

Programação .NET

Professor Ricardo Frohlich da Silva

Exemplo 12: escolher operação

```
static void Main(string[] args)
{
    double num1, num2, resultado;
    int op;
    Console.WriteLine("Digite dois numeros:");
    num1 = Double.Parse(Console.ReadLine());
    num2 = Double.Parse(Console.ReadLine());
    Console.WriteLine("\nEscolha a operacao:\n1.Soma\n2.Subtracao\n3.Multiplicacao\n4.Divisao\n5.Potencia\n\nEscolha:");
    op = int.Parse(Console.ReadLine());
    if (op == 1)
    {
        resultado = num1 + num2;
        Console.WriteLine("Soma = " + resultado);
    }
    else if (op == 2)
    {
        resultado = num1 - num2;
        Console.WriteLine("Subtração = " + resultado);
    }
    else if (op == 3)
    {
        resultado = num1 * num2;
        Console.WriteLine("Multiplicação = " + resultado);
    }
    else if (op == 4)
    {
        resultado = num1 / num2;
        Console.WriteLine("Divisão = " + resultado);
    }
    else if (op == 5)
    {
        resultado = Math.Pow(num1, num2);
        Console.WriteLine("Potência = " + resultado);
    }
}
```

Estruturas seletivas

Switch-case

- O comando `switch` é um comando de seleção múltipla;
- O `switch` testa sucessivamente o valor de uma expressão contra uma lista de constantes inteiras ou de caracteres:
 - Só funciona para INTEIROS e CARACTERES!
- O comando `switch` somente testa a igualdade da variável em relação a valores constantes, diferente do comando `if` que é capaz de testar expressões lógicas e/ou relacionais.

Switch-case

```
switch(expressão){  
    case constante1:  
        seqüência de comandos  
        break;  
    case constante2:  
        seqüência de comandos  
        break;  
    case constante3:  
        seqüência de comandos  
        break;  
    ...  
    default:  
        seqüência de comandos  
}
```

- O valor da expressão é testado, na ordem, contra os valores das constantes especificadas nos comandos case;
- Quando uma coincidência for encontrada, a seqüência de comando associada àquele case será executada até que o comando break ou o fim do comando switch seja alcançado;
- O comando default é executado se se nenhuma coincidência for detectada;
- O default é opcional.

Exemplo 1:

```
static void Main(string[] args)
{
    int num;
    Console.WriteLine("Digite um numero: ");
    num = int.Parse(Console.ReadLine());
    switch (num)
    {
        case 9:
            Console.WriteLine("o numero digitado é 9");
            break;
        case 10:
            Console.WriteLine("o numero digitado é 10");
            break;
        default:
            Console.WriteLine("o numero digitado não é 9 nem 10");
            break;
    }
}
```

- Obs. 1: abrimos e fechamos “{ }” ao iniciar um comando **switch**!
- Obs. 2: nos **case**, utilizamos “:” para abrir e **break** para terminar!!

Estruturas de repetição

Prof. Ricardo Frohlich da Silva

Estruturas de repetição

- As estruturas de repetições são bastante importantes pois muitas vezes necessitamos executar um mesmo procedimento diversas vezes.
- A linguagem C# basicamente possui três tipos de estrutura de repetição:
 - **for**, **while** e **do... while**

For

- O comando **for** permite implementar laços de repetição, ou seja, definir trechos do código de um programa que se repetem.
- O **for**, como qualquer iteração, necessita de uma variável para controlar a quantidade de loops (voltas) a serem executadas.
- O comando **for** inclui na sua própria definição uma variável contadora:
 - A variável que vai definir quantas vezes aquele trecho de código será executado.

For

- No comando **for**, a variável contadora deverá ser inicializada com um valor, e ser incrementada ou decrementada a cada loop;
- Além disso, deveremos ter uma condição, testando se a quantidade de vezes que o laço deveria ser executado chegou ao fim ou não (dependendo do valor atual da variável contadora);

For

- Exemplo:
 - executando um laço de repetição 4 vezes.
 - No **for**, inicializamos uma variável contadora **int i**, cujo valor inicial é **0**;
 - A cada loop/laço a variável é incrementada em **1**;
 - Após cada incremento, é testada uma condição para verificar se a variável chegou ao fim, ou seja, se o seu valor é igual a **4**:
 - Caso o valor tenha chegado a **4**, o laço termina;
 - Caso contrário, o laço se repete novamente.

For

- Portanto, o comando **for** necessita de 3 partes:

for(valor inicial; condição; incremento ou decremento){

<bloco de instruções>;

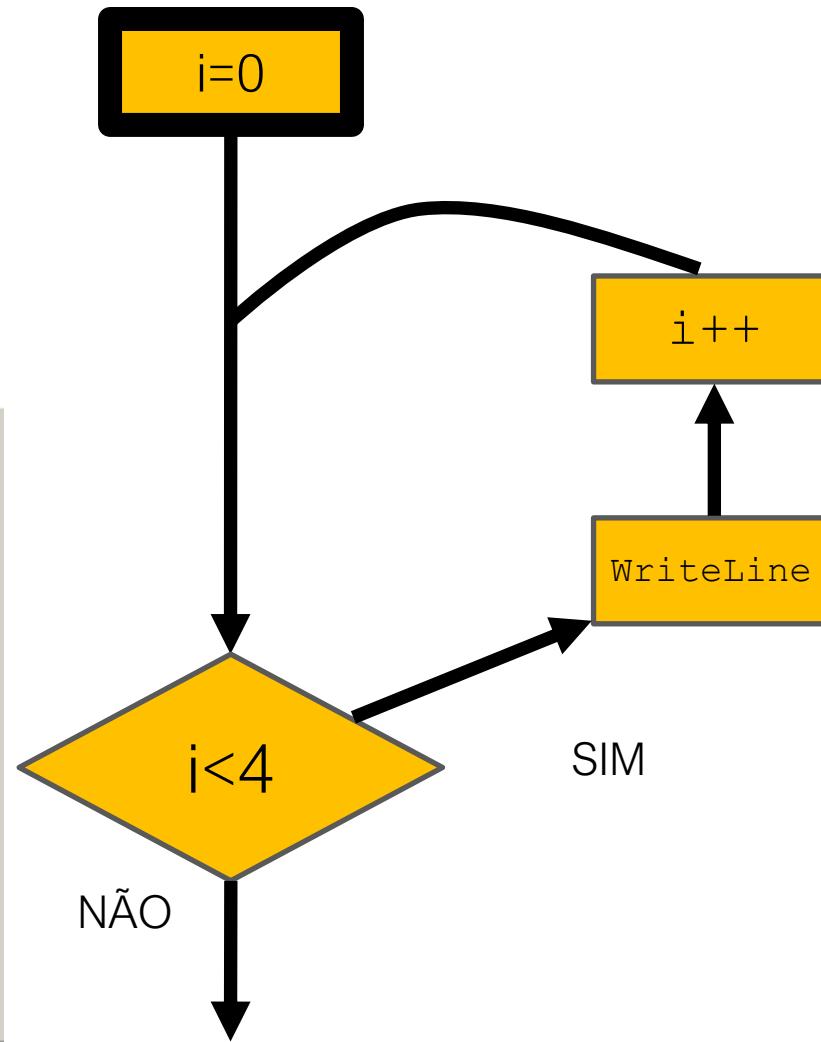
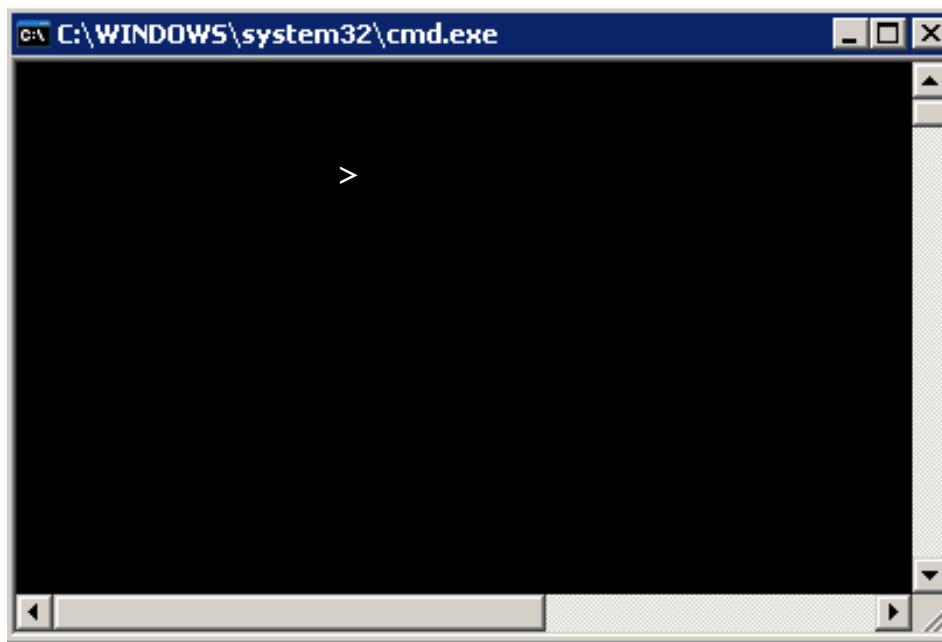
}

- Em que:

- valor inicial**: definimos um valor inicial para nossa variável contadora, que no caso do exemplo anterior, definimos que `i` vale 1 (`i=0`);
- condição**: testamos se o laço chegou ao fim, que no caso do exemplo anterior, se `i` chegou a 4 (`i < 4`);
- incremento**: `i` é incrementado em 1, para contar o número de vezes que o laço será repetido (`i++`)
- <bloco de instruções>**: bloco que será executado 4 vezes, neste exemplo.

