



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

# Observatorio de interacciones en Twitter

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

*Autor:* Elizabeth Vivanco Sánchez

*Tutor:* Vicente Juan Botti Navarro,  
Elena del Val Noguera

Curso 2016-2017



# Resumen

En este Trabajo de Fin de Grado (TFG) se ha desarrollado una herramienta, que permite recoger, observar, y analizar las interacciones entre los usuarios de una red social, en este caso, de Twitter. Las interacciones se analizan por hashtags y se aplican métricas de redes complejas y análisis de sentimientos además de información geoposicionada para comprender el comportamiento de los usuarios respecto a un tema. Los tweets pueden filtrarse eligiendo el rango de días que se desea analizar y se estudia a través del tiempo la evolución de diversas métricas. Las diferentes vistas permiten ver el grafo que representa la red de interacciones en el tiempo especificado, donde los nodos corresponden a los usuarios y los arcos a la existencia de una interacción entre usuarios. También es posible observar varias gráficas y los usuarios más influyentes correspondientes a la red generada. Además, se observa el sentimiento de los mensajes, tanto de los tweets geolocalizados como de los existentes en el rango de tiempo especificado y, por último, dos nubes de palabras, una para los mensajes con sentimiento positivo y otra para los negativos. El análisis proporcionado por la herramienta permite entender con más detalle el comportamiento de los usuarios en redes sociales. Este análisis puede ser aplicado, entre otros, en el campo de la recomendación y persuasión, en marketing, como sensor social, para detectar usuarios influyentes, entender los flujos de información, etc.

**Palabras clave:** redes complejas, análisis de sentimientos, análisis de redes sociales, información geolocalizada

---

# Resum

En aquest Treball de Fi de Grau (TFG) s'ha desenvolupat una eina, que permet recollir, observar, i analitzar les interaccions entre els usuaris d'una xarxa social, en aquest cas, de Twitter. Les interaccions s'analitzen per hashtags i s'apliquen mètriques de xarxes complexes i anàlisi de sentiments a més d'informació geolocalitzada per comprendre el comportament dels usuaris respecte a un tema. Els tweets es poden filtrar triant el rang de dies que es desitja analitzar i s'estudia a través del temps l'evolució de diverses mètriques. Les diferents vistes permeten veure el graf que representa la xarxa d'interaccions en el temps especificat, on els nodes corresponen als usuaris i els arcs a l'existència d'una interacció entre usuaris. També és possible observar diverses gràfiques i els usuaris més influents corresponents a la xarxa generada. A més, s'observa el sentiment dels missatges, tant dels tweets geolocalitzats com dels existents en el rang de temps especificat i, finalment, dos núvols de paraules, una per als missatges amb sentiment positiu i una altra per als negatius. L'anàlisi proporcionat per l'eina permet entendre amb més detall el comportament dels usuaris en xarxes socials. Aquesta anàlisi pot ser aplicat, entre altres, en el camp de la recomanació i persuasió, en màrqueting, com a sensor social, per detectar usuaris influents, entendre els fluxos d'informació, etc.

**Paraules clau:** xarxes complexes, anàlisi de sentiments, anàlisi de xarxes socials, informació geolocalitzada

---

# Abstract

This project presents a tool that allows to collect, visualize, and analyze interactions between users of the Twitter social network. The interactions (tweets) are analyzed by hashtags, location, or both. Based on these interactions, the tool applies complex network metrics and sentiment analysis to better understand users' behavior. Tweets can be filtered by choosing a range of days to be analyzed and the evolution of various metrics is studied over time. The different views allow to see the graph representing the network of interactions in the specified time, where the nodes correspond to the users and the links to the existence of an interaction between users. It is also possible to observe several graphics of structural network properties and the most influential users of the generated interaction network. In addition, it is possible to observe the sentiment of the users' messages (i.e., geolocated and/or no geolocated tweets) and two word clouds, one for the messages with positive sentiment and another for the negative ones. The analysis provided by the tool allows to understand in more detail the behavior of users in social networks. This analysis can be applied, among others, in the field of recommendation and persuasion, in marketing, as a social sensor, to detect influential users, to understand information flows, etc.

**Key words:** complex networks, sentiment analysis, social network analysis, geolocated information

---

# Índice general

---

<b>Índice general</b>	<b>V</b>
<b>Índice de figuras</b>	<b>VII</b>
<b>Índice de tablas</b>	<b>VII</b>
<hr/>	
<b>1 Introducción</b>	<b>1</b>
1.1 Motivación . . . . .	1
1.2 Objetivos . . . . .	4
1.3 Estructura de la memoria . . . . .	5
<b>2 Estado del arte</b>	<b>7</b>
<b>3 Propuesta</b>	<b>11</b>
3.1 Especificación de requisitos . . . . .	11
3.2 Diseño . . . . .	20
3.2.1 Descripción de la arquitectura . . . . .	20
3.2.2 Flujos de información . . . . .	21
3.2.3 Modelo de datos . . . . .	23
3.2.4 Diseño de la interfaz . . . . .	24
3.3 Desarrollo . . . . .	28
3.3.1 Implementación del modelo MVC . . . . .	28
3.3.2 Librerías y recursos . . . . .	30
3.3.3 Funcionalidad y vistas . . . . .	33
<b>4 Resultados y discusión</b>	<b>39</b>
4.1 Análisis <i>hashtag trump</i> . . . . .	39
4.2 Análisis <i>hashtag londonbus</i> . . . . .	43
<b>5 Conclusiones</b>	<b>47</b>
5.1 Valoración personal . . . . .	48
5.2 Trabajo Futuro . . . . .	49
5.3 Publicaciones del TFG . . . . .	49
<b>Bibliografía</b>	<b>51</b>



## Índice de figuras

---

1	Diagrama de casos de uso de la vista principal . . . . .	12
2	Diagrama de casos de uso de la vista estructural . . . . .	16
3	Diagrama de casos de uso de la vista de geolocalización y análisis de sentimientos . . . . .	18
4	Diagrama de la arquitectura del sistema . . . . .	21
5	Diagrama de secuencia cuando se realiza una petición HTTP . . . . .	22
6	Diagrama de secuencia cuando se realiza una petición websocket . . . . .	23
7	Modelo de datos del sistema . . . . .	24
8	Mockup de la vista principal . . . . .	25
9	Mockup de la vista estructural . . . . .	26
10	Mockup de la vista geolocalizada y de análisis de sentimientos . . . . .	27
11	Arquitectura interna de los componentes del patrón de diseño MVC . . . . .	29
12	Registro de aplicación en Twitter . . . . .	32
13	Vista principal . . . . .	33
14	Vista estructural 1 . . . . .	35
15	Vista estructural 2 . . . . .	35
16	Vista estructural 3 . . . . .	36
17	Vista geolocalizada y de análisis de sentimientos 1 . . . . .	37
18	Vista geolocalizada y de análisis de sentimientos 2 . . . . .	38
19	Métricas del <i>hashtag trump</i> y evolución del sentimiento. . . . .	40
20	Rankings del <i>hashtag trump</i> . . . . .	41
21	Mapa del sentimiento geolocalizado del <i>hashtag trump</i> . . . . .	42
22	Nubes de palabras del <i>hashtag trump</i> (positiva a la izquierda, negativa a la derecha). . . . .	42
23	Métricas del <i>hashtag londonbus</i> y evolución del sentimiento. . . . .	44
24	Rankings del <i>hashtag londonbus</i> . . . . .	44
25	Nubes de palabras del <i>hashtag londonbus</i> (positiva a la izquierda, negativa a la derecha). . . . .	45

## Índice de tablas

---

1	Análisis comparativo de trabajos previos [1, 15] en función de las características que tienen en cuenta: estructura de la red, información geolocalizada, análisis del sentimiento, tiempo . . . . .	10
---	--	----

---

2	Caso de uso Registrarse . . . . .	13
3	Caso de uso Iniciar sesión . . . . .	13
4	Caso de uso Crear nueva tarea . . . . .	13
5	Caso de uso Añadir datos de la tarea . . . . .	14
6	Caso de uso Modificar tarea . . . . .	14
7	Caso de uso Introducir nuevos datos . . . . .	14
8	Caso de uso Archivar tarea . . . . .	15
9	Caso de uso Parar recogida de mensajes . . . . .	15
10	Caso de uso Ver vistas de una tarea . . . . .	15
11	Caso de uso Dibujar grafo . . . . .	16
12	Caso de uso Analizar grafo . . . . .	16
13	Caso de uso Cambiar de vista . . . . .	17
14	Caso de uso Especificar rango de fechas . . . . .	17
15	Caso de uso Especificar modo de análisis del grafo . . . . .	17
16	Caso de uso Añadir geoJson . . . . .	18
17	Caso de uso Analizar sentimiento . . . . .	19
18	Caso de uso Elegir geoJson a utilizar o seleccionar región . . . . .	19



---

---

# CAPÍTULO 1

## Introducción

---

### Motivación

---

Las redes sociales online son cada vez más utilizadas y generan gran cantidad de datos; según el Estudio anual del uso de redes sociales 2016 del IAB; más de 15 millones de usuarios en España de entre 16 a 55 años utilizan redes sociales frente a los 14 millones de usuarios (representando el 82 % de los internautas en España) que las usaban según el estudio realizado en el 2015 por la misma organización y al 79 % de los internautas en el estudio del 2014, por lo que se observa un aumento del uso de las redes sociales. En el Estudio anual del 2016 del IAB también observaron que un 85 % de los usuarios de redes sociales declaró que seguían a influencers, por lo que sería interesante estudiar a esos usuarios. En el caso concreto de Twitter, que es la red social que se analizará en este trabajo, se estima que tiene más de 500 millones de usuarios, genera 65 millones de tweets al día y recibe 3000 millones de peticiones diarias.

Los datos que nos proporcionan las redes sociales al recoger un mensaje publicado por un determinado usuario son, entre otros, dónde está el usuario (si el usuario ha habilitado la opción sobre información geoposicionada), en qué momento se ha realizado una acción, qué acción ha realizado (pueden ser de diversos tipos como: publicación de un mensaje, contestar al mensaje de otro usuario, establecer una relación de amistad con otro usuario, interacción con otra persona por medio de un mensaje, etc.) y, si se trata de un mensaje, se puede tener acceso al contenido de este. Estos datos son importantes porque nos permiten de una manera objetiva extraer características relacionadas con el comportamiento de los usuarios.

Hoy en día, dado el nivel de información y datos que se crean en Internet es necesario disponer de herramientas que nos ayuden a realizar los análisis que nos interesen de una forma fácil y eficiente. Realizar a mano cualquier tipo de análisis o recogida de información de una red social podría ser demasiado costoso y se podría dar lugar a equivocaciones; con una herramienta que nos facilite el trabajo sólo tenemos que preocuparnos de sacar conclusiones a partir de los diversos análisis realizados.

En este TFG (Trabajo de Fin de Grado) nos hemos centrado en analizar las relaciones que aparecen a partir de las interacciones, ya sea por la interacción con otro usuario o por una acción individual. La razón de elegir analizar este tipo de relaciones y no las de amistad entre usuarios se debe a que muchas veces se da el caso de que se tienen muchas amistades con las cuales no se interactúa y, por lo tanto, no es considerada una relación “activa”. Sin embargo, las relaciones basadas en interacciones reflejan una relación más “real”.

En una red social en la que cada día se publican millones de tweets, aunque se escoja una determinada temática para realizar un análisis de la red puede que existan miles o millones de mensajes dependiendo de la popularidad del hashtag. Si realizamos un análisis de todos los mensajes existentes en los hashtags relacionados con un tema en concreto (un programa de televisión, un partido de fútbol, un evento social, etc.), sin tener en cuenta la fecha en la que fueron publicados esos mensajes, nos estaremos perdiendo una gran cantidad de información sobre la evolución de las interacciones de los usuarios.

La posibilidad de escoger un rango de fechas para realizar el análisis de esa red de interacciones nos permite: la detección de cambios en el comportamiento (i.e., cambios en las medidas de redes complejas y en el sentimiento) y analizar con más precisión fechas que consideremos relevantes. Por lo tanto, tener en cuenta el tiempo es un factor importante.

Dentro del análisis de las redes de interacciones de los usuarios, existen medidas que nos permiten observar quienes son los usuarios más influyentes (personas que más participan, a las que más contestan, a las que más nombran...). Los mensajes que publiquen esos usuarios serán vistos por un mayor número de personas que los mensajes de otro usuario con un nivel de influencia bajo, por lo que es interesante localizarlos ya que sus opiniones pueden influir en la de los demás.

Si analizamos el sentimiento de un conjunto de mensajes podemos observar cuál es la opinión general acerca de un tema o un evento, y si analizamos el cambio de este por días podemos detectar cambios bruscos en la opinión de los usuarios. Esto es muy útil si, además, tenemos disponible sobre qué se está hablando cuando el sentimiento de los mensajes es positivo o negativo ya que nos permite entender que es lo que está causando ese sentimiento y si se producen cambios drásticos del sentimiento podemos analizar si ha habido un cambio de las palabras más utilizadas, entendiendo que es lo que ha provocado el cambio.

---

Además, si se dispone de información sobre la geolocalización de los usuarios que interactúan en el tema a analizar, aunque sólo tengan activada la opción de hacer pública su ubicación un conjunto del total de usuarios participantes, es muy interesante realizar un análisis del sentimiento por zonas, ya que se puede analizar el sentimiento medio de los usuarios en los que su geolocalización obtenida esté dentro de una zona y así observar las tendencias de sentimiento que tienen diversas zonas reales y su evolución.

Este trabajo está motivado por una de las líneas de investigación del GTI-IA (<http://www.gti-ia.upv.es/>) centrada en el análisis del comportamiento a través de medios sociales. Concretamente, el trabajo forma parte de una herramienta cuyo objetivo es proporcionar distintos análisis y visualizaciones de información extraída de las redes sociales para facilitar al usuario la toma de decisiones [7, 8, 9, 16].

## Objetivos

---

En este TFG el objetivo principal es desarrollar una herramienta web que permita recoger datos de las comunicaciones que se llevan a cabo en las redes sociales y analizarlos de una manera intuitiva y visual para entender el comportamiento de las personas en las redes sociales.

Para alcanzar este objetivo principal se plantean los siguientes subobjetivos:

- Recoger y procesar grandes cantidades de datos de las redes sociales.
- Analizar la evolución de la comunicación entre usuarios de redes sociales mediante el uso de métricas de redes complejas.
- Analizar el sentimiento de la información generada por los usuarios en las redes sociales
- Análisis de información geoposicionada teniendo en cuenta el sentimiento.
- Encontrar las palabras más utilizadas por los usuarios, distinguiendo entre mensajes positivos y negativos.
- Desarrollar una interfaz web para la especificación de criterios de búsqueda y visualización de resultados de los análisis.

---

## Estructura de la memoria

---

Previamente, se ha presentado la motivación por la cual se cree necesaria la creación de una herramienta que permita analizar una red social de forma sencilla e intuitiva. También se han presentado los objetivos que se han querido cumplir con la creación del trabajo presentado. A continuación, la memoria de este trabajo está dividida en cuatro partes.

En el capítulo 2 se presentará el estado del arte, donde se analizarán estudios similares realizados anteriormente y se expondrán qué mejoras se pueden integrar que no hayan sido realizadas anteriormente o no se suelen tener en cuenta. En el capítulo 3 se realiza la presentación de la herramienta propuesta que se ha implementado. Este capítulo se divide en tres partes, primero se realiza una especificación de los requisitos necesarios para el correcto funcionamiento de la herramienta; después se presentará el diseño a nivel formal de la organización de la herramienta y, finalmente, se explicará cómo se ha realizado el desarrollo de esta de una forma más técnica, presentando las tecnologías utilizadas y la visualización final de la herramienta.

Después, en el capítulo 4 se mostrarán dos casos de estudio de dos eventos de Twitter para ver la funcionalidad de la herramienta con casos reales, analizándose los resultados obtenidos y su significado. Finalmente, en el capítulo 5 se exponen las conclusiones del trabajo y las posibles ampliaciones que se pueden implementar en el futuro en la herramienta creada.



---

## CAPÍTULO 2

# Estado del arte

---

Debido al incremento constante del uso de las redes sociales y a la capacidad de obtener información de los usuarios y sus actividades de una forma rápida y sencilla, se han realizado numerosos estudios sociales y se han creado herramientas que utilizan estos datos para entender el comportamiento de conjuntos de usuarios. Muchos de estos análisis son realizados teniendo en cuenta un sólo método, como utilizar la estructura de la red, la información geolocalizada o el análisis del sentimiento de los usuarios. Sólo unos pocos trabajos tienen en cuenta varios de estos puntos de vista para sacar mejores conclusiones acerca del comportamiento social.

