

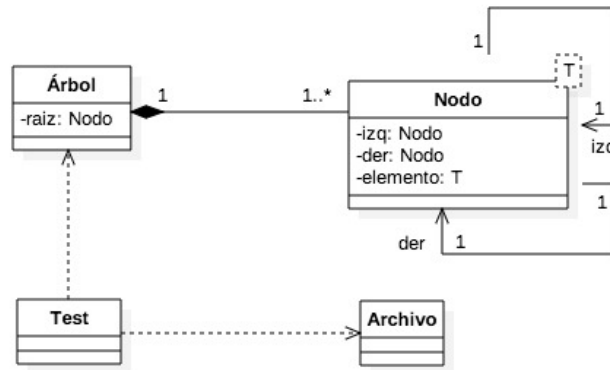


UNIVERSIDAD DEL CAUCA -
FACULTAD DE INGENIERIA ELECTRONICA Y TELECOMUNICACIONES
PROGRAMA DE INGENIERIA DE SISTEMAS
CURSO DE LABORATORIO DE ESTRUCTURAS DE DATOS II: GRUPO A

Práctica 6: Árboles Binarios de Búsqueda (ABB)

En esta práctica se llevará a cabo la implementación en java de un ABB. Recuerden que los árboles binarios son estructuras de datos no lineales y conocidos como árboles clasificados. Los ABB son estructuras de datos en donde se cumple que todos los elementos del subárbol izquierdo son menores a la raíz y los valores del subárbol derecho son mayores a la raíz.

Diagrama de Clases



Implementación Básica

```
//Arbol.java
package Negocio;
public class Arbol<T> {
    private Nodo raiz;
    public Arbol() {
        this.raiz = null;
    }
    public Nodo getRaiz() {
        return raiz;
    }
    public void setRaiz(Nodo raiz) {
        this.raiz = raiz;
    }
    public void insertar(Comparable elemento) {
        if (raiz == null) {
            raiz = new Nodo (elemento);
        } else {
            raiz = raiz.insertarNodo(raiz, elemento);
        }
    }
    public void imprimirArbolNiveles () {
        imprimirArbolNiveles(raiz, 0);
    }
    private void imprimirArbolNiveles(Nodo arbol, int nivel) {
        if (arbol != null) {
            imprimirArbolNiveles(arbol.getIzq(), nivel + 1);
            System.out.println(" "+nivel+" "+arbol.getElemento());
            imprimirArbolNiveles(arbol.getDer(), nivel + 1);
        }
    }
}
```



**UNIVERSIDAD DEL CAUCA -
FACULTAD DE INGENIERIA ELECTRONICA Y TELECOMUNICACIONES
PROGRAMA DE INGENIERIA DE SISTEMAS
CURSO DE LABORATORIO DE ESTRUCTURAS DE DATOS II: GRUPO A**

```
    }  
  }  
}  
//Nodo.java  
package Negocio;  
public class Nodo<T> {  
    private T elemento;  
    private Nodo izq, der;  
  
    public Nodo(T elemento) {  
        this.elemento = elemento;  
        this.izq = null;  
        this.der = null;  
    }  
    public T getElemento() {  
        return elemento;  
    }  
    public void setElemento(T elemento) {  
        this.elemento = elemento;  
    }  
    public Nodo getIzq() {  
        return izq;  
    }  
    public void setIzq(Nodo izq) {  
        this.izq = izq;  
    }  
    public Nodo getDer() {  
        return der;  
    }  
    public void setDer(Nodo der) {  
        this.der = der;  
    }  
    public Nodo insertarNodo (Nodo n, Comparable elemento) {  
        if (n == null) {  
            return new Nodo(elemento);  
        } else if (elemento.compareTo(n.getElemento())<0) {  
            n.izq = insertarNodo(n.getIzq(), elemento);  
        } else if (elemento.compareTo(n.getElemento())>0) {  
            n.der = insertarNodo(n.getDer(), elemento);  
        }  
        return n;  
    }  
}  
//Test.java  
package Presentacion;  
import Negocio.*;  
public class Test {  
    public static <T> void main(String[] args) {  
        Arbol ObjetoArbol = new Arbol();  
    }  
}
```

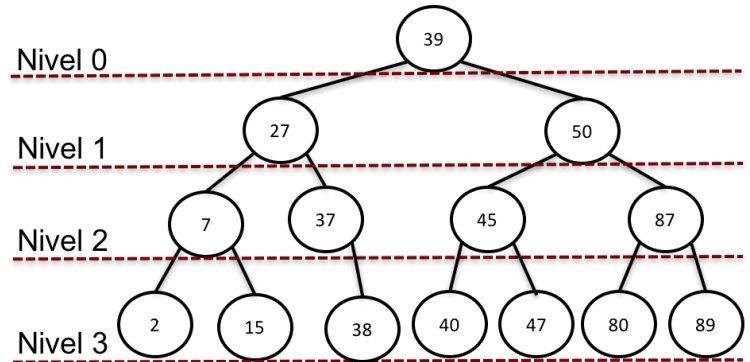


**UNIVERSIDAD DEL CAUCA -
FACULTAD DE INGENIERIA ELECTRONICA Y TELECOMUNICACIONES
PROGRAMA DE INGENIERIA DE SISTEMAS
CURSO DE LABORATORIO DE ESTRUCTURAS DE DATOS II: GRUPO A**

```
ObjetoArbol.insertar(39);
ObjetoArbol.insertar(27);
ObjetoArbol.insertar(50);
ObjetoArbol.insertar(7);
ObjetoArbol.insertar(37);
ObjetoArbol.insertar(45);
ObjetoArbol.insertar(87);
ObjetoArbol.insertar(2);
ObjetoArbol.insertar(15);
ObjetoArbol.insertar(38);
ObjetoArbol.insertar(40);
ObjetoArbol.insertar(47);
ObjetoArbol.insertar(80);
ObjetoArbol.insertar(89);
```

```
//mostrar el árbol resultante
ObjetoArbol.imprimirArbolNiveles ();
```

```
}
}
```



Funcionalidades para implementar:

1. Crear un árbol binario de búsqueda (números/caracteres) mediante un archivo.

```
//Árbol binario de Búsqueda
//1 Caracteres / 0 números
0
39 27 50 7 37 45 87 2 15 38 40 47 80 89
```

Nota 1: Se deberá garantizar respecto al tipo de árbol que los datos coincidan, en el caso de números que sean solo números, en el caso de caracteres solo caracteres. Haga uso de excepciones.

Nota 2: Se deberá garantizar que no se permitan ingresar valores repetidos al árbol.

2. Crear un menú que permita:
 - a. Mostrar todos los recorridos del árbol
 - b. Mostrar todos los nodos pertenecientes a un nivel del árbol.

Nivel	Información
0	39
2	7, 37, 45, 87
4	Este nivel no existe en el árbol.

- c. Mostrar cual es el elemento mayor y menor del árbol y en que nivel está

Mayor	
89	Es el elemento mayor del árbol y está en el nivel 3
Menor	
2	Es el menor elemento del árbol y esta en el nivel 3