Árboles en Java

1. Concepto de árbol:

Un árbol es una estructura de datos no lineal que se caracteriza por una organización jerárquica de nodos. Cada nodo puede tener uno o más hijos, y existe un único camino que conecta la raíz (el nodo superior) con cualquier otro nodo del árbol.

2. Componentes de un árbol:

Un árbol está integrado por dos elementos principales:

- **Nodos:** Son las unidades básicas que contienen la información almacenada en el árbol. Pueden compararse con las casillas de un organigrama.
- **Aristas:** Son las conexiones que se establecen entre los nodos, representando la relación padre-hijo. Imagínelos como las líneas que unen las casillas en un organigrama.

3. Requisito para trabajar con árboles:

Un requisito fundamental para trabajar con árboles es la existencia de una jerarquía de nodos bien definida. Esto significa que debe haber una relación padre-hijo clara entre los nodos del árbol.

Un nodo padre puede tener uno o más hijos, pero un nodo solo puede tener un padre. Esta jerarquía permite establecer el orden y la relación entre los datos almacenados en el árbol.

4. Nodo externo:

Un nodo externo, también conocido como nodo hoja, es un nodo que no tiene hijos. En otras palabras, son los nodos que se encuentran en los extremos del árbol, sin ninguna conexión a otros nodos. Siguiendo la analogía del organigrama, serían los empleados sin subordinados.

Características de un nodo externo:

- No tiene hijos.
- Se encuentra en el último nivel del árbol.
- Su profundidad es la máxima del árbol.
- Puede tener uno o más padres.

Ejemplos de nodos externos:

En el siguiente árbol, los nodos 4, 5, 6 y 7 son nodos externos:

Importancia de los nodos externos:

Los nodos externos son importantes porque representan los datos finales del árbol. Son los nodos que contienen la información que se busca o se procesa en las operaciones que se realizan sobre el árbol.

5. Profundidad de un nodo en un árbol

La profundidad de un nodo en un árbol se refiere a la longitud del camino que conecta ese nodo con la raíz del árbol. Se mide en número de aristas (conexiones) que recorres desde la raíz hasta llegar al nodo específico.

- La raíz del árbol siempre tiene una profundidad de cero (0).
- Los nodos hoja (externos) tienen una profundidad que depende de su ubicación en el árbol. Cuanto más abajo esté un nodo hoja, mayor será su profundidad.
- Para cualquier otro nodo, su profundidad se calcula sumando uno a la profundidad de su nodo padre.

Ejemplo:

Considera el siguiente árbol:

- La profundidad de la raíz (nodo 1) es 0.
- La profundidad de los nodos hoja (4, 5 y 6) es 1, ya que están a una arista de la raíz.
- La profundidad del nodo 2 es 1, porque es hijo de la raíz y 1 + 0 (profundidad de la raíz) = 1.
- La profundidad del nodo 3 también es 1, siguiendo la misma lógica.

6. Características de un árbol binario:

Un árbol binario es un tipo especial de árbol con las siguientes características:

