# Algoritmos - 4

Alexandre Diehl

Departamento de Física - UFPel

# Variáveis compostas homogêneas

## Definição

**Conjunto de variáveis** do **mesmo tipo** (**numérico**, **literal** ou **lógico**), referenciáveis pelo **mesmo identificador** (nome) e **alocadas sequencialmente na memória**.

- As **variáveis** são **individualizadas** dentro do conjunto **através de um (ou mais) índice**, ou posição, que referencia a localização da variável na estrutura.

## Tipos de variáveis compostas homogêneas

- **Vetores** (variáveis compostas **unidimensionais**)
- **Matrizes** (variáveis compostas **multidimensionais**)

# Vetores

## Definição

**Variáveis compostas homogêneas unidimensionais** com apenas **um índice** para identificar os elementos do conjunto.

## Forma de declaração

**declare** nome[tamanho] **tipo**

**nome** : nome da variável do tipo vetor

**tamanho** : quantidade de variáveis (ou elementos) que formam o vetor

**tipo** : tipo básico (**numerico**, **literal**, **logico**) dos dados que serão armazenados no vetor

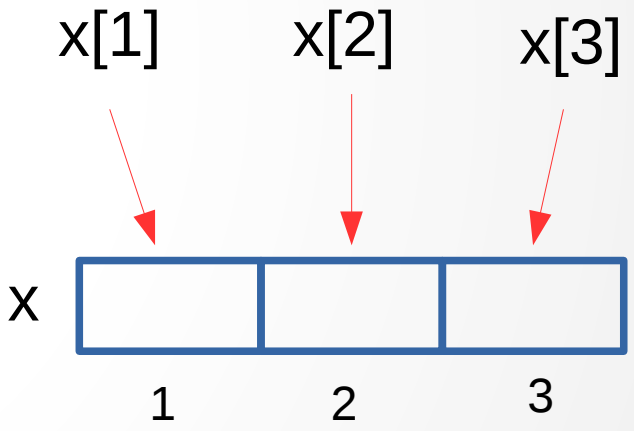
# Vetores

## Definição

**Variáveis compostas homogêneas unidimensionais** com apenas **um índice** para identificar os elementos do conjunto.

## Forma de declaração

```
declare x[3] numerico
```




**Vetor de nome x, com 3 posições**:

- nestas posições serão armazenados 3 dados numéricos.
- as posições de memória têm endereços sequenciais.

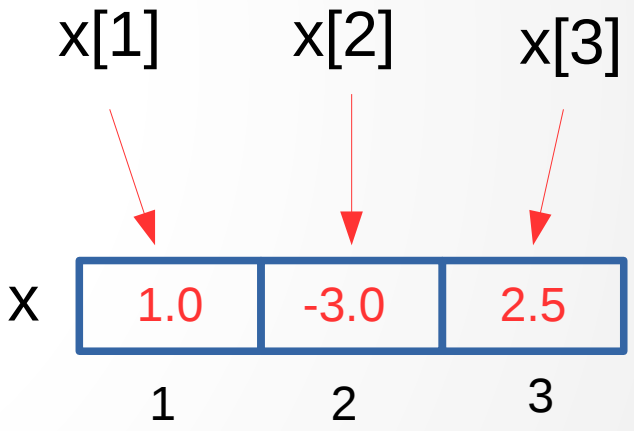
# Vetores

## Definição

**Variáveis compostas homogêneas unidimensionais** com apenas **um índice** para identificar os elementos do conjunto.

### Formas de atribuição

```
declare x[3] numerico
x[1] <- 1.0
x[2] <- -3.0
x[3] <- 2.5
```



# Vetores

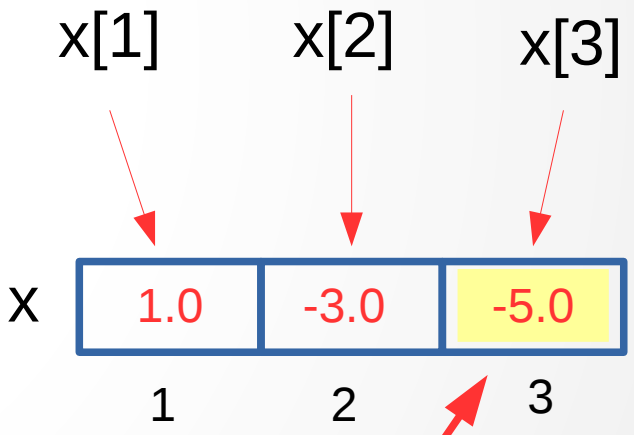
## Definição

**Variáveis compostas homogêneas unidimensionais** com apenas **um índice** para identificar os elementos do conjunto.

### Formas de atribuição

```
declare x[3] numerico
x[1] <- 1.0
x[2] <- -3.0
x[3] <- 2.5

x[3] <- x[1] + 2 * x[2]
```




Apenas a terceira posição do vetor é alterada

# Vetores

## Preenchendo um vetor usando leia

```
algoritmo
     declare x[3], i numerico
     para i <- 1 ate 3 faca passo 1
     inicio
          escreva "Digite o ", i,"º  número"
          leia x[i]
     fim
fim_algoritmo
```

Cuidado com o valor final. Deve **coincidir com o número de elementos** do vetor

Os elementos do vetor são preenchidos sequencialmente.

