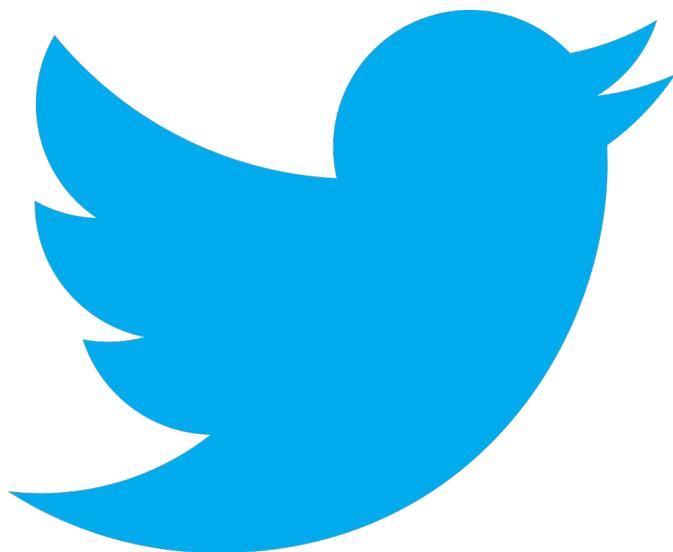


# **Práctica 2**

## **Caso Práctico de Análisis y Evaluación de Redes en Twitter**

---



**Gestión de Información en la Web**

**Máster en Ingeniería Informática**

**Curso 2016/17**

**Universidad de Granada**

Nombre: Aythami Estévez Olivas

DNI: 70918176E

Email: [aythae@correo.ugr.es](mailto:aythae@correo.ugr.es)

## Índice

• 1. Selección de un medio social, definición de una pregunta de investigación y obtención de un conjunto de datos asociado	3
◦ 1.1. Obtención y descripción de datos	3
◦ 1.2. Pregunta de investigación	3
• 2. Construcción de la red social on-line a analizar y visualizar	4
◦ 2.1. Importación de datos en Gephi	4
◦ 2.2. Reducción de la dimensionalidad de la red	4
■ 2.2.1. Estrategias de poda Pathfinder	5
■ 2.2.2. Estrategias de filtrado	6
• 3. Cálculo de los valores de las medidas de análisis	7
◦ 3.1. Medidas globales de la red	7
◦ 3.2. Conectividad de la red	7
• 4. Determinación de las propiedades de la red	7
◦ 4.1. Distribución de grados	7
◦ 4.2. Distribución de distancias	8
◦ 4.3. Distribución de coeficiente de clustering medio	9
◦ 4.4 Conclusiones	10
• 5. Calculo de los valores de las medidas de análisis de redes sociales	10
• 6. Descubrimiento de comunidades en la red	11
• 7. Visualización de la red social	13
◦ 7.1 Visualización 1	14
◦ 7.2 Visualización 2	15
◦ 7.3 Visualización 3	16
• 8. Discusión de los resultados obtenidos	17
• Anexo: twitter_search	17
◦ Tutorial	18
◦ Uso del programa	18
• Bibliografía	20

# 1. Selección de un medio social, definición de una pregunta de investigación y obtención de un conjunto de datos asociado

Estos últimos días se ha creado mucha polémica en España por la sentencia de la Audiencia Nacional el caso de los chistes de Carrero Blanco [1] hechos por la tuitera Cassandra Vera, los cuales le han llevado a una condena de 1 año de cárcel y siete de inhabilitación. Me resulta un tema interesante como se pueden interpretar los límites de la libertad de expresión, la delgada linea entre el humor negro y el delito. Por ello he decidido estudiar este tema usando Twitter como medio social.

## 1.1. Obtención y descripción de datos

La obtención de datos de Twitter se ha realizado de un modo distinto del comentado en clase debido a problemas de compatibilidad de NodeXL con mi versión de Office, por ello decidí descargar tuits usando la API de Twitter mediante el módulo de python **Tweepy**[2]. He creado un script en python llamado `twitter_search.py` que utiliza Tweepy para descargar tuits a partir de una cadena de caracteres usada para la búsqueda, además permite especificar el número de tuits a recuperar, las fechas entre las que se han escrito los tuits devueltos, o el idioma de los tuits. Ver el apartado Anexo: `twitter_search` para más información.

## 1.2. Pregunta de investigación

La pregunta concreta de investigación es **¿Cuales los usuarios más relevantes en la discusión de Twitter sobre las palabras "Carrero Blanco"?**

Es necesario acotar un poco más la pregunta de investigación, ya que no he determinado una ventana temporal para estudiar el tema, por ello empecé descargando todos los tuits que se habían escrito en distintos días para ver como evoluciona su repercusión a lo largo del tiempo y elegir el día más interesante para estudiarlo. Con estos datos he creado la siguiente gráfica que representa el número de tuits respecto a los días:

Fecha	Número de Tweets
26-03-2017	1.735
27-03-2017	797
28-03-2017	643
29-03-2017	115.254
30-03-2017	61.724
31-03-2017	20.846
01-04-2017	9.125



Como se puede apreciar se produce un importante incremento del número de Tweets que contienen "Carrero Blanco" el día **29 de Marzo** (llegando a 115.254 tuits), coincidiendo con la sentencia que condenaba a Cassandra Vera por sus chistes. Por ello este será el día seleccionado para realizar el estudio.

## 2. Construcción de la red social on-line a analizar y visualizar

Como se ha dicho en el apartado Anexo: twitter\_search para responder esta pregunta construye una red social en la que los **nodos corresponden a los usuarios** que han escrito tuits sobre el tema o han sido mencionados/retuiteados y los **arcos a las relaciones entre estos usuarios** a partir de las menciones y retuits que han realizado. El **peso** de estos determina el número de relaciones que unen a un usuario con otro y he optado por la construcción de un **grafo dirigido** de la forma que si el usuarioA menciona al usuarioB existirá un arco dirigido desde el usuarioA al usuarioB.

### 2.1. Importación de datos en Gephi

A la hora de importar los ficheros .csv generados a Gephi hay que seleccionar Archivo > Importar hoja de calculo, aquí se pide un fichero, su separador, su codificación y su es tabla de nodos o aristas.

Primero importaremos los nodos (el fichero queryX\_(dateSince\_dateUntil)\_Users.csv) para ello lo seleccionamos, dejamos como separador la coma, como codificación UTF-8 y marcamos que es una tabla de nodos. Al hacer click en siguiente se nos muestran las columnas a importar, es necesario cambiar los tipos de datos de las columnas Following, Followers, Tweets\_count, Favourites a Long para poder utilizarlos como números más adelante.

Una vez hecho esto pulsamos terminar y pasamos a importar las aristas, volvemos a seleccionar Importar hoja de calculo, buscamos el fichero queryX\_(dateSince\_dateUntil)\_Edges.csv, marcamos que es una tabla de aristas dejando el resto tal cual y pulsamos siguiente. Es importante marcar el checkbox de **crear nodos inexistentes** ya que es posible que existan menciones a usuarios no existentes en el fichero de usuarios. Para crear los usuarios que no estén en el fichero de nodos Gephi crea nuevas filas en el fichero con todas las columnas vacías a excepción del ID que saca del fichero de relaciones, por ello para que al menos aparezca el label es necesario acceder al Laboratorio de datos de Gephi y seleccionar la opción de copiar columna ID sobre la columna Label. Con esto podemos analizar la red social mejor al tener no solo los usuarios que han escrito tuits, si no aquellos que han sido mencionados o retuiteados en estos tuits, como inconveniente es que de esos usuarios generados por Gephi solo poseemos su ID y Label, no sus datos de twitter como el número de seguidores, por ello habrá que tener precaución al usar estas medidas en el análisis ya que existen valores perdidos.

