

LP2 – Aula 07

TEÓRICA

Profª Mª Denilce Veloso

- denilce.veloso@fatec.sp.gov.br
 - denilce@gmail.com

C# .NET - Declaração e uso de Vetores (Arrays)

Em C#, arrays são objetos cuja classe base é System.Array. Os arrays podem ser unidimensionais, multidimensionais ou ainda arrays de arrays, cujos elementos são outros arrays.

➤ Vetores (uma dimensão)

```
int[] numeros = new int[2];  
numeros[0] = 1;  
numeros[1] = 2;  
// numeros[2] = 3; não existe
```

➤ Formas de declaração

```
int[] numeros2 = new int[3] { 10, 20, 30 }; // OBJETO
```

```
int[] numeros3 = new int[] { 10, 20, 30 }; // OBJETO
```

```
int[] numeros4 = { 10, 20, 30 }; // FORMA LITERAL
```

```
string[] semana = { "segunda", "terça", "quarta", "quinta", "sexta", "sábado",  
"domingo" }; // FORMA LITERAL
```

C# .NET - Declaração e uso de Vetores (Arrays)

- Para obtermos elemento segunda fazemos:

```
string dia = semana[0];
```

- Quando um array é criado, todas as posições são inicializadas com valor padrão. Se for número serão inicializadas com 0, se for string null, se for booleano false e se for DateTime 1/1/1 00:00:00.

- Para varrer o array

```
for (int i = 0; i < arrInt.Length; i++)  
    MessageBox.Show(arrInt[i]);
```

```
foreach(int numero in arrInt)  
    MessageBox.Show(numero.ToString());
```

C# .NET - Declaração e uso de Vetores (Arrays)

- Matriz de duas dimensões

`int[,] arrInt = new int[2,2] ;` Inicializando o array:

`int[,] arrInt = new int[2,2] {{0,0},{0,1}}`

`int[,] arrInt = { { 0, 0 }, { 0, 1 } };`

`int[,] arrInt = new int[,]{{0,0}, {0, 1}};`

- Para obtermos o último elemento:

`int numero = arrInt[1,1];` → irá voltar 1

`double[,] retangulo = new double[5,10];` → aloca espaço para elementos de 0 a 4 e para elementos de 0 a 9

→ Acima quando declaramos um array de duas dimensões [5,10], ele possuirá 5 linhas (0 a 4) e 10 colunas (0 a 9) perfazendo um total de 50 elementos.

C# .NET - Declaração e uso de Vetores (Arrays)

Além disto podemos atribuir diretamente a cada elemento o seu valor:

```
DateTime[] DataAdmissao = new DateTime[3];  
DataAdmissao[0] = Convert.ToDateTime("28/08/1982");  
DataAdmissao[1] = Convert.ToDateTime("03/03/1986");  
DataAdmissao[2] = Convert.ToDateTime("31/03/2018");
```

C# .NET - Declaração e uso de Vetores (Arrays)

➤ Propriedades/Métodos dos Arrays

Rank: Propriedade que retorna o número de dimensões de um array.

Length: Propriedade que retorna o número total de elementos de todas as dimensões de um array.

GetLength: Método retorna o número total de elementos de uma dimensão (x,y,z,...) especificada do array. Recebe como parâmetro um número inteiro que corresponde ao número da dimensão da qual se deseja saber o total de elementos. A numeração das dimensões começa por zero.

C# .NET - Declaração e uso de Vetores (Arrays)

➤ Exemplos:

```
int[] numeros = new int[5] {1,2,3,4,5};
```

```
MessageBox.Show("Rank=" + numeros.Rank); //1
```

```
MessageBox.Show("Length=" + numeros.Length); //5
```

```
MessageBox.Show("GetLength=" + numeros.GetLength(0)); //5
```

```
int[,] numeros2 = new int[2,3] {{1, 2,0}, {3, 4,0}};
```

```
MessageBox.Show("Rank=" + numeros2.Rank); //2
```

```
MessageBox.Show("Length=" + numeros2.Length); //6
```

//→ dimensão 1 → número de colunas

```
MessageBox.Show("GetLength=" + numeros2.GetLength(1)); //3
```

// → dimensão 0 → número de linhas

```
MessageBox.Show("GetLength=" + numeros2.GetLength(0)); //2
```

C# .NET - Declaração e uso de Vetores (Arrays)

- Para varrer (ou percorrer) o array multidimensional e colocar dados

```
int[,] numeros = new int[10, 20];  
for(int i = 0; i < numeros.GetLength(0); i++) → vai até 9  
    for(int j = 0; j < numeros.GetLength(1); j++) → vai até 19  
        numeros[i, j] = i * j;
```


C# .NET - Declaração e uso de Vetores (Arrays)

➤ Exclusão de item

Não é possível excluir uma posição ou elemento de um array. Se o array tem 10 posições, ele sempre terá 10 posições.

