



UNIVERSIDAD TECNOLÓGICA DE  
PANAMÁCENTRO REGIONAL DE  
CHIRIQUÍ  
FACULTAD DE INGENIERÍA DE SISTEMAS  
COMPUTACIONALES



**CARRERA:**

Gestión y Desarrollo de Software

**ACTIVIDAD No. 6**

**LABORATORIO No. 2**

“Laboratorio 2”

**ASIGNATURA:** Estructura de Datos II

**DOCENTE:**

Profa. Nunehar Mondul

**ESTUDIANTE/s:**

Jorge Jiménez (4-826-874)

Briant Arango (4-825-620)

Jonathán Martínez (4-824-2482)

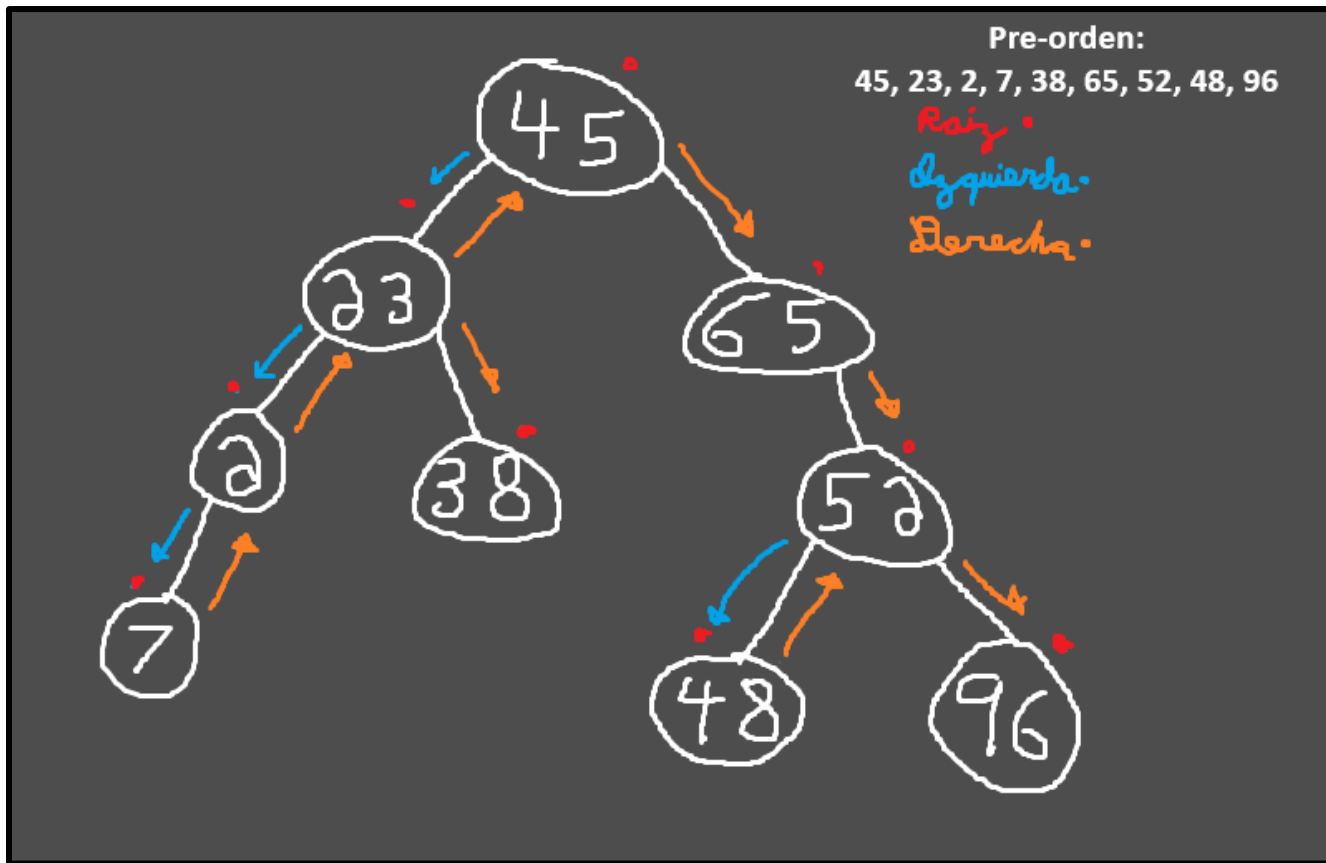
*I SEMESTRE 2025*

**FECHA:**

04/22/2025

## Desarrollo

1. "Dibujar árbol binario para el siguiente recorrido pre-orden 45, 23, 2, 7, 38, 65, 52, 48, 96."



2. "Mencionar nodos rama y nodos hoja."

Nodos rama – "Los que tienen al menos un hijo o más": 23, 2, 65 y 52.

Nodos hoja – "Los que no tienen hijos": 7, 38, 48 y 96.