### 2.2. Reducción de la dimensionalidad de la red

Usando los tweets del día 29 de marzo se obtiene un total de 49.640 usuarios (nodos) y 107.365 relaciones (aristas), cabe mencionar que al importar a Gephi las aristas se marca la opción de Crear nodos inexistentes y el número de nodos crece hasta los 50.258.

Como resulta obvio parándose a mirar los números de nodos y aristas del grafo no resulta manejable trabajar con un grafo de estas dimensiones, por lo que hay que reducir su dimensión. Teniendo en cuenta que nos encontramos ante un grafo ponderado dirigido se puede reducir la dimensión aplicando un filtrado de nodos por grado o de aristas por peso o bien una poda pathfinder.

### 2.2.1. Estrategias de poda Pathfinder

La poda pathfinder sería lo mejor ya que conservan los enlaces más importantes de la red proporcionando una representación única de la estructura subyacente de la red. Por ello busqué en los apuntes el mejor algoritmo pathfinder aplicable a mi red (teniendo en cuenta que existen múltiples versiones y optimizaciones), según estos el mejor algoritmo pathfinder es el **MST-Pathfinder** basado en un enfoque greedy logra ser mucho más rápido que sus hermanos a costa de restringir algunos parámetros, el problema es que solo es aplicable en grafos no dirigidos por lo que no es utilizable en mi caso.

Por tanto el siguiente candidato es **Fast Pathfinder**, partiendo de las referencias de los apuntes empecé mirando la una web que analiza las diversas variantes y aporta implementaciones en C de todas ellas [4]. El problema es que la implementación en C del algoritmo Fast pathfinder esta hecha para grafos no dirigidos pero dicha web menciona otras implementaciones conocidas, así es como dí con **Network Workbench** (NWB) [5], una herramienta software para el análisis de redes a gran escala que contiene multiples implementaciones de diversos algoritmos útiles para este análisis, entre ellas las diversas podas Pathfinder. A pesar de lo prometedor que suena esto he tenido múltiples problemas para usar esta herramienta, empezando por que parece estar abandonada (sin actualizaciones desde 2011), la documentación on-line no funciona pero logré obtener un manual de usuario en PDF [6]. Según esté, NWB admite diversos formatos entre los que se encuentra el formato .net de Pajek, por lo que exporté desde Gephi la red del grafo completo a ese formato obviando las posiciones de los nodos ya que solo me interesan sus relaciones para aplicar el podado. Tras cargar la red probé a realizar un Fast Pathfinder desde la opción *Analysis > Directed and Pondered > Fast pathfinder network* obteniendo errores por ser la red demasiado grande y tener autoenlaces o bucles (enlaces de un nodo hacia si mismo, cosa muy habitual en una conversación en twitter ya que al responder a alguien se van acumulando las menciones a todos los participantes en la conversación). Por ello realice un pequeño podado en Gephi aplicando un filtro de *Componente gigante* con un subfiltro *K-core* que permite eliminar nodos con grado menor que  $k$  ( $k = 2$ ) como se ha visto en la asignatura, pero además aplicando un subfiltro de *Bucle*. Exportando el resultado a .net y cargándolo en NWB esta vez no se produce error, pero tras una hora y media de ejecución no se llega a ningún resultado, debido a las dimensiones de la red a pesar de la reducción (20.165 vértices y 79.962 enlaces) y a la eficiencia del algoritmo que según [4] obtiene tiempos de más de una hora para una red de 10.000 nodos y 100.000.000 de enlaces. Hice una última prueba con NWB y la red mucho más podada, con un *K-core* de  $k = 12$  lo que da una red de 922 nodos y 13.700 enlaces. Por mera curiosidad de ver como quedaba mi red podada con pathfinder llegado a este punto. Esta vez el algoritmo si que acaba en unos segundos pero se produce un error interno haciendo un conversión a double, por lo que me quedé con las ganas de ver el algoritmo funcionando.

## 2.2.2. Estrategias de filtrado

Resignado tras los frustrados intentos con pathfinder me decanté por aplicar filtrados. Lo primero que parece lógico, teniendo en cuenta que es una red de usuarios de Twitter, es filtrar por el número de seguidores de usuarios como una forma de conocer su relevancia general. Pero con la creación de nodos inexistentes a partir de las relaciones los usuarios creados no disponen de esos datos por lo que en lugar de eso he decidido filtrar por grado de los nodos y por peso de las aristas. Esto se realiza con los filtros de Gephi **K-core** de la categoría *Topología* y **Peso de arista** de la categoría *Aristas* respectivamente. Además de esto he probado con el filtro **Componente gigante** de la categoría *Topología* que se basa en mantener solamente nodos del componente conectado mayoritario y con el filtro **Bucle** de la categoría *Aristas* que elimina los autoenlaces, ya que no me interesa que esos autoenlaces influyan en el filtrado de grado y peso de arista por no aportar información interesante.

El objetivo es llegar a una red manejable entorno a los 1000-2000 nodos. En la siguiente tabla se recoge el resultado de diversos filtrados

Estrategia de filtrado	Número de Nodos	Número de enlaces	% Nodos respecto a la red sin filtrar	% Enlaces respecto a al red sin filtrar
Ninguna	50.258	107.365	-	-
Componente gigante	46.811	106.809	93,14	99,48
K-core ( $k = 2$ )	20.264	80.280	40,32	74,77
K-core ( $k = 3$ )	11.443	62.853	22,77	58,54
Peso de arista $> 1$	50.258	5.864	100	5,46
Componente gigante + K-core ( $k = 2$ )	20.188	80.187	44,17	74,69
Componente gigante + K-core ( $k = 3$ )	11.431	62.830	22,74	58,52
Componente gigante + K-core ( $k = 10$ )	1.421	18.899	2,83	17,6
Componente gigante + K-core ( $k = 2$ ) + Peso de arista $> 1$	1.007	2.809	2	2,61
Componente gigante + Bucle + K-core ( $k = 2$ ) + Peso de arista $> 1$	1.002	2.770	1,99	2,58

Como se puede ver el filtrado *K-core* por si solo no elimina los suficientes enlaces para que la red sea manejable, a la inversa le pasa al filtro *Peso de arista*. La mejor solución por tanto parece la combinación de estos dos filtros como se ve en la última fila de la tabla. Partiré de ese filtrado para el resto del trabajo, pero para algunas tareas podría ser necesario filtrar aún más incluso.

