

# Estrutura de Dados

## Matrizes

---

Anderson Veiga da Silva  
anderson.silva@iff.edu.br

# Matrizes

Matrizes são estruturas de dados multidimensionais, que podem armazenar um conjunto de valores de um mesmo tipo.

Uma matriz de duas dimensões é formada por linhas e colunas

Exemplo de uma matriz de inteiros:

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15

# Índices de uma matriz

Para acessar cada célula da matriz é preciso informar a posição da linha e da coluna referente ao valor. Da mesma forma que ocorre com os vetores, o primeiro índice de cada linha ou de cada coluna é representado pelo número 0.

Exemplo de uma matriz com seus respectivos índices, onde o primeiro valor representa a linha e o segundo valor a coluna:

	coluna 0	coluna 1	coluna 2	coluna 3	coluna 4
linha 0	00	01	02	03	04
linha 1	10	11	12	13	14
linha 2	20	21	22	23	24
linha 3	30	31	32	33	34

# Criando uma matriz

Na linguagem C++, a criação de uma matriz segue a seguinte sintaxe:

```
tipo nome[linhas][colunas];
```

Onde:

- tipo: tipo de informação que será armazenada na matriz.
- nome: nome pelo qual a matriz será referenciada.
- linhas: quantidade de linhas que a matriz terá.
- colunas: quantidade de colunas que a matriz terá.

# Exemplos

Criando uma matriz de inteiros de 10 linhas e 5 colunas:

```
int numeros[10][5];
```

Criando uma matriz de inteiros de 2 linhas e 2 colunas com valores:

```
int numeros[][] = {{1, 2}, {3, 5}};
```

	0	1
0	1	2
1	3	5

# Inserindo valores de uma matriz

Para inserir um valor em uma posição da matriz, é preciso informar o índice da linha e o índice da coluna onde o valor será armazenado.

Exemplo de inserção de um valor na linha 0 e na coluna 2 da matriz:

```
int numeros[3][3];  
numeros[0][2] = 30;
```

	0	1	2
0			30
1			
2			

# Acessando valores de uma matriz

Para acessar o valor, basta informar os índices da linha e da coluna da matriz.

Exemplo:

```
int[2][3] numeros = {{1, 4, 8} {5, 6, 3}};  
cout << numeros[1][0];
```

Este código irá imprimir o valor **5**, que está na segunda linha e na primeira coluna da matriz.

# Percorrendo uma matriz

Para percorrer uma matriz, é preciso usar a estrutura de repetição **for** de forma aninhada, ou seja, um dentro do outro.

Exemplo de impressão de todos os valores de uma matriz:

```
int numeros[3][5];
for (int l = 0; l < 3; l++) {
    for (int c = 0; c < 5; c++) {
        cout << numeros[l][c] << endl;
    }
}
```



# Exemplo

Preencher uma matriz 9x9 e imprimir a soma dos valores inseridos:

```
#include <iostream>
using namespace std;
int main() {
    int matriz[3][3];
    for (int l = 0; l < 3; l++) {
        for (int c = 0; c < 3; c++) {
            cout << "informe o número: ";
            cin >> matriz[l][c];
        }
    }
    int soma = 0;
    for (int l = 0; l < 3; l++) {
        for (int c = 0; c < 3; c++) {
            soma += matriz[l][c];
        }
    }
    cout << soma;
}
```

# Referências

ASCENCIO, A. F. G., CAMPOS, E. V. **Fundamentos da Programação de Computadores**. 2ª Ed. São Paulo: Pearson/Prentice Hall, 2008.

FORBELLONE, A. L. V., EBERSPACHER, H. F. **Lógica de Programação**: a Construção de Algoritmos e Estruturas de Dados. São Paulo: Makron Books, 2000.