**UNIVERSIDAD POLITÉCNICA DE MADRID**
**ETS INGENIERÍA DE SISTEMAS INFORMÁTICOS**
**Departamento de Sistemas Informáticos**
***Asignatura: Estructuras de Datos***

# CURSO 2022-2023
# Práctica Grafos

---

## 1. OBJETIVOS:

- Adquirir destreza en el manejo de grafos y su representación en forma de matriz de adyacencias.
- Uso de la Clase GrafoMA (grafo implementado con una Matriz de Adyacencia).

## 2. DEFINICIÓN Y MANEJO DE UN GRAFO PARA LA CREACION DE UNA RED DE AMISTAD.

En la Escuela Técnica Superior de Ingeniería de Sistemas Informáticos se ha desarrollado una red de amistad, *AmigosETSISI.* En dicha red, los datos de las personas pertenecientes a esta se guardan en un vector y las relaciones de amistad entre estas personas, se reflejan en un grafo no dirigido representado por su matriz de Adyacencia.

Existen distintos grados de amistad:

Grado 1: Tienen grado 1 de amistad aquellas personas cuyos vértices representados en el grafo tiene una arista que los une.

Grado 2: Tienen grado 2 de amistad aquellas personas cuyos vértices están separados uno del otro por dos aristas y así sucesivamente.

Amigos indirectos son aquellos en los que el grado de amistad es superior a 1.

Si una persona es amiga de otra, en el grafo se representará a cada persona con un vértice y la relación de amistad entre ellas mediante una arista que une los dos vértices, indicando una relación de amistad directa o grado 1.

Las relaciones de amistad directa e indirecta forman lo que denominaremos grupos de amistad. Un grupo de amistad, por tanto, estará formado por todos aquellos vértices que tengan una relación de amistad tanto directa como indirecta (estos grupos coinciden con cada una de las componentes conexas del grafo).

Las clases que se utilizarán son:

- Persona. Incluye los datos relativos a una persona (nombre, dirección y teléfono).
- AmigosETSISI. Contiene los datos relativos a todas las personas integrantes de la red de amistad (**vector *contactos***), así como **el grafo** en el que indica las relaciones de amistad. Esta Clase AmigosETSISI proporcionados entre otros los siguientes métodos:
  - `public int devuelvePosNombre(String nombre)`: Devuelve la posición asociado a un nombre de persona en la tabla de contactos que además se corresponde con el vértice que le representa en el grafo.

- `public void imprimirRelaciones()`: Imprime la Matriz del Relaciones( Matriz de adyacencia del grafo) por consola.
- `public void mostrarRed()`: Imprime la información de la red y la Matriz de Relaciones por consola.
- `private boolean nombresIguales(String cad1, String cad2)`: Metodo que compara dos cadenas que representan dos nombres ignorando mayusculas y minúsculas
- `public void insertaRelacion(int o, int d)`: Método que añade una relación de amistad directa entre "o" y "d" ( inserta una arista en la Matriz de Relaciones con origen en "o" y destino en "d" )
- `public boolean existeRelacion(int o, int d)`: Método que indica si hay una relación de amistad directa entre dos personas representadas por los vértices "o" y "d" (devuelve cierto si existe una arista en el grafo que une esos dos vértices)
- `private boolean[]copia(boolean[] v1)`: Devuelve una copia del array v1
- `private boolean[] inicia_Visitados(boolean[] v1)`:Inicializa el array v1 a false.

En dicha red de amistad se desea incorporar cierta funcionalidad como operaciones de su interfaz:

- Cuáles son los grupos de relación (grupos disjuntos de personas relacionadas) conforman esta red y las personas que conforman cada grupo. En el grafo de ejemplo que hay al final de este documento, pueden verse 3 grupos de relación (componentes conexas), que se corresponden con los grupos:

  [Juan, José, Alicia, Alan, Guillermo] [Eva, Lucas, Clara] [Julio, Paula].

  Apartado 2.2. primer método**.**
- Disponer de un método para listar los amigos indirectos de una persona en concreto. Apartado 2.2. segundo método
- Indicar si dos personas no tienen ningún grado de amistad entre ellas. Apartado 2.2. tercer método.
- Si dos personas están en el mismo grupo de amistad mostrar los componentes de dicho grupo. Apartado 2.2. cuarto método

2.1. Descargar el proyecto **ED 5 Practica. Grafos** y abrirlo con *IntelliJ*.

2.2. En la clase **AmigosETSISI**, completar los siguientes método:

- `public int` **`mostrarGrupos`**`()`, devuelve el número de grupos de Amistad(disjuntos) que hay en la red de Amistad (número de componentes conexas del grafo) y muestra por pantalla el nombre de las personas que conforman cada grupo. En el grafo de ejemplo que hay al final de este documento, pueden verse 3 componentes conexas, que se

corresponden con 3 grupos: [Juan, José, Alicia, Alan, Guillermo] [Eva, Lucas, Clara] [Julio, Paula].

