ALINE FRANÇA

Mentoria Seja Tester Academy – Aula 8

Análise e Modelagem de Teste – Parte 01

ALINE FRANÇA

Sumário

Quais assuntos vamos aprender nessa aula?

ALINE FRANÇA

VISÃO GERAL DAS TÉCNICAS DE TESTE

As técnicas de teste são maneiras que ajudam quem testa um sistema a decidir o que deve ser testado e como fazer esses testes. Elas ajudam a criar uma quantidade pequena, mas suficiente, de testes de forma organizada e inteligente.

Essas técnicas também servem para definir o que exatamente precisa ser verificado, quais partes do sistema serão cobertas pelos testes, e quais informações ou dados serão usados nos testes. Se você quiser se aprofundar, existem padrões e livros que falam mais sobre isso (como o ISO/IEC/IEEE 29119-4, e autores como Beizer, Craig, Copeland, entre outros).

ALINE FRANÇA

VISÃO GERAL DAS TÉCNICAS DE TESTE

As técnicas de teste são divididas em três tipos principais:

1. **Técnicas de Teste Caixa-Preta (também chamadas de testes baseados em especificações):** Aqui, a pessoa que testa olha o sistema como um “caixa fechada”, ou seja, ela testa o que o sistema deve fazer, sem se preocupar com como ele foi construído por dentro.

Por exemplo: se você estiver testando um micro-ondas, você verifica se ele esquenta a comida ao apertar o botão, sem abrir o aparelho para ver os fios ou circuitos.

Esses testes continuam válidos mesmo que o sistema mude por dentro, desde que o comportamento do lado de fora continue o mesmo.

ALINE FRANÇA

VISÃO GERAL DAS TÉCNICAS DE TESTE

2. Técnicas de Teste Caixa-Branca (também chamadas de testes baseados na estrutura): Aqui, é o contrário: o testador analisa o que está acontecendo dentro do sistema, como ele foi programado. É como abrir o motor de um carro para ver se as peças estão funcionando do jeito certo. Esses testes só podem ser feitos depois que o sistema já foi criado ou pelo menos desenhado.

3. Técnicas de Teste Baseadas na Experiência: Essas técnicas usam a vivência e o “feeling” da pessoa que testa. Quem já tem experiência consegue imaginar onde podem surgir problemas e faz testes baseados nisso.

Esses testes são muito bons para encontrar erros que talvez não aparecessem nos testes caixa-preta ou caixa-branca. Por isso, são considerados um complemento importante aos outros dois tipos.

ALINE FRANÇA

TÉCNICAS DE TESTE CAIXA-PRETA

As técnicas de teste caixa-preta são aquelas em que a pessoa testa o sistema sem precisar saber como ele foi feito por dentro, ela só se importa com o que o sistema deve fazer.

Algumas formas comuns de fazer esse tipo de teste são:

- Particionamento de equivalência: É como dividir os possíveis resultados ou entradas em grupos que se comportam do mesmo jeito. Em vez de testar tudo, você testa um exemplo de cada grupo, o que já costuma ser suficiente.
- Análise de valor de limite: Aqui, o foco é testar os valores mais extremos, ou seja, os que ficam bem no começo ou bem no fim do que é permitido. Esses pontos costumam ser onde os erros aparecem com mais frequência.

ALINE FRANÇA

TÉCNICAS DE TESTE CAIXA-PRETA

Algumas formas comuns de fazer esse tipo de teste são:

- Teste de tabela de decisão: Quando existem muitas regras e combinações possíveis, essa técnica organiza tudo numa tabela para garantir que todos os cenários foram considerados e testados

- Teste de transição de estado: Essa técnica serve para sistemas que mudam de comportamento dependendo do que acontece. Por exemplo, como um caixa eletrônico que muda de tela quando você insere o cartão. O teste verifica se essas mudanças de “estado” estão funcionando corretamente.

ALINE FRANÇA

PARTICIONAMENTO DE EQUIVALÊNCIA

O **Particionamento de Equivalência** é uma técnica que ajuda a testar um sistema de forma mais inteligente, sem precisar testar absolutamente todos os casos possíveis.

