

Programação Visual

Trabalho de Laboratório S02 - Tutorial

Nível 1

1. Criar Solução/Projeto (ver Figura 1) em que _NA são as iniciais dos seus primeiro nome e apelido.

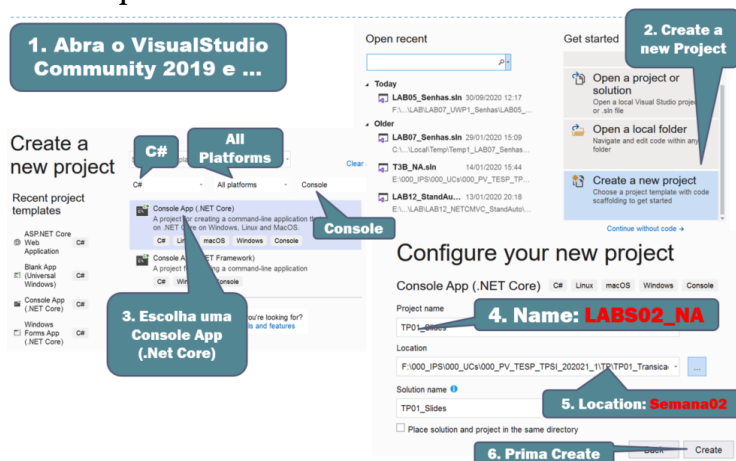


Figura 1: criação de uma solução

Nível 2

Estrutura dos programas

1. Compile, corra o programa e verifique se a saída coincide com as suas expectativas.
2. Adicione os comentários a negro carregado da Figura 2
3. Compile, elimine todos os erros sintáticos, corra o programa e verifique se a saída coincide com as suas expectativas, caso contrário corrija e volte a correr até obter o resultado desejado.
4. Logo abaixo da instrução que imprime na consola a frase “Hello World” imprima o texto abaixo usando apenas um **Console.WriteLine**, a combinação **\n** para mudar de linha e o operador de concatenação de strings **+** caso necessite:

\nNote que o Main em C# segue a convenção de que todos os identificadores públicos comecem por maiúsculas (notação PascalCase)

```
// Tal como em Java, as linhas antecedidas por //
// não são consideradas código pelo compilador
// Para comentar e descomentar código no VS
// selecciona-se o código e Ctrl k + Ctrl C

// Em C# usam-se namespaces com uma função equivalente
// aos packages do JAVA.
// A diretiva using do C# é semelhante ao import do JAVA e
// permite usar tipos de outro namespace
// sem ter de os especificar

using System;
namespace LABS01_NA
{
    // Em C# os comentários em XML, equivalentes ao Javadoc
    // do Java, iniciam-se por /// e originam ficheiros em XML

    /// <summary>
    /// ola mundo
    /// </summary>
```

(continua ...->)

Programação Visual

Trabalho de Laboratório S02 - Tutorial

```
(-> ... continuação)

public class Program
{

    public static void Main(string[] args)
    {
        Console.WriteLine("Olá Mundo!");

        Console.WriteLine("Prima return para continuar");
        Console.ReadKey();
    }
}
```

Figura 2: código para o nível 2

Nível 3

Tipo de Dados

1. Na linha a seguir à instrução para ler uma tecla do teclado insira o código da Figura 3
2. Compile, elimine todos os erros sintáticos e corra o programa.

```
Console.WriteLine("\nXXXXXX - TIPOS DE DADOS - Slide 7" +
    "\n\nUm tipo de dados define um conjunto de valores e as" +
    "\n    operações que podemos executar com esses valores");
Console.WriteLine("prima return para continuar");
Console.ReadKey();
```

