

# **Matrizes**

# **Unidimensionais**

# **\*Vetores\***

## **Algoritmos e Programação II**

Profa: Daniela Scherer dos Santos

[daniela.santos37@ulbra.edu.br](mailto:daniela.santos37@ulbra.edu.br)

[www.danielascherer.com.br](http://www.danielascherer.com.br)

Estruturas de dados homogêneas utilizadas para organização dos dados.

Estruturas de dados **homogêneas** utilizadas para organização dos dados.

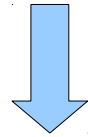


São capazes de armazenar um conjunto de dados do mesmo tipo (ou int, ou double, ou string, etc)

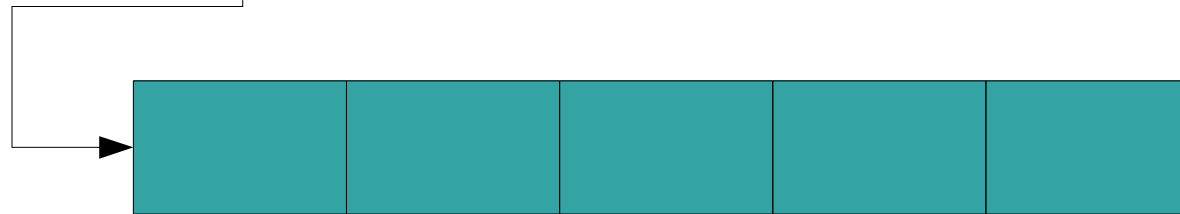
Enquanto uma variável comum (dos tipos int, double, etc.) é capaz de armazenar apenas um valor, uma variável definida como matriz pode armazenar a quantidade de valores que definirmos como capacidade desta matriz.

# Matriz Unidimensional → Vetores

São matrizes de uma dimensão



Área de armazenamento dos dados



0

1

2

3

4

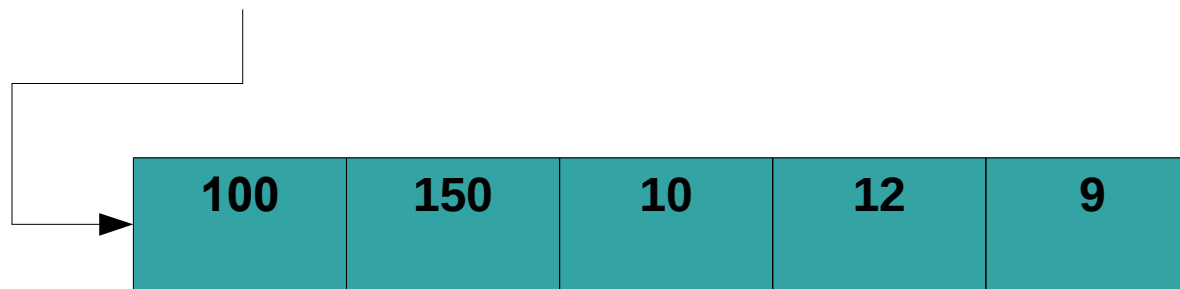
Posição / índice

Representação gráfica

# Matriz Unidimensional → Vetores

Exemplo: Vetor de números inteiros contendo suas 5 posições ocupadas.

Área de armazenamento dos dados



0 1 2 3 4

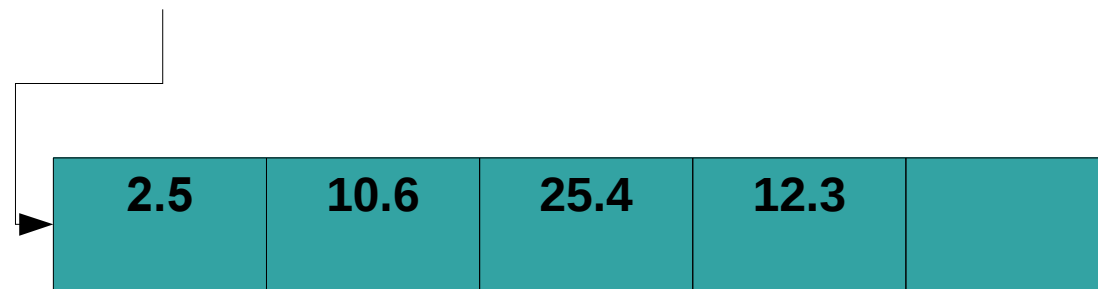
Posição / índice

Representação gráfica

# Matriz Unidimensional → Vetores

Exemplo: Vetor de números do tipo double contendo 4 de suas 5 posições ocupadas.

Área de armazenamento dos dados



0 1 2 3 4

Posição / índice

Representação gráfica

# Matriz Unidimensional → Vetores

Exemplo: Vetor de string contendo 3 de suas 5 posições ocupadas.