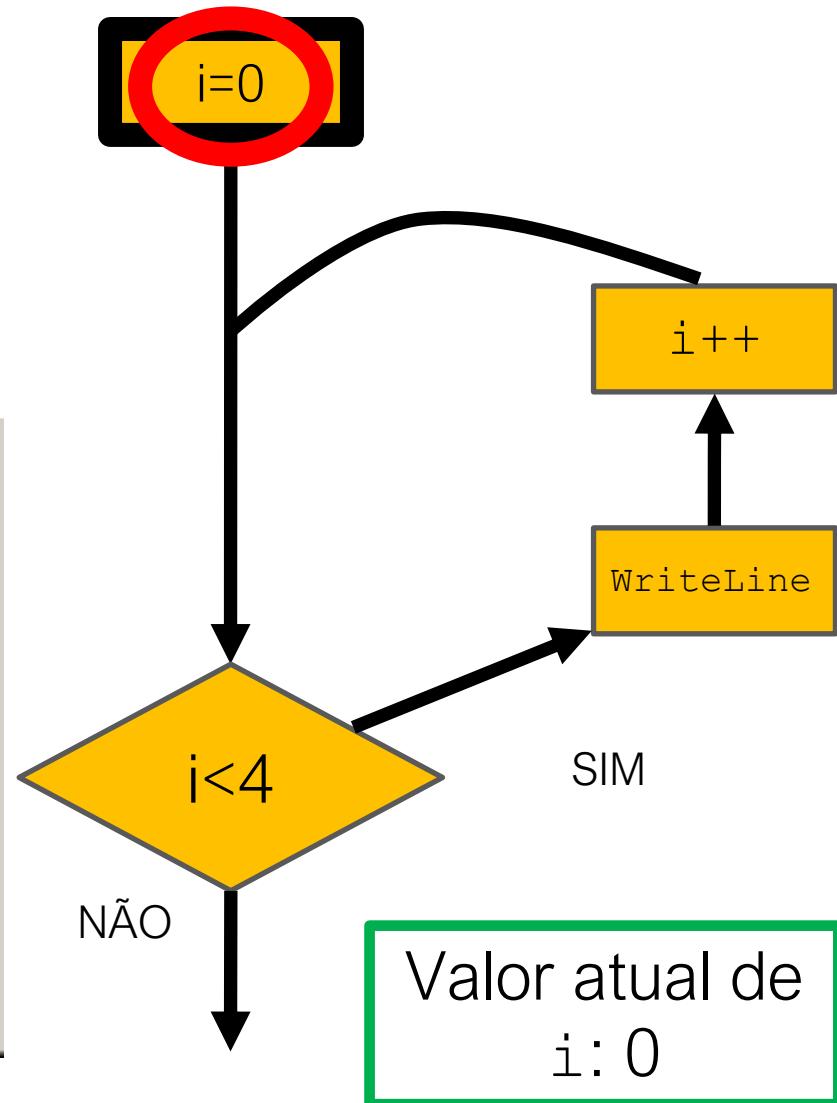
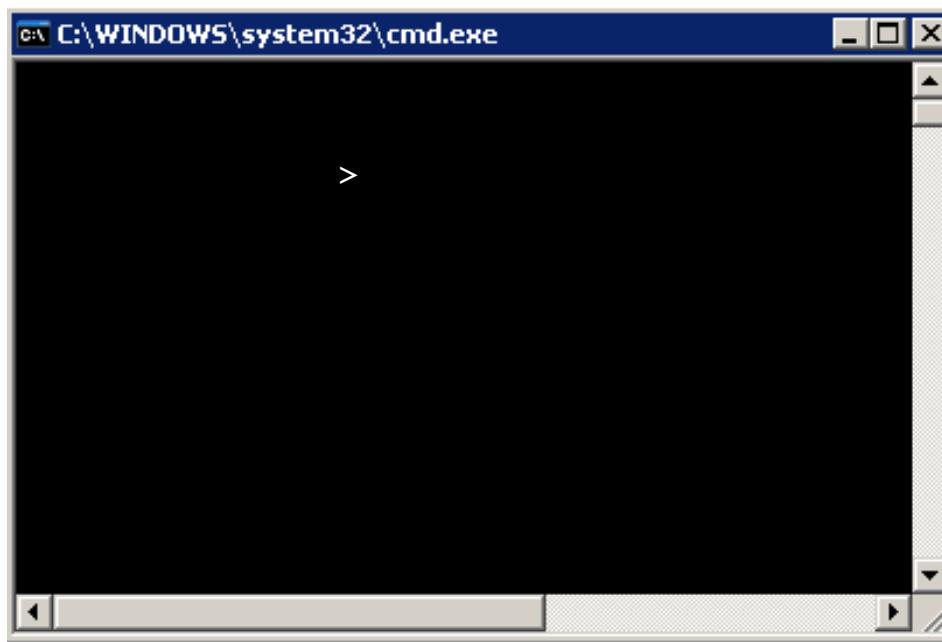
Entendendo o Exemplo 1

```
int i;
for (i = 0; i < 4; i++)
{
    Console.WriteLine("i vale "+i);
}
```



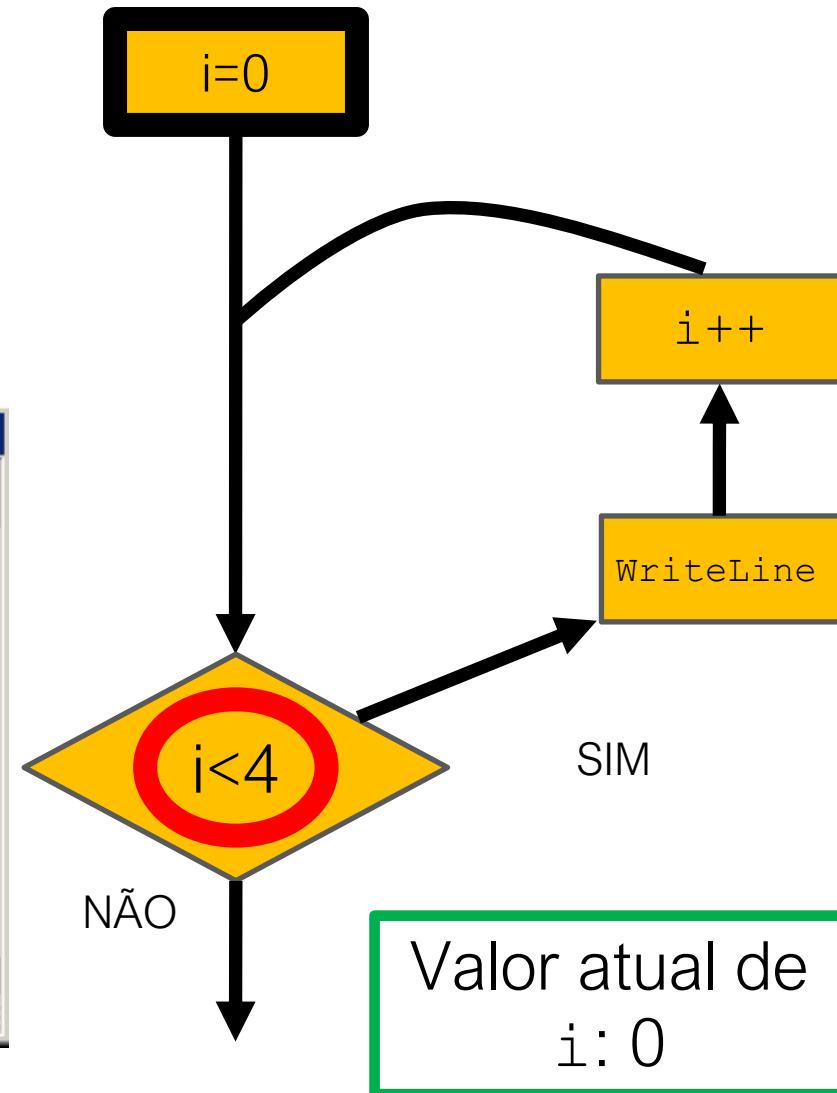
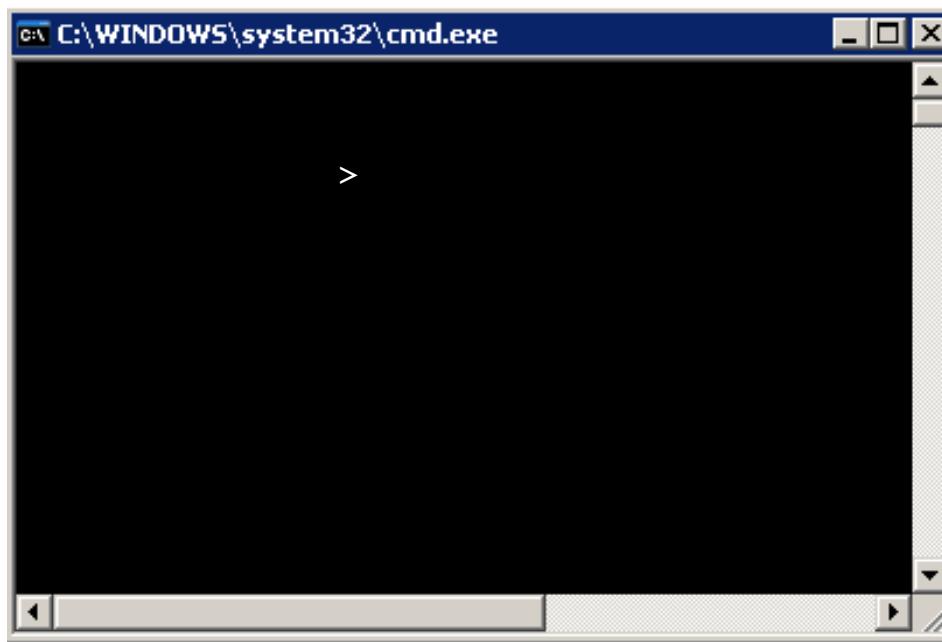
Entendendo o Exemplo 1

```
int i;
for i = 0  i < 4; i++)
{
    Console.WriteLine("i vale "+i);
}
```



