ESTUDIO DE TENDENCIAS DIARIAS EN TWITTER

Trabajo de fin del Grado de Ingeniería Informática
Facultad de Informática

Realizado por:
Ángel Luis Ortiz Folgado
Óscar Eduardo Pérez la Madrid
Esteban Vargas Rastrollo

Dirigido por:
Alberto Díaz Esteban
Virginia Francisco Gilmartín

Curso 2014/2015

UNIVERSIDAD
COMPLUTENSE
MADRID

Índice

Índice de figuras

Autorización

Los autores de este documento: Ángel Luis Ortiz Folgado, Óscar Eduardo Pérez la Madrid y Esteban Vargas Rastrollo, alumnos matriculados en la asignatura Trabajo de Fin de Grado autorizan, mediante el presente documento, a la Universidad Complutense de Madrid (UCM), a difundir y utilizar con fines académicos, y mencionando expresamente a sus autores, tanto la propia memoria, como el código, la documentación, y/o el prototipo desarrollado. Todo ello realizado durante el curso académico 2014 -2015 bajo la dirección de Alberto Díaz Esteban y Virginia Francisco Gilmartín profesores del Departamento de Ingeniería del Software e Inteligencia Artificial.

Ángel Luis Ortiz Óscar Pérez Esteban Vargas

Agradecimientos

En primer lugar, queremos dar las gracias a todos nuestros familiares y amigos por confiar en nosotros y estar siempre ahí durante los buenos y malos momentos por los que hemos pasado a lo largo de la carrera universitaria.

También dar las gracias a Virginia y Alberto por guiarnos y ayudarnos durante todo el proceso de realización de este trabajo. Además del resto de profesores que durante la carrera nos han enseñado los conocimientos sin los cuáles este trabajo no podría haberse realizado.

Por último, dar las gracias también a toda esa gran comunidad colaborativa de Internet que mediante aplicaciones útiles y foros de discusión han ayudado a resolver algunos escollos encontrados durante la realización de este trabajo.

Resumen

En la actualidad las redes sociales han cambiado el paradigma de las relaciones sociales y del acceso a la información. Twitter con el paso del tiempo se ha convertido en un medio de comunicación que está dejando poco a poco a los medios tradicionales como los periódicos y la televisión en un segundo plano. La ingente cantidad de información que se genera constantemente en esta plataforma provoca que surja una necesidad de agrupar y concretar esta información para que el usuario medio de Twitter no se vea abrumado por el exceso de información, muchas veces irrelevante, que recibe.

Una de las funcionalidades que ofrece Twitter es la posibilidad de agrupar los temas de los que se está hablando en la aplicación bajo un conjunto de tendencias, llamados trending topics. Estos trending topics tienen un tiempo de vida limitado, es decir, que una vez los usuarios dejan de hablar de ese tema en la aplicación, éste ya no formará parte de la lista de trending topics que Twitter ofrece a sus clientes. Esto puede llevar a que si el usuario no ha estado presente en la aplicación durante la aparición de una tendencia que le pueda resultar atractiva, le resultará complicado informarse sobre la misma.

El objetivo de este trabajo es el desarrollo de una aplicación que permita a los usuarios explorar el conjunto de tendencias que han ido apareciendo a lo largo del tiempo, permitiendo así que cada trending topic que ofrece Twitter quede registrado en la aplicación para su posterior análisis. La aplicación desarrollada ofrece funcionalidades de clasificación y agrupamiento de trending topics, así como visualización gráfica de la evolución en el tiempo de las tendencias más importantes. Además incluye un buscador de tweets populares que permita al usuario obtener aquellos tweets que más han destacado en la comunidad de usuarios de Twitter con respecto a cada trending topic.

Distintos aspectos de la aplicación implementada han sido evaluados. La evaluación del clasificador de trending topics obtiene una precisión del 90 %. La evaluación del agrupamiento de comunidades concluye que las tendencias están bien relacionadas entre sí aunque puede ser mejorable. Por último, en cuanto al buscador de tweets populares se concluye que los tweets devueltos son relevantes en su mayoría (por encima del 90 %) para cada criterio usado.

Los resultados obtenidos nos permiten concluir que la aplicación es capaz de informar al usuario de las tendencias que han surgido en Twitter a lo largo del tiempo, así como mostrarle información necesaria para que pueda percatarse sobre qué temas se ha hablado, cuáles han sido los mensajes más populares, cómo se relacionan esas tendencias entre sí y cómo ha sido la evolución en el tiempo de los trending topics más importantes.

Abstract

Nowadays, social networks have changed the paradigm of social relations and the access to information. Twitter has become a medium that is slowly pushing traditional media, like newspapers or television, into the background. The enormous amount of information that is constantly generated in this platform causes the need to group and sum up information for the average user to not overwhelmed him with information, often irrelevant. Twitter offers the possibility of grouping topics under a set of trends, called trending topics. These trending topics have a limited lifetime, i.e. once users stop talking about that topic, it will no longer be part of the list of trending topics that Twitter offers to its users. So, if the user has not been present in the application during the appearance of a trend that can be attractive to him, he will find it difficult to know about it.

The goal of this work is to develop an application that allows users to explore the set of trends that have appeared over time. The developed application also offers the categorization of trending topics, the generation of graphs which provides information about the impact over time of trending topics and a search engine of relevant tweets that allows the user to get those tweets that are most prominent in the community of Twitter users.

In this paper, we also performed an evaluation of the accuracy of the classification and clustering of trends and the popular tweets search engine. Classification accuracy of 90 % is obtained when classifying trends. Grouping communities regarding the trends observed are well interrelated but can be improved. Finally, respect to the search engine, the returned tweets are mostly relevant (over 90 %) for each criterion used.

The results allow us to conclude that the application inform about the trends that have emerged in Twitter over time and show information that allows the user to know what topics has been discussed in Twitter, what were the most popular tweets, how trends are related to each other and which has been the evolution of the main trending topics.

Palabras clave

Twitter, Tendencias, Trending topics, Clasificación, Temporalidad, Redes Sociales, Clustering.

1. Introducción

Hoy en día, Twitter[1] se ha convertido en una herramienta imprescindible para saber en tiempo real lo que sucede en la sociedad en la que convivimos, de hecho, esta plataforma se está convirtiendo poco a poco en un sustituto de los medios de comunicación tradicionales como método de acceso y publicación de la información. Ésto se debe principalmente a las siguientes características de Twitter:

- Limitación de 140 caracteres como longitud de texto o tweet máximo, lo que convierte a esta plataforma en un generador de titulares y opiniones concretas y concisas así como en un enlazador de noticias más extensas hacia otras webs.
- Capacidad de filtrar el contenido, es decir, el usuario elige a quién seguir, y éste decide si los contenidos que publica el otro usuario le resultan interesantes o no, con lo que se crean redes sociales internas de personas que tienen unos determinados gustos en común.
- Portabilidad de la plataforma, ya que cualquier usuario puede publicar contenidos en cualquier lugar y momento mediante un teléfono inteligente.
- Acceso y publicación en tiempo real de la información, y es que cualquier suceso que ocurra en alguna parte del mundo puede ser transmitido en el momento por usuarios que tengan conciencia de dicha información, en contraposición al modelo tradicional de los medios de comunicación, en el que para que las personas conocieran una información, esta debía ser recogida por un medio que le prestara atención, procesada por el mismo y presentada al usuario en un periódico o informativo televisivo.

Estas características en ocasiones suponen un lastre para el usuario, por lo siguiente:

- La limitación de caracteres provoca que una persona no sea capaz de concretar en tan poco espacio el trasfondo que le quiera dar a una noticia u opinión, ésto se soluciona la mayoría de las veces añadiendo un enlace a un contenido externo a Twitter en donde se explica de manera más detallada lo que el usuario ha querido transmitir. El usuario que consume ese tweet se encuentra con que no es capaz de contextualizar el contenido del mismo y para poder hacerlo ha de redirigirse a una web en la que probablemente tenga que leer una noticia extensa, con lo que el intento de concretar en 140 caracteres una información fracasa.
- La temporalidad de los contenidos en Twitter puede hacer que lo que el usuario estaba interesado en conocer, termine siendo ignorado si no ha estado pendiente de Twitter en ese intervalo de tiempo en el que la noticia ha sido importante.

[1] https://twitter.com/

- Twitter es principalmente una red social en la que los usuarios muestran sus opiniones respecto a algún tema, ésto puede hacer que un usuario que busque un tema de actualidad del que se hable en ese momento, se encuentre una cantidad importante de tweets sobre gente que opina sobre dicho suceso acaecido pero muy pocos tweets sobre el suceso en cuestión, lo que puede llevar a la situación de que el usuario que accede a Twitter conozca las opiniones que tiene la gente sobre una información, antes incluso de haberse formado la suya propia ya que le resulta complicado hacerse una idea exacta sobre el suceso que está dando que hablar.

El objetivo de este TFG es crear una aplicación para dar solución a los problemas presentados anteriormente, es decir, una aplicación que sea capaz de contextualizar los tweets publicados en Twitter y mostrar información al usuario para que pueda conocer el contenido de las tendencias que han surgido en Twitter mientras el usuario no estaba presente en dicha plataforma. Además, la aplicación será capaz de mostrar un conjunto de gráficas que permitan al usuario determinar la relevancia de determinadas tendencias en Twitter. Por último, la aplicación expondrá el resultado de una clasificación en categorías de tendencias y de un agrupamiento de tendencias relacionadas entre sí.

El documento está estructurado de la siguiente manera. En el capítulo 2, se hace un análisis de las redes sociales en general para, a continuación centrarnos en Twitter como ejemplo de red social sobre la que se basará el trabajo. Además se explican algunas técnicas de análisis de textos, y se presentan proyectos relacionados con la temática del presente trabajo. Se explican en el capítulo 3 en detalle las tecnologías que se usan para el desarrollo y la realización de la aplicación descrita en este documento. Durante el capítulo 4 se describirá la estructura de la aplicación desarrollada, explicando los diferentes módulos implementados. A lo largo del capítulo 5 se explican los resultados de la evaluación de la aplicación. En el capítulo 6 se detallará el trabajo concreto realizado por cada alumno. Por último, en el capítulo 7 se presentan las conclusiones obtenidas tras la realización del trabajo, así como el trabajo futuro.

2. Estado del arte

En este capítulo vamos a presentar el estado actual de los diversos campos de estudio en los que se maneja el presente proyecto. En la sección 2.1 se va a explicar qué son las redes sociales en general. Durante la sección 2.2 se tratará el análisis de las redes viéndolas como sistemas complejos. En las sección 2.3 se hablará de Twitter, sus funcionalidades, interfaces de programación y aplicaciones derivadas que extraen información de sus servicios. En la sección 2.4 se explicarán distintos algoritmos para la búsqueda de cadenas en un texto. Por último, en la sección 2.5 se va a hacer un análisis de algunos proyectos que tienen relación con la temática del presente trabajo.

2.1. Redes Sociales

Una red social es una estructura social compuesta por personas interconectadas entre sí por relaciones de diverso tipo como pueden ser amistad, parentesco o intereses comunes. Con la aparición de internet tenemos hoy en día servicios de redes sociales SNS (Social Networking Service) que trasladan estas estructuras sociales al mundo digital. Algunos ejemplos de estos SNS son Facebook[2], Twitter[3] o LinkdIn[4]. En general una SNS permite a los usuarios crear un perfil con información básica que le identifica dentro de la red y le proporciona una serie de servicios adicionales centrados en el intercambio de información como pueden ser fotos, enlaces u opiniones. Los servicios de redes sociales se pueden clasificar en (Castañeda & Gutiérrez, 2010):

- Generalistas: su propósito es permitir la comunicación entre usuarios, proporcionándoles un medio para compartir información y relacionarse. Twitter se incluiría en este tipo de red social.
- De propósito específico: los usuarios comparten información orientada a una temática o formato concreto. Ejemplos de este tipo de red social son YouTube[5] para compartir material multimedia o LinkedIn para establecer relaciones laborales.

En la actualidad el uso de los servicios de redes sociales se ha instalado y arraigado profundamente en la sociedad, sólo Facebook tiene un total de 1.110 millones de usuarios registrados y Twitter unos 500 millones de perfiles de los que 288 son usuarios activos (Statista, 2015).

En este trabajo nos centramos principalmente en Twitter ya que es una red social en auge muy presente en la vida pública de la que se pueden sacar datos muy interesantes. Además el análisis de textos tan limitados en espacio está suponiendo un reto para la comunidad académica, ya que realizar un buen estudio y análisis de tweets pueden llevar a la implementación de buenos sistemas de recomendación o herramientas de análisis sociales.

[2] https://www.facebook.com/
[3] https://twitter.com/
[4] https://es.linkedin.com/
[5] http://www.youtube.com/

2.2. Análisis de redes sociales

En esta sección vamos a introducir conceptos del estudio de las redes sociales, también llamados sistemas complejos, que serán de utilidad para comprender la estructura y las características de las redes sociales presentes en nuestro entorno. Muchas de estas características vienen descritas en el libro Network Science de Lázló Barabasi[6].

2.2.1. Teoría de grafos

En esencia, una red o grafo se compone de un conjunto de nodos o vértices y un conjunto de aristas o enlaces que los unen. El número total de nodos define el tamaño de la red, mientras que el número de aristas definirá el número de interacciones entre los elementos de la red. Los enlaces en una red pueden ser de dos tipos, dirigidos y no dirigidos, en la figura 1 se puede ver un ejemplo enlaces dirigidos, en el primer caso especificarán una relación unidireccional, como pueda ser la relación de seguidor o follower en Twitter la cual se explica en la sección 2.3.3, mientras que en el segundo caso especificarán una relación bidireccional, como pueda ser la relación de amistad de Facebook, en la que para poder relacionar dos nodos de esa red (los nodos representarán los usuarios) ambos usuarios tienen que aceptar una relación de amistad mutua.

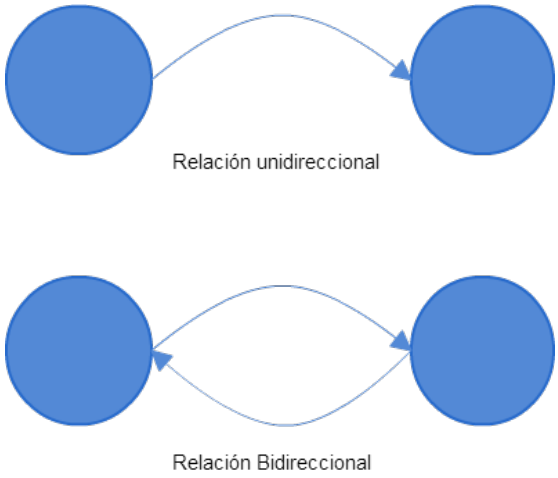


Figura 1: Tipos de enlace en una red.

Hay multitud de características o propiedades que se pueden extraer de todo grafo, como pueda ser el grado de los nodos (número de enlaces con el que está conectado un vértice), el camino mínimo entre cada par de nodos o el número

[6] http://barabasi.com/networksciencebook/

de componentes conexas del mismo entre otras. Todas estas propiedades significarán una cosa u otra dependiendo del tipo de grafo que vayamos a estudiar. Por ejemplo, si conformáramos una red en la que los nodos fueran usuarios de Twitter y los enlaces entre ellos fueran relaciones de seguidor o seguido, es decir, enlaces dirigidos, podríamos concluir estudiando el grado de entrada o salida de los nodos que aquellos con mayor grado de entrada serán aquellos usuarios con un mayor número de seguidores, mientras que aquellos con mayor grado de salida identificarán a los usuarios que sigan a más usuarios. Ésta misma propiedad para el caso de otra red social distinta dará a entender otro tipo de resultado distinto.

2.2.2. Modelos de redes

Un modelo es una abstracción de un sistema complejo real cuyo estudio estricto sería difícil de llevar a cabo debido a su complejidad, por eso a partir de un modelo dado que represente un sistema real se pueden extraer propiedades y características del sistema al que representan. En la rama de estudio del análisis de redes sociales se han llegado a elaborar diversos modelos sobre el estudio de las redes sociales, dos de los más importantes han sido:

- Modelo de red aleatoria (ERDdS & R&WI, 1959). Modelo propuesto por los matemáticos Pal Erdös y Alfred Renyi en 1959 que definen como red aleatoria aquella en la que los enlaces entre cada par de nodos se han creado de una manera completamente aleatoria. Este modelo extraía una serie de características y propiedades matemáticas de los grafos o redes, que en un principio parecía funcionar. Pero a partir de cierto momento en que se empezaron a recopilar datos de redes reales a los cuales se les aplicó este modelo, los resultados que mostraban eran incoherentes, por lo que se dedujo que el modelo de Erdös-Renyi fallaba para redes reales.

- Modelo Barabási-Albert(Barabási & Albert, 1999). Modelo de red social propuesto por los físicos Lászlo Barabási y Réka Albert en 1999, el cual fue usado para generar redes libres de escala, es decir, aquellas en las la distribución de grado sigue una ley potencial. Éste modelo se aplicó a una red que representaba los enlaces entre las distintas páginas del internet de la época y obtuvieron resultados satisfactorios. Las redes libres de escala se caracterizan principalmente por la existencia de hubs o concentradores en la red, es decir, aquellos nodos cuyo grado es significativamente mayor que el resto de nodos de la red y también por la propiedad de los mundos pequeños, lo que se refiere a que el camino mínimo entre cada par de nodo de una red libre de escala tiene un valor muy bajo, es decir, que se puede ir de un nodo a otro de la red atravesando muy pocos nodos intermedios. Twitter sigue el modelo de Barabási-Albert, ya que esta es una red libre de escala, se puede apreciar por el simple hecho de la existencia de hubs o concentradores, estos concentradores son aquellos usuarios que tienen gran cantidad de seguidores, así por ejemplo, las personalidades públicas, como

puedan ser políticos, músicos o actores entre otros tendrán un número de seguidores significativamente mayor que el resto de usuarios de Twitter.

2.2.3. Estructura de comunidades

Una de las principales propiedades que se dan en los sistemas complejos es la aparición de comunidades dentro de una red. Se entiende por comunidad un grupo de vértices altamente conectados entre sí con enlaces dispersos entre los demás grupos de vértices o comunidades (Newman, 2006). En la figura 2 se puede apreciar un ejemplo de un grafo con tres comunidades bien diferenciadas entre sí. Cada comunidad está formada por un conjunto de vértices con muchas uniones entre ellos, y cada comunidad está conectada con las demás con pocos enlaces, estos enlaces se pueden denominar puentes o bridges entre comunidades.

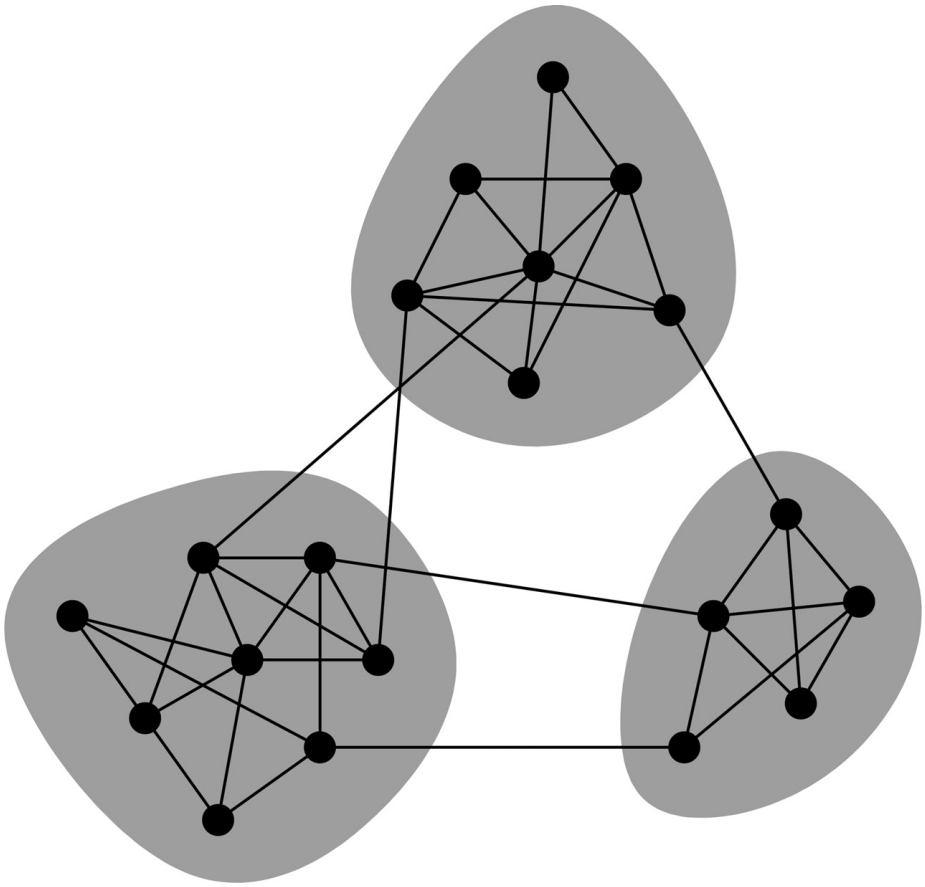


Figura 2: Ejemplo de un grafo con tres comunidades.

El detectar este tipo de estructuras puede llegar a ser un problema muy complejo, pero a partir del estudio de estas comunidades en redes sociales se

pueden abstraer datos muy significativos, ya que en la vida real, las personas tendemos a agruparnos entre nosotros con distintos grados de relación: familia, amistad, gustos o trabajo. Así, si a partir de una red dada podemos extraer las comunidades que la conforman podremos estudiar las relaciones entre dichas estructuras y sacar conclusiones muy interesantes sobre las mismas. Así por ejemplo en una red cuyos nodos representen grupos de música, los enlaces unirán dos grupos si algún disco de éstos ha sido comprado por la misma persona, se podrá apreciar al realizar un estudio de la estructura de comunidades, que el grafo se dividirá, si posee información suficiente, en comunidades que representen los distintos géneros de música y los puentes entre estas comunidades estarán formados por aquellos grupos de música o géneros que estén relacionados con ambas comunidades, por ejemplo, si suponemos la existencia de las comunidades formadas por el género del heavy metal, el rock y el pop, la comunidad de rock servirá de puente entre las de heavy metal y pop.

El problema de dividir un grafo en distintas comunidades puede llegar a ser muy complejo, algunos algoritmos para resolver este problema pueden llegar a ser NP-completos. Dos de los métodos más usados son:

- El algoritmo de Newman y Girvan (Newman & Girvan, 2004), que detecta comunidades en un sistema complejo a base de eliminar progresivamente aristas del grafo original hasta un cierto punto, llegando entonces a detectar que las componentes conectadas del grafo resultante tras aplicar el algoritmo son las comunidades del mismo.
- El algoritmo de Louvain (Blondel et al., 2008), que detecta comunidades en un grafo a base de optimizar la medida de modularidad durante su ejecución. La modularidad es un valor entre 1 y -1 y mide la densidad de enlaces de dentro de las comunidades y de fuera de las mismas.

2.3. Twitter

En esta sección se va a presentar qué es Twitter, su historia y sus principales características, así como un glosario de términos comunes que conforman una jerga particular en el ámbito de la aplicación.

2.3.1. Historia

Twitter es una red social creada en 2006 en Estados Unidos, con la intención de facilitar el intercambio de mensajes breves entre sus usuarios. El éxito de Twitter se centra en una limitación del servicio ofrecido a los usuarios: los mensajes (tweets) que pueden publicar los usuarios tienen una extensión limitada a 140 caracteres. De esta forma se consigue un intercambio de información fluida, concisa y rápida. En un principio se creó con la idea de comunicar a pequeños grupos de personas partiendo de la idea de los antiguos SMS de los dispositivos móviles.

El crecimiento de Twitter desde que se fundó ha sido exponencial, ha pasado de tener a principios del año 2010 unos 30 millones de perfiles activos a tener

288 millones de perfiles activos en el último trimestre del año 2014 (Statista, 2015).

