

DISCIPLINA: QUALIDADE DE SOFTWARE

Prof. Luiz Mesquita | Prática 4: Teste Estrutural e Diagnóstico de Software

1. Introdução e Fundamentação Teórica

Esta atividade prática laboratorial integra de forma dinâmica dois dos principais pilares do teste e garantia de qualidade de software do mercado moderno: o Teste Estrutural (Caixa Branca – Tema 9) e o Diagnóstico e Automação de Software (Tema 10). Ao longo desta prática, você aprenderá a exercitar as lógicas internas do código de uma aplicação, analisar métricas de cobertura, simular interações automatizadas como se estivesse utilizando o framework Selenium e estruturar um relatório de bug (diagnóstico) de alta precisão.

No Teste Estrutural, avaliamos o código-fonte por dentro. Usamos a Cobertura de Instruções (Statement Coverage) para garantir que cada linha de comando útil seja exercitada pelo menos uma vez, e a Cobertura de Decisões (Branch Coverage) para percorrer todos os caminhos (verdadeiro/falso) de cada estrutura condicional. Adicionalmente, esta atividade aborda métricas avançadas utilizadas em engenharia de software baseada em busca (Search-Based Testing):

- **Nível de Aproximação (Approximation Level):** Mede o número de bifurcações (nós condicionais) corretas que um teste percorreu em direção a um nó alvo específico do código antes de desviar pelo caminho errado. Pense no grafo de fluxo do programa como um labirinto: quanto mais próximo você chegar do tesouro (o alvo de cobertura) antes de cometer um erro, maior é o seu nível de aproximação.
- **Distância Local (Local Distance):** Quando o teste falha em satisfazer uma condição lógica específica (ex: 'renda > 2000'), a distância local calcula matematicamente o quão longe a entrada esteve de torná-la verdadeira. No caso de 'renda > 2000', se fornecermos '1500', a distância local para alcançar o limite correto é de exatamente 501 unidades (2001 - 1500).

No campo do Diagnóstico de Software (Tema 10), traçamos um paralelo com os rigorosos critérios de revisão sistemática da Cochrane Collaboration (famosa na área médica por analisar a acurácia de diagnósticos). Um teste de software funciona exatamente como um exame de diagnóstico médico: precisamos medir sua precisão para evitar Falsos Positivos e Falsos Negativos, encontrando a causa raiz exata (root cause) dos defeitos do sistema.

2. Código-Fonte Unificado da Aplicação

O código abaixo é uma aplicação HTML+JavaScript completa. Ela simula o funcionamento de um Motor de Triagem Clínica de Diagnóstico que contém desvios condicionais aninhados e de forma interativa calcula a Cobertura de Instruções (Statement Coverage), a Cobertura de Decisões (Branch Coverage), o Nível de Aproximação e a Distância Local à medida que novos casos de teste são injetados. Copie-o integralmente para realizar a prática.

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <title>PROZ - Prática de Caixa Branca e Diagnóstico</title>
  <style>
    body { font-family: 'Segoe UI', Arial, sans-serif; background-color: #f7fafc;
```

