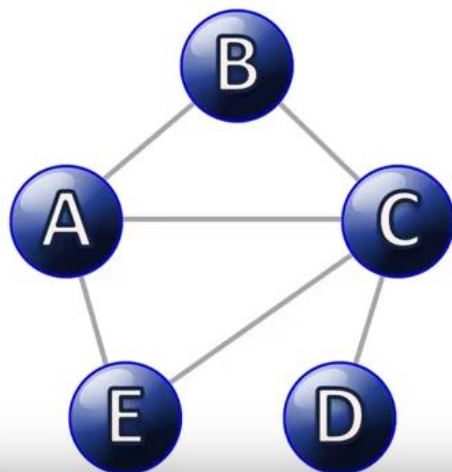




**Undirected Graph**



**Adjacency List**

A: B, C, E  
B: A, C  
C: A, B, D, E  
D: C  
E: A, C

# Complejidad Algorítmica

**Unidad 1: Comportamiento asintótico, métodos de búsquedas y grafos**

**Módulo 4: Teoría de Grafos**

# Complejidad Algorítmica

Semana 4 / Sesión 1

## MÓDULO 4: Teoría de Grafos



### Contenido

1. ¿Qué es un Grafo?
2. Tipos de Grafos
3. Casos de Uso para los grafos
4. Representación de un grafo
5. Grafos Completos, Subgrafos y Componentes



### Preguntas

# 1. ¿Qué es un Grafo?

¿Cómo definimos un Grafo?

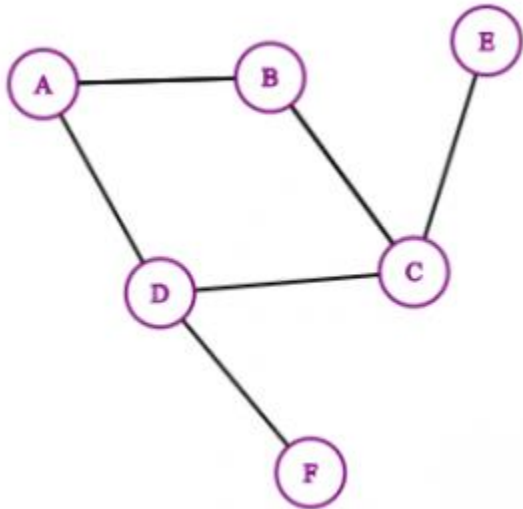


- Debemos referirnos primero a “**La teoría de grafos**” como el estudio de líneas y puntos.
- **La teoría de grafos** es un subcampo de las matemáticas que se ocupa de los gráficos, es decir, al estudio de la relación entre aristas y vértices.
- Formalmente, **un grafo es un par  $(V, E)$** , donde  **$V$**  es un conjunto finito de vértices (nodos o estados) y  **$E$**  un conjunto finito de aristas (arcos, bordes o ramas).

# 1. ¿Qué es un Grafo?

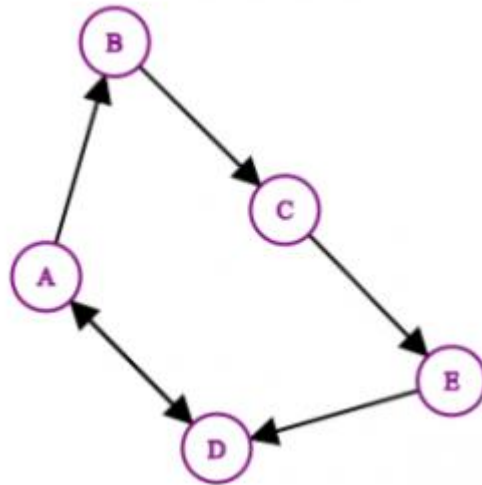
## ENTENDAMOS COMO ES LA ESTRUCTURA DE UN GRAFO

### Grafo no dirigido



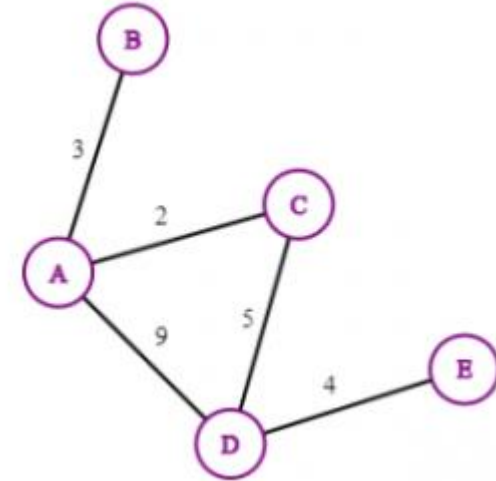
Cuando los bordes (también llamados aristas o ramas) no tienen una dirección.

### Grafo dirigido



Cuando los bordes tienen una dirección.  
También se le conoce como digrafo o direccionado

