

CURSO DE TECNOLOGIA EM DESENVOLVIMENTO DE SISTEMAS

Disciplina: Qualidade de Software

Prof. Luiz Mesquita — Roteiro de Laboratório

PRÁTICA 3: TESTES FUNCIONAIS (CAIXA PRETA)

Particionamento em Classes de Equivalência (PCE) e Análise do Valor Limite (AVL)

Olá, Aluno(a)!

Como seu professor de Qualidade de Software com atuação ativa nas principais empresas de tecnologia do mercado, preparei um laboratório prático fascinante e interativo sobre o Tema 8: Teste Funcional (Caixa Preta).

No dia a dia do desenvolvimento, o teste funcional foca em validar se o sistema executa as funções esperadas de acordo com as especificações técnicas, sem que o testador precise conhecer o código interno. Mas há um grande desafio prático: testar todas as entradas possíveis de um sistema (o chamado teste exaustivo) é matematicamente impossível e inviável.

Para resolver isso, nós engenheiros de QA usamos duas técnicas essenciais de otimização para obter alta cobertura de defeitos com o menor número de testes:

- **Particionamento em Classes de Equivalência (PCE):** Dividimos o domínio de entrada em grupos que o sistema trata de forma parecida. Se um valor do grupo funciona, assume-se que os outros também funcionam.
- **Análise do Valor Limite (AVL):** Focamos nossos testes nos extremos desses grupos (fronteiras), pois é exatamente ali onde a maioria dos bugs de programação ocorre devido a erros de lógica simples do desenvolvedor.

Hoje, vamos testar o Sistema de Reserva de Assentos de um Cinema com capacidade para até 100 lugares, especificação direta de nosso material didático. Desenhei um aplicativo web real e inseri, de propósito, dois bugs clássicos de limite.

Abra o seu navegador e acompanhe este passo a passo sequencial para caçar e corrigir esses bugs!

Parte 1: Configurando o Laboratório Virtual (Sem Instalações)

- 1. Abra o ambiente de testes:** No seu navegador, abra uma nova guia e acesse o site gratuito jsfiddle.net. O JSFiddle é um playground online onde podemos programar e testar códigos HTML e JavaScript instantaneamente.
- 2. Entenda a interface:** Você verá uma tela dividida em 4 quadrantes (HTML, CSS, JavaScript e o quadrante Result no canto inferior direito, onde nossa aplicação vai rodar).
- 3. Limpe o painel HTML:** Se houver algum texto padrão no painel superior esquerdo (HTML), selecione tudo e apague.
- 4. Copie o código do aplicativo:** Copie todo o código HTML estruturado abaixo (que contém nossa tela e as regras lógicas escritas pelo programador):

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <style>
    body { font-family: sans-serif; background-color: #f4f4f9; padding: 20px;
color: #333; }
    .card { background: white; padding: 20px; border-radius: 8px; max-width:
400px; box-shadow: 0 4px 8px rgba(0,0,0,0.1); margin: auto; }
    h2 { color: #1e3d59; text-align: center; margin-top: 0; }
    label { font-weight: bold; display: block; margin-bottom: 8px; }
    input { width: 100%; padding: 10px; box-sizing: border-box; margin-bottom:
15px; border: 1px solid #ccc; border-radius: 4px; font-size: 16px; }
    button { width: 100%; background-color: #17b978; color: white; border: none;
padding: 12px; border-radius: 4px; font-size: 16px; cursor: pointer; font-weight:
bold; }
    button:hover { background-color: #118f5d; }
    .msg { margin-top: 15px; padding: 10px; border-radius: 4px; font-weight: bold;
text-align: center; display: none; }
    .success { background-color: #d4edda; color: #155724; border: 1px solid
#c3e6cb; display: block; }
    .error { background-color: #f8d7da; color: #721c24; border: 1px solid #f5c6cb;
display: block; }
  </style>
</head>
<body>
  <div class="card">
    <h2>CineProz - Reserva de Ingressos</h2>
    <p style="font-size: 13px; text-align: center; color: #666; margin-bottom:
15px;">
      <strong>Especificação:</strong> É permitido reservar de 1 a 100 assentos por
compra.
    </p>
    <label for="seatsInput">Quantidade de Assentos:</label>
    <input type="text" id="seatsInput" placeholder="Ex: 5">
    <button onclick="realizarReserva()">Reservar Assentos</button>
    <div id="resultado" class="msg"></div>
```

