

## ***Projeto Integrador III-A (Recuperação)***

### ***1. Introdução***

Este documento descreve o projeto de um sistema de gerenciamento de pacientes em um hospital, desenvolvido como projeto integrador para as disciplinas de "Programação Orientada a Objetos" e "Estrutura de Dados". O sistema utiliza a estrutura de dados lista encadeada para armazenar e manipular as informações dos pacientes.

### ***2. Descrição do Problema***

O problema consiste em desenvolver um sistema que permita o gerenciamento de pacientes em um hospital. O sistema deve ser capaz de:

Inserir novos pacientes: registrar as informações básicas de um novo paciente, como nome, idade e doença.

Remover pacientes: remover um paciente da lista, simulando a alta do paciente.

Buscar pacientes: encontrar um paciente específico na lista pelo nome.

Imprimir a lista de pacientes: exibir a lista completa de pacientes cadastrados.

Ordenar a lista de pacientes por nome: organizar a lista em ordem alfabética de nome.

### ***3. Implementação da Solução***

A solução foi implementada em Python, utilizando o paradigma de orientação a objetos e a estrutura de dados lista encadeada.

#### ***3.1. Classes***

O sistema é composto por duas classes principais:

Paciente: representa um paciente, armazenando suas informações básicas (nome, idade e doença) e um ponteiro para o próximo paciente na lista.

ListaEncadeada: implementa a lista encadeada, fornecendo métodos para inserir, remover, buscar, imprimir e ordenar os pacientes.

#### ***3.2. Lista Encadeada***

A lista encadeada é uma estrutura de dados dinâmica que permite a inserção e remoção de elementos de forma eficiente. Cada elemento da lista é um nó que contém um dado (no caso, um objeto da classe Paciente) e um ponteiro para o próximo nó da lista.

#### ***3.3. Interface do Usuário***

O sistema possui uma interface de linha de comando que permite ao usuário interagir com as funcionalidades do sistema através de um menu de opções.

## 4. Código Fonte

[GitHub](#)

## 5. Execução do Sistema

Para executar o sistema, salve o código em um arquivo com extensão ".py" (por exemplo, "gerenciador\_pacientes.py"). Em seguida, abra um terminal ou prompt de comando, navegue até o diretório onde o arquivo foi salvo e execute o comando python gerenciador\_pacientes.py.

O sistema será iniciado e exibirá o menu de opções. O usuário poderá interagir com o sistema escolhendo as opções desejadas e fornecendo as informações solicitadas.

Caso contrário use o link abaixo para utilizar o compilador online e efetuar o teste do código [Online Compiler](#).

## 6. Conclusão

Este projeto demonstra a aplicação prática de conceitos de programação orientada a objetos e estrutura de dados na resolução de um problema real. A utilização da lista encadeada permite o gerenciamento eficiente das informações dos pacientes, proporcionando flexibilidade e desempenho na manipulação dos dados.

## 7. Trabalhos Futuros

O sistema pode ser aprimorado com a implementação de novas funcionalidades, como: Persistência de dados: salvar e carregar a lista de pacientes em um arquivo, permitindo que as informações sejam armazenadas e recuperadas entre diferentes execuções do programa.

Interface gráfica: desenvolver uma interface gráfica mais amigável para o usuário, facilitando a interação com o sistema.

Relatórios: gerar relatórios com estatísticas sobre os pacientes, como a quantidade de pacientes por doença, a média de idade dos pacientes, etc.

## 8. Referências

- Material da UCS
- Compilador em [Python Online](#)
- [Visual Studio Code](#)
- [GitHub](#)
- Colegas de empresa ([Inova](#))
- Minha Mãe 🙄 (Técnica de Enfermagem atuou como usuário de teste rosipaimsilva@gmail.com)

## Observações:

Os comentários no código fonte explicam a lógica de cada função e a finalidade de cada bloco de código.

Uso de IA criado com LLM, segue a referência da mesma no repositório [GitHub](#) (Desenvolvido por mim).