

VETORES

TP / LP

Professor Antonio Marcos Alvarez

Conceitos

Um vetor corresponde a uma coleção de dados de tamanho fixo, indexada, unidimensional e homogênea

- Indexada: os elementos são acessados por meio de índices
- Unidimensional: uma dimensão
- Homogênea: todos dados são do mesmo tipo

0	Maria
1	João
2	Carlos
3	Ana
4	Joaquim

A

Vetores

Vetores são também chamados de **arranjos unidimensionais**

Em C# a primeira posição de um vetor é a posição 0

Um arranjo deve ser alocado previamente, antes de ser utilizado. Uma vez alocado, sua quantidade de elementos é fixa

0	Maria
1	João
2	Carlos
3	Ana
4	Joaquim

A

Exemplificando...

Como criar um vetor?

declaração →
int[] A;
double[] B;
string[] C;

instanciação →
A = new int[10];
B = new double[5];
C = new string[8];

Memoria (RAM)

A	
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

B	
0	
1	
2	
3	
4	

C	
0	
1	
2	
3	
4	
5	
6	
7	

Como atribuir ou acessar “elementos” de um vetor

A[3] = 50;

C[7] = “Marcos”;

//Ou gerar um loop para inserir valores

```
for (int i=0; i<5; i++) {  
    B[i] = 100 + i;  
}
```

//Usando “i” como índice do vetor

Memoria (RAM)

A	
0	
1	
2	
3	50
4	
5	
6	
7	
8	
9	

B	
0	100
1	101
2	102
3	103
4	104

C	
0	
1	
2	
3	
4	
5	
6	
7	Marcos

Vamos analisar e Desenvolver um programa: EX_Vetor01

Fazer um programa para ler um número inteiro positivo N, depois ler N números quaisquer e armazená-los em um vetor. Em seguida, mostrar na tela todos elementos do vetor.

Entrada	Saída
4	10.5
10.5	4.2
4.2	-7.1
-7.1	15.0
15.0 ↴	

Obs.: Desafio, gerar lista dos números na ordem de entrada e outra na ordem inversa.

Codificação – Parte (01) EX_Vetor01

```
int N;  
double[] vetorN;  
int i;  
  
Console.Write("Informe a quantidade de números a ser armazenado no  
vetor:");  
  
N = int.Parse(Console.ReadLine());  
  
vetorN = new double[N];  
  
Console.WriteLine("Entrada de Dados de " + N + " Números.");  
  
for (i=0; i<N; i++) {  
    Console.Write("Digite " + (i+1) + "º Número:");  
    vetorN[i]=double.Parse(Console.ReadLine());  
}
```

Parte (02) EX_Vetor01

```
Console.WriteLine("Lista dos Numeros na Ordem de Entrada do Índice:");
for (i=0; i<N; i++) {
    Console.WriteLine("O índice [" + i + "] do vetorN Armazena: " + vetorN[i]);
}

Console.WriteLine("Lista dos Números na Ordem Inversa de Entrada do Índice:");
for (i = N-1; i >=0; i--) {
    Console.WriteLine("O índice [" + i + "] do vetorN Armazena: " + vetorN[i]);
}

Console.WriteLine("Final do Processo, pressione [Enter] para encerrar Console ...");
Console.ReadLine();
```

Desafio

Desenvolver um programa que permita a entrada de N informações (nome, sexo, idade e altura), ao terminar a entrada de dados, o programa informará:

- Total de pessoas do Sexo “F”;
- Total de Menores de 18 anos.

