

Universidad Autónoma de Nuevo León  
Facultad de Ingeniería Mecánica y Eléctrica

Laboratorio de Sistemas Adaptativos

Docente: Sara Elena Garza Villareal

Practica 2

“Redes Complejas”

Nombre	Matricula	Carrera	Semestre
Erick Fernando Romero Arias	1851501	ITS	5
Luis Adrián Lira Pérez	1848562	ITS	5
Mao de Jesús Lugo Oliva	1797546	ITS	4
Jesús Gabriel Rangel Hernández	1819217	ITS	7
Eduardo Salazar Treviño	1847972	ITS	5

Semestre Enero-Junio 2020

02 de Octubre de 2020

## **1.- Descripción de la práctica:**

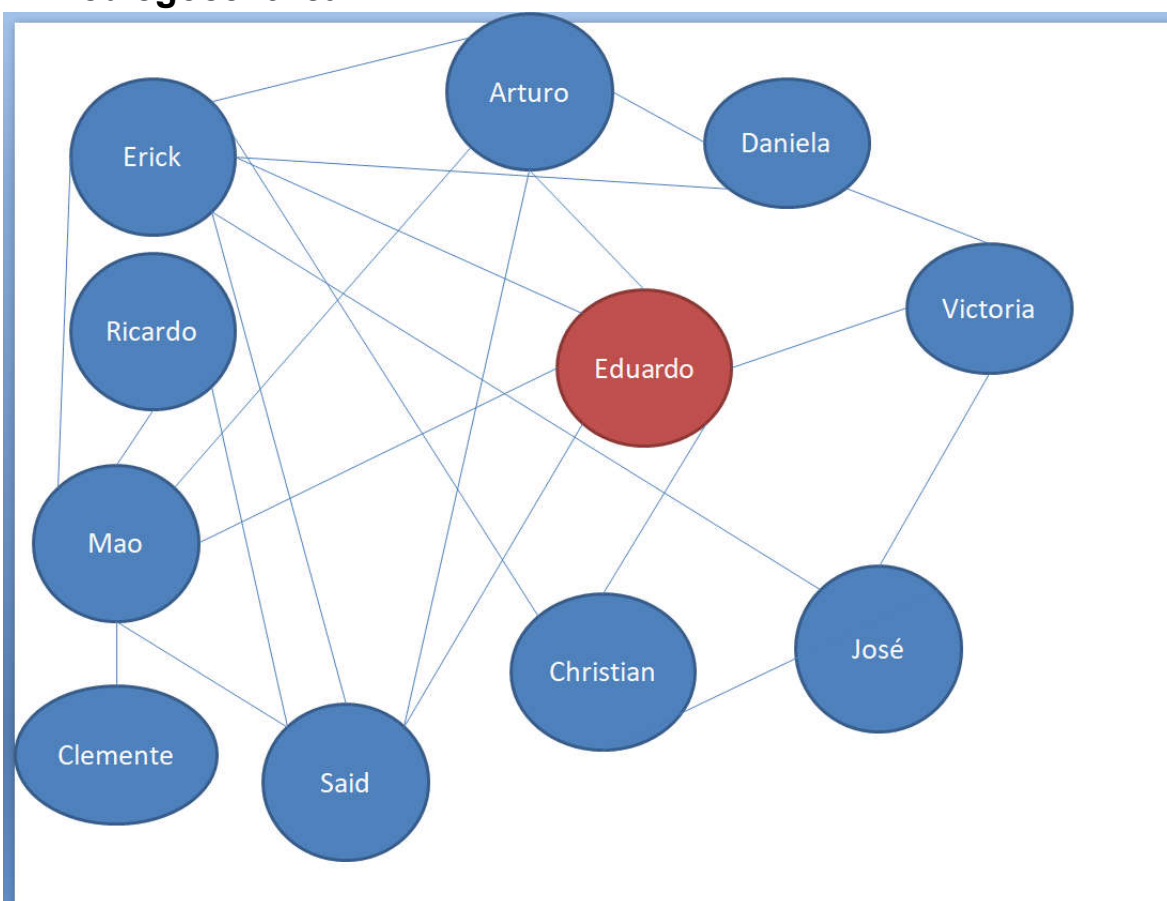
Para esta práctica, modelamos las relaciones sociales que tiene una persona en una red compleja, concretamente en la forma de un grafo no dirigido y no ponderado. Al principio planteamos una red egocéntrica, empezando con una persona y sus relaciones y ramificando poco a poco el grafo, hasta llegar a vértices que no estaban conectados necesariamente con el nodo que escogimos como el punto de partida del grafo. Sin embargo, como más adelante se verá, nos pasó algo muy curioso durante el proceso de estudio del grafo con las métricas.

