FACULTAD DE ESTUDIOS ESTADÍSTICOS

MÁSTER EN MINERÍA DE DATOS E INTELIGENCIA DE NEGOCIOS

Curso 2019/2020

Trabajo de Fin de Máster

Análisis de las distintas redes que se generan en Twitter e identificación de temáticas abordadas bajo el hashtag #Cuéntalo.

***Alumno*: Eva Garrido Herranz**

***Tutor*: Daniel Gómez González**

Septiembre de 2020

UNIVERSIDAD COMPLUTENSE
MADRID

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

1. INTRODUCCIÓN

A lo largo de los años las formas de comunicación han ido cambiando. Hoy en día, las más usada son las redes sociales, gracias a las que millones de personas interactúan entre sí diariamente desde cualquier punto del mundo.

Las redes sociales se han convertido en una gran fuente de información sobre gustos, preferencias y opiniones de los usuarios, es por esto que el análisis de redes sociales (ARS) es una de las claves de las compañías para captar clientes potenciales, crear influencia o hacer seguimiento de su rendimiento, entre otros usos.

Twitter se ha convertido en una de las redes sociales más utilizadas en la actualidad, de hecho, ellos mismo se definen como: "Twitter es lo que está sucediendo y el lugar donde la gente está hablando de ello." Por la accesibilidad al medio, las noticias se propagan muy rápido llegándose a hacer virales en cuestión de minutos, pero de igual manera ocurre con las "fake news", algunas llegando a alarmar a la comunidad sin motivo real. Un temprano análisis de la red podría llegar a prevenir que se difundan estos contenidos falsos. A partir de esta intención, surge la idea de establecer un patrón común de comportamiento en aquellas noticias que se están empezando a convertir en virales.

Un claro ejemplo de viralización de un tema fue el asociado al hashtag #Cuéntalo, cuyo origen se remonta al 26 de abril de 2018, cuando la Audiencia de Navarra dictó la sentencia del caso de La Manada, en el que a un grupo de cinco amigos se les acusaba de haber violado a una joven durante la celebración de san Fermín en Pamplona en 2016. Aunque más tarde se consideró que hubo violación, ese día los jueces decretaron que había sido un abuso sexual. Esta sentencia causó rechazo por gran parte de la población española, llegando a organizarse manifestaciones multitudinarias por todo el país. El mismo 26 de abril, la periodista Cristina Fallarás, ante la indignación por la sentencia, abrió un hilo en Twitter en el que instaba a las mujeres a contar su historia a través del hashtag #Cuéntalo. En pocas horas este hashtag ya era tendencia en España, donde miles de personas, principalmente mujeres, relataron todo tipo de agresiones, violaciones, abusos o intimidaciones sexuales que habían recibido. Muchas de ellas lo contaron por primera vez a través de este medio, ya que hasta el momento no se habían atrevido a contárselo a nadie. Dos días más tarde el hashtag #Cuéntalo también se hizo viral en Latinoamérica.

Este hashtag y las redes generada a partir de las interacciones de sus participantes serán el objeto de estudio de este trabajo, como se detallará a continuación.

2. OBJETIVOS

Este trabajo será la fase inicial de un proyecto que tiene como objetivo avanzar en el estado del arte de la investigación sobre las redes sociales, centrándose principalmente en el tipo de relaciones que se establecen entre los participantes de un tema viral dentro de una red social. La idea del proyecto final es comparar distintas redes sociales que se forman bajo un hashtag cuando empieza a convertirse en "trending topic" en Twitter, y obtener un patrón de comportamiento en las relaciones entre usuarios. Uno de los principales problemas, que se abordaran en este estudio, es que en Twitter existen diferentes formas de relacionarse con otros usuarios, mediante retuits, menciones, citas o respuestas, de aquí podemos diferenciar dos tipos de relaciones distintas:

- La primera sería una relación de transmisión de la información, en la que se comparte el tuit que ha generado otro usuario mediante el retuit.
- La segunda sería una relación de discusión, en la que se establece una conversación con otro usuario través de la mención, respuesta o cita. Además, mediante este tipo de relaciones se está generando contenido nuevo.

Este trabajo tendrá como objetivo estudiar por separado la red de retuits, y red formada por las menciones, citas y respuestas, llamada a partir de ahora red de menciones, para determinar si existen suficientes diferencias significativas entre ambas redes como para continuar estudiándolas por separado.

El hashtag escogido para llevar a cabo el estudio ha sido #Cuéntalo por dos motivos principales:

- Fue un tema que se hizo viral en poco tiempo y en el que hubo una gran participación, por lo que se dispondrá de un tamaño muestral grande.
- Este hashtag tiene gran importancia social ya que muchas mujeres relataron las agresiones sexuales sufridas en algún momento de sus vidas con el objetivo de evidenciar la veracidad de las denuncias asociadas al tema y la dimensión del conflicto.

Aunque ya se llevaron a cabo diferentes análisis de texto acerca de los tuits sobre #Cuéntalo, dado el gran impacto que tuvo el movimiento, otro de los objetivos que se pretende desarrollar en este estudio es profundizar en ciertos tipos de análisis, como por ejemplo llevar a cabo una identificación más detallada de las temáticas que se hablaron, y realizar un análisis más exhaustivo de la edad de las víctimas cuando sufrieron los hechos que comparten.

Según lo anteriormente expuesto y a modo resumen, el objetivo principal de este trabajo será determinar las diferencias y semejanzas entre una red de trasmisión de la información y una red de diálogo. Y, como objetivos secundarios se abordarán y estudiarán durante esta memoria algunos de los problemas asociados al análisis de redes sociales y problemas de text mining para esta temática, que son, entre otros, los siguientes: detección de temáticas que se abordaron en los tuits bajo el hashtag Cuéntalo, identificación de palabras clave, identificación de comunidades dentro la red, identificación de líderes de opinión globales y locales según su rol dentro de la red diferenciando entre emisor, intermediario y receptor.

Finalmente, se ha diseñado un algoritmo que permite identificar la edad que tenían las víctimas cuando sufrieron las agresiones en los casos que se relatan en la red, comparando después esta información con la proporcionada por estudios oficiales de género y agresiones en España, y planteándose como información adicional a la que se recoge mediante procedimientos oficiales y más costosos.

3. ESTADO DEL ARTE

En este apartado se mostrarán los resultados obtenidos en otras investigaciones relacionadas con los temas que se tratan en este trabajo.

3.1. Análisis de Redes Sociales (ARS)

El concepto de red social empezó a utilizarse a principios de siglo XX por los sociólogos para referirse a conjuntos complejos de relaciones entre miembros de sistemas sociales. A partir de entonces, se empezaron a desarrollar métodos analíticos básicos para medir las relaciones sociales. Más tarde se desarrollaron modelos capaces de explicar el comportamiento de gran parte de las redes sociales.

El ARS no solo abarca las redes de tipo social, sino que se podría decir que abarca todo tipo de estructuras compuestas por un grupo finito de actores con una serie de relaciones entre ellos. La forma de estudiar estas relaciones es a través de la teoría de grafos, donde los nodos serían los actores y las aristas las relaciones entre ellos. Hoy en día el análisis de redes sociales se está aplicando en diferentes áreas científicas aparte de la social como pueden ser, entre otras, la biología, física, antropología, informática o las ciencias de la información. Algunas de las aplicaciones prácticas del ARS son:

- Investigación de terrorismo, crímenes o fraude.
- Estudio de la propagación de una pandemia.
- Creación de algoritmos de recomendación de productos.
- Estudio de las conexiones neuronales.

Se ha de destacar que, debido a las nuevas formas de relacionarse la sociedad, el ARS se ha convertido en una técnica clave en la sociología moderna.

Un estudio sobre el mapeo de redes temáticas en Twitter realizado por Marc a. Smith, Lee Rainie, Ben Shneiderman e Itai Himelboim en 2014, detectó las siguientes estructuras distintivas de multitudes sociales:

- **Polarized Crowd:** las discusiones polarizadas presentan dos grupos grandes y densos que tienen poca conexión entre ellos.
- **Tight Crowd:** estas discusiones se caracterizan por personas altamente interconectadas con pocos participantes aislados.
- **Brand Clusters:** cuando se discuten en Twitter productos o servicios conocidos, o temas populares como celebridades, a menudo hay comentarios de muchos participantes “aislados”.
- **Community Clusters:** algunos temas populares pueden desarrollar varios grupos más pequeños, que, a menudo, se forman alrededor de unos pocos centros, cada uno con su propia audiencia, personas influyentes y fuentes de información.

- **Broadcast Network:** los comentarios de Twitter sobre noticias de última hora tienen una estructura de centro y radio distintiva en la que muchas personas repiten lo que tuitean las organizaciones de noticias y medios prominentes.
- **Support Network:** las quejas de los clientes para una empresa importante a menudo se manejan mediante una cuenta de servicio de Twitter que intenta resolver y administrar los problemas de los clientes en torno a sus productos y servicios.

3.2. #Cuéntalo

Cabe destacar el proyecto #Cuéntalo en el que se recopilaron, analizaron y visualizaron todos los tuits escritos durante los primeros 14 días de la iniciativa. En este proyecto estuvieron involucrados los archiveros Vicenç Ruiz y Aniol María que recogieron los tuits en tiempo real, la periodista especializada en Tecnologías de la Información y datos Karma Peiró, el equipo del BSC liderado por Fernando Cucchietti que, procesó, analizó y desarrolló la visualización, y la periodista e impulsora del hashtag, Cristina Fallarás. El objetivo de este proyecto era evidenciar la veracidad de las denuncias y la dimensión del conflicto.

Los principales resultados que se obtuvieron fueron:

- Tuvo un impacto internacional en el que participaron 60 paises distintos. El 38% de los tuits originales y 26% de los retuits procedían de España. Argentina también tuvo una gran participación, puesto que supuso un 30% de los tuits originales y un 43% de los retuits.
- El movimiento comenzó en España y tardó cerca de dos días en saltar a Latinoamérica.
- Unos 51.000 tuits relataban testimonios, 50.000 eran tuits de apoyo y cerca de 4.000 se posicionaban en contra del movimiento.
- Se categorizaron aquellos tuits testimoniales en 6 temáticas distintas y los resultados fueron que 5.000 hablan de un asesinato, 7.000 de violación, 14.000 de agresión sexual, 8.000 de maltrato, 18.000 de acoso y 15.000 de miedo.

Además, se detectaron 3.000 tuits en los que las víctimas tenían menos de 18 años, pero no se especifica si fue de la muestra que tomaron para la clasificación manual inicial o del total de los tuits, es un tema que se dejó para futuros trabajos.

La visualización que se desarrolló es la siguiente:
http://proyectocuentalo.org/index.html

4. METODOLOGÍA Y SOFTWARE

4.1. Metodología

Para poder cumplir los objetivos anteriormente establecidos se han seguido los siguientes pasos:

GRÁFICO 1: TAREAS QUE SE VAN A LLEVAR A CABO EN EL TRABAJO

- Extracción de los datos → recopilar los datos de Twitter para poder llevar a cabo el estudio.
- Depuración y análisis descriptivo → analizar las principales variables con el objetivo de depurar y explorar el set de datos.
- Text mining → detectar las principales temáticas que se abordan en los tuits y analizar la edad que tenían las víctimas cuando les ocurrieron los hechos que comparten. Para lograrlo antes habrá que depurar el texto.
- Análisis de redes → hallar las principales características de cada una de las redes: tipología, líderes de opinión y comunidades

4.1.1. Extracción de datos

La fuente de origen de los datos que se van a analizar es Twitter. Twitter es una página web de microblogueo que permite escribir y leer mensajes de hasta 280 caracteres llamados tweets o tuits. También es una plataforma bidireccional en las que se puede interactuar con otros usuarios respondiendo a sus tuits, retuiteando, citando o hasta se pude escribir un tuit nuevo en el que se mencione a otro usuario. Además, es frecuente el uso de etiquetas, llamadas hashtags, al escribir los tuits para categorizarlos, de esta manera se unifica toda la información que escriben los usuarios sobre un tema en concreto.

La manera más común de extraer la información de forma gratuita de Twitter es a través de la interfaz de programación de aplicaciones (API). Alguna de las API's que ofrece Twitter son:

- **API Rest.** Permite leer y escribir tuits, consultar la información de un usuario, buscar tuits, usuarios, etc.
- **API Streaming.** Permite recibir información de los tuits en tiempo real.

Para utilizar cualquiera de las API´s, es necesario estar registrado en la plataforma y, a partir de la cuenta, crear una Twitter App asociada. Al crear una Twitter App, Twitter proporciona una serie de claves y tokens de identificación que permiten acceder a la aplicación y extraer información. Seguidamente se muestra un ejemplo de código para conectarse a la API.

```
token <- create_token(
  app = "PR                ted",
  consumer_key = "Po7C7              tuS9z",
  consumer_secret = "KQRf45eJ                                        XJg8L")
```

GRÁFICO 2: EJEMPLO CÓDIGO PARA CONECTARSE A LA API DE TWITTER

Como medida de seguridad, cada vez que la aplicación se intente conectar habrá que autorizarla desde la cuenta de Twitter vinculada.

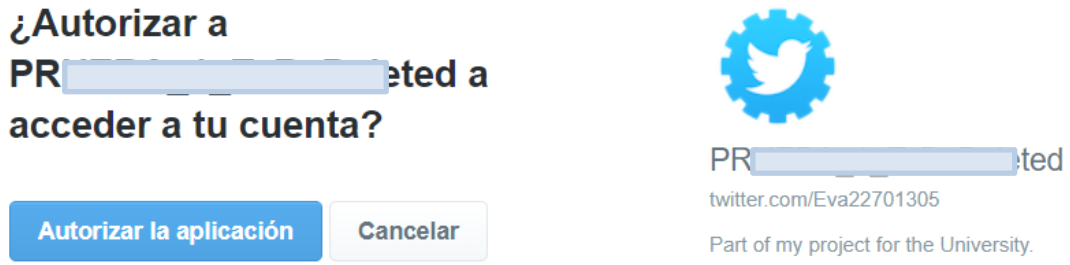

GRÁFICO 3: AUTORIZACIÓN DE LA APLICACIÓN PARA ACCEDER A LA CUENTA DE TWITTER VINCULADA

Para este trabajo se recopilarán todos aquellos tuits, y sus interacciones asociadas con otros usurarios, que se generaron entre el 26 y 29 de abril bajo el hashtag #Cúentalo. Se han escogido concretamente hasta el día 29 ya que, según se ha podido comprobar en el estado del arte, a partir de dicha fecha empezó a disminuir la actividad en España, aumentando en los países de Latinoamérica.

Debido a las limitaciones impuestas por Twitter sobre el uso de las API's en las que solo se permite extraer tuits de una semana atrás, cuando se empezó a realizar el trabajo no se consiguieron los datos del inicio del movimiento #Cuéntalo, por lo que fueron proporcionados por la profesora Dña. Mariluz Congosto.

4.1.2. Depuración y análisis descriptivo

En este apartado se pondrán de manifiesto todas las variables de las que se dispone y se explorarán algunas de ellas con el objetivo de hallar datos erróneos y extraer las características más representativas del conjunto. Para llevar esto a cabo se utilizarán técnicas visuales y medidas estadísticas principalmente.

4.1.3. Text mining

El text mining se puede definir como el análisis matemático para deducir patrones y tendencias que existen en los textos. Este tipo de análisis se desarrolla en dos fases claramente diferenciadoras, la fase de preprocesamiento o depuración, que trata de crear una representación estructurada intermedia útil para la siguiente etapa, eliminando aquellas palabras que no sean relevantes y normalizándolas, y la fase de data mining, que consiste en descubrir el conocimiento que está oculto en el corpus original.

En el caso del análisis de los tuits, la parte de depuración del texto es esencial, ya que muchos de ellos contienes urls de páginas web, caracteres ASCII y además, al ser texto libre, está sujeto a posibles errores ortográficos. Para llevar a cabo esta fase de preprocesamiento se seguirán los siguientes pasos:

- **Limpieza inicial del texto:** se eliminarán expresiones como urls de páginas web, caracteres ASCII, signos de puntuación, espacios en blanco inadecuados, etc.

- **Parsing:** se descompone la frase en sus unidades atómicas: la palabra. De esta manera se convierten los datos no estructurados en datos estructurados.

- **Elaboración de diccionarios:** el objetivo es eliminar las palabras irrelevantes y normalizar el resto. Según las necesidades del texto, se irán creando diccionarios que se aplicarán directamente al texto parseado, es decir, a la palabra.
    - Drop List: este diccionario recoge aquellas palabras que carecen de "significado" para posteriormente eliminarlas: preposiciones, artículos, adverbios, interjecciones, signos de puntuación, verbos auxiliares…
    - Errores ortográficos: Al tratase de texto libre siempre existe la posibilidad de que haya errores ortográficos en el texto o que se haya alterado el orden de las letras. Este diccionario estará compuesto por la palabra errónea y su corrección.

[1] Blei, David M.; Ng, Andrew Y.; Jordan, Michael I (January 2003). Lafferty, John (ed.). "Latent Dirichlet Allocation". Journal of Machine Learning Research. 3 (4–5).

- Lexematización: Para reducir la variabilidad de términos, se procederá a cambiar las desinencias verbales y las desinencias sustantivas. Para el caso de los verbos, todas las formas personales y las no personales se transforman a infinitivo.

- Sinónimos: Para aquellas palabras que comparten el mismo significado se utilizará solo una de ellas, unificándolas así.

Una vez con el texto limpio y estructurado, se identificarán las temáticas de las que se hablan. Para ello se utilizará el modelo Latent Dirichlet Allocation (LDA)[1]. En 2003 David Blei, Andrew Ng y Michael Jordan publicaron un artículo llamado "Latent Dirichlet Allocation (LDA)", en el cual se describía un novedoso y revolucionario modelo capaz de detectar las temáticas de un documento a través del aprendizaje automático.

> LDA es un modelo bayesiano jerárquico de tres niveles, en el que cada elemento de una colección se modela como una mezcla finita sobre un conjunto subyacente de temas. Cada tema, a su vez, se modela como una mezcla infinita sobre un conjunto subyacente de probabilidades de tema. En el contexto del modelado de texto, las probabilidades de tema proporcionan una representación explícita de un documento. (Blei, David M.; Ng, Andrew Y.; Jordan, Michael I, enero 2003).

La idea simple que subyace detrás del LDA es que cada documento se puede describir mediante un conjunto de temas y cada tema se puede describir mediante un conjunto de palabras.

En este trabajo, para dar respuesta a uno de los objetivos secundarios que se han planteado, se compararán varios modelos LDA con diferente número de temas y cada uno de esos temas compuesto por un conjunto de palabras.

Por último, en este apartado también se desarrollará otro de los objetivos expuestos, que será el análisis de la edad de las víctimas. Para deducir del texto una variable numérica, la técnica de extracción estará basada en expresiones regulares. Las expresiones regulares, también conocidas como "regex", son una secuencia de caracteres que conforma un patrón de búsqueda.

Se enlaza una guía de referencia para elaborar expresiones regulares:

https://cheatography.com/davechild/cheat-sheets/regular-expressions/

4.1.4. Análisis de las redes

El análisis de redes sociales parte de la utilización de grafos para representar las redes y analizarlas, por lo que es muy importante tener cierto conocimiento sobre la teoría de grafos. Un grafo es un conjunto no vacío de objetos llamados vértices, o nodos,

conectados a través de aristas, que pueden ser dirigidas o no. Este concepto es esencial para poder representar las redes que se estudiarán, donde los nodos serán los tuiteros y las aristas representarán las interacciones que han tenido entre ellos. La dirección de la arista representará el sentido en el que fluye el traspaso de información.

El análisis se centrará en hallar las características tipográficas de las redes, identificar a los líderes de opinión y detectar comunidades a partir de diferentes algoritmos.

A continuación, se explicarán los distintos tipos de redes y algunos modelos de detección de comunidades que se utilizarán en el desarrollo del trabajo.

Tipología de las redes sociales:

La tipología de las redes se puede considerar como la forma en la que están distribuidos los nodos y sus relaciones en la red. Su estudio es muy importante ya que permite identificar y explicar cómo se han ido generando las redes e inferir cómo van a evolucionar. Existen principalmente tres tipos de redes: aleatorias, libres de escala y de pequeño mundo.

Red aleatoria:

Una red aleatoria es aquella en la que sus nodos están conectados de forma aleatoria entre sí. El inicio de la teoría sobre las redes aleatorias se debe a los matemáticos húngaros Erdös y Rényi, quienes desarrollaron un modelo para generar redes aleatorias[2]. En este modelo se propone que un nuevo nodo se enlaza con igual probabilidad con los otros nodos, es decir, que posee una independencia estadística con el resto de nodos de la red. Además, se demostró que las aristas se generan siguiendo una distribución de probabilidad binomial, ya que la probabilidad de que un nodo de una red aleatoria tenga exactamente L enlaces es:

$$pL = \binom{\frac{N(N-1)}{2}}{L} p^L (1-p)^{\frac{N(N-1)}{2}-L}$$

Siendo:

- $\binom{\frac{N(N-1)}{2}}{L}$ el número de posibilidades de combinar los N vértices de L en L
- $p^L(1-p)^{\frac{N(N-1)}{2}-L}$ la probabilidad de que el vértice v esté conectado a L vértices concretos y no a los restantes

[2]Erdős, P.; Rényi, A. (1959). "On Random Graphs. I.".

Red libre de escala:

Estas redes se caracterizan por tener unos pocos nodos de la red con alto número de conexiones mientras que resto de nodos tienen pocos enlaces. Se descubrieron hace relativamente poco, cuando en 1998 Albert-László Barabási y otros investigadores empezaron a mapear las páginas dentro de la World Wide Web (WWW), con la idea de que encontrarían un patrón de comportamiento similar al modelo de Erdös y Rényi, pero se dieron cuenta de que unas pocas páginas web estaban mucho más conectadas que el resto. Además, demostraron como la distribución de los enlaces seguía una distribución potencial.

