



# Vetor e Matriz

---

Prof. MSc. Tiago Araújo

[tiagodavi70@gmail.com](mailto:tiagodavi70@gmail.com)

# Estrutura de Dados

---

Somente os tipos de dados vistos até agora (tipos primitivos) não são suficientes para representar toda e qualquer informação que possa surgir

A partir da composição de tipos primitivos, construiremos novos tipos, denominados tipos construídos

Esses novos tipos tem um formato denominado estrutura de dados, que define como os tipos primitivos estão organizados

# Vetor

---

Uma variável pode ser como um elemento e uma estrutura de dados, como um conjunto

Quando uma estrutura de dados é composta de variáveis com o mesmo tipo primitivo, temos um conjunto homogêneo de dados

**Vetor** é estrutura unidimensional homogênea

É importante ressaltar que vetores de qualquer dimensão são caracterizados por terem todos os seus elementos pertencentes ao mesmo tipo de dado

# Vetor

---

9	2	3	4	7	8	3	0	1	12
---	---	---	---	---	---	---	---	---	----

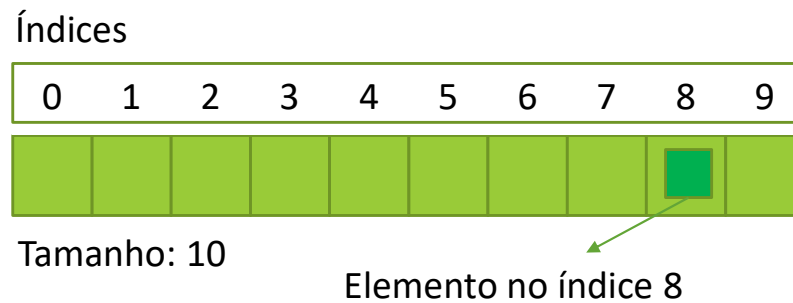
1.2	4	8.3	7.7	1.0	6.9	1.2	3.6	0.7	5.2
-----	---	-----	-----	-----	-----	-----	-----	-----	-----

# Vetor

---

Um vetor é definido inicialmente com um tamanho fixo, e a partir daí podemos percorrer sua estrutura buscando a informação desejada

Primeiro é necessário determinar qual vetor contém o valor desejado e, depois, especificar em qual posição este valor se encontra



# Vetor

---

O nome do vetor é determinado pelo mesmo nome utilizado na declaração de variáveis

A posição é determinada por meio de constante, expressão aritmética ou variável que estiver dentro dos colchetes, também denominada índice

É importante não confundir o índice com o elemento

O **índice** é a posição no vetor, enquanto o **elemento** é o que está contido naquela posição

Sobre a notação:

- O índice é referido como  $i$
- $a[i]$  é o elemento na posição  $i$  no vetor  $a$
- O tamanho do vetor é comumente referido como  $N$

# Vetor

---

a

0	1	2	3	4	5	6	7	8	9
9	2	3	4	7	8	3	0	1	12

$a[0] = 9$

$a[3] = 4$

$a[8] = 1$

b

0	1	2	3	4	5	6	7	8	9
1.2	4	8.3	7.7	1.0	6.9	1.2	3.6	0.7	5.2

$b[0] = 1.2$

$b[3] = 7.7$

$b[8] = 0.7$

# Sintaxe

---

`<nome>: vetor [<índice_inicial>..<índice_final>] de <tipo>`

`vet: vetor [0..9] de real`



# Exemplo

Crie um algoritmo que armazene 5 números em um vetor. Imprima uma listagem.

Algoritmo "salvar\_listar"

Var

vet: vetor[0..4] de inteiro

i: inteiro

Inicio

para i de 0 ate 4 faca

    leia(vet[i])

fimpara

para i de 0 ate 4 faca

    escreva(vet[i])

fimpara

Fimalgoritmo

# Exemplo

Criar um algoritmo que armazene números em dois vetores inteiros de cinco elementos cada. Gere e imprima um vetor que some ambos.

