

Implementações Corrigidas (com marcação de erros e correções)

Para cada problema abaixo, a implementação está **correta** e contém comentários mostrando a **linha problemática (comentada)** e logo abaixo a **linha corrigida**. As observações usam `// ERRO:` e `// Correto:`.

Problema 1 — Coletor de Dados Pessoais

```
using System;
using System.Globalization;

// ERRO: faltou ';' ao final da linha
// Console.WriteLine("Por favor, digite seu primeiro nome:")
// Correto:
Console.Write("Por favor, digite seu primeiro nome: ");
string primeiroNome = Console.ReadLine() ?? string.Empty;

// ERRO: atribuía string diretamente a int
// int idade = Console.ReadLine();
// Correto:
Console.Write("Agora, digite sua idade: ");
int idade = int.Parse(Console.ReadLine());

// ERRO: poderia falhar por cultura; aceitar ponto como separador
Console.Write("Por fim, digite sua altura em metros (ex: 1.80): ");
float altura = float.Parse(Console.ReadLine(),
CultureInfo.InvariantCulture);

// ERRO: Console.writeline e variável com capitalização incorreta
{PrimeiroNome}
// Console.writeline($"Resumo do Cadastro: Nome: {PrimeiroNome}, Idade:
{idade} anos, Altura: {altura}m.")
// Correto:
Console.WriteLine($"Resumo do Cadastro: Nome: {primeiroNome}, Idade: {idade}
anos, Altura: {altura.ToString("0.00", CultureInfo.InvariantCulture)} m.");
```

Problema 2 — Escolha de Tipos de Dados

```
using System;

// ERRO: literal float sem sufixo e demais tipos/aspas incorretos
// float notaProva = 8.5;
```

```
// int quantidadeAlunos = "42";
// string nomeCurso = 'Fundamentos de C#';
// bool pagamentoEfetuado = "false";
// char digitoVerificador = "7";
// Console.WriteLine(nomeCurso);
// Correto:
double notaProva = 8.5;           // ponto flutuante
int quantidadeAlunos = 42;        // inteiro
string nomeCurso = "Fundamentos de C#";
bool pagamentoEfetuado = false;
char digitoVerificador = '7';

Console.WriteLine(notaProva);
Console.WriteLine(quantidadeAlunos);
Console.WriteLine(nomeCurso);
Console.WriteLine(pagamentoEfetuado);
Console.WriteLine(digitoVerificador);
```

Problema 3 — Verificar Maioridade

```
using System;

Console.WriteLine("Digite sua idade: ");
int idade = int.Parse(Console.ReadLine());

// ERRO: operador e ponto e vírgula após condição
// if (idade => 18);
// Correto:
if (idade >= 18)
{
    Console.WriteLine("Você é maior de idade.");
}
else
{
    Console.WriteLine("Você é menor de idade.");
}
```

Problema 4 — Nome do Dia da Semana

```
using System;

Console.WriteLine("Digite um número de 1 a 7: ");

// ERRO: lia string, usava identificador errado no switch e misturava tipos
// em case
```

```
// string dia = Console.ReadLine();
// switch (diaa) { case 1: ... case "2": ... }
// Correto:
int dia = int.Parse(Console.ReadLine());

switch (dia)
{
    case 1: Console.WriteLine("Domingo"); break;
    case 2: Console.WriteLine("Segunda-feira"); break;
    case 3: Console.WriteLine("Terça-feira"); break;
    case 4: Console.WriteLine("Quarta-feira"); break;
    case 5: Console.WriteLine("Quinta-feira"); break;
    case 6: Console.WriteLine("Sexta-feira"); break;
    case 7: Console.WriteLine("Sábado"); break;
    default: Console.WriteLine("Número inválido"); break;
}
```

Problema 5 — Confirmação de saída (S/N)

```
using System;

string resposta;

do
{
    Console.Write("Deseja sair? (S/N): ");

    // ERRO: atribuía char a string e não tratava nulo/espacos
    // resposta = Console.ReadLine().ToUpper()[0];
    // Correto:
    resposta = (Console.ReadLine() ?? string.Empty).Trim().ToUpper();
}
// ERRO: condição fazia loop infinito (sempre verdadeira)
// while (resposta != "S" || resposta != "N");
// Correto:
while (resposta != "S");

// ERRO: writeline com caixa incorreta
// Console.writeline("Programa encerrado.");
// Correto:
Console.WriteLine("Programa encerrado.");
```

Problema 6 — Economizando até atingir a meta

```
using System;
using System.Globalization;

double total = 0;
while (total < 50)
{
    Console.Write("Digite o valor a ser guardado: ");

    // ERRO: usava string em variável numérica e operador de atribuição
    // incorreto
    // double total = "0";
    // total += valor;
    // Correto:
    double valor = double.Parse(Console.ReadLine()!,
CultureInfo.InvariantCulture);
    total += valor;
}

// ERRO: interpolação sem $ na string
// Console.WriteLine("Meta atingida! Total economizado: R$ {total}");
// Correto:
Console.WriteLine($"Meta atingida! Total economizado: R$
{total.ToString("0.00", CultureInfo.InvariantCulture)}");
```

Problema 7 — Tabuada de um número

```
using System;

Console.Write("Digite um número para ver a tabuada: ");
int num = int.Parse(Console.ReadLine()!);

// ERRO: direção do laço e variável inexistente no produto
// for (int i = 1; i <= 10; i--)
//     Console.WriteLine($"{num} x {i} = {num * j}");
// Correto:
for (int i = 1; i <= 10; i++)
{
    Console.WriteLine($"{num} x {i} = {num * i}");
}
```

