

# Operadores lógicos

CONCEITUAR SOBRE EXPRESSÕES ARITMÉTICAS E OS OPERADORES EM C#: OPERADORES ARITMÉTICOS, OPERADOR DE ATRIBUIÇÃO, COMBINAÇÃO DO OPERADOR DE ATRIBUIÇÃO COM OS OPERADORES ARITMÉTICOS, OPERADORES RELACIONAIS, OPERADORES LÓGICOS, OPERADORES DE INCREMENTO E DECREMENTO.

AUTOR(A): PROF. APARECIDA DE FATIMA CASTELLO ROSA

## Expressões Aritméticas

As expressões aritméticas em C# devem ser escritas na forma de linha reta. Dada a seguinte expressão:

$$\frac{a}{b}$$

Essa expressão deverá ser escrita como  $a / b$ .

As expressões também podem ser escritas utilizando-se parênteses, da mesma forma como utilizamos nas expressões algébricas, por exemplo, multiplicar  $a$  pelo valor da soma de  $b + c$ , escrevemos da seguinte forma:

$$a * (b + c)$$

As regras de precedência entre operadores, geralmente são as mesmas que na álgebra, e estas podem ser alteradas utilizando parênteses para forçar a ordem de avaliação.

Veja a tabela de precedência dos operadores aritméticos.

Operador	Operação	Ordem de precedência
( )	Parênteses	Primeira ordem. Se os parênteses estiverem aninhados, a expressão no par mais interno será avaliada primeiro. Se existirem vários parênteses no mesmo nível, ou seja, não aninhados, eles serão avaliados da esquerda para a direita.
* / %	Multiplicação Divisão Resto da Divisão	Segunda ordem. Se existirem vários desses operadores, eles serão avaliados da esquerda para a direita.
+ -	Adição Subtração	Terceira ordem. Se existirem vários desses operadores, eles serão avaliados da esquerda para a direita.

Vamos analisar a seguinte expressão algébrica:

$$m = \frac{a + b + c + d}{8}$$

Em C# devemos escrever:

$$m = (a + b + c + d) / 8.$$

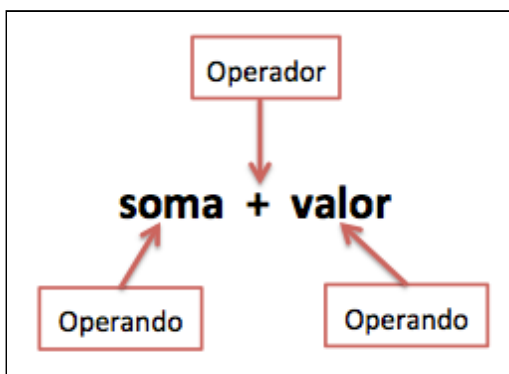
Caso os parênteses sejam erroneamente omitidos, por exemplo,  $m = a + b + c + d / 8$ , o compilador interpretará como sendo a expressão algébrica a seguir:

$$m = a + b + c + \frac{d}{8}$$

Operadores aritméticos

Os operadores são usados para operar sobre valores.

Os operadores aritméticos são usados para executar operações matemáticas simples. Eles são operadores binários, ou seja, cada um deles exige dois operandos, como por exemplo, a expressão `soma + valor` contém o operador binário `+` e os dois operandos `soma` e `valor`.



A tabela a seguir apresenta os operadores aritméticos.

Operador	Operação	Exemplo
+	Adição	(x=1; y=1) x + y = 2
-	Subtração	(x=2; y=3) x - y = -1
*	Multiplicação	(x=2; y=2.5) x * y = 5.0
/	Divisão	(x=5; y=2) x / y = 2
%	Resto da divisão de valores inteiros	(x=5; y=2) x % y = 1

A divisão inteira contém dois operandos int. O resultado dessa divisão é um quociente inteiro. Por exemplo: a expressão  $7 / 4$  tem como resultado 1; a expressão  $17 / 3$  tem como resultado 5. Nesses casos não podemos trabalhar com o resto da divisão utilizando o operador %, então  $7 \% 4$  é igual a 3 e  $17 \% 3$  é igual a 2, porque o operador % retorna o resto da divisão.

