



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escuela Técnica Superior de Ingeniería Informática
Universitat Politècnica de València

Extracción de información de Twitter para la creación de redes de interacciones entre usuarios

Proyecto Final de Carrera

Ingeniería Informática

Autor: Rubén Honrubia Pineda

Director: José Luis Poza Luján y Miguel Rebollo Pedruelo

27/09/2015

Tabla de contenidos

Introducción	7
Entorno	7
Objetivos	8
Descripción del documento	10
Estado del arte	12
Introducción	12
Sistemas similares	13
Twitter Analytics	14
Twitonomy	17
Tweetchup	19
http://tweetchup.com/es	19
Twitter counter	20
SocialBro	21
Tweet Archivist	22
Topsy	23
Tweetreach	24
Hootsuite	25
Análisis de los sistemas	26
Análisis subjetivo	27
Análisis subjetivo	29
Características del sistema a desarrollar	31
Métricas. Análisis en profundidad.	33
Conclusiones	34
Especificación de requisitos	35



Introducción	35
Propósito.....	35
Ámbito del sistema	35
Definiciones y acrónimos.....	35
Visión general del documento	36
Descripción general	37
Perspectiva del producto.....	37
Funcionalidad del producto	38
Características de los usuarios	40
Restricciones	41
Suposiciones y dependencias	41
Requisitos futuros	42
Requisitos específicos	43
Interfaces externas	43
Funciones	44
Requisitos de rendimiento	46
Restricciones de diseño	47
Atributos del sistema	47
Otros requisitos	47
Conclusiones	48
Diseño del sistema.....	49
Introducción	49
Especificación conceptual.....	49
Especificación formal.....	50
Persistencia	50
Negocio.....	53
Presentación.....	55
Dinámica de la aplicación	59
Conclusiones	60

Implementación e implantación	61
Introducción	61
Implementación.....	61
Tecnología Web.....	61
Datos	63
Proceso	65
Interfaz.....	68
Herramientas	70
Evaluación.....	71
Caja blanca	71
Caja negra.....	72
Implantación.....	73
Instalación.....	73
Pruebas.....	77
Conclusiones	78
Conclusiones.....	79
Dificultades	79
Aportaciones	80
Ampliaciones	81
Mejoras.....	81
Monetización.....	81
Ampliación de redes sociales a analizar	81
Referencias.....	82
Internet	82
Bibliografía.....	84

Introducción

Entorno

Hoy en día la información fluye de forma casi inmediata. Cualquier evento ocurrido en cualquier parte del globo aparece en cualquier otro punto de la tierra instantes después. La gente está más conectada que nunca. No existen distancias a la hora de comunicarte con alguien. Casi no existen barreras ni muros en las nuevas sociedades. El entorno cambia, la tecnología avanza y la sociedad con ella. Avanzamos, algunas veces lentos, otras veces sin rumbo fijo, con sorpresas por el camino que desaparecen casi al momento, pero seguimos avanzando.

Internet, en los últimos años, y gracias al auge de las redes sociales, ha hecho posible esta visión de la información. El acceso a ella en cualquier parte y en cualquier lugar es su principal baza. La socialización humana, el compartir lo que sientes, hacer que la gente que te rodea sepa que estás ahí, sin importar la distancia y en cualquier momento. Las redes sociales nos mantienen conectados siempre, a todas horas comentamos nuestros pensamientos, los compartimos, discutimos sobre ellos. Informamos sobre cosas que nos afectan de una forma inmediata, tanto buenas noticias como las no tan buenas.

Es cierto que acarrea problemas, el exceso de esta información provoca que estemos más insensibilizados con ella. El factor sorpresa ha dejado de tener peso. Todo es efímero. Por otra parte, la ausencia de privacidad es un factor muy a tener en cuenta. En cualquier momento se puede conseguir información tuya con simplemente una búsqueda. La lucha entre servicios y privacidad.

Siempre con modas pasajeras, contactos por internet que no has visto nunca. Se dice que hemos perdido parte de nuestra identidad social, que somos esclavos de la tecnología. Es posible.

Estarás a favor o en contra de la nueva sociedad de la información, pero ¿qué nos deparará el futuro? A esta pregunta no tenemos respuesta, pero seguramente seguirá cambiando, desafiante en algunos casos. Pero de lo que estoy seguro es que siempre seguiremos progresando.



Objetivos

Como las redes sociales son hoy en día una de las principales fuentes de difusión de información de los diferentes eventos que ocurren, nos centraremos en ellas para realizar un estudio de cómo se comportan los usuarios a determinados eventos.

La base que tomaremos para el estudio será interpretar las redes sociales como un conjunto de nodos y enlaces que se van generando a lo largo del tiempo. En particular, analizaremos Twitter por su facilidad a la hora de acceder a la información de los usuarios.

En el caso concreto de Twitter, vamos a considerar la red (grafo) que se genera a partir de las interacciones de los usuarios en un evento concreto que tiene asociado una etiqueta o “hashtag”¹. En un grafo asociado a un evento hay dos componentes: los nodos y los enlaces entre estos nodos.

- **Nodo:** Representa un usuario de la red que ha participado en el evento mediante la publicación de un tuit o “tweet”.
- **Enlace:** Representa una interacción entre usuarios, es decir, un usuario A (nodo) interactúa con otro usuario B cuando este último usuario es mencionado en el tweet del usuario A.

El objetivo principal del proyecto es, a partir de los nodos y de las interacciones asociadas a un evento, la creación de una red y analizar qué factores son los determinantes para su crecimiento, que nodos son relevantes en la expansión, quienes son receptores (los nodos que reciben más enlaces), en general, la centralidad de estos.

El estudio de redes complejas y con una gran cantidad de nodos y enlaces supone un gran desafío. En este proyecto se plantea el desarrollo de una herramienta visual, que ayude a la interpretación de las redes y lo que es más importante, ayude a alimentar la intuición del investigador. La información se mostrará de una forma visual, para que sea clara y sencilla.

En resumen, los objetivos del proyecto se podrían establecer en los siguientes pasos, todos ellos relevantes y de los que se podría hacer un proyecto de cada uno debido a su complejidad:

1. Recolección

Para poder realizar el estudio lo primero que haremos será la recolección de datos de redes sociales para poder trabajar con ellos. Tanto información que se ha transmitido en el pasado y que se ha almacenado en bases de datos, como información que se está compartiendo en el mismo instante.

¹ Un hashtag es una cadena de caracteres formada por una o varias palabras concatenadas y precedidas por una almohadilla. Sirve como etiqueta de metadatos con el fin de que tanto el sistema como el usuario la identifiquen de una forma rápida.

2. Almacenamiento

Es necesario almacenar esta información de una manera ordenada, con la estructura más adecuada para esta tipología de datos. Una vez almacenada, se tratará esta información realizando diversos análisis de centralidad que nos devolverán una serie de resultados relevantes.

3. Visualización

La última parte tendrá la funcionalidad de recabar todos los resultados almacenados y visualizar estos de una forma clara y directa al usuario, para que este pueda tener una idea general de lo más relevante de la red.

Todo el bloque aporta información suficiente como punto de partida para realizar estudios sobre los diferentes eventos que ocurren en la red. Estos resultados que servirían para la toma de decisiones a la hora de la transmisión de información.

Por último, cabe decir que en este proyecto nos centraremos, como se ha dicho antes, en la red social de Twitter, pero existen muchas otras redes sociales ampliamente utilizadas. Por esto se hará una comparativa, como añadido, con otro proyecto que trata también sobre redes sociales y que servirá como visión global de las pautas comunes que se han tomado para la realización de estos.

Descripción del documento

El documento se presentará de una forma sistemática donde se empezará teniendo una visión general del proyecto y que iremos ampliando para poder ver las diferentes partes en las que se compone. Por lo tanto, tendremos una primera parte más conceptual que luego iremos concretando. Por último se mostrarán las conclusiones generales que surgen de todo el proceso. Las diferentes partes del documento se muestran a continuación.

- **Estado del arte**

El punto de partida necesario para abordar este proyecto será el de observar globalmente el entorno donde nos vamos a asentar. Se hará un estudio detallado de la situación general del mercado, donde se mostrarán otras herramientas parecidas sobre el análisis de redes sociales. Obtendremos características globales de estas y sus respectivas bondades de cara a otras herramientas.

Una vez obtenidas todas las características se hará una comparativa que servirá como filtro para poder elegir aquellas características comunes en ellas y, que a su vez, nos mostrará aquellas partes en las que las distintas herramientas sean carentes y que en nuestro proyecto podremos atacar.

- **Especificación de requisitos**

Concretadas ya las características en los que nuestro sistema destacaría respecto a otras herramientas nos centraremos en la visión de conjunto de nuestro proyecto.

Se hará un análisis de funcionalidad de nuestra plataforma en la que se indicarán, haciendo uso de estándares documentales, el alcance y la perspectiva funcional del proyecto.

- **Diseño del sistema**

Definido el alcance del producto se realizará un diseño conceptual en el que se reflejarán las distintas funcionalidades descritas en el anterior punto pero de una manera más ampliada, para tener ya una visión específica de cada una de ellas.

Aquí trataremos tanto el desarrollo conceptual, como el desarrollo formal de la aplicación, donde se verán reflejadas definiciones cercanas a cómo será el sistema. Finalmente, se mostrarán un conjunto de mockups que describen la parte gráfica de la plataforma.

- **Implementación e Implantación**

Con el modelo ya consensuado y estudiado se realizará la implementación del producto con todas las características que se han listado en el anterior punto.

Además se hará uso de técnicas de evaluación que consistirán en pruebas para observar el correcto funcionamiento del sistema y su tratamiento de errores.

- **Conclusiones**

En esta parte se hará un repaso general de todas las pautas que se han seguido. Se hará un estudio sobre los resultados obtenidos con la aplicación y se comentará toda aquella información relevante de esta.

Cómo añadido se hará un estudio comparativo con la aplicación de otro proyecto para analizar las semejanzas y diferencias en el estudio de otras redes sociales.

Para finalizar se dará una apreciación personal que servirá como conclusión al proyecto y en el que se mostrará tanto mi visión general del ámbito de desarrollo del proyecto, como todo el proceso de creación de este.



Estado del arte

Introducción

Si hacemos un repaso a la historia de las redes sociales nos fijamos que todas ellas empiezan siguiendo un modelo de expansión enfocada a las diversas características con las que quieren destacar. Facebook con su tono generalista, Twitter con su simplicidad e inmediatez, LinkedIn con su orientación al mercado laboral. Todas ellas empiezan y evolucionan gracias al crecimiento continuo.

Por esto, gracias a la gran cantidad de información que los usuarios generan en las redes sociales es posible realizar estudios que puedan determinar tus aficiones, tus gustos, tus referencias en general. Todo ello genera un perfil lo suficientemente cercano cómo para poder enseñarte cómo más relevante aquello que te puede interesar.

Pero este perfil sirve para nutrir también otros perfiles conectados. Sirve para saber la importancia que tienes en las conexiones. Puedes ser alguien que reciba la información, o alguien que la expanda. Al fin y al cabo, un negocio en el que las redes sociales encuentran usuarios potenciales, o en su defecto, personas que debido a su importancia pueden generar otros usuarios potenciales.

Muchas empresas han visto una gran capacidad latente en la extracción de este gran cúmulo de información. Claramente, las redes sociales, como tal, son las primeras y a su vez las más grandes consumidoras de estos datos. Como se verá más adelante, los propios análisis públicos de perfil que te ofrece Twitter son ya de por sí amplios y directos, y ya sólo con estos datos se pueden extrapolar a otros más complejos generando resultados reveladores. Además, el propio Twitter te vende paquetes de mejora a estos resultados.

Existen empresas dedicadas exclusivamente al estudio de tus redes sociales. Herramientas de análisis que reflejan resultados de tus redes sobre aspectos relevantes que debes mejorar, dependiendo de tus interacciones, seguidores o amistades en cada una de ellas.

Haciendo un repaso general a primera vista, vemos que existe una gran variedad de enfoques sobre el análisis de las redes, pero a su vez todas ellas se centran en buscar los aspectos más relevantes y con los que deberías interactuar.

Veremos en esta sección un amplio abanico de analizadores de redes sociales, sobre todo las especializadas en Twitter. Analizaremos todas ellas y haremos un filtro de lo que nos ofrece cada una. Tanto sus ventajas, cómo sus desventajas respecto a otras.

Sistemas similares

La clave a la hora de realizar estrategias de negocio en Twitter es casi siempre el contenido. Twitter no es una red social en la que puedes publicitar un negocio a todas horas, necesitas sobretodo generar un contenido que motive a tus usuarios y crear así conversaciones entre tú y tus seguidores alcanzando así un equilibrio en el que puedas aportar algo que te diferencia de los demás.

Por ello, en el mercado han ido apareciendo cientos de aplicaciones para poder facilitar la tarea de gestionar la red social y centrarte en el contenido. Entre ellas podemos distinguir tres tipos, las herramientas de gestión, las de monitorización y las de análisis.

Las herramientas de gestión, como indica su nombre, sirven para ayudar al usuario a administrar la cuenta o cuentas de una manera fácil y rápida, para así centrar la atención en lo importante, el contenido.

Las herramientas de monitorización facilitan la obtención de información de una forma inmediata de los acontecimientos, tanto generados por ti como los que no y que te pueden afectar tanto directa como indirectamente.

Por último, encontramos las herramientas de análisis. Estas herramientas están enfocadas en obtener información relevante de la gran cantidad de datos de la red social. Esta información es la que te ayuda a la hora de tomar decisiones y sacar conclusiones sobre lo que está sucediendo en cada momento.

Como el ámbito de la aplicación a desarrollar es el análisis, vamos a comentar las herramientas más importantes en este momento y las compararemos para extraer las características propias de nuestro sistema.

El orden de aparición tendrá siempre en cuenta el orden de relevancia que se tiene en la web, situando en primer lugar las aplicaciones que aparecen más veces en las diferentes webs relacionadas con el social media marketing.



Twitter Analytics



<https://analytics.twitter.com>

Comenzamos hablando de la principal herramienta a la hora de hacer un análisis de la red social Twitter, la propia de Twitter. Esta herramienta abierta a cualquier usuario de la red muestra en tres pantallas la diversa información relevante que afecta a nuestra cuenta.

Esta herramienta, cómo se ha dicho, divide la información en tres pestañas fácilmente identificables. La primera de ellas es “Actividad de Tweets”, donde se tratará el impacto que ha generado en la red tus tweets.



Figura 1: Gráfica de impresiones de los tweets

En el centro nos muestra en una gráfica con las impresiones los tweets que has publicado. Twitter en este caso se refiere a impresiones cómo el número de veces que un tweet tuyo ha aparecido en las búsquedas o en los perfiles de otro usuario.

A la derecha de esta pantalla nos aparecen también diferentes gráficas que nos ayudan a visualizar de una forma gráfica el impacto que nuestros tweets han tenido en las fechas indicadas.

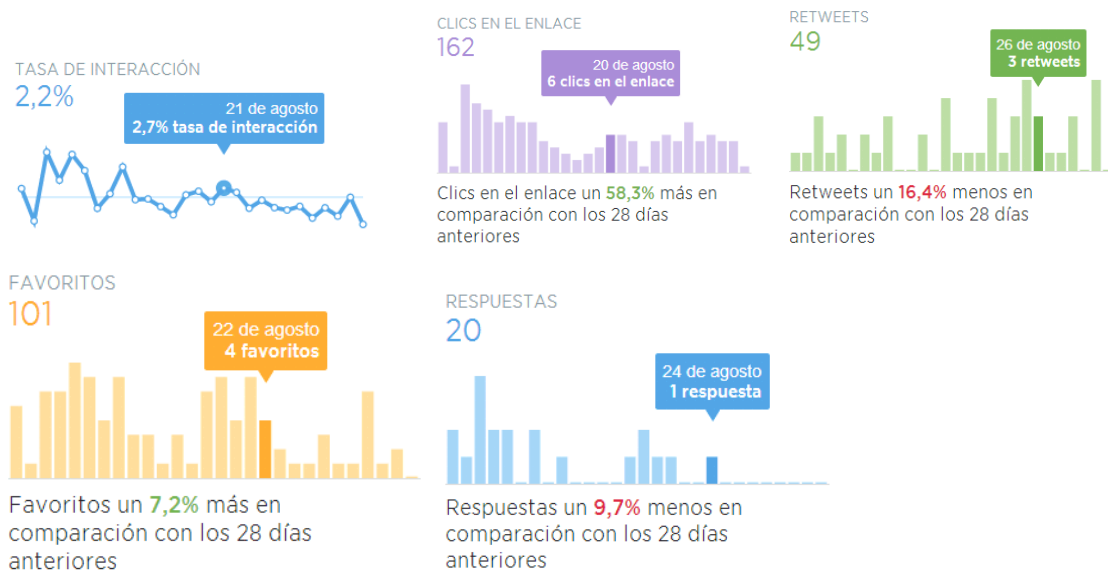


Figura 2: De izquierda a derecha y de abajo a arriba, gráficas sobre la tasa de interacción, clicks en enlaces, retweets, favoritos y respuestas

Por último, debajo de esta gráfica general podrás seleccionar uno por uno los tweets publicados y poder obtener toda la información anterior individualizada para cada uno de ellos.