Algoritmo "soma\_vetores"

Var

a, b, c: vetor[0..4] de inteiro

i: inteiro

Inicio

para i de 0 ate 4 faca

leia(a[i])

leia(b[i])

fimpara

para i de 0 ate 4 faca

c[i] <- a[i] + b[i]

fimpara

para i de 0 ate 4 faca

escreva(c[i])

fimpara

Fimalgoritmo

# Prática

---

Criar um algoritmo que armazene nome e duas notas de 5 alunos e imprima uma listagem contendo nome, as duas notas e a média de cada aluno.

Criar um algoritmo que armazene cinco nomes em um vetor e depois possa ser digitado um nome e, se for encontrado, imprimir a posição desse nome no vetor; caso contrário, imprimir uma mensagem de não encontrado.

# Matriz

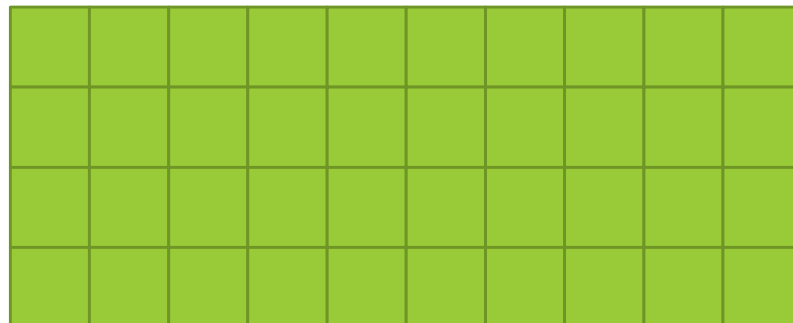
---

# Matriz

---

Vetores Multidimensionais

Estrutura de Dados Homogênea



# Matriz

Do mesmo jeito que um vetor, o acesso é feito por índices

O primeiro índice é o da linha, e o segundo o da coluna

Sobre a notação:

- Os índices são  $[i,j]$  para linha e coluna
- $[m,n]$  indica o tamanho das linhas e colunas respectivamente
- Em uma matriz quadrada o tamanho passa a ser  $N$

		Índices			
		0	1	2	3
0					8
1				9	
2		7			

$m[0,3] = 8$   
 $m[1,2] = 9$   
 $m[2,0] = 7$   
 $m[3,2] = ?$

$$A_{m \times n} = \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \end{bmatrix}$$

$a_{ij}$

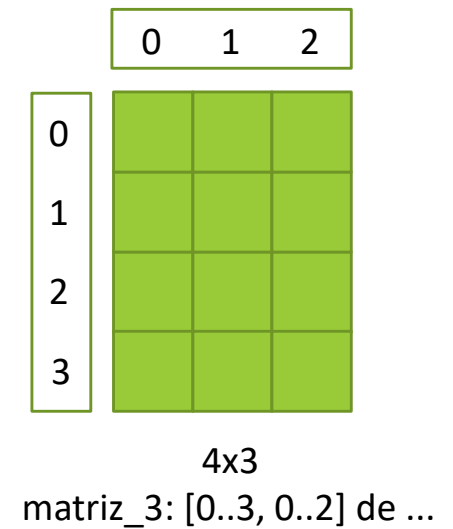
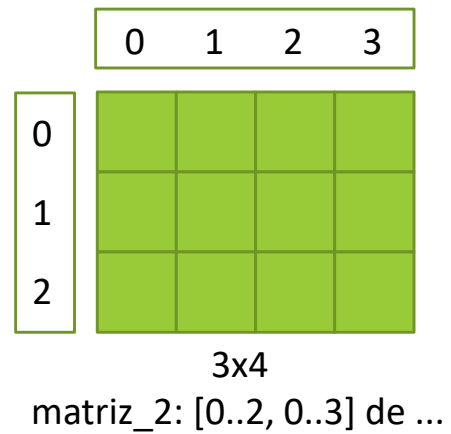
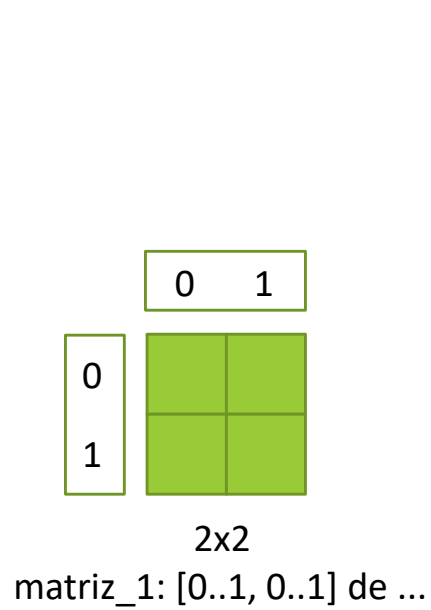
# Sintaxe