2.3.2. Interacciones entre usuarios y glosario de términos comunes

Las relaciones entre usuarios son de dos tipos: following y follower (seguidor y seguido). Se denomina following al conjunto de usuarios a los que un miembro sigue, suscribiéndose a sus publicaciones. Del mismo modo estos usuarios seguidos pasan a tener un nuevo follower o seguidor. La situación en la que dos usuarios se siguen mutuamente es la se podría considerar como la relación clásica de "amistad" en otras redes sociales.

El lugar en el que aparecen los contenidos de los usuarios es el timeline o línea de tiempo. Cada miembro de esta red dispone de un espacio personal o perfil donde se muestran sus mensajes ordenados cronológicamente. Además también se visualizan los mensajes de los usuarios que el usuario ha decidido seguir.

Las formas de interacción entre miembros de esta red son las siguientes:

- Menciones: Es la forma en la que un usuario se dirige a otro, u otros, con el fin de iniciar una conversación, o notificarles algo en particular. Para llevar a cabo este proceso se precede el nombre de usuario con el símbolo @. Si la mención es la respuesta a otro tweet se denomina reply.

- Mensaje privado: A diferencia del anterior el contenido del mensaje sólo es visible para el destinatario del mensaje.

- Hashtags: Palabras o etiquetas que comienzan con el símbolo # y se emplean para agrupar mensajes cuyo contenido tienen un tema común. De esta forma se puede conocer cuál es la opinión de los usuarios sobre temas concretos.

- Trending Topics: Son las tendencias o temas de actualidad de los que los usuarios están hablando en un determinado momento. Se pueden visualizar a nivel mundial o bien restringido a zonas geográficas. Aunque la mayoría de trending topics están identificados por un hashtag, no todas las tendencias tienen este formato. La manera de generar éstos trending topics es mediante un algoritmo secreto que usa Twitter. Aunque este algoritmo es secreto, si se han dado algunas claves de cómo funciona, como por ejemplo que prevalece el número de usuarios twitteando al número de tweets o que se tiene preferencia por aquellas tendencias que son novedosas y que están ocurriendo en el momento (Tweetsmarter, 2011; Bufferapp, 2011; Ignitesocialmedia, 2012).

- Retweet: Acción de copiar un tweet o mensaje de otro usuario al perfil propio, añadiéndolo al timeline o línea de tiempo.

- Hacer Follow: Acción de seguir un perfil de otro usuario, a partir de entonces, las actualizaciones del perfil de ese otro usuario las podrá ver el usuario que ha realizado la acción de hacer follow.

2.3.3. Características

Twitter es una red social no dirigida, no es necesario que exista una relación o conexión bidireccional entre dos usuarios, es decir es asimétrica, asemejándose al mundo real donde la comunicación entre entidades puede ser unidireccional, como pasa, por ejemplo, en los medios de comunicación.

En (Kwak et al., 2010) se hace un análisis de Twitter usando diversos algoritmos de teoría de grafos. Se estudia un grafo generado a partir de la recolección de datos de 41,7 millones de perfiles de Twitter, 1,47 billones de relaciones sociales, 4.262 trending topics y 106 millones de tweets. Una vez representada la red y aplicados los distintos algoritmos de teoría de grafos, se obtienen algunas conclusiones muy interesantes sobre Twitter:

- Reciprocidad: Según el estudio, un 77,9 % de los pares de usuarios están relacionados en una única dirección, es decir, un usuario sigue a otro, pero el otro no sigue al primero, sólo un 22,1 % de las relaciones son bidireccionales, que es la relación típica de amistad en otras redes sociales como Facebook. Según los datos anteriores se puede afirmar que Twitter es una red social con bajo nivel de reciprocidad. Otro dato interesante sacado del estudio de las relaciones es que un 67,6 % de los usuarios estudiados no son seguidos por ninguno de sus seguidores, con lo que los autores de dicho artículo conjeturan que para este alto porcentaje de usuarios, Twitter es más una herramienta de información que una red social.

- Homofilia: Es la tendencia a relacionarse entre sí, aquellos usuarios con gustos similares. El estudio muestra que los usuarios cuya relación es recíproca están geográficamente cercanos. Esto indica que el contexto cultural y social es muy importante a la hora de establecer relaciones en Twitter.

- La importancia del retweet. El retweet es un mecanismo que ofrece Twitter para dotar al usuario del poder de difundir a todos sus seguidores un tweet específico. Esta característica de Twitter hace que aquella información a la que un usuario le parezca intersante se extienda de manera exponencial por la red al llegar a todos sus seguidores. La temporalidad está muy presente a la hora de hacer retweets ya que la mitad del total de retweets que llega a conseguir un tweet se produce en la primera hora desde que el tweet original se publicó, y un 75 % del total se produce a lo largo de las primeras 24 horas, de lo que se deduce que el tiempo de vida de actualidad de un tweet viene a ser aproximadamente de un día.

Twitter ofrece la posibilidad de etiquetar sus mensajes mediante hashtags que, bien usados, ayudan al lector a contextualizar el mensaje en cuestión. Twitter provee funcionalidad para poder ordenar jerárquicamente los hashtags según sean utilizados por los usuarios en el tiempo, ésto provoca la aparición de una lista de los temas más hablados del momento (los trending topics) que está formado en su mayoría por hashtags, aunque también es habitual encontrar tendencias identificadas por un conjunto de palabras que no tengan formato de hashtag. Los trending topics son el conjunto de los temas de los cuales los

usuarios de Twitter están mayoritariamente hablando en un preciso momento. Ésta lista de trending topics tiene un componente de temporalidad muy elevado, es decir, que una vez se deja de hablar de un tema o bien ha habido otros que han irrumpido con más fuerza, este hashtag desaparece de la lista de los trending topics. Ésto último puede resultar perjudicial para un usuario, ya que, si en ese intervalo de tiempo en el que el suceso ha ocurrido, dicho usuario no ha accedido a la aplicación, no tendrá constancia del suceso mencionado. El usuario podría solucionar este problema buscando específicamente el suceso en cuestión a lo largo de su timeline, pero debido a la ingente cantidad de información que se genera de manera continua en Twitter esta solución podría resultar tediosa. Existe por tanto la posibilidad de que el usuario no termine conociendo la actualidad de aquello que le interesa.

2.3.4. Temporalidad de la información

Twitter ofrece un servicio de streaming, es decir, presenta la información exactamente en el momento en que se genera y de manera continua, a esta característica la llamaremos temporalidad de la información.

Twitter es una red social con un alto contenido de temporalidad, es decir, la mayoría de los sucesos que los usuarios twittean en la aplicación suelen ser aquellos que están ocurriendo en tiempo real. Por ejemplo durante un partido de fútbol suele surgir un determinado hashtag que identifica el partido, ese hashtag o bien lo crea la propia comunidad de usuarios y prevalece aquel que es más identificativo o popular sobre el tema, o bien la propia televisión suele facilitar un hashtag para que todos los usuarios que estén viendo el partido converjan rápidamente a un único hashtag y se pongan a twittear sobre el suceso en cuestión una mayor cantidad de usuarios, para así poder llegar a ser trending topic y producir una mayor repercusión en ese instante en la comunidad de usuarios de Twitter, lo que para la televisión podría repercutir en una mayor cantidad de espectadores. Los usuarios de Twitter que están viendo el partido se suelen dividir en dos grandes grupos: los usuarios que producen contenido y los usuarios que sólo consumen contenido, es decir, aquellos usuarios que estén twitteando de manera activa durante la retransmisión del partido pertenecerán al primer grupo, y aquellos que sólo consuman información, es decir, que apenas publiquen y sólo lean lo que los usuarios productores estén publicando pertenecerán al segundo grupo (Hipertextual, 2010). Para ambos grupos de usuarios surge el mismo problema, y es que la producción de contenido se hace de manera constante en tiempo real, y cuanto más usuarios estén twitteando sobre el partido mayor será la cantidad de información publicada por unidad de tiempo, es decir, que mientras un usuario está leyendo o publicando contenido, en ese intervalo de tiempo que ha empleado para ello, se han podido llegar a publicar varias decenas de tweets. Ésto produce que alguien que se acaba de conectar a Twitter y quiera informarse de los acontecimientos que han sucedido hasta ese momento en el partido pueda encontrarse con que no es capaz de encontrar lo que está buscando debido a la inmensa cantidad de tweets publicados, muchos de ellos opiniones personales o ruido que se cuela en medio de la tendencia (Cruz, 2014).

Una de las curiosidades en las que Twitter demuestra su alto grado de temporalidad está en que en un principio la aplicación, invitaba a los usuarios a publicar tweets mediante la pregunta ¿qué estás haciendo?, en la actualidad, esa pregunta se ha cambiado por ¿qué está pasando? (Cruz, 2014). Con esto se deduce que la aplicación está invitando al usuario a publicar sobre los hechos que están ocurriendo en ese momento a su alrededor, lo que haya pasado ya no interesa, lo que importa es el «aquí y el ahora».

2.3.5. Información disponible

Además de la información básica que provee Twitter (el tweet). Twitter ofrece también la siguiente información derivada de los tweets, de los usuarios de la aplicación y del entorno de la aplicación en general:

- Como se puede ver en la figura 3 los elementos básicos que conforman un tweet son: cuerpo, creador, fecha y hora en las que se creó el tweet, número de Retweets (cantidad de veces que dicho mensaje ha sido copiado en el perfil de otro usuario) y número de favoritos (cantidad de veces que dicho tweet ha sido seleccionado como favorito por los demás usuarios).
- Twitter ofrece en su interfaz la lista de los trending topics, los cuales se ordenan de manera jerárquica dependiendo de lo relevantes que sean para los usuarios, es decir, aquellos que más tweets generan se colocarán en las primeras posiciones. Ésta lista de trending topics se puede filtrar a nivel global, nacional o metropolitano.

Figura 3: Ejemplo de tweet.

- En la figura 4 se puede ver el conjunto de características que nos permite conocer Twitter sobre un usuario de su aplicación[7]:

[7] Esta información por defecto es pública aunque los usuarios pueden especificar ciertos criterios de privacidad en la aplicación

- Foto de perfil con la que el usuario se identifica.
- Nombre de usuario, el cual puede no ser único y el usuario puede cambiar a su gusto.
- Identificador de usuario para identificar de forma unívoca al usuario, es la información que se usa para realizar las menciones.
- Tweets que el usuario ha publicado, creado o retwitteado. Ésta lista se puede filtrar de manera que sólo muestre los tweets publicados, las fotos y vídeos o todos en conjunto.
- Lista de favoritos.
- Seguidores.
- Siguiendo (conjunto de otros perfiles a los que el usuario ha decidido seguir).

Figura 4: Ejemplo de un usuario de Twitter.

Toda esta información accesible desde Twitter permite crear diversas aplicaciones y servicios relacionados con la red social de los que hablaremos en la sección 2.3.7.

2.3.6. API de Twitter

Uno de los motivos de la expansión de Twitter ha sido la existencia de APIs gratuitas que proporciona la empresa, que han propiciado la creación de software de terceros que permite conectarse y manejar datos de la aplicación.

El API de Twitter está limitado ya que el acceso a la aplicación está limitado a 150 o 350 solicitudes por hora dependiendo si registramos o no la aplicación en el apartado de desarrolladores de Twitter. Twitter usa OAuth[8] para tener acceso a algunas APIs, OAuth es un protocolo abierto para permitir acceso seguro de manera simple y estándar. Hay tres APIs principales proporcionadas por Twitter:

[8] http://oauth.net/

- **Search API**[9]: Se encarga de suministrar los tweets buscados de hasta hace 7 días, con un máximo de 1.500 tweets. En esta API es posible filtrar por cliente utilizado, lenguaje y localización.

- **Rest API** [10]: Es una API web que funciona por HTTP a la cual accedemos a partir de URLs que devuelven contenido en formato JSON, XML, HTML, etc. A diferencia de la Search API no hay limitación temporal, pero sí una limitación del número de resultados devueltos establecido en 3.200 tweets.

- **Streaming API** [11]: Permite recibir información en tiempo real. Los contenidos devueltos tienen formato JSON. En esta API se pueden obtener muestras aleatorias o un filtrado por palabras clave o usuarios, aunque también existen métodos más interesantes como puedan ser el poder obtener el caudal de tweets, o filtrar solo por tweets con enlaces o tweets con retweets.

2.3.7. Herramientas de análisis de datos de Twitter

Hemos seleccionado un total de tres herramientas (Trendinalia, Tweet-Tag y Topsy) que brindan una funcionalidad interesante extrayendo y analizando los datos que se pueden obtener desde Twitter. Estas aplicaciones tienen objetivos muy parecidos aunque con matices, pero en general, son aplicaciones pensadas para suplir alguna falta de funcionalidad en Twitter o bien presentar la información al usuario de manera ligeramente distinta a como Twitter lo hace. De hecho, actualmente Twitter está empleando y absorviendo muchas de estas aplicaciones para incluirlas en su funcionalidad de serie para suplir sus carencias.

- Trendinalia[12] es un servicio web que proporciona una monitorización de los diferentes hashtags que se convierten en trending topic en algún momento del día. Dicha aplicación proporciona información diversa sobre el trending topic en cuestión, su duración, ubicación y gráficas para comparar unas tendencias con otras. En esta aplicación se puede filtrar por día y lugar ofreciendo una visión más local de la actualidad. Esta aplicación viene a suplir lo que ya dijimos en apartados anteriores sobre la temporalidad de los contenidos en Twitter, y es que, esta web mantiene un registro sobre los trending topics a lo largo del tiempo. En la figura 5 se presenta una captura de la aplicación donde se puede ver la lista de trending topics y la duración de los mismos como tendencia en Twitter. A la izquierda de la imagen se puede ver que la aplicación permite filtrar resultados por país, fecha e incluso la ciudad a la que pertenecen. A la derecha de la misma se puede apreciar un conjunto de gráficas que hacen más explicativos los resultados mostrados.

[9] https://dev.twitter.com/rest/public/search
[10] https://dev.twitter.com/rest/public
[11] https://dev.twitter.com/streaming/overview
[12] http://www.trendinalia.com/

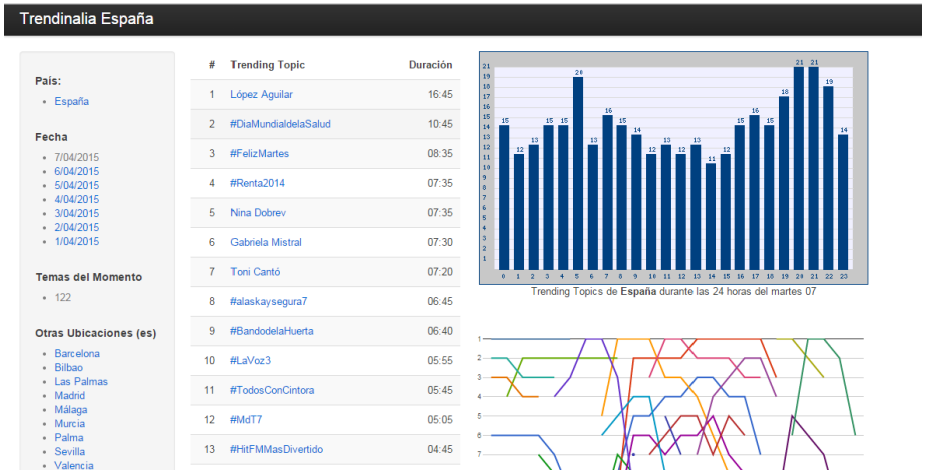


Figura 5: Aplicación Trendinalia.

- Tweet-Tag[13] es una web que permite al usuario introducir un hashtag y una fecha como parámetro de búsqueda para mostrar la información derivada de dicho hashtag o tendencia. Devuelve una información más completa que Trendinalia, ya que muestra la cantidad de usuarios que han usado ese hashtag, la audiencia del mismo o los tweets más relevantes o de la última hora. Ofrece una interfaz muy cuidada con gráficos explicativos que relacionan el número de tweets respecto a la cantidad de usuarios que twittean, ofreciendo además una lista de palabras clave que representan las más usadas por los usuarios a la hora de twittear. En otros aspectos Tweet-Tag pierde funcionalidad respecto a Trendinalia, ya que el número de hashtags que se pueden monitorizar es reducido y a la hora de buscar el hashtag y la fecha no da recomendaciones sobre que debe poner el usuario, es decir, el cliente de antemano tiene que saber que hashtag específico y correcto debe buscar y la fecha en la que se produjo. Es una aplicación más enfocada al mundo empresarial o televisivo, donde conocer el impacto temporal de un hashtag en concreto. Esta aplicación sin embargo tiene un límite de uso de hasta tres monitorizaciones, almacenándo los datos un máximo de una semana. En la figura 6 se puede ver una captura de pantalla de la aplicación, donde se puede ver la información asociada a la monitorización del trending topic #FelizMiercoles. A la izquierda de la imagen se puede apreciar el conjunto de tweets que se están publicando en ese momento, y a la derecha de la misma, los datos asociados a la monitorización de la tendencia, la gráfica por horas y la cantidad de participantes que están tuiteando sobre esa tendencia.

[13] http://www.tweet-tag.com/

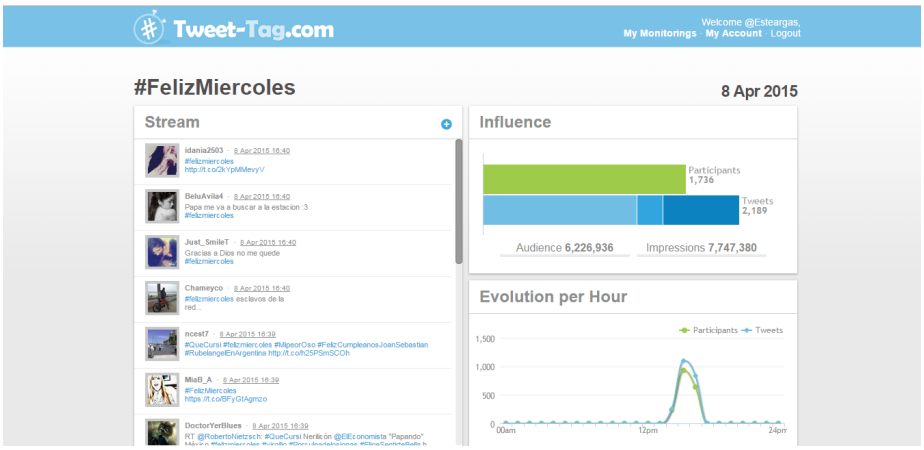


Figura 6: Aplicación Tweet-tag

- Topsy[14] es un buscador como el de la API search de Twitter, al que le han añadido funcionalidad y una interfaz cuidada. Lo más relevante de esta aplicación es que realiza un análisis de sentimiento de los tweets, ofrece de manera visual una relación positivo/negativo del hashtag buscado, así como gráficas en tiempo real sobre la repercusión del hashtag buscado. Además esta web permite al usuario filtrar por contenidos, es capaz de ofrecer al usuario sólo los tweets, los enlaces o las fotos y vídeos que los usuarios han publicado referenciando la tendencia buscada. En la imagen 7 se puede ver los tweets más relevantes respecto a un determinado trending topic, a la izquierda de la misma se puede filtrar por hora, lenguaje e incluso mostrar aquellos resultados que sólo contengan lo que nos interesa, ya sean fotos, enlaces o vídeos. Además como principal característica la aplicación muestra encima de la lista de tweets relevantes un análisis de sentimiento sobre la tendencia buscada.

Podemos concluir que aunque estas tres aplicaciones muestran información relevante sobre las tendencias de Twitter y llevan un registro de las mismas, a cada una de ellas les falta funcionalidad que la aplicación desarrollada en este trabajo quiere subsanar.Aunque Trendinalia lleva el registro de todas las tendencias de cada día, la duración de las mismas como trending topics en Twitter no está integrada con Twitter. Además, en esta aplicación si el usuario quiere explorar cada tendencia listada en la aplicación es redirigido al propio Twitter en otra pestaña. Tweet-tag sólo permite informarse y mantener un registro sobre tres tendencias a la vez. Además el usuario ha de saber de antemano que tendencia va a buscar, ya que no lleva un registro de las mismas. Topsy, sin embargo, aunque es una aplicación muy completa, y muestra información muy variada al usuario, no clasifica las tendencias en categorías y también tiene el

[14] http://topsy.com/

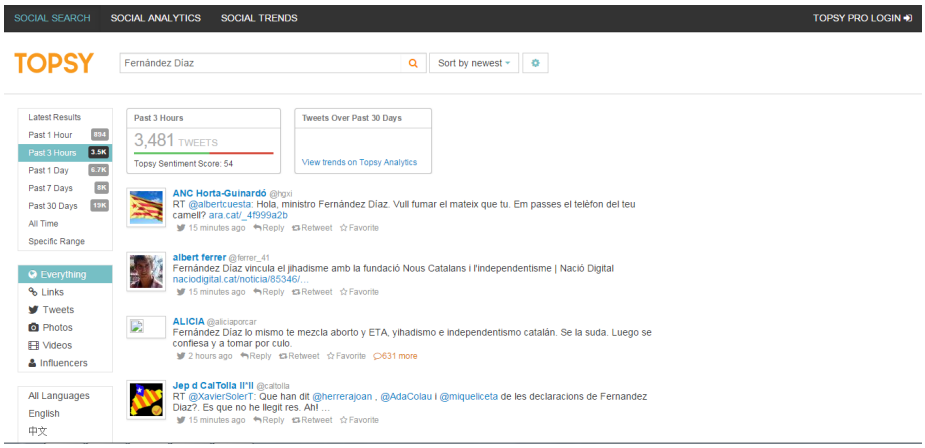


Figura 7: Aplicación Topsy

problema de Trendinalia de la integración con Twitter, y es que al hacer click sobre cualquier información referenciada en la aplicación, esta redirige al usuario a Twitter y deja de mostrarse la aplicación.

2.4. Algoritmos de búsqueda de subcadenas

Uno de las principales tareas del procesado de textos consiste en la búsqueda de subcadenas en un texto dado. Dos de los algoritmos más usados para realizar esta tarea son:

- Algoritmo Knuth-Morris-Pratt (KMP)(Knuth et al., 1977): se basa en usar técnicas de precondicionamiento con autómata para poder encontrar de manera eficiente la ocurrencia de un patrón p en una cadena dada con coste en tiempo de preprocesado y de ejecución lineal. En la figura 8 se puede apreciar el pseudocódigo del algoritmo KMP.

- Algoritmo de Rabin-Karp (Cormen et al., 2001; Karp & Rabin, 1987): utiliza también técnicas de precondicionamiento, aunque en menor medida que el KMP, y es capaz de encontrar las distintas apariciones de un patrón p de longitud m en un texto dado de longitud n en el caso mejor en tiempo O(n+m) , mientras que en el caso peor O(nm) con un coste O(p) en espacio. En la figura 9 se presenta en pseudocódigo el algoritmo.

La principal ventaja que aporta el algoritmo Rabin-Karp es que se comporta mejor en cuanto a la búsqueda de múltiples patrones en una cadena, además dada su sencillez resulta más fácil adaptar este algoritmo que el KMP para que pueda hacer esa búsqueda de patrones múltiples, el código del algoritmo adaptado a la búsqueda de varios patrones puede verse en la figura 10. El algoritmo de Rabin-Karp es ampliamente usado en aplicaciones de detección de plagio (Stoimen, 2012).

```
KMP-MATCHER(T, P)
 1  n = T.length
 2  m = P.length
 3  π = COMPUTE-PREFIX-FUNCTION(P)
 4  q = 0                                  // number of characters matched
 5  for i = 1 to n                         // scan the text from left to right
 6      while q > 0 and P[q + 1] ≠ T[i]
 7          q = π[q]                       // next character does not match
 8      if P[q + 1] == T[i]
 9          q = q + 1                      // next character matches
10      if q == m                          // is all of P matched?
11          print "Pattern occurs with shift" i − m
12          q = π[q]                       // look for the next match
```

Figura 8: Algoritmo KMP de búsqueda de un patrón.

```
RABIN-KARP-MATCHER(T, P, d, q)
 1  n = T.length
 2  m = P.length
 3  h = d^(m−1) mod q
 4  p = 0
 5  t_0 = 0
 6  for i = 1 to m                         // preprocessing
 7      p = (dp + P[i]) mod q
 8      t_0 = (dt_0 + T[i]) mod q
 9  for s = 0 to n − m                     // matching
10      if p == t_s
11          if P[1..m] == T[s + 1..s + m]
12              print "Pattern occurs with shift" s
13      if s < n − m
14          t_{s+1} = (d(t_s − T[s + 1]h) + T[s + m + 1]) mod q
```

Figura 9: Algoritmo de Rabin-Karp de búsqueda de un patrón.