Dichas métricas son el cálculo de la densidad del grafo, los grados de cada vértice y en base a ese dato su centralidad y por nuestra parte decidimos añadir también la métrica de diámetro, obteniendo primero las geodésicas o distancias de cada vértice, estructurándolas en una matriz de distancias.

De manera similar, también estructuramos la matriz de adyacencias del grafo, la cual nos sirvió mucho para poder representarlo de una manera más abstracta para el programa y para nuestro uso personal pudimos visualizar de una manera más compacta quienes eran amigos de quien.

Finalmente, con todos estos datos se usó el programa en Python proporcionado para la práctica, el cual intentamos modificar para poder automatizar el proceso de la obtención del diámetro pero nos dimos cuenta que no era práctico.

## 2.- Red egocéntrica:



	Eduardo	Erick	Arturo	Daniela	Victoria	Ricardo	Mao	Clemente	Said	Christian	Jose
Eduardo		1	1	0	1	0	1	0	1	1	1
Erick	1		1	1	0	0	1	0	1	1	1
Arturo	1	1		1	0	0	1	0	1	0	0
Daniela	0	1	1		1	0	0	0	0	0	0
Victoria	1	0	0	1		0	0	0	0	0	1
Ricardo	0	0	0	0	0		1	0	1	0	0
Mao	1	1	1	0	0	1		1	1	0	0
Clemente	0	0	0	0	0	0	1		0	0	0
Said	1	1	1	0	0	1	1	0		0	0
Christian	1	1	0	0	0	0	0	0	0		1
Jose	1	1	0	0	1	0	0	0	0	1	

MATRIZ.txt: Bloc de notas

Archivo	Edición	Formato	Ver	Ayuda						
0	1	1	0	1	0	1	0	1	1	1
1	0	1	1	0	0	1	0	1	1	1
1	1	0	1	0	0	1	0	1	0	0
0	1	1	0	1	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	1
0	0	0	0	0	0	1	0	1	0	0
1	1	1	0	0	1	0	1	1	0	0
0	0	0	0	0	0	1	0	0	0	0
1	1	1	0	0	1	1	0	0	0	0
1	1	0	0	0	0	0	0	0	0	1
1	1	0	0	1	0	0	0	0	1	0

	Eduardo	Arturo	Daniela	Victoria	José	Christian	Said	Erick	Mao	Ricardo	Clemente
Eduardo	0	1	2	1	2	1	1	1	1	2	2
Arturo	1	0	1	2	2	2	1	1	1	2	2
Daniela	2	1	0	1	2	2	2	1	2	3	3
Victoria	1	2	1	0	1	2	2	2	2	3	3
José	2	2	2	1	0	1	2	1	2	3	3
Christian	1	2	2	2	1	0	2	1	2	3	3
Said	1	1	2	2	2	2	0	1	1	1	2
Erick	1	1	1	2	1	1	1	0	1	2	2
Mao	1	1	2	2	2	2	1	1	0	1	1
Ricardo	2	2	3	3	3	3	1	2	1	0	2
Clemente	2	2	3	3	3	3	2	2	1	2	0

Anexamos también la matriz de distancias que obtuvimos.

### 3.- Capturas de pantalla: fragmentos de código y ejecución:

a. Obtención de la cantidad de vértices de la red (n)

```

try:
    archivo=open(sys.argv[1], 'r')
    lineas=archivo.readlines()
    archivo.close()
except IOError:
    print('No se pudo abrir el archivo')
    sys.exit(1)

n=len(lineas)