Figura 2: código para o nível 2

Nível 4A

Identificadores: PascalCase e CameCase

1. Na linha a seguir à instrução para ler uma tecla do teclado insira o código da Figura 4A
2. Compile, elimine todos os erros sintáticos e corra o programa.

```
Console.WriteLine("\nXXXXXX - IDENTIFICADORES Slide 7" +
    "\n\nUm identificador é um nome que damos " +
    "\n    a uma entidade de um programa de c#" +
    "\n    como classes, variáveis ou métodos" +
    "\nEm C# todos as entidades públicas (ou none) " +
    "\n    têm os seus nomes em PascalCase " +
    "\n    (PrimeiraLetraDeCadaPalavraMaiuscula" +
    "\n\nEm C# as entidades que não sejam públicas (ou none) " +
    "\n    têm os seus nomes em camelCase " +
    "\n    (primeiraLetraDaPrimeiraPalavraMinuscula");
Console.WriteLine("\nMandam as práticas de bem programar que" +
    "\n\nNão usemos no código caracteres específicos do português" +
    "\n    mas podemos usá-los nos comentários ou nas saídas" +
    "\n    Exemplo: não usar Ç ã, á, í etc." +
    (continua ...->)
```

Programação Visual

Trabalho de Laboratório S02 - Tutorial

```
(-> ... continuação)
"\n\nNão usemos acrónimos nem abreviaturas nos identificadores" +
"\n    Exemplo: não usar num para numero, opc para opcao etc." +
"\n\nEscolhemos cuidadosamente os nomes das variáveis " +
"\n    por forma a identificarem a entidade envolvida." +
"\n\nEscolhemos verbos para ações, procedimento e funções" +
"\n    Exemplo: CalculaMedia, TrocaValores" +
"\n\nEscolhemos substantivos para classes, e variáveis" +
"\n    Exemplo: Aluno, Escola, posicao, media " );

Console.WriteLine("\nprima return para continuar");
Console.ReadKey();
```

Figura 4: código para o nível 4

Nível 4B

Tipo de Dados por Valor e por Referência

3. Na linha a seguir à instrução para ler uma tecla do teclado insira o código da Figura 4B
4. Compile, elimine todos os erros sintáticos e corra o programa.

```
Console.WriteLine("\nXXXXXX - TIPOS DE DADOS POR VALOR E POR
REFERÊNCIA - Slide 8" +
"\n\ntipo por valor (int, double char etc): " +
"\n    na variável é escrito um valor" +
"\n\ntipo por referência (Int32, Double, Char etc): " +
"\n    na variável é escrita uma referência " +
"\n    (o endereço de memória onde está o objeto)" +
"\n\nEm c# os tipos simples (incorporados ou built-in) " +
"\n    podem ser vistos como nomes alternativos (alias)" +
"\n    para as classes existentes." +
"\n\nOs tipos simples são tipos por valor " +
"\n    significando que as suas variáveis armazenam valores");
Console.WriteLine("prima return para continuar");
Console.ReadKey();
```

Figura 4A: código para o nível 4A

Nível 4C

Instruções e operadores de atribuição

5. Na linha a seguir à instrução para ler uma tecla do teclado:
 - a) Declare uma variável inteira denominada variavel1;
 - b) Numa outra instrução atribua-lhe o valor 1;
 - c) Declare uma variável inteira denominada variavel2 e
 - d) Atribua-lhe o valor 2, tudo na mesma instrução;
 - e) Declare uma variável inteira denominada tmp e
 - f) Atribua-lhe a variavel1;
 - g) Atribua a variavel2 à variavel1;

(continua ...->)

Programação Visual

Trabalho de Laboratório S02 - Tutorial

- h) Atribua tmp à variavel2;
 - i) Imprima na consola os valores das variáveis 1 e 2 no formato da Figura 14C.
 - k) Incremente em um a variavel1 usando o operador ++;
 - l) Decrementa em um a variavel2 usando o operador --;
 - m) Imprima na consola os valores das variáveis 1 e 2 no formato da Figura 14C.
 - n) Incremente em dez a variavel1 usando o operador +=;
 - o) Decrementa em vinte a variavel2 usando o operador -=;
 - p) Imprima na consola os valores das variáveis 1 e 2 no formato da Figura 14C.
 - r) Insira as duas instruções abaixo:
`Console.WriteLine("\n\nprima return para continuar");`
`Console.ReadKey();`
6. Compile, elimine todos os erros sintáticos e corra o programa.

```
XXXXX - slide 9 - Instruções e operadores de atribuição
variável 1= 2
variável 2= 1
variável 1= 3
variável 2= 0
variável 1= 13
variável 2= -20

prima return para continuar
```