La segunda pestaña de Twitter Analytics es la de “Seguidores”, donde aparecen diversos aspectos relevantes de los usuarios que te siguen para así poder tener más claro cuáles son sus intereses.

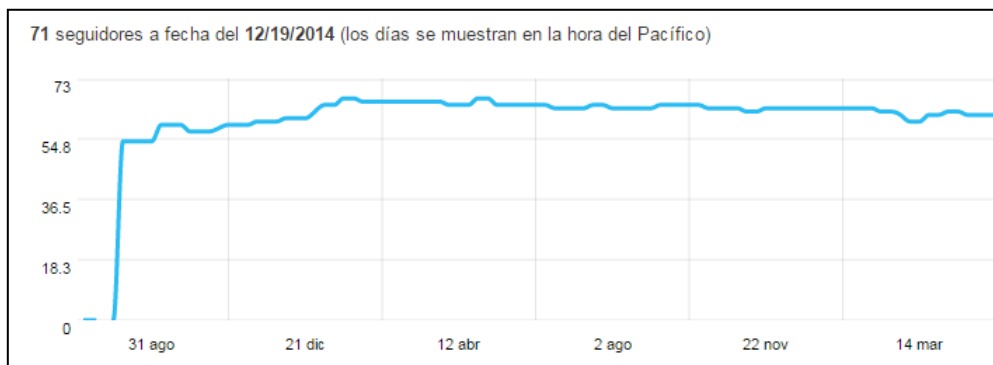


Figura 3: Gráfica del número de seguidores

En el centro de todo nos aparece una gráfica interactiva en la que nos aparece el número de seguidores que tenemos según el tiempo. Además, podemos ver, para cada día, el número de seguidores que tenemos y visualizar gracias a la gráfica en que momentos ganamos/perdimos seguidores y entender el por qué.

Debajo nos aparecen diversos paneles para encontrar patrones en los usuarios. El primero es el de intereses, donde nos aparecen porcentajes de los intereses de nuestros usuarios. Los intereses generales, como pueden ser Música, Informática, Ciencia, etc, y las específicas, se pueden subdividir en Música Clásica, Algoritmos, etc.

Luego aparece una gráfica de la ubicación donde se encuentran nuestros seguidores, haciendo primero una distinción por países y debajo por provincias. La última gráfica divide a tus seguidores por género.

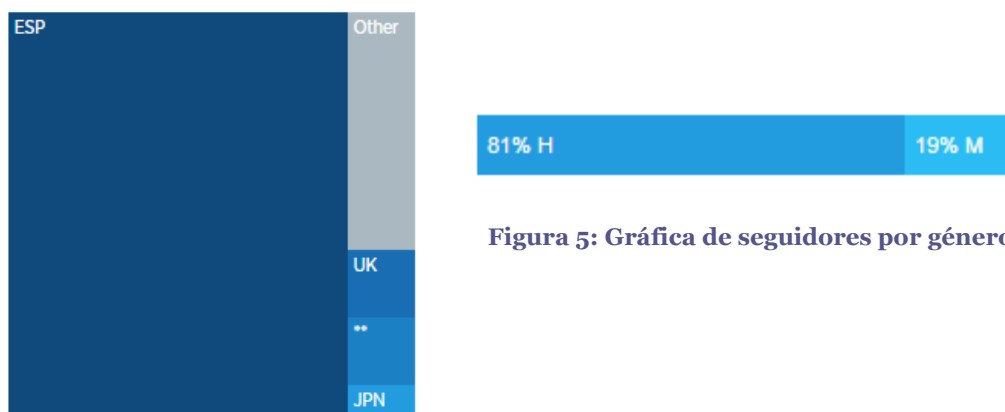


Figura 5: Gráfica de seguidores por género

Figur a4: Gráfica de seguidores por país

El último panel de tus seguidores es el que te ofrece información sobre a quién siguen tus seguidores, ofreciendo información de tus posibles competidores en ese ámbito de contenido.

Las siguientes pestañas tratan sobre aspectos que son más específicos de publicidad y marketing de marcas, por lo que sólo las nombraremos y daremos una breve explicación de ellas.

La primera trata sobre las “Twitter Cards”. Las Twitter Cards son un formato enriquecido para los tweets que le permiten agregar información multimedia en ellos, así como enlaces para su posterior análisis. Gracias a esta clase de tweets, Twitter Analytics es capaz de analizar cómo se comparte el contenido por Twitter.



Figura 6: Twitter Card de ejemplo

Una vez seleccionada la Twitter Card correspondiente, nos aparecerá para cada una de ellas la información correspondiente, dividida en varias secciones. Una imagen que nos mostrará la impresión que ha tenido, y varios paneles para filtrar como se ha ido compartiendo, como puede ser por el tiempo.

La siguiente pestaña, “App Manager”, que sirve de ayuda para poder promocionar aplicaciones móviles que hagan uso de Twitter, tanto con información sobre sus aplicaciones como su posterior promoción.

Para finalizar, la última pestaña es la de “Seguimiento de conversaciones”. Esta funcionalidad muestra información personalizada sobre las campañas de publicidad que ofrees, dándote información precisa sobre de que se está hablando en estos momentos sobre tu marca.

Twitonomy es una herramienta web que nos permite realizar un completo análisis de la cuenta con la que nos identifiquemos. Existen dos versiones, una gratuita y una de pago, que añade diversa funcionalidad.

Lo primero que nos aparece una vez entramos dentro de la aplicación es la pestaña de “Dashboard”, donde nos aparecerán los datos de nuestra cuenta, un panel de noticias sobre la web y una lista con tu timeline. Aquí podremos añadir además, otros timelines de otros usuarios, permitiendo su monitorización.

Por otra parte, puedes acceder a la funcionalidad de tracking de tus seguidores y hacer un backup de ellos para luego descargarla. Pero esta funcionalidad se encuentra en la versión de pago.

La siguiente pestaña es la de “Profile”, donde se muestra una completa ficha sobre nuestra cuenta o la que nosotros prefiramos. Por un lado tenemos en diversas listas la siguiente información:

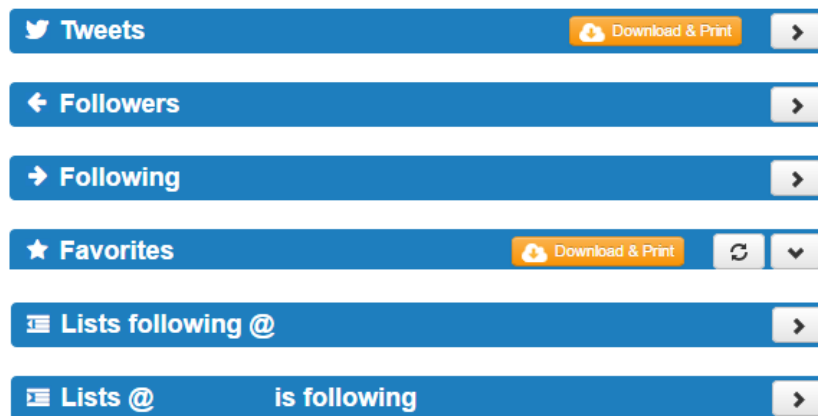


Figura 7: Listas con la información de tu cuenta

Por otra parte, nos hará un análisis detallado sobre nuestros tweets, mostrados todos ellos por métricas o gráficos.

1. Total tweets del periodo analizado (para poder elegir un rango tienes que ser premium), tweets por día, menciones, tweets lanzados con links, total tweets retweeteados, total tweets que han sido favoritos, etc.
2. Un gráfico historial de nuestros tweets.
3. Lista con avatares de los seguidores que con que más hemos interactuado: mencionado, retweeteado o respondido, así como los hashtags que más hemos utilizado.
4. Tweets más retweetados y los más favoritos.
5. Días de la semana y horarios de nuestra actividad
6. Plataformas que hemos utilizado (hootsuite, web de Twitter, aplicación móvil de Twitter por tipo de dispositivo, botones en webs o blogs...).

La siguiente pestaña es la de “Mentions & RTs”, dándonos información detallada de nuestras relaciones con otros usuarios y varia información sobre estos. Entre la información que podemos encontrar en ellas podemos observar la siguiente:

1. Menciones que recibimos por día.
2. Los usuarios más activos y más influyentes con avatares y los principales datos de sus cuentas.
3. Tweets, retweets y menciones favoritos (parte Premium).
4. Top hashtags y periodos con más retweets y favoritos (parte Premium).
5. Gráficos con el número total de cada una de las métricas del periodo analizado añadiendo un Reach potencial por cada uno de ellos.
6. Mapa interactivo que nos va mostrando y colocando en tiempo real cada uno de los países y ciudades desde donde se han emitido tweets que nos han mencionado en el periodo.

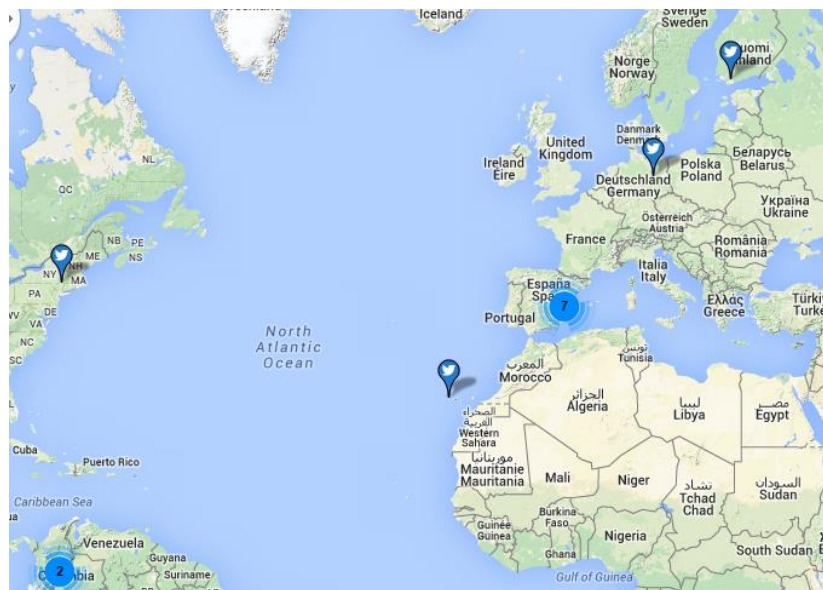


Figura 8: Mapa interactivo de tus menciones

Pestaña de “Search”, bloqueada para las cuentas Premium, donde se nos muestran resultados sobre búsquedas que hayamos hecho, como puede ser sobre usuarios concretos, keywords o hashtags, incluyendo también la posibilidad de visualizar el mapa interactivo geolocalizado de cada uno de ellos.

Continuamos con las pestañas de “Followers y Following”, donde nos muestra para cada uno de ellos información relevante sobre cada uno de estos usuarios que puede ser extendida simplemente clickando sobre ellas.

Para finalizar tenemos la pestaña de “Lists” donde tendremos acceso a todas las listas de Twitter que hemos creado (tanto públicas como privadas) y también en las que nos han incluido.



Tweetchup se trata de nuevo de una aplicación web que nos permite consultar ciertas métricas de la cuenta con la que te identifiques en Twitter. Nada más identificarte veremos una interfaz minimalista en la que tendremos una barra lateral que nos servirá de menú para poder acceder a las distintas opciones.

La primera de ellas, y que es la que te aparece cuando te identificas, es el panel de “Connections”. En ella se puede ver una gráfica donde aparecen las veces te han mencionado en los últimos meses y quienes. Tweetchup también calcula una media del número de menciones por día y ofrece la cifra de cuántas veces se retweetearon los mensajes que te mencionan. Además, ofrece un mapa donde puedes ver en qué lugar del mundo se te ha mencionado más. Las estadísticas sobre tus tweets retweeteados también se pueden consultar en este panel.

El siguiente panel es ‘Profiles’, donde puedes buscar perfiles de Twitter y comprobar cuáles son sus métricas. De un vistazo ves su número de seguidores, a los que siguen, en cuántas listas están, así como estadísticas más avanzadas. Entre estas últimas está la media del número de tweets por día, cuántos usuarios fueron mencionados, los enlaces contenidos en los tweets, así como cuántos mensajes son retweets y cuántos son respuestas a otros usuarios.

Este segundo panel contiene estadísticas verdaderamente aprovechables, como la media de tweets por día de la semana o por horas, cada día. Los mensajes más retweeteados, los usuarios más mencionados o los hashtags más utilizados también están dispuestos en forma de rankings.

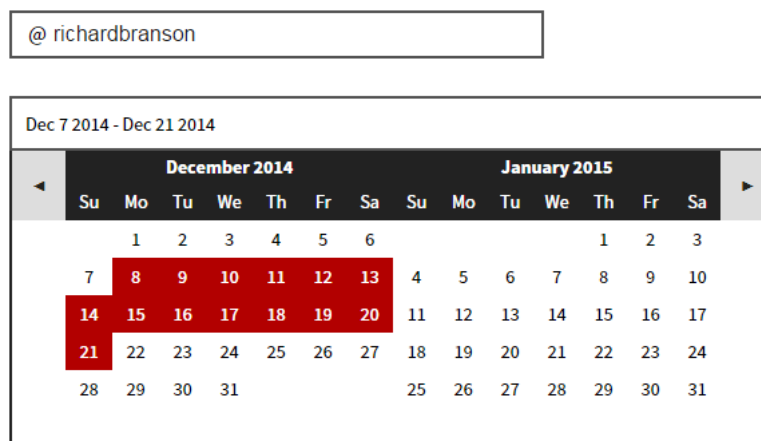


Figura 9: Buscador de usuarios y el período a analizar

El tercer panel, ‘Keywords’, permite buscar una palabra o varias y Tweetchup resumirá en un puñado de estadísticas lo más relevante que se puede encontrar en Twitter sobre ese término. Con gráficas que nos mostrarán la influencia de ésta, así como información sobre los usuarios que la compartieron y también los hashtags relacionados a esta búsqueda.

Twitter counter



<http://twittercounter.com/>

Twitter counter es un servicio web de análisis para Twitter que proporciona información relevante sobre tu cuenta de Twitter, además de poder buscar información sobre otros usuarios.

Una vez identificados, apareceremos en la pantalla de “Overview” de “Audience”. Esta pantalla nos ofrecerá información sobre los seguidores de nuestra cuenta, así como un ranking mundial de cómo de relevante es nuestra cuenta. Aquí podremos generar gráficas, según un periodo de tiempo, de nuestros followers, nuestros followings, tweets, unfollowers y sobre nuevos followers (estos dos últimos con una cuenta Premium). Permitiendo a su vez compararlos entre ellos.





Twitter users	Followers	Following	Tweets
1  KATY PERRY @katyperry	61,772,050	160	6,258
2  Justin Bieber @justinbieber	58,058,473	166,817	28,015
3  Barack Obama @BarackObama	51,514,727	645,963	12,887
4  Taylor Swift @taylorswift13	48,776,480	156	3,058

Figura 10: Listado de los usuarios más influyentes de Twitter

Si avanzamos por el menú de la derecha dentro del menú “Audience”, podemos elegir entre analizar los followers, los following, los unfollowers y los nuevos followers. Permittiéndonos en cada uno de ellos poder generar gráficos comparativos entre un usuario o entre dos usuarios.