A ideia é a seguinte: Você divide os dados (ou valores que o sistema pode receber) em grupos. Cada grupo é chamado de partição de equivalência. Todos os valores dentro de um mesmo grupo devem se comportar da mesma maneira no sistema. Então, em vez de testar todos os valores daquele grupo, você testa apenas um valor de cada grupo. Se houver um erro, ele provavelmente vai acontecer com qualquer outro valor do mesmo grupo. Isso economiza tempo e esforço nos testes.

ALINE FRANÇA

PARTICIONAMENTO DE EQUIVALÊNCIA

Exemplo simples: Imagine um campo onde o usuário precisa digitar a idade. Vamos supor que o sistema só aceita idades de 18 a 60 anos. Você pode criar três partições (grupos):

- Valores válidos: de 18 a 60
- Valores inválidos abaixo do permitido: como 10, 17
- Valores inválidos acima do permitido: como 61, 80

Em vez de testar todas as idades, você escolhe um valor de cada grupo (por exemplo, 30, 15 e 65), isso já cobre bem a lógica.

Entrada	Valores Permitidos	Classes	Casos de Teste
Idade	Idade entre 18 e 120	18 a 120	Idade = 20
		< 18	Idade = 10
		> 120	Idade = 150

ALINE FRANÇA

PARTICIONAMENTO DE EQUIVALÊNCIA

Partições válidas e inválidas: Um grupo com valores que o sistema deve aceitar é chamado de partição válida. Um grupo com valores que o sistema não deve aceitar é uma partição inválida. O que é considerado válido ou inválido pode variar de empresa para empresa. Às vezes, valores inválidos devem ser ignorados, e outras vezes devem gerar uma mensagem de erro, por exemplo.

Entrada	Valores Permitidos	Classes	Casos de Teste
Idade	Idade entre 18 e 120	18 a 120	Idade = 20
		< 18	Idade = 10
		> 120	Idade = 150

ALINE FRANÇA

PARTICIONAMENTO DE EQUIVALÊNCIA

Como saber se testou o suficiente? Cada partição que você identificar precisa ser testada pelo menos uma vez. Se você fizer isso, estará cobrindo 100% do que a técnica pede. A “cobertura” é medida pela quantidade de grupos testados em comparação com o total de grupos existentes e isso é mostrado como uma porcentagem.

E se o sistema tiver mais de um dado de entrada? Muitos sistemas pedem mais de uma informação ao mesmo tempo, como idade, senha, nome, etc. Nesse caso, os testes vão considerar os grupos de cada uma dessas informações. E o básico é garantir que cada grupo de cada dado seja testado ao menos uma vez, mesmo que não se testem todas as combinações possíveis entre eles.

Esse jeito básico de cobrir cada grupo pelo menos uma vez se chama Each Choice Coverage (ECC).

ALINE FRANÇA

PARTICIONAMENTO DE EQUIVALÊNCIA

RESUMO

O Particionamento de Equivalência ajuda a testar melhor, com menos esforço, dividindo os dados em grupos parecidos. Basta testar um valor de cada grupo, cobrindo os possíveis tipos de resposta que o sistema pode dar, tanto para dados certos quanto para dados errados.

ALINE FRANÇA

ANÁLISE DE VALOR DE LIMITE (BVA)

A Análise de Valor de Limite (ou BVA, do inglês Boundary Value Analysis) é uma técnica usada para testar programas de computador, focando nos “limites”, ou seja, nos valores bem no começo ou bem no final de uma faixa permitida.

Imagine que um programa aceita apenas números entre 1 e 100. A BVA vai testar especialmente os números que estão bem perto desses limites, como 1, 2, 99 e 100, porque é muito comum os programadores cometerem erros justamente nesses pontos extremos.

Essa técnica só funciona quando os valores têm uma ordem, como números, por exemplo. Se 10 e 20 estão no mesmo grupo, a BVA considera que todos os números entre eles (como 11, 12, 13...) também fazem parte do mesmo grupo.