# Vetores

## Preenchendo um vetor usando leia

```
algoritmo
      declare x[3], i numerico
      para i <- 1 ate 4 faca passo 1
      inicio
            escreva "Di
            leia x[i]
      fim
fim_algoritmo
```

Cuidado com o valor final.
Um **erro de execução**
pode ser produzido.

```
Console
Digite o 1º número
1
Digite o 2º número
2
Digite o 3º número
3
Digite o 4º número
Erro em tempo de execução na linha 7 coluna 8
[7,8] Não foi possível acessar a posição de memória X[4]

A execução do programa foi interrompida.
```

**Erro de execução**

# Vetores

## Mostrando os elementos usando escreva

```
algoritmo
      declare x[3], i numerico
      para i <- 1 ate 3 faca passo 1
      inicio
            escreva "Digite o ", i,"º  número"
            leia x[i]
      fim
      para i <- 1 ate 3 faca passo 1
            escreva "Elemento ", i," é : ", x[i]
fim_algoritmo
```

# Vetores

## Mostrando os elementos us

```
algoritmo
     declare x[3], i numeri
     para i <- 1 ate 3 faca
     inicio
          escreva "Digite o
          leia x[i]
     fim
     para i <- 1 ate 3 faca passo 1
          escreva "Elemento ", i," é : ", x[i]
fim_algoritmo
```

```
Console
Digite o 1º número
-1.0
Digite o 2º número
2.5
Digite o 3º número
5.0
Elemento 1 é : -1.0
Elemento 2 é : 2.5
Elemento 3 é : 5.0

11 comandos executados com sucesso em 9.101 segundos
```

# Matrizes

## Definição

**Variáveis compostas homogêneas multidimensionais** com **dois ou mais índices** para identificar os elementos do conjunto. As matrizes têm mais de uma dimensão.

## Forma de declaração

**declare** nome[dimensão1, dimensão2, dimensão3,…, dimensãoN] **tipo**

**nome** : nome da variável do tipo matriz

**dimensão1** : quantidade de elementos da 1ª dimensão (linha)
**dimensão2** : quantidade de elementos da 2ª dimensão (coluna)
**dimensão3** : quantidade de elementos da 3ª dimensão (profundidade)
**dimensãoN** : quantidade de elementos da N-ésima dimensão

**tipo** : tipo básico (**numerico**, **literal**, **logico**) dos dados que serão armazenados na matriz

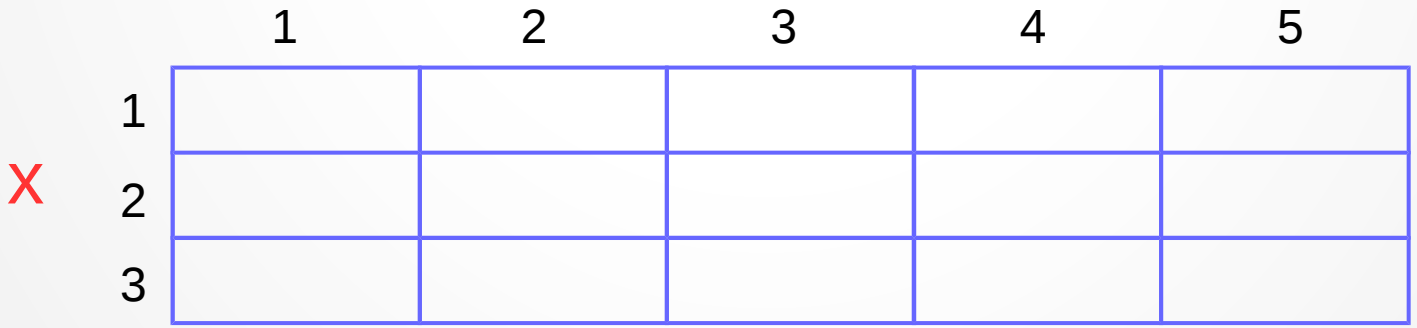
# Matrizes

## Forma de declaração

```
declare x[3,5] numerico
```

**Matriz bidimensional de nome x**:

- 1ª dimensão (linha) tem tamanho 3
- 2ª dimensão (coluna) tem tamanho 5
- A matriz tem 3 x 5 = 15 elementos