### 3. Cálculo de los valores de las medidas de análisis

#### 3.1. Medidas globales de la red

Medida	Valor
Número de nodos ( $N$ )	1.002
Número de aristas ( $L$ )	2.770
Densidad ( $D$ )	0,003
Grado medio ( $\langle k \rangle$ )	2,764
Grado medio con pesos	6,856
Diámetro ( $d_{max}$ )	8
Distancia media ( $\langle d \rangle$ )	2,447
Distancia media para una red aleatoria equivalente ( $\langle d_{aleatoria} \rangle$ )	6,7964
Coeficiente de clustering medio ( $\langle C \rangle$ )	0,082
Coeficiente de clustering medio para una red aleatoria equivalente ( $\langle C_{aleatoria} \rangle$ )	0,00276

#### 3.2. Conectividad de la red

Medida	Valor
Número de componentes conexas	1
Número de nodos de la componente gigante (% Respecto a la red total)	1.002 (1,99%)
Número de enlaces de la componente gigante (% Respecto a la red total)	2.770 (2,58%)

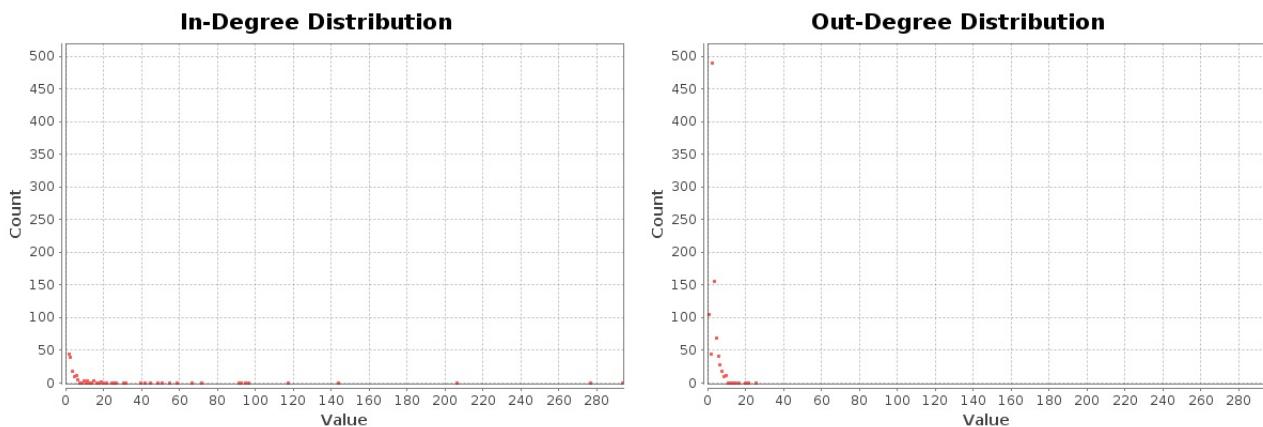
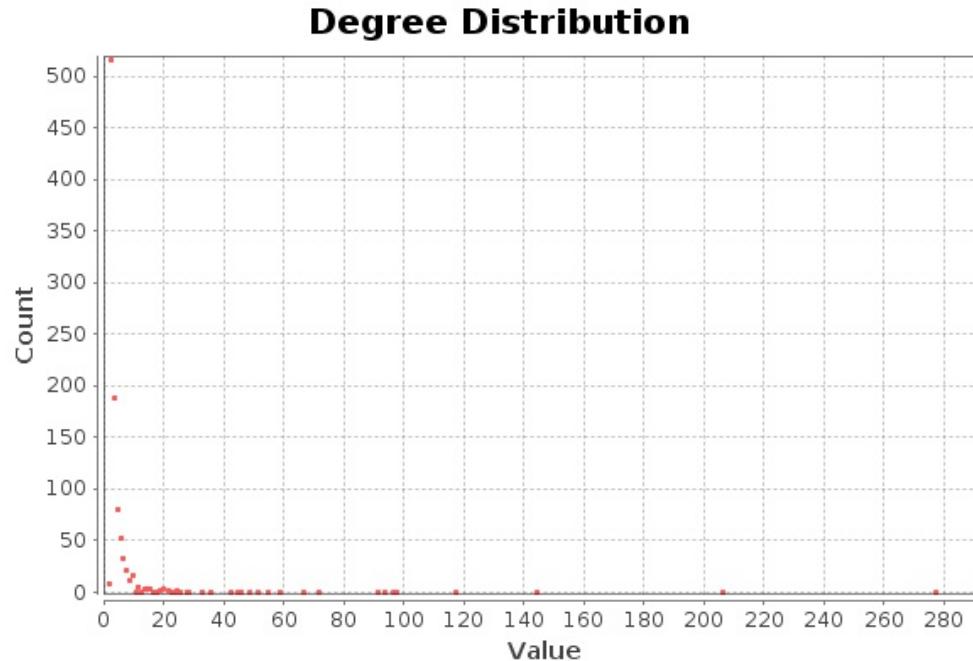
Estos valores obtenidos tienen todo el sentido del mundo, ya que para hacer la red manejable se ha filtrado por *componente gigante* lo que implica que solo se mantienen los nodos de la componente conexa mayoritaria.

### 4. Determinación de las propiedades de la red

#### 4.1. Distribución de grados

En las siguientes gráficas se recogen las distribuciones de grados totales, de entrada y de salida de la red. Como se puede observar todas tienen una clara forma de distribución de larga estela, lo que quiere decir que esta distribución sigue la ley de la potencia y por tanto parece ser una red social libre de escala.

También he realizado las gráficas de la distribución de grados con pesos, pero no he considerado significativo añadirlas al presentar la misma distribución pero con valores mayores.

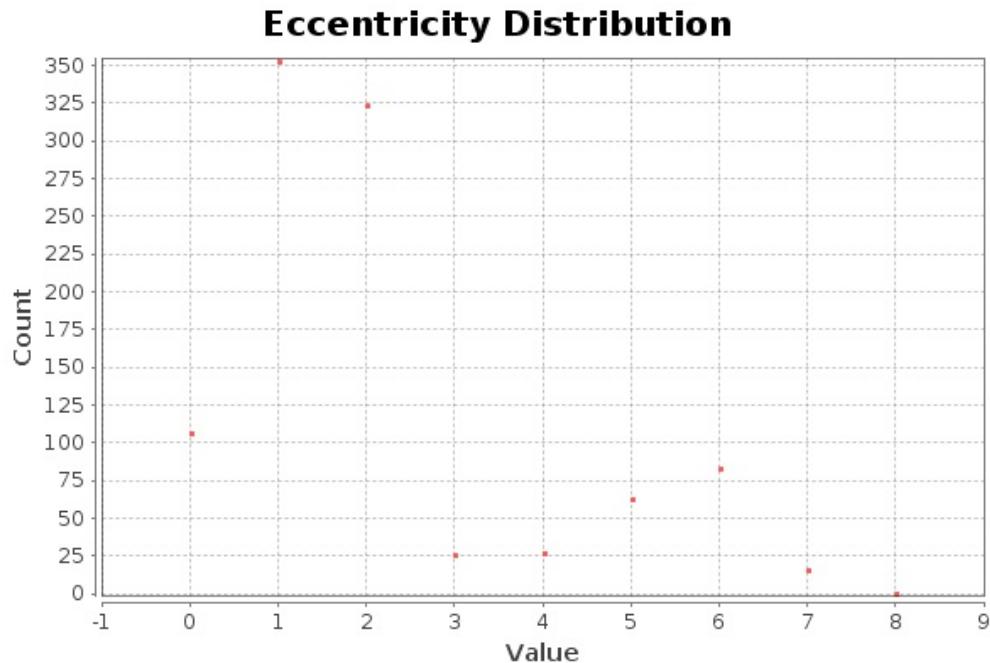


## 4.2. Distribución de distancias

En la siguiente gráfica se observa la distribución de la excentricidad que hace mide la distancia máxima de cada nodo al nodo más lejano de la red, lo primero que llama la atención es que existen nodos con valor 0, estos nodos son aquellos con grado de salida 0 por tanto no es posible llegar desde ellos al resto de nodos al tratarse de un grafo dirigido.

