



# Entendendo Vetores e Matrizes

## **D**EFINIÇÃO, DECLARAÇÃO E ATRIBUIÇÃO

Vetor é considerado uma estrutura de dados homogênea pois é capaz de armazenar vários dados, desde que todos sejam do mesmo tipo. Ou seja, declarando um único vetor com um nome, é possível manipular uma lista com vários dados de um mesmo tipo.

Um vetor, enquanto estrutura de dados homogênea, é quase considerado como outra variável, pois pode, inclusive, ser manipulado dentro de outras estruturas. Por exemplo, a de decisão - simples, composta ou aninhada - ou dentro de uma estrutura de múltipla escolha, mesmo nas de repetição.

O vetor é uma matriz unidimensional, tem este nome porque é uma matriz com uma única linha. Na hora de declarar um vetor, você precisa saber o tipo do dado que será armazenado e a quantidade de elementos. Além disso, também é necessário definir um nome para o vetor. Por exemplo:

Declarar

```
notas[5] real;
```

Nesse exemplo, foi declarado um vetor chamado "notas" com 5 elementos do tipo real. Para atribuir valores no vetor, você faz do mesmo modo como se fosse qualquer outra variável. É preciso indicar a posição:

```
notas[0] <- 9,5; // a posição zero (inicial) do vetor recebe 9,5
```

```
notas[1] <- 8,0;
```

```
notas[2] <- 9,0;
```

```
notas[3] <- 8,5;
```


```
notas[4] <- 9,5;
```

Assim como o Vetor, a Matriz é considerada uma estrutura de dados homogênea, possuindo as mesmas especificidades, porém, é composta por linhas e colunas. Entretanto, uma matriz unidimensional é chamada de vetor porque possui uma única linha.

Na hora de se declarar uma matriz, você precisa saber o tipo do dado que será armazenado e a quantidade de elementos da linha e da coluna. Você precisa também definir um nome para a matriz. Por exemplo:

Declarar

```
notas[3][2] real;
```

Nesse exemplo, foi declarada uma matriz chamada notas com  $3 \times 2 = 6$  elementos do tipo real. Para atribuir valores na matriz, você faz do mesmo modo como se fô  para qualquer outra variável, apenas precisa indicar a posição da linha e da coluna.

```
notas[0][0] <- 9,5;
```

```
notas[0][1] <- 8,0;
```

```
notas[1][0] <- 9,0;
```

```
notas[1][1] <- 8,5;
```

```
notas[2][0] <- 9,5;
```

```
notas[2][1] <- 9,0;
```

## ALGORITMOS COM VETORES E MATRIZ

Para exemplificar algoritmos com vetores, vamos mostrar um algoritmo que apresenta os valores que estão armazenados no vetor.

**Algoritmo** MostrarVetor

**início\_algoritmo**

**Declarar**

Vet [25], i **inteiro**;



**para** i **de** 0 **até** 24 **passo** + 1 **faça**

escrever (“Digite um valor inteiro”);

ler (Vet[i]);

**fimpara**;

escrever (Vet[i]);

**fim\_algoritmo.**

Quando os valores de um vetor são apresentados, você precisa saber qual o tipo de dado está armazenado nele e quantos elementos têm nesse vetor. Os dados são apresentados para cada posição do vetor e a forma em que os elementos são mostrados para o usuário são iguais quando você apresenta o valor de uma variável.

Para exemplificar algoritmos com matriz, vamos mostrar um algoritmo que apresenta os valores que estão armazenados na matriz.

**Algoritmo** Mostrar

**início\_algoritmo**

**Declarar**

Mat [5][5], i, j **inteiro**;



**para i de 0 até 4 passo +1 faça**

**para j de 0 até 4 passo + 1 faça**

escrever (“Digite um valor inteiro”);

ler (Mat[i][j]);

escrever (Mat[i][j]);


**fimpara;**

**fimpara;**

**fim\_algoritmo.**

Quando os valores de uma matriz são apresentados, você precisa saber o tipo de dado que está armazenado e também quantos elementos têm nela. Os dados são apresentados para cada posição, linha e coluna, e a forma em que os elementos são mostrados para o usuário são iguais quando você apresenta o valor de uma variável.