Utilize o método `Clear()` da classe `Array` para redefinir essa posição do array para o valor default desse tipo. Por exemplo:

```
int[] seuArray = new int[3];  
seuArray[0] = 1;  
seuArray[1] = 2;  
seuArray[2] = 3;
```

```
Array.Clear(seuArray, 0, 1);
```

```
foreach (var x in arr)  
    MessageBox.Show(x); // imprime um de cada vez
```

Resultado: 0

2

3

C# .NET - Declaração e uso de Vetores (Arrays)

DV0

➤ DICAS

Convertendo array de int para String

```
int[] vetorInt = new int[3] { 1, 2, 3 };  
string[] vetorSt = Array.ConvertAll(vetorInt, x =>  
x.ToString());  
string resultado = String.Join("", vetorSt);  
MessageBox.Show(resultado);
```

➤ ‘

Convertendo array de Char para string

```
char[] c = new char[3] {'a', 'b', 'c'};  
string s = new string(c);
```

Slide 10

DV0

```
string stringona=new string("a",20);
```

DENILCE DE ALMEIDA OLIVEIRA VEL; 2023-04-18T23:55:52.071

C# .NET - Declaração e uso de Vetores (Arrays)

➤ Procurando um item

```
string[] Cores = { "vermelho", "verde", "amarelo", "laranja", "azul" };
```

```
int posicao = Array.IndexOf(Cores, "verde");
```

→ Posição é a posição de onde foi encontrado "verde" → 1

C# .NET - Declaração e uso de Vetores (Arrays)

- Reverse: É um método estático cujo objetivo é inverter a ordem dos elementos do array. Essa inversão pode ser completa ou parcial, indicando o índice inicial e final para a inversão do array.

Exemplos:

```
int[] numeros = new int[5] {1,2,3,4,5};
```

```
Array.Reverse(numeros);
```

```
MessageBox.Show(numeros[0] + " " + numeros[4]); // 5 e 1
```

```
int[] numerosN = new int[5] {10,20,30,40,50};
```

```
Array.Reverse(numerosN,2,3); // a partir do elemento 2, 3 elementos
```

```
MessageBox.Show(numerosN[0] + " " + numerosN[1] + " " + numerosN[2] + " " +  
numerosN[3] + " " + numerosN[4]); // 10 20 50 40 30
```

C# .NET - Declaração e uso de Vetores (Arrays)

- Sort: Método que ordena o array passado como parâmetro.

Exemplo:

```
int[] numeros3 = new int[3] { 4, 3, 2 };
```

```
Array.Sort(numeros3);
```

```
MessageBox.Show(numeros3[0] + " " + numeros3[2]); // 2 e 4
```

```
int[] numeros = new int[5] { 0, 20, 3, 10, 4 };
```

```
Array.Sort(numeros); // 0,3,4,10,20
```

```
string[] numeros2 = new string[5] {"0", "20", "3", "10", "4"};
```

```
Array.Sort(numeros2); // 0,10,20,3,4
```

```
string[] str = new string[6] { "Ana", "denilce", "elsa", "Denilce", "carlos", "Fabio" }; //
```

minúsculas foram colocadas de propósito

```
Array.Sort(str); // Ana,carlos,denilce,Denilce,elsa,Fabio
```

*** qualquer forma de mudar a lógica do sort deve usar a interface IComparer*

C# .NET - Declaração e uso de Vetores (Arrays)

➤ Coleções (Matrizes Dinâmicas)

A coleção `List(T)` representa uma coleção de *objetos tipados* a qual é altamente otimizada oferecendo um bom desempenho e pode ser acessada usando índice.

Esta classe fornece métodos para percorrer, filtrar, ordenar, procurar, contar e manipular coleções. Para verificar todos os métodos e propriedades basta digitar variável do tipo `List.(ponto)` exemplo `lista`. (A *versão não genérica* seria a classe `ArrayList`).