Cabe recordar que la distancia media obtenida para esta red es  $\langle d \rangle = 2,447$ , mucho menor que la distancia media para una red aleatoria equivalente  $\langle d_{aleatoria} \rangle = 6,7964$ , lo que indica que nos encontramos ante una red de mundo pequeño, pero incluso si comparamos la distancia media con la distancia media de una red libre de escala (mundo ultra-pequeño,  $\log(N)/\log(\log(N))$ )  $\langle d_{libre-escala} \rangle = 3,5747$  vemos que sigue siendo menor por lo que podemos deducir claramente que se cumple la propiedad de mundos pequeños.

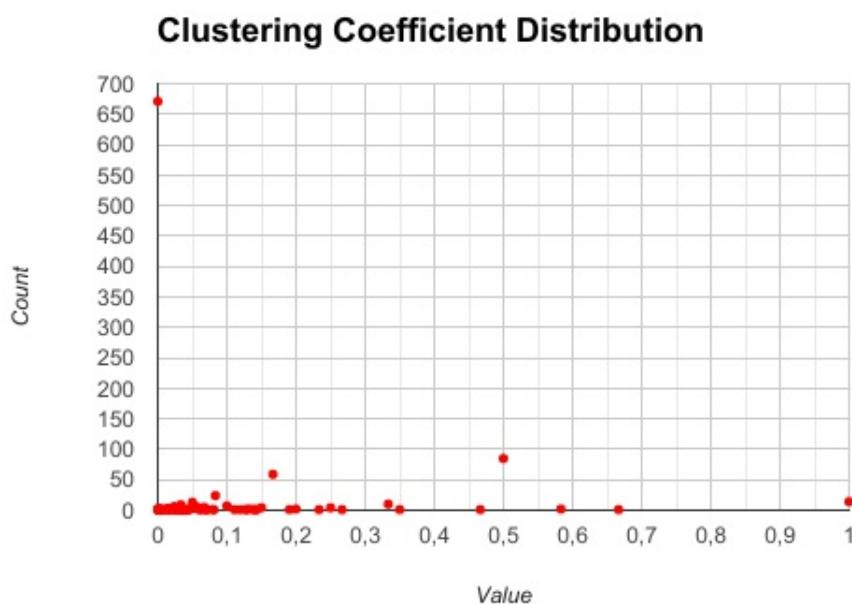
Se observa además, que a pesar de que la distancia máxima es 8, la frecuencia de distancias decrece mucho por encima de la media (aproximadamente el 77% de los nodos tiene una distancia máxima inferior a la media), lo que es consecuencia de la propiedad de mundos pequeños.



#### 4.3. Distribución de coeficiente de clustering medio

En la siguiente gráfica se observa la distribución de los coeficientes de clustering. Para comparar esta red con una red aleatoria hay que recordar el coeficiente de clustering medio para una red aleatoria equivalente ( $\langle C_{aleatoria} \rangle$ ) que era 0,00276, mucho menor que el coeficiente de clustering medio de esta red 0,082.

Se observa también que el coeficiente de clustering es mucho mayor en los nodos poco conectados que en los hubs, lo que indican que tienden a estar en zonas densamente pobladas, esto es una consecuencia de la jerarquía de redes.



## 4.4 Conclusiones

La ley de la potencia, comprobada con la distribución de grados, unido a la propiedad de mundos pequeños y al coeficiente de clustering tan alejado del de una red aleatoria me lleva a concluir que esta es una red social libre de escala como suele ser habitual con las redes sociales on-line

## 5. Cálculo de los valores de las medidas de análisis de redes sociales

Teniendo en cuenta de la pregunta de investigación planteada es quienes son los usuarios más influyentes en la discusión de Carrero Blanco es necesario realizar un análisis de redes sociales. En esencia hay que calcular medidas de centralidad para los nodos con el objetivo de determinar cuales son los más importantes según estas. En las siguientes tablas se pueden observar los diez nodos más importantes según las medidas de **grado, grado teniendo en cuenta los pesos, cercanía, intermediación y centralidad de vector propio**.

Nodo	Grado
kira_95	293
iunida	277
gerardotc	206
_ju1_	144
subversivos_	117
Yo_Soy_Asin	97
ctxt_es	96
protestona1	93
Xuxipc	91
rcabbrero75	71

Nodo	Grado con pesos
iunida	810
kira_95	789
_ju1_	461
gerardotc	444
subversivos_	302
protestona1	249
ctxt_es	247
Yo_Soy_Asin	233
Xuxipc	220
rcabbrero75	188

Nodo	Cercanía
protestona1	1
iunida	1
_ju1_	1
IsaAranjuez	1
ForretsGump	1
Well086	1
Famelica_legion	1
egel71	1
TRoderic	1
tecn_preocupado	1

Nodo	Intermediación
Yo_Soy_Asin	2341,4416666667
Klaseobreratk	1495,1083333333
carolacaracola5	1384,675
gerar666	1257,8166666667
PodemosAhora	995,2416666667
VictorGonz54	905,5416666667
vidushi_i	503,4166666667
LaloliFaz	453,7583333333
cantabriamiguel	265,325
protestona1	234

Nodo	Centralidad de vector propio
kira_95	1
gerardotc	0,7410629959
iunida	0,7241817322
_ju1_	0,3906274078
Yo_Soy_Asin	0,3362844387
subversivos_	0,2979991883
ctxt_es	0,243182962
protestona1	0,2374092107
Klaseobreratk	0,2309777553
Xuxipc	0,2198197312

Se pueden extraer diversas conclusiones de estos datos:

- La medida de cercanía no discrimina nada ya que existe un elevado número de nodos con

cercanía igual a uno (máxima ya que se encuentra normalizada). Esto probablemente se deba a que son nodos que se encuentran en el "medio", conectados a múltiples hubs, por lo que sus distancias al resto de nodos son bastante reducidas.

- La intermediación está pensada como una medida para capturar la "correduría", esto intuitivamente quiere decir que se considera mejor a un nodo cuando más grupos de nodos separados conecte. Esta medida puede ser interesante para algunas preguntas, pero en este caso concreto que busco determinar los actores más importantes, esta claro que esta muy relacionado con los usuarios más mencionados o retuiteados.
- En relación con esto último el grado parece una buena medida y como se puede ver el grado con o sin peso da unas medidas similares, se produce algún cambio de posición pero de entre los diez primeros ningún nodo es diferente.
- La centralidad de vector propio es una generalización de la medida de grado incorporando a la idea de que no solo influye que un nodo tenga un gran número de conexiones para ser importante, si no la "calidad" de esas conexiones, es decir la importancia de sus vecinos.

Por todo ello selecciono la **centralidad de vector propio** como la medida de centralidad más relevante para mi investigación.