La siguiente opción del menú trata sobre información relevante sobre los tweets que has publicado, y como norma general de la aplicación web, todo aparece en forma de gráfica con la que podrás información relevante a lo largo del tiempo y poder enfrentarla con otras métricas.

Existen dentro de este menú opciones para poder visualizar y comparar las menciones y lo retweets, pero está bloqueada para aquellos usuarios Premium. También bloqueada se encuentra la sección de “Interactions”, donde se pueden observar por medio de gráficas las interacciones que existen entre diversas cuentas.

Por último, la web ofrece una lista variada de servicios propios:

1. Un *plugin* para el navegador con acceso rápido a toda la funcionalidad de la web.
2. Un análisis que diagnostica el nivel de tu cuenta de Twitter y diversas ayudas para mejorarlo.
3. Un comparador y un tracking de tu cuenta.
4. Un *widget* y un botón para incluirlo en tu web

SocialBro es una plataforma de marketing que ofrece un conjunto de herramientas que analizan y actúan sobre los datos de Twitter para hacer crecer, contactar y monetizar la audiencia de tu cuenta de Twitter.

Una vez registrados y vinculada nuestra cuenta de Twitter, entraremos en la pantalla de inicio de la aplicación. Aquí encontraremos una lista de opciones para poder elegir aquellas que queremos monitorizar (esta parte es siendo Premium). Podemos monitorizar tanto cuentas, como listas, como cualquier palabra que nos interese.

La siguiente sección de la web es la de “Comunidad”, en la que nos aparece un listado de todos nuestros seguidores, pudiendo filtrarlos por una gran cantidad de opciones entre las que se incluyen por importancia de estos o por zonas geográficas. Además podrás añadir otras cuentas para poder hacer comparativas con ellas (esta parte también Premium).

La siguiente sección, “Reports”, sirve para visualizar toda la información que has generado en la sección de Analítica. Por ello, vamos a visualizar las diversas opciones que nos ofrece la herramienta a la hora de analizar la información.

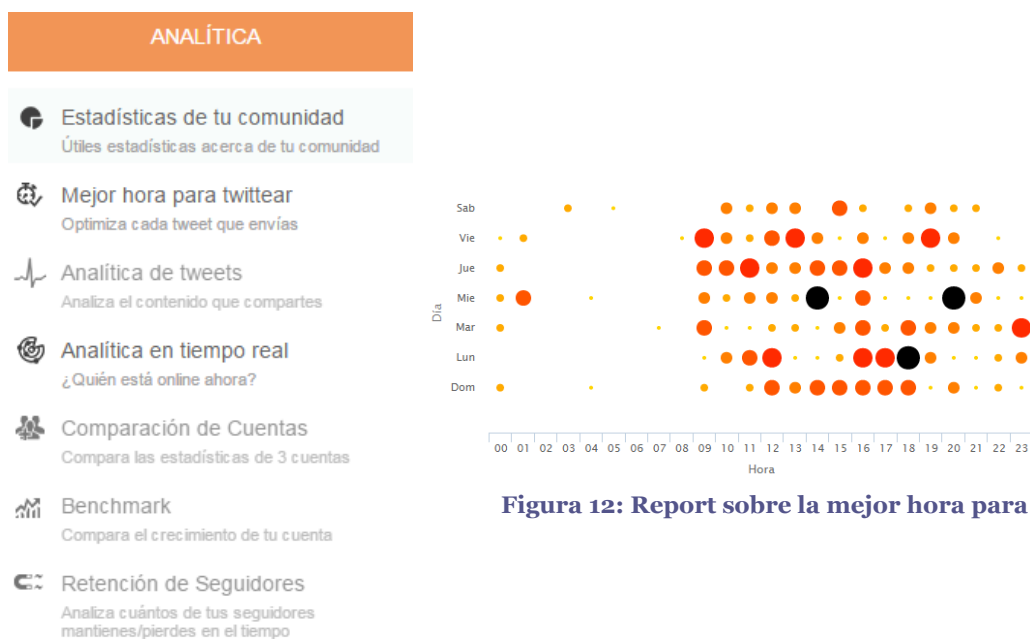


Figura 12: Report sobre la mejor hora para

Figura 11: Listado de reports

La siguiente sección es la de “Interacción” que nos sirve para automatizar de una forma sencilla diversas acciones en Twitter. Esta automatización se ayuda de los datos que has tomado en los análisis para poder tomar la mejor decisión a la hora de publicar el contenido.

La última sección es “Segmentación” y sirve para poder expandir tu cuenta en la red y así poder tener más relevancia en ella. De nuevo se ayuda de la información de los análisis para poder tomar decisiones que te ayudan a encontrar nuevos usuarios.

Tweet Archivist

TWEET ARCHIVIST <https://www.tweetarchivist.com/>

Aplicación web que busca información en la red social en base a una palabra o un término de búsqueda y crea un archivo a través de ella que monitoriza automáticamente. Usa los operadores para la búsqueda avanzada propios de la API de Twitter para poder encontrar fácilmente lo que se busca, tanto hashtags, como usuarios como búsquedas complejas

Una vez hecha una búsqueda aparecerán los últimos tweets relacionados con el término buscado y una larga lista de desplegables con información sobre el término.

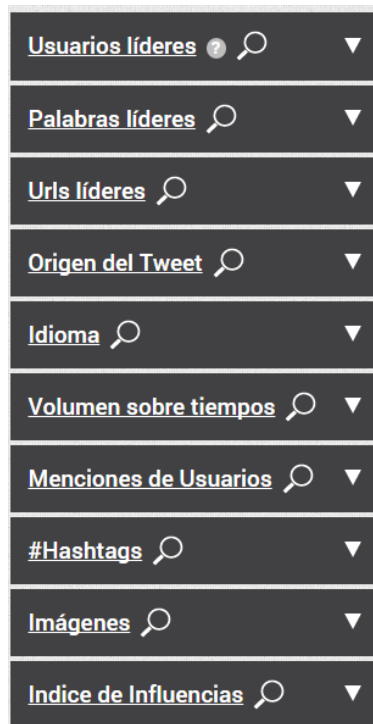


Figura 13: Opciones disponibles

Todas estas opciones mostradas con gráficos u otros mecanismos para facilitar su posterior análisis. Además se pueden acceder a ellos pulsando en el enlace, lo que abre una nueva ventana en la que aparece información más detallada sobre ellas.

Por otra parte, puedes descargarte los distintos análisis en distintos formatos para poder ejecutar sobre ellos diferentes análisis posteriores, pero en este caso, sería comprando una cuenta Premium.

Topsy

TOPSY

<http://topsy.com/>

Topsy es un buscador social que trabaja sobre Twitter. El servicio ofrece tres opciones diferentes según el enfoque con el que se desea trabajar. La primera sección es el “Social search”, que es una búsqueda a través de Twitter del término a buscar. En esta parte se pueden seleccionar diferentes filtros para enfocar la búsqueda.

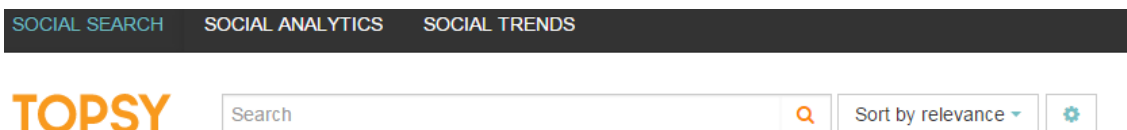


Figura 14: Buscador Topsy

La funcionalidad más significativa de este buscador es que te muestra los resultados según un orden de relevancia que se genera gracias a un algoritmo propio que mide el impacto de los tweets mostrados. Esto da información precisa sobre las fuentes y quiénes son los usuarios más influyentes de tu búsqueda.

La segunda sección es una gráfica en la que se pueden comparar los tweets por día de diferentes búsquedas, tanto los tweets por términos, como por usuario o enlace.

Por último, la sección “Social trends” muestra la información más relevante que está ocurriendo en la red social en tiempo real. De nuevo usa su algoritmo para mantener ordenada la lista que ofrece y que además podemos filtrar con diferentes opciones.

Tweetreach**tweetreach**<https://tweetreach.com/>

TweetReach es una herramienta que permite analizar el impacto que ha tenido tanto una url concreta, un término, un hashtag o un usuario y su alcance. Una vez introducido el término a buscar nos aparecerá en la pantalla distintos paneles con información. Los más importantes son el potencial estimado, que es el número de usuarios a los que podría llegar el término, y el número de tweets que se han repartido entre los usuarios de la red.

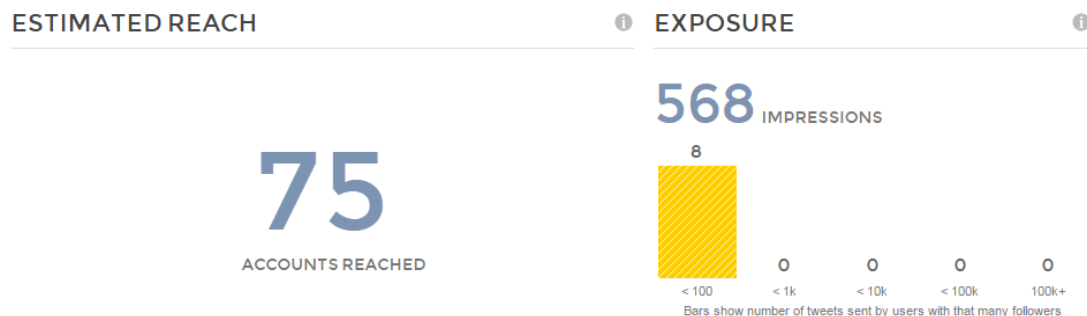


Figura 15: Potencial e impresiones

Además aparecen otros paneles con información relevante de la búsqueda realizada. Las opciones según orden de aparición son las siguientes:

1. Activity: ofrece 2 gráficos, uno que de forma lineal nos muestra cómo ha sido la semana, el número de Tweets y cuantos contribuidores ha habido y otro circular que nos ofrece el número de Tweets, ReTweets y Menciones.
2. Top de los máximos contribuidores a la búsqueda.
3. Top de los Tweets que más se han retweetado.
4. Listado con todos los contribuidores.
5. Timeline incluyendo el número de Tweets analizados.

6.

Hootsuite



<https://hootsuite.com/es/>

La aplicación web Hootsuite no se basa solo en una aplicación analítica, se trata de una herramienta para la gestión de redes sociales pero que cuenta con una parte de análisis para ayudarte en la toma de decisiones. En esta parte se pueden generar informes para las distintas redes sociales que hayas vinculado.

En la parte de gestión podrás filtrar en diferentes paneles, información relevante de tus cuentas. Pudiendo separar los tweets de las menciones para poder tener un orden claro a la hora de mantener conversaciones, o poder seguir la información que te sea relevante. Además de poder programar ciertos tweets a determinadas horas.

Si nos centramos en la parte de análisis, la parte significativa de esta aplicación es, siendo de pago, el poder generar tus propios informes de una forma sencilla, para tener todo ordenado según el criterio que para ti sea más relevante.

Los datos que muestra la aplicación son mostrados según diferentes gráficas y utilizando distintos filtros para poder proporcionar la información de una forma directa y fácil de entender.

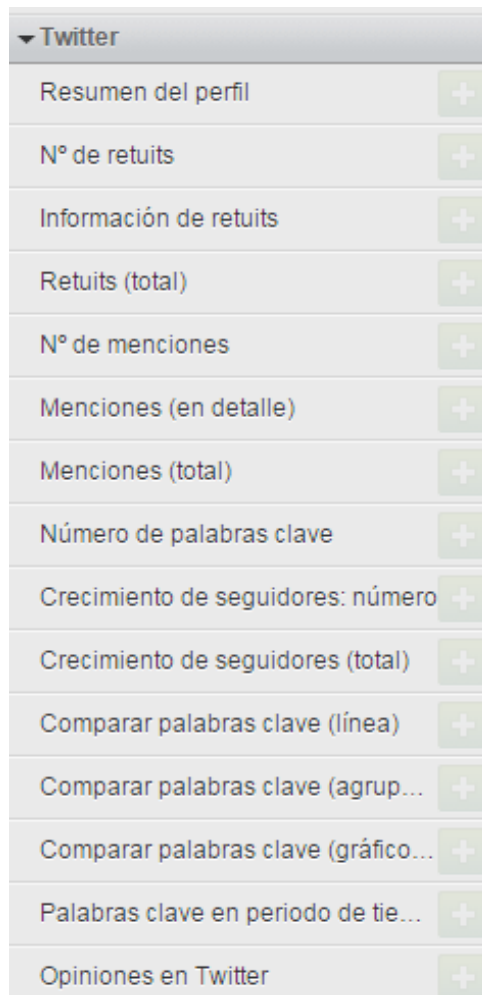


Figura 16: Opciones para la creación del informe sobre Twitter



Análisis de los sistemas

La comparativa de los sistemas se verá reflejada en dos partes que reflejarán las características más relevantes de cada uno de los sistemas similares respecto de otros. En la primera parte se tratarán temas relativos al sistema en sí, que no dependerán de nuestra apreciación y ya en la segunda parte, aquellos puntos en los que sí que dependerá nuestra apreciación.

Se listarán ahora las características que usaremos en las tablas de comparación divididas como se ha dicho antes en las características objetivas y subjetivas:

- **Características objetivas**

- ❖ Precio: Se estudiará el impacto económico de la herramienta.
- ❖ Plataforma: Sistema que sirve como base para hacer funcionar la aplicación.
- ❖ Métricas: Consideraciones de las métricas mostradas en el análisis tanto en cantidad como en la calidad de las mismas.
- ❖ Backward/Forward: El análisis se realiza sobre datos ya existentes, o se hará un “directo” del proceso, capturando los datos en el mismo instante.
- ❖ Gestión múltiple de redes sociales: Se medirá si la aplicación está orientada en modo general a distintas redes sociales o estará centrada en una sola.
- ❖ Requiere autenticación: Se observará si será necesaria la identificación en la aplicación, en la red social, o no será necesario.

- **Características subjetivas**

- ❖ Facilidad de Uso: Calificación a primera vista de la sencillez y la claridad de la funcionalidad.
- ❖ Visualización de la información: Calificación también de la sencillez y la claridad de los resultados mostrados.
- ❖ Finalidad del producto: Objetivos de la aplicación.

Una vez estudiadas las características propias de los sistemas que hay en el mercado, se hará una tarea de extracción de propiedades relevantes que tendrá nuestra herramienta. Esta distinción tendrá por su parte una fuerte componente subjetiva, propia del desarrollo de aplicaciones, pero que será apoyada por los propios resultados del análisis.

