

# Arranjos - Matrizes



Professor: Humberto Nigri

[hnigri@gmail.com](mailto:hnigri@gmail.com)

# Necessidade de uso de Matrizes

- # Certas situações exigem a manipulação de dados de uma forma que os vetores não são suficientes para resolver.

Ler a nota de todas as matérias de todos os alunos do 1o. período

1a Solução: `int [] mat1 = new int[40]; ...`  
`int [] mat5 = new int[40]; // 1 p/ cada matéria`

**Problema: tratar cada variável (vetor) individualmente**

```
Console.WriteLine("Entre com as notas de IC ");
for (i = 0; i < 40; i++) {
    Console.WriteLine("entre com a nota " + (i+1));
    mat1[ i ] = int.Parse(Console.ReadLine());
}

:      :      :

Console.WriteLine("Entre com as notas de ATP ");
for (i = 0; i < 40; i++) {
    Console.WriteLine("entre com a nota " + (i+1));
    mat5[ i ] = int.Parse(Console.ReadLine());
}
```

# Definição de Matriz

---

\* Uma matriz é uma variável, **composta**,  
**homogênea**, **multidimensional**

Sua representação é feita em apenas mais de uma dimensão, no caso, vamos usar linha e coluna

Armazena mais de um valor em uma mesma variável

Os valores devem ser de um mesmo tipo, por exemplo, int, float, double ou char

# Matriz - Representação

\* Considere que a matriz abaixo possui 3 linhas e 5 colunas e seu nome é **matrizX**:

```
int [ , ] matrizX = new int [3, 5]
```