Un factor tan importante como es el tiempo y la evolución de las distintas métricas y propiedades no suele ser considerado en estos análisis, perdiendo una gran cantidad de información ya que sólo consideran el total de los datos recogidos o un instante de tiempo concreto.

Hay varios trabajos que sólo tienen en cuenta el análisis del sentimiento asociado a los mensajes a estudiar. Por ejemplo, en el trabajo presentado por Romina et. al [1] se analizan varias herramientas que sólo tienen en cuenta el sentimiento de los mensajes para obtener un resumen de las opiniones disponibles de los usuarios, sin tener en cuenta el tiempo. Una de las herramientas que analizan, SumView [2], simplemente selecciona las opiniones más representativas y las muestra, sin presentar el grado del sentimiento de cada mensaje. También han analizado el sistema AZFinText [3], utilizado para predecir noticias financieras, junto a una herramienta de análisis de sentimientos, dándose cuenta que estas noticias tienen un impacto directo en la influencia de los precios de las materias primas y acciones. Chamlerwat et. al [4] proponen un sistema basado en el análisis de sentimientos en Twitter para automatizar el análisis de las opiniones que los clientes tienen sobre los productos investigados. Identifican la polaridad, intensidad y tema de los sentimientos para que las empresas sean capaces de aprovechar estas opiniones a la hora de mejorar la calidad de sus productos, proporcionar un mejor servicio o identificar nuevas oportunidades de negocio. Sin embargo, no tiene en cuenta la evolución de los mensajes en el tiempo, por lo que pierden la posibilidad de detectar qué es lo que molesta a los clientes en la actualidad o si ha habido un cambio de opinión acerca del producto y en qué momento se ha producido ese cambio.

Hay otros estudios que han decidido tener en cuenta sólo la estructura de la red con el fin de observar las métricas asociadas a la red creada.

Del Val et. al [7] presentan un análisis de la evolución de las interacciones de los usuarios en eventos de varios tipos (televisivos, políticos, conferencias, etc.) en Twitter, teniendo en cuenta el tiempo y la estructura de la red. Modelan las interacciones como redes con marcadores de tiempo. Realizan un estudio de los cambios en las propiedades estructurales de la red creada con el fin de detectar patrones y diferencias en estos dependiendo del evento analizado.

También, Del Val et. al [8] presentan un sistema multiagente (SMA) para automatizar el proceso de recoger datos de la actividad de los usuarios en las redes sociales y realizar un análisis de la evolución del comportamiento social a lo largo del tiempo con diferentes niveles de granularidad en eventos online basados en métricas de teoría de grafos.

Últimamente se están creando cada vez más estudios que utilizan la información geolocalizada disponible en las redes sociales como medio para estudiar el comportamiento social de los usuarios. Un ejemplo de este tipo de estudios es el realizado por Mai et. al [6], que utilizan la información geolocalizada de los mensajes en Twitter que tengan que ver sobre accidentes que han ocurrido en un lugar concreto para realizar un análisis sobre el transporte. Comparan los accidentes registrados en la Patrulla de Carreteras de California con los mensajes relacionados con este tipo de evento durante el mismo período de tiempo y que su localización esté cerca del lugar del accidente con el fin de obtener más información acerca del accidente.

Del Val et. al [9] realizan una herramienta llamada U-Tool que utiliza la información geolocalizada disponible en Twitter. La herramienta permite explorar diferentes análisis de la actividad en áreas de una ciudad concreta en tiempo real, pudiendo detectar donde está pasando un evento, encontrar patrones de movilidad, detectar zonas de la ciudad con un mayor número de visitantes, predecir la actividad en un determinado punto de interés, localizar la ubicación óptima para una nueva instalación urbana, medir la accesibilidad espacial entre diferentes partes de la ciudad y detectar rutas alternativas.

Kling et. al [15] investigan la dinámica espacio-temporal de la información de acceso de servicios como Foursquare, distribuidos sobre el servicio de Twitter. Los eventos con marcas de tiempo y etiquetados geográficamente pueden usarse para cuantificar la intensidad de las actividades urbanas de un área y un periodo de tiempo.

Por otra parte, hay algunos estudios que tienen en cuenta dos de los métodos mencionados anteriormente conjuntamente. Por ejemplo, Caragea et. al [10] realizan una clasificación de los sentimientos de los mensajes publicados en Twitter durante el huracán Sandy y visualizan esos sentimientos en un mapa geográfico centrado alrededor del huracán. Muestran cómo los sentimientos de los usuarios cambian no sólo dependiendo de su localización, sino también debido a la distancia del desastre.



Hasan et. al [11] investigan el alcance de la influencia social sobre la actividad individual y las decisiones sobre el estilo de vida desde los datos de acceso en los medios sociales. Para ello, analizan la estructura de la red social e infieren patrones de acceso y geolocalización usando modelos de temas.

West et. al [5] utilizan la estructura de la red junto al análisis del sentimiento para predecir la opinión de un usuario acerca de otro, utilizando los mensajes relativos a ambos usuarios y una síntesis del contexto estructural dentro de la red social.

Charlton et. al [12] estudian la relación entre los niveles de sentimientos en usuarios de Twitter y la estructura de la red que los usuarios crean al mencionarse unos a otros; utilizan tres algoritmos de puntuación de sentimientos, entre ellos el programa SentiStrength [13]. Los autores detectaron que la gente que tiene potencialmente el mayor alcance de comunicación tienen un sentimiento positivo con más frecuencia que el usuario promedio. Otro de los aspectos que encontraron en el análisis fue que las comunidades estables estructuralmente durante un periodo de meses tienen unos niveles de sentimiento estables donde cambios repentinos se deben a eventos externos que afecten a la comunidad.

O'Sullivan et. al [14] examinan la relación entre estructura social y sentimiento mediante el análisis de una gran colección de mensajes sobre el referéndum de matrimonio irlandés de 2015. Han encontrado que el sentimiento de las menciones publicadas por usuarios está correlacionado con el sentimiento de las menciones recibidas y hay más conexiones entre usuarios con puntuaciones de sentimiento similares que entre usuarios con puntuaciones opuestas.

Considerando el contenido de trabajos previos (ver tabla 1), se propone la realización de una herramienta que permita la visualización de las tres perspectivas citadas anteriormente (estructura de la red, información geolocalizada y análisis de sentimientos), teniendo en cuenta el tiempo en el que los datos fueron creados, permitiendo así un mejor entendimiento del comportamiento social en las redes.

	Estructura de la red	Información geolocalizada	Análisis del sentimiento	Tiempo
[1]			X	
[2]			X	
[3]			X	
[4]			X	
[5]	X		X	
[6]		X		
[7]	X			X
[8]	X			X
[9]		X		
[10]		X	X	
[11]	X	X		
[12]	X		X	
[13]			X	
[14]	X		X	
[15]		X		X
Publicación asociada al TFG (5.3)	X	X	X	X

**Tabla 1:** Análisis comparativo de trabajos previos [1, 15] en función de las características que tienen en cuenta: estructura de la red, información geolocalizada, análisis del sentimiento, tiempo

---

---

# CAPÍTULO 3

## Propuesta

---

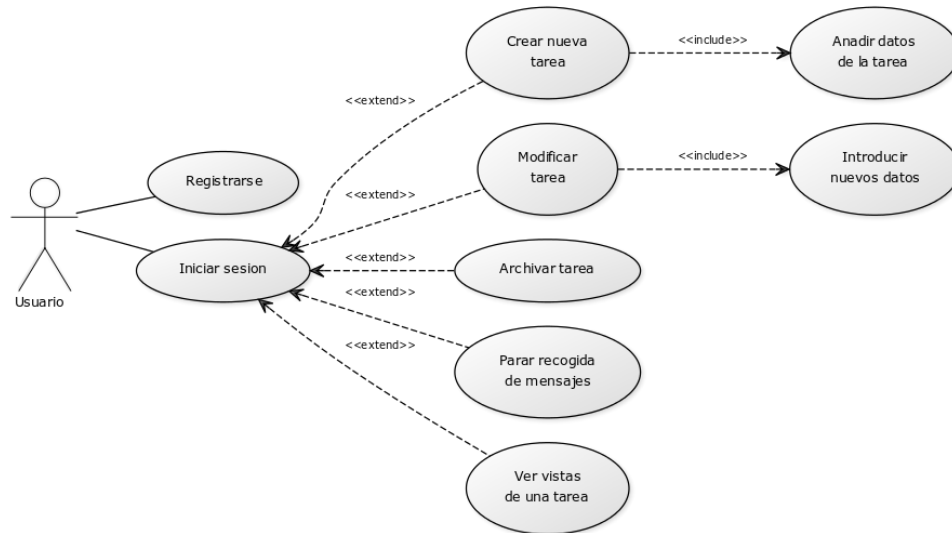
### Especificación de requisitos

---

En esta sección se presenta una especificación de los requisitos necesarios para la herramienta que se va a implementar en este TFG. En términos generales, la herramienta debe permitir la recogida automática de mensajes de una red social relacionados con el tema a analizar, realizar diversos tipos de análisis y visualizar los resultados de estos. La forma de visualizar los resultados se realizará a través de una aplicación web con diferentes vistas, donde estará contenida la herramienta.

En la vista principal se deben mostrar las distintas tareas, donde cada una corresponde a cada tema a analizar y cuyos mensajes asociados se recogen automáticamente y en tiempo real. Cada tarea se puede archivar/desarchivar, parar/iniciar la recogida de mensajes y modificar. También se puede crear una nueva tarea especificando el nombre deseado y el tema a analizar. Se crean otras dos vistas en la herramienta: la vista estructural y la vista de geolocalización y análisis de sentimientos.

En la vista estructural se muestra el análisis de la estructura de la red de interacciones de usuarios asociados a una tarea. Esta vista permite recoger y guardar los mensajes en un sistema de gestión de base de datos. Los mensajes deben poderse filtrar a la hora de realizar los análisis, modificando el rango de días que se desea tener en cuenta y se debe estudiar a través del tiempo la evolución de las diversas métricas. La vista debe permitir ver la red de interacciones en el tiempo especificado, donde los nodos corresponden a los usuarios y los arcos a la existencia de una interacción entre usuarios. Una interacción consiste en un mensaje donde se menciona y/o responde a otro usuario. También debe ser posible observar varias gráficas con la evolución de las distintas métricas útiles para el estudio del comportamiento social y la detección de los usuarios más influyentes correspondientes a la red generada.



**Figura 1:** Diagrama de casos de uso de la vista principal

En la vista de geolocalización y análisis de sentimiento se muestra el sentimiento de los mensajes, tanto de los mensajes geolocalizados como de los existentes en el rango de tiempo especificado. El sentimiento de los mensajes geolocalizados se muestra en el mapa incluido en la vista. Por otra parte, la evolución del sentimiento positivo y negativo del total de los mensajes se visualiza en una gráfica, no teniendo en cuenta los mensajes con un sentimiento neutral. La vista también permite la visualización de dos nubes de palabras, una para los mensajes con sentimiento positivo y otra para los negativos.

La vista permite al usuario realizar un filtrado de los mensajes que estén geolocalizados. Por una parte, se puede especificar una región sobre un mapa, mostrándose un indicador numérico del grado de sentimiento positivo o negativo de cada mensaje en la región especificada. Por otra parte, se puede añadir un geoJson y realizar el análisis sobre las regiones especificadas en el geoJson, cambiando el color de la región dependiendo de la media de las puntuaciones de los mensajes que están dentro de esa región. Un geoJson[17] es un archivo que representa datos de estructuras geográficas, junto con sus atributos no espaciales. Está basado en JSON, permitiendo un intercambio de datos rápido y sencillo.

A continuación, se muestran los diagramas de casos de uso de las funciones que se pueden realizar en cada una de las vistas disponibles y las tablas asociadas a cada función explicando con más detalle que se realiza en cada una.

Tablas para los casos de uso de la Fig. 1:

<b>Nombre</b>	<b>Registrarse</b>
<b>Actores</b>	Usuario
<b>Función</b>	Crear un nuevo usuario en el sistema
<b>Descripción</b>	Permite la creación de un nuevo usuario en el sistema para que el usuario sea capaz de acceder a la herramienta. Es un paso necesario, ya que no se puede acceder a las demás funciones del sistema sin tener un usuario propio.
<b>Comentarios</b>	

Tabla 2: Caso de uso Registrarse

<b>Nombre</b>	<b>Iniciar sesión</b>
<b>Actores</b>	Usuario
<b>Función</b>	Iniciar sesión para acceder a la herramienta
<b>Descripción</b>	Se introduce el nombre de usuario y la contraseña en las casillas correspondientes, mostrando la vista principal si los datos son correctos.
<b>Comentarios</b>	

Tabla 3: Caso de uso Iniciar sesión

<b>Nombre</b>	<b>Crear nueva tarea</b>
<b>Actores</b>	Usuario
<b>Función</b>	Crear una nueva tarea, iniciándose la recogida de mensajes
<b>Descripción</b>	Primero, se realiza el caso de uso <b>Añadir datos de la tarea</b> , y se acepta la creación de la tarea. La nueva tarea creada empieza a recoger mensajes según los datos especificados, creándose una nueva colección en la base de datos para guardar los mensajes.
<b>Comentarios</b>	
<b>Casos de uso relacionados</b>	Añadir datos de la tarea

Tabla 4: Caso de uso Crear nueva tarea

<b>Nombre</b>	<b>Añadir datos de la tarea</b>
<b>Actores</b>	Usuario
<b>Función</b>	Introducir los datos necesarios para la creación de una nueva tarea
<b>Descripción</b>	Se introduce el nombre deseado para la tarea y los hashtag o cuentas de Twitter sobre los que se desea que se recojan mensajes.
<b>Comentarios</b>	

**Tabla 5:** Caso de uso Añadir datos de la tarea

<b>Nombre</b>	<b>Modificar tarea</b>
<b>Actores</b>	Usuario
<b>Función</b>	Modificar una tarea ya existente
<b>Descripción</b>	Se indica, mediante el botón disponible, que se desea modificar una tarea concreta. Entonces, se lleva a cabo el caso de uso <b>Introducir nuevos datos</b> y modificando la tarea.
<b>Comentarios</b>	
<b>Casos de uso relacionados</b>	Introducir nuevos datos

**Tabla 6:** Caso de uso Modificar tarea

<b>Nombre</b>	<b>Introducir nuevos datos</b>
<b>Actores</b>	Usuario
<b>Función</b>	Introducir los nuevos datos que se desean cambiar
<b>Descripción</b>	Modificación de los datos de la tarea que se desean cambiar, introduciéndolos en la casilla correspondiente.
<b>Comentarios</b>	

**Tabla 7:** Caso de uso Introducir nuevos datos

<b>Nombre</b>	<b>Archivar tarea</b>
<b>Actores</b>	Usuario
<b>Función</b>	Archivar una tarea concreta en la herramienta
<b>Descripción</b>	Archivar, mediante el botón disponible para la acción, una tarea; moviéndola a otro apartado de la vista principal
<b>Comentarios</b>	

Tabla 8: Caso de uso Archivar tarea

<b>Nombre</b>	<b>Parar recogida de mensajes</b>
<b>Actores</b>	Usuario
<b>Función</b>	Detener la recogida automática de mensajes
<b>Descripción</b>	Detener, mediante el botón disponible, la recogida de mensajes, dejando la tarea en el estado en el que se encontraba a la hora de realizar la acción.
<b>Comentarios</b>	

Tabla 9: Caso de uso Parar recogida de mensajes

<b>Nombre</b>	<b>Ver vistas de una tarea</b>
<b>Actores</b>	Usuario
<b>Función</b>	Salir de la vista principal
<b>Descripción</b>	Ver las vistas donde se realizan los análisis de una tarea concreta, saliendo de la vista principal.
<b>Comentarios</b>	

Tabla 10: Caso de uso Ver vistas de una tarea

Tablas para los casos de uso de la Fig. 2:

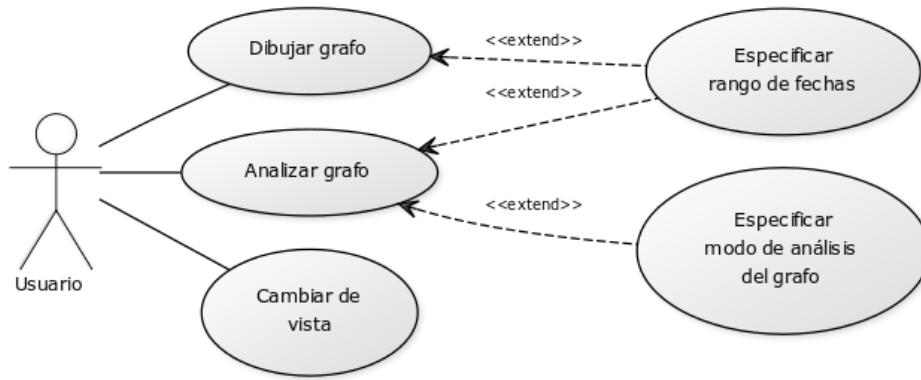


Figura 2: Diagrama de casos de uso de la vista estructural

<b>Nombre</b>	<b>Dibujar grafo</b>
<b>Actores</b>	Usuario
<b>Función</b>	Dibujar el grafo que representa la red de interacciones de usuarios una tarea
<b>Descripción</b>	Creación y visualización del grafo que representa la estructura de la red dentro del rango de fechas especificado. Los nodos corresponden a los usuarios que han publicado un mensaje con o sin hacer referencia a otro usuario. Las aristas representan una interacción entre usuarios pudiendo ser esta de distintos tipos (mención, respuesta o retweet). Se utiliza una librería específica para la creación y visualización.
<b>Comentarios</b>	