2.5. Proyectos relacionados

En esta sección se presentan tres trabajos que están en consonancia con el objetivo del presente proyecto.

```
1.  function RabinKarpSet(string s[1..n], set of string subs, m):
2.      set hsubs := emptySet
3.      foreach sub in subs
4.          insert hash(sub[1..m]) into hsubs
5.      hs := hash(s[1..m])
6.      for i from 1 to n-m+1
7.          if hs ∈ hsubs and s[i..i+m-1] ∈ subs
8.                  return i
9.          hs := hash(s[i+1..i+m])
10.     return not found
```

Figura 10: Algoritmo Rabin-Karp de búsqueda de múltiples patrones.

2.5.1. Itafy

Ésta aplicación (Anguita & Lorenzo, 2014) tiene como objetivo visualizar información extraída de Twitter en tiempo real y categorizar en tiempo real tweets además de representar de manera visual y atractiva dicha información mediante gráficos. El sistema creado ofrece las siguientes funcionalidades:

- Categorizador de textos: Mediante técnicas de procesamiento de lenguaje natural se clasifica la temática de los tweets que recogen en tiempo real mediante la Streaming API de Twitter, en concreto usan herramientas especializadas en el tratamiento de datos como WEKA[15] y Lucene[16]. Se definen un conjunto de tres categorías para clasificar los tweets: deportes, política y otros (categoría para aquellos tweets que no pertenezcan a las otras dos).

- Detector de género: Identifica el género del autor de los tweets recogidos por la aplicación. Para ello utilizan una base de datos con los nombres más usados en español aplicando técnicas de comparación y otros refinamientos como son la detección de diminutivos.

- Visualización de la información: La aplicación muestra mediante servicio web los resultados obtenidos. El usuario se conecta a la página web en la cual reside la aplicación y recibe en tiempo real en un mapa geográfico los distintos tweets que se están generando en el momento, su ubicación,

[15] http://www.cs.waikato.ac.nz/ml/weka/
[16] https://lucene.apache.org/core/

autor y cuerpo del tweet. Además la aplicación web implementa una API con la que el usuario se puede conectar y visualizar también el resultado de la categorización de los tweets que ha estado recolectando hasta entonces.

En la figura 11 se muestra una captura de la aplicación. Se puede apreciar que la aplicación muestra un mapa del mundo en el que van apareciendo marcas que representan los tweets que se están produciendo en ese preciso instante, al seleccionar una marca podremos ver el tweet en cuestión, tal como muestra la figura. Además la aplicación muestra unas estadísticas generadas en tiempo real sobre el género de las personas que están twitteando en ese momento.

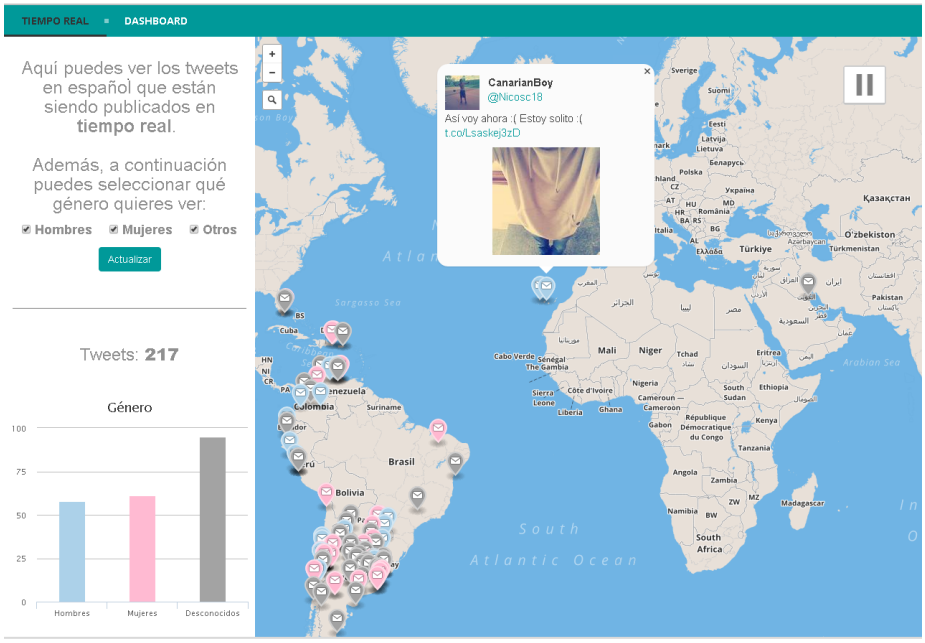


Figura 11: Aplicación Itafy

2.5.2. Diseño e implementación de un sistema para el análisis y categorización en Twitter mediante técnicas de clasificación automática de textos

Este trabajo (Alcázar Jaén et al., 2013) diseña e implementa un prototipo para capturar tweets para su posterior análisis y categorización usando técnicas de procesamiento del lenguaje natural.

Se apoya principalmente en la librería Tweepy[17] la cual da soporte para acceder a los datos que nos ofrece Twitter desde sus API. Así mismo el autor

[17] http://www.tweepy.org/

usa Django[18] para implementar su código como aplicación web. En cuanto herramientas de procesamiento de lenguaje natural, usan NLTK[19] como software de tratamiento de textos. El prototipo que presentan funciona de la siguiente manera:

1. La aplicación obtiene un conjunto de tweets publicados en Twitter mediante la Streaming API.
2. Detecta el idioma de los tweets usando el corpus de idiomas que la herramienta NLTK ofrece.
3. Realiza una estructuración de cada tweet, dividiendo el mismo en unidades más pequeñas de información, como puedan ser hashtag, mención o emoticonos.
4. Se categorizan los tweets.
5. Se realiza un análisis de sentimiento usando un clasificador bayesiano ingenuo.

2.5.3. Desarrollo de un clasificador jerárquico multi-etiqueta de tendencias de Twitter

Éste trabajo (Fiaidhi et al., 2013), lleva a cabo una clasificación de los trending topics que produce Twitter por temas.

En un primer momento el trabajo propone recolectar un conjunto significativamente amplio de tweets usando la Streaming API de Twitter con una serie de filtros a las búsquedas de los mismos para así poder devolver aquellos más relevantes y pertenecientes a los llamados trending topics. Este conjunto de datos lo almacenan en ficheros .csv. Los datos son recogidos en diferentes intervalos de tiempo y en diferentes ciudades de Norteamérica. Los autores identificaron un conjunto de 12 clases para etiquetar los tweets y poder llevar a cabo su clasificación: política, educación, sanidad, marketing, música, noticias, deportes y entretenimiento, ciencia y tecnología, mascotas, comida, familia y otros. A continuación, se sirven de la Search API de Twitter para buscar tendencias y asignar manualmente una etiqueta a las tendencias. Una vez hecho esto se tendrá entonces un conjunto amplio de datos para poder entrenar de manera efectiva los clasificadores que se usarán más tarde.

Una vez recogida y etiquetada la información, esta pasa por una serie de clasificadores, como puedan ser el Naïve Bayes o el Support Vector Machine entre otros, y tras pasar por el proceso de clasificación de cada clasificador, se elige de entre todos ellos el mejor clasificador, es decir, aquél que ha obtenido mejores resultados de precisión, tras lo cual, con éste clasificador entrenado, que ha sido elegido entre el resto por su eficacia se obtiene un clasificador de trending topics para cada categoría antes relatada.

[18] https://www.djangoproject.com/
[19] http://www.nltk.org/

2.5.4. Conclusiones

Estos tres proyectos aproximan de manera diferente el tratamiento de información desde Twitter, en los trabajos presentados en 2.5.1 y 2.5.3 se usa la Streaming API de Twitter para el procesamiento de la información en tiempo real mientras que en el trabajo referido en la sección 2.5.2 se capturan los tweets mediante Tweepy, librería de Python que sirve de envoltorio para la API REST de Twitter. Todos estos proyectos tratan de categorizar los tweets generados, olvidándose y dejando de lado la clasificación de los trending topics, por lo que nuestro proyecto se enfocará de otra manera para poder categorizar en un primer momento los trending topics los cuales ya contienen bastante información sobre el tema del que tratan, y una vez hecho esto poder asignar dicha categoría a los tweets que referencien dicha tendencia. Además, los proyectos presentados, de cara al usuario no llegan a ser muy atractivos, porque presentan información al usuario, sin permitir al mismo poder interactuar con ella ni sacar partido de la misma, nuestra aplicación permite al usuario conocer e interactuar con las tendencias de Twitter.

3. Tecnologías usadas

En este capítulo explicaremos en detalle las diferentes tecnologías usadas para llevar a cabo el proyecto. En esencia se usa Twitter4j (una librería que encapsula las API de Twitter), MongoDB (una base de datos no relacional orientada a documentos que usamos para almacenar los datos), Gephi (herramienta para la visualización y estudio de grafos) y JavaFX (un conjunto de librerías Java para la creación de interfaces de usuario).

3.1. Twitter4j[20]

Librería no oficial bajo licencia Apache 2.0 de la API de Twitter creada por el japonés Yusuke Yamamoto la cual proporciona una serie de métodos escritos en Java para poder acceder a la información que proporciona la API de Twitter.

Esta librería también proporciona soporte para la conexión mediante OAuth (Open Authorization) un protocolo abierto que permite una conexión segura a los datos protegidos de la API de Twitter.

Lo que hace esta librería principalmente es transformar las llamadas a la API de Twitter, las cuales se hacen mediante peticiones JSON, a código y métodos Java evitando al desarrollador tratar las peticiones JSON.

Hemos decidido seleccionar esta librería para desarrollar nuestra aplicación principalmente porque proporciona una implementación en Java, lenguaje en el que está realizado el proyecto además, esta librería es muy usada y tiene gran respaldo de la comunidad de desarrolladores, así como documentación, tutoriales y ejemplos explicativos.

3.2. MongoDB[21]

Base de datos no relacional orientada a documentos representados en formato JSON. Elegimos MongoDB como sistema para almacenar los datos de nuestra aplicación debido a que la propia estructura de este tipo de bases de datos no relacional está pensada para facilitar la lectura masiva de datos. Además, MongoDB posee las siguientes características que consideramos útiles para la realización de nuestro proyecto:

- Facilidad de instalación y uso. Existen multitud de tutoriales y documentación disponible en internet sobre su instalación y uso.
- Ofrece un modelo de datos muy flexible ya que no es necesario la creación de un esquema fijo y estricto previa creación de la base de datos. En las bases de datos relacionales tradicionales como pueden ser MySQL es obligatorio especificar de manera previa a la inserción de datos la estructura de las tablas mediante el estudio y la creación de unos diagramas relacionales los cuales representan la estructura de las tablas que compondrán

[20] http://twitter4j.org
[21] https://www.mongodb.org/

la base de datos. Aunque en MongoDB ésto no es necesario, si es recomendable tener pensada la estructura jerárquica mediante la cual se van a almacenar los datos, ésto nos aporta mucha flexibilidad ya que cuando hemos considerado añadir más o menos información a los documentos que guardábamos sólamente teníamos que añadir o quitar un campo de dicho documento, mientras que con las bases de datos relacionales ésto podría dar lugar a un conflicto de relaciones que podrían afectar a toda la base de datos.

- Es muy escalable por lo que favorece el almacenamiento masivo de datos mediante las técnicas de replicación y sharding consistente en poder dividir la base de datos entre distintos servidores, y es que además de almacenar los datos en formato binario (BSON) lo que compacta aún más la BD, es posible mediante MongoDB de manera sencilla replicar y particionar la BD para poder ser almacenada en varios servidores. Ésta característica resultaría muy útil si quisiéramos ampliar el alcance de nuestro proyecto y tuviéramos que necesitar más de un servidor para recolectar y tratar los datos.

- La facilidad de la creación de índices para realizar búsquedas más rápidas ya que de manera inmediata con una simple instrucción MongoDB ofrece la posibilidad de indizar de manera diferente los campos que queramos de los documentos, ofreciendo una acceso muy rápido a dichas consultas.

- Es una base de datos de código abierto que esta creciendo muy rápidamente debido a la gran comunidad que la respalda y que poco a poco está siendo utilizada en una gran cantidad de sistemas, algunos tan importantes como pueden ser eBay y Foursquare (Genbetadev, 2014).

3.3. Gephi[22]

Gephi es una herramienta bajo licencia GPL para la visualización y el análisis de redes, sistemas complejos o cualquier tipo de estructura que se pueda representar mediante grafos.

Gephi es un software de código abierto y gratuito que se desarrolló inicialmente en la UTC en Francia y en la actualidad es ampliamente usado por profesionales de diversos sectores como la informática, la biología o la sociología.

Esta herramienta está escrita en lenguaje Java y ofrece las siguientes funcionalidades para el estudio de grafos:

- Creación de grafos a partir de diversos formatos de ficheros como .gdf o .csv además de poder crear grafos de manera manual en la interfaz.

- Representación del grafo en distintas disposiciones o "layouts".

- Aplicación de distintos algoritmos para calcular métricas interesantes para el estudio de la estructura del grafo, como pueda ser el cálculo del grado,

[22] http://gephi.github.io/

las clases modulares, el coeficiente de clustering, caminos mínimos, page rank, etc.

Gephi ofrece la posibilidad de descargar su código fuente para poder usar y modificar sus funcionalidades como una librería.

3.4. JavaFX[23]

JavaFX es una herramienta software destinada a la creación de RIAs (Rich Internet Applications) (Merayo, 2011). Se introdujo en la versión 8 de Java para sustituir a Swing como herramienta para crear interfaces gráficas. Actualmente, junto con Adobe Flash y Microsoft Silverlight constituye una de las principales plataformas para diseñar interfaces modernas de diseño atractivo.

JavaFX recomienda la instalación de una aplicación llamada Scene Builder[24] la cual nos permite de manera fácil, rápida e intuitiva agregar elementos a nuestra interfaz, sin necesidad de programar desde cero los elementos que componen dicha interfaz. Dicha aplicación nos permite asociar un controlador a la vista de la interfaz, por lo tanto, promueve el uso del patrón de diseño Modelo-Vista-Controlador, permitiendo además aplicar hojas de estilo .css a los elementos de la aplicación y asociar eventos de usuario (pulsar un botón, arrastrar un elemento...) a procedimientos java de una manera mucho más sencilla y eficaz que lo que se podría conseguir con Java Swing.

Esta tecnología además permite incluir un motor web dentro de la propia aplicación, es decir, que podremos navegar en internet desde nuestra aplicación sin necesidad de recurrir a navegadores web externos.

[23] http://docs.oracle.com/javase/8/javase-clienttechnologies.htm

[24] http://www.oracle.com/technetwork/java/javase/downloads/javafxscenebuilder-info-2157684.html

4. TrendSpy

El objetivo de la aplicación creada, llamada TrendSpy, en este trabajo es recolectar el conjunto de trending topics o tendencias que produce Twitter a lo largo del día. Además de los trending topics la aplicación será capaz de obtener información relevante de Twitter para la clasificación de dichos trending topics en distintas categorías además del agrupamiento en estructura de comunidades de los trending topics, para ver la relación entre los mismos. La aplicación además incluirá funcionalidad para poder buscar tweets como si del propio buscador de Twitter se tratara. Incluirá también la generación y visualización de distintas gráficas que ayudarán a interpretar mejor la información que produce la aplicación. Para la instalación de la aplicación desarrollada habrá que seguir los pasos descritos en el Anexo C.

4.1. Arquitectura

En la figura 12 se puede apreciar la estructura de nuestra aplicación que se divide en siete módulos (todos ellos relacionados de alguna manera con la base de datos creada):

- Extractor de trending topics: se encarga de recoger los trending topics provenientes de Twitter de forma periódica.
- Extractor de links: se encarga de extraer la información necesaria de Twitter para poder agrupar y categorizar las tendencias que han surgido a lo largo del día.
- Clasificación por diccionario de palabras: se encarga de clasificar en categorías los trending topics recogidos.
- Agrupamiento por estructura de comunidades: se encarga de dividir en estructura de comunidades los trending topics y generar un grafo con la red asociada.
- Generador de gráficas: se ocupa de obtener los datos necesarios para realizar gráficas estadísticas para que el usuario pueda ver la evolución de la tendencia a lo largo de su tiempo de vida.
- Extractor de tweets populares: ofrecerá una lista de tweets considerados populares en base a unos criterios que el usuario podrá seleccionar.
- Interfaz de usuario: conecta todos estos módulos ofreciendo al usuario de manera visual la información extraída.

En la sección 2.3.6 se presentaron las distintas APIs que proporciona Twitter para extraer datos, debido al tipo de datos que queremos obtener de la plataforma, decidimos usar la API REST de Twitter, ya que no necesitamos recolectar información en tiempo real. API REST presenta una serie de limitaciones[25], las dos que más han afectado al desarrollo del proyecto han sido las siguientes:

[25] https://dev.twitter.com/rest/public/rate-limits

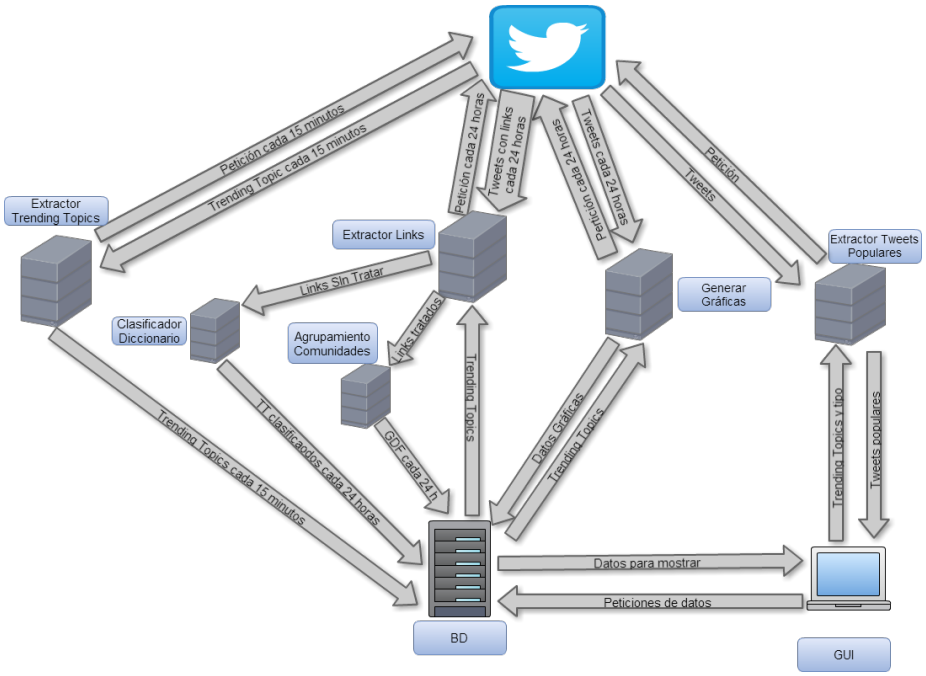


Figura 12: Arquitectura del sistema

- La API REST de Twitter permite un máximo de 15 peticiones cada 15 minutos para obtener los trending topics del momento, esto provoca que nuestra aplicación tenga que extraer de Twitter los trending topics cada 15 minutos, ya que de otra manera el servicio bloquearía y banearía por un tiempo limitado nuestra aplicación.

- En cuanto a las búsquedas de tweets, API REST sólo permite 180 peticiones cada 15 minutos. Esta limitación hay que tenerla en cuenta a la hora de recuperar tweets para almacenarlos en la base de datos, en concreto, la aplicación hará búsquedas de 100 peticiones para a continuación hacer una pausa por un tiempo de 15 minutos para luego continuar con la búsqueda y así evitar el bloqueo por parte de Twitter.

Para poder conectarnos a la API Rest de Twitter, en primer lugar habrá que registrar la aplicación en el apartado de Twitter developers[26] para obtener los tokens de acceso y los permisos para recibir datos de Twitter. Una vez obtenidos los tokens de acceso que nos permitirán autentificar nuestra aplicación en Twitter mediante el protocolo OAuth, nuestra aplicación, ayudándose de la librería Twitter4j será capaz de devolver una instancia de tipo Twitter, éste objeto es el que encapsula todos los métodos y funcionalidades de la librería Twitter4j y

[26] https://dev.twitter.com/

es el usado para realizar las distintas peticiones a Twitter. Mediante una clase estática llamada ConexionTwitter.java tenemos acceso autentificado y seguro a las distintas funcionalidades que ofrece la API REST de Twitter. En la figura 13 se puede ver un ejemplo del código que usa la aplicación para autentificarse y devolver la instancia Twitter requerida.

```
public static Twitter conexionApiRestForGraphic(){
      TwitterFactory factory = new TwitterFactory();
       AccessToken accessToken = UpdateUserState.loadAccessTokenForGraphics();
       Twitter twitter = factory.getInstance();
       twitter.setOAuthConsumer(tokenSecret,claveSecreta);
       twitter.setOAuthAccessToken(accessToken);
       return twitter;
}
```

Figura 13: Código para la autenticación de la aplicación en Twitter

4.2. Base de datos

La aplicación se sirve de una base de datos no relacional MongoDB para almacenar la información que extraeremos de Twitter. En concreto creamos un total de cuatro colecciones de datos para estructurar la información que vamos a almacenar. Las cuatro colecciones almacenan trending topics, la clasificación de las tendencias, gráficas y archivos gdfs, en la figura 14 se puede ver un esquema de lo que es la base de datos de la aplicación. A continuación se explican en más detalle cada una de estas colecciones además de la seguridad que conforma la base de datos.

4.2.1. Seguridad base de datos

La base de datos está en funcionamiento en un servidor de la universidad complutense proporcionado por el grupo NIL[27]. Para blindar la base de datos ante accesos indeseados se creó un usuario de tipo administrador para manejar los datos almacenados. Los detalles de la creación de dicho usuario y de la seguridad de la base de datos pueden verse en el Anexo A. El sistema cuenta con una clase estática llamada MongoDBHandler.java que se implementó para aislar todas las operaciones que se pueden hacer sobre la base de datos, entre ellas la autentificación. En el código mostrado en la figura 15 se puede ver el método de acceso mediante URI a una base de datos mongo, la base de datos que devuelve es el parámetro db, y es aquél sobre el que se realizarán las posteriores búsquedas o insercciones.