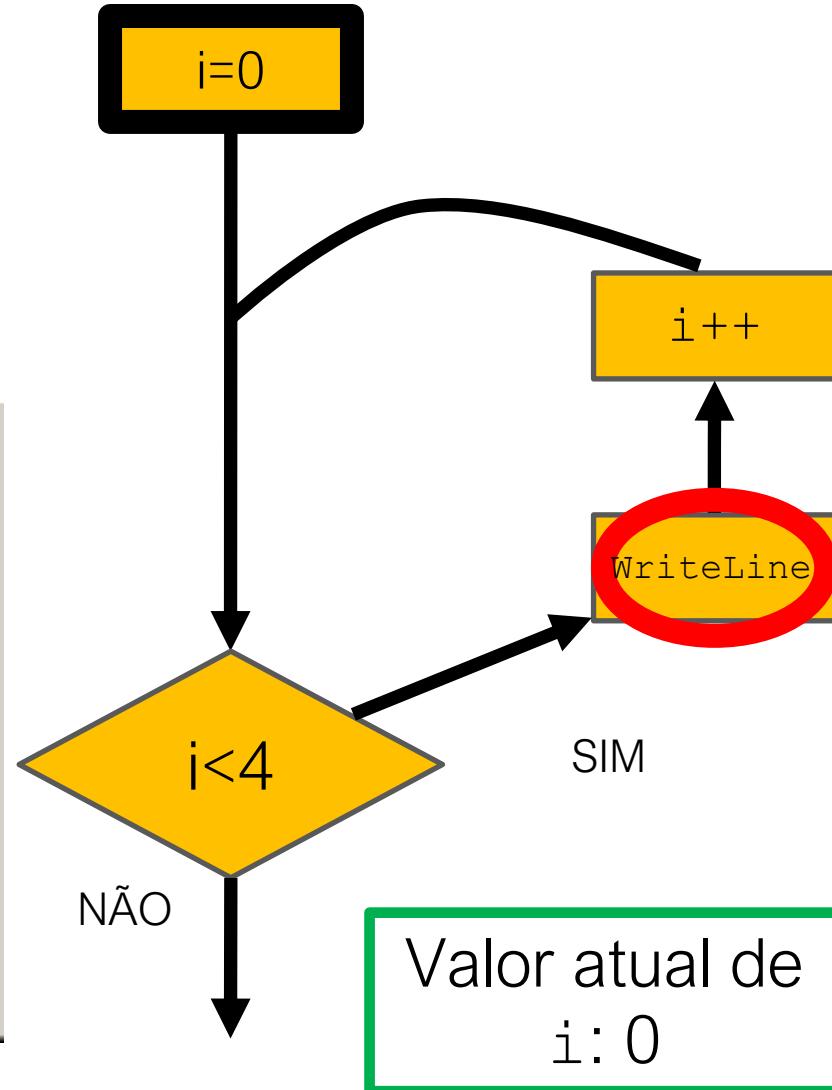
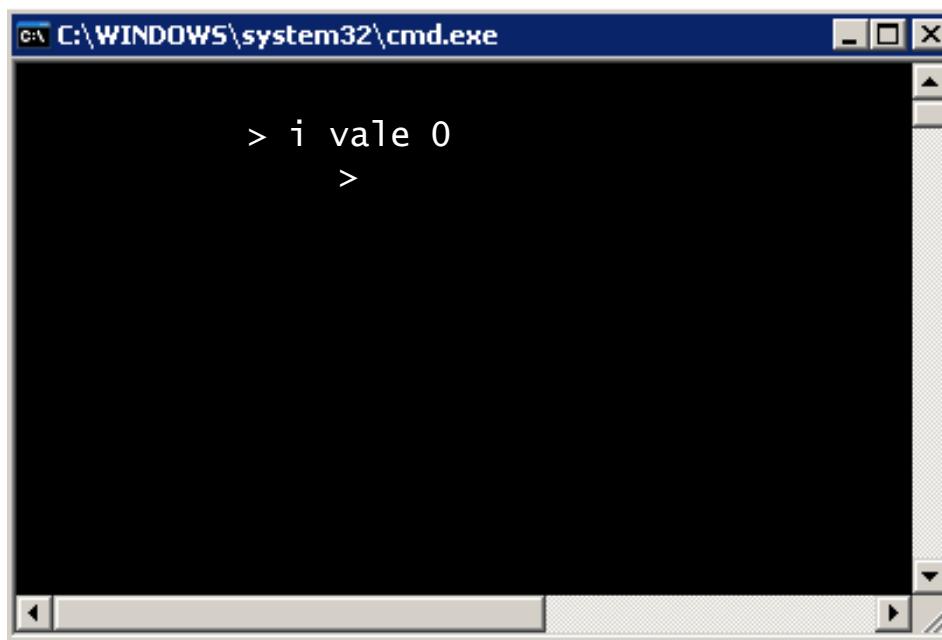
Entendendo o Exemplo 1

```
int i;  
for (i = 0; i < 4 i++)  
{  
    Console.WriteLine("i vale "+i);  
}
```



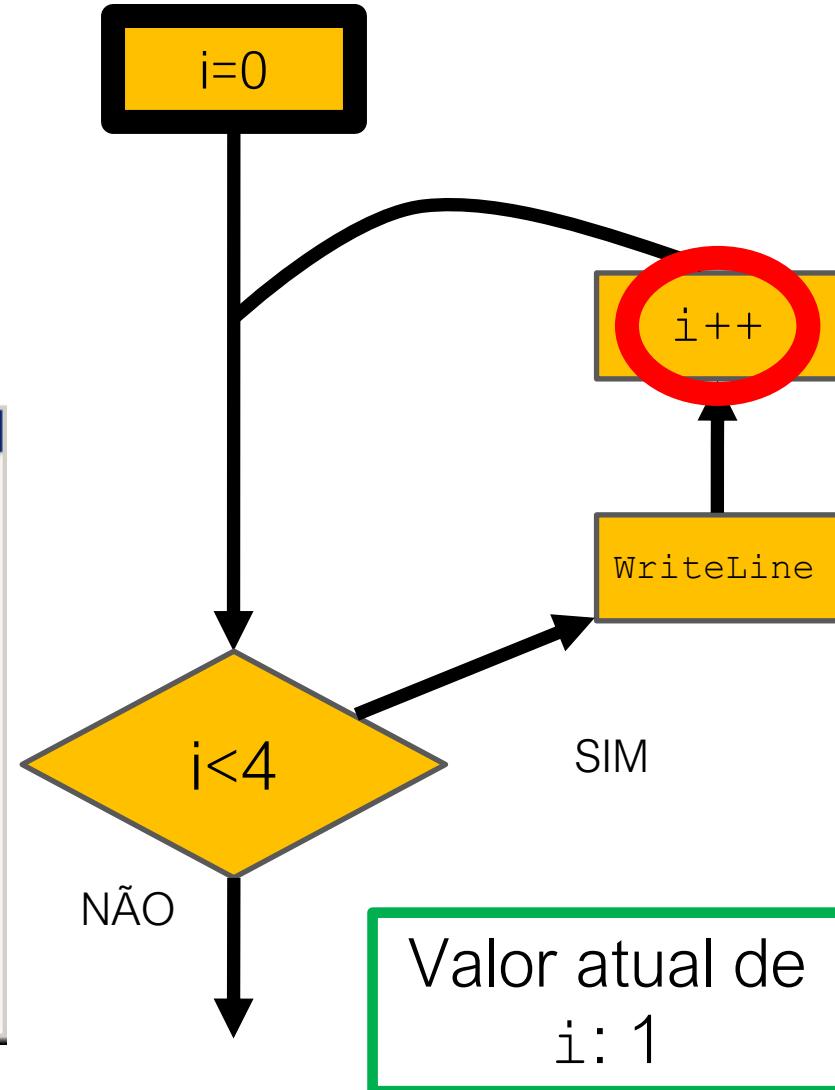
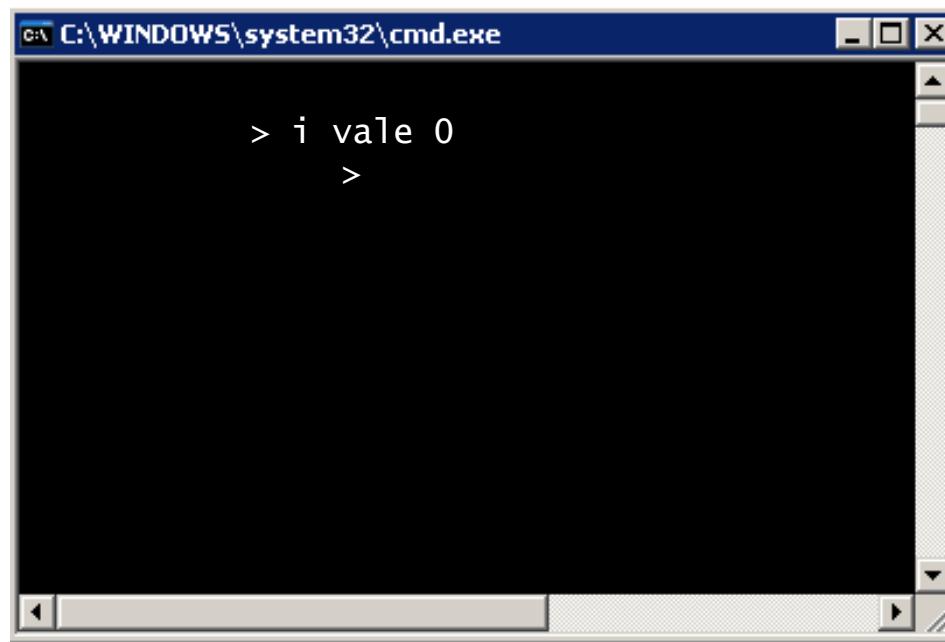
Entendendo o Exemplo 1

```
int i;  
for (i = 0; i < 4; i++)  
{  
    Console.WriteLine("i vale "+i);  
}
```