Problema 8 — Armazenar e Exibir Notas com Situação

```
using System;

int[] notas = new int[5];

// ERRO: off-by-one (usava <= Length)
// for (int i = 0; i <= notas.Length; i++)
// Correto:
for (int i = 0; i < notas.Length; i++)
{
    Console.Write($"Digite a nota {i + 1}: ");
    notas[i] = int.Parse(Console.ReadLine());
}

for (int i = 0; i < notas.Length; i++)
{
    // ERRO: usava atribuição no ternário (notas[i] = 7 ? ...)
    // string situacao = notas[i] = 7 ? "Aprovado" : "Reprovado";
    // Correto:
    string situacao = notas[i] >= 7 ? "Aprovado" : "Reprovado";
    Console.WriteLine($"Nota {i + 1}: {notas[i]} - {situacao}");
}
```

Problema 9 — Matriz 3×3: diagonais, maiores por linha e transposta

```
using System;

// ERRO: declarou jagged com sintaxe de matriz retangular
// int[][] matriz = new int[3,3];
// Correto:
int[,] matriz = new int[3, 3];

// ERRO: faltou ';'
// int somaTotal = 0
// Correto:
int somaTotal = 0;
int somaDiagonal = 0;
int[] maiores = new int[3];

// ERRO: limites com <= 3 (acessa índice 3)
// for (int i = 0; i <= 3; i++)
//     for (int j = 0; j <= 3; j++)
// Correto:
for (int i = 0; i < 3; i++)
{
```

```

int maxLinha = int.MinValue;
for (int j = 0; j < 3; j++)
{
    Console.WriteLine($"Valor [{i},{j}]: ");
    int valor = int.Parse(Console.ReadLine());

    // ERRO: invertia índices na atribuição
    // matriz[j, i] = valor;
    // Correto:
    matriz[i, j] = valor;

    somaTotal += valor;
    if (i == j)
    {
        // ERRO: somava matriz[i, i+1]
        // somaDiagonal += matriz[i, i + 1];
        // Correto:
        somaDiagonal += matriz[i, j];
    }

    if (valor > maxLinha) maxLinha = valor;
}
maiores[i] = maxLinha;
}

// ERRO: interpolação sem $ e uso de variável fora do escopo
// Console.WriteLine("Soma total: {somaTotal}");
// for (int i = 0; i < 3; i++) Console.WriteLine(maiores[j]);
// Correto:
Console.WriteLine($"Soma total: {somaTotal}");
Console.WriteLine($"Soma diagonal principal: {somaDiagonal}");
Console.WriteLine($"Maiores por linha: {maiores[0]}, {maiores[1]},
{maiores[2]}");

// ERRO: mistura retangular com jagged
// int[,] transposta = new int[3][3];
// Correto:
int[,] transposta = new int[3, 3];

for (int i = 0; i < 3; i++)
{
    for (int j = 0; j < 3; j++)
    {
        // ERRO: copiava sem transpor
        // transposta[i, j] = matriz[i, j];
        // Correto:
        transposta[j, i] = matriz[i, j];
    }
}

Console.WriteLine("Matriz transposta:");

```

```

for (int i = 0; i < 3; i++)
{
    for (int j = 0; j < 3; j++)
    {
        Console.Write(transposta[i, j] + " ");
    }
    // ERRO: faltava ';'
    // Console.WriteLine()
    // Correto:
    Console.WriteLine();
}

```

Problema 10 — Procurar número e exibir posição (3×3)

```

using System;

// ERRO: matriz criada sem dimensionar 2ª dimensão
// int[,] matriz = new int[3];
// Correto:
int[,] matriz = new int[3, 3];

for (int i = 0; i < 3; i++)
{
    for (int j = 0; j < 3; j++)
    {
        Console.Write($"Valor [{i},{j}]: ");
        matriz[i, j] = int.Parse(Console.ReadLine());
    }
}

Console.Write("Número a buscar: ");
// ERRO: faltou ';'
// int buscado = int.Parse(Console.ReadLine())
// Correto:
int buscado = int.Parse(Console.ReadLine());

// ERRO: bool recebido como null
// bool encontrado = null;
// Correto:
bool encontrado = false;
int linha = -1, coluna = -1;

for (int i = 0; i < 3 && !encontrado; i++)
{
    for (int j = 0; j < 3; j++)
    {
        // ERRO: usava atribuição dentro do if e invertia linha/coluna
        // if (matriz[i, j] = buscado) { encontrado == true; linha = j;

```

```

coluna = i; }
    // Correto:
    if (matriz[i, j] == buscado)
    {
        encontrado = true;
        linha = i;
        coluna = j;
        break;
    }
}

// ERRO: "=" em vez de "==" e Console.WriteLine
// if (encontrado = true) { Console.WriteLine($"Encontrado em ({linha + 1},
{coluna + 1})") }
// Correto:
if (encontrado)
{
    Console.WriteLine($"Encontrado em ({linha + 1}, {coluna + 1})");
}
else
{
    // ERRO: interpolação sem $
    // Console.WriteLine("Número não encontrado: {buscado}");
    // Correto:
    Console.WriteLine($"Número não encontrado: {buscado}");
}

```