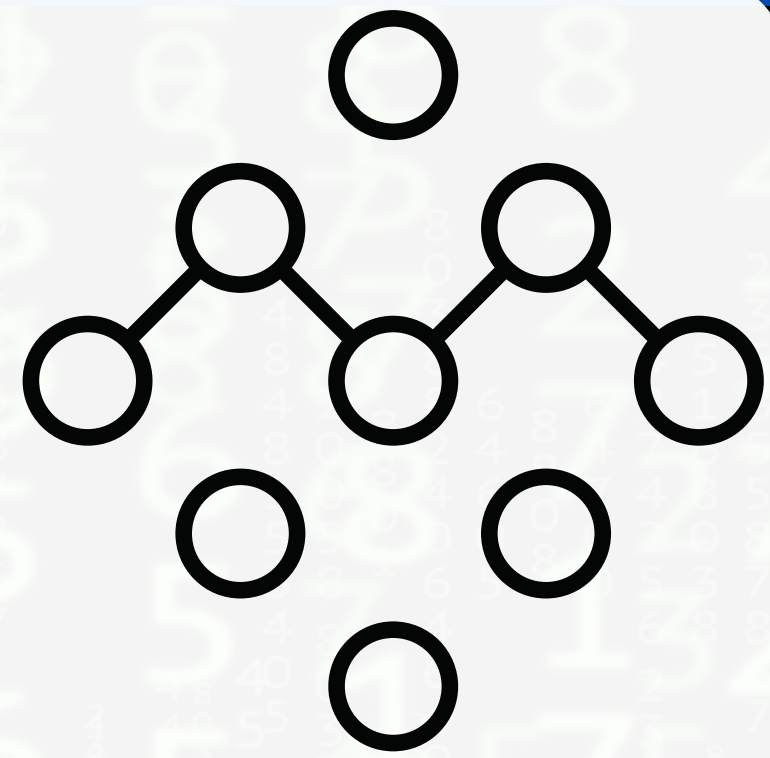


Curso de

Lógica de Programação



Por

Jeiverson C. V. M. Santos

Aula 15

Vetores

"A direção é mais importante do que a velocidade."

Edson Marques

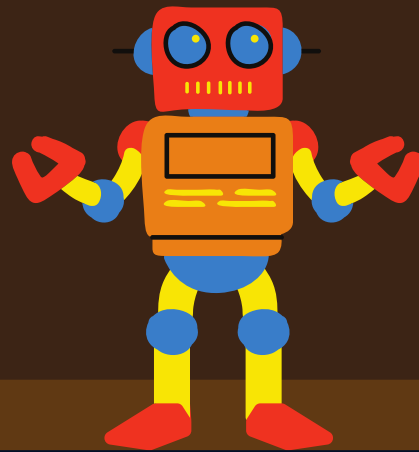


Primeiro

Vamos ver o resultado do "para casa"