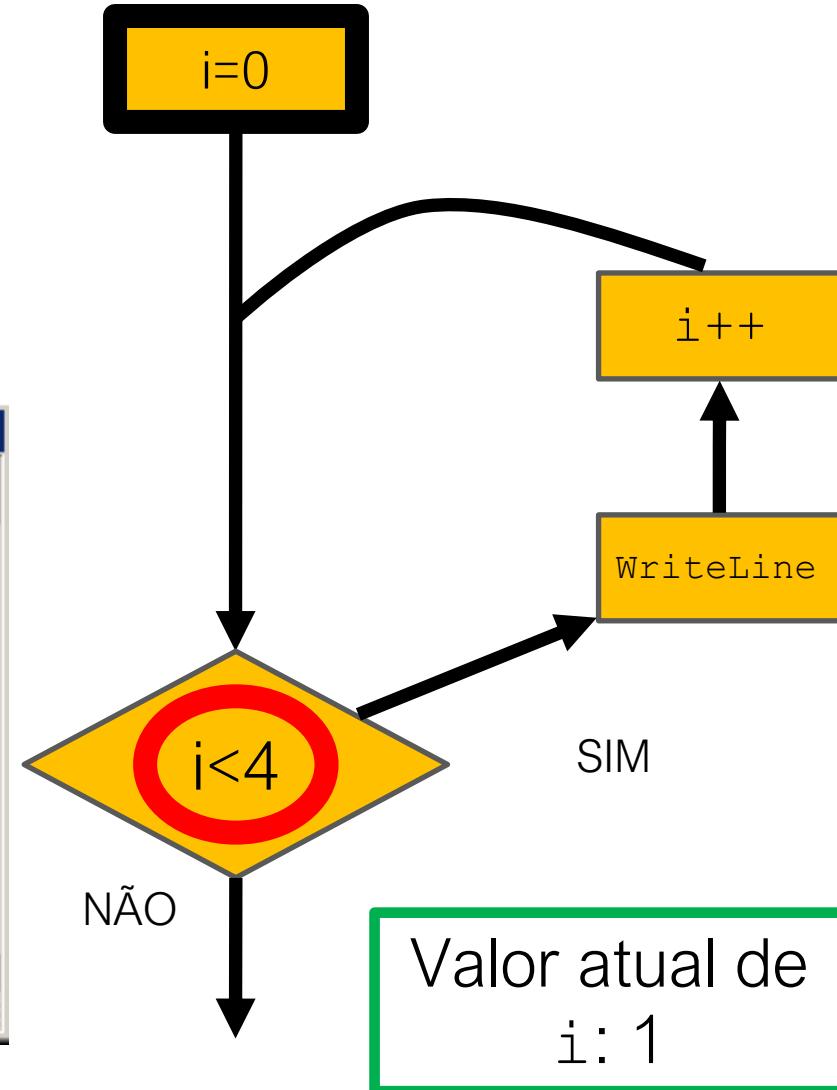
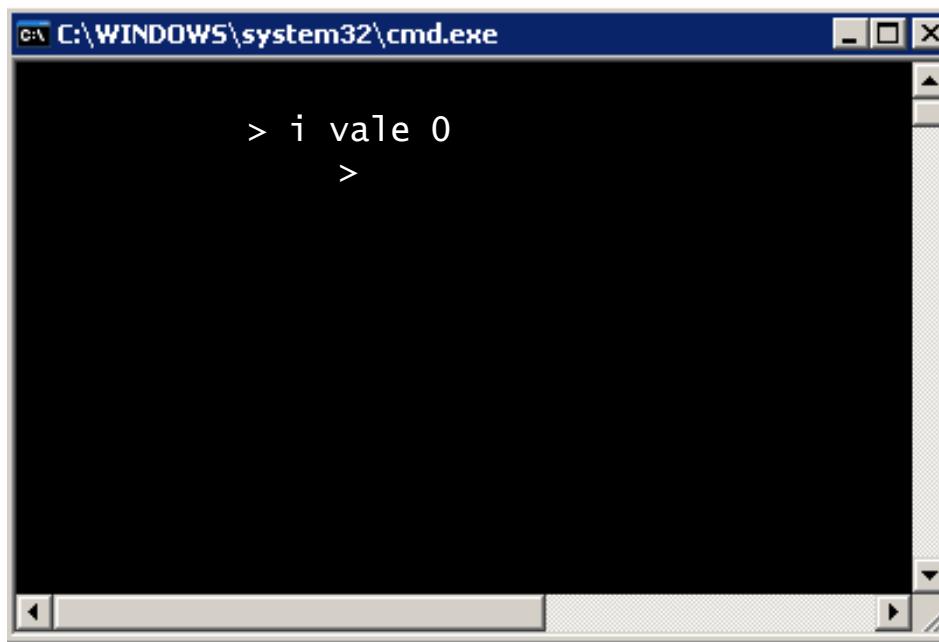
Entendendo o Exemplo 1

```
int i;  
for (i = 0; i < 4; i++)  
{  
    Console.WriteLine("i vale "+i);  
}
```



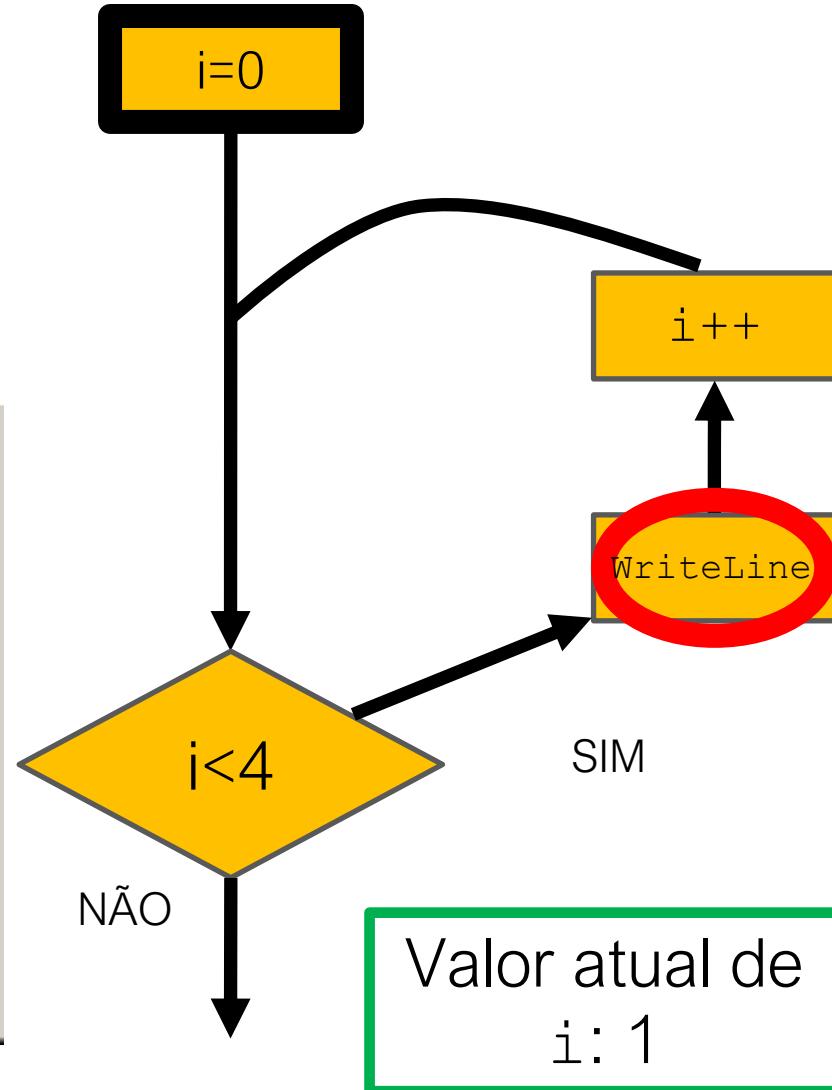
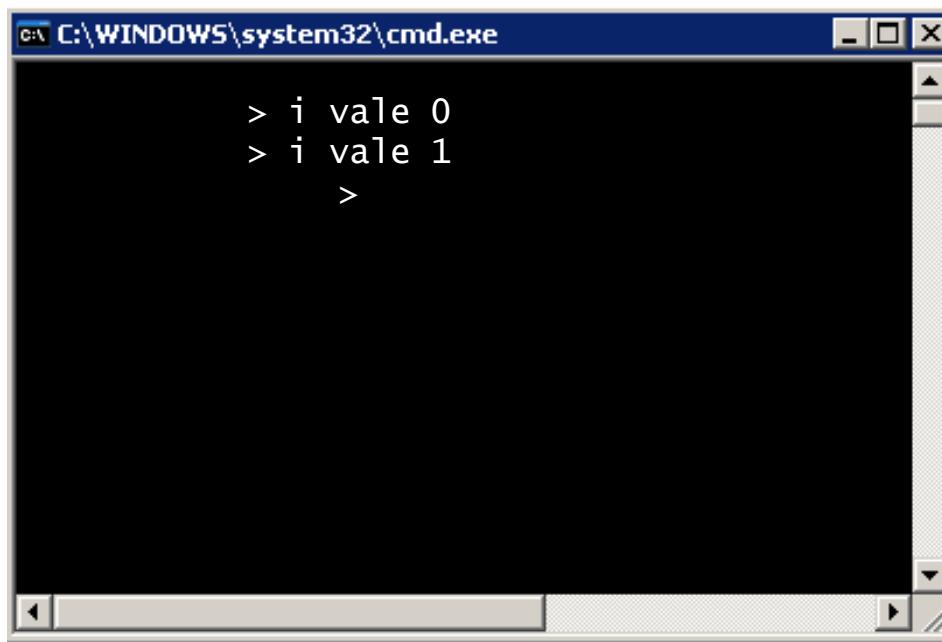
Entendendo o Exemplo 1

```
int i;  
for (i = 0; i < 4 i++)  
{  
    Console.WriteLine("i vale "+i);  
}
```



