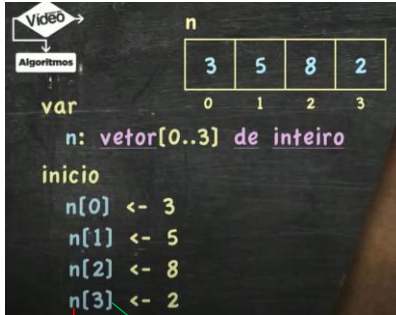


AULA 14 – VETORES

Variáveis compostas – vetores e matrizes

Em algoritmo:



Variável

Índice

Quer dizer que existem 4 posições. No Java, a primeira posição de um vetor é sempre 0

Em Java:

```
int n[] = new int[4];
n[0] = 3;
n[1] = 5;
n[2] = 8;
n[3] = 2;
```

Atribuição dos valores

Lembrando: ao utilizar a palavra **new**, estou criando um novo objeto! Ou seja, vai possuir métodos e atributos

Outra maneira de atribuir os valores:

```
int n[] = {3,5,8,2};
```

No NetBeans:

Para mostrar o vetor na tela:

```
public static void main(String[] args) {
    // TODO code application logic here
    int n [] = {3,2,8,7,5,4};

    for (int c = 0; c <= 5; c++) {
        System.out.print(n[c] + " ");
    }
}
```

run: 3 2 8 7 5 4

Lembrando: posso utilizar qualquer estrutura de repetição!

Alguns atributos dos vetores:

Length (comprimento):

```
public static void main(String[] args) {
    // TODO code application logic here
    int n [] = {3,2,8,7,5,4};
    System.out.println("O total de casas de N é " + n.length);
}
```

run: O total de casas de N é 6

Outra maneira do Length:

```
public static void main(String[] args) {  
    // TODO code application logic here  
    int n [] = {3,2,8,7,5,4};  
    for (int c = 0; c <= n.length-1; c++) {  
        System.out.println("Na posição " + c + " temos o valor " + n [c]);  
    }  
}
```

run:
Na posição 0 temos o valor 3
Na posição 1 temos o valor 2
Na posição 2 temos o valor 8
Na posição 3 temos o valor 7
Na posição 4 temos o valor 5
Na posição 5 temos o valor 4

Em vez de fazer o c começar com 0 (c = 0) e terminar em 5 (c <= 5), eu vou fazer c <= n.length-1, sabendo que vou ter que ir de 0 até 5, sendo que temos 6 posições.

Outra maneira seria c < n.length

Exercício quantos dias tem em cada mês do ano:

```
public static void main(String[] args) {  
    // TODO code application logic here  
    String mes [] = {"Jan", "Fev", "Mar", "Abr", "Mai", "Jun", "Jul",  
        "Ago", "Set", "Out", "Nov", "Dez"};  
    int tot [] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};  
    for (int c = 0; c < mes.length; c++) {  
        System.out.println(mes[c] + " tem " + tot [c] + " dias.");  
    }  
}
```

run:
Jan tem 31 dias.
Fev tem 28 dias.
Mar tem 31 dias.
Abr tem 30 dias.
Mai tem 31 dias.
Jun tem 30 dias.
Jul tem 31 dias.
Ago tem 31 dias.
Set tem 30 dias.
Out tem 31 dias.
Nov tem 30 dias.
Dez tem 31 dias.

Estrutura For It (Para Cada):

- Funciona exclusivamente para vetores!
- Dentro da estrutura **for** não vai ter ponto e vírgula e sim dois pontos, além de não ter 3 elementos, mas sim 2.

```
int num[] = {3, 5, 1, 8, 4};  
for (int valor: num) {  
    System.out.println(valor);  
}
```

Leia-se: "para cada elemento de num, coloque dentro de valor".

```
public static void main(String[] args) {  
    // TODO code application logic here  
    double v[] = {3.5, 2.75, 9, -4.5};  
    for (double valor: v) {  
        System.out.print(valor + " ");  
    }  
}
```

run:
3.5 2.75 9.0 -4.5

Para colocar o vetor em ordem: Arrays.sort

```
int num[] = {3, 5, 1, 8, 4};  
Arrays.sort(num);  
for (int valor: num) {  
    System.out.println(valor);  
}
```

```
public static void main(String[] args) {
    // TODO code application logic here
    double v[] = {3.5, 2.75, 9, -4.5};
    Arrays.sort(v);
    for (double valor: v) {
        System.out.print(valor + " ");
    }
}
```

run:
-4.5 2.75 3.5 9.0

Para buscar um vetor: binarySearch

```
int num[] = {3, 5, 1, 8, 4};
int pos = Arrays.binarySearch(num, 1);
System.out.println(pos);

public static void main(String[] args) {
    // TODO code application logic here
    int vet [] = {3, 7, 6, 1, 9, 4, 2};
    for (int v:vet) {
        System.out.print(v + " ");
    }
    System.out.println("");
    int p = Arrays.binarySearch(vet, 1);
    System.out.println("Encontrei o valor na posição " + p);
}
```

run:
3 7 6 1 9 4 2
Encontrei o valor na posição 3

Buscar a posição do valor 1

OBS: se eu buscar a posição de um valor que não existe, o índice dará negativo, pois não existem índices negativos no Java, então, se não encontrar o valor, o valor da chave será negativo

```
public static void main(String[] args) {
    // TODO code application logic here
    int vet [] = {3, 7, 6, 1, 9, 4, 2};
    for (int v:vet) {
        System.out.print(v + " ");
    }
    System.out.println("");
    int p = Arrays.binarySearch(vet, 5);
    System.out.println("Encontrei o valor na posição " + p);
}
```

run:
3 7 6 1 9 4 2
Encontrei o valor na posição -8

Preenchimento do vetor automático: Arrays.fill

```
int num[] = int[5];
Arrays.fill(num, 8);
for (int valor: num) {
    System.out.println(valor);
}
```

Todos os elementos do vetor
vão receber o valor 8

```
public static void main(String[] args) {
    // TODO code application logic here
    int v[] = new int [20];
    Arrays.fill(v, 0);
    for (int valor: v) {
        System.out.print(valor + " ");
    }
}
```

run:
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0