

# LISTAS

## DCE792 - AEDs II (Prática)

Atualizado em: 26 de agosto de 2024

Iago Carvalho

Departamento de Ciência da Computação



Estrutura de dados mais básicas existente em qualquer linguagem de programação

- Estrutura de dados linear
  - Seus elementos são salvos em pedaços contíguos de memória
  - Acesso pode ser realizado de forma aleatória (e rápida) utilizando índices
  - É possível guardar elementos homogêneos, i.e., do mesmo tipo
- 
- |                                 |   |
|---------------------------------|---|
| ○ Acesso aleatório ↑            | ○ Tamanho fixo ↓                            |
| ○ Fácil de ordenar elementos ↑  | ○ Difícil de inserir ou remover elementos ↓ |
| ○ Simples de trocar variáveis ↑ | ○ Desperdício de memória ↓                  |

# EXEMPLOS DE VETORES

```
int numeros[5];
```

```
char palavra[10];
```

```
float numeros_reais[8];
```

Uma lista simples é muito parecida com um vetor

- Estrutura de dados linear
- Seus elementos são salvos em pedaços contíguos de memória
- Acesso pode ser realizado de forma aleatória (e rápida) utilizando índices
- É possível guardar elementos homogêneos, i.e., do mesmo tipo

Teoricamente, só é possível inserir elementos no início e no fim da lista

Declarada como `<tipo> <nome>[<tamanho>];`

Operações possíveis em listas

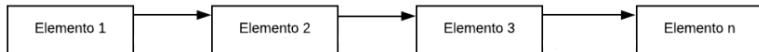
- `void inserirFrente(<valor>);`
- `void inserirAtras(<valor>);`
- `void removeFrente(void);`
- `void removeAtras(void);`
- `<tipo> obtemPrimeiro(void);`
- `<tipo> obtemUltimo(void);`
- `bool ehVazia(void);`

Estrutura sem nenhuma utilidade prática

- É melhor utilizar vetores

Uma lista implementada utilizando ponteiros

- Tamanho dinâmico
- Não possui desperdício de memória



O código completo de listas encadeadas está disponível no Github



► Link

# ATIVIDADE PRÁTICA



Modificar o código do Github para incluir três novos método

O primeiro método deve permitir inserir um novo item na  $i$ -ésima posição

- `void inserirPosicao(<valor>, <posicao>);`

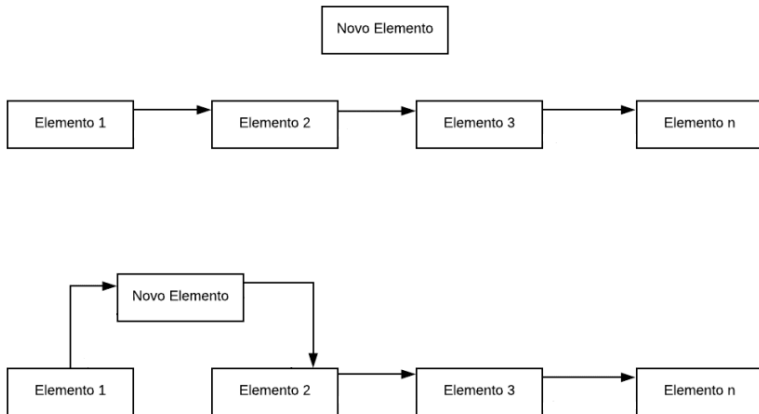
O segundo método deve permitir remover um item da  $i$ -ésima posição

- `void removerPosicao(<posicao>);`

O terceiro método deve permitir obter o valor do  $i$ -ésimo item

- `<tipo> obterPosicao(<posicao>);`

# ATIVIDADE - INSERIR



PRÓXIMA AULA:

LISTAS DUPLAMENTE  
ENCADEADAS