

# FICHA DE ERRO - TRABALHO Fundamentos de Programação C#

## PROBLEMA 8 — Armazenar e Exibir Notas com Situação

Código do erro:	CS0029, condição de loop, operador ternário
Tipo:	Conversão, lógica, sintaxe de operador ternário
Mensagem do compilador/exceção:	<ul style="list-style-type: none"> <li>• Condição de loop <code>&lt;= notas.Length</code> causará exceção de índice fora do limite</li> <li>• CS0029: Operador <code>=</code> incorreto no operador ternário (deve ser <code>&gt;=</code>)</li> <li>• Lógica de comparação errada para determinar aprovação</li> <li>• <code>IndexOutOfRangeException</code> em potencial</li> </ul>
Linha(s):	2, 7
Hipótese (por que ocorre?):	<ul style="list-style-type: none"> <li>• Confusão entre <code>&lt;=</code> e <code>&lt;</code> em arrays (<code>Length</code> é tamanho, índice vai até <code>Length-1</code>)</li> <li>• Uso de operador de atribuição <code>=</code> em vez de comparação <code>&gt;=</code> no operador ternário</li> <li>• Falta de conhecimento sobre limites de índices em arrays</li> <li>• Confusão entre operadores de atribuição e comparação</li> <li>• Lógica incorreta para determinar aprovação (nota <code>&gt;= 7</code>)</li> </ul>
Experimento (o que alterou/testou):	<p>CÓDIGO COM ERRO: <code>int[] notas = new int[5]; for (int i = 0; i &lt;= notas.Length; i++) { Console.WriteLine(\$"Digite a nota {i + 1}: "); notas[i] = int.Parse(Console.ReadLine()); } for (int i = 0; i &lt; notas.Length; i++) { string situacao = notas[i] == 7 ? "Aprovado" : "Reprovado"; Console.WriteLine(\$"Nota {i + 1}: {notas[i]} - {situacao}"); }</code></p> <p>CORREÇÃO APLICADA: <code>int[] notas = new int[5]; for (int i = 0; i &lt; notas.Length; i++) { Console.WriteLine(\$"Digite a nota {i + 1}: "); notas[i] = int.Parse(Console.ReadLine()); } for (int i = 0; i &lt; notas.Length; i++) { string situacao = notas[i] &gt;= 7 ? "Aprovado" : "Reprovado"; Console.WriteLine(\$"Nota {i + 1}: {notas[i]} - {situacao}"); }</code></p>
Correção final (resumo; sem colar o código todo):	<ol style="list-style-type: none"> <li>1. Alterado <code>&lt;=</code> para <code>&lt;</code> no primeiro loop (evitar índice fora do limite)</li> <li>2. Corrigido operador <code>=</code> para <code>&gt;=</code> na comparação do operador ternário</li> <li>3. Testado funcionamento com notas diferentes (abaixo e acima de 7)</li> <li>4. Verificado que não há exceções de índice</li> </ol>
Teste de regressão (entrada/saída esperada):	<p>Caso 1: Entrada → 8, 6, 9, 5, 7 Saída: "Nota 1: 8 - Aprovado" "Nota 2: 6 - Reprovado" "Nota 3: 9 - Aprovado" "Nota 4: 5 - Reprovado" "Nota 5: 7 - Aprovado"</p> <p>Caso 2: Entrada → 10, 3, 7, 6, 8 Saída: Situações corretas conforme notas <code>&gt;= 7</code></p>
Regra de prevenção:	<ul style="list-style-type: none"> <li>• <code>Array.Length</code> com <code>&lt;</code> (não <code>&lt;=</code>) para evitar índice fora do limite</li> <li>• Usar <code>&gt;=</code> para comparações no operador ternário</li> <li>• Testar operador ternário: condição <code>? verdadeiro : falso</code></li> <li>• Lembrar que índices de array vão de 0 a <code>Length-1</code></li> <li>• Sempre testar com valores limite (nota = 7)</li> </ul>