C# .NET - Declaração e uso de Vetores (Arrays)

Exemplos:

```
int[] lista = new int[10];
```

```
lista[0] = 0;
```

```
lista[1] = 2;
```

```
...
```

```
// se fosse declarar já incluindo
```

```
//List<int> lista = new List<int>(new int[] {0,2,4,6,8,10,12,14,16,18});
```

```
List<int> lista = new List<int>(); // pode escolher o tipo desejado
```

```
int x = 0;
```

```
for (x=0; x<10; x++)
```

```
{
```

```
    lista.Add(x*2); // adiciona os itens
```

```
}
```

```
foreach (int elemento in lista)
```

```
{
```

```
    MessageBox.Show("antes:" + elemento.ToString());
```

```
}
```

→ irá mostrar 0 2 4 6 8 10 12 14 16 18

C# .NET - Declaração e uso de Vetores (Arrays)

```
lista.Remove(10); // remove o item cujo conteúdo é 10
//ou
//lista.RemoveAt(5); // remove o item na posicao 5

foreach (int elemento in lista)
{
    MessageBox.Show("depois:" + elemento.ToString());
}
```

→ irá mostrar 0 2 4 6 8 12 14 16 18

C# .NET - Declaração e uso de Vetores (Arrays)

```
Lista.Insert(2,444); // inserindo 444 na posição 2
```

```
foreach (int elemento in lista)
{
    MessageBox.Show("depois insert:" + elemento.ToString());
}
```

→ irá mostrar 0 2 444 4 6 8 12 14 16 18

C# .NET - Declaração e uso de Vetores (Arrays)

Exemplo utilizando ArrayList:

```
ArrayList lista = new ArrayList(); // tipo string
lista.Add("um");
lista.Add("dois");
lista.Add("tres");

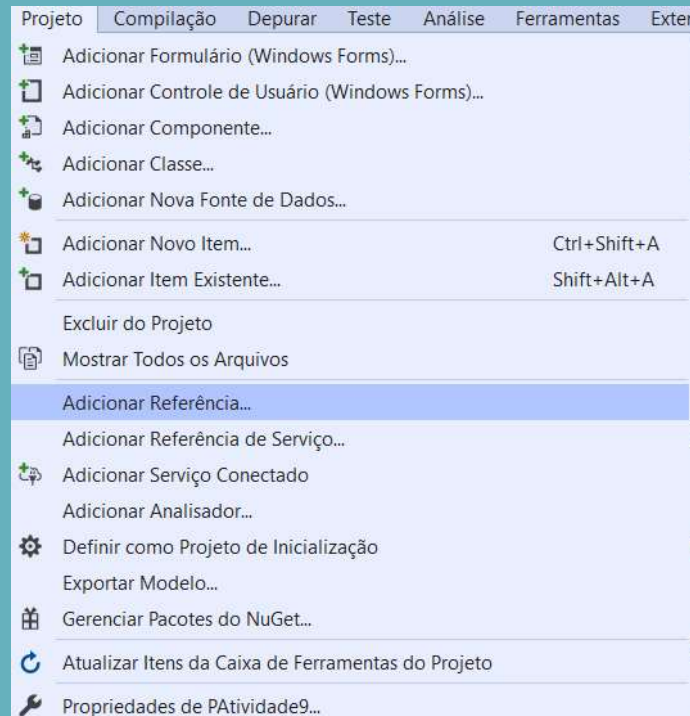
foreach (string item in lista)
{
    MessageBox.Show(item);
}
```

C# .NET - Declaração e uso de Vetores (Arrays)

Recebendo dados para colocar nos vetores ou matrizes.

