

## INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ

CAMPUS TERESINA-CENTRAL
DIRETORIA DE ENSINO

## Estrutura de Dados Aula 1 - Listas

Professora: Elanne Cristina O. dos Santos

<u>elannecristina.santos@gmail.com</u> <u>elannecristina.santos@ifpi.edu.br</u>

## Lista Linear

- Forma simples de interligar os elementos de um conjunto.
- Agrupa informações referentes a um conjunto de elementos que se relacionam entre si de alguma forma.
- Inúmeros tipos de dados podem ser representados por listas. Alguns exemplos de sistemas de informação são: informações sobre os funcionários de uma empresa, notas de alunos, itens de estoque, etc.

## Lista Linear

- Lista linear é uma estrutura de dados na qual elementos de um mesmo tipo de dado estão organizados de maneira sequencial. Não necessariamente, estes elementos estão fisicamente em sequência, mas a idéia é que exista uma **ordem lógica entre eles.**
- Um exemplo disto seria um consultório médico:
  - as pessoas na sala de espera estão sentadas em qualquer lugar, porém sabe-se quem é o próximo a ser atendido, e o seguinte, e assim por diante.

## Lista Linear

- Assim, é importante ressaltar que uma lista linear permite representar um conjunto de dados afins (de um mesmo tipo) de forma a preservar a relação de ordem entre seus elementos. Cada elemento da lista é chamado de nó, ou nodo.
- A representação do nó ou nodo no programa pode ser dada através de uma estrutura do tipo:
- Vetor chamado de lista estática, pois em um vetor primitivo é necessário a definição prévia do tamanho do vetor.
- Ponteiros chamado de lista dinâmica.

### **LISTAS**

É uma das estruturas de dados mais empregadas no desenvolvimento de programas. É possível realizar as operações de manipulação de elementos em qualquer posição da lista.

#### **VETORES**

#### **ALOCAÇÃO ESTÁTICA**

Ocorre quando a quantidade total de memória utilizada pelos dados é previamente conhecida e definida de modo imutável.

#### **PONTEIROS**

#### **ALOCAÇÃO DINÂMICA**

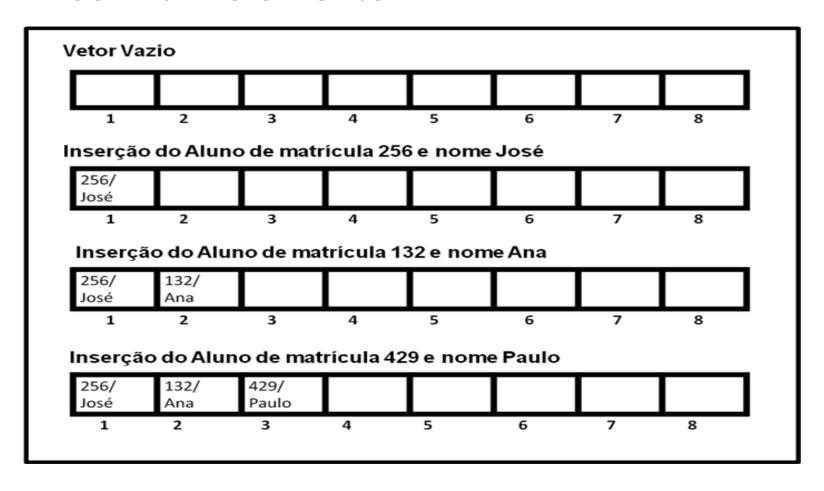
É o processo de solicitar e utilizar memória durante a execução de um programa. Ou seja, o programa aloca (SOLICITAR) espaço de memória quando há necessidade e desalocar quando houver necessidade de remoção de alguma informação.

## Lista Estática Desordenada

- Na lista desordenada os elementos são colocados na primeira posição vazia da lista (normalmente, no final).
   Na lista ordenada, é escolhido um dado que será o campo de ordenação da lista. Quando se deseja inserir um novo elemento na lista, primeiro tem que ser verificado em que local ele dever ser colocado para que seja mantida a ordem da lista.
- Operações básicas das listas: inserir elemento, remover elemento, consultar elemento, alterar elemento, listagem dos elementos da lista.

## Lista Estática Desordenada

Inserir um elemento

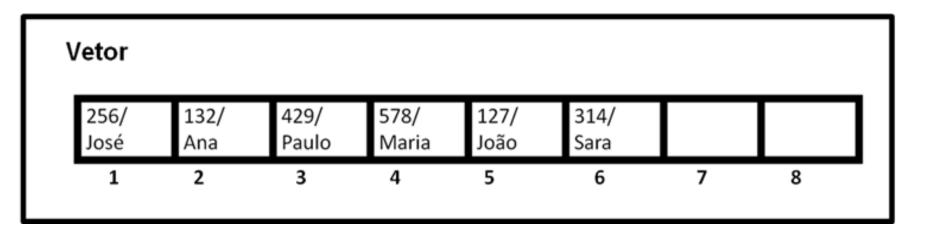


```
//Inserir novo aluno
typedef struct aluno{
   int mat;
   string nome;
}Aluno;
Aluno turma[30];
void inserir() {
  int qa=0; int cont;
  printf("\nInserir Novo Aluno\n\n");
  do {
    if (qa < maximo) { // verifica se o vetor pode receber novo aluno
      printf("\nMatricula do Aluno: ");
      scanf("%d",&turma[qa].mat);
      printf("\nNome: ");
      scanf("%s",turma[qa].nome);
      qa++;
      printf("\n\nAluno Inserido com Sucesso!!!\n\n");
```

```
else { // vetor cheio
    printf("\n\n Nao Pode Inserir - Turma Cheia!!!\n\n");
    break;
    }
    printf("\n\n Inserir outro(1-sim/2-nao)? ");
    scanf("%d",&cont);
    }while (cont == 1);
}
```