```

</div>

<script>
  function realizarReserva() {
    const input = document.getElementById('seatsInput').value.trim();
    const resultado = document.getElementById('resultado');
    resultado.className = "msg";

    // Validando se a entrada não é numérica ou vazia
    if (input === "" || isNaN(input) || input.includes('.')) {
      resultado.innerText = "Erro: Digite apenas números inteiros válidos.";
      resultado.className = "msg error";
      return;
    }

    const qtd = parseInt(input, 10);

    // ATENÇÃO: O desenvolvedor programou as regras de limite abaixo:
    if (qtd < 0) {
      resultado.innerText = "Erro: A quantidade não pode ser negativa.";
      resultado.className = "msg error";
    } else if (qtd >= 100) {
      resultado.innerText = "Erro: Capacidade máxima do cinema excedida (Máximo
100).";
      resultado.className = "msg error";
    } else {
      resultado.innerText = "Sucesso! " + qtd + " assento(s) reservado(s) com
sucesso.";
      resultado.className = "msg success";
    }
  }
</script>
</body>
</html>

```

5. Cole o código no JSFiddle: Cole este código completo diretamente dentro do quadrante HTML (superior esquerdo) do JSFiddle.

6. Execute o aplicativo: No menu superior do JSFiddle, clique no botão "Run" (ícone de engrenagem ou botão de play).

7. Verifique a renderização: Olhe para o quadrante inferior direito (Result). Você deverá ver a interface azul e verde do nosso sistema de cinema "CineProz" pronta para ser utilizada.

Parte 2: Mapeando os Casos de Teste (Estratégia de QA)

Antes de clicarmos em qualquer botão, vamos aplicar a teoria e desenhar nossos cenários de teste funcionais com base no intervalo de 1 a 100 assentos.

- 8. Identifique a classe de equivalência válida (PCE):** O intervalo aceitável de reserva é de [1 a 100] assentos. Qualquer número inteiro aqui dentro deve funcionar de maneira similar e resultar em sucesso.
- 9. Identifique as classes de equivalência inválidas:** Valores abaixo do mínimo: Qualquer número menor do que 1 (incluindo o zero e negativos). Valores acima do máximo: Qualquer número acima de 100 (ex: 101). Entradas não numéricas: Textos ou campos em branco.
- 10. Determine as fronteiras para a Análise do Valor Limite (AVL):** Vamos focar nos números exatos nos limites e vizinhos imediatos.
- 11. Limite Inferior Extremo:** Os valores a testar na borda de baixo são 1 (deve passar) e o seu vizinho imediato inválido 0 (deve falhar).
- 12. Limite Superior Extremo:** Os valores a testar na borda de cima são 100 (deve passar) e o seu vizinho imediato inválido 101 (deve falhar).

Parte 3: Executando os Testes e Caçando os Bugs

Agora vamos interagir com a tela para realizar a validação do produto.

- 13. Cenário 1 (Classe Válida):** Digite o valor 5 no campo "Quantidade de Assentos" e clique no botão "Reservar Assentos".
- 14. Analise o resultado:** O sistema exibirá uma mensagem verde de "Sucesso". Este teste passou como esperado.
- 15. Cenário 2 (Limite Inferior Válido):** Apague o número 5, digite o valor 1 e clique em "Reservar Assentos".
- 16. Analise o resultado:** O sistema exibirá uma mensagem de "Sucesso". Este teste de limite passou conforme a especificação.
- 17. Cenário 3 (Limite Inferior Inválido - O Teste do Vilão):** Limpe o campo, digite o número 0 e clique em "Reservar Assentos".
- 18. 🕵️ ACHAMOS O PRIMEIRO BUG!** O sistema exibiu a mensagem "Sucesso! 0 assento(s) reservado(s) com sucesso". Como um cinema pode aceitar uma reserva de zero lugares? A validação do limite inferior falhou!
- 19. Cenário 4 (Limite Superior Válido):** Limpe o campo, digite exatamente 100 e clique em "Reservar Assentos".
- 20. 🕵️ ACHAMOS O SEGUNDO BUG!** O sistema barrou sua entrada e exibiu a mensagem de erro "Capacidade máxima do cinema excedida (Máximo 100)". Mas o requisito técnico diz expressamente que é permitido reservar de 1 até 100 assentos! O limite válido de 100 foi injustamente bloqueado.
- 21. Cenário 5 (Limite Superior Inválido):** Limpe o campo, digite o valor 101 e clique em "Reservar Assentos".
- 22. Analise o resultado:** O sistema exibe o erro de capacidade máxima de forma correta.

23. Cenário 6 (Tratamento de Erros): Digite o texto "três" no campo e clique em "Reservar Assentos". O sistema exibe corretamente a mensagem de erro solicitando apenas números inteiros válidos.

Parte 4: Corrigindo o Código no Ambiente de Teste

O papel do QA moderno não é apenas apontar o dedo para o erro, mas entender por que ele ocorreu e, se possível, guiar a correção.

24. Encontre a falha no código: Olhe para o painel de código HTML que você colou no JSFiddle e procure pelas seguintes linhas lógicas de validação dentro da tag <script> (próximo à linha 43):

