



**Afi** Escuela  
de Finanzas

# Redes Sociales

## Máster en Data Science y Big Data

**Alejandro Llorente Pinto**

2022-04-26

# Índice

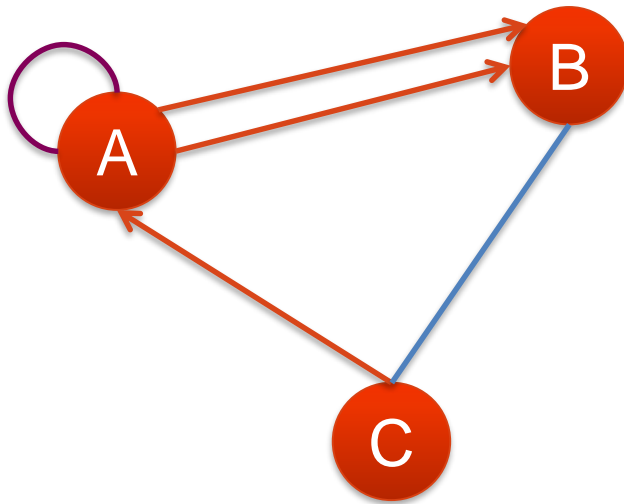
---

1. Introducción a las redes sociales
2. Conceptos básicos de SNA
3. Fundamentos prácticos básicos

# 1 | Introducción a las redes sociales

# ¿Qué es un grafo?

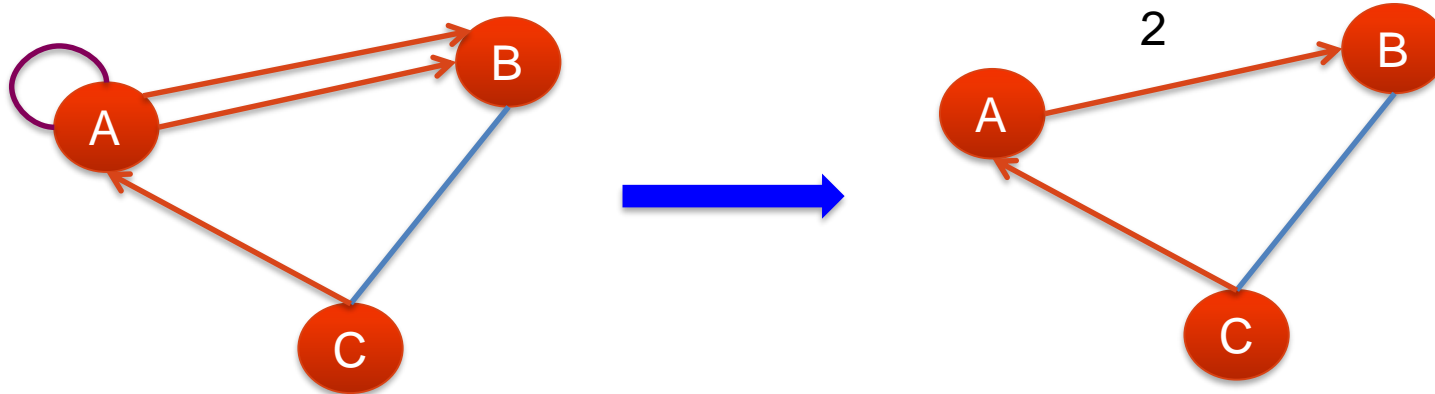
- Usaremos las palabras red y grafo indistintamente.
- Un grafo está definido por dos conjuntos de elementos.
  - **Nodos** (vertices): elementos que representan entidades con significado.
  - **Enlaces** (links): conexiones entre nodos.



- Los links pueden ser **dirigidos** o no. Dependiendo del contexto, un link sin dirección puede significar una relación bidireccional entre dos nodos.
- Un **bucle** es un enlace que empieza y acaba en el mismo nodo.
- Un enlace es **múltiple** si ha sido observado en más de una ocasión.

# ¿Qué es un grafo?

- **Simplificación** es el proceso de eliminar bucles y combinar enlaces múltiples en un peso.

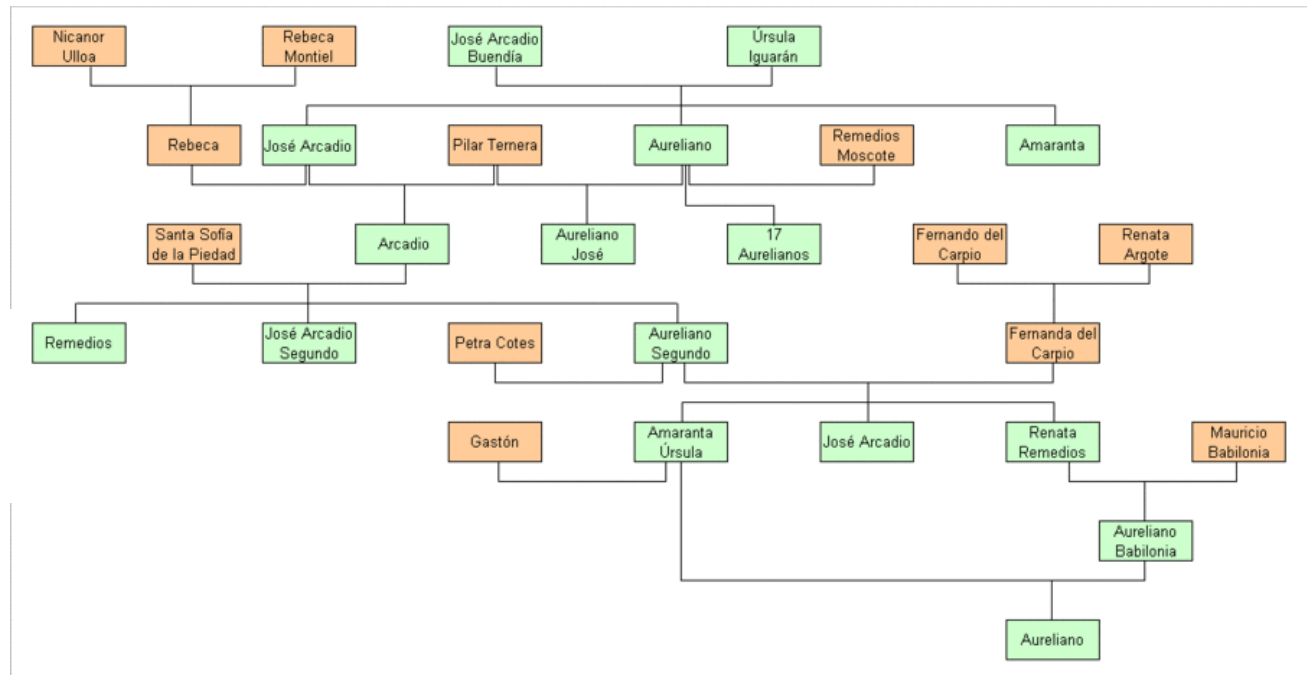


- Los pesos son una representación de la intensidad de una relación pero también, dependiendo del contexto, pueden representar costes.

# Tipos de redes sociales

- Las redes sociales son un tipo particular de red donde los nodos representan **personas**.
- Algunas de estas redes sociales son estáticas (o poco cambiantes) y representan una organización **formal**.

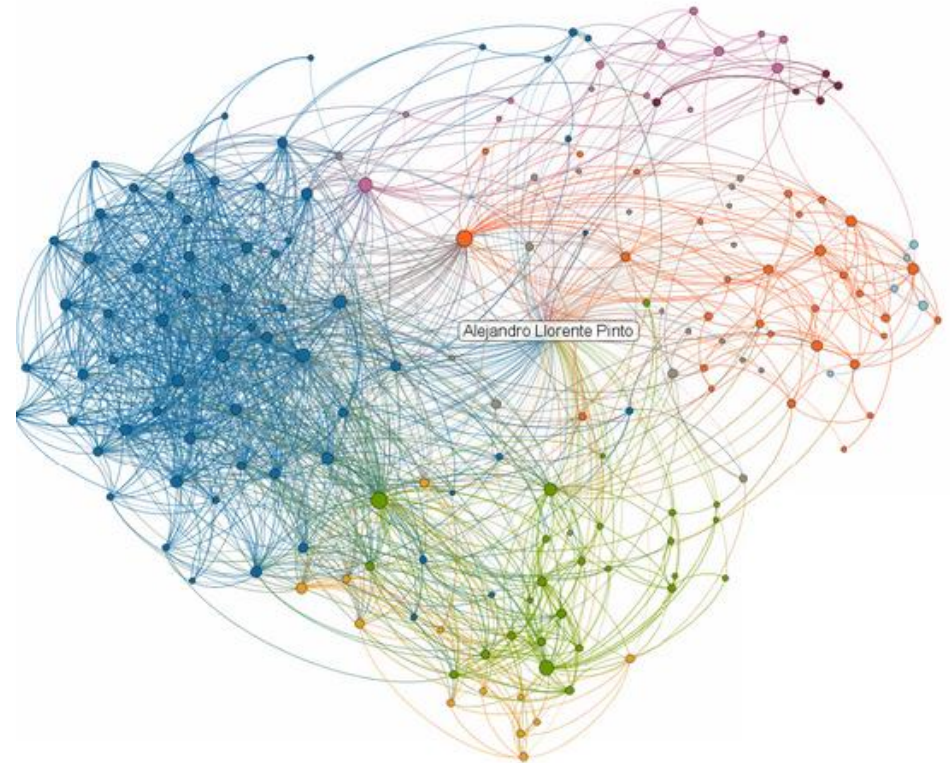
**La familia de Aureliano Buendía, protagonista de *Cien Años de Soledad*.**



# Tipos de redes sociales

- Pueden también representar relaciones sociales, como la **amistad**, compañeros de trabajo, etc.