## 6. Descubrimiento de comunidades en la red

Como método de descubrimiento de comunidades he aplicado el método de Lovain disponible en Gephi. Para aplicarlo hay que determinar la resolución lo que marca el tamaño de los grupos (cuanta más resolución, grupos más grandes). En la siguiente tabla recojo los valores de modularidad y número de comunidades detectadas respecto a la resolución empleada, hay que tener en cuenta que el proceso es algo aleatorio y que a igual resolución se pueden producir diferentes valores, aun así da una idea del rango de valores entre los que estará.

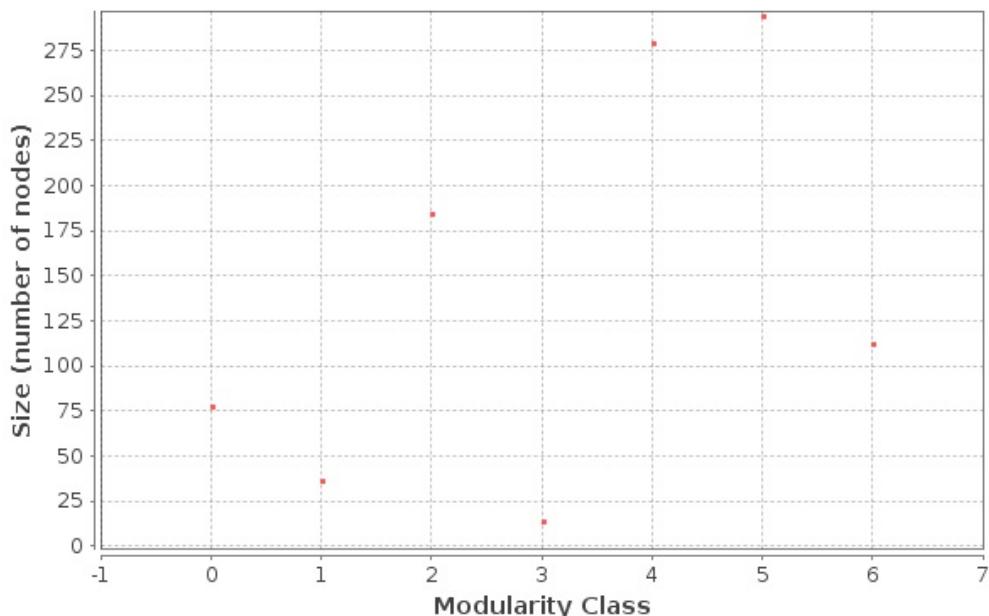
Resolución	Modularidad	Nº Comunidades
0,85	0,428	16
1	0,431	11
1,2	0,422	9
1,3	0,415	7
1,5	0,401	6

Hay que mencionar que la modularidad de una partición de comunidades es un valor definido entre [-1, 1] que mide la calidad de esta partición, siendo favorable cuanto mayor sea su valor. Valores superiores a 0,3 denotan que puede haber una estructura de comunidades subyacente a la red.

Para realizar un análisis de comunidades es necesario que no sean demasiadas ya que eso incrementa su complejidad, por ello sacrificando algo de modularidad he decidido aplicar una resolución de 1,3 obteniendo una modularidad de 0,415 y siete comunidades.

En la siguiente gráfica se observa la distribución de tamaño de estas comunidades.

### Size Distribution



Para analizar si los resultados son significativos se puede recurrir a los valores de modularidad, que son superiores a 0,3 como he comentado, pero tampoco excepcionalmente, por ello habrá que tomar las comunidades con cautela. Hay que tener en cuenta que esta red social proviene de una red mucho más grande que ha sido filtrada, por dar un ejemplo si aplicamos el método de Lovaina a la red total se obtienen valores del orden de 0,516 con resolución 1.

Otra forma de ver si los resultados son significativos, añadiendo conocimiento por mi parte, sería observar si los usuarios que forman los grupos tienen cosas en común. Para ello he añadido un filtro por *clase de modularidad* y he ido viendo una a una las comunidades fijándome en los nodos más centrales según la centralidad de vector propio. Se observan algunas cosas curiosas:

- La **comunidad 5** tiene como nodos más centrales a @iunida y @gerardotc (colaborador del jueves) además pertenecen a él @eljueves, @agarzon (Alberto Garzón, secretario general de IU), @iumadrid.
- En la **comunidad 4** aparecen @policia y @guardiacivil junto con @kira\_95, el usuario de Cassandra Vera, la chica condenada por sus chistes.
- La **comunidad 3** se encuentra formada por @TRoderic (Toni Roderic) presidente de los verdes, @verdesinfo (el usuario oficial del partido los verdes), los usuarios de los verdes en el país valenciano, cataluña o gandía.
- En la **comunidad 1** encontramos diversos medios de comunicación como @La\_Ser, @eldiarioes, @sextaNoticias o @DebatAlRojoVivo.
- Por último se puede observar en la **comunidad 0** algún medio de comunicación alternativo como @ctxt\_es y @ColpisaNoticias o divisiones alternativas de otros medios como @Tremending de Público y @verne de El País.

Todo esto nos lleva a pensar que sí que existe una estructura de comunidades subyacente a la red social y tienen cierto sentido los miembros de estas comunidades.