└─ Coluna  
└─ Linha

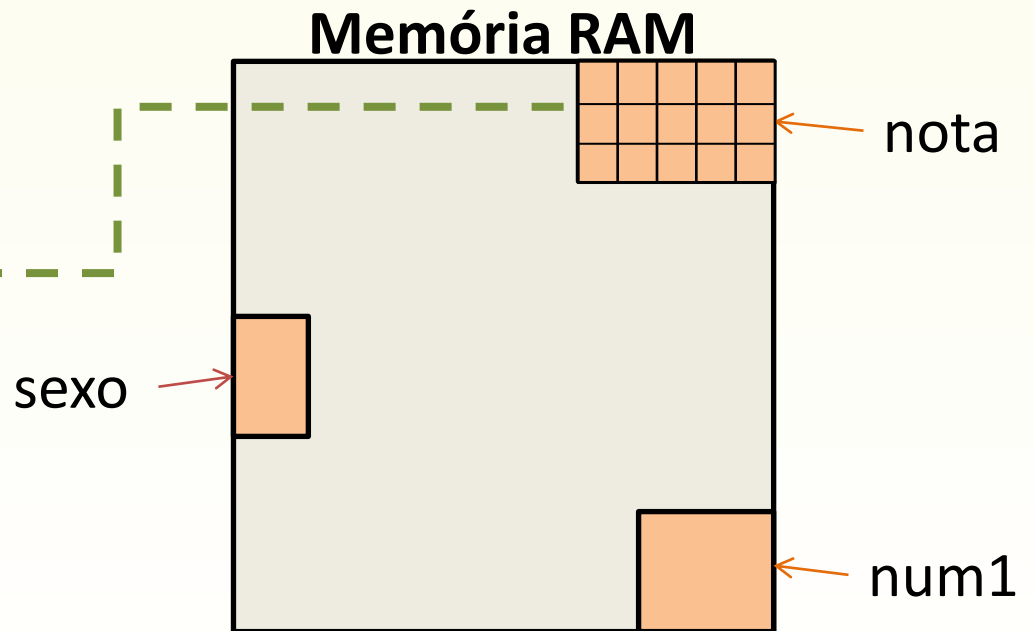
índice  
s

0					
1					
2					

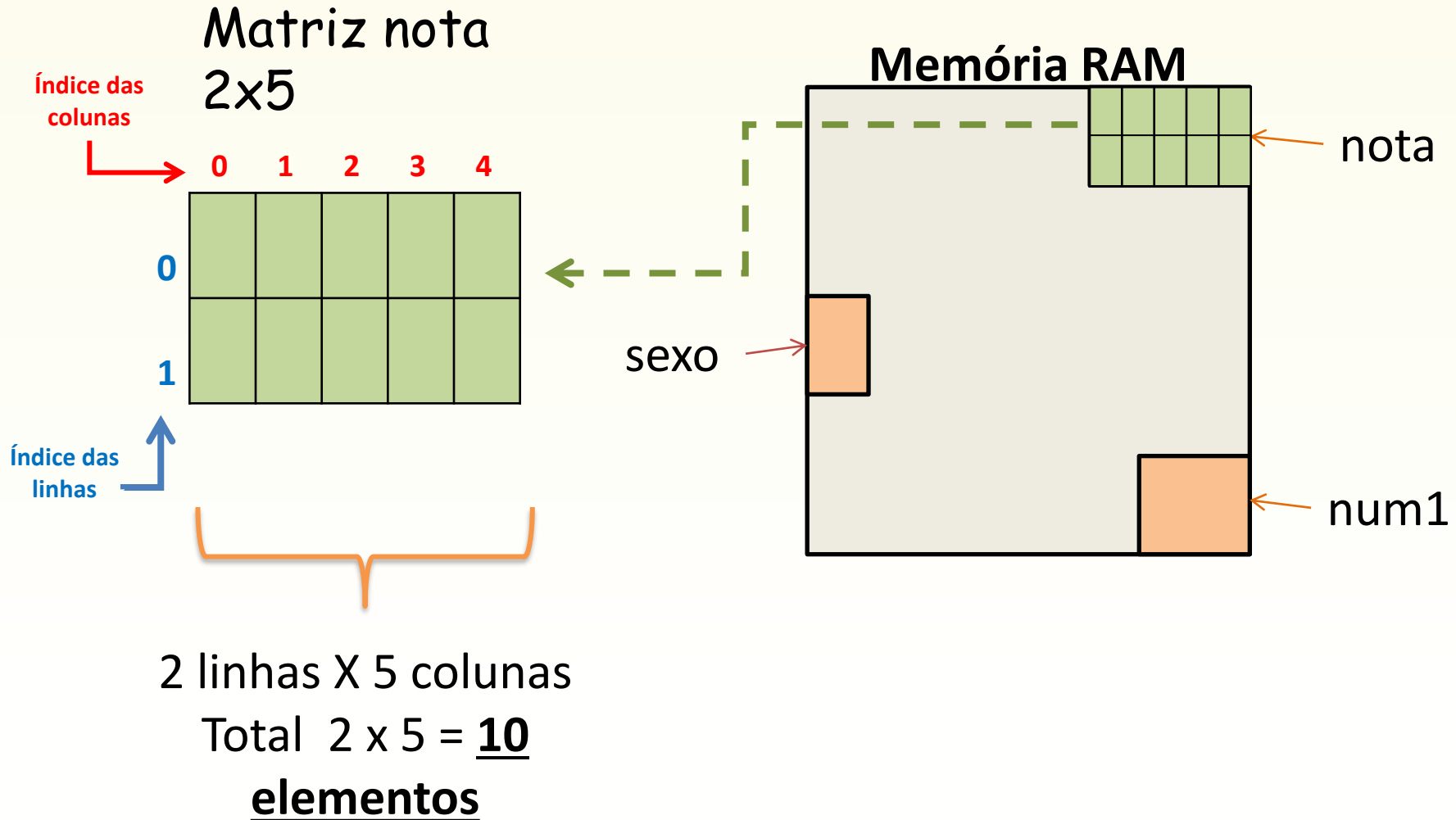
# Matriz - Representação

Com a matriz: é possível armazenar um valor em cada linha e coluna da variável.

Observe que é possível armazenar mais valores.



# Matriz - Representação



# Matriz - Exemplo

```
int[ , ] nota = new int [2,5];
```



Matriz de inteiros

nota[0,0], nota[0,1], nota[0,2], nota[0,3], nota[0,4]

nota[1,0], nota[1,1], nota[1,2], nota[1,3], nota[1,4]

Índice das  
colunas



0 1 2 3 4

0

	0	1	2	3	4
0	12	15	20	2	15
1	22	18	30	32	1

1

Índice das  
linhas



# Matriz - Declaração em C#

## # Declarando uma variável do tipo matriz

Tipo[ , ] nome = new Tipo[Qtde de linhas , Qtde de Colunas];

