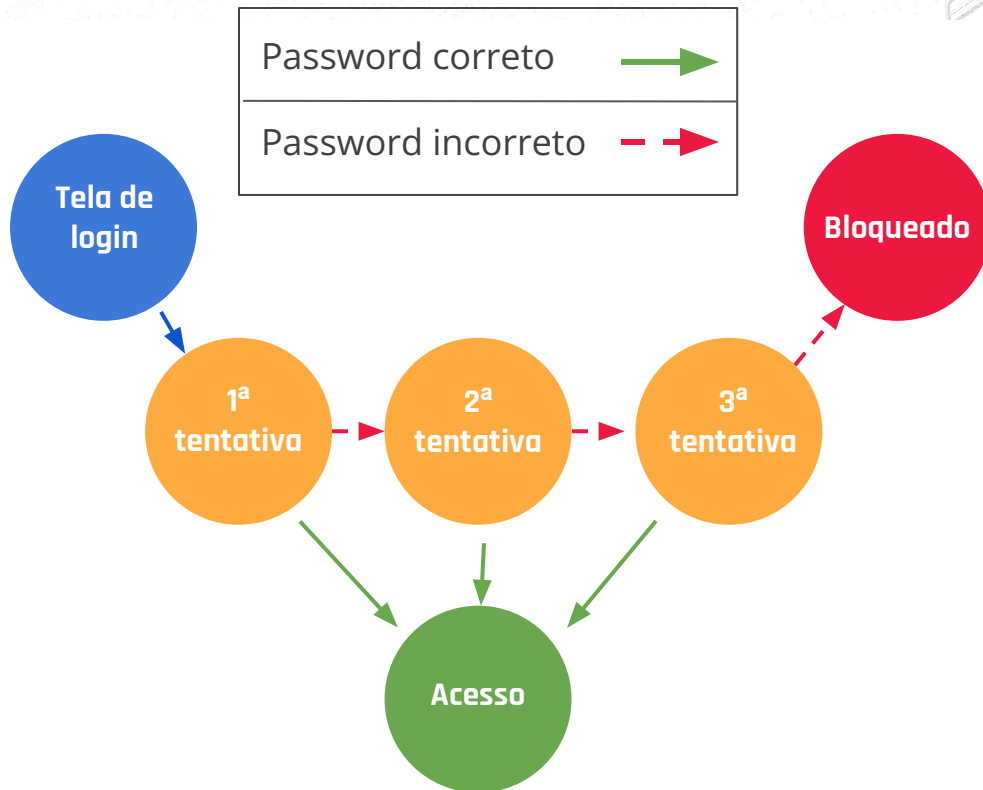
Atua para qualquer sistema no qual você tenha que tomar uma decisão de acordo com uma condição particular, como if-else ou switch case. Exemplo: se no caso de alteração temos uma declaração para uma opção, e outra declaração para outra opção.

	Regra 1	Regra 2	Regra 3	Regra 4
Condição				
Fim de mês	Não	Sim	Sim	Sim
Salario transferido	N/A	Não	Sim	Sim
Fundo de previdência	N/A	N/A	Não	Sim
Ação				
Imposto de renda	Não	Não	Sim	Sim
Fundo de previdência	Não	Não	Não	Sim



4) Teste de transição dos estados

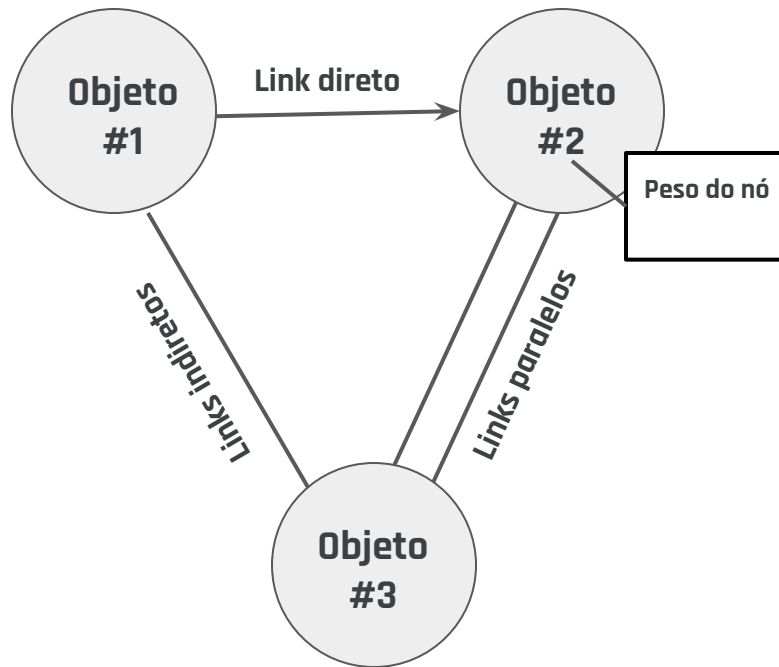
Quando um sistema tem muitos passos (como um site da Web: página inicial, página de login, página de logout, etc.), temos que verificar como o sistema fornece saídas para todas as entradas.





