

## ESTRUCTURAS DE DATOS 2

**CONTENIDO:** // Breve Descripción del Contenido

# TAREA-1. ARBOLES BINARIOS DE BÚSQUEDA.

**PORCENTAJE TERMINADO : 100%.**

**Integrantes:**

GRUPO	Nombre	Registro
14	Cristian Gabriel Gedalge Cayhuara	219022062
16	Enrique Jose Mamani Samurio	216164753

**Fecha de Presentación:** Martes ,07 de mayo de 2024

**COMENTARIO:**

El problema propuesto nos parecieron interesante ya que ocupamos una lista de arboles , lo cual fue nuevo para nosotros

Lista de menor a mayor

CLASE NODO

```
public class Nodo {
    public Nodo izq;
    public Nodo der;
    public int elem;
    public Nodo(int elem){
        this.elem=elem;
        izq=der=null;
    }
}
```

CLASE ARBOL

```
public class Arbol {
    public Nodo raiz;
    public Arbol(){
        raiz=null;
    }
    public void insertar(int x){
        raiz=insertar(raiz,x);
    }

    private Nodo insertar(Nodo p, int x) {
        if(p==null)
            return new Nodo(x);
        if(x<p.elem){
            p.izq=insertar(p.izq, x);
        }else{
            p.der=insertar(p.der, x);
        }
        return p;
    }

    public void inorden(){
        inorden(raiz);
    }

    private void inorden(Nodo p) {
        if (p==null)
            return;
        inorden(p.izq);
        System.out.println(p.elem);
    }
}
```

```

        inorden(p.der);
    }

    public void inordenMay(){
        inordenMay(raiz);
    }

    private void inordenMay(Nodo p) {
        if (p==null)
            return;
        inordenMay(p.der);
        System.out.println(p.elem);
        inordenMay(p.izq);
    }

}

                                CLASE LISTAARBOL
public class ListaArbol {
    public int cantElem;
    public int max;
    public Arbol arbol[];

    public ListaArbol(){
        this.max=this.cantElem=10;
        arbol=new Arbol[10];
        for (int i = 0; i < 10; i++) {
            arbol[i]=new Arbol();
        }
    }

    public void generar(int n,int a, int b){
        for (int i = 1; i <= n; i++) {
            int r=(int) (a+(b-1)*Math.random());
            arbol[r%10].insertar(r);
        }
    }

    public void mostrar(){
        for (int i = 0; i < 10; i++) {

```

```

        arbol[i].inorden();
        System.out.println();
    }
}

public void mostrarMayoraMenor(){
    for (int i = 0; i < 10; i++) {
        arbol[i].inordenMay();
        System.out.println();
    }
}

}

                                CLASE MAIN

public class Main {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        ListaArbol la1=new ListaArbol();
        la1.generar(100, 0, 1000);
        la1.mostrar();
        la1.mostrarMayoraMenor();
    }

}

```