ALINE FRANÇA

ANÁLISE DE VALOR DE LIMITE (BVA)

Erros comuns que a BVA ajuda a encontrar são aqueles em que o programa aceita ou rejeita valores que estão só um pouquinho fora do limite, por exemplo: aceitar 101 quando só deveria aceitar até 100, ou rejeitar 1 quando deveria aceitar desde 1.Existem duas formas principais de usar a BVA:

- **1. BVA de 2 valores:** Aqui, para cada limite, a gente testa:
- O valor do limite (ex: 100)
- E o número logo ao lado que está fora da faixa permitida (ex: 101)

Usando esse método, a gente garante que todos os limites foram testados com dois números: o limite em si e o vizinho que está na faixa errada. A gente mede a cobertura (ou seja, o quanto testamos) contando quantos limites testamos, comparando com o total de limites que identificamos.

ALINE FRANÇA

ANÁLISE DE VALOR DE LIMITE (BVA)

2. BVA de 3 valores: Esse método é um pouco mais completo. Aqui, para cada limite, a gente testa:

- O valor do limite (ex: 100)
- O número logo antes dele (ex: 99)
- E o número logo depois dele (ex: 101)

Nesse caso, além de testar os próprios limites, a gente também testa os valores vizinhos, para ter ainda mais certeza de que o sistema está funcionando direitinho.

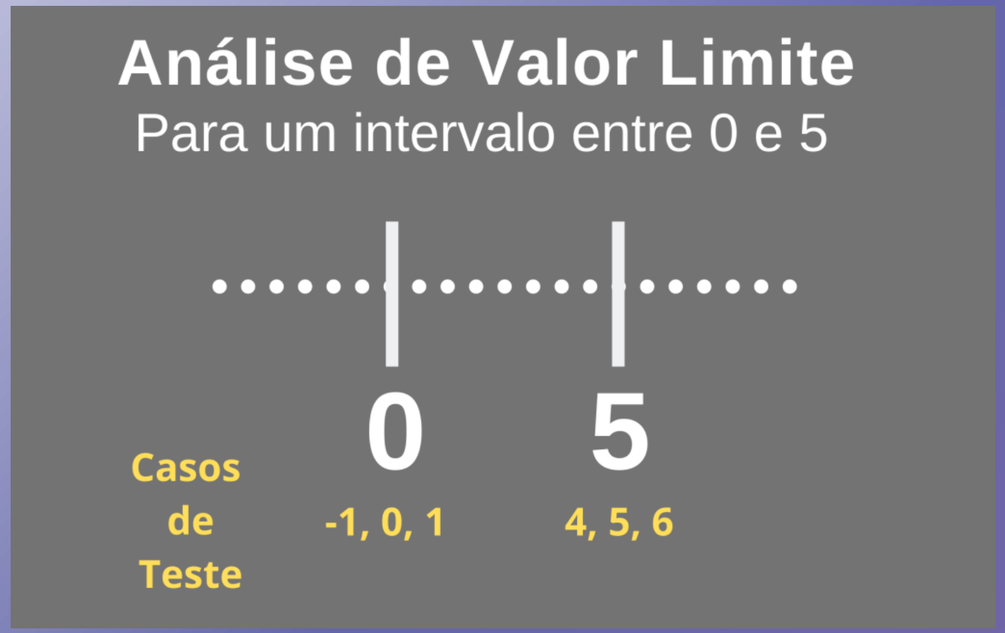
Esse método consegue encontrar erros que o de 2 valores pode não pegar. Por exemplo: imagine que o programa só deveria aceitar números até 10, mas por um erro ele só aceita exatamente 10. Se a gente só testar 10 e 11 (como no método de 2 valores), pode parecer que está tudo certo. Mas se testarmos também o 9 (como no de 3 valores), vamos perceber que 9 não está sendo aceito, e isso está errado.

ALINE FRANÇA

ANÁLISE DE VALOR DE LIMITE (BVA)

RESUMO

A BVA é como testar os "cantinhos" de uma regra, porque é aí que os erros mais acontecem. Quanto mais vizinhos a gente testa, mais chances temos de encontrar falhas escondidas.



ALINE FRANÇA

TESTE DE TABELA DE DECISÃO

As tabelas de decisão são uma ferramenta usada para ajudar a testar sistemas (como sites, aplicativos ou softwares) que têm regras para decidir o que fazer em diferentes situações.