Figura 4: Saída na consola do nível 4C

Nível 5

Estruturas de Selecção – Operador Condicional

1. Na linha a seguir à instrução para ler uma tecla do teclado insira o código da Figura 5
2. Compile, elimine todos os erros sintáticos e corra o programa.
3. Na linha a seguir à instrução para ler uma tecla do teclado:
 - a) Usando o operador condicional verifique se variavel1 é maior que 10 e, caso o seja imprima na consola **maior que dez**, caso contrário imprima na consola **menor ou igual a dez**.
 - b) Em seguida imprima o valor da variavel1.
 - c) Usando o operador condicional atribua à variavel2 o valor **-1** caso ela seja negativa ou **1** caso ela seja positiva ou zero.
 - d) Em seguida imprima o valor da variavel2.
 - e) Insira as duas instruções abaixo:
`Console.WriteLine("\n\nprima return para continuar");`
`Console.ReadKey();`

(continua ...->)

Programação Visual

Trabalho de Laboratório S02 - Tutorial

```
Console.WriteLine("\nXXXXXX slide 10 - Operador condicional ? :");
Console.Write("\nEscreva o nome de um mês em minúsculas >");
string str = Console.ReadLine();
Console.WriteLine(str == "fev" ? "tem 28 dias" : "tem 30 ou 31 dias");
Console.WriteLine("prima return para continuar");
Console.ReadKey();
```

Figura 5: código para o nível 5

Nível 6

Estruturas de Selecção – Switch

1. Na linha a seguir à instrução para ler uma tecla do teclado insira o código da Figura 4
2. Compile, elimine todos os erros sintáticos e corra o programa.
3. Na linha a seguir à instrução para ler uma tecla do teclado:
 - a) Peça ao utilizador para digitar um inteiro de **um a cinco**;
 - b) Leia o caracter digitado para uma String denominada **stringTmp_NA**, onde **_NA** são as iniciais dos seus primeiro nome e apelido.
 - c) Converta a String no inteiro equivalente usando a instrução **int.Parse(stringTmp_NA)** e atribua esse valor a um int denominado **inteiro_NA.**, onde **_NA** são as iniciais dos seus primeiro nome e apelido.
 - d) Use uma estrutura switch para imprimir na consola o nome em maiúsculas do inteiro inserido: UM, DOIS, TRÊS etc... e caso não seja nenhuma das opções imprima a String "DEU BRONCA".
 - e) Insira as duas instruções abaixo:
Console.WriteLine("\n\nprima return para continuar");
Console.ReadKey();

```
Console.WriteLine("\nXXXXXX Slide 11 - seleção if-else");
switch (str) // str tem de ser de um tipo enumeravel ou string
{
    case "janeiro":
    case "março":
    case "maio":
    case "julho":
    case "agosto":
    case "outubro":
    case "dezembro":
        Console.WriteLine("tem 31 dias");
        break;
```

(continua ...->)

Programação Visual

Trabalho de Laboratório S02 - Tutorial

```
case "abril":
case "junho":
case "setembro":
case "novembro":
Console.WriteLine("tem {0} dias", 30);
    break;
case "fevereiro":
Console.WriteLine("tem {0} ou {1} dias", 28, 29);
    break;
default:
    Console.WriteLine("mês desconhecido");
    break;
}
Console.WriteLine("prima return para continuar");
Console.ReadKey();
```

