

Matrizes

Definição

↳ São estruturas de dados que armazenam elementos, organizados em linhas e colunas.

* **vetor:** unidimensional

- * Matriz: multidimensional (no mínimo 2)

Sintaxe de declaração:

* tipo nome [linhas] [colunas]

* declaração com inicialização :

int mat [3] [3] = { {1, 2, 3}, {4, 5, 6}, {7, 8, 9} };

\downarrow \downarrow \downarrow
 linha 0 linha 1 linha 2

Sintaxe de acesso a posição

matriz [linha] [coluna]

* Posição por posição

nome [índice] [índice] Ex.: mat [0] [0] = 1;
posição mat [1] [1] = 5;

* Estrutura de repetição: imprimindo a matriz toda

11 percorre as linhas

```
for (linha = 0; linha < 3; linha++) {
```

11 percorre as colunas

```
for (coluna = 0; coluna < 3; coluna++) {
```

```
printf("%d", mat[i][j]);
```

}

}

0
1
2
...

1	2	3
4	5	6
7	8	9

→ quando o j é incrementado,
 $= j$ inicializa tudo de novo, incrementan-
do o i (linha)

*** Obs:** $i \in \mathbb{Z}$ } eles n° estão de acordo com
 $j \in \mathbb{Z}$ } o tamanho da tabela

* Diagonal principal de uma matriz quadrada!

↳ É formada pelos elementos que tem índice de linha e coluna iguais

```
int main() {
    int matriz[3][3] = {{1,2,3}, {4,5,6}, {7,8,9}};
    printf("Diagonal principal: \n");
    for(i=0; i<3; i++){
        for(j=0; j<3; j++){
            if(i==j)
                printf("%i\t", matriz[i][j]);
        }
    }
    return 0;
}
```

Perceber % os elementos da diagonal principal tem índice de linha e coluna iguais

Diagrama de uma matriz 4x4 com a diagonal principal destacada em amarelo. As linhas são numeradas 0 a 3 e as colunas também 0 a 3. Os elementos da diagonal principal são 45, 14, 65 e 55.

	0	1	2	3
0	45	32	83	95
1	26	14	37	42
2	40	43	65	77
3	74	79	48	55

DIAGONAL principal

* Diagonal secundária de uma matriz quadrada

```
int main() {
    int matriz[3][3] = {{1,2,3}, {4,5,6}, {7,8,9}};
    printf("Diagonal principal: \n");
    for(i=0; i<3; i++){
        for(j=0; j<3; j++){
            if(i+j==3-1)
                printf("%i\t", matriz[i][j]);
        }
    }
    return 0;
}
```

Diagrama de uma matriz 4x4 com a diagonal secundária destacada em amarelo. As linhas são numeradas 0 a 3 e as colunas também 0 a 3. Os elementos da diagonal secundária são 95, 37, 43 e 74.

	0	1	2	3
0	45	32	83	95
1	26	14	37	42
2	40	43	65	77
3	74	79	48	55

DIAGONAL secundária

Usuario digita a matriz

```
#include <stdio.h>
#include <stdlib.h>

int main() {
    int linha, coluna, mat[3][3]; // mat[linha][coluna]
    // primeiro for é pl linha
    for(linha = 0; linha < 3; linha++){
        // segundo for é coluna
        for(coluna = 0; coluna < 3; coluna++){
            printf("Digite o elemtno %d %d ", linha, coluna);
            scanf("%d", &mat[linha][coluna]);
        }
    }

    for(linha = 0; linha < 3; linha++){
        for(coluna = 0; coluna < 3; coluna++){
            printf("%d ", mat[linha][coluna]);
        }
        printf("\n");
    }

    return 0;
}
```

aparece a posição

imprime a matriz

Como alterar o conteúdo de uma matriz

↳ Alteração na matriz

↳ Atribuição:

↳ $\text{matriz}[\text{linha}][\text{coluna}] = \text{matriz}[\text{linha}][\text{coluna}] * 2$

↳ dobra os números da matriz

↳ atribuição dentro da estrutura de repetição

Alocação de matriz dinâmica

↳ É um vetor de vetores

↳ O que é?

Reservar um espaço na memória para armazenar uma matriz cujas dimensões podem ser determinadas em tempo de execução.

* É útil quando as dimensões não são conhecidas ou podem variar durante a execução do programa

↳ Criar vetor de ponteiro

↳ Cada posição desse vetor acessa a posição de outro vetor

↳ obs: Utiliza-se dois "*" pois é um vetor de vetor (ponteiro p/ ponteiro)

**matriz