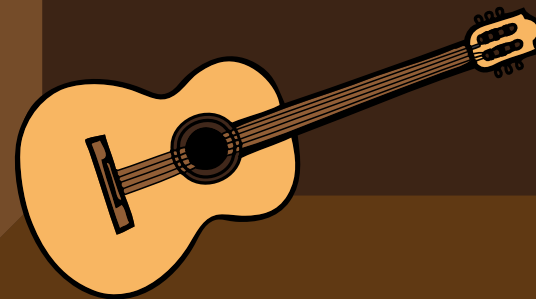
Relembrando



● Robo: ●
Brinquedo



● Bola: ●
Brinquedo



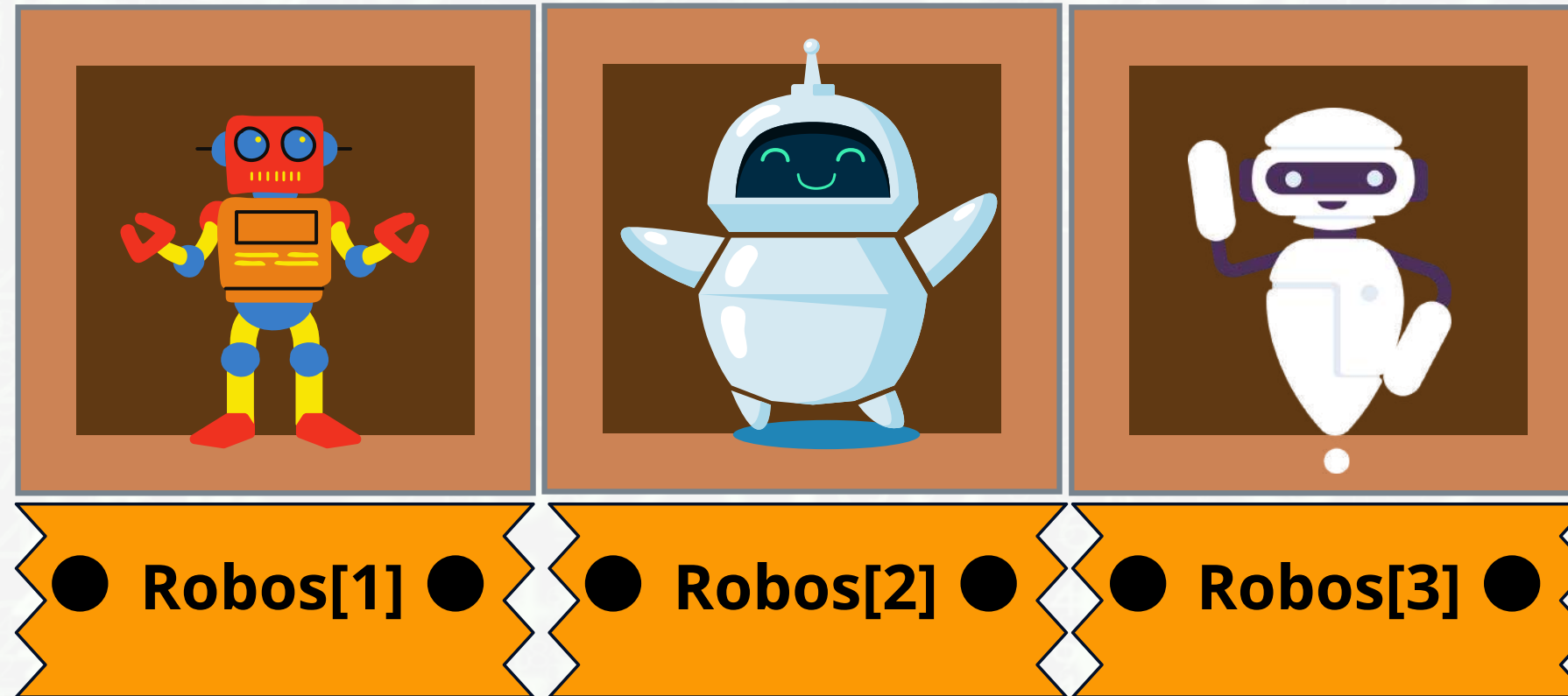
● Violao: ●
Instrumento



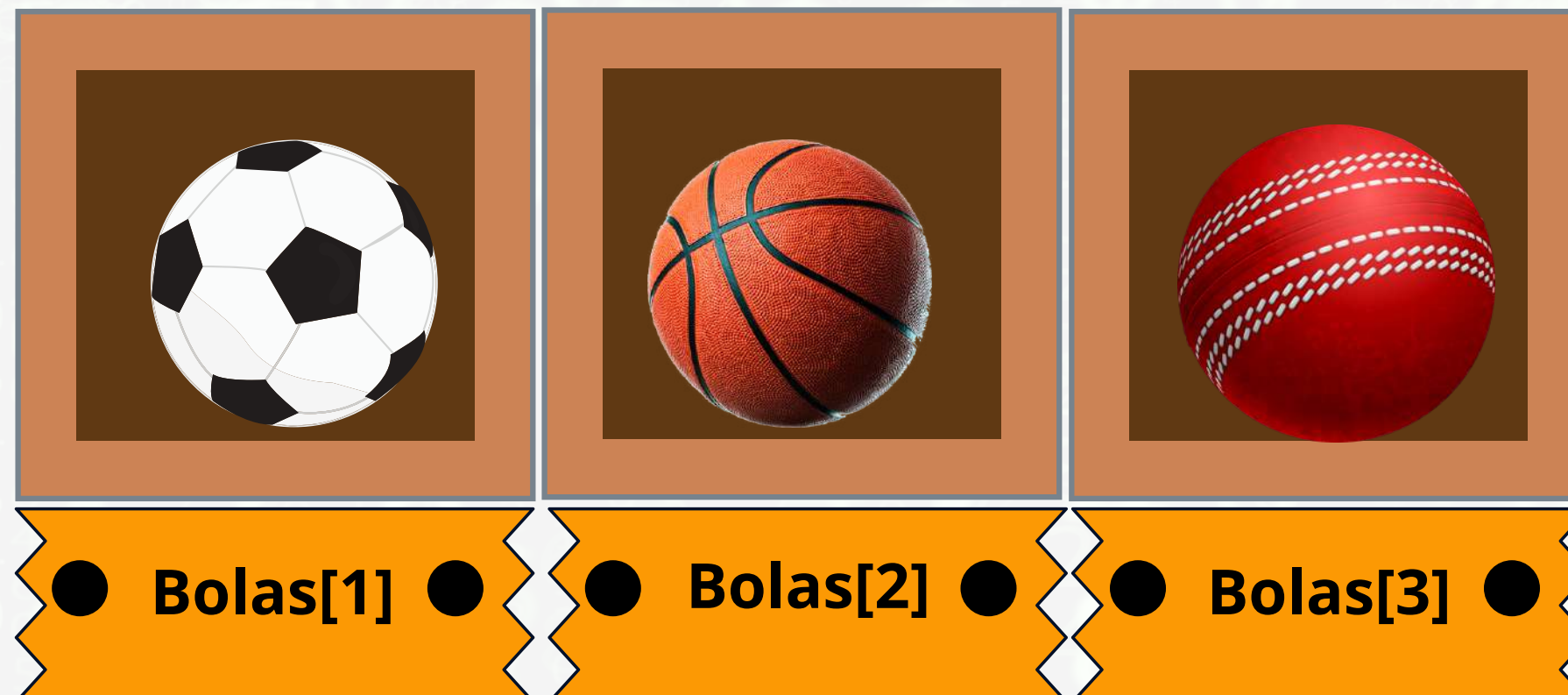
● Bateria: ●
Instrumento

Agora mudando um pouco

**Robos[1-3]:
Brinquedo**



**Bolas[1-3]:
Brinquedo**



Vetor

Exemplo

```
1 algoritmo "variáveis simples"
2 var
3     n1, n2, n3: inteiro
4 inicio
5     n1 <- 8
6     n2 <- 2
7     n3 <- 4
8     escreva(n1, n2, n3)|
9 fimalgoritmo
```

