

# LP2 – Aula 5

## TEÓRICA

Profª Mª Denilce Veloso

- [denilce.veloso@fatec.sp.gov.br](mailto:denilce.veloso@fatec.sp.gov.br)
  - [denilce@gmail.com](mailto:denilce@gmail.com)

# C# .NET - Operadores e funções básicas

**Tabela 1: Operadores Aritméticos e de Concatenação de Strings**

Operador	Utilização	Exemplo
=	Atribui um valor a uma variável ou constantes	y = 10; ou sobrenome = "Gates";
+	Soma dois valores	y=10; x = y + 5; ( o valor de x será igual a 15)
-	Subtrai um valor de outro	y=10; x = y - 2; ( o valor de x será igual a 8)
*	Multiplica dois valores	y=10; x = y * 2; ( o valor de x será igual a 20)

# C# .NET - Operadores e funções básicas

/	<p>Divide um valor por outro ** resultado depende do tipo da variável Se for inteiro, volta parte inteira</p>	<pre>int y = 10; int x; x = y / 2; // ( o valor de x será igual a 5)  y = 7; x = y / 2; // ( o valor de x será igual a 3)  y = 9; x = y / 5; // ( o valor de x será igual a 1)  double z = 7; double w; w = z / 2; // ( o valor de x será igual a 3,5)</pre>
%	<p>Divide um valor por outro e retorna o resto da operação</p>	<pre>int y=7; int x = y % 3; ( o valor de x será igual a 1) double x = 8.5; double y; y = x % 2; ( o valor de y será igual a 0,5)</pre>
+	<p>Combina (concatena) duas strings</p>	<pre>string sobrenome="Gates"; cliente = "Bill " + sobrenome; // ( cliente será igual a Bill Gates)</pre>

# C# .NET - Operadores e funções básicas

+=	soma o valor e atribui o resultado	y=10; y += 3; //( y será igual a 13)
-=	subtrai o valor e atribui o resultado	y=10; y -= 6; //( y será igual a 4)
*=	Multiplica o valor e atribui o resultado	y=10; y *= 2; //( y será igual a 20)
/=	Divide o valor e atribuiu o resultado	int y=10; y /= 2 ; //( y será igual a 5) double y=10; y /= 3; //( y será igual a 3,3) double y=9.8; y /=3; //( y será igual a 3,268)
+=	Concatena a string e atribui o resultado se ambos os operadores forem strings	sobrenome += ", Denilce" ; //( sobrenome será igual a "Veloso , Denilce"

# C# .NET - Operadores e funções básicas

++	O operador de incremento incrementa seu operando em 1.	(++) double x= 1.5; MessageBox.Show((++x).ToString()); ➔ Imprime 2,5 MessageBox.Show((x++).ToString()); ➔ <b>imprime 2,5</b> MessageBox.Show(x.ToString()); ➔ Imprime 3,5
--	O operador de decremento decrementa seu operando em 1.	(--) double y = 1.5; MessageBox.Show(--y.ToString()); ➔ Imprime 0,5 MessageBox.Show((y--).ToString()); ➔ <b>Imprime 0,5</b> MessageBox.Show(y.ToString()); ➔ Imprime -0,5

# C# .NET – Classe Math (métodos)

Pow	Eleva um valor a um determinado expoente	<pre>double d=3; d = Math.Pow(d, 3); // ( 3 elevado a 3 ; resultado = 27 )</pre>
Sqrt	Extraí a raiz quadrada de um valor (Namespace System classe Math)	<pre>x = Math.Sqrt(81); // ( O valor de x será igual a 9)</pre>
Round	Volta o número arredondado	<pre>double z1 = 3.5; z1 = Math.Round(z1); // (o valor de z1 será 4)</pre>

*Obs: A classe Math do namespace System possui diversos métodos que oferecem um suporte matemático. (Ex: Atan , Exp , Sqrt , Pow , Min , Round , Abs , Cos , Sin , Tan , etc...)*

# C# .NET - Operadores e funções básicas

**Tabela 2: Operadores de Comparação**

Operador	Utilização	Exemplo
>	Maior	if (x >y)
<	Menor	if (x< y)
<=	Menor ou Igual	if (x <=y)
>=	Maior ou Igual	if (x >=y)
==	Igual	if (x == y)
!=	Diferente	if (x != y)

# C# .NET - Operadores e funções básicas

**Tabela 3: Operadores Lógicos**

Operador	Utilização	Exemplo
&&	Retorna true se ambas as expressões forem verdadeiras(and)	if ((x==1) && (y>5))
	Retorna true se uma das expressões for verdadeira(or)	if ((x==1)    (y>5))
!	Retorna o inverso da expressão(not)	if (!(x==1))
^	Ou exclusivo, retorna true se apenas uma das expressões for verdadeira, mas NÃO SE AMBAS FOREM VERDADEIRAS(xor)	if ((x==1) ^ (y>5))
&	Similar ao &&, porém o operador && sempre avalia as duas condições. Por outro lado, o operador & não avalia a segunda condição se o valor da primeira condição for false.	if ((x==1) & (y>5))
	Similar ao   , porém o operador    sempre avalia as duas condições. Por outro lado, o operador   avalia a segunda condição se o valor da primeira condição for false.	if ((x==1)   (y>5))



