

Curso C# Completo Programação Orientada a Objetos + Projetos

Capítulo: Recapitulação de Lógica de Programação usando C#
<http://educandoweb.com.br>
Prof. Dr. Nelio Alves

Objetivo:

Construir programas básicos, aplicando o básico de Lógica de Programação, usando C# como linguagem

- Tipos de dados básicos em C#
- Estrutura sequencial (*entrada, processamento, saída*)
- Operadores (*aritméticos, comparativos, lógicos*)
- Estruturas de controle (*if-else, while, for*)

Dica para iniciantes



C# primeiros passos: Lógica de Programação e Algoritmos

Dica para experientes



Tipos básicos em C#

<http://educandoweb.com.br>
Prof. Dr. Nelio Alves

Checklist

<https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/built-in-types-table>

- C# é uma linguagem **estaticamente tipada**
- Tipos valor pré-definidos em C#
- Tipos referência pré-definidos em C#
- Variável não atribuída
- Overflow
- Padrão para float: sufixo "f"
- Padrão para char: aspas simples
- Padrão para string: aspas duplas
- Padrão para bool: true, false
- Opção: inferência de tipos com palavra "var" (dentro de métodos)

C# built-in data types (tipos valor)

C# Type	.Net Framework Type	Signed	Bytes	Possible Values
sbyte	System.SByte	Yes	1	-128 to 127
short	System.Int16	Yes	2	-32768 to 32767
int	System.Int32	Yes	4	-2^{31} to $2^{31} - 1$
long	System.Int64	Yes	8	-2^{63} to $2^{63} - 1$
byte	System.Byte	No	1	0 to 255
ushort	System.UInt16	No	2	0 to 65535
uint	System.UInt32	No	4	0 to $2^{32} - 1$
ulong	System.UInt64	No	8	0 to $2^{64} - 1$
float	System.Single	Yes	4	$\pm 1.5 \times 10^{-45}$ to $\pm 3.4 \times 10^{38}$ with 7 significant figures
double	System.Double	Yes	8	$\pm 5.0 \times 10^{-324}$ to $\pm 1.7 \times 10^{308}$ with 15 or 16 significant figures
decimal	System.Decimal	Yes	12	$\pm 1.0 \times 10^{-28}$ to $\pm 9.9 \times 10^{27}$ with 28 or 29 significant figures
char	System.Char	N/A	2	Any Unicode character
bool	System.Boolean	N/A	1/2	true or false

1 byte = 8 bit armazena 256 valores, pois cada bit pode ser 0 ou 1, resultando em $2^8 = 256$ combinações possíveis.

• Os mais utilizados

overflow = É quando o cálculo estoura o limite do tipo declarado

C# built-in data types (tipos referência)

Tipo C#	Tipo .NET	Descrição
string	System.String	Uma cadeia de caracteres Unicode IMUTÁVEL (segurança, simplicidade, thread safe)
object	System.Object	Um objeto genérico (toda classe em C# é subclasse de object) GetType Equals GetHashCode ToString

String é imutável!!

Você atribuir outro valor a variável não significa mudá-la.

Demo

```
bool completo = false;
char genero = 'F';
char letra = '\u0041';
byte n1 = 126;
int n2 = 1000;
int n3 = 2147483647;
long n4 = 2147483648L;
float n5 = 4.5f;
double n6 = 4.5;
String nome = "Maria Green";
Object obj1 = "Alex Brown";
Object obj2 = 4.5f;

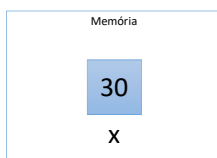
Console.WriteLine(completo);
Console.WriteLine(genero);
Console.WriteLine(letra);
Console.WriteLine(n1);
Console.WriteLine(n2);
Console.WriteLine(n3);
Console.WriteLine(n4);
Console.WriteLine(n5);
Console.WriteLine(n6);
Console.WriteLine(nome);
Console.WriteLine(obj1);
Console.WriteLine(obj2);
```

Funções para valores mínimos e máximos

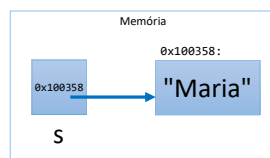
- `int.MinValue`
- `int.MaxValue`
- `sbyte.MaxValue`
- `long.MaxValue`
- `decimal.MaxValue`
- etc...

