

# ESTRUTURA DE DADOS

Prof.<sup>a</sup> Priscilla Abreu

[priscilla.braz@rj.senac.br](mailto:priscilla.braz@rj.senac.br)



# Estrutura de dados



## Roteiro de Aula

- Objetivo da aula
- Listas
- Listas Lineares
- Lista linear sequencial

# LISTAS

# Estrutura de dados



## LISTA

### O QUE É UMA LISTA?

**Consideraremos  
como listas  
conjuntos sem  
repetições!**

Uma lista é um conjunto de dados relacionados, e de número variável de elementos.

- Exemplo:
  - Lista de alunos de uma turma;
  - Lista de aprovados em um concurso;
  - Lista de produtos de uma loja;
  - ...

# Estrutura de dados



## LISTA

### Operações comuns:

- Busca;
- Inserção;
- Remoção.

### Outras operações:

- Alteração de um elemento da lista;
- Ordenação dos elementos da lista segundo uma determinada chave;
- Determinação do primeiro elemento da lista, ...

## LISTA LINEAR

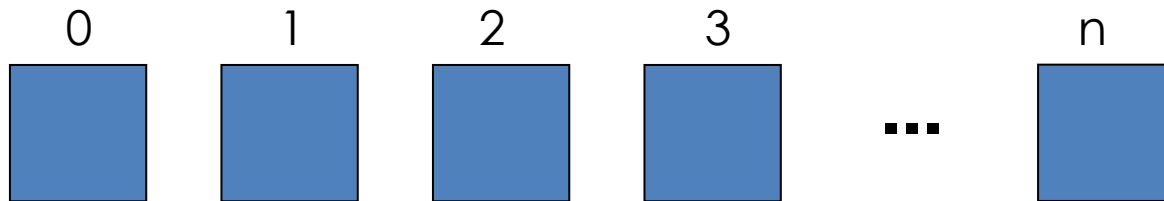
Estrutura que permite representar um conjunto de dados de forma a preservar a relação de ordem existente entre eles.

Conjunto de  $n \geq 0$  nós  $L[0]$ ,  $L[1]$ , ...,  $L[n-1]$ , cuja propriedade estrutural envolve as posições relativas de seus elementos.

# Estrutura de dados



## LISTA LINEAR



- Se  $n \geq 0$ ,  $L[0]$  é o primeiro elemento;
- Para  $0 < k < n$ , o nó  $L[k]$  é precedido por  $L[k-1]$ ;
- $L[n]$  é o último elemento.
- Se  $n < 0$ , então a lista está vazia.

## LISTA LINEAR

### Listas lineares

#### Listas lineares gerais

SEM restrição de inserção e remoção de elementos

#### Listas particulares

COM restrição de inserção e remoção de elementos



# Estrutura de dados



## LISTA LINEAR

### Casos particulares:

- Deque  
Inserção e remoção apenas nas extremidades;
- Pilha  
Inserção e remoção apenas em um extremo
- Fila  
Inserção em um extremo e remoção em outro extremo;

## LISTA LINEAR: TIPO DE ARMAZENAMENTO

O tipo de armazenamento de uma lista linear pode ser classificado de acordo com a posição relativa na memória (contínua ou não) de cada dois nós consecutivos na lista.

Existem dois tipos de listas:

- Lista sequencial
- Lista encadeada

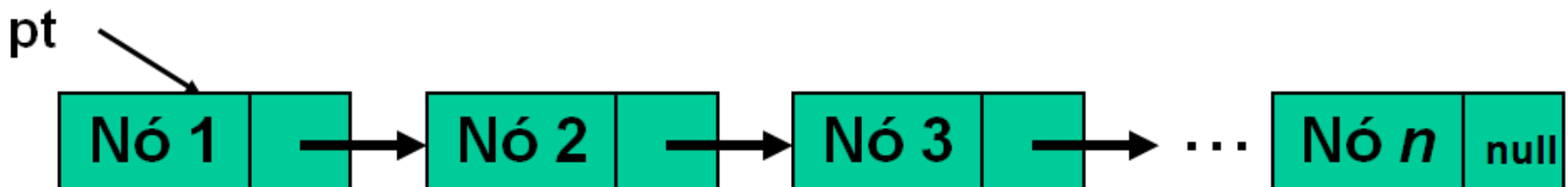
## LISTA LINEAR: TIPO DE ARMAZENAMENTO

Lista sequencial: lista linear com alocação estática de memória, em que cada elemento da lista ocupa posição física na memória sucessiva ao elemento anterior.



