Escuela Politécnica Superior de Jaén

Grado en Ingeniería Informática

UNIVERSIDAD DE JAÉN
Nombre del Centro

Trabajo Fin de Grado

ANÁLISIS DE LA RED SOCIAL TWITTER MEDIANTE TÉCNICAS BASADAS EN GRAFOS

Alumno: Javier García Garzón

Tutor: Arturo Montejo Ráez

Dpto: Lenguajes y Sistemas Informáticos

Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Don Arturo Montejo Ráez, tutor del Proyecto Fin de Carrera titulado: ANÁLISIS DE LA RED SOCIAL TWITTER MEDIANTE TÉCNICAS BASADAS EN GRAFOS, que presenta Javier García Garzón, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Junio de 2021

El alumno: Los tutores:

Firmado digitalmente por MONTEJO RAEZ ARTURO - 26026531F el día 30/06/2021

Javier García Garzón Arturo Montejo Ráez

Índice

1. INTRODUCCIÓN

En este TFG voy a plasmar mis motivaciones y objetivos a resolver durante el problema propuesto, ya que se trata de un TFG experimental mi objetivo es la aplicación de determinadas tecnologías para el análisis de tendencias en Twitter, de manera que el lector pueda entender los pasos que se han realizado y el porqué.

1.1. Motivación

Vivimos en la era de la información y como tal una de las fuentes principales somos nosotros mismos y nuestras interacciones, las redes sociales forman parte esencial en estas comunicaciones y por eso encuentro esencial su análisis. ¿Qué opina la gente sobre un determinado tema?, ¿Cómo se distribuye la información?,¿Cuál es el seguimiento sobre un determinado tema? Estas preguntas tienen una solución sencilla cuando se trata de unos pocos usuarios, pero en las redes sociales estamos hablando de millones de usuarios interactuando entre sí por lo que analizar y presentar esta información de manera automática, eficaz y visual es primordial para el análisis de tendencias.

1.2. Propósito

Con este TFG mi propósito es poder recopilar tweets de una tendencia en Twitter para posteriormente o en ese mismo momento analizar las diferentes comunidades y opiniones que se generan sobre un tema de actualidad en concreto para poder presentar esta información de manera gráfica y lo más clara posible.

1.3. Objetivos

1. Analizar tendencias en redes sociales usando la herramienta de análisis de grafos
2. Conocer las API de acceso a Twitter
3. Profundizar en los algoritmos de análisis de grafos
4. Definir y estudiar distintos métodos de extracción de relaciones entre usuarios y envíos: retweets, a partir de menciones… como forma de crear grafos

1.4 Estructura del Proyecto

El proyecto se divide en tres fases:

a) Captura de tweets (en vivo y por petición).
b) Procesamiento y exportación de los tweets.
c) Creación, procesamiento, análisis y exportación de los grafos.

A lo largo del *apartado 2* se explica que consiste cada una de estas fases, así como las soluciones propuestas para cada punto y el porqué.

En el *apartado 3* se explica el desarrollo y el análisis los diferentes experimentos.

En *apartado 4* se dan unas conclusiones generales del proyecto.

1.5 Resultados esperados

- Conjunto de scripts que nos permita recopilar información de Twitter mediante el acceso a su API.
- Conjunto de scripts que nos permitan procesar la información para poder ser exportada a un software de procesamiento de grafos.
- Grafo o conjunto de grafos que nos categoriza una tendencia.

2. Contexto del Proyecto

Ya he expuesto mi idea sobre este TFG, pero realmente existen muchas redes sociales válidas para realizar un análisis de tendencias, entonces ¿Por qué he elegido Twitter? ¿Cómo vamos a recopilar la información? ¿Qué consideraciones debemos tener para el análisis de opiniones?¿Que herramientas usaremos en nuestro proyecto?

A lo largo de este apartado explicaré el contexto del proyecto y las diferentes consideraciones realizadas para la solución del mismo.

2.1. Twitter como red social

Para empezar con un poco de contexto, diremos que Twitter es una red social creada en el año 2006, y que en enero de 2021 tiene 322,4 millones de usuarios activos. Twitter consiste en una red social de microblogging en la cual se pueden escribir tweets de 280 caracteres, esto favorece que los usuarios tengan que escribir un texto lo más corto y conciso posible lo que facilita su análisis.

La red social sigue un sistema de seguidores estándar de manera que los usuarios interesados en seguir a otro recibirán sus tuits (mensajes) en su página principal (tweetline), pero esto no es la única manera que tienen los usuarios de ver tweets, ya que podrán explorar a su gusto el contenido en la red social, en concreto suelen destacar los hashtags, que no dejan de ser temas de actualidad que más destacan en ese momento por la cantidad de tweets que reciben en un momento determinado. Además, los usuarios podrán interactuar sobre los tweets mediante RT (retweet), MG (me gusta) o respondiendo lo que provocará que compartan estos tweets a sus seguidores propagando así los tweets de forma exponencial.

Twitter es una red social no dirigida, esto significa que un usuario esté relacionado con otro no implica necesariamente que esta relación sea bidireccional, es decir es asimétrica, como muchas de las relaciones del mundo real.

Según el estudio *(Kwak et al., 2010)* si representamos la red social Twitter en un grafo a partir de millones de usuarios, relaciones sociales y tweets tras procesar algunos algoritmos de la teoría de grafos obtenemos unas características interesantes:

- **Reciprocidad**: El 77,9 % de los de usuarios están relacionaods en un solo sentido, es decir solo el 22,1% de las relaciones son biderecionales.
- **Homofilia**: Tendencia a que usuarios similares se sigan, puede ser por cuestiones geográficas, culturales o sociales.
- **La temporalidad**: El Retuit es el mecanismo por el cuál más se propagan los tweets, y el tiempo es una cuestión relevante, el estudio indica que la mitad total de los retuits se producen en la primera hora y un 75% se produce en las primeras 24 horas.

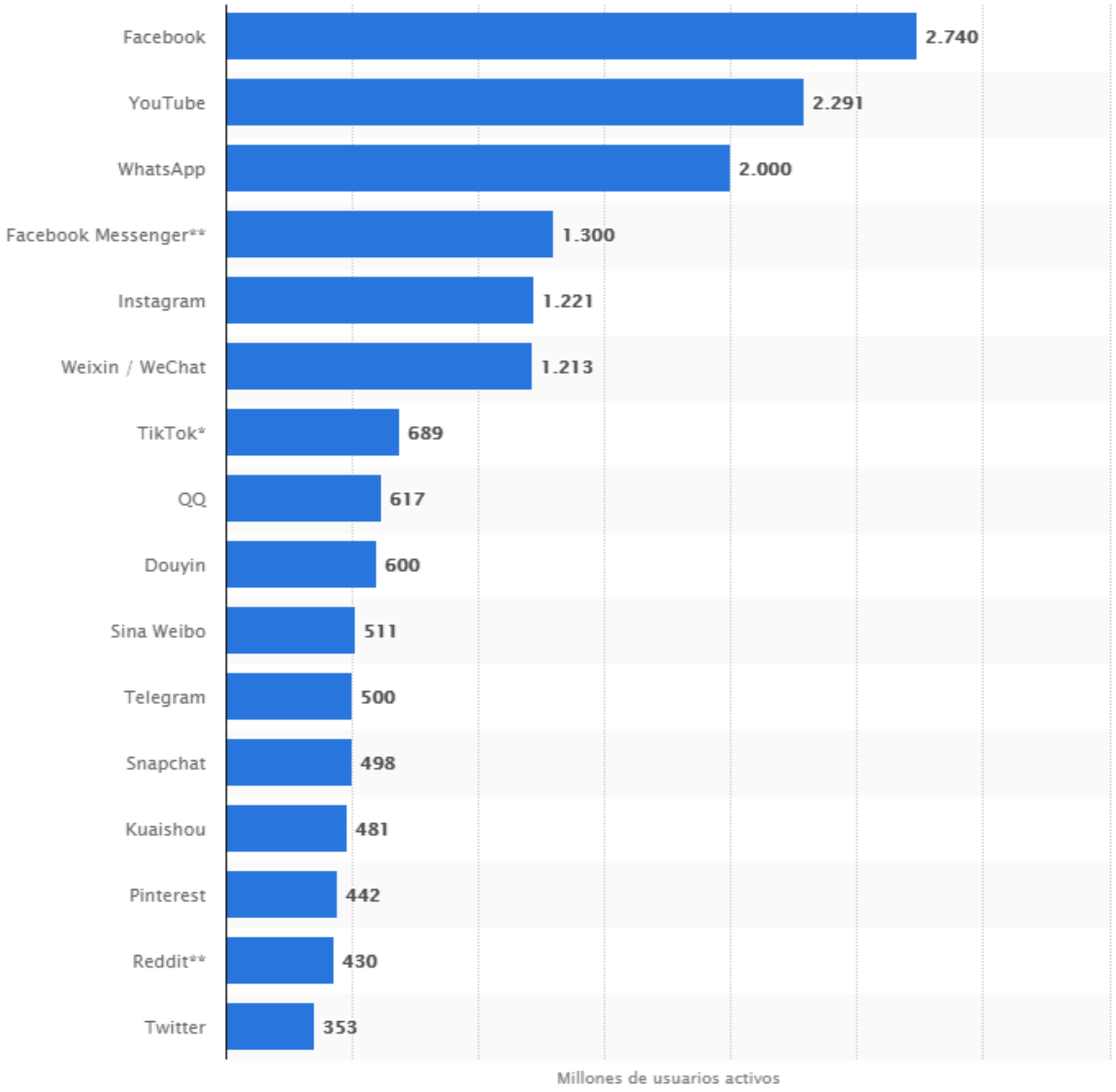

Gráfico 2.1 Usuarios activos en diferentes redes sociales fuente: es.statista.com

Como podemos ver en la *Gráfico 2.1,* Twitter no destaca precisamente por el número de usuarios activos con respecto a otras redes sociales. Entonces, ¿En que destaca Twitter?

Twitter es la red social que mayor impacto tiene en temas políticos. De hecho, el 64% de las personas usa Twitter al menos una vez a la semana para informarse sobre temas políticos.

Para el 75% de los españoles Twitter es muy útil para estar al tanto de la actualidad política, aprecian y consideran que favorece la interacción entre políticos y ciudadanos. En esta línea, siete de cada diez usuarios admiten que “Twitter les permite comprender información, decisiones o declaraciones de primera mano antes que cualquier otro medio de los políticos, y brinda la posibilidad de rastrear en tiempo real y en tiempo real los momentos políticos clave”.

De hecho, en comparación con los usuarios de otras redes sociales, los usuarios de Twitter están más dispuestos a debatir e interactuar con otras ideologías que no son las suyas.

En Twitter, el 76% de los usuarios suele interactuar con políticos y partidos políticos, y el 67% de los usuarios interactúa con signos ideológicos distintos a sus propios políticos o partidos políticos.

El 36 % de los usuarios de Twitter en España se consideran "micro líderes de opinión", personas que tienen la capacidad de influir en los demás en sus opiniones políticas.

El 74% de las personas piensa que se mojará por temas de actualidad, aunque sean controvertidos; el 68%, responden e interactúan con los usuarios; el 62% piensa que los políticos gestionan ellos mismos sus datos personales. [1]

Por esta razón, Twitter se ha convertido en un espacio ideal para expresar opiniones, no solo en torno a la política, podemos ver debates sobre muchos temas todos los días.

2.1.1. Twitter Developer

Una vez que hemos decidido Twitter como nuestra red social, tenemos que pensar en la extracción de datos, para eso necesitamos acceder a la API de Twitter, pero antes de eso será necesario tener una cuenta en Twitter developer y crear una aplicación, esta es la forma que tiene Twitter de controlar quién accede a sus datos y con qué intención.

El proceso de creación de la cuenta y posteriormente la creación de la aplicación es muy sencillo, tan solo debemos tener en cuenta las keys de acceso a nuestra aplicación pues la necesitaremos para poder acceder a la API a partir de nuestra aplicación.

En mi caso utilicé una aplicación anterior que tenía creada:

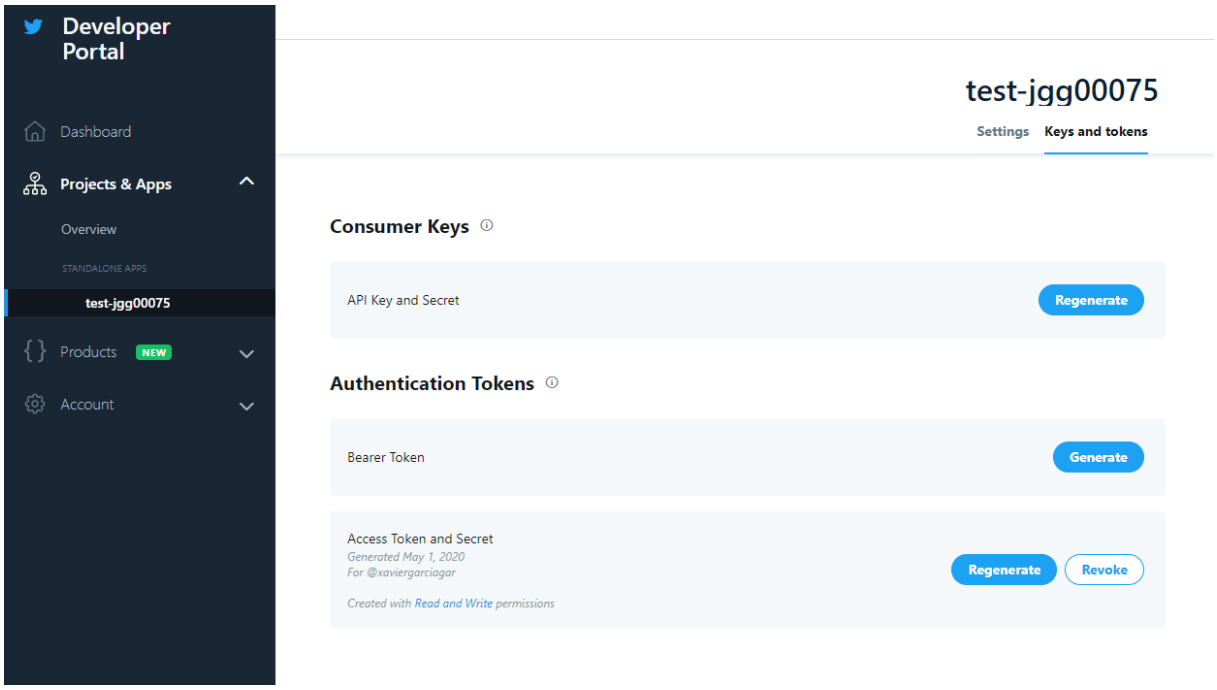

Ilustración 2.1.1 aplicación en Twitter developer

Ahora tan solo debemos guardar esta información en un fichero .key con el nombre de la aplicación.

2.1.2. Twitter API

Una vez tengamos nuestras keys de acceso y autentificación, podemos acceder a la API de Twitter.

Una de las cosas que tenemos que conocer a la hora de trabajar con la API de Twitter es que existen diferentes API y tendremos que escoger una dependiendo de nuestra intención y las funcionalidades que necesitemos.

Además, tenemos que tener en cuenta que Twitter te da acceso gratuito pero limitado, si quieres tener acceso ilimitado hay que usar un plan de pago.

API	Petición	Max. datos por petición (página)	Cada 15 minutos
REST	GET statuses/user_timeline	200 tuits	900 * 200 = 180.000 tuits
REST	GET users_show	1 perfil	1 * 900 = 900 perfiles
REST	GET follwers_list	200 perfiles	200 * 15 = 3.000 perfiles
REST	GET followers_ids	5.000 ids de usuario	5.000 * 15 = 75.000 Ids
Search	GET search/tweets	100 tuits	180 * 100 = 18.000 tuits
Streaming	POST statuses_filter	----	Máximo de 45.000 tuits

Tabla 2.1 Tipos de API y peticiones y sus limitaciones

La API estándar de Twitter y la que usaremos para nuestro proyecto, mediante esta api podremos acceder a ciertos recursos como:

- **Tweets** (mensajes o posts de los usuarios)
- **Usuarios** (actores de la red social)
- **Mensajes directos** (mensajes privados entre usuarios)
- **Listas** (listas de difusión)
- **Tendencias** (temas categorizados)
- **Archivos multimedia** (tanto los usados en los tweets como fotos de perfiles, descripciones...)
- **Ubicaciones** (ubicación del usuario, de una foto…)

Hay que tener en cuenta que el tiempo es una cuestión principal y el cuándo y de cuando queramos cierta información varía en el tipo de API que tenemos que usar, principalmente debemos tener en cuenta las siguientes cuestiones:

- **Rest API**: Ofrece a los desarrolladores el acceso a los datos de Twitter (mencionados anteriormente), independientemente de cuándo sean esos datos, es decir, podemos hacer peticiones de datos antiguos.
- **Search API**: Suministra tuits ajustándose a una petición en concreto, pero solo con un alcance de 7 días.
- **Streaming API**: proporciona tweets prácticamente en tiempo real pudiendo hacer peticiones con diferentes parámetros como la localización, usuarios, palabras clave…

Adicionalmente, Twitter cuenta con otras API para diferentes propósitos, aunque no le daremos uso en nuestro proyecto:

- **Twitter Ads API**: La API de anuncios de Twitter le permite integrarse mediante programación con la plataforma de anuncios de Twitter. Orientada a crear y mantener campañas basadas en objetivos y realizar un seguimiento de los análisis de esa campaña.
- **Twitter for website:** es la API de Twitter destina a funcionalidades web por ejemplo permite insertar el contenido en vivo de Twitter en una aplicación, directamente desde la fuente, insertar Tweets en sus historias y artículos en la web…
- **Labs:** La versión de Labs está diseñada para la comunidad de desarrolladores, brindándoles espacio para experimentar, aportar ideas y probarlas para que puedan agregarse a la API estándar de Twitter o cualquier versión en el futuro.

2.2. Captura de tweets

Para la extracción de tweets usaremos la API estándar de Twitter como hemos mencionado anteriormente, mediante la API Search y la API Streaming para la extracción de tweets en tiempo real, a veces combinaremos ambos métodos, esto dependerá de la cantidad de tweets que queramos capturar, ya que al tener en cuenta las limitaciones de la API podríamos dejar tweets relevantes fuera de nuestro análisis.

Para esta fase he utilizado la aplicación t-hoarder-kit que explicaremos a continuación.

2.3. T-hoarder-kit

T-hoarder-kit fue concebido por Mari Luz Congosto. Doctora en Telemática por la Universidad Carlos III y licenciada en Informática por la Universidad Politécnica de Madrid como un entorno de trabajo mínimo que pudiera recopilar, procesar y exportar la información de Twitter para posteriormente presentarla en grafos y poder analizarla. [8]

T-hoarder-kit está programado en Python en su gran mayoría y está diseñado para UNIX como sistema operativo. Fue diseñado con una arquitectura minimalista que evita la dependencia de otros paquetes software y además cuenta con una estructura de carpetas muy sencilla:

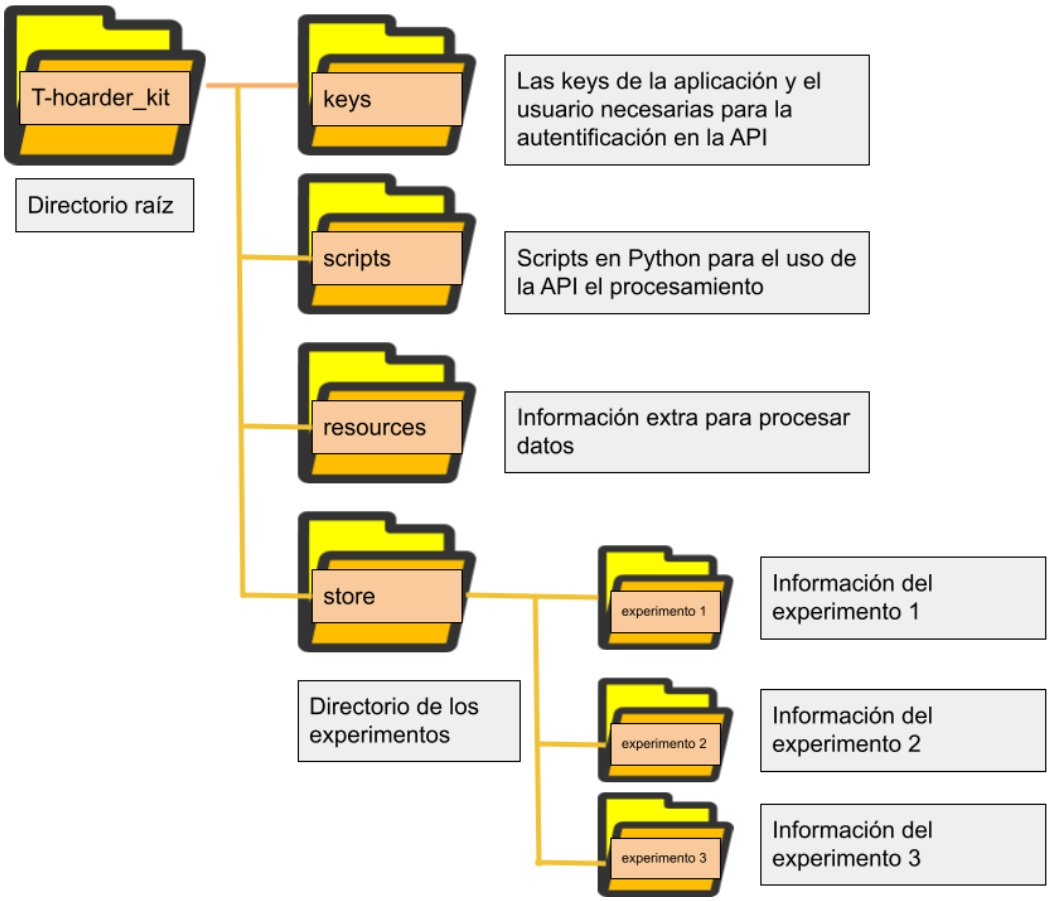

Imagen 2.3 Jerarquía de directorios en T-hoarder-kit

La mayoría de los enfoques que usan estos entornos de trabajo a la hora de almacenar tweets se basan en grandes bases de datos que se pueden procesar más tarde. Por el contrario, T-Hoarder-kit que almacena esta información en un sistema de archivos sencillo que se puede se accede más tarde para su análisis fuera de línea.