## **APLICANDO COM VETORES E MATRIZES**

Vamos exemplificar por meio de uma tabela com a média das notas de estudantes de uma instituição educacional que possui quatro turmas, A, B, C e D, para  disciplinas de Matemática, Português, História e Geografia.

Para construir a tabela, vamos colocar o nome das disciplinas nas colunas e das turmas em cada linha. Vamos chamar a tabela, ou matriz, por Notas, como podemos observar a seguir:

| Notas       | Matemática<br>[0] | Português<br>[1] | História<br>[2] | Geografia<br>[3] |
|-------------|-------------------|------------------|-----------------|------------------|
| Turma A [0] | 8                 | 9                | 8               | 9                |
| Turma B [1] | 7                 | 5                | 6               | 6                |
| Turma C [2] | 8                 | 7                | 7               | 7                |
| Turma D [3] | 7                 | 8                | 8               | 9                |

Desta forma, temos, por exemplo:

`Notas[0][0] = 8;` // linha 1, coluna 1

`Notas[0][2] = 8;` // linha 1, coluna 3

`Notas[1][1] = 5;` // linha 2, coluna 2

`Notas[1][3] = 6;`

`Notas[2][2] = 7;`

`Notas[3][1] = 8.`



A matriz Notas possui 4 linhas e 4 colunas, somando um total de  $4 \times 4$  elementos, que são as notas.

As matrizes são vastamente utilizadas, inclusive em disciplinas como a Matemática Aplicada à Computação. Também podemos desenvolver algoritmos e programas para diversos campos do conhecimento como Economia, Engenharias, a própria Matemática com a Álgebra e a Geometria, além de diversas aplicações em Física, Computação, entre outros.

## **EXERCITANDO COM VETORES E MATRIZES**

Desenvolva um algoritmo que receba 100 valores numéricos inteiros e mostre a soma desses 100 números.

Quando a gente resolve exercícios que são para o desenvolvimento de algoritmos, precisamos ficar atentos a todos os detalhes do enunciado, bem como à entrada, ao processamento e à saída.

A entrada neste algoritmo são 100 números inteiros.

O processamento neste algoritmo é somar os 100 números.

A saída neste algoritmo é apresentar a soma dos 100 números.



Algoritmo Somar //Nome do algoritmo é somar

início\_algoritmo

Declarar

//declarando o vetor VetSoma de tamanho 100 do tipo inteiro

// também declarou as variáveis i e soma, soma já começando com 0

VetSoma [100], i, soma  $\leftarrow$  0 inteiro;

// usando estrutura de repetição para passar pelas 100 posições do vetor.

// as posições são do 0 até 99

// dentro da estrutura de repetição, mostra mensagem para o usuário

// e atribui no vetor

// também já vai acumulando a soma dos valores do vetor na variável soma

para i de 0 até 99 passo + 1 faça

escrever (“Digite um valor inteiro”);

ler (VetSoma[i]);

soma  $\leftarrow$  soma + VetSoma[i];

fimpara;

// apresentando o valor da soma para o usuário



escrever ("A soma dos 100 valores digitados é: " + soma);



fim\_algoritmo.

## **Algoritmo Somar**

**início\_algoritmo**

**Declarar**

VetSoma [100], i, soma  $\leftarrow$  0 **inteiro**;

**para i de 0 até 99 passo + 1 faça**

escrever ("Digite um valor inteiro");

ler (VetSoma[i]);

soma  $\leftarrow$  soma + VetSoma[i];

**fimpara;**

escrever ("A soma dos 100 valores digitados é: " + soma);

**fim\_algoritmo.**



Desenvolva um algoritmo que receba 120 valores numéricos inteiros, numa matriz 10 x 12 e mostre a soma destes 120 números.