5) Métodos de teste baseados em gráficos

Cada aplicativo é um acúmulo de alguns objetos. Todos esses objetos são identificados e o gráfico é preparado. A partir desse gráfico de objeto, cada relacionamento de objeto é identificado e os casos de teste são gravados de acordo para descobrir erros.





6) Error guessing

O teste é realizado de acordo com a experiência do testador. Todos os erros comuns que os desenvolvedores normalmente cometem são testados e a codificação de exceção é testada.

7) Teste de comparação

Diferentes versões independentes do mesmo software são usadas para comparação entre si.





Tipos de testes de caixa preta

- **Teste funcional:** este tipo de teste é útil para os testadores identificarem os requisitos funcionais de um software ou sistema.
- **Teste de regressão:** Este tipo de teste é executado após o procedimento de manutenção do sistema, atualizações ou correções de código para descobrir o impacto do novo código no código antigo.
- **Testes não funcionais:** esse tipo de teste não está relacionado ao teste de nenhuma funcionalidade específica, mas sim a parâmetros não funcionais, como usabilidade, escalabilidade e desempenho.



3 | Caixa branca vs. caixa preta

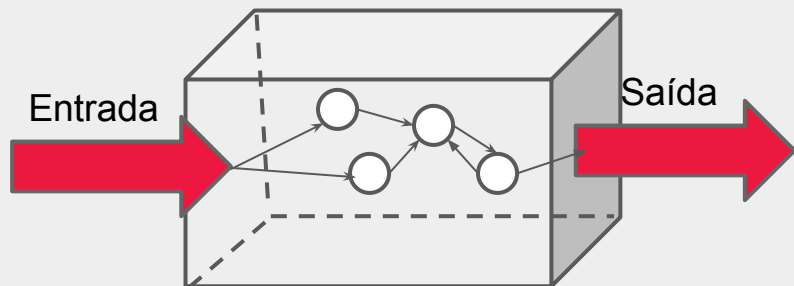


Caixa branca Vs. Caixa preta

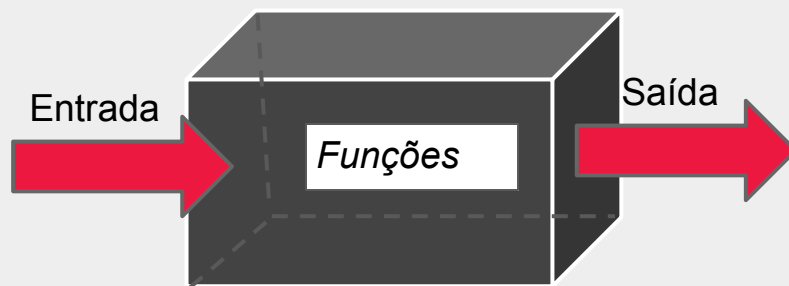
CAIXA PRETA	CAIXA BRANCA
Não há necessidade de conhecer a estrutura interna do software	Há necessidade de conhecer a estrutura interna do software
É feito pelos testadores	É feito pelos programadores
Nenhum conhecimento de programação necessário	É necessário conhecimento de programação
Requer conhecimento de implementação	Não requer conhecimento de implementação
É um teste de alto nível	É um teste de baixo nível
Consome pouco tempo	Consome bastante tempo



Caixa branca



Caixa preta



DigitalHouse>
Coding School