

[◀ VOLTAR](#)

# Testes de Software - Testes de Aferição e Homologação

Estudar os conceitos sobre os testes de aceitação, técnicas de teste caixa-preta, definição de teste alfa, teste beta e planilhas de aferição/homologação.

## NESTE TÓPICO

- › Teste de Aceitação
- › Técnicas de Teste
- › Particionamento de Equivalência
- › Análise de Valores-limites
- › Teste Alfa



## Teste de Aceitação

Antes que o sistema seja entregue ao cliente para uso, esse passa por um processo de aceite, onde ocorrem aferições (conferências) e homologações (aprovações) das funcionalidades dos produtos de software. Esse processo normalmente é referenciado como “Teste de Aceitação” ou “Teste de Aceite”.

Segundo Koscianski e Soares (2007), o propósito do teste de aceitação é o de “avaliar a qualidade externa do produto e, na medida do possível, também a qualidade de uso”. A qualidade externa do software tem uma visão do produto como uma caixa-preta, ou seja, não se vê sua arquitetura, código ou qualquer funcionamento interno. Já a qualidade em uso corresponde a algumas características sob o ponto de vista do usuário, em contextos como eficácia, produtividade, segurança e satisfação.

Esses testes só devem ocorrer depois que outros testes foram aplicados ao sistema, corrigindo e ajustando o que foi necessário até então. A homologação é a comprovação, pelo usuário habilitado a tal tarefa, de que o produto resultante do projeto de software atende aos critérios de aceite previamente estabelecidos com o cliente.

Teste de Aceite: teste formal relacionado às necessidades dos usuários, requisitos e processos de negócios. É realizado para estabelecer se um sistema satisfaz ou não os critérios de aceitação e para possibilitar aos usuários, aos clientes e às outras entidades autorizadas decidir aceitar ou não determinado sistema.

((IEEE, 1990))

Os principais objetivos do teste de aceitação são:

- Atestar se o software funciona da forma esperada pelo cliente no ambiente proposto;
- Identificar oportunidades de correção e aplicar os ajustes correlatos à aplicação;
- Obter o aceite do produto de software pelo cliente;
- Registrar sugestões para as próximas versões da aplicação.

O objetivo do teste de aceite é estabelecer a confiança no sistema, parte do sistema ou uma característica não específica do sistema. Procurar defeitos não é o principal foco do teste de aceite. Teste de aceite pode avaliar a disponibilidade do sistema para entrar em produção, apesar de não ser necessariamente o último nível de teste. Por exemplo, teste de integração em larga escala pode vir após o teste de aceite de um sistema.

((ISTQB, 2011))





Koscianski e Soares (2007) ressaltam a possibilidade de utilização, ainda, do Teste em Paralelo para apoio à decisão sobre a homologação do sistema. Esse teste é passível de utilização quando um sistema é desenvolvido para substituir outro, já em funcionamento. Colocam-se os dois sistemas para operar (antigo e novo) em paralelo, verificando-se o comportamento de ambos. Resultados importantes, como por exemplo, valores apurados em um processo contábil podem ser utilizados para verificar a acurácia do sistema novo.



Teste de Aceitação de Usuário.

Algumas importantes técnicas de teste de caixa-preta podem ser empregadas no teste de aceitação, uma vez que o foco do teste é verificar os requisitos funcionais do software, analisando se o comportamento do sistema está de acordo com o estabelecido nos requisitos.



## Técnicas de Teste

Quando os testes funcionais são criados, determinadas considerações devem ser feitas, como a quantidade de dados que devem ser fornecidos para que o teste seja executado com rapidez e abrangência. A execução de testes com um grande volume de dados de entrada não significa que o teste foi conduzido de forma melhor do que um teste que tenha sido realizado com uma quantidade inferior de dados. A qualidade das entradas é o mais importante.

Um caso de teste ideal é aquele que, somente ele, é capaz de apontar erros que só poderiam ser descobertos caso fossem executados muitos casos de teste (PRESSMAN, 2011). Vamos estudar uma importante técnica para criação de casos de teste.