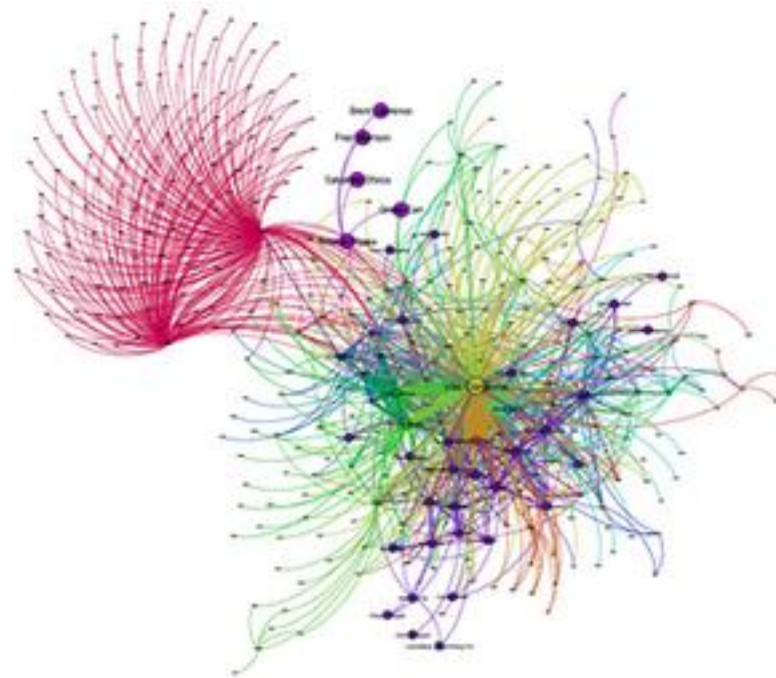
## Ejemplo de una red ego-network en LinkedIn



# Tipos de redes sociales

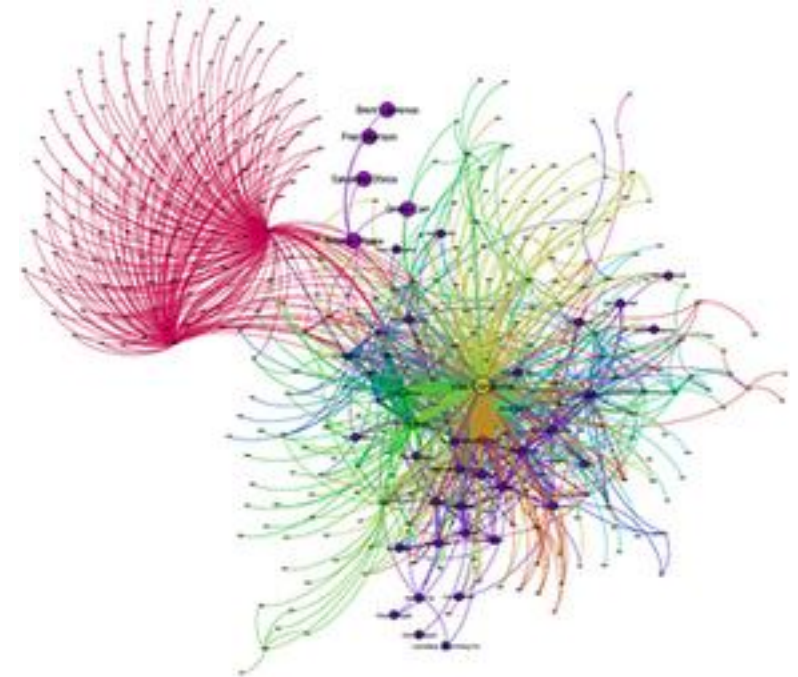
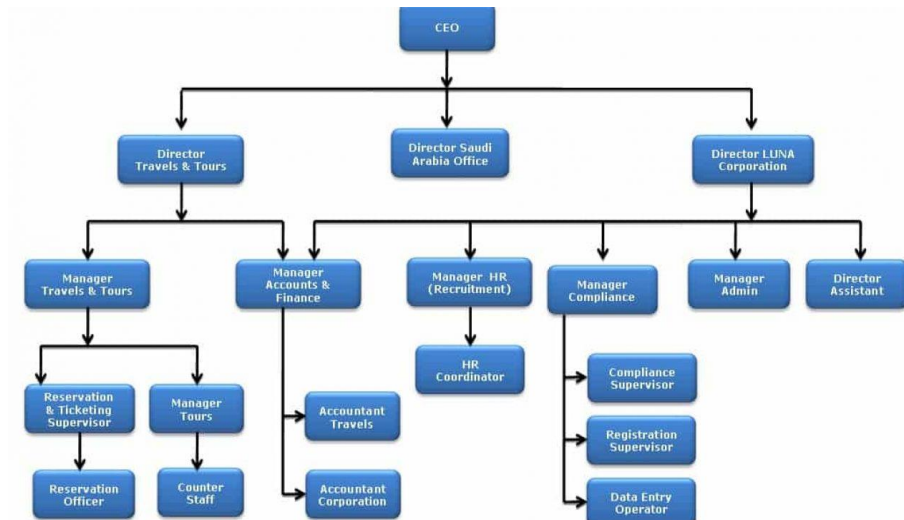
- Otras en cambio pueden ser construidas con las **interacciones** observadas entre individuos.

## Ejemplo de una red de emails en una empresa



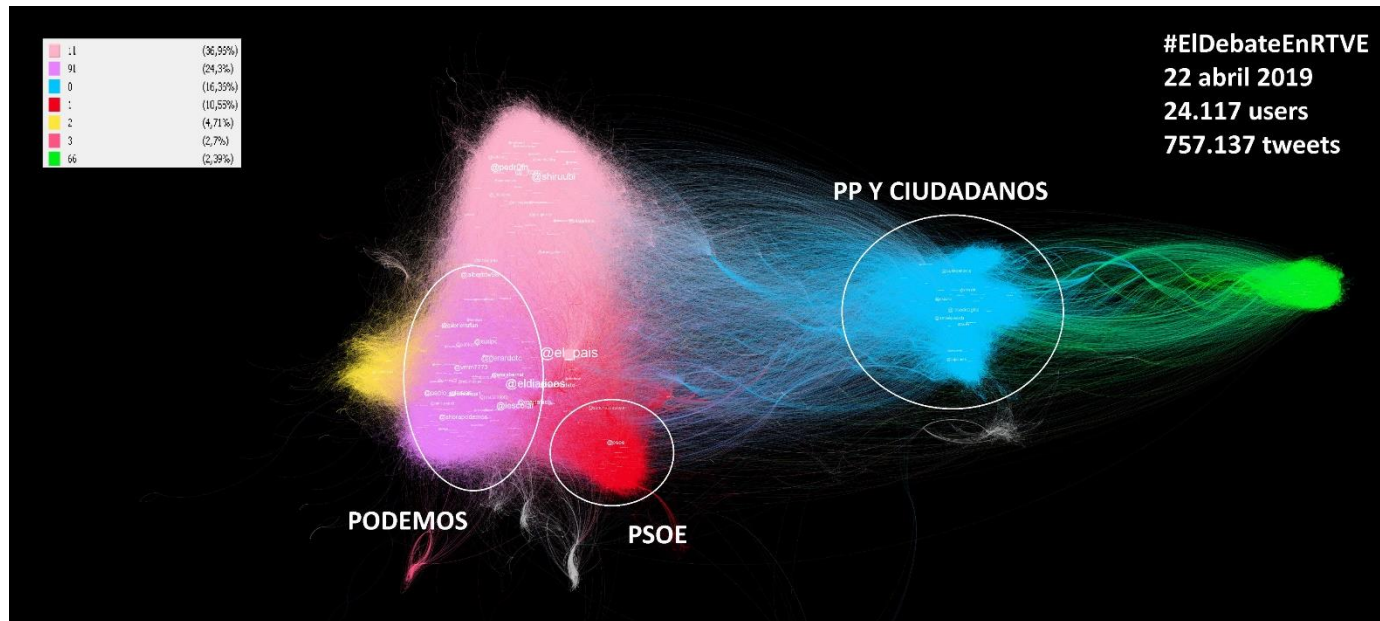


# Idea de aplicación: detección de líderes



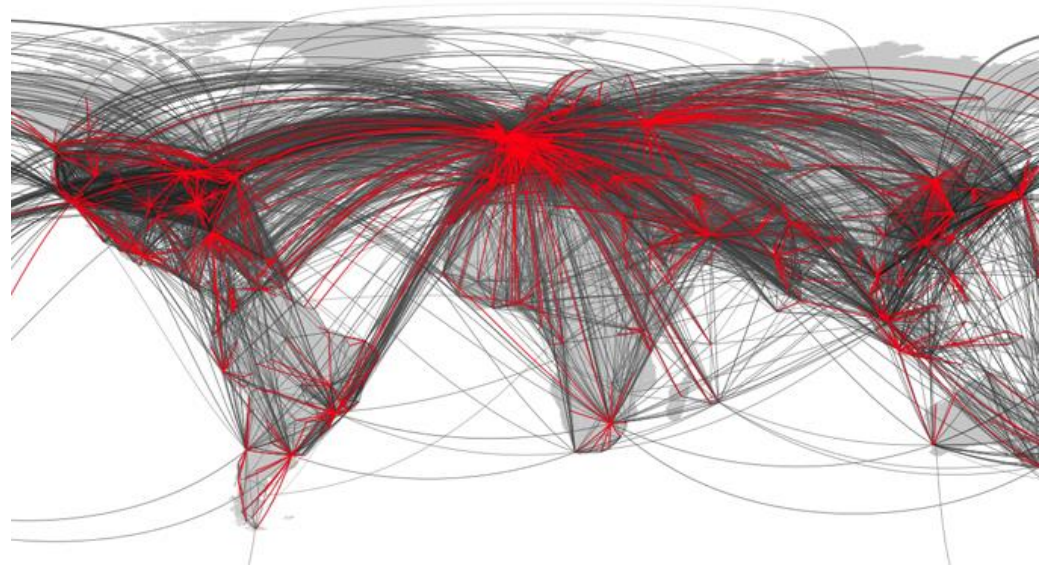
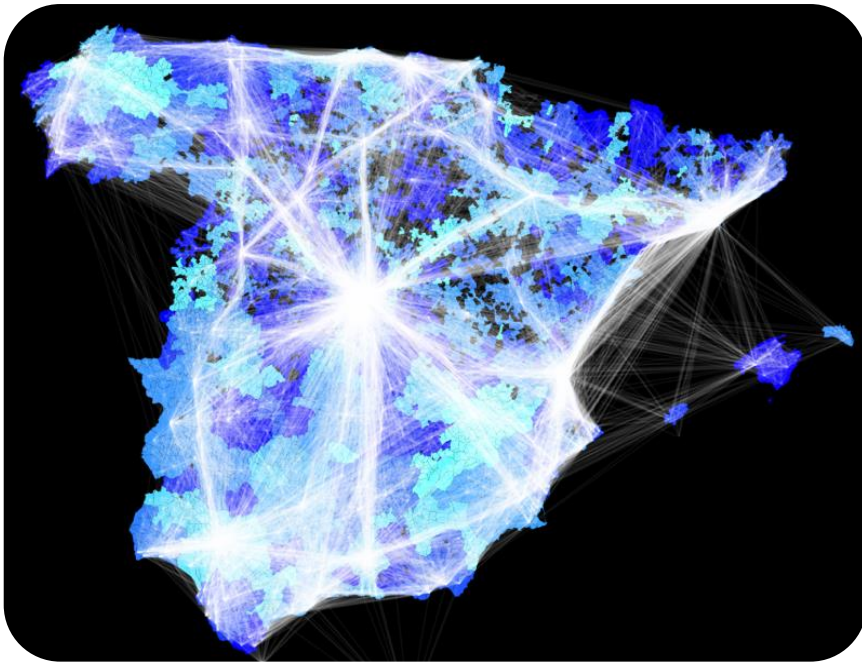
# Tipos de redes sociales

- En ocasiones, analizar la **estructura** de la red para averiguar qué grupos de nodos están más conectados entre sí, puede dar mucha información acerca de aquello en lo que se investiga.