Como assim “regras”?

Pense assim: um sistema pode ter várias condições (como "o cliente tem conta?", "o pagamento foi aprovado?", etc.) e, dependendo das respostas (sim ou não), o sistema toma uma ação (como "liberar o produto", "mostrar uma mensagem de erro", etc.).

O que a Tabela de Decisão faz?

Ela ajuda a organizar tudo isso em formato de tabela, com:

- Linhas: que mostram as condições (o que o sistema precisa verificar) e as ações (o que o sistema deve fazer).
- Colunas: cada coluna é uma regra diferente, ou seja, uma combinação de respostas que leva a uma determinada ação.

ALINE FRANÇA

TESTE DE TABELA DE DECISÃO

✅ Exemplo simples

Imagine um caixa eletrônico:

Tem cartão?	Digitou a senha certa?	Pode sacar dinheiro?
Sim	Sim	Sim
Sim	Não	Não
Não	-	Não

Esse é um exemplo básico de uma **tabela de decisão.** Ela mostra como o sistema deve agir em cada situação.

ALINE FRANÇA

TESTE DE TABELA DE DECISÃO

O que significam os símbolos?

- V = Verdadeiro (ou seja, a condição é satisfeita – como “sim”)
- F = Falso (condição não satisfeita – como “não”)
- - = Irrelevante (essa condição não faz diferença nessa situação)
- N/A = Não aplicável (essa combinação nem faz sentido acontecer)
- X = Ação que deve acontecer(em branco) = Ação que não deve acontecer

- **E se houver muitas combinações?**

Se houver muitas combinações e for inviável testar todas, podemos:

- Cortar as combinações que não fazem sentido
- Juntar as colunas parecidas
- Ou usar uma prioridade baseada no risco (testar primeiro o que é mais importante)

ALINE FRANÇA

TESTE DE TABELA DE DECISÃO

Por que isso é útil nos testes?

Porque permite verificar todas as possíveis combinações de situações e garantir que o sistema reage corretamente a cada uma delas. Isso é especialmente útil quando há muitas regras que se combinam.

Por exemplo, um sistema de descontos pode considerar:

- Se o cliente é VIP
- Se ele comprou mais de R$ 500
- Se é aniversário dele

Com 3 condições assim, já são 8 combinações possíveis! Com 4 condições, são 16... e assim por diante. Então a tabela ajuda a não esquecer nenhuma e a ver onde pode haver erros ou falhas nas regras.

Como medimos se testamos bem?

A gente mede o quanto testamos olhando quantas colunas (regras) viáveis foram testadas, comparado com o total de colunas que deveriam ser testadas. Esse número vira uma porcentagem, quanto mais perto de 100%, melhor!

ALINE FRANÇA

TESTE DE TABELA DE DECISÃO

RESUMO

O teste com tabela de decisão é como montar uma tabela para garantir que, para cada situação possível, o sistema faça a coisa certa. Ele é ótimo para testar regras complicadas, evitar erros escondidos e garantir que o sistema siga todas as instruções certinhas.

ALINE FRANÇA

TESTE DE TRANSIÇÃO DE ESTADO

O teste de transição de estado é uma técnica usada para verificar se um sistema (como um aplicativo ou site) está mudando corretamente de uma situação para outra conforme o esperado.

O que é um “estado”?

Um estado é como uma “fase” ou “condição” em que o sistema pode estar.

Por exemplo, pense em uma porta:

- Estado 1: Fechada
- Estado 2: Aberta
- Estado 3: Trancada

Você muda de um estado para outro com ações (como girar a chave ou empurrar a porta).

ALINE FRANÇA

TESTE DE TRANSIÇÃO DE ESTADO

O que é uma transição?

Uma transição é a mudança de um estado para outro, quando algo acontece (um evento).

Por exemplo:

- Evento: "Apertar o botão“
- Resultado: Porta muda de "Fechada" para "Aberta“

Às vezes, essa transição depende de uma condição especial. Exemplo:

“Só destrancar a porta se a chave estiver correta”.

