Actividad 4 Diseña e implementa un sistema que modele una red social utilizando grafos. Cada usuario será un nodo, y las amistades entre usuarios serán las aristas. El sistema debe permitir: Agregar usuarios a la red, cada uno con un identificador único y nombre. Conectar usuarios indicando que son amigos (relación bidireccional). Implementar un método para realizar un recorrido DFS (Depth First Search) desde un usuario dado, mostrando los usuarios alcanzables a través de sus amistades. Implementar un recorrido BFS (Breadth First Search) para explorar la red desde un usuario dado, siguiendo las conexiones de amistad.

import java.util.\*;

class Usuario {

private int id;

private String nombre;

public Usuario(int id, String nombre) {

this.id = id;

this.nombre = nombre;

}

public int getId() {

return id;

}

public String getNombre() {

return nombre;

}

@Override

public String toString() {

return "Usuario{" +

"id=" + id +

", nombre='" + nombre + '\'' +

'}';

}

}

class RedSocial {

private Map<Usuario, LinkedList<Usuario>> listaAdyacencia;

public RedSocial() {

listaAdyacencia = new HashMap<>();

}

public void agregarUsuario(Usuario usuario) {

listaAdyacencia.putIfAbsent(usuario, new LinkedList<>());

}

public void conectarUsuarios(Usuario usuario1, Usuario usuario2) {

listaAdyacencia.get(usuario1).add(usuario2);

listaAdyacencia.get(usuario2).add(usuario1); // Relación bidireccional

}

public void DFS(Usuario inicio) {

Set<Usuario> visitados = new HashSet<>();

DFSUtil(inicio, visitados);

}

private void DFSUtil(Usuario actual, Set<Usuario> visitados) {

visitados.add(actual);

System.out.println(actual);

for (Usuario amigo : listaAdyacencia.get(actual)) {

if (!visitados.contains(amigo)) {

DFSUtil(amigo, visitados);

}

}

}

public void BFS(Usuario inicio) {

Set<Usuario> visitados = new HashSet<>();

Queue<Usuario> cola = new LinkedList<>();

visitados.add(inicio);

cola.add(inicio);

while (!cola.isEmpty()) {

Usuario actual = cola.poll();

System.out.println(actual);

for (Usuario amigo : listaAdyacencia.get(actual)) {

if (!visitados.contains(amigo)) {

visitados.add(amigo);

cola.add(amigo);

}

}

}

}

}

public class SistemaRedSocial {

public static void main(String[] args) {

RedSocial redSocial = new RedSocial();

Usuario usuario1 = new Usuario(1, "Juan");

Usuario usuario2 = new Usuario(2, "Maria");

Usuario usuario3 = new Usuario(3, "Pedro");

Usuario usuario4 = new Usuario(4, "Ana");

redSocial.agregarUsuario(usuario1);

redSocial.agregarUsuario(usuario2);

redSocial.agregarUsuario(usuario3);

redSocial.agregarUsuario(usuario4);

redSocial.conectarUsuarios(usuario1, usuario2);

redSocial.conectarUsuarios(usuario1, usuario3);

redSocial.conectarUsuarios(usuario2, usuario4);

redSocial.conectarUsuarios(usuario3, usuario4);

System.out.println("Recorrido DFS desde Juan:");

redSocial.DFS(usuario1);

System.out.println("Recorrido BFS desde Juan:");

redSocial.BFS(usuario1);

}

}