```

padding: 20px; color: #2d3748; }
.container { display: flex; flex-wrap: wrap; gap: 20px; max-width: 1100px; margin:
auto; }
.card { background: white; padding: 20px; border-radius: 12px; box-shadow: 0 4px
15px rgba(0,0,0,0.05); flex: 1; min-width: 320px; }
h2 { color: #1a365d; text-align: center; margin-top: 0; border-bottom: 2px solid
#e2e8f0; padding-bottom: 10px; }
.form-group { margin-bottom: 12px; }
label { font-weight: bold; display: block; margin-bottom: 6px; font-size: 13px; }
input { width: 100%; padding: 10px; box-sizing: border-box; border: 1px solid
#cbd5e0; border-radius: 6px; font-size: 15px; }
button { width: 100%; background-color: #2b6cb0; color: white; border: none;
padding: 12px; border-radius: 6px; font-size: 15px; cursor: pointer; font-weight: bold;
transition: 0.2s; margin-bottom: 8px; }
button:hover { background-color: #2b5c8f; }
.btn-selenium { background-color: #319795; }
.btn-selenium:hover { background-color: #2c7a7b; }
.coverage-box { background-color: #edf2f7; padding: 15px; border-radius: 8px;
margin-top: 15px; font-size: 13px; line-height: 1.5; }
.code-view { font-family: 'Consolas', monospace; background-color: #1a202c; color:
#cbd5e0; padding: 15px; border-radius: 8px; font-size: 12px; line-height: 1.6; }
.executed-line { background-color: rgba(72, 187, 120, 0.25); color: #fff;
border-left: 3px solid #48bb78; padding-left: 5px; }
.unexecuted-line { border-left: 3px solid transparent; padding-left: 5px; }
.metric-title { font-weight: bold; color: #2c5282; margin-top: 10px; display:
block; }
.msg { margin-top: 15px; padding: 10px; border-radius: 6px; font-weight: bold;
text-align: center; }
.bg-success { background-color: #48bb78; color: white; }
.bg-danger { background-color: #f56565; color: white; }
.terminal-box { font-family: 'Consolas', monospace; background-color: #2d3748;
color: #a0aec0; padding: 10px; border-radius: 6px; font-size: 11px; max-height: 150px;
overflow-y: auto; margin-top: 10px; }
</style>
</head>
<body>
  <div class="container">
    <!-- Formulário Principal -->
    <div class="card">
      <h2>Painel da Clínica (Objeto de Teste)</h2>
      <p style="font-size: 12px; color: #4a5568; text-align: center; margin-bottom:
15px;">
        Insira os dados clínicos ou use o simulador do Selenium para realizar testes
        automatizados.
      </p>

      <div class="form-group">
        <label for="tempInput">Temperatura do Paciente (C) [ID: tempInput]</label>
        <input type="number" step="0.1" id="tempInput" placeholder="Ex: 37.5">
      </div>

      <div class="form-group">
        <label for="sintomasInput">Dias com Sintomas (Inteiro) [ID:
sintomasInput]</label>
        <input type="number" id="sintomasInput" placeholder="Ex: 5">
      </div>
    </div>
  </div>

```

```

<div class="form-group">
  <label for="oxigenioInput">Saturação de Oxigênio (%) [ID:
oxigenioInput]:</label>
  <input type="number" id="oxigenioInput" placeholder="Ex: 95">
</div>

<button id="btnExecutar" onclick="executarTriagemClinica()">Executar Fluxo Caixa
Branca</button>
<button class="btn-selenium" id="btnSelenium" onclick="simularSelenium()">Simular
Execução Selenium (Automático)</button>

<div id="statusResultado" class="msg" style="display: none;"></div>

<span class="metric-title">Terminal de Console de Teste:</span>
<div id="terminalConsole" class="terminal-box">Aguardando execução de
testes...</div>
</div>

<!-- Painel de Cobertura e Grafo de Código -->
<div class="card">
  <h2>Monitoramento de Cobertura</h2>

  <div class="coverage-box">
    <div><strong>Cobertura de Instruções (Linhas):</strong> <span
id="instrucoesPct" style="font-weight: bold; color: #2b6cb0;">0%</span></div>
    <div><strong>Cobertura de Decisões (Branch):</strong> <span id="decisooesPct"
style="font-weight: bold; color: #2b6cb0;">0%</span></div>

    <span class="metric-title">Nível de Aproximação (Alvo: ALTA GRAVIDADE):</span>
    <div id="nivelAproximacao" style="font-weight: bold; font-size: 13px;
margin-top: 3px;">N/A</div>

    <span class="metric-title">Cálculo de Distância Local na Falha Lógica:</span>
    <div id="distanciaLocal" style="font-family: monospace; font-size: 13px;
margin-top: 3px; color: #e53e3e;">N/A</div>
  </div>