4.2.2. Colecciones

La base de datos cuenta con un total de cuatro colecciones: trendingtopic, gdf, gráficas y clasificación. Los detalles de cada una de ellas se explican a

[27] http://nil.fdi.ucm.es/

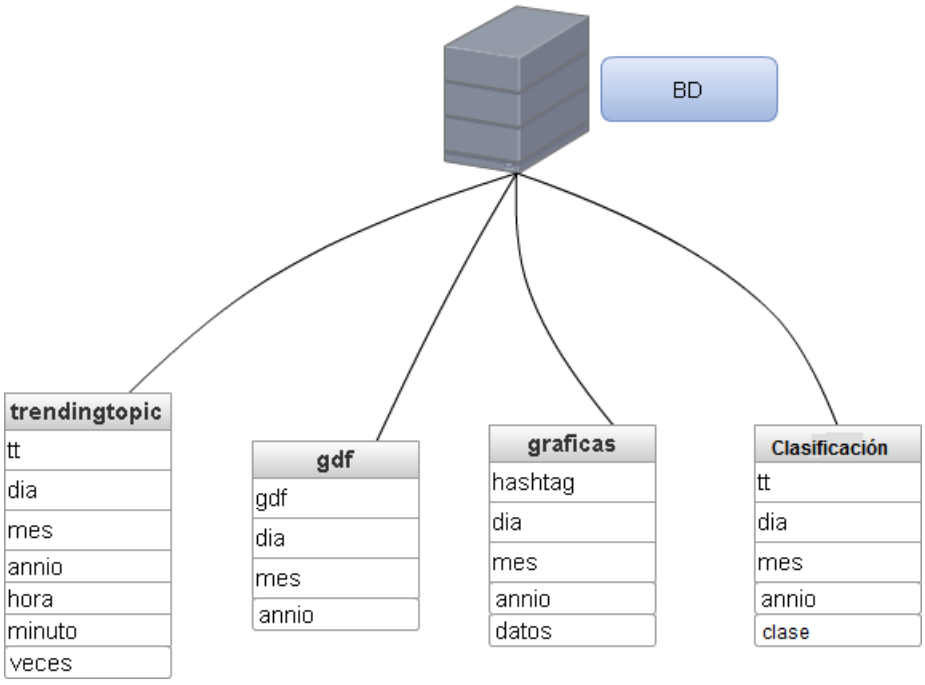


Figura 14: Estructura de la base de datos MongoDB y las colecciones creadas

```
try{
      String mongoDbUri = "mongodb://user:pass@hypatia.fdi.ucm.es:27017/bd";
      mongo = new MongoClient(new MongoClientURI(mongoDbUri));
      db=mongo.getDB("twitter");
}
```

Figura 15: Código para conectarse a la base de datos.

continuación.

- Trendingtopic: Se encarga de almacenar la estructura de los trending topics que extraemos de Twitter. Cada agregado JSON de la colección está compuesto por los siguientes atributos:

  - tt: Identifica el nombre del trending topic recogido de Twitter.
  - Fecha y hora en que se ha obtenido el trending topic.
  - lugar: Lugar geográfico al que pertenece el trending topic.
  - veces: Indica durante cuantos intervalos de tiempo (15 minutos por intervalo) ha permanecido como trending topic en Twitter.

  En la figura 16 se puede ver un ejemplo datos almacenados en la colección trendingtopic.

tt	dia	mes	annio	hora	minuto	lugar	veces
Piqué	8	6	2015	0	7	Spain	22
#DebateSV8	8	6	2015	0	7	Spain	26
#ChesterRuth	8	6	2015	0	7	Spain	18

Figura 16: Ejemplo de la colección trendingtopic

- Clasificación: Esta colección se encarga de almacenar el resultado de la clasificación en categorías de las tendencias. La estructura almacenada se compone de los siguientes atributos:

  - tt: Trending topic que contenían los tweets de los que se define su tendencia.
  - Fecha en que se produjo el trending topic.
  - clase: Categorías a las que pertenece el tt (política, cultura, ciencia/tecnología, entretenimiento, deportes y otros). Este parámetro se compone de una lista de categorías seguidas por el número de coincidencias de esa categoría para el trending topic.

  En la figura 17 se puede ver un ejemplo de los datos almacenados en clasificación.

tt	dia	mes	annio	clase
#Audiencias	8	6	2015	Deporte 1 Entretenimiento 76 Tecnologia 1 Cultura 2
Tamayazo	8	6	2015	Politica 9
Osasuna	8	6	2015	Deporte 17 Politica 1

Figura 17: Ejemplo de la colección clasificacion

- Gráficas: Esta colección se encarga de almacenar los datos necesarios para crear una gráfica que posteriormente representará el número de tweets que produce un trending topic por unidad de tiempo. Los atributos que definen su estructura son:

  - hashtag: Nombre del trending topic.
  - Fecha del trending topic.
  - datos: Lista de pares hora/número de tweets, es decir, almacenamos para cada hora en la que estuvo la tendencia en Twitter, el número de tweets publicados sobre esa tendencia por el conjunto de usuarios.

En la figura 18 se puede ver un ejemplo de los datos almacenados en la colección gráficas.

hashtag	dia	mes	annio	datos
#CuartoMilenio	8	6	2015	3-69 4-43 5-35 6-34 7-31 8-16 9-4 10-4 11-6 12-4 13-3 14-4 15-3 16-1 17-1 18-1 19-1 20-0 21-1 22-5 23-6
#DevateSV8	8	6	2015	3-100 4-80 5-70 6-10 7-8 8-15 9-25 10-9 11-6 12-5 13-12 14-6 15-2 16-8 17-1 18-1 19-1 20-0 21-1 22-5 23-0
Stannis	8	6	2015	3-1835 4-1982 5-653 6-534 7-327 8-256 9-215 10-189 11-204 12-286 13-354 14-454 15-526 16-589 17-714 18-689 19-452 20-361 21-362 22-412 23-451

Figura 18: Ejemplo de la colección graficas

- Gdf: Esta colección se ocupa de guardar el documento en formato gdf que crea la aplicación tras obtener el conjunto de enlaces asociados a los trending topics para posteriormente tratar dicho archivo mediante Gephi. Los atributos de esta colección son:

  - gdf: Contiene la información del archivo gdf.
  - Fecha de generación del archivo gdf.

  En la figura 19 se puede ver un ejemplo de los datos almacenados en la colección gdf. En este caso el atributo gdf contiene sólo una muestra pequeña del archivo gdf que se genera en la aplicación. Para ver un ejemplo de archivo .gdf completo ver en el Anexo D.

dia	mes	annio	gdf
8	6	2015	nodedef>name VARCHAR,label VARCHAR #FemBComú,#FemBComú #CampusSostenible,#CampusSostenible Roberto Carlos,RobertoCarlos Eurovisión Junior,EurovisiónJunior edgedef>node1 VARCHAR,node2 VARCHAR,weightINTEGER,labelVARCHAR,links VARCHAR Shumpert,#FemBComú,2,2, archivoparanormal.com cuerpoymente.es #FemBComú,#TuneaUnDeportista,2,2, archivoparanormal.com cuerpoymente.es #FemBComú,Tristan Thompson,2,2, cuerpoymente.es archivoparanormal.com

Figura 19: Ejemplo de la colección gdf

4.2.3. Recuperación de información

En la sección 4.2.1 se habló sobre la clase Java estática MongoDBHandler.java creada para aislar el procesamiento sobre la base de datos de la aplicación del resto de componentes, en concreto, se habló del método para autentificarse en la misma. En esta sección hablaremos sobre los distintos métodos que implementa esta clase Java y que nos sirven de punto de anclaje para el procesamiento de los datos almacenados en la base de datos. El conjunto de métodos a destacar de esta clase son:

- insertarTT(trends, lugar): Inserta un conjunto determinado de trending topics (trends) en la base de datos, concretamente en la colección «trendingtopic» identificando de manera adecuada el lugar de donde proceden el conjunto de tendencias a insertar, especificando además la fecha y hora a la que se ha recogido dicha tendencia que vienen especificados como atributos del parámetro trends.
- recuperarTT(dia,mes,annio): Recupera todos los trending topics de la fecha pasada por parámetro.
- guardaGDF(gdf,dia,mes,annio): Inserta en la la colección gdf un documento en formato gdf para un determinado dia, mes y año.
- recuperaGDF(dia,mes,annio):Recupera de la base de datos el archivo gdf que representa el grafo de trending topics de una fecha en concreto.
- recuperaGrafica(dia,mes,annio,tt): Recupera los datos asociados del número de tweets por hora que ha producido un determinado trending topic (tt) en una fecha concreta.

4.3. Modulo de extracción de Trending Topics

La principal funcionalidad de la aplicación consiste en la recolección de los distintos trending topics que produce Twitter a lo largo del día. La extracción de dichas tendencias se realiza en base a dos parámetros principales: la localización del mismo y el tiempo que se mantiene como tendencia en la aplicación.

4.3.1. Localización

Twitter nos ofrece la posibilidad de extraer el conjunto de temas que son tendencia en el momento en que se haga la petición de acuerdo a la localización de los mismos. En la sección 2.3.5 se explicó que Twitter puede filtrar dichas tendencias a nivel global, nacional o metropolitano. En nuestro caso, nos restringimos a nivel nacional y metropolitano, para así poder obtener tendencias y tweets en español que se clasificarán y se agruparán más adelante. Se extraen tendencias a nivel de España, y a nivel metropolitano, incluyendo un conjunto de las principales ciudades predefinidas por Twitter, entre las que se incluyen Madrid, Barcelona y Valencia.

Para poder obtener el conjunto de tendencias de una determinada localización, Twitter necesita que le pasemos un parámetro llamado WOEID[28] (Where On Earth IDentifier) que identifica mediante un código de 32 bits la localización exacta de un determinado lugar. En la figura 20 se pueden ver un subconjunto de los códigos WOEID de varias ciudades españolas.

[28] https://developer.yahoo.com/geo/geoplanet/guide/concepts.html

Figura 20: Conjunto de códigos WOEID devueltos

4.3.2. Tiempo como tendencia

La aplicación recolecta los trending topics de España y las ciudades españolas disponibles en intervalos de 15 minutos y los almacena en la base de datos. Si dicha tendencia ya se encontraba en ese día y en esa ciudad como tendencia, lo que se hace es incrementar un parámetro llamado «veces» que indica el número de peticiones en las que dicho trending topic ha aparecido como tendencia en Twitter. Éste parámetro posteriormente nos permitirá calcular el tiempo en que dicho trending topic ha permanecido como tendencia en la aplicación, lo que nos indicará la relevancia del mismo.

4.3.3. Extracción y almacenamiento de trending topics

Para recolectar todos los trending topics y almacenarlos en la base de datos el módulo recorre un HashMap de <Ciudades,WOEID> y llama a la función de Twitter4j correspondiente que devuelve los trending topics en ese momento para España y las ciudades españolas. Este procesamiento se hace cada 15 minutos debido a las limitaciones de la API REST de Twitter, ya que si calculásemos los trending topics en intervalos más pequeños de tiempo, la API nos banearía un tiempo limitado debido al exceso de peticiones. En la figura 21 se puede ver la estructura de la aplicación a la hora de extraer trending topics. En la figura 22 se puede ver el código del módulo extractor de trending topics. El código llama a la función getPlaceTrends(WOEID) a la que se le pasa como parámetro el código de la ciudad de la que queremos extraer los trending topics, a continuación mediante la clase mongoDBHandler hacemos una llamada al método insertarTT que insertar el conjunto de trending topics, en la colección trendingtopic de la base de datos. Este módulo estará ejecutándose de manera permanente en el servidor para que pueda ir recolectando a lo largo de todo el día las tendencias de Twitter.

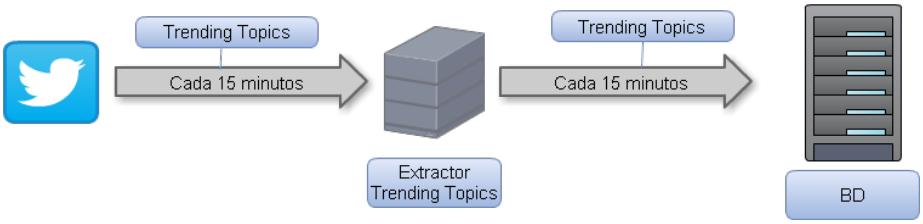


Figura 21: Esquema de la aplicación al extraer trending topics

```
while(true){
    Iterator<String> it = ciudades.keySet().iterator();
    while(it.hasNext()){
        String key = (String) it.next();
        int idTrendLocation =(int) ciudades.get(key) ;
        Trends trends = twitter.getPlaceTrends(idTrendLocation);
        mongoDBHandler.insertarTT(trends,trends.getLocation().getName());

    }
          TimeUnit.MINUTES.sleep(15);
}
```

Figura 22: Código para la extracción de trending topics.

4.4. Módulo de extracción de links

Este módulo se encarga de la extracción de información relevante para la posterior clasificación y agrupamiento de trending topics, enlaces a contenido externo de los tweets asociados a tendencias.

Debido a la limitación de caracteres que impone Twitter de la que ya se habló, la mayoría de los tweets que los usuarios consideran más relevantes, contienen en su mayoría uno o varios hashtags o palabras clave que identifican el tema del que se habla, y uno o varios enlaces a contenido externo en el que se explica de manera más amplia el contenido del tema referenciado por el tweet, normalmente este enlace suele ser a algún blog o periódico digital. En la figura 23, se puede apreciar un ejemplo del tipo de tweet que se considera como relevante ya que contiene un hashtag, en este caso «#EPDesayunoRajoy» y un enlace a la noticia ampliada, en este caso incluso Twitter describe este tweet como noticia destacada teniendo un total de 5 retweets y 5 favoritos a los 10 minutos de su publicación[29].

Twitter, con sus API, nos ofrece la posibilidad de realizar búsquedas muy específicas incluyendo algunos parámetros predefinidos en la consulta de búsqueda, en la página web de Twitter[30] se puede ver el conjunto de parámetros que se pueden añadir a la búsqueda para que Twitter únicamente devuelva los

[29] http://www.europapress.es/economia/macroeconomia-00338/noticia-rajoy-dice-bajara-impuestos-cuando-haya-mas-actividad-recaudacion-20150427100559.html

[30] https://dev.twitter.com/rest/public/rate-limits

Figura 23: Ejemplo de tweet relevante

tweets que pasen dicho filtro. En nuestro proyecto lo que haremos será hacer una búsqueda de aquellos tweets que contengan un hashtag o palabras claves determinadas, estén escritos en español y al menos contengan un enlace y excluyendo aquellos tweets que sean retweets para evitar obtener un conjunto más amplio de tweets. Por lo tanto la petición de tweets para el tweet de la imagen 23 sería así: #EPDesayunoRajoy filter:links lang:es exclude:retweets. Este tipo de consultas la realiza el módulo para todas las tendencias almacenadas durante el día en la base de datos, obteniendo un conjunto de enlaces que serán almacenados también en la base de datos de nuestra aplicación. Dichos enlaces, asociados a tendencias, nos serán muy útiles para el proceso de clasificación y categorización de los trending topics que se explicará en la sección 4.5.

El módulo de extracción de links hace peticiones a Twitter cada 24 horas y extrae los tweets relevantes para la clasificación y el agrupamiento, es decir, tweets con links y que contengan en su cuerpo las tendencias registradas durante el día por el módulo de extracción de Trending Topics. Una vez extraídos esos tweets, se tratan para extraer sólo los enlaces. Tras extraer éstos enlaces, se resuelve la dirección original de los mismos, ya que éstos se recogen de la aplicación con la URL minimizada. Una vez extraídos esos enlaces, son tratados por los módulos de clasificación por diccionario de palabras y por el módulo de agrupamiento de tendencias por comunidades, que almacenan el resultado de

sus cálculos en la base de datos.

4.4.1. Extracción de tweets con links

Este módulo se sirve de los datos almacenados por el módulo de extracción de trending topics para recuperar las tendencias que han surgido en Twitter a lo largo del día. Por lo tanto, este módulo se ejecutará en el servidor cada 24 horas, para que así en la base de datos estén todas las tendencias que han surgido ese día en Twitter. Para extraer los tweets con links seguimos este procesamiento:

1. Recuperaremos las tendencias de ese día mediante la clase MongoDBHandler con el método recuperarTT(dia,mes,annio), esta función devolverá la lista de todos los trending topics distintos[31] que han surgido en España y las ciudades españolas disponibles.

2. Hacemos uso de la clase implementada SearchWithLinks.java que lo que hará será buscar aquellos tweets con links en Twitter a partir de una fecha dada, que será el día en que se generó la tendencia. En la figura 24 se puede ver el constructor de la clase SearchWithLinks.java. El constructor de la clase SearchWithLinks tiene un parámetro que especifica el límite de tweets, esto se hace así ya que por defecto la API de Twitter sólo nos permite realizar una búsqueda de hasta un máximo de 100 tweets, que corresponderán a los 100 últimos publicados. Para evitar esta limitación, a la hora de buscar tweets, lo que hace el módulo es realizar múltiples búsquedas de 100 tweets hasta llegar al límite deseado. En esta misma clase, tenemos el método busqueda() que realizará la busqueda en Twitter de los tweets que pasen el filtro de que tengan enlaces, estén en español y no sean retweets.

```
public SearchWithLinks(String txt,int limiteTweets,String date){
      this.date = date;
      this.query=new Query(txt  +" filter:links lang:es exclude:retweets");
      if(limiteTweets>100)
            this.query.setCount(100);
      else
            this.query.setCount(limiteTweets);

      this.limiteTweets=limiteTweets;
      if (!this.date.equals(""))
            this.query.setSince(date);
}
```

Figura 24: Código del constructor de la clase SearchWithLinks.java.

[31] Es habitual que varios lugares compartan tendencias, así evitamos duplicidades de trending topics

4.4.2. Extracción y procesamiento de links

Una vez tenemos el conjunto de tweets devueltos por la clase SearchWithLinks, lo que hace el módulo es obtener los links asociados a esos tweets. El procesamiento que se sigue es el siguiente:

1. La aplicación extrae los enlaces presentes en cada tweet y los procesa. Para extraer los enlaces de los tweets nos servimos de expresiones regulares y de las utilidades de reconocimiento de las mismas de la clase String de Java. Así en la clase creada ProcesarInformación.java se implementa un método llamado pullLinks que se puede ver en la figura 25 que devuelve los enlaces presentes en un texto.

```
private  ArrayList<String> pullLinks(String text) {
 ArrayList<String> links = new ArrayList<String>();
 String regex = "\\b(((ht|f)tp(s?)\\:\\/\\/|~\\/|\\/)|www.)" +
             "(\\w+:\\w+@)?(([-\\w]+\\.)+(com|org|net|gov" +
             "|mil|biz|info|mobi|name|aero|jobs|museum" +
             "|travel|[a-z]{2}))(:[\\d]{1,5})?" +
             "(((\\/([-\\w~!$+|.,=]|%[a-f\\d]{2})+)+|\\/)+|\\?|#)?" +
             "((\\?([-\\w~!$+|.,*:]|%[a-f\\d{2}])+=?" +
             "([-\\w~!$+|.,*:=]|%[a-f\\d]{2})*)" +
             "(&(?:[-\\w~!$+|.,*:]|%[a-f\\d{2}])+=?" +
             "([-\\w~!$+|.,*:=]|%[a-f\\d]{2})*)*)*" +
             "(#([-\\w~!$+|.,*:=]|%[a-f\\d]{2})*)?\\b";
Pattern p = Pattern.compile(regex);
Matcher m = p.matcher(text);
while(m.find()) {
    String urlStr = m.group();
    if (urlStr.startsWith("(") && urlStr.endsWith(")")){
     urlStr = urlStr.substring(1, urlStr.length() - 1);
     }
     if(!urlStr.equalsIgnoreCase("http://t.co"))
     links.add(urlStr);
}
return links;
}
```

Figura 25: Implementación del método pullLinks.

En la figura 25 se aprecia que la cadena «regex» contiene la expresión regular que sirve para identificar un amplio rango de tipos de URL en un texto. Posteriormente mediante las clases Pattern y Matcher de Java se va analizando la cadena de entrada hasta encontrar las URLs presentes.

2. En segundo lugar, el módulo procesa los enlaces, decodificando los mismos, ya que éstos, en Twitter vienen minimizados, por lo tanto la mayoría son de la forma: «t.co/QqYtUkgDKB»[32]. Para decodificar estos enlaces y obtener la URL real de los mismos, realizamos varias peticiones HTTP

[32]Esto es un servicio que ofrece Twitter a sus usuarios para que los enlaces en los tweets

a las direcciones devueltas hasta que obtenemos la URL original. Estas peticiones HTTP se realizan en la clase creada para tal propósito ProcesarInformación.java en el método getLinkFromHashtagCompleto(hashtag, limite, fecha). Este método se encarga de:

a) Buscar los tweets asociados con SearchWithLinks y su método busqueda, para a continuación obtener los links codificados de esos tweets.

b) Posteriormente, mediante la llamada del método FastDecode decodifica los links pasados por parámetro y devuelve la URL original asociada a los mismos. En la figura 26 se puede ver la implementación del método getLinkFromHashtagCompleto.

```
public  ArrayList<DataStorage> getLinksFromHashTagCompleto(Hashtags hashtag, int
limit,String date) {
     //Realizo la busqueda para un determinado hashtag.
     SearchWithLinks search1 = new SearchWithLinks(hashtag.toString(),40,date);
     ArrayList<String> misLinks = new ArrayList<String>();
     //obtengo todos los links.
     misLinks = getLinks( search1.busqueda());
     //decodifico los links
      ArrayList<String> salidaDecodificados= FastDecode(misLinks);
     return cuentaRepeticiones(salidaDecodificados);
}
```

Figura 26: Implementación del método getLinksFromHashtagCompleto.

Debido a la cantidad de peticiones HTTP que se tienen que realizar el método FastDecode que recibe el conjunto de links a decodificar, se ha paralelizado para que su ejecución fuera más rápida y eficaz, debido a que si se hiciera de manera secuencial, el tiempo de espera de respuesta de cada petición HTTP haría de la decodificación de tantos enlaces una tarea muy pesada. En esencia para paralelizar el cálculo creamos un ExecutorService[33] con un conjunto de 100 hilos que se van ejecutando de manera concurrente, ya que el resultado de las peticiones HTTP son independientes entre ellas.

Además, tras obtener los enlaces ya decodificados y sabiendo la URL original, descartamos la mayoría de aquellas páginas web que no aportarían nada al tratamiento posterior de clasificación por diccionario de palabras y de agrupamiento por comunidades, como son páginas de imágenes (instagram.com), vídeos (youtube.com) o enlaces al propio Twitter.

4.5. Clasificación por diccionario de palabras

Una vez recolectados los datos explicados en la secciones 4.3 y 4.4 la aplicación categoriza las tendencias usando la información que nos brindan los tweets

ocupen el menor número de caracteres posibles, así aunque un usuario publique un tweet pegando un enlace completo a una página, el tweet publicado tendrá internamente un enlace minimizado de la forma anteriormente descrita.

[33] http://docs.oracle.com/javase/7/docs/api/java/util/concurrent/ExecutorService.html

con links. En esencia lo que hace el módulo es clasificar las tendencias que han surgido a lo largo del día en Twitter a partir de la información de los enlaces externos contenidos en los tweets.

Las categorías que hemos definido para la clasificación de los trending topics por diccionario de palabras son: Política, Deportes, Ciencia/Tecnología, Cultura, Entretenimiento y Otros. Elegimos este conjunto de categorías ya que abarca de manera amplia el conjunto de temas sobre los que se suele hablar en Twitter.

Para asociar los trending topics recolectados a lo largo del día a sus respectivas categorías definimos para cada una de ellas un conjunto de palabras clave asociadas a cada clase, por ejemplo, en el caso de deportes, algunas palabras claves serían «fútbol» , «baloncesto» o «tenis». Una vez definido dicho diccionario de palabras, la aplicación busca en cada enlace recogido por el extractor de links, las palabras claves y en el momento en que encuentre una, le asigna a la tendencia dicha categoría. Esto se repite para cada enlace extraído asociado a la tendencia, por lo que puede darse el caso que varios enlaces asocien una misma categoría a una tendencia. Este número de coincidencias se va contabilizando, por lo que al final, almacenamos en la base datos para cada tendencia una lista de categorías y número de coincidencias como puede verse en la figura 17. Ésta técnica resulta útil ya que la mayoría de enlaces recogidos son referencias a periódicos digitales o blogs, y éstos, dividen sus noticias en secciones según de lo que traten, así por ejemplo un enlace recogido en la aplicación podría ser: http://www.marca.com/2015/04/27/baloncesto/seleccion/1430090468.html que contiene la palabra «baloncesto» que es la sección de noticias a la que pertenece dicha publicación en la página. De esta manera el clasificador ubicaría el trending topic asociado a este enlace en la categoría de deportes.