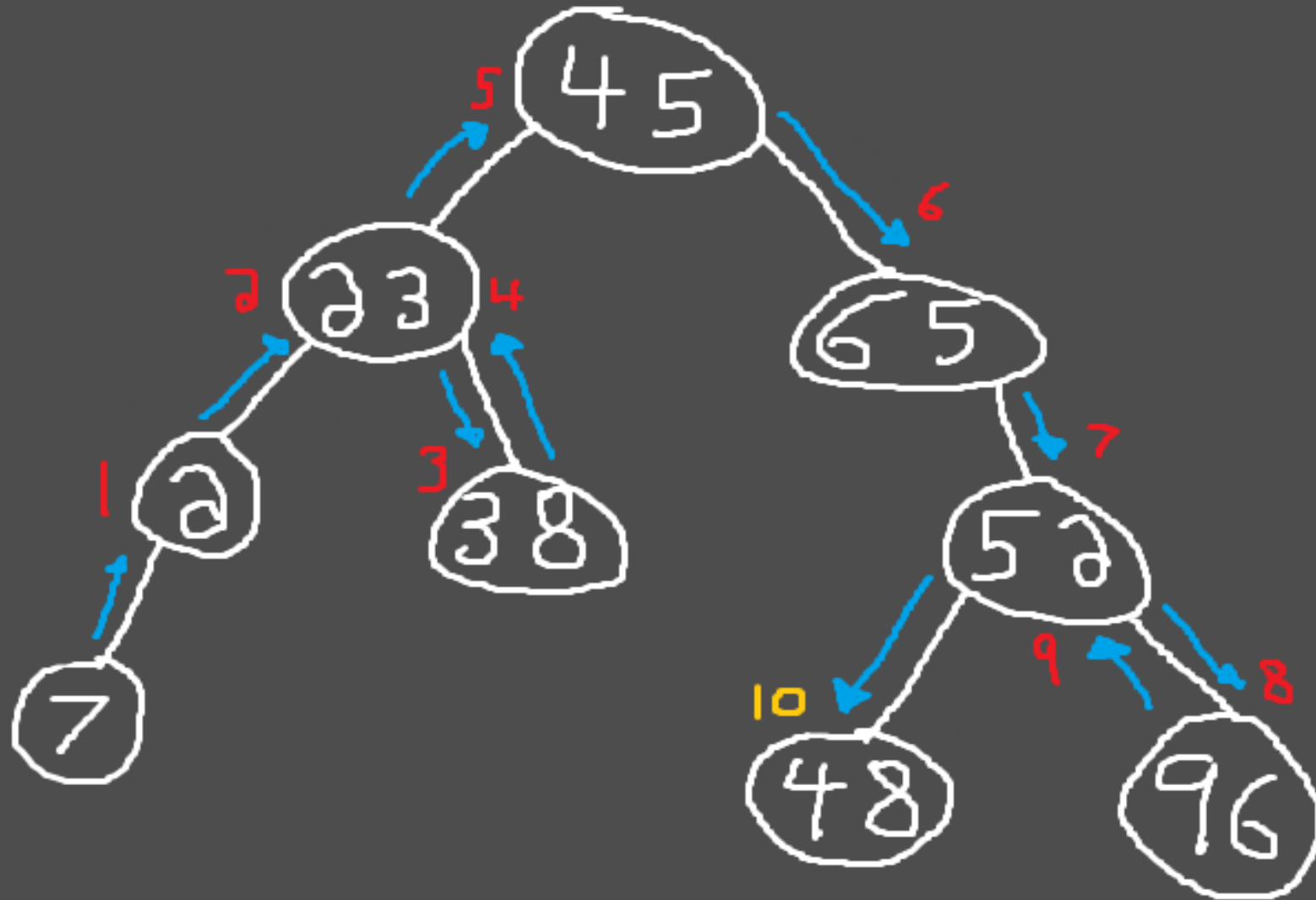
3. "Hacer recorrido in orden y post orden del árbol dibujado en la 1."

In-orden: 7, 2, 23, 38, 45, 65, 48, 52, 96.

Post-orden: 7, 2, 38, 23, 48, 96, 52, 65, 45.