### Grafo ponderado

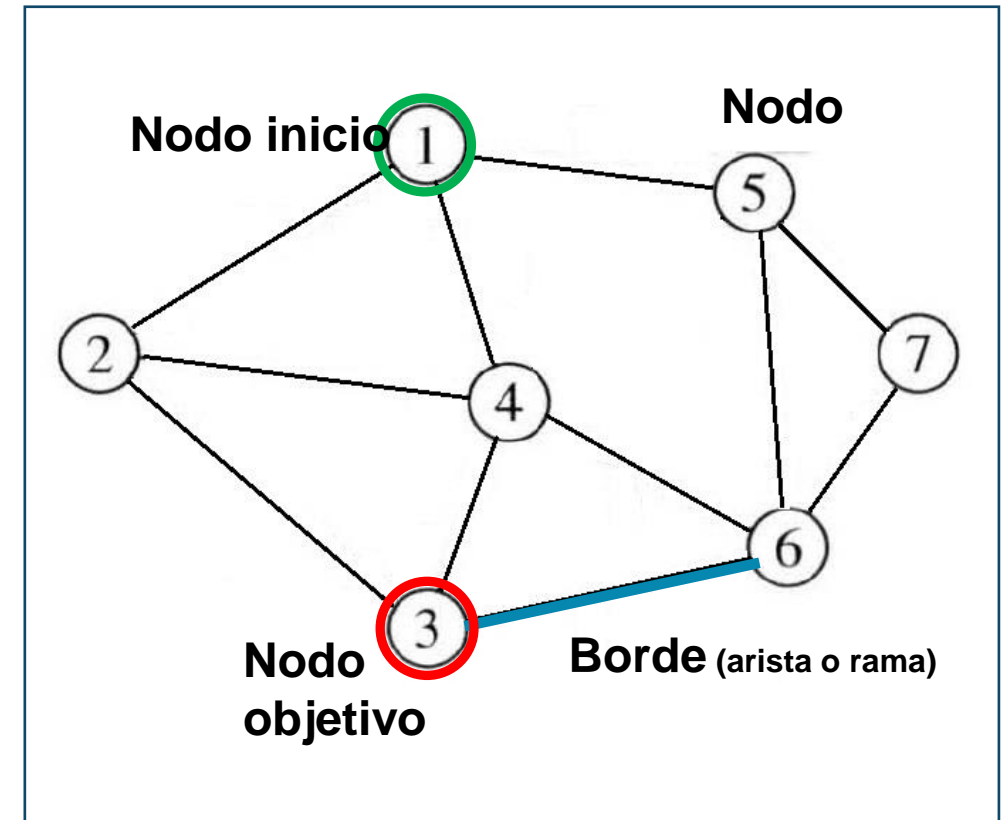


Cuando los bordes tienen un valor numérico.

# 1. ¿Qué es un Grafo?

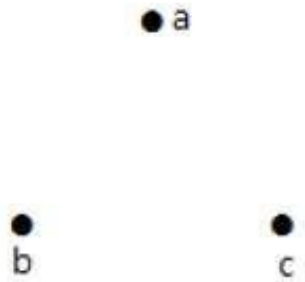
## ENTENDAMOS COMO ES LA ESTRUCTURA DE UN GRAFO

<b>Nodo :</b> (Estado o Vértice)	Todas las posibles posiciones o paradas con una identificación única.
<b>Borde :</b> (Arista o rama)	Cada conexión entre dos nodos.
<b>Transición :</b>	El acto de moverse entre nodos (o vértices)
<b>Nodo inicio:</b>	Dónde empezar a buscar (nodo inicial).
<b>Nodo objetivo:</b>	El objetivo para detener la búsqueda (nodo final).
<b>Espacio de búsqueda:</b>	Una colección de nodos, como todas las posiciones del tablero de un juego de mesa



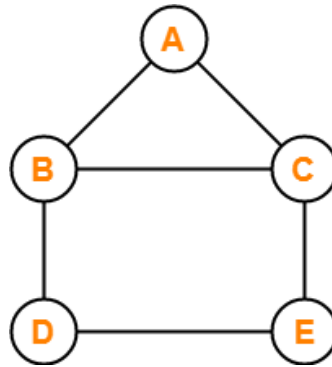
## 2. Tipos de Grafos

### GRAFO NULO



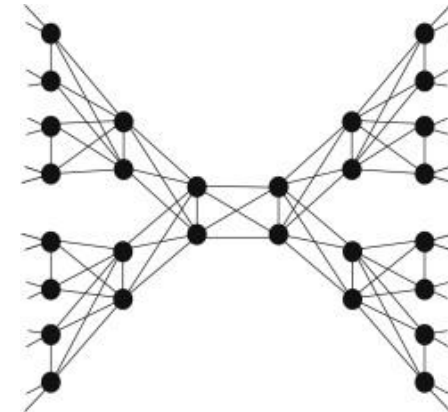
- Un grafo que no tiene bordes se llama grafo nulo.

### GRAFO FINITO



- Un grafo es **finito** si tiene un número finito de vértices y un número finito de aristas

### GRAFO INFINITO



- Un grafo es **infinito** si tiene un número infinito de vértices y un número infinito de aristas (contiene a todos los grafos finitos, como un subgrafo).

### Otros tipos de grafos:

- Grafo trivial, regular, completo, de ciclo, de rueda, etc.

### 3. Casos de Uso para los grafos

#### Redes Sociales





# 3. Casos de Uso para los grafos

## Redes Sociales

A partir de los datos recolectados de redes sociales, se realizan investigaciones de índole: social, educativo, económico y comercial.



**2019:** Redes en Twitter y la defensa de la mujer peruana

### Objetivos:

- Modelar redes de Twitter.
- Identificar a los principales actores influyentes por red.
- Identificar las relaciones entre los actores y sus comunidades.
- Ver relaciones existentes entre actores del Gobierno y de la sociedad civil.



# 3. Casos de Uso para los grafos

## Redes Sociales

### Minería de Datos



#### HASHTAGS:

#8M  
#Marcha8M  
#DialInternacionalDeLaMujer  
#DiaDeLaMujer  
#8marzo2019  
#8M2019

#### PALABRAS CLAVE:

Feminicidio  
Acoso



#### Programación en Python

- (1) Programa para captura de tweets por fecha y palabra clave
- (2) Programa para lectura y proceso de archivos de tweets
- (3) Programa para creación de archivos para grafos