En este sentido T-Hoarder-kit destaca por su accesibilidad con respecto a su competencia, es un entorno sencillo, simple y no necesita de una gran capacidad de computación, por lo que podríamos usarlo en una Rasberry Pi.

2.3.1. Captura de tweets en T-hoarder

Cuando vayamos a extraer datos en T-Hoarder debemos de tener en cuenta el método que queramos usar como hemos comentado anteriormente. En este TFG como nuestra intención es analizar las tendencias, usaremos la opción 4 de “Get tweets on real time” aunque también podemos usar la opción 3 “Make a query on Twitter” como podemos observar en la *Imagen 2.3.1*, la opción 4 trabaja con la Streaming API y la opción 3 con la Search API.

Imagen 2.3.1 Menú principal de T-hoarder

Una vez empecemos nuestro proceso de extracción de tweets, deberemos esperar un tiempo considerable que dependerá de la cantidad de tweets que queramos extraer para nuestro análisis, en este trabajo hemos capturado tweets durante unas 4 horas, ya que considero que es un tiempo suficiente para extraer al menos a los tweets más relevantes de una tendencia, ya que suelen ser tweets que son muy interactuados y es difícil que en un periodo de 4 horas un tweet relevante no haya usuarios que interactúen con él.

Tras esperar el tiempo que creamos preciso, cancelaremos el proceso y obtendremos un archivo con las siguientes variables:

- Id Tweet: es el identificador del tweet
- Date: fecha en la que se escribió el tweet
- Autor: usuario que interactuó con el tweet

- Text: texto del tweet
- App: aplicación con la que el usuario interactúo con el tweet
- Id User: identificador del usuario
- Followers: número de seguidores del usuario
- Following: número de usuarios que sigue el usuario
- Stauses Location: localización del usuario.
- Urls: dirección del tweet.
- Geolocation: localización del usuario cuando escribió el tweet.
- Name: nombre del usuario .
- Description: descripción del usuario.
- url_media: enlace del contenido multimedia.
- type media:tipo de contenido multimedia.
- quoted relation: tipo de interacción.
- replied_id: la contestación del id.
- user replied: el usuario que ha sido contestado.
- retweeted_id: el id del retuit.
- user retweeted: usuario que ha sido retuiteado.
- quoted_id: el id del usuario mencionado.
- user quoted: el usuario que ha sido mencionado.
- first HT: primer hastag del tweet.
- Lang: idioma del tweet.
- Link: enlace del tweet.

A partir de estas variables crearemos relaciones para formar los grafos.

2.4. Minería de Opinión

El análisis de sentimiento o minería de opinión trata de evaluar las emociones extrayéndolas de un texto. Es una tarea que combina técnicas de

minería de texto y Procesamiento del Lenguaje Natural (PLN). Desde la aparición de las redes sociales la minería de opinión se convirtió en un reto para analizar a cada usuario, determinar qué candidato político pretende votar, si está de acuerdo con la nuevas medidas de sanidad o simplemente para medir la aceptación de una campaña publicitaria.

Las utilidades de extraer el sentimiento de los tweets son muy diversas, por eso he considerado interesante añadir un análisis de sentimiento sobre los tuits para tener más información que analizar y facilitar el análisis de una tendencia en Twitter.

Una de las principales aplicaciones consiste en determinar la polaridad de las opiniones a nivel de documento, frase o característica. De esta manera, podemos clasificar binariamente las opiniones en positivas o negativas.

Por otra parte, y a pesar de que las opiniones y comentarios compartidos en Internet no tienen restricción en cuanto al idioma utilizado, la gran mayoría de la investigación llevada a cabo relacionada con la minería de opiniones se centra casi exclusivamente en textos escritos en inglés. Sin embargo, cada vez son más los textos subjetivos que utilizan otros idiomas. [12]

Antes de formar nuestros archivos gdf para crear nuestros grafos, añadiremos una nueva variable que exportar para nuestros tweets, la variable sentimiento.

Este proceso lo he realizado mediante un script de Python, que sea capaz de leer un archivo de entrada con las variables del punto anterior, y a partir del texto de un tuit, hacer análisis de opinión dando un valor de 0 o 1 si considera que el tweet expresa una opinión negativa o positiva respectivamente.

Para este proceso he optado por un modelo de predicción que parte de un corpus de entrenamiento, en este caso he partido del corpus COST [2] , formado por 17.317 tweets con emoticonos positivos y 17.317 emoticonos negativos, ya que este corpus se centraba en el análisis de sentimiento en una serie de tweets donde los emoticonos tenían un gran peso en su análisis, he considerado prudente suprimir estos emojis con

el fin de evitar falsos positivos o negativos, ya que cuando trabajemos con los datos reales estos emojis no serán tan significativos.

Para construir nuestro modelo he utilizado la librería *sklearn* [10] de Python, he utilizado un modelo predictivo o de regresión, estos modelos pretender obtener la representación de la relación entre dos (o más) variables, a través de una expresión lógico-matemática que, además de resumir esta relación, permite hacer predicciones de los valores que asumirá una de las dos variables (una variable dependiente, variable de respuesta, Criterio o Y) los valores del otro (el que actúa como un predictor explicativo, independiente o X), en nuestro caso los tweets son la variable X y su sentimiento asociado, la variable Y.

Antes de empezar a entrenar nuestro modelo debemos crear la matriz TFIDF, (Term Frequency Inverse Document Frequency). Es una estadística numérica que se basa en encontrar el documento más relevante para cierto término dentro de una colección de documentos.

En nuestro contexto lo usaremos para hallar la relevancia de los términos para una colección de tweets a la hora de predecir su sentimiento.

Para vecotrizar nuestros datos de entrada en una matriz TF-IDF hemos usado el método *TfidfVectorizer* de *Sklearn*, en nuestro caso solo hemos usado dos parámetros:

- **tokenizer**: El método que nos hará el preprocesamiento del texto, este se basa en:
  - Normalizar el texto en minúsculas.
  - Suprimir los signos de puntuación.
  - Suprimir los usuarios mencionados.
  - Suprimir los hastag del tweet.
  - Suprimir los emojis.
- **ngram-range(1,2)**: Secuencia contigua de n elementos que considera a la hora de realizar la matriz tfidf, en este caso significa que solo usará unigramas y bigramas, es decir n-gramas de tamaño 1 y 2.

El siguiente paso será dividir el conjunto de datos entre los datos de entrenamiento y los de testeo. Para eso usamos el método *train_test_split* de *Sklearn*, al cual debemos pasarle como parámetros el conjunto de datos X e Y. En esta función solo debemos de tener en cuenta adicionalmente otros 2 parámetros:

- **test-size**: es el tamaño que usamos para el conjunto test sobre el total de los datos, normalmente en estos ámbitos se suele usar un 0.3 esto equivale a que un 30% se usará para el conjunto *test* y un 70% se usará para el conjunto de entrenamiento. Aunque de normal se usa un 30% si consideramos que tenemos un conjunto de datos lo suficiente grande podríamos usar un 20 % para el tamaño de testeo.
- **Random state**: Es una semilla con la cual el algoritmo puede controlar el generador de números aleatorios esto influye en la manera en la que se dividen los datos, y siempre que utilicemos la misma semilla podremos reproducir los mismos resultados. Normalmente se busca optimizar esta semilla para encontrar los mejores resultados posibles, aunque en nuestro modelo este dato no ha sido influyente.

Nuestro siguiente paso será construir un clasificador para nuestro modelo, de nuevo mediante la librería Sklearn. Para ello usaremos el método LinearSVC.

El objetivo de LinearSVC (Support Vector Classifier) es ajustar los datos que proporciona, devolver un hiperplano "ideal" y dividir o clasificar sus datos. Este hiperplano no es más que una línea (gráficamente hablando) que divide nuestro conjunto de datos en n planos diferentes (tal y como podemos observar en la Imagen 2.4, en este caso el hiperplano, divide el conjunto de datos de la clase “azul” y de la clase “roja” en dos planos diferentes) por cada n clase que tengamos que predecir los datos se dividirán en n grupos diferentes.

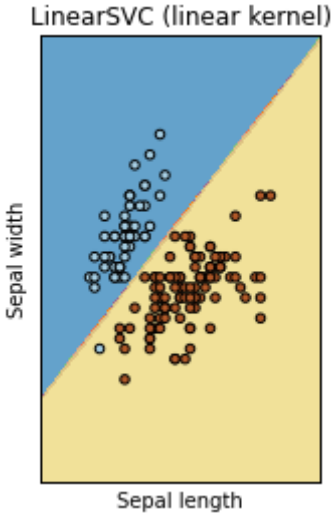

Imagen 2.4 resultados de un clasificador LinearSVC

Para terminar nuestro modelo tendremos que ajustar el modelo a los datos con el método *fit* y posteriormente ya podremos aplicar el método *predict* de nuestro clasificador (ambos métodos lo usamos por defecto, sin parámetros). Con este método nuestro modelo intentará predecir etiquetas de clase para las muestras en X, en este caso el sentimiento de los tweets.

Una vez entrenado nuestro modelo, obtenemos una tabla de resultados *Imagen 2.4.1*, Estas son las métricas utilizadas para evaluar el modelo entrenado:

- **recall** es la relación tp / (tp + fn) donde *tp* es el número de verdaderos positivos y *fn* el número de falsos negativos.
- **f1-score** puede interpretar como un promedio ponderado de la precisión y el *recall*. Cuanto más se acerque a la puntuación 1, mejo.
- **support** es el número de apariciones de cada clase en *y_true*, esto significa el verdadero número de veces que aparece esa clase en nuestro modelo sin contar los falsos positivos o negativos.

Una vez entrenado nuestro modelo obtenemos un valor de un 75% de precisión, que consideramos aceptable para nuestro propósito.

	precision	recall	f1-score	support
0	0.73	0.78	0.76	5187
1	0.77	0.72	0.74	5198
accuracy			0.75	10385
macro avg	0.75	0.75	0.75	10385
weighted avg	0.75	0.75	0.75	10385

Imagen 2.4.1 Matriz de clasificación

Cuando ya tenemos nuestro modelo entrenado simplemente volvemos a pasar nuestro *Tokenizer* sobre el nuevo conjunto de datos.

En este proceso simplemente tendremos que leer un archivo de entrada con tweets a predecir, procesar el texto y utilizar nuestro modelo entrenado para predecir los valores de sentimiento del tweet y posteriormente guardar estos valores en nuestro archivo.

2.5. Exportar datos para crear un grafo

Tras añadir nuestra variable sentimiento en los tweets, simplemente tendremos que generar un archivo gdf a partir de la información recopilada. Este paso es muy sencillo, tan solo tendremos que usar T-Hoarder-kit de nuevo, esta vez usaremos la opción 6 de la *Imagen 3.2.*

Para importar la variable sentimiento en Gephi, usaremos la función *exportData*() de nuestro código, este método simplemente exporta la variable sentimiento en un formato Excel, que luego podremos importar desde la tabla de datos de Gephi gracias al plugin *ImportPlugin.*

2.6. Grafos

Para la fase de importar, crear, procesar y exportar grafos usaremos la aplicación Gephi, es una herramienta open-source desarrollada en Java para visualizar y analizar grandes gráficos de red. [9]

Gephi usa un motor de renderizado 3D para mostrar gráficos en tiempo real y permite explorar, analizar, filtrar, agrupar, manipular y exportar diversos tipos de gráficos.

Las razones principales por las que he usado Gephi, es que es un software gratuito, accesible y a pesar de ello sigue siendo potente para la creación de grafos, además cuenta con un sencillo sistema de plugins que fácilmente podremos añadir y ampliar sus funcionalidades. Todavía en la actualidad, Gehpi se usa en una gran cantidad de proyectos relacionados con las redes sociales.

En la creación de grafos debemos tener claras dos preguntas, ¿qué buscamos representar? y ¿cuál es la manera más eficiente para nuestro problema?

Hay muchos algoritmos que nos permiten procesar los grafos de multitud de formas, algunos de los enfoques de análisis más conocidos son:

- **Análisis de ruta o camino**: Analiza las características de la ruta entre dos nodos.
- **Análisis de conectividad**: se utiliza para comprobar la "fuerza" de la relación entre dos nodos, permitiendo la detección de relaciones débiles o vulnerables. Un caso de uso para este tipo de análisis es detectar cuellos de botella dentro de una red.
- **Análisis de comunidad**: Este método de análisis consiste en detectar la comunidad de nodos en función de la distancia y densidad del gráfico, de modo que cada comunidad contenga nodos con características comunes o similares.

- **Análisis de centralidad**: se centra en conocer la correlación de nodos en el grafo, es decir, analizar la influencia de un nodo. Un ejemplo común es detectar a las personas más influyentes en las redes sociales.
- **Isomorfismo de subgrafos**: Analizan un grafo para obtener el patrón estructural, a fin de encontrar el patrón más repetitivo. La detección de patrones es un método de detección estructural que nos da algunas características tipo.

Debemos tener en cuenta el tipo de grafos que vamos a estudiar, forman redes sociales.

Una Red Social es la estructura social que facilita la comunicación entre un grupo de actores (individuos u organizaciones) que están relacionados de alguna manera (es decir, por intereses comunes, valores compartidos, intercambios financieros, amistad, desagrado, etc.).

Por ejemplo, una red de amigos forma una red social. Pero las redes sociales operan en muchos más niveles, desde las relaciones familiares y la propagación de la enfermedad hasta el nivel de las estrategias de la empresa, los movimientos sociales o incluso las naciones.

Además, la investigación en muchas áreas científicas ha demostrado que las redes sociales son importantes cuando estudiamos la forma en que se resuelven los problemas, se propagan las enfermedades, se administran las organizaciones y el grado en que las personas logran sus objetivos.

El Análisis de Redes Sociales (SNA) es una combinación de Sociología y Matemáticas, compuesta de varias técnicas interdisciplinarias para el estudio de dichas redes sociales.

Los investigadores de SNA conceptualizan las relaciones sociales en términos de nodos y bordes (enlaces) en gráficos matemáticos. Los nodos representan a los actores individuales dentro de las redes, mientras que los bordes visualizan las relaciones entre esos actores. [12]

Los resultados de un análisis de redes sociales se pueden utilizar para:

- Identificar las personas, equipos y unidades que desempeñan roles centrales.
- Discernir fallas de información, 7 cuellos de botella, 8 agujeros estructurales, así como individuos, equipos y unidades aislados.
- Busque oportunidades para acelerar los flujos de conocimiento a través de funciones y límites organizacionales.
- Fortalecer la eficiencia y eficacia de la comunicación formal existente.
- Sensibilizar y reflexionar sobre la importancia de las redes informales y formas de mejorar su desempeño organizacional.
- Mejorar la innovación y el aprendizaje.
- Refinar estrategias. [13]

2.6.2. Detencción de comunidades

En Twitter, como cualquier red social, cuando los usuarios interactúan entre sí, se crean comunidades, estas son un subconjunto de usuarios fuertemente cohesionados, lo que resulta muy útil en el análisis de marketing, economía, sociología, minería de datos y, por su puesto, en redes sociales.

La detencción de comunidades es tremendamente útil ya sea para encontrar un nicho de mercado, un grupo de opinión o cualquier otro tipo de conjunto de usuarios que mantengan algún tipo de cohesión.

2.6.3. Componente gigante

La componente gigante es un subgrafo conexo que aglutine la gran mayoría de nodos, es interesante comparar las propiedades de la red global y de este componente. La componente gigante pretende eliminar aquellos nodos más aislados de nuestra red y por tanto menos relevantes.

En grafos o redes dirigidas, podemos diferenciar dos tipos de componentes conexos:

- Cuando la componente gigante está compuesta con una fracción pequeña de los nodos.
- Cuando la componente gigante incorpora todas las componentes aisladas y la red queda con una única componente conexa.

Por lo general en la gran mayoría de redes sociales aparece una componente gigante. [3, Newman].

2.6.4. Modularidad

La modularidad Q es una función de calidad que mide la calidad de una partición concreta de una red en comunidades:

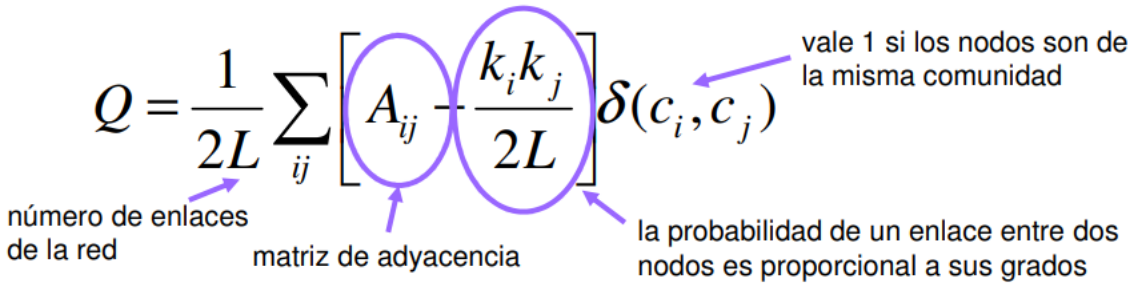

Imagen 2.6.4 Formula Modularidad

La idea básica es que la red muestra una estructura modular coherente si el número de enlaces entre comunidades es menor que el esperado en una red aleatoria Q ∈ [-1,1]. Cuanto mayor es su valor, mejor es la partición, es decir, las comunidades encontradas están densamente conectadas internamente (hay más enlaces de los que cabría esperar aleatoriamente) y dispersamente conectadas entre sí. En una red aleatoria, Q=0. En la práctica, un modularidad de 0.3 es un buen valor. [4]

2.6.5. Distribución

Para facilitar la visualización de los nodos y aristas de una red existen algoritmos de distribución (layouts), esto se basa en el principio de que una misma red se puede representar gráficamente de diversas formas, y dependiendo de nuestro propósito a la hora de analizar un grafo, una representación puede sernos mucho más útil ya que puede destacar con mayor facilidad aquellas características que nos interesen mostrar.

Podríamos clasificar los algoritmos de distribución en dos tipos:

- **Algoritmos puros**: simulan fuerzas “naturales”, como las fuerzas gravitatorias, simulan pesos, muelles a la hora de distribuir los nodos y aristas. Estos algoritmos están bien definidos y no suelen aceptar modificaciones.
- **Algoritmos de distribución fina o modificación**: Son algoritmos que nos permiten hacer algunas alteraciones de forma que podemos personalizar o parametrizar para nuestro entorno.

3. Experimentos

Durante los siguientes experimentos seguiremos la metodología descrita anteriormente, cabe destacar la inclusión de la nube de palabras como herramienta de apoyo para el análisis.

La nube de palabra o de etiquetas, es una imagen donde las etiquetas pueden aparecer en diferentes colores y tamaños formando una figura abstracta o no.

Las etiquetas más grandes y con colores más fuertes son las que aparecen con más frecuencia y las que aparecen con una frecuencia menor van a figurar en la imagen más pequeñas y en colores más apagados. [7]

Esto nos servirá de apoyo para caracterizar el contenido de los tweets.

3.1. Experimento Placido Domingo

En este experimento analizaremos la tendencia “Placido Domingo”, este hastag surgió durante 2021-03-25 cuando Placido Domingo fue vitoreado durante su última actuación tras su vuelta a los escenarios, de los cuales se retiró temporalmente debido a las acusaciones de agresiones sexuales.

Multitud de tuiteros expresaron su opinión y se formó un debate en torno a ello, en este experimento, analizaremos las diferentes comunidades de opinión que se formaron y su relación entre ellas.

Para la recopilación de tuits de este experimento, hemos utilizado la opción de T-hoarder-kit, para la recopilación de tweets en vivo, de aquellos tuits que aparece la palabra Placido Domingo.

El archivo gdf lo generamos a partir de la opción 6 como podemos ver en la *Imagen 2.3,* esta vez elegimos la opción de formar un grafo a partir de la relación RT.