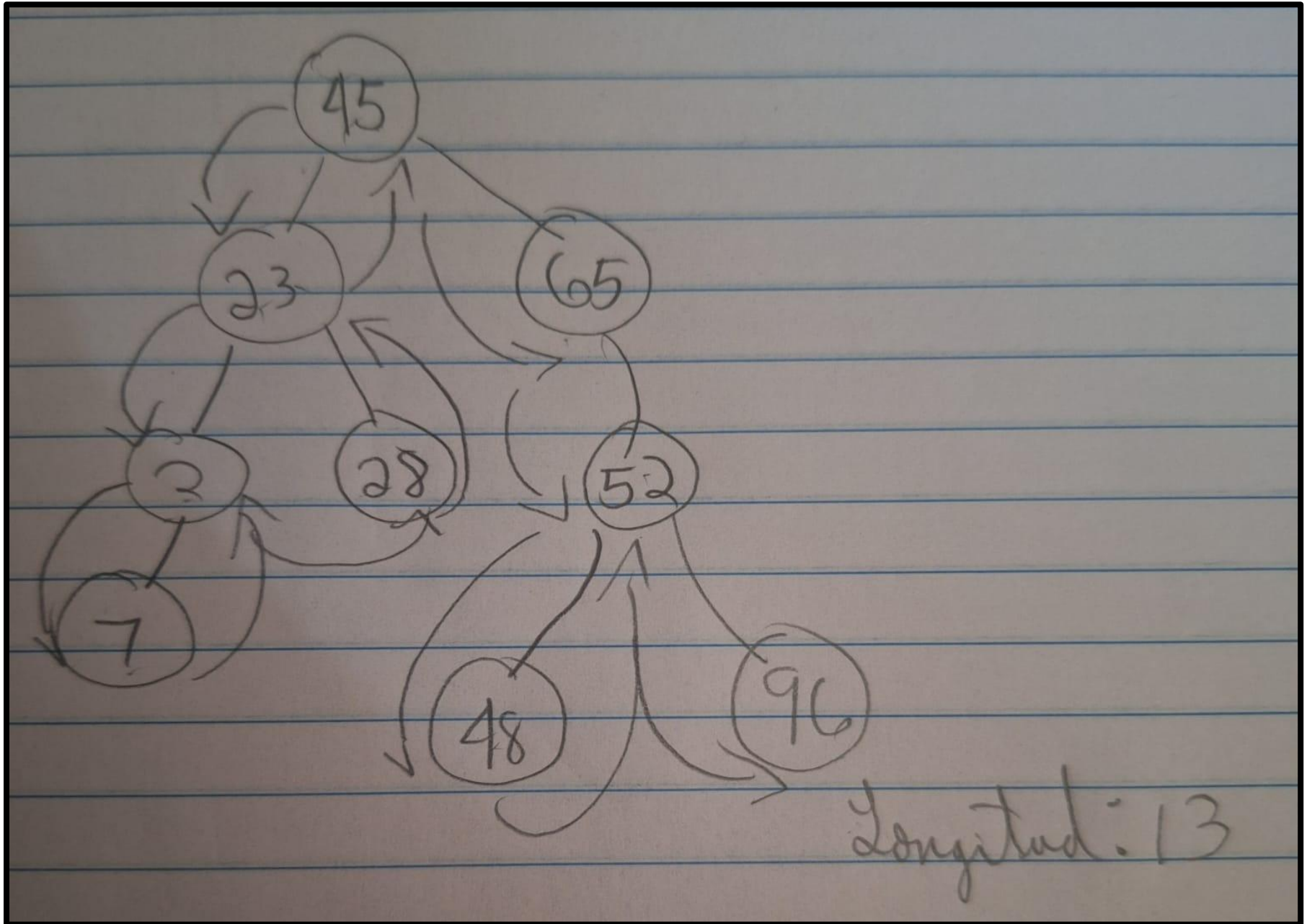
4. "Longitud del nodo 7 al 48 recorriendo todos los nodos."

7-2-23-38-23-45-65-52-96-52-48



5. "Longitud desde la raíz, iniciando desde el lado izquierdo y recorriendo todos los nodos del árbol."

45-23-2-7-2-23-28-45-65-52-48-52-96



6. "Agregar pre-orden y post-orden al siguiente código in-orden 30, 35, 40, 45, 50, 55, 60 (Captura de pantalla de salida de ambos)."

Como el código era el mismo que el de el anterior trabajo, decidí solo añadirle los métodos para recorrer el árbol de las maneras que se pidieron.

```
49 void preOrden(){ // El método para recorrer el árbol en pre orden
50     preOrdenRec(raiz);
51 }
52
53 void preOrdenRec(Nodo raiz){ // Recorrido en orden de: raíz, izquierda, derecha
54     if (raiz != null) {
55         System.out.print(raiz.valor + " ");
56         preOrdenRec(raiz.izquierdo);
57         enOrdenRec(raiz.derecho);
58     }
59 }
60 void postOrden(){ // El método para recorrer el árbol en post orden
61     postOrdenRec(raiz);
62 }
63
64 void postOrdenRec(Nodo raiz){ // Recorrido en orden de: izquierda, derecha, raíz
65     if (raiz != null) {
66         preOrdenRec(raiz.izquierdo);
67         enOrdenRec(raiz.derecho);
68         System.out.print(raiz.valor + " ");
69     }
70 }
```

```
Output - arbol (run) x
run:
Recorrido en orden del arbol binario:
30 35 40 45 50 55 60
Recorrido pre orden del arbol binario:
45 35 30 40 50 55 60
Recorrido post orden del arbol binario:
35 30 40 50 55 60 45
BUILD SUCCESSFUL (total time: 0 seconds)
```