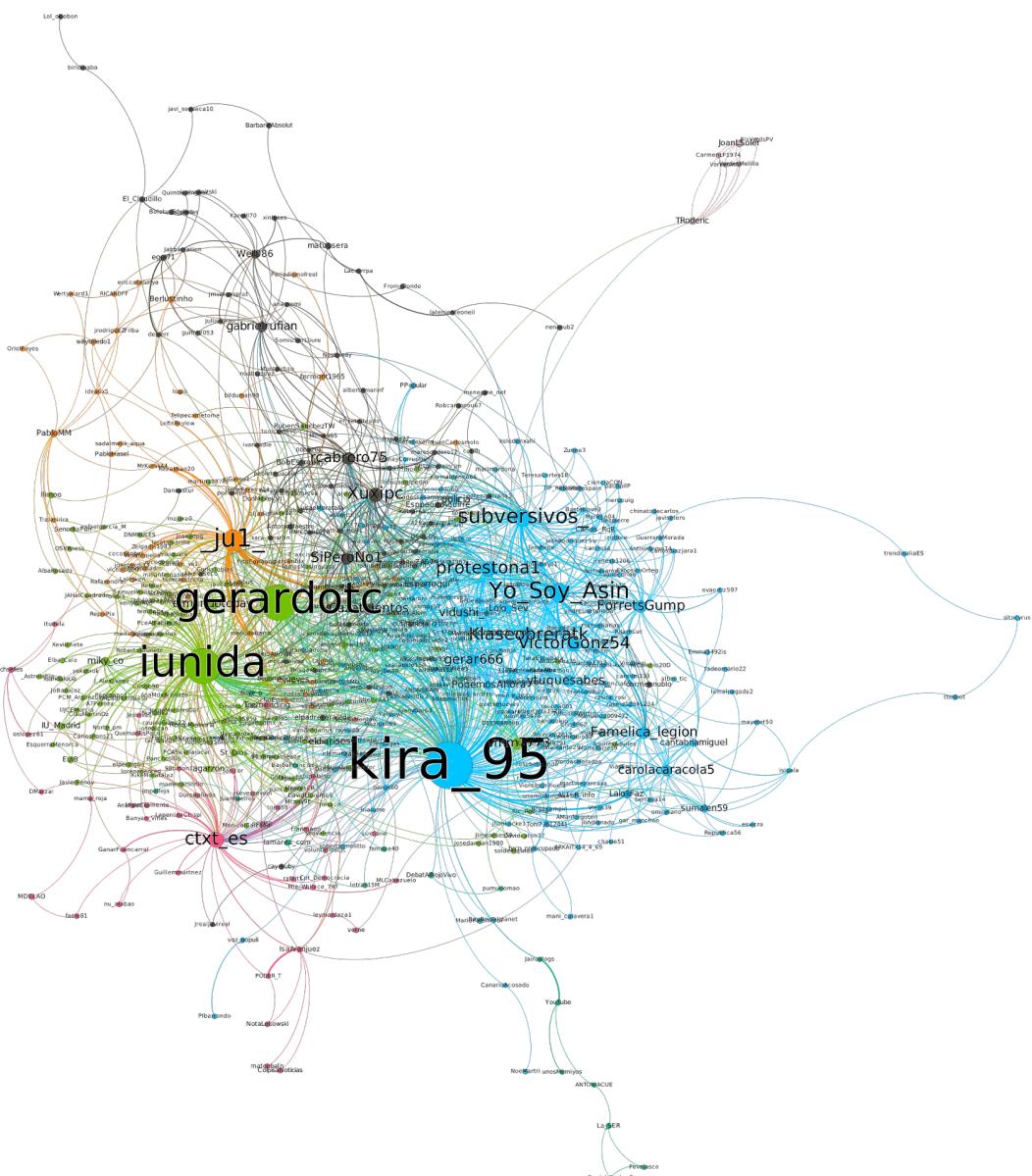
## 7. Visualización de la red social

Para una correcta visualización además del filtrado que he aplicado voy a filtrar los nodos "menos relevantes" usando para ello una medida de prestigio en Twitter, los seguidores o **followers**. Por ello he rellenado este valor manualmente en los usuarios creados al añadir los enlaces como ya comenté y filtraré quedándome con aquellos usuarios con al menos 500 seguidores. Haciendo esto paso a tener una red de 525 nodos y 1487 aristas.

Destacar que si se desea ver una visualización en más detalle es recomendable acceder a los archivos .png directamente en lugar de verlas aquí.

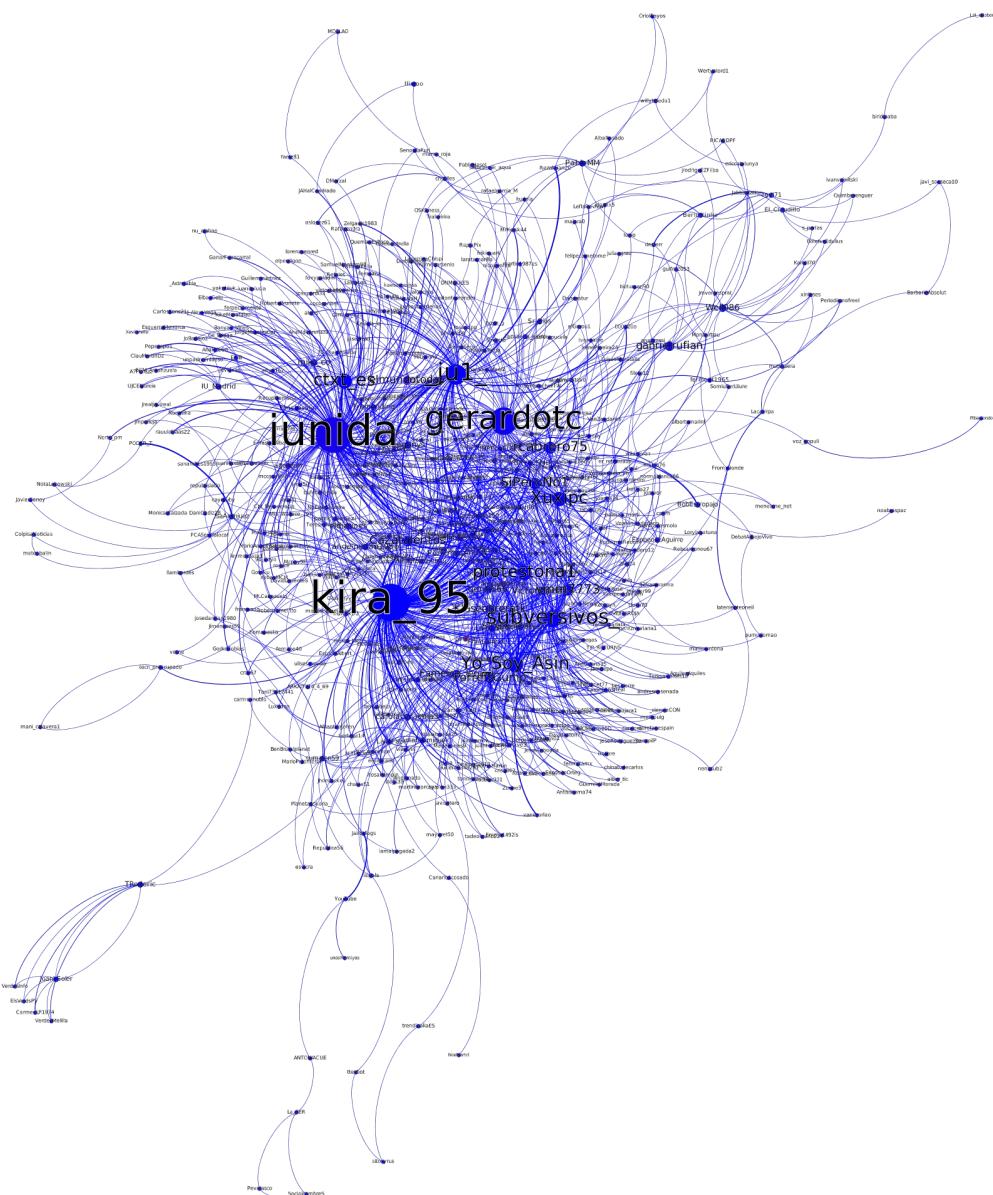
## 7.1 Visualización 1

Para una primera visualización estableceré el tamaño de los nodos de acuerdo a su centralidad de vector propio, el tamaño de las etiquetas de acuerdo al tamaño de los nodos y el color de los nodos de acuerdo a la estructura de comunidades calculada en el paso previo. Como algoritmo de layout usaré un **ForceAtlas2**, basado en el algoritmo de T.Kamada y S.Kawai, ha sido creado para espaciar las redes libres de escala como es la mía[7], como parámetros de este he usado: Escalado = 200, Gravedad = 3, Evitar el solapamiento = True y el resto de parámetros por defecto. Además de esto desde la ventana de *Previsualización* de Gephi he aplicado un contorno de 1px blanco a las etiquetas para mejorar su legibilidad y he acortado, moviendo los nodos manualmente, la cola mas inferior ya que se alejaba demasiado del centro y hacia que la imagen resultante fuera demasiado larga.