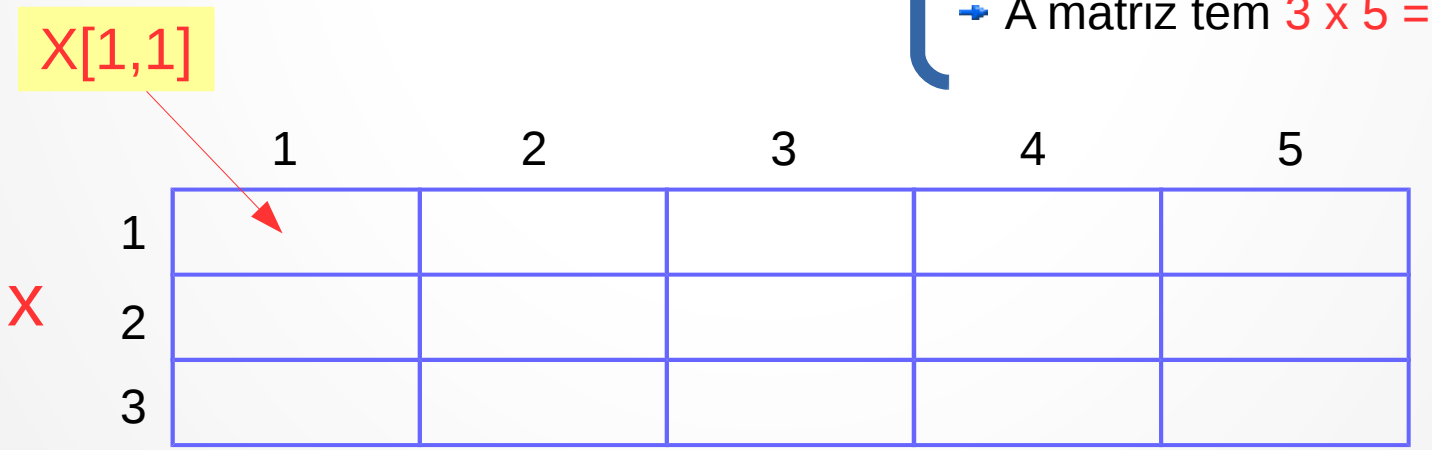
# Matrizes

## Forma de declaração

```
declare x[3,5] numerico
```

**Matriz bidimensional de nome x**:

- 1ª dimensão (linha) tem tamanho 3
- 2ª dimensão (coluna) tem tamanho 5
- A matriz tem 3 x 5 = 15 elementos



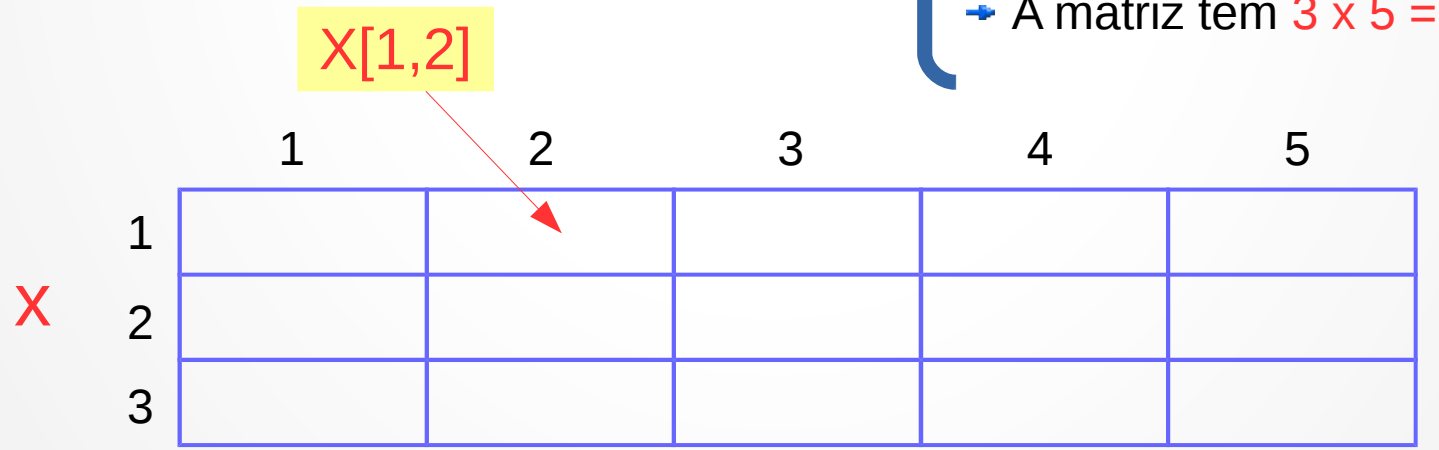
# Matrizes

## Forma de declaração

```
declare x[3,5] numerico
```

**Matriz bidimensional de nome x**:

- 1ª dimensão (linha) tem tamanho 3
- 2ª dimensão (coluna) tem tamanho 5
- A matriz tem 3 x 5 = 15 elementos



