



ALGORITMOS

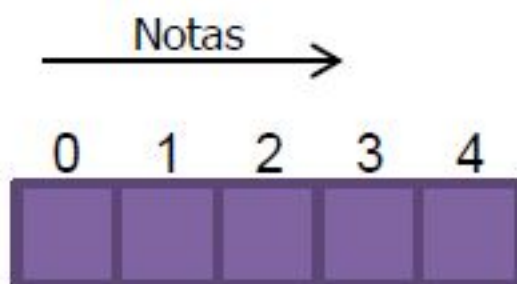
PROF^º: SIMONE DOMINICO

ARRANJOS MULTIDIMENSIONAIS

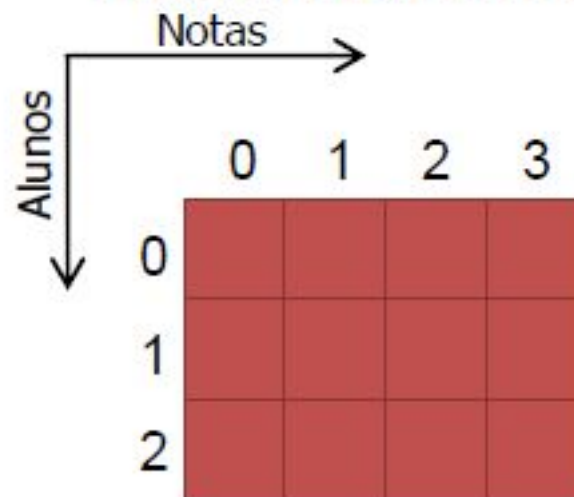
ARRANJOS MULTIDIMENSIONAIS

- Uma dimensão → vetores
- Duas dimensões → tabelas ou matrizes
- Três dimensões ou mais ...

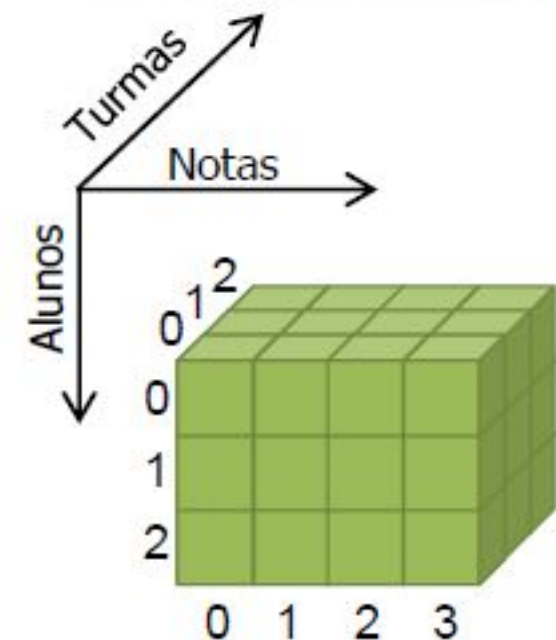
Vetor



Matriz Bidimensional



Matriz Tridimensional



MATRIZES

- São arranjos bidimensionais, nos quais as informações são organizadas em

Linhas e **Colunas**

	Coluna 0	Coluna 1	Coluna 2	Coluna 3
Linha 0	$M[0][0]$	$M[0][1]$	$M[0][2]$	$M[0][3]$
Linha 1	$M[1][0]$	$M[1][1]$	$M[1][2]$	$M[1][3]$
Linha 2	$M[2][0]$	$M[2][1]$	$M[2][2]$	$M[2][3]$

Uma matriz requer 2 subscritos para identificar um elemento particular.

Observe que os elementos de uma linha todos têm o primeiro subscrito iguais.

Observe que os elementos de uma coluna todos têm o segundo subscrito iguais.

