

# CLÍNICA DE BUGS – C# ESTRUTURADO

Nesta atividade colaborativa, você receberá pequenos trechos de código em C# (modo estruturado) com **erros propositalmente inseridos**. Seu trabalho é **reproduzir o ambiente, executar o código, ler as mensagens do compilador/runtime, formular hipóteses, testar correções e justificar** cada ajuste feito até alcançar o comportamento esperado.

## PROBLEMA 1 – CADASTRO SIMPLES

### ENUNCIADO (O QUE RESOLVER)

**Declare quatro variáveis** para armazenar as seguintes informações de uma pessoa: o **nome completo**, a **idade**, a **letra inicial do nome** e o **salário**. **Inicialize-as** com dados de exemplo e, em seguida, **exiba cada uma** dessas informações no console.

Saída esperada (conteúdo equivalente, formatação livre):  
Nome completo, idade (numérica), uma letra, e salário como número (preferencialmente com duas casas).

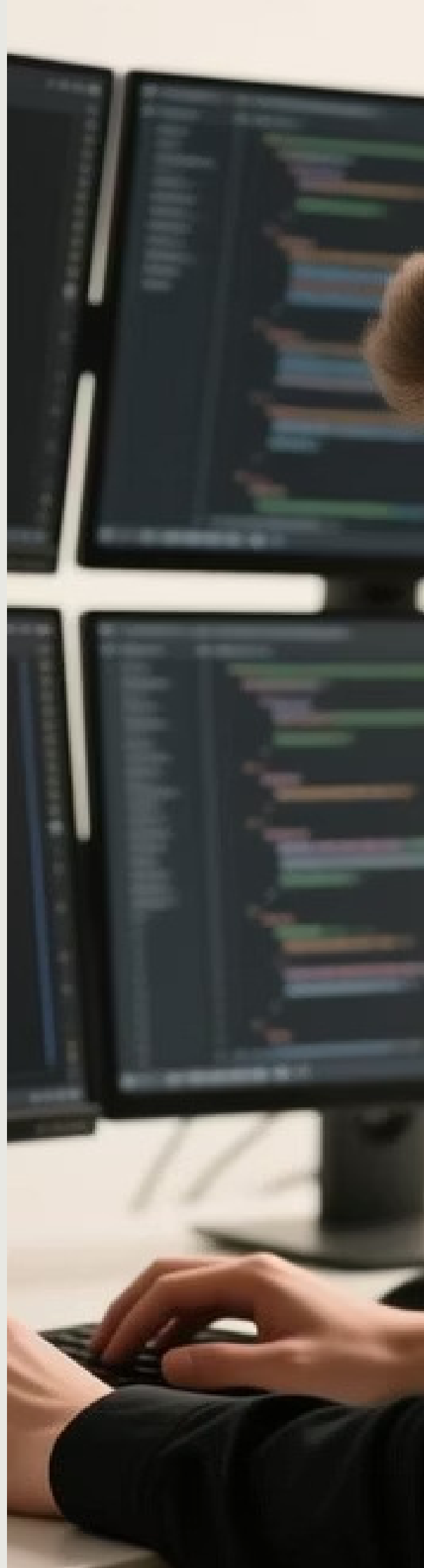
### OBJETIVOS PARA OS(AS) ESTUDANTES

Consolidar a transição C → C# pela **leitura ativa de mensagens** do compilador (CSxxxx) e de exceções comuns.

Diferenciar **erros sintáticos**, **erros em tempo de execução** e **erros lógicos/semânticos**.

Praticar **método de depuração**: hipótese → experimento → evidência → correção → prevenção.

Produzir **código comentado com propósito** (comentários explicam o *porquê*, não o óbvio).





# "RESOLUÇÃO" FORNECIDA COM ERROS (PARA DEPURAÇÃO)

⚠ **Atenção:** este código **tem muitos erros** de tipos diferentes. **Não comece corrigindo tudo de uma vez.** Rode, leia as mensagens e avance por hipóteses.

⚠ Código propositalmente defeituoso para o Problema 1

```
string nomeCompleto = "Maria Silva" // (E1) falta ';'
int idade = "35"; // (E2) conversão implícita inválida string→int (CS0029)
char letraInicial = "M"; // (E3) uso de aspas duplas para char (CS0029)
double salario = 4_550,75; // (E4) literal numérico inválido com vírgula (CS1003/CS1525)

Console.WriteLine("Dados do Funcionário:"); // (E5) identificador incorreto: 'Consol' (CS0103)
Console.WriteLine(nomeCompleto); // (E6) variável inexistente (case-sensitive) (CS0103)
Console.WriteLine(idade); // (E7) bloqueado pelos erros acima
Console.WriteLine(letraInicial); // (E8) idem
Console.WriteLine("Salário: R$ {salario}"); // (E9) falta '$' na interpolação → erro lógico (apenas imprime {salario})

Console.WriteLine("Cálculo estranho: " + (salario / letraInicial)); // (E10) divisão double/char → semântica absurda
if (idade = 35) { // (E11) atribuição em vez de comparação (CS0029)
    Console.WriteLine("Idade confirmada");
}

// (E12) possível confusão de cultura ao tentar formatar em outra linha:
Console.WriteLine(salario.ToString("C2",
    System.Globalization.CultureInfo.GetCultureInfo("en-US"))); // sem valor pedagógico aqui para pt-BR
```