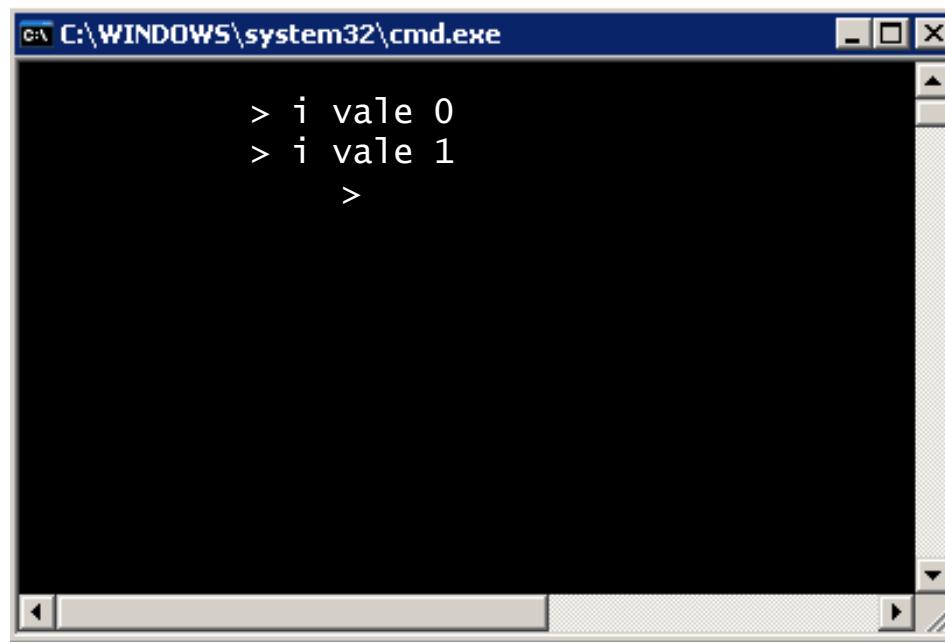
Entendendo o Exemplo 1

```
int i;  
for (i = 0; i < 4; i++)  
{  
    Console.WriteLine("i vale "+i);  
}
```

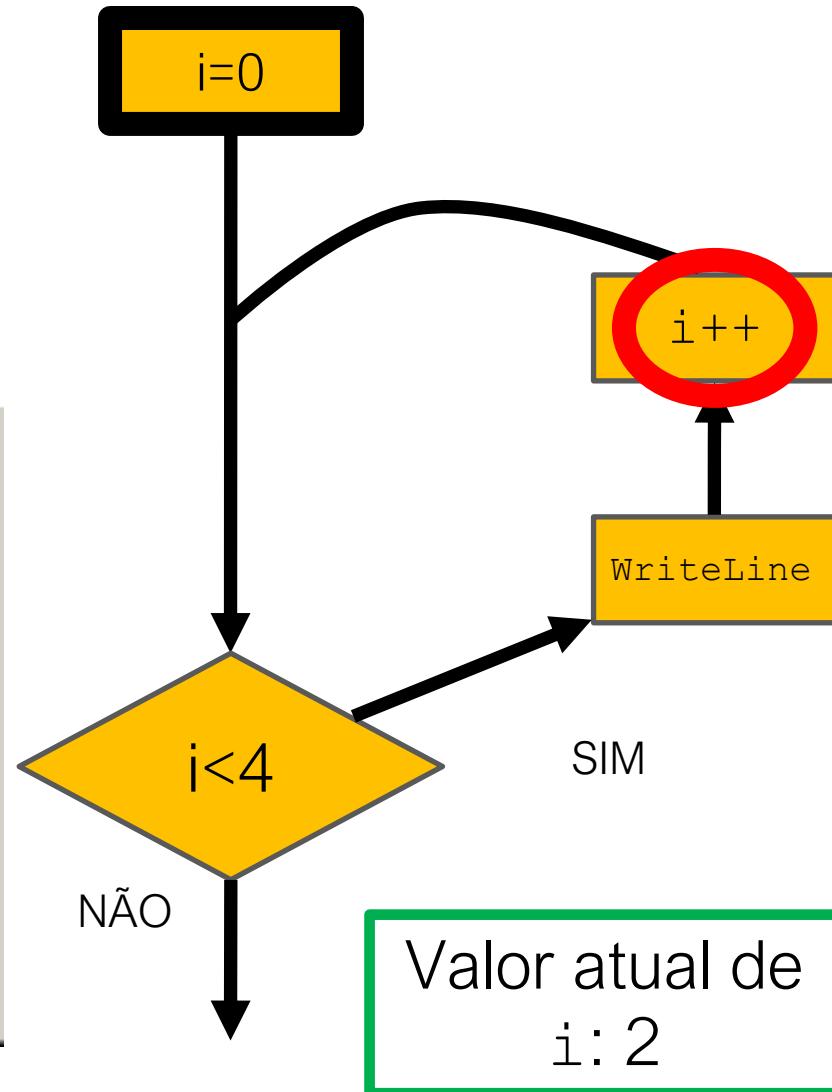


Entendendo o Exemplo 1

```
int i;  
for (i = 0; i < 4; i++)  
{  
    Console.WriteLine("i vale "+i);  
}
```

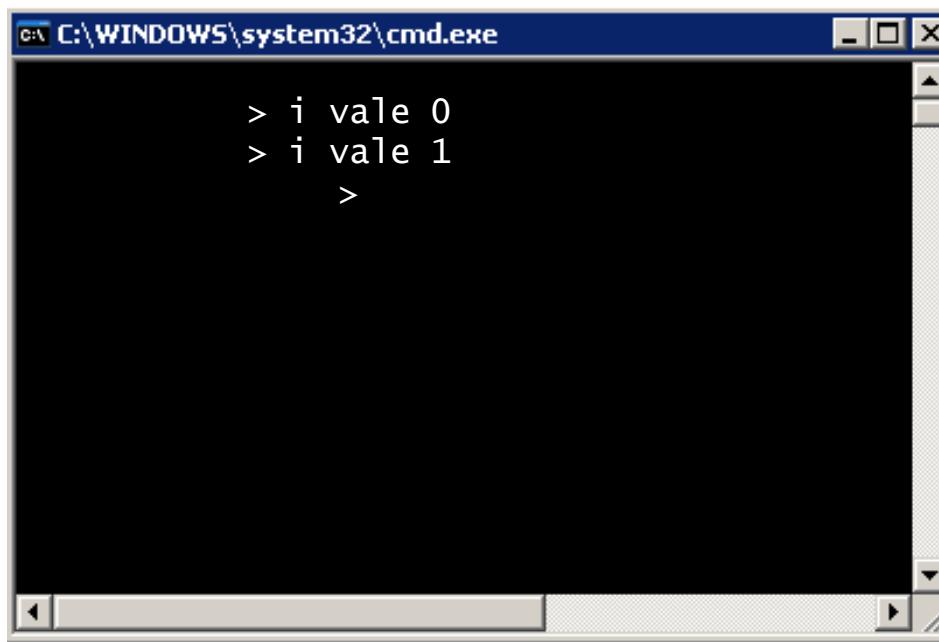


A screenshot of a Windows command prompt window titled 'cmd.exe' located at 'C:\WINDOWS\system32'. The window displays the following text:
> i vale 0
> i vale 1
>

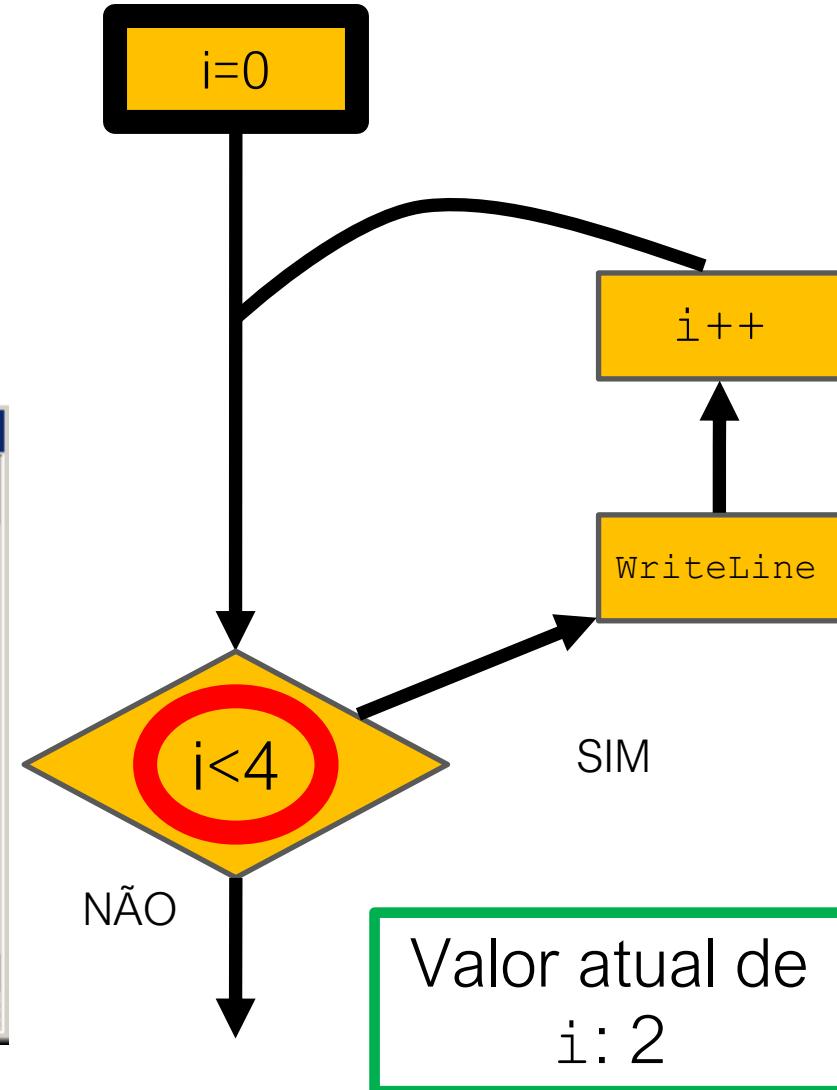


Entendendo o Exemplo 1

```
int i;  
for (i = 0; i < 4 i++)  
{  
    Console.WriteLine("i vale "+i);  
}
```