# C# .NET - Operadores e funções básicas

**Tabela 4: Outros Operadores**

Operador	Utilização	Exemplo
Is	Retorna true se objeto pode ser convertido no tipo	if (btn1 is Button) → true if (btn1 is TextBox) → false

# C# .NET - Operadores e funções básicas

Tabela 5: Outros Precedências

Precedência	Operador	Operação	Desempate
1	++ --	Pós incremento Pós decremento	
2	++ -- + - ! (tipo)	Pré incremento Pré decremento Mais unário Menos unário Negação Casting	Direita → Esquerda
3	* / %	Multiplicação Divisão Módulo	Esquerda → Direita
4	+ -	Adição Subtração	Esquerda → Direita
5	< <= > >=	Menor Menor ou igual Maior Maior ou igual	Esquerda → Direita
6	== !=	Igualdade Diferença	Esquerda → Direita
7	&	E simples	Esquerda → Direita
8	^	Ou exclusivo	Esquerda → Direita
9		Ou simples	Esquerda → Direita
10	&&	E duplo	Esquerda → Direita
11		Ou duplo	Esquerda → Direita
12	? :	Ternário	Direita → Esquerda
13	= += -= *= /= %=	Atribuição Incremental Decremental Multiplicativa Divisória Modular	Direita → Esquerda

## Slide 10

---

**DV10**

Console.WriteLine(+4); // output: 4

Console.WriteLine(-4); // output: -4

Console.WriteLine(--(-4)); // output: 4 --> 4

DENILCE VELOSO; 29/03/2021

**DV12**

**\*\* UNÁRIOS**

DENILCE VELOSO; 30/04/2022

**DV11**

DENILCE VELOSO; 30/04/2022

# C# .NET - Operadores e funções básicas

## →EXERCÍCIOS

1) Dados os valores das variáveis tipo int: **varX = 2**,  
**varA = -5**, **varB = 3**, **varC = 9** e **varD = 6**. Determine o resultado lógico das expressões mencionadas abaixo, assinalando se são Verdadeiras ou Falsas.

if (!( varX > 3))	- (   ) Verdadeiro	(   ) - Falso
if (( varX < 1 ) && ( varB >= varD))	- (   ) Verdadeiro	(   ) - Falso
if (!( varD > 3)    !( varB < 7))	- (   ) Verdadeiro	(   ) - Falso
if (( varA > varB )    !( varC > varB))	- (   ) Verdadeiro	(   ) - Falso
if (( varX >= 2 ) && ( varX < 7))	- (   ) Verdadeiro	(   ) - Falso
if (( varA < varB ) ^ ( varC > varB))	- (   ) Verdadeiro	(   ) - Falso

# C# .NET - Operadores e funções básicas

## → EXERCÍCIOS

2) Qual o resultado da expressão?

```
int z = 15;  
double y = 5;
```

```
double x = (((1 + ((7 * 3) / 3)) * 10) / ((double)(z) - y));  
MessageBox.Show(x.ToString());
```

# C# .NET - Operadores e funções básicas

## →EXERCÍCIOS

3) Se construída uma aplicação que tenha apenas um botão e qual será o valor final de N na seguinte operação?

```
private Button1_Click(object sender,
System.EventArgs e) {
    int N = 55;
    int B = 7;
    N /= 5;
    B -= 1;
    N = N % B;
    MessageBox.Show("N = " +
Convert.ToString(N));
}
```

# C# .NET - Operadores e funções básicas

## →EXERCÍCIOS

4) Quais os resultados que serão impressos depois de executado o código?

```
private Button1_Click(object sender, EventArgs e) {  
    decimal Numero = 1234;  
    decimal D1, D2, D3, D4;  
    D4 = Numero % 10;  
    D3 = (Numero / 10) % 10;  
    D2 = (Numero / 100) % 10;  
    D1 = (Numero / 1000) % 10;  
    MessageBox.Show("D1 =" + Convert.ToString(D1) +  
        " D2 = " + Convert.ToString(D2) +  
        " D3 =" + Convert.ToString(D3) +  
        "D4 =" + Convert.ToString(D4));  
}
```

# C# .NET - Operadores e funções básicas

## → EXERCÍCIOS

5) Quais os resultados que serão impressos depois de executado o código?

```
private Button1_Click(object sender, System.EventArgs e) {  
    int Numero = 1234;  
    int D1, D2, D3, D4;  
    D4 = Numero % 10;  
    D3 = (Numero / 10) % 10;  
    D2 = (Numero / 100) % 10;  
    D1 = (Numero / 1000) % 10;  
  
    MessageBox.Show("D1 =" + Convert.ToString(D1) + " D2 ="  
    + Convert.ToString(D2) + " D3 =" + Convert.ToString(D3) + "D4"  
    + Convert.ToString(D4));  
}
```