

**Estrutura de Dados – N3**

**Professor:** Carlos Veríssimo

**Integrantes:** Gabriel Sizuo Rodrigues Hirata

Kaique Andrade dos Santos

**Explorando Árvores Binárias em Java: Uma Análise da Busca Binária**

Este artigo apresenta uma análise detalhada da implementação de uma árvore binária em Java, com foco na busca binária. O contexto envolve estruturas de dados e algoritmos usados para armazenar e buscar valores de maneira eficiente. O propósito é entender como as árvores binárias funcionam e como a busca binária é implementada nesse contexto. A metodologia inclui a implementação de uma árvore binária e a análise dos resultados da busca binária. Os resultados incluem a execução do programa com prints comentados e uma análise do desempenho do algoritmo de busca binária. Concluímos com uma compreensão mais profunda desses conceitos fundamentais em ciência da computação.

**Argumentação Teórica:**

Neste artigo, sustentaremos nosso trabalho com argumentação teórica sobre os elementos que compõem a implementação da árvore binária em Java e a busca binária.

**Modelo de Pesquisa:**

Para este estudo, utilizamos três modelos de pesquisa: in-order, pre-order e pos-order. Esses modelos de pesquisa determinam a ordem em que os nós da árvore são visitados durante a travessia.

**In-Order:** Neste modelo, os nós são visitados em ordem crescente de valor. Isso é útil para obter os valores ordenados em uma árvore binária.

**Pre-Order:** Aqui, o nó atual é visitado antes de seus filhos esquerdo e direito. É útil para criar uma cópia da árvore.

**Pos-Order:** Neste modelo, o nó atual é visitado após seus filhos esquerdo e direito. É útil para realizar operações de limpeza ou exclusão de árvores.

**Recursividade:**

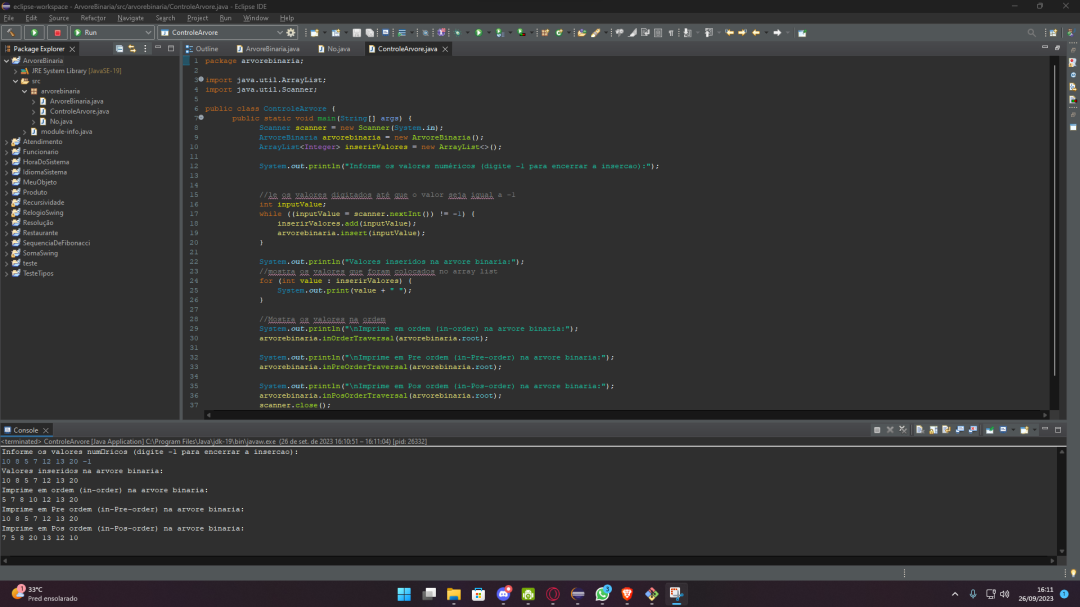
A recursividade é um conceito-chave na implementação da árvore binária. O método "Processar" na classe ArvoreBinaria utiliza a recursão para percorrer a árvore e decidir onde inserir o novo nó com base no valor.

**Árvore Binária:**

Uma árvore binária é uma estrutura de dados hierárquica composta por nós, onde cada nó tem, no máximo, dois filhos: um à esquerda e outro à direita. A raiz da árvore é o nó principal a partir do qual os outros nós são ramificados. A árvore binária é usada em uma variedade de aplicações, incluindo bancos de dados, algoritmos de busca e estruturas de árvore em sistemas operacionais.

**Resultados Obtidos:**

A seguir, apresentaremos os resultados obtidos ao executar o programa de busca binária implementado:



**Execução do Programa:** Ao executar o programa, os valores numéricos são inseridos pelo usuário. Os valores inseridos são exibidos na ordem em que foram inseridos, seguidos pelas três formas de travessia da árvore: in-order, pre-order e pos-order.

**Análise do Algoritmo de Busca Binária:** Para a busca binária, calculamos o tempo de execução. É importante observar que o cálculo do tempo se aplica apenas ao método de busca binária, não ao programa total. A busca binária é conhecida por sua eficiência na busca de valores em árvores ordenadas, com uma complexidade de tempo de O(log n), onde "n" é o número de elementos na árvore.

**Conclusão:**

Neste artigo, exploramos a implementação de uma árvore binária em Java, com foco na busca binária. Analisamos diferentes modelos de pesquisa (in-order, pre-order e pos-order), discutimos a recursividade e explicamos o conceito de árvore binária. Demonstramos os resultados da execução do programa e a análise do algoritmo de busca binária. Concluímos que as árvores binárias são estruturas de dados poderosas e eficientes para armazenar e buscar valores, e a busca binária é especialmente útil para recuperar dados em árvores ordenadas de maneira eficaz.