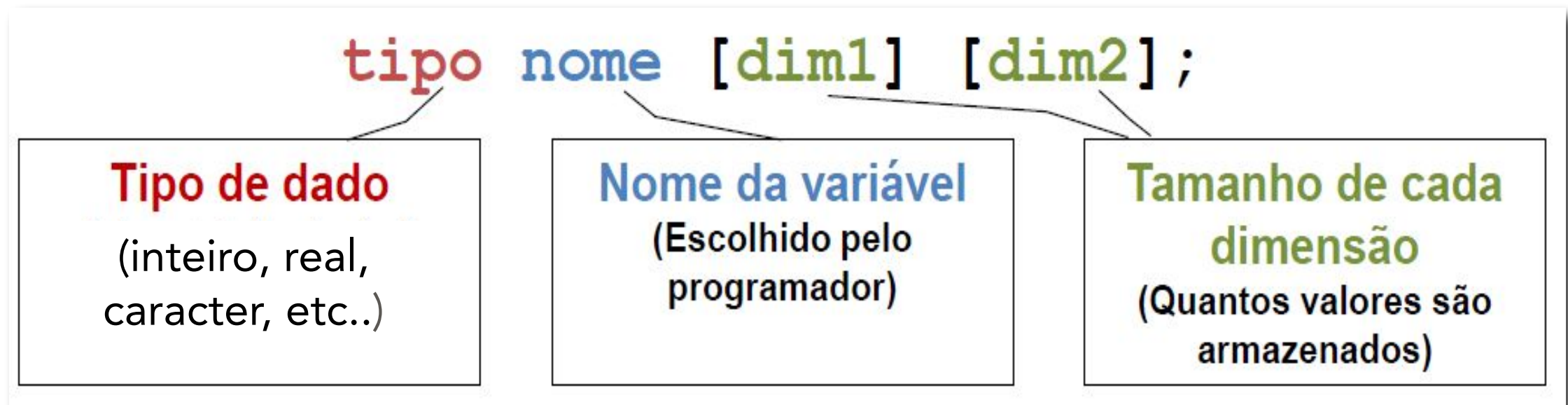
Índice ou subscrito da **Coluna**

Índice ou subscrito da **Linha**

nome da matriz

DECLARAÇÃO DE MATRIZES

- Forma geral



- Exemplos:
 - Info é uma matriz real de 10 elementos (100 elementos):
real **info** **[10]** **[10]**;
 - Cruzamentos é uma matriz de inteiros de 18x45 elementos:
inteiro **cruzamentos** **[18]** **[45]**;

INICIALIZAÇÃO DE MATRIZES

Na declaração:

```
inteiro m1[2][2] = {{1, 2}, {3, 4}}
```

Declara uma matriz quadrada 2 x 2 de inteiros e a inicializa da seguinte forma:

$$\rightarrow m1 = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

```
inteiro m2[2][2] = {{1}, {3, 4}};
```

Declara uma matriz quadrada 2 x 2 de inteiros e a inicializa da seguinte forma:

$$\rightarrow m2 = \begin{bmatrix} 1 & 0 \\ 3 & 4 \end{bmatrix}$$

```
real m3[2][3] = {{1.0, 2.0, 3.0}, {4.0, 5.0, 6.0}};
```

Declara uma matriz 2 x 3 de valores tipo float e a inicializa da seguinte forma:

$$\rightarrow m3 = \begin{bmatrix} 1.0 & 2.0 & 3.0 \\ 4.0 & 5.0 & 6.0 \end{bmatrix}$$

INICIALIZAÇÃO POR LEITURA

Por Leitura (linha)

–Considerando uma matriz de inteiros m (5x5)

```
1  Algoritmo "matriz"
2  INICIO
3      inteiro matriz[5][5], linha,coluna
4      para(linha=0;linha<5;linha++)faca
5          para(coluna=0;coluna<5;coluna++)faca
6              escreva("Digite o valor da linha",linha, "E da coluna", coluna)
7              leia(matriz[linha][coluna])
8          fimpara
9      fimpara
10 FINALGORITMO
```

INICIALIZAÇÃO POR LEITURA

Por Leitura (linha)

–Considerando uma matriz de inteiros m (5x5)

```
1  Algoritmo "matriz"
2  INICIO
3      inteiro matriz[5][5], linha,coluna
4      para(linha=0;linha<5;linha++)faca
5          para(coluna=0;coluna<5;coluna++)faca
6              escreva("Digite o valor da linha",linha, "E da coluna", coluna)
7              leia(matriz[linha][coluna])
8          fimpara
9      fimpara
10 FINALGORITMO
```

Neste caso, para cada valor do índice i de linha, o índice j de coluna varia em toda sua extensão.