# PISTAS E ORIENTAÇÕES (NÃO ENTREGUE CÓDIGO PRONTO)

## ERROS DE SINTAXE

- **E1 (falta ';'):** C# exige ; após cada instrução. *Mensagem típica: CS1002 "; esperado".*
- **E4 (literal double):** vírgula não separa decimais em literal C#. Use ponto: 4550.75 (a cultura afeta **formatação**, não o literal). *Mensagem: CS1003/CS1525.*
- **E5 (Console):** nome correto é Console. *Mensagem: CS0103 "nome não existe no contexto atual".*

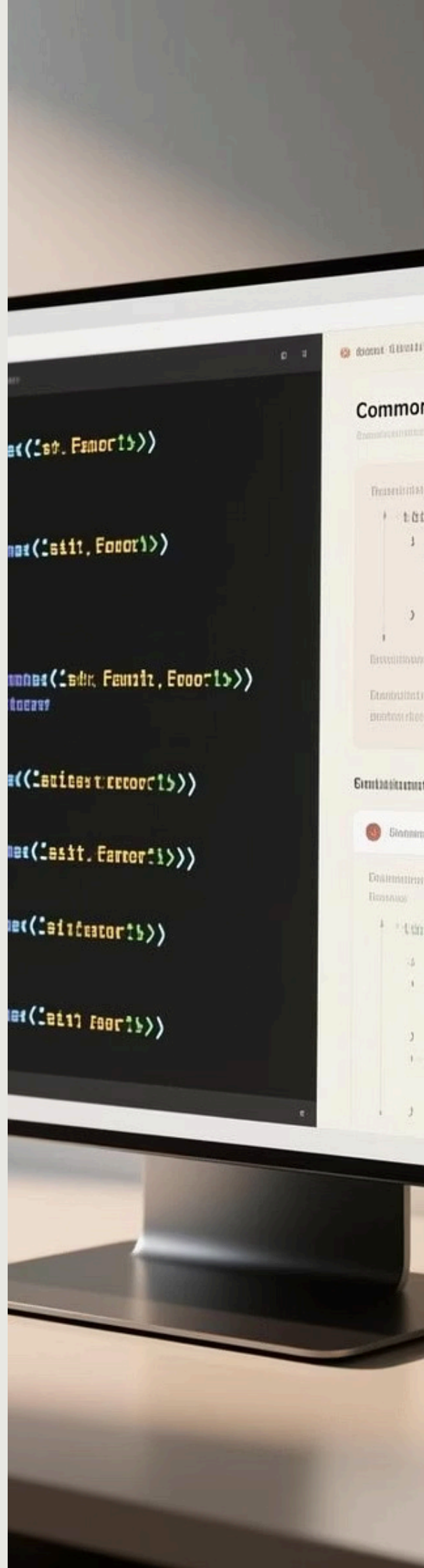
## ERROS DE TIPAGEM

- **E2 (tipagem):** "35" é string. Converta com `int.Parse`, `Convert.ToInt32` ou `int.TryParse`. *Mensagem: CS0029.*
- **E3 (char):** char usa **aspas simples**. Ex.: 'M'. *Mensagem: CS0029.*
- **E6 (nome da variável):** C# é *case-sensitive*. Use exatamente `nomeCompleto`. *Mensagem: CS0103.*

## ERROS LÓGICOS/SEMÂNTICOS

- **E9 (interpolação):** faltou \$. Sem ele, **não** insere o valor. Isso é **erro lógico** (compila, mas mostra `{salario}`).
- **E10 (semântica):** dividir salário por um char não faz sentido de domínio; char vira código numérico. Explique **por que** isso é incorreto e remova.
- **E11 (comparação):** em C# `if (idade == 35)`. `=` é atribuição – além disso, `if` espera `bool`. *Mensagem: CS0029.*
- **E12 (formatação de moeda):** se quiser moeda pt-BR, use `CultureInfo("pt-BR")`. Caso contrário, explique a escolha.

Dica geral: **corrija em camadas** – 1) sintaxe, 2) tipos, 3) semântica/lógica, 4) apresentação.