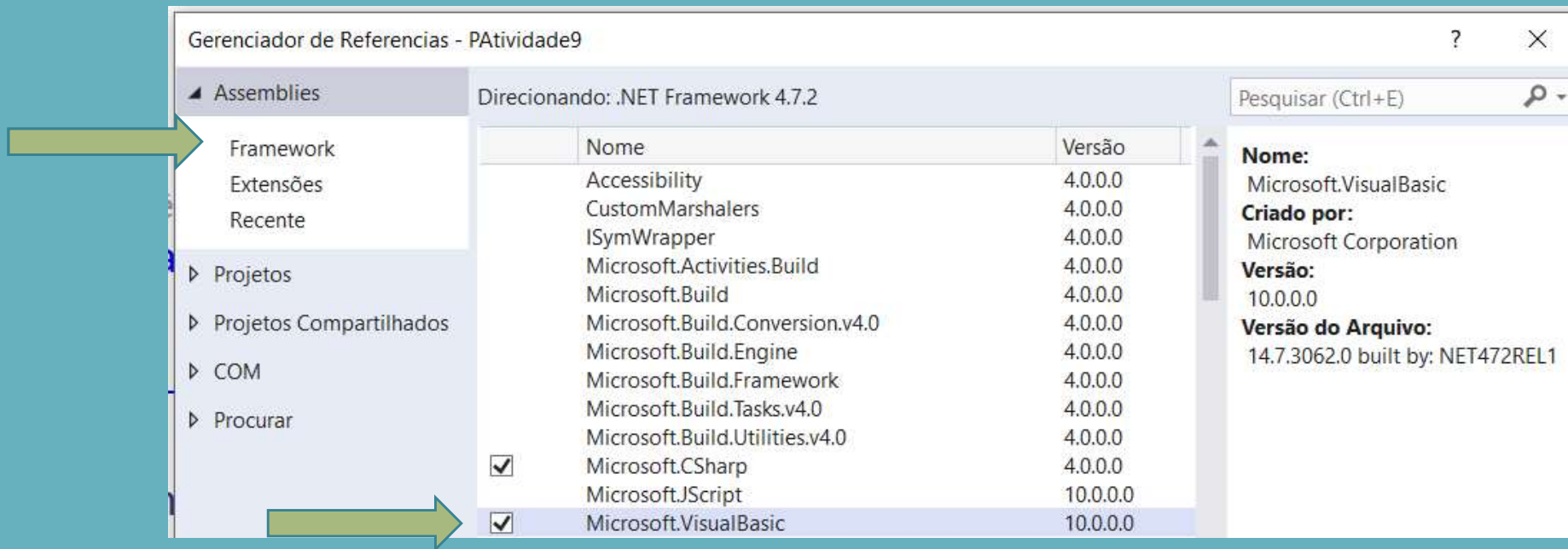
A linguagem C# não possui nada parecido com o InputBox da linguagem Visual Basic para a entrada de dados (uma espécie de prompt). Mas é possível fazer uma referência no projeto a Microsoft.VisualBasic e utilizar esse recurso.

a) No menu Project selecione Add Reference. (ou Projeto/Adicionar Referência)



C# .NET - Declaração e uso de Vetores (Arrays)

b) No item Assemblies\Framework selecione a caixa Microsoft.VisualBasic e clique em OK.



C# .NET - Declaração e uso de Vetores (Arrays)

c) Também é necessário incluir Microsoft.VisualBasic no using.

```
1  using System;
2      using System.Collections.Generic;
3      using System.ComponentModel;
4      using System.Data;
5      using System.Drawing;
6      using System.Linq;
7      using System.Text;
8      using System.Threading.Tasks;
9      using System.Windows.Forms;
10     using Microsoft.VisualBasic;
```

C# .NET - Declaração e uso de Vetores (Arrays)

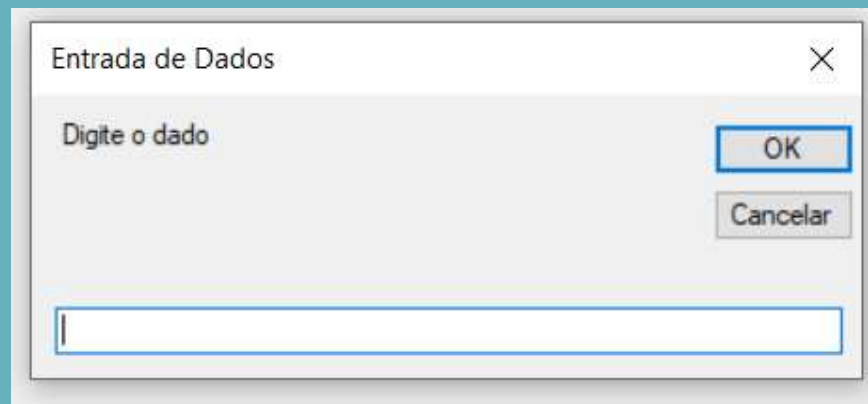
d) No código a sintaxe é a seguinte:

```
valorRetorno = InputBox(Prompt, Titulo, respostaPadrao, XPos, YPos)
```

→ Ele sempre retorna uma string e os parâmetros respostaPadrao, XPos, YPos são opcionais.