Este método devolvería 3 y visualizaría el siguiente resultado:

Juan José Alicia Alan Guillermo

Eva Lucas Clara

Julio Paula

- `public void` **`mostrarAmigosIndirectos`**`(String nombre)`, método que, si existe en el vector de contactos la persona con el nombre que se pasa como argumento, muestra los amigos indirectos grado 2 o más de esta persona. Si la persona no existe en el vector, se indicará en un mensaje.
  Para la búsqueda del nombre en el vector, se pueden utilizar el método de la clase implementado en la AmigosETSISI :
  `public int` **`devuelvePosNombre`**`(String nombre),`que devuelve el número de la posición que ocupa ese nombre en la tabla de contactos. Esta posición coincide con el número de vértice del grafo que representa a esa persona.

  Por ejemplo, en la red de amistad del ejemplo final de este documento, al realizar la llamada mostrarAmigosIndirectos("Alicia"), este método visualizaría el siguiente resultado:

  Amigos indirectos de Alicia(3):
  1: José
  4: Alan
  5: Guillermo

- `public boolean` **`noSonAmigos`**`(Persona p, Persona p1)`, método que devuelve cierto si las personas p y p1 no tienen ningún grado de amistad entre ellos. En caso de que p y/o p1 no se encuentren en el vector de contactos, este método devolverá falso.
- `public void` **`mostrarMiembrosSiAmigos`**`(Persona p, Persona p1)`, si las dos personas forman parte del mismo grupo de amistad, el método listará los miembros del grupo de amistad de las dos personas que pasan por parámetro. Si p y p1 no forman parte del mismo grupo de amistad, se tendrá que mostrar el nombre de p y de p1 indicando que no forman parte del mismo grupo de amistad. Finalmente, en caso de que p y/o p1 no se encuentren en el vector de contactos, el método visualizará un mensaje de error.

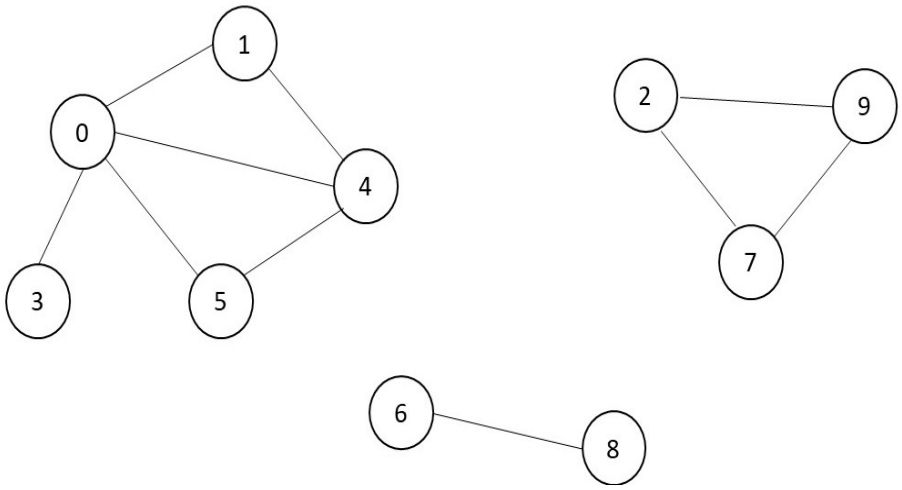
  Por ejemplo, en la red de amistad del ejemplo final de este documento, al realizar la llamada mostrarMiembrosSiAmigos(eva, clara), este método visualizaría el siguiente resultado:
  2: Eva
  7: Lucas
  9: Clara
  Y si se produce la llamada mostrarMiembrosSiAmigos(eva, paula):
  Eva no es del mismo grupo que Paula.

*2.3.* La clase Prueba comprobar el correcto funcionamiento de los métodos anteriores desarrollados. En dicha clase, crear un objeto de la clase **AmigosETSISI** en el que se inserten al menos estas 10 personas, así como las aristas del ejemplo que definen los tres grupos diferentes. A continuación, se muestra el ejemplo de vector de red social y un dibujo del grafo:

Personas:

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Juan<br>Nadie<br>C/ Uno<br>111111 | José<br>Cuervo<br>C/ Dos<br>222222 | Eva<br>Adán<br>C/ tres<br>333333 | Alicia<br>Maravillas<br>C/ Cuatro<br>777777 | Alan<br>Turing<br>C/ Cinco<br>999999 | Guillermo<br>Tell<br>C/ Seis<br>159267 | Julio<br>Catedrales<br>C/ Siete<br>123456 | Lucas<br>Pato<br>C/ Ocho<br>231465 | Paula<br>Vázquez<br>C/ Nueve<br>654321 | Clara<br>Oscuro<br>C/ Dos<br>123654 |

Grafo con las relaciones entre personas:




## 3. ENTREGA DE LA PRÁCTICA

**Únicamente se podrá implementar código en la Clase AmigosETSISI las demás clases proporcionadas deberán permaneces inalteradas.**

Se entregará el proyecto *IntelliJ* resultante de hacer la práctica "ED 5 Practica. Grafos", **que tendrá que tener exactamente ese nombre**. El nombre y el grupo al cual pertenece el alumno deberá aparecer como comentario al principio de la clase AmigosETSISI.

Para entregarlo, se comprimirá en un archivo, preferentemente ZIP, y se subirá a la plataforma. El nombre del archivo ZIP será el mismo: "ED Practica 5. Grafos.zip".

No olvide que el proyecto debe incluir las pruebas solicitadas en este enunciado, incluidas en el código del método *main* en las Clase *Prueba.*