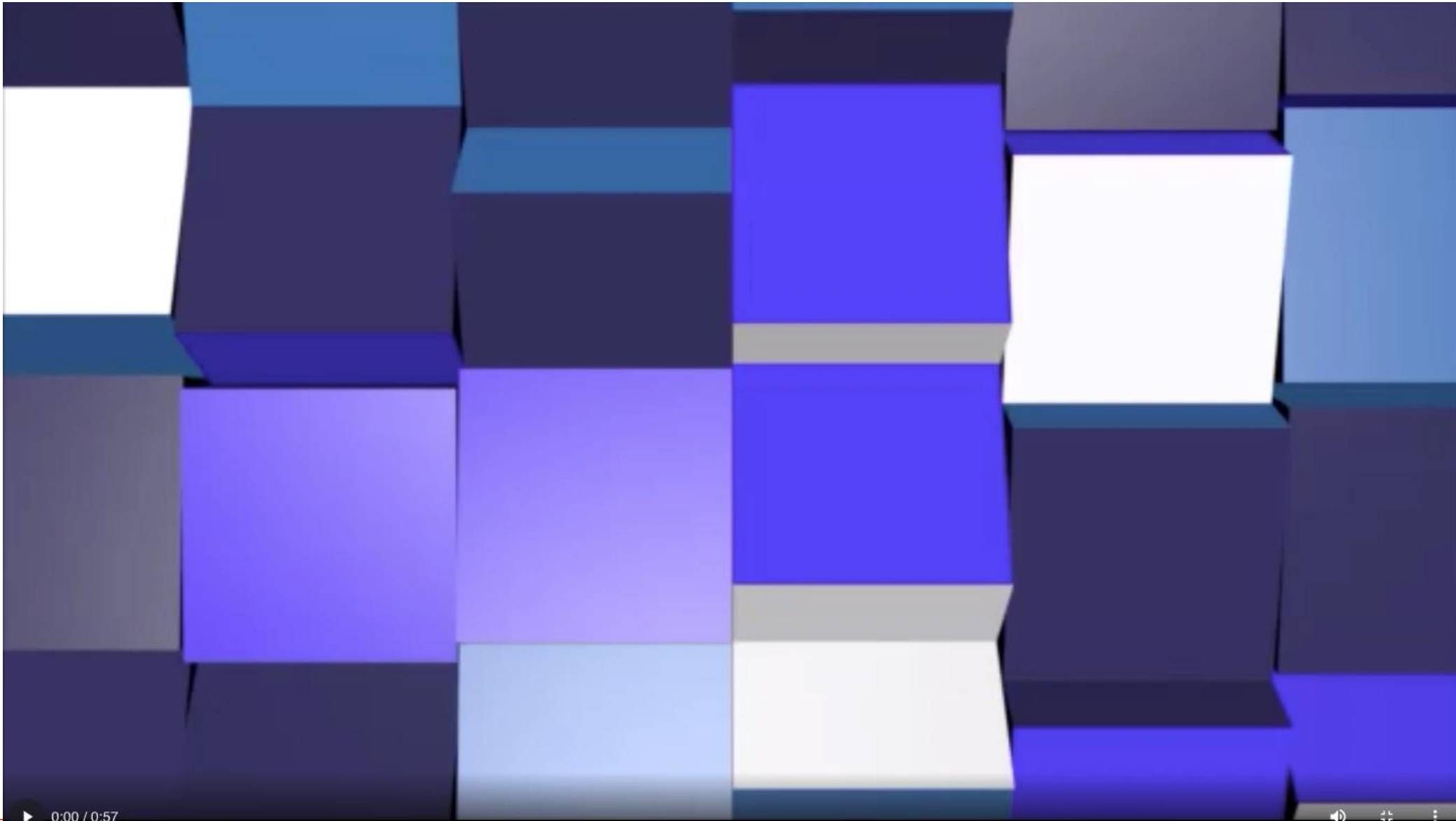


# Tipos de redes sociales

- Además, podemos añadir otros componentes, como la geografía, ya que, en muchos de los casos, los individuos están ubicados en un determinado lugar.



## Introducción a las redes sociales

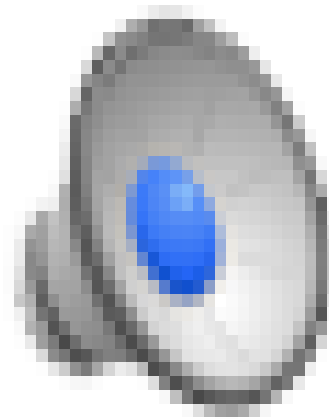


0:00 / 0:57



# Tipos de redes sociales

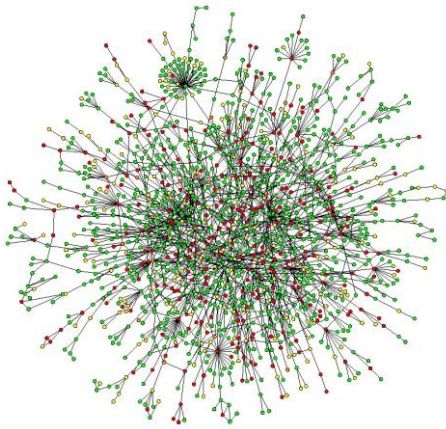
- Finalmente, las redes sociales son **dinámicas**: no interactuamos continuamente con nuestros contactos, aparecen y desaparecen relaciones, nodos, etc.



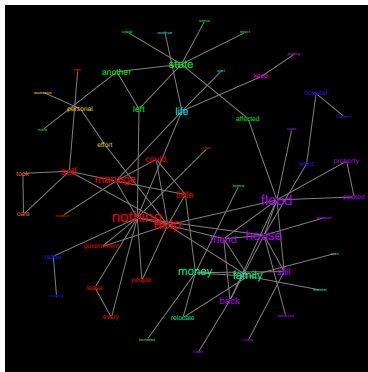
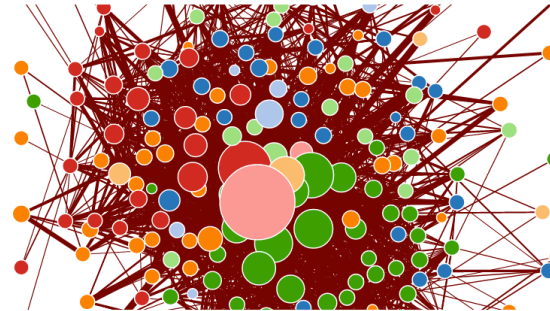


# Otros tipos de redes

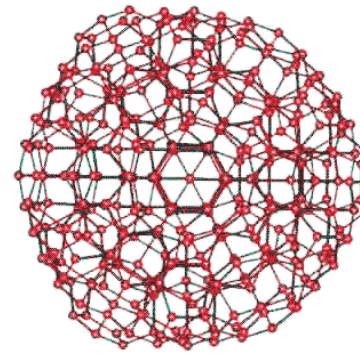
**Protein-Protein Network**



**Country-Country Network**

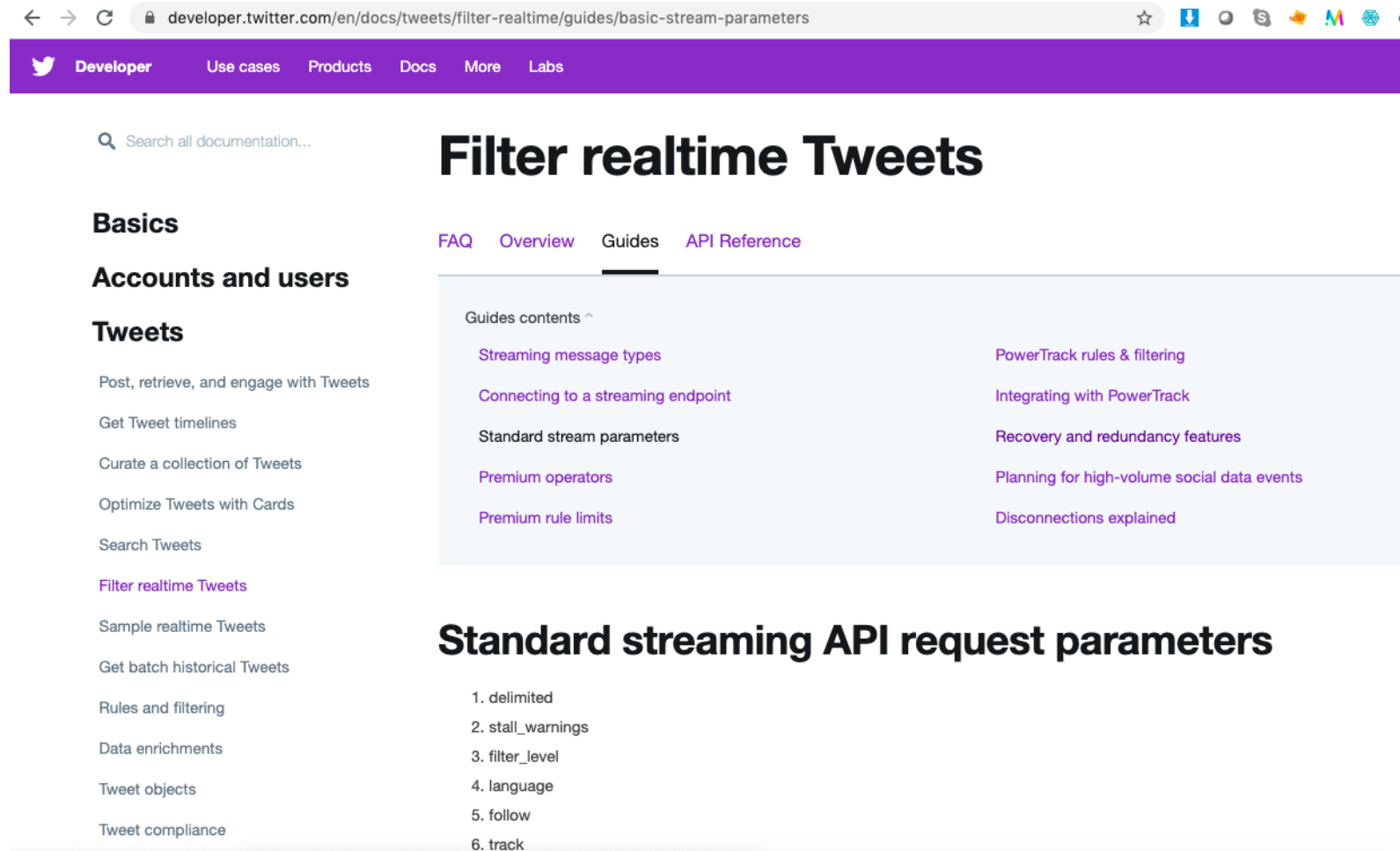


**Word network**



**Molecule network**

# Redes sociales online



The screenshot shows the Twitter Developer documentation page for 'Filter realtime Tweets'. The browser address bar shows the URL: `developer.twitter.com/en/docs/tweets/filter-realtime/guides/basic-stream-parameters`. The page has a purple header with navigation links: Developer, Use cases, Products, Docs, More, and Labs. A search bar is located below the header. The main content area is titled 'Filter realtime Tweets' and includes tabs for FAQ, Overview, Guides (selected), and API Reference. The 'Guides' tab is active, showing a list of guide contents: Streaming message types, Connecting to a streaming endpoint, Standard stream parameters, Premium operators, Premium rule limits, PowerTrack rules & filtering, Integrating with PowerTrack, Recovery and redundancy features, Planning for high-volume social data events, and Disconnections explained. A sidebar on the left lists various documentation topics under 'Basics', 'Accounts and users', and 'Tweets'. The 'Tweets' section is expanded, showing 'Filter realtime Tweets' as the selected item.