El modelo de Barabási–Albert[3] es un algoritmo empleado para generar redes libres de escala, también llamadas de conexión preferencial, ya que los nuevos nodos tienen preferencia en conectarse con los hubs (nodos de alto grado). Por este motivo, los nodos con alto grado tienden a acumular aún más conexiones, en cambio, los que tienen pocos, rara vez las adquieren. Concretamente la probabilidad de que un nodo i se conecte a un nodo j es:

$$p_i = \frac{k_i}{\sum_j k_j}$$

Siendo:

- k_i el grado del nodo i
- k_j el grado del nodo j

Red de pequeño mundo:

En este tipo de redes la mayoría de los nodos pueden ser alcanzados desde cualquier nodo origen a través de un número relativamente corto de saltos entre ellos. Este fenómeno también es conocido como Teoría de los Seis Grados de Separación, por el experimento realizado por Stanley Milgram con el que se demostró que, si se consideran dos personas aleatorias en cualquier parte del mundo o entorno, existiría la posibilidad de que, a través de seis personas se pueda establecer una cadena que conecte a esas dos personas entre sí.

[3] Albert-László Barabási & Réka Albert (octubre de 1999). «Emergence of scaling in random networks»

Los matemáticos Duncan Watts y Steven Strogatz desarrollaron un modelo para generar redes de pequeño mundo, conocido como modelo de Watts- Strogatz[4], en el que se establece una red inicial unidimensional con N nodos. Estos nodos se pueden disponer en forma de anillo de tal forma que cada uno de los vértices (o nodos) se una con 2k vecinos. La probabilidad de conectar un nodo con otro cualquiera es de *p*. Para un grafo con p=0 se puede ver que la conectividad es la misma y de valor 2k. Por otro lado, un valor no nulo de p introduce desorden en la red de tal forma que la conectividad no es uniforme, manteniendo todavía de media un valor de 2k.[5]

Algoritmos de detección de comunidades:

Las comunidades dentro una red se pueden considerar como conjunto de nodos que están más densamente conectados entre ellos que con el resto de nodos de la red. Debido a limitaciones de hardware, algunos algoritmos más complejos como por ejemplo el de Girvan-Newman, no han podido utilizarse.

Algoritmo de Louvain:

Este algoritmo está basado en la optimización de la modularidad de los datos. La modularidad compara la densidad de aristas dentro y fuera de una comunidad, de manera que una alta modularidad quiere decir que existen sólidas conexiones entre los nodos de una misma comunidad, pero escasas conexiones entre nodos de diferentes comunidades.

$$Q = \frac{1}{2m}\sum_{ij}\left[A_{ij} - \frac{k_i k_j}{2m}\right]\delta(c_i, c_j)$$

Siendo:

- A_{ij} la matriz de adyacencia.
- k_i el grado del nodo i.
- $2m$ la suma de los grados de la red.
- c_i la comunidad del nodo i.
- δ la función de delta.

En el algoritmo de Louvain, para detectar las comunidades, primero se encuentran comunidades pequeñas optimizando la modularidad localmente para todos los nodos, luego, cada comunidad pequeña se asocia a un nodo y se repite el proceso hasta alcanzar la convergencia. Es muy similar a su antecesor, el algoritmo de Clauset, Newman y Moore, que conecta las comunidades cuya fusión produce el mayor aumento de la modularidad[6]

---

[4] Watts, D.J.; Strogatz, S.H. (1998). «Collective dynamics of 'small-world' networks.». *Nature* **393** (6684)

[5] https://es.wikipedia.org/wiki/Modelo_Watts_y_Strogatz

[6] https://es.wikipedia.org/wiki/M%C3%A9todo_de_Louvain

Algoritmo Infomap:

El algoritmo fue creado por M. Rosvall y C. T. Bergstrom y se basa en los principios de la teoría de la información. Infomap caracteriza el problema de encontrar la agrupación óptima en una red como el problema de encontrar una descripción de la información mínima de un camino aleatorio en el grafo. El algoritmo encuentra la estructura de la comunidad que minimiza la longitud de descripción esperada de una trayectoria aleatoria[7].

4.2. Software

Tanto en la ejecución del proyecto como en la visualización de las redes se ha utilizado software libre, con la idea de que pueda ser replicado por cualquier persona en esta u otras redes.

4.2.1 R

R es un programa de procesamiento y análisis estadístico de datos que fue desarrollado por Robert Gentleman y Ross Ihaka en 1993. Entre sus características destacan:

- Es un programa avalado por una sólida comunidad científica que provee excelente documentación.
- Es multiplataforma, hay versiones para Linux, Windows, Mac, iPhone…
- Tiene capacidades avanzadas de gráficos.
- Es compatible con muchos formatos de datos (.csv, .xls, .sas…). Además de poderse conectar directamente a bases de datos como Oracle

Se utilizará concretamente RStudio, un entorno de desarrollo integrado para el lenguaje de programación R, para realizar todos los análisis que se llevan a cabo en este trabajo.

4.2.2 Gephi

Gephi es una plataforma interactiva para la visualización y exploración de todo tipo de redes y sistemas complejos con gráficos dinámicos y jerárquicos. Aunque en este trabajo solo se ha utilizado para la representación de los diferentes grafos sociales creados a través de R, este programa tiene diversas funcionalidades como:

- Cálculo de medidas de centralidad
- Detección de comunidades
- Filtros dinámicos
- Determinar el número de compontes conexas.

[7] http://castor.det.uvigo.es:8080/xmlui/bitstream/handle/123456789/115/TFG%20Yamila%20Bouhachmir%20Gonzalez.pdf?sequence=1&isAllow

5. DESARROLLO DEL TRABAJO

5.1. DEPURACIÓN Y ANÁLISIS DESCRIPTIVO

La base de datos que se va a analizar en este estudio contiene todos los tuits y sus relaciones (retuits, respuestas y citas) recogidos entre el 26 y el 29 de abril del 2018 con el hashtag #Cuéntalo. En esos cuatro días se publicaron 477360 tuits, en los que participaron 162180 usuarios distintos.

El data set está compuesto por las siguientes variables, aunque no todas se usarán en el estudio:

Variables	Descripción
id tweet	Identificador
date	Fecha de creación tuit
author	Autor de tuit
text	Texto de tuit
App	Tipo de dispositivo: Android, iPhone, Web Client
id user	Identificación del autor de tuit
followers	N.º seguidores del autor de tuit
following	N.º de usuarios a los que sigue el autor
statuses	Estados
location	Localización
urls	Enlaces que tiene el tuit
geolocation	Geolocalización
name	Nombre del autor de tuit
description	Descripción del autor
url_media	Enlaces que no sean páginas web
type_media	Tipo de archivo multimedia
relation	Tipo de relación con otros tuits
replied_id	Identificación usuario al que ha respondido
user_replied	Usuario al que ha respondido
retweeted_id	Identificación del usuario al que se ha retuiteado
user retweeter	Usuario al que se ha retuiteado
quoted_id	Identificación usuario al que se ha citado
user_quoted	Usuario al que se ha citado

firstHT	Primer hashtag
lang	Idioma en el que está el tuit
link	Link del perfil del autor

TABLA 1: LISTADO DE VARIABLES

A continuación, se hará un análisis descriptivo de algunas variables con el objetivo de tener una primera visión de los datos, depurar el data set y ver el alcance que ha tenido el hashtag #Cuéntalo.

- **DATE**
  Representa la fecha de creación del tweet.

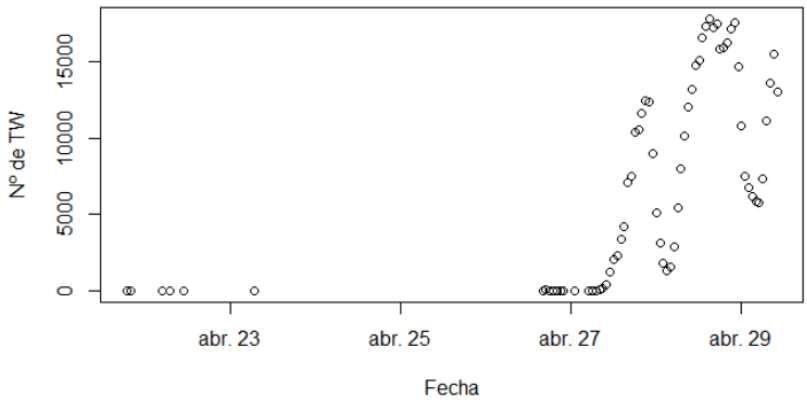

GRÁFICO 4: DISTRIBUCIÓN VARIABLE DATE

Hay observaciones que se crearon antes del 26 de abril (fecha de inicio de #Cuéntalo), por lo que se eliminarán. Esto da un indicativo de que puede que haya tuits con ese hashtag pero que no tengan que ver con la temática que se quiere analizar.

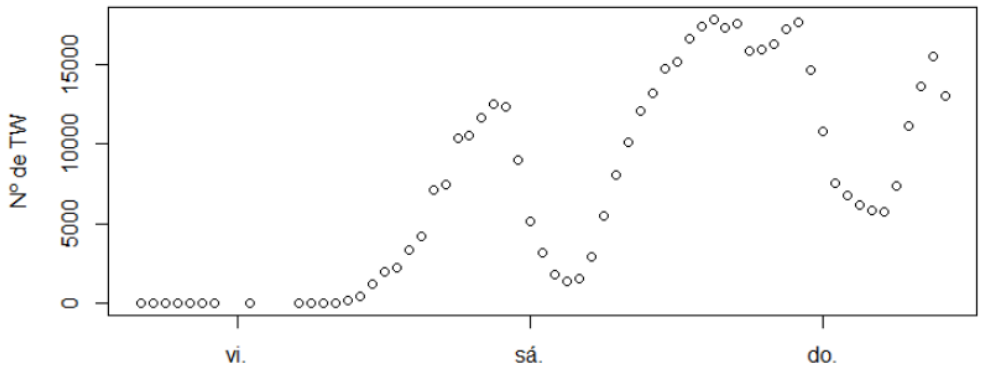

GRÁFICO 5: DISTRIBUCIÓN VARIABLE DATE DEPURADO

En general, y al ser un tema centrado en España esencialmente, la distribución de tuits a lo largo del tiempo tiene una mayor predominancia en las tardes que en las mañanas, formándose una distribución con forma de normal en el pico de la tarde, que es cuando más se tuitea. Se observa que, de los tres días, el de mayor intensidad fue el sábado 28.

- **LANG**

Representa el idioma en el que se ha escrito el tuit. Esta variable cuenta con 29 categorías, como algunas de ellas apenas tienen representación se ha categorizado la variable obteniendo el siguiente resultado:

Idioma	Frecuencia
Español	457.377
Catalán	12.283
Sin texto: urls, #, @	5.492
Inglés	1.008
Otros	507
Portugués	302
Francés	246
Euskera	137

TABLA 2: IDIOMA DE LOS TUITS

El idioma más usado con gran diferencia respecto al resto ha sido el español. También se han escrito tuits en otras lenguas oficiales de España como el catalán y el euskera.

- **LOCATION**

Representa la localización del autor del tuit. Esta localización no la genera el algoritmo de Twitter, sino que es el propio usuario quién la escribe, por lo que los usuarios se pueden referir a la misma ubicación escribiéndola de forma diferente, por ejemplo, “Madrid”, “Madrid, Comunidad de Madrid”, “Madrid, España”, “Madrid, Spain”, “Comunidad de Madrid, España”. Además, en muchas ocasiones los usuarios ponen ubicaciones generalistas: “en mi casa”. Este es uno de los motivos por los que hay más de 1500 categorías distintas y por el que se ha decidido no normalizar la variable, pero a grandes rasgos se puede observar como la mayoría de tuits proceden de España (principalmente de grandes comunidades como Madrid y

Cataluña) seguido de países de Latinoamérica donde se comparte el idioma (Argentina, Colombia, Perú, México).

Localización	Frecuencia	Localización	Frecuencia
España	8.900	Valencia	1.130
Madrid	8.257	Madrid, Spain	1.092
Barcelona	6.986	Barcelona, Cataluña	1.075
Madrid, Comunidad de Madrid	4.943	Andalucía, España	987
Barcelona, España	3.606	Colombia	971
Buenos Aires, Argentina	2.549	República de Catalunya	952
Catalunya	2.546	Zaragoza, España	929
Argentina	2.534	Granada, España	927
Barcelona, Catalunya	2.467	Lima, Peru	919
Madrid, España	2.273	México	917
Valencia, España	1.786	Comunidad de Madrid, España	891
Sevilla, España	1.785	Chile	889
Málaga, España	1.685	República Catalana	880
Sevilla	1.672	Málaga	817
Spain	1.510	Venezuela	816

TABLA 3: LOCALIZACIÓN DE LOS TUITS

Los datos de la localización concuerdan con el idioma, donde el más usado era el español.

- **RELATION**

Representa el tipo de relación del tuit analizado con otro tuit: si es un retuit (RT), una respuesta, una cita o si por el contrario no tenía relación con ningún otro tuit, en estos casos los llamaremos tuits originales. En esta variable se han hallado varios problemas: había 48 observaciones sin clasificar y se clasificaron manualmente. El otro problema fue que como cada tuit está limitado a 280 caracteres, algunos usuarios para continuar con su historia se respondían a sí mismos, por lo que el algoritmo de Twitter los contabilizaba como si fuese una respuesta, en estos casos se ha cambiado la relación de respuesta a tuit original.

A partir de esta variable se han creado otras 4 para facilitar el manejo de la base de datos:

Retuit	Tuit_original	Reply	Quote
424.930	38.507	11.117	2.798

TABLA 4: FRECUENCIA TIPO DE TUIT

La mayoría de la actividad generada ha sido mediante retuits.

La siguiente tabla representa estadísticos relacionados con la actividad de cada tuitero.

Retuit					
Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0,00	1,00	1,00	2,62	2,00	540,00
Tuit_originales					
Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0,00	0,00	0,00	0,24	0,00	108,00
Reply					
Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0,00	0,00	0,00	0,07	0,00	74,00
Quote					
Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0,00	0,00	0,00	0,02	0,00	235,00

TABLA 5: ESTADÍSTICOS DESCRIPTIVOS POR CADA CATEGORÍA DE LA VARIABLE RELATION

Más del 75% de los tuiteros que han participado en el hashtag cuéntalo han retuiteado a, al menos una vez, otro usuario. En cambio, en el resto de categorías la actividad se ha concentrado en menos del 25% de los tuiteros.

- **TEXT (detección hashtag más usados)**

Recoge el contenido del Tweet. Aunque posteriormente se haga un análisis más exhaustivo del texto, se analizaran los hashtags más usados para tener una primera visión de lo que tratan.

Se eliminan las tildes y se convierten todas las letras a minúscula para normalizar las palabras.

Hashtag normalizado	Frecuencia
#cuentalo	480.322
#lamanada	14.211
#noesno	11.079
#yositecreo	5.512
#notallmen	2.803
#taxi	2.756
#felizsabado	2.542
#lamanadasomostodas	2.448
#nagorelafagge	1.900
#noesabusoesviolacion	1.797

TABLA 6: HASHTAGS MÁS USADO

Como era de esperar, #cuentalo es el más usado. Además, muchos de ellos son hashtags a favor del movimiento, como #noesno, #yositecreo, #lamanadasomostodas. Aunque también se observa el hashtag #notallmen que, por lo general, es opuesto al movimiento.

Después de este pequeño análisis descriptivo de algunas variables se puede concluir que la mayoría de la participación en este hashtag ha sido a través de los retuits, con usuarios de habla hispana y no necesariamente con una misma opinión al respecto.

5.2. ANÁLISIS DE TEXTO

En este apartado se realizará un análisis del texto de los tuits para poder dar respuesta a los objetivos secundarios planteados referentes la identificación de palabras clave, detección de temáticas, y finalmente identificación de la edad que tenían las víctimas cuando sufrieron las agresiones en los casos que se cuentan en la red.

En este apartado no se tendrán en cuanta los retuits y se descartarán los tuits que no se hayan escrito en español. Siguiendo estos criterios, se cuenta con 47.463 tuits.

Para llevar a cabo este análisis, el mismo se dividirá en tres apartados: depuración del texto, identificación de temáticas y detección de la edad de las víctimas.

5.2.1 Depuración del texto

Lo primero de todo será depurar el texto eliminando urls, palabras sin significado y normalizando palabras siguiendo los pasos que se han detallado en la metodología.

- **Limpieza del texto:** Se eliminan las menciones, el hashtag cuéntalo, urls, números, signos de puntuación y espacios en blanco inadecuados. También se convierten todas las letras a minúscula y se quitan las tildes de las letras (á,é,í,ó,ú).

Tuit Original	Periodistas, escritoras, políticas y tuiteras convierten #Cuéntalo en te ndencia mundial https://t.co/b3WCCzh0q9 https://t.co/b3WCCzh0q9
Tuit Limpio	Periodistas escritoras politicas y tuiteras convierten en tendencia mu ndial

Tuit Original	@LaFallaras @AnaGarridoRamos #Cuéntalo , con 18 un hombre me p ersiguió a casa haciéndose una paja bajo sus pantalones. X suerte corr í + k él y encontré vecino n portal
Tuit Limpio	con un hombre me persiguio a casa haciendose una paja bajo sus pan talones X suerte corri k el y encontre vecino n portal

TABLA 7: EJEMPLO DEPURACIÓN DEL TEXTO

- **Parsing:** se descomponen las frases en palabras.

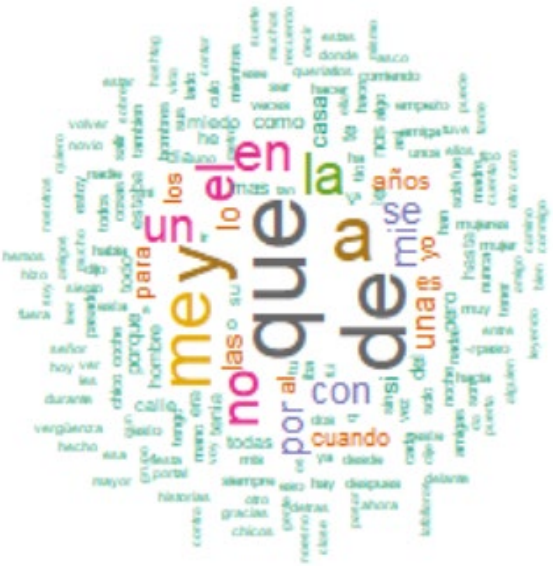

GRÁFICO 6: NUBE DE PALABRAS ORIGINAL

Este gráfico muestra las palabras que más se repiten, que como se puede observar son preposiciones, artículos, adverbios, etc. Son palabras que no aportan significado. A continuación, se crearán diccionarios para eliminar, corregir o normalizar ciertas palabras.

- **Diccionarios:**

  - Drop List: Se ha utilizado el diccionario de stopword en castellano de la librería tm (tidytext no tiene una lista de stopword en castellano). Aunque se han ido añadiendo más palabras en función de la necesidad del texto. Está compuesto por 536 palabras.

    Ejemplo:

Palabra
De
La
Que
El
En
Y
A
Los

TABLA 8: EJEMPLO DICCIONARIO DROP LIST

GRÁFICO 7: NUBE DE PALABRAS SIN LAS STOPWORD

Aquellas palabras carentes de significado ya están eliminadas. Solo falta corregir y normalizar las palabras para tener el texto depurado.

- <u>Errores ortográficos, lexematización y sinónimos:</u> agruparemos los tres diccionarios en uno, ya que se aplican de la misma manera en el texto, aunque se expondrán ejemplos de cada uno para entender la diferencia entre ellos. Estos diccionarios se han ido creando según la necesidad del texto que se analiza.

Palabra	Corrección	
anio	año	Errores ortográficos
hombres	hombre	
mujeres	mujer	Lexematización
iba	ir	
tocaba	tocar	
contarlo	contar	
tetas	pecho	Sinónimos

TABLA 9: EJEMPLO TABLA ERRORES, LEXEMATIZACIÓN Y SINÓNIMOS

GRÁFICO 8: NUBE DE PALABRAS TEXTO DEPURADO

Con el texto ya depurado se procederá a la detección de temáticas (topics).

5.2.2 Identificación de temáticas

Para ello se utilizará el Latent Dirichlet Allocation (LDA), que, como ya se ha descrito antes, es un modelo probabilístico no supervisado para identificar posibles temas, previamente desconocidos, en los documentos. Para detectar el número óptimo de topics se ejecutarán varios modelos LDA con diferentes números de topics. Además, se tendrá en cuenta el muestreo de Gibbs con 500 iteraciones para maximizar la calidad de los resultados.

La comparación de los diferentes modelos se realizará a través del valor de coherencia. Esta medida permite cuantificar la coherencia de cada tema, es decir, si las palabras del mismo tema tienen sentido cuando se juntan. Además, para garantizar una medida estable, se ejecutará cada modelo 15 veces.

Los términos con mucha frecuencia, es decir, que aparecen en muchos documentos, no discriminan bien entre temáticas, por lo que se limitará la matriz de entrada a aquellos términos con una frecuencia mayor que 500 y menor que 5.000.

N.º de topics	Mediana	Media
5	0,064	0,062
10	0,055	0,068
15	0,056	0,067
20	0,052	0,064

TABLA 10: ESTADÍSTICOS DE COHERENCIA MODELOS LDA 1ª ITERACIÓN

Aunque no se aprecia gran diferencia entre los modelos, se considerará el que tiene 5 topics como el mejor, ya que, aunque su media no es la más elevada, su mediana sí que es algo superior que el resto.

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
violacion	mano	coche	sola	historia
lamanada	amigo	calle	noche	abuso
noesno	meter	mirar	calle	sexual
españa	tocar	correr	llegar	valiente
violar	intentar	seguir	movil	vergüenza
yositecreo	culo	autobus	alguien	experiencia
denunciar	meter_mano	señor	llave	agresion
manada	quedar	llegar	fiesta	sufrir
sexual	pecho	portal	taxi	asco
sentencia	clase	detras	mano	vida

TABLA 11: TOPICS MODELO LDA 1ª ITERACIÓN

Los temas sobre los que se hablan son el caso de La Manada, que les han metido mano, tocado el culo o el pecho, que han corrido porque les seguía un coche o un señor, o sobre que estando solas de noche en la calle llevaban el móvil o las llaves en la mano.

El último topic es un poco ambiguo, por lo que se realizará una segunda iteración para detectar más temáticas, pero esta vez limitando la matriz de entrada a términos con una frecuencia de entre 250 y 1000.

N.º de topics	Mediana	Media
5	0,00685	0,00684
10	0,00996	0,01001
15	0,01237	0,01237
20	0,01425	0,01420

TABLA 12: ESTADÍSTICOS DE COHERENCIA MODELOS LDA 2ª ITERACIÓN

El mejor modelo es el de 20 topics.