Tabla 11: Caso de uso Dibujar grafo

<b>Nombre</b>	<b>Analizar grafo</b>
<b>Actores</b>	Usuario
<b>Función</b>	Realizar los análisis de la estructura del grafo
<b>Descripción</b>	Se realizan análisis de métricas de redes complejas sobre el grafo creado en las fechas especificadas, viendo la evolución de las mismas por día con el fin de sacar conclusiones acerca del comportamiento acerca del comportamiento de los usuarios en la red.
<b>Comentarios</b>	

Tabla 12: Caso de uso Analizar grafo



<b>Nombre</b>	<b>Cambiar de vista</b>
<b>Actores</b>	Usuario
<b>Función</b>	Cambiar a la vista deseada
<b>Descripción</b>	Cambiar de la vista actual a otra vista mediante los botones disponibles, pudiendo acceder a la vista principal, la vista estructural o la vista de geolocalización y análisis de sentimientos.
<b>Comentarios</b>	

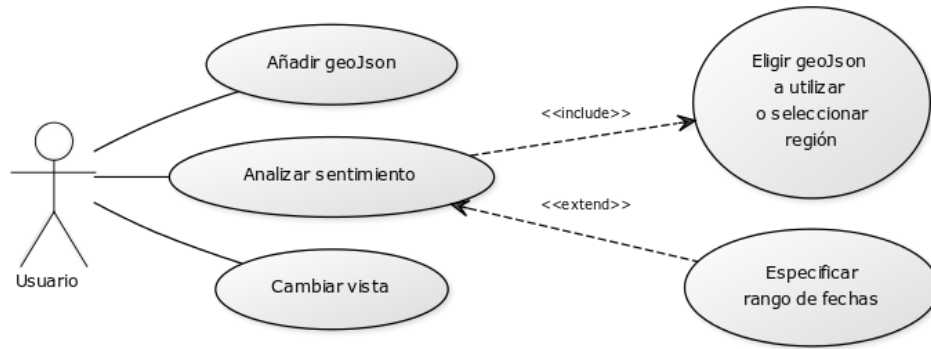
**Tabla 13:** Caso de uso Cambiar de vista

<b>Nombre</b>	<b>Especificar rango de fechas</b>
<b>Actores</b>	Usuario
<b>Función</b>	Modificar fechas
<b>Descripción</b>	Modificar el rango de fechas en el que se tendrán en cuenta los análisis a realizar.
<b>Comentarios</b>	

**Tabla 14:** Caso de uso Especificar rango de fechas

<b>Nombre</b>	<b>Especificar modo de análisis del grafo</b>
<b>Actores</b>	Usuario
<b>Función</b>	Especificar la forma en que se crea el grafo para, posteriormente, realizar el análisis de las métricas.
<b>Descripción</b>	Ya que el análisis de las métricas se realiza por días, existen dos formas de determinar el valor de cada métrica en cada día. Si se elige el modo agregado, se tiene en cuenta el grafo creado con los mensajes desde la fecha de inicio seleccionada hasta cada uno de los días, añadiendo los mensajes de cada día al total de mensajes del día anterior para realizar el análisis, y así sucesivamente, hasta la fecha de fin seleccionada. Si se elige el modo no agregado, en cada uno de los días especificados dentro del rango, se crea el grafo con sólo los mensajes de ese día y se obtiene el valor de la métrica.
<b>Comentarios</b>	

**Tabla 15:** Caso de uso Especificar modo de análisis del grafo



**Figura 3:** Diagrama de casos de uso de la vista de geolocalización y análisis de sentimientos

Tablas para los casos de uso de la Fig. 3:

<b>Nombre</b>	<b>Añadir geoJson</b>
<b>Actores</b>	Usuario
<b>Función</b>	Introducir un geoJson
<b>Descripción</b>	El usuario es capaz de subir su propio geoJson existente en su sistema de archivos. Se comprueba que el geoJson proporcionado es correcto y si no hay ningún problema con el archivo seleccionado por el usuario, este se guarda en la base de datos. El geoJson aparecerá disponible para ser seleccionado por el usuario en un menú desplegable.
<b>Comentarios</b>	

**Tabla 16:** Caso de uso Añadir geoJson

<b>Nombre</b>	<b>Analizar sentimiento</b>
<b>Actores</b>	Usuario
<b>Función</b>	Realizar el análisis de sentimiento
<b>Descripción</b>	Dependiendo de la opción elegida en el caso de uso <b>Elegir geoJson a utilizar o seleccionar región</b> , el análisis de sentimiento geolocalizado se realiza de forma diferente. En el caso de escoger un determinado geoJson, este se muestra en un mapa y se colorean las zonas con el sentimiento medio de los mensajes que están dentro de cada región. En el caso de no elegir ningún geoJson, se muestran los mensajes que están dentro de la región especificada por el usuario junto con su puntuación. Además, se muestra la evolución del sentimiento positivo y negativo del total de los mensajes dentro del rango de fechas seleccionado, sin necesidad de que estén geolocalizados y dos nubes de palabras con las palabras más frecuentes de los mensajes positivos y negativos.
<b>Comentarios</b>	
<b>Casos de uso relacionados</b>	Elegir geoJson a utilizar o seleccionar región

Tabla 17: Caso de uso Analizar sentimiento

<b>Nombre</b>	<b>Elegir geoJson a utilizar o seleccionar región</b>
<b>Actores</b>	Usuario
<b>Función</b>	Escoger la forma de análisis de sentimiento
<b>Descripción</b>	Mediante una lista se puede elegir uno de los geoJson específicos de la tarea a analizar guardados en la base de datos o elegir no utilizar ningún geoJson. En caso de elegir la segunda opción, el usuario debe seleccionar una región con la opción disponible en el mapa para ser capaz de realizar el análisis.
<b>Comentarios</b>	

Tabla 18: Caso de uso Elegir geoJson a utilizar o seleccionar región

---

## Diseño

---

Teniendo en cuenta la especificación de requisitos que hemos determinado en la sección anterior, ahora debemos definir los elementos necesarios para realizar la herramienta descrita. En esta sección se va a presentar la descripción de la arquitectura necesaria, los diagramas de secuencia para entender qué información se van a intercambiar los componentes del sistema, el modelo de datos escogido y los *mockups* de las distintas vistas a implementar.

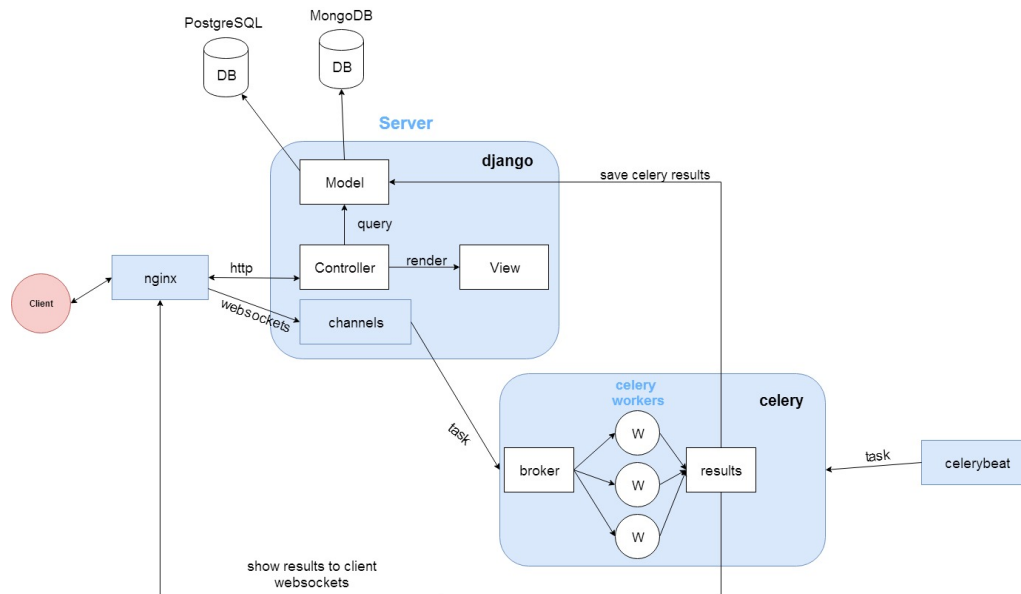
### Descripción de la arquitectura

En esta sección se describen los principales componentes de la arquitectura. Como puede observarse en el diagrama de la arquitectura (ver Fig. 4), en el que se muestran los componentes de la herramienta y como interactúan entre ellos, se ha utilizado principalmente un patrón de diseño de tipo *Model-View-Controller* mediante el *framework* de desarrollo web Django [18] y Celery [19] para realizar las tareas más pesadas en segundo plano. A continuación se va a describir cada uno de los componentes de la arquitectura.

En la parte *Client* se encuentra el navegador web que permite visualizar la herramienta. Se utiliza AngularJS [20] para conectar con el servidor web y realizar las peticiones necesarias a Django, obteniendo la respuesta y mostrando el resultado de la acción que se ha realizado. Se pueden realizar las peticiones de dos formas: mediante una petición HTTP, que es llevada a cabo por los componentes *Model-View-Controller*, o mediante *websockets*, donde las tareas más pesadas se pasan a Celery a través de *Channels* para realizarlas en segundo plano y pasar los resultados al servidor directamente, guardando el resultado también en la base de datos.

Existen dos *DBs* diferentes en el sistema: por una parte, se utiliza PostgreSQL [23] para guardar los modelos contenidos en la parte *Model* de la arquitectura; por otra parte, utilizamos MongoDB [22] sólo para almacenar los mensajes recogidos de las tareas disponibles, con los que se realizarán los análisis posteriormente. Nginx [21] es un servidor web que hace de *proxy* entre el *Client* y el resto de la arquitectura. Redirige las peticiones HTTP y *websockets* al componente adecuado, al *Controller* o a *Channels* respectivamente. El *Controller* realiza consultas al *Model* para conseguir información o actualizarla, y selecciona y renderiza la vista correspondiente mediante el *View*. En Django un *Controller* consiste en los archivos *urls.py* y *views.py* existentes. El *Model* se encarga de los datos, accede a la información y actualiza la base de datos. Los archivos correspondientes en Django se llaman *models.py*.

*View* se encarga de mostrar la salida de los datos que proporciona el componente *Model*; en nuestro caso, se trata de las plantillas HTML (archivos *\*.html*) que corresponden a cada vista.



**Figura 4:** Diagrama de la arquitectura del sistema

*Channels* es el nombre que Django le ha asignado a la extensión para soportar llamadas mediante *websockets*, enviando la tarea correspondiente al *Broker*. En el *Broker* se utiliza el servidor rabbitMQ [24] para hacer pasar las tareas mediante mensajes a los *CeleryWorkers*.

Los *CeleryWorkers* se encargan de realizar las tareas en segundo plano, una vez se tienen los resultados de la petición este se pasa al servidor *Nginx* para mostrar el resultado al *Client* y también se pasa al *Model* para guardar los resultados y no repetir cálculos costosos ya realizados.

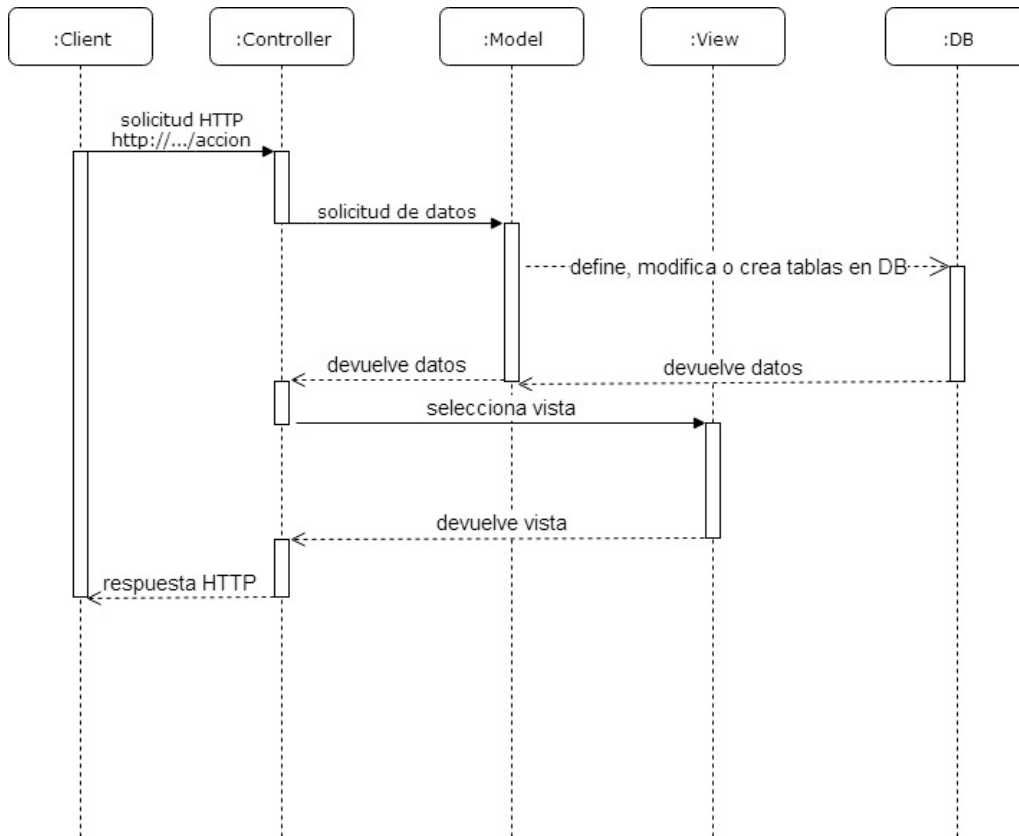
*Celerybeat* permite realizar tareas que son muy costosas y se programan a una hora determinada, realizándose de la misma forma que las tareas que se pasan a través de *Channels*.

## Flujos de información

Se han realizado dos diagramas de secuencia, uno para las peticiones HTTP (ver Fig. 5) y otro para las peticiones *websocket* (ver Fig. 6), para entender qué información se intercambian los distintos componentes del sistema a la hora de satisfacer las peticiones dependiendo del tipo de estas.

En ambos diagramas se ha omitido el componente *Nginx*, ya que sólo realiza la función de *proxy* entre el *Client* y el resto del sistema, por lo que sólo traslada las peticiones al componente que va a realizar la operación y devuelve el resultado al *Client*.

En el diagrama de secuencia para peticiones HTTP, cuando el *Client* desea realizar una acción disponible en el sistema, esta es pasada al *Controller*, quien accede al *Model* correspondiente para realizar la operación necesaria en la base de datos, ya sea crear una nueva tabla o recuperar información. Finalmente se selecciona la vista correspondiente, que se actualiza con los datos recuperados, y se renderiza, devolviendo la respuesta al *Client*.



**Figura 5:** Diagrama de secuencia cuando se realiza una petición HTTP

En el diagrama de secuencia para peticiones *websocket*, la petición es pasada a *Channels*, que a su vez pasa la tarea a realizar al *Broker* de Celery. Esta es pasada a un *CeleryWorker* disponible, que al terminar de realizar la tarea, entrega el resultado al *Client* para visualizar los resultados directamente y al *Model* para que el resultado sea guardado en la base de datos PostgreSQL.

Las tareas que están programadas, son enviadas a Celery mediante el componente Celerybeat y realizan el mismo intercambio de datos que en el diagrama de secuencia de peticiones *websockets* a partir del componente *Broker*, con la salvedad de que no necesitan enviar resultados al *Client*, que en este caso no existe, sino que tan sólo guardan los resultados en la *DB*.

