***Estrutura de Dados***

Estruturas de Dados - Avaliação Bimestral N3

**ADS - AMS 4**

Prof: Carlos Veríssimo - [carlos.pereira70@fatec.sp.gov.br](mailto:carlos.pereira70@fatec.sp.gov.br)

Marcos Vinicius da Silva – [marcos.silva435@fatec.sp.gov.br](mailto:marcos.silva435@fatec.sp.gov.br)

Rodrigo Veloso - [rodrigo.goncalves10@fatec.sp.gov.br](mailto:rodrigo.goncalves10@fatec.sp.gov.br)

**Sumário**

[1 – Resumo 2](#_Toc146625890)

[1.1 Contexto 2](#_Toc146625891)

[1.2 – Proposito 2](#_Toc146625892)

[1.3 - Metodologia 2](#_Toc146625893)

[1.4 Resultados 3](#_Toc146625894)

[1.5 Conclusão 7](#_Toc146625895)

[2.0 – Argumentação teórica 7](#_Toc146625896)

[2.1 - Modelo de pesquisa 7](#_Toc146625897)

[2.2 – Recursividade 8](#_Toc146625898)

[2.3 – Arvore Binaria 8](#_Toc146625899)

[3.0 – Link 8](#_Toc146625900)

# **– Resumo**

# **Contexto**

Este código em Java faz parte de um sistema que implementa uma estrutura de dados conhecida como "árvore binária de busca". Uma árvore binária de busca é uma estrutura hierárquica que organiza dados de maneira ordenada, seguindo a regra de que elementos menores são colocados à esquerda e elementos maiores à direita. O sistema apresenta a árvore por meio de uma interface gráfica utilizando a biblioteca Swing do Java.

# **– Proposito**

O objetivo deste código é criar uma aplicação interativa que permite adicionar, visualizar e buscar números em uma árvore binária de busca. Essa ferramenta é valiosa para fins educacionais, pois permite aos usuários compreenderem de forma visual o funcionamento dessa estrutura de dados. Além disso, oferece uma maneira prática de realizar operações em árvores binárias, tornando o aprendizado e a exploração dessa estrutura mais acessíveis e intuitivos.

# **- Metodologia**

**Implementação da Árvore Binária de Busca:**

O código utiliza a classe "ArvoreBinaria" para criar uma árvore binária de busca.

A árvore mantém os números inseridos organizados de acordo com as regras da árvore.

**Interface Gráfica Interativa:**

O sistema cria uma interface gráfica com botões para interação com o usuário.

Os botões permitem realizar diversas operações na árvore, como adicionar números, listar números em diferentes ordens e buscar números específicos.

**Operações Disponíveis**:

Adicionar Números: O usuário pode inserir números na árvore binária de busca através da interface.

Listar Números em Diferentes Ordens: A aplicação permite listar os números na árvore em ordem crescente, pré-ordem e pós-ordem.

Buscar Números: É possível realizar buscas na árvore para encontrar números específicos.

**Medição de Tempo de Execução**:

Durante o processo de busca binária, o código mede o tempo de execução da operação. Isso permite avaliar o desempenho do algoritmo de busca, fornecendo informações sobre sua eficiência.

# **Resultados**

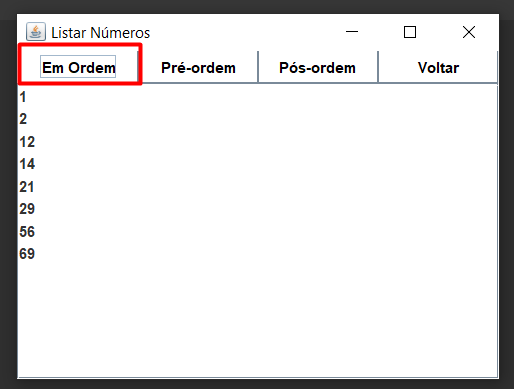
* + - O programa permite que os usuários adicionem números à árvore binária de busca, garantindo que a propriedade da árvore seja mantida.

