



**TECNOLOGIA E INOVAÇÃO  
EM PROL DA INDÚSTRIA**



**Técnico em Informática**

# Programação de Aplicativos– 140h

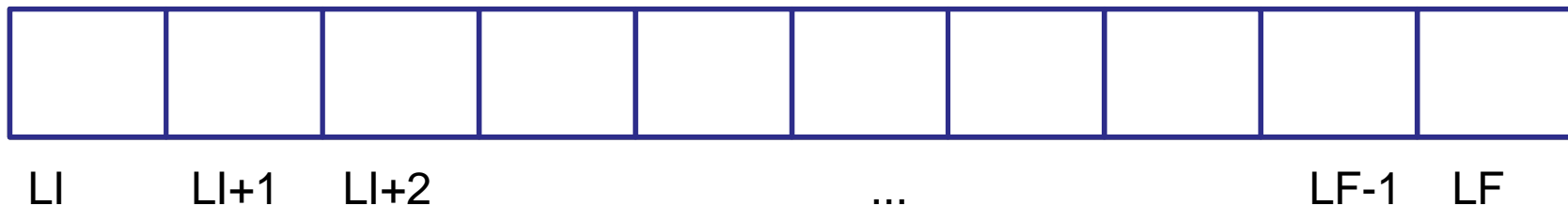
Prof<sup>a</sup>: Francisleide Almeida

# Vetores

- Um vetor é uma espécie de caixa com várias divisórias para armazenar coisas (dados)
- Variável que pode armazenar vários valores



- Os vetores são definidos pelo tipo de dados que eles devem armazenar e a quantidade de posições;



# Variáveis Compostas Homogênea

- **Homogênea:** Armazena os mesmos tipos de dados em todas as posições;
- **Ex:**
  - Vetor de 8 posições para armazenar números reais;
  - Vetor de 40 posições para armazenar caracteres;

```
algoritmo "exemplo_vetores"  
var  
    nome_alunos: vetor [1..50] de caractere  
    media_alunos: vetor [1..50] de real  
inicio
```

## Atribuição de valores no vetor

- `<nome_variavel> [<posição>] <- <valor>`
- `<nome_variavel> [<posição>] := <valor>`
- `Leia <nome_variavel> [<posição>]`

```
algoritmo "exemplo_vetores"
var
    nome_alunos: vetor [1..50] de caractere
    media_alunos: vetor [1..50] de real
inicio
    nome_alunos[1] ← "Pedro"
    leia(nome_alunos[2])
    nome_alunos[3] := "Joana"
    media_alunos[1] := 8.5
```

# Preenchimento do vetor com repetição

- Ex:

---

```
algoritmo "exemplo_vetores"  
var  
numeros: vetor [1..10] de inteiro  
i: inteiro  
inicio  
  para i de 1 ate 10 faca  
    escreva("Digite um valor para ser adicionado ao vetor: ")  
    leia(numeros[i])  
  fimpara  
fimalgoritmo
```



# Atividade

- Criar um algoritmo que leia 10 números pelo teclado e exiba os números na ordem correta que os números foram digitados.
- Criar um algoritmo que leia 10 números pelo teclado e exiba os números na ordem inversa da que os números foram digitados.

- Escreva um algoritmo que leia um vetor com 10 posições de números inteiros. Em seguida, receba um novo valor do usuário e verifique se este valor se encontra no vetor.

# Matrizes

- São como vetores, porém tem mais de uma dimensão.
- Imagine que no edifício, além de poder pegar um elevador para andares diferentes, agora possa acessar os apartamentos distintos de cada andar.

# Vetor x Matrizes

- Vetores têm apenas um índice;
- Matrizes possuem mais de um índice;
- Vetores são unidimensionais;
- Matrizes são bidimensionais.