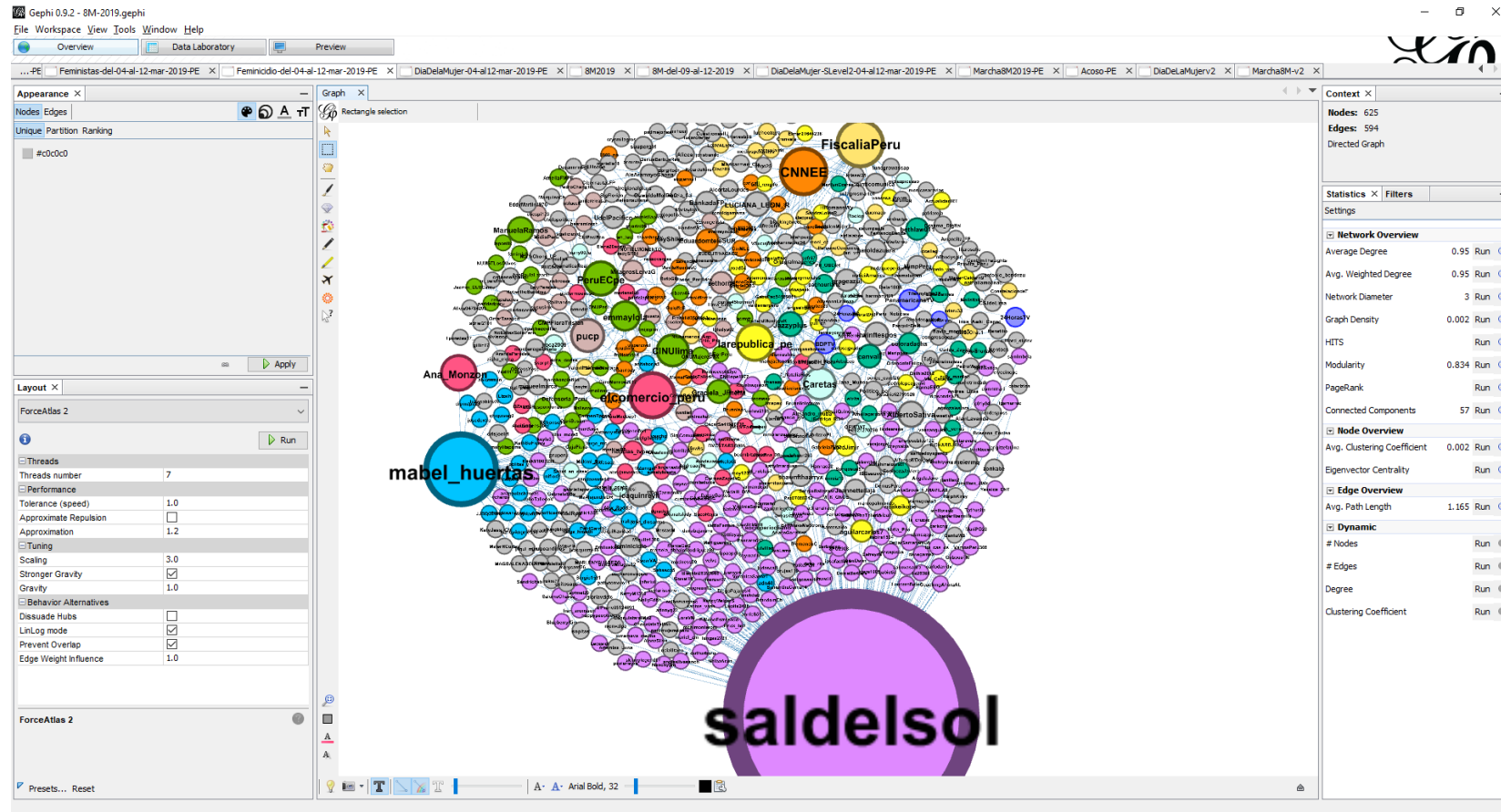


### Modelado y análisis de las Redes

- ◉ Consideraciones importantes:
  - Las redes se modelan a partir de los **Retweets** como tipo de interacción entre nodos.
  - Todas las redes son de tipo dirigidas.
- Las métricas calculadas son las siguientes:
  - **Densidad de la red**
  - **Centralidad**
  - **Modularidad de la red**
- La centralidad del vector propio o Eigenvector fue la métrica utilizada para la visualización de los nodos en cada una de las redes.

# 3. Casos de Uso para los grafos

## Redes Sociales



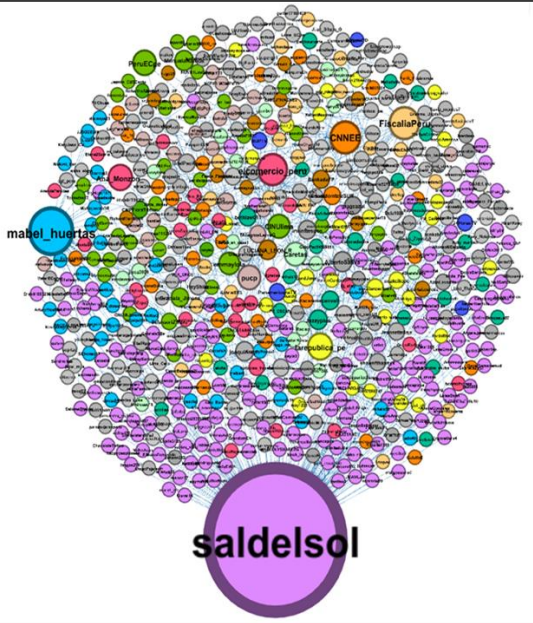
Tratamiento de los datos  
de los retweets como  
grafos.

# 3. Casos de Uso para los grafos

## Redes Sociales

### Modelado y análisis de las Redes

#### Red 5: Femicidio



#### Resumen de pruebas estadísticas

Tipo de red	Dirigida
Nro. de Nodos	625
Nro. de Conexiones	594
Densidad de la Red	0.003041025641025641
Modularidad	0.833
Núm. Comunidades	69

#### Nodos influyentes

Item	Nombre Nodo	Comunidad	Descripción
1	saldelsol		Presidente del Consejo de Ministros del Perú. <a href="https://twitter.com/saldelsol">https://twitter.com/saldelsol</a>
2	mabel_huertas		Periodista. Todas las mañanas en Buenos Días Perú @BDPTV por @PanamericanaTV <a href="https://twitter.com/mabel_huertas">https://twitter.com/mabel_huertas</a>
3	PeruECpe		Cuenta oficial de la sección Perú del diario El Comercio. <a href="https://twitter.com/PeruECpe">https://twitter.com/PeruECpe</a>

- La **densidad de la red** mide la proporción de enlaces que existen entre las relaciones posibles de una red en concreto.
- La **modularidad** es una medida de la estructura de las redes o grafos
- La **modularidad**, mide la fuerza de la división de una red en módulos, agrupaciones o comunidades

