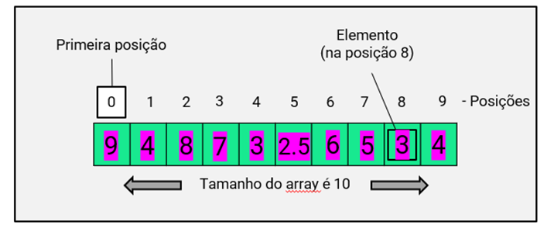
**Abstração de dados**

* As informações no computador estão armazenadas em células de memória com endereços específicos e muitos detalhes relacionados. Estruturas de dados são formas do cliente (programador, usuário, outro sistema) atuar com o sistema sem ficar preso aos detalhes do armazenamento da informação na memória do computador. Chamamos isso de “abstração de dados” que ganham forma nas estruturas de dados que são decisivas para que o programador expresse os algoritmos de forma mais eficiente.

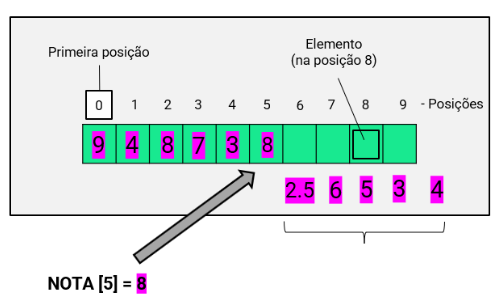
**Vetor (array):** coleção de elementos que aparecem sequencialmente. Permite acesso direto a qualquer elemento da lista, isto é, você não precisa percorrer todos os elementos de uma lista até encontrar o que você precisa, basta usar o índice da estrutura.

EX: Considere que a lista se chame LEITURAS.

LEITURAS [1] indica o conteúdo na posição 1, LEITURAS [2], a posição 2 e assim por diante. Essa forma de estrutura de dados em vetor permite acesso direto a uma determinada posição do vetor.

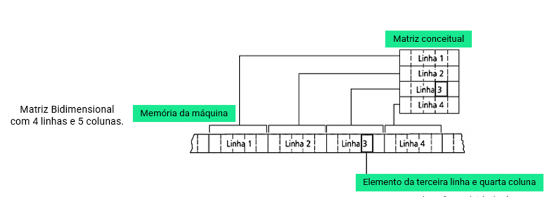


A inserção de um novo elemento irá forçar que os demais elementos sejam deslocados pela estrutura exigindo um laço de programação. Em outras palavras, a inserção de um elemento no vetor sem copiar por cima de um elemento pré-existente requer movimentar os elementos restantes da lista.



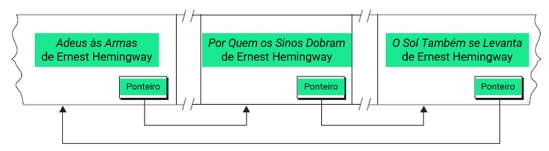
**Matriz:** consistem de vetores bidimensionais. Considere os dados como se estivesse todos dispostos em uma tabela, isto é, uma estrutura retangular contendo elementos de um mesmo tipo (no caso de matriz homogênea).

EX: simplificando bastante o processo, considerando uma estrutura de dados chamada LUGARES contendo a representação de todos os assentos do teatro contendo 20 fileiras de bancos e 15 assentos em cada fileira

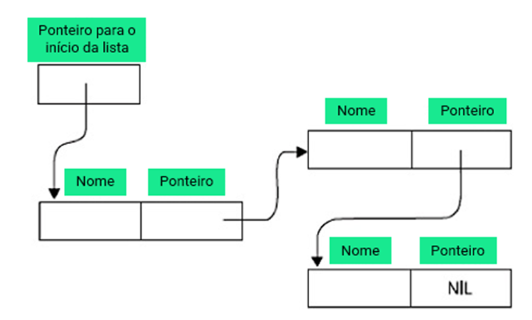


**Listas ligadas:** Se o armazenamento dos elementos da lista anterior estiver em blocos de letras espalhados pela memória e não “colados uns aos outros”, temos uma **lista ligada**. Tem importante diferença com relação a vetores porque geralmente será utilizada quando o tamanho da memória do computador a ser alocado não é conhecido anteriormentee mais memória deve ser alocada dinamicamente, isto é, durante a execução do próprio programa. Não permite acesso direto a qualquer elemento da lista (como em um vetor), isto é, você precisa percorrer todos os elementos de uma lista até encontrar o que você precisa. São muito usadas quando não se sabe o tamanho da memória requerida para uma determinada aplicação.