Tipo valor vs. tipo referência

`int x = 30;`



`string s = "Maria";`



Restrições e convenções para nomes

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Restrições para nomes de variáveis

- Não pode começar com dígito: use uma letra ou _
- Não usar acentos ou til
- Não pode ter espaço em branco
- Sugestão: use nomes que tenham um significado

Errado:

```
int 5minutos;  
int salário;  
int salario do funcionario;
```

Correto:

```
int _5minutos;  
int salario;  
int salarioDoFuncionario;
```

Convenções

- Camel Case: lastName (parâmetros de métodos, variáveis dentro de métodos)
- Pascal Case: LastName (namespaces, classe, properties e métodos)
- Padrão _lastName (atributos "internos" da classe)

```
namespace Curso
{
    class ContaBancaria
    {
        public string Titular { get; set; }
        private double _saldo;

        public void Deposito(double quantia)
        {
            _saldo += quantia;
        }

        public double GetSaldo()
        {
            return _saldo;
        }
    }
}
```

Conversão implícita e casting

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Checklist

- Conversão implícita entre tipos
<https://docs.microsoft.com/pt-br/dotnet/csharp/language-reference/keywords/implicit-numeric-conversions-table>
- Casting: conversão explícita entre tipos COMPATÍVEIS

Exemplo 1

```
double a;  
float b;  
  
a = 5.1;  
b = (float)a;  
  
Console.WriteLine(b);
```

Exemplo 2

```
double a;  
int b;  
  
a = 5.1;  
b = (int)a;  
  
Console.WriteLine(b);
```

Exemplo 3

```
int a = 5;  
int b = 2;  
  
double resultado = (double) a / b;  
  
Console.WriteLine(resultado);
```

Saída de dados em C#

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Imprimir na saída padrão (console)

- Comandos
 - `Console.WriteLine(valor);`
 - `Console.Write(valor);`

Demo

```
using System;
using System.Globalization;

namespace PrimeiroProjeto {
    class Program {
        static void Main(string[] args) {

            char genero = 'F';
            int idade = 32;
            double saldo = 10.35784;
            String nome = "Maria";

            Console.WriteLine("Bom dia!");
            Console.WriteLine("Boa tarde!");
            Console.WriteLine("Boa noite!");
            Console.WriteLine("-----");
            Console.WriteLine(genero);
            Console.WriteLine(idade);
            Console.WriteLine(saldo);
            Console.WriteLine(nome);
            Console.WriteLine(saldo.ToString("F2"));
            Console.WriteLine(saldo.ToString("F4"));
            Console.WriteLine(saldo.ToString("F4", CultureInfo.InvariantCulture));

        }
    }
}
```

Placeholders, concatenação e interpolação

Demo:

```
int idade = 32;
double saldo = 10.35784;
String nome = "Maria";

Console.WriteLine("{0} tem {1} anos e tem saldo igual a {2:F2} reais", nome, idade, saldo);

Console.WriteLine($"{nome} tem {idade} anos e tem saldo igual a {saldo:F2} reais");

Console.WriteLine(nome + " tem " + idade + " anos e tem saldo igual a "
    + saldo.ToString("F2", CultureInfo.InvariantCulture) + " reais");
```

Exercício de fixação

Em um novo programa, inicie as seguintes variáveis:

```
string produto1 = "Computador";
string produto2 = "Mesa de escritório";

byte idade = 30;
int codigo = 5290;
char genero = 'M';

double preco1 = 2100.0;
double preco2 = 650.50;
double medida = 53.234567;
```

Em seguida, usando os valores das variáveis, produza a seguinte saída na tela do console:

```
Produtos:
Computador, cujo preço é $ 2100,00
Mesa de escritório, cujo preço é $ 650,50

Registro: 30 anos de idade, código 5290 e gênero: M

Medida com oito casas decimais: 53,23456700
Arredondado (três casas decimais): 53,235
Separador decimal invariant culture: 53.235
```

