

PROGRAMAÇÃO DE INTERNET

Prof. Richard Brosler

E-mail: richard.brosler@sp.senai.br

Grupo WhatsApp: <https://bit.ly/3K8jFIg>





AGENDA

- Revisão da última aula
 - Procedimentos
 - Funções
- Vetores
- Matrizes
- Ordenação de valores com vetores
- Exercícios

PROCEDIMENTOS

- Blocos de códigos que não devolvem resposta após sua execução
- Vejam o exemplo ao lado, onde usaremos um procedimento para montagem da tela

```
1 Algoritmo "semnome"
2
3 procedimento montaTela()
4 inicio
5   limpatela()
6   escreval("Menu de opções")
7   escreval("1 - cadastrar")
8   escreval("2 - consultar")
9   escreval("3 - alterar")
10  escreval("4 - excluir")
11  escreval("9 - fim")
12  escreval()
13  escreva("Digite sua opção:")
14 fimprocedimento
15
16 Var
17   opc : inteiro
18 Inicio
19   repita
20     montaTela()
21     leia(opc)
22   ate opc = 9;
23 Fimalgoritmo
```

FUNÇÕES

- Funções, assim como procedimentos, são blocos de código que serão executados, porém, diferentemente de um procedimento, elas retornam valor.

```
1 Algoritmo "semnome"  
2  
3 funcao somar() : inteiro  
4 var ret : inteiro  
5 inicio  
6   ret := a + b  
7   retorne ret  
8 fimfuncao  
9  
10 Var  
11   a, b : inteiro  
12 Inicio  
13   a := 10  
14   b := 20  
15   escreval("a soma é ", somar())  
16 Fimalgoritmo
```


VETORES E MATRIZES

- Vetores – conjunto de variáveis do mesmo tipo de dados (estrutura homogênea) que pode ser acessado por um único nome em conjunto com um índice que representará a variável (gaveteiro com uma gaveta por vez)
- Matriz – conjunto de variáveis do mesmo tipo de dados (estrutura homogênea) assim como o vetor, porém, esse possuiu mais de uma dimensão (exemplo linha e coluna). Caixa postal de correios, planilhas de valores são exemplos similares para fazer uma relação de armazenamento, porém, lembrando que assim como o vetor, os dados devem ser do mesmo tipo em todos os espaços de armazenamento

VETORES E MATRIZES

- Vamos a um exemplo, vamos supor que você precise cadastrar 5 nomes de pessoas em seu sistema, como você faria?
- Talvez dessa forma...

```
1 Algoritmo "semnome"  
2 Var  
3  nom1, nom2, nom3, nom4, nom5 : caracter  
4  
5 Inicio  
6  escreva("Digite o nome da pessoa 1: ")  
7  leia(nom1)  
8  
9  escreva("Digite o nome da pessoa 2: ")  
10 leia(nom2)  
11  
12 escreva("Digite o nome da pessoa 3: ")  
13 leia(nom3)  
14  
15 escreva("Digite o nome da pessoa 4: ")  
16 leia(nom4)  
17  
18 escreva("Digite o nome da pessoa 5: ")  
19 leia(nom5)  
20  
21 Fimalgoritmo |
```


VETORES E MATRIZES

- Ou talvez dessa forma
- Agora note o padrão que temos nas linhas 8-9, 12-13, 16-17, 20-21 e 24-25
- Se pudéssemos utilizar o laço de repetição, seria muito interessante se não fosse o problema das variáveis nas leituras, não temos como colocar `nomintI`

```
1 Algoritmo "semnome"
2 Var
3  nom1, nom2, nom3, nom4, nom5 : caracter
4  intI : inteiro
5 Inicio
6   intI <- 0
7
8   intI <- intI + 1
9   escreva("Digite o nome da pessoa ",intI,": ")
10  leia(nom1)
11
12  intI <- intI + 1
13  escreva("Digite o nome da pessoa ",intI,": ")
14  leia(nom2)
15
16  intI <- intI + 1
17  escreva("Digite o nome da pessoa ",intI,": ")
18  leia(nom3)
19
20  intI <- intI + 1
21  escreva("Digite o nome da pessoa ",intI,": ")
22  leia(nom4)
23
24  intI <- intI + 1
25  escreva("Digite o nome da pessoa ",intI,": ")
26  leia(nom5)
27
28 Fimalgoritmo
```

VETORES E MATRIZES

- Com vetor, podemos fazer isso, vejamos o exemplo de vetor...

```
1 Algoritmo "semnome"  
2 Var  
3   nom : vetor [1..5] de caracter  
4   intI : inteiro  
5 Inicio  
6  
7   para intI de 1 ate 5 faca  
8     escreva("Digite o nome da pessoa ",intI,": ")  
9     leia(nom[intI])  
10  fimpara  
11 |  
12 Fimalgoritmo
```


VETORES E MATRIZES

- Agora vamos usar um exemplo com matrizes

```
1 Algoritmo "semnome"
2 Var
3   nom : vetor [1..5,1..2] de caracter
4   intI : inteiro
5 Inicio
6
7   para intI de 1 ate 5 faca
8     escreva("Digite o nome da pessoa ",intI,": ")
9     leia(nom[intI,1])
10    escreva("Digite o endereço da pessoa ",intI,": ")
11    leia(nom[intI,2])
12  fimpara
13
14 Fimalgoritmo
```

