



**CESED - CENTRO DE ENSINO SUPERIOR E DESENVOLVIMENTO**

**UNIFACISA – CENTRO UNIVERSITÁRIO**

**CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**DOCUMENTAÇÃO DE SISTEMA PARA GERENCIAMENTO DE ACERVO DE  
UMA BIBLIOTECA**

**Competência Curricular:** Aplicar Algoritmos de Apoio à Pesquisa Operacional.

**Professor:** Jonhnanthan Oliveira.

**Equipe:** Hugo Bezerra Figueiroa;  
Kleiton da Silva Ferreira.

**Campina Grande - PB  
Novembro-2024**

# DOCUMENTAÇÃO DE SISTEMA DE GERENCIAMENTO DE ACERVO DE UMA BIBLIOTECA

## 1. INTRODUÇÃO

O sistema de gerenciamento de acervo de biblioteca foi desenvolvido em Java e permite adicionar, listar, buscar, ordenar por título e remover livros de um acervo. O objetivo principal foi implementar uma estrutura de dados personalizada, seguindo a exigência de utilizar uma lista encadeada ao invés de uma lista sequencial.

## 2. ESCOLHA DA ESTRUTURA DE DADOS

### Lista Encadeada vs. Lista Sequencial

Optamos pela lista encadeada em vez de uma lista sequencial (array) para este sistema, devido às suas vantagens em termos de flexibilidade e eficiência:

- **Tamanho Dinâmico:** A lista encadeada não exige um tamanho fixo previamente definido, permitindo que o acervo cresça dinamicamente à medida que novos livros são adicionados. Em comparação, uma lista sequencial exige um tamanho inicial ou o redimensionamento constante do array, o que é custoso em termos de memória e processamento.
- **Facilidade para Inserção e Remoção:** Em uma lista encadeada, adicionar ou remover elementos em qualquer posição é mais eficiente, pois envolve apenas o ajuste dos ponteiros dos nós (endereços de memória) e não a realocação de elementos. Em uma lista sequencial, a remoção e inserção em posições intermediárias exigiriam o deslocamento dos elementos subsequentes, o que resulta em maior consumo de tempo e memória.
- **Uso de Memória Contínua:** A lista encadeada aloca memória conforme os elementos são adicionados, enquanto a lista sequencial exige um bloco de memória contínuo. Isso torna a lista encadeada mais adequada para sistemas com variação constante de tamanho, como uma biblioteca, onde o número de livros pode aumentar ou diminuir com frequência.

## 3. FUNCIONAMENTO DO CÓDIGO

### Estrutura Principal do Código

O sistema foi implementado com as seguintes classes:

- **Livro:** Representa os dados de cada livro com os atributos título, autor e ano de publicação.

- **Node:** Representa um nó na lista encadeada, contendo um objeto Livro e uma referência (próximo) ao próximo nó na lista.
- **ListaEncadeada:** Contém os métodos principais para manipulação dos livros na lista, incluindo a ordenação dos livros por título.
- **Program:** Classe principal, que fornece uma interface de linha de comando para interação com o usuário.

## Classes e Métodos

- **Classe Livro:** Define os atributos titulo, autor e anoPublicacao. Possui o método toString para exibir os dados de cada livro.
- **Classe Node:** Representa um nó da lista encadeada, com os atributos livro (para armazenar o livro) e proximo (para apontar ao próximo nó).
- **Classe ListaEncadeada:**
  - **adicionarLivro(Livro livro):** Adiciona um novo livro ao final da lista encadeada.
  - **listarLivros():** Percorre a lista e exibe os detalhes de cada livro; caso a lista esteja vazia, exibe uma mensagem apropriada.
  - **buscarLivroPorTitulo(String titulo):** Busca um livro pelo título, percorrendo a lista encadeada.
  - **ordenarPorTitulo():** Este método utiliza o algoritmo Bubble Sort para ordenar os livros pelo título em ordem alfabética.
  - **removerLivroPorTitulo(String titulo):** Remove um livro pelo título, ajustando os ponteiros para "pular" o nó correspondente, removendo o livro desejado sem necessidade de realocação de outros elementos.
- **Classe Program:** Implementa o menu interativo com opções para adicionar, listar, buscar, ordenar e remover livros, usando a classe ListaEncadeada para armazenar e gerenciar o acervo. Utiliza um Scanner para receber as entradas do usuário e um loop do-while para manter o menu ativo até a opção de saída ser selecionada.

## Fluxo de Execução do Código

O usuário interage com o menu do sistema na linha de comando. Ao adicionar um livro, ele é inserido no final da lista encadeada. A opção de listar exibe todos os livros na lista ou uma mensagem indicando que não há livros. A busca permite localizar um livro específico pelo título, retornando seus detalhes caso exista. A remoção de um livro ocorre ao buscar o título na lista encadeada e ajustar os ponteiros para remover o nó, garantindo eficiência sem deslocamentos de memória. A ordenação dos livros por título, com o método ordenarPorTitulo, organiza a lista em ordem alfabética de forma simples e eficiente.

## 4. ESCOLHA DO BUBBLE SORT PARA ORDENAÇÃO

Escolhemos o Bubble Sort por ser uma implementação simples e direta, suficiente para listas pequenas e facilmente aplicável a uma estrutura encadeada sem o uso de arrays auxiliares. O método usa um laço while, que facilita o

controle sobre a ordenação, parando automaticamente quando a lista estiver organizada, economizando comparações e tempo de execução

## **5. CONCLUSÃO**

A implementação da lista encadeada atende aos requisitos de flexibilidade, eficiência e organização de dados. A escolha dessa estrutura permitiu desenvolver um sistema adequado para uma aplicação como um acervo de biblioteca, onde a quantidade de itens pode variar e operações dinâmicas são frequentes. A escolha do Bubble Sort para ordenar os livros permite uma organização básica e eficiente para listas pequenas, aproveitando o controle do laço while para otimizar o tempo de execução. A estrutura e o código foram organizados para facilitar a manutenção e possíveis expansões futuras.