

**CESED – CENTRO DE ENSINO SUPERIOR E DESENVOLVIMENTO
CENTRO UNIVERSITÁRIO UNIFACISA**

**APLICAR ALGORITMOS DE APOIO À PESQUISA OPERACIONAL
PROF. JONHNANTHAN OLIVEIRA**

**IMPLANTAÇÃO DAS TÉCNICAS E ESTRUTURAS AVANÇADAS
JUSTIFICATIVAS DAS ESCOLHAS**

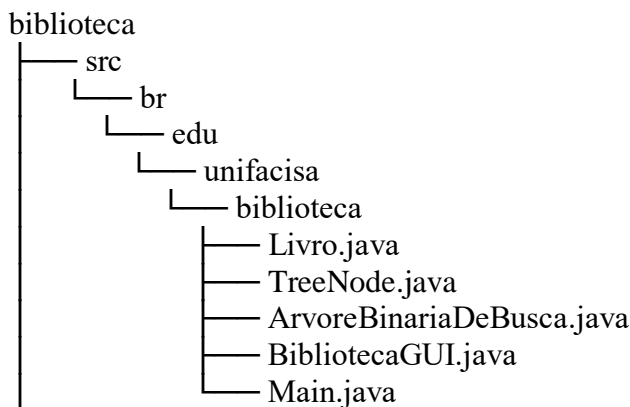
**ALUNOS:
GUSTAVO TOMIO MAGALHÃES KUBO
SÉRGIO MAGNO CASTOR PINHEIRO
THIAGO LIMEIRA DE ALENCAR**

**CAMPINA GRANDE – PB
2024.2**

1. Estrutura do Projeto

Este projeto é um sistema para gerenciar livros de uma biblioteca, utilizando estruturas de dados como **ArrayList** para organizar e armazenar as informações de forma eficiente. O sistema permite adicionar, listar, buscar e ordenar livros (por título ou autor) com uma interface gráfica simples.

A estrutura do projeto é a seguinte:



2. Técnicas e Estruturas Utilizadas

1. Estruturas de Dados

- A árvore binária de busca foi escolhida para armazenar e buscar livros rapidamente.
- O **ArrayList** foi usado para operações temporárias, como ordenação e exibição de livros.

2. Interface Gráfica

- Usamos **Swing** para criar a interface gráfica, facilitando a interação do usuário com o sistema.

3. Ordenação e Comparação

- A classe **Livro** implementa **Comparable** para ordenação por título, e **Comparator** permite ordenação por autor.

4. Recursividade

- A recursividade foi aplicada nas operações da árvore binária (busca, inserção e remoção), simplificando o código.

5. Ordenação com Collections

- O método **Collections.sort** foi usado para ordenar listas temporárias de livros.

6. Design Modular

- O sistema foi dividido em classes específicas, como **Livro**, **TreeNode**, **ArvoreBinariaDeBusca**, e **BibliotecaGUI**, facilitando a manutenção e futuras expansões.

3. Justificativa das Escolhas

1. Estruturas de Dados

- A árvore binária de busca é eficiente para busca e inserção de livros.
- 2. **Swing**
 - **Swing** foi escolhido por sua simplicidade e ampla documentação.
- 3. **Collections**
 - O **ArrayList** simplificou a implementação e aumentou a eficiência, cuidando do tamanho dinâmico da lista.
- 4. **Recursividade**
 - A recursividade reflete a estrutura hierárquica da árvore, tornando o código mais claro.
- 5. **Modularidade**
 - A separação das responsabilidades em classes torna o código mais organizado e fácil de expandir.

4. Benefícios da Implementação

- **Eficiência:** A combinação de árvore binária e **Collections** garante buscas e ordenações rápidas.
- **Simplicidade:** A interface gráfica facilita o uso do sistema.
- **Organização:** O design modular permite fácil manutenção e adição de funcionalidades.

Conclusão Final

O projeto aplica conceitos de programação, como estruturas de dados eficientes e recursividade. A escolha de **Swing** e **Collections.sort** equilibraram simplicidade e eficiência. O sistema atende aos objetivos propostos, sendo funcional, organizado e pronto para melhorias.