(correção na próxima página)

```
using System;
using System.Globalization;

namespace Course {
    class Program {
        static void Main(string[] args) {
            string produto1 = "Computador";
            string produto2 = "Mesa de escritório";

            byte idade = 30;
            int codigo = 5290;
            char genero = 'M';

            double preco1 = 2100.0;
            double preco2 = 650.50;
            double medida = 53.234567;

            Console.WriteLine("Produtos:");
            Console.WriteLine($"{0}, cujo preço é $ {1:F2}", produto1, preco1);
            Console.WriteLine($"{0}, cujo preço é $ {1:F2}", produto2, preco2);
            Console.WriteLine();
            Console.WriteLine("Registro: {0} anos de idade, código {1} e gênero: {2}", idade, codigo, genero);
            Console.WriteLine();
            Console.WriteLine("Medida com oito casas decimais: {0:F8}", medida);
            Console.WriteLine("Arredondado (três casas decimais): {0:F3}", medida);
            Console.WriteLine("Separador decimal invariant culture: " + medida.ToString("F3", CultureInfo.InvariantCulture));
        }
    }
}
```

Operadores aritméticos

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Operadores aritméticos

Operador	Significado
+	adição
-	subtração
*	multiplicação
/	divisão
%	resto da divisão

NOTAS:

1) * / % tem precedência maior que + -

2) Exemplos:
3 + 4 * 2 → Resultado: 11
(3 + 4) * 2 → Resultado: 14

3) Pode-se usar parêntesis à vontade

4) Exemplo com mod:
17 % 3 → Resultado: 2

$$\begin{array}{r} 17 \text{ } | \text{ } 3 \\ 2 \text{ } 5 \end{array}$$

Demo

```
int n1 = 3 + 4 * 2;  
int n2 = (3 + 4) * 2;  
int n3 = 17 % 3;  
double n4 = 10.0 / 8.0;  
  
double a = 1.0, b = -3.0, c = -4.0;  
  
double delta = Math.Pow(b, 2.0) - 4.0 * a * c;  
  
double x1 = (-b + Math.Sqrt(delta)) / (2.0 * a);  
double x2 = (-b - Math.Sqrt(delta)) / (2.0 * a);  
  
Console.WriteLine(n1);  
Console.WriteLine(n2);  
Console.WriteLine(n3);  
Console.WriteLine(n4);  
Console.WriteLine(delta);  
Console.WriteLine(x1);  
Console.WriteLine(x2);
```

$$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

Operadores de atribuição

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Operadores de atribuição

Operador	Exemplo	Significado
=	a = 10;	a RECEBE 10
+=	a += 2;	a RECEBE a + 2;
-=	a -= 2;	a RECEBE a - 2;
*=	a *= 2;	a RECEBE a * 2;
/=	a /= 2;	a RECEBE a / 2;
%=	a %= 3;	a RECEBE a % 3;

Demo

```
int a = 10;
Console.WriteLine(a);

a += 2;
Console.WriteLine(a);

a *= 3;
Console.WriteLine(a);

string s = "ABC";
Console.WriteLine(s);

s += "DEF";
Console.WriteLine(s);
```

Operadores aritméticos / atribuição

Operador	Exemplo	Significado
++	a++; ou ++a;	a = a + 1;
--	a--; ou --a;	a = a - 1;

```
int a = 10;
a++;
Console.WriteLine(a);

SAÍDA:
11
```

```
int a = 10;
int b = a++;
Console.WriteLine(a);
Console.WriteLine(b);

SAÍDA:
11
10
```

```
int a = 10;
int b = ++a;
Console.WriteLine(a);
Console.WriteLine(b);

SAÍDA:
11
11
```

Entrada de dados em C# - Parte 1

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Comando

```
Console.ReadLine();
```

- Lê da entrada padrão **até a quebra de linha**.
- Retorna os dados lidos **na forma de string**.

