



#Reskilling4Employment  
Software Developer

# Programação Estruturada

Vetores e Matrizes

Vitor Santos



# Conteúdo

- Vetores
- Matrizes

# Variáveis e Vetores

Podemos imaginar uma **variável** como sendo uma gaveta onde guardamos algo.

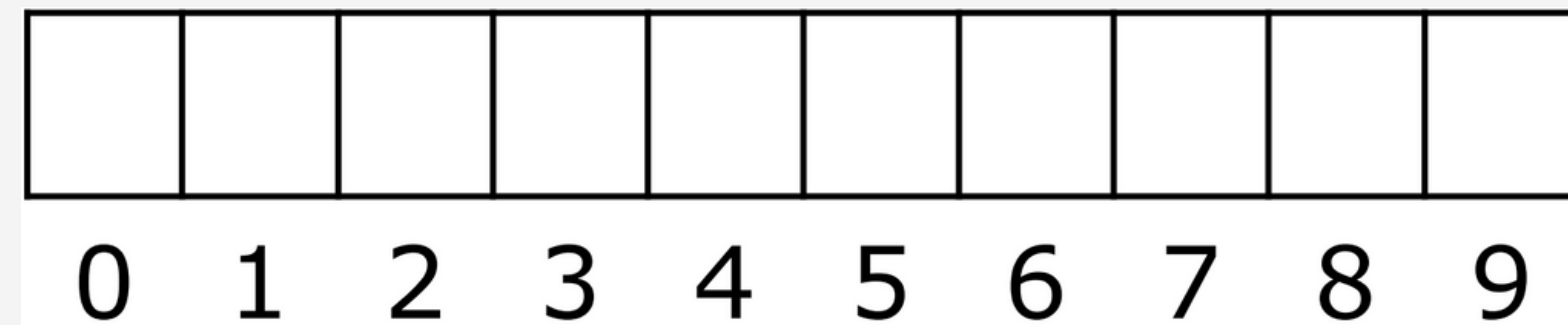


Um **vetor** (ou array) pode ser visto como uma fila de gavetas.



# Vetores

- Um **vetor**, ou **array**, é um conjunto de dados de um mesmo tipo (**homogêneos**), que são armazenados de forma **contígua**, e a que é possível aceder individualmente através de um **índice**.



- Exemplos de utilização para armazenamento:
  - Temperaturas médias de cada mês de um ano;
  - Comissões mensais de um vendedor;
  - Notas de um formando a um módulo;
  - Movimentos de uma conta bancária.

# Vetores - Declaração

- A declaração de um vetor é feita da seguinte forma:

```
tipo_dados[ ] nome_vetor = new tipo_dados[nº de elementos];
```

- **tipo\_dados**: tipo de dados dos elementos do vetor (int,double, String, boolean, etc...)
- **nome\_vetor**: nome a dar ao vetor;
- **nº de elementos**: números de elementos que o vetor irá conter (tem que ser um valor inteiro e positivo).
- Exemplo:
  - `double[ ] comissões = new double[12];`
  - `int[ ] array = new int[10];`
  - `int[ ] notas = new int[2];`

# Vetores - Manipulação

- A forma de acedermos a um determinado elemento do vetor é: `nome_vetor[índice]`;
- **nome\_vetor**: nome que foi dado ao vetor;
- **índice**: posição do elemento do vetor;
- **IMPORTANTE**: Os índices de um vetor com n elementos variam entre 0 e n-1.

- Exemplo:

```
int[ ] vetor = new int[10];
```

```
vetor[4]=2;
```

```
vetor[7]=5-1;
```

```
vetor[0]=vetor[4]+vetor[7];
```

```
vetor[vetor[4]]=9;
```

```
vetor[10]=1; ← IMPOSSÍVEL
```

<b>6</b>		<b>9</b>		<b>2</b>			<b>4</b>		
0	1	2	3	4	5	6	7	8	9

# Vetores e Matrizes

Se um **vetor** pode ser visto como uma fila de gavetas...



... então uma **matriz** pode ser vista como várias filas de gavetas.



# Matrizes

- Uma **matriz** não é mais que um vetor **bidimensional**.
- Podemos pensar numa matriz como uma **tabela**.
- Essa tabela é um conjunto de dados de um mesmo tipo (**homogêneos**), distribuídos por **colunas e linhas**, e a que é possível aceder individualmente através de um **par de índices**.

	0	1	2	3	4
0					
1					
2					
3					
4					



# Matrizes - Declaração

- A declaração de uma matriz é feita da seguinte forma:

```
tipo_dados[ ][ ] nome_matriz = new tipo_dados[nº de linhas][nº de colunas];
```

- **tipo\_dados**: tipo de dados dos elementos do vetor (int, float, char, etc)
- **nome\_matriz**: nome a dar à matriz;
- **nº de linhas e nº de colunas**: números de linhas e colunas que a matriz irá conter (tem que ser um valor inteiro e positivo).
- **Exemplo**:
  - `int[ ][ ] matriz = new int[3][3];`
  - `int[ ][ ] galo = new int[3][3];`
  - `double[ ][ ] tabela = new double[4][8];`

# Matrizes - Manipulação

- A forma de acedermos a um determinado elemento da matriz é:  
`nome_matriz[índice_linha][índice_coluna];`
- `nome_matriz`: nome que foi dado à matriz;
- `[índice_linha][índice_coluna]`: posição do elemento da matriz;
- **IMPORTANTE**: Os índices de uma matriz com  $n$  elementos variam entre 0 e  $n-1$ .

- Exemplo:

```
int[ ][ ] matriz = new int[5][5];  
matriz[4][3]=2;  
matriz[1][2]=4+1;  
matriz[3][0]=matriz[4][3]+matriz[1][2];  
matriz[matriz[4][3]][4]=6;  
matriz[2][5]=1;←IMPOSSÍVEL
```

	0	1	2	3	4
0					
1			5		
2					6
3	7				
4				2	

# Método Length

- Podemos saber o tamanho de um vetor através do método `.length( )`

```
int[ ] vetor = new int[15];  
int tamanho_vetor = vetor.length( );
```

Neste caso a variável “tamanho\_vetor” assume o valor 15

# Método Length

- Podemos saber o tamanho de uma matriz através do método `.length( )`

```
int[ ][ ] matriz = new matriz[5][3];  
int linhas_matriz = matriz.length( );  
int colunas_matriz = matriz[0].length( );
```

Neste caso a variável “linhas\_matriz” é 5  
e a variável “colunas\_matriz” é 3



#Reskilling4Employment  
Software Developer

# Programação Estruturada

Vetores e Matrizes

Vitor Santos