Figura 6: código para o nível 6

Nível 7

Estruturas de Selecção – if-else

1. Na linha a seguir à instrução para ler uma tecla do teclado insira o código da Figura 7
2. Compile, elimine todos os erros sintáticos e corra o programa.
3. Na linha a seguir à instrução para ler uma tecla do teclado:
 - a) Peça ao utilizador para digitar um número de **um** a **5**;
 - b) Leia o caracter digitado para a String denominada **stringTmp_NA**.
 - c) Converta a String no double equivalente usando a instrução **double.Parse(stringTmp_NA)** e atribua esse valor a uma variável do tipo double denominada **numero_NA**.
 - d) Use uma estrutura if/else para imprimir na consola a parte inteira do valor digitado. Use condições do tipo:
(numero_NA >=0.0 && numero_NA < 1.0)
(numero_NA >=1.0 && numero_NA < 2.0)
(numero_NA >=2.0 && numero_NA < 3.0)
Etc.
 - e) Insira as duas instruções abaixo:
Console.WriteLine("\n\nprima return para continuar");
Console.ReadKey();

```
Console.WriteLine("\nXXXXX Slide 12 - selecção if-else");
if (str == "fevereiro") {
    // se tiver mais de uma instrução temos de
    //criar um bloco com chavetas

    (continua ...->)
```

Programação Visual

Trabalho de Laboratório S02 - Tutorial

```
Console.WriteLine("tem {0} dias", 28);
Console.WriteLine("Se o bloco da cláusula" +
    "\ntiver duas ou mais instruções" +
    "\n temos de as agrupar num bloco");
}
else if ((str == "abril") ||
    (str == "junho") ||
    (str == "setembro") ||
    (str == "novembro"))
    // se a intrução for simples não precisa de chavetas
    Console.WriteLine("tem {0} dias", 30);
else
    Console.WriteLine("tem {0} dias", 31);
Console.WriteLine("prima return para continuar");
Console.ReadKey();
```

Figura 7: código para o nível 7

Nível 8

Estruturas de Seleção – Quando usar qual delas?

1. Na linha a seguir à instrução para ler uma tecla do teclado insira o código da Figura 8
2. Compile, elimine todos os erros sintáticos e corra o programa.

```
Console.WriteLine("\nXXXXX slide 13 - Quando usar qual delas?");
Console.WriteLine("\nOperador condicional ?" +
    "\n    Sempre que o resultado da avaliação seja atribuível a " +
    "\n    a uma variável e facilite a leitura." +
    "\n\nSwitch" +
    "\n    Quando str for uma String ou um enumerável." +
    "\n\nif /else" +
    "\n    Quando não pudermos usar os dois anteriores");
Console.WriteLine("\nprima return para continuar");
Console.ReadKey();
```

Figura 8: código para o nível 8

Nível 9

Estruturas de Repetição - foreach

1. Na linha a seguir à instrução para ler uma tecla do teclado insira o código da Figura 9
 2. Compile, elimine todos os erros sintáticos e corra o programa.
 - a) Peça ao utilizador para digitar o seu nome completo;
 - b) Leia o nome para a variável **stringTmp_NA**.
 - c) Usando um foreach imprima na consola o nome no formato de um caracter por linha (na vertical 😊)
 - e) Insira as duas instruções abaixo:
- ```
Console.WriteLine("\n\nprima return para continuar");
Console.ReadKey();
```

(continua ...->)

# Programação Visual

## Trabalho de Laboratório S02 - Tutorial

```
Console.WriteLine("\nXXXXXX slide 14 - Estruturas de Repetição -
foreach");
Console.WriteLine("Mais eficiente, usamos sempre que possível" +
 "\nInfelizmente só podemos usá-la para iterar coleções");
string str1 = "abcdefgh";
// ler para cada char chamado s no código abaixo
// existente na coleção str1
// execute o código do bloco abaixo
foreach (char s in str1)
{
 Console.Write(" {0}-", s);
 Console.Write(" {0}-", s + 1);
}
Console.WriteLine("\n\nprima return para continuar");
Console.ReadKey();
```