### 3. Casos de Uso para los grafos

#### Redes Sociales

## Análisis y Observaciones

#### ● Análisis de Comunidades

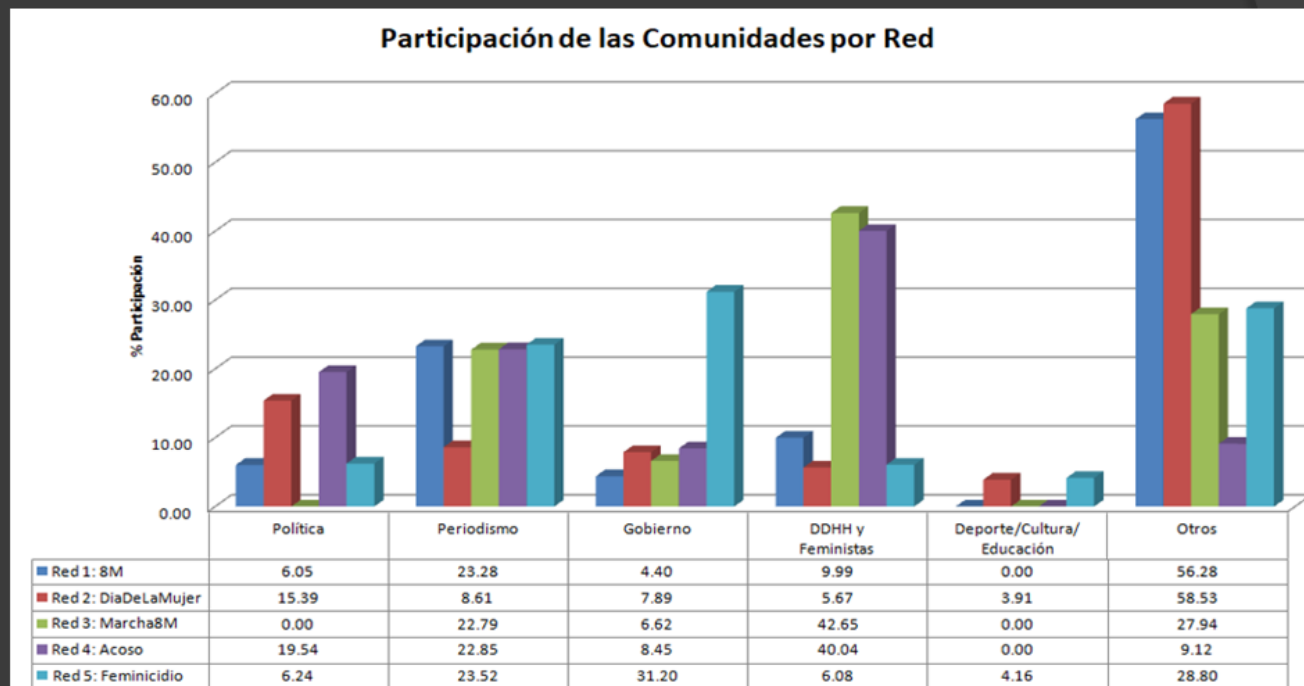
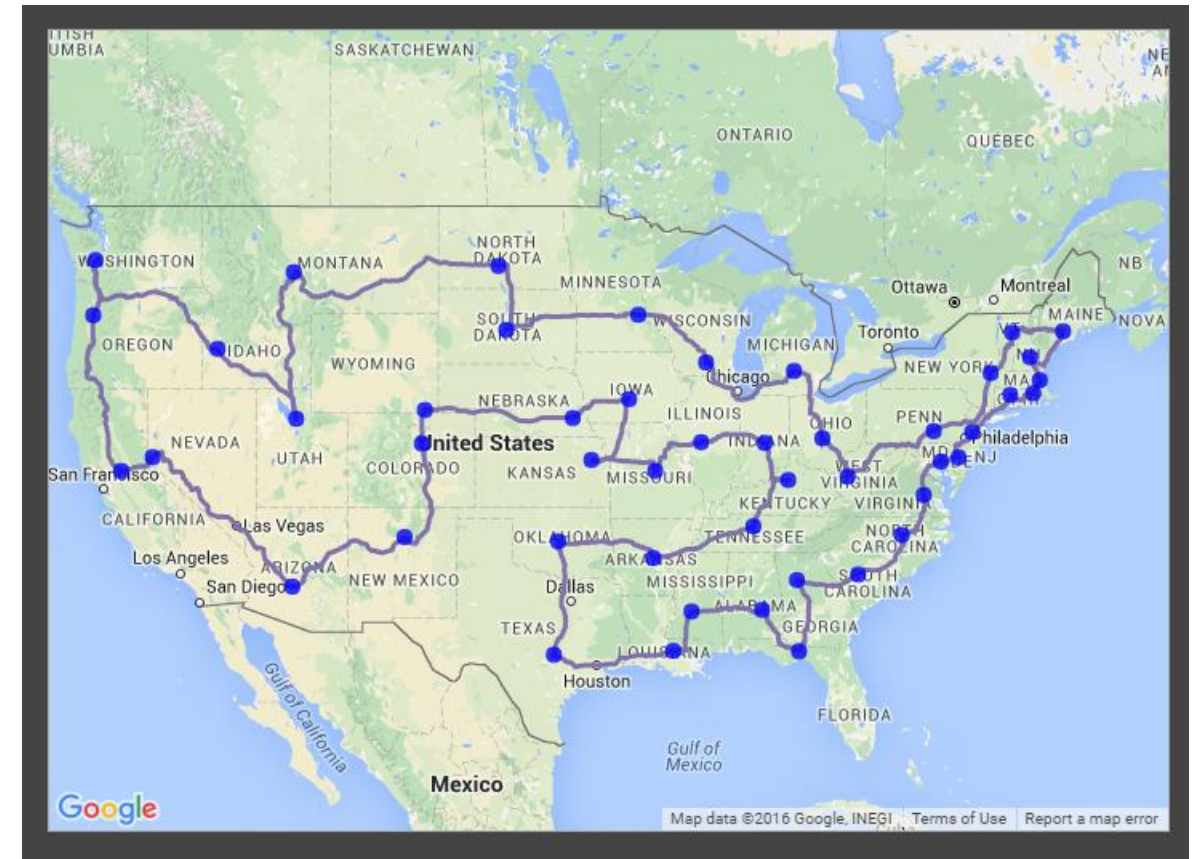


Figura 31. Tabla de participación de las Comunidades por Red

# 3. Casos de Uso para los grafos

## Sistemas Geográficos





### 3. Casos de Uso para los grafos

#### Redes Semánticas

Cuanto más a menudo aparece una palabra clave, mayor el tamaño de las letras y círculos.

