

# espaçamento

## Contents

---

---

Oi este é o post de teste para balas

### 1. Verificação 1

### 2. VERIFICAÇÃO 2

### 3. Tipos de testes ad-hoc

- Teste do amigo: Dois membros da equipe (geralmente um desenvolvedor e um testador) colaboram para testar um módulo específico juntos, oferecendo perspectivas diferentes.
- Teste de pares: Dois testadores trabalham juntos na mesma máquina para revisar e testar o aplicativo, gerando ideias espontâneas.
- Teste de macacos: Fortemente desestruturada, essa abordagem envolve a entrada de dados aleatórios ou a execução de ações aleatórias para ver como o software se comporta e garantir que não falhe.



### 4. MARCADOR APÓS IMAGENS

Vantagens e desvantagens

Pros

- Economia de tempo: Ignora o demorado processo de escrever casos de teste formais, tornando-o ideal para prazos apertados.
- Complementa o teste formal: Encontra bugs mais profundos ou obscuros que não foram encontrados nos conjuntos de testes estruturados.
- Flexibilidade: Adapta-se perfeitamente quando os requisitos estão incompletos ou mudando.

#### 1. Técnicas Usadas em Testes Adhoc

As técnicas de teste adhoc são métodos informais usados pelos testadores para identificar rapidamente defeitos sem casos de teste predefinidos. Essas técnicas dependem de experiência,

intuição e exploração aleatória do aplicativo.

- Adivinhação de erros: com base na experiência do testador para prever áreas propensas a defeitos, concentrando-se em erros comuns e módulos arriscados.
- Teste de entrada aleatório: usar dados inesperados ou inválidos para verificar o comportamento do sistema e garantir que o aplicativo manipule entradas incomuns corretamente.
- Teste de limite (Informal): Teste de valores de entrada mínimos e máximos para identificar erros nas bordas dos limites de entrada.
- Abordagem de testes exploratórios: aprender e testar a aplicação simultaneamente sem casos de teste predefinidos para detectar defeitos dinamicamente.
- Teste de macaco: Executando ações aleatórias sem lógica para verificar a estabilidade do sistema e identificar falhas.
- Teste baseado em sessão: realização de testes em sessões curtas e focadas para cobrir módulos específicos de forma eficiente.
- Teste negativo: uso de entradas incorretas ou inválidas para verificar se o sistema trata os erros e as validações corretamente.
- Teste de cenário do usuário: simular o comportamento real do usuário para garantir que o aplicativo funcione corretamente em situações reais e para identificar problemas de usabilidade

## Exemplo: Aplicativo Móvel

Um testador executa ações aleatórias em um aplicativo móvel:

- Alternar rapidamente entre telas
- Gire o dispositivo com frequência
- Usar o aplicativo com baixa conexão com a Internet
- Abra vários recursos de uma só vez

Isso pode revelar problemas de desempenho, falhas de interface do usuário ou falhas de aplicativo.

## Condições ideais para testes específicos

Mais eficaz quando o tempo é limitado, os testadores têm um forte conhecimento do domínio e são necessárias verificações exploratórias rápidas para descobrir defeitos ocultos.

- Quando o tempo de teste é limitado.
- Após a conclusão do teste formal.
- Quando a validação rápida de um recurso ou módulo é necessária.
- Quando não houver casos de teste detalhados disponíveis.
- Quando o desenvolvimento estiver quase completo.
- Ao testar áreas de alto risco ou propensas a erros.

## Limitações dos testes ad hoc

Não possui documentação e estrutura, o que dificulta a reprodução de defeitos e garante a cobertura completa do teste.

- Não há documentação adequada, o que dificulta o rastreamento de defeitos posteriormente.
- A cobertura do teste não é garantida devido à sua natureza não estruturada.
- Depende muito da experiência e do conhecimento do testador.
- Os problemas encontrados podem ser difíceis de reproduzir.
- Não pode substituir métodos de teste formais e sistemáticos.

## Vantagens dos testes ad hoc

Ajuda a identificar rapidamente defeitos inesperados sem casos de teste formais, economizando tempo e melhorando a cobertura geral do teste.

- Identifica bugs que não podem ser encontrados por meio de casos de teste escritos.
- Pode ser realizado em pouco tempo, especialmente quando os prazos são apertados.
- Incentiva o pensamento criativo e ajuda a gerar novos cenários de teste.
- Melhora a qualidade do produto ao revelar problemas inesperados.
- Pode ser realizado em qualquer estágio do [SDLC \(Software Development Life Cycle Process\)](#).

## Práticas recomendadas para realização de testes ad hoc

- Tenha um conhecimento sólido do produto: Entenda claramente o aplicativo e sua funcionalidade.
- Identificar Áreas Propensas A Erros: Concentre-se nos módulos onde é provável que ocorram defeitos.
- Priorizar Recursos Críticos: Teste primeiro as áreas importantes e de alto risco.
- Plano em um nível básico: Tenha uma ideia aproximada do que testar, mesmo sem casos de teste formais.
- Use as ferramentas apropriadas: Utilize ferramentas de depuração e teste para melhorar a eficácia.

1. <https://www.youtube.com/>
2. <https://fdk-stage.cisco.com/c/en/us/support/docs/licensing/ask-licensing/cda/device-management/lic219871-this-is-the-test-article-to-check-the.html>
3. <https://www.geeksforgeeks.org/software-engineering/adhoc-testing-in-software/>

## Sobre esta tradução

A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês ([link fornecido](#)) seja sempre consultado.