

# Matrizes



- Vetores Bidimensionais -





# Matrizes

---

- Matrizes são utilizadas para o tratamento de conjuntos de dados com as mesmas características, assim como os vetores, porém possuem linhas e colunas (são bidimensionais).
- Este conjunto de dados recebe um nome comum, que é o nome da matriz. Na figura, o nome da matriz é  $A$

$$A = \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \end{pmatrix}$$



# Matrizes

- Cada **elemento da matriz** têm uma posição: **índice da linha** e o **índice da coluna**

$$A = \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \end{pmatrix}$$

Elemento da matriz

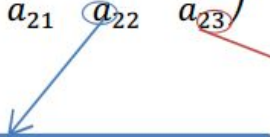
Índices de um elemento da matriz: linha e coluna



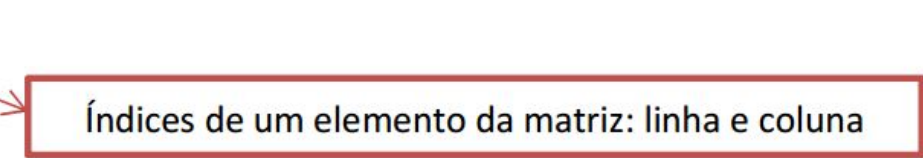
# Matrizes

- Os índices sempre iniciam em *zero*
  - Para uma matriz com L linhas e C colunas, os índices das linhas variam de  $[0, L-1]$  e os índices das colunas variam de  $[0, C-1]$
  - Por exemplo, para uma matriz A com 3 linhas e 4 colunas, como na figura, os índices das linhas variam de  $[0, 2]$  e os índices da coluna, de  $[0, 3]$
  - Um elemento de A é referenciado por  $a_{ij}$ , em que i indica o índice da linha e o j indica o índice da coluna.

$$A = \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \end{pmatrix}$$



Elemento da matriz



Índices de um elemento da matriz: linha e coluna



# Declaração de uma matriz

- A declaração de uma variável do tipo Matriz na linguagem C segue o formato:

```
tipo nome[quantidade_de_linhas][quantidade_de_colunas];
```

char  
int  
float  
double  
...

nome da variável

valor numérico que  
indica quantas linhas a  
matriz possuirá.

valor numérico que  
indica quantas linhas a  
matriz possuirá.



# Declaração de uma matriz

---

- A declaração de uma variável do tipo Matriz na linguagem C segue o formato:

```
tipo nome[quantidade_de_linhas][quantidade_de_colunas];
```

em que:

**tipo** é o tipo de dados dos elementos que serão armazenados;

**nome** é o nome da variável (nome da matriz)

**quantidade\_de\_linhas** é um valor numérico que indica quantas linhas estarão armazenadas na matriz, do tipo de dado especificado.

**quantidade\_de\_colunas** é um valor numérico que indica quantas colunas estarão armazenadas na matriz, do tipo de dado especificado.



# Declaração de uma matriz

---

- Exemplos de declaração de matrizes

```
int tabela[10][4];  
float chuva[12][31];  
double x[5][2];
```



# Elementos das Matrizes

---

- A matriz declarada como: `int A[3][4];` tem 3 linhas e 4 colunas.
- Os elementos são **referenciados** por índices (posições) de linha e coluna, por exemplo: `A[2][1] = 10`, indica que o elemento na 3ª linha e 2ª coluna recebe o valor 10.
- Assim, para acessar um elemento da matriz, usamos o nome da matriz e o valor dos índices da linha e da coluna do elemento, entre colchetes.

$$A = \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \end{pmatrix}$$





# Matrizes

---

É importante ter bem claro os conceitos de índice da matriz e valor de elemento da matriz. Os índices são um indicativo da posição do elemento (é a referencia da posição do elemento em determinada linha e coluna). Já o valor de um elemento da matriz é o conteúdo armazenado em determinada posição da matriz.



# Leitura de uma matriz

---

- Para fazer a leitura dos elementos de uma matriz, utiliza-se a função `scanf`, como para qualquer outra variável.
- Um elemento da matriz é lido de cada vez.
- Por isso, existem sempre dois laços de repetição: um para percorrer as linhas da matriz e outro para percorrer as colunas.
- Um elemento da matriz é referenciado com seus índices e na função `scanf` é utilizado `&` antes do elemento, como no exemplo a seguir:



# Leitura de uma matriz

---

```
int i, j;
int A[3][4];
for(i=0;i<3;i++) //laço que percorre as linhas da matriz
    for(j=0;j<4;j++){ //laço que percorre as colunas da matriz
        printf("Digite o elemento a[%d][%d]:", i, j);
        scanf("%d", &A[i][j]);
    }
```



# Exibição de uma matriz

---

- Assim, para listar todos os elementos da matriz na tela, seguinte código é válido:

```
int i, j;
int A[3][4];
for(i=0;i<3;i++) //laço de repetição que percorre as linhas da matriz
    for(j=0;j<4;j++) //laço de repetição que percorre as colunas da matriz
        printf("A[%d][%d] = %d\n", i, j, A[i][j]);
```



# Matrizes

---

## IMPORTANTE:

- na linguagem C não existe controle automático de final da matriz, assim como nos vetores, também não há este controle.
- O compilador não apontará como erros.
- Mas estes são **erros de programação**:
  - uma matriz declarada como 3x4 não possui os elementos de índices 5 e 7, por exemplo. Possui na linha, os índices de 0 a 2 e coluna os índices de 0 a 4.
- Este tipo de erro de programação tende a provocar o funcionamento incorreto do programa.



# Matrizes

- Vamos testar o exemplo que faz a leitura e a exibição dos elementos da matriz A

```
#include <stdio.h>

int main(){
    int A[3][4], i, j;

    printf("Digite os elementos da matriz A3x4\n");
    for(i = 0; i < 3; i++){
        for(j = 0; j < 4; j++){
            printf("A[%d][%d] = ", i, j);
            scanf("%d", &A[i][j]);
        }

        printf("\nMatriz A3x4\n");
        for(i = 0; i < 3; i++){
            for(j = 0; j < 4; j++){
                printf("%d\t", A[i][j]);
            }
            printf("\n");
        }

        return 0;
    }
```



# Matrizes

- Exemplo da execução do programa anterior:

Digite os elementos da matriz A3x4

A[0][0] = 1

A[0][1] = 2

A[0][2] = 3

A[0][3] = 4

A[1][0] = 2

A[1][1] = 3

A[1][2] = 4

A[1][3] = 5

A[2][0] = 3

A[2][1] = 4

A[2][2] = 5

A[2][3] = 6

Matriz A3x4

1	2	3	4
2	3	4	5
3	4	5	6



# Matrizes

---

- Vamos resolver os seguintes exercícios:
  - a) Ler uma matriz  $2 \times 3$ , multiplicar seus elementos por um número informado pelo usuário e mostrar a matriz modificada.
  - b) Gerar uma matriz  $10 \times 10$  e mostrar na tela. Após, solicitar ao usuário o número de uma linha da matriz e mostrar a respectiva linha.
  - c) Gerar uma matriz  $4 \times 4$  e mostrar sua diagonal principal.