Para hacer la búsqueda de las palabras clave nos servimos del algoritmo de Rabin-Karp, ya comentado en la sección 2.4, con las palabras del diccionario como patrones del mismo. El diccionario de palabras para el clasificador puede verse en el Anexo B.

4.6. Agrupamiento de tendencias por comunidades

En este otro módulo, relacionaremos los trending topics entre sí mediante estructura de comunidades, es decir, que con las tendencias y los enlaces asociados a los mismos, crearemos un grafo en el que tras aplicar un algoritmo, éste quede dividido en comunidades.

La estructura de comunidades, propiedad de los grafos que se explicó en la sección 2.2.3, nos servirá para que, vía Gephi podamos hacer un estudio de las distintas comunidades identificadas en un grafo generado diariamente tal y como se explica a continuación.

En este caso se crea un grafo diariamente en el que los nodos representan los trending topics, y las aristas que unen dichos nodos se crean si estos comparten un enlace. Para ello, hacemos un procesamiento previo, y nos quedamos con el host o raíz de los enlaces así por ejemplo en el enlace http://www.marca.com/ ... /baloncesto/... el enlace tras su procesamiento quedaría: http://www.marca.com. Por lo tanto la red que forma la aplicación unirá por ejemplo los hastags #Rafa-

Nadal y #CristianoRonaldo, ya que ambos contendrán enlaces a www.marca.com. Además si comparten más de un enlace en común la arista incrementará su peso en el grafo. Este procesamiento se hace para todos los hashtags recolectados durante todo el día creando un archivo en formato .gdf que puede leer Gephi para crear la red.

Esto era una primera aproximación, pero dado que hay páginas web generalistas que hablan de diversos temas, como son elpais.com o elmundo.es, lo anteriormente comentado podría llevar a cierta pérdida de información relevante que estemos buscando. Para solucionar ésto, el módulo lo que hace es buscar en cada URL sin acortar el conjunto de palabras definidas en el diccionario del Anexo B. Si en la dirección aparecen varias palabras del diccionario, nos quedaremos con aquella categoría, cuyas palabras asociadas presente más apariciones en la cadena. De este modo la red generada contendrá nodos formados por URL seguidas por una palabra clave que será la categoría, por ejemplo, pasaríamos de tener aristas elmundo.es a aristas elmundo.es.deporte. Además, si el host de la web ya tiene información sobre alguna categoría presente en el diccionario como pueda ser deportes.elpais.com evitaríamos todo este procesamiento.

La aplicación usa la librería que conforma la herramienta Gephi para generar una red y aplica un algoritmo que divide en clases modulares los distintos nodos del grafo. La implementación y uso de estos métodos podrá verse en la clase creada GephiMethods.java.

Una vez ejecutado el algoritmo de clases modulares, sólo nos falta mostrarlo al usuario mediante un archivo pdf. Para realizar esto, en primer lugar, aplicamos un layout o distribución a los nodos del grafo de tal manera que se coloquen de una manera vistosa al usuario. En concreto usamos YifanHu[34] como algoritmo de layout.

Una vez creado el archivo que identifica a la red, el usuario de la aplicación podrá obtener una imagen de un grafo en el que los nodos están divididos en distintas comunidades cada una representada por un color distinto. Un ejemplo de éste grafo puede verse en la figura 27 donde se aprecian un conjunto de 5 comunidades, tres de ellas mayoritarias representadas con los colores rojo, violeta y azul. La comunidad roja mayoritariamente tiene nodos que representan tendencias relacionadas con videojuegos, ya que en el momento de generar esta gráfica se estaba celebrando la conferencia E3 de videojuegos. Por otro lado la comunidad azul tiene tendencias más relacionadas con temas políticos hablados ese día (16 de junio). Por último la comunidad violeta tiene más relación con programas de televisión retransmitidos ese día.

4.7. Módulo de obtención de tweets populares

Este módulo se encarga de obtener una lista de tweets populares respecto a un hashtag dado. La aplicación mide la popularidad de los tweets en base al número de favoritos, al número de retweets o al número de seguidores que tenga el autor del tweet.

[34] http://yifanhu.net/PUB/graph_draw_small.pdf

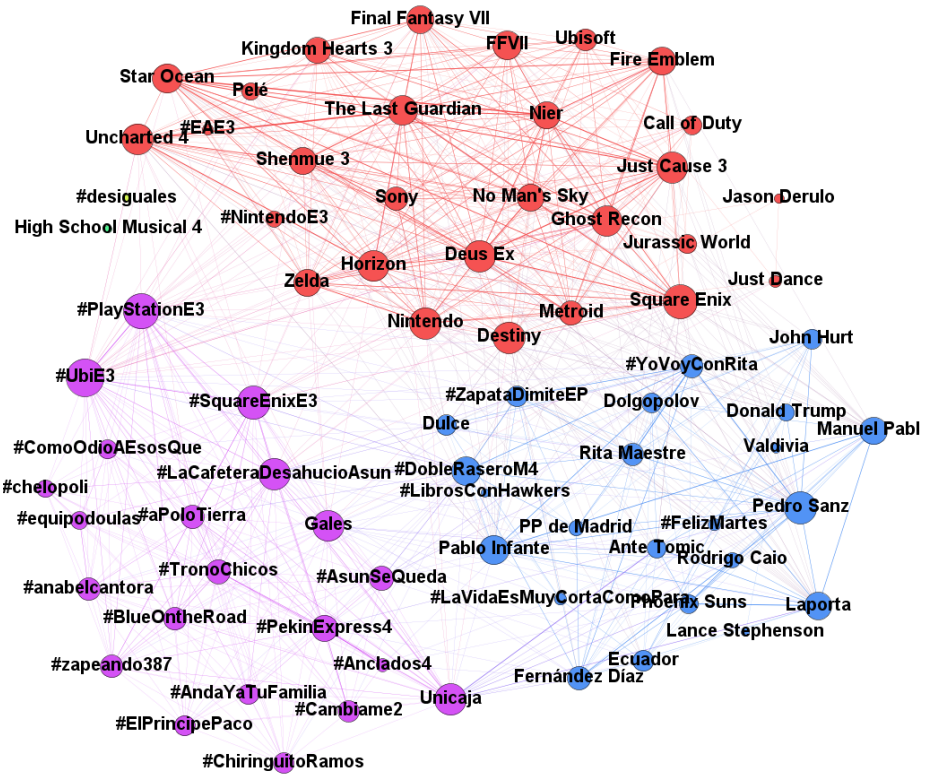


Figura 27: Ejemplo de archivo de grafo

Este módulo se ejecuta en el momento en que el usuario hace la petición, buscando a partir de un trending topic seleccionado un número amplio de tweets, en nuestro caso 1.000 tweets, que han ocurrido a partir de una determinada fecha. Una vez obtenido éste conjunto de tweets, el módulo los ordena en base a uno de los criterios seleccionados (favoritos, retweets o número de seguidores).

La estructura de la aplicación para este módulo se puede ver en la figura 28.

4.8. Módulo de generación de gráficas

Este módulo sirve para la generación de gráficas estadísticas que muestren el número de tweets publicados por hora para cada trending topic relevante. Un trending topic será relevante, si ha estado al menos 5 horas durante el día como tendencia. La aplicación tiene esta limitación debido a que generar gráficas para

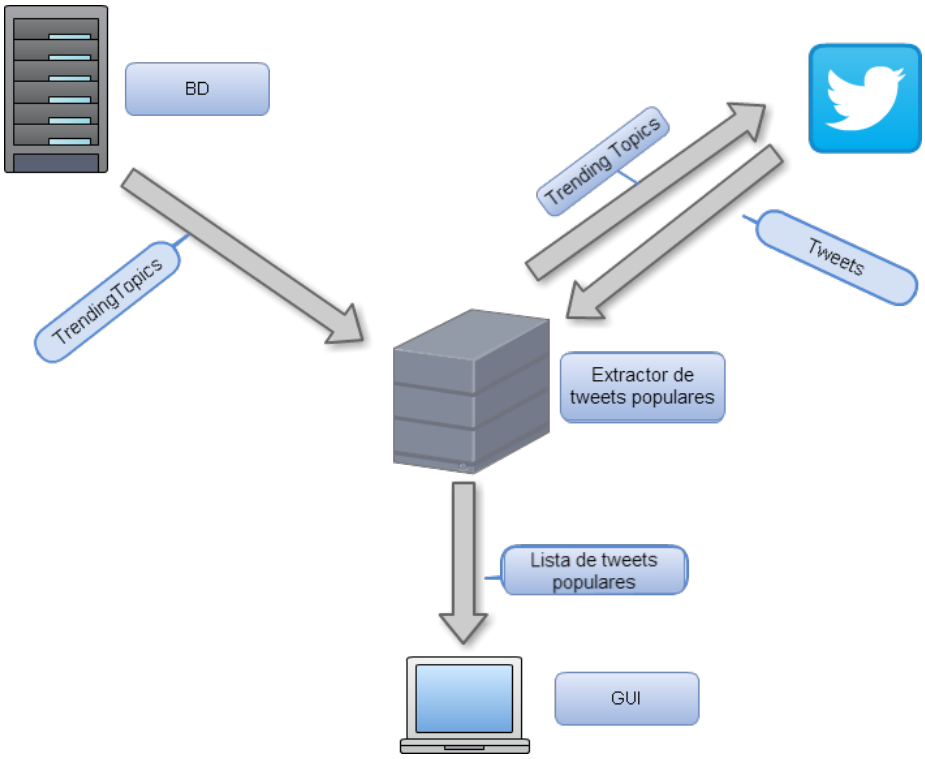


Figura 28: Arquitectura del módulo para extraer tweets populares

todos los trending topics que se generan al día, muchos de ellos con un tiempo de vida muy corto, sería una operación muy pesada para el servidor y tampoco aportaría información muy relevante al usuario.

Esta funcionalidad se ejecuta cada 24 horas en el servidor, para así poder tener disponibles el conjunto de tendencias surgidas durante el día. En esencia, el módulo lo que hace es buscar y contar el total de tweets que se han publicado por hora respecto a un determinado trending topic, generando entonces una lista de pares hora / número de tweets que nos servirá para la generación de las gráficas. En la figura 29, vemos un ejemplo del método constructor de la clase SearchByHour.java creada para que nos permita recolectar, con el método busqueda() el total de tweets por hora.

Una vez buscados y contabilizados los tweets almacenamos la información en la base de datos mediante el método GuardarGDF de la clase MongoDB-Handler.java para que después se pueda recuperar y generar una gráfica en la interfaz de usuario.

En la figura 30 se puede ver la arquitectura del módulo a la hora de extraer los datos necesarios para generar gráficas.

```
public SearchByHour(String txt,int limiteTweets, String since, String until, Date dia){
      this.since = since;
      this.until = until;
      this.dia = dia;
      this.txt=txt;
      this.query=new Query(txt  +" lang:es exclude:retweets");
      if(limiteTweets>100)
            this.query.setCount(100);
      else
            this.query.setCount(limiteTweets);
      this.limiteTweets=limiteTweets;
}
```

Figura 29: Implementación del método SearchByHour.

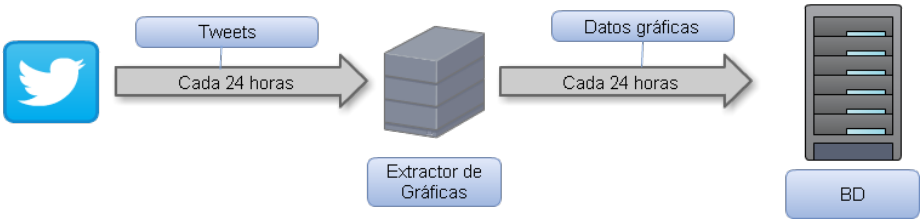


Figura 30: Arquitectura del sistema del módulo de generar gráficas

4.9. Interfaz de usuario

La aplicación posee una interfaz desarrollada en JavaFX. En la figura 31 se puede ver una imagen completa de la aplicación. En esencia la aplicación se compone de tres grandes apartados, una tabla de tendencias, un buscador de populares y conjunto de pestañas: vista web, gráficas generadas, clasificación por diccionario de palabras y relación de tendencias por estructura de comunidades.

En el apartado de tendencias (ver figura 32) podemos ver un conjunto de elementos con los que el usuario puede interactuar, básicamente son cuatro:

- Lugar: botón desplegable (combo-box) para que el usuario pueda elegir el origen de las tendencias que se vayan a buscar.
- Fecha: selector de fecha que permita al usuario seleccionar de qué dia quiere conocer las tendencias ocurridas en Twitter.
- Buscar TT: botón de búsqueda que permite al usuario realizar la búsqueda con los parámetros de fecha y lugar introducidos.
- Tabla de tendencias: Muestra los resultados de la búsqueda en dos columnas. La columna de la izquierda muestra el nombre del trending topic y la columna de la derecha muestra la duración del mismo como tendencia a lo largo de ese día.

En el apartado de tweets populares (ver figura 33), el usuario puede interactuar con varios elementos para recuperar aquellos tweets que sean relevantes para la

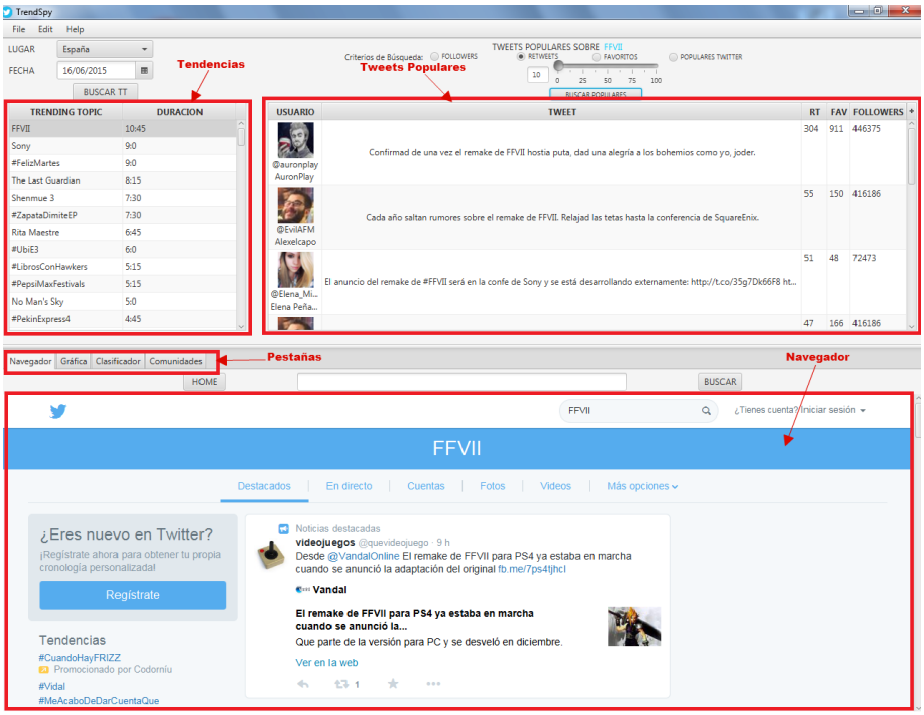


Figura 31: Ejemplo de la pantalla completa de tendencias y populares

tendencia seleccionada en la tabla anterior. Estos tweets pueden ser recuperados en base a tres factores: los más retweeteados, aquellos con mayor número de favoritos, o aquellos publicados por los usuarios con mayor número de seguidores. Este apartado implementa un cuadro de texto junto a un slide que permite especificar el número de tweets a recuperar. El botón de búsqueda carga en la tabla de abajo el conjunto de tweets. Esta tabla de tweets populares es interactiva, es decir, al seleccionar el usuario del tweet, en la vista web se cargará la información de ese usuario que ofrece Twitter, así por ejemplo si selecciona el tweet puede verlo ampliado en la vista web (ver figura 34) , permitiendo ver imágenes, vídeos o visitar enlaces asociados al mismo.

La aplicación también tiene un apartado para generar gráficas estadísticas sobre el número de tweets que se han generado por hora por cada trending topic cuyo tiempo de vida sea superior a 5 horas. En la figura 35 se puede apreciar la generación de dichas gráficas y cómo el usuario puede seleccionar qué gráficas de que trending topics mostrar, mediante los checkbox mostrados en el borde inferior, para así poder comparar la repercusión de un trending topic con otro.

En la siguiente pestaña, la aplicación tiene la opción de mostrar la clasificación de los trending topics del día en categorías, en la figura 36 se puede ver cómo se genera una tabla cuya primera columna identifica los trending topics y las demás las categorías a las que pertenecen, el porcentaje mostrado en cada co-

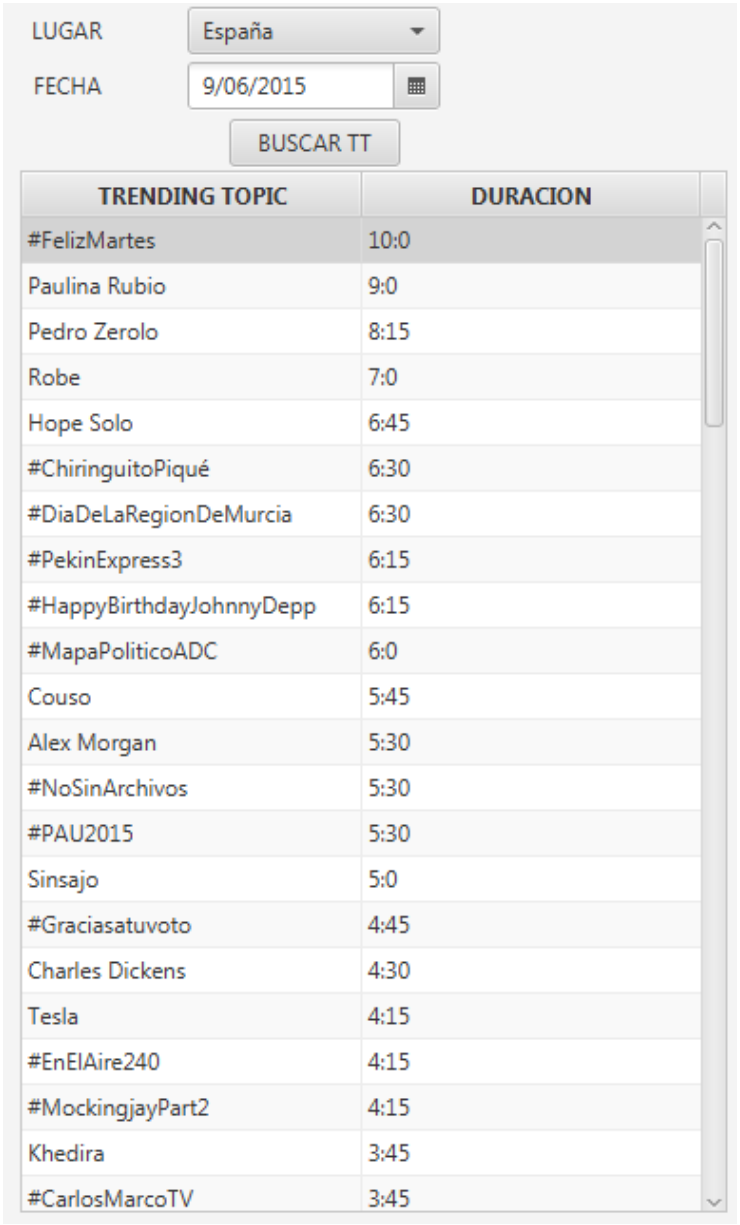


Figura 32: Tabla de tendencias de la aplicación.

lumna indica la proximidad del trending topic a esa categoría. Esta clasificación se hace mediante el clasificador por diccionario de palabras.

En la última pestaña se incluye una agrupación de los trending topics por estructura de comunidades. En esta pestaña se genera una tabla cuyas columnas

Figura 33: Apartado tweets populares de la aplicación.

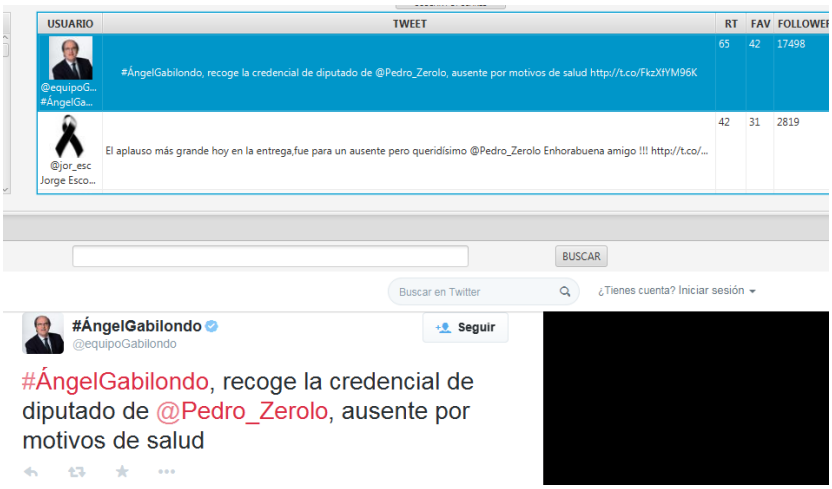


Figura 34: Asociación entre la vista web y la tabla de populares

identifican comunidades detectadas y las filas las tendencias asociadas a las mismas. Mediante esta tabla podremos ver la relación que tienen las tendencias entre sí. En la figura 38 se puede ver un ejemplo de la pestaña de comunidades. Se incluye un botón para generar el archivo pdf que contendrá la imagen generada

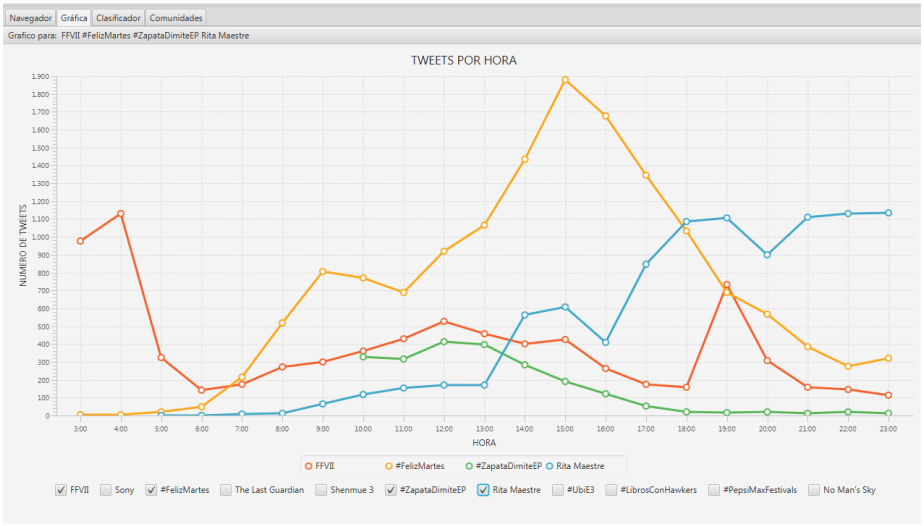


Figura 35: Ejemplo de generación de gráficas en la aplicación

Navegador | Gráfica | Clasificador | Comunidades

Buscar TT

Hashtag	Cultura	Tecnologia	Deportes	Politica	Entretenimiento	Otros
FFVII					100%	
Sony	14%	42%			42%	
#FelizMartes	30%	7%		15%	46%	
The Last Guardian	33%	33%			33%	
Shenmue 3	40%	10%			50%	
#ZapataDimiteEP		7%		92%		
Rita Maestre				100%		
#UbiE3	50%				50%	
#LibrosConHawkers						100%
No Man's Sky					100%	
#PekinExpress4					100%	
#AsunSeQueda					100%	
#TronoChicos					100%	
PP de Madrid				83%	16%	
#Cambiame2				14%	85%	
#EAE3		100%				
Horizon		20%		20%	60%	
#LaVidaEsMuyCortaComoPara			100%			
High School Musical 4	91%			8%		
Ante Tomic		4%	90%		4%	
Manuel Pablo		14%	85%			
Destiny					100%	
Final Fantasy VII					100%	
Ghost Recon	16%	33%			50%	
Square Enix	28%			28%	42%	
Nintendo		52%		11%	35%	

Figura 36: Ejemplo de clasificación de tendencias por diccionario de palabras

por Gephi a la hora de clasificar por estructura de comunidades, además se genera un archivo .gdf interpretable por Gephi para que cualquier usuario que quiera instalarse la aplicación pueda modificar y estudiar el grafo a su antojo. En la figura 37 se puede ver un ejemplo de grafo dividido en comunidades, en este caso, generado el día 14 de Junio de 2015, y está dividido principalmente en dos comunidades. Las tendencias de la comunidad de color rojo están relacionadas

con temática de deportes mientras que las tendencias de la comunidad azul tienen más que ver con el entretenimiento.