ORDENAÇÃO DE VALORES COM VETORES

- Algoritmo de ordenação bolha (BubbleSort)

```
3 procedimento bubbleSort()  
4 var  
5   li, lf : inteiro  
6   apoio  : caracter  
7 inicio  
8   para li de 1 ate 9 faca  
9     para lf de li ate 10 faca  
10      //ordem ascendente  
11      se itens[li] > itens[lf] entao  
12        apoio <- itens[lf]  
13        itens[lf] <- itens[li]  
14        itens[li] <- apoio  
15      fimse  
16    fimpara  
17  fimpara  
18 fimprocedimento
```

```
20 Var  
21 // Seção de Declarações das variáveis  
22   itens : vetor [1..10] de caracter  
23   i : inteiro  
24 Inicio  
25 // Seção de Comandos, procedimento, funções, operadores  
26   aleatorio on  
27   para i de 1 ate 10 faca  
28     leia(itens[i])  
29   fimpara  
30   aleatorio off  
31   bubbleSort()  
32   escreval("Listando os valores classificados!")  
33   para i de 1 ate 10 faca  
34     escreval(itens[i])  
35   fimpara  
36  
37 Fimalgoritmo
```


ORDENAÇÃO DE VALORES COM VETORES

- Algoritmo de ordenação quickSort

```
1 Algoritmo "semnome"
2
3 procedimento troca( li : inteiro; lf : inteiro)
4 var
5   apoio : caracter
6 inicio
7   apoio <- itens[li]
8   itens[li] <- itens[lf]
9   itens[lf] <- apoio
10 fimprocedimento
11
12 funcao particao(start : inteiro; end : inteiro) : inteiro
13 var
14   li, lj : inteiro
15 inicio
16   li <- start
17   para lj de start ate end - 1 faca
18     se itens[lj] <= itens[end] entao
19       troca(li,lj)
20       li <- li + 1
21   fimse
22   fimpara
23   troca(li,end)
24   retorne li
25 fimfuncao
```

```
27 procedimento quickSort(ini : inteiro; fim : inteiro)
28 var
29   pivo : inteiro
30 inicio
31   se ini < fim entao
32     pivo <- particao(ini, fim)
33     quickSort(ini, pivo - 1)
34     quickSort(pivo + 1, fim)
35   fimse
36 fimprocedimento
37
38 Var
39 // Seção de Declarações das variáveis
40   itens : veter [1..10] de caracter
41   i : inteiro
42 Inicio
43 // Seção de Comandos, procedimento, funções, operadores
44   aleatorio on
45   para i de 1 ate 10 faca
46     leia(itens[i])
47   fimpara
48   aleatorio off
49   quickSort(1,10)
50   escreval("Listando os valores classificados!")
51   para i de 1 ate 10 faca
52     escreval(itens[i])
53   fimpara
54
55 Fimalgoritmo
```

EXERCÍCIOS

- 1) Desenvolver um algoritmo para solicitar 10 nomes, armazená-los em um vetor, e após solicitar os nomes o algoritmo deverá lista-los com seu respectivo índice do vetor. Exemplo: 1 – Richard
- 2) Desenvolver um algoritmo para solicitar 15 números inteiros e armazená-los em um vetor. Após a inserção dos valores, o algoritmo deverá lista-los e classifica-los em par ou impar. Exemplo :
 - 15 – impar
 - 10 – par
 -

EXERCÍCIOS

- 3) Desenvolver um algoritmo para solicitar 8 números inteiros e armazená-los em um vetor. Após armazená-los, o algoritmo deverá listar os números e depois mostrar uma mensagem indicando o total de múltiplos de 6.
- 4) Desenvolver um algoritmo para solicitar a média dos 4 bimestre de 15 alunos e armazená-las em uma matriz que conterà as médias dos bimestres e a média final. Solicitar também o nome dos alunos para armazenar em outra matriz que conterà o nome e a situação AP ou RP (aprovado ou reprovado). As colunas media final e situação do aluno devem ser calculadas, ou seja, a média será a soma das médias dos 4 bimestres e divididas por 4. A situação será AP se a média for maior ou igual a 6.0. Após as entradas de dados e cálculos, listar o nome do aluno, média final e situação.

EXERCÍCIOS

- 5) Desenvolver um algoritmo para solicitar o nome do funcionário e seu salário de 20 funcionários e armazená-los em vetores (1 para o nome e 1 para o salário). Após solicitar os nomes e valores, o algoritmo deverá calcular um reajuste de 8% nos salários e mostrar uma listagem contendo o nome do funcionário e o novo salário.
- 6) Desenvolver um algoritmo para solicitar o valor de compra e de venda de 100 produtos. O algoritmo deverá verificar o lucro desses produtos e ao término mostrar o total de produtos que tiveram lucros abaixo de 10%, total de produtos que tiveram lucro entre 10% e 20% e o total de produtos que tiveram lucro acima de 20%. Cálculo do % de lucro é $((\text{preço de venda} - \text{preço de compra}) / \text{preço de compra}) * 100$

EXERCÍCIOS

- 7) Desenvolver um algoritmo para solicitar o nome e a data de nascimento (formato ddmm) de 20 pessoas e armazená-los (nome em 1 vetor e data em outro vetor). Considerar que as datas serão informadas corretamente. Após a entrada de dados, você deverá verificar qual signo é de cada pessoa e armazená-lo em um vetor, para isso utilize a tabela ao lado.
- Dica: Use um vetor para o último dia e um vetor para os nomes dos signos. O índice deles é o próprio mês.

Mês	Último Dia	Signo
1	20	Capricórnio
2	19	Aquário
3	20	Peixes
4	20	Áries
5	20	Touro
6	20	Gêmeos
7	21	Câncer
8	22	Leão
9	22	Virgem
10	22	Libra
11	21	Escorpião
12	21	Sagitário

EXERCÍCIOS

- 8) Desenvolver um algoritmo para solicitar 30 nomes. Após a entrada de dados, classificá-los em ordem ascendente e listá-los.
- 9) Desenvolver um algoritmo para solicitar 50 nomes. Após a entrada de dados, classifica-los em ordem descendente e listá-los.