Para se obter o valor da divisão de números inteiros com as casas decimais, um dos números deverá ser convertido para um número real (número com casas decimais). No exemplo acima, poderíamos fazer um cast da seguinte forma: `x = 17 / (float)3;`

Divisão de números inteiro por zero não é permitido e causa um erro em tempo de execução.

Veja alguns exemplos de operadores aritméticos e seus resultados.

```
1. int a = 21;
2. int b = 10;
3. int c;
4. c = a + b;      // resultado da expressão: c = 31
5. c = a - b;      // resultado da expressão: c = 11
6. c = a * b;      // resultado da expressão: c = 210
7. c = a / b;      // resultado da expressão: c = 2
8. c = a % b;      // resultado da expressão: c = 1
```

## Operador de Atribuição

O operador para atribuir um valor a uma variável é = (o mesmo símbolo das equações matemática).

A sintaxe para atribuir um valor é a seguinte:

operando = valor, expressão.

Veja o trecho de código com exemplo utilizando o operador de atribuição.

```
1. int x = 8;  
2. int y = x;  
3. x = y + 2;
```

Na linha 1, o valor 8 'e atribuido à variável x; na linha 2, a variável y é declarada e inicializada com o valor de x, ou seja, é atribuído 8 à variável y.

Na linha 3, o valor de y é somado com 2 e depois o resultado é atribuído a x, que recebe o valor 10.

## Combinando operador de atribuição e operador aritmético

Um uso comum dos operadores de atribuição e de aritmética é para operar em uma variável e, em seguida, guardar o resultado na mesma variável.

Estas operações podem ser escritas, de forma simplificada, com os operadores de atribuição combinados.

Supondo que: x = 10; y = 5;

Operador	Expressão	Equivalência	Resultado
+=	x += y	x = x + y	x = 15
-=	x -= y	x = x - y	x = 5
*=	x *= y	x = x * y	x = 50
/=	x /= y	x = x / y	x = 2

É um erro de sintaxe se os operadores de atribuição combinado com os operadores de atribuição contiverem espaços entre seus símbolos (como em + =, - =, \* =, / =).

O lado direito de uma expressão de atribuição é sempre calculado antes que a atribuição ocorra, então cuidado ao utilizar essa forma.

x = x/y + 5	x = (10/5) + 5	x = 7
x /= y + 5	x = 10 / (5 + 5)	x = 1

```
1. int x = 2;
2. int y = 4;
3. x *= y;    // equivalente a x = x * y;
4.
5. //resultado será 8
```

## Operador Relacional

O operador relacional compara dois valores e sempre retorna um valor verdadeiro ou falso. Eles são principalmente utilizados para especificar condições, nas quais envolvem avaliar se uma expressão é verdadeira ou falsa.

Considere a seguinte declaração: `int x = 3;`

Operador	Descrição	Exemplo	Resultado da comparação
<code>==</code>	Igual a	<code>x == 3</code>	true
<code>&gt;</code>	Maior que	<code>x &gt; 3</code>	false
<code>&lt;</code>	Menor que	<code>x &lt; 3</code>	false
<code>&gt;=</code>	Maior ou igual a	<code>x &gt;= 3</code>	true
<code>&lt;=</code>	Menor ou igual a	<code>x &lt;= 3</code>	true
<code>!=</code>	Diferente de	<code>x != 3</code>	false

É um erro de sintaxe se os operadores relacionais `==`, `!=`, `>=` e `<=` contiverem espaço entre seus símbolos (como em `=`, `!=`, `> =`, `< =`).

Também é um erro de sintaxe caso os operadores `!=`, `>=`, `<=` sejam invertidos (como em `=!`, `=>`, `=<`)

## Operadores Lógico

Os operadores lógicos são utilizados juntamente com os operadores relacionais. Eles avaliam expressões lógica que resultam em verdadeiro ou falso.

Operador	Significado	Exemplo
&&	Operador E condicional	<code>x &gt; 3 &amp;&amp; x &lt; 10</code>
&	Operador E lógico	<code>x &gt; 3 &amp; x &lt; 10</code>
	Operador OU condicional	<code>x &gt; 3    x &lt; 10</code>
	Operador OU inclusivo lógico	<code>x &gt; 3   x &lt; 10</code>
^	XOR Operador OU exclusivo lógico	<code>x ^ y</code>
!	Operado NÃO lógico, também chamado de negação lógica	<code>x != 3</code>

Vamos relembrar a tabela verdade que envolve os operadores lógicos E, OU e XOR. Assuma que X e Y são booleanos.