Análisis subjetivo

	Facilidad de uso	Visualización de la información	Finalidad del producto
Twitter Analytics	Rápida y sencilla. Una vez activada esta opción en tu cuenta de Twitter, acceder a ella es muy fácil y toda la información es accesible muy rápido	Todo se muestra en gráficas coloridas con patrones de color. Muy directo a la hora de mostrar los datos relevantes	Ayudar al usuario en la toma de decisiones a la hora de expandirse en Twitter. Intenta vender también sus paquetes de marketing, que sirve cómo sustento para la propia red social
Twitonomy	Al ser una herramienta demasiado completa peca de complejidad, lo que hace resentirse en su facilidad de uso	Toda la información aparece en gráficas o listas, pero es necesario un análisis detallado de cada uno por la gran cantidad de ellas.	Ayudar al usuario en la toma de decisiones para expandirse en Twitter.
Tweetchup	Muy limpia y muy simple, con toda la información accesible de forma muy cómoda	Al tener una interfaz sencilla, todas las gráficas y datos que se muestran se leen de una forma clara. La búsqueda también es acertada a la hora de mostrar la información	Ayudar al usuario en la toma de decisiones para observar que puntos son los más adecuados a la hora de expandirse en Twitter
Twitter counter	Aplicación rápida y limpia sobre la información que quiere mostrar. Los menús son a su vez muy grandes debido a la cantidad de opciones que tiene la aplicación	La información se muestra dividida en dos partes, ordenando siempre por la relevancia de los Tweets o del usuario buscado para hacer las comparativas	Se venden como una empresa especialidad en ayudas para la publicidad y el marketing en la red social
SocialBro	Gran cantidad de opciones bien resueltas debido a la simpleza de su interfaz	La actividad se disecciona en gráficos detallando fácilmente la información relevante	Herramienta de marketing para Twitter que ofrece paquetes de ayuda a las empresas para promocionarse en Twitter



Tweet Archivist	Directo gracias a su buscador y fácil en la muestra de los resultados	La información relevante sobre la búsqueda se muestra en sencillas gráficas	Sirve como analizador a posteriori de eventos que han ocurrido en la red social. Ofrece paquetes de pago para poder ir analizando al tiempo lo que ocurre para una determinada búsqueda
Topsy	El buscador actúa como eje en la herramienta, y gracias a esto se muy cómoda de usar	La información se muestra en forma de lista donde la mayoría de los datos se muestran de una forma clara y sencilla	Sirve para ver cómo se ha desarrollado una actividad reciente en la red social de Twitter
Tweetreach	El buscador es el eje principal de la aplicación, a partir de ella se muestra toda la información principal	La información se apoya en gráficas destacando el puntaje que se le da a la importancia de la búsqueda	Ayuda a visualizar de una forma inmediata la importancia que ha tenido un determinado evento en la red social de Twitter
Hootsuite	Muy fácil de gestionar la multitud de cuentas que se pueden tener la aplicación, pero debido a su complejidad, es necesaria un periodo de aprendizaje de la herramienta	Toda la información se reparte en diferentes pantallas dependiendo de la red social. Se apoya en diversas gráficas para ayudar en la claridad de los resultados	De nuevo ofrece paquetes para la gestión de redes centrándose sobre todo en soluciones para el marketing y promoción de las empresas

Análisis subjetivo

	Facilidad de uso	Visualización de la información	Finalidad del producto
Twitter Analytics	Rápida y sencilla. Una vez activada esta opción en tu cuenta de Twitter, acceder a ella es muy fácil y toda la información es accesible muy rápido	Todo se muestra en gráficas coloridas con patrones de color. Muy directo a la hora de mostrar los datos relevantes	Ayudar al usuario en la toma de decisiones a la hora de expandirse en Twitter. Intenta vender también sus paquetes de marketing, que sirve como sustento para la propia red social
Twitonomy	Al ser una herramienta demasiado completa peca de complejidad, lo que hace resentirse en su facilidad de uso	Toda la información aparece en gráficas o listas, pero es necesario un análisis detallado de cada una por la gran cantidad de ellas.	Ayudar al usuario en la toma de decisiones para expandirse en Twitter.
Tweetchup	Muy limpia y muy simple, con toda la información accesible de forma muy cómoda	Al tener una interfaz sencilla, todas las gráficas y datos que se muestran se leen de una forma clara. La búsqueda también es acertada a la hora de mostrar la información	Ayudar al usuario en la toma de decisiones para observar que puntos son los más adecuados a la hora de expandirse en Twitter
Twitter counter	Aplicación rápida y limpia sobre la información que quiere mostrar. Los menús son a su vez muy grandes debido a la cantidad de opciones que tiene la aplicación	La información se muestra dividida en dos partes, ordenando siempre por la relevancia de los Tweets o del usuario buscado para hacer las comparativas	Se venden como una empresa especialidad en ayudas para la publicidad y el marketing en la red social
SocialBro	Gran cantidad de opciones bien resueltas debido a la simpleza de su interfaz	La actividad se disecciona en gráficos detallando fácilmente la información relevante	Herramienta de marketing para Twitter que ofrece paquetes de ayuda a las empresas para promocionarse en Twitter

Tweet Archivist	Directo gracias a su buscador y fácil en la muestra de los resultados	La información relevante sobre la búsqueda se muestra en sencillas gráficas	Sirve como analizador a posteriori de eventos que han ocurrido en la red social. Ofrece paquetes de pago para poder ir analizando al tiempo lo que ocurre para una determinada búsqueda
Topsy	El buscador actúa como eje en la herramienta, y gracias a esto se muy cómoda de usar	La información se muestra en forma de lista donde la mayoría de los datos se muestran de una forma clara y sencilla	Sirve para ver cómo se ha desarrollado una actividad reciente en la red social de Twitter
Tweetreach	El buscador es el eje principal de la aplicación, a partir de ella se muestra toda la información principal	La información se apoya en gráficas destacando el puntaje que se le da a la importancia de la búsqueda	Ayuda a visualizar de una forma inmediata la importancia que ha tenido un determinado evento en la red social de Twitter
Hootsuite	Muy fácil de gestionar la multitud de cuentas que se pueden tener la aplicación, pero debido a su complejidad, es necesaria un periodo de aprendizaje de la herramienta	Toda la información se reparte en diferentes pantallas dependiendo de la red social. Se apoya en diversas gráficas para ayudar en la claridad de los resultados	De nuevo ofrece paquetes para la gestión de redes centrándose sobre todo en soluciones para el marketing y promoción de las empresas

Características del sistema a desarrollar

Con toda la información de la que disponemos después de realizar la comparativa correspondiente con los sistemas similares, se hará una extracción de características para concretar aquellas que tendrá nuestro sistema.

Primero empezaremos por las características objetivas:

❖ **Modelo Premium**

Todas las herramientas que hemos analizado son gratuitas en la parte de visualización de resultados. Algunos muestran más cantidad de resultados, otros apuestan por resultados más relevantes, pero es a la hora de la interpretación de estos resultados cuando se paga.

- Para el proyecto también se realizará la parte de visualización de los resultados de forma gratuita. Una vez tuviéramos esta base realizada y se pusiera a disposición del público, se realizaría entonces un estudio de cuál sería la viabilidad de realizar estos planes de pago.

❖ **Plataforma**

De nuevo, la totalidad de las aplicaciones se realiza a través de la web, algunos añadiendo, además, alguna otra plataforma, pero siempre teniendo de base la web.

- Se usará también una plataforma web para la realización del proyecto ya que debido a la naturaleza de las redes sociales, para facilitar la accesibilidad y evitar tener que instalar un software específico, estas siempre están en entornos conectados.

❖ **Métricas**

Todas las aplicaciones comentadas se centran en dar resultados de relevancia de los tweets asociados a la cuenta de un usuario, apoyadas en vistosas gráficas que muestran de una forma directa y clara. Es por esto que la decisión de estas métricas y la forma de mostrarlas es la característica más importante de todas, así que nos centraremos aquí para la parte del aspecto diferenciador de nuestra aplicación.

- La aplicación se apoyará en gráficas a la hora de mostrar los resultados más importantes.
- Utilizaremos métricas que sirvan para mostrar la relevancia de los usuarios en la red social que emerge de las interacciones entre usuarios que participan en un evento, pero haciendo un análisis mucho más profundo, creando así uno de los aspectos diferenciadores.

❖ **Backward/Forward**

La mayoría de herramientas que realizan un análisis de Twitter utilizan los datos que existen en la red de followers en lugar de en la red de interacciones de los usuarios para poder realizar el estudio. En cambio, las herramientas centradas en realizar análisis sobre interacciones en eventos que ocurren en Twitter añaden la opción de ver en tiempo real lo que ocurre.

- El sistema realizará un análisis de la red que emerge de las interacciones entre usuarios de un evento.

❖ **Gestión múltiple**

La gran mayoría de analizadores de redes sociales se suele centrar una herramienta para cada una de las redes sociales.

- Para la aplicación del proyecto nos centraremos en Twitter para hacer el analizador pero teniendo en mente la idea de que esta sea la base para el uso en otras redes sociales. Es bueno, además, centrarnos en esta para luego hacer la comparativa con el proyecto de análisis de la red social Facebook.

❖ **Requiere autenticación**

La mayoría de herramientas estudiadas son usadas para hacer análisis detallados de las cuentas con las que se han logueado. En cambio sí nos centramos en los eventos, este factor no hay que tenerlo en cuenta.

- La aplicación será un analizador de eventos que ocurren en la red social, no un estudio sobre un usuario concreto, por lo que esta opción no estará presente.

Ahora realizaremos el estudio para las características subjetivas:

❖ **Facilidad de uso**

El problema de las herramientas de análisis es que a la hora de realizar estos se tienen en cuenta muchas métricas. Debido a esto, la interfaz de usuario de las aplicaciones se ve resentida por la complejidad de mostrar tanta información.

- Las aplicaciones suelen solventar este problema usando la sencillez a la hora de mostrar las opciones, mostrando las cosas de una forma directa. En nuestra aplicación se buscará esta sencillez en la interfaz a la hora de desarrollarla.

❖ **Visualización de la información**

En la característica de las métricas ya se ha comentado que los resultados serán mostrados en gráficas. La visualización de la información será también uno de los puntos más importantes y tendrá también un cierto extra a la hora de afrontar la aplicación.

- En la aplicación se mostrará de manera visual la información del grafo de interacciones así como las métricas asociadas a él. De nuevo cómo valor añadido y diferenciador.

❖ **Finalidad del producto**

La gran finalidad de las aplicaciones con las que nos hemos encontrado se basan en aportar datos relevadores que sirven para ayudar a la toma de decisiones a la hora de publicar tweets. Todo ello se basa en referencias de qué tipo de tweets, a qué hora publicarlos, si es bueno ser más proactivo en algunos aspectos, y así un largo etcétera. Sabiendo esto, la finalidad es hacer estudios a partir de estos datos para prácticas de marketing. Esta publicidad orientada y directa a clientes potenciales es la más demandada y a su vez es el modelo de negocio de la gran mayoría de las aplicaciones consideradas.

- Nuestra aplicación debería tener en mente, que el modelo de negocio en el que se debe centrar es en los análisis en profundidad de los resultados para luego obtener beneficio de ellos. Esta característica se dejará en “standby” hasta que el producto fuese lo suficientemente reconocido cómo para poder monetizarlo. Por ello, se buscará primero un elemento diferenciador y más adelante se abordarían estos temas, ya con la aplicación con cierto renombre.

Métricas. Análisis en profundidad.

Cómo ya se ha comentado, las métricas que utilizaremos en la aplicación serán la característica diferenciadora con el resto de las aplicaciones. Por ello, vamos a realizar ahora un estudio sobre cuáles serán estas y el porqué de su utilización.

Cuando hablamos sobre el análisis de redes sociales tenemos que observar que lo que buscamos son los puntos de conexión de más impacto. Aquellos que afectan a la red de una manera significativa. Es cuando atacamos a estos nodos cuándo tendremos el mayor porcentaje de éxito para influir en el resto.

Por esto, al buscar la importancia de los vértices en la red, tendremos que hacer estudios sobre la centralidad. La centralidad es una medida que depende estrictamente de la localización en la red. Los nodos con una centralidad alta serían los que ocuparían posiciones privilegiadas en los flujos de información. Por lo tanto, todas las medidas que utilizaremos en la herramienta serán sobre centralidad²³.

² The “New” Science of Networks. Duncan J. Watts. 2004

³ Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality. M. E. J. Newman. 2001



Conclusiones

Muchas empresas jóvenes han nacido alrededor del análisis de redes sociales. Todas han visto el potencial que se encuentra ya que cumplen la máxima de cualquier campaña publicitaria, el boca a boca. La expansión que genera que algo se vuelva viral no tiene comparación con ninguna otra forma de publicidad.

Sólo con echar un vistazo a lo que ofrecen te das cuenta que el público al que se dirigen es siempre Márketing. La figura del Social Media Manager ha ganado enteros en muy poco tiempo en la empresa. Estos se encargan de definir estrategias, todas ellas dotadas de un enfoque empresarial con tal de seguir siendo más competitivas.

Según el último estudio de la asociación digital de la economía digital sobre el uso de las redes sociales en las empresas⁴, "más de la mitad de las empresas considera que el retorno obtenido de las redes sociales es igual o superior a la inversión en ellas".

Es por esto que el filón del sector en el que nos encontramos apoya a la creación de herramientas de análisis de estas. Buscando, eso sí, diferenciación con respecto a otras. Ese valor añadido que destaque.

Las características más comunes que nos encontramos al realizar el estudio son aquellas por las que se crearon las diferentes herramientas. Realizan análisis en los datos que la gente introduce y los muestra de forma gráfica. No hay ninguno que destaque con alguna característica diferenciadora del resto más que por visualizaciones distintas de la misma información.

Al realizar este proyecto actuaremos siempre con esa idea en mente. Realizaremos el desarrollo siguiendo la máxima de añadir ese extra a la hora de mostrar los resultados. Iremos un paso más allá a la hora de elegir las métricas que tendremos y la forma de visualizarlos. Sin olvidar nunca esa facilidad de uso propia de este tipo de herramientas.

Una vez finalizado esto, se debería abordar en el futuro el hito de monetizar estos resultados cuando la aplicación tenga la suficiente relevancia en el sector. Y a su vez, expandir estos análisis a las diferentes redes sociales que existen.

⁴ <http://www.adigital.org/emailing/2014/rss/informe.pdf>

Especificación de requisitos

Introducción

Concretadas las características propias del sistema, en la siguiente sección se realizará la Especificación de Requisitos Software (ERS), apoyada según la última versión del estándar IEEE 830.

Propósito

La presente sección servirá como guía para el personal encargado del diseño y el desarrollo de la aplicación. La sección contiene una descripción completa del sistema a desarrollar mostrando tanto la especificación funcional y no funcional de la aplicación.

Ámbito del sistema

El sistema parte con la máxima de realizar un análisis sobre las redes sociales con métricas distintas a las que las que ofrecen las herramientas que ya están disponibles. A su vez, estas métricas serán mostradas de una forma gráfica para ayudar a discernir rápidamente los puntos más importantes.

Para completar este objetivo el sistema partirá de un término de búsqueda (hashtag asociado a un evento, o una palabra clave, o un nombre) por el cual irá recuperando tweets y creando la red. Con esta red almacenada se podrá actuar sobre ella para generar los resultados que nos servirán para su posterior estudio.

Definiciones y acrónimos

- Definiciones
 - Búsqueda: Acción del usuario para iniciar el proceso de análisis. Se escribirá un término de búsqueda por el que filtrar el resultado del análisis.
 - Backward: La acción de búsqueda se realizará con datos almacenados tanto en la red, como en la propia base de datos de la aplicación.
 - Forward: Se hará un stream del evento para visualizar en tiempo real el crecimiento de la red y los aspectos más relevantes.
 - API de Twitter: Para la captura de datos, Twitter posee una interfaz de programación de aplicaciones a la que realizaremos las llamadas correspondientes para obtener los datos que necesitamos.
 - Grafo: Debido a la naturaleza de red de las redes sociales, representaremos esta como un grafo. Un grafo es una representación simbólica de los elementos constituidos de un sistema o conjunto, mediante esquemas gráficos.
 - Usuario: Persona que interaccionará con la aplicación.
 - Usuario Twitter: Persona que tiene una cuenta en la red social de Twitter y que interacciona con ella.
 - Seguidor: Usuario que está conectado a otro para poder visualizar su actividad.
 - Tweet: Texto de 140 caracteres por el que se comparte la información en la red social.
 - Muro: Tablón donde aparece toda la información que compartes.



- Hashtag: Palabra especial que sirve como conexión sobre un tema en el que está girando la información
- Tipos de interacciones o Mensajes: Acción especial que ocurre dentro de la red social de Twitter. Se diferencian las siguientes acciones.
 - Retweet: Cuando una información de otro usuario te parece lo suficientemente relevante, puedes compartirla en tu muro. Esta acción no te hace dueño de la publicación, ya que en tu muro aparece el usuario original.
 - Mención: Para dirigir la información a un usuario en especial, puedes mencionarlo para poder entablar una conversación con él.
 - Favorito: Puedes añadir como favorito la información de otro usuario, para indicar que esa información es relevante para ti.
- Acrónimos
 - SGBD: Sistema de Gestión de Base de Datos. Conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos.
 - API: Interfaz de programación de aplicaciones. conjunto de subrutinas, funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.
 - BBDD: Bases de datos. conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso
 - OAuth: Autorización abierta. Protocolo que permite flujos simples de autorización para sitios web o aplicaciones informáticas.
 - WEB: Red informática mundial. sistema de distribución de documentos de hipertexto interconectados y accesibles vía Internet

Visión general del documento

Las siguientes secciones explicarán primero una descripción general del producto a desarrollar, donde se verán desde una perspectiva genérica las principales funciones a desarrollar.