ALINE FRANÇA

TESTE DE TRANSIÇÃO DE ESTADO

Modelos que ajudam nos testes

Existem duas formas de representar isso:

1. **Diagrama de estados:** É um desenho mostrando os estados e as setas entre eles, indicando as possíveis transições.
2. **2. Tabela de estados:** É uma tabela onde:

- As linhas mostram os estados atuais
- As colunas mostram os eventos (ações)
- As células mostram para onde o sistema deve ir depois de cada evento

Se uma célula estiver vazia, significa que aquela ação não deveria funcionar naquele estado.

ALINE FRANÇA

TESTE DE TRANSIÇÃO DE ESTADO

Como são feitos os testes?

Os testes simulam situações reais para ver se o sistema:

- Está mudando de estado corretamente
- Está impedindo mudanças erradas

Exemplo: Se o sistema está no estado “Deslogado” e o usuário clica em “ver perfil”, o sistema não deve deixar acessar (transição inválida).

Tipos de cobertura (quanto foi testado):

1. Cobertura de todos os estados:

-Garante que o sistema passou por todos os estados possíveis pelo menos uma vez.

-Exemplo: Testar se o sistema já ficou em “logado”, “deslogado”, “bloqueado”, etc.

ALINE FRANÇA

TESTE DE TRANSIÇÃO DE ESTADO

2. Cobertura de transições válidas:

- Garante que todas as mudanças corretas entre estados foram testadas.

- Exemplo: Ir de “deslogado” para “logado”, de “logado” para “deslogado”, etc.

3. Cobertura de todas as transições (válidas e inválidas)

- Garante que o sistema também reage bem às tentativas erradas de mudar de estado.

- Exemplo: Tentar entrar em "modo administrador" sem permissão → o sistema deve bloquear.

Por que isso é importante? Porque muitos erros nos sistemas acontecem quando algo muda de um estado para outro. Exemplos comuns:

- Deixa entrar sem login
- Permite alterar um pedido já entregue
- Libera acesso mesmo com senha errada

Testar as transições ajuda a garantir que o sistema só faz o que deveria fazer, e reage corretamente quando alguém tenta fazer algo que não é permitido.

ALINE FRANÇA

TÉCNICAS DE TESTE CAIXA-BRANCA

Existem muitas formas de testar se um sistema ou programa está funcionando direitinho. Uma delas é chamada de teste caixa-branca. Nesse tipo de teste, a pessoa que está testando olha dentro do código do sistema para verificar se tudo está sendo executado corretamente. É como abrir o motor de um carro para ver se todas as peças estão funcionando.

Foco em duas técnicas mais usadas:

1. **Teste de Instruções:** Verifica se cada linha do código foi executada pelo menos uma vez durante os testes.

   Exemplo: Se o código tem 100 linhas, os testes tentam passar por todas elas.

2. **Teste de Ramificações:** Verifica se todas as decisões no código (tipo "se isso acontecer, faça isso; senão, faça aquilo") foram testadas.

   Exemplo: Se o programa pode seguir por dois caminhos diferentes, testamos os dois.

Essas duas técnicas são populares porque são simples e funcionam bem na maioria dos casos.

ALINE FRANÇA

TÉCNICAS DE TESTE CAIXA-BRANCA

E quanto a situações mais críticas? Em áreas onde a segurança é essencial, como:

- Aviões
- Hospitais
- Usinas nucleares...

são usadas técnicas de teste mais rígidas e detalhadas para garantir que absolutamente tudo esteja funcionando perfeitamente. Essas técnicas cobrem o código de forma muito mais profunda, mas não são abordadas aqui.

Outros tipos de testes mais avançados: Existem também técnicas de teste que:

- Não olham só para o código, mas para como o sistema se comporta em níveis mais altos (como testar a comunicação entre sistemas, API).
- São usadas em inteligência artificial, como testar redes neurais (tipo o cérebro de um robô).

Mas esses tipos de teste também não fazem parte deste conteúdo.