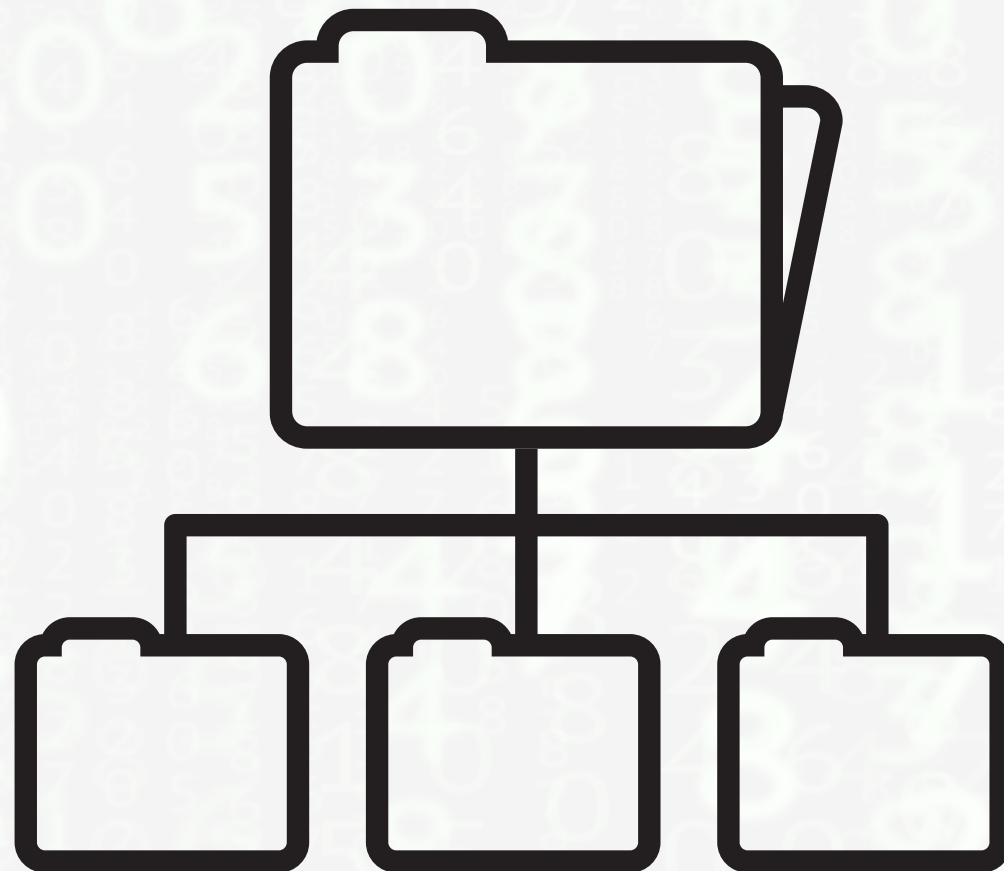
```
1 algoritmo "vetor"
2 var
3     n: vetor[1..3] de inteiro
4 inicio
5     n[1] <- 8
6     n[2] <- 2
7     n[3] <- 4
8     escreva(n[1], n[2], n[3])|
9 fimalgoritmo
```


Exemplo 2

```
1 algoritmo "vetor"
2 var
3   nomes: vetor[1..3] de caractere
4   idades: vetor[1..3] de inteiro
5   i: inteiro
6 inicio
7   para i <- 1 ate 3 passo 1 faca
8     escreva("Nome: ")
9     leia(nomes[i])
10    escreva("Idade: ")
11    leia(idades[i])
12  fimpara
13  limpatela
14  para i <- 1 ate 3 passo 1 faca
15    escreval(nomes[i], idades[i])
16  fimpara
17 fimalgoritmo |
```

Exercício

Fazer um algoritmo que leia 10 números, guarde-os em um vetor e conte quantos pares e quantos ímpares foram digitados.



Resposta

```
1 algoritmo "Pares | Ímpares"
2 var
3   numeros: vetor[1..10] de inteiro
4   i, q_par, q_impar: inteiro
5 inicio
6   q_par <- 0
7   q_impar <- 0
8   para i <- 1 ate 10 passo 1 faca
9     escreva("n°: ")
10    leia(numeros[i])
11  fimpara
12  para i <- 1 ate 10 passo 1 faca
13    se numeros[i] % 2 = 0 entao
14      q_par <- q_par + 1
15    senao
16      q_impar <- q_impar + 1
17  fimse
18  fimpara
19  escreval("Quantidade de pares: ", q_par)
20  escreval("Quantidade de ímpares: ", q_impar)
21 fimalgoritmo |
```


Exercício

Aproveite o código anterior e faça uma modificação para que ele exiba a posição dos pares e a posição dos ímpares.