## LISTA LINEAR: TIPO DE ARMAZENAMENTO

Lista encadeada: lista linear com alocação dinâmica de memória, em que as posições de memória são alocadas a medida que são necessárias. Elementos da lista encontram-se aleatoriamente dispostos na memória e são interligados por ponteiros, que indicam a localização do próximo nó.



# LISTAS LINEARES SEQUENCIAIS

## LISTA LINEAR SEQUENCIAL

- Forma mais simples de se manter uma lista na memória: nós em posições contínuas.
- Uso de vetores.



- M é quantidade máxima de elementos que a lista poderá armazenar.
- **n representa a última posição inserida.**

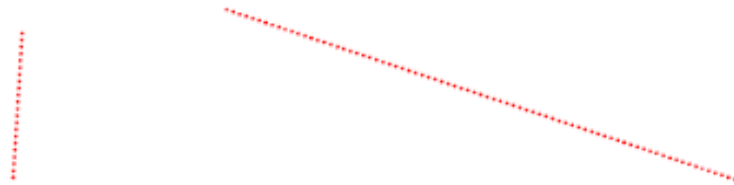
# Estrutura de dados



## LISTA LINEAR SEQUENCIAL

- Uso de registros / estruturas

<b>Nó 1</b>	<b>Nó 2</b>	<b>Nó 3</b>	<b>Nó 4</b>	<b>Nó 5</b>
-------------	-------------	-------------	-------------	-------------



<b>Chave</b>	<b>Nome</b>	<b>Endereço</b>
--------------	-------------	-----------------

# Estrutura de dados



## LISTA LINEAR SEQUENCIAL

```
#include <stdio.h>
#define M 100
// FUNÇÕES
int main(){
    int L[M];
    int n=-1, op, x;
    ...
}
```



# Estrutura de dados



## LISTA LINEAR SEQUENCIAL

- Operações
  - Busca
    - Sequencial
    - Ordenada
    - Binária
  - Inserção
  - Remoção

# Como exibir a lista???

# Estrutura de dados



## LISTA LINEAR SEQUENCIAL

```
void imprimir(int n, int *L){
    int i;
    if (n>=0){
        printf("\nIMPRESSÃO DA LISTA\n");
        for(i=0; i<=n;i++){
            printf("%d ",L[i]);
        }
        printf("\n");
    }
    else{
        printf("\nLista vazia!\n");
    }
}
```

# Como buscar um elemento na lista???

# Estrutura de dados



## LISTA LINEAR SEQUENCIAL

### Busca

- Lista  $[0..n]$
- $x \Rightarrow$  elemento procurado
- Função retorna:
  - índice do nó desejado;
  - -1, caso o elemento não esteja na lista

# Estrutura de dados



## LISTA LINEAR SEQUENCIAL

### Busca

```
int busca( int x, int n, int *L) {  
    int i=0;  
    while (i<= n){  
        if (L[i] == x) {  
            return i;  
        }  
        i++;  
    }  
    return -1;  
}
```

# Como inserir um elemento na lista???

# Estrutura de dados



## LISTA LINEAR SEQUENCIAL

### Inserção de um elemento na lista

```
void inserir(int x, int *n, int *L){
```

```
    ????
```

```
}
```



# Estrutura de dados



## LISTA LINEAR SEQUENCIAL

Valor a ser  
inserido

### Inserção de um elemento na lista

x

16

n



0	1	2	3	4	5	6	7	8	9
15	52	36	48						

# Estrutura de dados



## LISTA LINEAR SEQUENCIAL

Tem espaço disponível?

### Inserção de um elemento na lista

x

16

M = 10

**n**  
↓

0	1	2	3	4	5	6	7	8	9
15	52	36	48						

# Estrutura de dados



## LISTA LINEAR SEQUENCIAL

### Inserção de um elemento na lista

O valor a ser inserido  
já está cadastrado na  
lista?

x

16

**n**  
↓

0	1	2	3	4	5	6	7	8	9
15	52	36	48						

# Estrutura de dados



## LISTA LINEAR SEQUENCIAL

`n++;`

### Inserção de um elemento na lista

x

16

n



0	1	2	3	4	5	6	7	8	9
15	52	36	48						

# Estrutura de dados



## LISTA LINEAR SEQUENCIAL

$L[n] = x;$

### Inserção de um elemento na lista

x

16

n



0	1	2	3	4	5	6	7	8	9
15	52	36	48	16					

## LISTA LINEAR SEQUENCIAL – INSERÇÃO

```
void inserir(int x, int *n, int *L){  
    if (*n < M-1) {  
        if (busca(x,*n,L) == -1){  
            *n = (*n)+1;  
            L[*n]=x;  
        }  
        else  
            printf("\nElemento já existe\n");  
    }  
    else  
        printf("\nLista cheia!\n");  
}
```