```

b. Obtención de la cantidad de aristas de la red (m)

```
for linea in lines:  
    linea=linea.strip()  
    celdas=linea.split()  
    grado=0  
    for celda in celdas:  
        m=m + int(celda)  
        grado=grado + int(celda)  
    grados.append(grado)  
  
m=m/2
```

c. Obtención del grado para cada vértice

```
for linea in lines:  
    linea=linea.strip()  
    celdas=linea.split()  
    grado=0  
    for celda in celdas:  
        m=m + int(celda)  
        grado=grado + int(celda)  
    grados.append(grado)
```

d. Calculo de centralidad de grado para cada vértice

```
for i in range(len(grados)):  
    centralidad=grados[i]/(n-1)
```

e. Calculo de densidad de la red

```
densidad=2*m/(n*(n-1))
```

f. Ejecución del programa en la terminal

```
→ eduar cd Documents/AG-EN2021/Adaptativos/lab/p2
→ p2 python3 analisis_red_social.py MATRIZ.txt
n: 11
m: 23.0
Densidad: 0.418181818181815
Grado v0=7 Centralidad: 0.7
Grado v1=7 Centralidad: 0.7
Grado v2=5 Centralidad: 0.5
Grado v3=3 Centralidad: 0.3
Grado v4=3 Centralidad: 0.3
Grado v5=2 Centralidad: 0.2
Grado v6=6 Centralidad: 0.6
Grado v7=1 Centralidad: 0.1
Grado v8=5 Centralidad: 0.5
Grado v9=3 Centralidad: 0.3
Grado v10=4 Centralidad: 0.4
→ p2
```

#### 4. Preguntas de análisis

a. ¿Qué representa el grafo que dibujaste?

Representa algunas de las amistades y relaciones sociales de cada uno de los miembros que aparecen en el grafo, concretamente centrándose en Eduardo.

b. ¿Qué tipo de red compleja analizaste?

Red social: amistad

c. ¿Quién es el mas popular (o los mas populares) en tu red? ¿ En que te basas para decir eso? ¿Qué metrica de las vistas en el laboratorio te indica este grado de popularidad?

En este caso hay dos populares, los cuales son: Eduardo y Erick, ya que tienen la misma centralidad, ya que el program de salida nos que los vértices V0 y V1 tienen un mismo grado y por subsecuente, su centralidad es la misma.

```
Grado v0=7 Centralidad: 0.7
Grado v1=7 Centralidad: 0.7
```

d. ¿Podría decirse que los amigos de tu red, en general, son todos amigos entre ellos o no? ¿Por qué o porque no? ¿Qué métrica de las vistas en laboratorio te ayuda a decir esto y porque?

No, ya que desde el momento en el que hay 0s en la matriz de adyacencia concluimos que existen vértices que no están conectados entre ellos. Usamos las métricas de la matriz de adyacencia y la matriz de distancias también, ya que si hay distancias mayores a 1 significa que existe mínimo un intermediario en nuestra red entre esos 2 vertices. En nuestro caso, Ricardo y Clemente tienen múltiples distancias de 3 por lo que podemos decir que ellos son los más desconocidos para el resto de la red compleja.

## **5.- Aplicación:**

Algunas de los usos que le podemos dar a las herramientas de análisis de grafos que acabamos de usar se aplican en las redes sociales. Una aplicación que viene la mente es ofrecer un servicio adicional a, por ejemplo Twitter usando su API, que le permita al usuario si así lo desea obtener con quién es que se relaciona más (la menor distancia en sus relaciones), quién es la persona más parecida a el usuario (con el uso de subgrafos inducidos), quién es la persona más influyente en su red (centralidad) o qué tanto se conocen sus seguidores/seguídos entre ellos (densidad). Podría incluso sugerir también hacer unfollows de manera periódica usando el diámetro del grafo.

## **6.- Preguntas de reflexion**

a.- ¿Qué conocimientos nuevos adquiriste después de haber realizado esta práctica?

Como modelar relaciones complejas en una red compleja por medio de un grafo y su matriz de adyacencias, procesar esa matriz en Python para obtener métricas de evaluación de la red y como interpretar esas métricas.

b.- ¿Qué habilidades nuevas adquiriste despues de haber realizado esta practica?

Pasar relaciones a redes y trabajar con ellas a través de los conceptos de teoría de grafos.

c.- ¿Qué parte de la practica se te dificultó más? ¿Por qué?

Lo mas complicado en nuestro caso fue obtener el diámetro, ya que es tardado y se tiene que hacer cuidadosamente, ya que son muchos datos, teniendo que vaciar cada distancia en la matriz, por lo que no podíamos equivocarnos.

d.- ¿Qué parte de la practica te agradó mas?¿Porque?

Entender el codigo y como interactúa con la matriz, ya que el codigo ya estaba realizado, y la dinamica era entender el cómo funciona y como interactua con el archivo .txt.

e.- ¿Qué mejorarías de tu parte para la siguiente practica?

Poder realizar el código solos, implementar los algoritmos por nuestra propia cuenta, una disposición de trabajar en equipo mejor.