

# **Estruturas de Dados**

## **Trabalho Prático de implementação – Listas encadeadas**

### **1. Regras**

- O trabalho deverá ser desenvolvido individualmente, em linguagem C.
- O trabalho vale 25 pontos.
- Trabalho copiado vale 0(zero).

O professor poderá realizar arguição individual sobre o trabalho a fim de averiguar a compreensão de cada aluno sobre o trabalho realizado podendo a nota do trabalho ser substituída pela nota da arguição.

### **2. Introdução:**

Listas encadeadas ordenadas são estruturas de dados lineares em que as operações de busca exigem, nos piores caso,  $n$  comparações (onde  $n$  é o número de elementos da lista). Diversas outras estruturas de dados foram criadas para melhorar essa característica, como, por exemplo, árvores binárias, árvores B, tabelas hash, etc. Nesse curso ainda veremos árvores binárias. Há duas implementações clássicas de listas: simplesmente encadeadas e duplamente encadeadas. Nesse trabalho você deverá desenvolver um programa que se utilize de listas encadeadas ordenadas para implementar uma agenda de telefones. Você pode escolher entre implementar simplesmente ou duplamente encadeada.

### **3. Requisitos Funcionais:**

Desenvolver um programa de agenda de contatos. Cada registro de sua agenda deve armazenar o nome e o número de telefone de um contato.

Seu programa deve prover as seguintes funcionalidades:

- Incluir contato/telefone: o programa deve solicitar o nome do contato a ser incluído.
  - Caso já exista um contato com o nome informado, o sistema deve informar que já existe um contato com esse nome e exibir o telefone dele.
  - Caso não exista um contato com o nome dado, o sistema deve solicitar o telefone do contato e criar um novo contato com o nome e o telefone dado.

Dica: mantenha a lista de contatos já ordenada pelo nome ao inserir o contato. Isso facilitará a implementação do próximo requisito.

- Exibir Agenda Inteira: O programa exibirá os dados (nome e telefone) de todos os contatos da agenda ordenados alfabeticamente pelo nome.
- Buscar contato por nome: o programa solicita o nome do contato.
  - Caso exista um contato com o nome informado, o programa deve exibir o telefone do mesmo;
  - Caso não exista um contato com o nome informado o programa deve exibir uma mensagem falando que não existe contato com o nome dado.
- Buscar contato por telefone: o usuário informa um número de telefone. Se o número informado for telefone de algum contato o sistema exibe o nome do contato. Caso contrário exibe uma mensagem informando que o número é desconhecido.
- Excluir contato por nome: o programa solicita o nome do contato.
  - Caso exista um contato com o nome informado, o programa deve excluir o contato e exibir uma mensagem informando que o contato foi excluído com sucesso.
  - Caso não exista um contato com o nome informado o programa deve exibir uma mensagem falando que não existe contato com o nome dado.

#### **4. Requisitos não funcionais:**

- Faça uma biblioteca contendo as estruturas e métodos referentes a manipulação de listas.

#### **5. Critérios de avaliação**

Além da adequação aos requisitos acima listados, serão considerados na avaliação:

- Modularidade do código (divisão do código em funções e procedimentos);
- Qualidade dos comentários;
- Indentação do código.