INICIALIZAÇÃO POR LEITURA

Por Leitura (coluna)

–Considerando uma matriz de inteiros m (5x5)

```
1  Algoritmo "matriz"
2  INICIO
3      inteiro matriz[5][5], linha,coluna
4      para(coluna=0;coluna<5;coluna++)faca
5          para(linha=0;linha<5;linha++)faca
6              escreva("Digite o valor da linha",linha, "E da coluna",
7                  leia(matriz[linha][coluna])
8          fimpara
9      fimpara
10 FINALGORITMO
```


INICIALIZAÇÃO POR LEITURA

Por Leitura (coluna)

–Considerando uma matriz de inteiros m (5x5)

```
1  Algoritmo "matriz"
2  INICIO
3      inteiro matriz[5][5], linha,coluna
4      para(coluna=0;coluna<5;coluna++)faca
5          para(linha=0;linha<5;linha++)faca
6              escreva("Digite o valor da linha",linha, "E da coluna",
7                  leia(matriz[linha][coluna])
8          fimpara
9      fimpara
10 FINALGORITMO
```

Neste caso, para cada valor do índice j de coluna, o índice i de linha varia em toda sua extensão.

INICIALIZAÇÃO POR ATRIBUIÇÃO

- Considerando uma matriz de inteiros m (5x5)

```
1  Algoritmo "matriz"
2  INICIO
3      inteiro matriz[5][5], linha,coluna
4      para(linha=0;linha<5;linha++)faca
5          para(coluna=0;coluna<5;coluna++)faca
6              matriz[linha][coluna] = linha+1
7          fimpara
8      fimpara
9  FINALGORITMO
```

MATRIZES

- Matriz → $a[5][5]$

colunas	1	2	3	4	5
linha 1	5	9	6	4	3
linha 2	4	1	3	5	6
linha 3	3	1	4	5	6
linha 4	1	3	4	5	6
linha 5	1	3	4	5	6

APRESENTAÇÃO DE MATRIZES

```
1  Algoritmo "matriz"  
2  INICIO  
3      inteiro matriz[5][5], linha,coluna  
4      para(linha=0;linha<5;linha++)faca  
5          para(coluna=0;coluna<5;coluna++)faca  
6              escreva(matriz[linha][coluna])  
7          fimpara  
8      fimpara  
9  FIMALGORITMO
```



colunas	1	2	3	4	5
linha 1	5	9	6	4	3
linha 2	4	1	3	5	6
linha 3	3	1	4	5	6
linha 4	1	3	4	5	6
linha 5	1	3	4	5	6

Neste caso, para cada valor do índice i de linha, o índice j de coluna varia em toda sua extensão.

APRESENTAÇÃO DE MATRIZES

```
1  Algoritmo "matriz"  
2  INICIO  
3      inteiro matriz[5][5], linha,coluna  
4      para(coluna=0;coluna<5;coluna++)faca  
5          para(linha=0;linha<5;linha++)faca  
6              escreva(matriz[linha][coluna])  
7          fimpara  
8      fimpara  
9  FIMALGORITMO
```



colunas	1	2	3	4	5
linha 1	5	9	6	4	3
linha 2	4	1	3	5	6
linha 3	3	1	4	5	6
linha 4	1	3	4	5	6
linha 5	1	3	4	5	6

Neste caso, para cada valor do índice j de coluna, o índice i de linha varia em toda sua extensão.

EXEMPLOS DE APLICAÇÃO DE MATRIZES

- Dada uma matriz M (MAXLIN, MAXCOL), preenchê-la por leitura e imprimir:
 1. O **maior** elemento de cada linha da matriz;
 2. A **média** dos elementos de cada coluna;
 3. O **produto** de todos os elementos diferentes de zero;
 4. **Quantos elementos** são negativos;
 5. Posição ocupada (linha-coluna) por um elemento cujo valor será lido pelo programa.

1. PROCURANDO O MAIOR ELEMENTO DE CADA LINHA...

```
15     printf("encontrando o maior elemento");
16     for ( linha = 0; linha < size; linha++)
17     {
18         maior = 0;
19         for ( coluna = 0; coluna < size; coluna++)
20         {
21             if(matriz[linha][coluna]>maior){
22                 maior = matriz[linha][coluna];
23             }
24         }
25         printf("O maior elemento da linha %d é: %d\n",linha, maior);
26     }
27
```

Usa-se uma variável auxiliar “maior” para guardar o maior elemento, linha por linha

3. CALCULANDO O PRODUTO DOS VALORES != 0

...

```
produto = 1;
for ( linha = 0; linha < size; linha++)
{
    for ( coluna = 0; coluna < size; coluna++)
    {
        if(matriz[linha][coluna] != 0){
            produto = produto * matriz[linha][coluna];
        }
    }
}
printf("O produto e %d\n",produto);
```

Similar a soma, usa-se uma variável auxiliar “produto” para acumular a multiplicação dos elementos