Nº de Informações	Nome	Sexo	Idade	Altura
7	João	M	52	1,70
	Maria	F	20	1,60
	Mayara	F	16	1,55
	Marina	F	9	1,40
	Pedro	M	15	1,80
	Fernando	M	18	1,90
	Estela	F	19	1,60

Informações	
Total do Sexo F:	4
Total de Menores (18 anos):	3

Codificação (Parte 1) - EX_Vetor02

```
//Declarar as Variáveis e Vetores
```

```
string[] nome;
```

```
char[] sexo;
```

```
int[] idade;
```

```
int N=0, i=0, s=0, m=0; // N (nº Entradas), i (indice), s(total de sexo), m (total de menor)
```

```
//Solicitação do Nº de Entradas de Dados (nome, sexo e idade)
```

```
Console.Write("Informe o Nº de Entradas:");
```

```
N = int.Parse(Console.ReadLine());
```

```
//Instanciar Vetores
```

```
nome = new string[N];
```

```
sexo = new char[N];
```

```
idade = new int[N];
```

Codificação (Parte 2) EX_Vetor02

```
//Entradas
for (i = 0; i < N; i++)
{
    Console.WriteLine("Informações do Indice :" + i);
    Console.Write("Nome.....:");
    nome[i] = Console.ReadLine();
    Console.Write("Sexo.....:");
    sexo[i] = char.Parse(Console.ReadLine());
    Console.Write("Idade....:");
    idade[i] = int.Parse(Console.ReadLine());
    Console.WriteLine("-----");
    if (sexo[i] == 'F' || sexo[i] == 'f') { s = s + 1; }
    if (idade[i] < 18) { m = m + 1; }
}
```

Codificação (Parte 3) EX_Vetor02

```
//Exibir Informações  
Console.WriteLine("Número do Sexo Feminino: " + s);  
Console.WriteLine("Menores que 18 anos.....: " + m);  
  
Console.WriteLine("Tecle [Enter] para Finalizar o Console...");  
Console.ReadLine();
```

Usando “foreach” (Para Cada)

A instrução “foreach” executa uma instrução ou um bloco de instruções para cada elemento em uma instância (Exemplo um vetor) que pode ser usado como um “alias” (Apelido).

Problema exemplo

Fazer um programa para ler um valor N e depois o nome de N pessoas. Em seguida, mostrar os nomes lidos.

Entrada	Saída
3 Maria Joaquim Ana	Nomes lidos: Maria Joaquim Ana

Código Simples com “for”- Parte 1 (EX_foreach)

```
//Declarar as Variáveis e Vetores
string[] nome;
int N = 0, i = 0; // N (nº Entradas), i (Indice)
//Solicitação do Nº de Entradas de Dados (nome)
Console.Write("Informe o Nº de Entradas:");
N = int.Parse(Console.ReadLine());
//Instanciar Vetor
nome = new string[N];
for (i = 0; i < N; i++)
{
    Console.WriteLine("Informações do Indice :" + i);
    Console.Write("Nome....:");
    nome[i] = Console.ReadLine();
}
```

Parte 2

```
//Exibir os Nomes do Vetor  
for (i = 0; i < N; i++)  
{  
    Console.Write("Informações do Indice :" + i+ " | ");  
    Console.WriteLine("Nome....:" + nome[i]);  
}  
  
Console.ReadLine();
```

Substituir o for por “foreach” – parte 03

```
//Exibir os Nomes do Vetor
    for (i = 0; i < N; i++)
    {
        Console.Write("Informações do Indice :" + i+ " | ");
        Console.WriteLine("Nome....:" + nome[i]);
    }
//Por
    //A variável x (elemento) abstrém o conteúdo do vetor nome (coleção) - in (dentro de)
    //Para Cada elemento "x" (resgata os dados) dentro do vetor "nome"
    foreach (string x in nome)
    {
        Console.WriteLine("Nome....:" + x);
    }
```

Matriz

As matrizes podem ter mais de uma dimensão, bidimensional e tridimensional.

Exemplo: Bidimensional

Uma matriz corresponde a uma coleção de dados de tamanho fixo, indexada, bidimensional e homogênea

- Indexada: os elementos são acessados por meio de índices
- bidimensional: duas dimensões
- Homogênea: todos dados são do mesmo tipo

