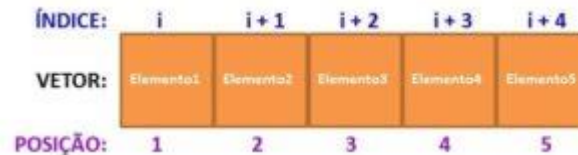


ESTRUTURAS UNIDIMENSIONAIS

O que é?

- São variáveis compostas que necessitam de apenas um índice para individualizar um elemento do conjunto. Essas variáveis também são chamadas de Vetores.



- A Figura ilustra um Array de uma dimensão com cinco posições.
- Cada quadrado desse Array é correspondente a um espaço de memória que armazena um dado. Como o Array tem cinco posições, então, são cinco espaços de memória que serão utilizados para armazenar esses dados

ÍNDICES

ÍNDICE:	i	i+1	i+2	i+3	i+4
VETOR:	Elemento1	Elemento2	Elemento3	Elemento4	Elemento5
POSIÇÃO:	1	2	3	4	5

O índice ajuda a percorrer o Array, permitindo que o elemento daquela determinada posição possa ser armazenado, acedido, atualizado, excluído, enfim, manipulado.

O índice pode andar da esquerda pra direita, por isso vocês veem escrito i , $i + 1$, e assim por diante. O índice i é a variável de incremento (contador) e a cada iteração que fizermos, ela é somada de um, para podermos caminhar no array.

Exemplo (algoritmo)

- Vamos ver um exemplo simples:
- armazenar dez números do tipo real.

```
1 algoritmo "vetores"
2
3 var
4     indice: inteiro
5     vet: vetor [1..10] de real
6
7 inicio
8
9     para indice de 1 ate 10 faca
10         escreva(" Digite um número real: ")
11         leia (vet[indice])
12     fimpara
13
14     escreval("")
15
16     para indice de 1 ate 10 faca
17         escreval(" | indice: | ", indice, " | elemento: | ", vet[indice])
18     fimpara
19
20 fimalgoritmo
```

Listagem 1: Armazenando e imprimindo dez números reais em um ,

VETOR:

1	2	3	4	5	6	7	8	9	10
5.6	8.7	1.2	9	9	10.7	1.5	81.7	123.87	2.4

Explicações (algoritmo)

```
8  
9  para indice de 1 ate 10 faca  
10     escreva(" Digite um número real: ")  
11     leia (vet[indice])  
12 fimpara
```

Nas linhas de 9 a 12 temos o código necessário para PREENCHER o vetor e usamos o comando de controle FOR para isso, assim podemos incluir um valor em cada posição do vetor.

O FOR vai ser repetido por dez vezes, conforme a linha 9 define:

para ÍNDICE de 1 até 10 (começando em 1 e terminando em 10).

Dentro do FOR colocamos uma mensagem para o usuário, pedindo que ele digite um valor, e o mesmo é armazenado pelo comando

Explicações (algoritmo)

```
16 para indice de 1 ate 10 faca
17     escreval(" | indice: | ", indice, " | elemento: | ", vet[indice])
18 fimpara
19
```

As linhas de 16 a 18 IMPRIMEM o conteúdo do ARRAY e podem verificar a forma como é impresso o conteúdo.

iteração	indice	para indice de 1 até 10	leia (vet [indice])
1	1	i = 1	vet[1] = 5.6
2	2	i = 2	vet[2] = 8.7
3	3	i = 3	vet[3] = 1.2
4	4	i = 4	vet[4] = 3
5	5	i = 5	vet[5] = 9
6	6	i = 6	vet[6] = 10.7
7	7	i = 7	vet[7] = 3.5
8	8	i = 8	vet[8] = 81.7
9	9	i = 9	vet[9] = 123.07
10	10	i = 10	vet[10] = 2.4
11	-	-	-