# FICHA DE ERROS (PREENCHA UMA POR ERRO)

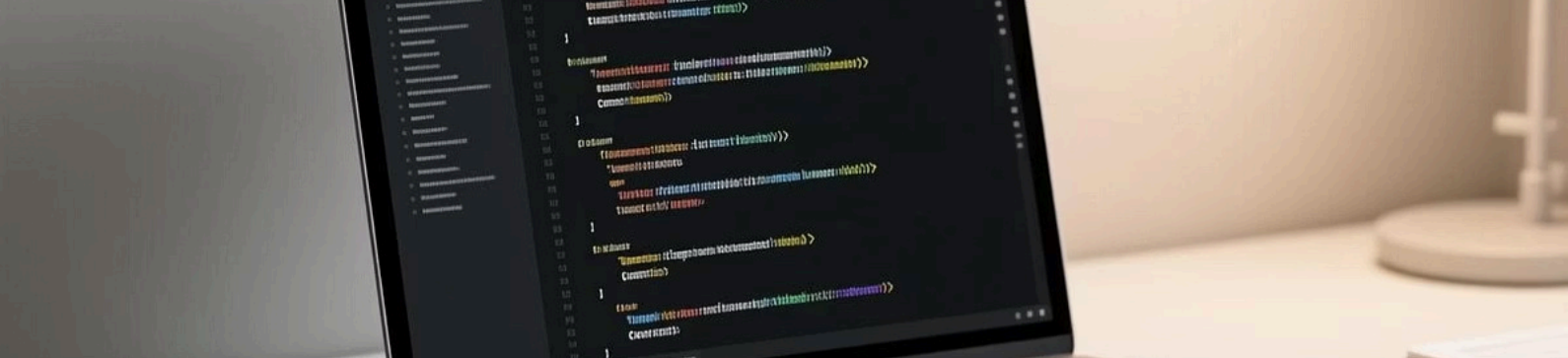
Código do erro:	E1/E2/... (e CSxxxx se houver)
Tipo:	Sintático / Execução / Lógico / Semântico
Mensagem do compilador/exceção:	
Linha(s):	
Hipótese (por que ocorre?):	
Experimento (o que alterou/testou?):	
Correção final (resumo; sem colar o código todo):	
Teste de regressão (entrada/saída esperada):	
Regra de prevenção:	

## MAPA RÁPIDO DE CÓDIGOS ÚTEIS (PARA CONSULTA)

CS1002 ; esperado – sintaxe	CS0103 nome não existe no contexto – escopo/typo
CS0029 conversão implícita inválida – tipos	CS1525/CS1003 token/termo inválido – sintaxe geral

# TABELA DE TESTES (MÍN. 3 CASOS)

Caso	Entradas simuladas	O que observar	Saída esperada
1	Nome "Maria Silva"	Ordem das impressões	Linhas com nome, idade, letra, salário
2	Idade 35	Tipagem correta	Número sem aspas
3	Salário 4550.75	Formatação e precisão	Número legível (ideal: 2 casas, opcional "R\$")



# COMENTÁRIOS (CLEAN CODE) – O QUE PEDIR AOS ALUNOS

1

## EXPLIQUE O PORQUÊ

Explique *por que* certas escolhas foram feitas (ex.: "formatamos com pt-BR para aderir ao contexto local"), **não** descreva o óbvio ("incrementa i").

2

## EVITE COMENTÁRIOS QUE APODRECEM

Evite comentários que **apodrecem** (ex.: repetir valor que o código já expressa).

3

## PREFIRA NOMES CLAROS

Prefira **nomes claros** às explicações longas; comentários complementam, não salvam maus nomes.

4

## CABEÇALHO CURTO

**Cabeçalho curto** no arquivo: propósito, entradas/saídas esperadas, e **principais riscos** (ex.: cultura na formatação de números).

## O QUE FAZER

- Comentar decisões não óbvias
- Explicar o "porquê" das escolhas
- Documentar riscos e limitações
- Usar comentários para clarificar intenção

## O QUE EVITAR

- Comentar o óbvio ("incrementa contador")
- Repetir o que o código já diz
- Comentários desatualizados
- Usar comentários para compensar código ruim

# RESUMO DE APRENDIZAGEM (OBRIGATÓRIO, 5-8 LINHAS)

Escreva **com suas palavras**:

O que você aprendeu sobre **C#** (tipagem, literais, interpolação, comparação).

O que você aprendeu sobre o **propósito** do problema (representar dados simples de uma pessoa e exibi-los corretamente).

O **erro mais interessante** que encontrou e como **diagnosticou**.

Uma **regra de prevenção** que você levará para os próximos exercícios.