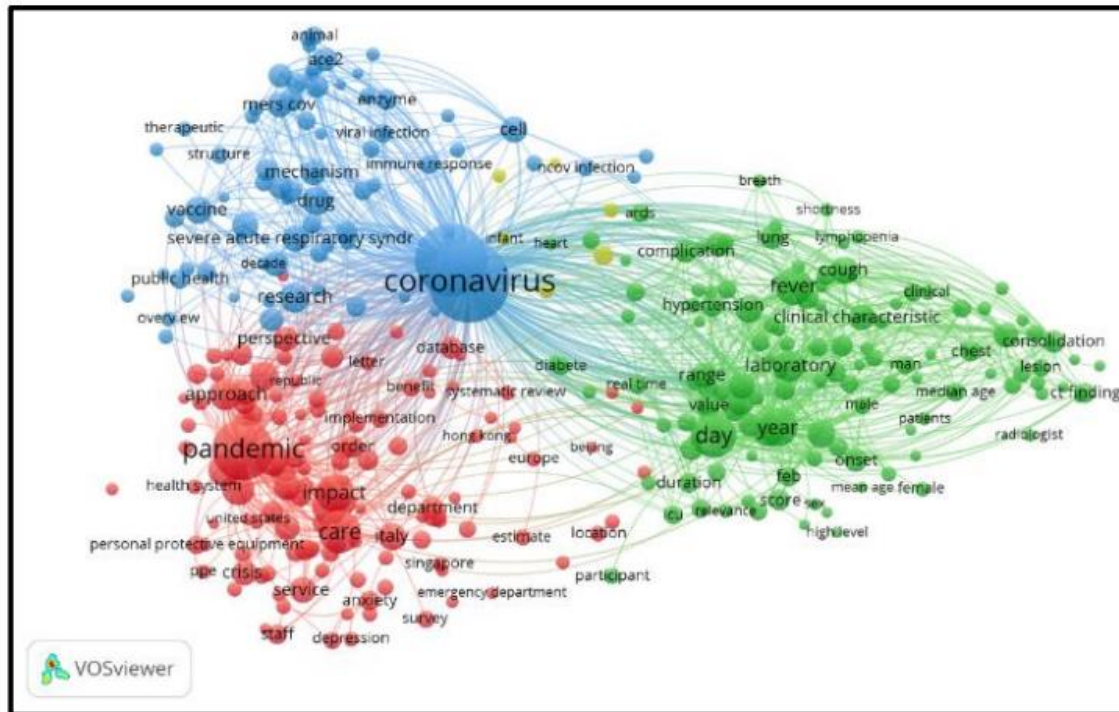


Figure 1. Visualization topic area using VOSviewer using network visualization

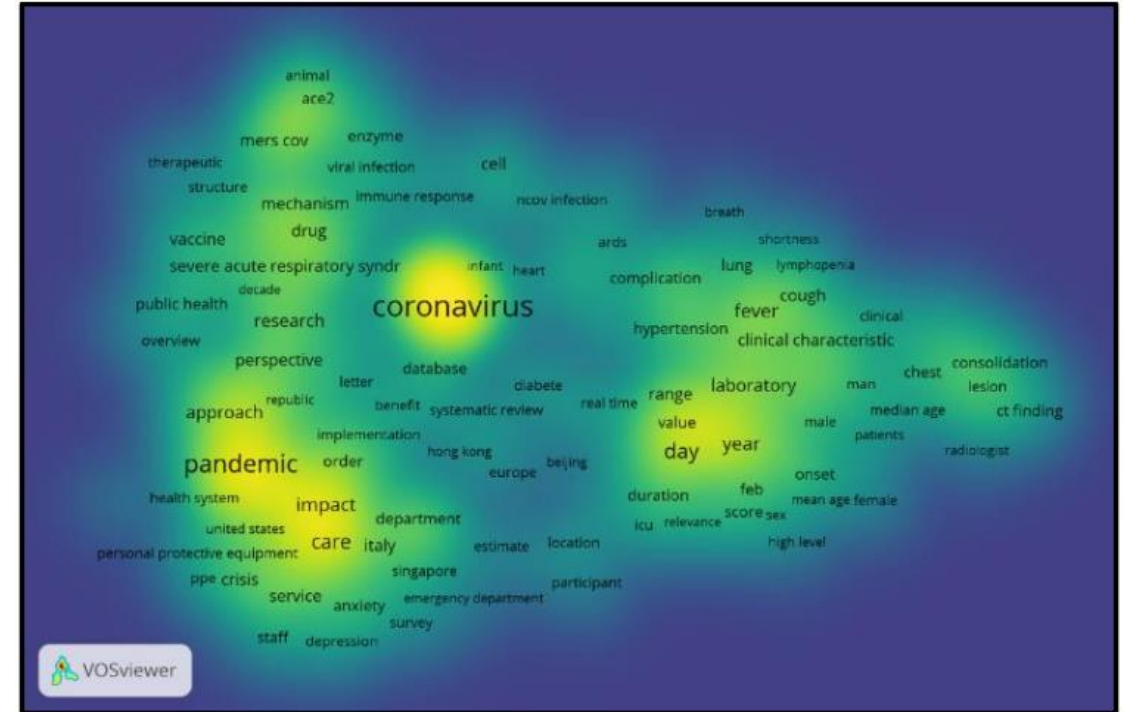
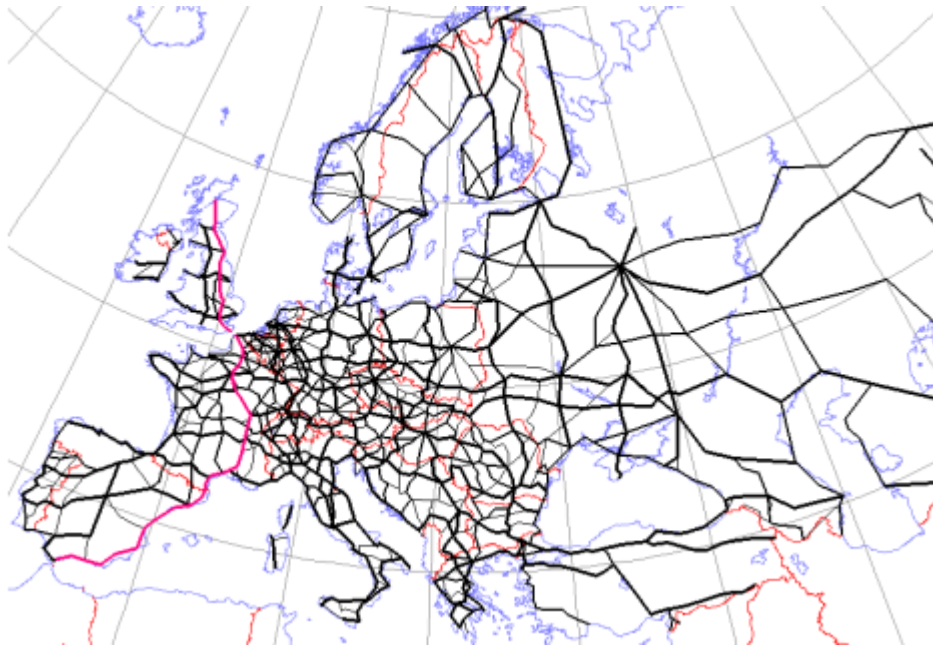