El RT en Twitter, funciona como una manera de compartir la información a tus seguidores, dando a entender que estás de acuerdo con el mensaje original. Es por esto, que es el mejor tipo de relación que funciona, a la hora de detectar comunidades, ya que otras relaciones como las respuestas o las citaciones a un tuit, no tienen por qué ser entre usuarios que compartan la misma opinión, por lo que las comunidades creadas a partir de la relación de RT, tienen más sentido que en el resto de casos, sobre todo si lo que buscamos es agrupar en comunidades de opinión.

3.1.2. Componente Gigante

En este filtro descartamos nodos y aristas en función de lo conexos que estén con el resto de nodos, es decir se descartan aquellos nodos que estén más desolados.

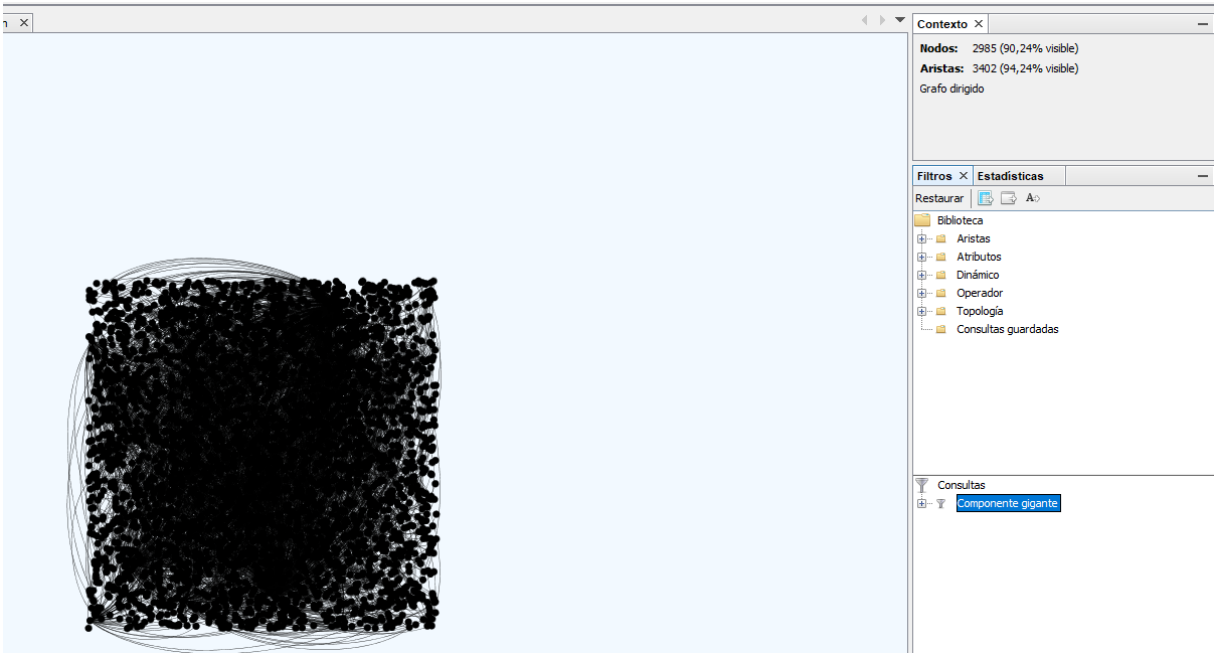

Imagen 3.1.2 Componente Gigante

Tal y como podemos ver en la *Imagen 3.1.2* este sería el resultado de aplicar la componente gigante sobre los datos recopilados en el hastag Placido Domingo, en este caso hemos descartado un 3,57% de los nodos y un 5,76% de las aristas.

Este filtro tan solo pretende deshacerse de los nodos poco relevantes para luego aplicar una serie de filtros, por lo que aún no tenemos un resultado visual.

3.1.3. Modularidad

Para calcular la modularidad en Gephi por defecto usa el algoritmo *de Louvain que está basado en la modularidad de los datos. Este realiza una evaluación del conjunto de datos y compara la densidad de aristas que están presentes dentro o fuera de la comunidad. Al optimizar este valor de iteración se obtiene un estimado de agrupación de los nodos que pertenecen a una red*. [5]

A la hora de utilizar este algoritmo debemos tener en cuenta que Gephi tiene un parámetro de entrada “Resolución”, este parámetro por defecto será 1, esto significa

que se utilizará el algoritmo de Louvain estándar, sin embargo, si disminuimos este valor por debajo de 1, obtendremos más comunidades, pero más pequeñas y por consiguiente si aumentamos este valor por encima de 1, obtendremos menos comunidades, pero más grandes.

Durante este experimento he usado el valor por defecto pues he obtenido un número de comunidades que he considerado suficiente para el análisis que vamos a realizar.

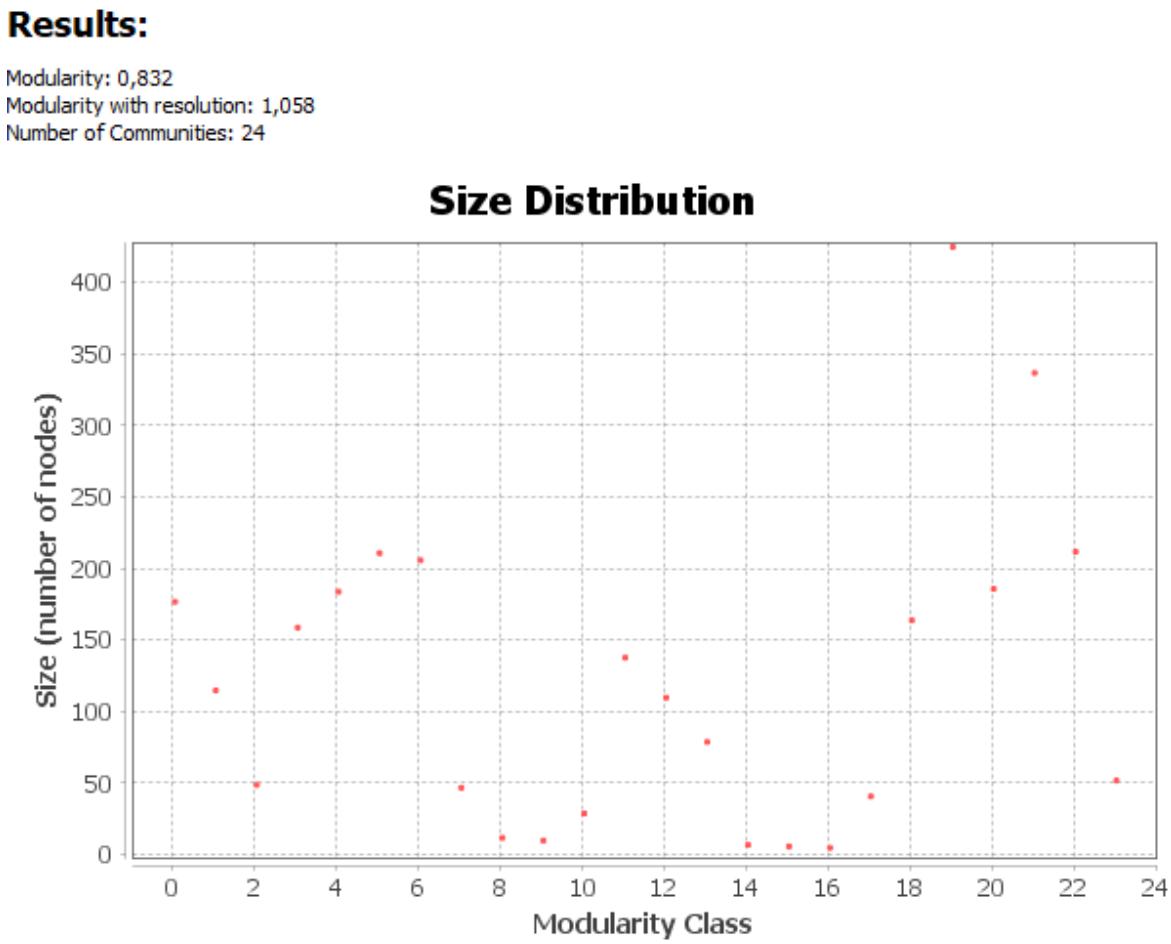

Imagen 3.1.3 Distribución de las clases

Una vez ejecutemos el algoritmo de modularidad en Gephi nos presenta los resultados distribuidos como podemos observar en la *Imagen 3.1.3*, en este caso hemos obtenido 22 comunidades.

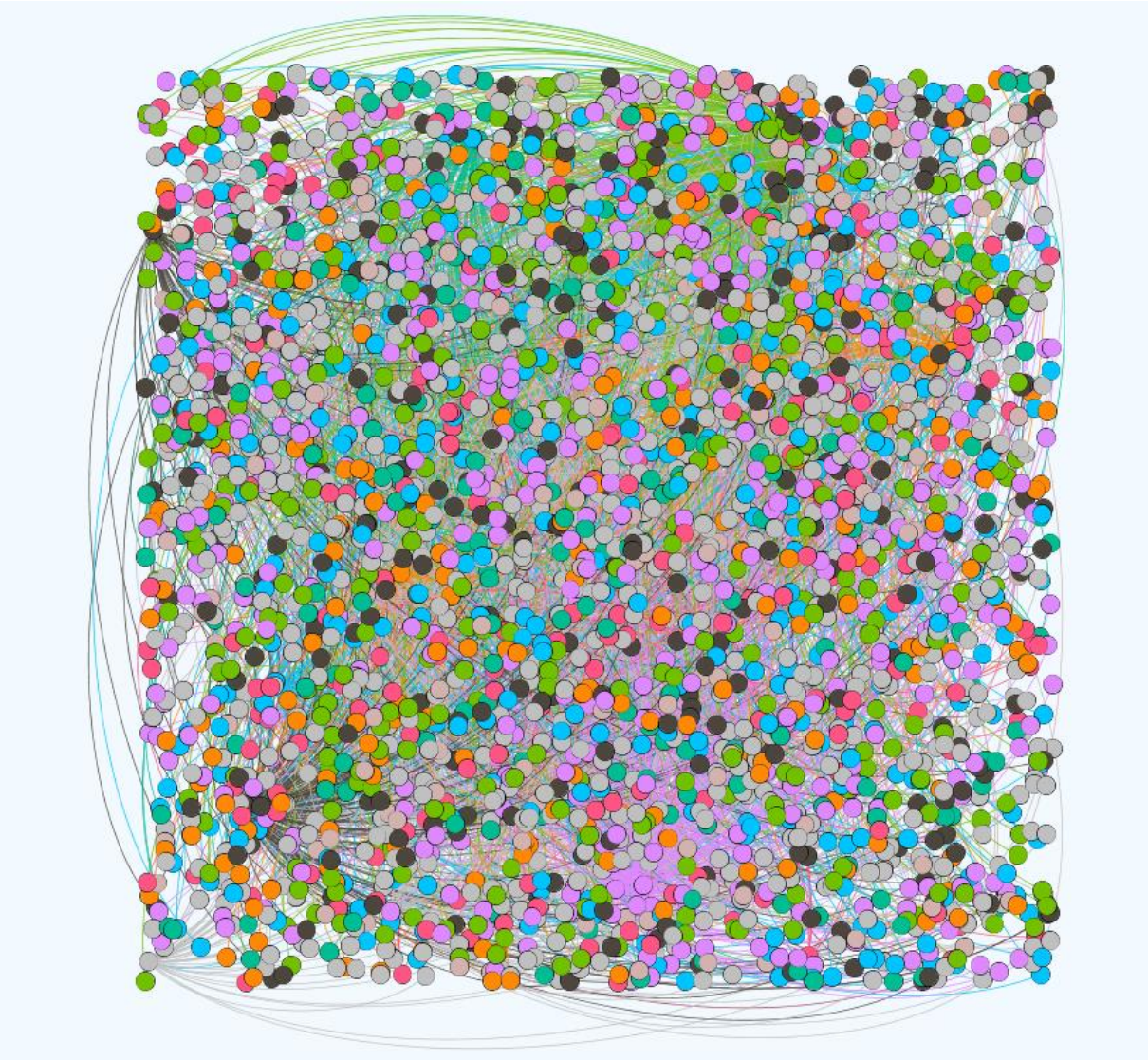

Imagen 3.1.3.1 grafo tras aplicar color por modularidad a los nodos

El siguiente paso sería aplicar el color de los nodos por “Modularity Class” que no deja de ser el color de la comunidad que hemos asociado a cada clase. En la *Imagen 3.1.3.1* podemos observar los nodos diferenciados por el color de su comunidad.

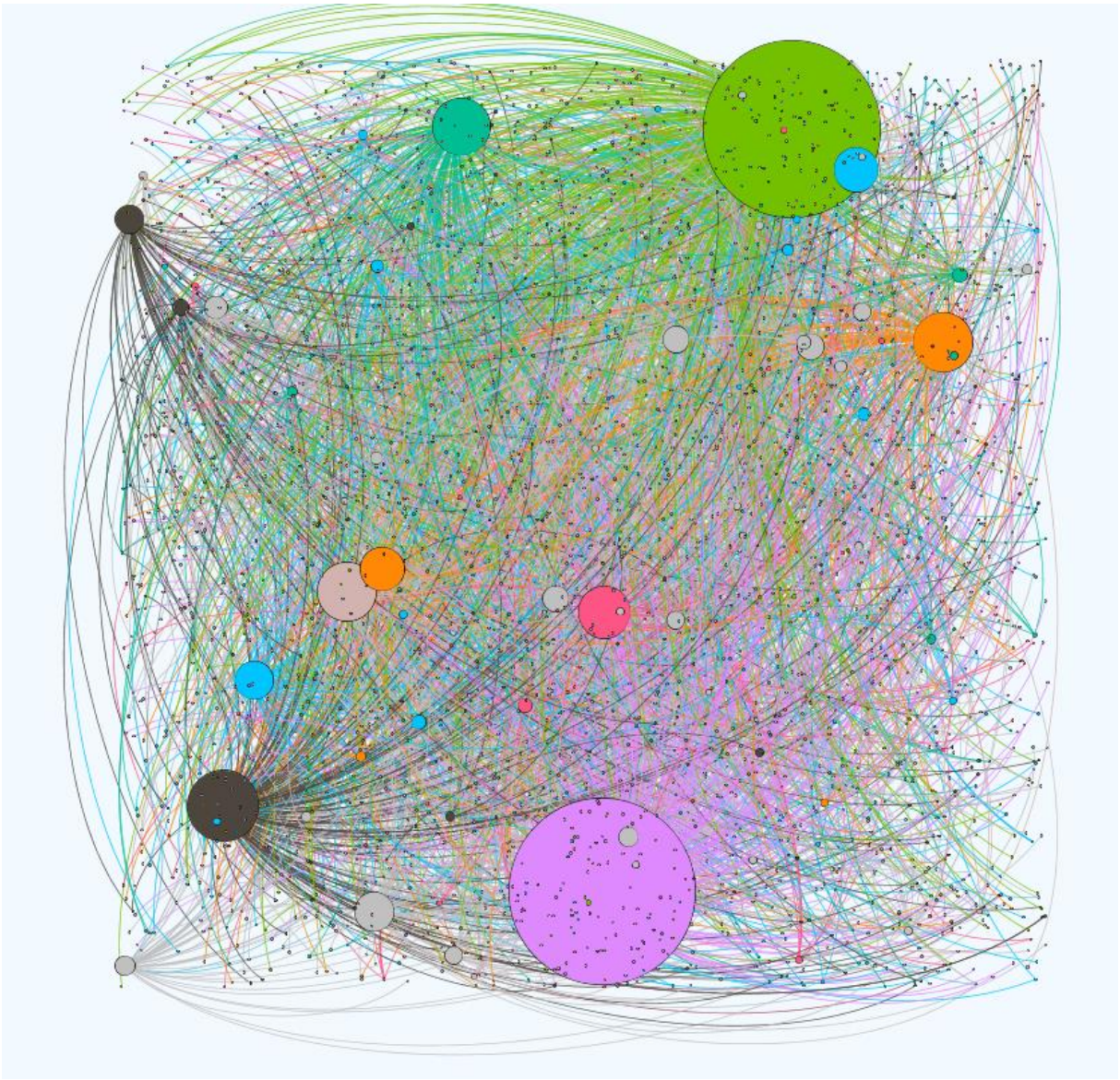

Imagen 3.1.3.2 grafo tras aplicar tamaño proporcional a los nodos

Posteriormente queremos dar un tamaño a los nodos proporcional a su importancia, en este caso hemos configurado para que Gephi de un tamaño proporcional al nodo dependiendo del grado de entrada, con una escala de 1-100. En la *Imagen 3.1.3.1* podemos ver el resultado de aplicar esta modificación.

3.1.4. Distribución

ForceAtlas 2 es el algoritmo que he elegido para la distribución, este afectará visualmente el conjunto de comunidades. ForceAtlas 2 es un algoritmo de vector de fuerza que optimiza la eficiencia sobre la precisión, gracias a esto permite una convergencia rápida en su ejecución. Esta característica permite representar redes de gran tamaño en cortos tiempos, además la ejecución se para de forma manual por lo que el usuario decide cuando parar en función de la precisión que busque, pero por lo general no necesita mucho tiempo para obtener unos resultados óptimos.

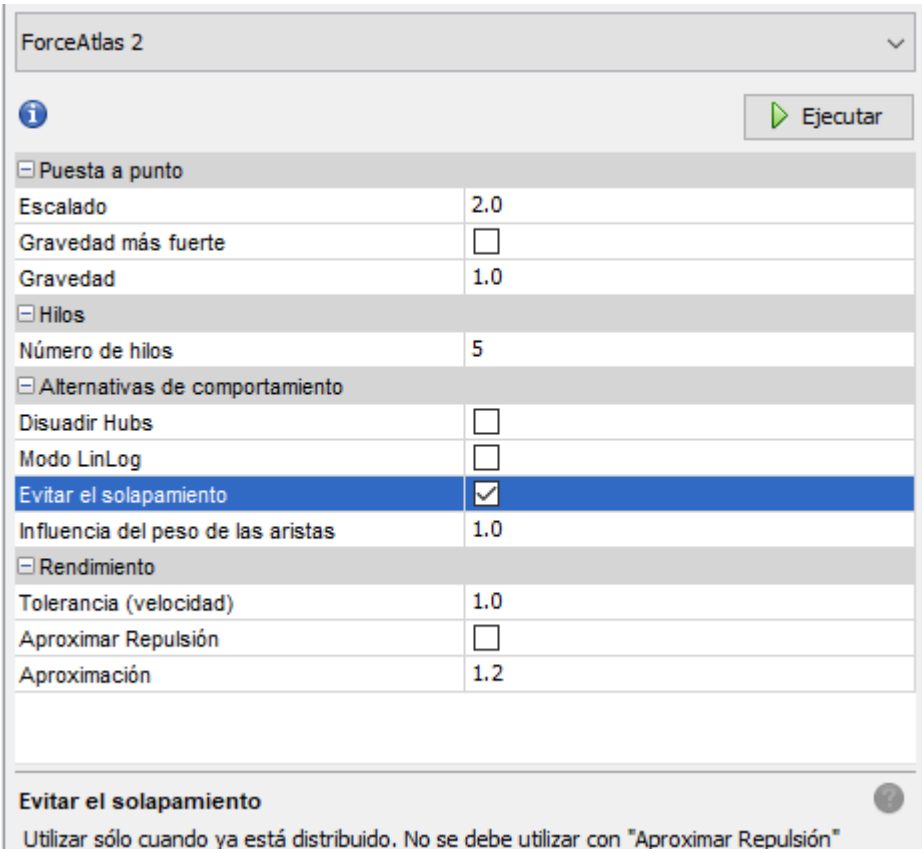

Imagen 3.1.4 parámetros ForceAtlas 2

En nuestro caso Aplicamos ForceAtlas 2 con los parámetros que observamos en la *Imagen 3.1.4,* la opción de evitar solapamiento nos sirve para mostrar unos resultados más claros evitando que los nodos se junten demasiado.