Figura 9: código para o nível 9

### Nível 10

#### Estruturas de Repetição - for

1. Na linha a seguir à instrução para ler uma tecla do teclado insira o código da Figura 10
2. Compile, elimine todos os erros sintáticos e corra o programa.
3. Execute:
  - a) 5 vezes o bloco de instruções que escreveu no nível 6 alíneas 3a) a 3e).
  - b) Insira as duas instruções abaixo:  
Console.WriteLine("\n\nprima return para continuar");  
Console.ReadKey();

```
Console.WriteLine("\nXXXXXX slide 14 - Estruturas de Repetição - for");
Console.WriteLine("Usamos quando não podemos usar o foreach" +
 "\n Infelizmente só podemos usá-la quando sabemos" +
 "\n antecipadamente quantas vezes o ciclo deve ser executado");
int i = 0;
for (i = 0; i < 10; i++)
 Console.Write(" {0}-", i);
Console.WriteLine("\n\nprima return para continuar");
Console.ReadKey();
```

Figura 10: código para o nível 10

### Nível 11

#### Estruturas de Repetição – do/while

1. Na linha a seguir à instrução para ler uma tecla do teclado insira o código da Figura 11
2. Compile, elimine todos os erros sintáticos e corra o programa.
3. Execute:
  - a) O bloco de instruções que escreveu no nível 7 alíneas 3a) a 3d) pelo menos uma vez e até que seja digitado um valor inferior a 0 ou igual ou superior a 6.

(continua ...->)



# Programação Visual

## Trabalho de Laboratório S02 - Tutorial

b) Insira as duas instruções abaixo:  
`Console.WriteLine("\n\nprima return para continuar");`  
`Console.ReadKey();`

```
Console.WriteLine("\nXX slide 15 - Estruturas de Repetição - do/while");
Console.WriteLine("Usamos quando a execução do bloco" +
 "\n depende de uma condição, mas deve ser " +
 "\n sempre executado pelo menos uma vez");
do
{
 Console.Write(" {0}-", i);
 i++;
} while (i < 20);
Console.WriteLine("\n\nprima return para continuar");
Console.ReadKey();
```

Figura 11: código para o nível 11

### Nível 12

#### Estruturas de Repetição – while

1. Na linha a seguir à instrução para ler uma tecla do teclado insira o código da Figura 12
2. Compile, elimine todos os erros sintáticos e corra o programa.
3. Execute:
  - a) O bloco de instruções que escreveu no nível 7 alíneas 3a) a 3d) enquanto o valor digitado for inferior a 0 ou igual ou superior a 6.
  - b) Insira as duas instruções abaixo:  
`Console.WriteLine("\n\nprima return para continuar");`  
`Console.ReadKey();`

```
Console.WriteLine("\nXX slide 15 - Estruturas de Repetição - while");
Console.WriteLine("Usamos quando a execução do bloco" +
 "\n depende de uma condição e " +
 "\n pode nunca ser executado");
while (i < 30)
{
 Console.Write(" {0}-", i);
 i++;
}
Console.WriteLine("\n\nprima return para continuar");
Console.ReadKey();
```

Figura 12: código para o nível 12

### Nível 13

#### Arrays – Uni e Multidimensionais

1. Na linha a seguir à instrução para ler uma tecla do teclado insira o código da Figura 13
2. Compile, elimine todos os erros sintáticos e corra o programa.