ALINE FRANÇA

TESTE DE INSTRUÇÃO E COBERTURA DE INSTRUÇÃO

O que é Teste de Instrução? Imagine que um programa é como uma receita de bolo, cheia de instruções passo a passo. O teste de instrução é uma técnica usada para garantir que cada passo dessa receita foi realmente seguido durante os testes.

O que é “Cobertura de Instrução”? A cobertura de instrução mede quantas dessas instruções foram executadas nos testes.

- Se o programa tem 100 instruções e os testes passaram por 80 delas, a cobertura é de 80%.
- O ideal é tentar chegar o mais perto possível de 100%, ou seja, testar todas as instruções.

ALINE FRANÇA

TESTE DE INSTRUÇÃO E COBERTURA DE INSTRUÇÃO

- **Por que isso é importante?**

Se uma instrução nunca é testada, ela pode esconder um erro. Testando todas, aumentamos a chance de encontrar defeitos.

Mas atenção: Mesmo testando todas as instruções, nem todos os erros aparecem. Por quê?

- Às vezes o erro só aparece com determinados dados.

Exemplo: uma conta de divisão pode funcionar com vários números, mas falha se tentar dividir por zero.

Além disso, esse tipo de teste não garante que todas as decisões do código (como “se isso acontecer, faça isso, senão faça aquilo”) foram testadas.

Isso é outra técnica, chamada teste de decisão, que vai além.

ALINE FRANÇA

TESTE DE INSTRUÇÃO E COBERTURA DE INSTRUÇÃO

Resumo:

- **Teste de instrução** = testar se cada linha do código foi executada.
- **Cobertura de instrução** = porcentagem de linhas que foram testadas.
- 100% de cobertura é ótimo, mas não garante que tudo esteja perfeito.
- Ainda podem existir erros escondidos, especialmente em decisões ou situações específicas.

ALINE FRANÇA

TESTE DE RAMIFICAÇÃO

O que é Teste de Ramificação? Imagine que o código de um programa é como um caminho com várias bifurcações(Bifurcação é quando você chega num ponto onde precisa escolher entre dois ou mais caminhos diferentes), como se você estivesse andando por uma trilha que, em vários momentos, oferece duas opções:

- Se estiver chovendo, vá para a esquerda.
- Se estiver sol, vá para a direita.

Essas bifurcações são chamadas de **ramificações**. Elas acontecem quando o programa precisa tomar decisões, como:“

- Se isso for verdadeiro, faça isso. Se for falso, faça outra coisa.
- ""Escolha entre estas opções."“
- Continue ou pare de repetir algo."

ALINE FRANÇA

COBERTURA DE RAMIFICAÇÃO

O que é Cobertura de Ramificação? É a porcentagem dessas bifurcações que foram testadas.

- Se o programa tem 10 ramificações e os testes passam por 8 delas, a cobertura é de 80%.
- O ideal é atingir 100%, ou seja, testar todos os caminhos possíveis que o código pode seguir.

Por que isso importa? Se um caminho do código **nunca for testado**, ele pode esconder um erro. Testando todas as ramificações, temos mais chance de encontrar defeitos que só aparecem em **certas situações**.

⚠️ Mas atenção: Mesmo testando todos os caminhos, alguns defeitos podem não aparecer. Isso acontece quando o problema só surge se você seguir uma sequência específica de passos, não apenas passar por um caminho isolado.

ALINE FRANÇA

TESTE DE INSTRUÇÃO E COBERTURA DE INSTRUÇÃO

E qual a diferença para o teste de instrução? Cobertura de instrução garante que cada linha de código foi executada.

- Cobertura de ramificação garante que cada decisão foi testada com todas as suas opções (verdadeiro/falso, sim/não).
- Se você atingir 100% de cobertura de ramificação, você também já testou todas as instruções. Mas o contrário não é verdade, dá pra ter 100% de instruções testadas sem ter testado todas as decisões.

Resumindo:

- Ramificação = quando o programa precisa tomar uma decisão.
- Teste de ramificação = garantir que todos os caminhos possíveis foram testados.
- Isso ajuda a descobrir mais erros, especialmente os que surgem em condições específicas.

ALINE FRANÇA

- DÚVIDAS?

- SUGESTÕES?

- RECLAMAÇÃO?

- ELOGIO?