Seguidamente nos apoyaremos en diagramas de casos de uso para mostrar la funcionalidad del sistema, mostrando sus características y sus restricciones. Pasaremos a continuación a una detallada descripción de los requisitos del sistema.

Para finalizar añadiremos aquella información relevante que no ha tenido cabida en las anteriores secciones.

Descripción general

Abordaremos la siguiente sección describiendo todos los factores que afectan al producto y a sus requisitos. Daremos un repaso al contexto, para luego poder definir con detalle los requisitos, que serán mostrados en la siguiente sección. Gracias a esto, será más fácil de entender el porqué de estos.

Dividiremos esta sección en las diferentes partes para poder agrupar los distintos factores que actúan en el sistema y poder concretarlos para su posterior tratamiento. Mostraremos los factores internos, los externos, y aquellos que puedan surgir en el futuro.

Perspectiva del producto

La aplicación de análisis de Twitter que desarrollaremos será independiente de otros productos, ya que nacerá cómo un producto único, pero debido a su naturaleza WEB, será necesaria la conexión con otros sistemas para su funcionamiento.

❖ Interfaces lógicas

- La interfaz WEB de la aplicación actúa como capa para la interacción entre el usuario y la herramienta. La herramienta se encontrará en un servidor al que el usuario se conectará a partir de un navegador que servirá como vía para esta comunicación.
- El sistema necesitará conexión directa con el sistema de base de datos para poder realizar los análisis pertinentes sobre la información almacenada en ella.

❖ Interfaces externas

- La extracción de datos se realizará sobre la red social de Twitter, por lo que tendremos una conexión directa con su API.
Hay que reseñar que para la conexión a la API de Twitter será necesaria la configuración de un protocolo de conexión segura con esta, OAuth.

❖ Interfaces de comunicación

- HTTP: Protocolo de comunicación de la capa de aplicación de red. Nos servirá tanto para hacer peticiones cómo para recibir las contestaciones a estas.
- HTTPS: Protocolo anterior añadiendo una capa de seguridad.
- Lenguaje Cypher: Lenguaje de conexión con el sistema gestión de base de datos para la extracción de la información.

Funcionalidad del producto

Mostraremos ahora un resumen de las funciones, a grandes rasgos, del sistema. Nos apoyaremos para mostrar la información en diagramas de casos de uso en los que se verá de forma directa la idea de la funcionalidad mostrada. Utilizaremos para estos, el programa de dibujo Microsoft Visio.

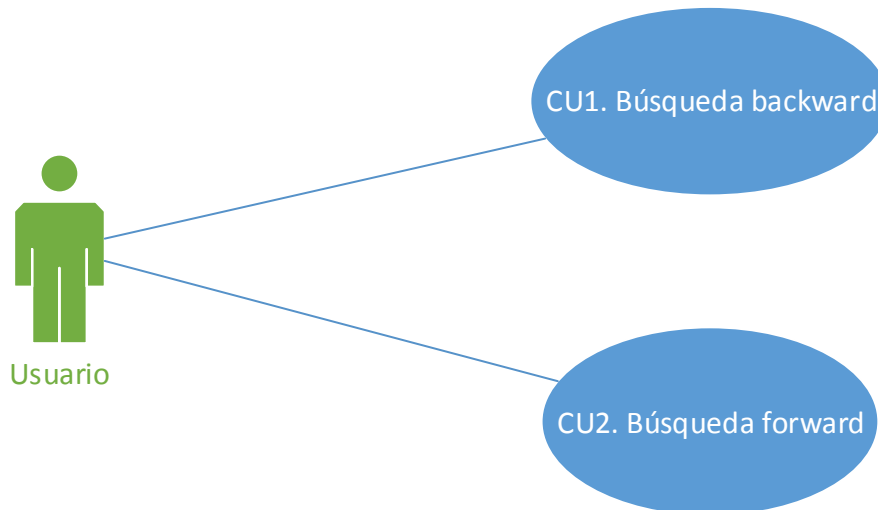


Figura 17: Funcionalidad de búsqueda de evento

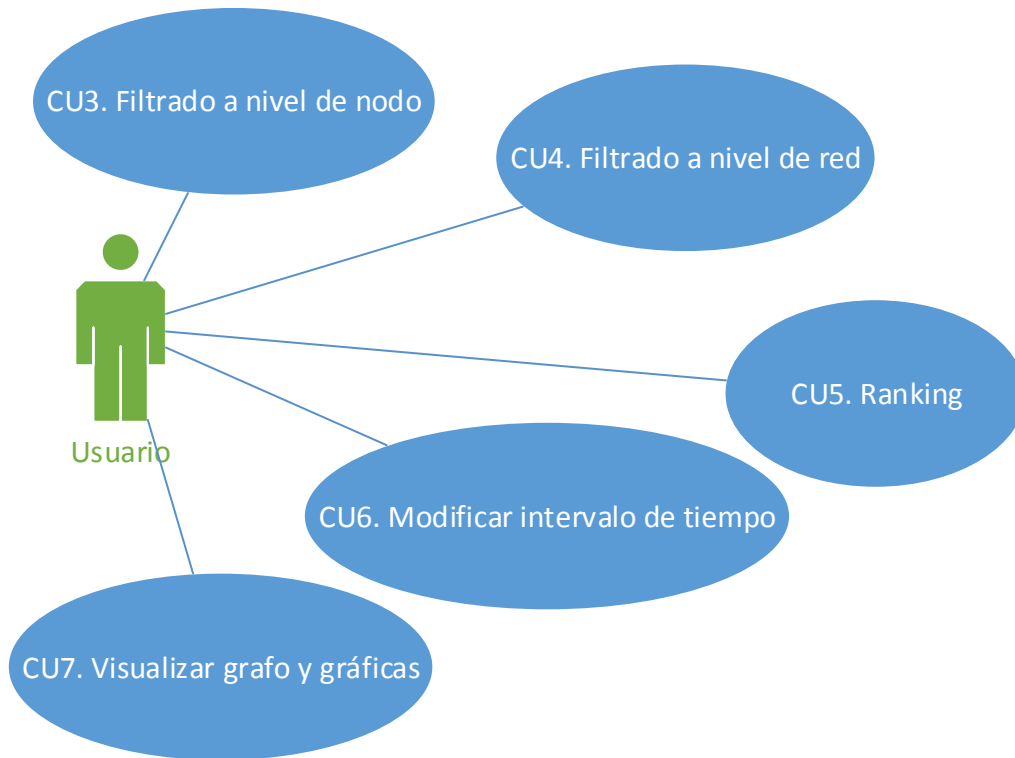


Figura 18: Visualización de resultados para búsqueda backward

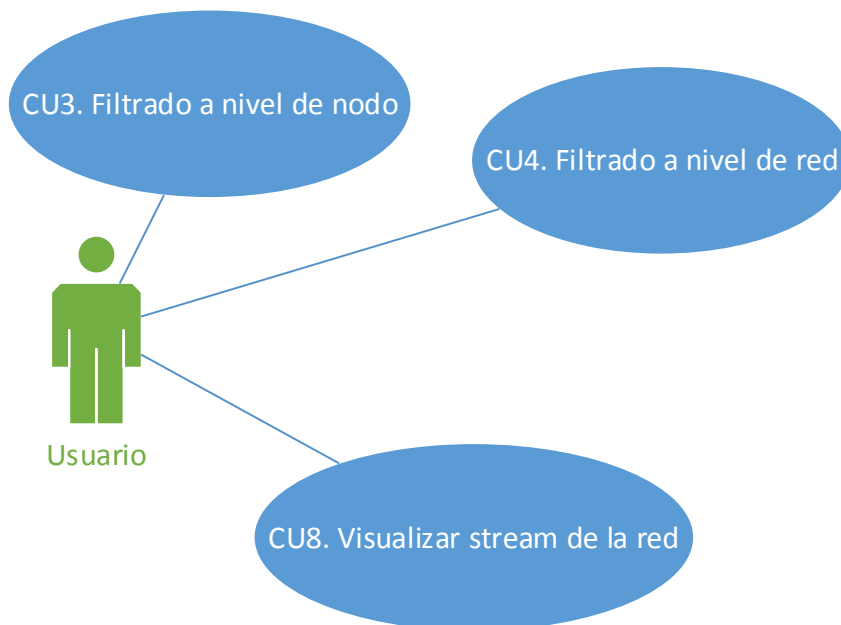


Figura 18: Visualización de resultados para búsqueda forward

La primera elección de búsqueda será determinante a la hora del comportamiento del sistema, aun cuando se compartan gran cantidad de usos. Es dependiendo de esta selección que aparecerá una opción de visualización de la red u otra. La primera se dividirá en intervalos temporales que podrás avanzar y retroceder, y la segunda mostrará automáticamente cómo crece la red.

Características de los usuarios

Para la aplicación, debido a la máxima de simplicidad de cara al usuario, no es requerido distinguir varios tipos de usuario. La autenticación con Twitter será realizada bajo una cuenta propia específica de la aplicación.

En cuanto al tipo de perfil del usuario al que va dirigido, también aplicando la máxima de la simplicidad, sólo será necesario tener conocimientos de la propia red social de Twitter para su uso. Es cierto que vamos a usar métricas que pueden resultar avanzadas para todos aquellos que no tienen un conocimiento propio de teoría de grafos, pero es gracias a la simplicidad y a la idea de mostrar todos los resultados de forma gráfica, que se podrá intuir por dónde empezar el análisis.

Para realizar un análisis detallado y poder evaluar los resultados obtenidos sí que haría falta un conocimiento previo sobre teoría de grafos y centralidad.

Restricciones

Se mostrarán ahora las restricciones que se deben suplir para poder levantar el sistema:

- El ámbito online de la aplicación obliga a estar conectado a Internet para poder realizar la conexión con la red social de Twitter.
- La aplicación debe mostrar correctamente los caracteres especiales, por lo que servirá los contenidos con la codificación de caracteres UTF-8.
- El sistema debe ser capaz de atender interacciones con los usuarios de forma concurrente. También existirá concurrencia cuando se haga la búsqueda forward.
- La aplicación está desarrollada con la última tecnología de lenguajes de programación web. Algunos navegadores no aceptan algunas de estas características, por lo que se tendría que utilizar las últimas versiones de estos.
- Los análisis sobre eventos pasados que no han sido almacenados en la base de datos no podrán ser recuperados de Twitter en el rango de una semana. Esto es debido a restricciones de la propia API de Twitter.
- El lenguaje de programación debe de ser Python, y el framework usado para su desarrollo Django. Por tanto, el servidor WEB debe de ser compatible con estas y actualizados a la versión correcta.

Suposiciones y dependencias

El sistema se ha desarrollado bajo estándares web de última generación, por lo que las dependencias software dependerán siempre de las actualizaciones de estos. Los problemas que podrían hacer fallar la aplicación con mayor probabilidad serían aquellos que surgen de estos.

Si la API de Twitter se actualiza y cambia la forma de acceder a ella, se tendría que modificar la aplicación. También podría surgir este error a la hora de actualizar los componentes de la aplicación. Ya sea por una actualización en el SGBD que modifica el acceso, o una actualización de alguna librería que cambiara la forma de actuar con ella.



Requisitos futuros

Los requisitos futuros se dividirán en dos secciones, las mejoras que se podrían abordar en la aplicación para incrementar la experiencia de usuario y las ampliaciones que se podrían realizar en el futuro.

- **Mejoras:** Se realizarán dependiendo del usuario final cuando se realicen las mejoras en este caso, pero debido a restricciones técnicas y/o temporales, se podrían realizar las siguientes mejoras de cara al futuro.
 - ❖ Estudio sobre la interfaz de usuario. Aunque la pauta en el desarrollo siempre ha sido la simplicidad, se han tomado decisiones que se deberían tener en cuenta a la hora de su mejora. Se aplicarían técnicas de diseño sobre ciertos aspectos, cómo el color, la disposición de los widgets, claridad en textos, etc. Todo ello para aumentar la usabilidad, tanto en facilidades para los usuarios, cómo quitar impedimentos para los usuarios con dificultades.
 - ❖ Mejoras de rendimiento. Se realizará un análisis sobre el rendimiento para detectar aquellos cuellos de botella que podrían estar afectando al sistema y sus posibles soluciones.
 - ❖ Mejoras en la seguridad. Cifrado de datos a la hora de compartir la información por Internet.

- **Ampliaciones:** Desarrollos que servirían para extender el proyecto.
 - ❖ Incluir nuevas redes sociales. Cómo base se utilizará la red social de Twitter para el desarrollo de la aplicación. Pero gracias a la condición que tienen las redes sociales, se podrían hacer ampliaciones para poder utilizar la herramienta en otras redes sociales.
 - ❖ Monetizar la aplicación. Ya se ha explicado que el verdadero potencial de este tipo de aplicaciones ha sido explotar el recurso de la información extraída de las redes sociales para la toma de decisiones en las campañas de marketing. Las empresas apuestan claramente por esta publicidad dirigida, por lo que sería el siguiente paso a realizar una vez la aplicación tuviera el suficiente renombre.
 - ❖ Internacionalización. Traducir la aplicación a diversos idiomas.

Requisitos específicos

Esta sección contendrá los requisitos a un nivel de detalle lo suficientemente explicado para que se satisfagan. Además ayudará a la realización posterior de pruebas que demuestren que el sistema las satisface. Todo requisito aquí especificado describirá comportamientos externos del sistema, perceptibles por parte de los usuarios y otros sistemas.

Interfaces externas

Número de requisito	R.001		
Nombre de requisito	Formulario WEB		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción	La solicitud de datos por parte del sistema al usuario se realizará siempre por formularios web.		

Número de requisito	R.002		
Nombre de requisito	Validación de los formularios WEB		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción	Se validarán siempre los campos de los formularios web, y se avisará al usuario si algún campo no es correcto.		

Número de requisito	R.003		
Nombre de requisito	Comunicación por red		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción	La herramienta estará montada en un servidor con conexión a Internet por lo que será necesaria la habilitación de los puertos necesarios para su conexión.		

Número de requisito	R.004		
Nombre de requisito	Servidor WEB		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción	El servidor web desplegado tendrá que ser compatible con el lenguaje de programación Python y el framework web django.		

Número de requisito	R.005		
Nombre de requisito	Base de datos		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción	La base de datos usada será neo4j. Será necesaria la configuración correcta de la conexión a ella y su autenticación, así como el funcionamiento continuado de ella. El intercambio de información con ella se realizará con el lenguaje Cypher.		

Número de requisito	R.006		
Nombre de requisito	Comunicación con API Twitter		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción	Para la conexión con la API de Twitter será necesario un paso previo de autenticación con ella para poder interactuar. Una vez realizada esta, se podrá atacar su servicio para recuperar los datos que usaremos para el análisis.		

Funciones

Ahora se detallaran ahora los requisitos funcionales del sistema que se habían mostrado en los diagramas de casos de uso de la anterior sección.

Número de requisito	R.007		
Nombre de requisito	Búsqueda forward		
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input checked="" type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción	Una vez introducido y validado un término de búsqueda, el sistema pasará a la siguiente pantalla en la que se podrá observar el stream resultante del término buscado.		

Número de requisito	R.008		
Nombre de requisito	Búsqueda backward		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción	La búsqueda backward se dividirá en dos posibles formas de realizar el análisis: <ul style="list-style-type: none"> • Un análisis en el que se llamará a la API de Twitter para traer los datos de la última semana. Esto es debido a las limitaciones de la propia API. • Un análisis que buscará en la base de datos a partir de un rango de fechas. Una vez elegida una opción, el sistema mostrará el resultado del análisis el término buscado.		

