

# Operadores

## Linguagem C#

## Operadores

Podem ser definidos na linguagem C# da seguinte forma:

- Aritméticos
- Unários
- Lógicos
- Condicionais
- Relacionais
- Igualdade

# Operadores Aritméticos

## Linguagem C#

<i>Operadores Aritméticos</i>	
Operador	Uso
+	Soma tipos numéricos. Também é usado para concatenar strings
-	Efetua a diferença de tipos numéricos
/	Efetua a divisão de tipos numéricos
*	Efetua o produto de tipos numéricos
%	Retorna o resíduo da divisão

# Operadores Unários

## Linguagem C#

<i>Operadores Unários</i>	
Operador	Uso
+	Especifica números positivos
-	Especifica números negativos
!	Negação booleana
~	Complemento bit a bit
<b>++ (pré)</b>	Incremento pré-fixado. Primeiro é efetuado o incremento e depois a variável é avaliada. Exemplo: ++x
<b>++ (pós)</b>	Incremento pós-fixado. Primeiro a variável é avaliada e depois é efetuado o incremento. Exemplo: x++
<b>-- (pré)</b>	Decremento pré-fixado. Primeiro é efetuado o incremento e depois a variável é avaliada. Exemplo: --x
<b>-- (pós)</b>	Decremento pós-fixado. Primeiro a variável é avaliada e depois é efetuado o decremento. Exemplo: x--
<b>(type)E</b>	Operador de mascaramento de tipos. Exemplo: (int) var.

# Operadores Lógicos

Linguagem C#

<i>Operadores Lógicos</i>	
Operador	Uso
&	Operador lógico AND. Efetua a operação AND bit a bit em tipos inteiros
	Operador lógico OR. Efetua a operação OR bit a bit em tipos inteiros
^	Efetua a operação XOR em tipos inteiros
>>	Efetua o deslocamento de um bit à direita de tipos numéricos
<<	Efetua o deslocamento de um bit à esquerda de tipos numéricos

# Operadores Condicionais

## Linguagem C#

<i>Operadores Condicionais</i>	
Operador	Uso
&&	Operador lógico AND usado para comparar expressões booleanas
	Operador lógico OR usado para comparar expressões booleanas
?:	Operador ternário usado da seguinte forma: expr. a: expr. b? expr. c. Isso equivale a dizer: if (expr. a) expr. b; else expr. c;

# Operadores Relacionais

## Linguagem C#

<i>Operadores Relacionais</i>	
Operador	Uso
<	Condição "menor que" para tipos numéricos
>	Condição "maior que" para tipos numéricos
>=	Condição "maior ou igual que" para tipos numéricos
<=	Condição "menor ou igual que" para tipos numéricos
is	Compara em tempo de execução se um objeto é compatível como um tipo qualquer. Esse assunto será abordado mais adiante
as	Efetua mascaramento de tipos e caso este falhe, o resultado da operação será null. Esse assunto será abordado mais adiante

# Operadores de Igualdade

## Linguagem C#

<i>Operadores de Igualdade</i>	
Operador	Uso
==	Avalia a igualdade de dois tipos
!=	Avalia a desigualdade de dois tipos