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
historia	sola	coche	Ganas	madre
valiente	noche	calle	Llorar	padre
sociedad	llegar	seguir	España	familia
mundo	calle	correr	ganas_llorar	padres
vivir	alguien	camino	Seguro	abuso
experiencia	llave	noche	Dan	pequeña
testimonio	taxi	andar	Wwegrr	hermana
hastag	movil	caminar	graciasiniesta	recordar
triste	mano	vuelta	Llevar	niña
fuerte	llegar_casa	llegar	dan_ganas	hija

Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
noesno	vergüenza	relacion	Señor	violar
lamanada	culpa	exnovio	Calle	violacion
yositecreo	vida	pareja	Pene	violador
españa	seguir	sexo	Parque	red
sentencia	sentido	novio	Mirar	denunciar
manada	peor	sexual	Plena	manada
justicia	mia	follar	Luz	guardia
juez	狯	daño	Acerco	sociales
noesabusoesviolacion	asco	conmigo	Mayor	victima
machista	culpable	primer	masturbandose	pais

Topic 11	Topic 12	Topic 13	Topic 14	Topic 15
hablar	calle	amigo	Portal	tocar
llamar	comentario	fiesta	Puerta	culo
foto	piropo	intentar	Correr	tocar_culo
conmigo	asco	noche	Llegar	discoteca
movil	aguantar	beso	Entrar	grupo
policia	derecho	quedar	Seguir	calle
quedar	mirada	dormir	Suerte	amigo
amigo	cuerpo	cama	Llorar	fiesta

trabajo	tipo	conmigo	Ascensor	paz
numero	puto	habitacion	Detrás	pecho

Topic 16	Topic 17	Topic 18	Topic 19	Topic 20
autobus	persona	clase	Sexual	mano
mirar	mierda	compañero	Abuso	meter
metro	gente	llevar	Agresión	meter_mano
parada	forma	colegio	Sufir	agarrar
señor	parecer	niño	Acoso	intentar
gente	verdad	trabajo	agresion_sexual	culo
tren	machista	profesor	Historia	quedar
pierna	tema	pecho	Rabia	brazo
bajar	bien	falda	Asco	pecho
quedar	entender	tocar	Violación	tocar

TABLA 13: TOPICS MODELOS LDA 2ª ITERACIÓN

Aparte de los detectados en la anterior iteración, se han encontrado temáticas nuevas en las que se narran abusos dentro de la unidad familiar (topic 5), o del ámbito del trabajo/colegio (topic 18). Tener un sentimiento de vergüenza o culpa tras lo que les sucedió (topic 7). Señores que se les acercaban masturbándose o se sacaban el pene por la calle (topic 9). Situaciones dañinas que han sufrido con sus parejas o exparejas (topic 8). Comentarios, piropos o miradas que recibían las víctimas mientras iban por la calle (topic 12). Sufrir tocamientos en las discotecas o de fiesta (topic 15).

5.2.3 Detección de la edad de las víctimas

En la nube de palabras del anterior apartado se apreciaba como el término año era de los que más se repetían en el texto, esto se debe a que en numerosas ocasiones las víctimas cuentan los años que tenían cuando se produjo la agresión/abuso, hace cuantos años fue o la edad del agresor. A continuación, se extraerán los fragmentos del texto que hagan referencia a la edad que tenía la víctima para posteriormente hacer un análisis.

Esta extracción de fragmentos del texto se ha realizado sobre el texto sin limpiar, ya que el texto limpio no contiene números y hay diferencia entre los distintos tiempos verbales, por ejemplo, en algunos tuits hablan de la edad que tienen en la actualidad, no cuando les sucedió, usando el tiempo verbal presente.

Lo primero será buscar patrones que tengan en común los fragmentos que hay en el texto y borrar aquellos que no hagan referencia a lo que se quiere analizar. Por ejemplo:

"tenía [0-9]* años más" [7]

"tenía [0-9]* años menos"

"él tenía [0-9]* años"

"[a-z]* que tenía [0-9]* años"

No hacen alusión a la edad exacta de la víctima o se refieren a la edad del agresor.

Una vez borrados esos fragmentos se procede a detectar aquellos que realmente hablen de la edad de la víctima cuando ocurrieron los hechos. Los patrones detectados y usados han sido los siguientes:

Patrón	Frecuencia
tenía [0-9]* años	1.361
tenia [0-9]* años	228
#cuéntalo [0-9]* años	310
#cuentalo [0-9]* años	142
con [0-9]* años	1.753
a los [0-9]* años	403

TABLA 14: FRECUENCIA PATRONES DE TEXTO EDAD DE LAS VICTIMAS

[7] [0-9]* el asterisco significa que los valores que hay dentro del corchete se pueden repetir varias veces.

Se extrajeron los números de cada fragmento según los patrones anteriores y estos fueron los resultados:

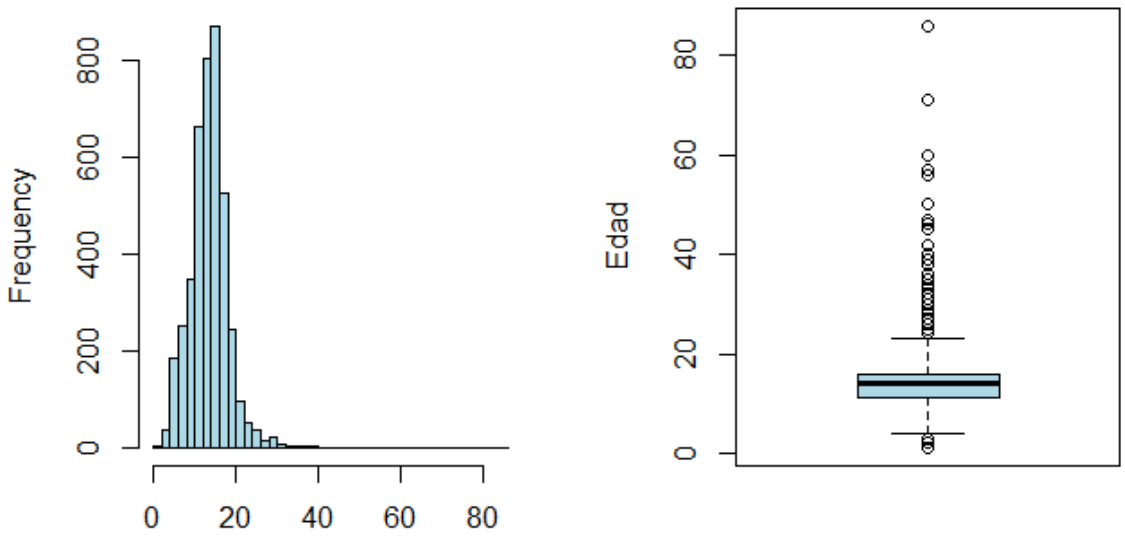

GRÁFICO 9: HISTOGRAMA Y BOX-PLOT CON LA EDAD DE LAS VICTIMAS

Lo que mas destaca es el amplio intervalo de edades que se abarca, aunque el 50% de ellas se encuentra en un rango muy estrecho de edad. Se ajustarán los ejes de los gráficos para ver con mayor detalle dónde se concetran los datos.

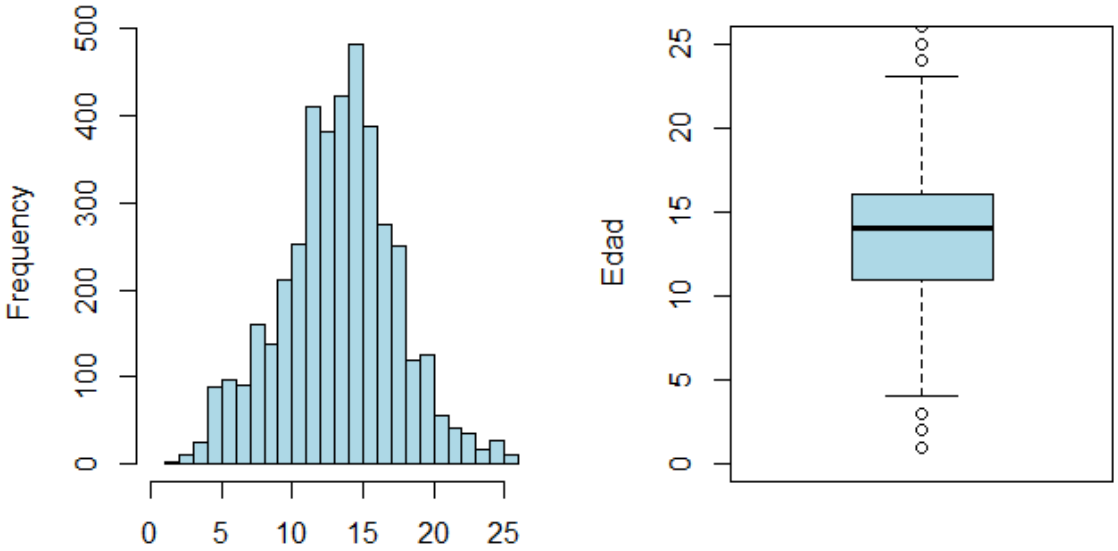

GRÁFICO 10: HISTOGRAMA Y BOX-PLOT CON LA EDAD DE LAS VÍCTIMAS. EJES AJUSTADOS

Se aprecia como la edad más frecuente son los 14 años, seguido de los 13, 11, 15 y 12 años, además, al menos el 75% de las victimas no llegaban a los 17 años. A continuación se monstrará una tabla que recoge los estadisticos asociados a la edad.

				Percentil								
Min.	Max	Mean	Var	10%	20%	30%	40%	50%	60%	70%	80%	90%
1	86	14,1	26,9	8	11	12	13	14	15	16	17	19

TABLA 15: ESTADÍSTICOS EDAD DE LAS VICTIMAS

Como ya se había comprobado antes, el rango de la edad de las víctimas es muy amplio, concretamente oscila entre 1 año y 86 años, que, aunque puedan parecer datos atípicos, son reales. Esta gran amplitud en la edad provoca una varianza muy elevada, por lo que es más conveniente fijarse en los percentiles que en la media. En torno al 20% de las víctimas eran niños (menos de 12 años) y, el 60% eran adolescentes (entre 12 y 17 años), es decir, concretamente el 80% de las víctimas era menor de edad cuando sufrió algún tipo de violencia sexual. Este porcentaje tan alto de menores de edad hace replantearse la validez de los resultados por lo que se consultaron fuentes oficiales.

Según el informe sobre delitos contra la libertad e indemnidad sexual en España[8] elaborado por el Ministerio del Interior en 2018, se afirma que casi el 50% de las víctimas de un delito sexual fueron menores de edad.

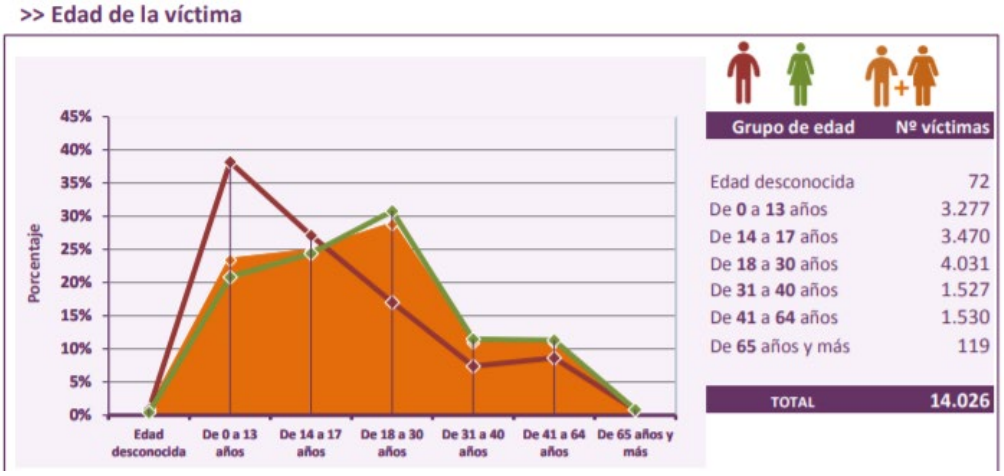

GRÁFICO 11: EDAD VICTIMAS SEGÚN EL INFORME DEL MINISTERIO DEL INTERIOR 2018

[8] En este estudio los datos han sido obtenidos del Sistema Estadístico de Criminalidad (SEC) y contemplan los siguientes delitos denunciados en España: abuso sexual, agresión sexual, agresión sexual con penetración, abuso sexual con penetración, exhibicionismo, acoso sexual, contacto tecnológico con menores 16 años, delitos relativos a la prostitución, corrupción de menores o personas con discapacidad, pornografía de menores y provocación sexual.

Aunque el análisis de los tuits también recoge información de otras regiones y diversos momentos temporales y no solo de España en 2018 como el del Ministerio, esta discrepancia en la edad de las víctimas puede deberse principalmente a dos motivos:

- La brecha digital: cuando se trata de inferir que le pasa a una población en base a lo que se opina en las redes, es evidente que existe un sesgo ya que la gente mayor no usa tanto las redes sociales como la gente joven.
- El informe del Ministerio solo contempla delitos que han sido denunciados. Y, en la macroencuesta de violencia contra la mujer de 2019[9] elaborada por el Ministerio de Igualdad, se afirma que el 35% de las mujeres encuestadas no denunciaron la violencia sexual que sufrieron porque eran unas niñas en el momento de los hechos, subiendo hasta un 40% en el caso de tratarse de una violación. Es decir, gran porcentaje de los menores que han sufrido violencia sexual no lo denunció.

Motivos para no denunciar la violencia sexual fuera de la pareja a lo largo de la vida
(N=frecuencia muestral, %=porcentaje)

	Violencia sexual		Violación	
	N	% sobre mujeres que han sufrido violencia sexual fuera de la pareja y no han denunciado ellas mismas (N=570)	N	% sobre el total de mujeres que han sufrido una violación y no han denunciado ellas mismas (N= 184 mujeres)
Tuvo muy poca importancia/no era lo suficientemente grave/no era necesario/no lo consideró violencia	174	30,5	31	16,8
Por miedo al agresor, por temor a las represalias	67	11,8	43	23,5
Por vergüenza, apuro, no quería que nadie lo supiera	148	25,9	74	40,3
Piensa/pensó que era su culpa	48	8,4	34	18,4
Temor a que no la creyeran	118	20,8	67	36,5
Por desconocimiento/no se le ocurrió/no sabía lo que la policía podía hacer	94	16,4	37	20,2
Otra persona la disuadió de denunciar	15	2,6	2	1,1
El problema se terminó	86	15,2	26	14,1
Carece/carecía de recursos económicos propios	5	0,8	5	2,6
Fue a otro lugar para obtener ayuda	9	1,5	4	2,3
Era menor, era una niña	202	35,4	74	40,2
Eran otros tiempos, otra época y no se hablaba de estas cosas	126	22,1	45	24,6
Sucedió en otro país	30	5,3	12	6,6
Otros motivos	47	8,3	14	7,4
NC	5	1,0	1	0,3

Pregunta de respuesta múltiple

GRÁFICO 12: MOTIVOS PARA NO DENUNCIAR SEGUN MACROENCUESTA MINISTERIO DE IGUALDAD 2019

Al comparar el dato obtenido mediante el análisis de los tuits con los informes expuestos, no resulta tan dudosa la validez del mismo. Por lo que se pude afirmar que el 80% de los tuits analizados, las víctimas eran menores de edad cuando sufrieron algún tipo de violencia sexual.

[9] Macroencuesta realizada a 9.568 mujeres de 16 o más años residentes en España.

5.3. ANÁLISIS DE REDES

Para poder dar respuesta al objetivo principal planteado anteriormente, en este apartado se presenta un análisis de redes sociales diferenciando aquellas relaciones que se forman al retuitear, de las que se forman al mencionar, citar o responder. Además, se abordarán objetivos secundarios tales como la identificación de comunidades dentro la red o la identificación de líderes de opinión globales y locales según su rol dentro de la red diferenciando entre emisor, intermediario o receptor.

5.3.1 CREACIÓN DE LOS GRAFOS

Tanto la red formada por los retuits como la red formada por las menciones, respuestas y citas, red de menciones, se modelarán siguiendo la teoría de grafos, donde los nodos del grafo corresponderán a los usuarios y las aristas representarán las relaciones entre ellos, además, estas aristas serán dirigidas. Para determinar la dirección de la arista se ha determinado el tuitero influyente, al que se retuitea o menciona, como el emisor de la información. Es decir, para dos nodos 1 y 2, la arista dirigida con dirección 2 hacia 1 representa que el tuitero 1 ha mencionado o retuiteado al tuitero 2. De esta manera el tuitero 2 ha influido en el tuitero 1, por lo que se considera al 2 como influyente y al 1 como receptor.

Los grafos se han modelado a través de una lista de aristas formada por la persona a la que se ha retuiteado/mencionado (influyente) y el autor del tweet (receptor).

Por ejemplo:

GRÁFICO 13: EJEMPLO TUIT-RETUITS

@YamaxanaduFTW ha recibido un retuit y ha retuiteado dos veces: a @NoPennywise y a @aandest.

Lista de aristas:

Influyente (2)	Receptor (1)
YamaxanaduFTW	nuriallibertatc
NoPennywise	YamaxanaduFTW
aandest	YamaxanaduFTW

TABLA 16: LISTA DE ARISTAS EJEMPLO ILUSTRATIVO

Grafo:

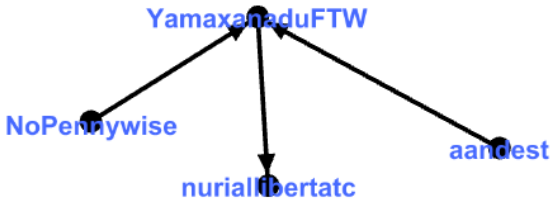

GRÁFICO 14: GRAFO RETUITS EJEMPLO ILUSTRATIVO

A continuación, se muestran las redes que se han creado, aunque para la visualización se ha limitado el número de nodos.

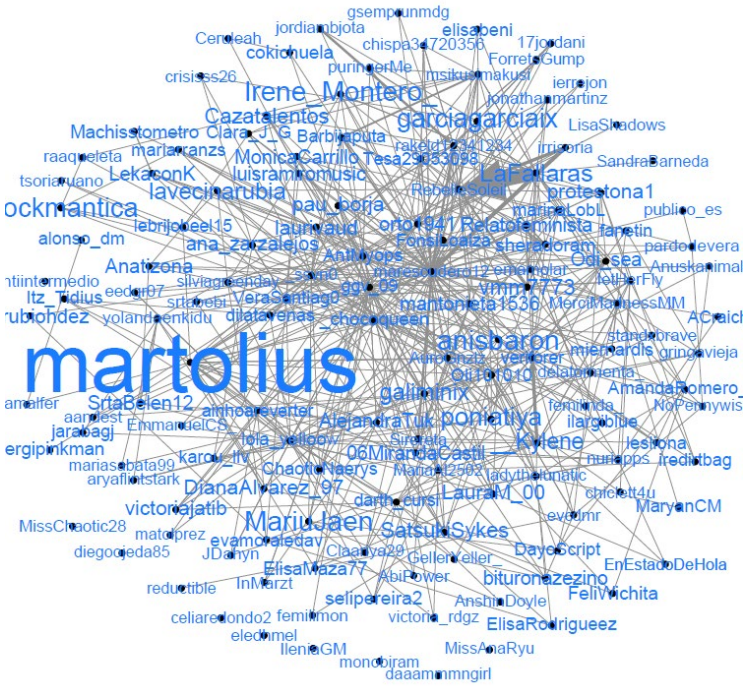

GRÁFICO 15: RED DE RETUITS CON NODOS DE MÁS DE 545 GRADOS

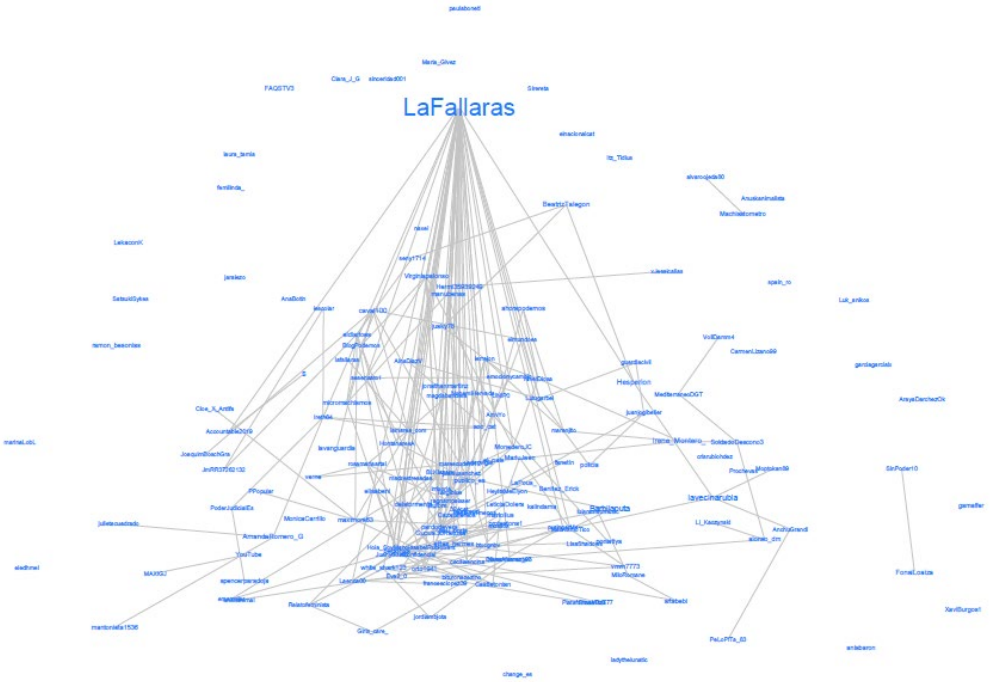

GRÁFICO 16: RED DE MENCIONES CON NODOS CON MÁS DE 7 GRADOS

5.3.2 ANÁLISIS DE LA TIPOLOGÍA DE LAS REDES

El análisis del tipo de red permite identificar y explicar cómo se han ido generando las redes hasta la fotografía que se tiene de ella actualmente, así como permite inferir como va a evolucionar. Es importante saber si las conexiones se generan de manera preferencial o si por el contrario la gente menciona o retuitea aquellos tuits con los que más se identifica.

