**Maneiras inteligentes de escrever casos de teste — técnicas que todo QA deve conhecer**

Quando comecei em QA, pensei que escrever casos de teste significava apenas listar etapas e verificar se as coisas funcionavam.

Clique no botão  
Digite o nome  
Enviar formulário

Simples, certo?

Mas com o tempo — depois de alguns bugs de produção e uma pilha crescente de casos de teste repetitivos — percebi: ***não se trata de quantos casos de teste você escreve, mas de quão inteligentemente você os projeta****.*

Então, aqui estão algumas técnicas de design de casos de teste que me ajudaram a subir de nível. Se você é um engenheiro de QA ou um SDET, elas facilitarão sua vida (e melhorarão sua cobertura de testes).

**1. Particionamento de Equivalência**

Pense nisso como agrupar entradas semelhantes.

Digamos que um campo aceite números de 1 a 100. Você não precisa testar todos os 100 números. Basta escolher:

* Uma entrada válida (digamos, 50)
* Uma entrada inválida abaixo do intervalo (0)
* Uma entrada inválida acima do intervalo (101)

Pronto. Você cobriu todos os "tipos" de entradas sem perder tempo.

**Quando usar:** Você está lidando com campos de entrada, validação de formulário ou APIs.

**2. Análise de Valor Limite**

É aqui que as coisas geralmente quebram — nas bordas.

Se um sistema permite valores de 10 a 100, teste:

* Logo abaixo: 9
* Limite inferior: 10
* Limite superior: 100
* Logo acima: 101

Você ficaria surpreso com a frequência com que os insetos se escondem nos cantos.

**Quando usar:** Há um intervalo numérico, limite de tamanho de arquivo ou contagem de caracteres envolvidos.

**3. Teste de Tabela de Decisão**

Se houver *muitas condições* , faça uma tabela.

Imaginar:

* Se o usuário estiver logado
* Se o item estiver em estoque
* Se for fim de semana

Cada combinação pode levar a um resultado diferente (por exemplo, aplicar desconto ou não). Em vez de chutar, organize-os em uma tabela e aborde cada cenário importante.

**Quando usar:** As regras ou lógicas de negócios dependem de várias condições.

**4. Teste de transição de estado**

Esta é ótima para sistemas que se comportam de maneira diferente com base no estado atual.

Pense em fluxos de login/logout, status de pedidos ou fluxos de trabalho. Você testa como o sistema reage ao passar de um estado para outro.

Por exemplo:

* Não logado → login → logado
* Conectado → sair → não conectado

**Quando usar:** O aplicativo se comporta de maneira diferente com base no que aconteceu antes.

**5. Adivinhação de erro**

Às vezes, os melhores casos de teste vêm da experiência.

Você sabe onde as coisas *geralmente* dão errado — como caracteres especiais, falhas de rede ou dados ausentes.

Você não encontrará isso na documentação, mas eles geralmente expõem bugs ocultos.

**Quando usar:** Você está curioso ou cético em relação ao sistema.

***Projetar casos de teste é muito mais do que apenas seguir instruções****. Trata-se de ser estratégico e testar os cenários certos. As técnicas compartilhadas aqui me ajudaram a me concentrar em escrever menos casos de teste, porém mais eficazes — e evitar os temidos*testes repetitivos e redundantes*. Se você é novo em QA, ou mesmo um profissional experiente, recomendo que incorpore essas técnicas ao seu fluxo de trabalho. Teste de forma mais inteligente, não com mais esforço.*

Se você achou este artigo útil, sinta-se à vontade para me seguir para obter mais dicas, insights e tutoriais sobre testes, automação e tudo relacionado ao controle de qualidade.