## Lista Desordenada Estática

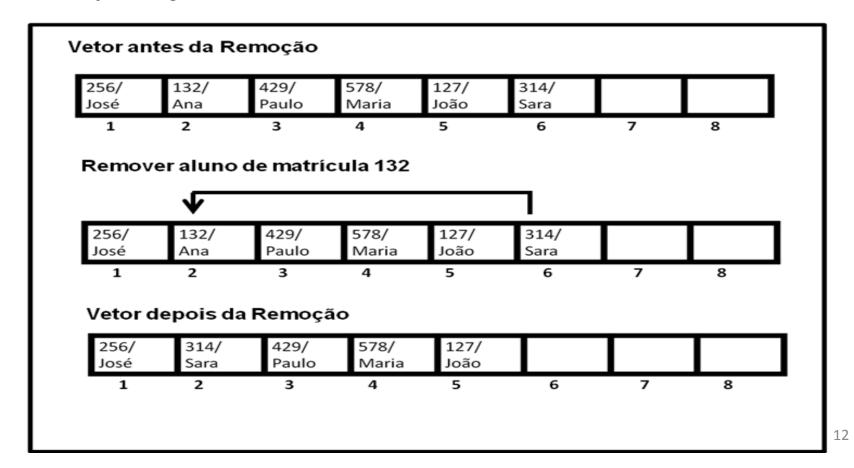
Consultar um elemento



```
void consultarmat() {
  int i, matprocurada, achou, cont;
  do {
    printf("\nConsultar Aluno por Matricula\n\n");
    printf("\nMatricula do Aluno: ");
    scanf("%d",&matprocurada);
    achou = procura(matprocurada);
    if (achou != -1)
     mostre(achou);
    else // aluno nao foi encontrado
     printf("\n\n\aNumero de Matricula Incorreto!!!!!\n");
    printf("\n\nConsultar outro(1-sim/2-nao)? ");
    scanf("%d",&cont);
  } while (cont == 1);
```

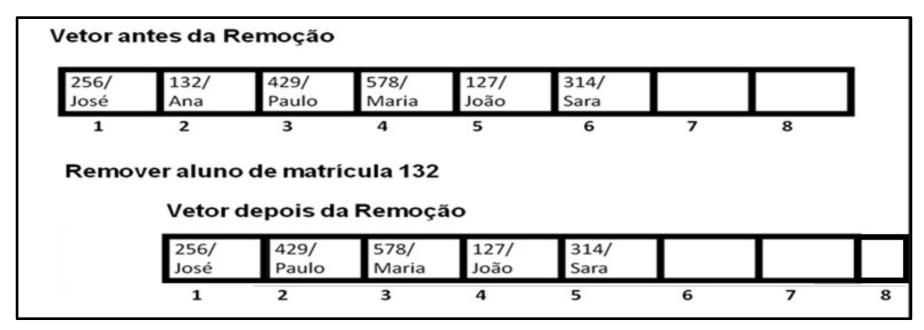
# Remover um elemento na lista estática

 Estratégia 1: coloca o ultimo elemento da lista na posição do elemento removido.



# Remover um elemento na lista estática

Estratégia 2: Uma outra estratégia seria mover
 TODOS os elementos que estão APÓS o elemento que deve ser removido UMA
 POSIÇÃO A FRENTE.



```
void remover() {
  int matprocurada, i, cont, achou, conrem;
  do{
    printf("\nRemover Aluno\n\n"); printf("\nMatricula do Aluno: ");
    scanf("%d",&matprocurada);
    achou = procura(matprocurada);
    if (achou != -1) {
      mostre(achou);
      printf("\nDeseja remover o aluno (1-sim/2-nao)? ");
      scanf("%d",&resp);
      if (resp==1) { // verifica se quer remover
        ???? COMPLETE AQUI O CÓDIGO PARA AS DUAS ESTRATÉGIAS
              DE REMOÇÃO???
       printf("\n\nAluno removido com Sucesso!!!\n");
                                                                 14
```

```
else
       printf("\n\n\aO aluno nao foi removido!!!\n");
      break;
 else // aluno nao foi encontrado
      printf("\n\naNumero de Matricula Incorreto!!!!!\n");
 printf("\n\nRemover outro(1-sim/2-nao)?");
 scanf("%d",&cont);
  }while (cont == 1);
```