X	Y	X && Y	X    Y	X ^ Y
V	V	V	V	F
V	F	F	V	V
F	V	F	V	V
F	F	F	F	F

## Operadores de Incremento e Decremento

Uma outra operação comum em computação é incrementar ou decrementar uma variável em uma unidade. Por exemplo:

```
int x = 2;
```

```
x = x + 1; //o valor de x é incrementado em 1 e passa a ter o valor igual a 3
```

```
int y = 5;
```

```
y = y - 1; // o valor de y é decrementado em 1 e passa a ter o valor igual a 4
```

Nesses casos podemos utilizar o operador de incremento ( ++ ) e o operador de decremento ( -- ).

Operador	Significado	Operadores lógicos	Expressão	Equivalência
++	Incremento pós-fixado		<code>x++</code>	<code>x = x + 1</code>

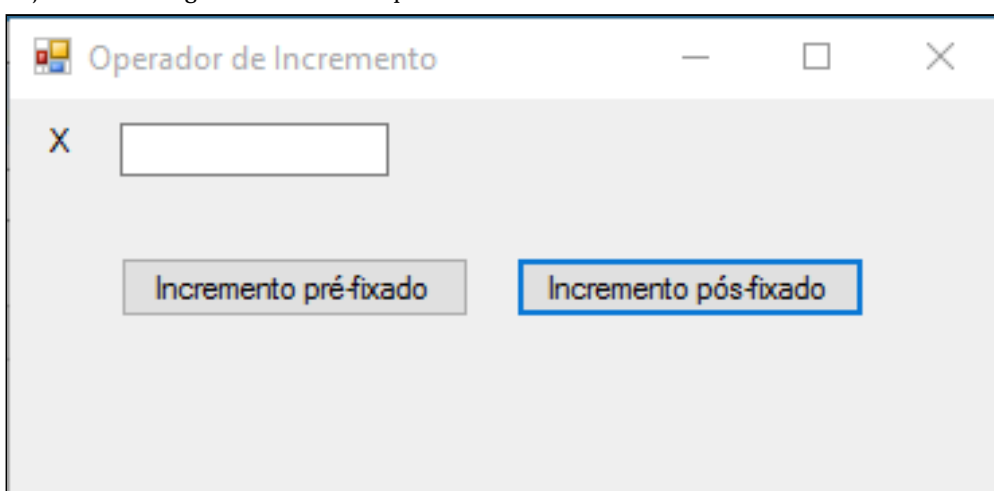
$y = x++$	$y = x ; x = x + 1 ;$	$y = 10 ; x = 11 ;$
$z = ++x$	$x = x + 1 ; z = x ;$	$z = 11 ; x = 11 ;$

## Hora de Praticar

Vamos criar um projeto para implementar o operador de incremento pré-fixado e pós-fixado.

Vamos ver também como colocar um título em um MessageBox.

Veja a interface gráfica desse exemplo.



Observação: A aparência pode ser diferente no seu computador dependendo da versão do seu sistema operacional.

O form contém os seguintes controles:

- 1 Label;
- 2 Button.

Vamos começar.

1. Inicie o Visual Studio (se ainda não o fez).
2. Na Start Page escolha New Project, ou caso a Start Page não esteja visível, na barra de menu escolha File, New Project.