## Particionamento de Equivalência

Existem algumas técnicas para agrupar e otimizar casos de teste, cujo objetivo é dar a maior cobertura possível no teste do sistema. Uma delas é a técnica do particionamento de equivalência. Consiste em dividir a entrada de dados para o teste em classes de dados válidos e inválidos. Veja o quadro abaixo:

Entrada	Valores permitidos	Classes	Casos de testes
Idade ( este valor é obtido através da digitação da data de aniversário)	Número entre 18 e 120	18 e 120	idade = 20
		< 18	idade = 10
		> 120	idade = 150

Particionamento de Equivalência

No quadro acima, temos um exemplo de como os casos de teste devem ser elaborados. A criação de uma classe válida e duas classes inválidas cobre as possibilidades de teste para o atributo idade. A execução do teste com essas classes pode identificar se a funcionalidade está correta ou se possui algum tipo de erro. No exemplo citado, espera-se que a idade gerada como 20 seja aceita e as demais (10 e 150), rejeitadas. Qualquer resultado diferente desses esperados é indicativo de erro.

### Passos do Método

#### 1) Definir as classes de equivalência

##### 1.1) Definir as chamadas condições de entrada

*Condição de entrada: Uma condição de entrada é qualquer intervalo de entradas válidas de um programa.*

Tipos comuns de condições de entrega:

##### a) Faixa de valores.

Ex.:  $1 < \text{ITEM} < 1.000$ .

##### b) A entrada envolve um certo número de valores.

Ex: Um vetor de até 6 elementos.

##### c) Conjunto de Valores.

Ex: Conjunto dos veículos motorizados = {caminhão, carro, ônibus}.

##### d) Condição booleana.

Ex: O primeiro caractere de um identificador deve ser uma letra.

##### 1.2) Definir as classes de equivalência propriamente ditas, que podem ser válidas ou inválidas. Estas classes são definidas a partir da condição de entrada.



Exemplos:**a)**  $1 < \text{ITEM} < 1000$ .

Classe válida:  $1 < \text{ITEM} < 1000$ .

Classes inválidas:  $\text{ITEM} \leq 1$  e  $\text{ITEM} \geq 1000$ .

**b)** Vetor de até 6 elementos.

Classe válida: vetor com 4 elementos

Classes inválidas: Vetor com zero elemento e Vetor com 7 elementos.

**c)** Conjunto de valores.

Classe válida: {caminhão, carro, ônibus}.

Classe inválida: {carroça}

**d)** Condição

Classe válida: O primeiro caractere do identificador é uma letra.

Classe inválida: O primeiro caractere do identificador não é uma letra.

## Análise de Valores-limite

Técnica que complementa a técnica de particionamento de equivalência. A intenção é selecionar valores que se encontram nos extremos que devem ser validados, ou seja, leva à seleção de casos de teste nas extremidades da classe.



Entrada	Valores permitidos	Classes	Casos de testes
Idade ( este valor é obtido através da digitação da data de aniversário)	Número entre 18 e 120	18 e 120	idade = 18 idade = 120
		< 18	idade = 17
		> 120	idade = 121

### Análise de valores-limite

Convém notar que os valores que são gerados encontram-se próximos aos valores que são os limites dos valores permitidos (18 e 120). Neste caso, estão sendo criados dois valores que verificam a validade dos limites impostos e que devem ser aceitos (18 e 120). E, também, um valor imediatamente abaixo do limite inferior (17) e um valor imediatamente acima do limite superior (121), que se espera que sejam rejeitados.

Vejamos agora duas abordagens possíveis a serem realizadas para o teste de aceitação: os testes denominados alfa e beta.

## Teste Alfa

É realizado imediatamente após o término do desenvolvimento do sistema. No momento em que o sistema se torna plenamente operacional, os usuários passam a testá-lo em ambiente apropriado (geralmente ambiente de homologação), que está sob a supervisão da equipe de desenvolvimento. Esse tipo de teste busca erros que podem ter sido negligenciados nos planos de teste, cenários e casos de teste elaborados. É o primeiro contato do cliente/usuário com o produto.