3.1.5. Ajustes de apariencia

Para darle una apreciación correcta a los nodos en nuestro grafo vamos a cambiar algunas de las configuraciones como podemos observar en *Imagen 3.1.5*, en este caso las variables que he cambiado fueron las siguientes:

- Nodos → color del borde = “parent”
- Nodos → activamos la opacidad del nodo
- Etiquetas → activamos Mostrar etiquetas
- Etiquetas → activamos tamaño proporcional
- Etiquetas → seleccionamos Arial 8 sin Formato como fuente

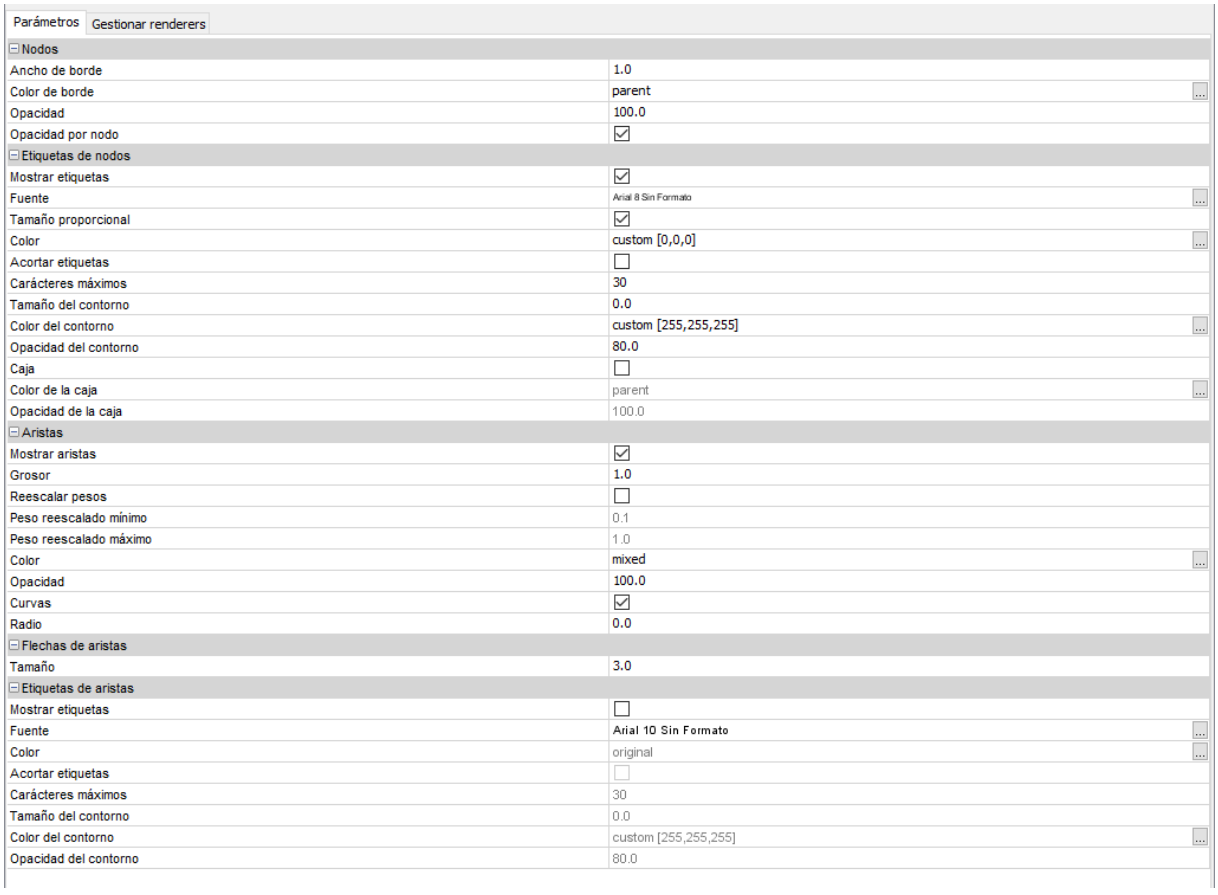

Imagen 3.1.5 Ajustes de Apariencia

Después de aplicar estas transformaciones, lo ideal sería asignar el color manualmente a algunas comunidades para que no haya confusiones con aquellas que estén cerca.

3.1.6. Resultado y Análisis

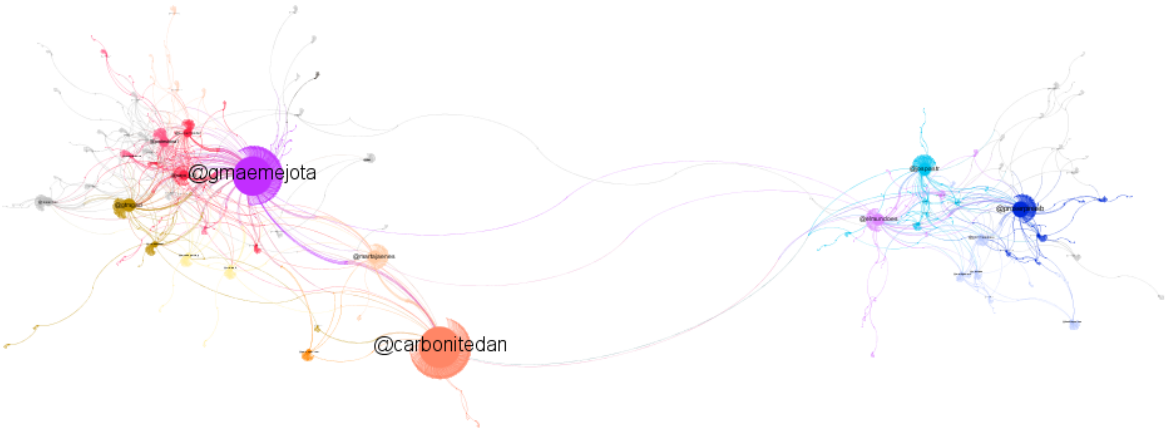

Imagen 3.1.6 Grafo de la tendencia Palcido Domingo distribuido por comunidades

En la *Imagen 3.1.6,* podemos ver el resultado final de nuestro grafo, antes de analizar el resultado final me gustaría comentar algunos puntos importantes:

Hay que recordar que el tamaño de los nodos es proporcional al grado de entrada, que en este caso es el número de RTs recibidos, de esta manera destacan los perfiles con más impacto.

Los nodos con conexiones más comunes se atraen y los que tienen menos se repelen. Esto hace que la distancia entre los nodos sea inversamente proporcional a las conexiones comunes, es decir, cuanto mayor sea la distancia en el grafo menor es la conexión que los une. En este contexto que estamos segregando por comunidades de opinión, esto nos indica que son opiniones distantes.

La distribución de grupos nos muestra las macro agrupaciones en el bloque. Cuando se llega a un consenso solo hay un bloque y cuando hay más de uno es cuando no se llega a un consenso. En este caso, hay dos bloques, que representan la estructura de un argumento a favor de un grupo y oposición al otro. Cuando hay dos grupos muy separados y con conexiones débiles entre ellos, generalmente representa una polarización en un tema de opinión.

En este caso tenemos dos grupos, uno formado por las comunidades gris, azul y azul oscuro y otro grupo muy separado formado por las comunidades morado, púrpura, rojo y amarillo.

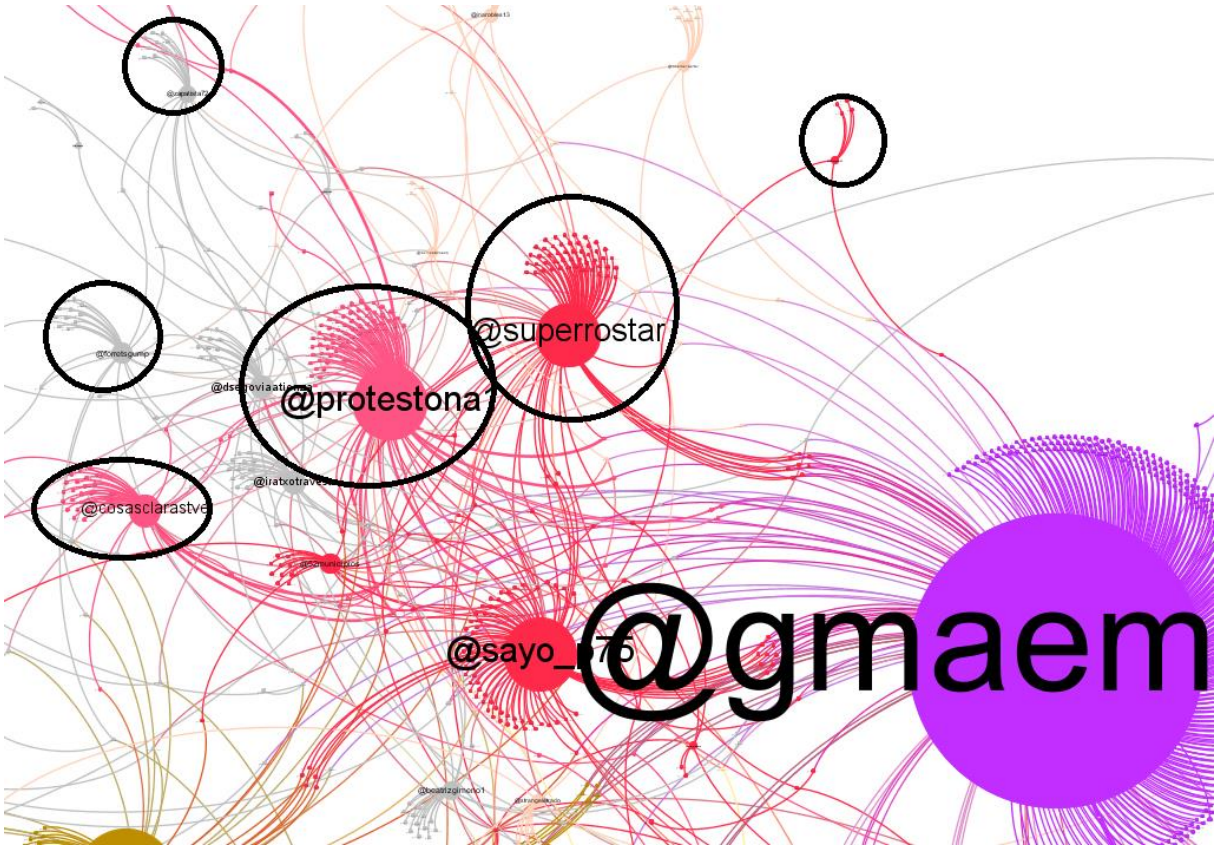

Imagen 3.1.6.1 colas de cometa

El coeficiente de agrupamiento se puede apreciar visualmente en el grafo. Por ejemplo, en los usuarios @protestona1, @superrostar que vemos señalados por círculos negros en la *Imagen 3.1.6.1,* podemos observar algo parecido a una “cola de cometa” esto significa que hay muchos usuarios que han retuiteado pero que no están conectados entre sí. Esto en el coeficiente de clustering se conoce como un caso C=0, entendiendo C como el número de conexiones entre sus vecinos, dividido entre el número de conexiones posibles.

Algunos usuarios pueden retuitear a varias comunidades, la asignación a una comunidad depende de a qué grupo haya retuiteado más y en caso de igualdad de RT’s, la asignación en una comunidad es aleatoria. Estos usuarios forman “autopistas”

que conectan a nodos que han tenido mucho impacto (tal y como podemos ver en la *Imagen 3.1.6.3)*.

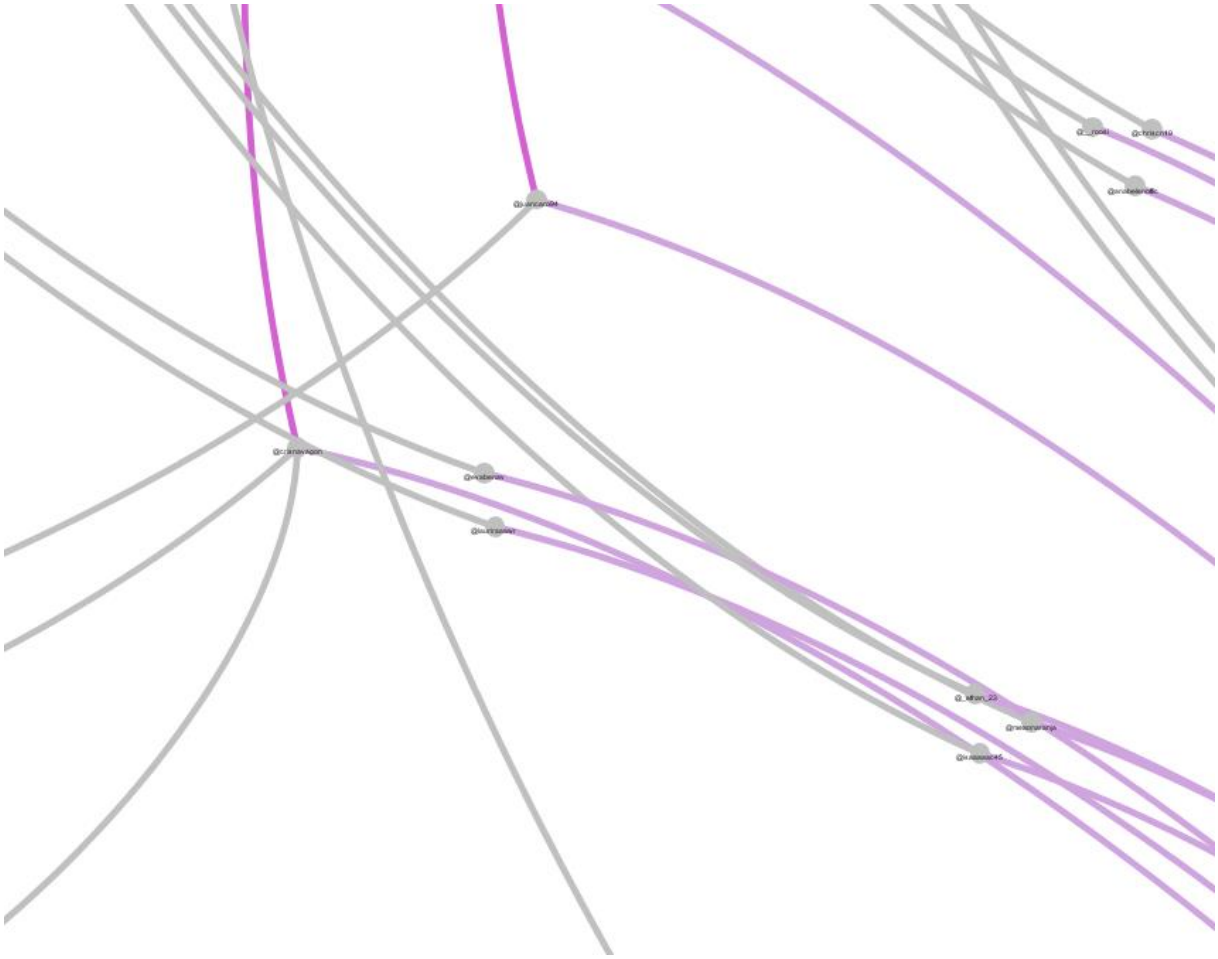

Imagen 3.1.6.2 usuarios que hacen RT a varias comunidades

Cuando un usuario hace RT's a varias comunidades, sus aristas serán de diferente color, tal y como podemos apreciar en la *Imagen 3.1.6.2*.

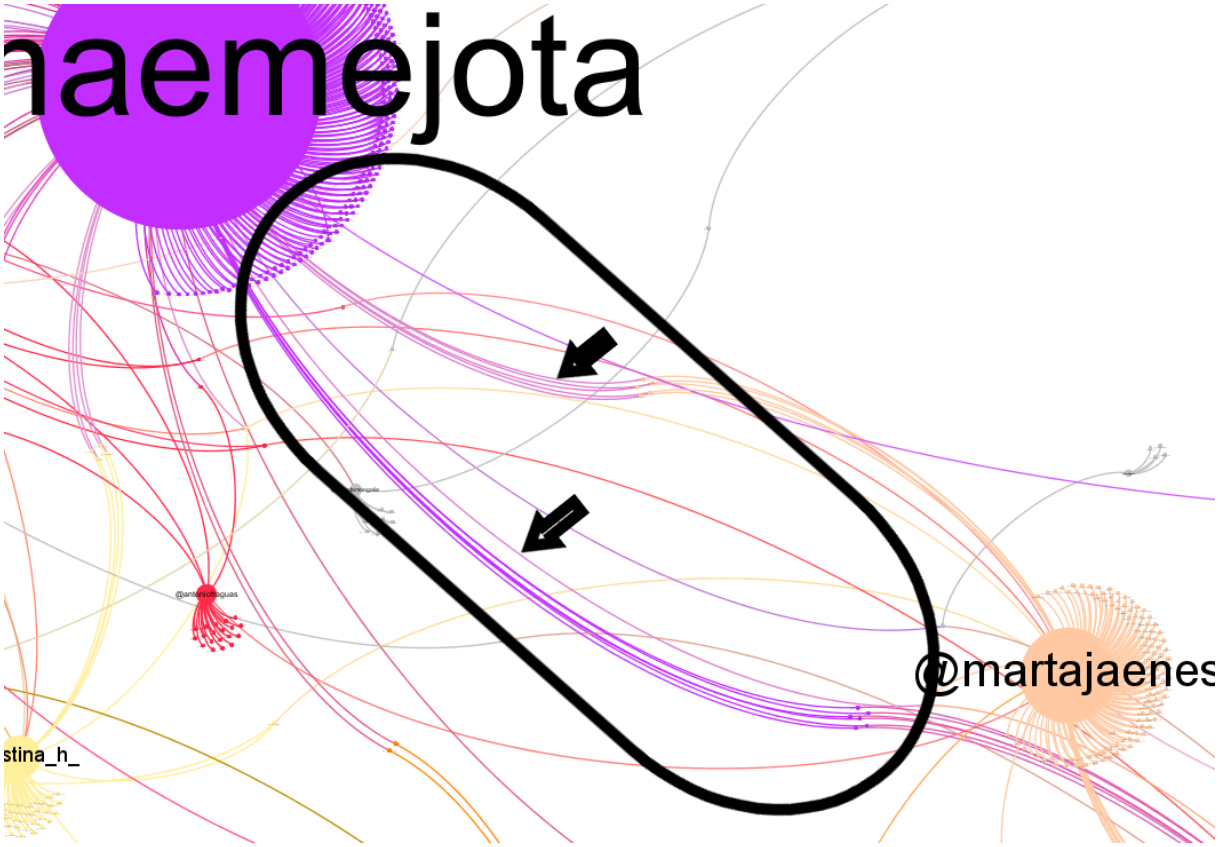

Imagen 3.1.6.3 efecto "Autopista"

Cuando en una comunidad los perfiles se retuitean mucho entre sí (mayor coeficiente de clustering) aparecen zonas densamente conectadas que visualmente son zonas de color "tupido". Esto representa a comunidades activas e implicadas con el tema que se analiza. Estas zonas son perceptibles en la *Imagen 3.1.6.4.*

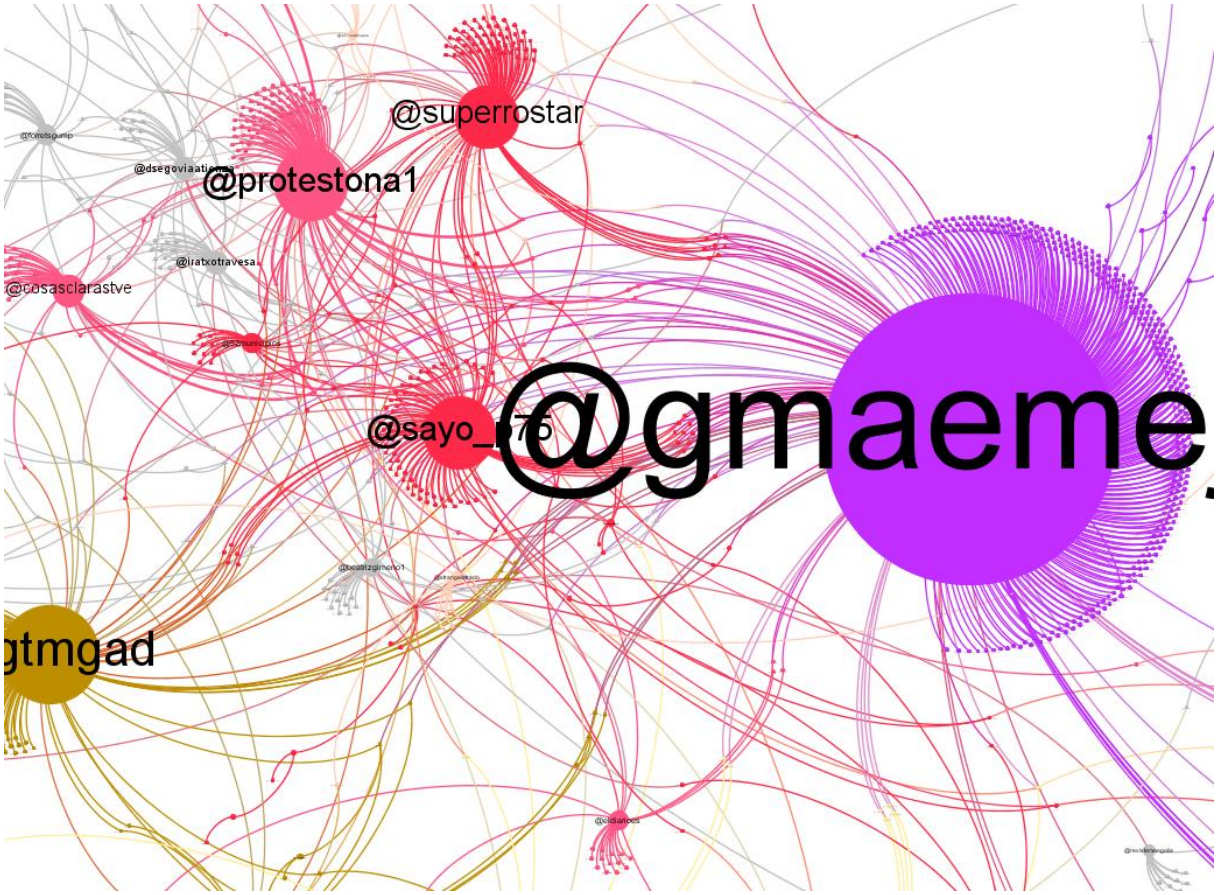

***Imagen* 3.1.6.4 zonas densamente conectadas**

Cuando una comunidad no tiene un color “tupido” y se aprecia la estructura de conexiones de varios nodos muy retuiteados unidos entre sí por conexiones poco densas representa a una agrupación no muy implicada por el tema que se analiza.