Interface gráfica do usuário, Tabela

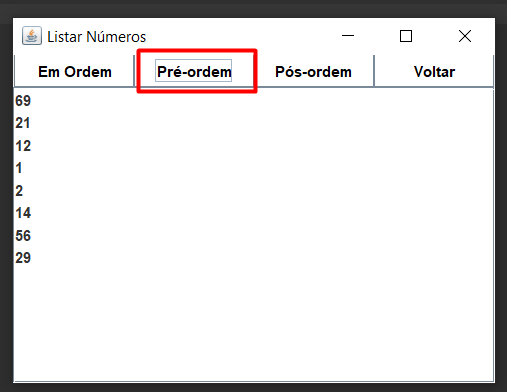
Descrição gerada automaticamente

* + - Ao selecionar a opção ”Listar Números” os números podem ser listados em três ordens diferentes, permitindo que os usuários visualizem a estrutura da árvore e a ordem dos elementos.

**Em Ordem:**



**Pré-ordem:**



**Pós-ordem:**

Interface gráfica do usuário

Descrição gerada automaticamente

* + - Ao selecionar a opção “Buscar Número” no menu principal a busca binária retorna se um número específico, fornecido pelo usuário, está presente na árvore.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

* Ao finalizar, o usuario tem a opção de sair da aplicação ao selecionar o botão “Sair” que logo irá mostrar uma mensagem informando “Volte sempre!”.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

# **- Conclusão**

Este código oferece uma implementação prática e instrutiva de uma árvore binária de busca, apresentada por meio de uma interface gráfica amigável. Além de tornar mais fácil a compreensão das estruturas de árvores binárias de busca, ele também disponibiliza uma ferramenta prática para realizar diversas operações nesse tipo de árvore. A medição do tempo durante a busca binária permite avaliar o desempenho do algoritmo, o que pode ser valioso em situações em que a eficiência é uma consideração importante.

# **– Argumentação teórica**

# **- Modelo de pesquisa**

O código ilustra o uso de dois métodos de travessia em árvores binárias, conhecidos como "pré-ordem" e "pós-ordem". Esses métodos de travessia são técnicas que permitem percorrer e examinar todos os elementos de uma árvore binária de forma organizada.

**Pré-ordem:** Na técnica de pré-ordem, o algoritmo começa pelo nó raiz da árvore, depois visita o nó esquerdo e, por fim, o nó direito. Esse processo é repetido recursivamente para todos os nós da árvore.

**Pós-ordem**: Já na técnica de pós-ordem, o algoritmo começa pelo nó esquerdo, seguido pelo nó direito e, por último, visita o nó raiz. Esse padrão também é aplicado de maneira recursiva a todos os nós da árvore. O método de pós-ordem é frequentemente usado para liberar a memória alocada para a árvore, uma vez que garante que os nós folha sejam liberados antes dos nós pais.

# **– Recursividade**

A recursividade é um conceito essencial na implementação de algoritmos relacionados a árvores binárias. Ela permite que uma função se invoque a si mesma para lidar com partes menores ou subárvores da árvore, simplificando a divisão de problemas complexos em problemas menores e mais acessíveis. A recursividade é aplicada nas funções de travessia da árvore, tais como enumeração em pré-ordem, pós-ordem e em ordem. Cada função recursiva atua em uma seção específica da árvore, realizando inicialmente o processamento dos nós à esquerda e, em seguida, combinando os resultados conforme necessário.

# **– Arvore Binaria**

Uma árvore binária é uma estrutura de dados composta por nós interconectados através de arestas, criando uma hierarquia. Cada nó em uma árvore binária pode ter, no máximo, dois descendentes, frequentemente denominados filho esquerdo e filho direito. A árvore começa com um nó especial chamado raiz, que serve como ponto de partida para acessar todos os outros nós na árvore.

Em uma árvore binária de busca, a ordem dos elementos desempenha um papel crucial. Elementos menores são colocados à esquerda da raiz, enquanto elementos maiores são posicionados à direita. Essa organização oferece uma maneira eficaz de realizar operações de busca, uma vez que em cada etapa podemos descartar metade da árvore, reduzindo a necessidade de comparações. A implementação de uma árvore binária de busca é ilustrada através da classe "ArvoreBinaria", que demonstra como os elementos são inseridos e percorridos de acordo com as regras estabelecidas para manter a propriedade de busca.

# **3.0 – Link**

Github: <https://github.com/RodrigoVeloso994/Estrutura-de-Dados-N3>

[https://github.com/Marcos03102004/Estrutura-de-Dados-N3](%20%20%20https://github.com/Marcos03102004/Estrutura-de-Dados-N3)