← → ↻ 🔒 developer.twitter.com/en/docs/tweets/filter-realtime/guides/basic-stream-parameters ☆ 📄 📄 📄 📄 📄 📄 📄

🐦 Developer Use cases Products Docs More Labs

🔍 Search all documentation...

## Filter realtime Tweets

FAQ Overview **Guides** API Reference

Guides contents ^

- Streaming message types
- Connecting to a streaming endpoint
- Standard stream parameters
- Premium operators
- Premium rule limits
- PowerTrack rules & filtering
- Integrating with PowerTrack
- Recovery and redundancy features
- Planning for high-volume social data events
- Disconnections explained

## Standard streaming API request parameters

1. delimited
2. stall\_warnings
3. filter\_level
4. language
5. follow
6. track

**Basics**

**Accounts and users**

**Tweets**

- Post, retrieve, and engage with Tweets
- Get Tweet timelines
- Curate a collection of Tweets
- Optimize Tweets with Cards
- Search Tweets
- Filter realtime Tweets**
- Sample realtime Tweets
- Get batch historical Tweets
- Rules and filtering
- Data enrichments
- Tweet objects
- Tweet compliance

# Redes sociales online


rtweet **0.7.0**
Reference
Articles
Changelog

## rtweet

build passing
CRAN 0.7.0
codecov 64%
DOI 10.5281/zenodo.2528481
Peer Reviewed

repo status Active
downloads 11K/month
downloads 167K
lifecycle maturing
JOSS 10.21105/joss.01829

R client for accessing Twitter's REST and stream APIs. Check out the [rtweet package documentation website](#).



## Package Functionality

There are several R packages for interacting with Twitter's APIs. See how {rtweet} compares to these others in the chart below.

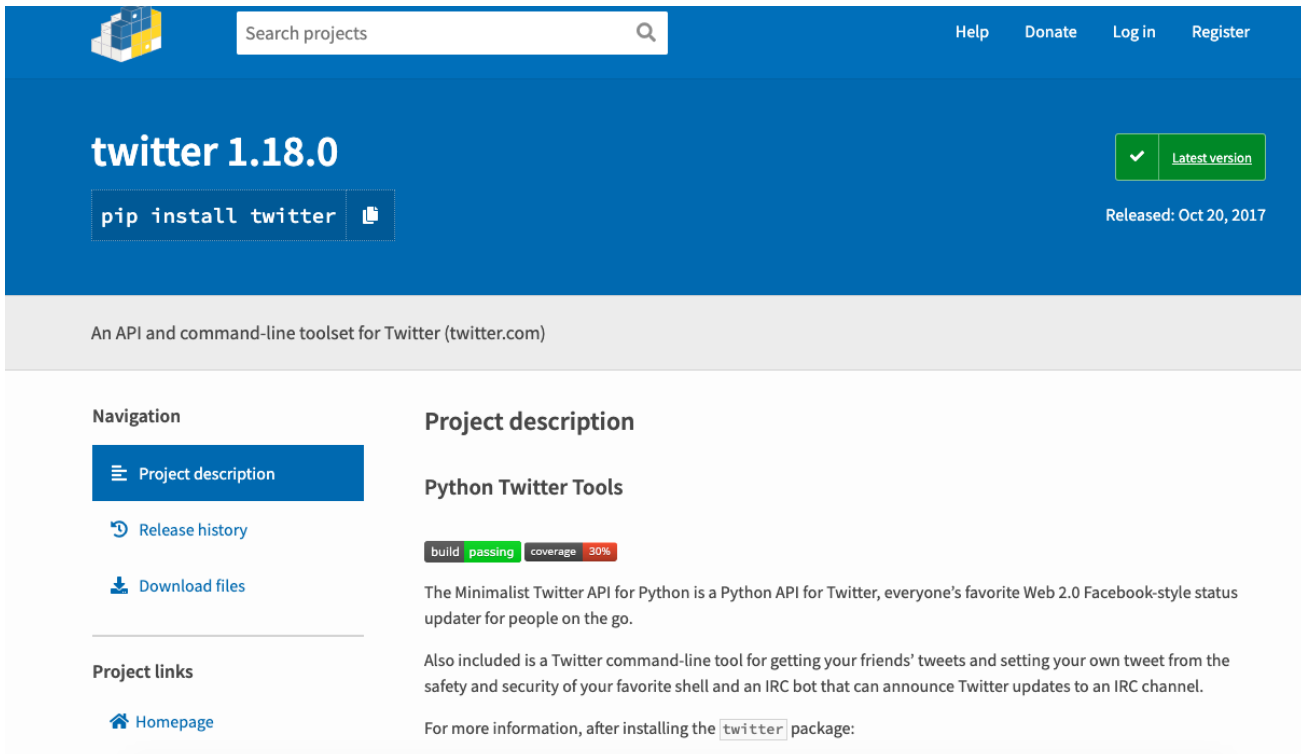
Task	rtweet	twitterR	streamR	RTwitterAPI
Available on CRAN	✓	✓	✓	✗
Updated since 2016	✓	✗	✓	✗
Non-'developer' access	✓	✗	✗	✗
Extended tweets (280 chars)	✓	✗	✓	✗
Parses JSON data	✓	✓	✓	✗
Converts to data frames	✓	✓	✓	✗

<https://rtweet.info/>



# Redes sociales online

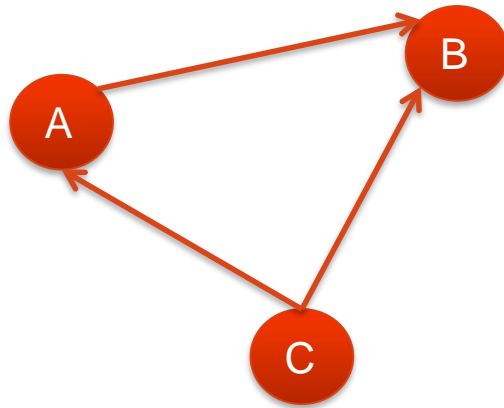
<https://pypi.org/project/twitter/>



The screenshot shows the PyPI project page for the 'twitter' package. The header is blue with a search bar and navigation links (Help, Donate, Log in, Register). The main section features the package name 'twitter 1.18.0' in large white text, a green 'Latest version' badge, and a 'pip install twitter' button. Below this, a description states 'An API and command-line toolset for Twitter (twitter.com)'. The left sidebar contains a 'Navigation' menu with 'Project description' (selected), 'Release history', and 'Download files'. The 'Project description' section is titled 'Python Twitter Tools' and includes a status bar showing 'build passing' and 'coverage 30%'. The description text reads: 'The Minimalist Twitter API for Python is a Python API for Twitter, everyone's favorite Web 2.0 Facebook-style status updater for people on the go. Also included is a Twitter command-line tool for getting your friends' tweets and setting your own tweet from the safety and security of your favorite shell and an IRC bot that can announce Twitter updates to an IRC channel. For more information, after installing the `twitter` package:'.

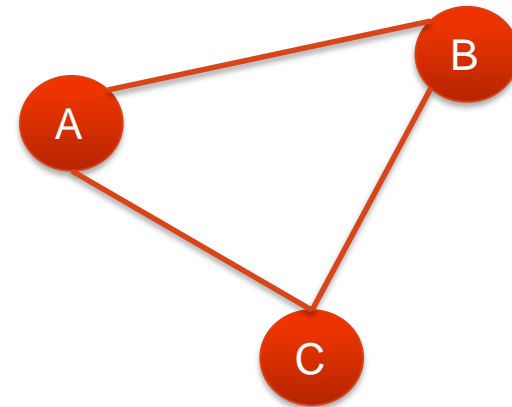
# 2 | Conceptos básicos de SNA

# Grafos dirigidos vs no dirigidos



- Red de emails
- Redes de movilidad
- Red de followers en Twitter

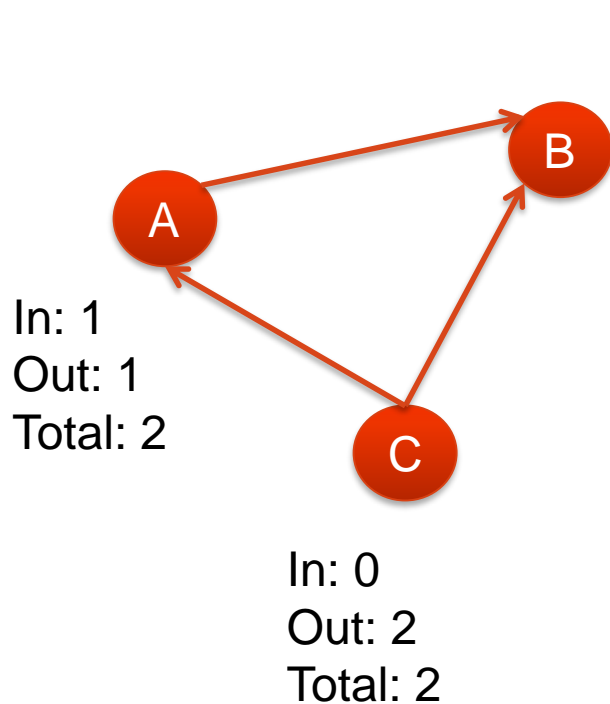
**¿Cómo describirías una red de calles en forma de grafo dirigido?**



- Conexiones entre servidores de Internet.
- Enlaces entre moléculas
- Red de contactos en LinkedIn

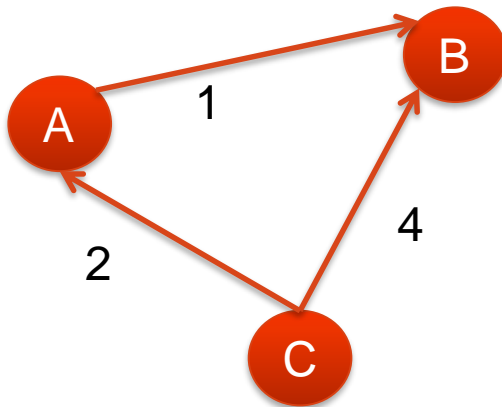
**¿Algún ejemplo real con enlaces direccionales y bidireccionales?**

# Propiedades de nodos: Grado



- El grado es el número de conexiones en el que un nodo está involucrado.
- Podemos calcular tres tipos: el de entrada (in), el de salida (out) y el grado total (in+out).
- En grafos no dirigidos, solamente tiene sentido hablar del grado total

## Propiedades de nodos: Grado (en redes con pesos)



- De la misma manera, podemos contar simplemente el número de enlaces pero, habitualmente, también se suele extender la definición del grado en función del peso del enlace.
- Redes de interacciones
  - Número de mensajes.
  - Duración de la llamada.
- Capacidad de comunicación entre dos ubicaciones.
- Número de peajes entre dos puntos geográficos.