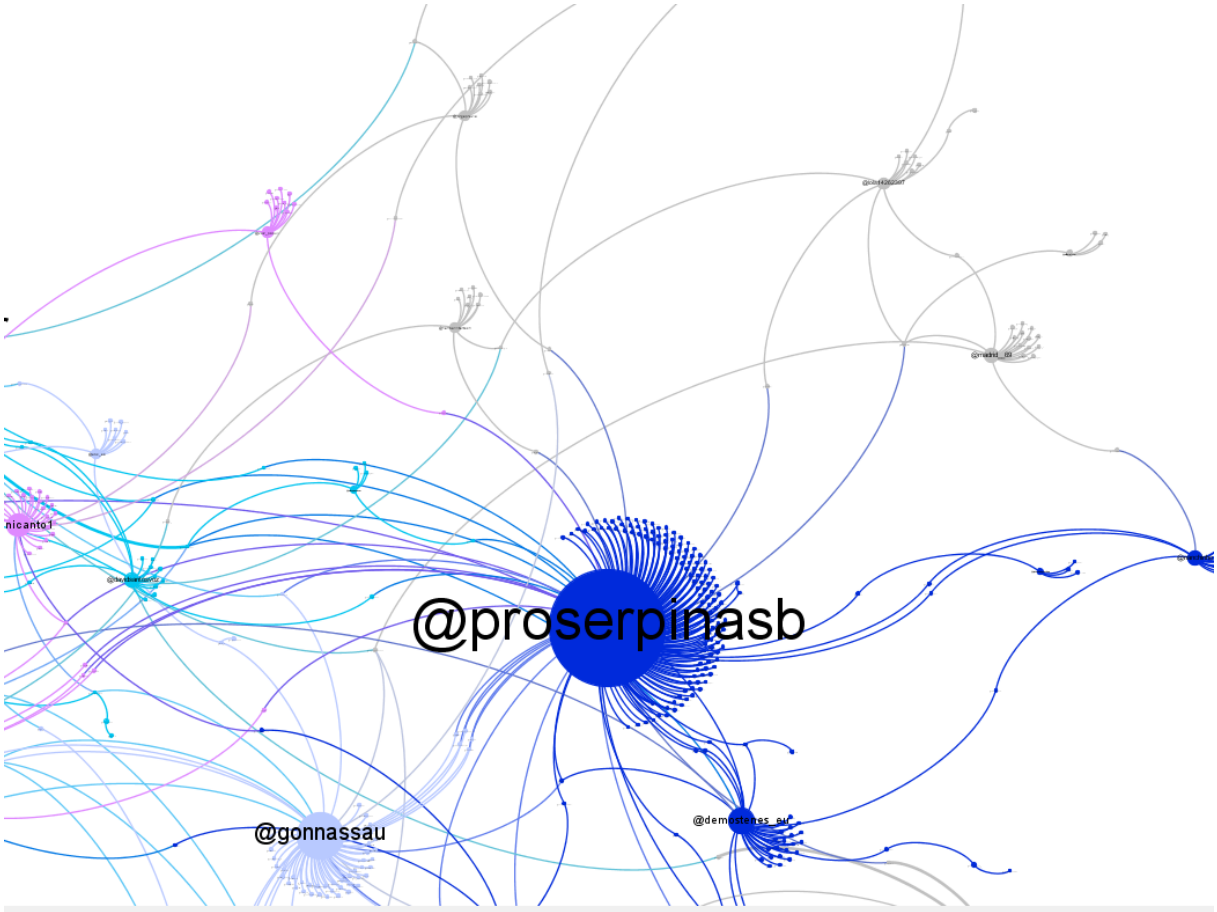

***Imagen* 3.1.6.5 zonas menos densamente conectadas**

Por ejemplo, podemos apreciar estas zonas en la *Imagen 3.1.6.5,* si nos fijamos la comunidad de color azul oscuro gris no es muy densa de hecho está bastante dispersada.

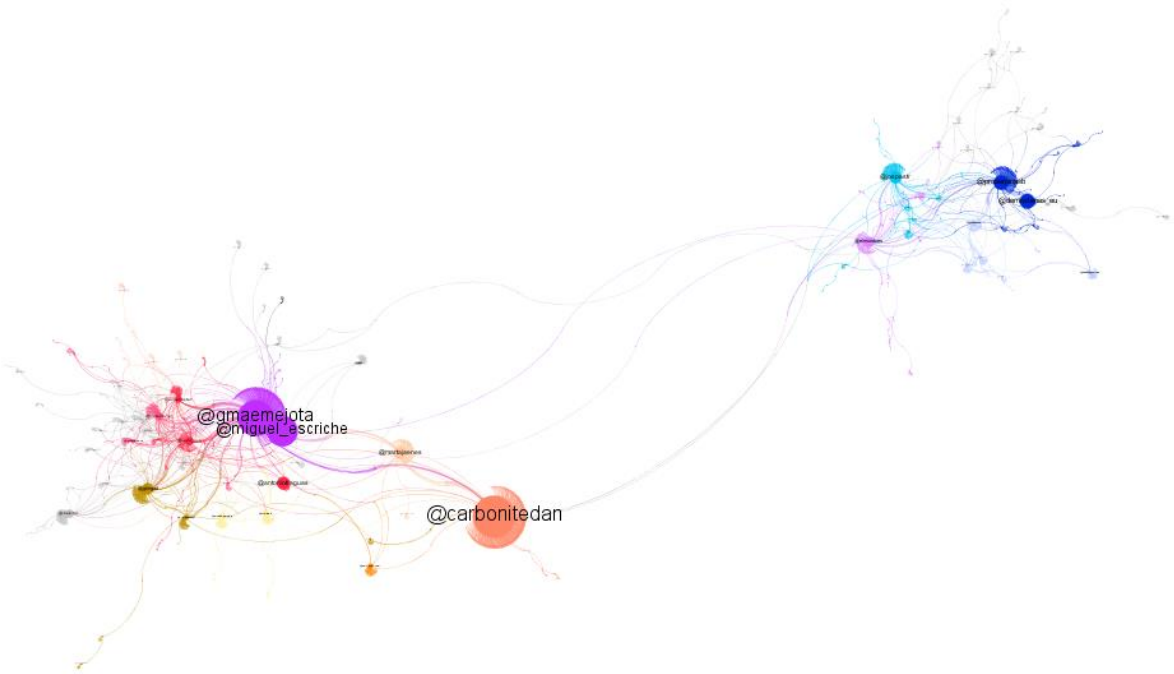

***Imagen* 3.1.6.6 distribución por comunidades y tamaño por PageRank**

Si cambiamos la proporcionalidad del tamaño de los nodos de grado de entrada (RTs), a PageRank, obtenemos el resultado de la *Imagen 3.1.6.6*, lo que nos da un resultado prácticamente idéntico al grafo original. Teniendo en cuenta, que el PageRank es una métrica que nos sirve para calcular los actores más influyentes dentro de nuestra red social. Concluimos que el RT, en una tendencia de Twitter, nos sirve como métrica para calcular los tweets más influyentes, al menos en este caso.

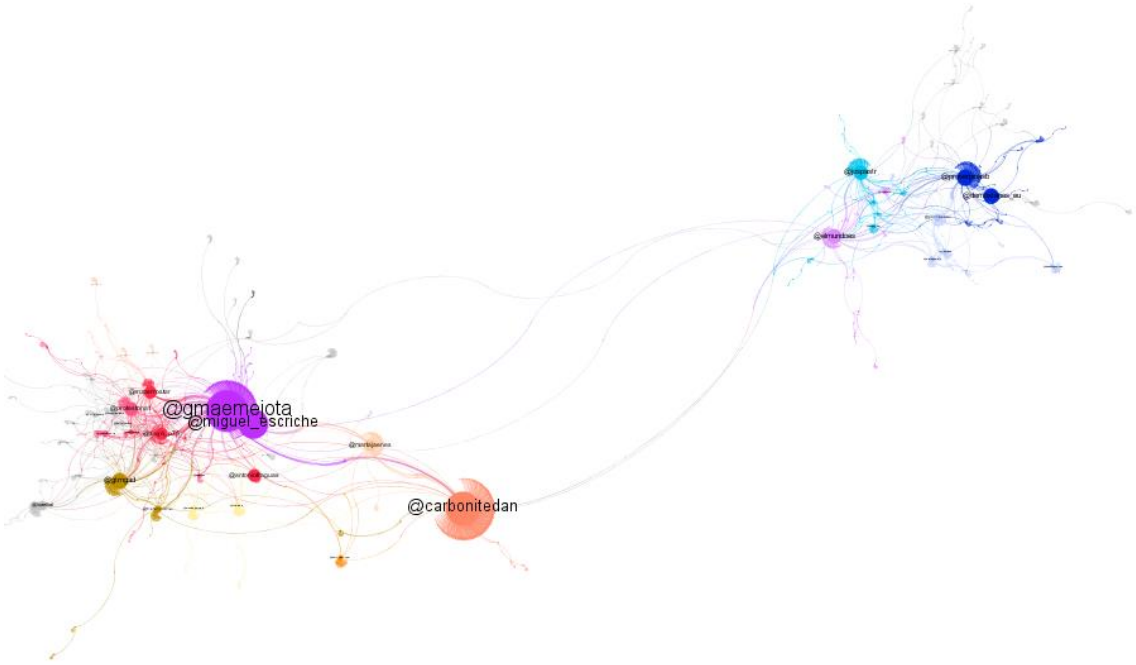

***Imagen* 3.1.6.7 distribución por comunidades y tamaño por Prestigio**

Si cambiamos la proporcionalidad del tamaño de los nodos de grado de entrada (RTs), a Prestigio (“Prestige Rank”), obtenemos el resultado de la *Imagen 3.1.6.7*, de nuevo nos pasa como en el punto anterior y observamos unos resultados prácticamente idénticos. Para calcular el prestigio de un nodo se tiene en cuenta el número de conexiones que tiene, y el número de conexiones que tienen sus vecinos. Al igual que antes, la conclusión es la misma, el RT es una métrica muy similar al prestigio, al menos en este contexto.

El “Presitge Rank” se calcula a partir de un atributo de prominencia, en este caso el número de enlaces.

***Imagen* 3.1.6.8 distribución por comunidades y tamaño por "Centralidad puente"**

Centralidad puente es una métrica de centralidad en sociometría que nos sirve para calcular aquellos nodos centrales que nos funcionan como puente entre ellos, es decir, nodos conectados a vecinos inmediatos que, a su vez, están conectando grupos de nodos. Estos nodos representan a las personas que unen varias comunidades con importancia, por tanto, podemos decir, que son enlaces importantes entre comunidades. En este caso en la *Imagen 3.1.6.8* podemos visualizar una diferencia de cuáles son los nodos que más han destacado con este cambio, por un lado, el más beneficiado sería @ortizblanesjf.

Si nos fijamos, el resto de nodos que antes destacaban por su tamaño han quedado reducidos drásticamente y solo se observa las aristas que unen los nodos cercanos. [11]

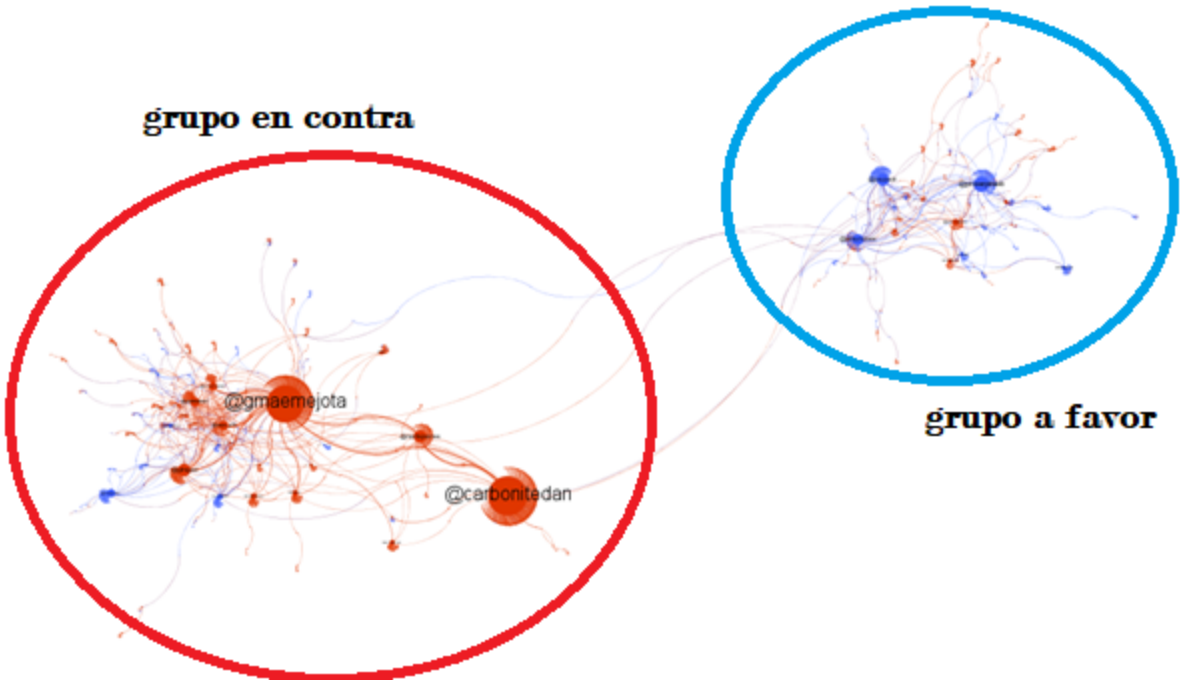

***Imagen* 3.1.6.9 distribución por comunidades y color por opinión**

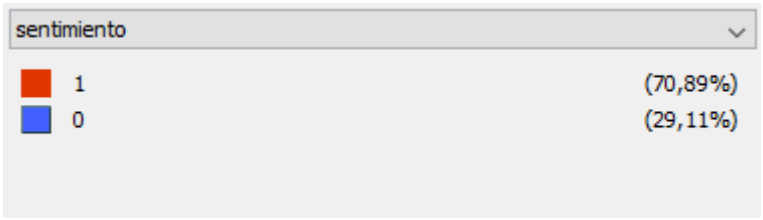

***Imagen* 3.1.6.10 distribución del sentimiento**

Si nos fijamos en la *Imagen 3.1.6.9,* tenemos a los dos grupos claramente diferenciados por el color del sentimiento.

El color rojo representa a la comunidad en contra del vitoreo a Plácido Domingo, mientras que el color azul representa a la comunidad a favor.

Cabe mencionar, que, aunque los grupos no son enteramente de un solo color, esto podría deberse a que nuestro modelo de predicción tiene una precisión del 75%, por lo que la polarización de los grupos por opinión, podría ser aún mayor aún.

3.1.6.1. Conclusiones

Recopilando todos estos puntos sacamos las siguientes conclusiones:

1. La opinión en este hastag es un elemento identificatorio de las comunidades, es decir, las comunidades tienden a agruparse por su opinión creando grupos.
2. La estructura del grafo muestra claramente 2 grupos polarizados, tal y como podemos apreciar en la *Imagen3.1.6.9*. El primero formado por las comunidades azul oscuro, azul claro y rosa, que llamaremos grupo a favor, y el segundo grupo, que está formado por las comunidades de color púrpura, morado, rojo, naranja y amarillo lo llamaremos grupo en contra.
3. El grupo a favor, tiene un menor impacto en comparación con el grupo en contra.
4. El grupo en contra, está compuesto por más comunidades y por usuarios con más impacto.
5. El grupo en contra presenta, por lo general, zonas menos densamente conectadas, por lo tanto, podemos decir que es una comunidad con menor grado de implicación comparado con el grupo a favor.
6. El RT en este hastag, es una medida muy similar al prestigio o al PageRank.

3.2. Experimento OVNI

En este experimento analizaremos la tendencia “OVNI”, este hastag surgió durante 2021-06-25 cuando el gobierno estadounidense publicó un informe sobre los objetos voladores no identificados ese mismo día.

Multitud de tuiteros reaccionaron de diferentes formas ante este suceso. En este experimento de nuevo trataremos ver que comunidades se han formado y como se ha distribuido la información en la red, ya que se trata de un entorno muy diferente, aunque repitamos algunos de los procedimientos realizados en el experimento anterior, me parece muy interesante ya que así podremos compara la naturaleza de los hastag.

Para la recopilación de tuits de este experimento, hemos utilizado una consulta mediante T-hoarder-kit, buscando los tweets en los que aparecen la palabra ovni o #OVNI.

3.2.2. Componente gigante

Imagen 3.2.2 Componente Gigante

Tal y como podemos ver en la *Imagen 3.2.2* este sería el resultado de aplicar la componente gigante sobre los datos recopilados en el hastag ovni, en este caso hemos descartado un 52,41% de los nodos y un 46,06% de las aristas.

A diferencia de nuestro experimento anterior, observamos que estamos descartando prácticamente la mitad de los nodos y aristas, lo que nos dice que hay muchos nodos aislados.

3.2.3. Modularidad

Durante este experimento he usado un valor para la resolución de 2 ya que obtenía demasiadas comunidades.

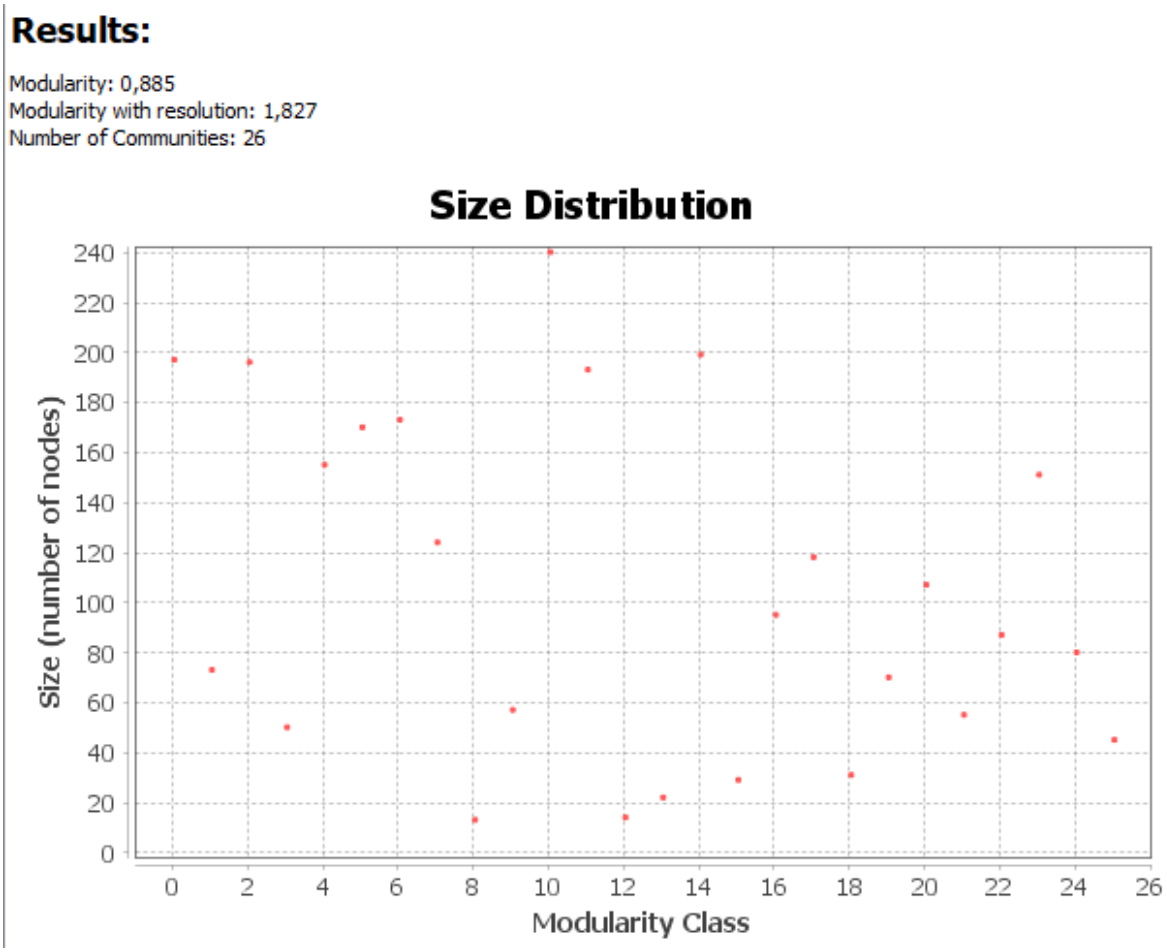

Imagen 3.2.3 distribución de las clases

Una vez ejecutemos el algoritmo de modularidad en Gephi nos presenta los resultados distribuidos como podemos observar en la *Imagen 3.2.3*, en este caso hemos obtenido 26 comunidades.

Si nos fijamos hemos obtenido un valor de modularidad de 0,885, este es un valor bastante alto y que nos indica que no hay comunidades tan bien definidas.