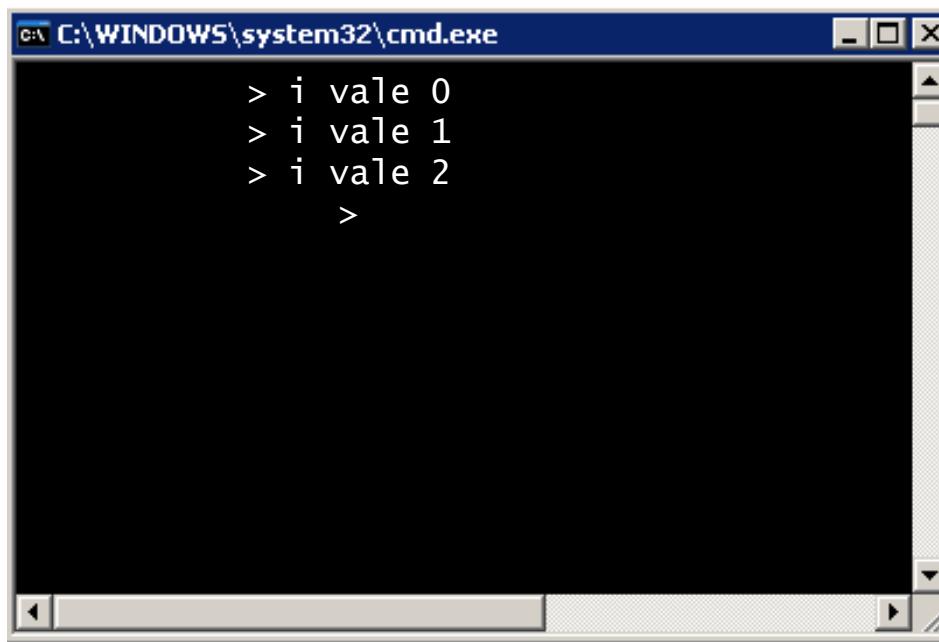


The screenshot shows a Windows command prompt window titled 'cmd.exe' with the path 'C:\WINDOWS\system32'. It displays the following text:
> i vale 0
> i vale 1
>

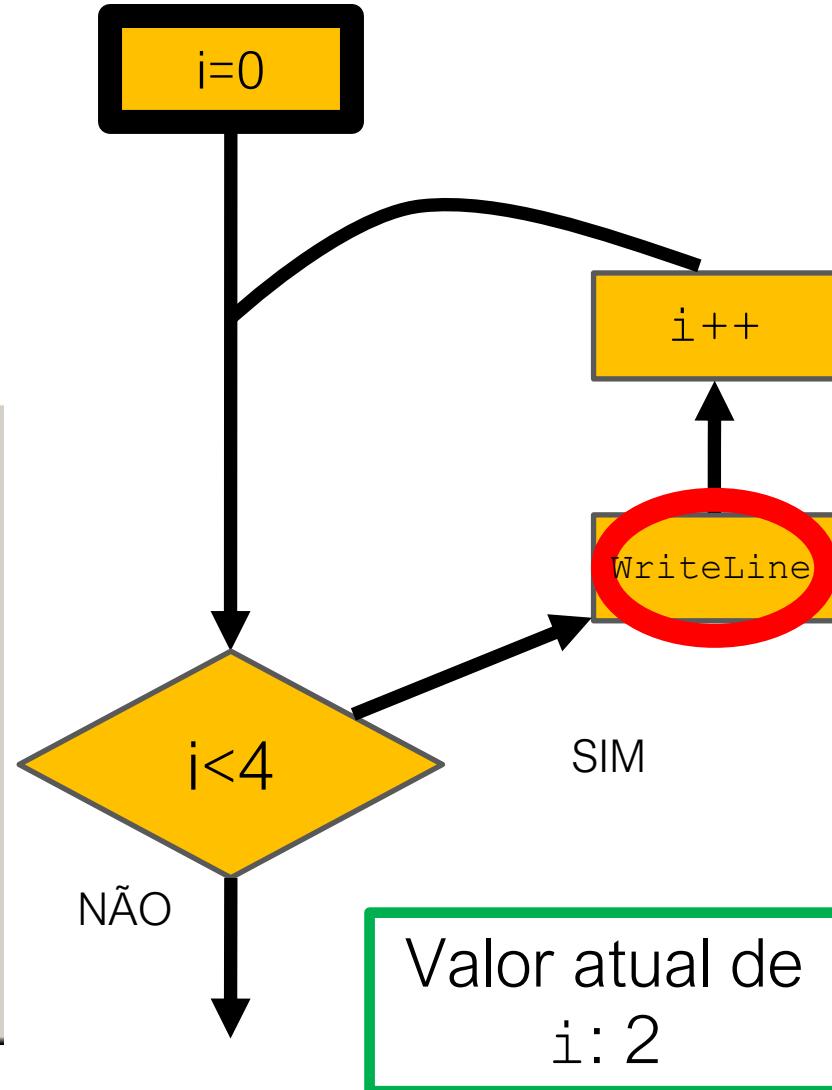


Entendendo o Exemplo 1

```
int i;  
for (i = 0; i < 4; i++)  
{  
    Console.WriteLine("i vale "+i);  
}
```

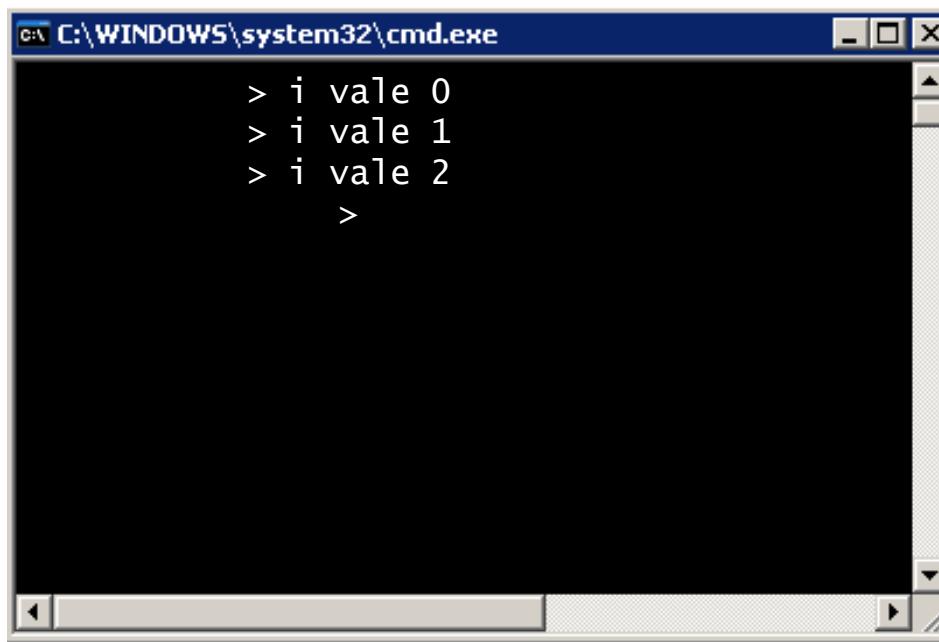


The screenshot shows a Windows command prompt window titled 'cmd.exe' with the path 'C:\WINDOWS\system32'. The window displays the following text:
> i vale 0
> i vale 1
> i vale 2
>

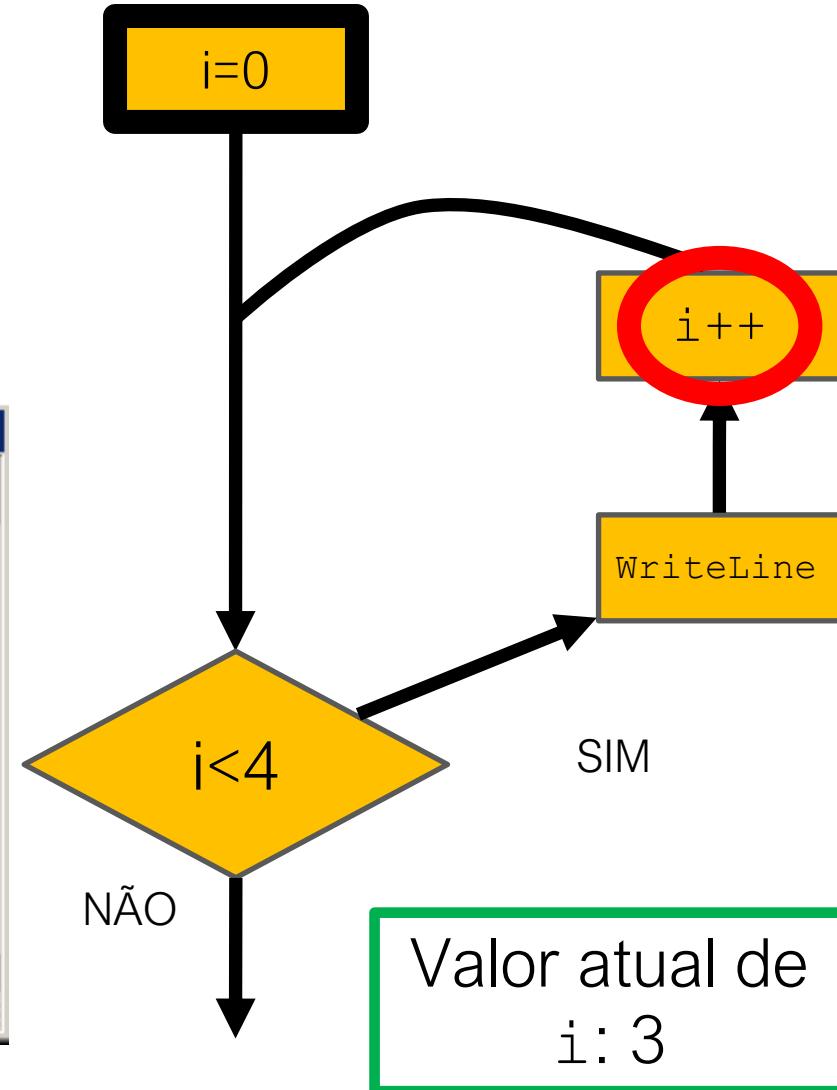


Entendendo o Exemplo 1

```
int i;  
for (i = 0; i < 4; i++)  
{  
    Console.WriteLine("i vale "+i);  
}
```