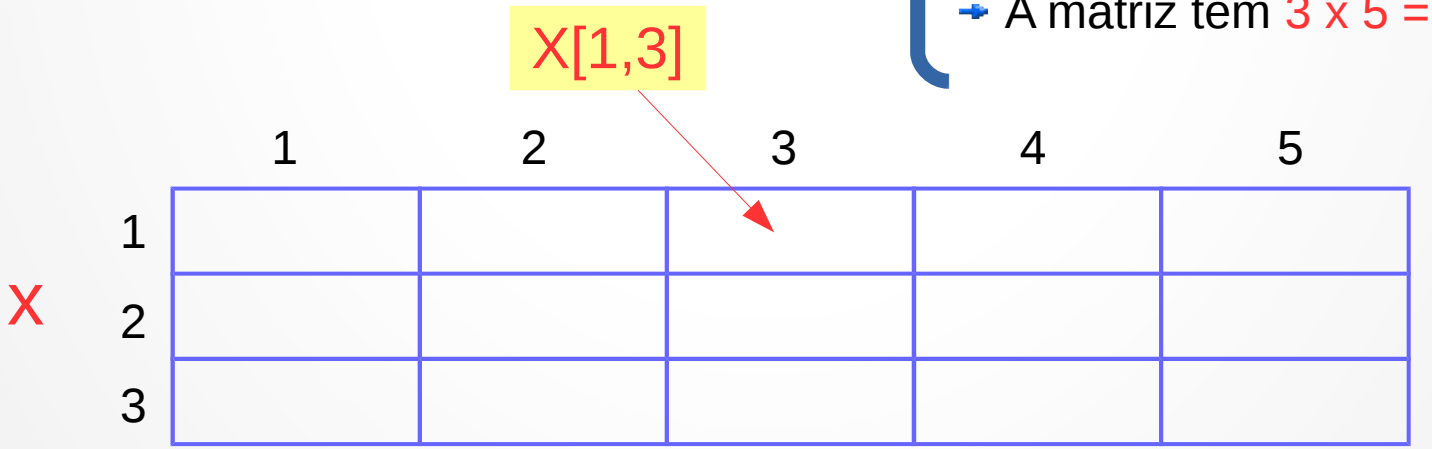
# Matrizes

## Forma de declaração

```
declare x[3,5] numerico
```

**Matriz bidimensional de nome x**:

- 1ª dimensão (linha) tem tamanho 3
- 2ª dimensão (coluna) tem tamanho 5
- A matriz tem 3 x 5 = 15 elementos



# Matrizes

## Forma de declaração

```
declare x[3,5] numerico
```

**Matriz bidimensional de nome x**:

- 1ª dimensão (linha) tem tamanho 3
- 2ª dimensão (coluna) tem tamanho 5
- A matriz tem 3 x 5 = 15 elementos

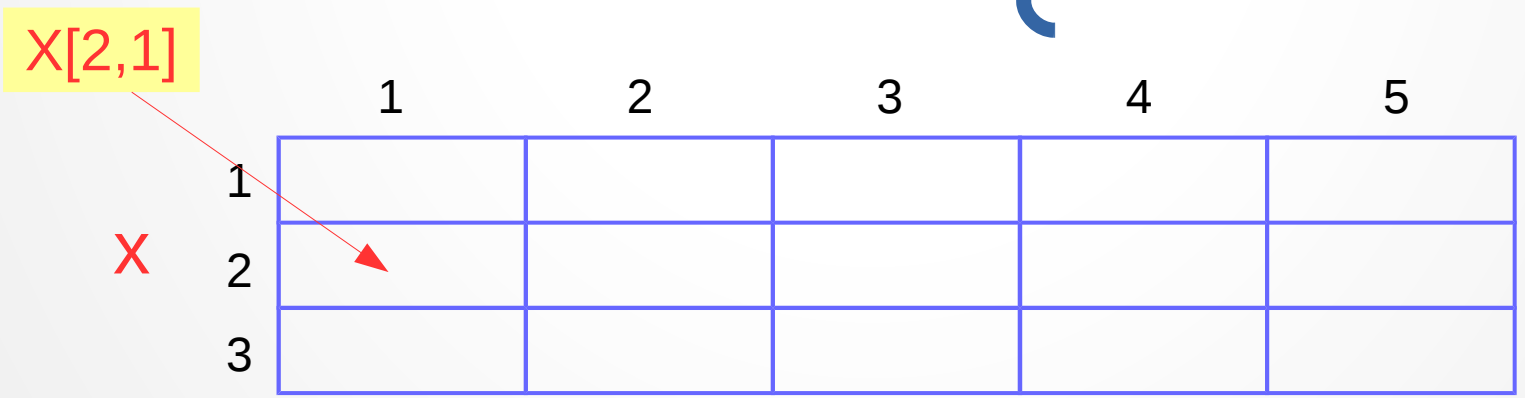
# Matrizes

## Forma de declaração

```
declare x[3,5] numerico
```

**Matriz bidimensional de nome x**:

- 1ª dimensão (linha) tem tamanho 3
- 2ª dimensão (coluna) tem tamanho 5
- A matriz tem 3 x 5 = 15 elementos