- # **Tipo** : indica do tipo de cada elemento da matriz
- # **nome**: indica o nome da coleção de variáveis (seguir mesmas regras para um nome de variável qualquer)
- # **Qtde de linhas**: Quantidade de linhas que a matriz deverá ter
- # **Qtde de Colunas**: Quantidade de colunas que a matriz deverá ter



# Matriz - Declaração em C#

---

```
Tipo[ , ] nome = new Tipo[Qtde de linhas ,  
                           Qtde de Colunas];
```

# Exemplos:

```
int[ , ] idade = new int[2, 4];
```

```
double[ , ] peso = new double[10, 5];
```

```
string[ , ] nome = new string[3, 6];
```

# Necessidade de uso de Matrizes

- # Para resolver nosso problema inicial (e outros) usaremos as matrizes, que estendem o conceito dos vetores para duas (ou mais) dimensões
- # Em C# declaramos uma matriz da seguinte forma:

```
int [,] materia = new int [ 5, 40 ]; // temos 5 matérias, cada uma com 40 alunos
```

Agora temos:

```
int [,] materia = new int [ 5, 40 ];

for ( int i = 0 ; i < 5; i++ ) {
    Console.WriteLine("Entre com as notas da matéria " +( i+1));
    for ( int j = 0; j < 40; j++ ) {
        Console.WriteLine("\n Entre com a nota  do aluno " +( j+1));
        materia [ i , j ] = int.Parse(Console.ReadLine());
    }
}
```

# Matrizes - Inicialização

- # Problemas que são possíveis de tratar como uma tabela são os mais indicados para uso de matrizes

Exemplo:

```
int [,] nota = new int [ 3, 4 ];
```

Podemos inicializar :

```
int [,] nota = { {0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0, 0} }
```

Em lugar de ...

```
nota[0, 0] = 0; nota[0, 1] = 0; nota[0, 2] = 0; nota[0, 3] = 0;  
nota[1, 0] = 0; nota[1, 1] = 0; nota[1, 2] = 0; nota[1, 3] = 0;  
nota[2, 0] = 0; nota[2, 1] = 0; nota[2, 2] = 0; nota[2, 3] = 0;
```

Ou usar comandos for:

```
for ( i = 0; i < 3; i++ )  
    for ( j = 0; j < 4; j++ )  
        nota [ i , j ] = 0;
```

# Exemplo

- # Ler duas matrizes 4 x 4 de inteiros e imprimir a soma das duas

Primeiro: fazer uma função para ler as matrizes ...

```
void LeMatriz(int [,] Matriz)
{
    for (int i = 0; i < 4; i++)
        for (int j = 0; j < 4; j++) {
            Console.WriteLine("[{0}, {1}] = ", i, j);
            Matriz[i, j] = int.Parse(Console.ReadLine());
        }
}
```

# Exemplo

- # Ler duas matrizes 4 x 4 de inteiros e imprimir a soma das duas

Mas a função ficará melhor assim ...

```
void LeMatriz(int [,] Matriz)
{
    for (int i = 0; i < Matriz.GetLength(0); i++)
        for (int j = 0; j < Matriz.GetLength(1); j++) {
            Console.WriteLine("{0}, {1} = ", i, j);
            Matriz[i, j] = int.Parse(Console.ReadLine());
        }
}
```

# Exemplo

```
void Main() {  
    int [,]A = new int [4, 4];  
    int [,]B = new int [4, 4];  
    int [,]C = new int [4, 4];  
  
    LeMatriz(A);    LeMatriz(B);  
    for (int i = 0; i < A.GetLength(0); i++) // soma matricial  
        for (int j = 0; j < A.GetLength(1); j++)  
            C[i, j] = A[i, j] + B[i, j];  
    ImprimeMatriz(C);  
}
```

# Exercícios

1. Faça um programa que leia duas matrizes  $A$  e  $B$  ( $3 \times 3$ ) de elementos inteiros e calcule e imprima a matriz  $C$  que é a soma matricial das duas anteriores
2. Faça um programa que leia uma matriz  $A$  ( $3 \times 3$ ), calcule e imprima a soma dos elementos da diagonal principal da matriz
3. Faça um programa que leia uma matriz  $A$  ( $3 \times 3$ ), calcule e imprima a matriz  $T$  que é a transposta de  $A$ . Definição pra quem não "lembra": Em uma matriz transposta as linhas viram colunas e as colunas viram as linhas.
4. Faça um programa que leia duas matrizes  $A$  e  $B$  ( $3 \times 3$ ), calcule e imprima a matriz  $E$ , onde cada elemento de  $E$  contém o maior elemento da respectiva posição de  $A$  e  $B$ . Ou seja:  $E[i, j] = \text{MAIOR}(A[i, j], B[i, j])$ , para cada  $i$  e  $j$ .