Resposta

```
1 algoritmo "Pares | Ímpares"
2 var
3   numeros: vetor[1..10] de inteiro
4   i, q_par, q_impar: inteiro
5 inicio
6   q_par <- 0
7   q_impar <- 0
8   para i <- 1 ate 10 passo 1 faca
9     escreva("nº: ")
10    leia(numeros[i])
11  fimpara
12  para i <- 1 ate 10 passo 1 faca
13    se numeros[i] % 2 = 0 entao
14      q_par <- q_par + 1
15      escreval("PAR na posição: ", i)
16    senao
17      q_impar <- q_impar + 1
18      escreval("ÍMPAR na posição: ", i)
19    fimse
20  fimpara
21  escreval("Quantidade de pares: ", q_par)
22  escreval("Quantidade de ímpares: ", q_impar)
23 fimalgoritmo |
```


Exercício

Fazer um algoritmo que leia 7 nomes e pergunte se a pessoa é homem, mulher ou não binário e depois os escreva separados de acordo com o gênero.



Resposta



```
1 algoritmo "homem | mulher | não binário"
2 var
3   nomes: vetor[1..7] de caractere
4   generos: vetor[1..7] de caractere
5   i: inteiro
6 procedimento EscreveNomes (genero: caractere)
7 inicio
8   para i <- 1 ate 7 passo 1 faca
9     se generos[i] = genero entao
10      escreva(nomes[i], " ")
11    fimse
12  fimpara
13  escreval()
14 fimprocedimento
15 inicio
16   para i <- 1 ate 7 passo 1 faca
17     escreva("Nome: ")
18     leia(nomes[i])
19     escreva("Gênero [M], [F] ou [NB]: ")
20     leia(generos[i])
21   fimpara
22   limpatela
23   EscreveNomes("M")
24   EscreveNomes("F")
25   EscreveNomes("NB")
26 fimalgoritmo |
```


Ordenando um vetor

[9, 0, 5, 1] → [0, 1, 5, 9]

código

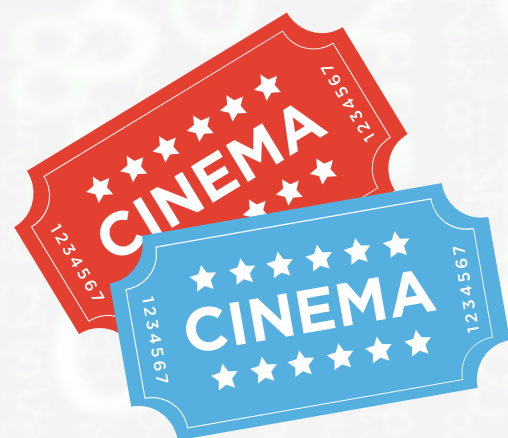
```
1 algoritmo "ordenando vetor"
2 var
3   v: vetor[1..4] de inteiro
4   i, j, memoria: inteiro
5 inicio
6   para i <- 1 ate 4 passo 1 faca
7     escreva("nº: ")
8     leia(v[i])
9   fimpara
10  para i <- 1 ate 4 passo 1 faca
11    escreva(v[i])
12  fimpara
13  para i <- 1 ate 3 faca
14    para j <- i+1 ate 4 faca
15      se v[i] > v[j] entao
16        memoria <- v[i]
17        v[i] <- v[j]
18        v[j] <- memoria
19      fimse
20    fimpara
21  fimpara
22  escreval()
23  para i <- 1 ate 4 passo 1 faca
24    escreva(v[i])
25  fimpara
26 fimalgoritmo |
```



Para casa

Escolha um pra fazer

Fazer um algoritmo que ordene um vetor de forma decrescente.



Fazer um algoritmo que consiga reserva a cadeira em uma fileira de cinema. Antes da reserva, mostrar as cadeiras, depois mostrar também, mas com um x no lugar da cadeira reservada.