Haciendo referencia al objetivo principal del trabajo, en este apartado se podrán establecer diferencias en la tipología de las redes cuando únicamente se consideran los retuits y cuando se consideran las menciones. Las menciones indican una relación más estrecha con el mensaje que se ha leído y normalmente fomentan el debate y la discusión, mientras que la red de retuits únicamente tiene por objeto la trasmisión de un mensaje. En este sentido el análisis que aquí se hace permite diferenciar ambos aspectos.

	Grafo Retuits	Grafos Menciones
N.º de nodos	154668	10360
N.º de aristas	424.930	16.515
Densidad	1,77E-05	1,53E-04

TABLA 17: TAMAÑO DE LAS REDES

Tal y como se esperaba por el gran volumen de retuits, el grafo que los recoge es mucho mayor que el de las menciones. Por otro lado, ambos grafos tienen una densidad muy baja, esto quiere decir que el número de enlaces que tiene cada grafo es muy inferior al de todos los enlaces que podría haber, concretamente para el de los retuits solo hay un 0,0017% de enlaces posibles y para el grafo de las menciones un 0,015%.

Para estudiar las relaciones entre los tuiteros se tendrán en cuenta las siguientes medidas de centralidad:

- **Degree:** El grado de un vértice es el número de aristas que están conectadas al vértice, es decir, el número de conexiones que tiene cada tuitero con el resto de tuiteros de la red.
- **Degree-in:** representa el número de aristas entrantes a un vértice, es decir, el número de retuits o menciones que ha realizado cada tuitero.
- **Degree-out:** es el número de aristas salientes de un vértice, es decir, el número de retuits o menciones que ha recibido cada tuitero

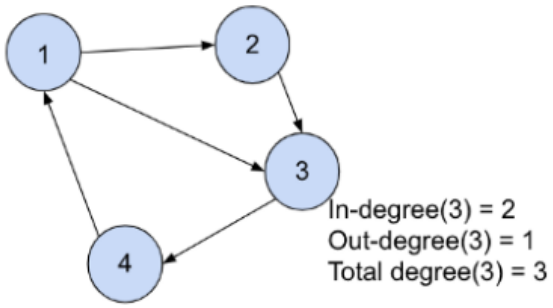

GRÁFICO 17: EJEMPLO ILUSTRATIVO MEDIDAS DE CENTRALIDAD

En cierto modo se podría decir que degree-out mide la capacidad de influencia de un usuario y degree-in la capacidad de transmisión de información.

A continuación, se muestra un histograma y gráfico de cajas y bigotes que representan la frecuencia de usuarios que hay con cada valor de las tres medidas anteriormente comentadas. Se han limitado los ejes para observar con mayor detalle los valores que tienen la mayor parte de los usuarios.

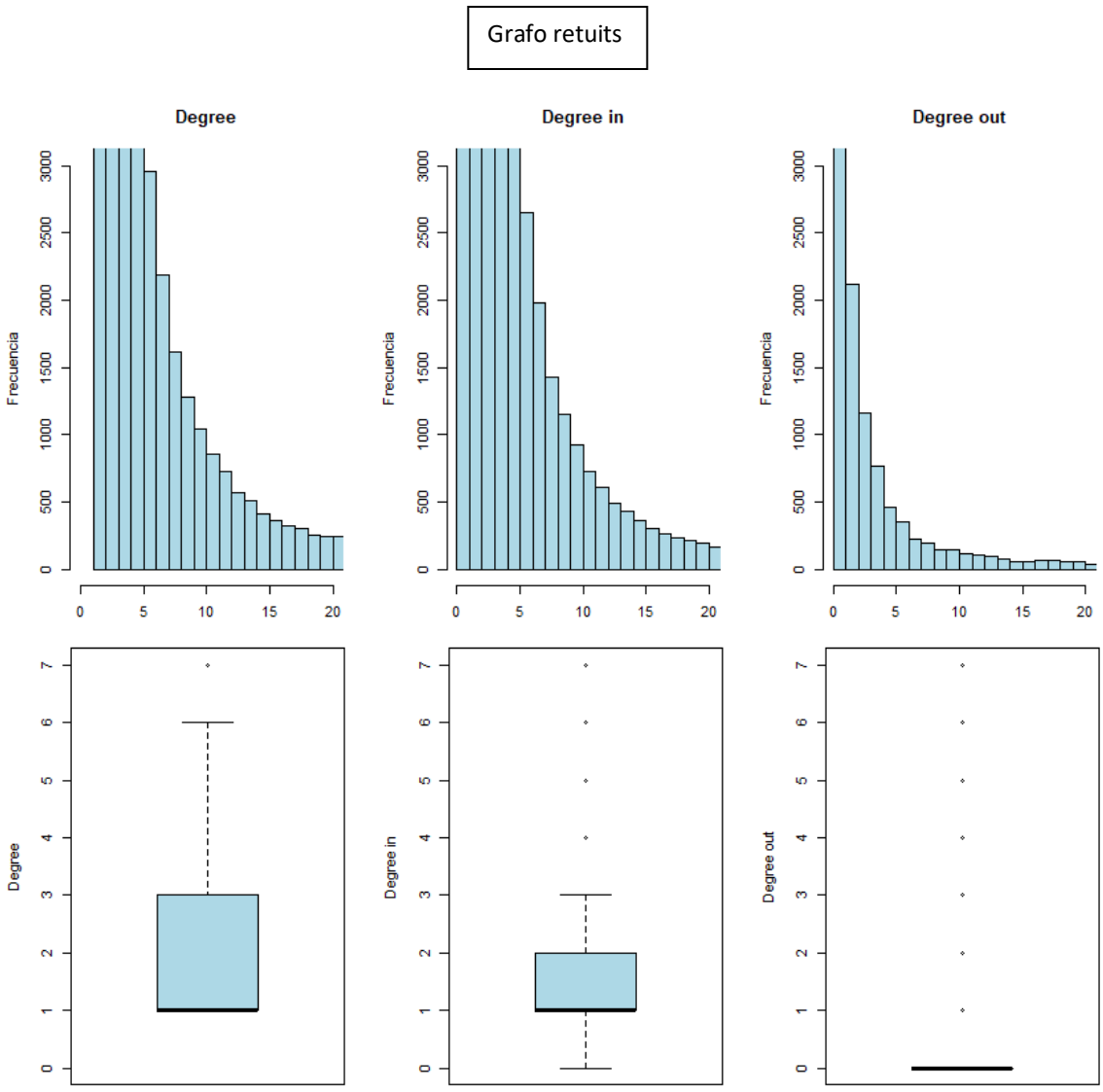

GRÁFICO 18: HISTOGRAMA Y BOX-PLOT MEDIDAS DE CENTRALIDAD GRAFO RETUITS

Lo que más destaca es que, en general, las relaciones que han tenidos los usuarios entre ellos han sido bajas, al menos el 75% de los usuarios que componen la red han retuiteado 2 o menos veces, y más del 75% de ellos no ha recibido ningún retuit. De los que sí que lo han recibido la mayor parte solo ha tenido un retuit (cerca de 2.100 usuarios).

Como era de esperar, no hay ningún usuario dentro de la red que tenga grado cero, ya que esto significaría que no ha recibido ni realizado ningún retuit. Pero, tanto para Degree out como Degree sí que hay usuarios con valores de 0, esto implica que ha habido usuarios que han retuiteado, pero no han recibido ningún retuit, esto es algo normal porque muchas personas no suelen crear contenido, sino que solo retuitean. También hay usuarios, aunque en menor proporción que los anteriores, que no han retuiteado ninguna vez, pero otros a ellos sí.

Grafo menciones

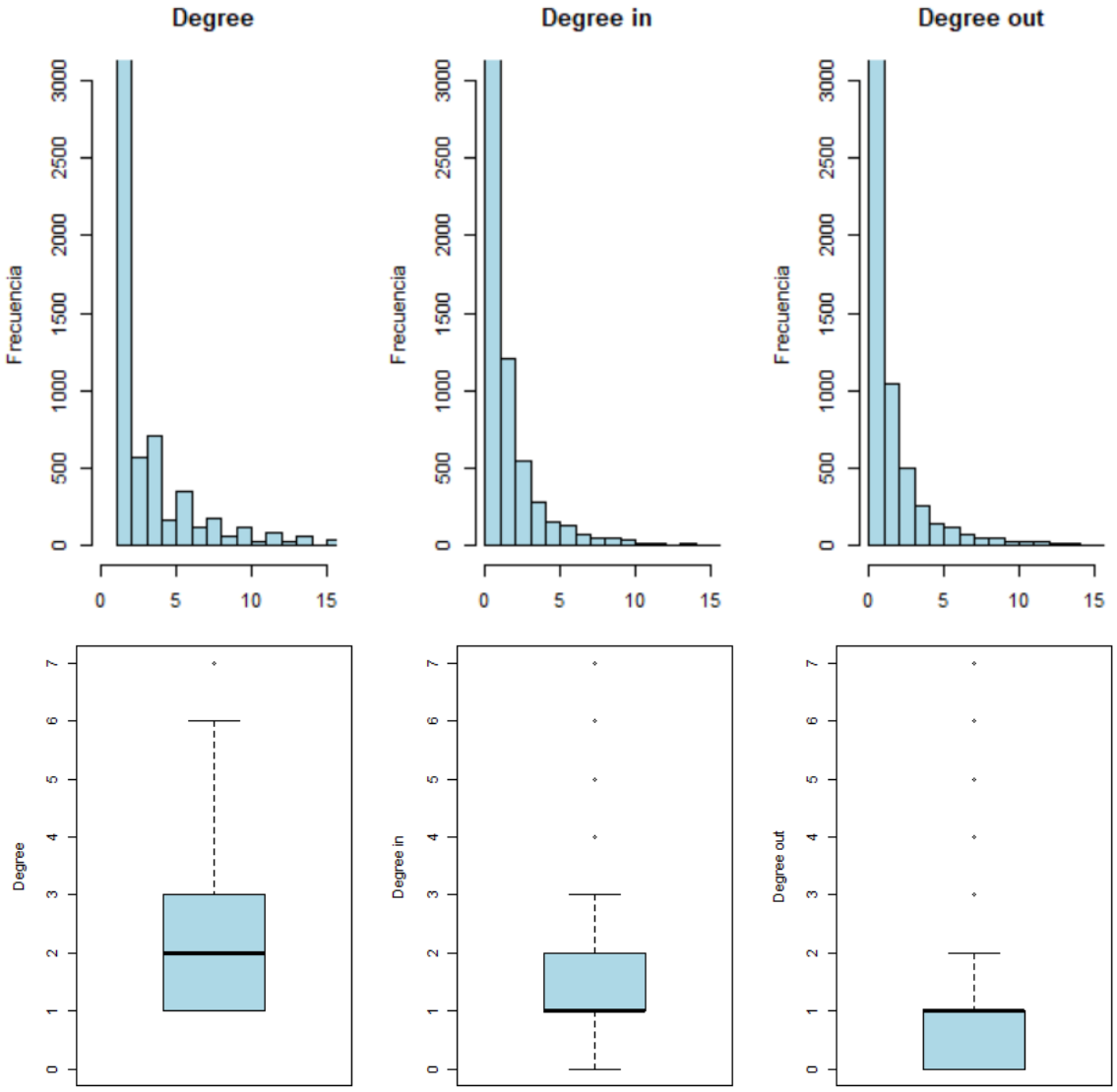

GRÁFICO 19: HISTOGRAMA Y BOX-PLOT MEDIDAS DE CENTRALIDAD GRAFO DE LAS MENCIONES

Al igual que pasaba en la red de retuit, las relaciones que han tenidos los usuarios dentro de la red de menciones han sido bajas, predominando aquellos tuiteros que solo han establecido una única relación con otro usuario. El 75% de los participantes han mencionados menos de 2 veces, pero cabe destacar que cerca del 50% de los tuiteros han recibido al menos una mención por parte de otro usuario.

En esta red también se encuentran tuiteros que no han realizado ninguna mención, pero sí que se han recibido, y viceversa.

A continuación, se muestra una tabla con los estadísticos descriptivos asociados a las medidas de estudio.

	Deegre		Deegre - in		Deegre - out	
	Grafo RTs	Grafo Menciones	Grafo RTs	Grafo Menciones	Grafo RTs	Grafo Menciones
Mínimo	1,00	1,00	0,00	0,00	0,00	0,00
Máximo	38.889,00	1164,00	540,00	238,00	38.888,00	926,00
Media	5,49	3,19	2,74	1,59	2,75	1,59
Varianza	14.911,33	165,12	46,77	14,47	14.811,17	98,25
10%	1,00	1,00	1,00	0,00	0,00	0,00
20%	1,00	1,00	1,00	0,00	0,00	0,00
30%	1,00	1,00	1,00	1,00	0,00	0,00
40%	1,00	1,00	1,00	1,00	0,00	1,00
50%	1,00	2,00	1,00	1,00	0,00	1,00
60%	2,00	2,00	1,00	1,00	0,00	1,00
70%	2,00	2,00	2,00	1,00	0,00	1,00
80%	3,00	4,00	3,00	2,00	0,00	2,00
90%	6,00	6,00	5,00	3,00	0,00	3,00

TABLA 18: ESTADÍSTICOS MEDIDAS DE CENTRALIDAD

Observando la variable deegre, la media de relaciones en ambas direcciones en el grafo de los retuits (5,49) es mayor que el de las menciones (3,19), esto era lo esperable ya que los usuarios tienden a retuitear con más facilidad que a comentar. Otra gran diferencia entre ambas redes es que el 90% de los usuarios no ha recibido ningún retuit frente al 30% de usuarios que no ha sido mencionando en ninguna ocasión, esto se debe a que la red la red de menciones es una red de debate y discusión mientras que la de retuit es de transmisión.

Por otra parte, el 90% de los usuarios dentro de la red de retuits ha retuiteado al menos 1 vez, concretamente, el 50% solo lo ha hecho una vez. En cambio, en la otra red, el 20% de los participantes no ha mencionado a ningún otro tuitero.

Tanto los retuits como las menciones se han concentrado en un grupo reducido de tuiteros, esto da un indicio de que las redes no son aleatorias, ya que, en caso de serlo, la variable grado estaría distribuida de forma igualitaria entre todos los participantes. De todas formas, a continuación, se estudiarán tres posibles redes: aleatoria, pequeño mundo y libre de escala.

5.3.2.1 Redes aleatorias: Modelo de Erdos-Renyi

Se dice que una red es aleatoria cuando las aristas se generan siguiendo una distribución binomial de parámetros n y p, siendo n el tamaño de la red y p la densidad de la misma.

Mediante un contrate de bondad de ajuste se comprobará si la variable grado se aproxima a una variable aleatoria Binomial. Se utilizará el estadístico Chi cuadrado.

Grafo de retweets

H_0: sigue una distribución Binomial (162.062, 1,62E-05)

H_1: no sigue una distribución Binomial (162.062, 1,62E-05)

```
        Chi-squared test for given probabilities

data:  tabla_RT
X-squared = Inf, df = 442, p-value < 2.2e-16
```

GRÁFICO 20: CONTRASTE BONDAD DE AJUSTE BINOMIAL, GRAFO DE RETUITS

Grafo de menciones

H_0: sigue una distribución Binomial (14.465, 7,89E-05)

H_1: no sigue una distribución Binomial (14.465, 7,89E-05)

```
        Chi-squared test for given probabilities

data:  tabla_menc
X-squared = 2.0039e+71, df = 63, p-value < 2.2e-16
```

GRÁFICO 21: CONTRASTE BONDAD DE AJUSTE BINOMIAL, GRAFO DE MENCIONES

Ninguno de los dos grafos se aproxima a una distribución Binomial. No obstante, se comprobará si se distribuye según otras variables aleatorias como la normal o la exponencial. En este caso se utilizará el estadístico de Kolmogorov Smirnov.

Contrate de normalidad:

```
data:  Degree_RT
D = 0.48532, p-value < 2.2e-16
alternative hypothesis: two-sided
```

```
data:  Degree_menciones
D = 0.43239, p-value < 2.2e-16
alternative hypothesis: two-sided
```

GRÁFICO 22: CONTRASTE BONDAD DE AJUSTE NORMALIDAD

Contraste de exponencial:

```
data:  Degree_RT
D = 0.43925, p-value < 2.2e-16
alternative hypothesis: two-sided
```

```
data:  Degree_menciones
D = 0.26923, p-value < 2.2e-16
alternative hypothesis: two-sided
```

GRÁFICO 23: CONTRASTE BONDAD DE AJUSTA EXPONENCIAL

En todas ellas se puede rechazar la hipótesis nula, por lo que se concluye que ninguna de las dos redes es aleatoria.

5.3.2.2 Redes libres de escala: Modelo Barabasi-Albert

Como ya se ha detallado en la metodología, este tipo de redes tienen una distribución de grado de tipo potencial.

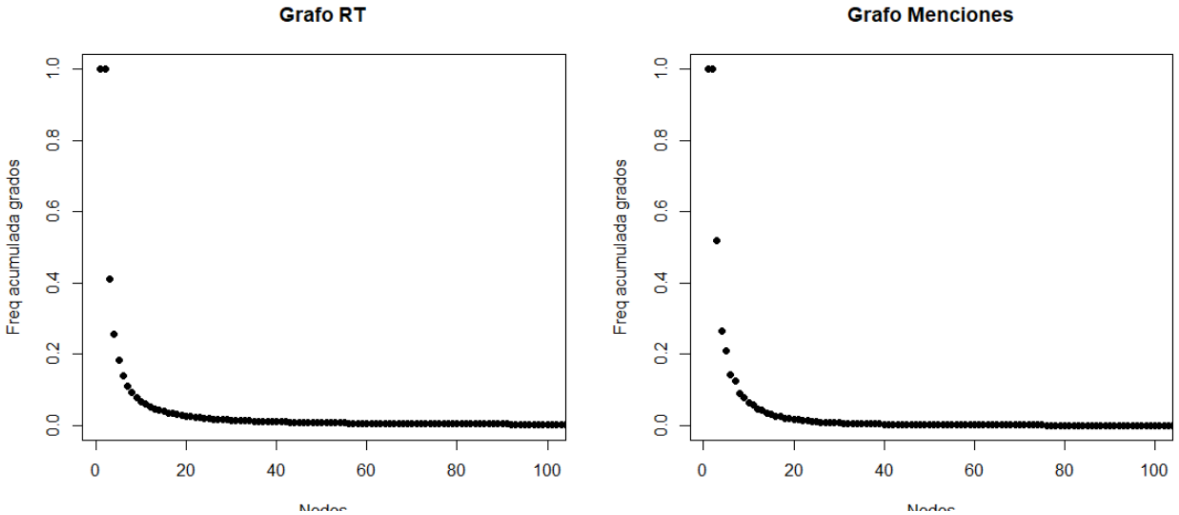

GRÁFICO 24: DISTRIBUCIÓN VARIABLE GRADO

En ambos grafos se observa como la variable grado se asemeja a una distribución potencial, donde la mayoría de los grados se concentran en un grupo reducido de nodos. No obstante, se realizará un contraste de hipótesis para comprobarlo estadísticamente:

H_0= sigue una distribución potencial

H_1= no sigue una distribución potencial

Grafo RT

```
$KS.stat
[1] 0.01573329

$KS.p
[1] 1
```

Grafo menciones

```
$KS.stat
[1] 0.06263384

$KS.p
[1] 0.9999562
```

GRÁFICO 25: CONTRASTE BONDAD DE AJUSTE POTENCIAL

En ambos grafos, no hay evidencia suficiente para rechazar la hipótesis nula por lo que la variable grado sigue una distribución potencial. Por tanto, se pueden considerar que son redes libres de escala.

5.3.2.3 Redes de pequeño mundo: Modelo de Wats-Strogaz

Son un tipo de redes en el que la mayoría de los nodos pueden ser alcanzados desde cualquier nodo origen a través de un número relativamente bajo de saltos entre ellos. Para ello es necesario que los grafos sean conexos, sin embargo, en redes del tipo social es fácil encontrar particiones no conectadas. Por eso lo primero de todo será comprobar si son grafos conexos y, en caso de no serlo, calcular el tamaño de su componente conexa gigante y, si esta es lo suficientemente grande, se podrá analizar:

	Grafo RT	Grafo Menciones
¿Es conexo?	No	No
N.º de componentes conexas	732	4.251
Tamaño componente conexa gigante	152.677	4.152
Tamaño total de la red	154.668	10.360
% tamaño componente conexa gigante	98,71%	40,08%

TABLA 19: COMPONENTES CONEXAS DE LOS GRAFOS

Ambos grafos son no conexos, pero cabe destacar que el de las menciones tiene casi 6 veces más componentes conexas que el de la red formada por los retuits aun siendo ésta última 15 veces más grande. Debido a este gran número de particiones, la componente conexa gigante del grafo de menciones tan solo representa el 40,08% de la red, mientras que la del grafo de los retuits equivale al 98,71% de los nodos de la red. Según lo anterior, se puede afirmar que el grafo de menciones no es una red de pequeño mundo y se analizará en más detalle la componente conexa gigante de los retuits.

Una red se considera de mundo pequeño cuando presenta un alto coeficiente medio de agrupamiento, es decir, que la probabilidad de que dos vecinos de un nodo sean vecinos entre sí sea alta y que, además, la longitud media del camino mínimo entre nodos sea pequeña.

La componente conexa gigante de la red analizada tiene un coeficiente medio de agrupamiento de 0,00045 y una longitud media de 6,09 de camino mínimo. Al ser una red con 152.677 nodos en la que la densidad es muy baja, podemos afirmar que tener que pasar, en media, por 6 nodos para llegar a cualquier otro nodo de la red, supone

una longitud media de camino mínimo pequeña. En cambio, para saber si el coeficiente medio de agrupamiento es lo suficientemente grande para considerar la red de pequeño mundo, se simularán 10.000 redes libres de escala del mismo tamaño que la componente conexa gigante que se está analizando, para estimar la probabilidad de que por pura aleatoriedad se obtenga una red similar a la estudiada, pero con un coeficiente medio de agrupamiento mayor.