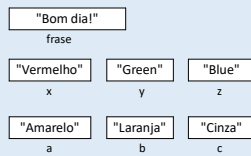
Checklist

- Ler um texto até a quebra de linha e armazenar em uma variável
- Ler três palavras, uma em cada linha, armazenando cada uma em uma variável
- Ler três palavras na mesma linha, separadas por espaço, armazenando cada uma em uma variável

Console:

```
Bom dia!  
Vermelho  
Verde  
Azul  
Amarelo Laranja Cinza
```

Memória:

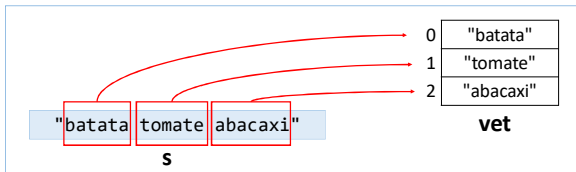
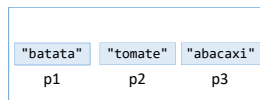


Split

```
string s = Console.ReadLine();  
  
string[] vet = s.Split(' ');  
string p1 = vet[0];  
string p2 = vet[1];  
string p3 = vet[2];
```

Console:

```
batata tomate abacaxi
```



Demo

```
string frase = Console.ReadLine();  
string x = Console.ReadLine();  
string y = Console.ReadLine();  
string z = Console.ReadLine();  
  
string[] v = Console.ReadLine().Split(' ');  
string a = v[0];  
string b = v[1];  
string c = v[2];  
  
Console.WriteLine("Você digitou: ");  
Console.WriteLine(frase);  
Console.WriteLine(x);  
Console.WriteLine(y);  
Console.WriteLine(z);  
Console.WriteLine(a);  
Console.WriteLine(b);  
Console.WriteLine(c);
```

Entrada de dados em C# - Parte 2

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

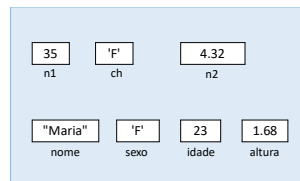
Checklist

- Ler um número inteiro
- Ler um caractere
- Ler um número double
- Ler um nome (única palavra), sexo (caractere F ou M), idade (inteiro) e altura (double) na mesma linha, armazenando-os em quatro variáveis com os devidos tipos

Console:

```
35
F
4.32
Maria F 23 1.68
```

Memória:



Demo

```
int n1 = int.Parse(Console.ReadLine());
char ch = char.Parse(Console.ReadLine());
double n2 = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);

string[] vet = Console.ReadLine().Split(' ');
string nome = vet[0];
char sexo = char.Parse(vet[1]);
int idade = int.Parse(vet[2]);
double altura = double.Parse(vet[3], CultureInfo.InvariantCulture);

Console.WriteLine("Você digitou:");
Console.WriteLine(n1);
Console.WriteLine(ch);
Console.WriteLine(n2.ToString("F2", CultureInfo.InvariantCulture));
Console.WriteLine(nome);
Console.WriteLine(sexo);
Console.WriteLine(idade);
Console.WriteLine(altura.ToString("F2", CultureInfo.InvariantCulture));
```

Exercício de fixação

Fazer um programa para executar a seguinte interação com o usuário, lendo os valores destacados em vermelho, e depois mostrar os dados na tela:

```
Entre com seu nome completo:
Alex Green
Quantos quartos tem na sua casa?
3
Entre com o preço de um produto:
500.50
Entre seu último nome, idade e altura (mesma linha):
Green 21 1.73
```

SAÍDA ESPERADA (NÚMEROS REAIS
COM DUAS CASAS DECIMAIS):

```
Alex Green
3
500.50
Green
21
1.73
```

(correção na próxima página)

```
using System;
using System.Globalization;
namespace Course {
    class Program {
        static void Main(string[] args) {
            Console.WriteLine("Entre com seu nome completo:");
            string fullName = Console.ReadLine();
            Console.WriteLine("Quantos quartos tem na sua casa?");
            int bedrooms = int.Parse(Console.ReadLine());
            Console.WriteLine("Enter product price:");
            double price = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            Console.WriteLine("Entre seu último nome, idade e altura (mesma linha):");
            string[] vect = Console.ReadLine().Split(' ');
            string lastName = vect[0];
            int age = int.Parse(vect[1]);
            double height = double.Parse(vect[2], CultureInfo.InvariantCulture);

            Console.WriteLine(fullName);
            Console.WriteLine(bedrooms);
            Console.WriteLine(price.ToString("F2", CultureInfo.InvariantCulture));
            Console.WriteLine(lastName);
            Console.WriteLine(age);
            Console.WriteLine(height.ToString("F2", CultureInfo.InvariantCulture));
        }
    }
}
```

