

Matrizes

Matrizes/Vetores Bidimensionais

- Um vetor possui índices para uma determinada posição:

- `int[] vet = new int[10];`

0	1	2	3	4	5	6	7	8	9
5	6	-7	2	0	-98	56	23	1	1

Matrizes/Vetores Bidimensionais

- Uma matriz possui índices para linhas e colunas:

- `int[,] mat = new int[5,5];`

	0	1	2	3	4
0	0	6	2	-9	8
1	2	3	6	55	1
2	1	8	-5	2	8
3	1	4	7	32	7
4	0	2	3	9	8

Ou seja, o elemento
marcado a esquerda é
referenciado por:
`mat[2,3]`

Propriedades:

- Os índices sempre iniciam em zero.
- Por exemplo, para uma matriz com **L** linhas e **C** colunas:
 - os índices das linhas variam de **[0, L-1]**
 - os índices das colunas variam de **[0, C-1]**
- Logo, uma matriz com 5 linhas e 10 colunas:
 - os índices das linhas variam de **[0, 4]**
 - os índices das colunas variam de **[0, 9]**

Declaração:

- A **declaração** de uma variável do tipo Matriz na linguagem C# segue o formato:
 - `tipo[,] nome = new tipo[qtde_linhas,qtde_colunas];`
- onde:
 - **tipo** é o tipo de dados dos elementos que serão armazenados;
 - **nome** é o nome da variável (nome da matriz)
 - **qtde_linhas** é um valor numérico que indica quantas linhas estarão armazenadas na matriz, do tipo de dado especificado.
 - **qtde_colunas** é um valor numérico que indica quantas linhas estarão armazenadas na matriz, do tipo de dado especificado.

Elementos:

- Os elementos são **referenciados** por **índices** de linha e coluna.
- Na ilustração a seguir, há a representação de uma matriz de elementos inteiros, com 3 linhas e 4 colunas, declarada como `int[,] A = new int[3,4]`.

$$A = \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \end{pmatrix}$$



Elemento da matriz

Índice de um elemento da matriz:
linha e coluna

Manipulação:

- Para acessar um elemento da matriz, deve-se usar o **nome da matriz** e os **valor dos índices da linha e da coluna** do elemento, entre colchetes;

```
double[,] A = new double[2,2];
```

```
A[0,0] = 9.0;
```

```
A[0,1] = 8.0;
```

```
A[1,0] = 7.0;
```

```
A[1,1] = 7.0;
```

$$A = \begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix}$$



:>

Manipulação:

- Para acessar um elemento da matriz, deve-se usar o **nome da matriz** e os **valor dos índices da linha e da coluna** do elemento, entre colchetes;

```
double[,] A = new double[2,2];
```

```
A[0,0] = 9.0;
```

```
A[0,1] = 8.0;
```

```
A[1,0] = 7.0;
```

```
A[1,1] = 7.0;
```

$$A = \begin{pmatrix} 9.0 & 8.0 \\ 7.0 & 7.0 \end{pmatrix}$$



```
:>
```


Manipulação:

- Para acessar um elemento da matriz, deve-se usar o **nome da matriz** e os **valor dos índices da linha e da coluna** do elemento, entre colchetes;

```
double[,] A = new double[2,2];  
    A[0,0] = 9.0;  
    A[0,1] = 8.0;  
    A[1,0] = 7.0;  
    A[1,1] = 7.0;  
quant=A[0,0]+A[0,1]+A[1,0]+A[1  
    ,1];  
    Console.WriteLine(quant);
```

$$A = \begin{pmatrix} 9.0 & 8.0 \\ 7.0 & 7.0 \end{pmatrix}$$



:>

Manipulação:

- Para acessar um elemento da matriz, deve-se usar o **nome da matriz** e os **valor dos índices da linha e da coluna** do elemento, entre colchetes;

```
double[,] A = new double[2,2];
```

```
    A[0,0] = 9.0;
```

```
    A[0,1] = 8.0;
```

```
    A[1,0] = 7.0;
```

```
    A[1,1] = 7.0;
```

```
quant=A[0,0]+A[0,1]+A[1,0]+A[1  
    ,1];
```

```
    Console.WriteLine(quant);
```

$$A = \begin{pmatrix} 9.0 & 8.0 \\ 7.0 & 7.0 \end{pmatrix}$$

:>31.0000

Manipulação:

- Para acessar um elemento da matriz, deve-se usar o **nome da matriz** e os **valor dos índices da linha e da coluna** do elemento, entre colchetes;

```
double[,] A = new double[2,2];  
    A[0,0] = 9.0;  
    A[0,1] = 8.0;  
    A[1,0] = 7.0;  
    A[1,1] = 7.0;  
quant=A[0,0]+A[0,1]+A[1,0]+A[1,1];  
    Console.WriteLine(quant);  
    media = quant/4;  
    Console.WriteLine(media);
```

$$A = \begin{pmatrix} 9.0 & 8.0 \\ 7.0 & 7.0 \end{pmatrix}$$

:>31.0000

:>7.75000

Manipulação através de um laço:

- Em um vetor, utilizamos um **for** para realizar a leitura de elementos;
- Em matrizes, utilizaremos um **for** dentro de um outro **for**;

$$A = \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \end{pmatrix}$$

– Assim, para listar todos os elementos da matriz na tela, pode-se usar o seguinte código:

```
int i, j;
int[,] A = new int[3,4];
for(i=0;i<3;i++){ //laço que percorre as linhas da matriz
    for(j=0;j<4;j++){ //laço que percorre as colunas da matriz
        Console.WriteLine("A["+i+", "+j+"] = "+A[i,j]);
    }
}
```

Manipulação:

- para fazer a leitura dos elementos de uma matriz, utiliza-se a função **Console.ReadLine()**:

$$A = \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \end{pmatrix}$$

- para fazer a leitura dos elementos de uma matriz:

```
int i, j;
int[,] A = new int[3,4];
for(i=0;i<3;i++){ //laço que percorre as linhas da matriz
    for(j=0;j<4;j++){ //laço que percorre as colunas da matriz
        Console.WriteLine(" Digite o elemento "A["+i+", "+j+"]");
        A[i,j] = int.Parse(Console.ReadLine());
    }
}
```