Exemplo:

```
string auxiliar = "";  
    auxiliar = Interaction.InputBox("Digite o dado", "Entrada de Dados");
```



C# .NET - Declaração e uso de Vetores (Arrays)

Exercícios

1) Qual das linhas a seguir instancia um array de 10 elementos?

a) `int[] numeros = new int[9];`

b) `int[] numeros = new int[10];`

c) `int[] numeros = new int["dez"];`

d) `int[10] numeros = new int[10];`

C# .NET - Declaração e uso de Vetores (Arrays)

2) Supondo o array abaixo:

```
int[] numeros = new int[15];
```

Como acessar o quinto elemento nessa lista?

- a) numeros[3];
- b) numeros[4];
- c) numeros["quinto"];
- d) numeros[5];

C# .NET - Declaração e uso de Vetores (Arrays)

3) Dado um array número, como descobrir seu tamanho?

- a) tamanho=numero.Length;
- b) tamanho=numero.Size;
- c) tamanho=numero.Size();
- d) tamanho=numero.Length();
- e) tamanho= numero.Capacity();

C# .NET - Declaração e uso de Vetores (Arrays)

4) Supondo o código abaixo qual o resultado de Soma?

```
private void button7_Click(object sender, EventArgs e)
{
    int[] arrayNaturais = new int[10];
    int soma = 0;
    for (int i = 0; i < arrayNaturais.Length; i++)
    {
        arrayNaturais[i] = i;
    }
    for (int i = 0; i < arrayNaturais.Length; i++)
    {
        if ((arrayNaturais[i] % 2) != 0)//não é par
        {
            soma += arrayNaturais[i];
        }
    }
    MessageBox.Show("Soma="+soma.ToString());
}
```

C# .NET - Declaração e uso de Vetores (Arrays)

5) Supondo o código abaixo, informe qual será o resultado da variável stringona.

```
private void button1_Click(object sender, EventArgs e)
{
    double[] vetor = new double[10];
    int contador = 0;
    for (contador = 0; contador < vetor.Length; contador++)
    {
        vetor[contador] = contador * 2;
    }
    Array.Reverse(vetor); ←
    string stringona = "";
    foreach (double x in vetor)
        stringona += x + "\n";
    MessageBox.Show(stringona);
}
```

C# .NET - Declaração e uso de Vetores (Arrays)

6) Supondo o código abaixo, informe qual será o resultado da variável text?

```
private void button12_Click(object sender, EventArgs e)
{
    var text = "xxxxxxxxxxxxxxxxxxxxxx João yyyyyyyyyy João @ zzzzzzz #João ";
    var myName = "João";
    List<string> hits = new List<string>();
    int i;
    int j;
    for (i = 0; i < text.Length; i++)
    {
        if (text.Substring(i, 1) == "J") // if text[i]=='J'
        {
            for (j = i; j < i + myName.Length; j++)
                hits.Add(text.Substring(j, 1));
        }
    }
```

C# .NET - Declaração e uso de Vetores (Arrays)

```
if (hits.Count == 0)
    MessageBox.Show("João não foi encontrado!");
else
{
    text = "";
    foreach (string p in hits)
        text += p + " ";
    MessageBox.Show(text);
}
}
```

C# .NET - Declaração e uso de Vetores (Arrays)

7) Supondo o código abaixo que calcula a média de **um** aluno em **5** disciplinas, o que irá acontecer se o usuário digitar uma letra? E se o usuário digitar uma média -6?

```
{  
    double[] vetor = new double[5];  
  
    int x = 0;  
  
    double media = 0;  
    string valor = "";  
  
    for (x = 0; x < 5; x++)  
    {  
        valor = Interaction.InputBox($"Entre com a nota da disciplina {i + 1}", "Entrada de  
Dados");  
        // ou valor = Interaction.InputBox("Entre com a nota da disciplina:" + (x + 1).ToString(), "Entrada de dados");  
        if (valor=="")  
            break;
```

C# .NET - Declaração e uso de Vetores (Arrays)

```
if (double.TryParse(valor, out vetor[x]))
{
    if (vetor[x] >= 0 && vetor[x] <= 10)
        media += vetor[x];
    else
    {
        MessageBox.Show("Média deve estar em 0 e 10!");
        x--;
    }
}
else
{
    MessageBox.Show("Número Inválido!");
    x--;
}
}
```

```
MessageBox.Show("A média entre as cinco disciplinas é: " + (media / 5).ToString("N2"));
```

```
}
```