  <span class="metric-title">Estrutura Interna de Código (Caixa Branca):</span>
  <div class="code-view">
    <div id="ln1" class="unexecuted-line">01: function triagemPaciente(temp, dias,
oxigenio) {</div>
    <div id="ln2" class="unexecuted-line">02:   if (temp >= 37.8) {</div>
    <div id="ln3" class="unexecuted-line">03:     if (dias > 3) {</div>
    <div id="ln4" class="unexecuted-line">04:       if (oxigenio < 92) {</div>
    <div id="ln5" class="unexecuted-line">05:         return "ALTA GRAVIDADE
(ALVO)";</div>
    <div id="ln6" class="unexecuted-line">06:       } else {</div>
    <div id="ln7" class="unexecuted-line">07:         return "MONITORAMENTO
DOMICILIAR";</div>
    <div id="ln8" class="unexecuted-line">08:       }</div>
    <div id="ln9" class="unexecuted-line">09:     } else {</div>
    <div id="ln10" class="unexecuted-line">10:       return "SINTOMAS INICIAIS
LEVES";</div>
    <div id="ln11" class="unexecuted-line">11:     }</div>
    <div id="ln12" class="unexecuted-line">12:   } else {</div>

```

```

    <div id="ln13" class="unexecuted-line">13:    return "PACIENTE ESTÁVEL";</div>
    <div id="ln14" class="unexecuted-line">14:    }</div>
    <div id="ln15" class="unexecuted-line">15: }</div>
  </div>
</div>
</div>

<script>
const linhasExecutadasGlobais = new Set();
const branchesExecutadosGlobais = new Set();
const consoleLogs = [];

function logConsole(msg) {
  consoleLogs.push(msg);
  const box = document.getElementById('terminalConsole');
  box.innerHTML = consoleLogs.slice(-6).join('<br>');
  box.scrollTop = box.scrollHeight;
}

function executarTriagemClinica() {
  const temp = parseFloat(document.getElementById('tempInput').value);
  const dias = parseInt(document.getElementById('sintomasInput').value);
  const oxigenio = parseInt(document.getElementById('oxigenioInput').value);

  if (isNaN(temp) || isNaN(dias) || isNaN(oxigenio)) {
    alert("Erro: Preencha todos os campos com valores válidos antes de testar!");
    return;
  }

  logConsole("Iniciando varredura estrutural...");

  // Reset visual das linhas
  for (let i = 1; i <= 15; i++) {
    document.getElementById('ln' + i).className = "unexecuted-line";
  }

  const linhasDesteTeste = [];
  let resultado = "";
  let nivelAprox = "";
  let distLocal = "";

  linhasDesteTeste.push(1, 2);

  // Decisão 1: Temperatura
  if (temp >= 37.8) {
    branchesExecutadosGlobais.add("D1_True");
    linhasDesteTeste.push(3);

    // Decisão 2: Dias com Sintomas
    if (dias > 3) {
      branchesExecutadosGlobais.add("D2_True");
      linhasDesteTeste.push(4);

      // Decisão 3: Saturação de Oxigênio
      // ATENÇÃO: O desenvolvedor cometeu um bug lógico na linha 4!
      // Ele codificou 'oxigenio <= 92' em vez do limite correto especificado de

```

```

'oxigenio < 92'.
    if (oxigenio <= 92) {
        branchesExecutadosGlobais.add("D3_True");
        linhasDesteTeste.push(5);
        resultado = "ALTA GRAVIDADE (ALVO)";
        nivelAprox = "Nível 1 (Alvo Alcançado! )";
        distLocal = "Nenhuma falha (0)";
    } else {
        branchesExecutadosGlobais.add("D3_False");
        linhasDesteTeste.push(6, 7);
        resultado = "MONITORAMENTO DOMICILIAR";
        nivelAprox = "Nível 2 (Desviou na Decisão 3 - Oxigênio)";
        distLocal = "Oxigênio esperado < 92 | Distância Local: " +
Math.abs(oxigenio - 91);
    }
    } else {
        branchesExecutadosGlobais.add("D2_False");
        linhasDesteTeste.push(9, 10);
        resultado = "SINTOMAS INICIAIS LEVES";
        nivelAprox = "Nível 3 (Desviou na Decisão 2 - Dias)";
        distLocal = "Dias de sintomas esperados > 3 | Distância Local: " + (4 -
dias);
    }
    } else {
        branchesExecutadosGlobais.add("D1_False");
        linhasDesteTeste.push(12, 13);
        resultado = "PACIENTE ESTÁVEL";
        nivelAprox = "Nível 4 (Desviou na Decisão 1 - Temp)";
        distLocal = "Temperatura esperada >= 37.8 | Distância Local: " + (37.8 -
temp).toFixed(1) + "C";
    }
    linhasDesteTeste.push(15);

