

Vetor ou Array

O que é um vetor (array)?

Um **vetor (array)** é uma **estrutura de dados** que armazena **vários valores do mesmo tipo**, organizados em sequência, dentro de **uma única variável**.

Um vetor é como **uma lista de caixas** onde cada caixa guarda um valor, e todas têm o mesmo tipo (todos int, ou todos String, etc.).

índice:	0	1	2	3
valor:	10	20	30	40

Cada posição tem um **índice**, que começa no **0**.

- Você acessa os valores pelo índice:
numeros[0] → 10 (no indice zero você tem o 10)

Definição técnica (Java):

Um vetor é um **objeto** que armazena uma quantidade fixa mas sua criação pode ser dinâmica de elementos do mesmo tipo, acessíveis por um índice.

Na prática, por que usar vetores?

Você usa um vetor quando precisa trabalhar com **uma coleção de valores de uma forma em mais baixo nível**, como processar estrutura de dados:

- notas de alunos
- lista de produtos
- números lidos do teclado
- componentes em html
- dados para análise e processamento
- muito usado em javascript para manipular elementos em tela
- e etc.

EXEMPLO PRÁTICO NO MUNDO REAL

Este exemplo é da plataforma de conteúdo online

```
/*Obtem os ids de video aulas na tela que está em array*/
var idvideoaulaCkeck = document.getElementsByClassName('idvideoaulaCkeck');

for (var i = 0; i < idvideoaulaCkeck.length; i++) {

    var AulaConcluida = false;

    /*Identifica as aulas que o aluno realizou e marca como realizada*/
    for (var x = 0; x < arrayAulas.length; x++) {
        if (arrayAulas[x] == idvideoaulaCkeck[i].value){
            idvideoaulaCkeck[i].checked = "checked";
            AulaConcluida = true;
            break;
        }
    }
}
```

Declarar e Criar Vetores

1) Declarar + inicializar

```
java

int[] numeros = {10, 20, 30, 40};
```

[Ver exemplo 1 na ide](#)

2) Declarar vazio e depois preencher

```
java

int[] notas = new int[5];
notas[0] = 10;
notas[1] = 8;
notas[2] = 7;
```

[Ver exemplo 2 na ide](#)

3) Declarar sem tamanho (somente com valores)

```
String[] nomes = {"Ana", "Pedro", "Lucas"};
```

[Ver exemplo 3 na ide](#)

4) Acessar e modificar elementos

```
System.out.println(notas[0]); // primeiro elemento
```

[Ver exemplo 4 na ide](#)

Ponto importante:

- Índices começam em **0**
- Último índice = **tamanho – 1 ou (ex: 10 elementos então 10 – 1 = 9)**
- *Teremos um array de 0 até 9*

```
notas[2] = 9;
```

5) Percorrer Vetores

```
for (int i = 0; i < notas.length; i++) {  
    System.out.println("Nota na posição " + i + ": " + notas[i]);  
}
```

[Ver exemplo 5 na ide](#)

ForEach

```
for (int nota : notas) {  
    System.out.println("Nota: " + nota);  
}
```

[Ver exemplo 5 na ide](#)

Quando usar cada um:

- for → quando precisa do índice numero
- for-each → quando só precisa do valor

Arrays com Scanner (Recebendo do usuário)

```
public class AulaVetores {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
  
        int[] valores = new int[5];  
  
        for (int i = 0; i < valores.length; i++) {  
            System.out.print("Digite um número: ");  
            valores[i] = sc.nextInt();  
        }  
  
        System.out.println("Valores digitados:");  
        for (int v : valores) {  
            System.out.println(v);  
        }  
    }  
}
```



[Ver exemplo 6 na ide](#)

Somar todos os valores

```
int soma = 0;  
  
for (int n : valores) {  
    soma += n;  
}  
  
System.out.println("Soma total = " + soma);
```

[Ver exemplo 7 na ide](#)

Encontrar o maior valor

```
int maior = valores[0];

for (int n : valores) {
    if (n > maior) {
        maior = n;
    }
}

System.out.println("Maior número = " + maior);
```

[Ver exemplo 8 na ide](#)

Média de notas

```
double total = 0;

for (double n : notas) {
    total += n;
}

double media = total / notas.length;

System.out.println("Média = " + media);
```

[Ver exemplo 9 na ide](#)

Array de objetos de uma classe

```
public class Aluno {
    String nome;
    int idade;

    public Aluno(String nome, int idade) {
        this.nome = nome;
        this.idade = idade;
    }

    public void mostrar() {
        System.out.println("Nome: " + nome + ", Idade: " + idade);
    }
}
```

[Ver exemplo 10 na ide](#)

Array de objetos Aluno

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

        Aluno[] alunos = new Aluno[3];

        alunos[0] = new Aluno("Ana", 20);
        alunos[1] = new Aluno("Bruno", 22);
        alunos[2] = new Aluno("Carlos", 19);

        for (Aluno a : alunos) {
            a.mostrar();
        }
    }
}
```

[Ver exemplo 10 na ide](#)

⭐ Explicação Final sobre Arrays em Java

Um **array** (vetor) é uma das estruturas de dados mais importantes e fundamentais do Java. Ele permite armazenar **vários valores do mesmo tipo** dentro de uma única variável, organizados em posições chamadas **índices**, que começam em **0**.

O array funciona como uma sequência de caixas enfileiradas: cada caixa guarda um valor e todas têm o mesmo tipo. É ideal quando você precisa trabalhar com **múltiplos dados relacionados**, como notas, nomes, números, produtos, objetos etc.

✓ Principais características:

- **Tamanho fixo:** depois de criado, não pode aumentar nem diminuir.
- **Acesso rápido:** você acessa qualquer posição diretamente usando `array[indice]`.
- **Tipo único:** todos os elementos precisam ser do mesmo tipo (`int`, `double`, `String`, objetos, etc.).
- **Índice começa em 0:** o primeiro elemento é `array[0]`, e o último é `array[array.length - 1]`.

✓ Por que arrays são importantes?

Porque formam a base para outras estruturas mais avançadas, como:

- `ArrayList`
- `List`
- `Collections`
- Estruturas de dados complexas
- Processamentos de massa de dados
- Algoritmos de busca, ordenação, filtros, etc.

Além disso, arrays permitem trabalhar desde tarefas simples (armazenar 5 notas) até operações mais complexas, como processamento de milhares de registros em sistemas reais.

✓ Resumo:

Array é uma coleção ordenada de elementos do mesmo tipo, acessados por índice, com tamanho fixo, e extremamente útil para organizar e processar múltiplos valores de forma eficiente.

TAREFA DE CASA

Crie uma classe chamada Aluno contendo:

1. Um atributo **nome** (String)
2. Um **array de 4 posições** para armazenar as **notas** do aluno (tipo double ou int)

Em seguida, faça:

- Um construtor para receber o **nome** do aluno
- Um método para **preencher o array de notas** (pode ser via Scanner ou manualmente)
- Um método para **exibir todas as notas** do aluno
- Um método para **calcular e retornar a média** das 4 notas

Por fim, na classe principal (Main ou TesteAluno):

1. Crie um objeto da classe **Aluno**
2. Preencha o nome
3. Preencha as 4 notas
4. Mostre todas as notas do aluno
5. Mostre a média final

FIM