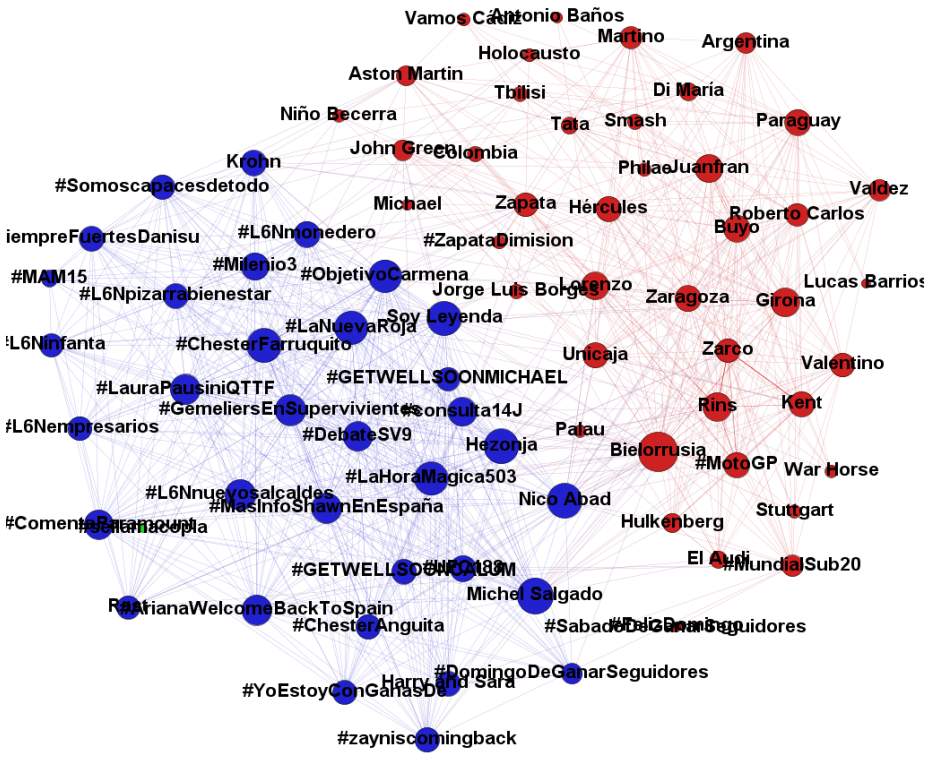
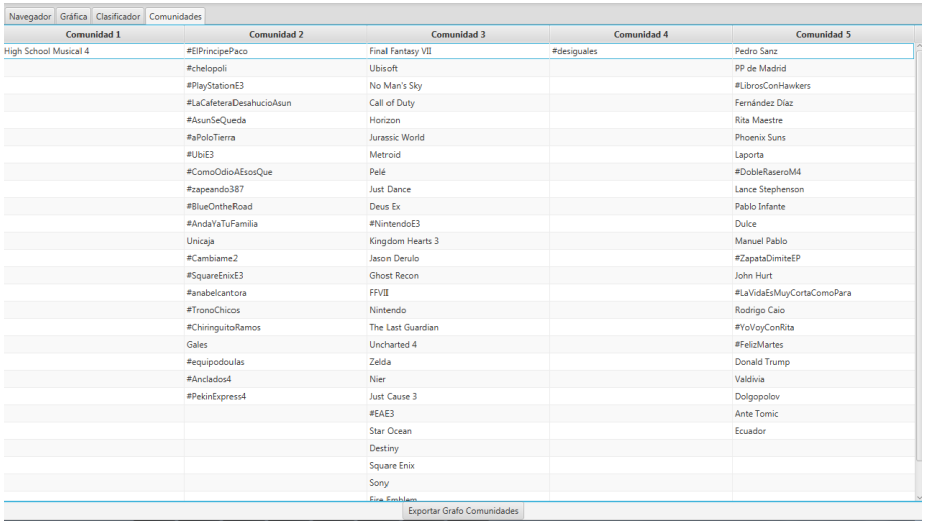


Figura 37: Grafo de tendencias agrupados en comunidades del día 14/6/2015.

En las figuras 39 y 40 se puede ver fragmentos del grafo generado de la figura 37 en el que se muestran los nodos, que son las tendencias, clasificados por colores que identifican la comunidad a la que pertenecen, en un caso, se puede intuir que una comunidad de nodos identifican tendencias relacionadas con los deportes, en ese día se jugó el ascenso a primera división de fútbol del Girona contra el Zaragoza, así como el partido de España contra Bielorrusia, además del mundial de motociclismo y un partido de la ACB, y otra con entretenimiento (programas de televisión), los nodos relacionan programas de televisión emitidos el domingo 14 de junio, entre ellos destacan «La Sexta Noche», películas en la cadena «Paramount Comedy» y el reality «Supervivientes». En ambas imágenes no aparecen las aristas para que se pueda ver mejor la composición de los nodos.



Navegador | Gráfica | Clasificador | Comunidades

Comunidad 1	Comunidad 2	Comunidad 3	Comunidad 4	Comunidad 5
High School Musical 4	#ElPrincipePaco	Final Fantasy VII	#desiguales	Pedro Sanz
	#chelopoli	Ubisoft		PP de Madrid
	#PlayStationE3	No Man's Sky		#LibrosConHawkers
	#LaCafeteraDesahucioAsun	Call of Duty		Fernández Díaz
	#AsunSeQueda	Horizon		Rita Maestre
	#aPoloTierra	Jurassic World		Phoenix Suns
	#UbiE3	Metroid		Laporta
	#ComoOdioAEsosQue	Pelé		#DobleRaseroM4
	#zapeando387	Just Dance		Lance Stephenson
	#BlueOntheRoad	Deus Ex		Pablo Infante
	#AndaYaTuFamilia	#NintendoE3		Dulce
	Unicaja	Kingdom Hearts 3		Manuel Pablo
	#Cambiame2	Jason Derulo		#ZapataDimiteEP
	#SquareEnixE3	Ghost Recon		John Hurt
	#anabelcantora	FFVII		#LaVidaEsMuyCortaComoPara
	#TronoChicos	Nintendo		Rodrigo Caio
	#ChiringuitoRamos	The Last Guardian		#YoVoyConRita
	Gales	Uncharted 4		#FelizMartes
	#equipodoulas	Zelda		Donald Trump
	#Anclados4	Nier		Valdivia
	#PekinExpress4	Just Cause 3		Dolgopolov
		#EAE3		Ante Tomic
		Star Ocean		Ecuador
		Destiny		
		Square Enix		
		Sony		
		Fire Emblem		

Exportar Grafo Comunidades

Figura 38: Ejemplo de clasificación de tendencias por estructura de comunidades.

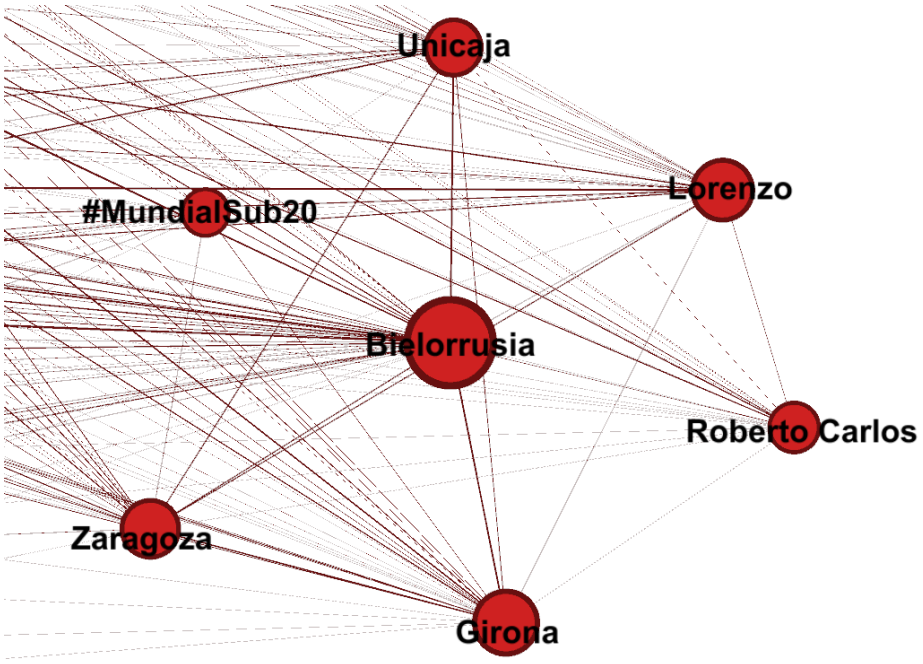


Figura 39: Fragmento de una de las comunidades, cuyos nodos están asociados por temática deportiva.

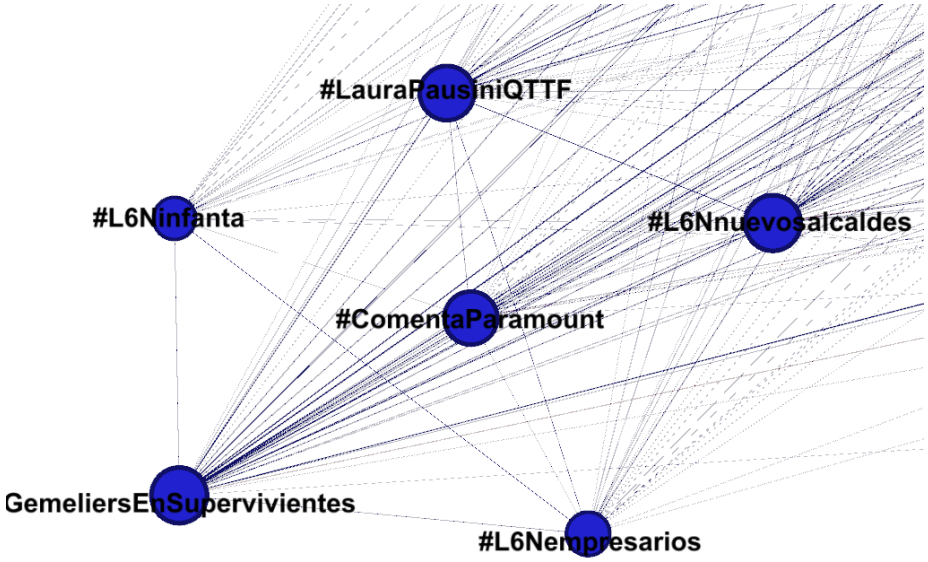


Figura 40: Fragmento de una de las comunidades, cuyos nodos representan en su mayoría programas de radio o televisión.

5. Evaluación

En esta sección se realiza la evaluación de algunos aspectos de la aplicación. En concreto evaluamos el método de clasificación por diccionario de palabras, el agrupamiento de tendencias por estructura de comunidades y el método para la obtención de los tweets populares.

5.1. Evaluación del clasificador por diccionario de palabras

En esta sección se detalla la evaluación realizada sobre el clasificador por diccionario de palabras descrito en la sección 4.5. Se explicará cómo se ha realizado la evaluación, qué resultados se han obtenido y a qué conclusiones se han llegado.

5.1.1. Diseño

Tomaremos los 20 trending topics de mayor duración de un día determinado, en concreto del 8 de junio de 2015, y procederemos a realizar una clasificación manual de ellos. Esta clasificación se realizará por parte de uno de los autores de este trabajo y los clasificará en base a su conocimiento de la actualidad y después de analizar los tweets asociados a ese trending topic. Posteriormente compararemos esta clasificación manual, con la obtenida por medio del clasificador por diccionario de palabras.

5.1.2. Resultados

Las clasificación manual para cada trending topic seleccionado es la siguiente:

- **Stannis:** se refiere a un personaje de la serie Juego de Tronos y de la saga literaria Canción de hielo y fuego. Por ello estaría encuadrado en las categorías de Cultura y Entretenimiento.
- **#DiaMundialdelosOceanos:** este hashtag corresponde a una celebración ocurrida ese día. No se encuadra en ninguna de las categorías establecidas en la aplicación, por lo tanto debe clasificarse como Otros.
- **#FelizLunes:** este hashtag es una tendencia contenedor de temas de diversa índole, le corresponde la categoría de Otros.
- **#SusanaPresidenta:** se refiere a la proclamación de Susana Díaz como presidenta de la Junta de Andalucía. La clasificación correspondiente es, por tanto, Política.
- **El FMI:** se refiere al Fondo Monetario Internacional, la clasificación adecuada es Política.

- **#CuartoMilenio:** es un hashtag asociado a un programa de televisión, que trata especialmente sobre temas asociados al misterio y lo desconocido, contando con la participación de distintos expertos. La clasificación sería, por tanto Entretenimiento, pero como ese día se invitó a algunos escritores, la clasificación correcta será Entretenimiento y Cultura.
- **#DebateSV8:** es un hashtag asociado a un programa de televisión, del tipo reality show. Por ello la clasificación en la que se encuadra es Entretenimiento.
- **Piqué:** es un jugador del FC Barcelona, la clasificación es Deportes. Además ese día estaba de actualidad porque Piqué realizó diversas opiniones políticas. Por lo que la clasificación correcta es Deportes y Política.
- **Kate Moss:** es una modelo. La clasificación es Entretenimiento.
- **Lagarde:** se refiere a la directora gerente del Fondo Monetario Internacional. Por ello, la categorización adecuada es Política.
- **#elultimomono8:** es un hashtag asociado a un programa de variedades emitido en televisión. Por ello la clasificación es Entretenimiento.
- **#TronoChicas:** es un hashtag asociado a un reality show televisivo. Por tanto, se encuadra en la categoría de Entretenimiento.
- **The Martian:** esta tendencia se refiere a una nueva película de ciencia ficción así que su clasificación es Entretenimiento y Cultura.
- **#NerviónNoSeCierra:** reacción popular en contra del cierre del estadio del Sevilla FC. Su clasificación es Deporte.
- **#WWDC15:** es un evento mundial de desarrolladores, realizado por Apple. Su categoría es Tecnología.
- **#ChesterRuth:** es un hashtag asociado a un programa de entrevistas emitido en televisión. Luego, le corresponde la categoría de Entretenimiento.
- **Cleveland:** es una ciudad de Estados Unidos, que esta de actualidad debido a que el equipo de esa ciudad esta jugando la final de la NBA. Por tanto su categorización es Deportes.
- **Konoplyanka**: es un jugador de fútbol pretendido por varios clubes españoles. La categorización es Deportes.
- **Carvajal:** es un jugador de fútbol del Real Madrid. Por tanto su categorización debe ser Deportes.
- **Camp Nou:** es el estadio del FC Barcelona. Se categoriza como Deportes.

En la tabla de la figura 41 se pueden ver los resultados que devuelve la aplicación para las tendencias clasificadas del día 8/6/2015, así como, los resultados de la clasificación manual de dichas tendencias. Se destacan en negrita las categorías dadas por el clasificador por diccionario que coinciden con las categorías dadas manualmente.

Trending Topic	Categorías clasificador diccionario	Clasificación manual
Stannis	**Cultura** 42%, **Entretenimiento** 42%, Política 14%,	Entretenimiento/Cultura
#DiaMundialdelos Oceanos	Entretenimiento 71%, Tecnología 28%,	Otros
#FelizLunes	**Otros** 100%	Otros
#SusanaPresidenta	**Política** 81%, Tecnología 18%,	Política
El FMI	**Política** 50%, Tecnología 50%,	Política
#CuartoMilenio	**Entretenimiento** 80%, Cultura 20%,	Entretenimiento/Cultura
#DebateSV8	**Entretenimiento** 60%, Política 30%, Tecnología 10%,	Entretenimiento
Piqué	**Deportes** 73%, Tecnología 20%, Política 6%,	Deportes/Política
Kate Moss	**Entretenimiento** 60%, Cultura 20%, Política 20%,	Entretenimiento
Lagarde	**Política** 66%, Entretenimiento 33%	Política
#elultimomono8	**Entretenimiento** 71%, Tecnología 28%,	Entretenimiento
#TronoChicas	**Entretenimiento** 93%, Tecnología 6%,	Entretenimiento
The Martian	**Cultura** 57%,**Entretenimiento** 28%, Tecnología 14%	Entretenimiento/Cultura
#NervionNoSeCierra	**Deportes** 100%	Deportes
#WWDC15	**Tecnología** 54%, Cultura 31% Entretenimiento 15%	Tecnología
#ChesterRuth	**Entretenimiento** 87%, Tecnología 12%,	Entretenimiento
Cleveland	**Deportes** 77%,Cultura 15%, Entretenimiento 5%, Política 1%,	Deportes
Konoplyanka	**Deportes** 100%	Deportes
Carvajal	Entretenimiento 30%, Política 30%, **Deportes** 23%, Cultura 15%	Deportes
Camp Nou	**Deportes** 50%, Entretenimiento 37%, Tecnología 12%,	Deportes

Figura 41: Comparación de la clasificación manual y la clasificación por diccionario de palabras.

5.1.3. Conclusiones

Analizando los datos presentados en la tabla de la figura 41 tenemos que en un 20 % de los trending topics evaluados coinciden las categorías manuales con las categorías del clasificador, esto ocurre para los trending topics #**FelizLunes**, **Konoplyanka**, #**CuartoMilenio** y #**NervionNoSeCierra**.

En un 70 % de los casos la categoría asignada por el clasificador con un mayor porcentaje coincide con la categoría manual. Vamos a analizar uno a uno estos trending topics para comprobar qué ocurre con el resto de las categorías que son asignadas por el clasificador por diccionario de palabras:

- **Stannis.** Este personaje en la serie es un gobernante, de ahí su clasificación del 14 % en Política.
- #**SusanaPresidenta** es clasificado con un 18 % en Tecnología. Esta clasificación es incorrecta y se debe a que para dicha tendencia hay asociados tweets que contienen links, a páginas como: «periodistadigital» o «economiadigital» que el clasificador interpreta como de Tecnología, porque en el diccionario de palabras, la palabra digital está asociada a la categoría de tecnología.
- **El FMI** es clasificado con un 50 % en Tecnología. Este error, se debe al igual que el caso anterior a la errónea clasificación de las páginas «periodistadigital», o «economiadigital».
- #**DebateSV8** se clasifica con un 30 % erróneamente en Política debido a las votaciones que se realizan en dicho programa y que el clasificador identifica con política. El 10 % en Tecnología se debe a que algunos de los enlaces obtenidos de los tweets de este trending topic son a una página llamada «cuerpoymente», cuyo contenido es a distintas investigaciones sobre la alimentación.
- **Piqué** es clasificado erróneamente con un 20 % en Tecnología es debido a la páginas «periodistadigital» y «libertadigital», que el clasificador asocia con Tecnología.
- **Kate Moss** esta clasificada erróneamente con un 20 % en Cultura, esto se debe a links asociado a este trending topic del tipo www.campeche.com.mx/ ... / cultura / ... /. La clasificación incorrecta del 20 % en Política se debe a otro link, en este caso: www.confirmado.com.ve/politica/....
- **Lagarde** ha sido clasificado como Entretenimiento en un 33 %, esto se debe al link www.elboletin.com/.../telegrama/..., el cual incluye telegrama y la palabra tele están asociada a Entretenimiento en el diccionario.
- #**elultimomono8** tiene una clasificación del 28 % en Tecnología, esto se debe a que en los enlaces recuperados para este trending topic, hay alguno a la página «archivoparanormal.com/ciencia» y ciencia es una de las palabras asociadas a la categoría de Tecnología.
- #**TronoChicas** tiene una clasificación de un 6 % en Tecnología. Esto es debido a que aparece un enlace a la página «archivoparanormal.com/ciencia» y el clasificador lo considera en la categoría de Tecnología.

- **The Martian** tiene una categorización del 14 % en Tecnología. Esto se debe a que al ser esta película del género ciencia ficción, algunos links como www.cinecuatro.com/ ... /ciencia - ficcionincluyen la palabra ciencia, con el resultado de que el clasificador las incluya en la categoría de Tecnología.

- **#ChesterRuth** la clasificación del 12 % de esta tendencia en Tecnología se debe a que hay un enlace a la web «semanaldigital», que el clasificador interpreta como de Tecnología.

- **Cleveland** tiene una clasificación erronea del 15 % en Cultura debido a que unos links de la página de la ESPN, contiene la siguiente subcadena: fashion + cultura + arte + cine. La clasificación del 5 % en Entretenimiento se debe a la aparición de la palabra serie en algún link, esta palabra hace referencia a la serie de partidos de los que se compone la final de la NBA, pero en el diccionario de la aplicación la asocia a Entretenimiento, debido a las series de televisión. En cuanto al 1 % de Política se debe a que entre los links se encuentra un enlace a la página «cubadebate.cu» y la palabra debate en el diccionario esta asociada a la categoría de Política.

- **Camp Nou** se clasifica con un 37 % en Entretenimiento, esto es debido enlaces a «radioolot.com» y «btv.cat» y las palabras radio y tv están asociadas a la categoría de Entretenimiento. La clasificación del 12 % en Tecnología se debe a que entre los enlaces, hay uno a «naciodigital.cat» y digital esta asociado a Tecnología en el diccionario.

- **#WWDC15** la categorización del 31 % en Cultura, se debe a que una de las novedades presentadas en esta convención ha sido un servicio para escuchar música, llamado Apple music, debido a esto la palabra music, que esta asociada en el diccionario del categorizador con Cultura, aparece también en varios enlaces, provocando el error en la clasificación. El encuadre del 15 % en la categoría de Entretenimiento es provocado por la aparición entre los links de uno a la web «mediatelecom.com.mx», y la palabra tele, en el diccionario esta asociada a la categoría de Entretenimiento.

En un 5 % de los trending topics evaluados, una categoría minoritaria del clasificador por diccionario de palabras es la que se corresponde con la clasificación manual. Vamos a analizar estos trending topics más en detalle:

- **Carvajal** es clasificado mayoritariamente por el clasificador de diccionario en Entretenimiento y Política con un 30 % cada una. Esto es así porque entre los enlaces hay varios a la página «telegrafo.com», y tele, en el diccionario esta asociado a Entretenimiento, en cuanto a la clasificación en Política es porque hay una algunos links que hacen referencia a la ministra Carvajal de Ecuador. La categorización en Cultura con 15 % es debido a que entre los links recogidos para este trending topic aparece: www.llanera.com/musica/...y música esta recogida en el diccionario dentro de la categoría Cultura.

Finalmente sólo en uno de los trending topics evaluados (5 %), la clasificación automática no coincide en nada con la clasificación manual, es el trending topic **#DiaMuundialdelosOceanos**. Este trending topic debería haber sido clasificada en Otros y lo ha sido en Entretenimiento (71 %) y Tecnología (28 %).

En vista de los resultados se puede concluir que el categorizador por diccionario, tiene una tasa de acierto bastante alta, ya que en un 90 % de los casos el clasificador devuelve una categorización correcta.

5.2. Evaluación agrupamiento de tendencias por comunidades

En esta sección se detalla la evaluación del módulo de agrupamiento de tendencias por estructura de comunidades descrito en la sección 4.6. Se detallará el cómo se ha realizado la evaluación así como los resultados y las conclusiones obtenidas.