## Propiedades básicas del grafo

- A menudo, queremos describir el grafo como un objeto en sí mismo, con propiedades que surgen de la propia estructura de red. Esta caracterización la hacemos a través de métricas.
- Las métricas más básicas de un nodo son, obviamente, el número de nodos y enlaces que posee.
- Derivado de ello, definimos la **densidad** del grafo como:

$$\text{densidad} = \frac{\text{num. enlaces observados}}{\text{num. enlaces posibles}}$$

## Propiedades básicas del grafo

- A menudo, queremos describir el grafo como un objeto en sí mismo, con propiedades que surgen de la propia estructura de red. Esta caracterización la hacemos a través de métricas.
- Las métricas más básicas de un nodo son, obviamente, el número de nodos y enlaces que posee.
- Derivado de ello, definimos la **densidad** del grafo como:

$$\text{densidad} = \frac{\text{num. enlaces observados}}{\text{num. enlaces posibles}}$$

Caso dirigido

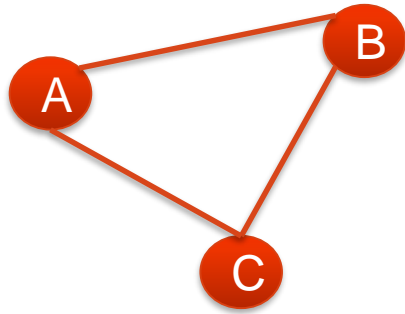
$$N(N - 1)$$

Caso no dirigido

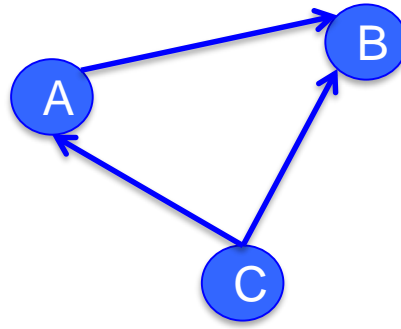
$$\frac{N(N - 1)}{2}$$

## Conectividad

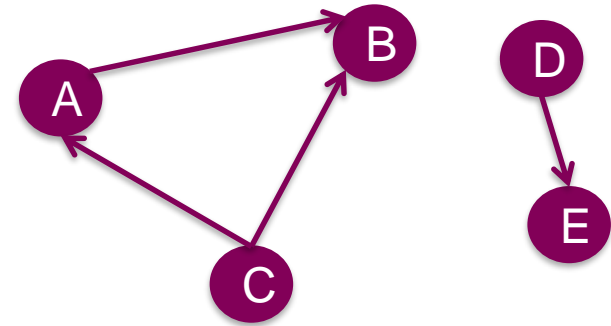
- Un grado está conectado (o es conexo) si, desde cualquier nodo, yo puedo alcanzar cualquier otro yendo a través de los enlaces.



Este grafo es **conexo**.



Este grafo no es **fuertemente conexo** pero sí es **débilmente conexo**.

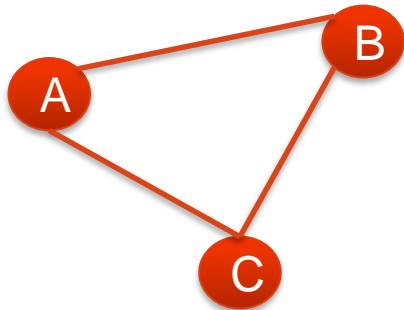


Este grafo no es conexo... pero tiene dos **componentes débilmente conexas**.

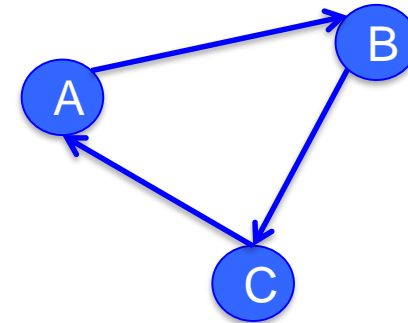


## Ciclos

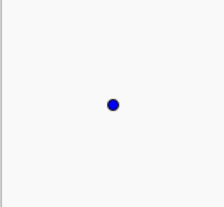
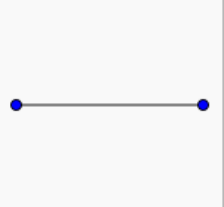
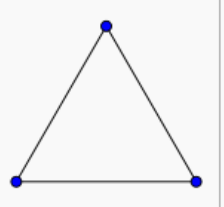
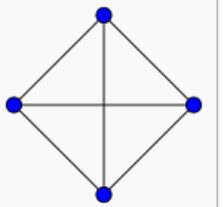
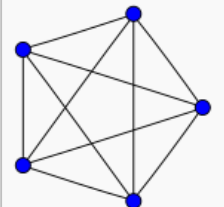
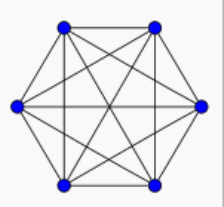
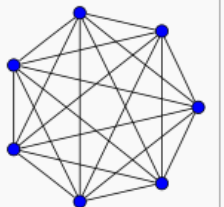
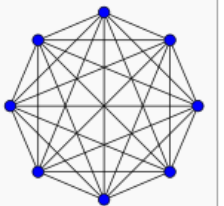
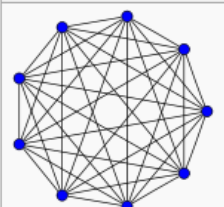
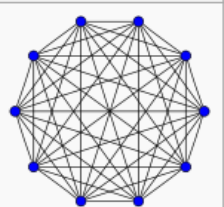
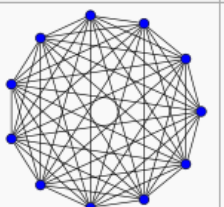
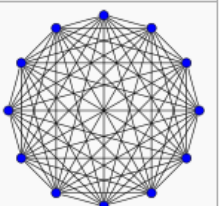
- Un ciclo es un camino que parte de un nodo y acaba en el mismo nodo visitando cada enlace solamente una vez.



Puede haber ciclos tanto en grafos no dirigidos como dirigidos.

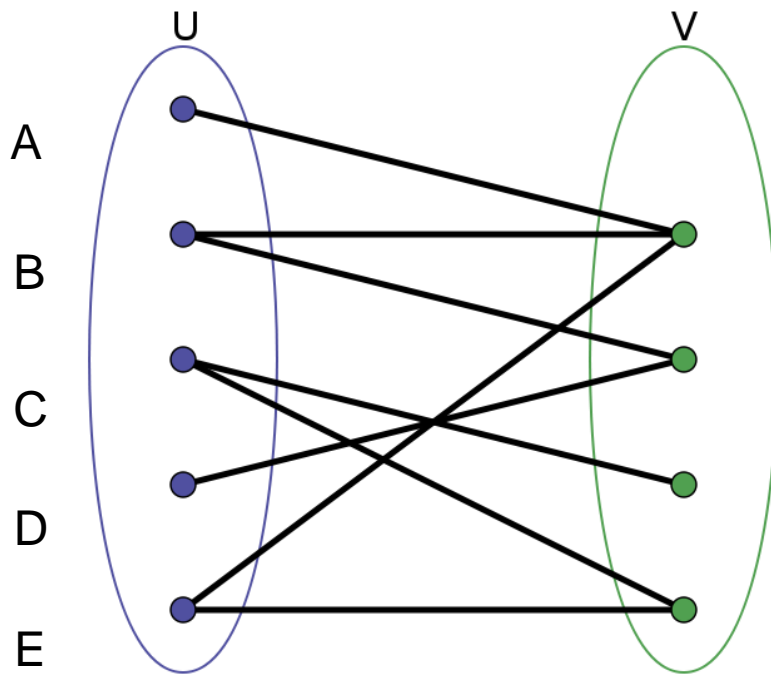


## Tipos de grafos: completos

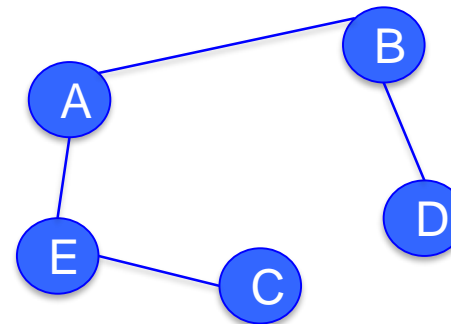
$K_1: 0$	$K_2: 1$	$K_3: 3$	$K_4: 6$
			
$K_5: 10$	$K_6: 15$	$K_7: 21$	$K_8: 28$
			
$K_9: 36$	$K_{10}: 45$	$K_{11}: 55$	$K_{12}: 66$
			

- Los grafos completos son grafos no dirigidos donde aparecen todos los enlaces posibles.
- ¿Cuánto es el grado para un nodo en un grafo completo?
- ¿Cuántos links hay en un grafo completo con N nodos?

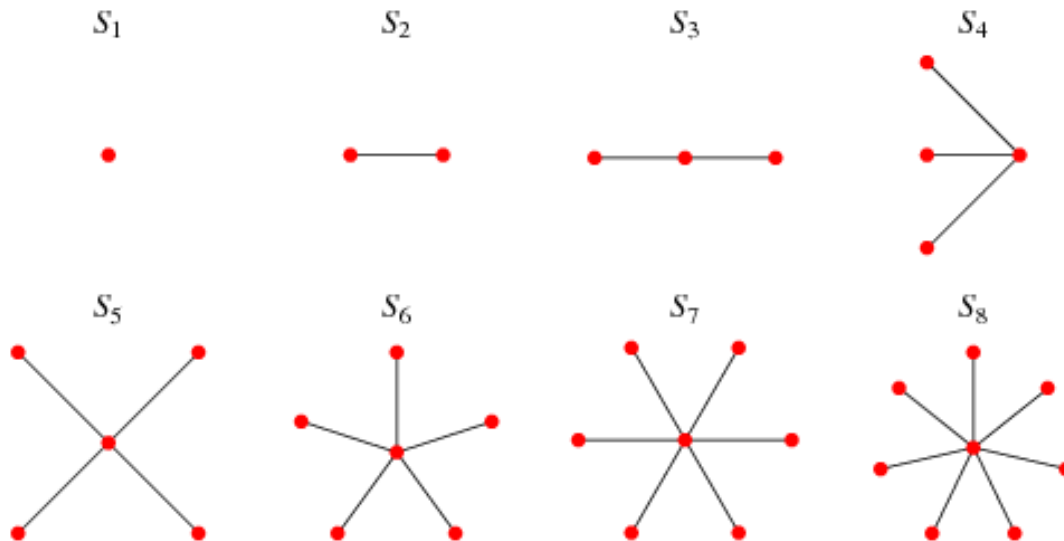
## Tipos de grafos: bipartitos



- Son grafos donde existen links entre dos grupos de nodos completamente separados.
- Estos grafos pueden ser proyectados para descubrir relaciones entre los nodos de cada grupo.