Exercícios propostos - PARTE 1

Estrutura sequencial (entrada, processamento, saída)

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Operadores comparativos

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Operadores comparativos

Operador	Significado
>	maior
<	menor
>=	maior ou igual
<=	menor ou igual
==	igual
!=	diferente

Demo

```
int a = 10;
bool c1 = a < 10;
bool c2 = a < 20;
bool c3 = a > 10;
bool c4 = a > 5;
Console.WriteLine(c1);
Console.WriteLine(c2);
Console.WriteLine(c3);
Console.WriteLine(c4);
Console.WriteLine("-----");

bool c5 = a <= 10;
bool c6 = a >= 10;
bool c7 = a == 10;
bool c8 = a != 10;
Console.WriteLine(c5);
Console.WriteLine(c6);
Console.WriteLine(c7);
Console.WriteLine(c8);
```

Operadores lógicos

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Operadores lógicos

Operador	Significado
&&	E
	OU
!	NÃO

C1	C2	C1 E C2
F	F	F
F	V	F
V	F	F
V	V	V

C1	C2	C1 OU C2
F	F	F
F	V	V
V	F	V
V	V	V

NOTAS:

1) Precedência: ! > && > ||

2) Pode-se usar parêntesis à vontade

3) Exemplos:

2 > 3 || 4 != 5 → Resultado: true
!(2>3) && 4 != 5 → Resultado: true

Demo

```
bool c1 = 2 > 3 || 4 != 5; // true
bool c2 = !(2 > 3) && 4 != 5; // true
Console.WriteLine(c1);
Console.WriteLine(c2);

Console.WriteLine("-----");

bool c3 = 10 < 5; // false
bool c4 = c1 || c2 && c3; // true
Console.WriteLine(c3);
Console.WriteLine(c4);
```


Estrutura condicional (if-else)

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Estrutura condicional

Simples

```
if ( condição ) {  
    comando 1  
    comando 2  
}
```

Composta

```
if ( condição ) {  
    comando 1  
    comando 2  
}  
else {  
    comando 3  
    comando 4  
}
```

Encadeamentos

```
if ( condição 1 ) {  
    comando 1  
    comando 2  
}  
else if ( condição 2 ) {  
    comando 3  
    comando 4  
}  
else if ( condição 3 ) {  
    comando 5  
    comando 6  
}  
else {  
    comando 7  
    comando 8  
}
```

Nota: se o bloco de comandos possuir apenas um comando, as chaves são opcionais.

Demo #1

Entre com um número inteiro:

10
Par!

Entre com um número ímpar?

15
Ímpar!

```
using System;

namespace Course {
    class Program {
        static void Main(string[] args) {

            Console.WriteLine("Entre com um número inteiro:");
            int x = int.Parse(Console.ReadLine());

            if (x % 2 == 0) {
                Console.WriteLine("Par!");
            }
            else {
                Console.WriteLine("Impar");
            }
        }
    }
}
```

Demo #2

Qual a hora atual?
10
Bom dia!

hora < 12

Qual a hora atual?
14
Boa tarde!

12 <= hora < 18

Qual a hora atual?
19
Boa noite!

hora >= 18

```
using System;

namespace Course {
    class Program {
        static void Main(string[] args) {

            Console.WriteLine("Qual a hora atual?");
            int hora = int.Parse(Console.ReadLine());

            if (hora < 12) {
                Console.WriteLine("Bom dia!");
            }
            else if (hora < 18) {
                Console.WriteLine("Boa tarde!");
            }
            else {
                Console.WriteLine("Boa noite!");
            }
        }
    }
}
```

Escopo e inicialização

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Checklist

- Escopo de uma variável: é a região do programa onde a variável é válida, ou seja, onde ela pode ser referenciada.
- Uma variável não pode ser usada se não for iniciada.
- Falaremos de escopo de métodos no Capítulo "Comportamento de memória, arrays, listas"

Demo

```
double preco = double.Parse(Console.ReadLine());  
  
if (preco > 100.0) {  
    double desconto = preco * 0.1;  
}  
  
Console.WriteLine(desconto);
```

Exercícios propostos - PARTE 2

Estrutura condicional (if-else)

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Funções (sintaxe)

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Funções

- Representam um processamento que possui um significado
 - `Math.Sqrt(double)`
 - `Console.WriteLine(string)`
- Principais vantagens: modularização, delegação e reaproveitamento
- Dados de entrada e saída
 - Funções podem receber dados de entrada (parâmetros ou argumentos)
 - Funções podem ou não retornar uma saída
- Em orientação a objetos, funções em classes recebem o nome de "métodos"

Problema exemplo

Fazer um programa para ler três números inteiros e mostrar na tela o maior deles.