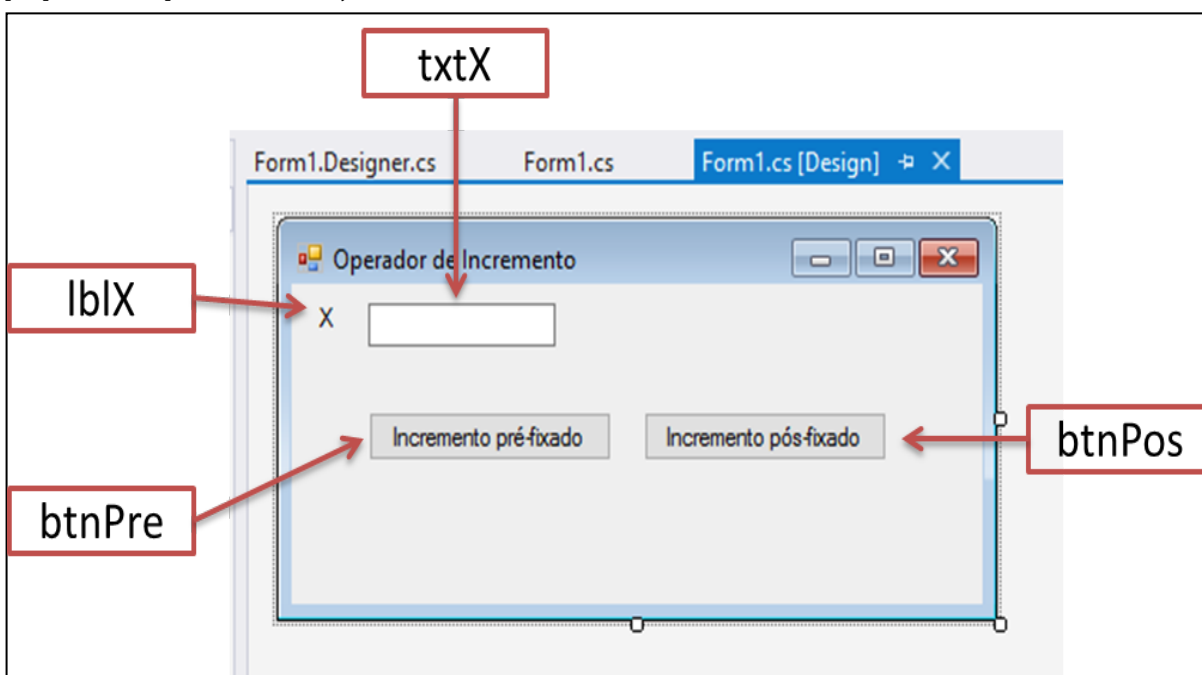
A caixa de diálogo é exibida.

1. Selecione no painel à esquerda o template Visual C#, no painel central escolha Windows Forms Application.
2. Na caixa de texto Name: Operadores.
3. Location: escolha o local onde deseja salvar o seu projeto.
4. Mantenha selecionado: Create directory for solution.
5. Clique em OK

#### Criando o Design do Form

1. Após fixar (Auto Hide) a Toolbox, abra o nó Common Controls, e insira os controles no Form. Lembre-se para inserir os controles, basta selecionar o controle na ToolBox, clicar, segurar e arrastar para a posição que desejar dentro do Form.

Você poderá ter um layout parecido com a figura a seguir, mas não se preocupe se o posicionamento e tamanho dos controles não estiverem iguais ao da figura porque hoje você vai trabalhar com novas propriedades que alteram o layout.



Faça um layout parecido com a imagem acima.

Selecione os controles e altere as propriedades conforme definido na tabela abaixo:



Controle	Propriedade = Valor
Form	Text = Operador de Incremento
Label x	(name) = lblX Text = X
Button Incremento Pré-fixado	(name) = btnPre Text = Incremento Pré-fixado
Button Incremento Pós-fixado	(name) = btnPos Text = Incremento Pós-fixado Location =

### Escrevendo o código do programa Operadores

Agora que você já criou a interface e alterou as propriedades dos controles, vamos codificar o botão btnPre. Para abrir o editor de código dê duplo clique no botão btnPre e digite o código a seguir.

```

1. int x, y;          //declaração das variáveis
2. x = Convert.ToInt32(txtX.Text);    //entrada de dados, converte o texto do Te
3.
4. y = ++x;          //incremento pré-fixado
5.
6. MessageBox.Show("Y = " + y + " X = " + x , "Incremento Pré-fixado");    //s

```

Agora vamos codificar o botão btnPos. Volte para o modo design e dê duplo clique no botão btnPos e digite o código a seguir.

```
1. int x, y;          //declaração das variáveis
2. x = Convert.ToInt32(txtX.Text); //entrada de dados, converte o texto do TextB
3.
4. y = x++;          //incremento pós-fixado
5.
6. MessageBox.Show("Y = " + y + " X = " + x , "Incremento Pós_Fixado"); //saída
```

