

# Lógica de Programação

Professor: Emerson Camargo

# Agenda

- Dicas

## USING

A using instrução obtém um ou mais recursos, executa uma instrução e, em seguida, descarta o recurso.

## USING

Crie um novo aplicativo console C#; criará o Hello World

***Console.WriteLine("Hello, World!");***

## USING

Digite uma nova linha com a instrução USING, chamando as classes System e Console, pronto agora você poderá digitar as instruções sem chamar a classe Console em seu algoritmo.

```
using static System.Console;  
WriteLine("Hello, World!");
```

## Variáveis na mesma linha.

Você pode declarar uma ou mais variáveis do mesmo tipo em uma mesma linha. Ao invés de declarar assim;

```
int n1 = 10;
```

```
int n2 = 15;
```

```
int n3 = 25;
```

**Variáveis na mesma linha.**

Você pode simplesmente fazer assim, separando por vírgula;

```
int n1=10, n2=15, n3=25;
```

## **Declaração implícita de variáveis.**

Quando aprendemos a programar, uma das primeiras coisas que aprendemos é declaração de variáveis e os tipos de variáveis que podemos declarar, sendo que as mais comuns são o int, double e string. Mas existe um tipo de variável que é genérica, ou seja, que aceita praticamente qualquer tipo de valor, o valor pode ser tanto numérico (int ou double) como literal (string).



## **Declaração implícita de variáveis.**

Este tipo de variável é a var, ao invés de declarar as variáveis assim;

**string nome = "Anderson";**

**int idade = 18;**

**double salario = 1250.79;**

**Declaração implícita de variáveis.**  
podemos simplesmente declarar assim;

```
var nome = "Anderson";  
var idade = 18;  
var salario = 1250.79;
```

## **Declaração implícita de variáveis.**

O uso das aspas é suficiente para indicar ao var que o valor da variável é uma string, o uso do ponto flutuante indica que se trata de um double, e um valor numérico inteiro fora das aspas indica que se trata de um inteiro.

## Declaração de valores

***// forma errada de usar o var com ReadLine()***

```
var numero = Console.ReadLine();
```

Prefira;

***// forma correta de usar o var com ReadLine()***

```
var numero = 0;
```

```
numero = Console.ReadLine();
```

## Interpolação de strings

Usando concatenação

```
string nome = "Anderson";  
int idade = 23;  
Console.WriteLine("Olá " + nome + ", você tem " + idade  
+ " anos.");
```

## Interpolação de strings

Para usarmos interpolação precisamos apenas adicionar um cifrão no início da string e substituir o sinal de + por um jogo de chaves, ficando assim;

```
Console.WriteLine($"Olá {nome}, você tem  
aproximadamente {idade} anos.");
```

## Constantes

Em algumas aplicações temos variáveis cujo o valor nunca muda, para elas podemos acrescentar a declaração **const** que nos permite organizar melhor o nosso programa. É importante entender que declararmos o **const**, a variável não poderá mais receber um novo valor, ou o programa retornará erro.

## Constantes

Veja um exemplo;

```
string nome = "Anderson";  
const int anoAtual = 2021; // este valor sempre será fixo  
int anoNascimento = 1984;
```



## Exemplo de Programa

### - Programa 5

**Sistema**  
**Fiep**  
nosso i é de indústria.

*FIEP*

*SESI*

*SENAI*

*IEL*