INTRODUÇÃO À COMPUTAÇÃO

UD II - ALGORITMOS

ESTRUTURA DE DADOS

Estrutura de dados

ELEMENTOS DE COMPETÊNCIA

- Empregar recursos para operar em ambientes humanizados, integrando as dimensões física, humana e informacional deste ambiente operacional.
- Tomar decisões e conduzir ações, em situações de crise.

UD II - INTRODUÇÃO À ALGORITMOS

Estrutura de dados

OBJETIVOS

1. Identificar as estruturas de dados. (FACTUAL)
2. Compreender a aplicação das estruturas de dados na elaboração de algoritmos. (CONCEITUAL)
3. Aplicar as estruturas de dados na elaboração de algoritmos. (PROCEDIMENTAL)

Estrutura de dados

ATITUDES

1. Organização: capacidade de desenvolver atividades de forma sistemática e eficiente.
2. Dedicação: agir, realizando espontaneamente, com empenho e entusiasmo, as atividades necessárias ao cumprimento da missão.
3. Responsabilidade: capacidade de cumprir suas atribuições assumindo e enfrentando as consequências de suas atitudes e decisões.

Estrutura de dados

Os dados que são utilizados por um programa de computador necessitam estar armazenados de maneira organizada, de modo a possibilitar e facilitar, o acesso a eles durante o processamento, bem como as ações de inclusão, modificação e exclusão.

Estrutura de dados é um modo de armazenar e organizar dados com o objetivo de facilitar acesso e modificações. Nenhuma estrutura de dados única funciona bem para todas as finalidades e, por isso, é importante conhecer os pontos fortes e as limitações de várias delas.

Aqui abordaremos vetores e matrizes.

Vetores (ou array)

É uma estrutura de dados que compreende uma sequência ordenada e indexada de elementos mutáveis. Um vetor se constitui em uma variável composta, onde é armazenado uma sequência de dados do mesmo tipo e, por isso, essas variáveis são classificadas como homogêneas.

Declaração de Vetores

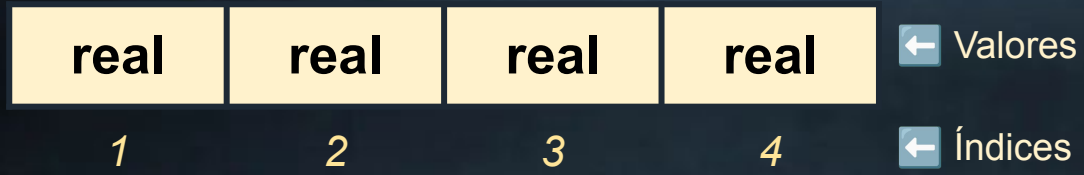
Sintaxe de declaração de vetor em pseudocódigo (seção “Var”):

```
<nome>: vetor [<início..fim>] de <tipo de dado>
```

Exemplo:

```
notas: vetor [1..4] de real
```

Corrigir apostila, sem <>



Atribuição de valores em vetores

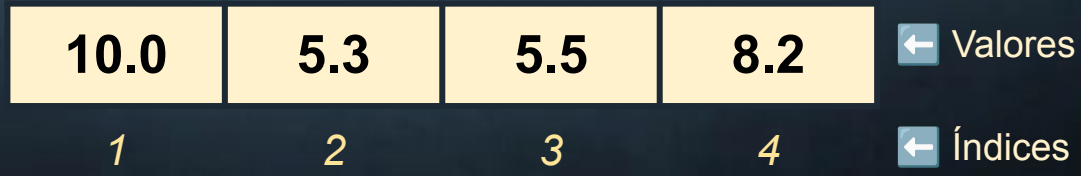
Sintaxe para atribuir valor a uma posição do vetor:

```
<nome>[índice] ← <valor>
```