Os códigos dos botões devem ter ficado parecido com a imagem a seguir.

```
1 reference
private void btnPre_Click(object sender, EventArgs e)
{
    int x, y;
    x = Convert.ToInt32(txtX.Text);

    y = ++x;

    MessageBox.Show("Y = " + y + " X = " + x , "Incremento Pré-fixado");
}

1 reference
private void btnPos_Click(object sender, EventArgs e)
{
    int x, y;
    x = Convert.ToInt32(txtX.Text);

    y = x++;

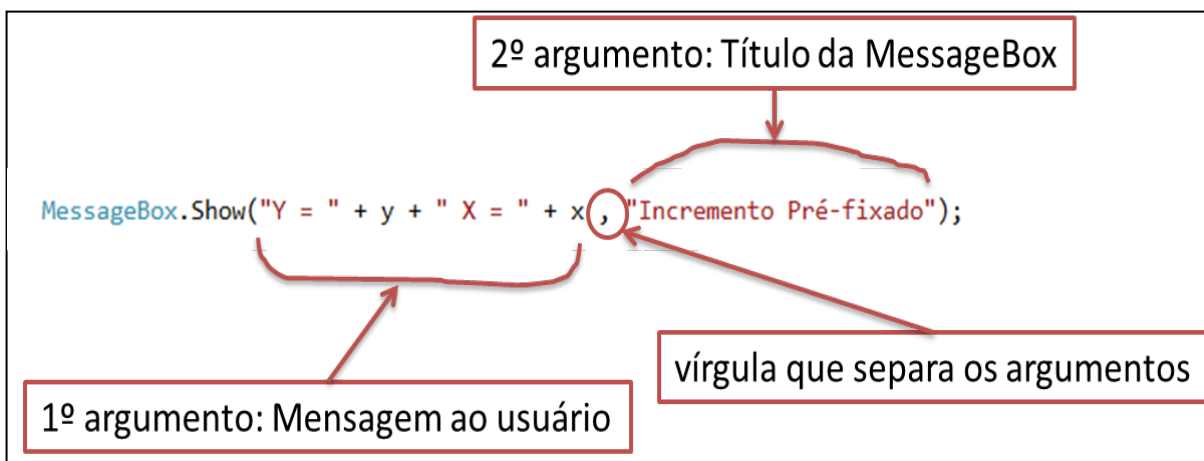
    MessageBox.Show("Y = " + y + " X = " + x , "Incremento Pós_Fixado");
}
```

Entendendo o código do MessageBox.