Área de armazenamento dos dados



0

1

2

3

4

Posição / índice

Representação gráfica

# Matriz Unidimensional → Vetores

Como declarar e criar uma variável como vetor?

int  
double  
string  
etc

([10],[20],[100]...) de  
acordo com a  
quantidade de dados  
que deseja armazenar

```
tipo[ ] nomeDaVariável = new tipo[tamanho];
```

Respeitando as regras para criação  
de variáveis



# Matriz Unidimensional → Vetores

## Exemplos de declarações de vetores:

```
double[ ] medias = new double[20];
```

Declara um vetor chamado “*medias*”, onde cada elemento do vetor será do tipo double e o vetor terá capacidade para armazenar 20 valores;

```
int[ ] vetor = new int[100];
```

Declara um vetor chamado “*vetor*”, onde cada elemento será do tipo int e o vetor terá capacidade para armazenar até 100 valores;

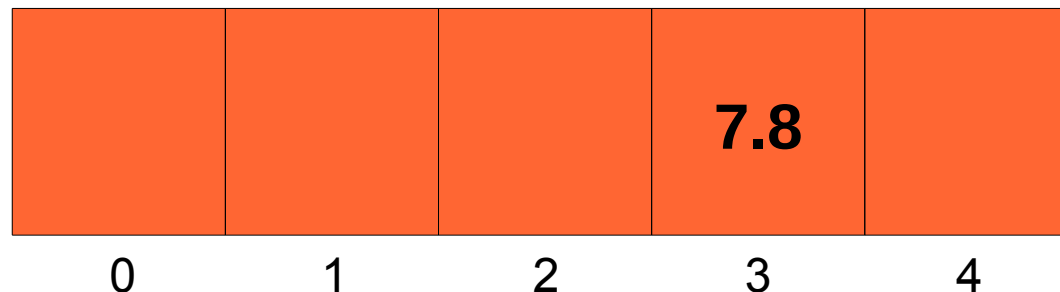
```
string[ ] palavras = new string[20];
```

Declara um vetor chamado “*palavra*” onde cada elemento deste vetor será do tipo string e o vetor terá capacidade para armazenar 20 palavras.

# Exemplos

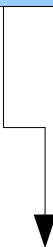
```
double[ ] notas = new double[5];  
notas[3] = 7.8;
```

1. Declara um vetor chamado "notas" onde cada elemento deste vetor será do tipo double e o vetor terá capacidade para armazenar até 5 valores.
2. Atribui o valor 7.8 para a posição 3 do vetor notas;



# Exemplos

```
string[ ] diasSemana = {"segunda", "terça", "quarta", "quinta",  
                        "sexta", "sábado", "domingo"};
```



Declara um vetor chamado “*diasSemana*” onde cada elemento deste vetor será do tipo string. O vetor foi **inicializado** com os valores entre as chaves.

segunda	terça	quarta	quinta	sexta	sábado	domingo
0	1	2	3	4	5	6

# Exemplos

```
int[ ] nums = {10, 20, 30, 40, 50, 60};
```



Declara um vetor chamado “nums” onde cada elemento deste vetor será do tipo int. O tamanho do vetor não foi definido explicitamente, desta forma o compilador contará a quantidade de itens da lista de valores e o fixará como o tamanho do vetor. O vetor nums foi inicializado com os valores entre as chaves.

10	20	30	40	50	60
0	1	2	3	4	5

# Exemplo:

Programa para preencher um vetor com 10 valores do tipo inteiro digitados pelo usuário

```
1.static void Main(string[] args)
2.{
3.    const int t = 10;
4.    //declara um vetor de inteiros com 10 posições
5.    int[] vetor = new int[t];
6.    int x;
7.    Console.WriteLine("Digite os 10 valores para preencher o vetor:");
8.
9.    //leitura dos valores
10.    for (x = 0; x < t; x++)
11.        vetor[x] = Convert.ToInt16(Console.ReadLine());
12.
13.    //escrita dos valores
14.    for (x = 0; x < t; x++)
15.        Console.WriteLine("posição " + x + " valor " + vetor[x]);
16.    Console.ReadKey();
17.}
```

# Exemplo (string):

```
1.static void Main(string[] args)
2.{
3.    string nome;
4.    Console.Write("Digite seu nome: ");
5.    nome = Console.ReadLine();
6.    //verifica a letra que está na terceira posição
7.    for(int x=0;x<nome.Length;x++)
8.    {
9.        if (x == 2)
10.            Console.WriteLine("A terceira letra da string digitada é: " + nome[x]);
11.    }
12.    Console.ReadKey();
13.}
```