# Como remover um elemento na lista???

# Estrutura de dados



## LISTA LINEAR SEQUENCIAL

### Remoção de um nó na lista L

Remoção não ocorre de fato



**simulação**



# Estrutura de dados



## LISTA LINEAR SEQUENCIAL

Valor a ser  
removido

### Remoção de um elemento na lista

x

36

**n**  
↓

0	1	2	3	4	5	6	7	8	9
15	52	36	48	16					

# Estrutura de dados



## LISTA LINEAR SEQUENCIAL

A lista está vazia?

### Remoção de um elemento na lista

x

36

n



0	1	2	3	4	5	6	7	8	9
15	52	36	48	16					

# Estrutura de dados



## LISTA LINEAR SEQUENCIAL

### Remoção de um elemento na lista

O valor a ser removido encontra-se na lista?

x

36

**n**  
↓

0	1	2	3	4	5	6	7	8	9
15	52	36	48	16					

# Estrutura de dados



## LISTA LINEAR SEQUENCIAL

### Remoção de um elemento na lista

O valor a ser removido encontra-se na lista?

x

36

**n**  
↓

0	1	2	3	4	5	6	7	8	9
15	52	36	48	16					

↑  
**índice**

# Estrutura de dados



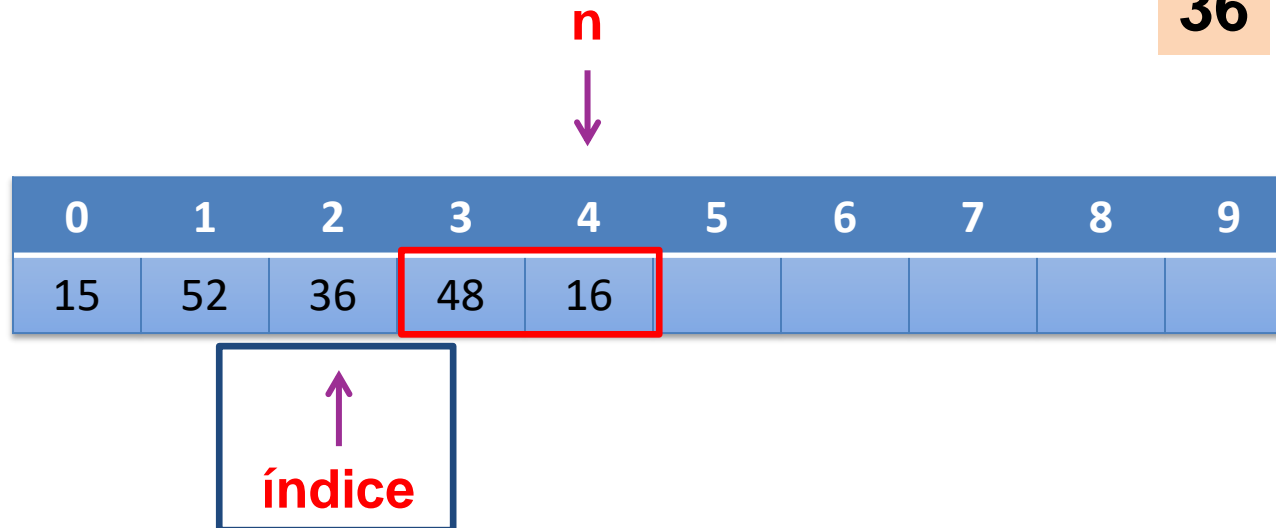
## LISTA LINEAR SEQUENCIAL

### Remoção de um elemento na lista

Vamos arrastar os elementos que estão após o valor a ser removido para à esquerda.

x

36



# Estrutura de dados



## LISTA LINEAR SEQUENCIAL

### Remoção de um elemento na lista

Vamos arrastar os elementos que estão após o valor a ser removido para à esquerda.

x

36

0	1	2	3	4	5	6	7	8	9
15	52	48	48	16					

↑  
índice

# Estrutura de dados



## LISTA LINEAR SEQUENCIAL

### Remoção de um elemento na lista

Vamos arrastar os elementos que estão após o valor a ser removido para à esquerda.

x

36

0	1	2	3	4	5	6	7	8	9
15	52	48	16	16					

↑  
índice

# Estrutura de dados



## LISTA LINEAR SEQUENCIAL

### Remoção de um elemento na lista

Elementos arrastados!  
Precisamos ajustar o  
valor de  $n$

x

36

0	1	2	3	4	5	6	7	8	9
15	52	48	16	16					

índice

Último valor da lista  
foi passado para a  
posição  $n-1$ !



