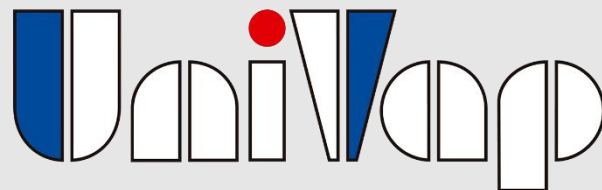


MANIPULAÇÃO DE STRINGS C#



Universidade do Vale do Paraíba

PROF. ME.
HÉLIO
ESPERIDIÃO



Faculdade de Engenharias
Arquitetura e Urbanismo

SUBSTRING

- Retorna uma parte da string.
- Inicia a partir de um determinado número de caracteres e vai até um limite N

```
string nome = "Hélio Esperidião";  
string sobreNome = nome.Substring(6, 10);  
  
MessageBox.Show(sobreNome);
```

Início

Término

- O retorno do exemplo acima é “**Esperidião**”

O MÉTODO INDEXOF

- O método IndexOf é utilizado para localizar uma determinada palavra dentro da string.
- Este método retornará a posição da string desejada.
- Caso a string não seja encontrada, será retornado o valor -1.

```
string nome = "Hélio Esperidião";  
int pos = nome.IndexOf("Esperidião");  
  
MessageBox.Show(pos.ToString());
```

TOUPPER E TOLOWER

- As funções ToUpper e ToLower permitem colocar uma string em letra minúsculas ou maiúsculas, conforme o exemplo a seguir.

```
string nome = "Maurício";  
nome = nome.ToUpper();  
MessageBox.Show(nome);  
  
nome = nome.ToLower();  
MessageBox.Show(nome);
```

STARTSWITH E ENDSWITH

- As funções startswith e endswith verificam se a string começa ou termina com uma determinada palavra ou caráter.
- Estas funções retornam um valor booleano (true ou false).

```
string arquivo = "Hélio.dll";

if (arquivo.StartsWith("Hélio"))
{
    MessageBox.Show("Começa com Hélio!");
}

if (arquivo.EndsWith(".dll")){
    MessageBox.Show(("É uma dll!"));
}
```

TRIMSTART, TRIMEND E TRIM

- As funções de Trim servem para remover espaços em branco das strings.
- A função TrimStart remove os espaços em branco do início da string.
- A função TrimEnd remove os espaços em branco do final da string.
- A função Trim remove os espaços em branco do início e do fim da string.

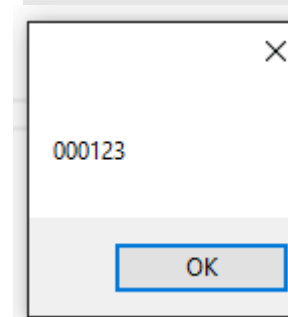
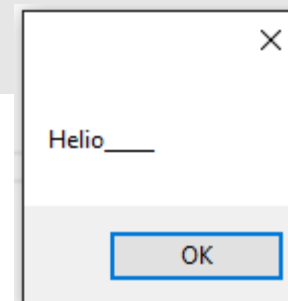
TRIMSTART, TRIMEND E TRIM

```
string nome = "  MAURICIO  ";  
nome = nome.TrimEnd();  
MessageBox.Show(nome);  
nome = nome.TrimStart();  
MessageBox.Show(nome);  
nome = nome.Trim();  
MessageBox.Show(nome);
```

PADLEFT E PADRIGHT

- As funções PadLeft e PadRight servem para preencher uma string a esquerda ou a direita com um caractere específico.

```
string nome = "Hélio";  
  
nome = nome.PadRight(10, '_');  
  
MessageBox.Show(nome);  
  
string codigo = "123";  
codigo = codigo.PadLeft(6, '0');  
  
MessageBox.Show(codigo);
```



STRING.JOIN E STRING.SPLIT

- A função Split serve para quebrar uma string em um array de strings de acordo com um separador.
- Ao contrário da função split, a função Join concatena um array de string inserindo um separador.

```
string linha = "Hélio, Engenheiro, Professor";  
  
string[] campos = linha.Split(',');  
MessageBox.Show(campos[0]);  
  
string linhaNova = String.Join(";", campos);  
  
MessageBox.Show(linhaNova);
```

STRING.FORMAT

```
int idade=17;  
string teste = String.Format("Vou colocar o {0} aqui.", idade);  
MessageBox.Show(teste);
```

- No exemplo entre {0} é o primeiro índice que será inserido. Este token indica que este valor deve ser substituído pelo segundo parâmetro passado para a função (neste caso a variável idade).

LETRA POR LETRA

- Para analisar letra por letra de uma string utilizamos o comando abaixo:
- Entende-se como:
 - Para cada caractere da string nome:

```
string nome = "Hélio";  
foreach (char letra in nome)  
{  
    MessageBox.Show(letra.ToString());  
}
```