Figure 3. Visualization topic area using VOSviewer using density visualization



# 3. Casos de Uso para los grafos

## Redes Viales

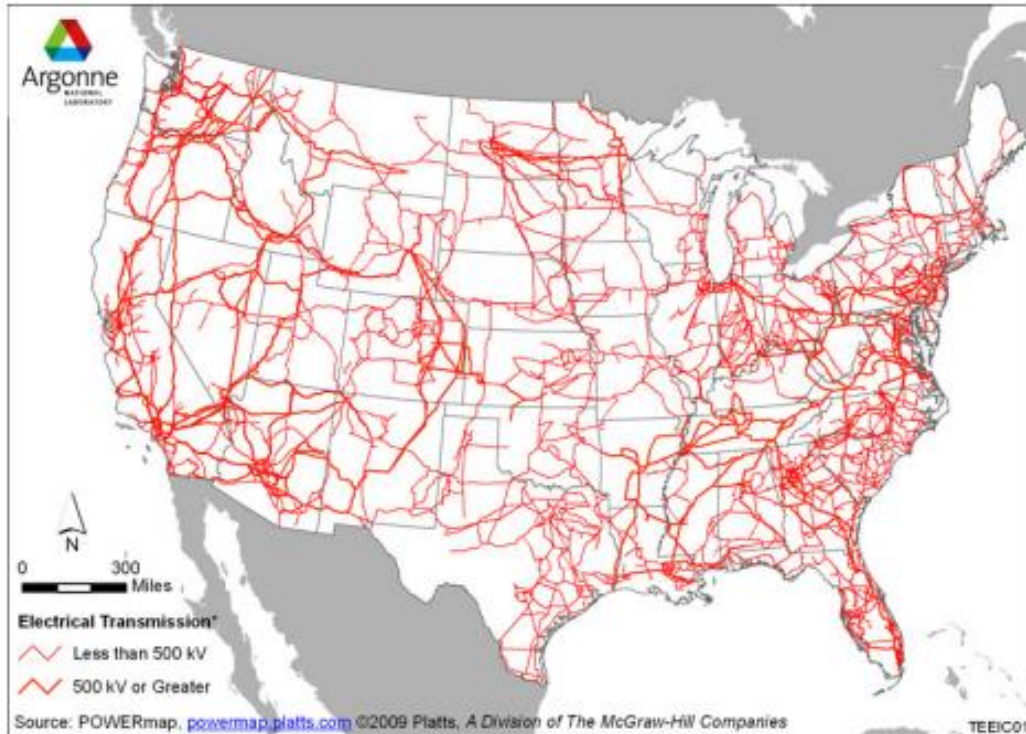


## Redes de transporte publico



### 3. Casos de Uso para los grafos

#### Redes de energía eléctrica

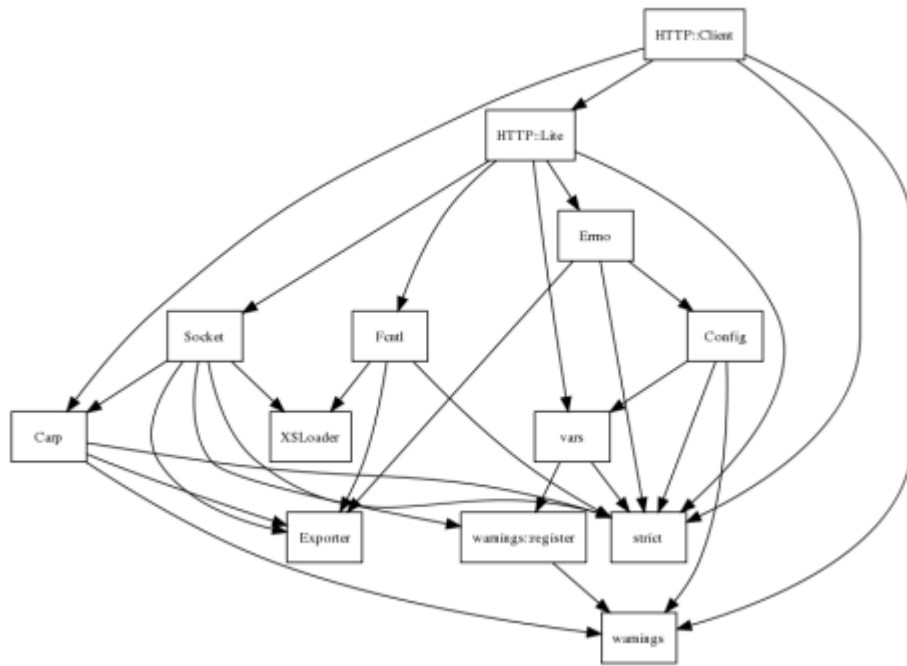


#### Redes de computadoras



### 3. Casos de Uso para los grafos

Redes de módulos (p.e. flujos de información)

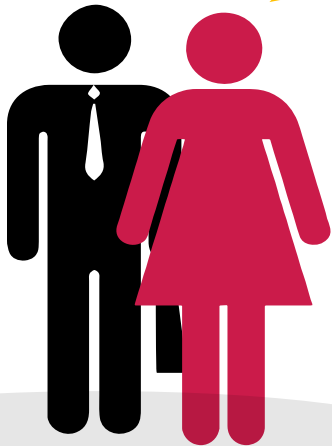


Redes de computadoras



## 4. Representación de un grafo

¿Cómo  
representamos un  
Grafo?