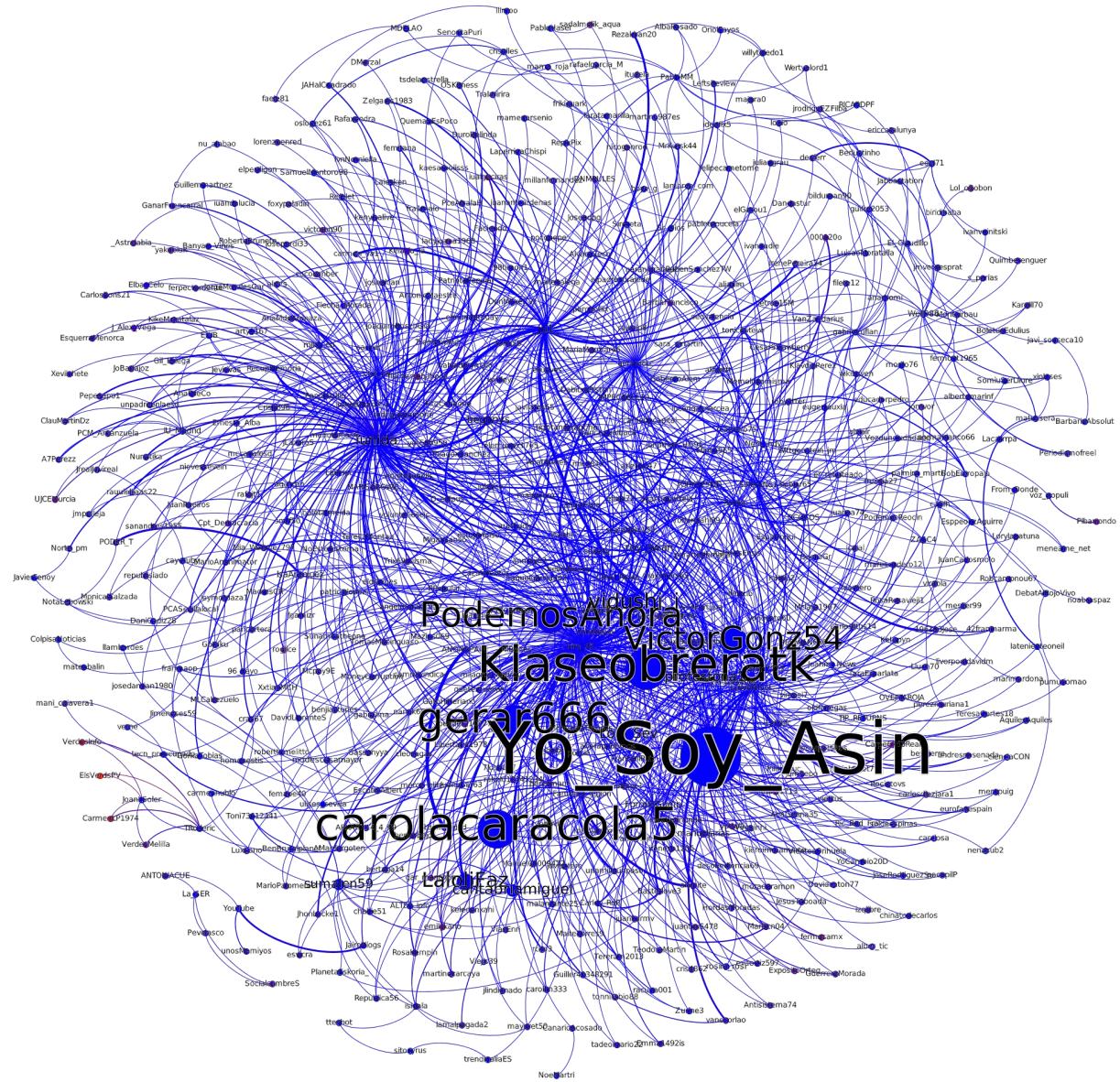
## 7.2 Visualización 2

Para esta visualización el tamaño de los nodos determina el grado, el tamaño de las etiquetas está relacionado con el del los nodos y el color un degradado de azul(-) a rojo(+) que marca el número de seguidores, teniendo en cuenta esto he cambiado el número de seguidores de @YouTube y considerarlo un *outlayer* al no tratarse de un perfil en español y tener un numero mucho más elevado de seguidores que el resto. He utilizado ***Yifan Hu*** como algoritmo de layout con los parámetros Distancia óptima = 300, Ratio de paso = 0.99 y el resto por defecto; tras esto he aplicado un ***Noverlap*** para evitar el solapamiento de nodos según recomienda [7] con los parámetros Velocidad = 1, Ratio = 2 y Margen = 10. Esta vez también he comprimido la cola inferior manualmente.



### 7.3 Visualización 3

Por último presento una visualización en la que el tamaño de los nodos representa la intermediación, el tamaño de las etiquetas esta relacionado con el del los nodos, el color un degradado de azul(-) a rojo(+) que marca el coeficiente de clustering. Para el layout he utilizado el algoritmo **Fruchterman-Reingold** con los parámetros Área = 10.000, Gravedad = 7 y Velocidad = 1, tras esto he aplicado otra vez un **Noverlap** con los mismos parámetros que en la visualización previa.



## 8. Discusión de los resultados obtenidos

Volviendo a la pregunta de investigación ¿Cúales los usuarios más relevantes en la discusión de Twitter sobre las palabras "Carrero Blanco"? Se puede observar mediante las medidas de centralidad *grado* y *centralidad de vector propio* así como en las visualizaciones 1 y 2 que los 2 nodos más centrales son **@kira\_95** (nombre de usuario de Cassandra Vera, condenada por los tuits) e **@iunida** (perfil oficial de Izquierda Unida).

Que Cassandra Vera séa el nodo más central tiene todo el sentido ya que su sentencia fue el hecho que elevó "Carrero Blanco" a *Trending Topic* en España. Respecto a IU esto se explica porque a causa de esta sentencia se inició una campaña de apoyo liderada por Podemos e IU en Twitter difundiendo los tuits que condenaron a Cassandra [8].

Como otras conclusiones destacar que como se aprecia en la visualización 2 los nodos con más grado no son necesariamente aquellos con más seguidores, de echo ni uno de los nodos más grandes tiene una tonalidad rojiza como debería. En la visualización 3 he pretendido destacar como las medidas de intermediación o coeficiente de clustering no resultan muy significativas para esta investigación.

A modo de conclusiones más generales basadas en mis observaciones a lo largo de esta práctica se comprueba que la gente implicada en la discusión demuestran tendencias ideológicas de izquierdas como de nota la presencia de partidos políticos como IU, Los Verdes o Podemos y personajes de la esfera política de este país como Gabriel Rufián (diputado de ERC) o Alberto Garzón (Secretario de IU y diputado de Unidos Podemos). También aparecen numerosos medios de comunicación no mayoritarios como CTXT, El Diario o Público así como periodistas o diversas publicaciones satíricas como la Revista Mongolia o El Jueves.

Esto se puede entender porque mucha gente tanto dentro como fuera de España ha considerado la sentencia de Cassandra como un ataque a libertad de expresión no justificado en el enaltecimiento del terrorismo teniendo en cuenta que Carrero Blanco fue una figura política muy importante en la última época del franquismo y fue asesinado en 1973, muchos años antes de que Cassandra naciera. Esto resulta preocupante pero no se trata de un hecho aislado, la Audiencia Nacional ha condenado a al menos 30 personas por enaltecimiento del terrorismo en las redes sociales desde 2016 [9]. Con esto no quiero decir que esté en contra de las sentencias si no que hay que entender los comentarios en una red social como Twitter en su contexto. No se pueden usar los mismos métodos que en siglo pasado para juzgar delitos en las redes, es necesario que intervengan expertos que conozcan y sepan analizar redes sociales porque, a día de hoy, hay agentes en los cuerpos y fuerzas de seguridad del estado dedicados a perseguir delitos en las redes sociales que tienen un conocimiento muy superficial de estas y se limitan a introducir algunas palabras en buscadores observar el resultado [10].

