

**UNIVERSIDAD DE SONORA**

**Unidad Regional Centro**

**División de Ingeniería**

# **Ingeniería en Sistemas de Información**

**Fundamentos de Computación 3**

**Maestro: IVAN DOSTOYEWSKI MEZA IBARRA**

**Grupo 1 – Hora 5 PM – 6 PM- Edif. 5G – L205**

**Nombre del Alumno:**

* **Vallejo Leyva Marcos**

**Actividad 5: Recorrido con árboles.**

# **Reporte de Código Java - Recorrido con árboles.**

**Índice**

1. **Introducción**
2. **Marco Teórico**
3. **Descripción General del Código**
4. **Métodos Implementados en el Código**
5. **Ejecución y código**
6. **Conclusiones**

**Introducción**

Los árboles son estructuras de datos fundamentales en informática que modelan relaciones jerárquicas entre elementos. En este contexto, un árbol binario es un tipo de árbol en el que cada nodo tiene como máximo dos hijos. Este código implementa un programa en Java que permite crear, mostrar y recorrer un árbol binario utilizando los tres tipos de recorridos principales: preorden, inorden y postorden.

**Marco Teórico**

1. **Árboles:** En informática, un árbol es una estructura de datos jerárquica que consiste en un conjunto de nodos conectados por aristas. Cada nodo tiene un valor y cero o más nodos hijos. Uno de los nodos se designa como raíz del árbol, mientras que los nodos restantes se dividen en subárboles.
2. **Recorridos en árboles:**
   * **Preorden:** Se visita primero el nodo raíz, luego el subárbol izquierdo y finalmente el subárbol derecho.
   * **Inorden:** Se visita primero el subárbol izquierdo, luego el nodo raíz y finalmente el subárbol derecho.
   * **Postorden:** Se visita primero el subárbol izquierdo, luego el subárbol derecho y finalmente el nodo raíz.

**Descripción General del Código**

1. **Creación del árbol:** El programa solicita al usuario que ingrese los nodos del árbol binario separados por comas. Luego, utiliza el método **insert** para insertar los nodos en el árbol.
2. **Selección del recorrido:** Después de crear el árbol, se le pregunta al usuario qué tipo de recorrido desea realizar: preorden, inorden o postorden. Dependiendo de la elección del usuario, se llama al método correspondiente (**preOrder**, **inOrder**, **postOrder**) para mostrar los nodos del árbol en el orden seleccionado.
3. **Finalización del programa:** El programa permite al usuario realizar recorridos adicionales o salir del programa seleccionando la opción "Salir".

**Métodos Implementados en el Código**

En el código se implementan tres métodos principales para recorrer un árbol binario y un método para insertar nodos en el árbol.

**Métodos de Recorrido:**

1. **preOrderTraversal(TreeNode node):** Este método realiza un recorrido preorden en el árbol binario. Toma un nodo como parámetro y realiza las siguientes operaciones: - Imprime el valor del nodo actual. - Realiza un recorrido preorden en el subárbol izquierdo. - Realiza un recorrido preorden en el subárbol derecho.

Texto

Descripción generada automáticamente

1. **inOrderTraversal(TreeNode node):** Este método realiza un recorrido inorden en el árbol binario. Toma un nodo como parámetro y realiza las siguientes operaciones: - Realiza un recorrido inorden en el subárbol izquierdo. - Imprime el valor del nodo actual. - Realiza un recorrido inorden en el subárbol derecho.

Texto

Descripción generada automáticamente

1. **postOrderTraversal(TreeNode node):** Este método realiza un recorrido postorden en el árbol binario. Toma un nodo como parámetro y realiza las siguientes operaciones: - Realiza un recorrido postorden en el subárbol izquierdo. - Realiza un recorrido postorden en el subárbol derecho. - Imprime el valor del nodo actual.

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

**Método de Inserción:**

* + **insert(TreeNode root, int data):** Este método inserta un nuevo nodo en el árbol binario. Toma como parámetros el nodo raíz del árbol y el valor del nodo que se va a insertar. Si el nodo raíz es nulo, crea un nuevo nodo con el valor proporcionado. De lo contrario, compara el valor con el valor del nodo raíz y decide si debe insertarse en el subárbol izquierdo o derecho, llamando recursivamente a **insert** en el subárbol correspondiente.

Pantalla de computadora con letras

Descripción generada automáticamente

**Clase Main**

El programa principal (**main**) se encarga de solicitar al usuario los nodos del árbol binario, crear el árbol, ofrecer opciones para realizar los recorridos y manejar la entrada del usuario para ejecutar los recorridos o salir del programa.

**Métodos:**

1. Se crea una instancia de **Scanner** para la entrada de datos.
2. Se crea una instancia de **BinaryTree** para representar el árbol binario.
3. Se solicita al usuario que ingrese los nodos del árbol binario.
4. Se insertan los nodos ingresados por el usuario en el árbol binario utilizando el método **insert**.
5. Se presenta al usuario un menú para seleccionar el tipo de recorrido que desea realizar.
6. Se ejecuta un bucle infinito que espera la elección del usuario.
7. Dependiendo de la opción seleccionada por el usuario, se realiza el recorrido correspondiente utilizando los métodos **preOrder**, **inOrder** o **postOrder**, o bien se sale del programa si el usuario elige la opción de salida.

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamente

**Ejecución y Código:**

El código se ejecuta solicitando al usuario que ingrese los nodos del árbol binario y luego proporciona las opciones para realizar los recorridos preorden, inorden o postorden. El usuario puede repetir los recorridos o salir del programa según su elección.

Texto

Descripción generada automáticamente

**Conclusiones:**

Este programa implementa de manera efectiva la creación, visualización y recorrido de un árbol binario en Java, brindando al usuario la flexibilidad de elegir el tipo de recorrido que desea realizar. El uso de árboles y sus recorridos es fundamental en la ciencia de la computación y tiene una amplia gama de aplicaciones en campos como la informática, la inteligencia artificial y las bases de datos.