



# Estrutura de Dados

Prof. Marques Sousa  
[marques.sousa@ifsp.edu.br](mailto:marques.sousa@ifsp.edu.br)

Este material consiste de adaptações e extensões dos originais  
gentilmente cedidos pelo Prof. Gilberto Viana - IFTM

Imagens retiradas do Google



# Na Aula Passada...

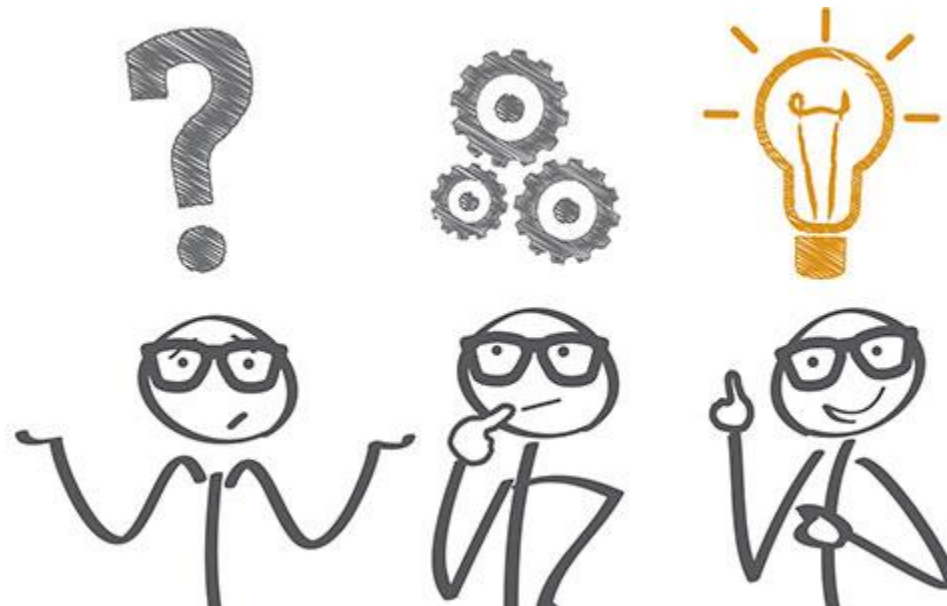
- Lista Linear
  - Implementação dinâmica.

- Lista Encadeada Ordenada
  - Tipos de ordenação



# Lista Encadeada

- Como é feita a busca na lista encadeada?
- Quais os critérios de parada para a busca na lista encadeada?
- Esse tipo de busca pode ser um problema?




# Lista Encadeada

- Formas para organizarmos nossa lista encadeada:
  - Método de auto organização:
    - Mover para frente;
    - Transposição;
    - Contagem.
  - Método de ordenação.
- Motivação: é possível melhorarmos a eficiência das nossas lista encadeadas!



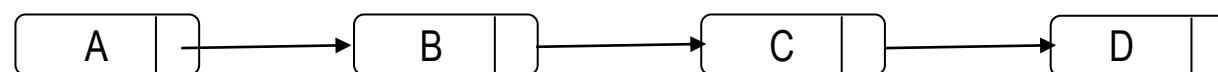
# Listas Auto Organizadas

- Ordenam os elementos em função das **pesquisas** realizadas durante o uso da aplicação.
  - Estes métodos buscam colocar os elementos mais prováveis de serem acessados no início da lista.
- 

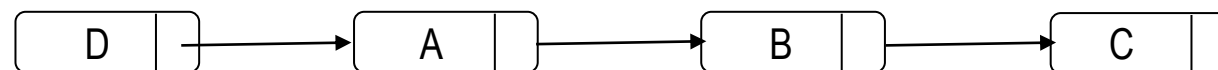


# Listas Auto Organizadas

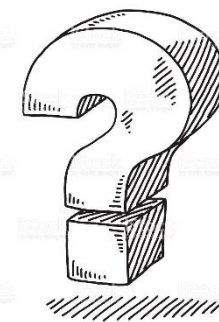
- **Método de mover para frente**
  - O elemento pesquisado vai para o início da lista.
- Desvantagem:
  - Uma única pesquisa pelo elemento não implica que o registro será frequentemente pesquisado.



**Elemento procurado → D**

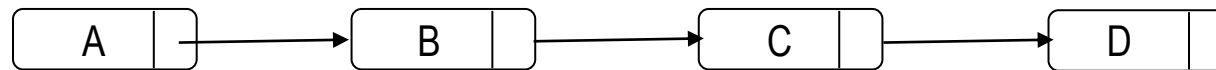


O custo desse método é o mesmo para uma implementação estática???

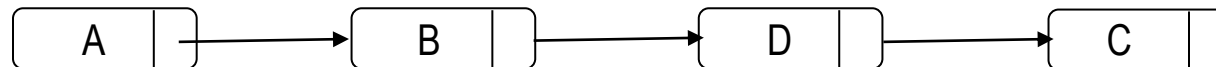




- **Método de transposição**
  - Sempre que encontrar o elemento, troque-o com seu predecessor, exceto se for o primeiro elemento.

