

Clínica-De-Bugs(POO: Programação Orientada a Objetos).

Implementação contendo erros, fornecida pelo professor para ser corrigida.

Atividade 06: Economizando_Ate_Atingir_Meta.

```
CSharp
double total = "0";
while (total < 50)
{
    Console.WriteLine("Digite o valor a ser guardado: ");
    double valor = double.Parse(Console.ReadLine().Replace('.', ','));
    total += valor;
}
Console.WriteLine("Meta atingida! Total economizado: R$ {total}");
```

Implementação corrigida, contendo comentários para marcar erros e correções

```
CSharp
double total = 0; // Valores atribuídos a esta variável devem ser do tipo
double.
while (total < 50)
{
    Console.WriteLine("Digite o valor a ser guardado: ");
    string? input = Console.ReadLine();
    if (input is not null) // Verifica se a entrada não é nula antes de
    tentar usá-la.
    {
        double valor = double.Parse(input.Replace('.', ','));
        total += valor; // Adiciona o valor digitado ao total, e não o
        substitui.
    }
}
Console.WriteLine($"Meta atingida! Total economizado: R$ {total}"); //
Deve-se utilizar o símbolo $ para interpolação de strings.
```

Ficha de Erros

Problema nº	ERR - 001	ERR - 002	ERR - 003	ERR - 004
Projeto/Arquivo:	Program.cs	Program.cs	Program.cs	Program.cs
Linha(s) Afetada(s):	Linha 01, Col 16	Linha 05, Col 33	Linha 06, Col 11	Linha 08, Col 18
Tipo:	Sintático	Execução	Lógico	Semântico
Mensagem do compilador/exceção:	error CS0029: Não é possível converter implicitamente tipo "string" em "double"	warning CS8602: Desreferência de uma referência possivelmente nula.		
Hipótese (por que ocorre?):	Porque o valor atribuído a uma variável deve ser equivalente ao tipo dessa variável.	Porque existe a possibilidade da variável ser nula em tempo de execução.	Porque o operador “+=” apenas substitui o valor da variável <i>total</i> em vez adicionar.	É necessário o símbolo “\$” para apresentar o valor da variável <i>total</i> .
Experimento (o que testei?):	Remoção das aspas do valor atribuído à variável.	Existem várias formas de solucionar este erro. Foi adicionado uma condição if-else para verificação do estado da variável.	O “+=” foi substituído por “+=”.	Adição do símbolo “\$”.

Descrição da correção:	Atribui um valor válido para variável do tipo <i>double</i> .	Verifica o estado da variável inserida pelo usuário e apenas executa a lógica do código caso a entrada não seja nula.	Garante que o valor de entrada do usuário será adicionado ao valor total.	Garante que o valor apresentado ao usuário será o da variável <i>total</i> .
Teste de regressão (entradas/saídas esperadas):		Entrada: 50 → Saída Esperada: "Meta atingida! Total economizado: R\$ 50"	Entrada: 50 → Saída Esperada: "Meta atingida! Total economizado: R\$ 50"	Entrada: 50 → Saída Esperada: "Meta atingida! Total economizado: R\$ 50"
Regra/checklist:	Sempre verificar a tipagem das variáveis antes de fazer qualquer atribuição.	Sempre verificar o estado de variáveis possivelmente nulas.	Garantir que o código está funcionando da maneira e lógica correta.	Sempre verificar erros de semântica.

MAPA RÁPIDO DE CÓDIGOS ÚTEIS (PARA CONSULTA)

CS0029 tipagem errada

CS8602 variável pode ser nula

TABELA DE TESTES			
Caso	Entradas simuladas	O que observar	Saída esperada
1	'palavra'	Tipagem correta de variáveis	The input string 'palavra' was not in a correct format.
2	50	Limite estabelecido no loop while.	Meta atingida! Total economizado: R\$ 50
3	49	Funcionamento do loop.	Digite o valor a ser guardado:

Resumo de Aprendizagem

O exercício foi um ótimo exemplo de aprendizagem dos conceitos de tipagem de variáveis e de tratamento de variáveis possivelmente nulas. Foi possível perceber como é importante entender a tipagem forte da linguagem e a comparação correta entre variáveis. O propósito do problema, que tinha como objetivo apresentar o valor acumulado das entradas inseridas pelo usuário quando acima de R\$50, possibilitou também ficar atento a possíveis erros de lógica e semântica. O erro mais interessante foi o de variável possivelmente nula, que tem uma gama grande de conteúdo de soluções e de motivos para o erro, tanto na documentação oficial quanto na comunidade de programação.

Registro de uso de IA

A I.A. foi utilizada principalmente para identificação dos tipos de erros (se é Sintático, de Execução, Lógico ou Semântico). Ex.: “total += valor em vez de total += valor é que tipo de erro?”

<3