

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

Implementação contendo erros, fornecida pelo professor para ser corrigida.

Atividade 08: Armazenar_Exibir_Notas_Situacao.

```
CSharp
int[] notas = new int[5];

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++)
{
    string situacao = notas[i] >= 7 ? "Aprovado": "Reprovado";
    Console.WriteLine($"Nota (i + 1): {notas[i]} - {situacao}");
}
```

Implementação corrigida, contendo comentários para marcar erros e correções

```
CSharp
int[] notas = new int[5];

for (int i = 0; i < notas.Length; i++) // O loop deve iterar até o tamanho
do array.
{
    Console.Write($"Digite a nota {i + 1}: ");
    string? input = Console.ReadLine();
    if (input is null) // Verifica se a entrada não é nula antes de tentar
usá-la.
    {
        Console.WriteLine("Entrada inválida.");
        return;
    }
    notas[i] = int.Parse(input);
}
```

```

for (int i = 0; i < notas.Length; i++)
{
    string situacao = notas[i] >= 7 ? "Aprovado": "Reprovado"; // Corrigido
    o operador ternário para refletir a lógica correta.
    Console.WriteLine($"Nota (i + 1): {notas[i]} - {situacao}");
}

```

Ficha de Erros			
Problema nº	ERR - 001	ERR - 002	ERR - 003
Projeto/Arquivo:	Program.cs	Program.cs	Program.cs
Linha(s) Afetada(s):	Linha 11, Col 34	Linha 06, Col 26	Linha 03, Col 18
Tipo:	Sintático/Lógico	Execução	Execução
Mensagem do compilador/exceção:	error CS0029: Não é possível converter implicitamente tipo "int" em "bool"	warning CS8604: Possível argumento de referência nula para o parâmetro 's' em 'int.Parse(string s)'.	
Hipótese (por que ocorre?):	O operador de comparação do bloco deve ser >= e não =.	Porque existe a possibilidade da variável ser nula em tempo de execução.	A condição do loop estoura o tamanho do vetor

Experimento (o que testei?):	Modificado = para \geq .	Existem várias formas de solucionar este erro. Foi adicionado uma condição if-else para verificação do estado da variável.	Modificado \leq para $<$.
Descrição da correção:	Modificado = para \geq .	Verifica o estado da variável inserida pelo usuário e apenas executa a lógica do código caso a entrada não seja nula.	Modificado \leq para $<$, assim interrompendo o loop antes de estourar o tamanho do vetor.
Teste de regressão (entradas/saídas esperadas):	Entrada: 1 → Saída Esperada: "Nota (1): 1 - Reprovado"	Entrada: 7 → Saída Esperada: "Nota (1): 7 - Aprovado"	Entrada: 8 → Saída Esperada: "Nota (1): 8 - Aprovado"
Regra/checklist:	Sempre verificar lógica de comparação.	Sempre verificar o estado de variáveis possivelmente nulas.	Sempre se atentar ao tamanho do vetor e à lógica do loop.

MAPA RÁPIDO DE CÓDIGOS ÚTEIS (PARA CONSULTA)

CS0029 tipagem errada

CS8604 variável pode ser nula

TABELA DE TESTES

Caso	Entradas simuladas	O que observar	Saída esperada
1	1	Limite estabelecido pela condição if-else	Reprovado
2	7	Limite estabelecido pela condição if-else	Aprovado
3	8	Limite estabelecido pela condição if-else	Aprovado

Resumo de Aprendizagem

O exercício permitiu o reforço de conceitos de lógica condicional por meio do operador ternário, deixando claro a diferença entre os operadores lógicos. Também permitiu o reforço do conhecimento em relação à declaração de variáveis possivelmente nuláveis, e trouxe um maior entendimento quanto a tamanho de vetores e como declarar e operar com vetores dentro de loops condicionais. O erro mais interessante foi o erro dentro do operador ternário.

Registro de uso de IA

A I.A. foi utilizada principalmente para transcrever o texto do pdf com os exercícios, pois não é possível copiar diretamente do arquivo disponibilizado no GitHub. Ex. de prompt: “copia e cola exatamente, sem mudar nada e sem corrigir”.