    // Pintar linhas executadas
    linhasDesteTeste.forEach(l => {
        linhasExecutadasGlobais.add(l);
        document.getElementById('ln' + l).className = "executed-line";
    });

    // Cálculo de Cobertura de Instruções (Statement Coverage)
    // Linhas úteis de processamento real: 1, 2, 3, 4, 5, 7, 10, 13, 15 (9 linhas
estruturais)
    const linhasUteis = [1, 2, 3, 4, 5, 7, 10, 13, 15];
    let executadasContadas = 0;
    linhasUteis.forEach(l => {
        if (linhasExecutadasGlobais.has(l)) executadasContadas++;
    });
    const pctInstrucoes = Math.round((executadasContadas / linhasUteis.length) *
100);

    // Cobertura de Decisões (6 branches possíveis: D1_True, D1_False, D2_True,
D2_False, D3_True, D3_False)
    const pctDecisoes = Math.round((branchesExecutadosGlobais.size / 6) * 100);

    // Exibir resultados lógicos e métricas
    document.getElementById('instrucoesPct').innerText = pctInstrucoes + "%";

```

```

document.getElementById('decisoosPct').innerText = pctDecisoos + "%";
document.getElementById('nivelAproximacao').innerText = nivelAprox;
document.getElementById('distanciaLocal').innerText = distLocal;

const rBox = document.getElementById('statusResultado');
rBox.innerText = "Resultado: " + resultado;
rBox.className = "msg " + (resultado.includes("ALTA") ? "bg-danger" :
"bg-success");
rBox.style.display = "block";

logConsole("Testes executados com sucesso.");
}

function simularSelenium() {
  logConsole("Carregando Selenium WebDriver...");
  logConsole("Configurando WebDriver local...");

  setTimeout(() => {
    logConsole('driver.findElement(By.id("tempInput")).sendKeys("38.5")');
    document.getElementById('tempInput').value = "38.5";
  }, 500);

  setTimeout(() => {
    logConsole('driver.findElement(By.id("sintomasInput")).sendKeys("5")');
    document.getElementById('sintomasInput').value = "5";
  }, 1000);

  setTimeout(() => {
    logConsole('driver.findElement(By.id("oxigenioInput")).sendKeys("85")');
    document.getElementById('oxigenioInput').value = "85";
  }, 1500);