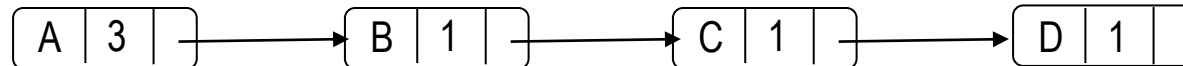


**Elemento procurado → D**

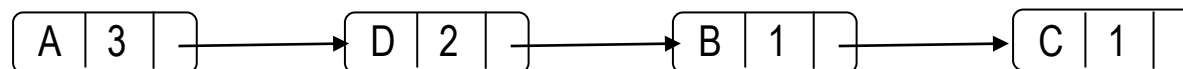


Qual a vantagem deste método se comparado ao método mover para frente?

- **Método de contagem**
  - Armazena um campo contagem, sempre que um elemento é pesquisado esse campo é incrementado.
  - A lista é ordenada pelo número de vezes que os elementos são acessados.



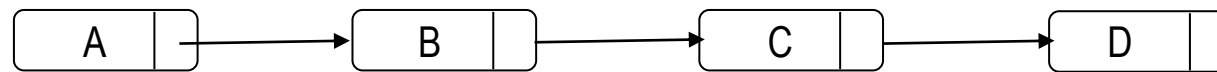
**Elemento procurado → D**



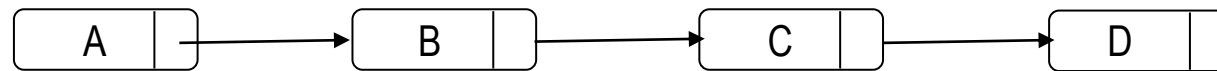
Caso o campo de contagem seja parte da informação, este método pode ser considerado como **método ordenado**.

- **Método de ordenação**

- A ordenação da lista depende do campo de informação. Ao contrário do método de contagem, neste tipo de ordenação qualquer campo pode ser escolhido. Por exemplo: menor nota, maior salário, ordem alfabética, etc.



**Elemento procurado → D**



- Ordena os valores em função do campo de conteúdo de seus nós (nas aulas anteriores, o campo *info*).
- Vantagem na busca pela informação, porém se a informação não se encontrar na lista, a pesquisa percorre até encontrar o seu fim.
  - Campo *next* igual a NULL.

# Listas Ordenada

- Insere um nó mantendo sempre a lista ordenada.
- Se a lista estiver vazia:
  - O ponteiro externo deve apontar para o nó a ser inserido.
- Senão:
  - Encontra o nó maior ou igual ao nó atual;
  - O novo nó aponta para o nó atual;
  - O nó anterior aponta para o novo nó.



**SIMPLES!**

# Lista Ordenada (Implementação)

- Inserir ordenado matrícula (Parte I).



# Lista Ordenada (Implementação)

- Inserir ordenado matrícula (Parte I).

```
int inserirOrdenado(CELULA **lista, PESSOA elemento){
    CELULA *nova, *aux, *anterior;

    nova = criarCelula();
    if(nova == NULL){
        printf("ERRO: Memoria cheia.\n");//se n alocou
        return 0;
    }

    nova->info = elemento;
    nova->next = NULL;

    aux = (*lista);
    //encontra o ponto de insercao
    while(aux != NULL){
        if(nova->info.matricula <= aux->info.matricula){
            break;
        }
        anterior = aux;
        aux = aux->next;
    }
}
```

# Lista Ordenada (Implementação)

- Inserir ordenado matrícula (Parte II).

```
//prox. da nova celula aponta para o maior que ela
nova->next = aux;

//neste caso, aux é a primeira.
if(aux == (*lista)){
    (*lista) = nova;
}else{
    anterior->next = nova;
}
return 1;
}
```



1. Considere uma lista encadeada com os seguintes elementos inteiros:

6, 2, 3, 13, 56, 13, 77, 88, 2, 78, 90, 1, 5, 133.

Realize as seguintes operações nesta lista. Desenhe a lista após cada operação.

a - PesqMoveParaFrente(90)

b - PesqTransposicao(56)

c - PesqTransposicao(5)

d - PesqTransposicao(56)

e - PesqMoveParaFrente(133)

f - PesqContagem(5)

g - PesqTansposicao(5)

h - PesqTransposicao(90)

i - PesqContagem(78)

j - PesqContagem(78)

k - PesqTransposicao(78)

l - PesqMoverParaFrente(2)

# Na Próxima Aula...

Implementação de um tipo de lista ordenada 😊





# Estrutura de Dados

Prof. Marques Sousa  
[marques.sousa@ifsp.edu.br](mailto:marques.sousa@ifsp.edu.br)