5.2.1. Diseño

Para evaluar este método de agrupamiento de tendencias se discutirá la relación entre los trending topics en distintas comunidades. Para ello se elegirá un día, que coincide con el elegido para realizar las pruebas del clasificador por diccionario(8/6/2015), y un miembro del grupo analizará si esta agrupación es correcta, conociendo él la temática a las que pertenecen dichos trending topics y si existe una relación, fuerte, entre ellos. Para conocer la temática y si existe una relación empleará su conocimiento de la actualidad y los tweets asociados a dicho trending topic.

5.2.2. Resultados

El programa establece 5 comunidades distintas para este día, las comunidades y las tendencias asociadas a cada uno son:

- **Comunidad 1**: #FelizLunes
- **Comunidad 2**: #LaHoraMagica497, Paco Bustos, #CitesTV3, Shumpert, #NegociacionesClaveARV, #ARVenezuela, #TronoChicas, #zapeando381, #LaCafeteraBuenosPactos, #MtvSoyGerman2015, Manuel Bustos, Klay Thompson, #elultimomono8, #BuenosDiasConAndaYa, #AhoraoNuncaEH, #LoMas40aPorElLunes, #Alliknow, Mozgov, #superliantes, #AndaLevantaCanalFiestaL8, #CuartoMilenio, #dianabel, #ChesterRuth, #malasbelenguas y Kerr.
- **Comunidad 3:** Dani Rovira, Stannis, #PekinExpress3, #Anclados3, Shireen, #DebateSV8, La UDEF yTen Walls.
- **Comunidad 4:** #LuzuEnDirecto.

- **Comunidad 5:** #TonyAwards, Danny Ings, Piqué, Uwe Boll, Verza, Iron Man, Delly, Stosic, Girona, Game 2, Kevin Roldan, The Martian, Asamblea de Madrid, Apple Music, Lagarde, Club Bilderberg, Cavaliers, LeBron, Camp Nou, Bustos, El PRI, Speights, #WWDC15, Kate Moss, Cleveland, #SusanaPresidenta, Curry, Carvajal, #NervionNoSeCierra, Konoplyanka, Alonso, El Capitan, El FMI, #HolaMangel, Cavs, #DiaMundialdelosOceanos.

En la primera comunidad se puede observar, que sólo engloba a un trending topic, que no pertenece a una categoría definida.

La segunda comunidad se compone, principalmente, de tendencias relativas al entretenimiento, con distintos programas de televisión o radio: **#BuenosDiasConAndaYa**, **#LoMas40aPorElLunes,** **#CuartoMilenio**, **#elultimono8,** **#zapeando381**, **#AhoraoNuncaEH**, **#AndaLevantaCanalFiestaL8**, **#Mtv SoyGerman2015**, **#ChesterRuth**, **#TronoChicas**, o personas que salen en ellos como **#dianabel** o **#malasbelenguas**, **#superlliantes**. Pero también incluye otros programas de televisión de corte político como **#NegociacionesClaveARV**, **#ARVenezuela**, **#LaCafeteraBuenosPactos**, o personas relacionadas con la política como **Manuel Bustos**. Pero en esta comunidad también aparecen tendencias de temática deportiva como **Klay Thompson, Mozgov, Kerr.**

La tercera comunidad engloba trending topics relacionados con el entretenimiento y la cultura, como **Dani Rovira**, actor y humorista, **Ten Walls**, productor musical, los programas de televisión **#PekinExpress3** y **#DebateSV8**, la serie **#Anclados3,** o los personajes de la serie de televisión Juego de tronos y la saga literaria Canción de hielo y fuego, **Stannis y Shireen**. Pero también aparece **La UDEF**, la Unidad Central de Delincuencia Económica y Fiscal, que no tiene relación aparente con los otros elementos de esta comunidad, ya que pertenece a la categoría de política.

La cuarta comunidad, al igual que la primera engloba a un trending topic, de una categoría indefinida.

La quinta comunidad, engloba principalmente trending topics de tres clases muy distintas, como son deportes: **Piqué, Carvajal**, **Curry**, **Konoplyanka**, **Alonso**, **Cleveland**, **Cavaliers**, **Cavs**, **Kevin Roldan**, **Verza**, **Girona**, **Camp Nou**, **Stosic**, **#NervionNoSeCierra**, **Delly**, **Danny Ings**, **Speights**; y política, **Club Bilderberg, Asamblea de Madrid**, **Lagarde**, **El FMI, El PRI**,**Bustos**, **#SusanaPresidenta**, cultura: **#TonyAwards**, **Iron Man** y alguno de tecnología como: **#WWDC15**, **El Capitan**, **Apple Music** .

5.2.3. Conclusiones

El el agrupador de tendencias por comunidades, relaciona los distintos trending topics, de forma que suelen tener una relación bastante alta, si bien es cierto que a veces, si una tendencia esta relacionada con dos categorías distintas, puede agrupar tendencias de esas dos categorías distintas en una misma comunidad, quedando esta compuesta por dos subcomunidades.

Como se observa la segunda y la tercera comunidad se podrían unir, ya que tienen contenido parecido, de entretenimiento. También la quinta comunidad se podría dividir en otras dos comunidades, una de deportes y otra de política.

5.3. Evaluación de los tweets populares

En esta sección detallaremos cómo se ha llevado a cabo la evaluación del buscador de tweets populares descrito en la sección 4.7. Se describe tanto el diseño de la evaluación como los resultados y las conclusiones obtenidas.

5.3.1. Diseño

La aplicación permite obtener 100 tweets populares para cada método, pero para realizar la evaluación de los tweets populares devueltos por la aplicación, se procederá a analizar los 10 tweets más populares para cada criterio de los 10 trending topic con más duración del día 16/6/2015. Primero comprobaremos si los tweets devueltos son relevantes y posteriormente compararemos los tweets populares según cada criterio.

5.3.2. Resultados

En la tabla de la figura 42 se recoge el porcentaje de tweets que tienen relevancia con el tema del trending topic al que están asociados. Que un tweet tenga relevancia con un tema quiere decir que su contenido se corresponde con el tema tratado en ese trending topic. Por ejemplo, en el trending topic FFVII, cuyo tema es la presentación de un remake de dicho juego en la conferencia E3, sería relevante el tweet «Final Fantasy VII Remake confirmado para PS4, grande FFVII», pero no sería relevante el tweet «Voy a jugar un rato al FFVII». Twitter no siempre devuelve un número fijo de tweets populares cuando en la tabla aparece un guión significa que Twitter no devolvió tweets para esa tendencia.

En la figura 43 se recoge el porcentaje de tweets, que se comparten entre los diferentes métodos.

5.3.3. Conclusiones

De los resultados de la tabla de la figura 42, se observa que los tweets populares que obtiene la aplicación son relevantes en su gran mayoría (mas del 90 %)

Twitter devuelve menos tweets populares que nuestra aplicación. Twitter suele devolver alrededor de diez tweets populares, pero en algunas ocasiones devuelve menos, o directamente no devuelve ninguno. Además, los tweets populares de Twitter sólo están disponibles hasta 48 horas después de que esa tendencia se creara.

También se puede inferir, de los resultados anteriores, que el número de tweets que se comparten entre los distintos criterios de búsqueda de la aplicación es bastante bajo, a excepción de entre el número de favoritos y número de retweets, que esta en un 60 %.

Trending Topics	Nº favoritos	Nº Retweets	Nº Seguidores	Populares Twitter
FFVII	80%	80%	90%	50% (2 Tweets)
Sony	80%	90%	70%	100% (10 Tweets)
No man's Sky	100%	100%	100%	-
The Last Guardian	100%	100%	100%	100% (1 Tweet)
Shenmue 3	100%	100%	100%	100% (1Tweet)
PP de Madrid	100%	90%	100%	80% (10 Tweets)
#UbiE3	90%	100%	100%	100% (10 Tweest)
Rita Maestre	70%	80%	100%	100% (10 Tweets)
#PekinExpress4	100%	90%	100%	100% (10 Tweets)
#AsunSeQueda	100%	100%	100%	-
Total	**92%**	**93%**	**96%**	**94,4%**

Figura 42: Tabla con los porcentajes de tweets relevantes para cada trending topic

	Nº Seguidores	Nº Retweets	Nº Favoritos	Populares Twitter
Nº Seguidores	100%	33%	37%	31,54%
Nº Retweets	33%	100%	60%	18.52%
Nº Favoritos	37%	60%	100%	22,22%
Populares Twitter	17%	10%	12%	100%

Figura 43: Tabla con los porcentajes de tweets compartidos entre métodos

Según se muestra, los tweets considerados populares por Twitter, aparecen con un porcentaje muy bajo, en el resto de criterios. Aunque el método con el que Twitter elige a los tweets populares no esta publicado, en vista a estos resultados se puede inferir, que una de las cosas que más tiene en cuenta Twitter, es el número de seguidores del usuario que lo publicó. Ya que en la tabla de la figura 43, se muestra que la repetición de tweets entre el criterio de número de seguidores y los populares de Twitter está por encima del 30 %.

Como conclusión final decir que no hay un método que destaque por encima de otros, ya que todos tienen una tasa de relevancia parecida. Debido ello la elección del mejor método queda a la elección del usuario y lo que busque.

6. Aportación individual al proyecto

En este capítulo se detalla el trabajo individual y concreto realizado por cada alumno en el presente proyecto, aunque cada integrante del grupo ha trabajado en algunos aspectos por separado y se ha encargado de distintas partes del mismo, la práctica totalidad del trabajo se ha realizado trabajando de manera conjunta.

6.1. Ángel Luis Ortiz Folgado

Lo primero que se hizo a la hora de comenzar el proyecto fue investigar sobre el tema que trataría el mismo. Ángel se encargó de investigar la parte del estado del arte, más concretamente sobre redes sociales, Twitter y minería de datos. Durante ese periodo recopiló distinta información, artículos y otros trabajos de fin de grado realizados en años precedentes.

Una vez fijados los objetivos del proyecto, y una vez decidido que se usaría el lenguaje Java, se centró en encontrar librerías que facilitaran la tarea de realizar una conexión con Twitter para, obtener la información que se necesitaba. Ángel encontró Scribe, una librería que permite emplear autenticación tipo OAuth, imprescindible para realizar peticiones a la API de Twitter, y Twitter4J, la cual permite realizar las peticiones a la API de Twitter de manera transparente, y sin elaborar peticiones Get/Post. Una vez encontradas las librerías se decidió realizar un sencillo programa de búsqueda de tweets con ambas para así decidir cual se emplearía en el proyecto. Ángel Luis se encargó de implementar un programa usando la librería Scribe y creando las peticiones Get a mano. Finalmente con ambas implementaciones, se decidió usar la librería Twitter4J, ya que, en opinión del grupo, simplificaba más el trabajo.

Inicialmente, se intentó aprovechar un trabajo de fin de grado, desarrollado el año anterior en la facultad: Itafy. Al principió se obtuvo su código, el cúal, en un principio se pensó que se ejecutaría como un proyecto Java normal, pero causaba multitud de errores, debido a falta de librerías. Se buscaron las librerías, pero una vez eliminados los errores, se siguió sin saber como ejecutar. Una vez descubierto que se necesitaba Play Framework, Ángel corrigió el modo en el que se realizaba la autenticación con Twitter, así como la conexión con la base de datos MongoDB, para lograr su funcionamiento.

Tras la decisión de abandonar el proyecto Itafy, se comenzó con la implementación del proyecto, Ángel desarrolló el extractor de Trending Topics, que permitió obtener los de España y las ciudades españolas que permitía Twitter. También diseñó la base de datos MongoDB, para almacenar la información obtenida, desplegándola en el servidor cedido para realizar este proyecto,creándo un script para arrancar y configurar la base de datos, sin necesidad de escribir comandos, así como los métodos que permitieron el almacenamiento y la recuperación de dicha información en la clase MongoDBHandler.

Junto a Esteban, encontraron la forma de dejar ejecutando en el servidor de manera permanente los distintos programas que habían realizado para este proyecto, y también la base de datos.

En ese momento, también se realizó el diseño de la GUI de nuestro programa en la que Ángel participó.

Más adelante, junto a su compañero Óscar realizó la adaptación e inclusión del proyecto que se había desarrollado para la asignatura de Análisis de Redes Sociales. Ángel desarrolló métodos que permitían guardar los links, que usaban en MongoDB, para así agilizar las consultas al no tener que consultar a Twitter cada vez. Y finalmente creó junto a Óscar un método extractor de links, que está corriendo en el servidor, y cada día a medianoche obtiene los links de las tendencias del día anterior.

Después de esto Ángel se centró en la implementación del categorizador de los Trending Topic, para ello, se pensó en usar la información de los links que contenían los tweets, de los hashtags. Buscó información sobre distintas formas de realizar búsqueda de patrones en cadenas, ampliando los métodos que ya conocía de la asignatura de Métodos Algorítmicos en Resolución de Problemas. Finalmente se decidió usar el algoritmo de Rabin-Karp, ya que era el que mejor eficiencia lograba, a la hora de buscar múltiples patrones en cadenas. A partir de una versión de búsqueda simple de dicho algoritmo, Ángel desarrolló la búsqueda múltiple, así como las distintas categorías principales, y los patrones asociados a ellas.

Más adelante, también desarrolló el método de crear el archivo gdf (archivo interpretable por Ghephi), y guardarlo y recuperarlo en MongoDB, para permitir generar el grafo de relaciones de los hashtags. También realizó las adaptaciones necesarias en la GUI, para permitir mostrar el archivo pdf generado por Ghephi.

Más adelante junto a su compañero Óscar, desarrollaron los métodos necesarios para almacenar las gráficas del número de tweets por hora de los principales Trending Topic del día. Además del botón y los métodos necesarios para la visualización de estas en la aplicación. También desarrollaron un método generador de gráficas que está corriendo en el servidor facilitado, y que todos los días a las 12 de la noche, crea las gráficas de los Trending Topics con una duración mayor o igual a cinco horas, en el día anterior.

Después de eso, junto con Óscar terminaron de crear las pestañas de clasificador y populares, el contenido enlazable, es decir, los links que contienen los tweets, así como búsquedas en Twitter de los hashtags o menciones que aparecen en dichos tweets.

Más adelante, con el cambio a la nueva interfaz, desarrollada con JavaFX, Ángel realizó algún cambio pequeño en la visualización de dicha GUI, como que el clasificador se muestre en la ventana inferior, en lugar de en una aparte.

Además también hizo los distintos diagramas que aparecen en la memoria, como el diagrama de la arquitectura del proyecto, los de los distintos módulos que lo componen, el esquema de la base de datos, el del pseudocódigo de los distintos algoritmos de búsqueda de patrones en cadenas o los ejemplos de las distintas colecciones que componen la base de datos.

Finalmente Ángel realizó la evaluación de los métodos de clasificación de la aplicación. Para el método del diccionario de palabras, comparó los resultados que producía para las tendencias de un día, con las categorías en las que las

clasificaría una persona, obteniendo resultados satisfactorios. Para el método de las comunidades analizó la composición de las distintas comunidades en las que se agrupaba a los trending topics, discutiendo si era una separación correcta. Como conclusión a este proceso escribió la parte de la memoria que explica dicho procedimiento y los resultados obtenidos en él.

Además, Ángel realizó tareas de revisión y corrección en la memoria, con el fin de encontrar y solventar distintas erratas presentes en ella.

6.2. Óscar Eduardo Pérez la Madrid

Al principio, se investigó sobre los servicios que ofrecían las API Rest y Streaming de Twitter. Una vez que se tuvo claro tanto los servicios ofrecidos por dichas API, además de las limitaciones de cada una de ellas, se eligió usar Java para la implementación del proyecto, y la librería Twitter4J para facilitar las conexiones y autentificaciones con Twitter. Lo primero que hizo fue obtener las credenciales y tokens necesarios para poder utilizar la API Rest de Twitter y así poder tener acceso a los datos disponibles de Twitter. Implementó varios métodos para la autentificación y para hacer consultas a la API REST de Twitter, y así obtener tweets que incluyesen hashtags y los links necesarios para el funcionamiento del clasificador de tendencias. Una vez detectadas estas entidades, tuvo que procesar los links obtenidos, dado que la API de Twitter devolvía links minimizados o acortados, y lo que interesaba era obtener el link completo subyacente al mismo, de modo que implementó un método para realizar las redirecciones necesarias mediante peticiones HTTP en Java a estos links, hasta obtener la URL real del enlace. Descartó aquellos enlaces con redirección a fotos, videos y contenidos que no aportaban la información que se buscaba, mediante un filtro de páginas web, luego procesó esos links para quedarnos solo con el dominio de la URL. El procesamiento de las URL tenía un alto tiempo de procesamiento, por lo que decidió implementar un algoritmo que pudiese ejecutar estas tareas paralelamente, reduciendo considerablemente el tiempo de procesamiento.

Una vez obtenida y procesada la información necesaria, implementó junto a Esteban, un grafo. En este grafo los trending topics representaban los nodos, y la relación entre dos nodos venia dada por el un link que compartían. Aplicando la medida de modularidad mediante el uso de la herramienta Gephi, se pudo visualizar el grafo por comunidades que trataban un tema determinado. Luego implementó el clasificador de trending topics por diccionario de palabras, para realizarlo, necesitábamos los links completos así que tuvo que re-implementar los métodos creados anteriormente, para guardar las URL completas y procesadas, en lugar de solo guardar los dominios de los links. De esta forma se conseguía no duplicar información dado que solo hacía falta guardar las URL completas, en la base de datos, dado que podíamos procesar luego esos links para obtener el dominio de la URL, y así crear el grafo.

Posteriormente implementó la obtención de los tweets más populares para un determinado trending topic teniendo en cuenta el número de followers del creador del tweet, el número de veces que ha sido favorito el tweet y el número

de retweets. Luego, implementó, las gráficas que muestran la evolución de un trending topic durante el día anterior, es decir, la cantidad de veces que apareció un hashtag o palabra importante por hora, solo obtenemos esta información de los trending topics con duración mayor a cinco horas, que es la cota inferior a partir de la cual se pensó que se obtendría información más relevante, dado que esto requería mucho tiempo de procesamiento se decidió guardar estos datos en la base de datos, para poder mostrar los resultados obtenidos rápidamente. Utilizando la clase mongoDBHandler, implementada por Ángel, logró almacenar los datos de las gráficas en la base de datos, además se creó de forma que se ejecutaba siempre a medianoche, utilizando otras credenciales y tokens para evitar superar el límite de consultas, dado que a esa misma hora como mencionó Ángel se ejecuta el módulo que extrae los links.

Se encargó de crear la tabla del clasificador para poder visualizar los trending topics clasificados de tal forma que si un hashtag/palabra importante tenía alguna coincidencia con alguna de las categorías que teníamos en cuenta, se marcaba con una "X", además las tendencias estaban ordenados de mayor a menor por el tiempo en que se mantuvieron como trending topics. Luego añadió un filtro para poder buscar el hashtag/palabra importante rápidamente, todo esto fue implementado con JavaFX.

Después, para poder mostrar con rapidez la clasificación de las tendencias se decidió guardar el resultado de la clasificación en la base de datos, así que se encargó de la implementación y de las modificaciones necesarias para poder realizar esa idea.

Modificó la forma en que dos nodos se relacionan en el grafo creado, es decir, antes se relacionaban dos nodos si compartían algún host de una URL, se observó que con esta implementación se perdía información valiosa que ayudaría a crear mejores comunidades en el grafo, así que se decidió agregar información al host cuando no tuviese suficiente información, es decir, si el host no contiene palabras clave que lo identifican dentro de las categorías antes mencionadas, se hace una búsqueda en la URL completa de palabras claves, utilizando el algoritmo Rabin Karp que modificó Ángel, luego se añadía al host la categoría que hubiera tenido más coincidencias. Si el host ya tenía información de a que categoría pertenece no se aplica lo anterior. Modificó la interfaz del programa en concreto de la tabla que mostraba la clasificación anteriormente creada porque se pensó que la mejor forma de visualizar los resultados de la clasificación sería ver un porcentaje de las coincidencias asociadas a cada categoría siempre y cuando haya tenido alguna coincidencia, para realizar esto modificó la forma en que se guardaban los resultados de la clasificación añadiendo el número de coincidencias por categoría, facilitando así el cálculo de dichos porcentajes. Creó otra tabla donde se podían visualizar todos los trending topics de un determinado día, y lugar, y a que comunidad pertenecían. Por último se modificó la implementación del grafo anteriormente mencionado, para que solo tuviese trending topics cuya duración sea mayor igual a 1 hora, para evitar que se formen comunidades de 1 solo elemento, por ejemplo si un hashtag / palabra importante ha sido trending topic durante quince minutos, no se podrán obtener los suficientes tweets con hashtag y link, necesarios para poder relacionarlo con los demás hashtag/palabra

importante. Además cabe destacar que con la colaboración de Ángel, estuvieron atentos al servidor revisando diariamente que los programas que se ejecutaban en el servidor Hypatia, funcionasen correctamente.

6.3. Esteban Vargas Rastrollo

La principal aportación al proyecto de Esteban ha sido la redacción casi al completo de la presente memoria, aunque también ayudó en momentos puntuales a la implementación de la aplicación y participó en la toma de decisiones sobre la implementación, aportando ideas y soluciones.

En un primer momento, los directores del proyecto recomendaron realizar la memoria del proyecto en una herramienta gráfica de edición de textos que tuviera LaTeX como lenguaje subyacente, entre los distintos editores recomendados, se seleccionó LyX por su sencillez de uso y su extensa documentación. Tras decidir que sería LyX la herramienta a usar para la edición de la memoria, se tuvieron que leer diversos tutoriales sobre la herramienta para aprender a usarla.

Esteban dedicó luego una gran parte del tiempo a intentar adaptar y actualizar el trabajo de fin de carrera de nuestros compañeros de facultad del año anterior (Itafy) que en un principio pensamos que nos iba a servir como base para la continuación y realización del presente proyecto. Tras intentar ejecutar el proyecto en Java como una aplicación normal aparecieron multitud de errores debido principalmente a la falta de librería y a errores en el código que nos prestaron nuestros compañeros. Resultó al final que el código formaba parte de una aplicación web, sin cuyo framework de diseño resultaba imposible arrancar la aplicación. Tras la descarga y posterior entendimiento de dicho framework, se consiguió arrancar la aplicación definitivamente de manera local, ya que esta estaba diseñada para ejecutarse en un servidor remoto. Tras comprobar que la práctica funcionaba, se intentó entender el código del mismo para poder desarrollar a partir del mismo nuestra trabajo. Resultó que lo que era la aplicación, estaba hueca, es decir, mucha de la funcionalidad que prometía llevar a cabo no estaba implementada, además de que el código era bastante confuso y estaba sin comentar. Por lo que se decidió desechar la idea de partir de dicho trabajo y empezar de cero nuestra aplicación.

A continuación, se realizó una búsqueda de información sobre el estado del arte del tema que queríamos tratar, en concreto, se realizó la búsqueda de distintos artículos de investigación que hablasen de las características de Twitter, así como artículos y trabajos relacionados que usasen Twitter para la categorización y análisis de textos. También se efectuó la búsqueda junto a Ángel de distintas herramientas y aplicaciones que usaran Twitter como base para la categorización y monitorización de la información que se genera en Twitter. Una vez buscada toda esta información sobre el estado del arte, se incorporó en la presente memoria.

Una vez se supo en concreto que es lo que queríamos hacer, Esteban redactó la introducción y las páginas iniciales de la memoria del proyecto. Tras tener un prototipo de la aplicación que queríamos implementar, se escribió el capítulo de tecnologías usadas, buscando información más concisa de las herramientas que

se usaba hasta ese momento.

También le aplicó a la base de datos la seguridad necesaria para que únicamente usuarios registrados con el rol de administrador pudieran acceder y hacer cambios en la misma. Por otro lado, también ayudó a la puesta a punto del servidor proporcionado para que los distintos módulos de la aplicación empezaran a realizar peticiones a Twitter y recogieran y almacenaran los datos provenientes de la aplicación.