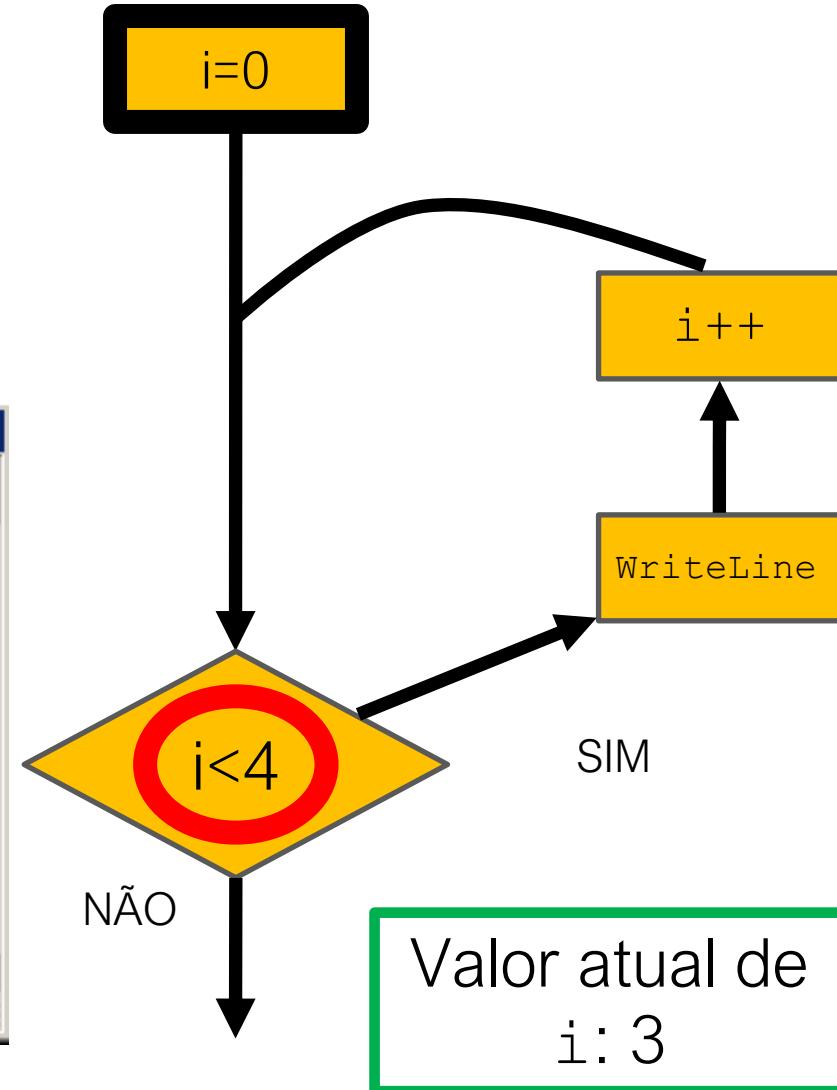
A screenshot of a Windows command prompt window titled 'cmd.exe' located at 'C:\WINDOWS\system32'. The window displays the following text:
> i vale 0
> i vale 1
> i vale 2
>



Entendendo o Exemplo 1

```
int i;  
for (i = 0; i < 4; i++)  
{  
    Console.WriteLine("i vale "+i);  
}
```

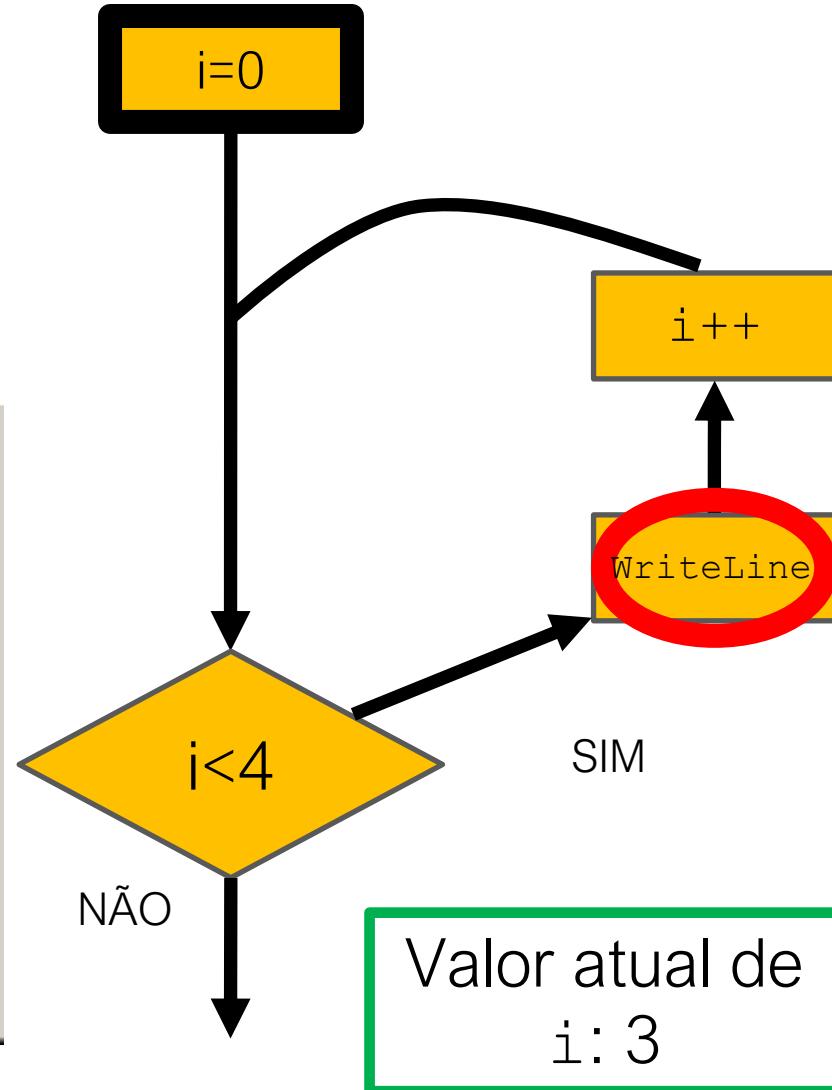
```
c:\> C:\WINDOWS\system32\cmd.exe  
> i vale 0  
> i vale 1  
> i vale 2  
>
```



Entendendo o Exemplo 1

```
int i;  
for (i = 0; i < 4; i++)  
{  
    Console.WriteLine("i vale "+i);  
}
```

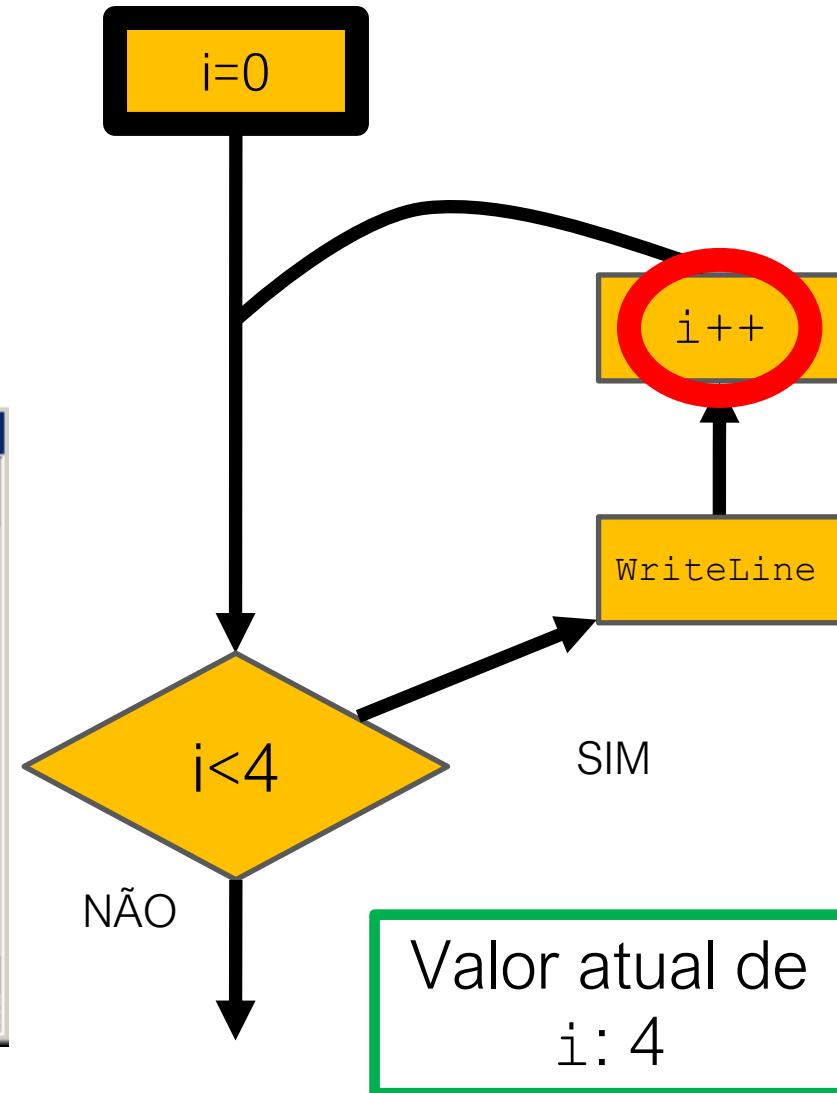
```
c:\> C:\WINDOWS\system32\cmd.exe  
> i vale 0  
> i vale 1  
> i vale 2  
> i vale 3  
>
```