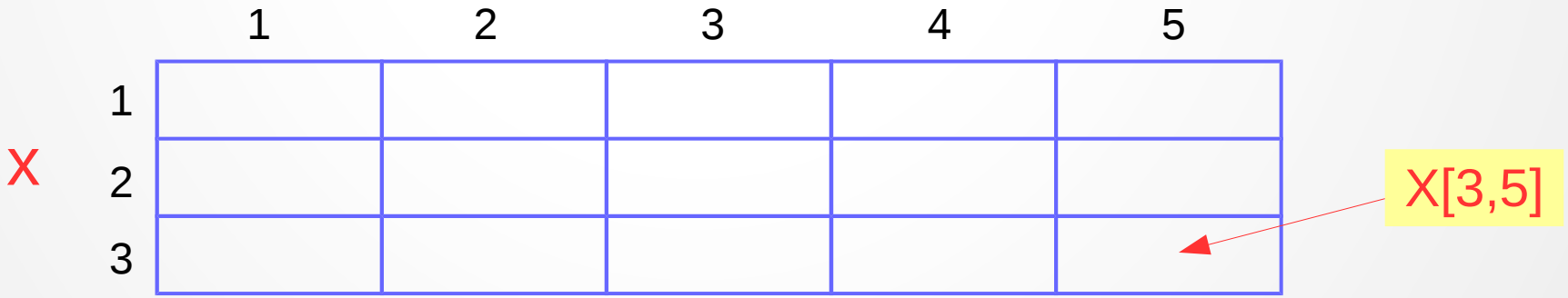
# Matrizes

## Forma de declaração

```
declare x[3,5] numerico
```

**Matriz bidimensional de nome x**:

- 1ª dimensão (linha) tem tamanho 3
- 2ª dimensão (coluna) tem tamanho 5
- A matriz tem 3 x 5 = 15 elementos



# Matrizes

## Forma de declaração

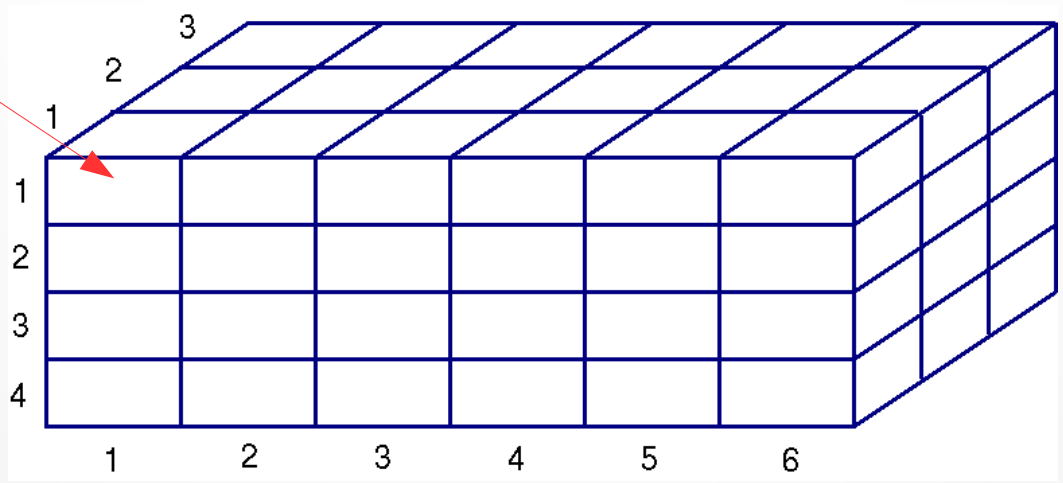
```
declare x[4,6,3] numerico
```

**Matriz tridimensional de nome x**:

- 1ª dimensão (linha) tem tamanho 4
- 2ª dimensão (coluna) tem tamanho 6
- 3ª dimensão (profundidade) tem tamanho 3
- A matriz tem 4 x 6 x 3 = 72 elementos

X[1,1,1]