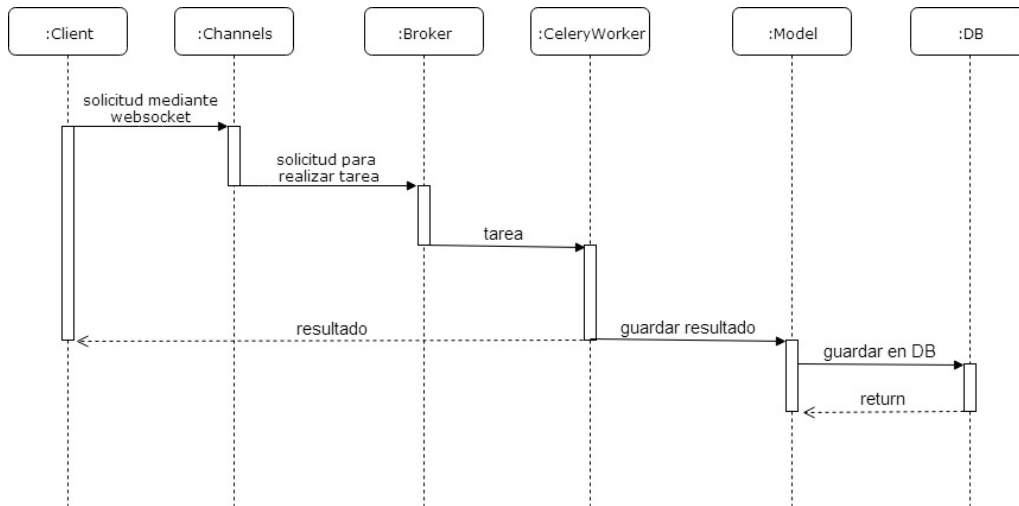


Figura 6: Diagrama de secuencia cuando se realiza una petición websocket

## Modelo de datos

A continuación se presenta el modelado de datos (ver Fig. 7) de la base de datos PostgreSQL (base de datos de tipo SQL), con las tablas necesarias para implementar las vistas disponibles.

Primero, tenemos la tabla *CollectHashtagTask*, que se identifica por un identificador único y los campos sirven para localizar la tarea de la que se deben recoger mensajes. Por cada tarea creada existe una colección diferente en MongoDB, donde se guardan los mensajes recogidos de la red social.

Por otra parte, las tablas *DailyGraph* se encargan de guardar los grafos diarios por tareas, guardando el día y el grafo correspondiente de ese día en data; las tablas *GeoJsonPolygons* guardan los archivos GeoJson subidos por el usuario en la tarea específica donde se ha subido el archivo, guardando el nombre del archivo y los contenidos si es un archivo correcto, y las tablas *Job* guardan las tareas que se deben realizar mediante *websockets* en Celery, actualizándola con el resultado una vez terminada. Los resultados intermedios de las tareas que realiza Celery y el porcentaje del progreso de la tarea que se está realizando se guardan en PostgreSQL.

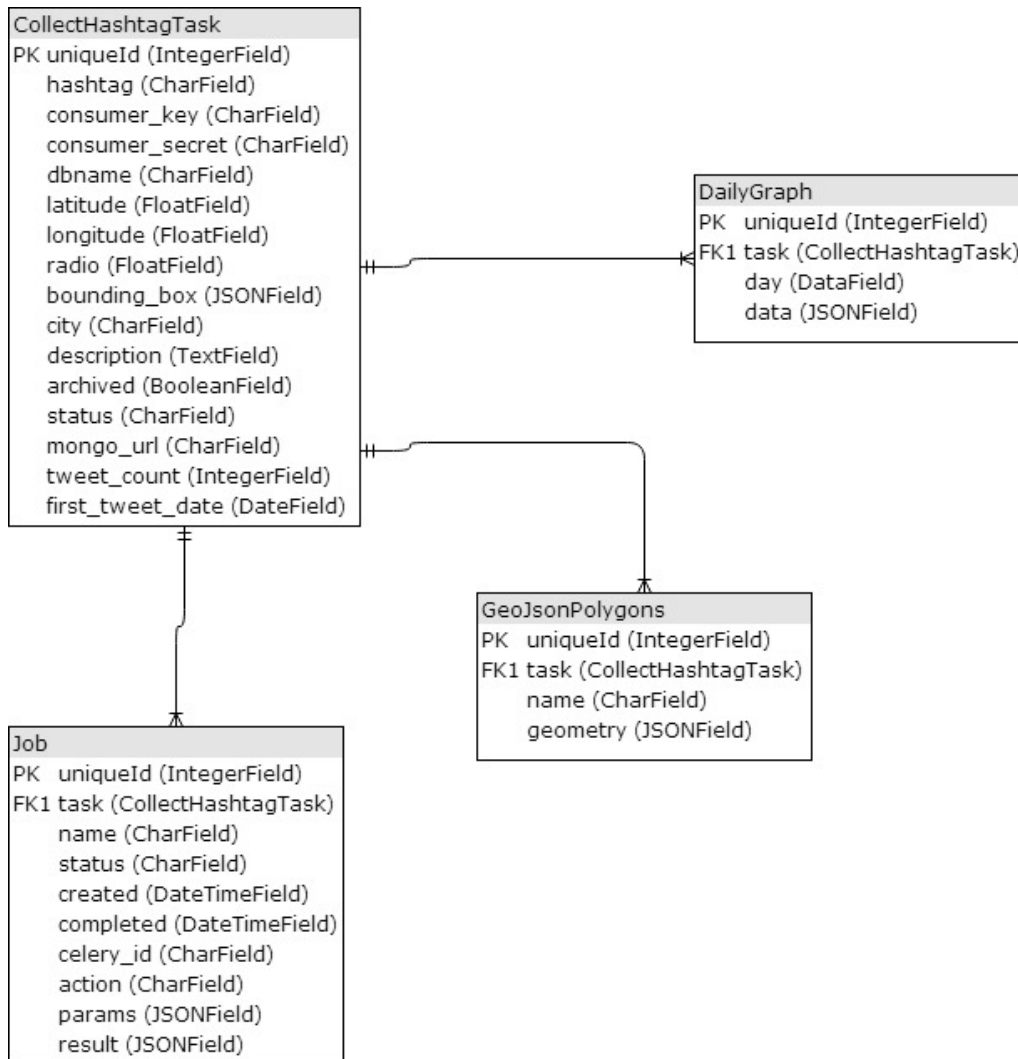


Figura 7: Modelo de datos del sistema

## Diseño de la interfaz

En esta sección se muestra el diseño de las vistas antes especificadas (Capítulo 3.1): la vista principal (ver Fig. 8), la vista estructural (ver Fig. 9) y la vista geolocalizada y de análisis de sentimientos (ver Fig. 10).

En la **vista principal** se pueden observar las acciones que se pueden realizar en cada una de las tareas mediante los botones disponibles, a la izquierda se encuentran los botones para ir a las tareas archivadas y a la vista principal, y a la derecha se encuentra el formulario para crear una nueva tarea y unos botones para realizar las acciones en todas las tareas disponibles. Para acceder a las otras dos vistas específicas de una tarea en concreto, se accede a través del nombre de la tarea.



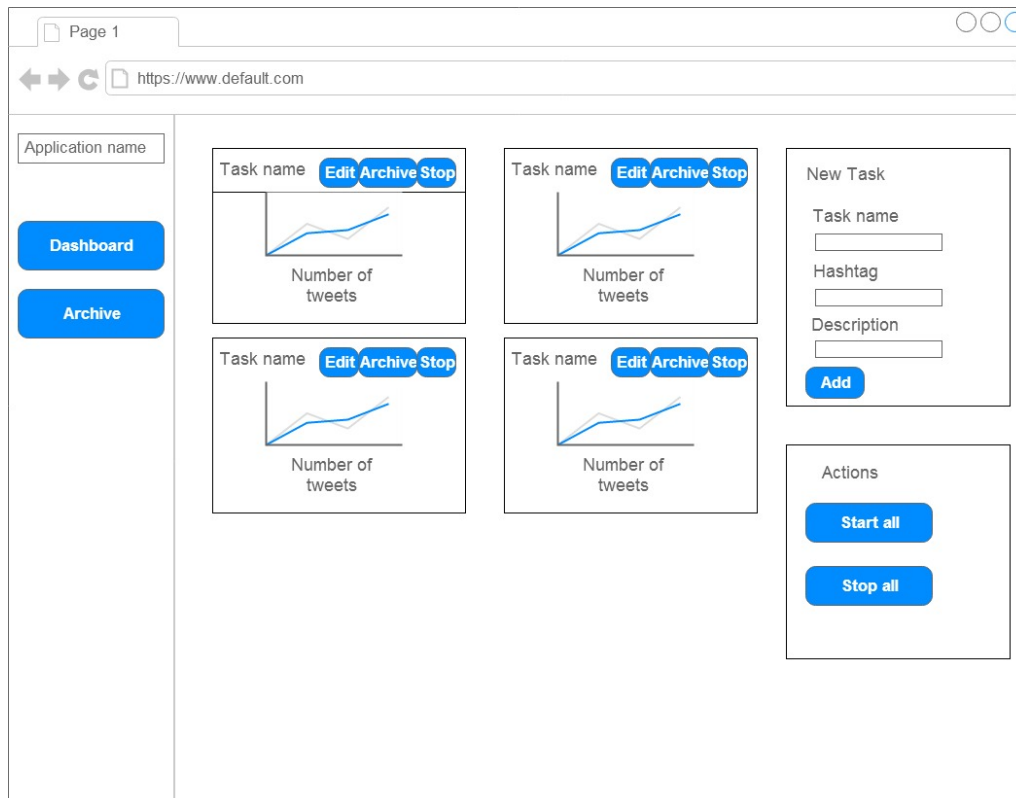


Figura 8: Mockup de la vista principal

En la **vista estructural** podemos cambiar de vista a partir del apartado View y para realizar los análisis de esta vista se pueden filtrar los mensajes seleccionados cambiando las fechas en las que tener en cuenta el análisis, a través de los calendarios disponibles o del slider. También se puede cambiar si se desea hacer el análisis en modo agregado o no. Se puede visualizar el grafo del conjunto de los mensajes en el rango de días escogido o mostrar la evolución de las diferentes métricas de redes complejas y los 10 usuarios más relevantes dependiendo de la métrica utilizada.

En la **vista geolocalizada y de análisis de sentimientos** podemos especificar la fecha de igual forma que en la vista anterior y elegir si queremos incluir los mensajes que incluyan el atributo places. Un mensaje no geolocalizado puede contener el atributo places, que representa la ciudad o país de origen del mensaje, por lo que incluyendo los mensajes con este atributo somos capaces de tener un mayor número de mensajes a tener en cuenta en el análisis geolocalizado. En el caso de los mensajes con geolocalización, se utiliza el atributo coordinates de estos. El combobox incluye la opción de no usar un geoJson (en cuyo caso se debe seleccionar una región en el mapa a la hora de realizar el análisis), y los nombres de los geoJson que se han subido a la herramienta, en ese caso se realiza el análisis de los mensajes contenidos en la región definida en el fichero.

Al realizar el análisis también se muestra la evolución del total de los mensajes en el período de tiempo especificado y los tag clouds de los sentimientos positivos y negativos. Los geoJson se añaden a la aplicación mediante la selección de un fichero del sistema del usuario, comprobando si es correcto.

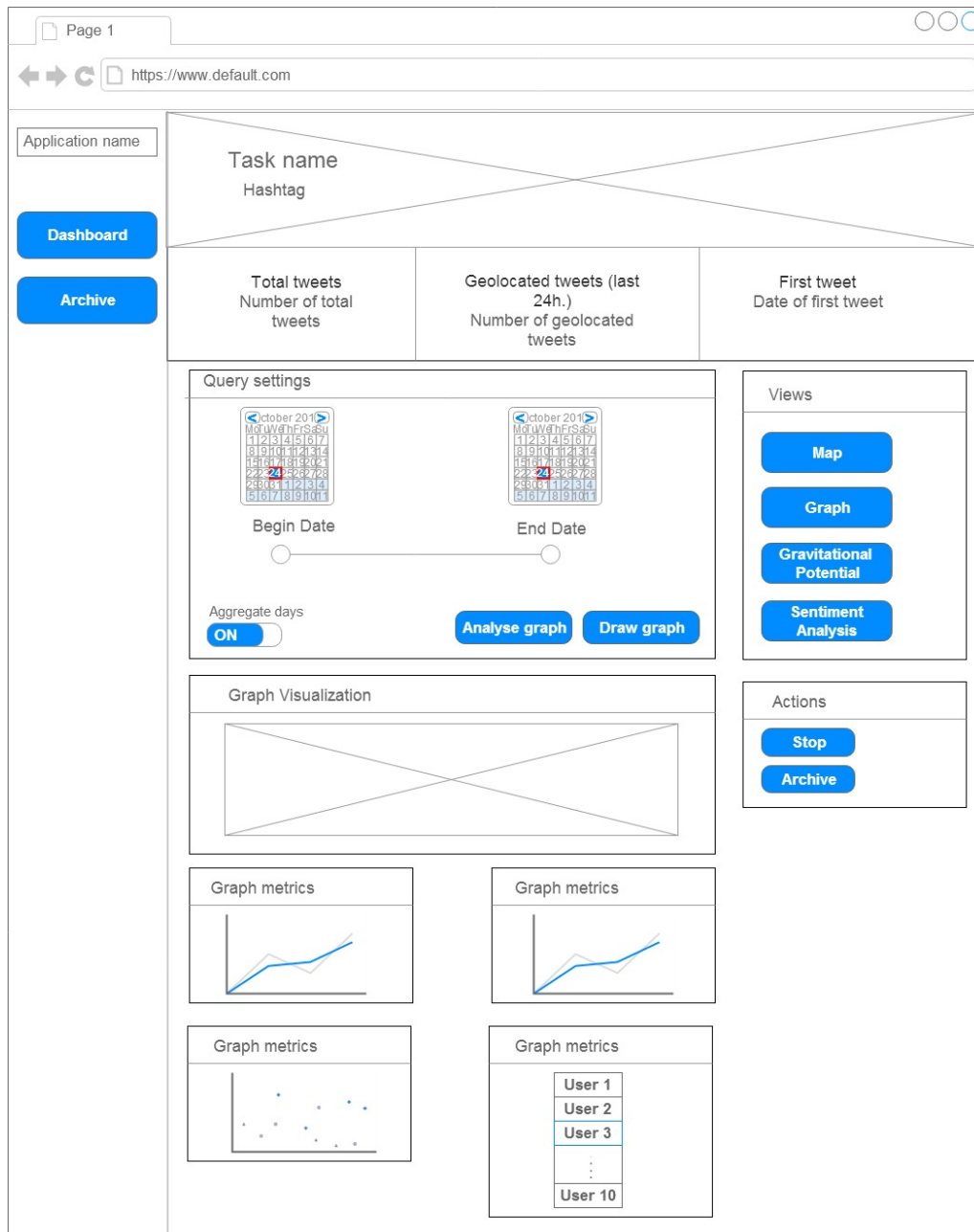


Figura 9: Mockup de la vista estructural

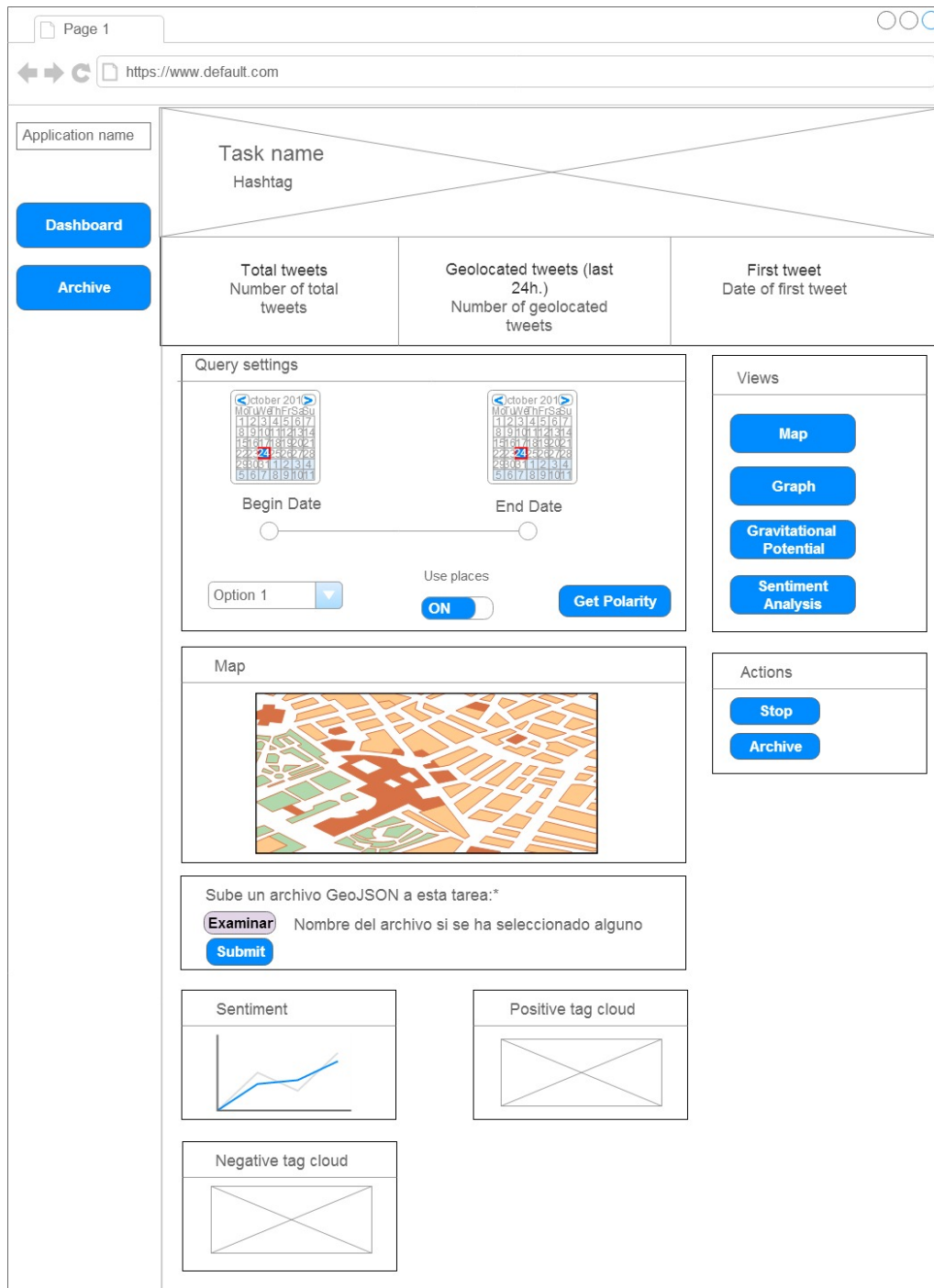


Figura 10: Mockup de la vista geolocalizada y de análisis de sentimientos

---

## Desarrollo

---

En esta sección vamos a presentar con más detalles las tecnologías y herramientas usadas para el desarrollo de la aplicación mencionada en este TFG. Además, se mostrará la funcionalidad y la visualización que presenta la herramienta ya implementada desde el punto de vista del usuario.

