

# Documentação do Sistema de Gerenciamento de Acervo de Biblioteca

## Introdução

Este sistema foi desenvolvido para gerenciar o acervo de uma biblioteca utilizando conceitos básicos de estruturas de dados e algoritmos de ordenação, sem o uso do Collections Framework. O objetivo é organizar e manipular os dados de livros (título, autor e ano de publicação) de maneira eficiente, utilizando uma interface simples.

## Estrutura de Dados Utilizada

### Lista Sequencial

Optou-se por utilizar uma **lista sequencial** (array) para armazenar os livros. A escolha desta estrutura de dados foi feita devido à simplicidade de implementação e ao fato de que a lista sequencial proporciona um acesso direto aos elementos via índice, o que torna a busca e a manipulação de elementos uma operação com custo constante  $O(1)$ .

```
package library;

public class BooksList {
    private Book[] books;
    private int size;

    public BooksList(int ListCapacity) {
        books = new Book[ListCapacity];
        size = 0;
    }

    public void addBook(Book book) {
        if (books.length == size) {
            Book[] newList = new Book[books.length * 2];
            for (int i = 0; i < books.length; i++) {
                newList[i] = books[i];
            }
            books = newList;
        }
        books[size] = book;
        size++;
    }

    public void printBooks() {
        for (int i = 0; i < size; i++) {
            System.out.println(books[i]);
        }
    }

    public int listSize() {
        return size;
    }
}
```

Justificativa:

- Simplicidade na inserção e manipulação dos dados.
- Custo de acesso direto aos elementos.
- Implementação manual de redimensionamento da lista permite maior controle sobre a memória utilizada.

## Algoritmo de Ordenação Utilizado

### Quick Sort

Para a ordenação dos livros, tanto por título quanto por autor, foi escolhido o algoritmo Quick Sort. Esse algoritmo foi selecionado devido à sua eficiência em cenários práticos, apresentando uma complexidade média de tempo de  $O(n \log n)$ , sendo mais rápido em comparação a outros algoritmos de ordenação como o Bubble Sort.

O Quick Sort é um algoritmo de divisão e conquista, onde um pivô é escolhido para particionar o array em duas partes: uma com elementos menores e outra com elementos maiores que o pivô. Posteriormente, as duas partes são ordenadas recursivamente.

A implementação do Quick Sort foi feita separadamente para ordenação por título e por autor, usando comparações de strings para definir a ordem lexicográfica.

```
public void sortByTitle() {
    quickSortByTitle(start:0, size - 1);
}

public void sortByAuthor() {
    quickSortByAuthor(start:0, size - 1);
}

private void quickSortByTitle(int start, int end) {
    if (start < end) {
        int pivotIndex = partitionByTitle(start, end);
        quickSortByTitle(start, pivotIndex - 1);
        quickSortByTitle(pivotIndex + 1, end);
    }
}
```

```
private int partitionByTitle(int start, int end) {
    Book pivot = books[end];
    int i = start - 1;

    for (int j = start; j < end; j++) {
        if (books[j].getTitle().compareTo(pivot.getTitle()) <= 0) {
            i++;
            books[i] = books[j];
        }
    }
}
```

Justificativa:

- Melhor desempenho em comparação a outros algoritmos de ordenação básicos.
- Complexidade média de  $O(n \log n)$ , sendo adequada para a ordenação de listas de tamanho moderado a grande.
- Capacidade de trabalhar de maneira eficiente com grandes quantidades de dados desordenados.

## Funcionalidades do Sistema

Adicionar novo livro: O usuário pode inserir novos livros informando o título, autor e ano de publicação. Caso a lista atinja sua capacidade máxima, ela será automaticamente redimensionada para o dobro do tamanho atual.

- Listar livros: Todos os livros adicionados podem ser listados, exibindo seus detalhes (título, autor e ano).
- Ordenar livros por título: Os livros podem ser ordenados em ordem alfabética com base no título, utilizando o algoritmo Quick Sort.
- Ordenar livros por autor: Similar à funcionalidade anterior, mas a ordenação é feita com base no nome do autor.
- Exibir número de livros: Exibe a quantidade total de livros atualmente na lista.
- Encerrar o sistema: Finaliza a execução do sistema.

## Considerações Finais

A implementação de uma lista sequencial junto ao Quick Sort garante simplicidade e eficiência no gerenciamento dos livros da biblioteca. Embora o sistema utilize uma estrutura e algoritmos básicos, ele é capaz de lidar com quantidades razoáveis de dados, apresentando desempenho aceitável para o propósito do projeto.