Exemplo:

```
Digite três números:  
5  
8  
3  
Maior = 8
```

```
using System;  
  
namespace Course {  
    class Program {  
        static void Main(string[] args) {  
  
            Console.WriteLine("Digite três números:");  
            int n1 = int.Parse(Console.ReadLine());  
            int n2 = int.Parse(Console.ReadLine());  
            int n3 = int.Parse(Console.ReadLine());  
  
            double resultado = Maior(n1, n2, n3);  
  
            Console.WriteLine("Maior = " + resultado);  
        }  
  
        static int Maior(int a, int b, int c) {  
            int m;  
            if (a > b && a > c) {  
                m = a;  
            }  
            else if (b > c) {  
                m = b;  
            }  
            else {  
                m = c;  
            }  
            return m;  
        }  
    }  
}
```

Debugging com Visual Studio

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Tópicos

- Teclas
 - F9 - marcar/desmarcar breakpoint
 - F5 - iniciar/continuar o debug
 - F10 - executar um passo (pula função)
 - F11 - executar um passo (entra na função)
 - SHIFT+F11 - sair do método em execução
 - SHIFT+F5 - parar debug
- Janelas
 - Watch (expressões personalizadas)
 - Autos (expressões "interessantes" detectadas pelo Visual Studio)
 - Locals (variáveis locais)

Estrutura repetitiva while

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Estrutura "enquanto"

```
while ( condição ) {  
    comando 1  
    comando 2  
}
```

Regra:

V: executa e volta
F: pula fora

Problema exemplo

Digitar um número e mostrar sua raiz quadrada com três casas decimais, depois repetir o procedimento. Quando o usuário digitar um número negativo (podendo inclusive ser na primeira vez), mostrar uma mensagem "Número negativo" e terminar o programa.

```
Digite um número: 25
5.000
Digite outro número: 10
3.162
Digite outro número : 9
3.000
Digite outro número : -4
Número negativo!
```

```
using System;
using System.Globalization;

namespace Course {
    class Program {
        static void Main(string[] args) {
            Console.Write("Digite um número: ");
            double x = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);

            while (x >= 0.0) {
                double raiz = Math.Sqrt(x);
                Console.WriteLine(raiz.ToString("F3", CultureInfo.InvariantCulture));
                Console.Write("Digite outro número: ");
                x = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            }

            Console.WriteLine("Número negativo!");
        }
    }
}
```

Exercícios propostos - PARTE 3

Estrutura while

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Estrutura repetitiva for

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Estrutura "para"

Executa somente
na primeira vez

V: executa e volta
F: pula fora

Executa toda vez depois
de voltar

```
for ( início ; condição ; incremento ) {  
    comando 1  
    comando 2  
}
```

Demo

Digitar um número N e depois N valores inteiros. Mostrar a soma dos N valores digitados.

```
Quantos números inteiros você vai digitar? 3  
Valor #1: 10  
Valor #2: 7  
Valor #3: 8  
Soma = 25
```

```
using System;
namespace Course {
    class Program {
        static void Main(string[] args) {

            Console.WriteLine("Quantos números inteiros você vai digitar? ");
            int N = int.Parse(Console.ReadLine());

            int soma = 0;
            for (int i = 1; i <= N; i++) {
                Console.WriteLine("Valor #{0}: ", i);
                int valor = int.Parse(Console.ReadLine());
                soma += valor;
            }

            Console.WriteLine("Soma = " + soma);
        }
    }
}
```

Exercícios propostos - PARTE 4

Estrutura for

<http://educandoweb.com.br>

Prof. Dr. Nélcio Alves