Un grafo lo representaremos mediante:

1. Una matriz de adyacencia.
2. Una lista de adyacencia.

Pero antes definamos el grado de un grafo y el grado de un nodo o vértice:

**Grado de un grafo** = Cantidad de nodos o vértices que contiene.

**Grado de un nodo** = Cantidad de aristas que son incidentes a él (de entrada o de salida).

## 4. Representación de un grafo

### Representación #1: Matriz de Adyacencia

- Dado un grafo G, este será representado por una matriz de  $n \times n$  que denominaremos **A**, donde:

$$A_{ij} = 1 \iff G \text{ tiene una arista } i \rightarrow j$$

Variante:  $A_{ij}$  = peso de la arista  $i \rightarrow j$

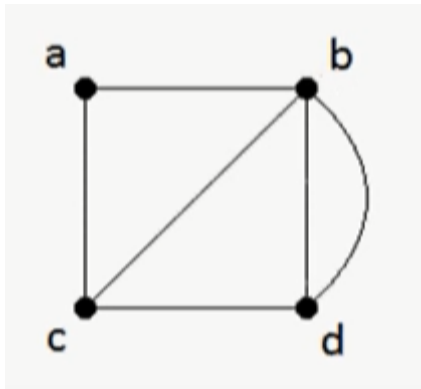
- El grafo está representado por una matriz cuadrada A de tamaño  $n \times n$ , donde n es el número de nodos o vértices.
- Si hay una arista (borde) entre un **vértice (nodo) i** y un **vértice (nodo) j**, entonces el elemento  $A(i, j)$  es 1, de lo contrario, es 0.

**Veamos un ejemplo...**

## 4. Representación de un grafo

**EJEMPLO #1:** Convertir un grafo en una matriz de adyacencia.

Dado el siguiente grafo:



- ¿Cuál es la matriz de adyacencia?

**SOLUCION:**

1. Contamos la cantidad de vértices o nodos y creamos una matriz de  $n * n$
2. Evaluamos las relaciones (aristas o bordes) entre cada par de la matriz  $(A_{i,j})$   
Asignamos 1 si existe una arista o 0 si no existe.

	a	b	c	d	Grado del nodo	
a	0	1	1	0	=	2
b	1	0	1	2	=	4
c	1	1	0	1	=	3
d	0	2	1	0	=	3

Cantidad de aristas que inciden en cada nodo.

- **Grado del grafo** = 4 (cantidad de nodos o vértices)
- Hemos creado una matriz **simétrica** (una matriz de orden  $n$  con el mismo número de filas y columnas donde su matriz traspuesta es igual a la matriz original).



## 4. Representación de un grafo

### EJEMPLO #2: Convertir una matriz de adyacencia en un grafo

Dada la siguiente matriz dirigida:

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

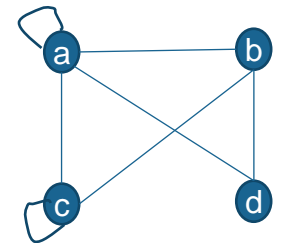
- ¿Cuál es el grafo? (visualmente)
- ¿Cuánto espacio es necesario para almacenar una matriz de adyacencia, en función a **n**, el número de vértices y **m**, el número de aristas?

### SOLUCION:

- ¿Que nos dice la matriz? Que tiene 4 nodos (grado 4), entonces representamos cada nodo con una letra.
- Calculamos el grado de cada nodo. ¿Es una matriz simétrica?

	a	b	c	d	grado
a	1	0	1	0	= 2
b	1	0	1	1	= 3
c	0	1	1	0	= 2
d	1	0	0	0	= 1

- Dibujamos los 4 nodos con sus etiquetas
- Dibujamos las aristas o bordes entre nodos para obtener el grafo
- No es una matriz simétrica.



# 4. Representación de un grafo

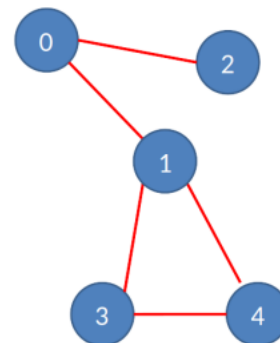
## Representación #2: Listas de Adyacencia

- Una forma sencilla de representar un grafo es a partir de una lista, o un arreglo, de  $|E|$  aristas, a la que llamamos una lista de aristas.
- Para representar una arista, solo tenemos un arreglo de dos números de vértices, o un arreglo de objetos que contienen los números de vértices sobre los que inciden las aristas.

### Elementos:

- Arreglo (o lista) de vértices (nodos)
- Arreglo (o lista) de aristas
- Cada arista apunta a sus vértices (nodos)
- Cada vértice (nodo) apunta a las aristas incidentes a él.

### Ejemplo:



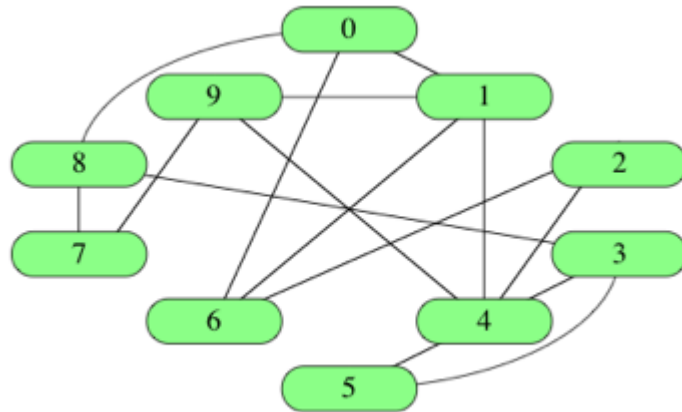
0		
1		
2		
3		
4		

<u>Vértice inicial</u>	<u>Vértices en los que incide</u>
0:	1,2
1:	0,3,4
2:	0
3:	1,4
4:	1,3

## 4. Representación de un grafo

**EJEMPLO #1:** Convertir un grafo en una lista de adyacencia.

Dado el siguiente grafo:



- ¿Cuál es la lista de adyacencia?