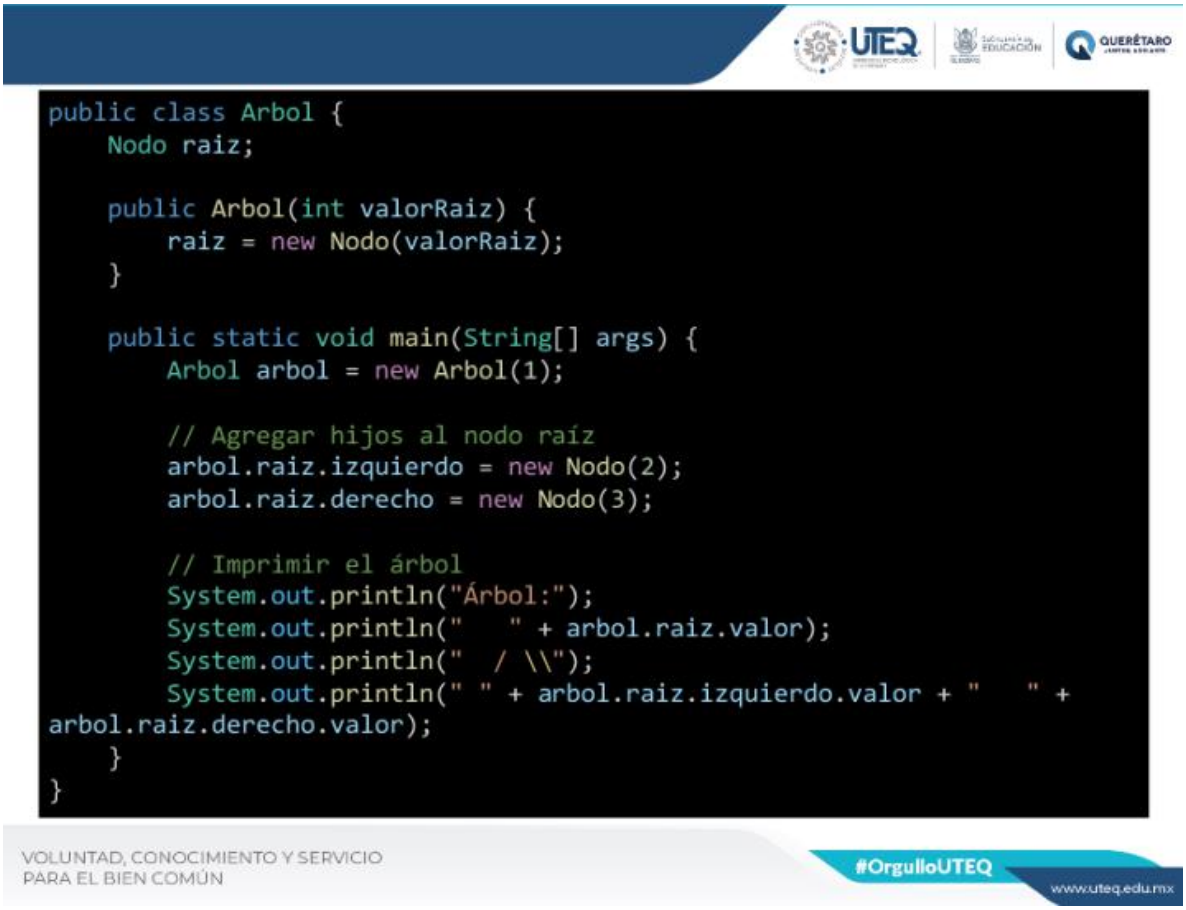
- **Máximo dos hijos:** Cada nodo en un árbol binario puede tener como máximo dos hijos: un hijo izquierdo y un hijo derecho.
- **Estructura jerárquica:** Al igual que en cualquier árbol, existe una jerarquía bien definida con una raíz en la parte superior y los nodos descendiendo de ella.
- **Orden (opcional):** Los árboles binarios pueden ser ordenados o no ordenados. En un árbol binario ordenado, existe una relación específica entre el valor de un nodo y el valor de sus hijos. Por ejemplo, en un árbol binario de búsqueda, el valor de cada nodo es mayor que todos los valores en su subárbol izquierdo y menor que todos los valores en su subárbol derecho.
- **Representación:** Los árboles binarios se suelen representar gráficamente con la raíz en la parte superior y los hijos conectados por líneas hacia abajo.

Ejemplos de árboles binarios:

Árbol binario sin ordenar:

Árbol binario de búsqueda:

Explica el Siguiente Código



define una clase llamada Arbol que representa un árbol binario. La clase tiene las siguientes características:

Atributos:

raiz: Es un objeto de la clase Nodo que representa la raíz del árbol.

Constructores:

Arbol(int valorRaiz): Crea un nuevo árbol con un único nodo raíz con el valor especificado.

Métodos:

main(String[] args): Es el método principal que se ejecuta al iniciar la aplicación. Este método crea un nuevo árbol binario, agrega dos hijos al nodo raíz e imprime el árbol en la consola.

Clase Nodo:

El código también define una clase interna llamada Nodo que representa un nodo individual en el árbol. La clase Nodo tiene un único atributo:

valor: Es un valor entero que se almacena en el nodo.

Explicación paso a paso:

1. Creación del árbol: Se crea un nuevo objeto de la clase Arbol con el valor 1 como valor de la raíz.
2. Adición de hijos: Se agregan dos nuevos nodos como hijos del nodo raíz. El nodo izquierdo tiene el valor 2 y el nodo derecho tiene el valor 3.
3. Impresión del árbol: Se imprime el árbol en la consola. La salida muestra el valor de la raíz, seguido de los valores de sus hijos izquierdo y derecho.

Ejemplo de salida:

```
Árbol:
 1
 / \
 2 3
```