  setTimeout(() => {
    logConsole('driver.findElement(By.id("btnExecutar")).click()');
    executarTriagemClinica();
  }, 2000);
}
</script>
</body>
</html>

```

3. Roteiro de Realização Prática (42 Passos)

- 01.** Abra uma nova aba de navegação e acesse o ambiente de desenvolvimento gratuito JSFiddle no endereço: jsfiddle.net.
- 02.** Visualize a interface do JSFiddle, observando que ela é composta por quatro quadrantes principais: HTML, CSS, Javascript e Result.
- 03.** Selecione qualquer código de exemplo ou texto que já esteja escrito por padrão dentro do quadrante superior esquerdo (HTML) e apague-o completamente.
- 04.** Copie todo o código-fonte unificado HTML e Javascript fornecido na Seção 2 deste roteiro.
- 05.** Cole todo o código copiado diretamente no quadrante superior esquerdo (HTML) do seu ambiente no JSFiddle.

06. No menu superior horizontal azul do JSFiddle, clique sobre o botão 'Run' (representado por um ícone de engrenagem ou sinal de reprodução) para carregar o sistema.
07. Observe o quadrante inferior direito, rotulado como 'Result'. Ele agora exibe a interface gráfica do nosso Motor de Triagem de Crédito e Monitor de Cobertura.
08. Familiarize-se com a tela de testes: veja os três campos de entrada de dados de triagem no painel esquerdo e os indicadores de métricas de cobertura no painel direito.
09. Observe o painel 'Estrutura Interna de Código (Caixa Branca)' na parte inferior direita. Note que as 15 linhas estão na cor cinza escuro, pois nenhum teste foi executado ainda.
10. Verifique que os marcadores de Cobertura de Instruções e Cobertura de Decisões estão atualmente marcando 0%, indicando ausência de testes executados.
11. Identifique a linha número '05: return \'ALTA GRAVIDADE (ALVO)\';' no painel de código interno. Nosso principal objetivo de teste estrutural é descobrir quais caminhos de dados ativam este alvo.
12. Vamos iniciar o Caso de Teste 1: clique no campo de entrada 'Temperatura do Paciente' e digite o valor de febre leve: 37.5.
13. No campo de entrada 'Dias com Sintomas', informe o número 2.
14. No campo de entrada 'Saturação de Oxigênio (%)', informe o valor normal de 98.
15. Clique sobre o botão azul 'Executar Fluxo Caixa Branca'.
16. Observe no terminal integrado de log a mensagem indicando o início e a conclusão da varredura estrutural.
17. Verifique no painel de código-fonte que as linhas 01, 02, 12, 13 e 15 ficaram pintadas de verde. Elas representam o caminho lógico percorrido por esta primeira entrada.
18. Analise os indicadores de Cobertura: a Cobertura de Instruções subiu de 0% para 44% acumulados.
19. Observe a métrica de Nível de Aproximação (Approximation Level). O indicador exibe 'Nível 4 (Desviou na Decisão 1 - Temp)', indicando que o fluxo errou o caminho logo no primeiro nó condicional.
20. Observe o indicador de Distância Local na Falha Lógica. Ele exibe 'Temperatura esperada >= 37.8 | Distância Local: 0.3C'. Essa é a distância matemática exata que faltou para a entrada de 37.5 atingir os 37.8 graus exigidos pelo desvio condicional.
21. Vamos iniciar o Caso de Teste 2: limpe os campos de entrada do formulário.
22. Informe uma temperatura febril alta: digite 38.0.
23. No campo de dias com sintomas, informe o número 1.
24. No campo de saturação de oxigênio, informe o valor normal de 95.
25. Clique no botão azul 'Executar Fluxo Caixa Branca'.
26. Observe no painel de código-fonte que o fluxo avançou no sistema. Agora, as linhas 03, 09, 10 e 15 também ficaram pintadas de verde, representando o tratamento lógico para sintomas iniciais leves.
27. Verifique que a Cobertura de Instruções aumentou de forma cumulativa, atingindo agora a marca de 67% de linhas úteis cobertas.

28. Verifique a métrica de Nível de Aproximação. Ela atualizou para 'Nível 3 (Desviou na Decisão 2 - Dias)', provando que o fluxo avançou mais um nível em direção ao Alvo em relação ao teste anterior.
29. Analise a Distância Local. Ela exibe 'Dias de sintomas esperados > 3 | Distância Local: 3', indicando a lacuna numérica para o fluxo passar pelo segundo desvio estrutural.