Tras simular las 10.000 redes libres de escala, se obtuvo:

coeficiente medio agrupamiento	
Media	N.º redes con coef. superior al coef. de la red estudiada
9,07E-05	0

TABLA 20: RESULTADO AL SIMULAR 10.000 REDES LIBRES DE ESCALA

De las 10.000 redes generadas ninguna presentó un coeficiente mayor, en consecuencia, queda demostrado que la red tiene un alto coeficiente de agrupamiento medio y por ello la red de retuits se puede considerar de pequeño mundo.

5.3.3 IDENTIFICACIÓN DE LOS LÍDERES DE OPINIÓN

Este apartado consistirá en identificar aquellos usuarios que han sido más influyente en cada una de las redes. El cálculo de esta influencia se basará en las medidas de centralidad ya utilizadas anteriormente y otras dos nuevas:

- **Betweenness**: Es el poder que tiene un nodo para intermediar en las comunicaciones de otros individuos. Una alta centralidad de intermediación significa que el nodo actúa como intermediario entre otros nodos en la red.
- **Page rank**: Medida según la cual el poder de un nodo es la suma del poder de sus amigos.

Cabe destacar una medida muy importante para la detección de líderes de opinión que, debido al tipo de redes con las que se está trabajando, no se puede usar: Closeness, mide la facilidad con que un nodo puede llegar al resto de nodos de la red. Esta medida se calcularía como la suma inversa de las distancias más cortas a todos los demás nodos desde un nodo focal. Al no ser conexas las redes, los nodos nunca van a llegar al resto de los nodos de la red, por lo que no es coherente utilizarla en este caso.

Cada medida evalúa una métrica diferente y por tanto ordenarán de manera diferente, por lo que primeramente se hará un análisis de correlaciones entre las medidas para ver cuáles están midiendo un tipo de liderazgo de forma parecida.

La siguiente matriz representa, en la parte superior, el coeficiente de correlación entre las distintas medidas de centralidad y, en la parte inferior, el gráfico de dispersión asociado.

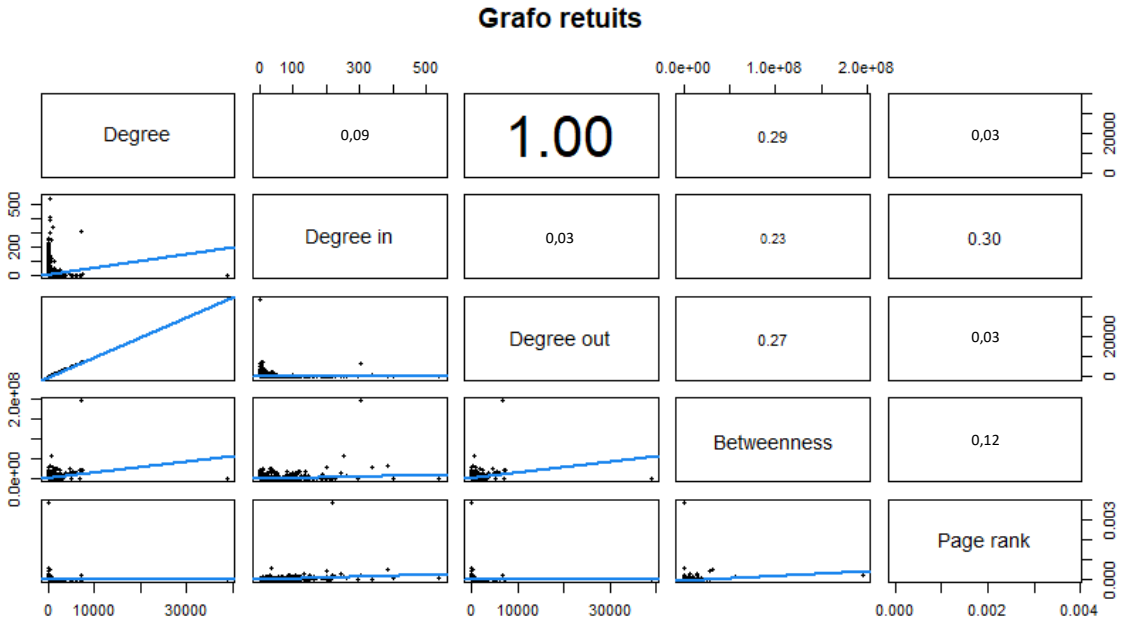

GRÁFICO 26: MATRIZ DE CORRELACIÓN Y DISPERSIÓN ENTRE LAS MEDIDAS DE CENTRALIDAD, GRAFO RETUITS

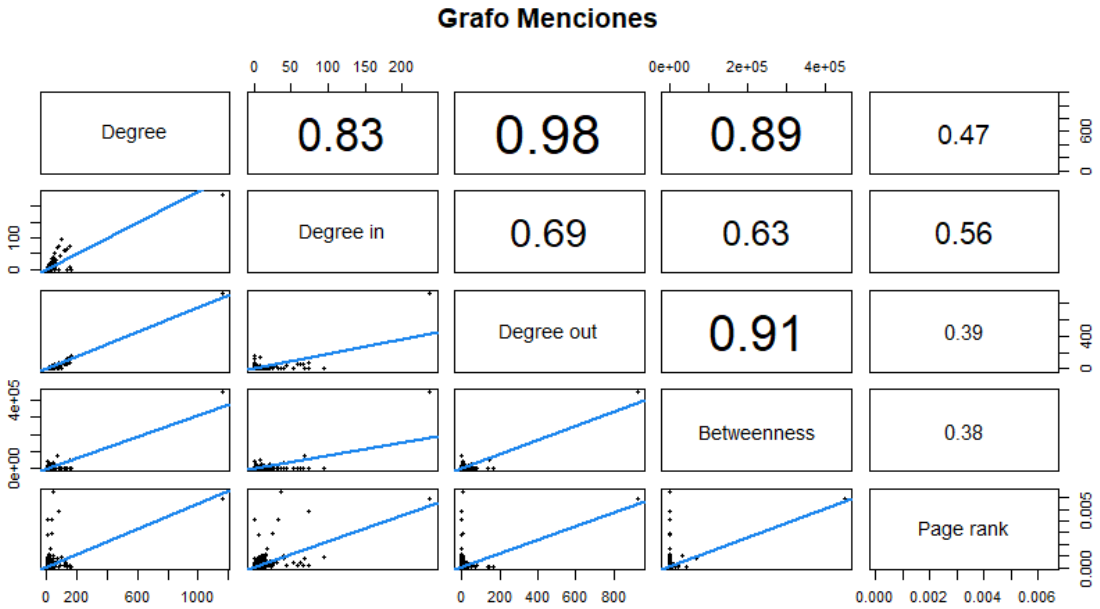

GRÁFICO 27: MATRIZ DE CORRELACIÓN Y DISPERSIÓN DE LAS MEDIDAS DE CENTRALIDAD, GRAFO MENCIONES

En el grafo de los retuits solo las medidas degree y degree out tienen una fuerte correlación ya que en el resto de medidas es prácticamente nula. Esto se debe a que en este tipo de redes existen usuarios que únicamente generan contenido sin luego ellos

retuitear a nadie, por lo que su degree out va a ser prácticamente el mismo que su Degree.

En cambio, en el grafo de las menciones, a excepción de page rank, el resto de medidas están altamente correlacionadas entre ellas, aunque la de degree in con degree y betweenness es algo más débil. Esto se debe a que, en la red de menciones, al existir conversación entre tuiteros, los que difunden la información también trasmiten y actúan de intermediarios.

A continuación, se presentará un ranking de las 5 personas más influyentes según cada medida. Los que tengan mayor degree out serán los influyentes en el sentido de emitir información, en el caso de degree in serán los más receptores/transmisores de la información y en caso de betweenness serán los que mayor poder de intermediación tengan.

Grafo retuits

Autor	Degree	Degree_in	Degree_out	Betweenness	Page_rank
martolius	38.889	1	38.888	67.700	5,89E-06
anisbaron	7.397	7	7.390	19.987.663	2,11E-05
MariuJaen	7.264	3	7.261	19.171.557	1,59E-05
Irene_Montero_	6.992	1	6.991	22.402.891	5,91E-06
Rockmantica	6.847	0	6.847	0	5,89E-06
LaFallaras	7.088	304	6.784	195.127.029	2,11E-04

TABLA 21: TOP INFLUYENTES SEGÚN DEGREE-OUT, GRAFO RT

Autor	Degree	Degree_in	Degree_out	Betweenness	Page_rank
neobabylonian7	541	540	1	2.272	9,37E-05
1denmadrid	536	536	0	0	2,64E-04
PalmiraTercera	404	402	2	3.781	1,16E-04
evallory	503	385	118	31.567.950	5,52E-04
marescudero12	1.094	339	755	28.282.410	6,09E-05
LaFallaras	7.088	304	6.784	195.127.029	2,11E-04

TABLA 22: TOP INFLUYENTES SEGÚN DEGREE-IN, GRAFO RT

Autor	Degree	Degree_in	Degree_out	Betweenness	Page_rank
LaFallaras	7.088	304	6.784	195.127.029	2,11E-04
RebelleSoleil	815	252	563	55.654.430	1,70E-04
evallory	503	385	118	31.567.950	5,52E-04
EslaLubary	209	199	10	28.995.193	4,62E-04
marescudero12	1.094	339	755	28.282.410	6,09E-05
_chocoqueen	1.897	30	1.867	25.215.778	1,24E-05

TABLA 23: TOP INFLUYENTES SEGÚN BETWEENNESS, GRAFO RT

Autor	Degree	Degree_in	Degree_out	Betweenness	Page_rank
Jose_A_Sotillo	221	220	1	0	3,87E-03
sorollmar	36	35	1	0	6,28 E-04
evallory	503	385	118	31.567.950	5,52E-04
EslaLubary	209	199	10	28.995.193	4,62E-04
mpiayza	142	134	8	7.526.586	3,41E-04
white_shark123	271	244	27	14.683.655	3,26E-04

TABLA 24: TOP INFLUYENTES SEGÚN PAGE RANK, GRAFO RT

Grafo menciones

Autor	Degree	Degree_in	Degree_out	Betweenness	Page_rank
LaFallaras	1.164	238	926	448.260	5,89E-03
s	165	0	165	0	2,80E-05
Barbijaputa	148	7	141	44.417	5,09E-05
lavecinarubia	134	0	134	0	2,80E-05
dramalesbian_	151	74	77	0	1,52E-04
Irene_Montero_	74	0	74	0	2,80E-05

TABLA 25: TOP INFLUYENTES SEGÚN DEGREE OUT, GRAFO MENCIONES

Autor	Degree	Degree_in	Degree_out	Betweenness	Page_rank
LaFallaras	1.164	238	926	448.260	5,89E-03
caval100	95	95	0	0	8,91E-04
dramalesbian_	151	74	77	0	1,52E-04
Hesperion	75	74	1	473	4,85E-03
begonys	69	68	1	69.576	7,40E-04
MujeresUC3M	132	66	66	0	1,86E-04

TABLA 26: TOP INFLUYENTES SEGÚN DEGREE IN, GRAFO MENCIONES

Autor	Degree	Degree_in	Degree_out	Betweenness	Page_rank
LaFallaras	1.164	238	926	448.260	5,89E-03
begonys	69	68	1	69.576	7,40E-04
Barbijaputa	148	7	141	44.417	5,09E-05
MadAnnMarie	4	1	3	43.583	3,34E-05
Cucua30Ratusa	46	40	6	34.898	1,01E-03
BeatrizTalegon	51	2	49	21.863	1,21E-04

TABLA 27: TOP INFLUYENTES SEGÚN BETWEENNES, GRAFO MENCIONES

Autor	Degree	Degree_in	Degree_out	Betweenness	Page_rank
Castletonian	38	35	3	0	6,53E-03
LaFallaras	1.164	238	926	448.260	5,89E-03
Hesperion	75	74	1	473	4,85E-03
srtamasa	2	1	1	0	4,15E-03
white_shark123	34	33	1	0	4,11E-03
SoldadoDescono3	28	25	3	0	2,90E-03

TABLA 28: TOP INFLUYENTES SEGÚN PAGE RANK, GRAFO MENCIONES

Según el grafo de retuits, hay bastante disparidad sobre a quién considerar líder/influyente de la red. Por ejemplo, martolius fue la usuaria que más retuits recibió (Degree-in), en cambio, el que más veces retuiteó fue neobabylonian7 (Degree-out). La que más poder de intermediación tuvo fue LaFallaras, usuario de Twitter de Cristina

Fallarás, impulsora del hashtag. Haciendo balance de las cuatro medidas se puede considerar que LaFallaras y evallory han sido las más influentes en la red de retuits.

Por otro lado, en el grafo de menciones no hay tanta disparidad sobre los influyentes, pues se aprecia como, con diferencia, LaFallaras tiene mayor poder de influencia que el resto de tuiteros.

5.3.4 DETECCIÓN DE COMUNIDADES DE TUITEROS.

En este apartado se identificarán las comunidades que se han generado dentro de cada una de las redes.
Una comunidad dentro de un grafo puede ser definida como un conjunto de nodos que están más densamente conectados entre ellos que con el resto de nodos de la red. Dependiendo del método y criterio que se utilice, se obtendrán diferentes particiones del grafo. Para detectar la mejor estructura comunitaria se tendrá en cuenta la modularidad, que mide la fuerza de la división de una red. Las redes con alta modularidad tienen sólidas conexiones entre los nodos de una misma comunidad, pero escasas conexiones entre nodos de diferentes comunidades.

Se han aplicado 3 algoritmos de detección de comunidades detallados en profundidad en la metodología.

	Grafo retuits			
	Comunidades	Comunidades en la conexa gigante	Modularidad	Modularidad en la conexa gigante
Louvain	824	95	0,507	0,505
Infomap	4549	3829	0,393	0,389
Clauset Newman & Moore	1055	357	0,509	0,509

TABLA 29: MODULARIDAD ALGORITMOS DE DETECCIÓN DE COMUNIDADES GRAFO RETUITS

	Grafo menciones			
	Comunidades	Comunidades en la conexa gigante	Modularidad	Modularidad en la conexa gigante
Louvain	4302	61	0,896	0,838
Infomap	4665	423	0,844	0,770
Clauset Newman & Moore	4311	79	0,891	0,834

TABLA 30: MODULARIDAD ALGORITMOS DE DETECCIÓN DE COMUNIDADES GRAFO MENCIONES

La modularidad en el grafo de las menciones es mayor porque, pesar de que la red de menciones es menos densa, tiene las comunidades mucho más identificadas, ya que la mención tiende a establecer comunicación y lazos más fuertes entre los miembros de la comunidad, mientras que en la red de retuits, aunque es una red modular, la tendencia es que las comunidades sean menos cohesionadas.

Cabe destacar la disparidad de los resultados dependiendo del algoritmo aplicado, por ejemplo, Infomap tiende a generar un gran número de comunidades, mientras que, Louvian y Clauset Newman & Moore generan menos, lo que provoca que tengan una mayor modularidad y, por tanto, una mejor partición. Debido a que con el modelo de Louvian se generan menos comunidades, se escoge como el mejor modelo.

Lo primero será analizar el tamaño cada comunidad, tanto dentro como fuera de la componente conexa gigante.

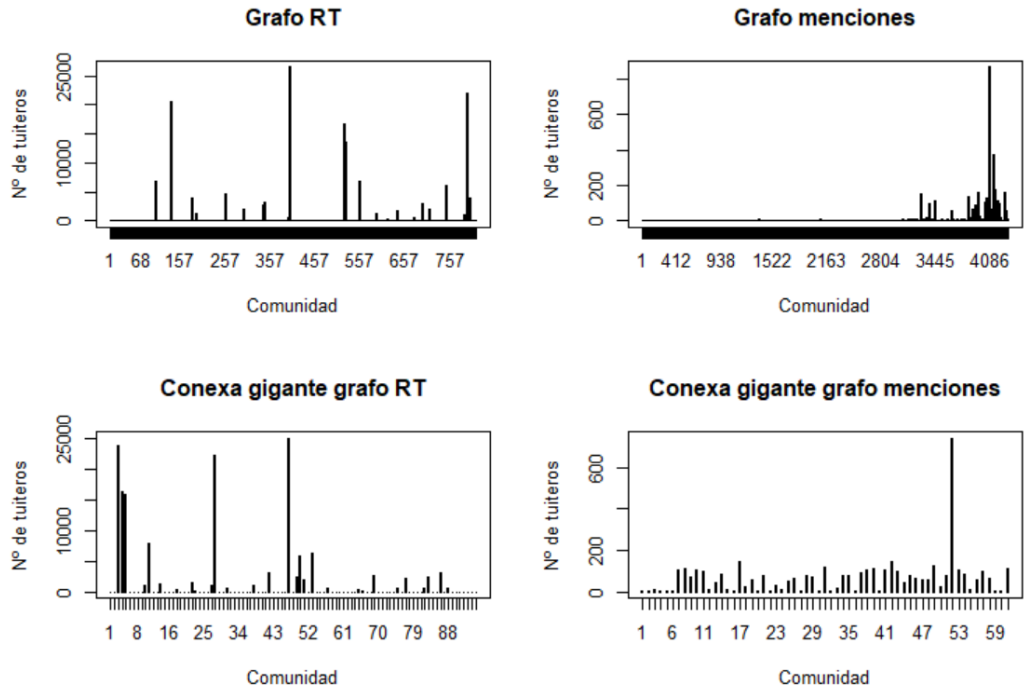

TABLA 31: TAMAÑO DE LAS COMUNIDADES

A simple vista lo que más destaca es que el grafo de los retuits tiene comunidades mucho más grandes y todas ellas se encuentran dentro de la componente conexa gigante. En cambio, en la de las menciones, la segunda comunidad más grande no está dentro de la componente conexa gigante, por lo que se utilizará la red entera para analizar las comunidades.

Grafo de retuits (componente conexa gigante):

	Tamaño comunidades
	Grafo RT
Mínimo	3
Máximo	25.096
Media	1.607
10%	3
20%	3
30%	3
40%	4
50%	5
60%	8
70%	94
80%	1.109
90%	2.915

Top comunidades	
Comunidad	**Tamaño**
47	25.096
3	23.974
28	22.400
4	16.340
5	15.999
11	7.966
53	6.234
50	5.951
86	3.108
42	3.038
69	2.730
49	2.468

TABLA 32: ESTADÍSTICOS TAMAÑO COMUNIDADES Y COMUNIDADES MÁS GRANDES GRAFO RETUITS

El 60% de las comunidades dentro de la componente conexa gigante están formadas por 8 o menos individuos, lo que implica que la mayoría de los tuiteros de red se concentre en un pequeño número de comunidades: solo 10 de ellas están compuestas por más de 3.000 tuiteros.

Grafo menciones:

	Tamaño comunidades
	Grafo Menciones
Mínimo	1
Máximo	867
Media	2
99,0%	11
99,1%	12
99,2%	13
99,3%	21
99,4%	59
99,5%	73
99,6%	91
99,7%	98
99,8%	131

Top comunidades	
Comunidad	Tamaño
4090	867
4136	370
4141	220
4158	175
3954	163
4271	162
3283	151
4151	146
3834	138
4047	126
4189	115
3449	112

TABLA 33: ESTADÍSTICOS TAMAÑO COMUNIDADES Y COMUNIDADES MÁS GRANDES GRAFO MENCIONES

En este grafo se observa como las comunidades son muchas, y muy pequeñas: tan solo en el 0.2% se superan los 100 tuiteros.

5.3.4.1 Detección de temáticas e "influencers" en las grandes comunidades

En este apartado se analizarán las comunidades de mayor tamaño de cada una de las redes, detectando posibles temáticas diferenciadoras en cada una de ellas y viendo qué lideres las componen. Para ello, se representará una nube de palabras con los términos más frecuentes y se graficará las relaciones entre las comunidades destacando aquellos tuiteros con mayor grado.

Grafo de los retuits

Cada nube de palabras representa un máximo 100 de los términos que se repiten más de 500 veces.

GRÁFICO 28: NUBE DE PALABRAS POR COMUNIDAD GRAFO DE RETUITS

No se aprecia gran diferencia temática entre cada comunidad, aunque en todas ellas se abordan muchos temas, a excepción de la comunidad 28, la tercera más grande, que está formada únicamente por los retuits del tuit de martolius:

GRÁFICO 29: TUIT ESCRITO POR MARTOLUIS

Debido a la gran dimensión de la red, solo se representarán en cada comunidad aquellos nodos con un grado mayor a 1.000.

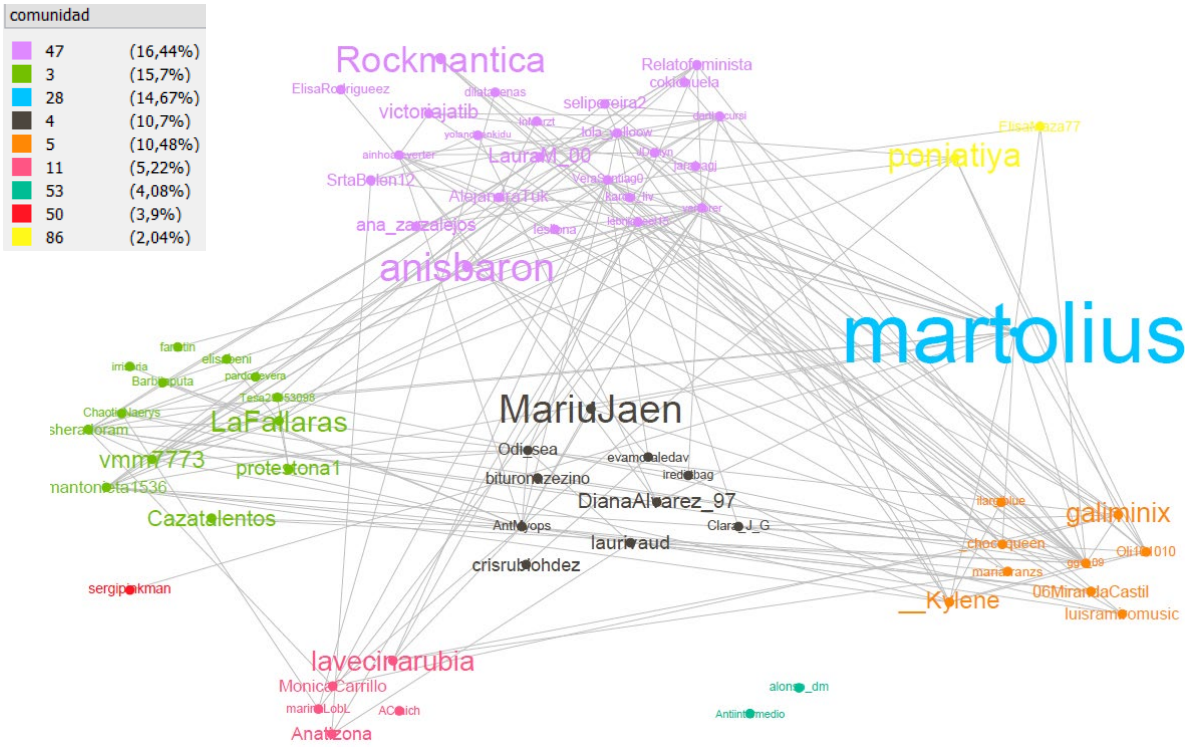

GRÁFICO 30: RED DE RETUITS CON LAS PRINCIPALES COMUNIDADES, NODOS CON GRADO MAYOR A 1.000

Se detecta como los nodos de alto grado están muy relaciones entre ellos, a este tipo de situaciones dónde usuarios con gran número de seguidores, “influencers” están densamente relacionados entre ellos se le conoce como “rich club”. Este efecto da lugar

a la teoría que dice de que, dentro de los grupos sociales, la élite tiende a asociarse entre sí, en este caso la élite serían los "influencers". Además, se observa como las comunidades de gran tamaño están formadas por varios líderes.