Número de requisito	R.009		
Nombre de requisito	Filtrado a nivel de nodo		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción	<p>Opciones de visualización y filtrado para la visualización del grafo que afectan a los nodos.</p> <ul style="list-style-type: none"> • In Degree: Grado de entrada del nodo • Out Degree: Grado de salida del nodo • Betweenness: Intermediación • Closeness: Cercanía • Eigenvector: Centralidad del vector propio <p>De todas estas se podrán elegir dos que se unirán a un color o a un tamaño para que se muestre en el grafo la métrica.</p>		

Número de requisito	R.010		
Nombre de requisito	Filtrado a nivel de red		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción	<p>Opciones de visualización y filtrado para la visualización del grafo que afecta a toda la red.</p> <ul style="list-style-type: none"> • Nodes: Número de nodos a visualizar • Links <ul style="list-style-type: none"> ○ Retweets ○ Mentions ○ Reply To • Degree • Path Length • Diameter • Clustering • Density • Nodes in Giant Component 		

Número de requisito	RI.011		
Nombre de requisito	Ranking		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción	<p>Listado de los usuarios de la red ordenados según la medida elegida.</p> <ul style="list-style-type: none"> • In Degree • Out Degree • Betweenness • Closeness • Eigenvector 		

Número de requisito	RI.012		
Nombre de requisito	Modificar intervalo de tiempo		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción	En las búsquedas backward, se partirá la visualización del grafo en pequeños intervalos de tiempo para poder ir avanzando o retrocediendo en él. Observando así el crecimiento de la red según el tiempo.		

Número de requisito	RI.013		
Nombre de requisito	Visualizar grafo y gráficas		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción	Para las búsquedas hacia atrás se podrán observar tanto el grafo como las gráficas con los resultados obtenidos.		

Número de requisito	RI.014		
Nombre de requisito	Visualizar stream de la red		
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input checked="" type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción	Aparecerá un espacio en blanco en el que se irá generando el grafo automáticamente según se vayan creando relaciones en la red.		

Requisitos de rendimiento

Número de requisito	RI.015		
Nombre de requisito	Concurrencia		
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input checked="" type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción	Será necesario que el sistema pueda ejecutar simultáneamente procesos de recogida de datos y visualización. Se debe hacer hincapié en esta característica a la hora de la visualización por stream.		
Número de requisito	RI.016		
Nombre de requisito	Tiempo de respuesta		
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input checked="" type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción	A la hora de realizar análisis. El proceso de recogida de datos y posterior tratamiento puede sufrir una ligera sobrecarga en el sistema. En estos casos, el sistema no puede sufrir una penalización de más de 30 segundos para el 95% de las acciones.		

Restricciones de diseño

No existen requisitos o restricciones del sistema aplicables a esta categoría.

Atributos del sistema

No existen requisitos o restricciones del sistema aplicables a esta categoría.

Otros requisitos

Número de requisito	RI.017		
Nombre de requisito	Política de uso de Twitter		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional
Descripción	Se deben aceptar y aplicar todas las obligaciones o limitaciones del uso de la API de Twitter, así como de avisar al usuario para el cumplimiento de las mismas.		

Conclusiones

La mayor dificultad que existe en la especificación de requisitos es plasmar las ideas que tienes en la cabeza en un lenguaje claro, explicado de una forma que cualquier persona lo pueda entender. Esta parte es muy difícil. Nunca se puede plasmar todo lo que se tiene en la cabeza.

En la especificación de requisitos que se ha descrito se ha pretendido dejar clara las tres divisiones que existen en la aplicación. La primera, la imprescindible para arrancar el sistema, que sirve para seleccionar el objeto de búsqueda y cómo actuará el sistema.

Aunque la visualización de la red pueda parecer idéntica, el funcionamiento es completamente distinto. La búsqueda hacia atrás realiza un análisis donde se podrá observar cómo ha evolucionado la red. Teniendo los datos, los resultados sobre estos se pueden corroborar y lo que ha ocurrido será demostrable.

Respecto a las conclusiones sobre las mejoras futuras, estamos ante un proyecto que analiza un tipo software que avanza demasiado rápido. Tal cual como ocurre ahora en la industria del software, crear una herramienta e intentar mantenerla siempre igual sería contraproducente. Al igual que las redes sociales, la aplicación debería continuar desarrollando mejoras y actualizaciones continuas.

Además, la expansión de negocio también es algo que sería propio de esta herramienta. No paran de surgir redes sociales nuevas que en muy poco tiempo se expanden de una manera muy grande. Actuar siempre en ellas sería un modelo de negocio de expansión continua.

Diseño del sistema

Introducción

El siguiente apartado contiene la descripción del diseño del producto. Primero se hará una explicación de la arquitectura del sistema, apoyándonos en modelos para su mejor entendimiento. Luego se hará la especificación formal, donde se definirá el sistema, detallando los artefactos de este.

Especificación conceptual

Mostraremos ahora un resumen de las funciones, a grandes rasgos, del sistema. Nos apoyaremos en el siguiente dibujo, creado en Microsoft Visio, para tener una visión general de este.

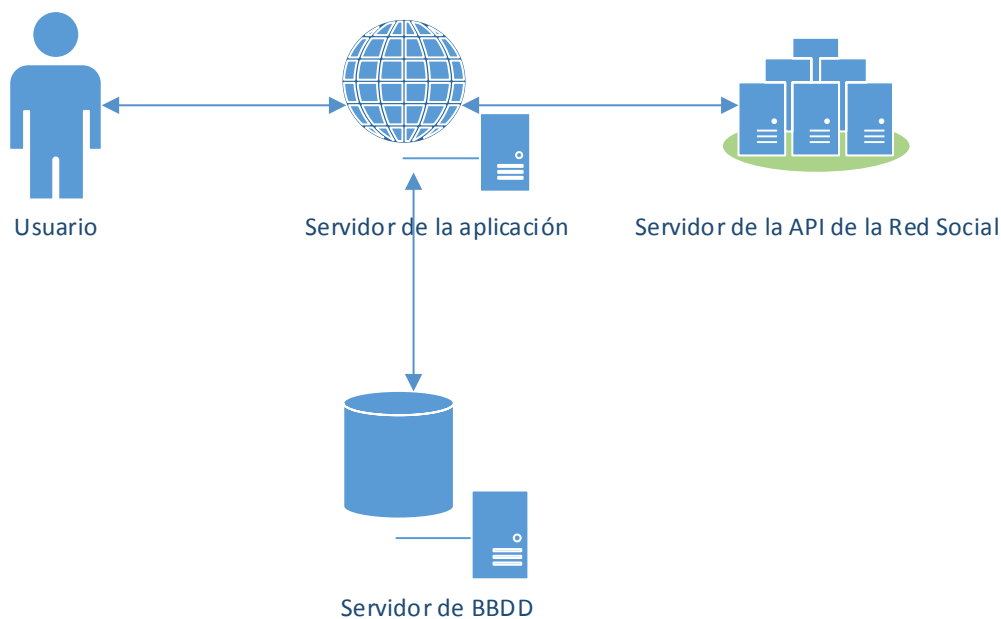


Figura 19: Especificación conceptual

El usuario se conecta al servidor de la aplicación. Según la petición enviada por este, el servidor interactúa con el servidor de la API de la red social para recuperar datos en el caso de que no se encuentren en la base de datos. Una vez los recupera los almacena dentro de la base de datos. Con los datos almacenados en la base de datos se procesan estos para representar la información al usuario, donde se devuelve una respuesta a la petición con estos datos.

Especificación formal

Una vez tenemos la vista general del proyecto, se realizará la especificación de las diferentes partes que componen la aplicación. Para ello utilizaremos una arquitectura por capas que dividirá las partes del desarrollo. Se realizará una capa para la presentación de la información que se relacionará con la capa de negocio. Esta capa de negocio será la lógica de la aplicación y usará los datos de la primera capa para poder trabajar con ellos y pasárselos a la tercera capa. La tercera capa se trata de la interfaz, que describirá la forma de mostrar los datos que se han procesado en el sistema, o las llamadas para poder visualizar estos.

Primero se mostrará con un diagrama de entidad-relación cómo se almacenarán los datos en la base de datos. Seguidamente se mostrará el procesado que se realiza de estos dentro del servidor, apoyándonos aquí de diagramas UML (Lenguaje Unificado de Modelado). Para finalizar, mostraremos diferentes mockups de la aplicación para representar la interfaz de estos.

Persistencia

La capa de persistencia de la aplicación muestra la forma en la que la información se almacena dentro de la base de datos. En nuestro caso, la información no se guarda en una base de datos relacional, la información se almacena en una base de datos en forma de grafos. Por este motivo, las representaciones clásicas para las bases de datos que existen no serían del todo precisas.

En nuestro caso utilizaremos un diagrama entidad-relación, que aunque este pensado como una forma de representar el modelo relacional, para el modelo de grafos que aplicaremos queda lo suficientemente claro.

En el modelo aparecen las tres entidades que generarán el grafo con sus respectivas relaciones. Los atributos de estas se muestran en la figura de la página siguiente debido a su tamaño.

Para realizar los diagramas entidad-relación se ha usado, al igual que en anteriores casos, el software Microsoft Visio, que nos permite realizar los diagramas.

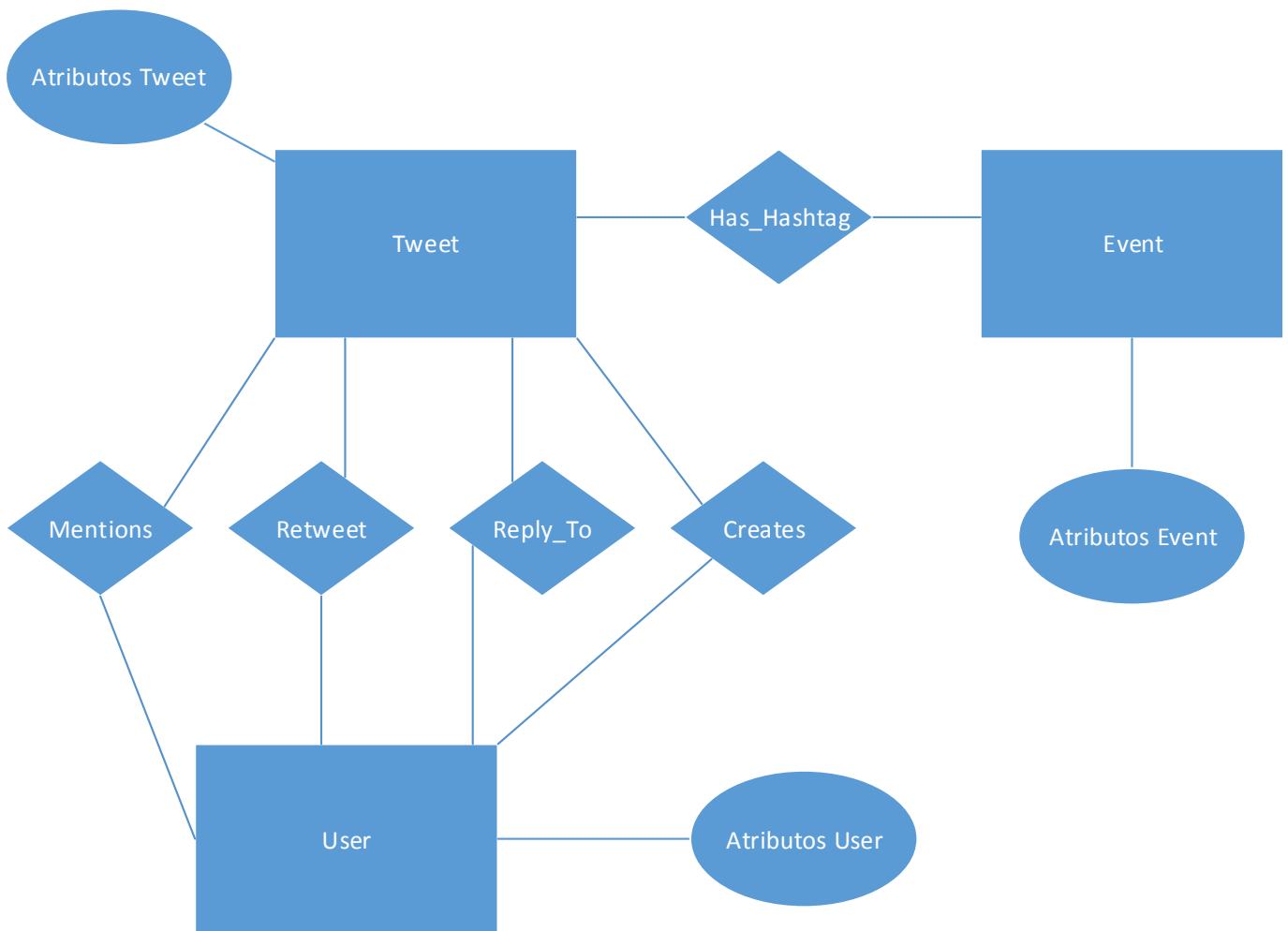


Figura 19: Modelo E/R

Atributos Tweet:

- id
- coordinates
- created_at
- favorite_count
- entities
- user_mentions
- in_reply_to_screen_name
- in_reply_to_status_id
- in_reply_to_user_id
- lang
- place
- retweet_count
- text

Atributos Event:

- hashtag

Atributos User:

- id
- contributors_enabled
- created_at
- default_profile
- default_profile_image
- description
- entities
- favourites_count
- followers_count
- friends_count
- geo_enabled
- lang
- listed_count
- location
- name
- profile_background_color
- profile_background_image_url
- profile_image_url
- profile_link_color
- profile_use_background_image
- protected
- screen_name
- statuses_count
- time_zone
- url
- utc_offset
- verified

Figura 20: Información de los datos a almacenar.

El modelo representado muestra lo fuertemente ligados que se encuentran las tres entidades. Existen una gran cantidad de relaciones entre un usuario y un Tweet dependiendo de las diferentes acciones que se pueden realizar en esta. La relación que se crea entre Event y Tweet es la que nos permite realizar las búsquedas y es la base de todas las demás dependencias.

Por otra parte, la información que almacenamos de cada una de las entidades muestra también la cantidad de información que nos aporta cada una de ellas. Mientras en el caso del usuario podemos almacenar gran cantidad de información perteneciente a él mismo, en el caso de los eventos sólo necesitamos almacenar la palabra clave que nos permita hacer la búsqueda.

Negocio

Para la capa de negocio nos centraremos en aquellas partes relevantes de la aplicación que son propias del proceso dejando de lado aquellas partes que ya vienen como modelo estándar en los diferentes frameworks de desarrollo web. En nuestro caso, y adelantándonos a la siguiente sección, pero que resulta importante destacar aquí para su comprensión, utilizaremos un framework llamado Django que nos ayudará a generar la parte del servidor de la aplicación web. Estos frameworks cuentan con estándares para las diferentes partes de cualquier desarrollo web y que no explicaremos aquí debido a la poca relevancia la sección.

Las partes que trataremos en la capa de negocio de la aplicación se listarán a continuación. Todas ellas con su correspondiente modelo UML, creado en Microsoft Visio, para su visualización.