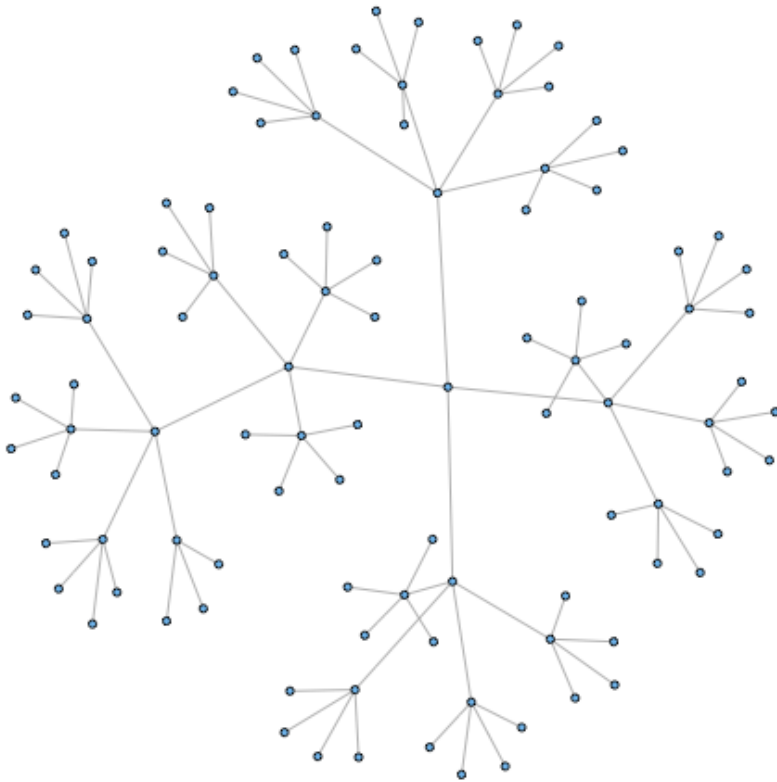


## Tipos de grafos: estrella



- Los grafos en estrella son un caso particular de grafo bipartido donde uno de los grupos está compuesto solamente de un nodo.
- Además, los grafos en estrella también son árboles...

# Tipos de grafos: árboles



- Un árbol es un grafo conexo y sin ciclos.
- Tienen características peculiares ya que podemos hablar de nodos raíz, nodos hijos, subárboles, profundidad, etc.

## Tipos de grafos: árboles

- Hay procesos que últimamente nos suenan mucho los cuales generan este tipo de grafos.

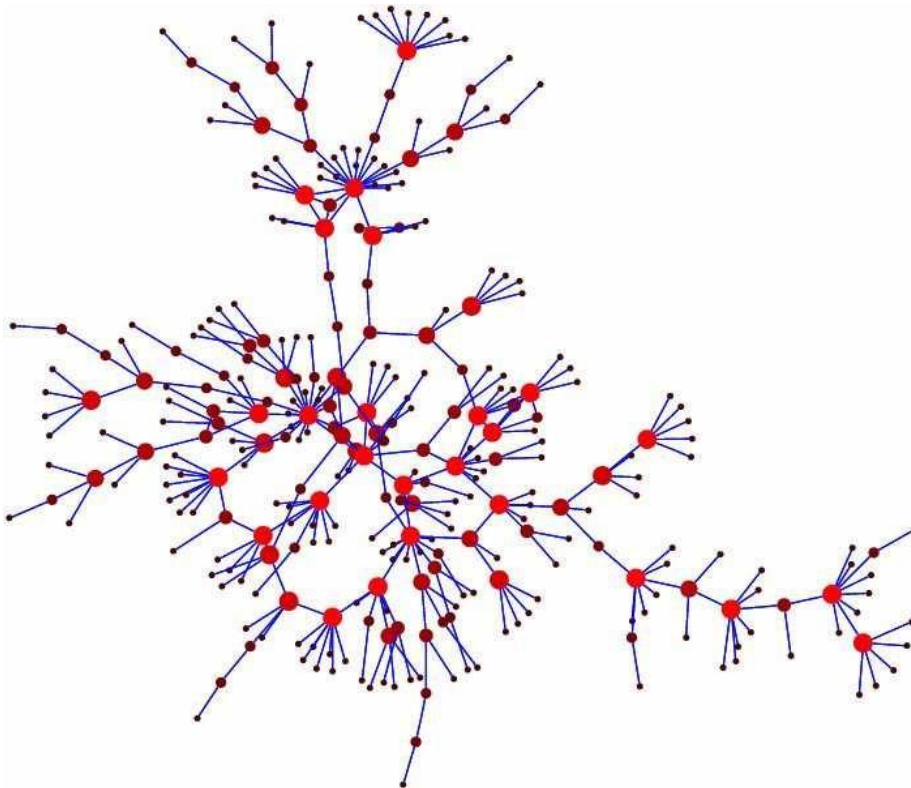


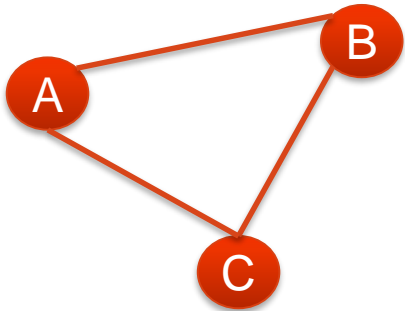
Gráfico extraído de la tesis de Gourab Ghoshal “Structural and dynamical properties of Complex Networks”.

An epidemic network of HIV transmission, generated from the data of Potterat et al (2015).

Vertices are individuals, and an edge between any two represents transmission of HIV (either through sexual contact or intravenous drug usage). Picture courtesy of Mark Newman.

## Representaciones formales: listas de enlaces

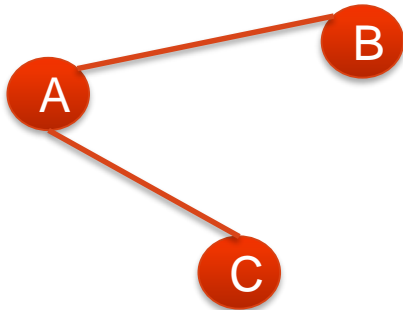
- Es un tipo de descripción de los ejes de un grafo.



- En este ejemplo,  $\{ [A,B], [B,C], [C,A] \}$  es la lista de enlaces del grafo.
- Cuando el grafo es dirigido, los elementos del enlace se suelen escribir entre paréntesis.

## Representaciones formales: matriz de adyacencia

- Es otra representación formal de los enlaces de un grafo.



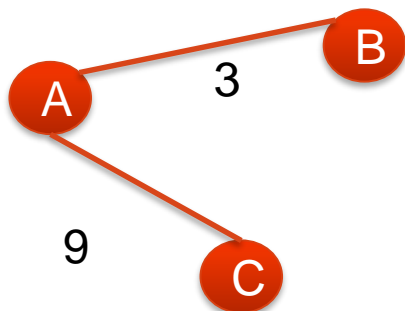
- En el caso de los grafos no dirigidos, la matriz de adyacencia es simétrica.

	A	B	C
A	0	1	1
B	1	0	0
C	1	0	0



## Representaciones formales: matriz de adyacencia

- Es otra representación formal de los enlaces de un grafo.

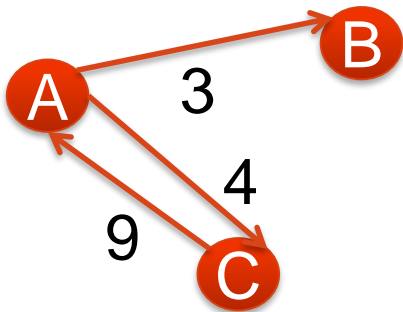


- Cuando el grafo es con pesos, se modifica el uno anterior por el peso en cuestión en la matriz.

	A	B	C
A	0	3	9
B	3	0	0
C	9	0	0

## Representaciones formales: matriz de adyacencia

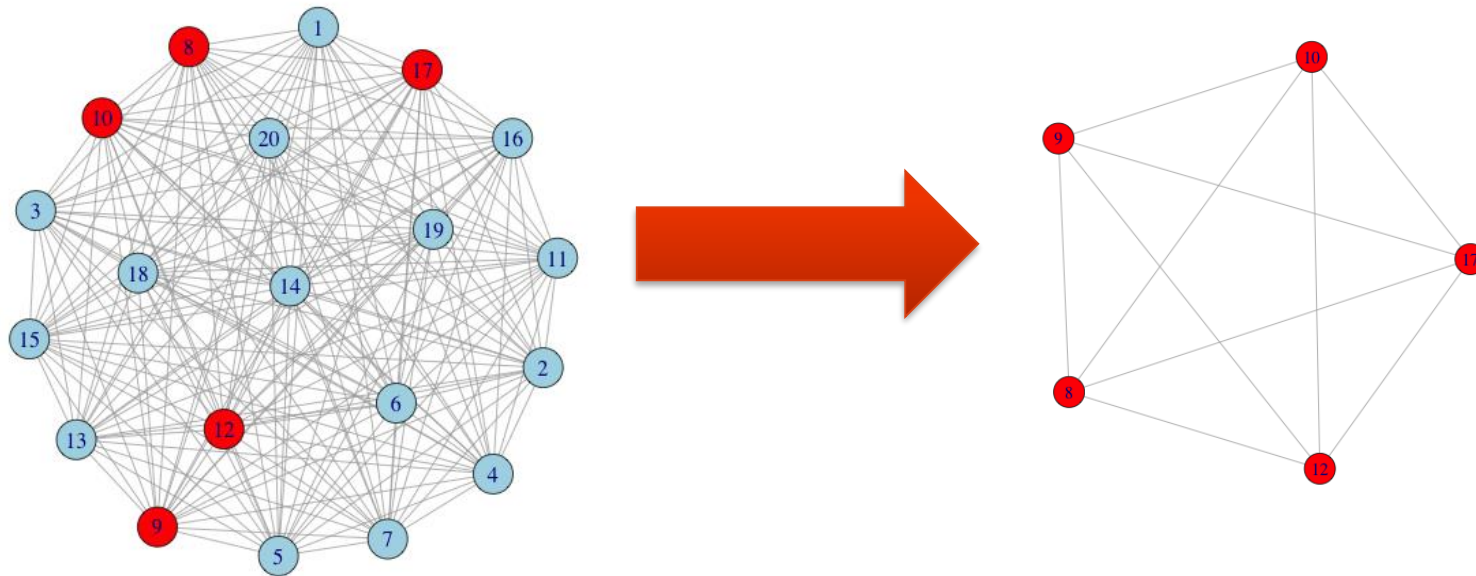
- Es otra representación formal de los enlaces de un grafo.



- En el caso de que el grafo sea dirigido y con pesos, la matriz de adyacencia deja de ser simétrica.