X



# Matrizes

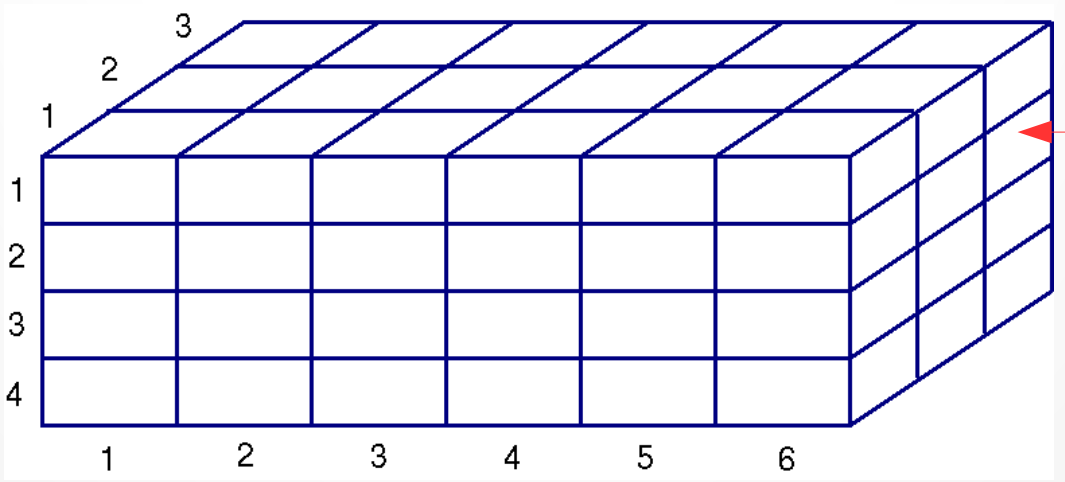
## Forma de declaração

```
declare x[4,6,3] numerico
```

**Matriz tridimensional de nome x**:

- 1ª dimensão (linha) tem tamanho 4
- 2ª dimensão (coluna) tem tamanho 6
- 3ª dimensão (profundidade) tem tamanho 3
- A matriz tem 4 x 6 x 3 = 72 elementos

X



# Matrizes

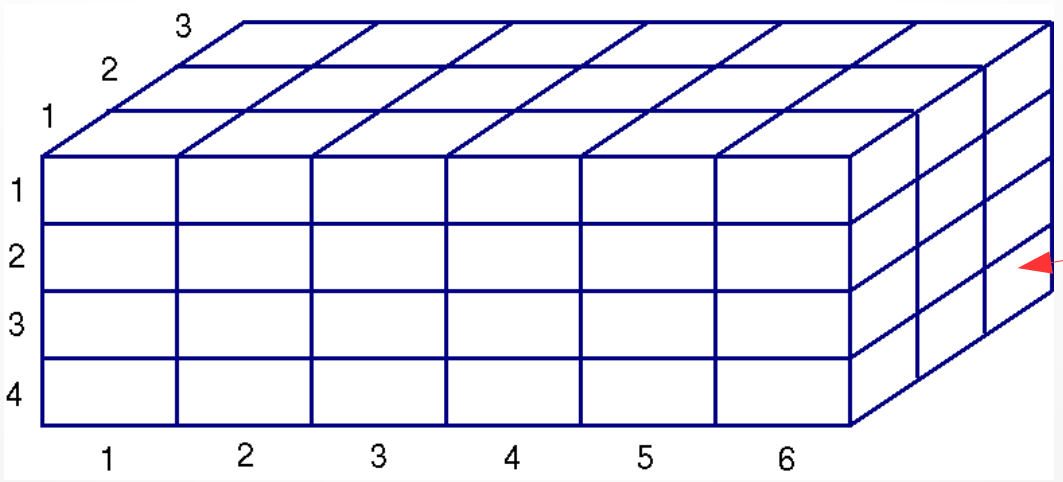
## Forma de declaração

```
declare x[4,6,3] numerico
```

**Matriz tridimensional de nome x**:

- 1ª dimensão (linha) tem tamanho 4
- 2ª dimensão (coluna) tem tamanho 6
- 3ª dimensão (profundidade) tem tamanho 3
- A matriz tem 4 x 6 x 3 = 72 elementos

X




X[4,6,3]

# Matrizes

## Formas de atribuição

```
algoritmo
      declare x[2,2] numerico
      x[1,1] <- 7
      x[1,2] <- 20
      x[2,1] <- -10
      x[2,2] <- 15
fim_algoritmo
```

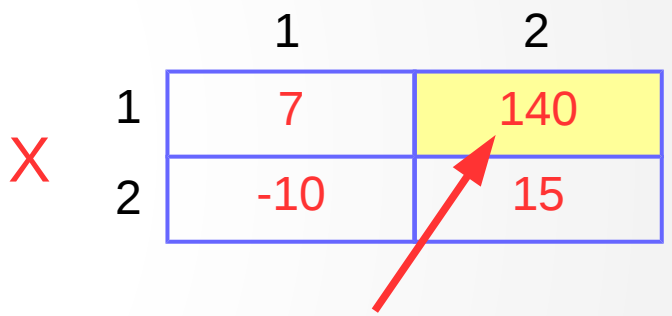
X

| | 1 | 2 |
|---|---|---|
| 1 | 7 | 20 |
| 2 | -10 | 15 |

# Matrizes

## Formas de atribuição

```
algoritmo
      declare x[2,2] numerico
      x[1,1] <- 7
      x[1,2] <- 20
      x[2,1] <- -10
      x[2,2] <- 15
      x[1,2] <- x[1,2] * x[1,1]
fim_algoritmo
```