- **Parte Servidor**

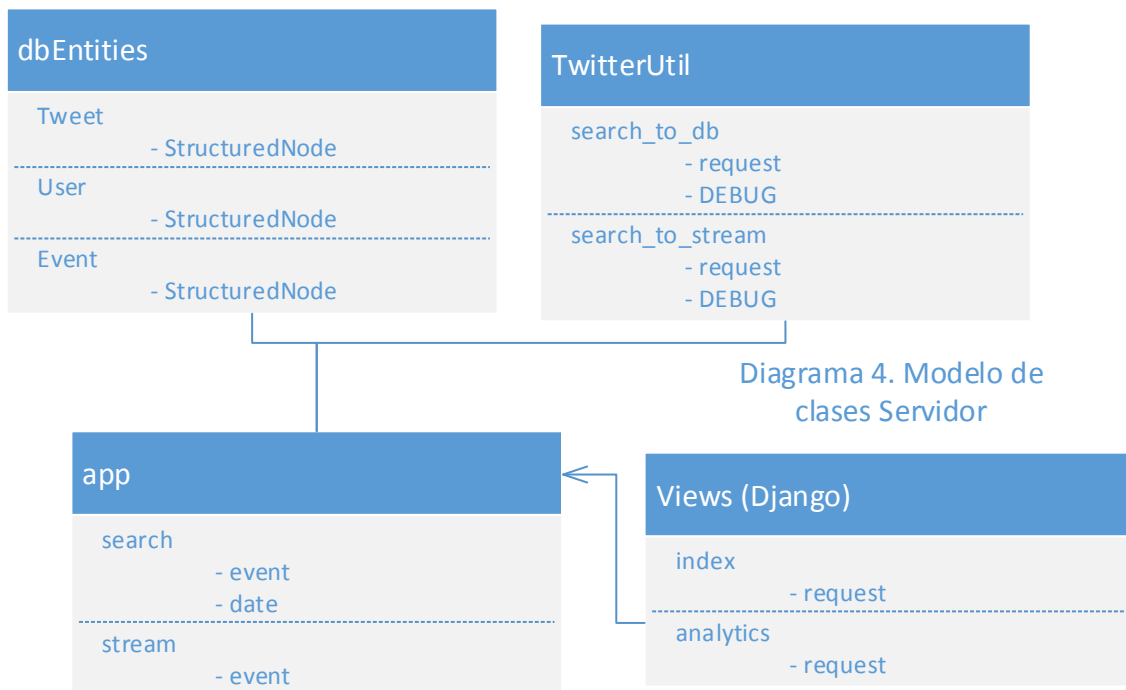


Diagrama 4. Modelo de clases Servidor

La funcionalidad principal de la aplicación se centra en la clase app. Esta clase es llamada por la clase propia de Django Views, que se encarga de devolver la respuesta a una determinada petición. Dependiendo de la opción introducida, este llamará a uno de los dos métodos de la clase app.

La clase app, una vez tiene la información del evento y la fecha, en el caso de que sea una búsqueda, que recupera de la información de la petición, realiza el procesado de esta. Para ello delega las tareas correspondientes a la conexión con Twitter y el almacenado en la base de datos en dos clases.

La clase `TwitterUtil` crea la conexión a la API de Twitter y realiza la tarea asignada. En el caso de la búsqueda devolverá la información en un formato determinado para el tipo de información pedida. Si es una búsqueda en el directorio o es una búsqueda en stream, en la que se irá repitiendo el proceso en background.

Una vez tenemos la información recuperada, realizáramos la llamada a la clase `dbEntities`, que nos ayudará a almacenar esta información recuperada en la base de datos. Con esta información almacenada, generamos el grafo correspondiente y sus respectivas medidas sobre esta información y lo devolvemos a la vista.

Cuando la vista ya tiene los datos preparados genera el HTML con los datos que le hemos pasado para su visualización.

- **Parte Usuario**

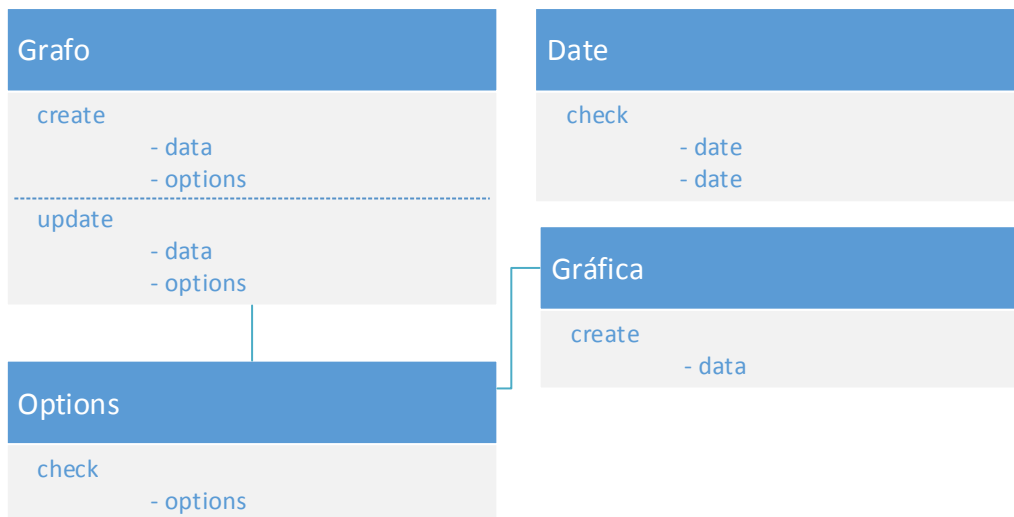


Figura 21: Modelo de clases Usuario

En este caso, dependiendo de donde nos encontremos se utilizará una u otra de las clases listadas.

Para el primer caso, el Grafo, llamaremos a la función de creación una vez se genere el primer HTML de la visualización del análisis. Se cargarán los datos de las opciones con valores predefinidos. Para cualquier modificación de visualización que se realice en este grafo, se llamará al método `update`, que volverá a generar el grafo con las opciones de configuración deseadas.

La segunda opción, relacionada con esta primera, tendremos una clase que se encargará de controlar que todas las opciones que introducimos para la visualización del grafo sean correctas. En este caso tenemos comprobaciones de valores de configuración imposible, o el tener siempre seleccionado dos opciones de visualización máximo.

La clase fecha nos ayuda a comprobar que todas las fechas que introducimos en el sistema sean correctas. Esta función se llama en la pantalla de inicio cuando lanzamos la aplicación para que busque fechas en la base de datos.

Por último tenemos la clase que genera las gráficas del análisis. En este caso se trata de un método en el que se le añade toda la información a mostrar y el genera el gráfico.

Presentación

La capa de presentación mostrará una idea esquemática de cómo se presentará la aplicación al usuario. Nos apoyaremos en bocetos sistemáticos que nos ayudarán a entender cómo funciona la aplicación. En estos bocetos aparecerán también notas que explicarán en pocas palabras las acciones relevantes que se podrán realizar en las diferentes partes de la aplicación.

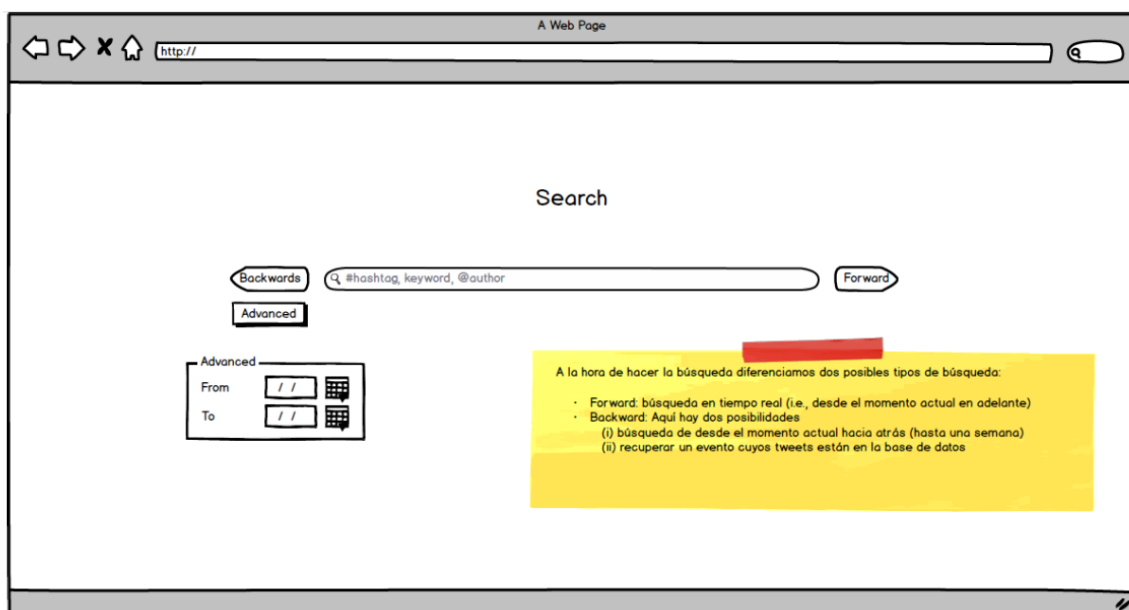


Figura 21: Mockup Pantalla principal

El usuario cuando entre en la pantalla principal tendrá ubicado en el centro el campo para introducir la búsqueda. Esto se encuentra colocado en el centro ya que es el valor en el que se centrará todo el análisis, por eso esta prioridad de colocarlo en el centro.

A los lados de este campo tenemos los dos botones que nos servirán para realizar el análisis. A la izquierda tenemos la búsqueda para eventos que ya han sucedido o que ya se encuentran en la base de datos y a la derecha tenemos el botón para hacer stream. Estos botones se colocan así ya que está establecido socialmente, la izquierda representa el pasado, y la derecha el futuro. Además, los botones se colocan también con forma de señal para acrecentar esa sensación.

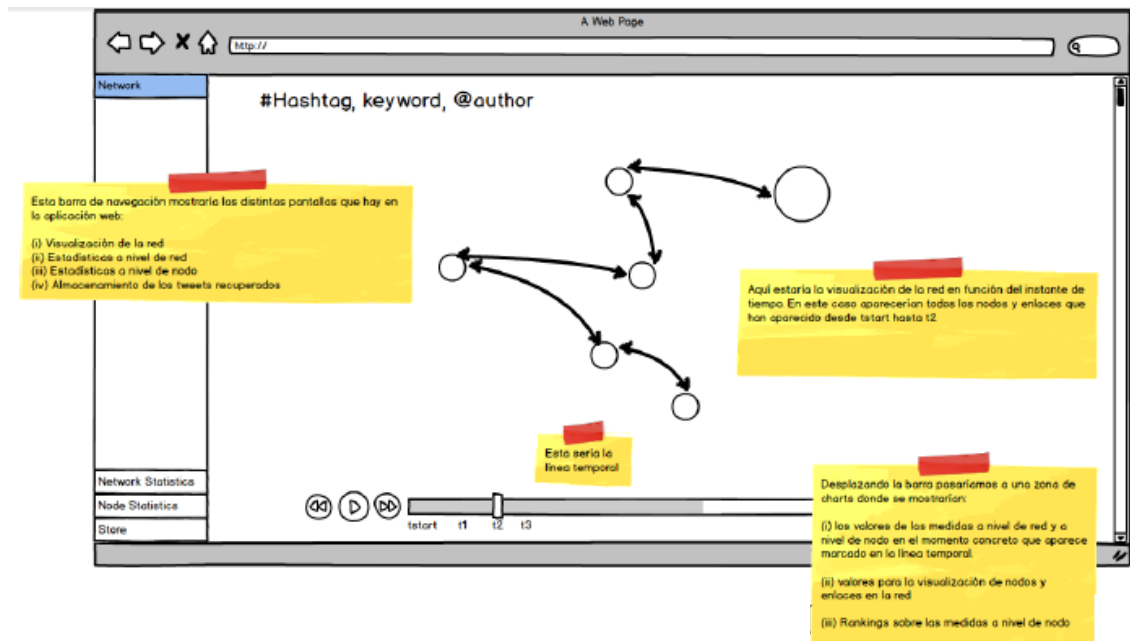


Figura 22: Mockup Visualización grafo

El segundo boceto ya muestra el grafo con la información a mostrar. En este caso ya se ha realizado el análisis y el grafo generado se nos representa en el centro. De nuevo la prioridad de mostrar en el centro las cosas importantes que el usuario necesita de la aplicación.

Debajo del grafo se muestra la barra de avance y retroceso para observar cómo ha sido el proceso de creación del grafo del análisis. Esto implica interacción con el usuario, por eso se muestra con una barra y unos botones de avanzar y retroceder. De nuevo aplicamos iconografía establecida en aplicaciones para que el usuario se sienta cómodo trabajando con la aplicación.

El menú para entrar en las diferentes pantallas que nos aparecen en la aplicación se mostrará a la izquierda de la pantalla, marcando siempre la opción de visualización en la que te encuentres.

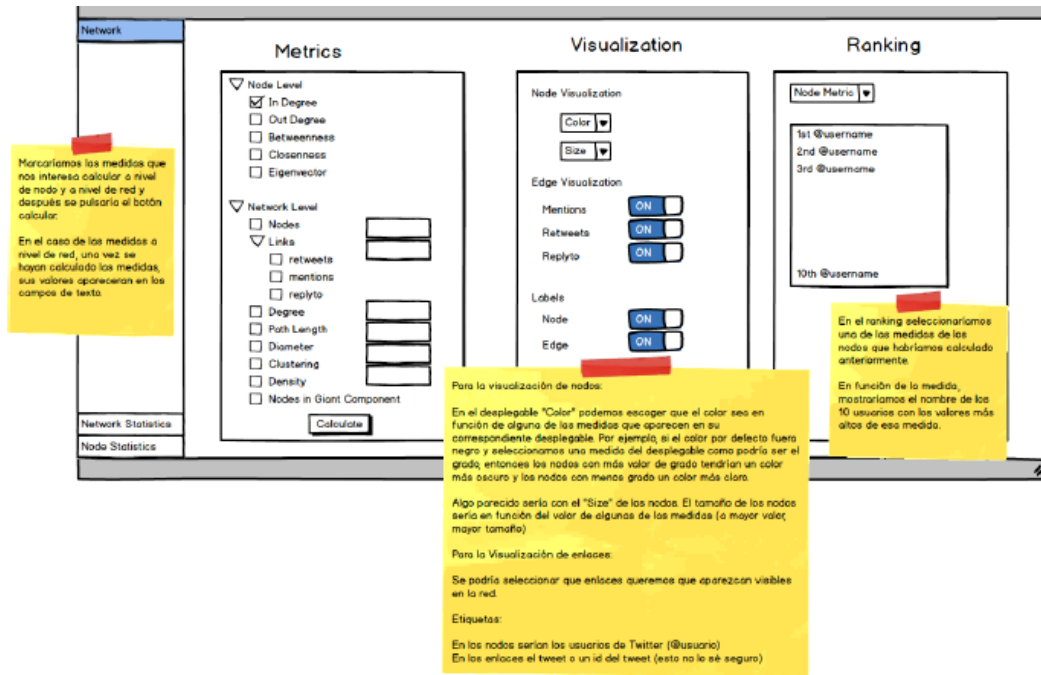


Figura 23: Mockup Pantalla de opciones

El tercer boceto nos muestra la pantalla de opciones de visualización. En este caso tendremos divididas las opciones en tres columnas. La primera de ellas mostrará las opciones que controlan los límites que se pondrán en cada una de las métricas que se visualizarán en el grafo.

La columna del medio mostrará las opciones de visualización del grafo para añadir esa información extra que ayudará a analizar el grafo. Primero tendremos opciones para controlar métricas y observar cuales son los nodos/aristas que se ven más afectados por las métricas. Seguidamente tendremos opciones de visualización de determinados tipos de enlaces y por último opciones de selección para mostrar los textos que aparecen en el grafo, tanto de los nodos, como de los vértices.

La última columna nos ofrecerá un ranking de nodos que podremos filtrar según la métrica elegida.