# Relembrando matrizes

Matriz  $m$  por  $n$

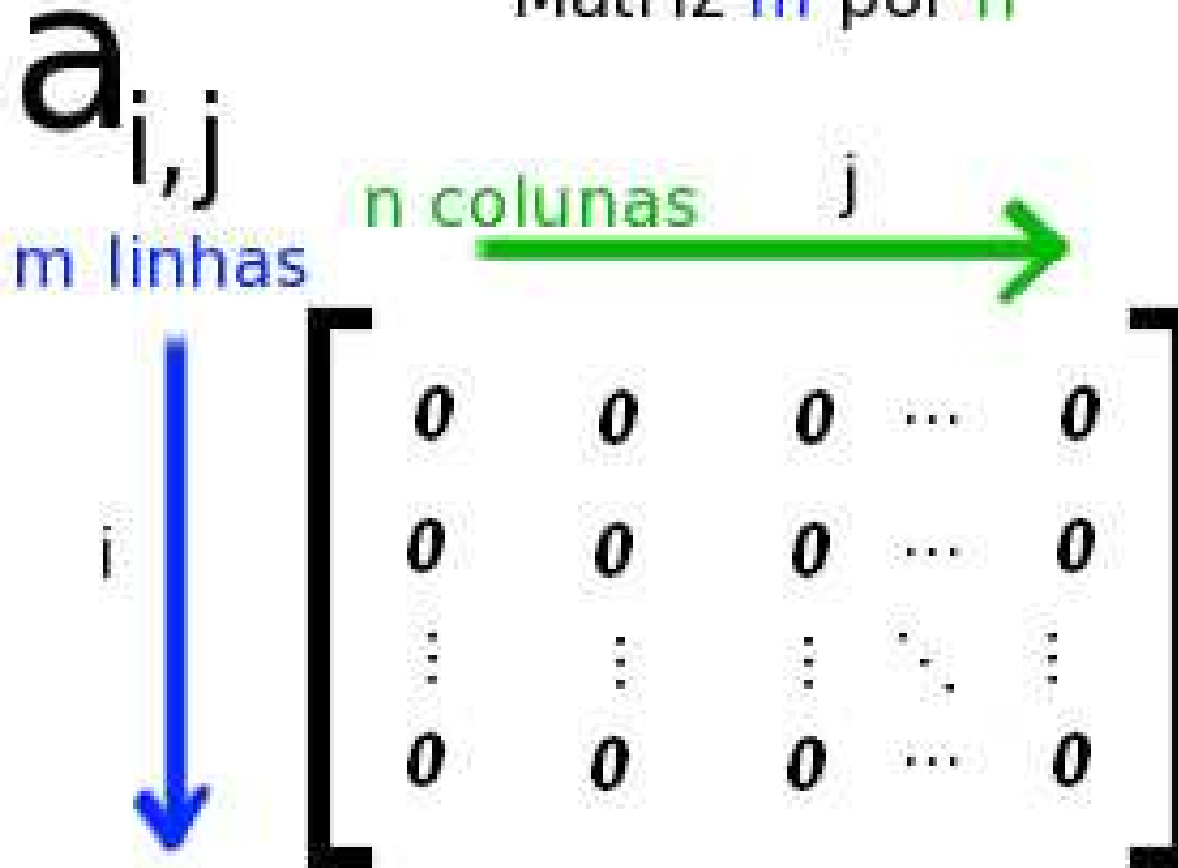
$a_{i,j}$

$m$  linhas

$n$  colunas

$j$

$i$



0	0	0	...	0
0	0	0	...	0
...	...	...	...	...
0	0	0	...	0

# Matrizes

- As matrizes são, comumente referenciadas através de suas dimensões (quantidade de linhas e colunas)
- A notação comum é:  $M \times N$ , onde
  - $M$  é a dimensão vertical (quantidade de linhas)
  - $N$  é dimensão horizontal (quantidade de colunas)



# Matrizes

- Ex:

**3x2**


**2x3**


**4x1**


**3x3**


# Declaração de matrizes

- <nome\_variavel>: vetor [li..lf, ci..cf] de <tipo>
  - li e lf representam, respectivamente o índice inicial e final das linhas e
  - ci e cf representam, respectivamente o índice inicial e final das colunas

algoritmo "matriz"

var

exMatriz: vetor [1..3, 1..2] de inteiro

Colunas: de 1 a 2

Linhas: de 1 a 3

# Preenchendo a matriz

- Se quisermos atribuir valores a todas as posições da matriz, podemos fazer:

```
algoritmo "preencher_matrizes"
var
numeros: vetor[1..3, 1..2] de inteiro
i: inteiro
inicio
para i de 1 ate 3 faca //fazer o laço para as linhas
    escreva("Digite o valor para a posicao ", i, ", ", 1")
    leia(numeros[i, 1])
    escreva("Digite o valor para a posicao ", i, ", ", 2")
    leia(numeros[i, 2])
fimpara
finalgoritmo
```

# Preenchendo a matriz

- Entretanto, à medida que a quantidade de elementos da matriz aumenta, fica complicado fazermos manualmente para todas as posições.
- O melhor caminho é utilizar laços de repetição.

# Preenchendo a matriz

```
algoritmo "preencher_matrizes"  
var  
numeros: vetor[1..3, 1..2] de inteiro  
i,j: inteiro  
inicio  
para i de 1 ate 3 faca  
    para j de 1 ate 2 faca  
        escreva("Digite um valor para a posição [", i,",", j, "]: ")  
        leia (numeros[i,j])  
    fimpara  
fimpara  
fimalgoritmo
```

# Preenchendo a matriz

```
algoritmo "exemplo01"
var
  numeros: vetor[1..3, 1..3] de inteiro
  i, j: inteiro
inicio
  para i de 1 ate 3 faca
    para j de 1 ate 3 faca
      escreva("Digite um valor para a posição [", i, ",", j, "]: ")
      leia(numeros[i,j])
    fimpara
  fimpara

  para i de 1 ate 3 faca
    para j de 1 ate 3 faca
      escreva(numeros[i, j])
    fimpara
  escreval
fimpara
fim
```

Ativar o Window  
Acesse Configurações



# Atividade

- Criar um algoritmo que leia uma matriz 3x3. Em seguida, exiba a soma dos elementos de cada uma das linhas.
- Ex:

1	2	2
3	2	3
4	1	1

Soma Linha 1 = 5

Soma Linha 2 = 8

Soma Linha 3 = 6

- Escreva um algoritmo que leia uma matriz 4x3. Em seguida, receba um novo valor do usuário e verifique se este valor se encontra na matriz. Caso o valor se encontre na matriz, escreva a mensagem “O valor se encontra na matriz na posição m,n”. Caso contrário, escreva a mensagem “O valor NÃO se encontra na matriz”.