### SOLUCION:

1. Construimos una lista de vértices o nodos según el grafo
2. Identificamos la lista de vértices o nodos en los que incide cada nodo de la lista construida en el paso 1).

Lista de Vértices	<u>Vértices en los que incide</u>
-------------------	-----------------------------------

0	1,6,8
1	0,4,6, 9
2	4,6
3	4,5, 8
4	1,2,3,5,9
5	3,4
6	0,1,2
7	8,9
8	0,3,7
9	1,4,7

<u>Representación de la lista de aristas</u>
--

[ [0,1], [0,6], [0,8], [1,4], [1,6], [1,9], [2,4], [2,6], [3,4], [3,5],[3,8], [4,5], [4,9], [7,8], [7,9] ]
---

En algún lenguaje de programación

## 4. Representación de un grafo

### EJEMPLO #2: Convertir una lista de adyacencia en un grafo

A partir de la lista de adyacencia:

```
1 :  
2 : 3 4  
3 : 2 4  
4 : 5 7  
5 : 1 2 4  
6 : 3 5  
7 : 5
```

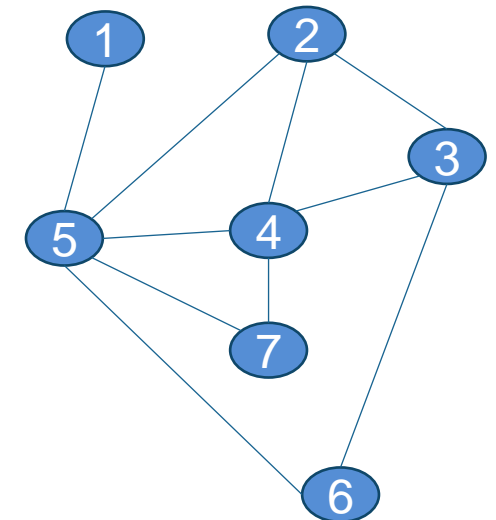
- ¿Cuál es el grafo?

### SOLUCION:

1. Construimos un grafo y lo etiquetamos con cada uno de los vértices de la lista.
2. Identificamos cada una de las incidencias desde cada uno de los vértices de la lista en 1) hacia los otros vértices.

**Lista de Vértices**      **Vértices en los que incide**

```
1 :  
2 : 3 4  
3 : 2 4  
4 : 5 7  
5 : 1 2 4  
6 : 3 5  
7 : 5
```



# 4. Representación de un grafo

## Ventajas / Desventajas: Matriz de Adyacencia vs Lista de Adyacencia

### Matriz de Adyacencia



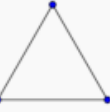
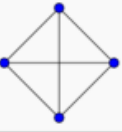

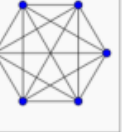


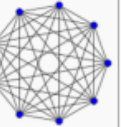
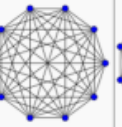
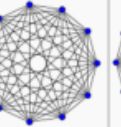
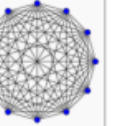
- Muy simple de implementar.
- Rápido para ver si hay una conexión entre dos nodos -  $O(1)$ .
- Lento para atravesar nodos adyacentes -  $O(|V|)$
- Alto desperdicio de memoria (en gráficos dispersos) -  $O(|V|^2)$
- Gráfico pesado implica solo almacenar peso en la matriz.
- Agregar/eliminar enlaces es solo cambiar la celda de matriz -  $O(1)$ .

### Lista de Adyacencia

- Lento para ver si hay una conexión entre los nodos  $u$  y  $v$  -  $O(\text{grado}(u))$ .
- Rápido para atravesar nodos adyacentes -  $O(\text{grado}(u))$ .
- Memoria bien utilizada -  $O(|V| + |E|)$ .
- Gráfico pesado implica agregar un campo a la lista.
- Eliminar enlace  $(u, v)$  implica recorrer la lista -  $O(\text{grado}(u))$ .

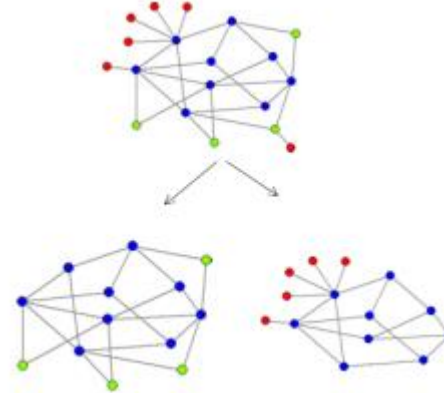
# 5. Grafos completos, Subgrafos y Componentes

## Grafos completos

$K_1: 0$	$K_2: 1$	$K_3: 3$	$K_4: 6$	$K_5: 10$	$K_6: 15$
					
$K_7: 21$	$K_8: 28$	$K_9: 36$	$K_{10}: 45$	$K_{11}: 55$	$K_{12}: 66$
					

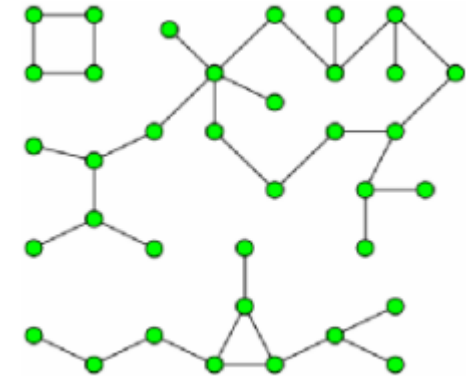
- Un **grafo es completo** si para cualquier pareja de vértices existe una arista que los une (en ambos sentidos si el grafo es no dirigido).
- El número máximo de aristas en un grafo de  $n$  vértices es:  $n(n - 1)/2$ .

## Subgrafos



- Un **subgrafo** se define como un grafo con vértices y aristas que son un subconjunto de un grafo.

## Componentes



- Un subconjunto de vértices (nodos) que forman un subgrafo completo.



# PREGUNTAS

Dudas y opiniones