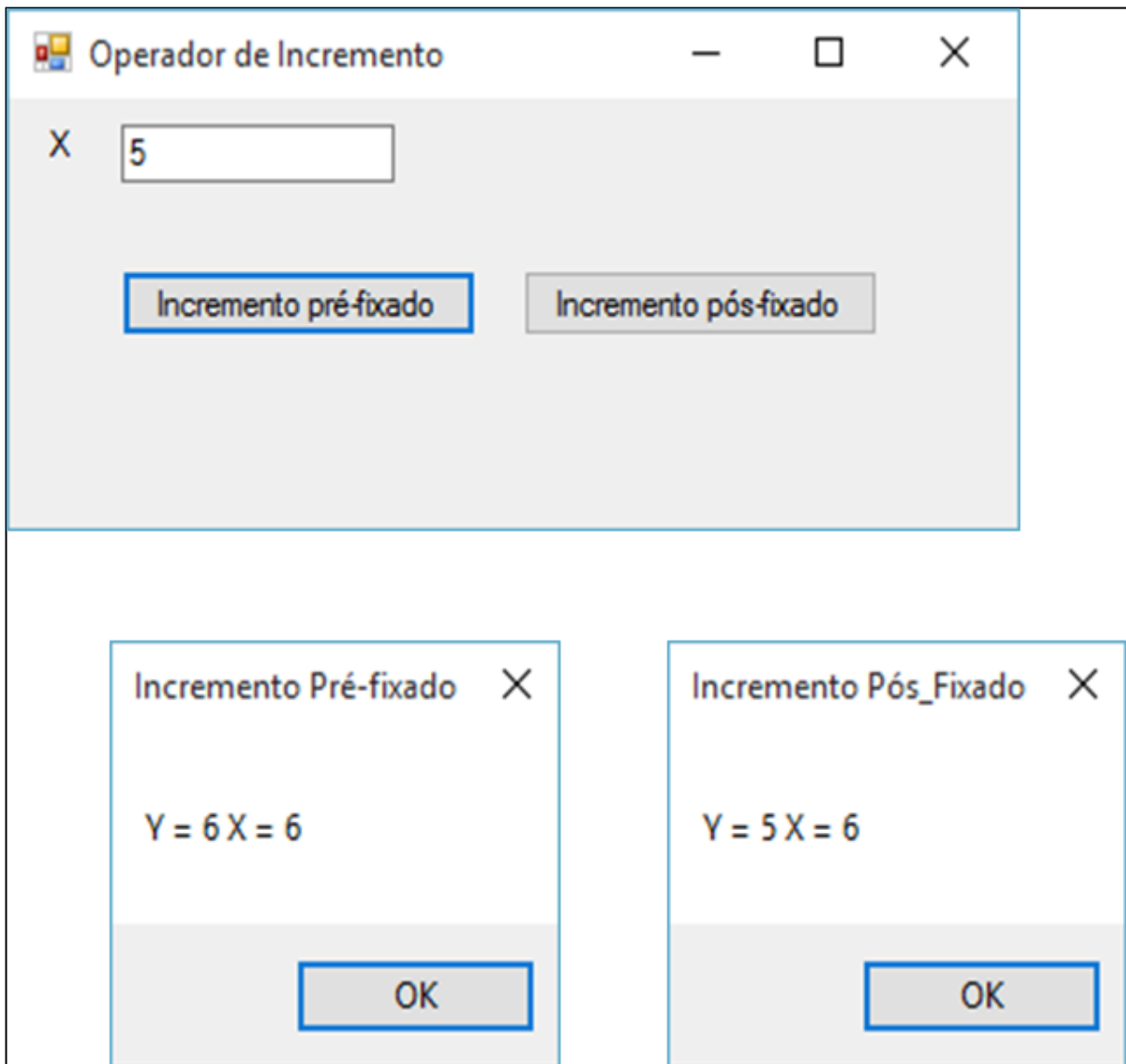
O primeiro argumento é o texto da mensagem que queremos apresentar ao usuário.

O segundo argumento é o texto que queremos que apareça na barra de título da MessageBox.

Veja a figura a seguir.



Veja o resultado final do programa.



## ATIVIDADE

Dado o trecho de código a seguir.

```
int x, y;
```

```
x += y;
```

Foi atribuído o valor 27 para a variável x e o valor 5 para a variável y.  
Após a execução das instruções acima, qual será o valor da variável x?

- A. 27
- B. 5
- C. 22
- D. 32

## ATIVIDADE

Dado o trecho de código a seguir.

Dado o trecho de código a seguir.

```
int x, y;
```

```
y = 26;
```

```
x = y % 3;
```

Após a execução das instruções acima, qual será o valor da variável x?

A. 2

B. 8

C. 8.6666

D. 78

## ATIVIDADE

Dado o código a seguir:

```
1.    int x, y;  
2.    y = 2;  
3.    x == y;
```

Analise as afirmações a seguir.

- I. A instrução da linha 1 declara as variáveis x e y do tipo inteiro.
- II. A instrução da linha 2 atribui o valor 2 a variável y.
- III. A instrução da linha 3 atribui o valor de y a variável x.

É correto apenas o que se afirma em

- A. I e II
- B. I e III
- C. II e III
- D. III

## REFERÊNCIA

DEITEL, H. M. *et al.* C# Como Programar. São Paulo: Pearson Makron Books, 2003.

DORMAN, Scott. *Sams Teach Yourself Visual C#®2010 in 24 Hours: Complete Starter Kit*. 2010.

OLSSON, Mikael. *C# Quick Syntax Reference*. Berkeley, CA: Apress, 2013.

### SITES

Microsoft <<https://msdn.microsoft.com/pt-br/library/ms173145.aspx>> (<https://msdn.microsoft.com/pt-br/library/ms173145.aspx>). Acesso em 20/09/2015

