



TAD - Um segundo exemplo

Nesta aula vamos implementar um **TAD Vetor** com algumas regras de negócio.

Um vetor, para quem não está lembrado, é um array de coluna (ou linha) única que armazena, em princípio, elementos do mesmo tipo, sendo que esse armazenamento é seqüencial e os elementos podem ser acessados pelo nome da variável vetor acompanhado do índice que dá a posição do elemento no vetor (o primeiro elemento tem índice 0).



Figura 1

A Figura 1 ilustra um vetor com um tamanho genérico de N posições sendo que cada posição pode armazenar um elemento ou estar vazia. Ou seja, o vetor, mesmo sem nenhum elemento armazenado, tem todas as posições alocadas na memória e disponíveis para uso.

Ler e escrever no vetor são operações de acesso direto e a busca por determinado elemento é implementada com o laço **for**, de forma que para essas operações não justifica a implementação de um **TAD Vetor**.

No entanto, vamos supor que nosso programa requeira um armazenamento em vetor que esteja submetido às seguintes regras:

Professor *Marcio Feitosa*



- O armazenamento é numérico do tipo **int**.
- Há um valor mínimo e um valor máximo suportados no vetor.
- Há um valor (fora do intervalo mínimo → máximo) que será armazenado nas posições vagas.
- Deve ter a opção de possibilitar números repetidos ou não permitir.
- Os métodos necessários são:
 1. Atribuir um valor a determinada posição.
 2. Alterar o valor de determinada posição.
 3. Remover o valor de determinada posição (atribui o valor indicativo de posição vaga).
 4. Ler o conteúdo de uma posição.
 5. Localizar um valor e retornar sua posição (se permitir repetição tem que retornar todos).
 6. Inserir na primeira posição vaga (busca no sentido 0 → N).
 7. Remover da última posição ocupada (busca no sentido 0 → N).
 8. Imprimir o conteúdo do vetor.

Vamos ver como isto funciona na prática!!

----- 10101010101010 -----