Como se explicó anteriormente, un usuario es capaz de acceder a la aplicación desde cualquier navegador web a su disposición, de forma que el usuario puede acceder a la herramienta de una forma sencilla sin necesidad de instalar nada. Junto a Python, se ha elegido utilizar el framework de desarrollo web Django, que es uno de los más populares para Python, para que el proceso de desarrollo fuera más sencillo.

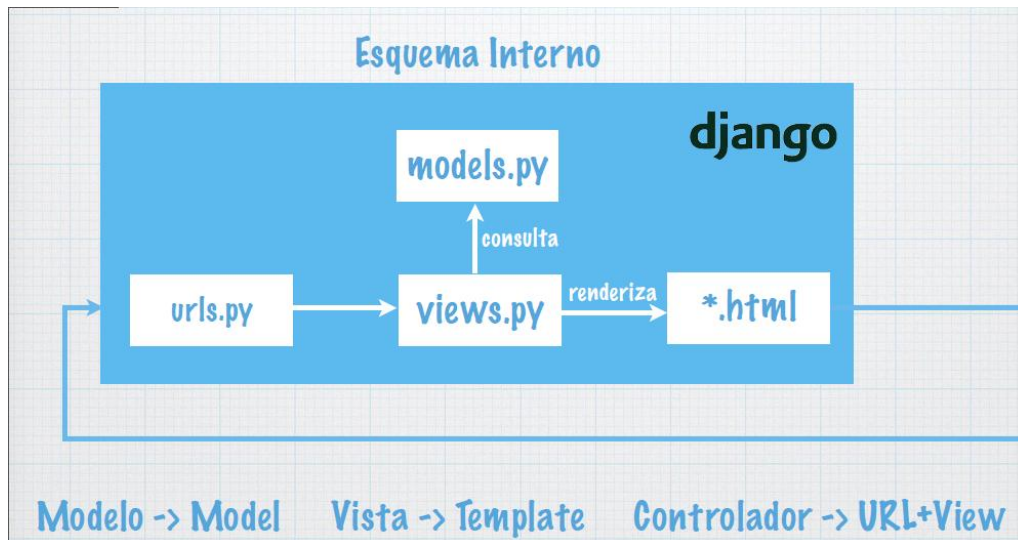
Se ha elegido Python como lenguaje de programación; ya que se puede emplear para implementar aplicaciones web, es un lenguaje popular para el que existen multitud de paquetes que permiten implementar más fácilmente elementos específicos y es un lenguaje sencillo de aprender, pudiendo utilizarse en múltiples plataformas.

Django [25] es un framework para aplicaciones web gratuito y open source, escrito en Python. Es un conjunto de componentes que te ayudan a desarrollar sitios web más fácil y rápidamente. Tal como explicamos en el punto anterior (Capítulo 3.2), Django utiliza un patrón de diseño de tipo *Model-View-Controller* (MVC). El patrón de diseño MVC [26] permite separar los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Como ya se indica en su nombre, está separado en tres componentes: *Model*, *View* y *Controller*.

El patrón de arquitectura de software MVC se basa en las ideas de reutilización de código y la separación de conceptos, buscando facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento. Además, al permitir dividir la lógica de negocio del diseño, el proyecto es más escalable y existen numerosas clases dentro de Django que nos facilitan la implementación, ya que incluyen numerosas funciones (como controlar fechas, URLs, respuestas para el cliente, etc.). Debido a todas estas ventajas se ha decidido utilizar el patrón MVC.

### Implementación del modelo MVC

Cuando se realiza una petición al servidor web, ésta se pasa a la parte *Controller*, formada por los archivos *urls.py* y *views.py*, de Django; este utiliza el *urlresolver* para encontrar un patrón que coincida con la URL. Una vez encontrado un patrón que coincide, se pasa la solicitud a la función del *views.py* que corresponda, a partir del cual se puede consultar al *Model* para acceder a la base de datos y obtener o guardar información. Luego, se elige el template (archivo *\*.html*) que corresponda del *View*, actualizándose con la información proporcionada por el *Model* para ser mostrada y se genera la respuesta que Django envía al navegador del usuario. Todo esto se refleja en la Fig. 11.



**Figura 11:** Arquitectura interna de los componentes del patrón de diseño MVC

Los componentes del MVC se pueden definir de la siguiente forma [26]:

*Model:* representa la información con la que opera el sistema y gestiona todos los accesos a la información que contiene, tanto consultas como actualizaciones. Se utilizan dos bases de datos: MongoDB y PostgreSQL. MongoDB contiene los mensajes recogidos asociados a una tarea. Los mensajes se almacenan en colecciones. Existe una colección por cada tarea. A la hora de realizar una consulta en la base de datos, se importa el *model* del que se quiere conseguir información y se utiliza el método *find()* para obtener los datos necesarios (normalmente se obtiene un conjunto de mensajes). Elegimos usar MongoDB para almacenar sólo los mensajes, ya que este tipo de base de datos no relacional es mucho más rápida al almacenar o acceder a mucha información en poco tiempo y la cantidad de mensajes recogidos de Twitter es muy grande; es una base de datos perfecta para usar en *Big Data*.

En cambio, si queremos consultar PostgreSQL utilizamos la operación *get()*, especificando un atributo para obtener el objeto deseado (normalmente se obtiene un sólo objeto), una vez se ha importado el modelo correspondiente. PostgreSQL contiene las tareas de las cuales se debe recoger mensajes junto con los datos que la define, los trabajos que se deben ejecutar en Celery, los grafos diarios y los archivos geoJson subidos por el usuario.

La razón para elegir PostgreSQL para almacenar los modelos definidos en los archivos *models.py* se debe a que PostgreSQL crea bases de datos robustas y eficientes. Además, no se requiere usar bloqueos de lectura al realizar transacciones, se pueden hacer tareas de mantenimiento o recuperación sin bloquear el sistema y es compatible con Django, necesitando sólo definir las tablas y relaciones en *models.py*.

*Controller*: Responde a los eventos (normalmente acciones del usuario) e invoca las peticiones a resolver. Hace de intermediario entre el *View* y el *Model*. Para implementar los archivos *urls.py* necesarios se sigue el esquema descrito en la documentación de Django [27], añadiendo tantos patrones de *URL* como sean necesarios y asociando cada uno de estos a una función del *views.py*. En los *views.py* se realizan las operaciones necesarias para satisfacer la petición realizada, tanto consultando al *Model* como procesando los datos obtenidos, devolviendo al final la respuesta.

Las acciones del usuario que implican la ejecución de tareas más pesadas son gestionadas a través de *websockets*. Celery recibe estas tareas a través de un *broker* RabbitMQ [28]. Celery es una cola de tareas asíncrona basada en el paso de mensajes distribuidos. Las tareas se ejecutan simultáneamente en uno o más *CeleryWorkers*, normalmente devolviendo la respuesta directamente al *Client*. Para integrar Celery con Django, se usa la extensión Channels [29].

*View*: Presenta la información del *Model* en un formato adecuado para facilitar la interacción con el usuario (usualmente la interfaz de usuario). Esta formado por cada uno de los archivos HTML necesarios para mostrar cada una de las vistas. Se utilizan hojas de estilo CSS para dar un estilo común a todos los HTML. Las acciones del usuario generan peticiones. Estas peticiones disparan eventos que se manejan con AngularJS, un *framework* de JavaScript de código abierto, en la parte *Client* de la arquitectura. El HTML contiene etiquetas específicas de AngularJS, por lo que cuando se realiza una acción, el JavaScript asociado al HTML actual se encarga de enviar la petición correcta al *Controller* o a *Channels*, dependiendo de si se debe completar la solicitud mediante Django (método *http.post(...)*) o mediante Celery (método *socket.send(...)*). Una vez se devuelve la respuesta, AngularJS se encarga de actualizar los nuevos datos en los elementos del *View*.

## Librerías y recursos

A continuación se van a presentar los paquetes de Python más relevantes que se han utilizado con la finalidad de implementar las funcionalidades deseadas.

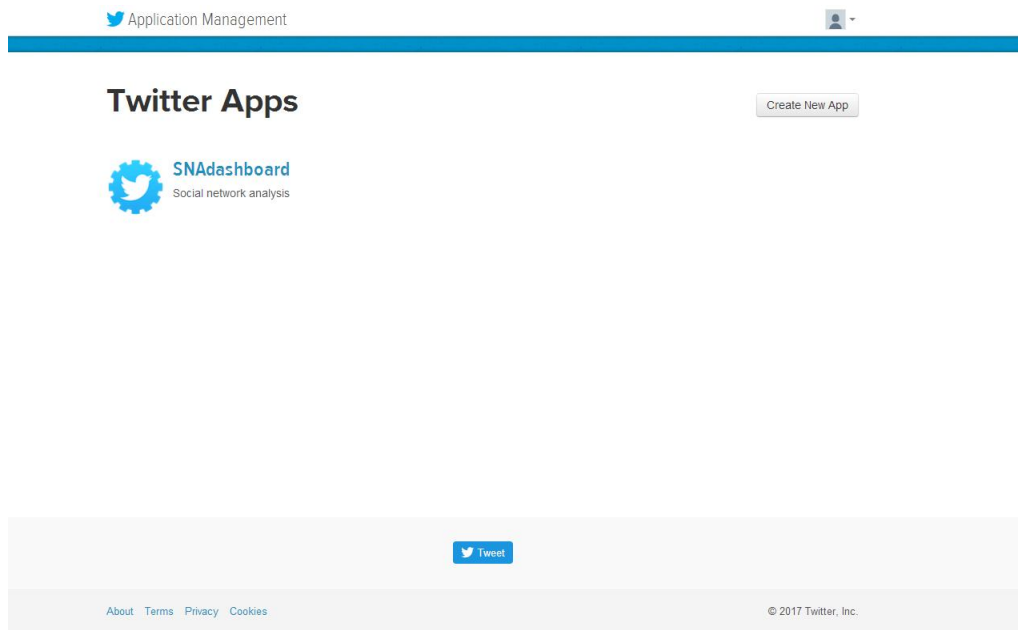
A la hora de crear y analizar un grafo, usamos NetworkX [30], un paquete creado específicamente para la creación, manipulación y estudio de la estructura, dinámicas y funciones de redes complejas. Este paquete incluye muchos algoritmos de grafos estándar y medidas de análisis de redes. Además, permite cargar y guardar redes en varios formatos si es necesario. En nuestro caso, los nodos representan a los usuarios que han escrito mensajes en el tema a analizar y/o están involucrados en una interacción. Las aristas corresponden a una interacción entre usuarios (mención, respuesta o retweet), cuyo tipo de interacción puede especificarse por medio de un atributo en la arista. Por lo tanto, Networkx nos permite realizar los análisis que necesitamos de una forma fácil y con un código claro.

Para la visualización del grafo se utiliza el paquete Graphistry [31]. Graphistry es una biblioteca que permite cargar grandes grafos en el servidor en la nube de esta. Soporta grafos grandes con una visualización interactiva, en la que el usuario puede moverse por las zonas que más le interesen del grafo. Puede dibujar el grafo directamente desde el creado en NetworkX, sin necesidad de transformarlo.

A la hora de visualizar los resultados del análisis de las propiedades (métricas) del grafo, utilizamos Bokeh [32]. Esta biblioteca nos permite visualizar de forma interactiva las gráficas en las que se muestra la evolución de las métricas. Se integra fácilmente en cualquier navegador web.

Para realizar las nubes de palabras se utiliza el paquete WordCloud [33], que permite crearlas de una forma muy sencilla. Los mensajes con una polaridad neutral no se tienen en cuenta, y se crean una nube de palabras para los mensajes con sentimiento positivo y otra para los negativos. Para crearlas, una vez detectada la polaridad del mensaje, el texto se va encadenando con los textos de la misma polaridad. Una vez conseguida ambas cadenas, la positiva y la negativa, se realiza un filtrado de las palabras contenidas en estas; se eliminan las *stopwords*, las URLs, las menciones, posibles etiquetas HTML, los *hashtags* y los *retweet*. Una vez que el filtrado se ha realizado, la biblioteca crea las nubes de palabras con las palabras más relevantes.

Para saber que polaridad y valor que tiene un mensaje, se usa la biblioteca TextBlob [34]. TextBlob es una librería para procesar textos. Permite realizar tareas de procesamiento de lenguaje natural, entre ellas el análisis de sentimientos. Una vez una cadena es pasada a TextBlob, el objeto que se crea tiene una propiedad llamada *sentiment*, cuyo valor corresponde un valor del sentimiento entre los valores [-1.0, 1.0]. Un valor de 0.0 representa un mensaje neutral, los valores de ]0.0,1.0] representan una polaridad positiva, cuyo grado más positivo se corresponde a un valor de 1.0, y los valores entre [-1.0, 0.0[ representan una polaridad negativa, cuyo grado más negativo se corresponde a un valor de -1.0. El único impedimento es que TextBlob sólo obtiene un valor para el sentimiento si el texto está en inglés, por lo que mensajes en otros idiomas siempre tienen un sentimiento neutral (valor 0.0 de la propiedad *sentiment*). Las herramientas de traducción para Python suelen tener una limitación en el uso de la misma, por lo que dado el gran volumen de datos manejados se ha decidido no traducir los mensajes. En la versión actual, la herramienta sólo realiza análisis de sentimientos de mensajes en inglés.



**Figura 12:** Registro de aplicación en Twitter

El paquete utilizado para recoger los *tweets* de las tareas introducidas en la aplicación se llama Twython [35], que obtiene los datos de la forma especificada en la API de Twitter. Lo que más nos interesa de la API de Twitter es la estructura interna de un *tweet* [36]. Nos muestra qué información está contenida en este, qué nombre tiene cada uno de los campos y qué tipo de valor pueden tener. De los mensajes recogidos se pueden obtener, entre otros, los siguientes campos y su valor correspondiente: la fecha de publicación de estos, el usuario que lo publicó, si el mensaje es producto de alguna interacción o las coordenadas en caso de ser un mensaje geolocalizado.

Para poder usar la funcionalidad de Twython primero se debe registrar una nueva aplicación de Twitter [Fig. 12], se deben rellenar los datos básicos de la aplicación y se generarán los valores '*Consumer Key*' y '*Consumer Secret*' necesarios para realizar las consultas y obtener la información de la API de Twitter. Es muy importante mantener las claves generadas en secreto para evitar suplantaciones de identidad. En la API de Twitter se encuentra la documentación y las referencias necesarias para poder extraer la información deseada.