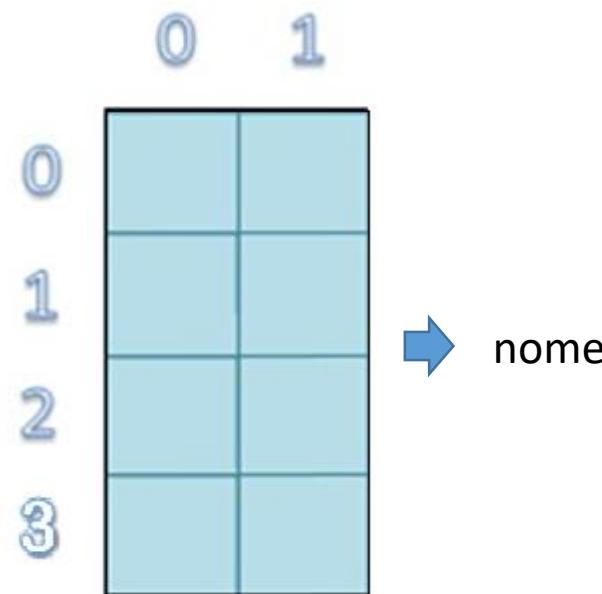
	0	1	2
0			
1			
2			
3			
4			

A

Vamos fazer um programa exemplos:

Desenvolver um programa que construa uma matriz bidimensional para serem armazenados nomes, solicitando o numero de linhas e colunas que instanciará a matriz (nome).

Nº Linha	Nº Coluna
4	2



Código Parte 01 - EX_Matriz2D

```
Console.WriteLine("Criando Matriz 2D (Bidimensional);  
//Declarando Variaveis;  
int nL, nC, cL, cC; // NL (nº Linhas), nC (nº Colunas), cL (contador Linha), cC (Contador Coluna)  
int cD = 0; // cD (contador de Dados de Entrada (nome)  
  
//Declarando Matriz nome  
string[,] nome;  
  
Console.Write("Nº Linhas: ");  
nL = int.Parse(Console.ReadLine());  
Console.Write("Nº Colunas: ");  
nC = int.Parse(Console.ReadLine());  
  
//Instanciando Matriz  
nome = new string[nL,nC];
```

Código Parte 01 - EX_Matriz2D

```
//Instanciando Matriz
nome = new string[nL,nC];

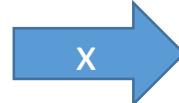
//Entrada dos Dados na Martriz
for (cL=0; cL < nL; cL++)
{
    for (cC=0; cC < nC; cC++)
    {
        cD += 1; // cD = cD + 1
        Console.Write("Informe " + cD + "º Dado: ");
        nome[cL, cC] = Console.ReadLine();
    }
}

Console.WriteLine("Pressione [ENTER] para encerrar Console....");
Console.ReadLine();
```

Vamos Analisar Exemplos (2D e 3D):

```
//Declarando, instanciando e inserindo informações bidimensional (4,2)
```

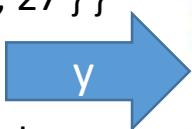
```
int[,] x2D = new int[,] {  
    { 1, 2 }, { 3, 4 },  
    { 5, 6 }, { 7, 8 }  
};
```



0	1
3	4
5	6
7	8

```
//Declarando, instanciando e inserindo informações tridimensional (3,3,3)
```