## Teste Beta

Trata-se de um tipo de teste de aceitação. É conduzido pelo cliente/usuário, sem a presença do desenvolvedor, no ambiente do cliente visando identificar possíveis erros resultantes da configuração. O cliente registra os problemas (reais ou imaginários) que são encontrados e relata-os ao desenvolvedor a intervalos regulares.



Teste Alfa / Teste Beta

## Avaliação de Produtos de Software

De acordo com a norma ISO 14598, o processo de avaliação de um software é definido como:

“Operação técnica que consiste em efetuar o julgamento das características de um produto de software através de procedimentos previamente definidos”. (Norma ISO 14598 – parte 5).

Ainda, em concordância com essa norma, um processo de avaliação deve possuir quatro características básicas. Essas características indicam que o processo deve ser:

- Repetível: deve ser capaz de ser reutilizado.
- Reprodutível: dadas as mesmas condições, os resultados obtidos devem ser iguais.
- Imparcial: não deve ser dependente de pessoas, grupos ou entidades.
- Objetivo: deve produzir resultados claros.

## Homologação de Produtos de Software

Em se tratando de homologar um software, a “autoridade” é o cliente. No caso de software aplicativo, desenvolvido para ser utilizado internamente na empresa ou de produto criado especificamente para um cliente externo, o cliente é responsável por atestar que o produto está apto a ser utilizado no ambiente de produção. Ou seja, a liberação do produto está atrelada à satisfação do cliente em relação aos requisitos, desempenho, usabilidade e outros elementos que determinam se o software está ou não em conformidade com as suas expectativas. Essa aprovação deve ser formal, onde normalmente é entregue um documento para que o usuário (cliente) assine um termo de aceite, o qual ratifica que o sistema foi validado e aprovado. Abaixo você poderá conferir um exemplo de documento conhecido como termo de aceite.

### MATERIAL COMPLEMENTAR

(<https://img.uninove.br/static/0/0/0/0/0/0/0/2/6/4/0/264087/15264.Pdf>)



Quando um software é desenvolvido para ser distribuído em larga escala, ou seja, é um produto que será vendido para uma aplicação de forma genérica (Por exemplo, um sistema de contabilidade para empresas), a homologação deve ficar a cargo de empresas que são especializadas em testes de software. Isso garante que exista, principalmente, imparcialidade no processo.

Geralmente, existem planilhas que são elaboradas, segundo os critérios das empresas, para verificar o andamento do processo de homologação. Como os critérios podem ser diversos, não existe um modelo padrão a ser seguido. É importante, contudo, que esses critérios sejam objetivos, práticos e que reflitam, de forma inequívoca, os resultados obtidos, permitindo, assim, a correta avaliação do produto.

## Quiz

Exercício Final



INICIAR ➤

## Referências

BARTIÉ, A. *Garantia da qualidade de software*. Rio de Janeiro: Campus, 2002.

IEEE STANDARDS ASSOCIATION. *Standard Glossary of Software Engineering Terminology*. [S.l.]: Std 610.12-1990, 1990.

ISTQB - INTERNATIONAL SOFTWARE TESTING QUALIFICATIONS BOARD. *Certified Tester Foundation Level Syllabus*. [S.l.]: ISTQB, 2011.

KOSCIANSKI, A.; SOARES, M. D. S. *Qualidade de software*: aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software. 2. ed. São Paulo: Novatec, 2007.

PRESSMAN, R. S. *Engenharia de Software*: Uma Abordagem Profissional. 7. ed. Porto Alegre: McGraw Hill, 2011.

SOMMERVILLE, I. *Engenharia de Software*. 8. ed. São Paulo: Pearson Addison-Wesley, 2007.



Avalie este tópico



ANTERIOR

Testes de Software - Planos de teste, Cenários,  
Casos de Teste Biblioteca



Índice

(<https://www.uninove.br/conheca->

a-

[uninove/biblioteca/sobre-](https://www.uninove.br/biblioteca/sobre-)

a-

[biblioteca/apresentacao/](https://www.uninove.br/biblioteca/apresentacao/))

Portal Uninove

(<http://www.uninove.br>)

Mapa do Site

Ajuda?

(<https://ava.uninove.br/ava.php?idCurso=>)

® Todos os direitos reservados