Inicializando o array

- Uma boa prática, sempre que trabalhar com Array, é inicializá-lo, o que significa inserir um valor nulo em todas as posições do Array. O código correspondente, para um vetor de números, é o seguinte:

```
1 algoritmo "vetores"
2
3 var
4     indice:inteiro
5     vet: vetor [1..10] de real
6
7 inicio
8
9     para indice de 1 ate 10 faca
10         vet[indice] <- 0
11     fimpara
12
13     escreval("")
14
15     para indice de 1 ate 10 faca
16         escreval(" | indice: | ", indice, " | elemento: | ", vet[indice])
17     fimpara
18
19 fimalgoritmo
```

Operações em array

- Se desejar aceder a uma posição específica do Array, deve indicar dentro do colchetes a posição desejada, exemplo: `vet[5]` é a posição 5 do vetor. Pode imprimir somente o conteúdo dessa posição, assim como pode substituir o valor que está armazenado.

```
1 algoritmo "vetores"
2
3 var
4     indice:inteiro
5     vet: vetor [1..10] de real
6
7 inicio
8
9     para indice de 1 ate 10 faca
10         escreval(" Digite um número real para a posição ", indice , " : ")
11         leia(vet[indice])
12     fimpara
13
14     escreval("")
15
16     escreval(" Elemento Original da Posição 5: ", vet[5])
17
18     vet[5] <- 567
19
20     escreval(" Elemento Modificado da Posição 5: ", vet[5])
21
22 fimalgoritmo
```

- A linha 16 imprime na tela o valor da posição 5 do vetor. A linha 18 altera o valor da posição 5 do vetor para 567. Por fim, a linha 20 imprime, novamente, na tela, o novo valor da posição 5.

Busca Sequencial

- Podemos buscar um valor desejado no Array. O código apresentado abaixo mostra como encontrar um elemento a partir de um valor digitado pelo usuário.
- Precisaremos de uma variável auxiliar, que chamei de **a**, que será iniciada com 0 para quando não encontrar o valor que se está procurando, ou 1, para quando o valor for encontrado. Usaremos um SE dentro do FOR para setar a variável **a** e encontrar o valor que se está procurando, e fora do FOR precisaremos de outro SE, para comparar o valor da variável **a** e verificar se o valor não foi encontrado.

```
1 var
2   indice, a: inteiro
3   vet: vetor [1..10] de real
4   x, soma, media : real
5
6 inicio
7
8   a <- 0
9
10  para indice de 1 ate 10 faca
11    escreval(" Digite um número real para a posição ", indice , " : ")
12    leia(vet[indice])
13  fimpara
14
15  para indice de 1 ate 10 faca
16    escreval(" | Índice: ", indice , " | Vetor [ índice ] : ", vet[indice])
17  fimpara
18
19  escreval(" Qual valor deseja procurar? Digite: ")
20  leia(x)
21
22  para indice de 1 ate 10 faca
23    se(vet[indice] = x)entao
24      escreval(" Valor encontrado na posição: ", indice)
25      a <- 1
26    fimse
27  fimpara
28
29  se ( a = 0 ) entao
30    escreval(" Valor NÃO encontrado ")
31  fimse
32
33 fimalgoritmo
```

Copiar os valores de um array

Copiar um Array para outro é bem fácil e simples, o código abaixo mostra como fazer isso. As linhas de 15 a 17 fazem a cópia do vetor 1 para o vetor 2.

```
1 algoritmo "vetores"
2
3 var
4     indice:inteiro
5     vet1: vetor [1..10] de real
6     vet2: vetor [1..10] de real
7
8 inicio
9
10    para indice de 1 ate 10 faca
11        escreval(" Digite um número real para a posição ", indice , " : ")
12        leia(vet1[indice])
13    fimpara
14
15    para indice de 1 ate 10 faca
16        vet2[indice] <- vet1[indice]
17    fimpara
18
19    para indice de 1 ate 10 faca
20        escreval(" | Índice: ", indice, " | Vetor 1: ", vet1[indice])
21        escreval(" | Índice: ", indice, " | Vetor 2: ", vet2[indice])
22    fimpara
23
24 fimalgoritmo
```