# Exercícios

- # Faça um programa que, utilizando funções faça:
- a) Leia duas matrizes A e B (3x3) de elementos inteiros
  - b) Calcule e imprima a matriz C que é a soma matricial das duas anteriores
  - c) Calcule e imprima a soma dos elementos da diagonal principal da matriz C
  - d) Calcule e imprima a matriz D que é a transposta de C.  
Definição pra quem não "lembra": Em uma matriz transposta as linhas viram colunas e as colunas viram as linhas.
  - e) Calcule e imprima a matriz E, onde cada elemento de E contém o maior elemento da respectiva posição de A e B.  
Ou seja:  $E[i,j] = \text{MAIOR}(A[i,j], B[i,j])$ .



# Solução - Programa principal

```
void main () {  
    int [,] A = new int[3, 3]; int [,] B = new int [3, 3];  
    int [,] C = new int [3, 3]; int [,] D = new int [3, 3]  
    int [,] E = new int [3, 3];  
  
    LeMatriz(A); LeMatriz(B);  
    SomaMatrizes(A, B, C);  
    ImprimeMatriz(C, "Soma das Matrizes");  
    Console.WriteLine("Soma da diagonal principal = " +  
                      SomaDiagonal(C) );  
  
    TranspoeMatriz(C, D);  
    ImprimeMatriz(D,, "Matriz da soma Transposta");  
    MatrizMaior(A, B, E);  
    ImprimeMatriz(E, "Matriz dos maiores valores");  
}
```

# Solução - Funções

a) Função para ler as matrizes ...

```
void LeMatriz(int [,]Mat) {  
    for (int i = 0; i < Mat.GetLength(0); i++)  
        for (int j = 0; j < Mat.GetLength(1); j++) {  
            Console.WriteLine("[{0}, {1}] = ", i, j);  
            Mat[i, j] = int.Parse(Console.ReadLine());  
        }  
}
```

b) Função para Soma de Matrizes

```
void SomaMatrizes(int [,] A, int [,] B, int [,] C) {  
    for (int i = 0; i < A.GetLength(0); i++)  
        for (int j = 0; j < A.GetLength(1); j++)  
            C[i, j] = A[i, j] + B[i, j];  
}
```

# Solução

c) Função somar a diagonal principal

```
int SomaDiagonal(int [,] Matriz)
{ // A diagonal principal é aquela onde a linha == coluna
  int i, soma = 0;
  for (i = 0; i < Matriz.GetLength(0); i++) // matriz quadrada !
    soma = soma + Matriz[i, i];
  return soma;
}
```

d) Matriz Transposta

```
void TranspoeMatriz(int [,] A, int [,] T)
{ // Supondo dimensões das matrizes permitem a transposição
  for (int i = 0; i < A.GetLength(0); i++)
    for (int j = 0; j < A.GetLength(1); j++)
      T[i, j] = A[j, i]; // moleza !
}
```

# Solução

e) Gerar matriz com o maior de cada

```
void MatrizMaior(int [,] A, int [,] B, int [,] M)
```

```
{  
    int i, j;  
    for (i = 0; i < A.GetLength(0); i++)  
        for (j = 0; j < A.GetLength(1); j++)  
            if ( A[i, j] > B[i, j] )  
                M[i, j] = A[i, j];  
            else M[i, j] = B[i, j];  
}
```

```
void ImprimeMatriz(int [,] A, String mensagem)
```

```
{ int i, j;  
    Console.WriteLine( mensagem );  
    for (i = 0; i < A.GetLength(0); i++) {  
        for (j = 0; j < A.GetLength(1); j++)  
            Console.Write( A[i, j] + "\t" );  
        Console.WriteLine(); // para a próxima linha  
    }  
}
```

# Exercício

---

- \* Fazer um programa que leia uma matriz  $4 \times 4$  e calcule:
  - a) A média dos seus elementos
  - b) O maior e o menor valor dos números armazenados nela
  - c) Imprimir a matriz