

FICHA DE ERRO - TRABALHO Fundamentos de Programação C#

PROBLEMA 4 — Nome do Dia da Semana

Código do erro:	CS0103, CS1002, CS0029
Tipo:	Sintaxe, conversão de tipos, lógica
Mensagem do compilador/exceção:	<ul style="list-style-type: none"> • CS0103: Nome 'diaa' não existe no contexto atual • CS1002: ; esperado após Console.WriteLine • CS0029: Não é possível converter implicitamente 'int' para 'string' • Falta de break; em cases do switch • Mistura de tipos int e string nos cases
Linha(s):	2, 3, 6, 8, 9
Hipótese (por que ocorre?):	<ul style="list-style-type: none"> • Erro de digitação no nome da variável (diaa em vez de dia) • Variável declarada como string mas usada como int no switch • Falta de conhecimento sobre sintaxe do switch-case • Inconsistência entre tipos de dados (int vs string) • Cases incompletos e falta de break statements
Experimento (o que alterou/testou):	<p>CÓDIGO COM ERRO: <code>string diaa = Console.ReadLine(); switch (diaa) { case 1: Console.WriteLine("Domingo"); break case "2": Console.WriteLine("Segunda-feira"); } }</code></p> <p>CORREÇÃO APLICADA: <code>int dia = int.Parse(Console.ReadLine()); switch (dia) { case 1: Console.WriteLine("Domingo"); break; case 2: Console.WriteLine("Segunda-feira"); break; // ... demais cases }</code></p>
Correção final (resumo; sem colar o código todo):	<p>1. Alterado tipo da variável de string para int</p> <p>2. Adicionado <code>int.Parse()</code> para conversão</p> <p>3. Corrigido nome da variável (diaa → dia)</p> <p>4. Padronizados todos os cases como int</p> <p>5. Adicionados <code>break;</code> em todos os cases</p> <p>6. Completados todos os 7 dias da semana</p> <p>7. Adicionado ponto e vírgula após <code>Console.WriteLine</code></p>
Teste de regressão (entrada/saída esperada):	<p>Caso 1: Entrada → 1 Saída: "Domingo"</p> <p>Caso 2: Entrada → 7 Saída: "Sábado"</p> <p>Caso 3: Entrada → 8 Saída: "Número inválido"</p>
Regra de prevenção:	<ul style="list-style-type: none"> • Sempre verificar consistência de tipos entre variável e cases do switch • Todo case deve terminar com <code>break;</code> (exceto quando há fall-through intencional) • Verificar nomes de variáveis para evitar erros de digitação • Sempre incluir case default para valores inválidos • Não esquecer ponto e vírgula após statements