



UNIVERSIDAD TECNOLÓGICA DE PANAMÁ CENTRO REGIONAL DE
CHIRIQUÍ
FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES



CARRERA:
Gestión y Desarrollo de Software

ACTIVIDAD No. 13

LABORATORIO No. 8

“Laboratorio 8”

ASIGNATURA: Estructura de Datos II

DOCENTE:
Profa. Nunehar Mondul

ESTUDIANTE/s:
Jorge Jiménez (4-826-874)

-

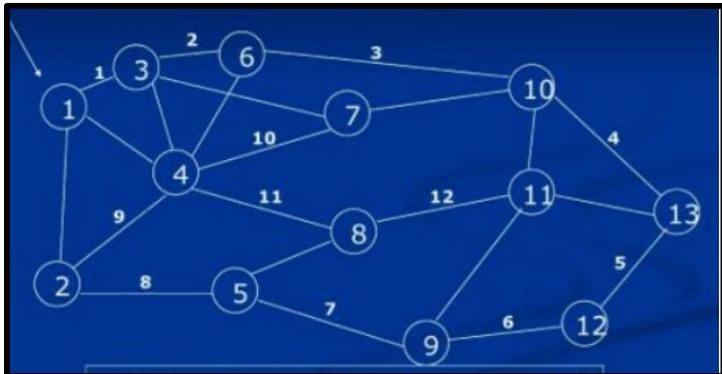
-

I SEMESTRE 2025

FECHA:
06/10/2025

Desarrollo

1. "Hacer recorrido en anchura y profundidad del siguiente grafo."



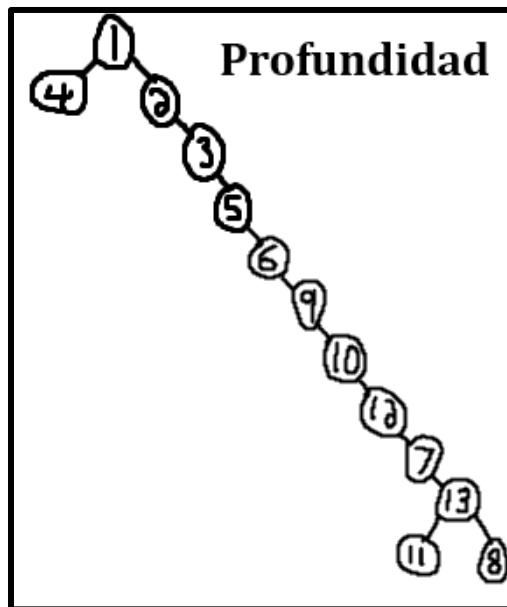
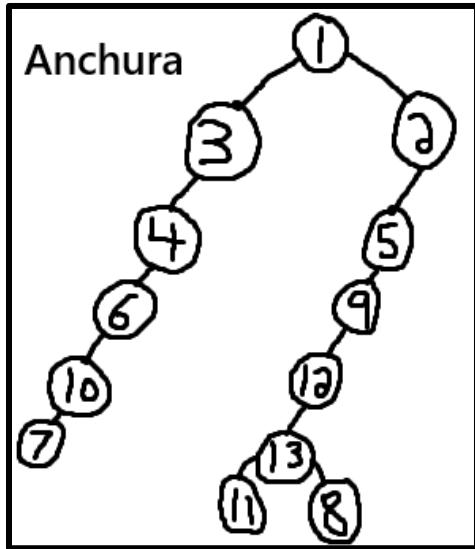
Anchura: 1-3-2-4-5-6-9-10-12-7-13-11-8

Profundidad: 1-2-3-5-6-9-10-12-7-13-8-11-4

2. "En el grafo de la 1 los nodos representan los puntos de entrega de una línea de Courier. Si no tiene peso significa que el camión no usa esa ruta. ¿Qué nodo es fundamental para llegar al nodo 4?"

-El nodo 2 es fundamental para llegar al 4, ya que es el único con peso que conecta con ella.

3. "Hacer arboles binarios resultantes de los recorridos."



4. “¿Que nodos son fundamentales para lograr hacer recorrido en profundidad y en anchura del grafo?”

-Diria que los nodos fundamentales para ambos recorridos serian el 4 y el 8.

5. “¿Cuál es la utilidad de los recorridos de anchura y profundidad en el área de redes?”

-Los recorridos en anchura y profundidad son útiles en redes para encontrar rutas óptimas y analizar la estructura de conexión, como detectar ciclos, componentes conexas o caminos más cortos.

6. "Para el siguiente código documentar, dibujar grafo."

```
1 package grafos4;
2 import java.util.LinkedList;
3 import java.util.Queue;
4
5 public class anchura {
6
7     private int vertices; // Número de vértices en el grafo
8     private LinkedList<Integer>[] listaAdyacencia; // Lista de adyacencia
9
10    public anchura(int vertices) {
11        this.vertices = vertices;
12        listaAdyacencia = new LinkedList[vertices];
13        for (int i = 0; i < vertices; i++) {
14            listaAdyacencia[i] = new LinkedList<>();
15        }
16    }
17
18    // Agrega una arista desde 'origen' hasta 'destino'
19    public void agregarArista(int origen, int destino) {
20        listaAdyacencia[origen].add(destino);
21    }
22
23    // Realiza recorrido en anchura desde un nodo inicial
24    public void recorridoEnAnchura(int inicio) {
25        boolean[] visitado = new boolean[vertices];
26        Queue<Integer> cola = new LinkedList<>();
27
```

```
28         visitado[inicio] = true;
29         cola.add(inicio);
30
31         System.out.println("Recorrido en anchura desde el nodo " + inicio + ":");
32
33         while (!cola.isEmpty()) {
34             int nodo = cola.poll();
35             System.out.print(nodo + " ");
36
37             for (int adyacente : listaAdyacencia[nodo]) {
38                 if (!visitado[adyacente]) {
39                     visitado[adyacente] = true;
40                     cola.add(adyacente);
41                 }
42             }
43         }
44
45
46         public static void main(String[] args) {
47             anchura grafo = new anchura(6);
48
49             grafo.agregarArista(0, 1);
50             grafo.agregarArista(0, 2);
51             grafo.agregarArista(1, 3);
52             grafo.agregarArista(1, 4);
53             grafo.agregarArista(2, 5);
54
55             grafo.recorridoEnAnchura(0);
56         }
57     }
```

Output - grafos4 (run) ×
run:
Recorrido en anchura desde el nodo 0:
0 1 2 3 4 5 BUILD SUCCESSFUL (total time: 0 seconds)