(continua ...->)

# Programação Visual

## Trabalho de Laboratório S02 - Tutorial

```
Console.WriteLine("\nXXXXXX slide 16 - Arrays - "+
 "Uni e Multidimensionais");
Console.WriteLine("Unidimensionais");
double[] valores = new double[5];
valores[2] = 5.0;
Console.WriteLine("3º elemento do array = " + valores[2]);
Console.WriteLine("Multidimensionais");
int[,] valores1 = new int[10, 4];
valores1[2, 3] = 23;
Console.WriteLine("Elemento na posição [2,3] = " + valores1[2,
3]);
Console.WriteLine("\nprima return para continuar");
Console.ReadKey();
```

Figura 13: código para o nível 13

### Nível 14 Arrays – Tabelas de Tabelas

1. Na linha a seguir à instrução para ler uma tecla do teclado insira o código da Figura 14
2. Compile, elimine todos os erros sintáticos e corra o programa.

```
Console.WriteLine("\nXXXXXX slide 15 - Arrays - " +
 "Tabelas de Tabelas");
byte[][] vals2 = new byte[5][];
vals2[1] = new byte[5];
vals2[2] = new byte[10];
vals2[1][0] = 10;
vals2[1][3] = 13;
Console.WriteLine(vals2[1][0]);
Console.WriteLine(vals2[1][3]);
Console.WriteLine("\nprima return para continuar");
Console.ReadKey();
```

Figura 14: código para o nível 14

### Nível 15 Arrays – Multidimensionais X Tabelas de Tabelas

1. Na linha a seguir à instrução para ler uma tecla do teclado insira o código da Figura 15
2. Compile, elimine todos os erros sintáticos e corra o programa.

```
Console.WriteLine("\nXXX - slide 15 - Arrays - Multidimensionais"
 + "versus Tabelas de Tabelas" +
 "\n\n As tabelas multidimensionais são mais " +
 "eficientes em termos de rapidez de iteração" +
 "\n\nAs tabelas de tabelas são mais " +
 "eficientes em termos de ocupação de memória" +
 "\n\nQuando a principal preocupação é rapidez devemos" +
 "optar pelas tabelas multidimensionais" +
```

(continua ...->)

# Programação Visual

## Trabalho de Laboratório S02 - Tutorial

```
"\n\nQuando a principal preocupação é economizar memória " +
"devemos optar pelas tabelas de tabelas.");
Console.WriteLine("\nprima return para continuar");
Console.ReadKey();
```

Figura 15: código para o nível 15

### Nível 16

#### Tipos enumerados (enums ou enumerativos)

1. Com o cursor do rato sobre o nome do projeto no Solution Explorer, prima o botão direito do rato e escolhe: Add -> Class. -> Name: Season0.cs (para que a classe se chame Season0)
2. **NOTE QUE**, em c# tal como em Java, cada classe ou tipo enumerado deve estar no seu ficheiro que deve ter como nome o nome da classe.
3. A partir do Solution Explorer, abra o ficheiro Season0.cs se ainda não estiver aberto e adicione-lhe o código a negro na figura 16A.
4. **NOTE QUE** o nome do **namespace do ficheiro Season01** é igual ao namespace do ficheiro com a classe Program que contém o método Main.
5. Procedendo como em 1, crie um enumerado Season1 cujo tipo subjacente seja inteiro e comece em 3 usando o código da Figura 16B.
6. Procedendo como em 1, crie um enumerado Season2 cujo tipo subjacente seja byte e comece em 0 usando o código da Figura 16C.
7. Procedendo como em 1, 5 e 6 crie um enumerado Season3 cujo tipo subjacente seja long e comece em 5.
8. No método Main, logo abaixo do código que escreveu no nível 15 digite o código da Figura 16D.

```
namespace LABS02_NA
{
 // por omissão: tipo subjacente é int começa em 0
 enum Season0 {
 Spring, Summer, Fall, Winter
 };
}
```

Figura 16A: código para o ponto 1 do nível 16

```
namespace LABS02_NA
{
 // por omissão: tipo subjacente é int começa em 0
 enum Season1 {
 Spring= 3, Summer, Fall, Winter
 };
}
```

Figura 16B: código para o ponto 5 do nível 16

```
namespace LABS02_NA
{
 // por omissão: tipo subjacente é int começa em 0
 enum Season2 : byte {
 Spring, Summer, Fall, Autumn = Fall, Winter
 };
}
```