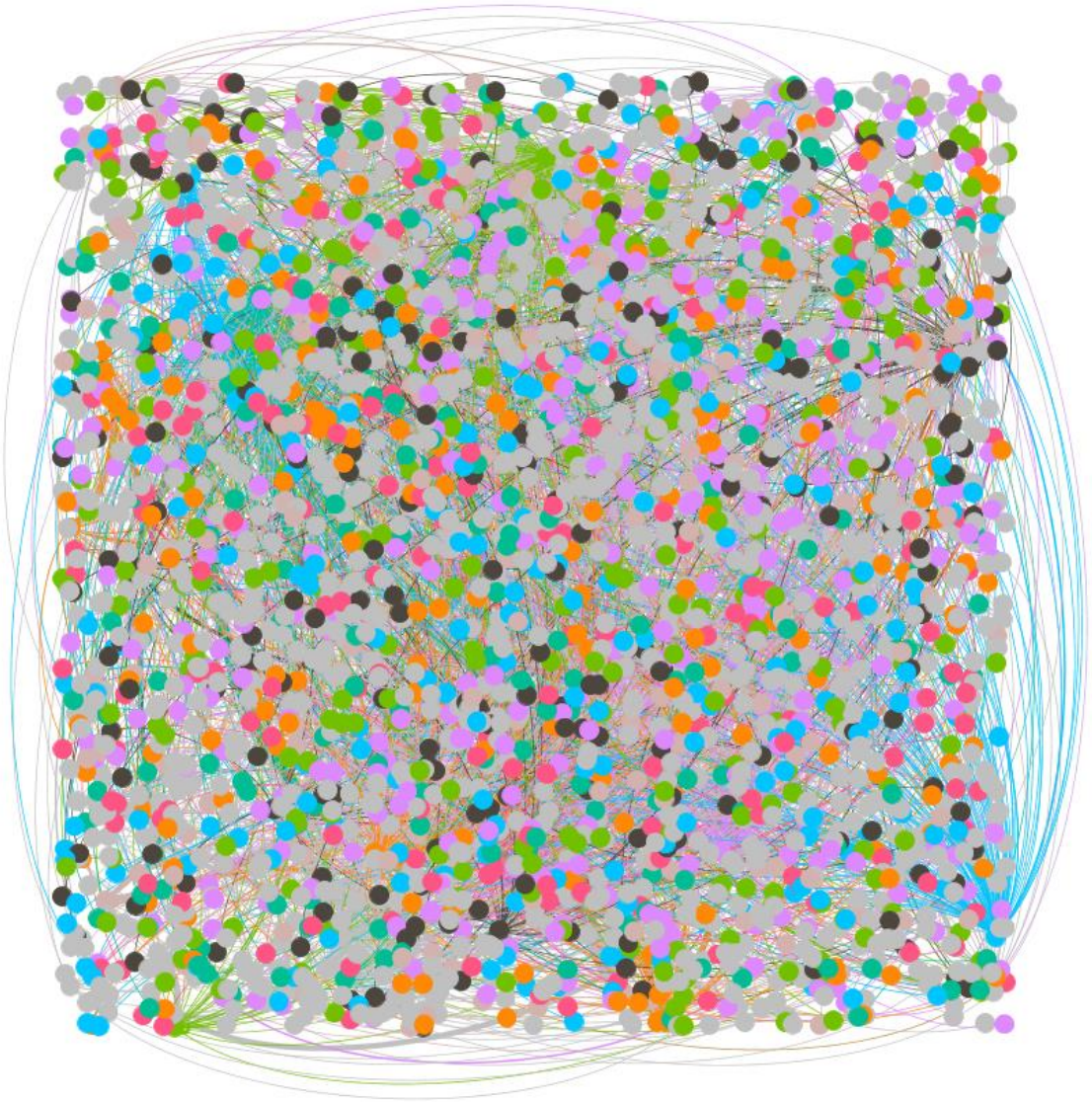

Imagen 3.2.3.1 grafo tras aplicar color por modularidad a los nodos

El siguiente paso de nuevo volvemos a aplicar el color de los nodos por “Modularity Class” que no deja de ser el color de la comunidad que hemos asociado a cada clase.

Observamos que hay muchos nodos grises, estos son nodos que pertenecen a comunidades residuales, a los que Gephi asigna automáticamente este color, posteriormente tendremos que asignar algunos colores a estas comunidades manualmente para evitar confusiones.

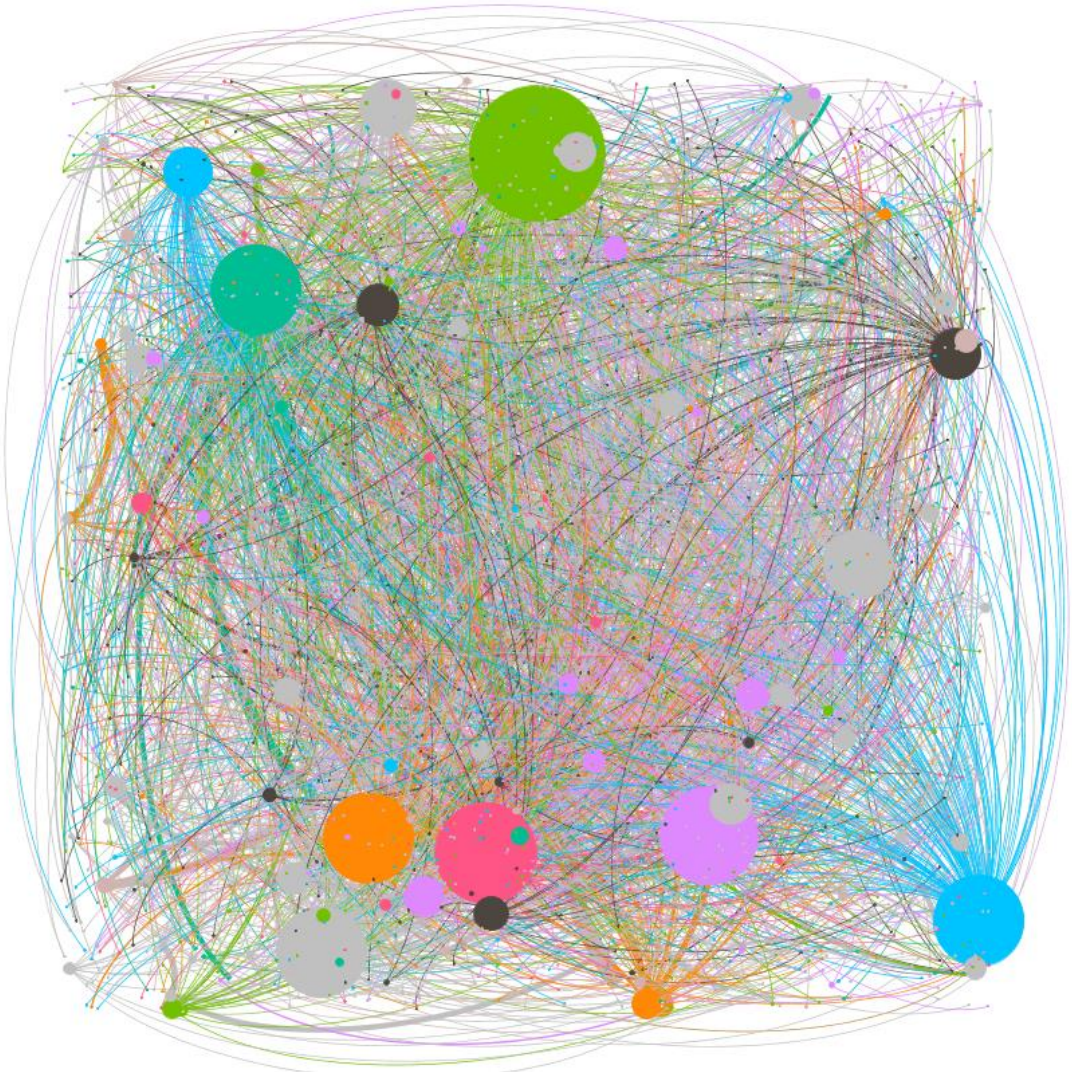

Imagen 3.2.3.2 grafo tras aplicar tamaño proporcional a los nodos

Para continuar queremos dar un tamaño a los nodos proporcional a su importancia, en este caso hemos configurado para que Gephi de un tamaño proporcional, con una escala de 1-200 (en función al grado de entrada), a diferencia del experimento anterior, esto se debe a que, al no haber nodos más tan grandes, podemos jugar más con la escala para que sea más visual. En la *Imagen 3.2.3.2* podemos ver el resultado de aplicar esta modificación.

3.2.4. Distribución

ForceAtlas es el algoritmo que he elegido para la distribución en este experimento, el funcionamiento de este algoritmo es muy similar a ForceAtlas 2, la mayor diferencia es que ForceAtlas es un algoritmo más lento que ForceAtlas 2, pero por el contrario suele dar unos resultados ligeramente mejores, ya que en este experimento contamos con menos nodos, he optado por este algoritmo, puesto que en tiempo de ejecución son similares.

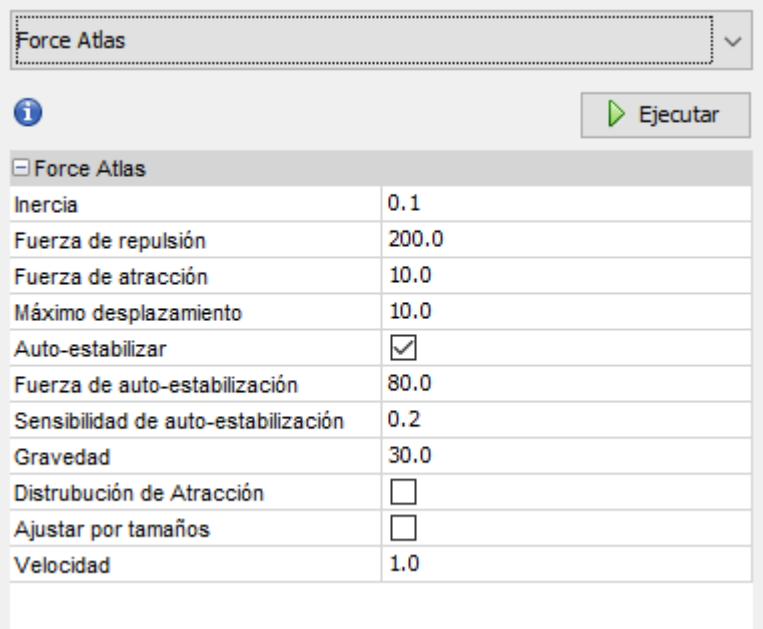

Imagen 3.2.4 parámetros ForceAtlas

.

3.2.5. Ajustes de apariencia

En este caso, los ajustes de apariencia son los mismos que optamos en el experimento anterior, los cuales corresponden con la *Imagen 3.1.5.*

3.2.6. Resultado y Análisis

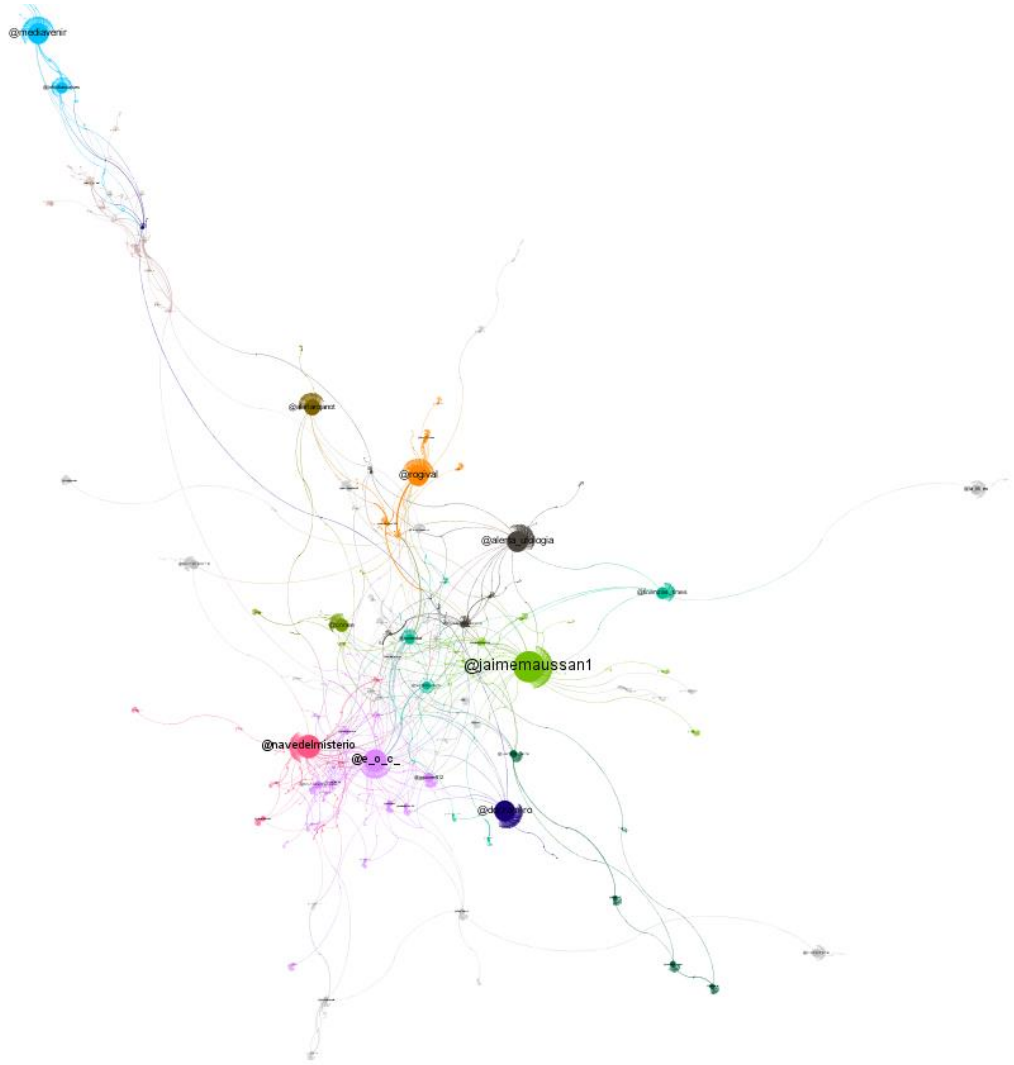

Imagen 3.2.6 Grafo de la tendencia OVNI distribuido por comunidades

En la *Imagen 3.2.6,* podemos ver el resultado final de nuestro grafo. Lo primero que llama a la vista es que no hay un grupo consolidado, o una polarización de comunidades, sino más bien son un conjunto de comunidades dispersas, sin formar zonas densas y con pocas interacciones entre sí.

Aun así, si podemos apreciar a algunas comunidades residuales más aisladas con respecto al resto. Con esto podemos concluir, que se trata de un hastag de una naturaleza diferente al que pudimos apreciar en el experimento “Placido Domingo”,

que presentaba una estructura más típica, de un debate político. Entonces si estas comunidades no se formaron por una opinión, ¿porque lo han hecho?

Aunque en un principio parecía una tendencia española, ya vemos algunos usuarios como @infofrancaises que nos indican que esto no es así, esto se debe a que las siglas OVNI (objeto volador no identificado) que usamos para filtrar nuestros datos, en la fase de recopilación de tuits, no solo son así en español, en francés por ejemplo también coincide con la palabra OVNI (objet volant non identifié).

Nuestra prioridad entonces es, intentar averiguar la coherencia de estas comunidades. Para eso, vamos a empezar a aplicando un filtro, de manera que solo se nos muestren aquellos nodos cuyo atributo idioma (lang), sea igual al español (es).

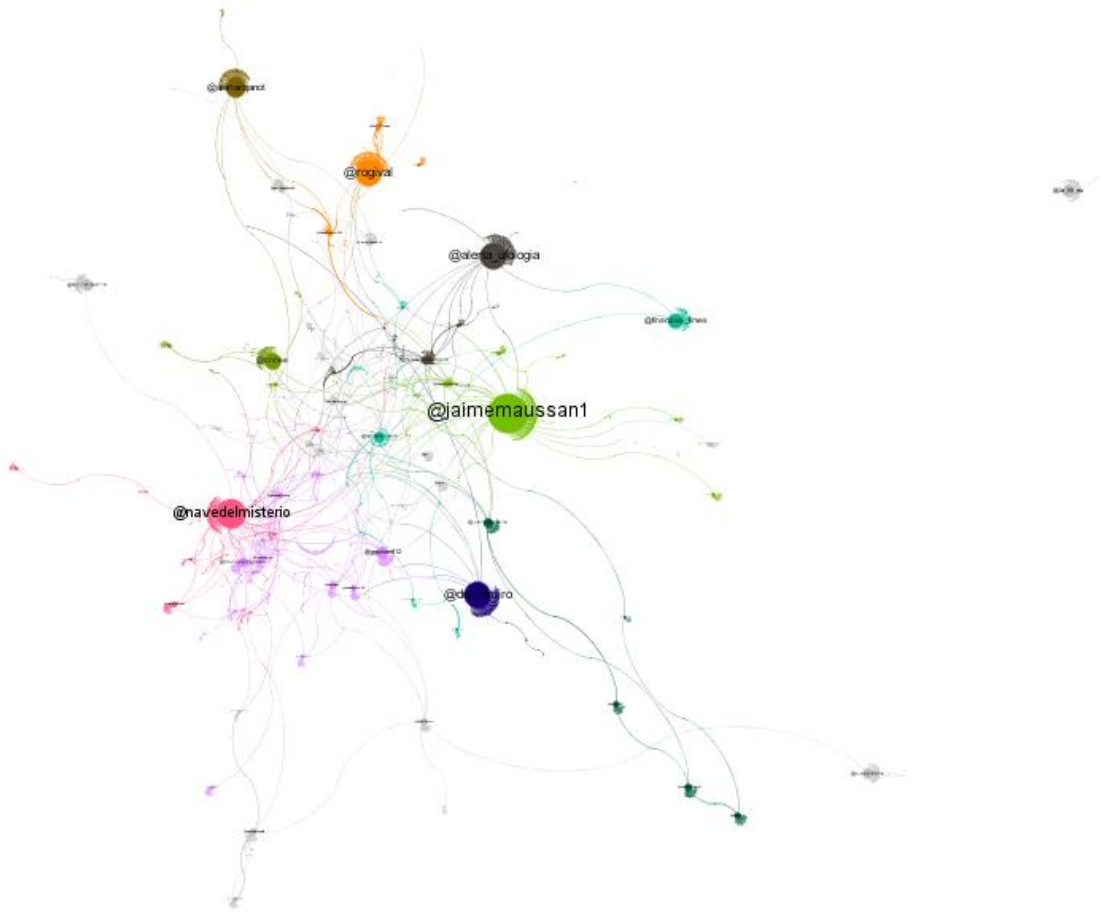

Imagen 3.2.6.1 Grafo de la tendencia OVNI con usuarios de habla hispana

En la *Imagen 3.2.6.1* podemos observar el resultado de haber aplicado el filtro.

Si nos fijamos casi todos los nodos centrales han permanecido, esto nos indica que la mayoría de comunidades centrales son de habla hispana.

El siguiente paso será aplicar el filtro contrario, ya que hemos visto que la mayoría de las comunidades son hispanas, vamos a ver que nodos no son de habla hispana.

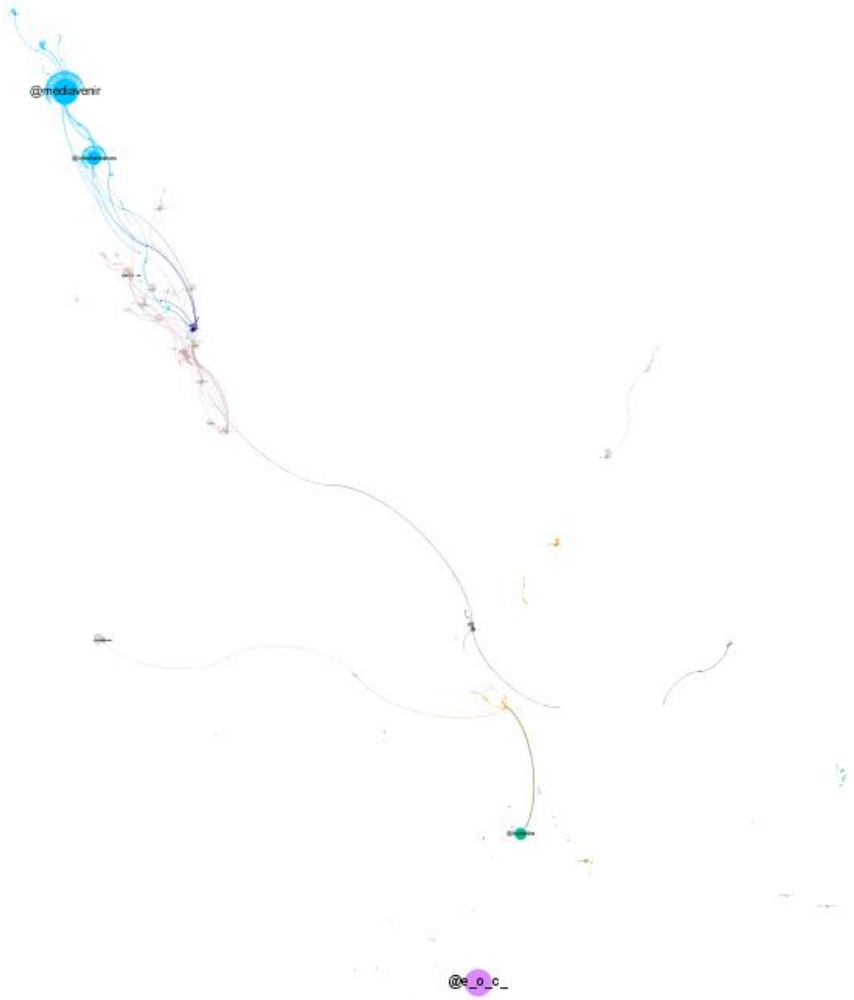

Imagen 3.2.6.3 Grafo de la tendencia OVNI con usuarios de habla no hispana

En la *Imagen 3.2.6.3*, podemos ver el resultado del filtro aplicado, en este caso vemos que la mayoría de nodos forman comunidades más alejadas del centro, aunque también vemos nodos aislados, esto nos indica que la mayoría de los nodos de habla no hispana forman una comunidad, pero hay usuarios sueltos que han interactuado

con comunidades en otros idiomas, aunque esto es menos común, parece que las comunidades se forman por el idioma. Pero para consolidar esta hipótesis vamos a explorar los diferentes idiomas que encontramos y analizar sus comunidades. Para eso podemos seleccionar un nodo y ver sus atributos o filtrar un atributo o nodo en la tabla de datos.