A entrada neste algoritmo são 120 números inteiros.

O processamento neste algoritmo é somar os 120 números.

A saída neste algoritmo é apresentar a soma dos 120 números.

Algoritmo Somar //Nome do algoritmo é somar

início\_algoritmo

Declarar

//declarando a matriz MatSoma de tamanho 10x12 do tipo inteiro

// também declarou as variáveis a, b e soma, soma já começando com 0

MatSoma [10][12], i, soma  $\leftarrow$  0 inteiro;

// usando estruturas de repetição, passando pelas 120 posições da matriz.

// as posições são do 0 até 9 na linha e de 0 a 11 na coluna

// dentro das estruturas de repetição, mostra mensagem para o usuário

// e atribui na matriz

// já vai acumulando a soma dos valores da matriz na variável soma



**para a de 0 até 9 passo +1 faça**

**para b de 0 até 11 passo + 1 faça**

escrever (“Digite um valor inteiro”);

ler (MatSoma[i]);

soma  $\leftarrow$  soma + MatSoma[i];

**fimpara;**

**fimpara;**

// apresentando o valor da soma para o usuário

escrever ("A soma dos 120 valores digitados é: " + soma);

fim\_algoritmo.

## **Algoritmo Somar**

**início\_algoritmo**

**Declarar**

MatSoma [10][12], a, b, soma  $\leftarrow$  0 inteiro;

**para a de 0 até 9 passo +1 faça**

**para b de 0 até 11 passo + 1 faça**



escrever ("Digite um valor inteiro");

ler (MatSoma[a][b]);

soma  $\leftarrow$  soma + MatSoma[a][b];

**fimpara;**

**fimpara;**

escrever ("A soma dos 120 valores digitados é: " + soma);

**fim\_algoritmo.**

## IMPLEMENTANDO VETORES E MATRIZES EM JAVA

A linguagem de programação que estamos utilizando é a linguagem Java.

Lembre-se que você deve deixar seu programa bem alinhado e indentado (alinhado graficamente) para que posteriormente, você consiga entender e dar suporte para ele.

A documentação é uma parte importante no desenvolvimento de um programa. Faça sempre comentários no seu código.

**class** Somar {



**public static void** main(String args [ ]) {

//declarando o vetor VetSoma de tamanho 100 do tipo inteiro

// também declarou as variáveis i e soma, soma já começando com 0

**int** VetSoma [ ] , i , soma = 0;

// construindo o vetor no java

VetSoma = **new int** [100];

// usando estrutura de repetição para passar pelas 100 posições do vetor.

// as posições são do 0 até 99

// dentro da estrutura de repetição, mostra mensagem para o usuário

//e atribui no vetor

// também já vai acumulando a soma dos valores do vetor na variável soma

**for** ( i = 0 ; i <= 99 ; i++ ) {

VetSoma[i] = Integer.parseInt(JOptionPane.showInputDialog (“Digite um valor inteiro”));

soma = soma + VetSoma[i];

} // for

// apresentando o valor da soma para o usuário

```
System.out.println ("A soma dos 100 valores digitados é: "+ soma);
```