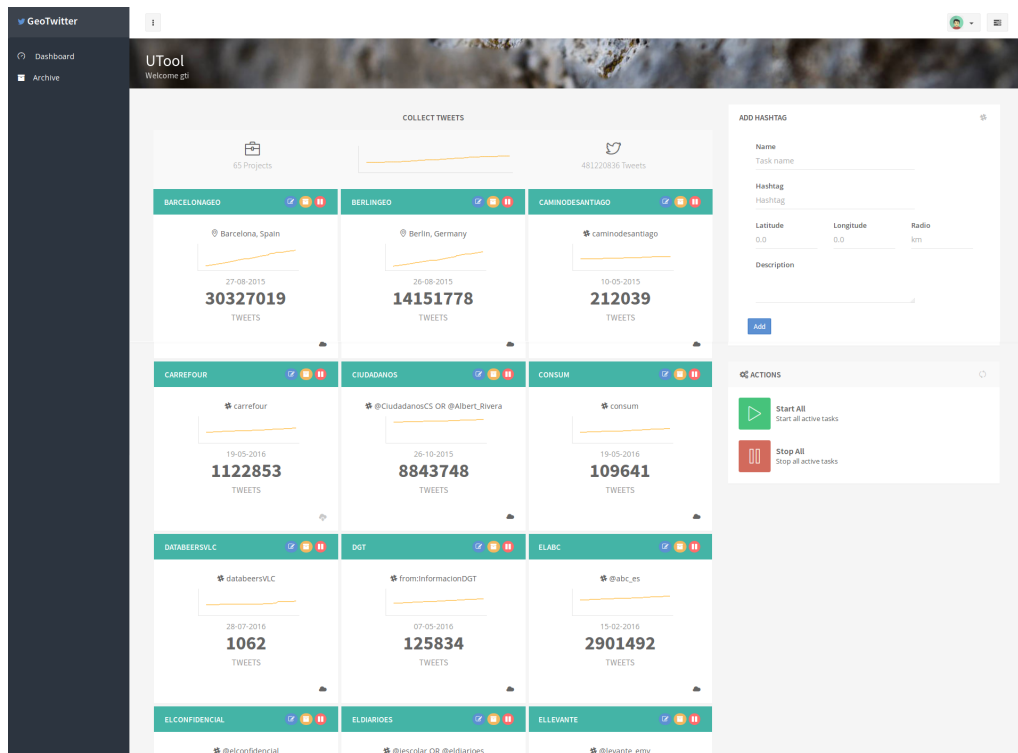


Figura 13: Vista principal

## Funcionalidad y vistas

Por último, se mostrará la funcionalidad de la herramienta y las diferentes vistas existentes.

En la Fig. 13 observamos la vista principal de la herramienta con una serie de tareas introducidas, en las cuales se recogen mensajes de manera automática en tiempo real.

Se han añadido 3 campos a la hora de añadir una nueva tarea (latitud, longitud y radio), por si se desea especificar una región del mundo para que sólo se recojan mensajes de esa zona.

En cada tarea se muestran las opciones de filtrado, el número total de mensajes, una gráfica con la evolución del total de mensajes recogidos y la fecha de inicio de recogida de mensajes.

En la Fig. 14 se muestra la primera captura de la vista estructural, donde se observan las opciones de selección de rango de fechas, junto con los botones para iniciar la ejecución de las funciones disponibles en esta vista (i.e., analizar grafo y visualizar grafo). Se ha elegido el *hashtag* 'Trump' y el rango entre el 1 de junio de 2017 y el 20 del mismo mes.

Una vez se han completado las operaciones, se muestra un grafo interactivo de la red generada, donde el paquete utilizado para mostrarlo incluye por defecto varias herramientas, como realizar consultas, hacer zoom, detectar comunidades, etc.

En la Fig. 15 se muestran 4 de las gráficas que muestran la evolución de las métricas; no se han incluido todas las gráficas en la muestra de la herramienta para permitir una mejor visualización de las imágenes. En los casos de uso que se expondrán en el siguiente apartado, se mostrarán los resultados de todas ellas. Las métricas tenidas en cuenta para analizar su evolución son: diámetro (*diameter*, representa la mayor distancia de entre todos los pares de vértices), coeficiente de agrupamiento (*clustering*, representa la media del grado de interconexión de los nodos con sus vecinos), modularidad (*modularity*, representa el grado de la división de una red en comunidades), densidad (*density*, representa el máximo número posible de aristas en el grafo), longitud media de camino (*average path length*, representa el número medio de pasos entre cada par de nodos), enlaces simétricos (*symmetric links*, representa el número de pares de nodos conectados entre sí en ambas direcciones), asortatividad (*assortativity*, representa el grado en el que nodos parecidos se conectan entre ellos) y distribución de grado (*degree distribution*, representa, para cada grado  $k$ , el número de nodos del grafo con grado  $k$ ). El paquete utilizado permite guardar la gráfica creada, hacer zoom, actualizar la gráfica, moverla, etc.

En la Fig. 16 se muestran 2 de las clasificaciones de los usuarios más influyentes en la red social utilizando medidas de centralidad, debido a la misma razón que en el caso anterior. En cada una de las métricas tenidas en cuenta se muestran los 10 usuarios con más valor en estas. Las métricas incluidas son: intermediación (*betweenness*, representa el número de caminos más cortos que pasan por cada nodo, es decir, el número de veces que el nodo actúa de puente a lo largo del camino más corto entre dos nodos), cercanía (*closeness*, representa la media de las distancias más cortas de un nodo hasta los demás), grado de entrada (*in-degree*, representa el número de enlaces de entrada de cada nodo), grado de salida (*out-degree*, representa el número de enlaces de salida de cada nodo) y *pagerank* (se computa un ranking de los nodos del grafo basándose en la estructura de los enlaces entrantes).

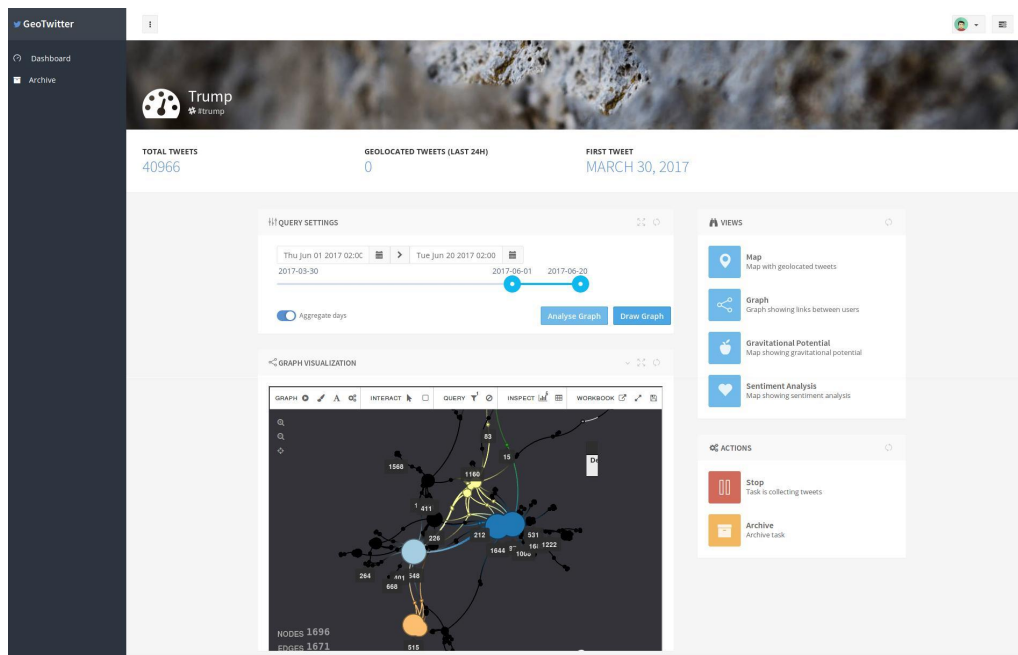


Figura 14: Vista estructural 1

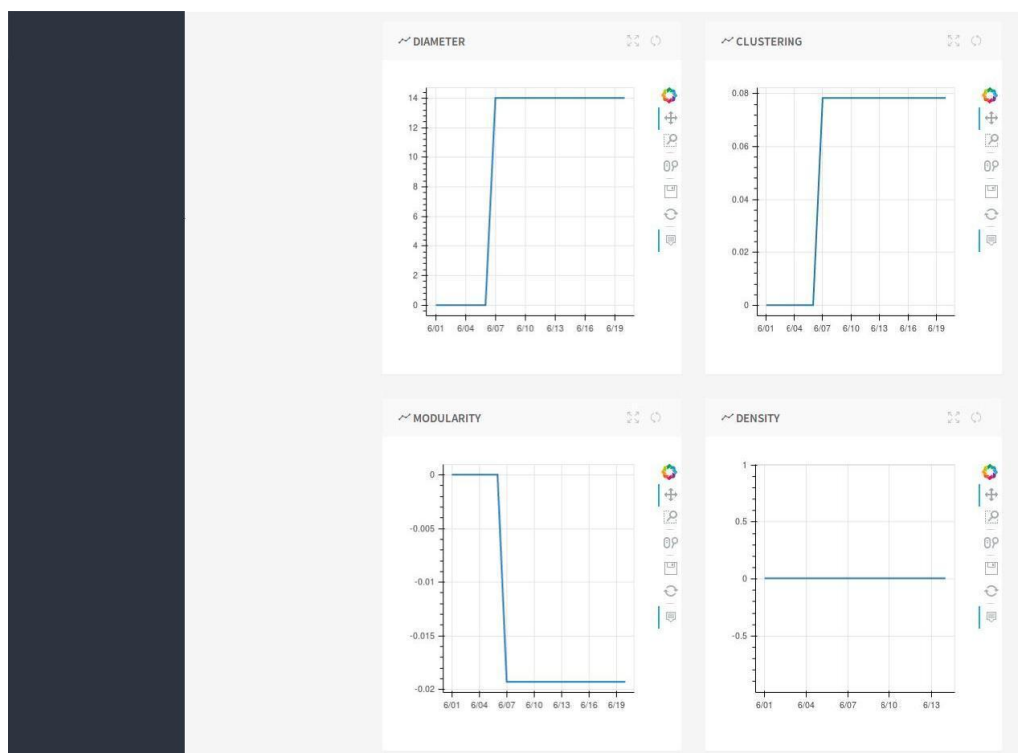
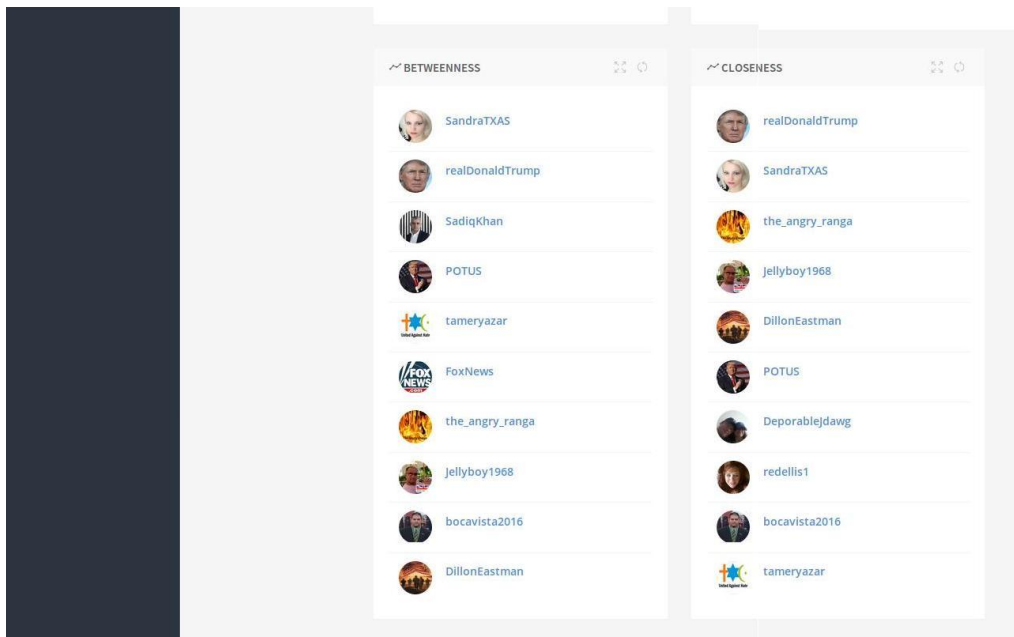


Figura 15: Vista estructural 2



**Figura 16:** Vista estructural 3

En la Fig. 17, se muestra la vista geolocalizada y de análisis de sentimientos. En esta vista se muestra el mismo caso de uso que en la vista anterior y en el mismo rango de fechas. En esta vista, el usuario tiene la opción de calcular el sentimiento medio especificando la zona mediante un fichero geoJson o seleccionando una región en el mapa. Para ilustrar el primer caso, se ha subido un fichero geoJson que contiene información de las zonas de Estados Unidos. Si no aparece color en una región, significa que no existen mensajes geolocalizados que estén dentro de esa zona en el rango de fechas especificado. Si existe color, éste puede ser rojo, verde o gris (sentimiento negativo, positivo o neutral, respectivamente), donde un color más intenso significa un mayor valor de la media del sentimiento y un color más claro significa un sentimiento poco negativo o positivo, dependiendo del color. Los valores de sentimiento neutrales no tienen variaciones en la intensidad del color.

En la Fig. 18 se muestra un ejemplo del segundo caso. Concretamente, se muestra la captura de una tarea de Valencia, donde a la hora de realizar el análisis de sentimientos geolocalizados se ha elegido la opción 'No GeoJson'; en tal caso, se selecciona una región en el mapa y cada mensaje detectado se muestra como un *pin*, donde el color muestra la polaridad del sentimiento (positivo, negativo o neutral).