Entendendo o Exemplo 1

```
int i;  
for (i = 0; i < 4; i++)  
{  
    Console.WriteLine("i vale "+i);  
}
```

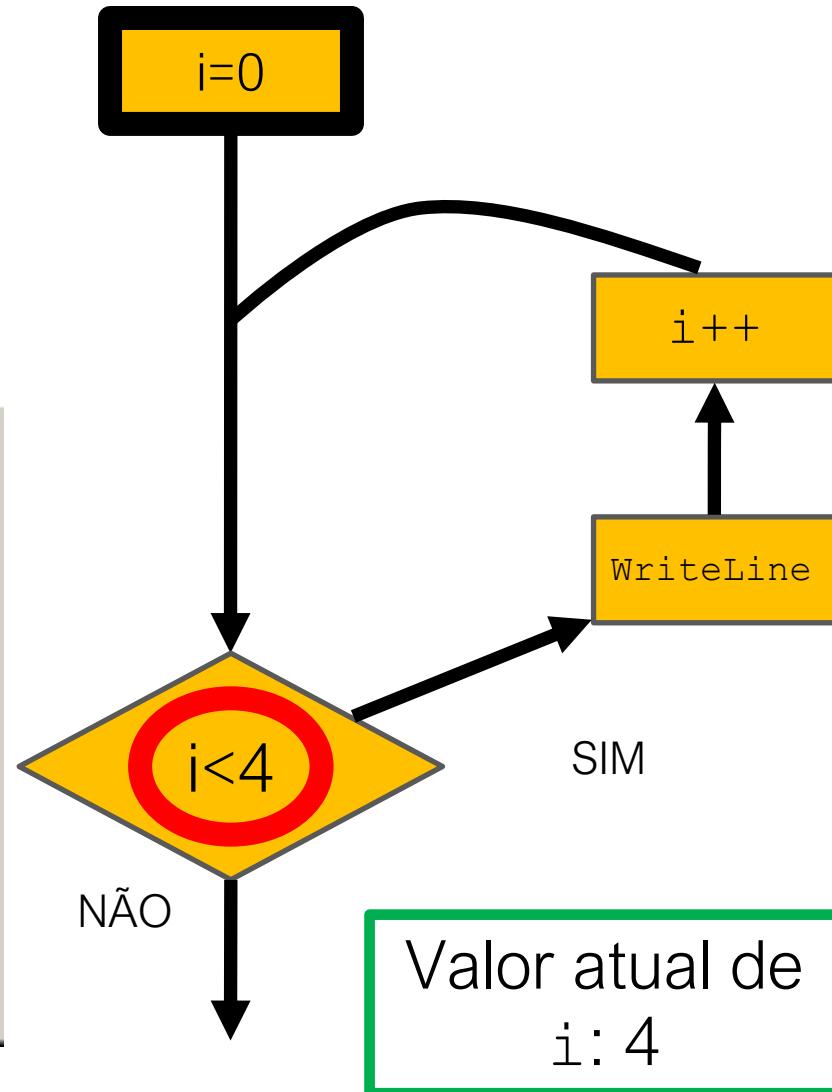
```
c:\> C:\WINDOWS\system32\cmd.exe  
> i vale 0  
> i vale 1  
> i vale 2  
> i vale 3  
>
```



Entendendo o Exemplo 1

```
int i;  
for (i = 0; i < 4 i++)  
{  
    Console.WriteLine("i vale "+i);  
}
```

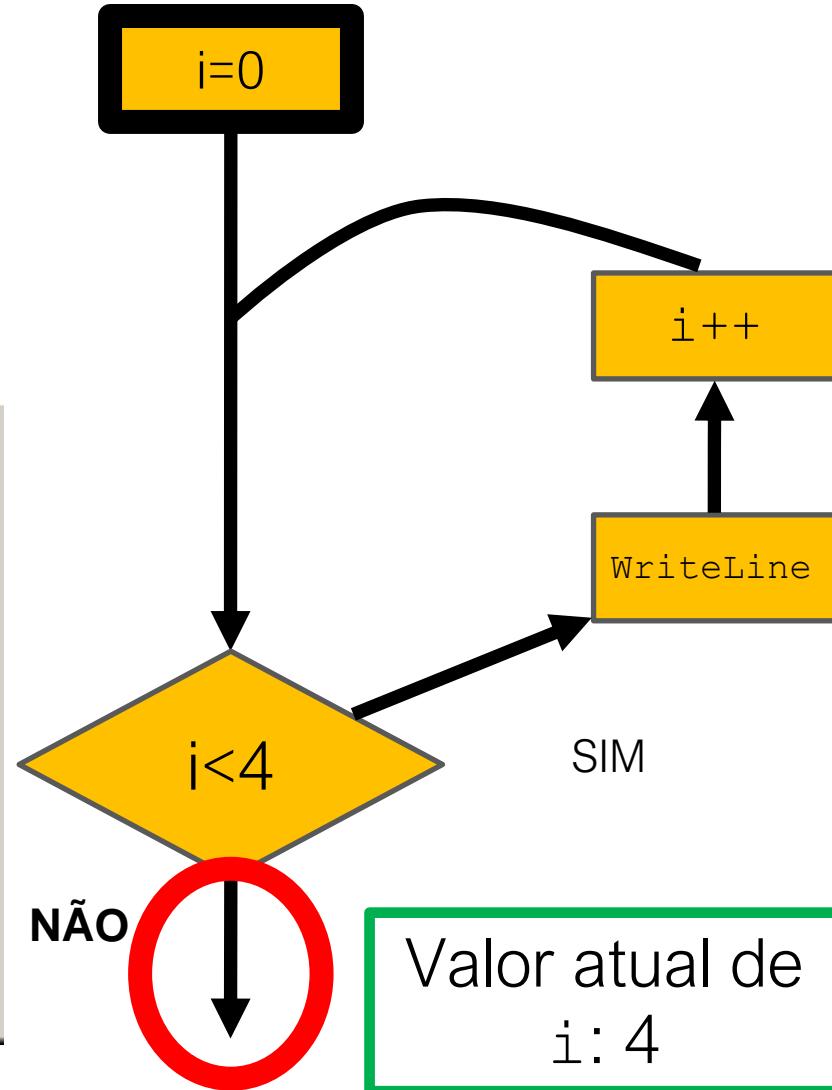
```
c:\> C:\WINDOWS\system32\cmd.exe  
> i vale 0  
> i vale 1  
> i vale 2  
> i vale 3  
>
```



Entendendo o Exemplo 1

```
int i;  
for (i = 0; i < 4 i++)  
{  
    Console.WriteLine("i vale "+i);  
}
```

```
c:\> C:\WINDOWS\system32\cmd.exe  
> i vale 0  
> i vale 1  
> i vale 2  
> i vale 3  
>
```



For – controle do laço

- O comando **break** faz com que todo laço de repetição seja finalizado, antes mesmo da condição do for se tornar falsa.
- O comando **continue** faz com que a interação atual do laço de repetição seja finalizada, e então seguir para a próxima interação.

Exemplo 1

- Escreva um algoritmo leia continuamente pelo teclado o valor de uma variável inteira chamada op, e só finalize a execução do programa quando o valor lido para op seja zero.

Exemplo 1 – comando **break**

```
static void Main(string[] args)
{
    int i, op;
    for (i = 0; true; i++)
    {
        Console.WriteLine("Quer continuar? \nDigite 0 para sair!");
        op = int.Parse(Console.ReadLine());
        if (op == 0)
        {
            Console.WriteLine("Obrigado pela presença neste lindo laço de repetição");
            break;
        }
    }
}
```

O comando **break** quebra o laço, indo para a próxima linha após o bloco **for**, finalizando a sua execução

Exemplo 2

- Escreva um algoritmo leia pelo teclado o valor de uma variável inteira chamada opcao, porém, o programa só pode seguir sua execução caso o valor de opcao seja entre 1 e 4. Se o valor digitado não for entre 1 e 4, a variável opcao deverá ser lida novamente pelo teclado.

Exemplo 2 – comando **continue**

```
static void Main(string[] args)
{
    int i, op;
    for (i = 0; true; i++)
    {
        Console.WriteLine("Digite sua opção:");
        op = int.Parse(Console.ReadLine());
        if (op > 4 || op < 1)
        {
            Console.WriteLine("Digite um valor entre 1 e 4!!! \n");
            continue;
        }
        else
        {
            Console.WriteLine("Opção "+op+" escolhida!");
            break;
        }
    }
    switch (op)
    {
        case 1:
            Console.WriteLine("Primeira opção");
            break;
        case 2:
            Console.WriteLine("Segunda opção");
            break;
        case 3:
            Console.WriteLine("Terceira opção");
            break;
        case 4:
            Console.WriteLine("Quarta opção");
            break;
    }
}
```

E comando **continue** manda seguir no laço para a próxima interação direto, ou seja, **vai direto pro incremento/decremento (*i++*)**, ignorando todo o resto do bloco!:

Exemplo 3 – for dentro de um for

```
static void Main(string[] args)
{
    int i, j;
    for (i = 0; i < 20; i++)
    {
        Console.WriteLine("i: " + i);
        for (j = 0; j < 10; j++)
        {
            Console.WriteLine("    j: " + j);
        }
    }
}
```

Exemplo 4 – for dentro de um for dentro de um for

```
static void Main(string[] args)
{
    int i, j, k;
    for (i = 0; i < 20; i++)
    {
        Console.WriteLine("i: " + i);
        for (j = 0; j < 10; j++)
        {
            Console.WriteLine("      j: " + j);
            for (k = 0; k < 15; k++)
            {
                Console.WriteLine("          k: " + k);
            }
        }
    }
}
```