Esteban también implementó ciertas funcionalidades de la aplicación. En concreto, codificó el conjunto de métodos necesarios para generar imágenes de grafos y aplicarles distintos algoritmos a los mismos, sobre todo la medida de modularidad para dividir el grafo en comunidades, mediante la librería que nos proporcionaba Gephi.

También creó, junto a Óscar, el agrupamiento de tendencias por estructura de comunidades que se usa en la aplicación.

Además, durante todo el proceso de realización del proyecto, se redactaron una serie de actas que recogían el resultado de las reuniones que manteníamos con nuestro directores periódicamente. También se subsanaron errores, y se introdujo contenido que los directores de proyecto corregían en las distintas entregas de la memoria que hacíamos en cada reunión.

Esteban implementó además un primer prototipo de interfaz usando Java Swing, en este punto se participó principalmente en hacer interactiva la aplicación, es decir, en poder interactuar con ella, y que esta efectuase llamadas a un navegador web remoto para poder visualizar en Twitter la información que presentábamos en nuestra aplicación.

Esteban trasladó el primer prototipo de la aplicación realizada en Swing a JavaFX, una moderna tecnología desconocida para nosotros que hubo que aprender y aplicar al proyecto, en concreto, se implementó la interfaz principal del proyecto en la que se muestra las tendencias y los tweets populares generados por dicha tendencia, además de la inclusión de una vista web que permitía al usuario interactuar mejor con la aplicación, haciendo que al hacer click sobre los distintos elementos presentados por la interfaz, esta cargara en esa vista web la página web correspondiente a la información con la que se interactuaba.

En este punto, Esteban redactó lo que sería el núcleo de la memoria, es decir, el capítulo 4 donde se detalla el trabajo realizado para la implementación de la aplicación descrita en este documento. Redactó tanto la descripción de la arquitectura de la aplicación, enumerando sus distintos módulos que componían la misma. Se redactó la descripción de la base de datos mongoDB utilizada para almacenar los datos de la aplicación, así como la explicación en detalle de los distintos módulos de la aplicación, es decir, el módulo de extracción de trending topics, el módulo de extracción de links, el módulo de clasificación y el de agrupamiento, la extracción de tweets populares, la generación de gráficas y la interfaz de usuario desarrollada respectivamente.

Por último, Esteban redactó el capítulo de conclusiones y trabajo futuro de la memoria, en el que se explican los resultados obtenidos tras la finalización del proyecto, es decir, si se han cumplido los objetivos del mismo. También incluyó los distintos gráficos presentes en la memoria, unos proporcionados por

su compañero Ángel y otros hechos por él mismo. Además terminó la memoria con la inclusión de los distintos apéndices presentes y la revisión de toda la memoria en su conjunto, así como la traducción al inglés del resumen y las conclusiones.

7. Conclusiones y trabajo futuro

En esta sección se explican las conclusiones a las que se han llegado tras realizar el trabajo, así como líneas de trabajo futuro para mejorar el mismo. Además se hace un resumen de los conocimientos aprendidos del Grado de Ingeniería Informática en su itinerario de computación y que han sido indispensables para la realización de este trabajo.

7.1. Conclusiones

Como usuarios noveles de Twitter que éramos al principio, uno de los problemas que detectamos en la aplicación era que la información que se generaba en la misma fluía de manera muy rápida y que esa información contenía un alto porcentaje de ruido, es decir, información no relevante que distorsiona el tema de conversación. Además, debido al grado de temporalidad de la información que se explicó en la sección 2.3.4, mucha de la información pasaba desapercibida para el usuario si éste no estaba conectado a Twitter durante el tiempo de vida de la tendencia. Por otro lado, debido a la limitación de caracteres de Twitter, muchas de las tendencias generadas en la misma e identificadas por un hashtag o un grupo reducido de palabras clave hacían que el usuario de un vistazo no lograra dilucidar el contenido del tema de actualidad del que se estaba hablando en esos momentos en Twitter.

El objetivo de este trabajo era crear una aplicación que diese solución a estos tres problemas que detectamos en Twitter y podemos concluir que se han logrado subsanar en mayor o menor medida: En cuanto a la temporalidad de la información, la aplicación soluciona éste problema manteniendo un registro localizado de las tendencias surgidas en Twitter a lo largo de los días. Así si el usuario no ha entrado en Twitter durante un período de tiempo, mediante nuestra aplicación podrá explorar las tendencias de las que no ha tenido constancia, explorando además en el navegador integrado los resultados que devuelve Twitter para ese trending topic. Respecto a la fluidez de la información, la aplicación es capaz de mostrar al usuario aquellos tweets más populares (con más favoritos, retweets o número de seguidores) de un trending topic. Ésta funcionalidad Twitter no la implementa y el usuario sólo es capaz de ver aquellos tweets publicados recientemente, resultando que la mayoría de las veces el usuario no sabe de que trata un tema o que han considerado los usuarios como más importante. En cuanto a la clasificación de tendencias, la aplicación clasifica las tendencias más importantes en base a un diccionario de palabras descrito en la sección 4.5. Además la aplicación agrupa las tendencias por estructura de comunidades como se describió en la sección 4.6, que permite relacionar tendencias entre si. Esto le sirve al usuario para contextualizar los trending topics, ya que las tendencias, al ser hashtags o conjunto de palabras reducidos muchas veces no permiten al usuario hacerse una idea sobre la categoría del tema que se está hablando o las relaciones que presentan las tendencias entre ellas.

7.2. Conocimientos aplicados de la carrera

La realización del presente proyecto no hubiera sido posible sin los conocimientos que se han aprendido a lo largo del Grado de Ingeniería Informática en su itinerario de Computación y en las diversas asignaturas cursadas en el mismo. Las asignaturas destacadas a continuación, son aquellas que hemos cursado y que más nos han ayudado para la realización de este trabajo de fin de grado:

- Tecnología de la Programación (TP). Programación en lenguaje Java. El proyecto está íntegramente escrito en Java.
- Ingeniería del Software (IS). Organización del proyecto y uso de repositorios.
- Programación Concurrente (PC). Técnicas de concurrencia en Java. En este proyecto usamos programación paralela y concurrente a la hora de resolver las direcciones de los links minimizados.
- Métodos Algorítmicos y Resolución de Problemas (MAR). Técnicas algorítmicas para resolver problemas. El algoritmo de Rabin-Karp es utilizado en el clasificador por diccionario de palabras.
- Desarrollo de Sistemas Interactivos (DSI). Diseño de interfaces amigables para el usuario y técnicas de evaluación de las mismas.
- Gestión de Información en la Web (GIW). Bases de datos no relacionales como MongoDB y competencias relacionados como los formatos JSON o el lenguaje php.
- Análisis de Redes Sociales (SOC). Construcción del clasificador por estructura de comunidades. Sin la misma no habríamos conocido Gephi, herramienta que nos ha sido muy útil para la realización del proyecto.

7.3. Líneas de trabajo futuro

Aunque la aplicación desarrollada cumple en mayor o menor medida con los objetivos propuestos de este trabajo, esta aún tiene mucho potencial a desarrollar para mejorar lo ya implementado y añadir funcionalidades. A continuación se hace un listado de aquellos aspectos que pensamos que se pueden mejorar de la aplicación así como el desarrollo de otras funcionalidades completamente nuevas y que pueden ser útiles en futuros trabajos:

- Utilización de técnicas de procesado de lenguaje natural para mejorar la clasificacion de trending topics.
- Añadir nuevos criterios de obtención de tweets populares, como por ejemplo el análisis de sentimiento, para decidir si un tweet tiene carácter positivo o negativo.
- Desarrollar la aplicación en formato de servicio web o de aplicación móvil.

- Aumentar el tamaño del diccionario de palabras o las categorías del mismo para obtener una clasificación más precisa.
- Realizar una evaluación de la aplicación desde el punto de vista del usuario, es decir, la usabilidad de la interfaz gráfica y la utilidad de la aplicación en general.
- Mejorar el apartado del agupamiento de tendencias por estructura de comunidades para que relacione las tendencias mejor, y se muestre mejor el grafo que generamos con Gephi.
- Generar nombres para cada comunidad del agrupamiento de tendencias por estructura de comunidades, para que dicho nombre esté asociado a una categoría o palabra clave que englobe a todos los nodos (tendencias) presentes en el grafo.

8. Conclusions and future work

In this section we explain the conclusions that have been reached after carrying out the work and future work to improve it. Furthermore a summary of learned knowledge in Computer Engineering Degree in Computer itinerary that where is presented necessary for carrying out this work.

8.1. Conclusions

As new users of Twitter, one of the problems we detected was that the information generated flowed very quickly and contained a high percentage of noise, ie, irrelevant information that distorted. In addition, due to the degree of timeliness of the information, as it was explained in section 2.3.4, much of the information went unnoticed for the user if it has not been connected to Twitter during the lifetime of the trend. Furthermore, because the character limit of Twitter, many of the trends do not elucidate the content at a glance.

The aim of this work was to create an application that would give solution to these problems that we detected on Twitter and we can conclude that they has been more or less overcome. Our application sales the problem of transitoriness by maintaining trends emerged in Twitter throughout the day. If the user has not entered into Twitter for a period of time, through our application he can explore the trends recorded in the application. Regarding the flow of information, the application is able to show the user the most popular tweets (with more favorites, retweets or number of followers) of a trending topic. This functionality is not implemented on Twitter and the user is only able to see those tweets recently published, with the result that most of the time the user does not know what an issue is about or what users have considered as more important. As for the classification of trends, the application classifies trending topics based on two methods, one more statistic, the dictionary of words of the 4.5 section, and another more visual that relates trends together by community structure as explained in section 4.6.

8.2. Degree applied knowledge

The realization of this project has not been possible without the knowledge learned throughout the degree in Computer Engineering and Computer itinerary. Highlighted below are those subjects which have been taken and that have been most useful to carry out this work:

- Tecnología de la Programación (TP).Java programming language.
- Ingeniería del Software (IS). Project organization and use of repositories.
- Programación Concurrente (PC). Java concurrency techniques. In this project we use parallel and concurrent programming to resolve the addresses of the links minimized.

- Métodos Algorítmicos y Resolución de Problemas (MAR). Algorithmic problem solving techniques. The Rabin-Karp algorithm is used in the classifier.
- Desarrollo de Sistemas Interactivos (DSI). Design user friendly applications and assessment techniques.
- Gestión de Información en la Web (GIW). Non-relational data bases as MongoDB and related skills such as JSON formats or php language.
- Análisis de Redes Sociales (SOC). Graph theory. Classifier of community structure. Without it we would not have known Gephi tool that has been very useful for the project.

8.3. Future work

Although the developed application complies more or less the objectives of this work, it still has much potential to improve it. Here is a list of those aspects that we think can be improved:

- Using NLP techniques to improve trending topics classification..
- Add new criteria for obtaining popular tweets, such as sentiment analysis to decide if a tweet is positive or negative.
- Transform the application to web applicattion or mobile application.
- Increase the size of the dictionary at words or categories to obtain a more accurate classification.
- Conduct an evaluation of the application from the point of view of the user, ie, the GUI usability and usefulness of the application in general.
- Improve the section of the classification by structure of communities to classify better and improve with Gephi generated graph.
- Build community names for each classification structure of communities, so the name can be associated to the category that includes all nodes (trends) present in the graph.

9. Anexo A: Seguridad base de datos

El archivo mongodb.conf que se presenta a continuación ha sido el utilizado para crear el servidor de la base de datos mongo.

```
fork = true
port = 27017
quiet = true
dbpath = /home/ssii1415/datos/db
logpath = /home/ssii1415/mongolog
logappend = true
journal = true
auth = true
```

Este archivo se pasa como parámetro a la hora de arrancar el servidor con la instrucción:

```
mongod --config /home/ssii1415/mongodb.conf
```

Los parámetros más importantes son el dbpath, que especificará el lugar donde se crea la base de datos y el parámetro auth = true, que especifica que para poder operar sobre la base de datos, el usuario tiene que estar autenticado.

10. Anexo B: Diccionario de palabras para el clasificador

A continuación se muestra la clase Categorias.java la cual contiene un enumerado de conjuntos de palabras y sus categorías asociadas.

```
public enum Categorias {
    deportes("deportes", "Deporte"),
    futbol("futbol","Deporte"),
    tenis("tenis","Deporte"),
    baloncesto("baloncesto","Deporte"),
    liga("liga","Deporte"),
    sport("sport","Deporte"),
    diariogol("diariogol","Deporte"),
    formula1("formula1","Deporte"),
    ciclismo("ciclismo","Deporte"),
    natacion("natacion","Deporte"),
    atletismo("atletismo","Deporte"),
    reality("reality","Entretenimiento"),
    tele("tele","Entretenimiento"),
    tv("tv","Entretenimiento"),
    videojuegos("videojuegos","Entretenimiento"),
    audiencia("audiencia","Entretenimiento"),
    emisora("emisora","Entretenimiento"),
    programa("programa","Entretenimiento"),
    serie("serie","Entretenimiento"),
    cadena("cadena","Entretenimiento"),
    telespectador("telespectador","Entretenimiento"),
    tve("tve","Entretenimiento"),
    telecinco("telecinco","Entretenimiento"),
    antena3("antena3","Entretenimiento"),
    cope("cope","Entretenimiento"),
    rne("rne","Entretenimiento"),
    radio("radio","Entretenimiento"),
    psoe("psoe","Politica"),
    podemos("podemos","Politica"),
    upyd("upyd","Politica"),
    rajoy("rajoy","Politica"),
    politic("politic","Politica"),
    ministro("ministro","Politica"),
    comunista("comunista","Politica"),
    debate("debate","Politica"),
    pacto("pacto","Politica"),
    eleccion("eleccion","Politica"),
    ejecutivo("ejecutivo","Politica"),
    nacionalista("nacionalista","Politica"),
```

```
parlamento("parlamento","Politica"),
electoral("electoral","Politica"),
diputad("diputad","Politica"),
socialista("socialista","Politica"),
oposicion("oposicion","Politica"),
tecnologi("tecnologi","Tecnologia"),
ciencia("ciencia","Tecnologia"),
cientific("cientific","Tecnologia"),
csic("csic","Tecnologia"),
digital("digital","Tecnologia"),
electronic("electronic","Tecnologia"),
investigacion("investigacion","Tecnologia"),
robot("robot","Tecnologia"),
ordenador("ordenador","Tecnologia"),
teoria("teoria","Tecnologia"),
sonda("sonda","Tecnologia"),
nasa("nasa","Tecnologia"),
laboratorio("laboratorio","Tecnologia"),
innovacion("innovacion","Tecnologia"),
cultura("cultura","Cultura"),
manga ("manga","Cultura"),
comic("comic","Cultura"),
literatura("literatura","Cultura"),
cine("cine", "Cultura"),
musica("musica", "Cultura"),
cantante("cantante", "Cultura"),
music("music", "Cultura"),
actor("actor", "Cultura"),
actriz("actriz","Cultura"),
espectaculo("espectaculo", "Cultura"),
novela("novela", "Cultura"),
poesia("poesia", "Cultura"),
poet("poet", "Cultura"),
teatro("teatro", "Cultura"),
exposicion("exposicion", "Cultura"),
danza("danza", "Cultura"),
artista("artista", "Cultura"),
festival("festival", "Cultura"),
concierto("concierto", "Cultura"),
actuacion("actuacion", "Cultura"),
escritor("escritor","Cultura");

private String categoria;
private String subcategoria;

Categorias(String subcat ,String cat) {
```

```
        this.categoria = cat;
        this.subcategoria=subcat;
    }
    public String getCategoria(){
        return this.categoria;
    }
    public String getSubCategoria(){
        return this.subcategoria;
    }
    public static ArrayList<String> getSubCategories() {
        Categorias[] cat = Categorias.values();
        ArrayList<String> subCategorias = new ArrayList<
            String>();
        for(int i = 0; i < cat.length; i++)
            subCategorias.add(cat[i].getSubCategoria().
                toString());
        return subCategorias;
    }
}
```

11. Anexo C: Manual de instalación

En esta sección se detallan los pasos que hay que seguir para la correcta instalación y funcionamiento de la aplicación desarrollada. La aplicación consta de dos partes bien diferenciadas, una de ellas un conjunto de módulos que se ejecutarán en un servidor, y por otro lado una aplicación de escritorio destinada al usuario. Con la presente memoria se entrega además el conjunto de módulos implementados, tanto el código fuente como ficheros ejecutables del mismo. El código fuente estará en la carpeta llamada «Fuentes», mientras que los ejecutables estarán en la carpeta llamada «Ejecutables», dividiéndose en las carpetas «Servidor» y «Aplicación».

Servidor

Para la correcta ejecución de todos los módulos habrá que instalar primero la base de datos mongoDB[35] en el servidor donde se desea alojar los datos de la aplicación y seguir los pasos descritos en el apéndice 9 para blindar la seguridad de la base de datos. Si se desea cambiar la ubicación del servidor o hacerlo de manera local en una máquina bastaría modificar el archivo bd.conf a los parámetros deseados.

Adjunto a esta memoria se dan un conjunto de archivos .jar ejecutables que se tienen que poner a funcionar en un servidor de manera continua de modo que dichos módulos estén recolectando la información de Twitter de manera constante. Cada ejecutable tiene sus tiempos de ejecución propios, por lo que sólo habría que ponerlos a ejecutar en el servidor uno tras otro mediante línea de comandos (con la instrucción java -jar archivo.jar &). Los tres archivos .jar ejecutables que se encuentran en la carpeta «Ejecutables/Servidor» son los siguientes:

- tt.jar: Se encarga de la recolección automática cada 15 minutos de los trending topics de Twitter. Generado a partir del proyecto Java CapturaTT.
- links.jar: Se encarga de recolectar los tweets relevantes, y de clasificarlos por los dos métodos descritos en la sección 4.4. Este programa se ejecuta cada 24 horas, al final del día. Generado a partir del proyecto Java links.
- graficas.jar: Se encarga de la recolección de los datos necesarios para generar las gráficas de tweets por hora de las tendencias más importantes. Generado a partir del proyecto Java Graficas.

Aplicación de escritorio

Para poder ejecutar la aplicación desarrollada en primer lugar deben estar funcionando todos los módulos descritos anteriormente en la parte del servidor y tener instalada la última versión de Java, en concreto Java8[36]. Una vez

[35] https://www.mongodb.org/
[36] https://www.java.com/es/download/

conseguidos estos dos pasos, ejecutar la aplicación haciendo doble click en la aplicación «TrendSpy.jar» presente en la carpeta «Ejecutables/Aplicacion» y se mostrará la interfaz explicada en la sección 4.9.

De nuevo, si se desea cambiar la ubicación del servidor, modificar los parámetros pertinentes en el archivo «bd.conf».

12. Anexo D: Ejemplo de archivo gdf

Fragmento de archivo en formato gdf que genera la aplicación para su posterior lectura desde las librerías de Gephi, en este caso es un archivo generado el dia 7/6/2015. En una primera parte del archivo se definen los nodos del grafo los cuales tienen un nombre y una etiqueta que coinciden (nodedef>name VARCHAR,label VARCHAR). Posteriormente se definen las aristas (edgedef>node1 VARCHAR,node2 VARCHAR,weight INTEGER,label VARCHAR,links VARCHAR), las cuales definen los nodos que conectan, un peso, una etiqueta y una lista de enlaces que relacionan ambos nodos.

```
nodedef>name VARCHAR, label VARCHAR
#ExperienceEndesa,#ExperienceEndesa
#ChampionsLeagueFinal,#ChampionsLeagueFinal
#RepescaIsabelR,#RepescaIsabelR
Sergio García,Sergio García
#FCBarcelona,#FCBarcelona
#FelizDomingo,#FelizDomingo
#ChiringuitoChampions,#ChiringuitoChampions
Luis Enrique,Luis Enrique
Nedovic,Nedovic
Ter Stegen,Ter Stegen
Vamos Zaragoza,Vamos Zaragoza
Girona,Girona
Windmill Hill,Windmill Hill
Kevin Roldán,Kevin Roldán
....Continúa definiendo nodos
edgedef>node1 VARCHAR, node2 VARCHAR, weight INTEGER, label
    VARCHAR, links VARCHAR
El Barça,#ChampionsLeagueFinal,2,2, lavanguardia.com.
    Deporte marca.com.Deporte
Piqué,#ChampionsLeagueFinal,3,3, vanitatis.elconfidencial.
    comlavanguardia.com.Deportemarca.com.Deporte
#RolandGarros2015,Roland Garros,2,2, highlightsvenezuela.
    com elmundo.es.Deporte
Valencia Basket,La Juve,2,2, zipzp.eu rtve.es.
    Entretenimiento
...Continúa definiendo aristas
```

Referencias

Alcázar Jaén, S. et al. (2013). Diseño e implementación de un sistema para el análisis y categorización en twitter mediante técnicas de clasificación automática de textos. 2.5.2

Anguita, M. A. & Lorenzo, R. M. (2014). Extracción, análisis y visualización de información social desde twitter. Proyecto de Sistemas Informáticos (Facultad de Informática, Curso 2013-2014). 2.5.1

Barabási, A.-L. & Albert, R. (1999). Emergence of scaling in random networks. *science*, 286(5439), 509–512. 2.2.2

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10), P10008. 2.2.3

Bufferapp (2011). Six secrets about whether twitter censors trending topics. www.blog.bufferapp.com. Accedido: 2014-12-12. 2.3.2

Castañeda, L. & Gutiérrez, I. (2010). Redes sociales y otros tejidos online para conectar personas. *Aprendizaje con Redes Sociales. Tejidos educativos en los nuevos entornos. Sevilla: MAD Eduforma*. 2.1

Cormen, T., Leiserson, C., Rivest, R., & Stein, C. (2001). The rabin–karp algorithm. *Introduction to Algorithms*, (pp. 911–916). 2.4

Cruz, E. G. (2014). ¿ hay un ethos en twitter? *VIRTUalis*, 2(3), 18–24. 2.3.4

ERDdS, P. & R&WI, A. (1959). On random graphs i. *Publ. Math. Debrecen*, 6, 290–297. 2.2.2

Fiaidhi, J., Mohammed, S., Islam, A., Fong, S., & Kim, T.-h. (2013). Developing a hierarchical multi-label classifier for twitter trending topics. *International Journal of u-and e-Service, Science and Technology*, 6(3), 1–12. 2.5.3

Genbetadev (2014). Bases de datos nosql. elige la opción que mejor se adapte a tus necesidades. www.genbetadev.com. Accedido: 2014-11-30. 3.2

Hipertextual (2010). ¿qué información quiere leer la gente en twitter? www.hipertextual.com. Accedido: 2014-11-30. 2.3.4

Ignitesocialmedia (2012). Trending on twitter: A look at algorithms behind trending topics. www.ignitesocialmedia.com. Accedido: 2014-12-12. 2.3.2

Karp, R. M. & Rabin, M. O. (1987). Efficient randomized pattern-matching algorithms. *IBM Journal of Research and Development*, 31(2), 249–260. 2.4

Knuth, D. E., Morris, Jr, J. H., & Pratt, V. R. (1977). Fast pattern matching in strings. *SIAM journal on computing*, 6(2), 323–350. 2.4

Kwak, H., Lee, C., Park, H., & Moon, S. (2010). What is twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10 (pp. 591–600). New York, NY, USA: ACM. 2.3.3

Merayo, R. V. (2011). Rich internet applications (ria) y accesibilidad web. *Hipertext. net*, (9), 2. 3.4

Newman, M. E. (2006). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23), 8577–8582. 2.2.3

Newman, M. E. J. & Girvan, M. (2004). Finding and evaluating community structure in networks. *Phys. Rev. E*, 69, 026113. 2.2.3

Statista (2015). Number of monthly active twitter users worldwide from 1st quarter 2010 to 1st quarter 2015 (in millions). www.statista.com. Accedido: 2014-10-30. 2.1, 2.3.1

Stoimen (2012). Computer algorithms: Rabin-karp string searching. www.stoimen.com. Accedido: 2015-01-10. 2.4

Tweetsmarter (2011). How i uncovered twitters trending topics secrets. www.tweetsmarter.com. Accedido: 2014-12-12. 2.3.2