Cabe destacar que, en la comunidad que está formada por martolius, solo se encuentra ella como líder, esto se debe a que, como ya se vió antes, en esa comunidad solo están los retuits que ha recibido.

En el gráfico se observan algunos de los tuiteros detectados anteriormente como líderes. Por ejemplo, encontramos a LaFallaras en la comunidad 3, a Rockmantica y anisbaron en la 47, a _chocoqueen (5º usuario con mayor betweenness) en la 5 y a MariuJaen en la comunidad 4.

Grafo de las menciones

En las siguientes nubes de palabras representan todos los términos que se repiten más de 5 veces.

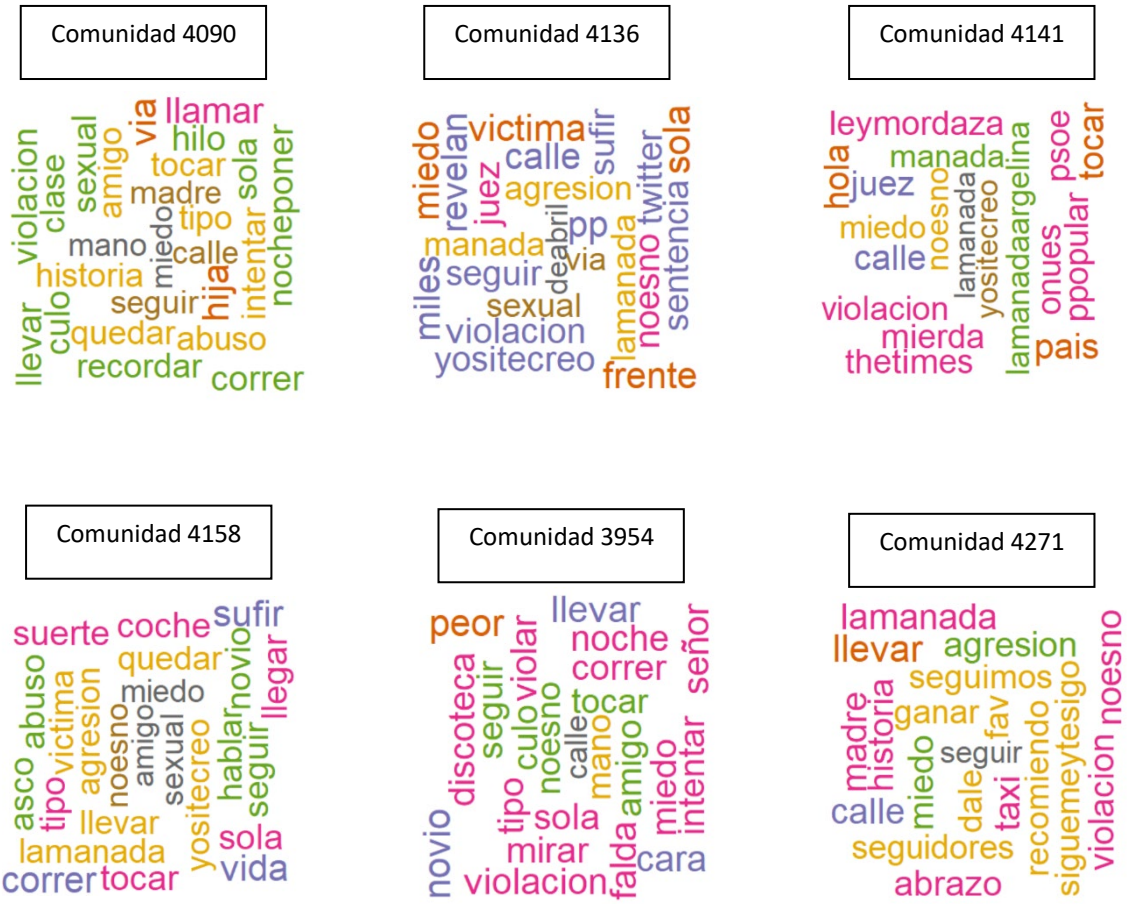

GRÁFICO 31: NUBE DE PALABRAS POR COMUNIDAD GRAFO MENCIONES

Se observa cómo, aparte de expresar sus relatos, también se hace más hincapié en el caso de La Manada, en temas políticos incluyendo “pp”, “psoe” o “leymordaza”. Y, en numerosas ocasiones, aparece la palabra “yositecreo”, símbolo de apoyo hacia otras personas que han compartido sus historias.

El siguiente gráfico representa las relaciones entre los componentes de las comunidades más grandes.

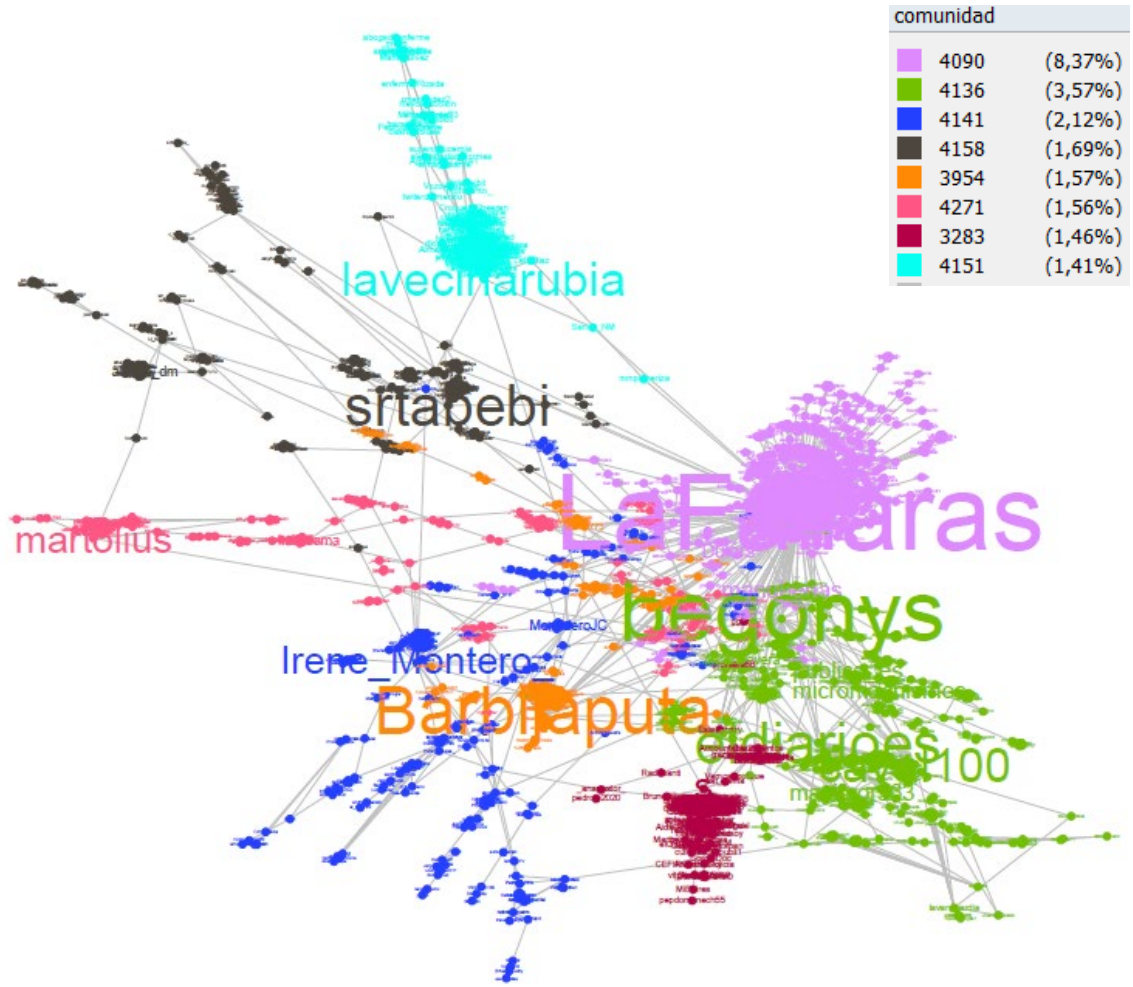

GRÁFICO 32: RED DE MENCIONES CON LAS PRINCIPALES COMUNIDADES

Al contrario que pasa con el grafo de los retuits, en esta red, de forma general, solo hay un líder por comunidad, aunque en la 4136 (verde) encontramos al periódico eldiarioes, a caval100 y a begonys. Este último se observa gráficamente cómo se comporta de intermediario entre comunidades (segundo mayor influyente según betweenness).

La comunidad más grande está liderada por Cristina Fallarás. También encontramos a otros “influencers” liderando comunidades como lavecinarubia (4151) o Barbijaputa (3954).

Por último, cabe destacar como en la comunidad 4141, que habla sobre el PP, el PSOE y la Ley mordaza, está liderada por Irene Montero, que en el 2018 era una diputada del grupo parlamentario PODEMOS.

6. CONCLUSIONES

Respondiendo al objetivo principal del estudio, se ha comprobado como las relaciones que se generan dentro de una red de trasmisión de la información no se comportan igual que las generadas en una red de diálogo, aunque sí que presentan nexos comunes:

En cuanto a la tipología de las redes, la primera diferencia que se ha detectado ha sido el tamaño: la red de retuits es mucho mayor que la de menciones, debido a que los usuarios tienden más a retuitear que a mencionar. Ninguna de las dos redes es homogénea ya que las relaciones en cada una de ellas se han distribuido de manera preferencial, es decir, el mayor porcentaje de conexiones se ha concentrado en un grupo reducido de tuiteros, los "influencers", pero las relaciones que se han creado con los retuits han dado lugar a una única red completamente conexa formada por casi el 99% de los participantes que han retuiteado, donde los nodos están separados, en media, por 6 nodos intermedios, formando así una red de pequeño mundo. Por otro lado, las relaciones formadas a partir de menciones, respuestas o citas han generado numerosos grupos pequeños aislados del resto de la red, siendo la subred conexa de mayor tamaño la representante del 40% del total. Este es uno de los principales motivos por los que la red de menciones tiene casi 100 veces más comunidades que la de retuits, aunque cada una de ellas está más cohesionada.

Se ha observado como en la red de menciones, dado que existe conversación, los que trasmiten la información también la difunden y actúan de intermediarios, mientras que en la red de retuits, al existir grandes generadores de información que no retuitean a nadie, solo actúan de emisores de la información.

Al analizar las grandes comunidades se ha detectado como, en la red de menciones, cada una de ellas está liderada por un único "influencer". Por el contrario, en la red de retuits encontramos varios "influencers" en cada una, estando, además, muy conectados entre todos ellos, lo que da lugar al fenómeno "rich club".

En la red de menciones se han detectado, en algunas de las grandes comunidades, temas algo más específicos, como la política, el caso de La Manada y, en varias de ellas también destacaba el hashtag #yositecreo, en apoyo a las víctimas. En cambio, en la red de retuits, había mayor diversidad temática.

Respondiendo al objetivo secundario referente a las temáticas abordadas bajo el hashtag Cuéntalo, se hallaron algunas bastante específicas como: sufrir tocamientos en las discotecas y de fiesta, ser perseguidas por un coche o un señor, ir con las llaves o el móvil en la mano por la noche cuando están solas por la calle, aguantar comentarios, piropos o miradas mientras iban por la calle, señores que se les acercaban masturbándose o se sacaban el pene por la calle y tener un sentimiento de vergüenza o culpa tras lo que les sucedió.

Un dato muy importante que se ha obtenido y contrastado, es que, de los 4.197 tuits analizados, cerca del 80% de las mujeres que sufrieron algún tipo de abuso, agresión o intimidación sexual no llegaban a la mayoría de edad cuando ocurrieron los hechos, concentrándose el 75% de las menores de edad entre los 12 y 17 años, edades adolescentes. Estos datos se alejan de los obtenidos a través de fuentes oficiales, pero probablemente se acerquen más a la realidad, por lo que complementar estudios oficiales con estudios de la red puede dar una visión global más aproximada de la situación real.

7. SIGUIENTES PASOS

Siguiendo el análisis de los tuits bajo el hashtag #Cuéntalo, el estudio podría complementarse:

- Diferenciando los tuits que están a favor del movimiento de los que están en contra. Para ello se obtendría una muestra aleatoria cuyos tuits se clasificarían manualmente, a fin de entrenar varios modelos de clasificación y seleccionar el mejor se ajustase.
- Extrayendo nuevos tuits con el hashtag cuéntalo que se estén publicando en la actualidad para comprobar si las redes que se generan son distintas. El objetivo sería comprobar si las redes se distribuyen de igual manera cuando el hashtag sobre el que se crean es tendencia que cuando no lo es.

Un enfoque más global de análisis sería estudiar las redes de tuits y menciones que se generan bajo otros hashtags, con el fin de compararlas y poder sacar un patrón común de comportamiento. De igual manera se analizarían tanto cuando son tendencia como cuando no.

8. BIBLIOGRAFÍA

Teoría

Freeman, L. (2004). The Development of Social Network Analysis.

Álvarez, A. & Aguilar-Gallegos, N. (2005). Manual introductorio al análisis de redes sociales. Medias de centralidad.

Aguirre, J.L. (2011). Introducción al Análisis de Redes Sociales. Documentos de Trabajo, 82, Centro Interdisciplinario para el Estudio de Políticas Públicas, Buenos Aires.

Solares Hernández, P. A. (2017). Redes aleatorias, de pequeño mundo y libres de escala (trabajo final de master). Universidad Politécnica de Valencia, España.

Análisis Estadístico de redes sociales. Apuntes de Daniel Gómez González.

https://www.pewresearch.org/internet/2014/02/20/mapping-twitter-topic-networks-from-polarized-crowds-to-community-clusters/

http://www.bsc.es/viz/corner/?p=210&lang=es

https://en.wikipedia.org/wiki/Social_network_analysis#Other_methods_used_alongside_SNA

https://es.wikipedia.org/wiki/Modelo_Erd%C3%B6s%E2%80%93R%C3%A9nyi#cite_ref-1

https://es.wikipedia.org/wiki/Modelo_Barab%C3%A1si%E2%80%93Albert

https://es.wikipedia.org/wiki/Modelo_Watts_y_Strogatz

https://riunet.upv.es/bitstream/handle/10251/86297/SOLARES%20-%20Redes%20aleatorias,%20de%20peque%C3%B1o%20mundo%20y%20libres%20de%20escala.pdf?sequence=1

http://castor.det.uvigo.es:8080/xmlui/bitstream/handle/123456789/115/TFG%20Yamila%20Bouhachmir%20Gonzalez.pdf?sequence=1&isAllowed=y

Uso de paquetes de R:

https://swcarpentry.github.io/r-novice-gapminder-es/13-dplyr/

https://www.tidytextmining.com/

https://quanteda.io/articles/pkgdown/quickstart_es.html

https://igraph.org/r/doc/

https://www.cs.us.es/~fran/curso_unia/red.html

https://www.academia.edu/23701692/An%C3%A1lisis_de_grafos_usando_R_e_igraph

https://github.com/elaragon/R-igraph-Network-Workshop/blob/master/NetSciX%202016%20Workshop.R

https://www.sixhat.net/finding-communities-in-networks-with-r-and-igraph.html

https://programminghistorian.org/es/lecciones/procesamiento-basico-de-textos-en-r

https://www.r-bloggers.com/community-detection-with-louvain-and-infomap/

Uso de Gephi

https://gephi.org/users/

Informes oficiales:

http://www.interior.gob.es/documents/10180/8736571/INFORME+DELITOS+CONTRA+LA+LIBERTAD+E+INDEMNIDAD+SEXUAL+2018.pdf/72779215-38b4-4bb3-bb45-d03029739f5c

https://violenciagenero.igualdad.gob.es/violenciaEnCifras/macroencuesta2015/pdf/Macroencuesta_2019_estudio_investigacion.pdf