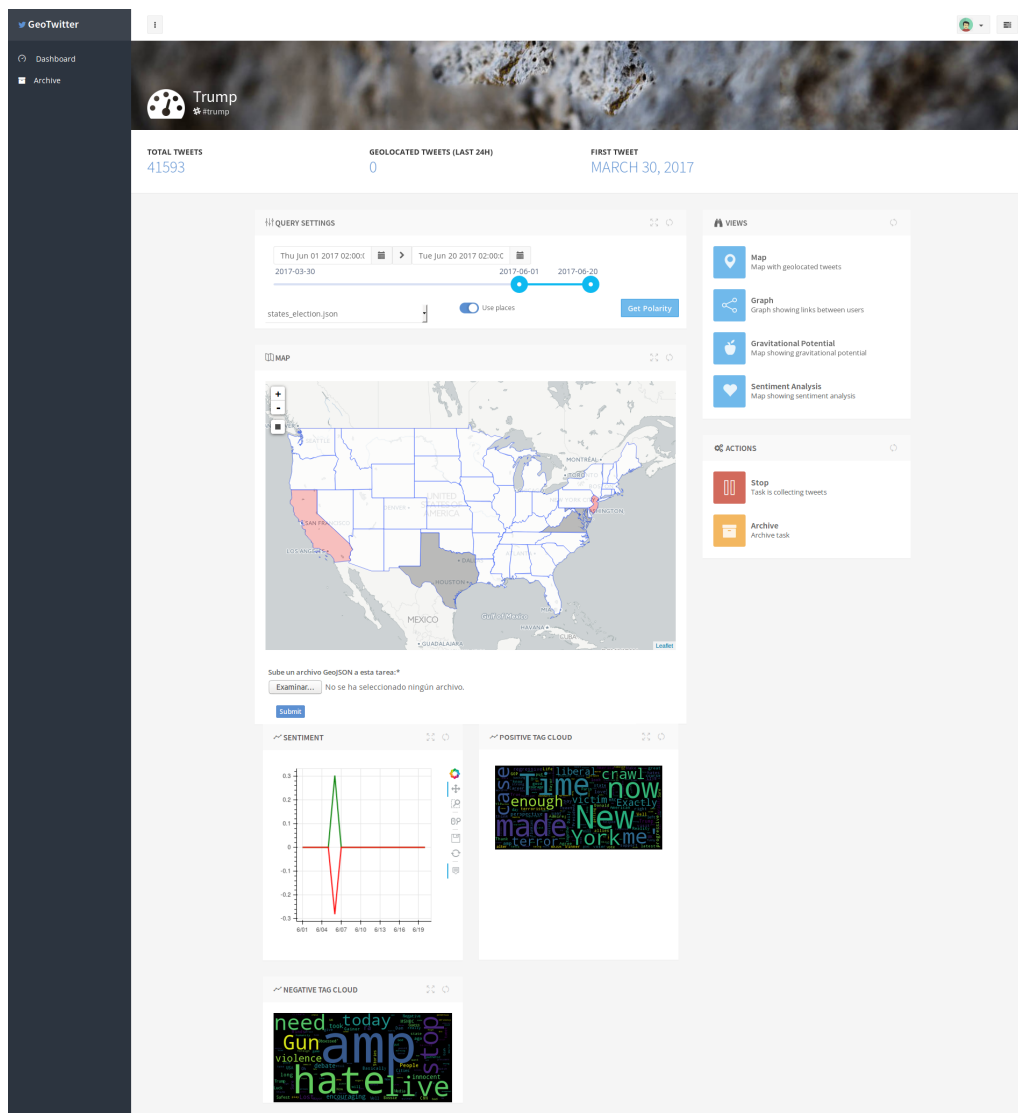


Figura 17: Vista geolocalizada y de análisis de sentimientos 1

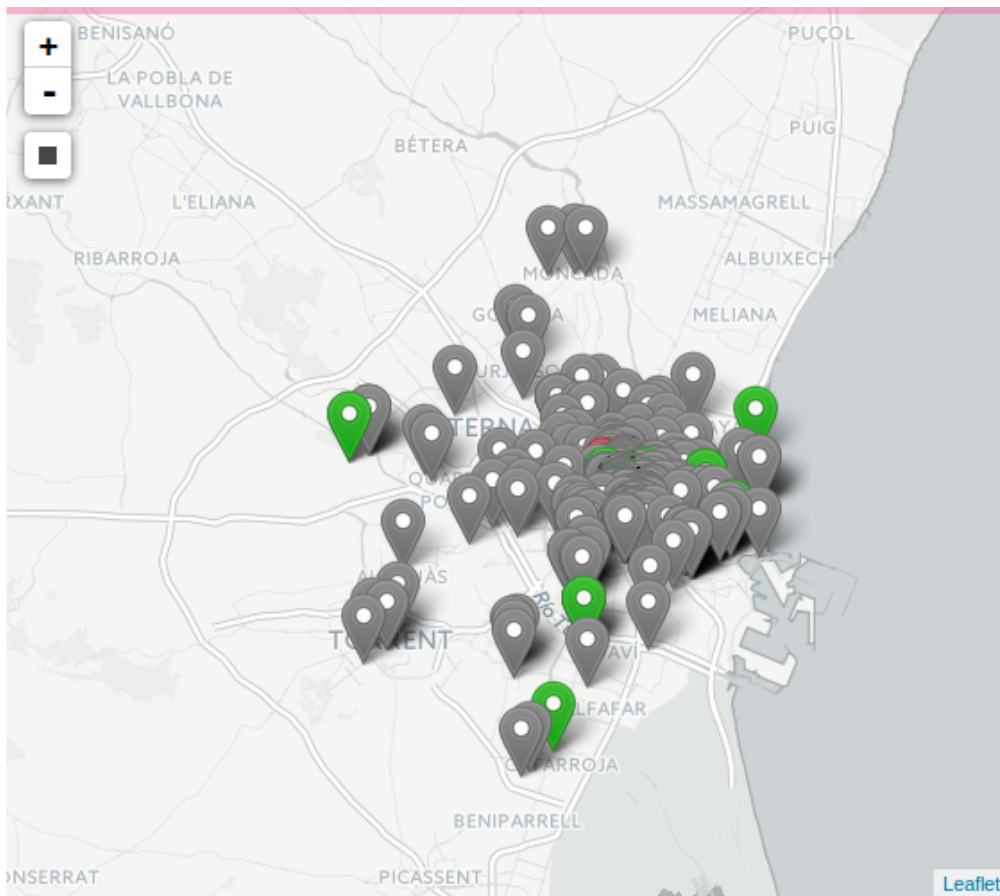


Figura 18: Vista geolocalizada y de análisis de sentimientos 2

---

---

## CAPÍTULO 4

# Resultados y discusión

---

En esta sección se presentan dos casos de prueba que se han realizado con dos de las tareas (*trump* y *londonbus*) disponibles en la herramienta desarrollada en unas fechas determinadas, con el fin de observar su funcionalidad con casos reales. Se comentarán las métricas que se consideren que presentan resultados interesantes.

### *Análisis hashtag trump*

---

Primero, se ha realizado un análisis de los mensajes que contienen el *hashtag trump* entre el 29 de abril de 2017 y el 29 de junio de 2017. Como se puede observar en la Fig. 19, el valor del diámetro aumenta mucho al principio de la creación de la red, indicando que aumenta la mayor distancia entre cada par de nodos rápidamente, sin embargo, luego se estabiliza, manteniéndose un valor constante, y finalmente aumenta un poco más; ya que el diámetro no disminuye en ningún momento, quiere decir que hay nodos muy lejanos entre sí que nunca llegan a encontrar un camino entre ellos más corto a través de otro. Se observa algo similar en la longitud media de camino, debido a la misma razón. La densidad también nos indica que se van creando más nodos (usuarios que participan publicando *tweets* con el *hashtag* del evento) que aristas (interacciones entre usuarios a través de mensajes que contienen *retweets*, *reply-to*, o menciones) entre ellos. Sin embargo, el coeficiente de agrupamiento y los enlaces simétricos nos indican que la interacción con nodos vecinos va aumentando bastante según pasan los días. La asortatividad indica que las conexiones se producen entre nodos con distinto grado. Todos los días tienen una distribución de grado muy similar, habiendo muchos nodos con poco grado y pocos nodos con un grado elevado. La gráfica de la evolución del sentimiento nos muestra que muchos de los días el sentimiento se ha mantenido neutral, con picos en algunos días, pero siempre el valor positivo es algo más alto que el negativo.



Figura 19: Métricas del *hashtag trump* y evolución del sentimiento.



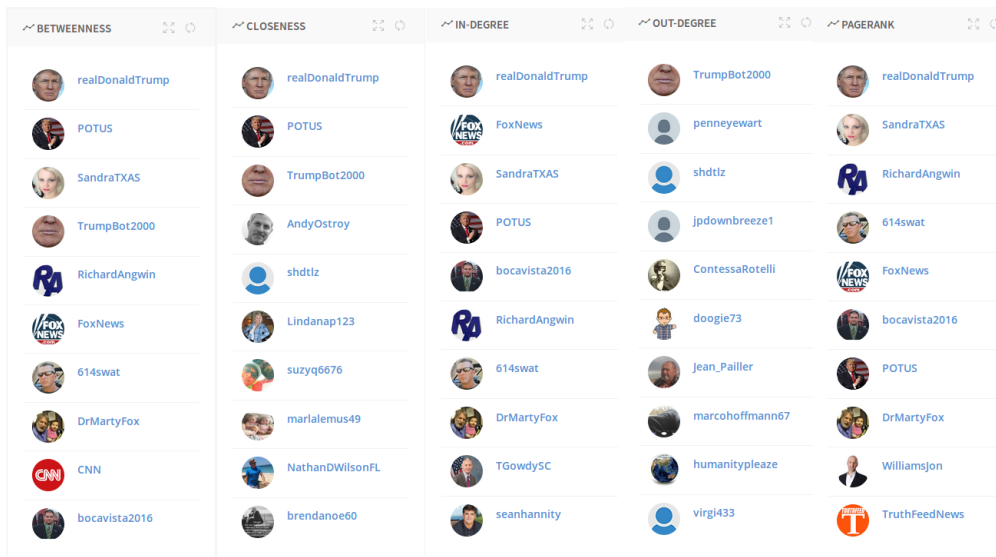


Figura 20: Rankings del *hashtag trump*.

En los rankings sobre la relevancia de los usuarios teniendo en cuenta métricas de centralidad (ver Fig. 20) se observa que hay tres usuarios que aparecen en los primeros puestos excepto en el de grado de salida, lo que quiere decir que son usuarios muy influyentes, pero no realizan interacciones con otros usuarios siendo ellos los emisores. Estos usuarios son *realDonaldTrump*, *POTUS* y *SandraTXAS*. Tiene mucho sentido la aparición de los dos primeros usuarios mencionados en los primeros puestos de varios rankings, ya que pertenecen a cuentas muy relacionadas con el *hashtag* a tratar. El usuario *TrumpBot2000* aparece como el usuario que más interacciones produce, apareciendo también en los rankings de intermediación y cercanía.

En la Fig. 21 puede observarse claramente, que aunque se ha realizado un análisis de dos meses completos, existen muchas zonas de Estados Unidos que no tienen ningún mensaje geolocalizado (zonas blancas). Esto se debe a que existen muy pocos mensajes geolocalizados comparado con el total de los mensajes, ya que no todos los usuarios habilitan la opción para ello. Se observa más actividad geolocalizada en la parte derecha del mapa, aunque las zonas en el extremo tienen una tonalidad muy clara, denotando un valor en el sentimiento muy bajo ya sea positivo o negativo. El valor más alto se encuentra en la parte sur, teniendo un sentimiento positivo de 0,7.

Las nubes muestran acerca de qué están hablando los usuarios que publican mensajes con sentimiento positivo o negativo sobre el total de los mensajes (ver Fig. 22). En el caso de los mensajes positivos, las palabras más utilizadas son: *elección*, *ganar*, *republicano* y *nuevo*. En el caso de los mensajes negativos, estas son: *medios de comunicación*, *demócrata*, *vídeo* y *trashing*.

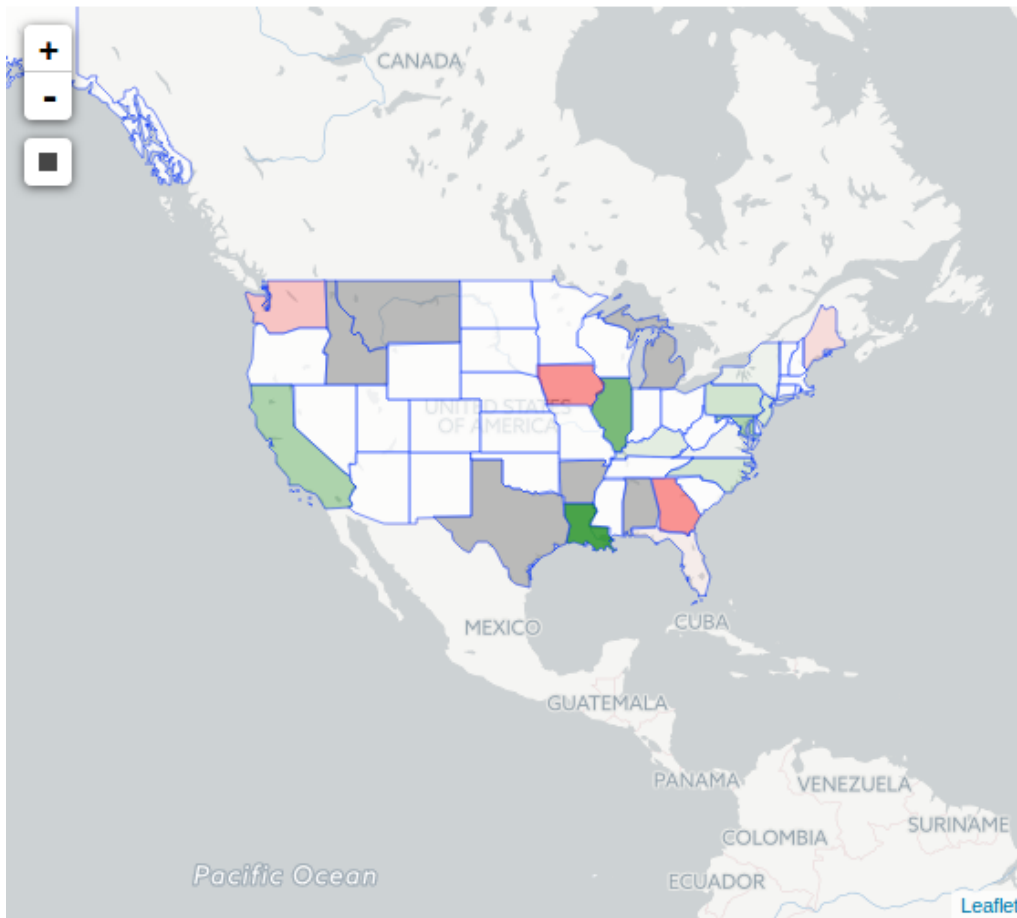


Figura 21: Mapa del sentimiento geolocalizado del *hashtag trump*.

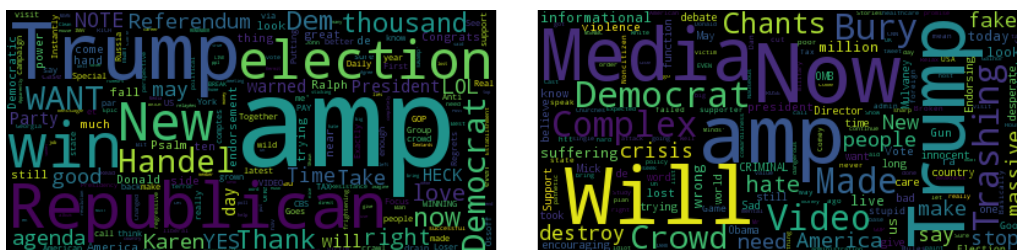


Figura 22: Nubes de palabras del *hashtag trump* (positiva a la izquierda, negativa a la derecha).

---

## Análisis *hashtag londonbus*

---

El segundo y último análisis se ha realizado de los mensajes que contienen el *hashtag londonbus* entre el 25 de junio de 2017 y el 29 de junio de 2017. Como se puede observar en la Fig. 23, el valor del diámetro aumenta también como en el caso anterior, ya que la red se vuelve más grande y los nodos se van distanciando al no haber interacciones entre los extremos. La modularidad nos indica que no existe una gran conectividad entre los nodos de una comunidad, algo que ya nos indicaba el diámetro. La densidad tiene un valor alto al principio, ya que la red tiene pocos nodos y pueden existir un menor número de aristas entre ellos; sin embargo, conforme aumenta la red, la densidad disminuye ya que aumenta más rápidamente el número de nodos que el de aristas. Cuando el valor de asortatividad es 0, quiere decir que la red es no asortativa (los nodos de grado similar no se conectan entre ellos), y cuando pasa a ser -1, quiere decir que la red es completamente no asortativa, pero en los dos últimos días vuelve a tener un valor de 0, por lo que en esos días deben de haberse conectado algún nodo similar entre sí. En la gráfica que muestra la evolución del sentimiento se observa que durante el rango analizado el sentimiento de los mensajes es muy positivo, llegando hasta el valor máximo (valor 1.0) el día 27 de junio.

En los rankings obtenidos (ver Fig. 24) se observa que hay cuatro usuarios que aparecen en los primeros puestos excepto en el de grado de salida, como en el caso anterior. Estos usuarios son *TfL*, *RadicalSnaps*, *liampdwyer* y *BigRonStevenson*. Los usuarios *TfL* y *TfLBusAlerts* son dos cuentas en Twitter que publican noticias acerca del bus de Londres, así que los demás usuarios interaccionan con esos usuarios pero no al contrario, por lo que no aparecen en el ranking de grado de salida.

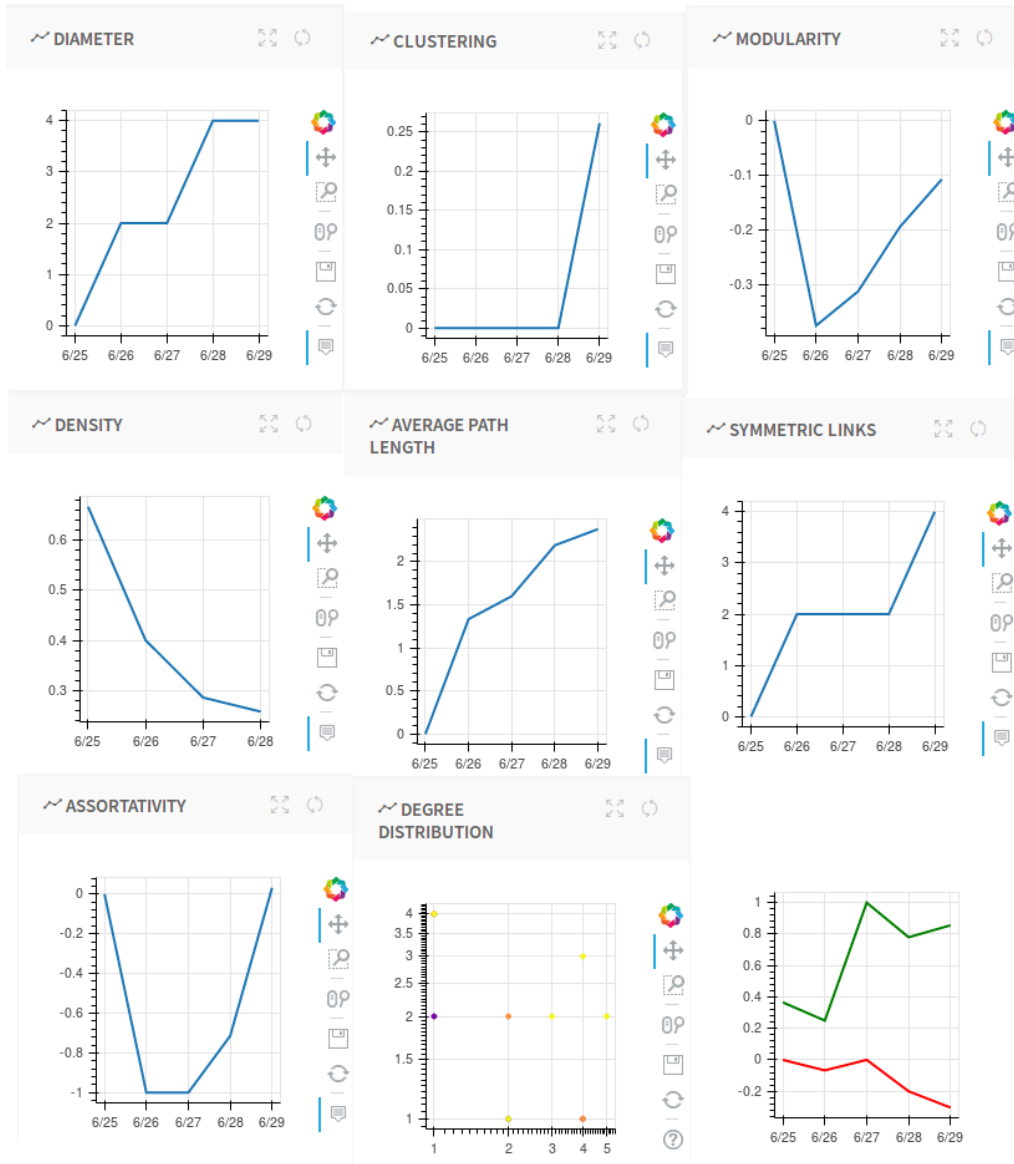


Figura 23: Métricas del *hashtag londonbus* y evolución del sentimiento.