Figura 16C: código para o ponto 6 do nível 16

# Programação Visual

## Trabalho de Laboratório S02 - Tutorial

```

Console.WriteLine("\nXXXXXX slide 15 - enunms");
Season0 estação0 = Season0.Fall;
Console.WriteLine(estação0); // escreve 'Fall'
Console.WriteLine((int)estação0); // escreve '2'

Season1 estação1 = Season1.Fall;
Console.WriteLine(estação1); // escreve 'Fall'
Console.WriteLine((int)estação1); // escreve '5'

Season2 estação2 = Season2.Autumn;
Console.WriteLine(estação2); // escreve 'Autumn'
Console.WriteLine((int)estação2); // escreve '2'

Season3 estação3 = Season3.Fall;
Console.WriteLine(estação3); // escreve 'Fall'
Console.WriteLine((int)estação3); // escreve '5'

Console.WriteLine("\nprima return para continuar");
Console.ReadKey();

```

Figura 16D: código para o ponto 7 do nível 16

### Nível 17A

#### Métodos (procedimentos e funções) – **sem** parâmetros e sem retorno

1. Note que, como o método Main é static (de classe), só podemos invocar nele métodos de classe (static) e por esse motivo os métodos que definiremos neste e nos níveis seguintes terão de ser antecidos da palavra reservada static que indica ao compilador que se tratam de métodos de classe.
2. Para criar/definir um método que não aceita parâmetros nem retorna nada, logo depois da chave que fecha o Método Main e antes da que fecha a classe Program, digite o código da Figura 17A1 (Note que pode copiar o código que escreveu para o nível 6 alínea 3a) a 3d).
3. Para invocar (executar) o método 5 vezes digite o código da Figura 17A2 logo depois do último Console.ReadKey do método Main.

```

// void significa que o método não retorna nada,
// ou, por outras palavras, não inclui nenhuma
// instrução com return
// a lista de parâmetros vazia,
//ou, por outras palavras, as chavetas vazias
// significam que o método não aceita parâmetros
static void imprimeParteInteira()
{
 Console.Write("\nEscreva um valor de 1 a 5 >");
 String stringTmp_NA = Console.ReadLine();
 double numero_NA = double.Parse(stringTmp_NA);
 Console.WriteLine(numero_NA);
}

```

(continua ...->)

# Programação Visual

## Trabalho de Laboratório S02 - Tutorial

```
if (numero_NA >= 0.0 && numero_NA < 1.0)
 Console.WriteLine("0");
else if (numero_NA >= 1.0 && numero_NA < 2.0)
 Console.WriteLine("1");
else if (numero_NA >= 2.0 && numero_NA < 3.0)
 Console.WriteLine("2");
else if (numero_NA >= 3.0 && numero_NA < 4.0)
 Console.WriteLine("3");
else if (numero_NA >= 4.0 && numero_NA < 5.0)
 Console.WriteLine("4");
else if (numero_NA >= 5.0 && numero_NA < 6.0)
 Console.WriteLine("5");
else
 Console.WriteLine("DEU BRONCA");
}
```

Figura 17A1: código para o nível 17A1

```
Console.WriteLine("\nXXXXXX INVOCAÇÃO de MÉTODOS sem PARÂMETROS" +
 "\n... e sem retorno");
for (int j = 0; j < 5; j++)
 imprimeParteInteira();
Console.WriteLine("\n\nprima return para continuar");
Console.ReadKey();
```