	A	B	C
A	0	3	4
B	0	0	0
C	9	0	0

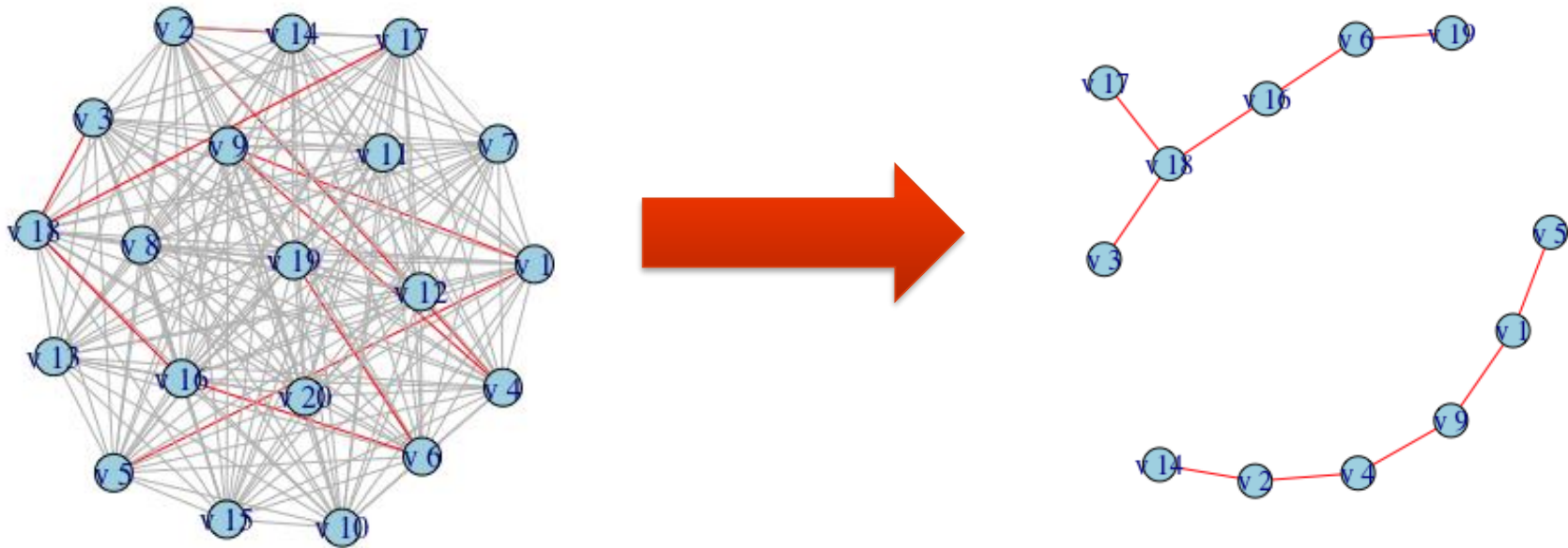
# Funciones sobre grafos: subgrafos



¿Cómo construirías un subgrafo solamente con los nodos rojos?

El grafo resultante está compuesto por los nodos señalados y los enlaces entre ellos

## Funciones sobre grafos: subgrafos

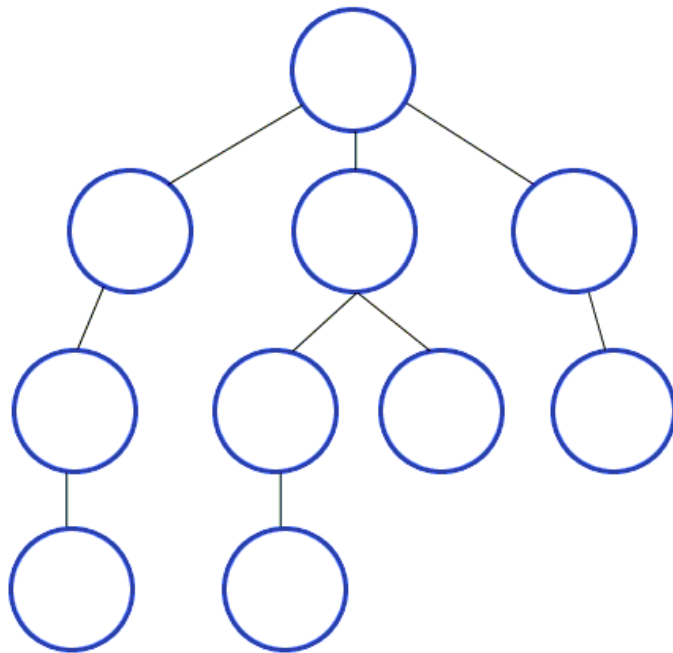


¿Cómo construirías un grafo solamente con los enlaces en rojo?

El grafo resultante está compuesto por los enlaces señalados y los nodos involucrados en ellos

## Funciones sobre grafos: búsqueda en profundidad

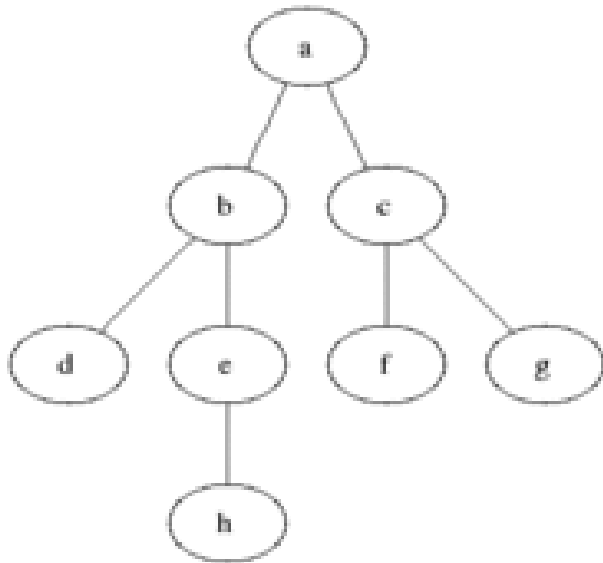
Una **búsqueda** es un proceso para recorrer los nodos a partir de uno dado y visitándolos solamente una vez.



- El proceso empieza en un nodo dado (raíz).
- Cada nodo solamente se visita una vez.
- Una vez visitado un nodo, se sigue el proceso por sus vecinos.
- Si se llega a un nodo sin vecinos, se vuelve al padre y se sigue por otro vecino.

## Funciones sobre grafos: búsqueda en anchura

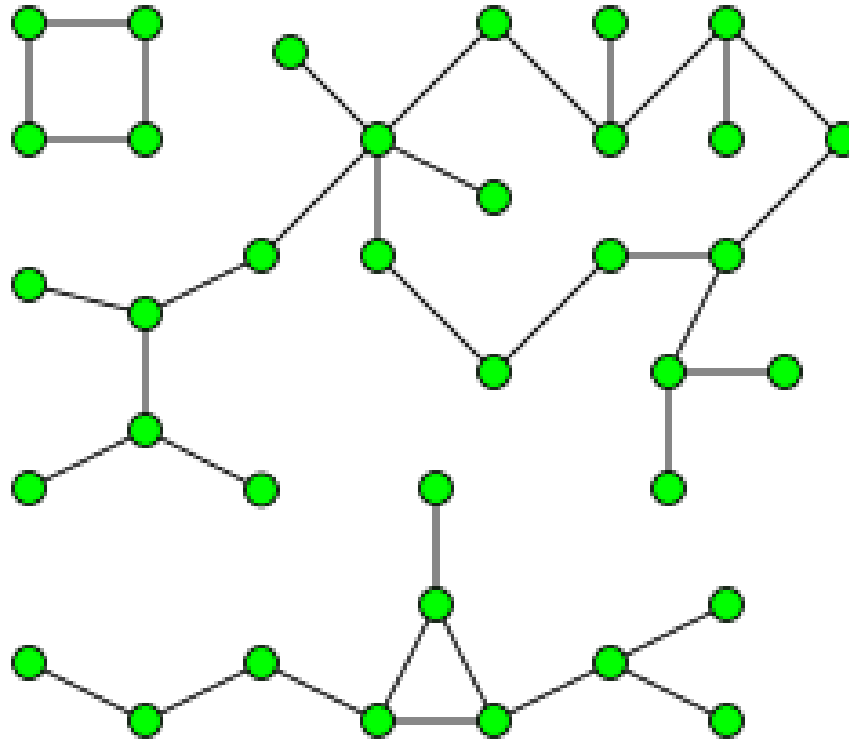
Una **búsqueda** es un proceso para recorrer los nodos a partir de uno dado y visitándolos solamente una vez.



- El proceso comienza en un nodo dado.
- Cada nodo solamente es visitado una vez.
- Tras un nodo visitado, se visita a aquellos en el mismo nivel jerárquico.
- Si agotamos los de un nivel, procedemos a los encolados del siguiente nivel.

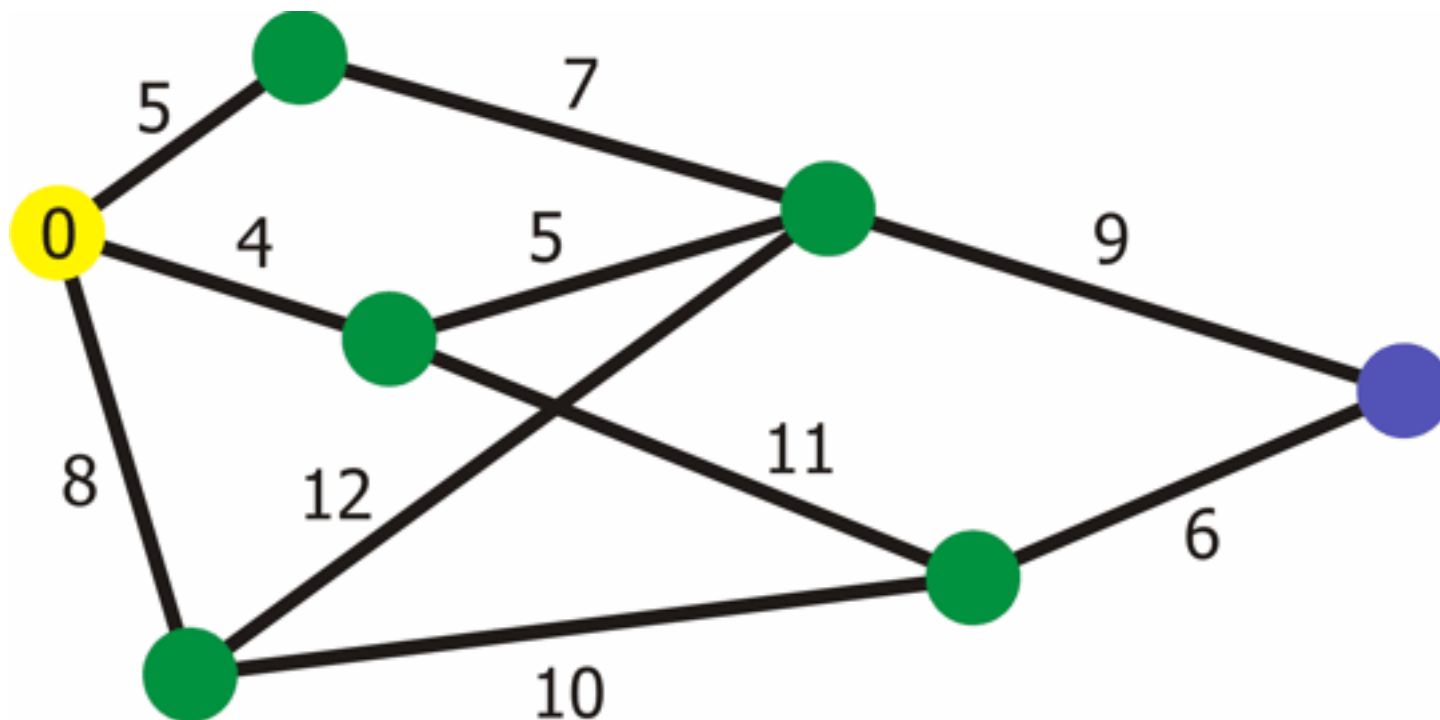
## Funciones sobre grafos: detectar componentes conexas

¿Se os ocurre alguna manera de detectar componentes conexas del siguiente grafo?