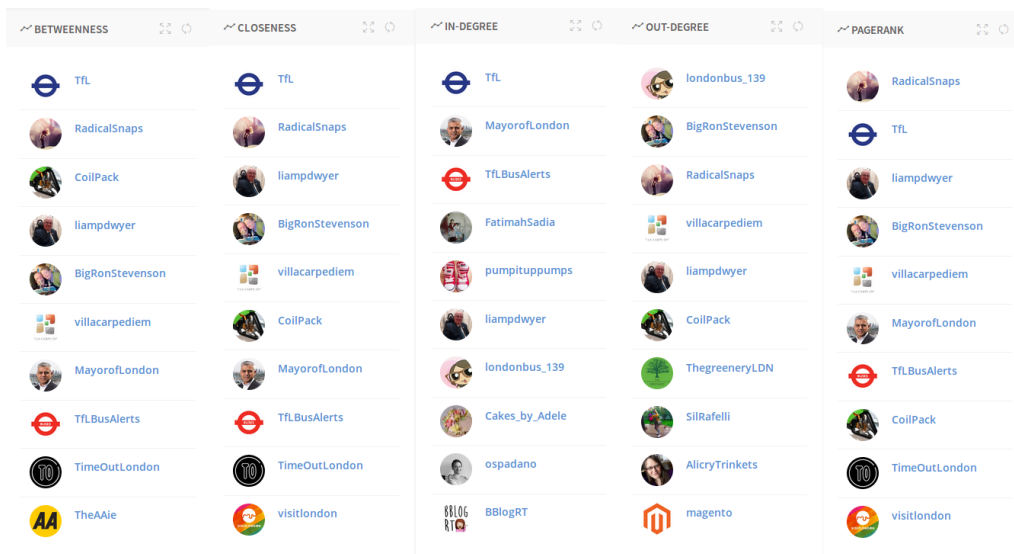


Figura 24: Rankings del *hashtag londonbus*.



**Figura 25:** Nubes de palabras del *hashtag londonbus* (positiva a la izquierda, negativa a la derecha).

Por último, analizamos las nubes de palabras de la tarea especificada (ver Fig. 25). En esta tarea, en el caso de los mensajes positivos, las palabras más utilizadas son: hermoso, parte y mejor. En el caso de los mensajes negativos, estas son: consejo, red, usuarios y bus.

A partir de los resultados obtenidos en la nube de palabras con sentimiento negativo, se detecta que parte del servicio público es lo que más molesta a los usuarios, por lo que mejorar esos aspectos mejoraría la opinión general de los usuarios.



---

---

## CAPÍTULO 5

# Conclusiones

---

Gracias a las tecnologías y métodos utilizados se han conseguido completar el objetivo principal y los objetivos secundarios planteados al principio de la memoria. Se ha implementado una herramienta web en la que se recogen de manera automática grandes cantidades de mensajes de redes sociales a partir de un hashtag, una geolocalización, o ambas cosas. Este análisis se puede realizar en un intervalo temporal especificado por el usuario y utiliza métricas de redes complejas y análisis de sentimientos. La visualización de la evolución de las métricas ha permitido entender la forma en la que se comunican los usuarios en una red social y de qué forma evoluciona.

Además, la herramienta desarrollada incluye el análisis de sentimientos en mensajes geolocalizados y no geolocalizados. En el caso de los mensajes geolocalizados, la herramienta permite comprobar si ciertas áreas tienden a tener una polaridad determinada en los mensajes publicados en redes sociales. Las nubes de palabras incluidas permiten detectar qué es lo que provoca un sentimiento positivo o negativo. Esto puede ser usado por empresas o instituciones que deseen mejorar la opinión general que se tiene sobre ellos al detectar qué es lo que deben cambiar o mejorar. La especificación de las fechas y forma de análisis permite entender mejor los cambios que se producen, ya que se visualizan los resultados de un subconjunto de mensajes y no del total de ellos.

En definitiva, en este proyecto se ha implementado una herramienta funcional y completa que permite analizar visualmente una red social y la forma en la que evoluciona. Se ha conseguido integrar los diferentes tipos de estudios que existen para analizar redes sociales en una sola herramienta, lo que ayudará al usuario a tener una visión global de qué está pasando y poder tomar decisiones de una manera más informada. El proyecto también aporta una de las muchas formas en las que se pueden desarrollar aplicaciones de este tipo, por lo que puede servir como referencia para la realización de un proyecto similar si se tiene pensado utilizar las tecnologías implementadas en este trabajo. El haber tenido que aprender nuevas tecnologías y métodos, como se ha explicado anteriormente, ha supuesto un gran enriquecimiento profesional.

---

## Valoración personal

---

Debido a la naturaleza del proyecto que se ha realizado, se han tenido que aprender a utilizar nuevas herramientas no vistas anteriormente en la carrera (como Django, la API de Twitter, las librerías específicas mencionadas en el capítulo 3.3, etc.). Aunque estas herramientas han facilitado en gran medida la implementación de la herramienta, se ha tenido que acceder a la documentación de estas y realizar consultas en Internet cuando algo no funcionaba como debía, aunque la curva de aprendizaje ha sido buena. El aprendizaje de nuevas herramientas y tecnologías puede entenderse como un objetivo adicional del proyecto realizado, ya que estos conocimientos servirán en mi futuro profesional a la hora de realizar un proyecto con una estructura similar. El rápido aprendizaje de estas técnicas ha sido posible gracias a los conocimientos previos y las prácticas mostrados a lo largo de la carrera; también se han utilizado conocimientos aprendidos anteriormente para llevar a cabo el proyecto (como los conocimientos acerca de bases de datos, Python, JavaScript, técnicas de programación, algoritmos, etc.).

A la mitad del proyecto se decidió cambiar la base de datos donde se guardan los mensajes recogidos MongoDB a Neo4j [37], ya que es una base de datos orientada a grafos y se pensó que se reduciría el tiempo de espera al ir creando los grafos para cada tarea según se fueran guardando mensajes en Neo4j, en vez de obtener los mensajes en el rango de tiempo especificado y tener que crear el grafo. Al modificar el sistema, se observó que, debido a las grandes cantidades de información almacenada, se tardaba más tiempo en obtener la parte ya creada del grafo en Neo4j, por lo que se volvió a cambiar la base de datos a MongoDB. Quizás este error podría haberse evitado buscando y encontrando un caso similar, en el que alguien hubiera usado Neo4j para manejar grandes cantidades de datos y explicase su experiencia de como se ralentiza en gran medida la obtención de datos.

Se planeaba añadir una métrica basada en *Eigenvector* para obtener otro ranking de los 10 usuarios con más valor en ella. La operación de la librería NetworkX para obtener el valor de *eigenvector* de cada nodo tiene un argumento del máximo número de iteraciones del método de potencia. Esta operación no tiene garantía de convergencia, por lo que en grafos grandes la operación suele fallar y se decidió eliminar la métrica de la herramienta.

Como conclusión personal, estoy contenta con el trabajo realizado, con el aprendizaje de nuevas técnicas y con el objetivo del TFG realizado, ya que entender mejor como evolucionan las relaciones humanas y los sentimientos en Internet me parece un tema muy interesante.



---

## Trabajo Futuro

---

Debido a la limitación de tiempo y a la naturaleza de los proyectos en informática, siempre existen cosas que se pueden añadir o modificar en un proyecto. A continuación se van a presentar varias ampliaciones que pueden resultar interesantes.

Se podría encontrar e implementar una solución alternativa para medir el sentimiento de los mensajes que están en otro idioma que no sea el inglés, con el fin de tener la posibilidad de analizar el sentimiento de un mayor rango de mensajes. Se encontraron bastantes librerías que no tenían límites y eran gratuitas, pero usaban el servicio de traducción gratuito de Google y desde que Google dejó de proveer este servicio, estas librerías ya no funcionan o tienen un límite gratuito de 1000 palabras al día (un límite demasiado bajo para la cantidad de datos manejados en la herramienta). Así que esto podría realizarse encontrando una librería en el futuro que ofrezca traducción sin límites gratuita o realizando un análisis del sentimiento para los idiomas más frecuentes de cada palabra de un mensaje para obtener la puntuación final del sentimiento.

El número de mensajes geolocalizados comparado con el total de los mensajes es muy escaso, por lo que sería interesante en el futuro, lograr clasificar más de los mensajes como geolocalizados a partir de algún atributo más, aparte del *places* ya integrado en la herramienta, que permitiese localizar más o menos de forma precisa más mensajes.

Por último, como ampliación, podría integrarse en el mapa de la vista geolocalizada y de análisis de sentimientos, la red formada por los mensajes que estén dentro de la zona seleccionada para analizar. Para esto, debería hacerse primero la ampliación anterior, ya que si el número de mensajes geolocalizados es muy bajo, difícilmente existirán interacciones entre los mensajes de la zona.

---

## Publicaciones del TFG

---

*Vivanco, E., Palanca, J., del Val, E., Rebollo, M., Botti, V. (2017, June). Using Geo-Tagged Sentiment to Better Understand Social Interactions. In International Conference on Practical Applications of Agents and Multi-Agent Systems (pp. 369-372). Springer, Cham.*



# Bibliografía

---

- [1] GOVINDARAJAN, M.; ROMINA, M. A Survey of Classification Methods and Applications for Sentiment Analysis. *The International Journal Of Engineering And Science (IJES)*, 2013, vol. 2, no 12, p. 11-15.
- [2] Wang, D., Zhu, S., Li, T. SumView: A Web-based engine for summarizing product reviews and customer opinions. *Expert Systems with Applications*, 2013, 40(1), 27-33.
- [3] Schumaker, R. P., Zhang, Y., Huang, C. N., Chen, H. Evaluating sentiment in financial news articles. *Decision Support Systems*, 2012, 53(3), 458-464.
- [4] Chamlerwat, W., Bhattarakosol, P., Rungkasiri, T., Haruechaiyasak, C. Discovering Consumer Insight from Twitter via Sentiment Analysis. *J. UCS*, 2012, 18(8), 973-992.
- [5] West, R., Paskov, H. S., Leskovec, J., Potts, C. Exploiting social network structure for person-to-person sentiment analysis. *arXiv preprint*, 2014, arXiv:1409.2450.
- [6] Mai, E., Hranac, R. Twitter interactions as a data source for transportation incidents. *In Proc. Transportation Research Board 92nd Ann.*, 2013, Meeting (No. 13-1636).
- [7] del Val, E., Rebollo, M., Botti, V. Does the type of event influence how user interactions evolve on Twitter? *PLOS one.*, 2015, 10(5), e0124049.
- [8] del Val, E., Martínez, C., Botti, V. Analyzing users' activity in online social networks over time through a multi-agent framework. *Soft Computing*, 2016, 20(11), 4331-4345.
- [9] del Val, E., Palanca, J., Rebollo, M. U-Tool: A Urban-Toolkit for Enhancing City Maps Through Citizens' Activity. *In Advances in Practical Applications of Scalable Multi-agent Systems. The PAAMS Collection*, 2016, (pp. 243-246). Springer International Publishing.
- [10] Caragea, C., Squicciarini, A., Stehle, S., Neppalli, K., Tapia, A. Mapping moods: geo-mapped sentiment analysis during hurricane Sandy. *ISCRAM*, 2014.
- [11] Hasan, S., Ukkusuri, S. V., Zhan, X. Understanding Social Influence in Activity Location Choice and Lifestyle Patterns Using Geolocation Data from Social Media. *Frontiers in ICT*, 2016, 3, 10.

- 
- [12] Charlton, N., Singleton, C., Greetham, D. V. In the mood: the dynamics of collective sentiments on Twitter. *Royal Society Open Science*, 2016, 3(6), 160162.
- [13] Thelwall, M., Buckley, K., Paltoglou, G. Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology*, 2012, 63(1), 163-173.
- [14] O'Sullivan, D. J., Garduño-Hernández, G., Gleeson, J. P., Beguerisse-Díaz, M. Integrating sentiment and social structure to determine preference alignments: The Irish Marriage Referendum. *arXiv preprint*, 2017, arXiv:1701.00289.
- [15] Kling, F., Pozdnoukhov, A. When a city tells a story: urban topic analysis. *In Proceedings of the 20th international conference on advances in geographic information systems*, 2012, (pp. 482-485). ACM.
- [16] Rebollo, M., del Val, E., Carrascosa, C., Palomares, A., Pedroche, F. Consensus over multiplex network to calculate user influence in social networks. *Int. J. Complex Systems in Science*, 2013, 3(1), 71-75.
- [17] Información acerca del formato geoJson y de los tipos de estructuras geográficas disponibles.  
Consultado en <https://es.wikipedia.org/wiki/GeoJSON>.
- [18] Documentación del framework de desarrollo web Django.  
Consultado en <https://www.djangoproject.com/>.
- [19] Documentación de la cola de tareas Celery.  
Consultado en <http://www.celeryproject.org/>.
- [20] Documentación del framework javaScript AngularJS.  
Consultado en <https://angularjs.org/>.
- [21] Documentación del servidor proxy Nginx.  
Consultado en <https://nginx.org/en/>.
- [22] Documentación de la base de datos MongoDB.  
Consultado en <https://www.mongodb.com/es>.
- [23] Documentación de la base de datos PostgreSQL.  
Consultado en <https://www.postgresql.org/>.
- [24] Documentación del broker RabbitMQ.  
Consultado en <https://www.rabbitmq.com/>.
- [25] Información acerca de Django, ¿Qué es Django?  
Consultado en <https://tutorial.djangogirls.org/es/django/>.
- [26] Información acerca del patrón de diseño Model-View-Controller.  
Consultado en <https://es.wikipedia.org/wiki/Modelo%2%80%93vista%2%80%93controlador>.

- 
- [27] Documentación de Django acerca de cómo implementar los archivos `urls.py`. Consultado en <https://docs.djangoproject.com/en/1.11/topics/http/urls/>.
- [28] Documentación del paquete específico para Python para usar el broker RabbitMQ. Consultado en <https://pyrabbit.readthedocs.io/en/latest/>.
- [29] Documentación de la extensión Channels para Django. Consultado en <https://channels.readthedocs.io/en/stable/>.
- [30] Documentación del paquete NetworkX para Python. Consultado en <https://networkx.github.io/>.
- [31] Documentación del paquete Graphistry para Python. Consultado en <https://github.com/graphistry/pygraphistry>.
- [32] Documentación del paquete Bokeh para Python. Consultado en <http://bokeh.pydata.org/en/latest/>.
- [33] Documentación del paquete WordCloud para Python. Consultado en [https://github.com/amueller/word\\_cloud](https://github.com/amueller/word_cloud).
- [34] Documentación del paquete TextBlob para Python. Consultado en <https://textblob.readthedocs.io/en/dev/>.
- [35] Documentación del paquete Twython para Python. Consultado en <https://twython.readthedocs.io/en/latest/>.
- [36] Documentación del API de Twitter. Consultado en <https://dev.twitter.com/overview/api/tweets>.
- [37] Documentación de la base de datos Neo4j. Consultado en <https://neo4j.com/>.