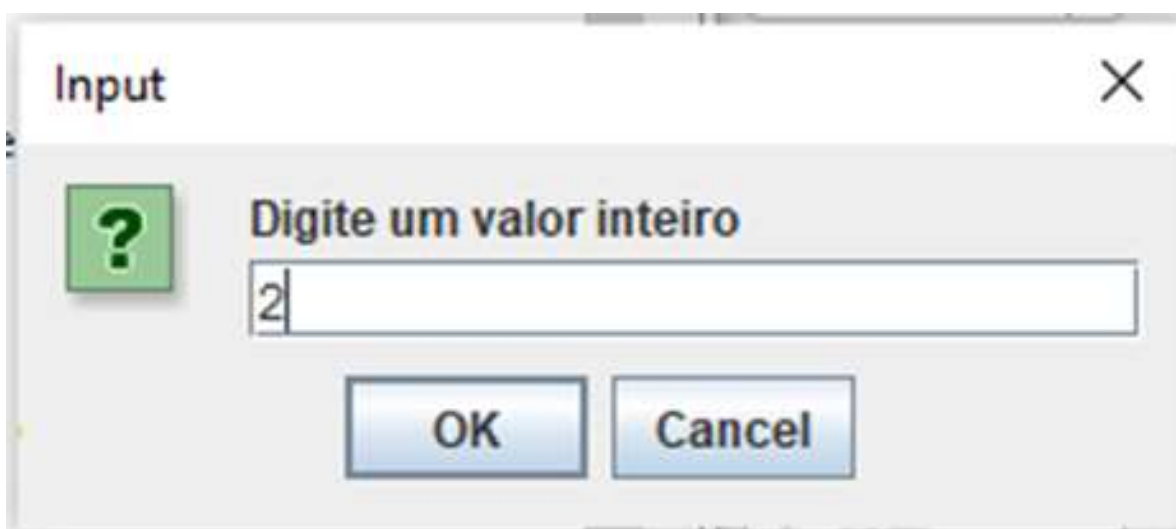


```
System.exit(0);
```

```
} // void main
```

```
} // classe Somar
```

```
1  import javax.swing.*;
2  class Somar {
3      public static void main(String args [ ]) {
4          int VetSoma [ ] , i , soma = 0;
5          VetSoma = new int [100];
6
7          for ( i = 0 ; i <= 99 ; i++ ) {
8              VetSoma[i] = Integer.parseInt(JOptionPane.showInputDialog
9              ("Digite um valor inteiro"));
10             soma = soma + VetSoma[i];
11         }
12         System.out.println ("A soma dos 100 valores digitados é: "+
13         soma);
14         System.exit(0);
15     }
16 }
```



Output - Running Single Java File X



A soma dos 100 valores digitados é: 45

A linguagem de programação que estamos utilizando é a linguagem Java.

Lembre-se que você deve deixar seu programa bem alinhado e indentado para que posteriormente, você consiga entender e dar suporte para o programa.

A documentação é uma parte importante quando você desenvolve um programa. Faça sempre comentários no seu código.

```
class Somar {
```

```
    public static void main(String args [ ]) {
```

```
        //declarando a matriz MatSoma de tamanho 10x12 do tipo inteiro
```

```
        // também declarou as variáveis a, b e soma, soma já começando com 0
```

```
        int MatSoma [ ][ ] , i , soma = 0;
```

```
        // construindo a matriz no java
```

```
        MatSoma = new int [10][12];
```

// usando estruturas de repetição para passar pelas 120 posições da matriz.



// as posições são do 0 até 9 na linha e de 0 a 11 na coluna

// dentro das estruturas de repetição, mostra mensagem para o usuário

// e atribui na matriz

// já vai acumulando a soma dos valores da matriz na variável soma

**for (a = 0 ; a <= 9 ; a++){**

**for ( b = 0 ; b <= 11 ; b++ ) {**

MatSoma[a][b] = Integer.parseInt(JOptionPane.showInputDialog ("Digite um valor inteiro"));

soma = soma + MatSoma[a][b];

**} // for**

**} // for**

// apresentando o valor da soma para o usuário

System.out.println ("A soma dos 120 valores digitados é: "+ soma);

System.exit(0);