# Estrutura de dados



## LISTA LINEAR SEQUENCIAL

### Remoção de um elemento na lista

n--

**n**  
↓

0	1	2	3	4	5	6	7	8	9
15	52	48	16	16					

# Estrutura de dados



## LISTA LINEAR SEQUENCIAL

```
void remover(int x, int *n, int *L){
    int i, indice;
    if (*n >= 0) {
        indice= busca(x,*n,L);
        if (indice != -1){
            for (i=indice; i<=*(n-1); i++)
                L[i]=L[i+1];
            (*n)--;
        }
        else
            printf("\nElemento não existe\n");
    }
    else
        printf("\nLista vazia!\n");
}
```

# AMPLIANDO O PROGRAMA...

## LISTA LINEAR SEQUENCIAL

Falta desenvolver a função main para que as funções implementadas de listas sequenciais possam ser utilizadas, de acordo com o que o usuário desejar fazer de operação na lista.

# Estrutura de dados



## LISTA LINEAR SEQUENCIAL

```
int main(){
    int L[M], n=-1, op,x;
    do{
        system("cls");
        printf("\n*****LISTA SEQUENCIAL*****\n");
        printf("1- Inserir valor\n");
        printf("2- Buscar valor\n");
        printf("3- Remover valor\n");
        printf("4- Exibir lista\n");
        printf("5- Sair\n");
        printf("Escolha sua opção: ");
        scanf("%d",&op);
```

# Estrutura de dados



## LISTA LINEAR SEQUENCIAL

```
switch(op){  
    case 1:{  
        printf("Informe o valor a ser inserido: ");  
        scanf("%d",&x);  
        inserir(x,&n,L);  
        break;  
    }
```

## LISTA LINEAR SEQUENCIAL

```
case 2:{
    int res;
    printf("Informe o valor a ser encontrado: ");
    scanf("%d",&x);
    res=busca(x,n,L);
    if(res!=-1){
        printf("\nValor %d encontrado na posição
%d.\n",L[res],res);
    }
    else{
        printf("\nValor não encontrado!\n");
    }
    break;
}
```

# Estrutura de dados



## LISTA LINEAR SEQUENCIAL

```
case 3:{  
    printf("Informe o valor a ser removido: ");  
    scanf("%d",&x);  
    remover(x,&n,L);  
    break;  
}
```



# Estrutura de dados



## LISTA LINEAR SEQUENCIAL

```
        case 4:{
            imprimir(n,L);
            break;
        }
    }
    system("Pause");
}while(op!=5);
}
```

# DÚVIDAS???



# PRÓXIMA AULA

# SEMINÁRIO: APRESENTAÇÃO DOS GRUPOS

## SEMINÁRIO

- Cada grupo terá até 40min. para apresentar seu seminário e realizar algum exercício com a turma;
- Ordem de apresentação:
  - Métodos de Busca;
  - Método Bubblesort;
  - Método Selection sort;
  - Método Inserção Direta;
  - Método Quicksort.

# Estrutura de dados



## ATIVIDADE AVALIATIVA 2

Após a apresentação de todos os métodos, cada grupo deverá ainda elaborar um sistema completo que implemente os métodos de busca e de ordenação vistos em sala de aula.

Descrição do sistema a seguir...

**ENTREGA FINAL ATÉ: 07/05**

## ATIVIDADE AVALIATIVA 2

Elabore um programa que, considerando uma lista linear de números inteiros (máximo de 20 valores), apresente as seguintes opções de operações para o usuário:

- 1- Inserir valor
- 2- Buscar valor:
  - a) busca sequencial
  - b) busca binária
- 3- Ordenar o vetor:
  - a) Bubblesort
  - b) Selection sort
  - c) Insertion sort
  - d) Quicksort
- 4- Exibir lista
- 5- Remover elemento
- 6- Sair

**A atividade pode ser feita em grupo e deve ser enviada utilizando o link disponibilizado no espaço do Moodle.**

**ENTREGA PARCIAL:  
24/04**