### Anexo: **twitter\_search**

**Twitter\_search** es un programa en Python para recuperar tuits que contengan ciertas palabras o hashtags y exportarlos en ficheros .csv, permite seleccionar las fechas entre las que buscar los tuits, el número de tuits a recuperar y el idioma de los tuits. Usa el módulo Tweepy [2] para manejar las llamadas a la API de Twitter.

Tras descargar los tuits genera tres ficheros .csv:

- `queryX_(dateSince_dateUntil)_Tweets.csv`: Contiene los siguientes campos
  - "ID": Identificador del tuit usado por Twitter.
  - "User": Nombre del usuario que escribió el tweet, este corresponde al "screen\_name" devuelto por la API, es decir al @usuario.
  - "Created\_at": Fecha y hora de creación del tuit.
  - "Tweet": Texto del tuit en sí.
  - "Following": Número de personas a las que sigue este usuario.
  - "Followers": Número de personas que siguen a este usuario.
  - "Tweets\_count": Número de tweets escritos por este usuario.
  - "Favourites": Número de me gustas de este usuario.
  - "Time\_Zone": Zona horaria del usuario.
  - "Location": Localización del usuario (si la tiene puesta en su perfil).
- `queryX_(dateSince_dateUntil)_Users.csv`: Contiene el ID y Label de los nodos que corresponden al nombre de usuario (@usuario) además de los mismos datos del usuario que el fichero previo (Following, Followers, Tweets\_count, Favourites, Time\_Zone y Location) pero con una sola entrada por cada usuario, en el caso previo si un mismo usuario ha escrito varios tuits aparecerá varias veces en el fichero. Está preparado para ser importado en [Gephi](#) como una hoja de cálculo, en concreto como **tabla de nodos**.
- `queryX_(dateSince_dateUntil)_Edges.csv`: Contiene las relaciones de los tuits descargados, es decir menciones a usuarios y retuits. Está preparado para ser importado en [Gephi](#) como una hoja de cálculo, en concreto como **tabla de aristas**.

Para usar este programa es necesario registrar una aplicación en Twitter para obtener los credenciales necesarios para consultar la API, una vez obtenidos hay que copiarlos como cadenas de caracteres en el fichero `credentials.py`.

## Tutorial

Para obtener una descripción detallada de los pasos a realizar para utilizar este programa se puede seguir el siguiente tutorial que también utiliza Tweepy para descargar datos de Twitter y utilizarlos en Gephi [3].

## Uso del programa

Es necesario instalar Python 3 y virtualenv antes de usar este programa, una vez hecho esto se pueden instalar el resto de dependencias ejecutando los siguientes comandos (los comandos han sido probados en una máquina Debian, pueden ser diferentes en su S.O.):

```
virtualenv -p$(which python3) venv
source venv/bin/activate
pip install -r requirements.txt
```

Y para ejecutar el script hay que introducir el siguiente comando:

```
python twitter_search.py
```

```
(venv)aythae@debian:~/Escritorio/GIW/P2/twitter_search (master *%=$) $ python twitter_search.py
Twitter_search
Recover tweets of some topic between 2 dates and export it into .csv files to be imported in Gephi
Author: Aythami Estévez Olivas <aythae [at] gmail [dot] com>
Repository: https://github.com/AythaE/twitter_search

Insert the search query to look for tweets: Carrero Blanco
Choose the number of tweets to recover: 250
Insert the first creation date of the tweets to recover formatted as YYYY-MM-DD (no more than 7 days before now)[Leave empty for not limit]:
Insert the last creation date of the tweets to recover formatted as YYYY-MM-DD (no more than 7 days before now)[Leave empty for not limit]:
Insert the desire language for the tweets (given by an ISO 639-1): es

First Search, recovering 100 tweets
Another Search, recovering another 100 tweets
Last Search, recovering last 50 tweets
Total tweets recovered: 250

Saving tweets to Carrero_Blanco_(2-4-2017)_Tweets.csv
Saving tweets users to Carrero_Blanco_(2-4-2017)_Users.csv
Saving tweets relationships to Carrero_Blanco_(2-4-2017)_Edges.csv
Number of relationships lost due to errors: 2/274

To import them in Gephi, import the Users.csv as a nodes table spreadsheet and the Edges.csv as edges table spreadsheet
(venv)aythae@debian:~/Escritorio/GIW/P2/twitter_search (master *%=$) $
```

## Bibliografía

- [1]: La Audiencia Nacional condena a un año de cárcel a Cassandra por los tuits sobre Carrero Blanco (n.d.). Recuperado el 1 de Abril de 2017, desde [http://www.eldiario.es/politica/Audiencia-Nacional-condena-tuitera-Cassandra\\_0\\_627487833.html](http://www.eldiario.es/politica/Audiencia-Nacional-condena-tuitera-Cassandra_0_627487833.html)
- [2]: Tweepy: Twitter for Python! (n.d.). Recuperado el 1 de Abril de 2017, desde <https://github.com/tweepy/tweepy>
- [3]: Maths with Python 6: Twitter API – Tweepy for social media and networks (with Gephi) (n.d.). Recuperado el 2 de Abril de 2017, desde <https://thebrickinthesky.wordpress.com/2014/06/26/math-with-python-6-twitter-api-tweepy-for-social-media-and-networks-with-gephi/>
- [4]: A. Quirin (2014). The Pathfinder algorithm: the original, binary, Fast and MST-variants. Recuperado el 3 de Abril de 2017, desde <http://aquirin.ovh.org/research/mstpathfinder.html>
- [5]: Network Workbench: A Workbench for Network Scientists (n.d.). Recuperado el 3 de Abril de 2017, desde <http://nwb.cns.iu.edu/index.html>
- [6]: Network Workbench Tool User Manual 1.0.0. (2009). Recuperado el 3 de Abril de 2017, desde <http://nwb.cns.iu.edu/Docs/NWBTool-Manual.pdf>
- [7]: Gephi Tutorial Layouts (2011). Recuperado el 5 de Abril de 2017, desde <https://gephi.org/users/tutorial-layouts/>
- [8]: IU ‘inunda’ su Twitter con los chistes sobre Carrero Blanco (n.d.). Recuperado el 5 de Abril de 2017, desde <http://www.elplural.com/politica/2017/03/30/iu-inunda-su-twitter-con-los-chistes-sobre-carrero-blanco>
- [9]: La Audiencia ha condenado al menos a 30 personas por enaltecimiento de ETA y Grapo en redes sociales desde 2016 (n.d.). Recuperado el 5 de Abril de 2017, desde <http://www.publico.es/sociedad/deriva-justicia-audiencia-condenado-30.html>
- [10]: El experto en redes sociales de la Guardia Civil que detuvo a Strawberry no sabe qué son las impresiones de un tuit (n.d.). Recuperado el 5 de Abril de 2017, desde <http://www.publico.es/sociedad/twitter-experto-redes-sociales-guardia.html>