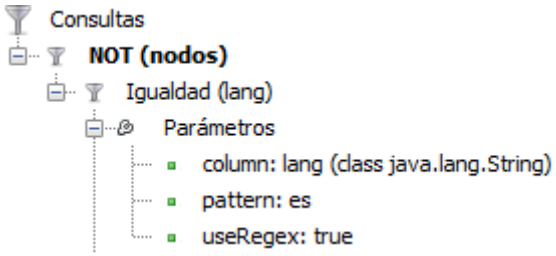

Imagen 3.2.6.4 filtro habla no hispana

Echando un vistazo a la tabla de datos, descubrimos que los idiomas no españoles con más impacto son el francés el portugués y el inglés.

Lo siguiente será filtrar por estos idiomas para ver si son causa de pertenencia de alguna comunidad o no.

En vez de hacer un filtro para cada idioma, lo que vamos a hacer será crear una nueva columna de datos con el idioma del tweet que la llamaremos “langID”, esto es necesario para que Gephi te deje hacer algunas operaciones, ya que necesitamos que sea una columna personalizada para poder aplicar algunas personalizaciones de apariencia. El objetivo es dibujar los nodos dependiendo del idioma.

Imagen 3.2.6.5 distribución por idiomas

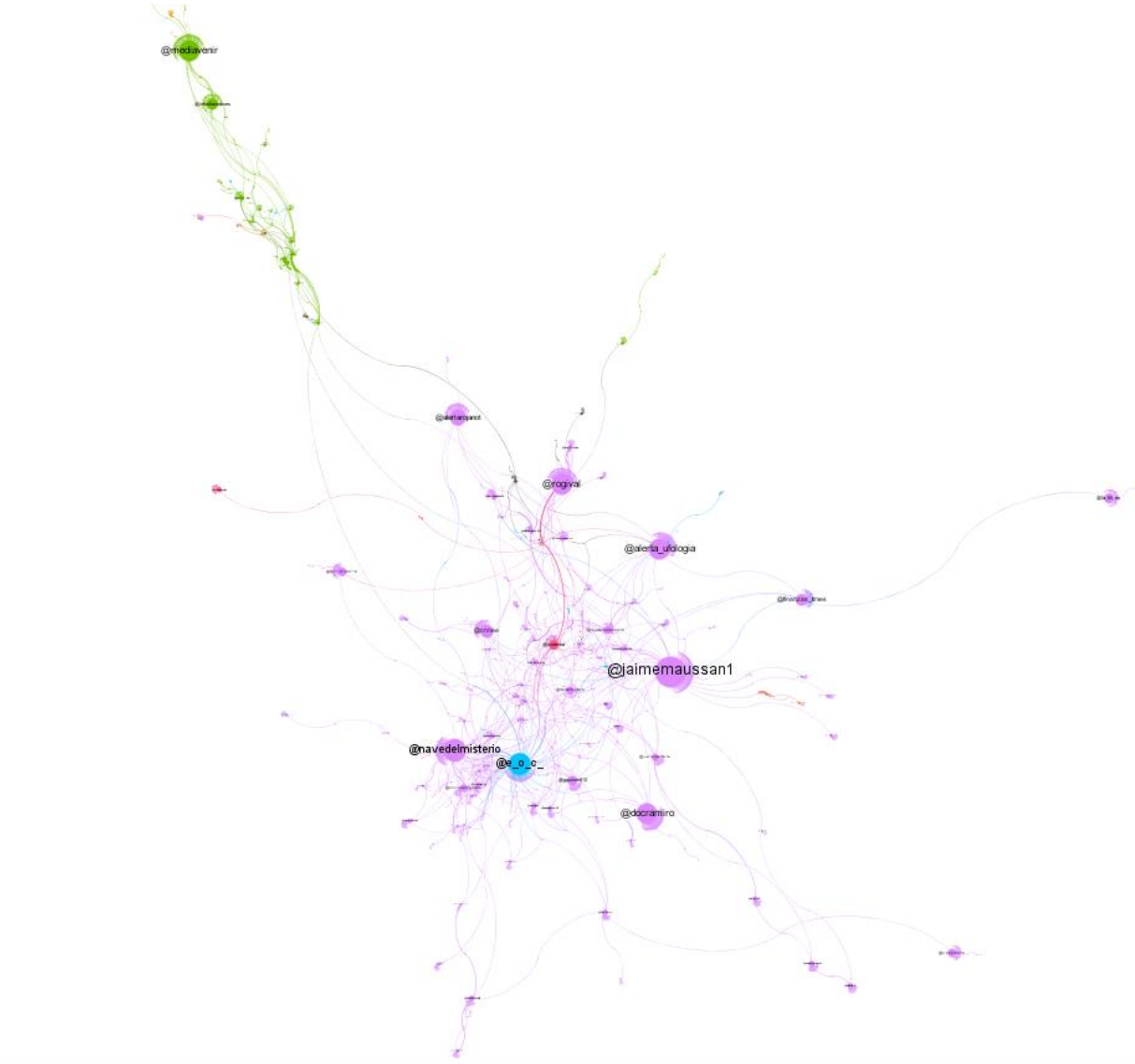

Imagen 3.2.6.6 grafo coloreado por idiomas

En la *Imagen 3.2.6.5* podemos ver la distribución por idiomas, prácticamente solo destacan la comunidad española y la francesa, en la *Imagen 3.2.6.6* podemos ver el resultado visual de esta distribución, si la comparamos con nuestro grafo original, la *Imagen 3.2.6* nos damos cuenta que tanto la comunidad azul como marrón coinciden con los nodos franceses, lo que nos indica que al menos esas comunidades se han formado en base al idioma.

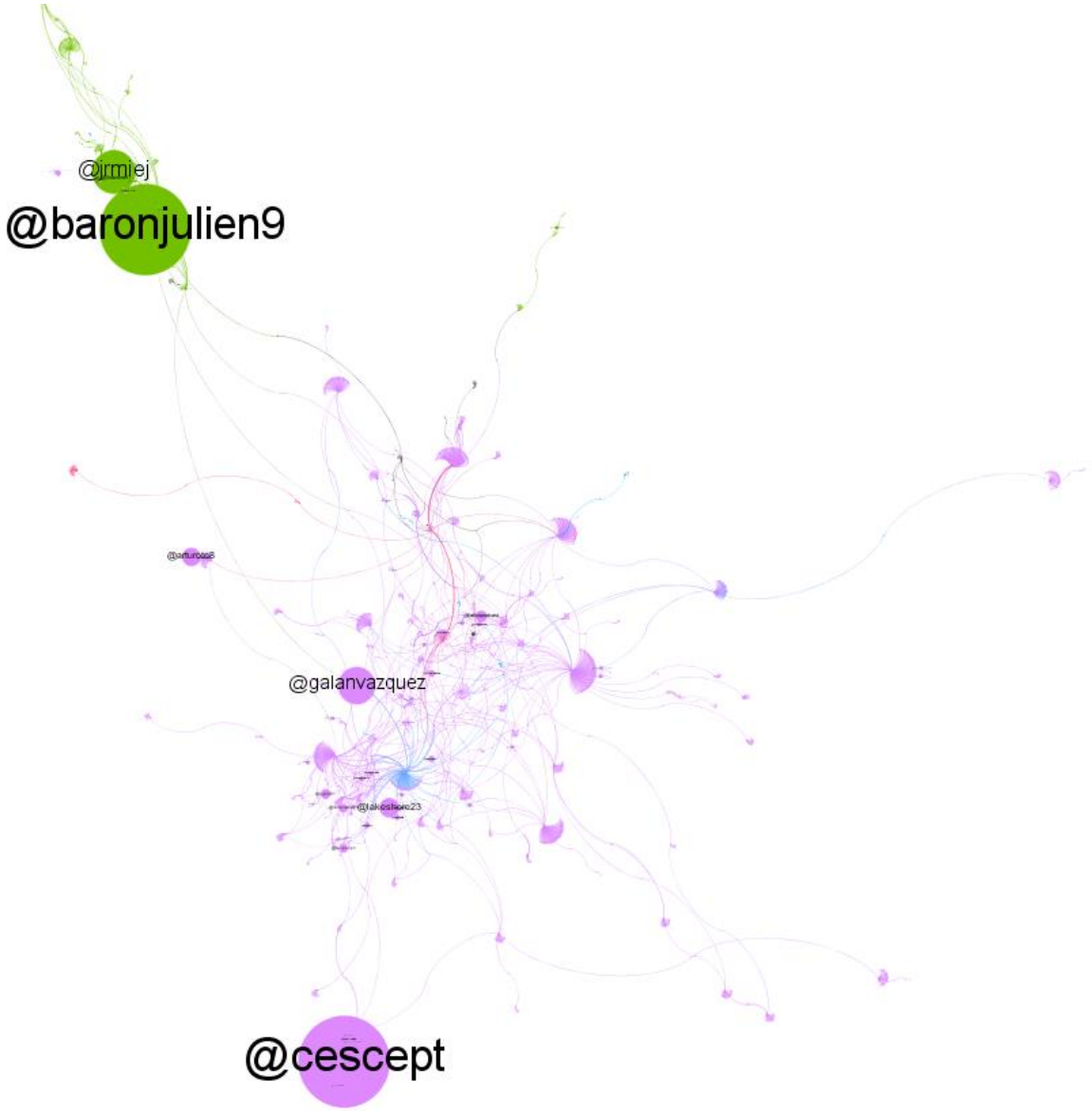

***Imagen* 3.2.6.7 distribución por comunidades, tamaño por "Centralidad puente" y color por idioma**

El siguiente paso será analizar generalidades de nuestra red, para eso de nuevo volveremos a aplicar los filtros de prestigio y “centralidad puente”.

En la *Imagen 3.2.6.7* podemos ver el resultado de aplicar centralidad puente a nuestro grafo, podemos a preciar claramente cuales son los nodos que han destacado con este cambio, estos son @baronjulien9, @jmiej y @cescept. Esto se debe a que son nodos que unen diferentes comunidades, podríamos decir que hacen cuello de botella, por ejemplo, en el caso de @baronjulien9, @jmiej obtienen tanta relevancia debido a que son nodos que unen la comunidad francesa con la comunidad española. Por otro lado @cescept hace de puente con algunas comunidades aisladas y muchos usuarios que solo interactuaron con él, tal y como podemos apreciar en la *Imagen 3.2.6.8* el usuario ha “absorbido” a estos nodos debido a su tamaño y cercanía.

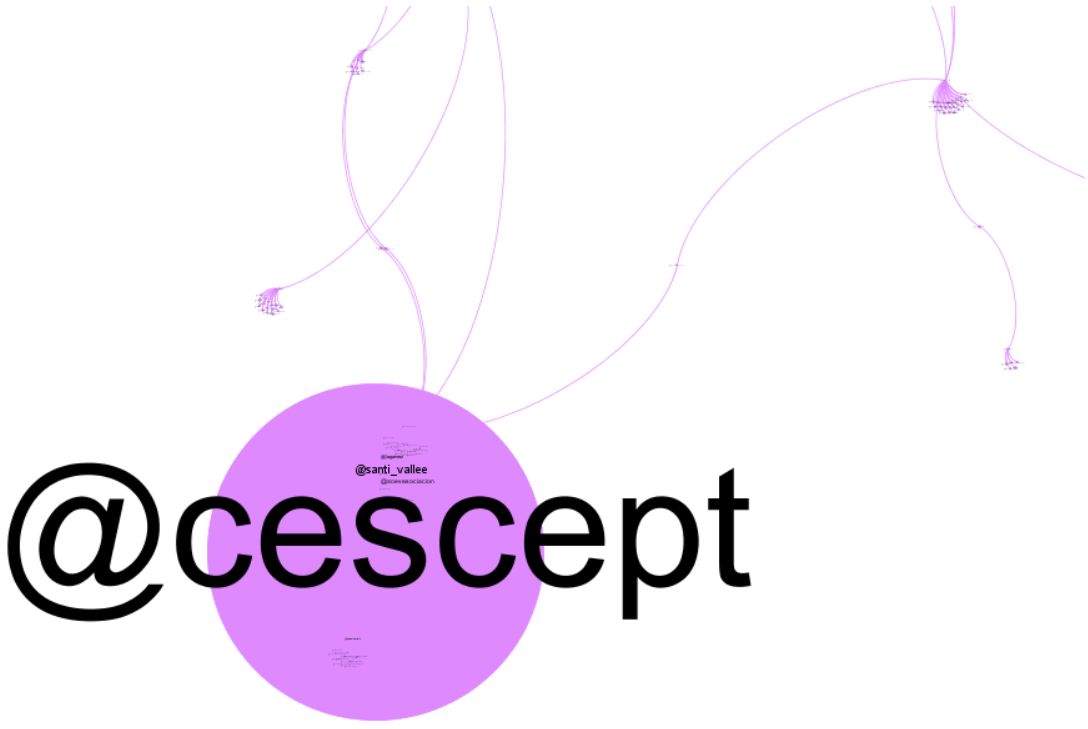

***Imagen* 3.2.6.8 usuario @cescept y su entorno**

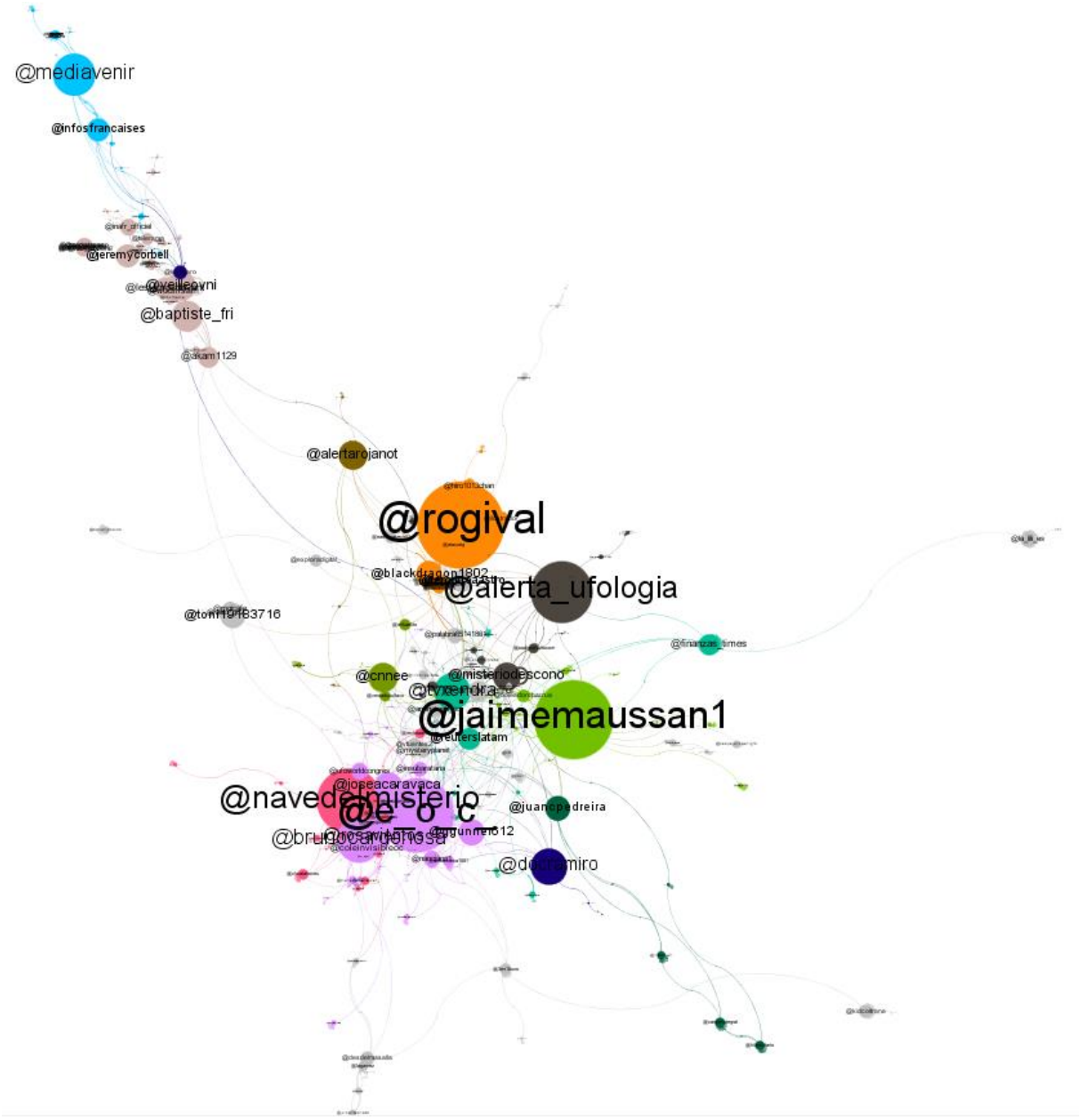

***Imagen* 3.2.6.9 distribución y color por comunidades y tamaño por Prestigio**

En la *Imagen 3.2.6.9* podemos ver el resultado de aplicar las transformaciones descritas, como podemos observar claramente tenemos un cambio con respecto a nuestro gafo original. En este caso los nodos de las comunidades más centrales destacan mucho más con respecto al resto, es decir, a diferencia del experimento anterior, en este si hay un cambio significativo, entre la métrica prestigio y grado de entrada (número de RT's). Esto nos indica que el "valor" del RT es diferente en este hastag y no está tan asociado al prestigio.

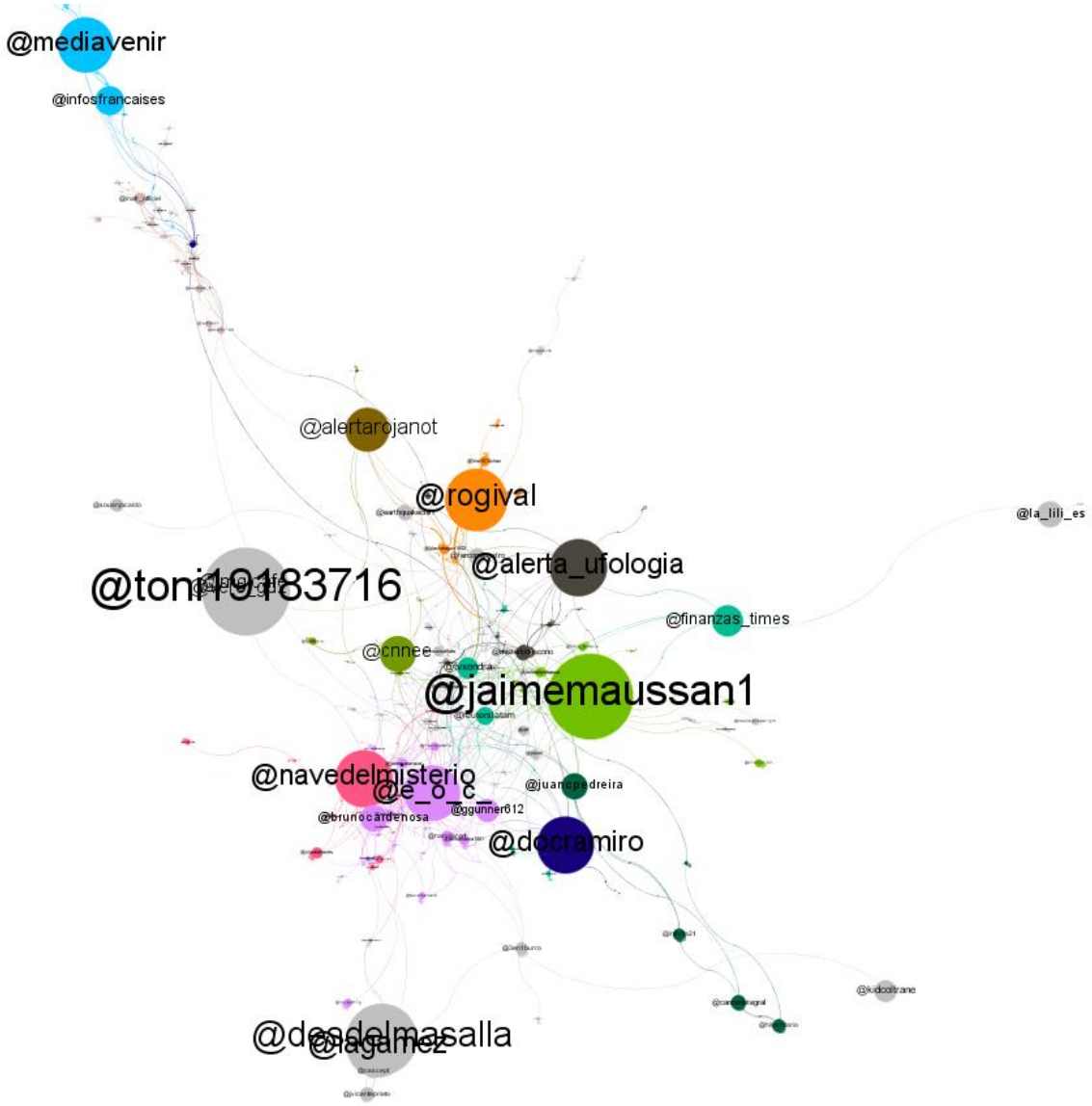

***Imagen* 3.2.6.10 distribución y color por comunidades y tamaño por PageRank**

En la *Imagen 3.2.6.10* podemos apreciar el resultado de aplicar las transformaciones expresadas, en este caso el cambio con respecto a la *Imagen 3.2.6.9* son bastante menores, sin embargo, si destaca el cambio de relevancia de los usuarios @toni19183716, @desdelmasalla y @lagamez.

