



## VETORES

Em programação existem várias estruturas que auxiliam na resolução de problemas. Podemos citar aqui a estrutura VETOR. Segundo Ascencio e Campos:

*Um vetor é uma variável composta homogênea unidimensional formada por uma sequência de variáveis, todas do mesmo tipo, com o mesmo identificador (mesmo nome) e alocadas sequencialmente na memória. Uma vez que as variáveis têm o mesmo nome, o que as distingue é um índice, que referencia sua localização dentro da estrutura. (ASCENSIO; CAMPOS, 2012, p. 131).*

Desta forma, podemos armazenar diversos valores em vetores, no entanto, todos com o mesmo tipo de dados. Cada valor pode ser armazenado em uma certa posição e é chamado de elemento do vetor.

O número de posições que pode conter em um vetor é especificado conforme a necessidade de um problema. Assim, um vetor de tamanho 5 armazenará 5 posições de valores do mesmo tipo de dado. Cada valor será armazenado em uma posição diferente e o que irá distinguir cada elemento do vetor é a sua posição. Cada posição é identificada por um valor inteiro, que é o seu índice e sequencialmente numerado.

Por exemplo, imaginemos um vetor com o nome X de 5 posições, disposto da seguinte forma:

1	2	3	4	5

Temos assim nosso vetor. Para realizar o acesso às suas posições devemos, então, nos utilizar dos índices. Assim, para eu atribuir um valor na posição 3, eu devo acessar a sua posição-identificação da seguinte forma:  $X[3]$ . Supondo, então, que eu queira adicionar os valores 20 e 40 nas posições 3 e 5, eu devo realizar os seguintes procedimentos:

$X[3] \leftarrow 20$

$X[5] \leftarrow 40$

E nosso vetor ficará da seguinte forma:

1	2	3	4	5
		20		40

Bom, agora que aprendemos a como acessar um índice em um vetor, como devemos declarar um vetor em um algoritmo? A sintaxe é bem simples. Para isso devemos fazer o seguinte:

*tipo identificador = vetor [LI .. LF] de tipo primitivo;*

Ou seja, devemos atribuir um nome identificador a este vetor, especificando a palavra vetor, identificando os seus limites de início e fim e indicando qual tipo de dado deve ser aceito. Assim, para o nosso exemplo, podemos declarar nosso vetor da seguinte forma:

Tipo X = vetor [1..5] de inteiros

X: vetor1, vetor2

Onde vetor1 e vetor2 são vetores do tipo X que possuem 5 posições cada um. Vetor assim declarado, podemos começar a realizar as adições de dados. Para adicionar um dado ao vetor, podemos realizar indicando o índice, como vimos anteriormente (`X[5] <- 40`), ou ainda, podemos nos utilizar da estrutura de repetição PARA (FOR).

Em uma implementação simples, podemos realizar da seguinte forma:

início

    tipo x = vetor [1..5] de inteiros;

    x: Vet;

    i: inteiro;

    para i de 1 até 5 passo 1 faça

        leia (Vet[i]);

    fimpara.

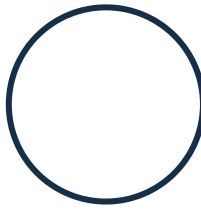
fim.

Trabalhar com vetores auxilia na programação, mas como toda estrutura, ela deve ser analisada de forma adequada para que não gere problemas no futuro. Em diversas linguagens de programação existe a implementação de vetores, para isso separamos três links interessantes para vetores na linguagem Java, C++ e Python. Vamos ler e adquirir novos conhecimentos?

Link 1: <https://www.devmedia.com.br/vetores-em-java/21449>

Link 2: <http://www.tiexpert.net/programacao/c/vetores.php>

Link 3: <https://www.devmedia.com.br/como-trabalhar-com-listas-em-python/37460>



## Vamos Praticar

Vetores são interessantes, não acha? Bom, agora que já aprendemos um pouco mais sobre esse recurso, vamos para a prática? Veja o estudo de caso abaixo:

Hélder está cursando o segundo semestre da disciplina de Lógica de Programação e seu professor passou um trabalho que deve ser entregue na próxima aula. O trabalho consiste em armazenar números sequenciais, sendo um total de 100 posições. Após o armazenamento, deverá ainda mostrar os números armazenados na tela. Para isso, Hélder deverá se utilizar de conceitos aprendidos em aula, como laços de repetição e vetores, que são requisitos do trabalho passado pelo professor.

Analisando o estudo de caso, vemos que Hélder deverá criar uma estrutura de dados do tipo vetor. Supondo que você seja Hélder, como você resolveria o problema do estudo de caso acima? Vamos para mais esse desafio? Ao final disponibilize seu trabalho no fórum da seção.

## Referências

ASCENCIO, A. F. G.; CAMPOS, E. A. V. de. **Fundamentos da Programação de Computadores**: Algoritmos, Pascal, C, C++ (padrão ANSI) e Java. 3. ed. São Paulo: Pearson, 2012.