ANEXO

Código R

```
library(gsubfn)
library(readr)
library(foreign)
library(dplyr)
library(lubridate)
library(tm)
library(SnowballC)
library(wordcloud)
library(RColorBrewer)
library(quanteda)
library(readxl)
library(stringr)
library(textmineR)
library(igraph)
library(CINNA)
library(tidyverse)
library(readr)
library(foreign)
library(gsubfn)
library(stringr)
library(tidyr)
library(car)
library(fitdistrplus)
require(MASS)
#Importación de los datos
Datos<- read_delim("D:/TFM/Datos_cuentalo.txt","\t", escape_double = FALSE, trim_ws = TRUE)

#=============================================================================#
#================  DEPURACIÓN Y ANÁLISIS DESCRIPTIVO ========================#
#=============================================================================#

#Date
TW_por_fecha<- Datos %>%
```

```
  mutate(fecha_hora=floor_date(Datos$date,unit = 'hour')) %>%
  group_by(fecha_hora) %>% count(fecha_hora)

plot(TW_por_fecha$fecha_hora,TW_por_fecha$n,xlab = "Fecha",ylab = "Nº de TW")

Datos<-Datos%>%
  filter(date >="2018-04-26 00:00:00")

TW_por_fecha<- Datos %>%
  mutate(fecha_hora=floor_date(Datos$date,unit = 'hour')) %>%
  group_by(fecha_hora) %>% count(fecha_hora)
TW_por_fecha$p<-mdy_h(TW_por_fecha$fecha_hora)

plot(TW_por_fecha$fecha_hora,TW_por_fecha$n,xlab = "Fecha",ylab = "Nº de TW")

#Lang
idioma_freq<-as.data.frame(table(Datos$lang))

for (i in 1:nrow(idioma_freq)){
  if(idioma_freq$Var1[i]=='es'){
    idioma_freq$Idioma[i]='Español'
  }else if(idioma_freq$Var1[i]=='ca'){
    idioma_freq$Idioma[i]='Catalán'
  }else if(idioma_freq$Var1[i]=='und'){
    idioma_freq$Idioma[i]='Sin texto: urls,#,@'
  }else if(idioma_freq$Var1[i]=='en'){
    idioma_freq$Idioma[i]='Ingles'
  }else if(idioma_freq$Var1[i]=='pt'){
    idioma_freq$Idioma[i]='Portugues'
  }else if(idioma_freq$Var1[i]=='fr'){
    idioma_freq$Idioma[i]='Francés'
  }else if(idioma_freq$Var1[i]=='eu'){
    idioma_freq$Idioma[i]='Euskera'
  }else{idioma_freq$Idioma[i]='Otros'}
}
```

```
idioma<- idioma_freq %>%
  group_by(Idioma) %>%
  summarise(Freq = sum(Freq))
idioma<- idioma[order(idioma$Freq,decreasing = TRUE),]
View(idioma)


#Location

localización_freq<-as.data.frame(table(Datos$location))
localización_freq<-localización_freq[order(localización_freq$Freq,decreasing = TRUE),]
head(localización_freq,n=30)

#Relation

relation<-as.data.frame(table(Datos$relation))
relationla

for (i in 1:nrow(Datos)){
  if (Datos$relation[i]=='None'){
    Datos$TW_ORIGINAL[i]<-1
    Datos$RT[i]<-0
    Datos$Reply[i]<-0
    Datos$Quote[i]<-0

  }else if (Datos$relation[i]=='RT'){
    Datos$TW_ORIGINAL[i]<-0
    Datos$RT[i]<-1
    Datos$Reply[i]<-0
    Datos$Quote[i]<-0

  }else if (Datos$relation[i]=='reply'){
    Datos$TW_ORIGINAL[i]<-0
    Datos$RT[i]<-0
    Datos$Reply[i]<-1
    Datos$Quote[i]<-0
```

```
  }else if (Datos$relation[i]=='quote'){
    Datos$TW_ORIGINAL[i]<-0
    Datos$RT[i]<-0
    Datos$Reply[i]<-0
    Datos$Quote[i]<-1

  }else if (substring (Datos$text[i] , 1,4 )=='RT @'){
    Datos$relation[i]<-'RT'
    Datos$TW_ORIGINAL[i]<-0
    Datos$RT[i]<-1
    Datos$Reply[i]<-0
    Datos$Quote[i]<-0

  }else{
    Datos$relation[i]<-'None'
    Datos$TW_ORIGINAL[i]<-1
    Datos$RT[i]<-0
    Datos$Reply[i]<-0
    Datos$Quote[i]<-0
  }
}


for (i in 1:nrow(Datos)){
  if (Datos$Reply[i]==1 & Datos$author[i]==Datos$`user replied`[i]){
    Datos$relation[i]<-'None'
    Datos$TW_ORIGINAL[i]<-1
    Datos$Reply[i]<-0

  }
}

num_usuarios_pp<- Datos %>%
  group_by(author) %>%
  summarise(max_followers = max(followers),min_followers=min(followers),num_RT=sum(RT),
```

```
        num_TW_Original=sum(TW_ORIGINAL),num_Reply=sum(Reply),num_quote=sum(Quote))


USER<- Datos %>%
  summarise(RT=sum(RT),
        TW_Original=sum(TW_ORIGINAL),Reply=sum(Reply),quote=sum(Quote))
summary(num_usuarios_pp$num_RT)
summary(num_usuarios_pp$num_TW_Original)
summary(num_usuarios_pp$num_Reply)
summary(num_usuarios_pp$num_quote)



#TEXT (detección hashtag más usados)

hagtag<- strapplyc(Datos$text,"#\\w+")
hagtag<-data.frame(unlist(hagtag))
hagtag_norm <-  tolower(hagtag$unlist.hagtag.)
hagtag_norm <-chartr("áéíóú", "aeiou",hagtag_norm)
hagtag_norm_freq<-as.data.frame(table(hagtag_norm))
hagtag_norm_freq<-hagtag_norm_freq[order(hagtag_norm_freq$Freq,decreasing = TRUE),]
top_n(hagtag_norm_freq,10)

#=====================================================================#
#=========================  TEXT MINING ==========================#
#=====================================================================#
Datos<- mutate(Datos, id = rownames(Datos))

#DEPURACIÓN DEL TEXTO
tweets <- select(Datos,id,text,lang) %>% filter(Datos$RT == 0)
tweets <- select(tweets,id,text) %>% filter(tweets$lang=="es")
tweets$text_dep = gsub("@\\w+", "", tweets$text)
tweets$text_dep <- tolower(tweets$text)
tweets$text_dep = gsub("#cuentalo", "", tweets$text_dep)
tweets$text_dep = gsub("#cuéntalo", "", tweets$text_dep)
tweets$text_dep <- removePunctuation(tweets$text_dep)
tweets$text_dep = nov_text <- removeNumbers(tweets$text_dep)
```

```
tweets$text_dep = gsub("http\\w+", " ", tweets$text_dep)
tweets$text_dep  <- stripWhitespace(tweets$text_dep)
tweets$text_dep <-chartr("áéíóú", "aeiou",tweets$text_dep)

corpus<- corpus(tweets$text_dep)
matriz<- dfm(corpus)
topfeatures(matriz, 12)
textplot_wordcloud(matriz, min_count = 1000, random_order = FALSE,
          rotation = .55,
          color = RColorBrewer::brewer.pal(8, "Dark2"))


#Drop list
lista_stop_word<- stopwords("spanish")
drop_list<-read_excel("D:/TFM/Diccionarios.xlsx",sheet = "drop_list")
lista_stop_word[309:529 ]<-drop_list$Palabra
head(lista_stop_word,n=10)
matriz_stop_word<- dfm(corpus,remove = lista_stop_word)
textplot_wordcloud(matriz_stop_word, min_count = 500, random_order = FALSE,
          rotation = .55,color = RColorBrewer::brewer.pal(8, "Dark2"))


#Resto de diccionarios
resto_diccionarios<-Diccionarios <- read_excel("D:/TFM/Diccionarios.xlsx",sheet = "resto_diccionarios")
tweets$text_dep<-tweets$text_dep %>%str_replace_all(resto_diccionarios$`palabra=palabra_corr`)
corpus_limp<- corpus(tweets$text_dep)
matriz_limp<- dfm(corpus_limp,remove = lista_stop_word)
textplot_wordcloud(matriz_limp, min_count = 450, random_order = FALSE,
          rotation = .55, color = RColorBrewer::brewer.pal(8, "Dark2"))


#CREACIÓN DE TOPICS
set.seed(123456)
dtm <- CreateDtm(doc_vec = tweets$text_dep,doc_names = tweets$id,ngram_window = c(1,
2),stopword_vec = lista_stop_word,
        lower = TRUE,remove_punctuation = TRUE,remove_numbers = TRUE,verbose = FALSE,cpus = 2)
```

```
dtm_500_5000 <- dtm[,colSums(dtm) >500 ]
dtm_500_5000 <- dtm[,colSums(dtm) <5000 ]

model_51 <- FitLdaModel(dtm = dtm_500_5000, k = 5,iterations = 200, optimize_alpha = TRUE,
                        calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_52 <- FitLdaModel(dtm = dtm_500_5000, k = 5,iterations = 200, optimize_alpha = TRUE,
                        calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_53 <- FitLdaModel(dtm = dtm_500_5000, k = 5,iterations = 200, optimize_alpha = TRUE,
                        calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_54 <- FitLdaModel(dtm = dtm_500_5000, k = 5,iterations = 200, optimize_alpha = TRUE,
                        calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_55 <- FitLdaModel(dtm = dtm_500_5000, k = 5,iterations = 200, optimize_alpha = TRUE,
                        calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_56 <- FitLdaModel(dtm = dtm_500_5000, k = 5,iterations = 200, optimize_alpha = TRUE,
                        calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_57 <- FitLdaModel(dtm = dtm_500_5000, k = 5,iterations = 200, optimize_alpha = TRUE,
                        calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_58 <- FitLdaModel(dtm = dtm_500_5000, k = 5,iterations = 200, optimize_alpha = TRUE,
                        calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_59 <- FitLdaModel(dtm = dtm_500_5000, k = 5,iterations = 200, optimize_alpha = TRUE,
                        calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_510 <- FitLdaModel(dtm = dtm_500_5000, k = 5,iterations = 200, optimize_alpha = TRUE,
                         calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_511 <- FitLdaModel(dtm = dtm_500_5000, k = 5,iterations = 200, optimize_alpha = TRUE,
                         calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_512 <- FitLdaModel(dtm = dtm_500_5000, k = 5,iterations = 200, optimize_alpha = TRUE,
                         calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_513 <- FitLdaModel(dtm = dtm_500_5000, k = 5,iterations = 200, optimize_alpha = TRUE,
                         calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_514 <- FitLdaModel(dtm = dtm_500_5000, k = 5,iterations = 200, optimize_alpha = TRUE,
                         calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_515 <- FitLdaModel(dtm = dtm_500_5000, k = 5,iterations = 200, optimize_alpha = TRUE,
                         calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)


model_101 <- FitLdaModel(dtm = dtm_500_5000, k = 10,iterations = 200, optimize_alpha = TRUE,
```

```
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_102 <- FitLdaModel(dtm = dtm_500_5000, k = 10,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_103 <- FitLdaModel(dtm = dtm_500_5000, k = 10,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_104 <- FitLdaModel(dtm = dtm_500_5000, k = 10,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_105 <- FitLdaModel(dtm = dtm_500_5000, k = 10,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_106 <- FitLdaModel(dtm = dtm_500_5000, k = 10,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_107 <- FitLdaModel(dtm = dtm_500_5000, k = 10,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_108 <- FitLdaModel(dtm = dtm_500_5000, k = 10,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_109 <- FitLdaModel(dtm = dtm_500_5000, k = 10,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_1010 <- FitLdaModel(dtm = dtm_500_5000, k = 10,iterations = 200, optimize_alpha = TRUE,
                 calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_1011 <- FitLdaModel(dtm = dtm_500_5000, k = 10,iterations = 200, optimize_alpha = TRUE,
                 calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_1012 <- FitLdaModel(dtm = dtm_500_5000, k = 10,iterations = 200, optimize_alpha = TRUE,
                 calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_1013 <- FitLdaModel(dtm = dtm_500_5000, k = 10,iterations = 200, optimize_alpha = TRUE,
                 calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_1014 <- FitLdaModel(dtm = dtm_500_5000, k = 10,iterations = 200, optimize_alpha = TRUE,
                 calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_1015 <- FitLdaModel(dtm = dtm_500_5000, k = 10,iterations = 200, optimize_alpha = TRUE,
                 calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)


model_151 <- FitLdaModel(dtm = dtm_500_5000, k = 15,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_152 <- FitLdaModel(dtm = dtm_500_5000, k = 15,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_153 <- FitLdaModel(dtm = dtm_500_5000, k = 15,iterations = 200, optimize_alpha = TRUE,
```

```
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_154 <- FitLdaModel(dtm = dtm_500_5000, k = 15,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_155 <- FitLdaModel(dtm = dtm_500_5000, k = 15,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_156 <- FitLdaModel(dtm = dtm_500_5000, k = 15,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_157 <- FitLdaModel(dtm = dtm_500_5000, k = 15,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_158 <- FitLdaModel(dtm = dtm_500_5000, k = 15,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_159 <- FitLdaModel(dtm = dtm_500_5000, k = 15,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_1510 <- FitLdaModel(dtm = dtm_500_5000, k = 15,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_1511 <- FitLdaModel(dtm = dtm_500_5000, k = 15,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_1512 <- FitLdaModel(dtm = dtm_500_5000, k = 15,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_1513 <- FitLdaModel(dtm = dtm_500_5000, k = 15,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_1514 <- FitLdaModel(dtm = dtm_500_5000, k = 15,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_1515 <- FitLdaModel(dtm = dtm_500_5000, k = 15,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)


model_201 <- FitLdaModel(dtm = dtm_500_5000, k = 20,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_202 <- FitLdaModel(dtm = dtm_500_5000, k = 20,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_203 <- FitLdaModel(dtm = dtm_500_5000, k = 20,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_204 <- FitLdaModel(dtm = dtm_500_5000, k = 20,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_205 <- FitLdaModel(dtm = dtm_500_5000, k = 20,iterations = 200, optimize_alpha = TRUE,
```

```
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_206 <- FitLdaModel(dtm = dtm_500_5000, k = 20,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_207 <- FitLdaModel(dtm = dtm_500_5000, k = 20,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_208 <- FitLdaModel(dtm = dtm_500_5000, k = 20,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_209 <- FitLdaModel(dtm = dtm_500_5000, k = 20,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2010 <- FitLdaModel(dtm = dtm_500_5000, k = 20,iterations = 200, optimize_alpha = TRUE,
                 calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2011 <- FitLdaModel(dtm = dtm_500_5000, k = 20,iterations = 200, optimize_alpha = TRUE,
                 calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2012 <- FitLdaModel(dtm = dtm_500_5000, k = 20,iterations = 200, optimize_alpha = TRUE,
                 calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2013 <- FitLdaModel(dtm = dtm_500_5000, k = 20,iterations = 200, optimize_alpha = TRUE,
                 calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2014 <- FitLdaModel(dtm = dtm_500_5000, k = 20,iterations = 200, optimize_alpha = TRUE,
                 calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2015 <- FitLdaModel(dtm = dtm_500_5000, k = 20,iterations = 200, optimize_alpha = TRUE,
                 calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)


dtm_200_1000 <- dtm[,colSums(dtm) >250 ]
dtm_200_1000 <- dtm[,colSums(dtm) <1000 ]

model_2_51 <- FitLdaModel(dtm = dtm_200_1000, k = 5,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_52 <- FitLdaModel(dtm = dtm_200_1000, k = 5,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_53 <- FitLdaModel(dtm = dtm_200_1000, k = 5,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_54 <- FitLdaModel(dtm = dtm_200_1000, k = 5,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_55 <- FitLdaModel(dtm = dtm_200_1000, k = 5,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
```

```
model_2_56 <- FitLdaModel(dtm = dtm_200_1000, k = 5,iterations = 200, optimize_alpha = TRUE,
                          calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_57 <- FitLdaModel(dtm = dtm_200_1000, k = 5,iterations = 200, optimize_alpha = TRUE,
                          calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_58 <- FitLdaModel(dtm = dtm_200_1000, k = 5,iterations = 200, optimize_alpha = TRUE,
                          calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_59 <- FitLdaModel(dtm = dtm_200_1000, k = 5,iterations = 200, optimize_alpha = TRUE,
                          calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_510 <- FitLdaModel(dtm = dtm_200_1000, k = 5,iterations = 200, optimize_alpha = TRUE,
                           calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_511 <- FitLdaModel(dtm = dtm_200_1000, k = 5,iterations = 200, optimize_alpha = TRUE,
                           calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_512 <- FitLdaModel(dtm = dtm_200_1000, k = 5,iterations = 200, optimize_alpha = TRUE,
                           calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_513 <- FitLdaModel(dtm = dtm_200_1000, k = 5,iterations = 200, optimize_alpha = TRUE,
                           calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_514 <- FitLdaModel(dtm = dtm_200_1000, k = 5,iterations = 200, optimize_alpha = TRUE,
                           calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_515 <- FitLdaModel(dtm = dtm_200_1000, k = 5,iterations = 200, optimize_alpha = TRUE,
                           calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)

model_2_101 <- FitLdaModel(dtm = dtm_200_1000, k = 10,iterations = 200, optimize_alpha = TRUE,
                           calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_102 <- FitLdaModel(dtm = dtm_200_1000, k = 10,iterations = 200, optimize_alpha = TRUE,
                           calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_103 <- FitLdaModel(dtm = dtm_200_1000, k = 10,iterations = 200, optimize_alpha = TRUE,
                           calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_104 <- FitLdaModel(dtm = dtm_200_1000, k = 10,iterations = 200, optimize_alpha = TRUE,
                           calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_105 <- FitLdaModel(dtm = dtm_200_1000, k = 10,iterations = 200, optimize_alpha = TRUE,
                           calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_106 <- FitLdaModel(dtm = dtm_200_1000, k = 10,iterations = 200, optimize_alpha = TRUE,
                           calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_107 <- FitLdaModel(dtm = dtm_200_1000, k = 10,iterations = 200, optimize_alpha = TRUE,
                           calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_108 <- FitLdaModel(dtm = dtm_200_1000, k = 10,iterations = 200, optimize_alpha = TRUE,
```

```
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_109 <- FitLdaModel(dtm = dtm_200_1000, k = 10,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_1010 <- FitLdaModel(dtm = dtm_200_1000, k = 10,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_1011 <- FitLdaModel(dtm = dtm_200_1000, k = 10,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_1012 <- FitLdaModel(dtm = dtm_200_1000, k = 10,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_1013 <- FitLdaModel(dtm = dtm_200_1000, k = 10,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_1014 <- FitLdaModel(dtm = dtm_200_1000, k = 10,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_1015 <- FitLdaModel(dtm = dtm_200_1000, k = 10,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)

model_2_151 <- FitLdaModel(dtm = dtm_200_1000, k = 15,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_152 <- FitLdaModel(dtm = dtm_200_1000, k = 15,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_153 <- FitLdaModel(dtm = dtm_200_1000, k = 15,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_154 <- FitLdaModel(dtm = dtm_200_1000, k = 15,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_155 <- FitLdaModel(dtm = dtm_200_1000, k = 15,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_156 <- FitLdaModel(dtm = dtm_200_1000, k = 15,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_157 <- FitLdaModel(dtm = dtm_200_1000, k = 15,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_158 <- FitLdaModel(dtm = dtm_200_1000, k = 15,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_159 <- FitLdaModel(dtm = dtm_200_1000, k = 15,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_1510 <- FitLdaModel(dtm = dtm_200_1000, k = 15,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
```

```
model_2_1511 <- FitLdaModel(dtm = dtm_200_1000, k = 15,iterations = 200, optimize_alpha = TRUE,
                            calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_1512 <- FitLdaModel(dtm = dtm_200_1000, k = 15,iterations = 200, optimize_alpha = TRUE,
                            calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_1513 <- FitLdaModel(dtm = dtm_200_1000, k = 15,iterations = 200, optimize_alpha = TRUE,
                            calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_1514 <- FitLdaModel(dtm = dtm_200_1000, k = 15,iterations = 200, optimize_alpha = TRUE,
                            calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_1515 <- FitLdaModel(dtm = dtm_200_1000, k = 15,iterations = 200, optimize_alpha = TRUE,
                            calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)

model_2_201 <- FitLdaModel(dtm = dtm_200_1000, k = 20,iterations = 200, optimize_alpha = TRUE,
                           calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_202 <- FitLdaModel(dtm = dtm_200_1000, k = 20,iterations = 200, optimize_alpha = TRUE,
                           calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_203 <- FitLdaModel(dtm = dtm_200_1000, k = 20,iterations = 200, optimize_alpha = TRUE,
                           calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_204 <- FitLdaModel(dtm = dtm_200_1000, k = 20,iterations = 200, optimize_alpha = TRUE,
                           calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_205 <- FitLdaModel(dtm = dtm_200_1000, k = 20,iterations = 200, optimize_alpha = TRUE,
                           calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_206 <- FitLdaModel(dtm = dtm_200_1000, k = 20,iterations = 200, optimize_alpha = TRUE,
                           calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_207 <- FitLdaModel(dtm = dtm_200_1000, k = 20,iterations = 200, optimize_alpha = TRUE,
                           calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_208 <- FitLdaModel(dtm = dtm_200_1000, k = 20,iterations = 200, optimize_alpha = TRUE,
                           calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_209 <- FitLdaModel(dtm = dtm_200_1000, k = 20,iterations = 200, optimize_alpha = TRUE,
                           calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_2010 <- FitLdaModel(dtm = dtm_200_1000, k = 20,iterations = 200, optimize_alpha = TRUE,