***Imagen* 3.2.6.11 distribución del sentimiento**

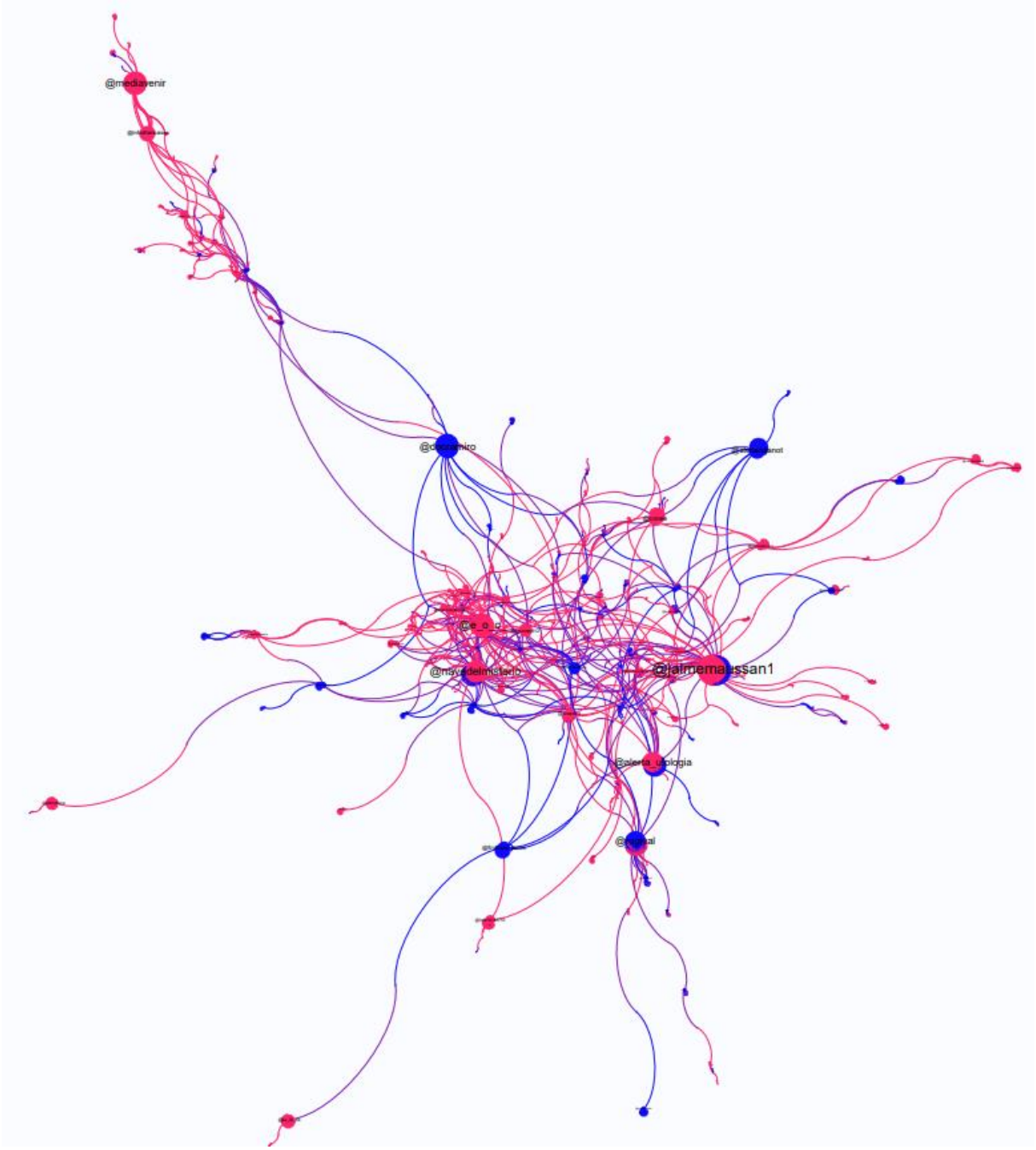

***Imagen* 3.2.6.12 distribución comunidades y sentimiento por color**

La *Imagen 3.2.6.12* es el resultado de aplicar el sentimiento como color a los nodos, sin embargo, como podemos observar, no podemos apreciar ninguna relación o apreciación destacable, que dote de significado a las comunidades.

Para averiguar más sobre las comunidades centrales, vamos a coger los tweets de los autores más importantes de cada comunidad para hacer una nube de palabras.

***Imagen* 3.2.6.13 nube de palabras de los tuits principales de la comunidad morada**

***Imagen* 3.2.6.14 nube de palabras de los tuits principales de la comunidad verde**

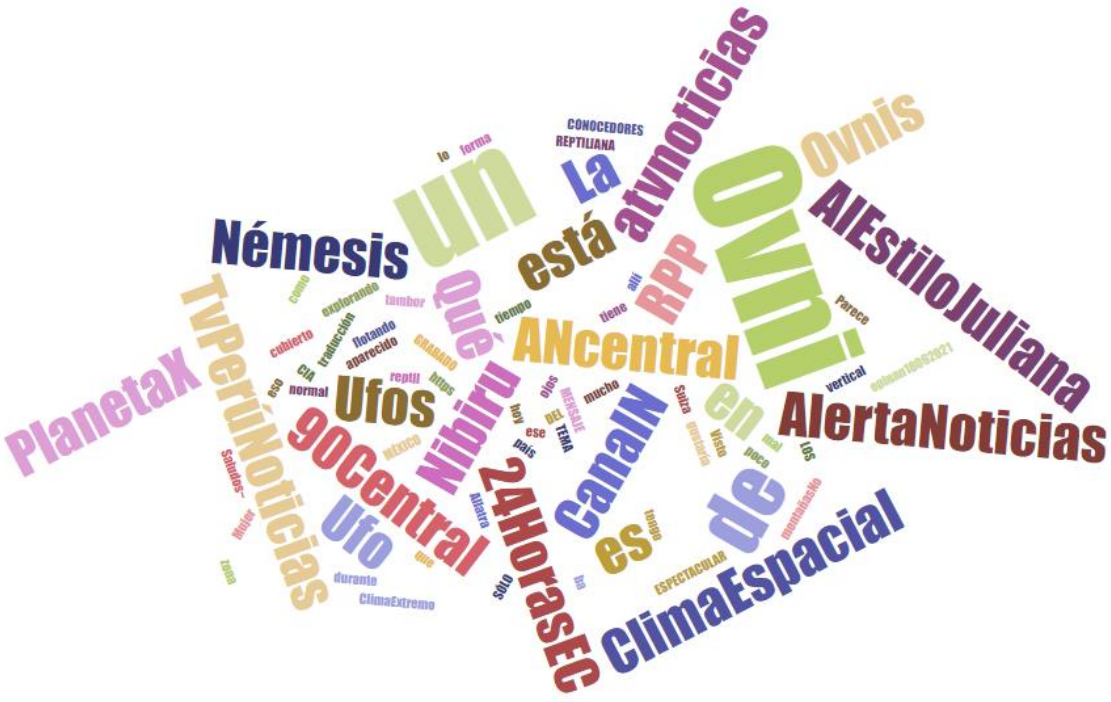

***Imagen* 3.2.6.15 nube de palabras de los tuits principales de la comunidad naranja**

***Imagen* 3.2.6.16 nube de palabras de los tuits principales de la comunidad rosa**

Si nos fijamos en el resultado de las diferentes nubes de palabras para cada comunidad, observamos que obtenemos unos resultados muy similares.

De hecho, a la hora de inspeccionar los tweets y los usuarios, descubrimos un patrón. En prácticamente todas las comunidades hay algún medio local, nacional o similar que se ha hecho eco de la noticia (Onda Cero, 24HorasEC, TvPerúNoticias, atvnoticias, sopia..) y diversos tuiteros que responden o propagan la noticia, además suele haber perfiles de usuarios que se dedican a hablar de ese tema en específico, que normalmente son los que unen comunidades ya que retuitean noticias de diferentes medios.

3.2.6.1. Conclusiones

En este experimento hemos visto unos resultados muy diferentes al anterior, y esto se debe a que claramente son hastag de naturaleza muy diferentes.

1. Es un hastag más internacional (hay más usuarios de diferentes idiomas y países participando) que en el experimento anterior por lo que esto afectará a nuestra red.
2. Es un hastag con menos impacto en general, tanto al nivel de número de tuits como influencia de estos.
3. Las comunidades están dispersas y no se crean bloques claramente diferenciados.
4. Las comunidades son menores y con menos implicación.
5. No hay muchos enlaces entre las comunidades centrales, y las más aisladas, por lo que los nodos que las unen, tienen bastante importancia en cuanto a la propagación del hastag.
6. Las comunidades se crean por pura distribución de la información, con esto quiero decir, que no hay una razón de ser de las de las comunidades, como en el experimento anterior que se distribuían claramente por la opinión. En este caso se distribuyen por las relaciones de los usuarios y sus círculos, en los que a veces el idioma o la nacionalidad es lo que influye a la creación de estos círculos.

7. Los usuarios implicados en este hastag son por lo general usuarios muy interesados específicamente en estos temas, por lo que ya cuentan con una red de seguidores que a su vez están interesados en su contenido, podríamos decir que es un hastag más de “nicho”.

4. Conclusiones generales del proyecto

Este es un TFG, aunque plantea un tema muy concreto, tiene muchísimas posibilidades para seguir expandiéndose, hay numerosas herramientas y tecnologías que nos permiten o nos ayudan a analizar los temas de tendencia en Twitter, sin embargo, muchas de ellas no son accesibles o son de pago. Yo he usado las que he considerado más óptimas o las que ya conocía, pero hay multitud de aplicaciones con más recursos que nos dan unas funcionalidades adicionales.

Además, parte de un tema muy complicado, el procesamiento del lenguaje y las relaciones sociales, a día de hoy esto sigue siendo ámbito de estudio e investigación.

Hay numerosos artículos que tratan este tema, y hay una cantidad enorme de algoritmos, métricas y conceptos de teoría de grafos que podríamos seguir aplicando para mejorar nuestra representación y análisis de una tendencia, sin embargo, considero que he conseguido un buen balance entre una solución práctica que resolviera el problema propuesto para este proyecto, acorde con los requisitos del mismo, sin abandonar el tema o propasarme con ello.

También es cierto que gran parte del trabajo realizado no está plasmado en esta memoria, por ejemplo, a la hora de optimizar nuestro modelo de predicción de sentimiento, hay un largo proceso de formación, probar diferentes tecnologías, librerías, corpus, diferentes métodos de procesamiento de texto, optimización de parámetros entre otras cosas, por lo que se consume mucho tiempo solo a base de prueba y error.

En el apartado de procesamiento de grafos pasa más de lo mismo, hay cantidad de métricas, filtros y algoritmos similares que producen diferentes resultados en

nuestro grafo, y aunque es cierto que aquí podemos guiarnos por la teoría de grafos, no siempre está claro que algoritmo/métrica es más óptimo en que problema.

Si tuviera que decir algunas conclusiones más específicas sobre el tema del proyecto diría:

1. Tener en cuenta la naturaleza de la tendencia que estamos analizando, el contexto es muy importante y nos da mucha información, por ejemplo, en este proyecto hemos analizado 2 tendencias, la primera que podríamos categorizar como una tendencia de debate o contenido político, y la segunda tendencia que es más una reacción a una noticia, cada una con sus características propias.
2. Saber identificar las estructuras de grafos e interpretar su significado.
3. Las comunidades es un concepto muy importante en este tipo de grafos, por lo que analizarlas es una tarea principal.
4. A veces las comunidades, nos dan más información o son más precisas, que las propias técnicas de minería de opinión en cuanto al sentimiento del contenido que propagan.
5. Cuando tengamos dudas sobre algunos fenómenos, lo mejor es filtrar la información al máximo, identificar los elementos y explorarlos manualmente. No todas las respuestas las he obtenido mediante los algoritmos de forma automática, a veces para asegurarnos de ciertos comportamientos lo mejor es revisarlo uno mismo. Pero no es lo mismo tener que hacer este proceso sobre una cantidad inmensa de información, que habiéndola filtrado y aislado previamente.
6. Tener claro el funcionamiento de los algoritmos y la funcionalidad de las métricas, nos ayudan a saber que es mejor aplicar en cada momento, dependiendo de aquello que queramos destacar o averiguar.
7. Hacer un perfil sobre el tipo de usuarios que participan en un hastag, también nos da una información valiosa a considerar.

5. Anexo: Instalación

Para poder reproducir los resultados de este proyecto necesitamos 3 herramientas:

1) T-hoarder-kit: Podemos descargar esta herramienta de forma gratuita desde: https://github.com/congosto/t-hoarder_kit, los pasos de instalación están descritos en la misma página.
    a. Los únicos requisitos son, utilizar un sistema operativo LINUX y tener las keys de acceso a una aplicación Twitter developer tal y como explicamos en el punto *2.1.1.*
2) Proyecto *SentimentalAnalysis* desarrollado en Python, esta accesible mediante mi GitHub: https://github.com/JavierGarciaGarzon/sentimentalAnalysis.
    a. Tan solo debemos de tener en cuenta las dependencias con las librerías de sklearn, pandas, csv,numpy,random y string.
3) Gephi: su instalación es sencilla y directa tal y como nos muestra en su página web: https://gephi.org/users/install/

5.1. Estructura del Proyecto

En cuanto a la estructura del proyecto *SentimentalAnalysis,* es la siguiente:

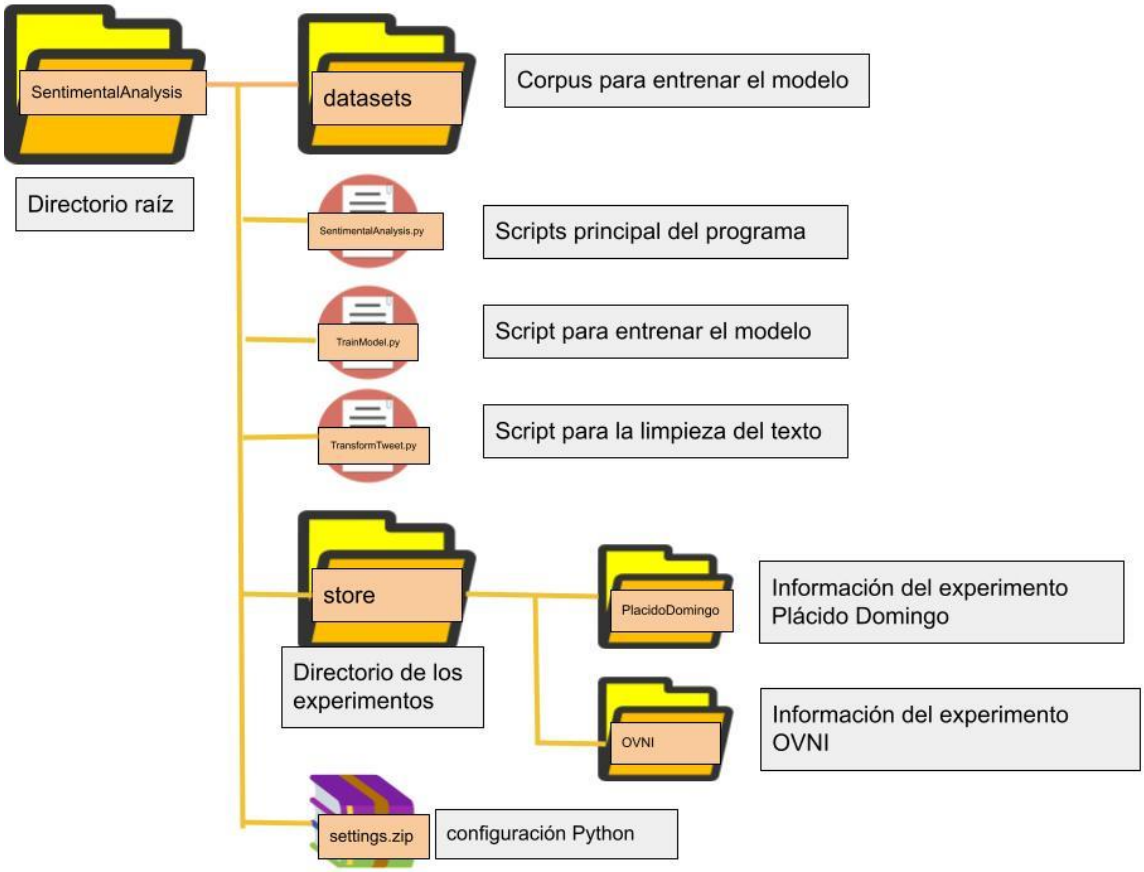

***Imagen* 5.2 Estructura del proyecto SentimentalAnalysis**

Tan solo añadir que la configuración de Python “settings.zip” se puede importar para Pycharm, no debería ser necesario importarla para que los scripts funcionen, pero si da algún error es recomendable.

Bibliografía

[1] blog.twitter, 2019. Twitter es la red donde la información política tiene mayor relevancia. Recuperado de: https://blog.twitter.com/es_es/topics/insights/2019/twitter-es-la-red-donde-la-informacion-politica-tiene-mayor-rele

[2] Martínez-Cámara, E., Martín-Valdivia, M. T., Ureña-López, L. A., Mitkov, R. (2015). Polarity classification for Spanish tweets using the COST corpus. Journal of Information Science, 41(3), 263-272. DOI: 10.1177%2F0165551514566564.

[3] Newman, M. E. J. (2000). Who is the best connected scientists. A study of scientific co-authorship networks.

[4] Oscar Cordón García. Redes y Sistemas Complejos. Modularidad, Particionamiento y Comunidades. Recuperado de: https://sci2s.ugr.es/sites/default/files/files/Teaching /GraduatesCourses/RedesSistemasCompejos/Tema06-DeteccióndeComunidades-13-14.pdf

[5] Kwak, Haewoon & Lee, Changhyun & Park, Hosung & Moon, Sue. (2010). What Is Twitter, a Social Network or a News Media?. Proceedings of the 19th International Conference on World Wide Web, WWW '10. 19. 10.1145/1772690.1772751.

[6] CONGOSTO, M., 2019. Explicando los detalles de un grafo. [Blog] barriblog, Recuperado de: http://www.barriblog.com/2019/06/explicando-los-detalles-de-un-grafo/.

[7] Muente, G., 2018. Nube de palabras: qué es y para qué sirve. [Blog] rockcontent, Recuperado de: https://rockcontent.com/es/blog/nube-de-palabras/

[8] Congosto, M., Basanta-Val, P., & Sanchez-Fernandez, L. (2017). T-Hoarder: A framework to process Twitter data streams. Journal of Network and Computer Applications, 83, 28-39.

[9] Learn how to use Gephi. (s. f.). https://gephi.org/. Recuperado de: https://gephi.org/users/

[10] scikit-learn. (s. f.). scikit-learn. https://scikit-learn.org/stable/user_guide.html

[11] TY BOOK AU - de Morais Pereira, Getúlio PY - 2018/02/23SP - T1 - Bridging Centrality Gephi Plugin ER

[12] Social Network Visualizer: SocNetV Manual. (s. f.). socnetv.org. Recuperado de https://socnetv.org/docs/index.html

[13] Serrat, O. (2017). Social network analysis. In Knowledge solutions (pp. 39-43). Springer, Singapore.

[14] MARTÍNEZ CÁMARA, Eugenio, et al. “Técnicas de clasificación de opiniones aplicadas a un corpus en español”. Procesamiento del Lenguaje Natural. N. 47 (2011). ISSN 1135-5948, pp. 163-170