```
if (qtd < 0) { ... }
else if (qtd >= 100) { ... }
```

Análise do Professor:

- O desenvolvedor escreveu 'qtd < 0' para barrar negativos, mas se esqueceu do 0. Por isso o zero passou como sucesso!
- Ele também escreveu 'qtd >= 100' (maior ou igual a 100) para o limite superior, o que incluiu e bloqueou o próprio valor 100!

25. Realize a correção (Hotfix): Altere essas duas condições lógicas diretamente no painel do código no JSFiddle para o seguinte padrão correto:

```
if (qtd < 1) { // Garante que o zero e negativos sejam bloqueados!
    resultado.innerText = "Erro: A quantidade deve ser de pelo menos 1 assento.";
    resultado.className = "msg error";
} else if (qtd > 100) { // Bloqueia apenas valores estritamente maiores que 100!
    resultado.innerText = "Erro: Capacidade máxima do cinema excedida (Máximo 100).";
    resultado.className = "msg error";
}
```

26. Rode e revalide o sistema: Clique novamente no botão "Run" no topo do JSFiddle para atualizar a aplicação. Refaça os testes informando 0 (agora deve dar erro) e 100 (agora deve dar sucesso).

Parabéns! Você acabou de aplicar com sucesso as técnicas de Particionamento em Classes de Equivalência e Análise do Valor Limite, caçou dois bugs críticos de fronteira e corrigiu o produto para o cliente final!

Folha de Entrega — Tabela de Resultados do Aluno

Preencha os resultados de sua execução prática na tabela de verificação a seguir:

Caso de Teste	Valor Testado	Resultado Esperado	Resultado Encontrado (Antes)	Resultado Encontrado (Depois)
CT-01 (PCE Válida)	5	Sucesso (Mensagem Verde)		
CT-02 (Limite Inf Válido)	1	Sucesso (Mensagem Verde)		
CT-03 (Limite Inf Inválido)	0	Mensagem de Erro		
CT-04 (Limite Sup Válido)	100	Sucesso (Mensagem Verde)		
CT-05 (Limite Sup Inválido)	101	Mensagem de Erro		
CT-06 (Erros - Texto)	"três"	Mensagem de Erro		