**} // void main**

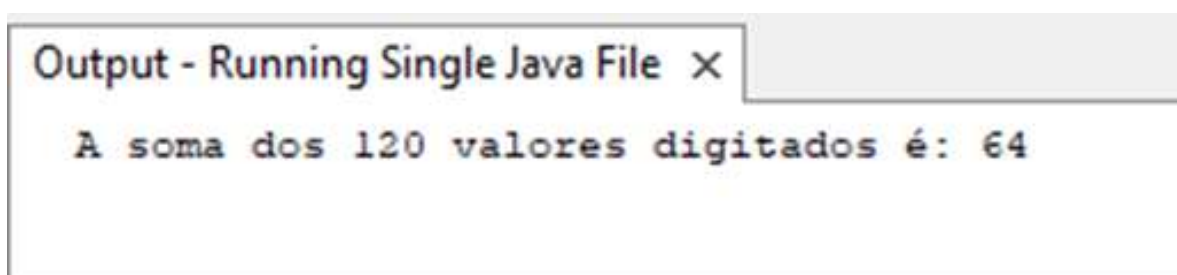
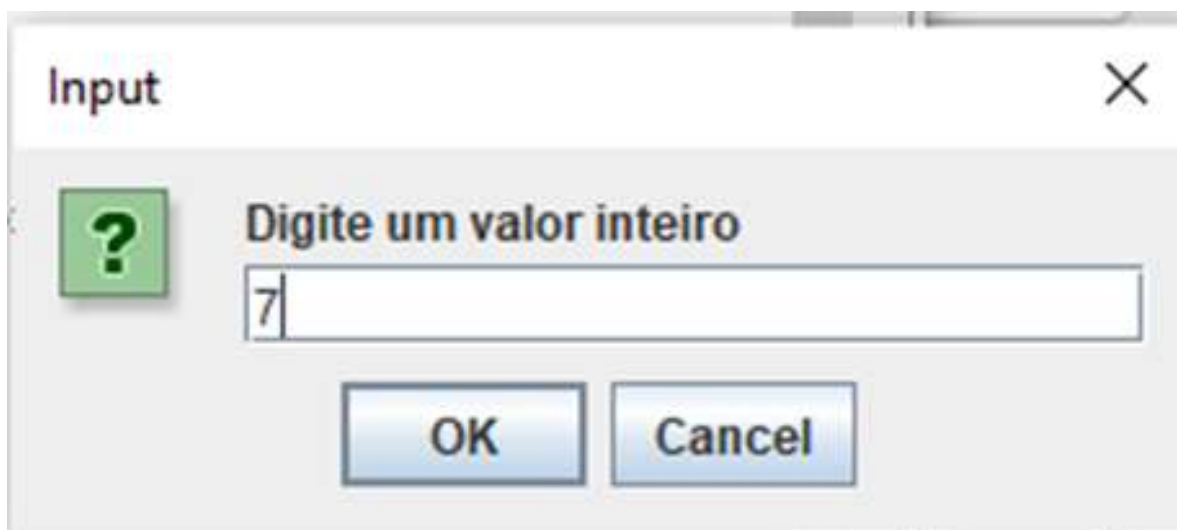
**} // classe Somar**



```

1  import javax.swing.*;
2  class Somar {
3      public static void main(String args [ ]) {
4          int MatSoma[][], a, b, soma = 0;
5          MatSoma = new int [10][12];
6
7          for (a = 0 ; a <= 9 ; a++){
8              for ( b = 0 ; b <= 11 ; b++ ) {
9                  MatSoma[a][b] = Integer.parseInt(JOptionPane.showInputDialog
10                     ("Digite um valor inteiro"));
11                  soma = soma + MatSoma[a][b];
12              }
13          }
14          System.out.println ("A soma dos 120 valores digitados é: " + soma);
15          System.exit(0);
16      }
17  }

```



**Atividade extra**

Indicação de leitura: **Fundamentos da Programação de Computadores algoritmos, pascal, C/C++ e Java**, da Ana Fernanda Gomes Ascencio e Edile Aparecida Veneruchi de Campos, capítulos 6 e 7.



## **Referência Bibliográfica**

CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. **Algoritmos Teoria e Prática**. Editora Campus. 3a Edição. 2012.

GOODRICH, M. T.; TAMASSIA, R. **Estruturas de Dados & Algoritmos em Java**. Editora Grupo A: Bookman, 5a Edição. 2013.

**Ir para exercício**