---

`<nome>: vetor [<inicial_linha>..<final_linha>, <inicial_coluna>..<final_coluna>] de <tipo>`

`matriz: vetor [0..9, 0..4] de real`

# Matriz





# Matriz

---

## Matriz Quadrada

- O número de linhas é igual o número de colunas
- Também conhecida por Matriz de Ordem  $x$  ( $x = 2, 3, 4, \dots$ )

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix} \quad \text{Ordem 3}$$

# Matriz

Diagonais Principal e Secundária

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix}$$

$$i = j$$

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix}$$

$$i + j = N + 1$$

# Matriz

---

## Matriz Identidade

- Todos os elementos da diagonal principal são iguais a 1 e todos os outros são 0
- $A[i,j] = 1$ , para  $i = j$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

## Matriz Inversa

- Uma Matriz é Invertível se for quadrada e o produto da matriz e o seu inverso for a matriz de identidade.

$$A \cdot A^{-1} = I$$

# Matriz

## Matriz Triangular Superior

- Todos os elementos à esquerda e abaixo da diagonal principal são nulos
- $A[i,j] = 0$ , para  $i > j$

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} \\ 0 & a_{11} & a_{12} \\ 0 & 0 & a_{22} \end{bmatrix}$$

## Matriz Triangular Inferior

- Todos os elementos à direita e acima da diagonal principal são nulos
- $A[i,j] = 0$ , para  $i < j$

$$\begin{bmatrix} a_{00} & 0 & 0 \\ a_{10} & a_{11} & 0 \\ a_{20} & a_{21} & a_{22} \end{bmatrix}$$

# Exemplo

Crie um algoritmo que armazene valores em uma Matriz 3x2. Imprima essa matriz em uma listagem.

Console simulando o modo texto do MS-DOS

```
Entre como numero da posição 0 0:
1
Entre como numero da posição 0 1:
2
Entre como numero da posição 1 0:
3
Entre como numero da posição 1 1:
4
Entre como numero da posição 2 0:
5
Entre como numero da posição 2 1:
6
1 2
3 4
5 6

>>> Fim da execução do programa !
```

Algoritmo "salvar\_matriz"

Var

mat: vetor[0..2,0..1] de inteiro  
i, j: inteiro

Inicio

```
para i de 0 ate 2 faca
    para j de 0 ate 1 faca
        escreva("Entre como numero da posição " ,i,j,":")
        leia(mat[i,j])
    fimpara
fimpara
```

```
para i de 0 ate 2 faca
    para j de 0 ate 1 faca
        escreva(mat[i,j])
    fimpara
    escreva()
fimpara
```

Fimalgoritmo

# Exemplo

Escreva um algoritmo que verifique se uma matriz 2x2 é uma Matriz Identidade.

Algoritmo "checar\_identidade"

Var

mat: vetor[0..1,0..1] de inteiro  
i, j: inteiro  
identidade: logico

Inicio

para i de 0 ate 1 faca  
  para j de 0 ate 1 faca  
    escreval("Entre como numero da posição " ,i,j, ":")  
    leia(mat[i,j])  
  fimpara  
fimpara

identidade <- VERDADEIRO  
para i de 0 ate 1 faca  
  para j de 0 ate 1 faca  
    se ((i <> j) e (mat[i,j] <> 0)) ou ((i = j) e (mat[i,j] <> 1)) entao  
      identidade <- FALSO  
    fimse  
  fimpara  
  escreval()  
fimpara

escreval(identidade)

Fimalgoritmo

# Prática

---

Entrar com valores para uma matriz 3x4. Gerar e imprimir uma matriz B que é o triplo da matriz A.

Leia uma matriz 4x4 e troque os valores da 1ª linha pelos da 1ª coluna.