## Funciones sobre grafos: distancias entre nodos

Utilizando el algoritmo de Dijkstra podemos encontrar la distancia de un nodo al resto (en grafos con pesos positivos).





## Funciones sobre grafos: distancias entre nodos

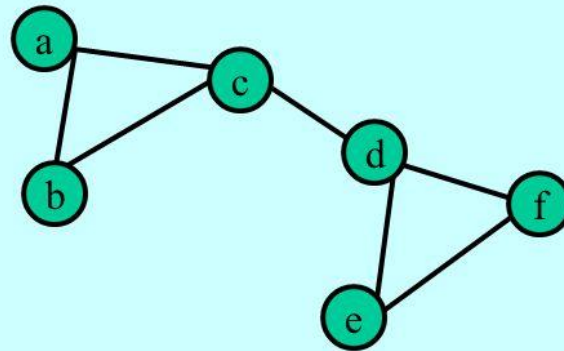
La distancia toma distintos significados dependiendo del contexto:

- **En grafos sin pesos:** es el número de saltos mínimo necesario para llegar de un nodo a otro. Al conjunto de saltos lo llamamos camino.
- **En un grafo con pesos:** es el camino que minimiza la suma de los pesos. Esto implica que, para la mayoría de algoritmos, el peso tiene una noción de distancia, es decir, de lejanía social entre nodos.

# Funciones sobre grafos: diámetro y camino medio

## Average Path length

Average path length  $L$  of a network is defined as the **mean distance** between all pairs of nodes.



a to b 1

a to c 1

a to d 2

a to e 3

a to f 3

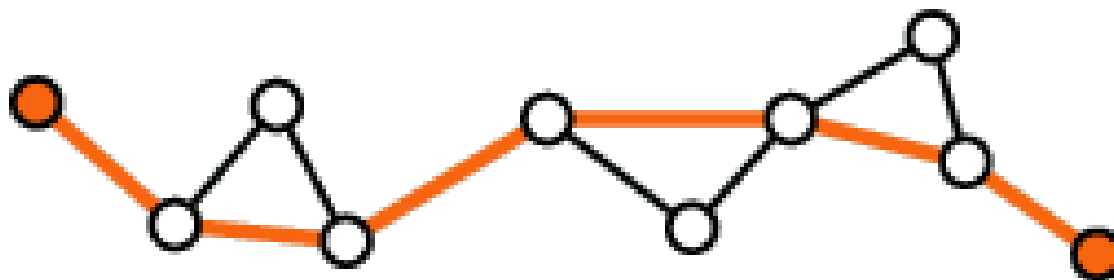
-----  
-----

$$L=27/15=1.8$$

Average path length of most  
real complex network is small

15 pairs 27(total length)

## Funciones sobre grafos: diámetro y camino medio

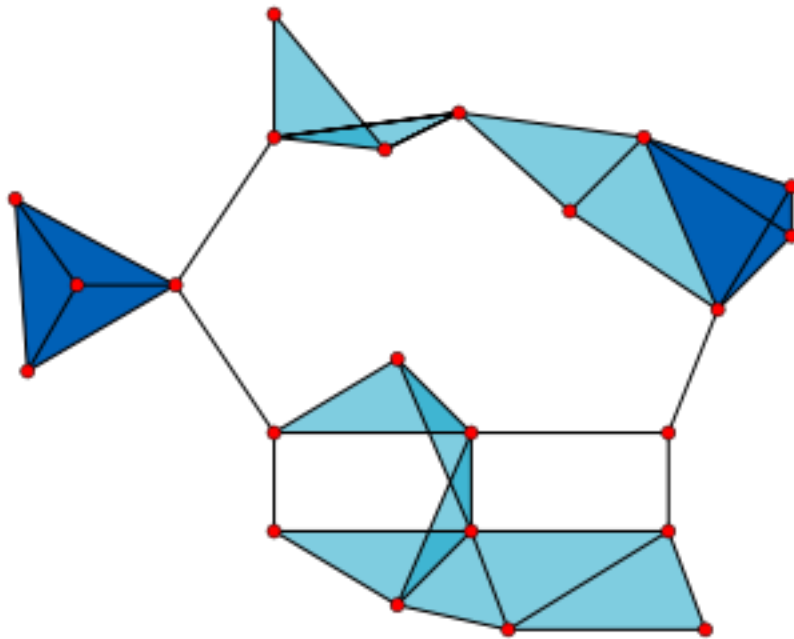


El diámetro es el mayor de todos los caminos mínimos.

En este caso el diámetro del grafo es 6.

# Funciones sobre grafos: detectar cliques

Un clique es un subgrafo completo dentro de un grafo.

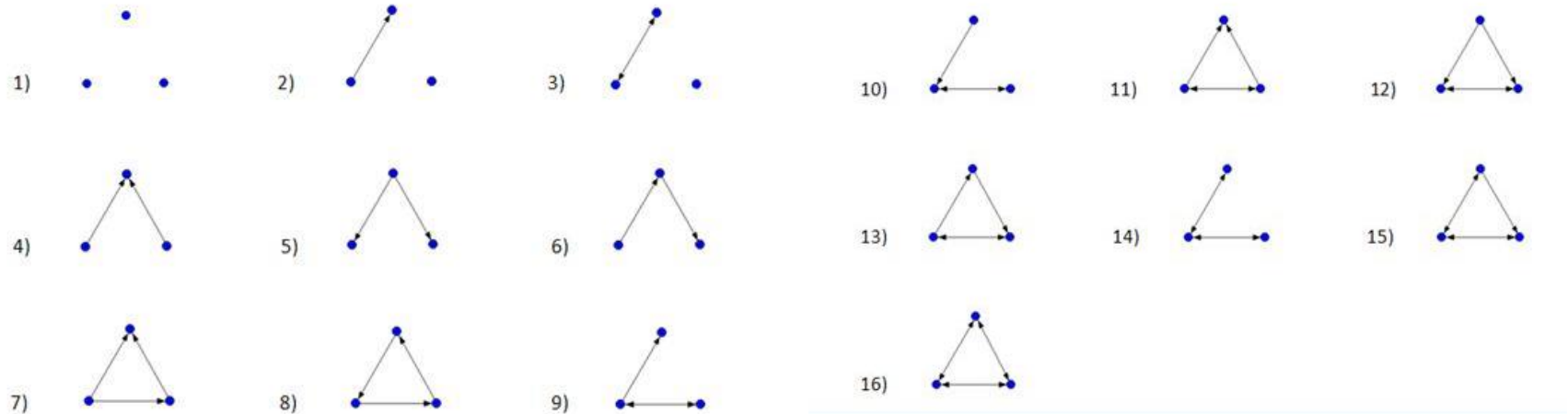


En este grafo podemos ver:

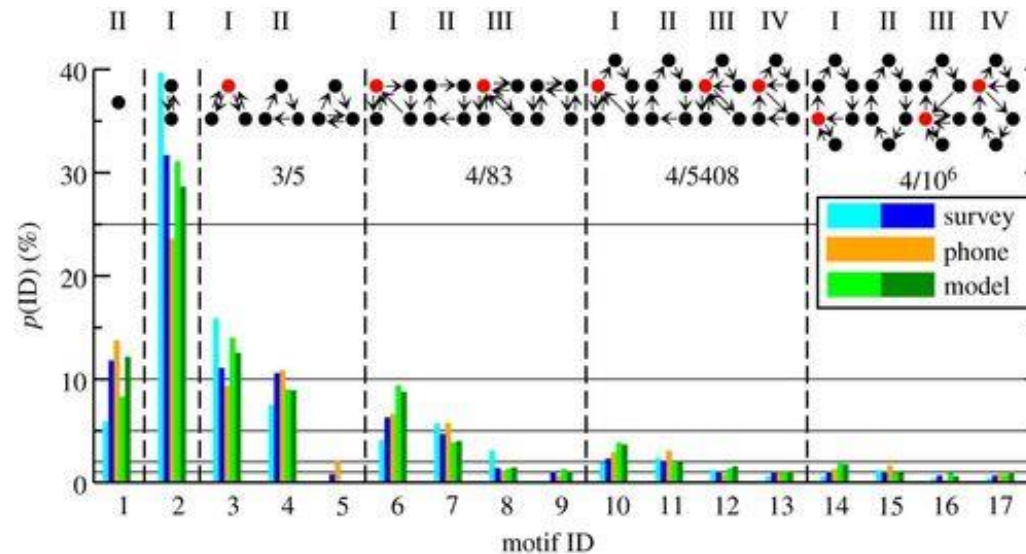
- 23 × cliques de 1 vértice (los vértices),
- 42 × cliques de 2 vértices (los links),
- 19 × cliques de 3 vértices (triángulos azules claros y oscuros), and
- 2 × cliques de 4 vértices (triángulos de azul oscuro).

# Funciones sobre grafos: detectar motivos

Un motivo es una forma particular de disponer los enlaces dentro de un grafo.



# Detectando motifs para estudiar movilidad



Research articles

## Unravelling daily human mobility motifs

Christian M. Schneider, Vitaly Belik, Thomas Couronné, Zbigniew Smoreda and Marta C. González

Published: 06 July 2013 | <https://doi.org/10.1098/rsif.2013.0246>

# 3 | Fundamentos Prácticos Básicos

# Requisitos software

Para los ejercicios prácticos de este curso necesitaremos:

- Rstudio
- Rmarkdown
- Paquete igraph



# Creación de grafos aleatorios y operaciones básicas

1. Creación de grafos aleatorios con la función `erdos.renyi.game` con 200 nodos y 400 enlaces.
2. Obtener un resumen de los datos del grafo y la semántica de lo que devuelve la función.
3. ¿Cómo se nombran los enlaces y los vértices?
4. Crear un subgrafo escogiendo aleatoriamente con 10 nodos y otro con 20 enlaces.
5. Comparar los resúmenes de ambos.

## Funciones avanzadas

1. Calcular las distancias de un nodo aleatorio y analizar el resultado.
2. Calcular el mínimo árbol abarcador del grafo.
3. Visualizar el grafo original con los nodos con un tamaño proporcional a la distancia del nodo “1”.
4. Visualizar el grafo original y colorear de un color diferente los enlaces que están en el árbol abarcador.



**Afi** Escuela  
de Finanzas

---

© 2021 Afi Escuela de Finanzas. Todos los derechos reservados.