                            calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_2011 <- FitLdaModel(dtm = dtm_200_1000, k = 20,iterations = 200, optimize_alpha = TRUE,
                            calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_2012 <- FitLdaModel(dtm = dtm_200_1000, k = 20,iterations = 200, optimize_alpha = TRUE,
                            calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_2013 <- FitLdaModel(dtm = dtm_200_1000, k = 20,iterations = 200, optimize_alpha = TRUE,
```

```
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_2014 <- FitLdaModel(dtm = dtm_200_1000, k = 20,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)
model_2_2015 <- FitLdaModel(dtm = dtm_200_1000, k = 20,iterations = 200, optimize_alpha = TRUE,
                calc_likelihood = TRUE,calc_coherence = TRUE,calc_r2 = TRUE)

load("D:/TFM/modelos_topics_OK.RData")

model_5<-
rbind(model_51$coherence,model_52$coherence,model_53$coherence,model_54$coherence,model_5
5$coherence

,model_56$coherence,model_57$coherence,model_58$coherence,model_59$coherence,model_510$c
oherence

,model_511$coherence,model_512$coherence,model_513$coherence,model_514$coherence,model_5
15$coherence)

model_10<-rbind( model_101$coherence, model_102$coherence, model_103$coherence,
model_104$coherence, model_105$coherence
        , model_106$coherence, model_107$coherence, model_108$coherence,
model_109$coherence, model_1010$coherence
        , model_1011$coherence, model_1012$coherence, model_1013$coherence,
model_1014$coherence, model_1015$coherence)

model_15<-rbind( model_151$coherence, model_152$coherence, model_153$coherence,
model_154$coherence, model_155$coherence
        , model_156$coherence, model_157$coherence, model_158$coherence,
model_159$coherence, model_1510$coherence
        , model_1511$coherence, model_1512$coherence, model_1513$coherence,
model_1514$coherence, model_1515$coherence)
model_20<-rbind( model_201$coherence, model_202$coherence, model_203$coherence,
model_204$coherence, model_205$coherence
        , model_206$coherence, model_207$coherence, model_208$coherence,
model_209$coherence, model_2010$coherence
        , model_2011$coherence, model_2012$coherence, model_2013$coherence,
model_2014$coherence, model_2015$coherence)

model_5<-as.numeric(model_5)
model_10<-as.numeric(model_10)
```

```
model_15<-as.numeric(model_15)
model_20<-as.numeric(model_20)
summary(model_5)
summary(model_10)
summary(model_15)
summary(model_20)

model_51$top_terms <- GetTopTerms(phi = model_51$phi, M = 10)
head(t(model_51$top_terms))



model_5_r<-rbind(model_51$r2,model_52$r2,model_53$r2,model_54$r2,model_55$r2
            ,model_56$r2,model_57$r2,model_58$r2,model_59$r2,model_510$r2
            ,model_511$r2,model_512$r2,model_513$r2,model_514$r2,model_515$r2)

model_10_r<-rbind( model_101$r2, model_102$r2, model_103$r2, model_104$r2, model_105$r2
             , model_106$r2, model_107$r2, model_108$r2, model_109$r2, model_1010$r2
             , model_1011$r2, model_1012$r2, model_1013$r2, model_1014$r2, model_1015$r2)

model_15_r<-rbind( model_151$r2, model_152$r2, model_153$r2, model_154$r2, model_155$r2
             , model_156$r2, model_157$r2, model_158$r2, model_159$r2, model_1510$r2
             , model_1511$r2, model_1512$r2, model_1513$r2, model_1514$r2, model_1515$r2)
model_20_r<-rbind( model_201$r2, model_202$r2, model_203$r2, model_204$r2, model_205$r2
             , model_206$r2, model_207$r2, model_208$r2, model_209$r2, model_2010$r2
             , model_2011$r2, model_2012$r2, model_2013$r2, model_2014$r2, model_2015$r2)

model_5_r<-as.numeric(model_5_r)
model_10_r<-as.numeric(model_10_r)
model_15_r<-as.numeric(model_15_r)
model_20_r<-as.numeric(model_20_r)
summary(model_5_r)
summary(model_10_r)
summary(model_15_r)
summary(model_20_r)

model_203$top_terms <- GetTopTerms(phi = model_203$phi, M = 10)
```

```
head(t(model_203$top_terms),20)


#EDAD DE LAS VICTIMAS
tweets$origen_años<- tolower(tweets$text) #pone todo el texto en minisculas
tweets$origen_años<-tweets$origen_años %>%
  str_replace_all(c(" año " = " años " ," anno "=" años "," anio "=" años "," anho "=" años "," annos "=" años "," anios "=" años "," anhos "=" años " ))

tweets$origen_años<- sub("tenia [0-9]* años mas", "",tweets$origen_años)
tweets$origen_años<- sub("tenía [0-9]* años mas", "",tweets$origen_años)
tweets$origen_años<- sub("tenia [0-9]* años más", "",tweets$origen_años)
tweets$origen_años<- sub("tenía [0-9]* años más", "",tweets$origen_años)
tweets$origen_años<- sub("tenia [0-9]* años menos", "",tweets$origen_años)
tweets$origen_años<- sub("tenía [0-9]* años menos", "",tweets$origen_años)
tweets$origen_años<- sub("el tenia [0-9]* años", "",tweets$origen_años)
tweets$origen_años<- sub("él tenia [0-9]* años", "",tweets$origen_años)
tweets$origen_años<- sub("el tenía [0-9]* años", "",tweets$origen_años)
tweets$origen_años<- sub("él tenía [0-9]* años", "",tweets$origen_años)
tweets$origen_años<- sub("[a-z]* que tenía [0-9]* años", "",tweets$origen_años)
tweets$origen_años<- sub("[a-z]* que tenia [0-9]* años", "",tweets$origen_años)
años1<- str_extract(tweets$origen_años,"tenía [0-9]* años")
años2<- str_extract(tweets$origen_años,"tenia [0-9]* años")
años3<- str_extract(tweets$origen_años,"#cuéntalo [0-9]* años")
años4<- str_extract(tweets$origen_años,"#cuentalo [0-9]* años")
años5<- str_extract(tweets$origen_años,"con [0-9]* años")
años6<- str_extract(tweets$origen_años,"a los [0-9]* años")
freqaños1<-años1[!is.na(años1)]
freqaños2<-años2[!is.na(años2)]
freqaños3<-años3[!is.na(años3)]
freqaños4<-años4[!is.na(años4)]
freqaños5<-años5[!is.na(años5)]
freqaños6<-años6[!is.na(años6)]

union_años<-as.vector(rbind(años1,años2,años3,años4,años5,años6))
union_años <- union_años[!is.na(union_años)]
```

```
union_años<- sub("tenía", "",union_años)
union_años<- sub("tenia", "",union_años)
union_años<-sub("#cuéntalo","",union_años)
union_años<-sub("#cuentalo","",union_años)
union_años<-sub("con","",union_años)
union_años<-sub("a los ","",union_años)
union_años<-sub("años","",union_años)
union_años<- as.numeric(union_años)
summary(union_años)
var(union_años)
quantile(union_años,c(seq(0.1,0.9,0.1)))
par(mfrow=c(1, 2))
hist(union_años,col = "lightblue",xlab = "Edad",breaks=60)
boxplot(union_años, col = "lightblue",ylab = "Edad")
hist(union_años,col = "lightblue",xlim= c(0,25),xlab = "Edad",breaks=100)
boxplot(union_años, col = "lightblue",ylab = "Edad",ylimit = c(0,25))

#=============================================================================#
#============================== ANÁLISIS DE LA RED ===============================#
#=============================================================================#

#CREACCIÓN DE LOS GRAFOS
#Grafo retuits
Datos$author<-substring(Datos$author,2,length(Datos$author))
retweets <- select(Datos,`user retweeted`,author) %>% filter(Datos$relation == "RT")
names(retweets)[1] = "Influyente"
g_RT <- graph.data.frame(retweets,vertices = NULL)

#Grafo menciones
citas<-select(Datos,`user quoted`,author,text) %>% filter(Datos$relation == "quote")
respuestas<- select(Datos,`user replied`,author,text) %>% filter(Datos$relation == "reply")
Datos$menciones<- strapplyc(Datos$text,"@\\w+")
menciones_TWOrig<- select(Datos,menciones,author,text) %>% filter(Datos$relation == "None"  &
Datos$menciones != "character(0)")
menciones<-menciones_TWOrig %>% mutate(Original=str_split(menciones, ",")) %>% unnest()
menciones$menciones<-substring(menciones$menciones,2,length(menciones$menciones))
```

```
menciones<-menciones[,c(1,2,3)]
names(citas)[1] = "Influyente"
names(respuestas)[1] = "Influyente"
names(menciones)[1] = "Influyente"
lista_aristas<- rbind(citas,respuestas,menciones)
g_menciones <- graph.data.frame(lista_aristas,vertices = NULL)

write.graph(g_menciones,file="g_menciones.graphml",format="graphml")
write.graph(g_RT10,file="g_RT10.graphml",format="graphml")



#ANÁLISIS TIPOLOGÍA DE LA RED

vcount(g_RT)
vcount(g_menciones)
ecount(g_RT)
ecount(g_menciones)
graph.density(g_RT)
graph.density(g_menciones)

Degree_RT <- degree(g_RT,mode = "all")
Degree_menciones <- degree(g_menciones,mode = "all")
Degree_in_RT <-  degree(g_RT,mode = "in")
Degree_in_menciones <- degree(g_menciones,mode = "in")
Degree_out_RT <- degree(g_RT,mode = "out")
Degree_out_menciones <- degree(g_menciones,mode = "out")

par(mfrow=c(1,3))

boxplot(Degree_RT,main="Degree",ylab="Degree", col = "lightblue",ylimit = c(0,7))
hist(Degree_RT,main="Degree",ylim = c(0,3000),xlim = c(0,20),breaks=100000, col = "light blue", xlab =
"", ylab = "Frecuencia")
boxplot(Degree_in_RT,main="Degree in", ylab="Degree in", col = "lightblue",ylimit = c(0,7))
hist(Degree_in_RT,main="Degree in",ylim = c(0,10000),xlim = c(-10,20),breaks=800, col = "light blue",
xlab = "", ylab = "Frecuencia")
boxplot(Degree_out_RT,main="Degree out",ylab="Degree out", col = "lightblue",ylimit = c(0,7))
```

```
hist(Degree_out_RT,main="Degree out",ylim = c(0,3000),xlim = c(0,20),breaks=100000, col = "light blue", xlab = "", ylab = "Frecuencia")


boxplot(Degree_menciones,main="Degree", col = "lightblue",ylimit = c(0,7),ylab="Degree")

hist(Degree_menciones,main="Degree",ylim = c(0,3000),xlim = c(0,15),breaks=1000, col = "light blue", xlab = "", ylab = "Frecuencia")

boxplot(Degree_in_menciones,main="Degree in", col = "lightblue",ylimit = c(0,7),

      ylab="Degree in")

hist(Degree_in_menciones,main="Degree in",ylim = c(0,3000),xlim = c(0,15),breaks=300, col = "light blue",

   xlab = "", ylab = "Frecuencia")

boxplot(Degree_out_menciones,main="Degree out",col = "lightblue",ylimit = c(0,7),

      ylab="Degree out")

hist(Degree_out_menciones,main="Degree out",ylim = c(0,3000),xlim = c(0,15),breaks=1000, col = "light blue",

   xlab = "", ylab = "Frecuencia")


summary(Degree_RT)

quantile(Degree_RT,probs=c(seq(0.1,0.9,0.1)))

var( Degree_RT )

summary(Degree_menciones)

quantile(Degree_menciones,probs=c(seq(0.1,0.9,0.1)))

var( Degree_menciones )

summary(Degree_in_RT)

quantile(Degree_in_RT,c(seq(0.1,0.9,0.1)))

var(  Degree_in_RT)

summary(Degree_in_menciones)

sd(Degree_in_menciones)

quantile(Degree_in_menciones,c(seq(0.1,0.9,0.1)))

var(Degree_in_menciones  )

summary(Degree_out_RT)

quantile(Degree_out_RT,c(seq(0.1,0.9,0.1)))

var( Degree_out_RT )

summary(Degree_out_menciones)

quantile(Degree_out_menciones,c(seq(0.1,0.9,0.1)))

var(Degree_out_menciones  )
```

```
##REDES ALEATORIAS: Modelo de Erdos-Renyi

tabla_RT<- table(Degree_RT)

pr_teor_RT <- dbinom(0:442, vcount(g_RT), graph.density(g_RT))

chisq.test(tabla_RT, p = pr_teor_RT)

tabla_menc <- table(Degree_menciones)

pr_teor_mec <- dbinom(0:63, vcount(g_menciones), graph.density(g_menciones))

chisq.test(tabla_menc, p = pr_teor_mec)


Ajusten_RT<-fitdistr(Degree_RT, "normal")

ks.test(Degree_RT, "pnorm", mean =Ajusten_RT$estimate[1], sd= Ajusten_RT$estimate[2])

Ajusten_menc<-fitdistr(Degree_menciones, "normal")

ks.test(Degree_menciones, "pnorm", mean =Ajusten_menc$estimate[1], sd=
Ajusten_menc$estimate[2])


Ajustex_RT <- fitdistr(Degree_RT, "exponential")

ks.test(Degree_RT, "pexp", rate=Ajustex_RT$estimate[1])

Ajustex_menc <- fitdistr(Degree_menciones, "exponential")

ks.test(Degree_menciones, "pexp", rate=Ajustex_menc$estimate[1])


##REDES LIBRES DE ESCALA: Modelo Barabasi-Albert

deg_distr_RT <-degree.distribution(g_RT, cumulative=T, mode="all")

deg_distr_menciones <-degree.distribution(g_menciones , cumulative=T, mode="all")


par(mfrow=c(1,2))

plot(deg_distr_RT, bg="black",pch=21, xlab="Nodos", ylab="Freq acumulada grados")

plot(deg_distr_menciones, bg="black",pch=21, xlab="Nodos", ylab="Freq acumulada grados")

plot(deg_distr_RT, xlim = c(1,100), bg="black",pch=21, main = 'Grafo RT',xlab="Nodos", ylab="Freq
acumulada grados")

plot(deg_distr_menciones, xlim = c(1,100), bg="black",pch=21, main = 'Grafo Menciones',xlab="Nodos",
ylab="Freq acumulada grados")


power_RT <- power.law.fit(deg_distr_RT)

power_RT[c(5,6)]

power_menciones <- power.law.fit(deg_distr_menciones)

power_menciones[c(5,6)]


##REDES DE PEQUEÑO MUNDO: Modelo de Wats-Strogaz
```

```
is_connected (g_RT)
componentes_conexas2<-components(g_RT)
length(componentes_conexas$csize)
max(componentes_conexas2$csize)
(max(componentes_conexas$csize)/vcount(g_RT))*100

is_connected (g_menciones)
componentes_conexas_m<-components(g_menciones)
length(componentes_conexas_m$csize)
max(componentes_conexas_m$csize)
vcount(g_menciones)
(max(componentes_conexas_m$csize)/vcount(g_menciones))*100

CCG_menciones<- giant_component_extract(g_menciones, directed = TRUE)
CCG_RT<- giant_component_extract(g_RT, directed = TRUE)
CCG_menciones<-graph.data.frame(CCG_menciones[2])
CCG_RT<-graph.data.frame(CCG_RT[2])
average.path.length(CCG_RT, directed=TRUE)
transitivity(CCG_RT)

grado<-as.numeric(degree(CCG_RT))
grado_CCG_RT<-as.data.frame(grado)
grado_CCG_RT<-grado_CCG_RT[order(grado_CCG_RT$grado,decreasing = FALSE),]
mitad_grado<-grado_CCG_RT/2
randon_libre_escala <- barabasi.game(n=152677,out.seq=mitad_grado,directed=TRUE)
coef_agrupamiento <- transitivity(randon_libre_escala)

for(i in 2:10000){
  print(i)
  randon_libre_escala <- barabasi.game(n=152677,out.seq=mitad_grado,directed=TRUE)
  coef_agrupamiento[i] <- transitivity(randon_libre_escala)
}

summary(coef_agrupamiento)
sum(coef_agrupamiento>0.0004517978)/10000
```

```
#IDENTIFICACIÓN LÍDERES DE OPINIÓN

Betweenness_RT<- betweenness(g_RT)

Page_rank_RT <- page_rank(g_RT)$vector

Medidas_centralidad_RT <-
data.frame(author=V(g_RT)$name,Degree_RT,Degree_in_RT,Degree_out_RT,Betweenness_RT,Page_ran
k_RT)

head(Medidas_centralidad_RT)


Degree_in_menc<-Degree_in_menciones

Degree_menc<-Degree_menciones

Degree_out_menc<-Degree_out_menciones

Betweenness_menc<- betweenness(g_menciones)

Page_rank_menc <- page_rank(g_menciones)$vector

Medidas_centralidad_menc <-
data.frame(author=V(g_menciones)$name,Degree_menc,Degree_in_menc,Degree_out_menc,Between
ness_menc,Page_rank_menc)

head(Medidas_centralidad_menc)


#Análisis de correlación

cor(Medidas_centralidad_RT[,2:6],method = "pearson")

cor(Medidas_centralidad_menc[,2:6],method = "pearson")

pairs(Medidas_centralidad_menc[,2:6],col="lightblue")

library(PerformanceAnalytics)

chart.Correlation(Medidas_centralidad_menc[,2:6],col="lightblue")

panel.reg <- function (x, y)

{

  points(x, y, pch=20)

  abline(lm(y ~ x), lwd=2, col='dodgerblue2')

}


# Función para obtener la correlación

panel.cor <- function(x, y, digits=2, prefix="", cex.cor)

{

  usr <- par("usr"); on.exit(par(usr))

  par(usr = c(0, 1, 0, 1))

  r <- abs(cor(x, y))

  txt <- format(c(r, 0.123456789), digits=digits)[1]
```

```
  txt <- paste(prefix, txt, sep="")
  if(missing(cex.cor)) cex <- 0.8/strwidth(txt)
  text(0.5, 0.5, txt, cex = cex * r)
}
pairs(Medidas_centralidad_menc[,2:6],
      lower.panel = panel.reg,
      main='Grafo Menciones',
      labels=c('Degree', 'Degree in', 'Degree out', 'Betweenness','Page rank'),
      upper.panel = panel.cor)

pairs(Medidas_centralidad_RT[,2:6],
      lower.panel = panel.reg,
      main='Grafo retuits',
      labels=c('Degree', 'Degree in', 'Degree out', 'Betweenness','Page rank'),
      upper.panel = panel.cor)

#¿Cual son los usuarios más influyentes?
head(Medidas_centralidad_RT[order(Medidas_centralidad_RT$Degree_out_RT,decreasing = TRUE),])
head(Medidas_centralidad_RT[order(Medidas_centralidad_RT$Degree_in_RT,decreasing = TRUE),])
head(Medidas_centralidad_RT[order(Medidas_centralidad_RT$Betweenness_RT,decreasing = TRUE),])
head(Medidas_centralidad_RT[order(Medidas_centralidad_RT$Page_rank_RT,decreasing = TRUE),])

head(Medidas_centralidad_menc[order(Medidas_centralidad_menc$Degree_out_menc ,decreasing =
TRUE),])
head(Medidas_centralidad_menc[order(Medidas_centralidad_menc$Degree_in_menc ,decreasing =
TRUE),])
head(Medidas_centralidad_menc[order(Medidas_centralidad_menc$Betweenness_menc ,decreasing =
TRUE),])
head(Medidas_centralidad_menc[order(Medidas_centralidad_menc$Page_rank_menc ,decreasing =
TRUE),])


#DETECCIÓN de COMUNIIDADES

g_RT_ind<-as.undirected(g_RT)
g_menc_ind<-as.undirected(g_menciones)
CCG_RT_ind<-as.undirected(CCG_RT)
```

```
CCG_menc_ind<-as.undirected(CCG_menciones)
g_RT_ind<-simplify(g_RT_ind)
g_menc_ind<-simplify(g_menc_ind)
CCG_RT_ind<-simplify(CCG_RT_ind)
CCG_menc_ind<-simplify(CCG_menc_ind)

luvian_RT<-cluster_louvain (g_RT_ind, weights = NULL)
luvian_menc<-cluster_louvain (g_menc_ind, weights = NULL)
luvian_CCG_RT<-cluster_louvain (CCG_RT_ind, weights = NULL)
luvian_CCG_menc<-cluster_louvain (CCG_menc_ind, weights = NULL)

infomap_RT <- cluster_infomap(g_RT_ind)
infomap_menc <- cluster_infomap(g_menc_ind)
infomap_CCG_RT <- cluster_infomap(CCG_RT_ind)
infomap_CCG_menc <- cluster_infomap(CCG_menc_ind)

fastgreedy_RT<-fastgreedy.community(g_RT_ind,modularity = TRUE)
fastgreedy_menc<-fastgreedy.community(g_menc_ind,modularity = TRUE)
fastgreedy_CCG_RT<-fastgreedy.community(CCG_RT_ind,modularity = TRUE)
fastgreedy_CCG_menc<-fastgreedy.community(CCG_menc_ind,modularity = TRUE)

max(membership(luvian_RT))
max(membership(luvian_menc))
max(membership(luvian_CCG_RT))
max(membership(luvian_CCG_menc))
max(membership(infomap_RT))
max(membership(infomap_menc))
max(membership(infomap_CCG_RT))
max(membership(infomap_CCG_menc))
max(membership(fastgreedy_RT))
max(membership(fastgreedy_menc))
max(membership(fastgreedy_CCG_RT))
max(membership(fastgreedy_CCG_menc))

modularity(luvian_RT)
modularity(luvian_menc)
```

```
modularity(luvian_CCG_RT)
modularity(luvian_CCG_menc)
modularity(infomap_RT)
modularity(infomap_menc)
modularity(infomap_CCG_RT)
modularity(infomap_CCG_menc)
modularity(fastgreedy_RT)
modularity(fastgreedy_menc)
modularity(fastgreedy_CCG_RT)
modularity(fastgreedy_CCG_menc)
modularity(walktrap_RT)
modularity(walktrap_menc)
modularity(walktrap_CCG_RT)
modularity(walktrap_CCG_menc)

par(mfrow = c(2, 2))
plot(sizes(luvian_RT),bg="black", xlab="Comunidad", ylab="Nº de tuiteros",main="Grafo RT")
plot(sizes(luvian_menc),bg="black", xlab="Comunidad", ylab="Nº de tuiteros",main="Grafo menciones")
plot(sizes(luvian_CCG_RT),bg="black", xlab="Comunidad", ylab="Nº de tuiteros",main="Conexa gigante grafo RT")
plot(sizes(luvian_CCG_menc),bg="black", xlab="Comunidad", ylab="Nº de tuiteros",main="Conexa gigante grafo menciones")

size_RT<-as.data.frame(sizes(luvian_CCG_RT))
summary(size_RT$Freq)
quantile(size_RT$Freq,c(seq(0.1,0.9,0.1)))
head(size_RT[order(size_RT$Freq ,decreasing = TRUE),],20)
head(luvian_RT$memberships)
size_menc<-as.data.frame(sizes(luvian_menc))
summary(size_menc$Freq)
quantile(size_menc$Freq,c(seq(0.99,0.999,0.001)))
head(size_menc[order(size_menc$Freq ,decreasing = TRUE),],20)

#Nube de palabras por comunidad
RT<-Datos%>% select(author,text)%>%
  filter(Datos$RT ==1)
```

```
freq_RT<-as.data.frame(table(RT$author))
RT$blanco<-""
group_RT<-RT %>%
  split(.$author) %>%
  lapply(function(x) paste(
    paste0(x$text, collapse=" "),
    paste0(x$blanco, collapse =" "))) %>%
  cbind() %>%
  as.data.frame()
as<-as.character(group_RT$.)
RT_group<-as.data.frame(as.character(group_RT$.))
names(RT_group)[1] = "texto"
RT_group$author<-freq_RT$Var1
RT_group$author <- gsub("@", "", RT_group$author)
n_cluster<-luvian_CCG_RT$membership
nombres<-luvian_CCG_RT$names
nom_cluster<-as.data.frame(n_cluster)
nom_cluster$author<-nombres
union<-merge(x = RT_group, y = nom_cluster, by = "author")
union$text_dep <- gsub("@\\w+", "", union$text)
union$text_dep <- tolower(union$text_dep)
union$text_dep = gsub("#cuentalo", "", union$text_dep)
union$text_dep = gsub("#cuéntalo", "", union$text_dep)
union$text_dep <- removePunctuation(union$text_dep)
union$text_dep = nov_text <- removeNumbers(union$text_dep)
union$text_dep = gsub("http\\w+", " ", union$text_dep)
union$text_dep  <- stripWhitespace(union$text_dep)
union$text_dep <-chartr("áéíóú", "aeiou",union$text_dep)

union$text_dep<-union$text_dep %>%
  str_replace_all(resto_diccionarios$`palabra=palabra_corr`)
for (i in c(47,3,28,4,5,11,53,50,86,42,69,49,83,77)){
  message("Cluster ", i)
  texto<-union%>%
    filter(union$n_cluster ==i)
  corpus<- corpus(texto$text_dep)
```

```
  dfm <- dfm(corpus,
          remove_punct=TRUE,remove = drop_list$Palabra)
  print(topfeatures(dfm, n=10))
  textplot_wordcloud(dfm, min_count = 500,max_words = 100, main="dsd", random_order = FALSE,
                rotation = .55,
                color = RColorBrewer::brewer.pal(8, "Dark2"))
}


menciones<-lista_aristas[,c(2,3)]
freq_Menciones<-as.data.frame(table(menciones$author))

menciones$blanco<-""
group_menc<-menciones %>%
  split(.$author) %>%
  lapply(function(x) paste(
    paste0(x$text, collapse=" "),
    paste0(x$blanco, collapse =" "))) %>%
  cbind() %>%
  as.data.frame()
as<-as.character(group_menc$.)

menc_group<-as.data.frame(as.character(group_menc$.))
names(menc_group)[1] = "texto"


menc_group$author<-freq_Menciones$Var1
menc_group$author <- gsub("@", "", menc_group$author)

n_cluster_mec<-luvian_menc$membership
nombres_mec<-luvian_menc$names
nom_cluster_mec<-as.data.frame(n_cluster_mec)
nom_cluster_mec$author<-nombres_mec

union_menc<-merge(x = menc_group, y = nom_cluster_mec, by = "author")
```

```
union_menc$text_dep <- gsub("@\\w+", "", union_menc$text) #quita menciones
union_menc$text_dep <- tolower(union_menc$text_dep) #pone todo el texto en minisculas
union_menc$text_dep = gsub("#cuentalo", "", union_menc$text_dep) #quita hasgtag
union_menc$text_dep = gsub("#cuéntalo", "", union_menc$text_dep)
union_menc$text_dep <- removePunctuation(union_menc$text_dep)  #quita signos de puntuación
union_menc$text_dep = nov_text <- removeNumbers(union_menc$text_dep) #quita números
union_menc$text_dep = gsub("http\\w+", " ", union_menc$text_dep) #quita links http
union_menc$text_dep  <- stripWhitespace(union_menc$text_dep)
union_menc$text_dep <-chartr("áéíóú", "aeiou",union_menc$text_dep)

union_menc$text_dep<-union_menc$text_dep %>%
  str_replace_all(resto_diccionarios$`palabra=palabra_corr`)

for (i in c(4090,4136,4141,4158,3954,4271,3283,4151)){
  message("Cluster ", i)
  texto<-union_menc%>%
    filter(union_menc$n_cluster ==i)
  corpus<- corpus(texto$text_dep)
  dfm <- dfm(corpus,
             remove_punct=TRUE,remove = drop_list$Palabra)
  print(topfeatures(dfm, n=10))
  textplot_wordcloud(dfm,min_size =5  ,max_words = 100, random_order = FALSE,
                     rotation = .55,
                     color = RColorBrewer::brewer.pal(8, "Dark2"))
}
```