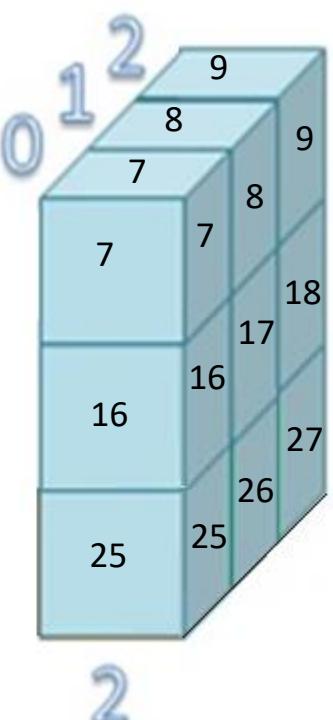
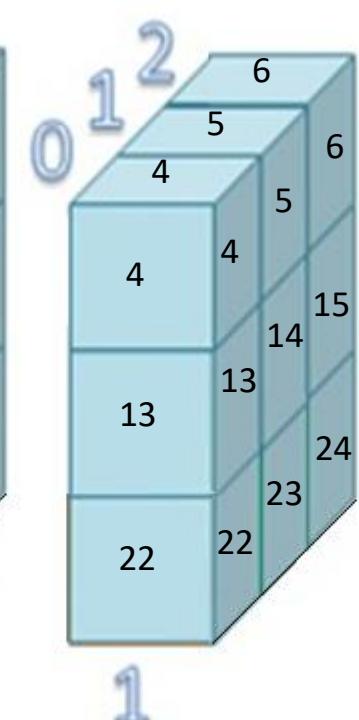
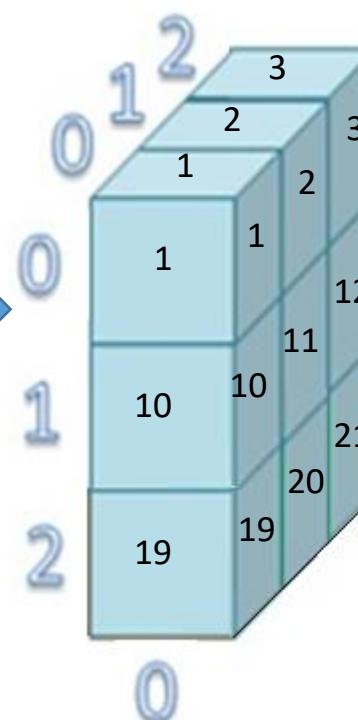
```
int[, ,] y3D = new int[, ,] {  
    { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 } },  
    { { 10, 11, 12 }, { 13, 14, 15 }, { 16, 17, 18 } },  
    { { 19, 20, 21 }, { 22, 23, 24 }, { 25, 26, 27 } }  
};
```



```
//Exibindo parâmetros das Matrizes Declaradas e Instaciadas
```

```
Console.WriteLine("Elementos em x2D: " + x2D.Length);
```

```
Console.WriteLine("Dimensão de x2D: " + x2D.Rank);
```



```
Console.WriteLine("Elementos em y3D: " + y3D.Length);
```

```
Console.WriteLine("Dimensão de y3D: " + y3D.Rank);
```

Ex_Matrices (Parte 01)

```
//Exibindo Valores armazenados nas matrizes 2D e 3D
```

```
Console.WriteLine(x2D[0,0]); //1
```

```
Console.WriteLine(x2D[2,1]); //6
```

```
Console.WriteLine(y3D[0,0,0]); //1
```

```
Console.WriteLine(y3D[1,1,1]); //14
```

