

**INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
RIO GRANDE DO SUL
Câmpus Feliz

Estruturas de Dados

Conceitos e Listas Lineares

Prof. Moser Fagundes

Programação II
Técnico em Informática

Algoritmos

- Os **algoritmos** podem ser visto como o "coração" da computação.
 - Maioria das subáreas implementam algoritmos
 - Um programa é uma materialização de uma ideia, e essa ideia é o algoritmo.
- Os algoritmos manipulam **dados** e a sua organização é crucial na criação de programas eficientes.

O que é estrutura de dados?

O que é estrutura de dados?

Na computação, uma estrutura de dados é um modo de armazenamento e organização de dados tal que estes dados possam ser usados eficientemente.

Estruturas de dados

Exemplos

- Listas Lineares
 - Listas Sequenciais
 - Listas Encadeadas
- Pilhas
- Filas
- Deques
- Árvores

Estruturas de dados

Exemplos

- Listas Lineares
 - Listas Sequenciais
 - Listas Encadeadas
- Pilhas
- Filas
- Deques
- Árvores

Listas Lineares (LL)

Uma **Lista Linear (LL)** é um agrupamento de informações sobre um conjunto de elementos que se relacionam entre si.

Exemplos:

- Notas de alunos de uma turma
- Nomes de pessoas
- Quantidade de produtos em um estoque
- Objetos da classe Filme (ou classe Pessoa)

Listas Lineares (LL)

Operações básicas sobre uma Lista Linear:

- Busca de elemento
 - Não altera a lista
 - Resposta da busca é sim ou não (a resposta também poderia ser o índice do elemento buscado)
- Inserção de um elemento
- Remoção de um elemento

Classificações de LL

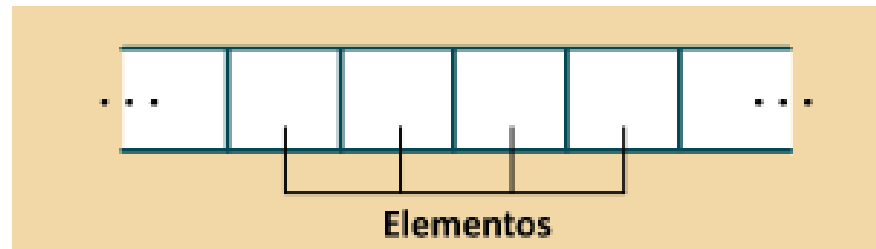
Podemos **classificar** uma LL de acordo com diferentes critérios. São eles:

1. Armazenamento na memória
2. Modo como elementos são inseridos/removidos
3. Indexação de elementos

Classificação quanto ao armazenamento

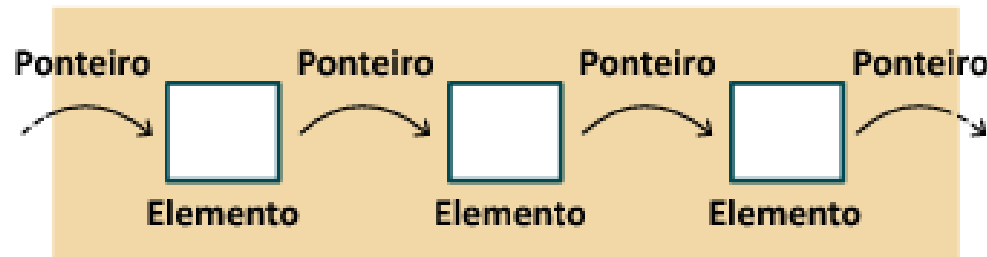
Lista Sequencial

Os elementos ocupam posições contíguas na memória.



Lista Encadeada

Os elementos encontram-se **dispersos** pela memória, sendo ligados por variáveis (*ponteiros*).



Classificação quanto ao modo de inserção e remoção de elementos

- **Listas** (em geral)

Inserções e remoções são permitidas em qualquer posição (por exemplo, um ArrayList de objetos).

- **Deque**

Inserções e remoções são permitidas apenas nas extremidades da lista.

- **Pilhas**

Inserções e remoções são permitidas somente em uma das extremidades (a outra permanece fixa [fundo da pilha]).

- **Fila**

Inserções são realizadas em uma extremidades, e as remoções são feitas na outra (processos, fila de impressão).

Classificação quanto ao modo de indexação

Esta classificação é tradicionalmente usada nas listas sequenciais.

Podemos classificar uma lista sequencial em:

- Vetor**
- Matriz**

Busca linear em LL

```
public static int busca(int[] valores, int valorProcurado) {  
    int i = 0;  
    while(i < valores.length) {  
        if(valores[i] == valorProcurado) {  
            // Retorna o indice o elemento encontrado  
            return i;  
        } else {  
            i++;  
        }  
    }  
  
    // Retorna -1 se nao encontrar  
    return -1;  
}
```

Busca em LL ordenadas

Supondo que a lista linear está ordenada em ordem crescente, como podemos otimizar o algoritmo de busca do slide anterior?

Em Java podemos ordenar um array de inteiros usando o método `Arrays.sort()`:

```
int[] valores = {2,44,23,45,39,89,17,12,76};  
Arrays.sort(valores);
```

Busca Binária

O que é uma busca binária?

Exercício

Implemente um algoritmo de **busca binária** para listas ordenadas.

Usando uma lista com um grande número de elementos ordenados, compare o tempo de busca gasto pelo algoritmo de busca sequencial com o tempo gasto pelo algoritmo de busca binária.