Figura 17A2: código para o nível 17A2

### Nível 17B Métodos (procedimentos e funções) – **COM** parâmetros e sem retorno

1. Para criar/definir um método que aceita parâmetros e não retorna nada, logo depois da chave que fecha o Método da figura 17A1 e antes da que fecha a classe Program, digite o código da Figura 17B1.
2. Para invocar (executar) o método enquanto fo digitado um valor entre 0 e 6 digite o código da Figura 17B2 logo depois do último Console.ReadKey do método Main.

```
//A lista de parâmetros com um único identificador,
//indica que o método deve ser invocado com um único
//valor ou variável do mesmo tipo do parâmetro
static void imprimeParteInteiraDoParametro(String parametro)
{
 double numero_NA = double.Parse(parametro);
 Console.WriteLine(numero_NA);
 if (numero_NA >= 0.0 && numero_NA < 1.0)
 Console.WriteLine("0");
 else if (numero_NA >= 1.0 && numero_NA < 2.0)
 Console.WriteLine("1");
 else if (numero_NA >= 2.0 && numero_NA < 3.0)
 Console.WriteLine("2");
}
```

(continua ...->)

# Programação Visual

## Trabalho de Laboratório S02 - Tutorial

```
else if (numero_NA >= 3.0 && numero_NA < 4.0)
 Console.WriteLine("3");
else if (numero_NA >= 4.0 && numero_NA < 5.0)
 Console.WriteLine("4");
else if (numero_NA >= 5.0 && numero_NA < 6.0)
 Console.WriteLine("5");
else
 Console.WriteLine("DEU BRONCA");
}
```

Figura B1: código para o nível 17B1

```
Console.WriteLine("\nXXXXXX INVOCAÇÃO de MÉTODOS COM PARÂMETROS" +
 "\n ... e sem retorno");
do {
 Console.Write("\nEscreva um valor de 1 a 5 >");
 stringTmp_NA = Console.ReadLine();
 imprimeParteInteiraDoParametro(stringTmp_NA);
} while (numero_NA > 0 && numero_NA < 6.0);
Console.WriteLine("\n\nprima return para continuar");
Console.ReadKey();
```

Figura 17B2: código para o nível 17B2

### Nível 17C

Métodos (procedimentos e funções) – **COM** parâmetros e **COM** retorno

1. Para criar/definir um método que aceita parâmetros e retorna um inteiro, logo depois da chave que fecha o Método da figura 17B1 e antes da que fecha a classe Program, digite o código da Figura 17C1.
2. Para invocar (executar) o método enquanto for digitado um valor entre 0 e 6 digite o código da Figura 17C2 logo depois do último Console.ReadKey do método Main.

```
//a declaração do tipo int que antecede
//o identificador do método indica que o método
//deve retornar um valor do tipo int
static int retornParteInteiraDoParametro(String parametro)
{
 double numero_NA = double.Parse(parametro);
 if (numero_NA >= 0.0 && numero_NA < 1.0)
 return 0;
 if (numero_NA >= 1.0 && numero_NA < 2.0)
 return 1;
 if (numero_NA >= 2.0 && numero_NA < 3.0)
 return 2;
 if (numero_NA >= 3.0 && numero_NA < 4.0)
 return 3;
}
```

(continua ...->)

# Programação Visual

## Trabalho de Laboratório S02 - Tutorial

---

```
if (numero_NA >= 4.0 && numero_NA < 5.0)
 return 4;
if (numero_NA >= 5.0 && numero_NA < 6.0)
 return 5;
// -1 funciona como um retorno de um código de erro
//depois de daro excepções deixaremos de usar
// códigos de erro
return -1;
}
```

Figura 17C1: código para o nível 17C1

```
Console.WriteLine("\nXXXXXX INVOCAÇÃO de MÉTODOS COM PARÂMETROS" +
 "\n ... e COM retorno");
do
{
 Console.Write("\nEscreva um valor de 1 a 5 >");
 stringTmp_NA = Console.ReadLine();
 Console.WriteLine(retornaParteInteiraDoParametro(stringTmp_NA));
} while (numero_NA > 0 && numero_NA < 6.0);
Console.WriteLine("\n\nprima return para continuar");
Console.ReadKey();
```

Figura 17C2: código para o nível 17C2

---

**The End**

•  
•  
•

**My Friend  
The End**