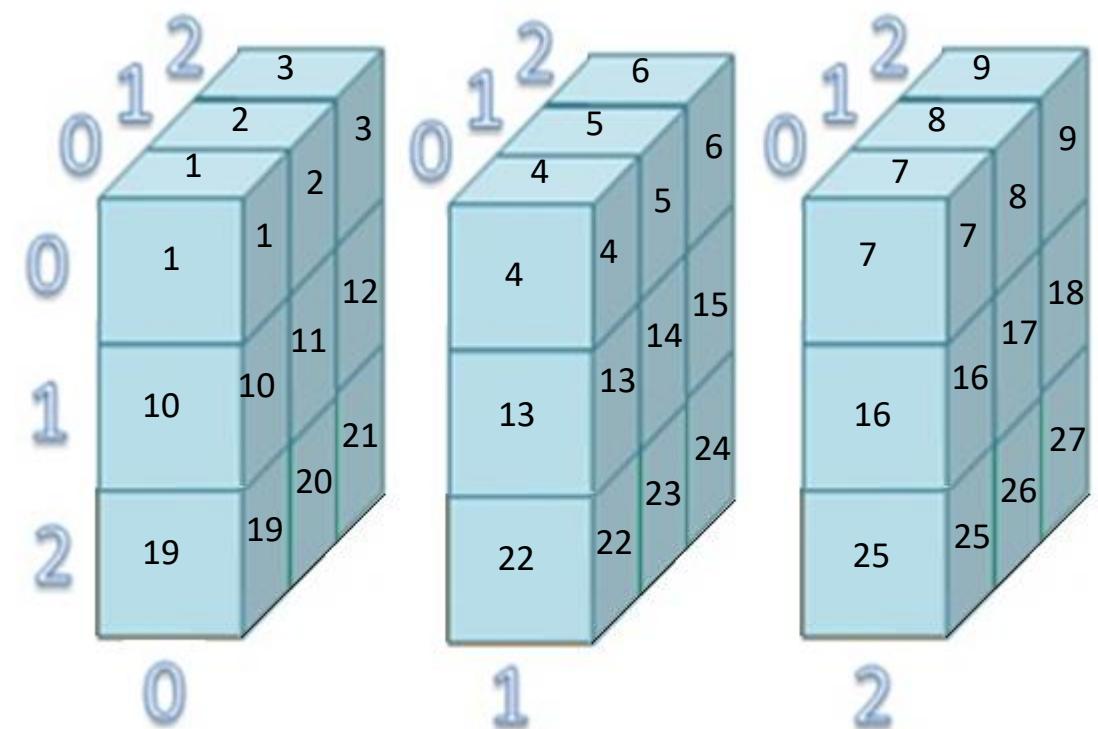
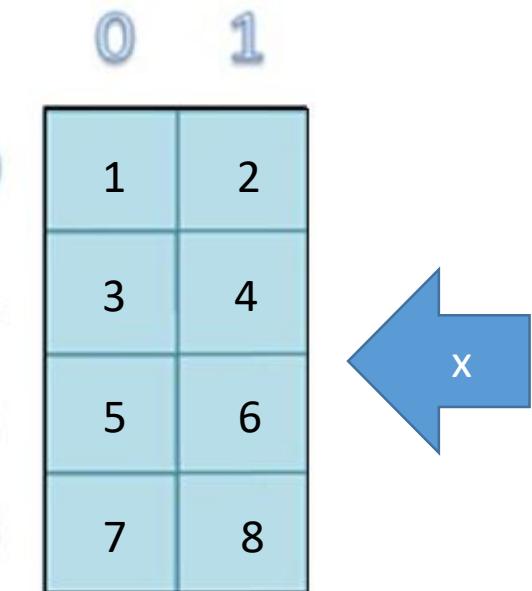
```
Console.WriteLine(y3D[2,0,0]); //19
```

```
Console.WriteLine(y3D[2,1,2]); //24
```

```
Console.WriteLine(y3D[0,2,0]); //7
```

```
//Esperando [Enter] para Encerrar o Console...
```

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



Atividades a serem apresentadas pelos Alunos

No Caderno

TP:

- Resumo dos Conceitos Iniciais Apresentados;
- Analise lógica dos Exercícios propostos e seus Fluxos de Processo.

LP:

Resumo da Apresentação do C#;

Resumos dos Comandos e Instruções apresentados;

Os códigos dos Programas, conforme as temas apresentados.

BD:

- Resumo dos Conceitos iniciais Apresentados;
- Analise dos Comandos Básicos do SQL;
- Analise dos Dois Bancos Relacionais Disponibilizados (Agenda e Cadastros).

Exercícios Práticos (Visual Studio 2019) C#

Personalizados com seus Dados e comentários sobre as funções principais do programa.

- Equação de 2º Grau (Bhaskara);
- Emissão de Tabuada (for);
- IMC (Calcula Índice de Massa Corpórea);
- Plano100 (Calcula o Consumo em minutos, emite total à pagar);
- Split (Separa informações digitadas em uma única linha (ReadLine));
- Celsius e Fharenheit (while);
- Calculo Valor Venal de um Terreno;
- Entradas de dados (Nome, Sexo e Idade) Total Feminino, Total de Menos (Vetor - 02);
- Emissão de Listas dos Vetores por índice (Crescente e Decrescente).

Forma de entrega dos trabalhos e Avaliações

- Neste momento estamos analisando com a Coordenação;
- Avaliações serão provavelmente digitais/eletrônicas, no horário estipulado com antecedência e forma assistida...
- Ainda em estudo...