Exemplos:

```
notas[1] ← 10.0
notas[2] ← 5.3
notas[3] ← 5.5
notas[4] ← 8.2
```

Vetor notas



Acessando valores em vetores

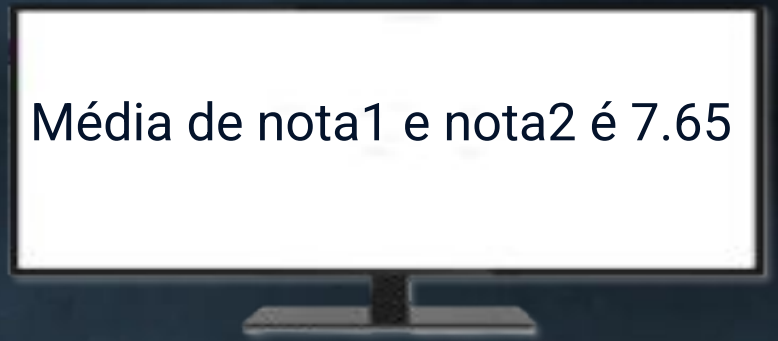
10.0	5.3	5.5	8.2
1	*2*	*3*	*4*

⬅ Valores

⬅ Índices

Para acessar um valor do vetor, utilize o índice indicativo da posição no vetor. Exemplo:

escreva(“Média de nota1 e nota2 é ”, (notas[1] + notas[2]) / 2)



Algoritmo

```
1  Algoritmo "09_01 Uso de Vetor"
2  //Exemplo de preenchimento de vetor e acesso por índices
3  Var
4    // Declaração de um vetor de 4 posições de números reais
5    notas : vetor [1..4] de Real
6    media : Real
7  Inicio
8    Escreval("--- Atribuindo valores ao vetor ---")
9    // A ordem de atribuição não importa, pois usamos o índice
10   notas[1] ← 9.5
11   notas[4] ← 6.3
12   notas[3] ← 4.7
13   notas[2] ← 8.7
14
15   // Acessando um valor específico
16   Escreval("Primeira nota: ", notas[1])
17
18   // Soma-se o conteúdo de cada "gaveta" e divide pelo total
19   media <- (notas[1] + notas[2] + notas[3] + notas[4]) / 4
20   Escreval("Média calculada: ", media)
21
22 FimAlgoritmo
```

O vetor ficaria:

9.5	8.7	4.7	6.3
1	2	3	4

Primeira nota: 9.5
Média: 7.3

Matriz

Uma matriz se constitui uma variável homogênea e mutável de duas dimensões: uma representando as linhas e a outra representando as colunas. Sintaxe da declaração de uma matriz (seção “var”):

linha *coluna*

<nome>: vetor [<início..fim>,<inicio..fim>] de <tipo de dado>

corrigir na apostila

Exemplo:

notas: vetor [1..3 , 1..3] de real

corrigir na apostila

	1	*2*	*3*
1	**real**	**real**	**real**
2	**real**	**real**	**real**
3	**real**	**real**	**real**

Atribuição de valores em matrizes

Sintaxe para atribuir valor a uma posição do vetor:

<nome>[linha, coluna] ← <valor>

Exemplos:

notas[1,1] ← 10
notas[2,2] ← 5
notas[3,2] ← 5
notas[3,3] ← 8

Matriz de notas

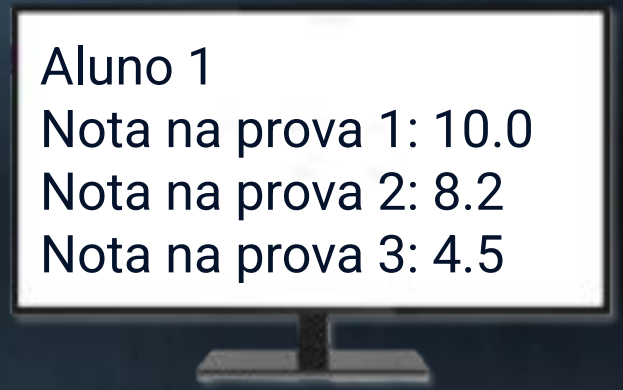
	1	2	3
1	**10**		
2		**5**	
3		**5**	**8**

Acessando valores em matrizes

Para acessar um valor de uma célula da matriz, utilize o índice (linha x coluna), indicativo da posição desejada.
Exemplo:

	1	2	3
1	10.0	8.2	4.5
2	3.8	5.0	7.3
3	8.3	5.0	8.0

```
escreva("Aluno 1:")
escreva("> Nota na prova 1: ", notas[1,1])
escreva("> Nota na prova 2: ", notas[1,2])
escreva("> Nota na prova 2: ", notas[1,3])
```



Algoritmo

```
1  Algoritmo "09_02 Uso de Matriz"
2  // Descrição: Exemplo de preenchimento e leitura de Matriz (Vetor Bidimensional)
3  Var
4    // Declaração de uma Matriz 3x3 (3 Linhas, 3 Colunas)
5    notas : vetor [1..3, 1..3] de Real
6    mediaAluno3 : Real
7  Inicio
8    // Preenchendo linhas da matriz
9    Escreval("# Atribuindo valores à Matriz")
10   notas[1,1] <- 8.0
11   notas[1,2] <- 9.5
12   notas[1,3] <- 7.6
13   notas[3,1] <- 8.8
14   notas[3,2] <- 4.8
15   notas[3,3] <- 8.8
16
17   // Acessando valores randomicamente
18   Escreval("--- Leitura dos Dados ---")
19   Escreval("Aluno 1 / nota 2: ", notas[1,2])
20   Escreval("Aluno 3 / nota 3: ", notas[3,3])
21
22   // Cálculo da Média do Aluno 3 (Linha 3 completa)
23   mediaAluno3 <- (notas[3,1] + notas[3,2] + notas[3,3]) / 3
24   Escreval("Média aluno 3: ", mediaAluno3:4:2)
25
26 FimAlgoritmo
```

	1	2	3
1	8.0	9.5	7.6
2			
3	8.8	4.8	8.8

```
# Atribuindo valores à Matriz
Aluno 1 / nota 2:  9.5
Aluno 3 / nota 3:  8.8
Média aluno 3:  7.47
```

Observação

Diferente de linguagens modernas, o Visualg não possui funções internas para manipular vetores e matrizes (como ordenar, redimensionar ou buscar automaticamente).

Exercícios em sala 1

Produza um algoritmo, em pseudocódigo, que permita a entrada de 10 números. Em seguida, permita que seja informado um número e verificado se está contido nos 10 números informados inicialmente.

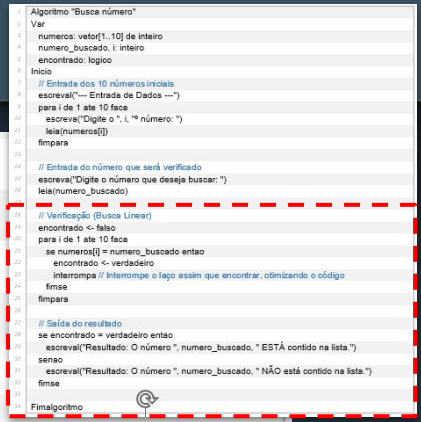
```
--- Entrada de Dados ---
Digite o  1° número: 1
Digite o  2° número: 2
Digite o  3° número: 3
Digite o  4° número: 4
Digite o  5° número: 5
Digite o  6° número: 6
Digite o  7° número: 7
Digite o  8° número: 8
Digite o  9° número: 9
Digite o  10° número: 10
Digite o número que deseja buscar: 6
Resultado: O número  6 ESTÁ contido na lista.

>>> Fim da execução do programa !
```

Exercícios em sala 1 - Solução

```
1  Algoritmo "Busca número"
2  Var
3     numeros: vetor[1..10] de inteiro
4     numero_buscado, i: inteiro
5     encontrado: logico
6  Inicio
7     // Entrada dos 10 números iniciais
8     escreval("--- Entrada de Dados ---")
9     para i de 1 ate 10 faca
10       escreva("Digite o ", i, "º número: ")
11       leia(numeros[i])
12    fimpara
13
14    // Entrada do número que será verificado
15    escreva("Digite o número que deseja buscar: ")
16    leia(numero_buscado)
17
```