Apenas o elemento x[1,2] da matriz é alterado

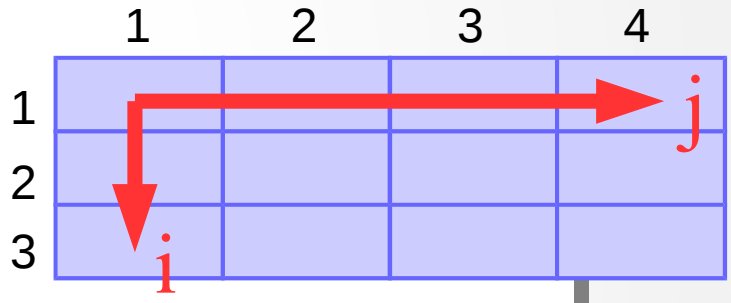
# Matrizes

## Preenchendo uma matriz usando leia

Precisamos usar **um índice inteiro** **para** percorrer **cada uma das dimensões** da matriz.

```
algoritmo
     declare x[3,4], i, j numerico
     para i <- 1 ate 3 faca passo 1
     inicio
          para j <- 1 ate 4 faca passo 1
          inicio
               escreva "Digite o número da linha ",i," e coluna ",j
               leia x[i,j]
          fim
     fim
fim_algoritmo
```