4. CALCULANDO O TOTAL DE VALORES NEGATIVOS ...

```
}  
negativos = 0;  
for ( linha = 0; linha < size; linha++)  
{  
    for ( coluna = 0; coluna < size; coluna++)  
    {  
        if(matriz[linha][coluna] != 0){  
            negativos = negativos + matriz[linha][coluna];  
        }  
    }  
}  
printf("O total de negativos e %d\n",negativos);
```

Usa-se uma variável auxiliar “negativos” para **contar** os valores menores que zero

5. PROCURANDO A POSIÇÃO OCUPADA POR UM ELEMENTO CUJO VALOR SERÁ LIDO PELO PROGRAMA

```
15 printf("Digite o valor a ser procurado: ");
16 scanf("%d", &valor);
17 achou = 0;
18 for ( linha = 0; linha < size; linha++)
19 {
20     for ( coluna = 0; coluna < size; coluna++)
21     {
22         if(matriz[linha][coluna]== valor){
23             achou = 1;
24             printf("O valor está na linha %d e na coluna %d\n", linha, coluna);
25             linha=size;
26             coluna = size;
27         }
28     }
29 }
30 if(achou==0)
31     printf("Valor não encontrado");
32
33
```

Compara-se os elementos da matriz um a um com o valor digitado

ÁLGEBRA MATRICIAL: SOMA E MULTIPLICAÇÃO

- Seja o trecho de código que declara as matrizes A, B e C:

```
// Cria a matriz  
inteiro A[TAMANHO][TAMANHO], B[TAMANHO][TAMANHO], C[TAMANHO][TAMANHO]
```

- Como efetuar?
 - A Soma: $C = A + B$
 - A Multiplicação: $C = A \times B$

SOMANDO DUAS MATRIZES

```
16   for ( linha = 0; linha < size; linha++)  
17   {  
18       for ( coluna = 0; coluna < size; coluna++)  
19       {  
20           C[linha][coluna] = A[linha][coluna] + B[linha][coluna];  
21       }  
22   }
```

Exemplo:

$$\begin{matrix} & A & + & B & & = & C \\ \begin{bmatrix} 4 & -2 \\ 1 & 0 \end{bmatrix} & + & \begin{bmatrix} 3 & 2 \\ -1 & 5 \end{bmatrix} & = & \begin{bmatrix} 4+3 & -2+2 \\ 1-1 & 0+5 \end{bmatrix} & = & \begin{bmatrix} 7 & 0 \\ 0 & 5 \end{bmatrix} \end{matrix}$$

MULTIPLICANDO DUAS MATRIZES

$$C_{ij} = \sum_{k=0}^{n-1} A_{ik} * B_{kj}$$

$$\begin{bmatrix} 3 & 1 \\ 2 & -1 \\ 0 & 4 \end{bmatrix} \times \begin{bmatrix} 1 & -1 & 2 \\ 3 & 0 & 5 \end{bmatrix}$$

$$c_{11} = \sum_{k=1}^2 a_{1k} \times b_{k1}$$

$$c_{11} = a_{11} \times b_{11} + a_{12} \times b_{21}$$

$$c_{11} = 3 \times 1 + 1 \times 3 = 6$$

MULTIPLICANDO DUAS MATRIZES

$$C_{ij} = \sum_{k=0}^{n-1} A_{ik} * B_{kj}$$

$$\begin{bmatrix} 3 & 1 \\ 2 & -1 \\ 0 & 4 \end{bmatrix} \times \begin{bmatrix} 1 & -1 & 2 \\ 3 & 0 & 5 \end{bmatrix} =$$

$$\begin{bmatrix} 6 & -3 & 11 \\ -1 & -2 & -1 \\ 12 & 0 & 20 \end{bmatrix}$$

$$\begin{bmatrix} 3 \times 1 + 1 \times 3 & 3 \times (-1) + 1 \times 0 & 3 \times 2 + 1 \times 5 \\ 2 \times 1 + (-1) \times 3 & 2 \times (-1) + (-1) \times 0 & 2 \times 2 + (-1) \times 5 \\ 0 \times 1 + 4 \times 3 & 0 \times (-1) + 4 \times 0 & 0 \times 2 + 4 \times 5 \end{bmatrix}$$

MULTIPLICANDO DUAS MATRIZES

```
16     for ( linha = 0; linha < size; linha++)
17     {
18         for ( coluna = 0; coluna < size; coluna++)
19         {
20             soma = 0;
21             for(aux = 0; aux < size;aux++){
22                 soma += A[linha][aux] * B[aux][coluna];
23             }
24             C[linha][coluna] = soma;
25         }
```