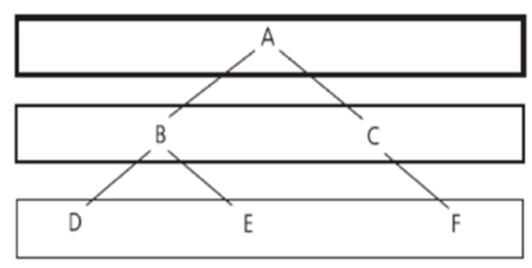
Assim, a lista é preferível ao vetor, quando houver necessidade de inserir e remover elementos. Já em um vetor, é necessário fazer muitos deslocamentos de nomes se for necessário preencher a lista com novos elementos. Mas a vantagem do vetor é o acesso direto. Quando queremos eliminar um elemento da lista, podemos simplesmente desviar a posição do ponteiro e nenhum elemento aponta para o elemento a ser eliminado. Isso equivale à sua eliminação.



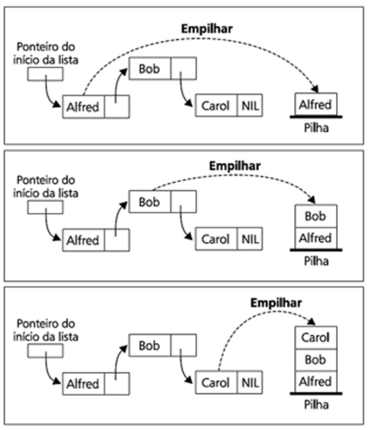
Visto de outra forma, a lista tem um início e cada elemento aponta para a localização do elemento seguinte (essa é a ideia do ponteiro representado por uma seta).



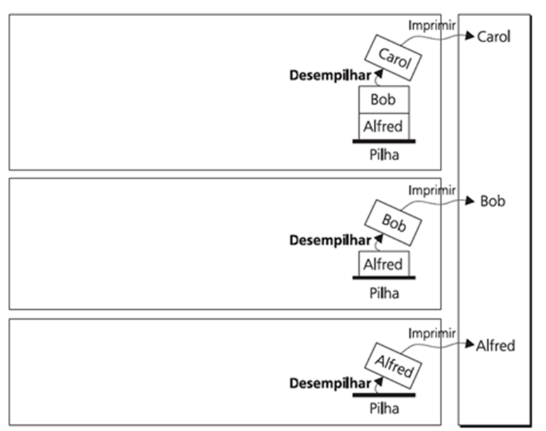
**Árvore:** a organização dos elementos se dá de forma hierárquica, existindo um elemento que fica no topo da árvore, chamado de raiz e os elementos subordinados a ele, seus nós filhos.

****

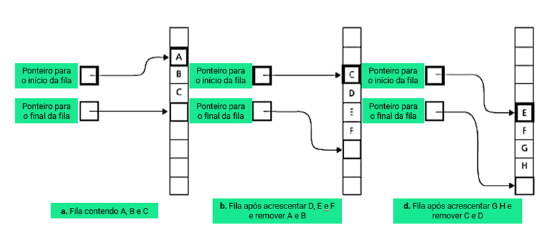
**Pilha:** Quando operamos uma estrutura em que as operações de inserção ou remoção ocorrem só em um dos extremos da lista, chamamos essa estrutura de uma pilha. A consequência dessa restrição é que **o último elemento a entrar na estrutura é o primeiro a ser removido**, isso leva as pilhas a serem conhecidas como estruturas “last-in, first-out – **LIFO**”. A extremidade em que as operações ocorrem é chamada de topo da pilha, a outra é chamada de base da pilha. O processo de inserir um objeto no topo da pilha é chamado de empilhamento (**push**), o de remover, operação de desempilhamento (**pop**).

****

Uma vez criada seus elementos podem ser descarregados, mas sempre a partir do topo da pilha:



**Filas:** as inserções são realizadas em uma extremidade e as remoções na outra. É o esquema first-in, first-out – FIFO. Outra forma de considerar fila é pensar que nela **os objetos são retirados de acordo com a ordem com que chegaram**. A extremidade na qual são retirados os elementos é chamada de início da fila, onde eles são inseridos é chamado de fim da fila.

****