Exercícios em sala 1 - Solução



```
Algoritmo "Busca número"
Var
  numeros: vetor[1..10] de inteiro
  numero_buscado, i: inteiro
  encontrado: logico
Inicio
  // Entrada dos 10 números iniciais
  escreval("--- Entrada de Dados ---")
  para i de 1 ate 10 faca
    escreva("Digite o ", i, "º número: ")
    leia(numeros[i])
  fimpara

  // Entrada do número que será verificado
  escreva("Digite o número que deseja buscar: ")
  leia(numero_buscado)

  // Verificação (Busca Linear)
  encontrado <- falso
  para i de 1 ate 10 faca
    se numeros[i] = numero_buscado entao
      encontrado <- verdadeiro
      interrompa // Interrompe o laço assim que encontrar, otimizando o código
    fimse
  fimpara

  // Saída do resultado
  se encontrado = verdadeiro entao
    escreval("Resultado: O número ", numero_buscado, " ESTÁ contido na lista.")
  senao
    escreval("Resultado: O número ", numero_buscado, " NÃO está contido na lista.")
  fimse

Fimalgoritmo
```

```
18    // Verificação (Busca Linear)
19    encontrado <- falso
20    para i de 1 ate 10 faca
21      se numeros[i] = numero_buscado entao
22        encontrado <- verdadeiro
23        interrompa // Interrompe o laço assim que encontrar, otimizando o código
24      fimse
25    fimpara
26
27    // Saída do resultado
28    se encontrado = verdadeiro entao
29      escreval("Resultado: O número ", numero_buscado, " ESTÁ contido na lista.")
30    senao
31      escreval("Resultado: O número ", numero_buscado, " NÃO está contido na lista.")
32    fimse
33
34  Fimalgoritmo
```

Exercícios em sala 1b

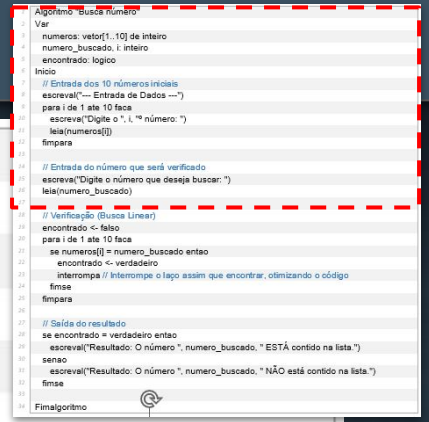
Produza um algoritmo, em pseudocódigo, que permita a entrada de 10 números. Em seguida, permita que seja informado um número e verificado se está contido nos 10 números informados inicialmente e em qual posição na lista.

```
--- Entrada de Dados ---
Digite o  1° número: 33
Digite o  2° número: 77
Digite o  3° número: 88
Digite o  4° número: 99
Digite o  5° número: 11
Digite o  6° número: 23
Digite o  7° número: 39
Digite o  8° número: 97
Digite o  9° número: 18
Digite o  10° número: 33
Digite o número que deseja buscar: 11
O número  11 ESTÁ contido na lista (posição: 5

>>> Fim da execução do programa !
```

Exercícios em sala 1b - Solução

```
1  Algoritmo "Busca número versão 2"
2  Var
3     numeros: vetor[1..10] de inteiro
4     numero_buscado, i, posicao: inteiro
5     encontrado: logico
6  Inicio
7     // Entrada dos 10 números iniciais
8     escreval("--- Entrada de Dados ---")
9     para i de 1 ate 10 faca
10       escreva("Digite o ", i, "º número: ")
11       leia(numeros[i])
12    fimpara
13
14    // Entrada do número que será verificado
15    escreva("Digite o número que deseja buscar: ")
16    leia(numero_buscado)
17
```



Exercícios em sala 1b - Solução

```
18    // Verificação (Busca Linear)
19    encontrado <- falso
20    para i de 1 ate 10 faca
21      se numeros[i] = numero_buscado entao
22        encontrado <- verdadeiro
23        posicao <- i
24        interrompa // Interrompe o laço assim que encontrar, otimizando o código
25      fimse
26    fimpara
27
28    // Saída do resultado
29    se encontrado = verdadeiro entao
30      escreval("O número ", numero_buscado, " ESTÁ contido na lista (posição:", posicao)
31    senao
32      escreval("O número ", numero_buscado, " NÃO está contido na lista.")
33    fimse
34  Fimalgoritmo
```