



Níveis de Teste



**Certified
Developer**
The Ultimate Tech Degree

DigitalHouse >
Coding School



Índice

1

Teste de
Componente

2

Teste de
Integração

3

Teste de Sistema

4

Teste de Aceitação





“

Uma parte de um software é mais que linhas de código. É geralmente um sistema complexo multi-camada, incorporando componentes funcionais separados e integrações de terceiros. Sendo assim, testes de software eficientes devem ir muito além de apenas encontrar erros no código.

”





Os **níveis de teste** se caracterizam pelos seguintes atributos:

- Objetivos específicos
- Bases de teste
- Objeto de teste
- Defeitos e falhas característicos
- Foco e responsabilidades específicas

1 | Teste de Componente



Objetivos específicos

- Reduzir o risco.
- Verificar se os comportamentos funcionais e não funcionais do **componente** estão de acordo com os projetados e especificados.
- Gerar confiança na qualidade do componente.
- Encontrar defeitos no componente.
- Prevenir a propagação de defeitos para níveis de teste superiores.





Base de teste

Alguns exemplos de produtos de trabalho que podem ser usados como base de teste incluem:

- Projeto detalhado.
- Código.
- Modelo de dados.
- Especificações dos componentes.



Objetos de teste

Os objetos de teste característicos para o teste de componente incluem:

- Componentes, unidades ou módulos.
- Código e estruturas de dados.
- Classes.
- Módulos de banco de dados.





Defeitos e falhas característicos do teste

Exemplos de defeitos e falhas característicos do teste de componente incluem:

- Funcionamento incorreto — por exemplo: não funciona conforme descrito nas especificações do projeto.
- Problemas de fluxo de dados.
- Código e lógica incorretos.





Foco e responsabilidades específicas

Em geral, o desenvolvedor que escreveu o código realiza o teste de componente. Os desenvolvedores podem alternar o desenvolvimento de componentes com a localização e correção de defeitos.

Eles geralmente escrevem e executam testes depois de escreverem o código de um componente. No entanto, especialmente no desenvolvimento ágil, a escrita de casos de teste de componentes automatizados pode preceder a escrita do código da aplicação.



2 | Teste de Integração



Objetivos específicos

O teste de integração se concentra nas interações entre componentes ou sistemas.

- Reduzir o risco.
- Verificar se os comportamentos funcionais e não funcionais das **interfaces** estão de acordo com os projetados e especificados.
- Gerar confiança na qualidade das interfaces.
- Encontrar defeitos — que podem estar nas próprias interfaces ou dentro dos componentes ou sistemas.
- Prevenir a propagação de defeitos para níveis de teste superiores.





Base de teste

Alguns exemplos de produtos de trabalho que podem ser usados como base de teste para teste de integração incluem:

- Projeto de software e sistemas.
- Diagrama de sequência.
- Especificações de interface e protocolos de comunicação.
- Casos de uso.
- Arquitetura a nível de componente ou sistema.
- Fluxos de trabalho.
- Definições de interface externa.



Objetos de teste

Os objetos de teste característicos para o teste de integração incluem:

- Subsistemas.
- Base de dados.
- Infraestrutura.
- Interfaces.
- Interfaces de programação de aplicações — API por sua sigla em inglês.
- Microserviços.





Defeitos e falhas característicos

- Dados incorretos, dados ausentes ou codificação incorreta de dados
- Sequenciamento ou sincronização incorreta de chamadas para interface.
- Incompatibilidade da interface.
- Falhas na comunicação entre os componentes.
- Falhas de comunicação entre componentes não tratados ou tratados incorretamente.
- Suposições incorretas sobre o significado, das unidades ou dos limites dos dados que são transmitidos entre os componentes.



Foco e responsabilidades específicas

O teste de integração deve se concentrar na própria integração. Pode-se utilizar os tipos de teste funcional, não funcional e estrutural. Em geral, é responsabilidade dos testadores.



3

Teste de Sistema



Objetivos específicos

- Reduzir o risco.
- Verificar se os comportamentos funcionais e não funcionais do **sistema** estão de acordo com os projetados e especificados.
- Validar se o sistema está completo e funcionará conforme o esperado.
- Gerar confiança na qualidade do sistema considerando o como um todo.
- Encontrar defeitos.
- Prevenir a propagação de defeitos para níveis de teste superiores ou para produção.





Base de teste

Alguns exemplos de produtos de trabalho que podem ser usados como base de teste incluem:

- Especificações de requisitos do sistema e do software - funcionais e não funcionais.
- Relatórios de análise de risco.
- Casos de uso.
- História de usuários.
- Modelo de comportamento do sistema.
- Manuais do sistema e do usuário.



Objetos de teste

- Aplicações.
- Sistemas de hardware/software.
- Sistemas operacionais.
- Sistema sujeito ao teste.
- Configuração do sistema e dados de configuração.





Defeitos e falhas característicos

- Cálculos incorretos.
- Comportamento funcional ou não funcional incorreto ou inesperado do sistema.
- Controle e/ou fluxo de dados incorretos dentro do sistema.
- Incapacidade de realizar, de forma adequada e completa, tarefas funcionais de ponta a ponta.
- Falha do sistema em operar corretamente no(s) ambiente(s) de produção.
- Falha do sistema para funcionar conforme descrito nos manuais do sistema e do usuário.



Foco e responsabilidades específicas

O teste do sistema deve se concentrar no comportamento geral e de ponta a ponta do sistema como um todo, tanto funcional quanto não funcional.

Deve-se utilizar as técnicas mais adequadas para os aspectos do sistema a serem testados.

Os testadores independentes, geralmente, realizam o teste do sistema.



4 | Teste de Aceitação



Objetivos específicos

O teste de aceitação, como o teste de sistema, geralmente se concentra no comportamento e nos recursos de todo um sistema ou produto. Os objetivos do teste de aceitação incluem:

- Estabelecer confiança na qualidade do sistema como um todo.
- Validar se o sistema está completo e funcionará conforme o esperado.
- Verificar se os comportamentos funcionais e não funcionais do sistema são os especificados.





Base de teste

Exemplos de produtos de trabalho que podem ser utilizados como base de teste incluem:

- Processos de negócio.
- Requisitos de usuário ou de negócio.
- Regulamentos, contratos legais e normas.
- Casos de uso.
- Requisitos do sistema.
- Documentação do sistema ou do usuário.
- Procedimentos de instalação.
- Relatórios de análise de risco.



Objetos de teste

- Sistema sujeito ao teste.
- Configuração do sistema e dados de configuração.
- Processos de negócio para um sistema totalmente integrado.
- Sistemas de recuperação e sites críticos - para testes de continuidade do negócio e recuperação de desastres.
- Processos operacionais e de manutenção.
- Formulários.
- Dados de produção existentes e transformados.



Defeitos e falhas característicos

Exemplos de defeitos característicos de qualquer forma de teste de aceitação incluem:

- Os fluxos de trabalho do sistema não atendem aos requisitos de negócios ou de usuário.
- As regras de negócios não são implementadas corretamente.
- O sistema não atende aos requisitos contratuais ou regulamentares.
- Falhas não funcionais, como vulnerabilidades de segurança, eficiência de desempenho inadequado sob altas cargas ou funcionamento inadequado em uma plataforma compatível.



Foco e responsabilidades específicas

Frequentemente, é responsabilidade dos clientes, usuários de negócio, proprietários de produtos ou operadores de um sistema, e outras partes interessadas também podem estar envolvidas.

O teste de aceitação é geralmente visto como o último nível de teste em um ciclo de vida de desenvolvimento sequencial.



DigitalHouse>
Coding School