X

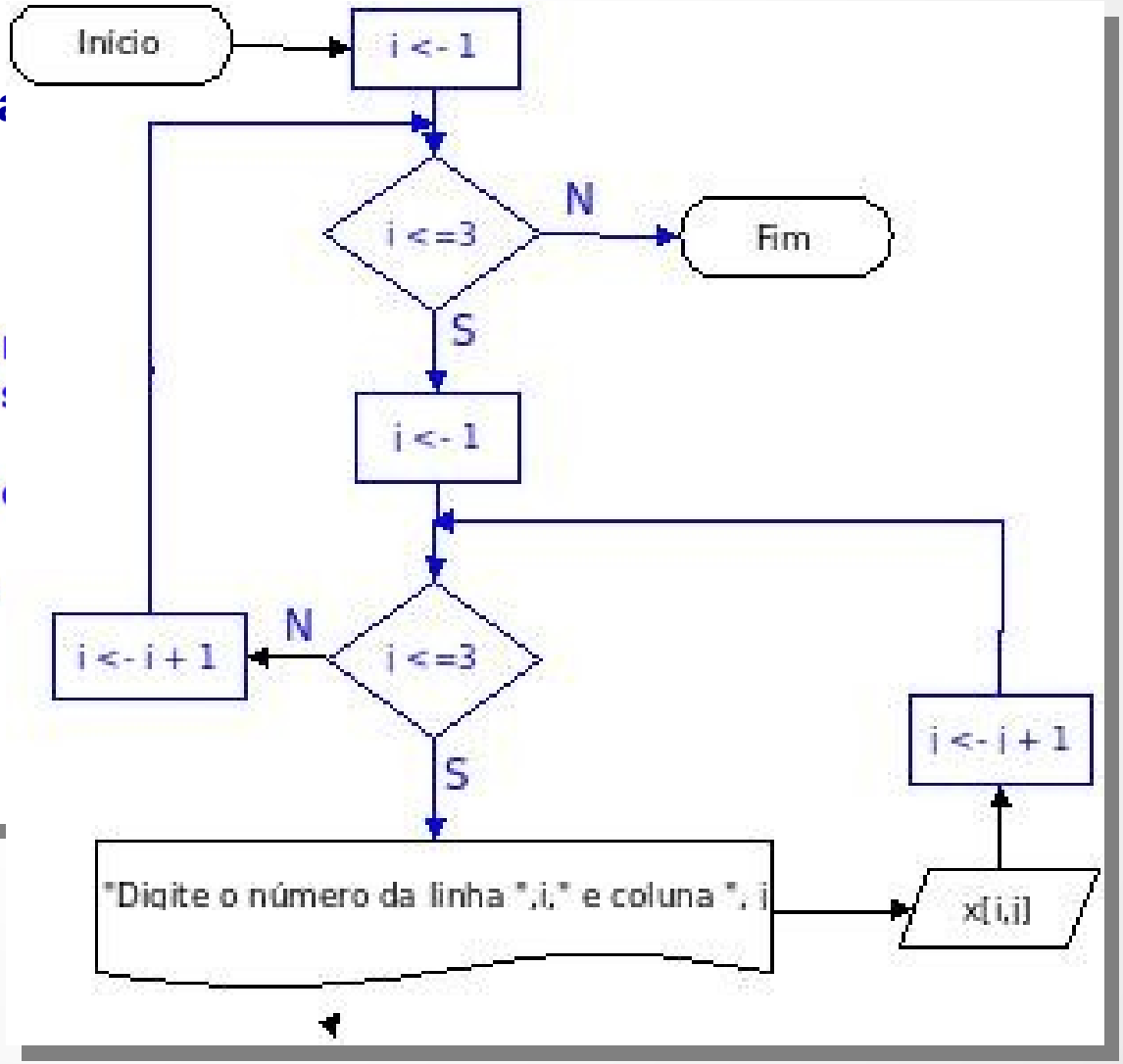


# Matrizes

## Preenchendo uma matriz usando leia

Precisamos usar **um índice inteiro para**

```
algoritmo
     declare x[3,4], i, j numer
     para i <- 1 ate 3 faca pas
     inicio
          para j <- 1 ate 4 fa
          inicio
               escreva "Digite
               leia x[i,j]
          fim
     fim
fim_algoritmo
```



# Matrizes

## Preenchendo uma matriz usando leia

Precisamos usar **um índice inteiro** **para** percorrer **cada uma das dimensões** da matriz.

```
algoritmo
     declare x[3,4,5], i, j, k numerico
     para i <- 1 ate 3 faca passo 1
     inicio
          para j <- 1 ate 4 faca passo 1
          inicio
               para k <- 1 ate 5 faca passo 1
               inicio
                    escreva "Digite o número da linha ",i," e coluna ",j,
                            " e profundidade ", k
                    leia x[i,j,k]
               fim
          fim
     fim
fim_algoritmo
```

# Matrizes

## Mostrando os elementos usando escreva

Precisamos usar **um índice inteiro** **para** percorrer **cada uma das dimensões** da matriz.

```
para i <- 1 ate 3 faca passo 1
inicio
      para j <- 1 ate 4 faca passo 1
      inicio
            para k <- 1 ate 5 faca passo 1
            inicio
                  escreva x[i,j,k]
            fim
      fim
fim
```

# Matrizes

## Percorrendo uma matriz

Uma vez definido um índice para cada dimensão da matriz, podemos percorrê-la de diferentes formas.

```
algoritmo
      declare x[2,2], i, j numerico
      x[1,1] <- 10
      x[1,2] <- -2
      x[2,1] <- 5
      x[2,2] <- 7
```

X

| | 1 | 2 |
|---|---|---|
| 1 | 10 | -2 |
| 2 | 5 | 7 |

# Matrizes

## Percorrendo uma matriz

Uma vez definido um índice para cada dimensão da matriz, podemos percorrê-la de diferentes formas.

```
algoritmo
      declare x[2,2], i, j numerico
      x[1,1] <- 10
      x[1,2] <- -2
      x[2,1] <- 5
      x[2,2] <- 7
```

X

| | 1 | 2 |
|---|---|---|
| 1 | 10 | -2 |
| 2 | 5 | 7 |

```
para i <- 1 ate 2 faca
inicio
      para j <- 1 ate 2 faca
      inicio
            escreva x[i,j]
      fim
fim
```

# Matrizes

## Percorrendo uma matriz

Uma vez definido um índice para cada dim
diferentes formas.

```
algoritmo
      declare x[2,2], i, j nume
      x[1,1] <- 10
      x[1,2] <- -2
      x[2,1] <- 5
      x[2,2] <- 7
```

```
para i <- 1 ate 2 faca
inicio
      para j <- 1 ate 2 faca
      inicio
            escreva x[i,j]
      fim
fim
```

Console

```
10
-2
5
7
```

X

| | 1 | 2 |
|---|---|---|
| 1 | 10 | -2 |
| 2 | 5 | 7 |

11 comandos executados com sucesso em 0.017 segundos

# Matrizes

## Percorrendo uma matriz

Uma vez definido um índice para cada dimensão da matriz, podemos percorrê-la de diferentes formas.

```
algoritmo
      declare x[2,2], i, j numerico
      x[1,1] <- 10
      x[1,2] <- -2
      x[2,1] <- 5
      x[2,2] <- 7
```

| | 1 | 2 |
|---|---|---|
| 1 | 10 | -2 |
| 2 | 5 | 7 |

```
para i <- 1 ate 2 faca
inicio
      para j <- 1 ate 2 faca
      inicio
            escreva x[j,i]
      fim
fim
```

# Matrizes

## Percorrendo uma matriz

Uma vez definido um índice para cada dim
diferentes formas.

```
algoritmo
      declare x[2,2], i, j nume
      x[1,1] <- 10
      x[1,2] <- -2
      x[2,1] <- 5
      x[2,2] <- 7
```

```
para i <- 1 ate 2 faca
inicio
      para j <- 1 ate 2 faca
      inicio
            escreva x[j,i]
      fim
fim
```

Console

```
10
5
-2
7
```

X

| | 1 | 2 |
|---|---|---|
| 1 | 10 | -2 |
| 2 | 5 | 7 |

11 comandos executados com sucesso em 0.015 segundos