30. Vamos iniciar o Caso de Teste 3: mantenha a temperatura em 38.0 e mude o campo de sintomas para 5 dias de duração.
31. No campo de saturação de oxigênio, informe o valor de 95%.
32. Clique em 'Executar Fluxo Caixa Branca'.
33. Analise o avanço de cobertura. O código passou pelas linhas 03, 04, 06 e 07. A Cobertura de Instruções saltou para 89% e a Cobertura de Decisões alcançou 83% de caminhos testados.
34. Observe o Nível de Aproximação: ele atualizou para 'Nível 2 (Desviou na Decisão 3 - Oxigênio)'. Estávamos a apenas uma bifurcação condicional de atingir o Alvo do sistema.
35. Observe a Distância Local. Ela exibe 'Oxigênio esperado < 92 | Distância Local: 4'. O valor de entrada (95) falhou por uma diferença de 4 unidades em relação ao limite mínimo de 91.
36. Agora utilizaremos o Selenium! Limpe todos os campos de entrada do simulador.
37. Clique sobre o botão verde 'Simular Execução Selenium (Automático)'.
38. Observe os logs no console integrado. O simulador recria exatamente o comportamento de um driver automatizado carregando o navegador, preenchendo os seletores por seus IDs (tempInput, sintomasInput, oxigenioInput) e disparando o clique automatizado no botão.
39. Aguarde os dois segundos de delay de simulação programados.
40. Observe que, de forma automatizada via script, o motor foi executado com as entradas: Temperatura 38.5, Sintomas 5 e Oxigênio 85.
41. Analise a resposta: O resultado no painel exibe a mensagem vermelha 'Resultado: ALTA GRAVIDADE (ALVO)'. O Nível de Aproximação indica 'Nível 1 (Alvo Alcançado! 🎉)'.
42. Comemore: seu teste de caixa branca e automação atingiu as marcas máximas de 100% de Cobertura de Instruções (Statement Coverage) e 100% de Cobertura de Decisões (Branch Coverage)!

4. Diagnóstico de Bugs e Relatório (Padrão Cochrane)

Acurácia Diagnóstica e a Relação de Qualidade: Assim como o padrão metodológico de revisão sistemática da Cochrane Collaboration analisa ensaios clínicos para validar a precisão de ferramentas de diagnóstico médico, na Qualidade de Software analisamos a precisão de testes automatizados e regras de decisão. Precisamos identificar erros do tipo Falso Positivo (o sistema reporta um bug que não existe) e Falso Negativo (o sistema deixa passar um bug crítico, camuflando o problema).

IDENTIFICAÇÃO DO BUG DE LIMITE: De acordo com os requisitos e especificações fornecidos, o estado de ALTA GRAVIDADE (Alvo) só deve ser sinalizado quando a Saturação de Oxigênio for estritamente menor que 92% (< 92). Contudo, no código-fonte interno do desenvolvedor (Linha 4), a lógica foi implementada como 'if (oxigenio <= 92)'. Isso significa que se um paciente estiver com exatamente 92% de saturação (limite normal), o sistema o classificará indevidamente como ALTA

GRAVIDADE. Isto gera um diagnóstico impreciso no produto, provocando o desperdício de leitos hospitalares (um típico Falso Positivo em controle de qualidade clínico).

RELATÓRIO DE BUG (A SER PREENCHIDO PELO ALUNO)

ID do Defeito:	BUG-001 – Falha de Limite no Critério de Oxigênio
Severidade / Impacto:	ALTA – Risco de falso positivo clínico e desperdício de recursos.
Componente afetado:	Motor de Triagem de Decisões (Função triagemPaciente – Linha 4)
Passos para Reproduzir:	1. Preencher Temperatura = 38.0\n2. Preencher Sintomas = 4 dias\n3. Preencher Oxigênio = 92\n4. Clicar em Executar Fluxo
Resultado Esperado (Especificado):	O sistema deve retornar MONITORAMENTO DOMICILIAR (pois oxigênio é 92, não menor que 92).
Resultado Atual Observado (Bug):	O sistema retorna indevidamente ALTA GRAVIDADE (ALVO) devido ao operador '<='.

5. Folha de Coleta de Métricas de Caixa Branca

Instruções: Registre abaixo as métricas observadas na tela durante a realização dos testes de cobertura.

Caso de Teste	Dados Injetados	Statement Coverage (%)	Approximation Level	Local Distance
CT-01	Temp=37.5, Dias=2, Oxi=98	44%	Nível 4 (Desvio na Temp)	0.3 C
CT-02	Temp=38.0, Dias=1, Oxi=95	67%	Nível 3 (Desvio no Score/Dias)	3
CT-03	Temp=38.0, Dias=5, Oxi=95	89%	Nível 2 (Desvio no Oxigênio)	4
CT-04 (Selenium)	Temp=38.5, Dias=5, Oxi=85	100%	Nível 1 (Alvo Alcançado! 🎉)	0