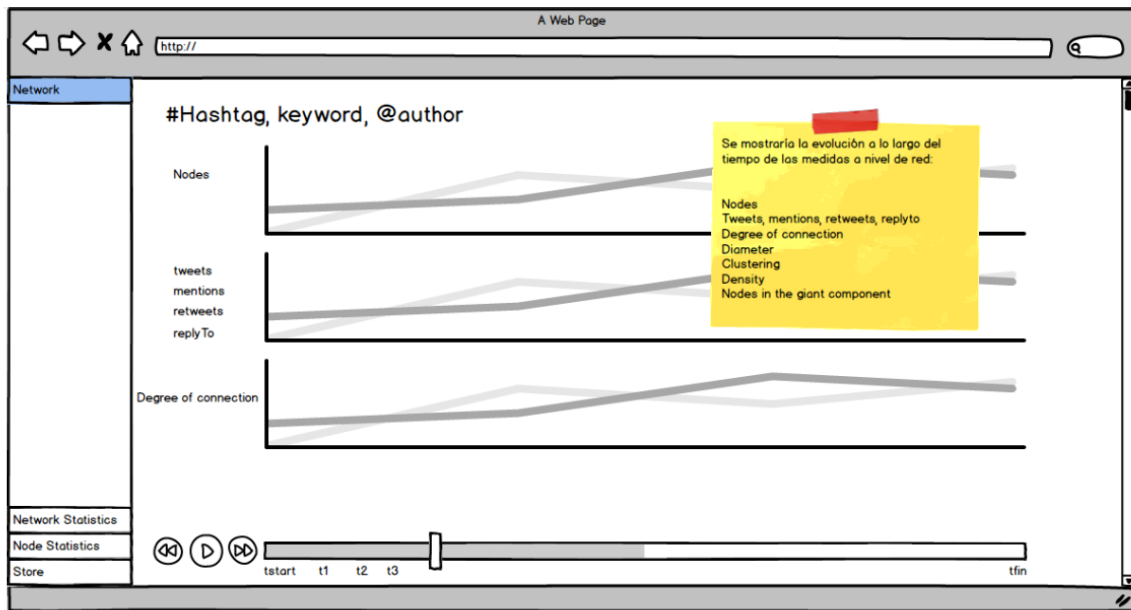


Figura 24: Mockup Pantalla de gráficas

El último de los bocetos nos mostrará las diferentes graficas de las métricas del resultado del análisis.

Dinámica de la aplicación

Una vez mostradas las tres partes en la que se encuentra dividida la aplicación, mostraremos ahora el diagrama de secuencia general de la interacción de los objetos de la aplicación a lo largo del tiempo. Esto nos servirá para poder tener una visión global de la relación entre las diferentes partes que conforman el programa.

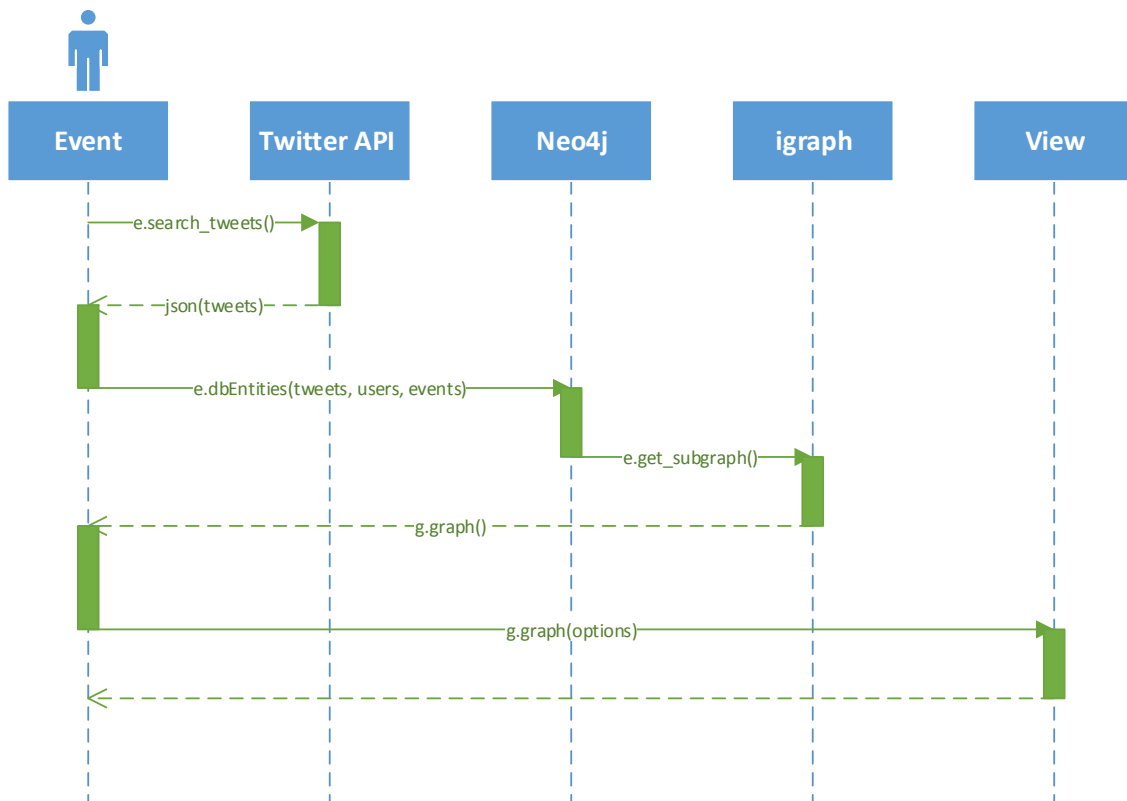


Figura 25: Diagrama de secuencia general

Los eventos, tanto en una búsqueda hacia atrás como una búsqueda en directo, se conectan a la API de Twitter para poder recuperar la información sobre el evento. Este devuelve un mensaje en formato JSON que se transformará al formato propio de la base de datos.

Una vez se han introducido los datos en la base de datos, estos pasarán a su procesamiento para que se devuelva el grafo y sus métricas a la vista para su correcta visualización.

Conclusiones

El desarrollo de una aplicación web conlleva generar una especificación del diseño del sistema muy amplia para poder cubrir todas las características correspondientes a este. Si nos encontráramos en el desarrollo de una aplicación en la que se tuviera que realizar toda la implementación, sería recomendable generar un extenso documento para esta sección. En nuestro caso, y debido a la gran cantidad de frameworks que se encargan de realizar toda esta implementación, nos hemos centrado en aquello que de verdad es propio del sistema.

Esto se ve reflejado muy bien en la capa de negocio. Para el desarrollo de una web, tendríamos que haber identificado desde la llegada de la petición hasta su devolución. Estas tareas ya las realizan los propios frameworks, por lo que nos hemos centrado en el procesado de esta información, que es donde la aplicación difiere de las otras.

Por otra parte, en las capas de persistencia y presentación nos hemos apoyado mucho en diagramas y bocetos para que nos ayuden a visualizar de una forma rápida y sencilla toda la información importante de la aplicación.

Implementación e implantación

Introducción

El apartado de implementación e implantación detallará las tecnologías aplicadas en el sistema y las herramientas utilizadas en la construcción de la aplicación web.

A la hora de tratar estos apartados se hará una definición general de estos haciendo especial hincapié en aquellos puntos relevantes del desarrollo.

Implementación

Para decidir las tecnologías que vamos a utilizar en el proyecto nos fijaremos en las características de este. Al igual que la red social que vamos a analizar, la característica más importante para la toma de la decisión, es la accesibilidad de las propias redes sociales.

Al poder ser accesible desde cualquier dispositivo simplemente con una conexión a internet, hace que la decisión de la tecnología deba ser semejante a esta. Por lo tanto necesitaremos utilizar tecnología de plataforma web.

Tecnología Web

Decidido el ámbito de la tecnología a desarrollar, se detallará el porqué de la decisión de la utilización del lenguaje de programación Python con el framework Django para el desarrollo web.



Python es un lenguaje de programación interpretado que favorece la sencillez del código, haciendo que este sea legible rápidamente. Es un lenguaje ampliamente conocido y utilizado, por eso es fácil de encontrar tanto información, como desarrollos. Gracias a esta cualidad, el desarrollo se basará en frameworks de Python, ya que cubren ampliamente las características del sistema a desarrollar.



El primer framework en el que nos basaremos para levantar el sistema será Django. Django es un framework Web que permite construir aplicaciones web siguiendo el patrón de arquitectura de software “modelo-vista-controlador”.

La definición del patrón se divide en los siguientes tres componentes.

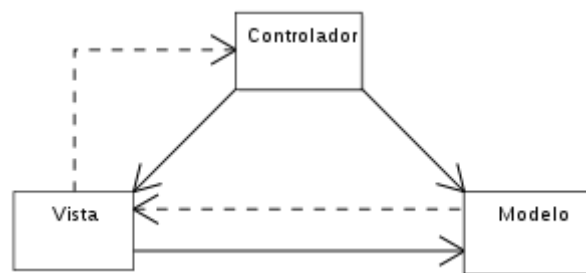


Figura 26. Model-Vista-Controlador

- **El Modelo:** Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.
- **El Controlador:** Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su 'vista' asociada si se solicita un cambio en la forma en que se presenta el 'modelo', por tanto se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo'.
- **La Vista:** Presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho 'modelo' la información que debe representar como salida

Una vez definida la metodología que seguiremos en el proyecto, debido a la elección del framework Django, detallaremos ahora las tecnologías y su uso en cada una de estas capas.

Datos

Para la capa de almacenamiento de la información se utilizará la base de datos Neo4j. Neo4j es idónea para el almacenamiento de la información que vamos a capturar para el análisis de las redes sociales. Esto es debido a que se trata de una base de datos orientada a grafos.



Debido a la cualidad de la información en forma de grafo que necesitamos almacenar, las bases de datos de este tipo son las que mejor se adaptan a estas.

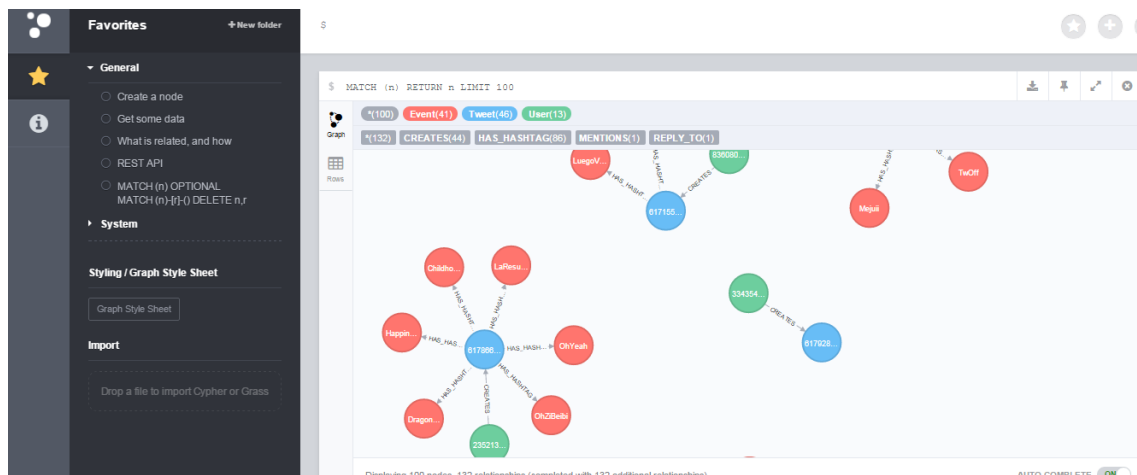


Figura 27. Página principal Neo4j

La base de datos Neo4j, almacena la información estructurada como se guardaría en un grafo. Se tiene la información almacenada en los nodos de los grafos y sus relaciones se almacenan en las aristas del grafo. Cada uno de ellos puede ser de diferente tipo creando así la estructura necesaria para almacenar la información que será usada en el análisis.

Cómo cualquier sistema de base de datos, además de almacenar los datos dentro de esta, será necesario poder recuperarlos. Al igual que el estándar para la selección de datos en las base de datos relacionales es el lenguaje SQL, en Neo4j el estándar para el manejo de los datos es el lenguaje Cypher.

Ejemplo propio del programa desarrollado:

```
MATCH (u:User)-[:CREATES]->(t:Tweet)-[:HAS_HASHTAG]->(event)
OPTIONAL MATCH (u2:User)-[:MENTIONS]->(t)
OPTIONAL MATCH (u3:User)-[:RETWEET]->(t)
OPTIONAL MATCH (u4:User)-[:REPLY_TO]->(t)
RETURN u, t.id, u2, u3, u4, t.created_at
```

Por otra parte, Neo4j es una base de datos implementada en Java, pero podemos acceder a ella gracias a los diferentes drivers que se han implementado en ella para acceder a esta desde diferentes tipos de lenguajes de programación.

El que nos interesa a nosotros es el driver desarrollado en Python. Pero no actuaremos directamente sobre este. Como ya se ha comentado anteriormente, debido a la propiedad de Python de que es ampliamente utilizado, se han desarrollado diferentes frameworks que actúan como intermediario entre la base de datos y Python.



Para el proyecto utilizaremos el “Object Graph Mapper” NeoModel. NeoModel es un desarrollo construido sobre otro framework que sirve también para gestionar el driver con la base de datos Neo4j.

Gracias a este framework definiremos la estructura de la información que queremos almacenar para el análisis de los datos, todo ello escrito en Python y con un estilo de definición de modelo propio de Django.

Para el proyecto se han definido tres tipos de nodos en NeoModel:

- Tweet, que almacenará la información relevante sobre los Tweets descargados de la red social.
- Event, donde se almacenará la forma en que la información se ha generado dentro de la red.
- User, donde se almacenará la información significativa sobre los usuarios de la red social del análisis.

Estos nodos se unirán de la siguiente forma:

- El nodo Evento se relaciona con Tweet gracias a la relación ‘HAS_HASHTAG’
- El nodo Usuario se relaciona con Tweet gracias a las relaciones ‘MENTIONS’, ‘RETWEET’ y ‘REPLY_TO’.

Proceso

La parte del procesado en la aplicación comienza con la introducción de los datos que se usarán para la búsqueda y análisis dentro de la red social. Este procesado, según el tipo de análisis que se requiera activará una u otra opción.

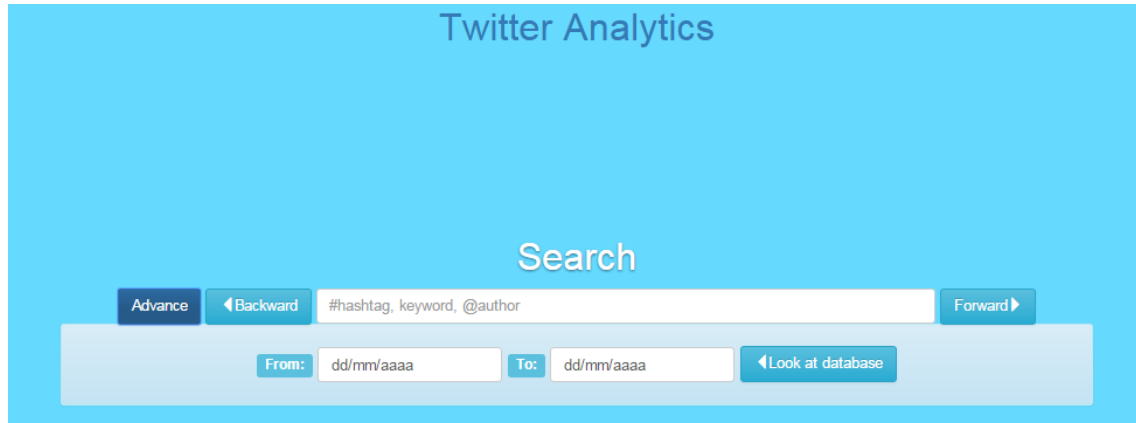


Figura 28. Página principal de la aplicación

La primera opción es la de buscar en la red social a partir de una búsqueda en los datos que ya se encontraban en ella o en su defecto en los datos que ya se encontraban almacenados en la base de datos local de la aplicación.

Cuando se realiza esta función, se activa el primero de los frameworks que se utilizan en esta parte, TwitterAPI. Este framework escrito en Python se encarga de interactuar, al igual que el driver de Neo4j, tanto con la API REST de Twitter, cómo de su versión en stream.



Para hacer uso de este framework será necesario crear el objeto de TwitterAPI con las key generadas por la API de Twitter para poder hacer uso de ella. Para esto será necesario registrar una aplicación dentro de la aplicación de Twitter para que a partir de estas claves puedas identificarte unívocamente, y así poder realizar consultas a su API.

Una vez generada el objeto, podremos hacer llamadas a las distintas funciones que nos ofrece la API de Twitter. En nuestro caso haremos una búsqueda a las funciones rest a partir de la información que se ha introducido para realizar el análisis.

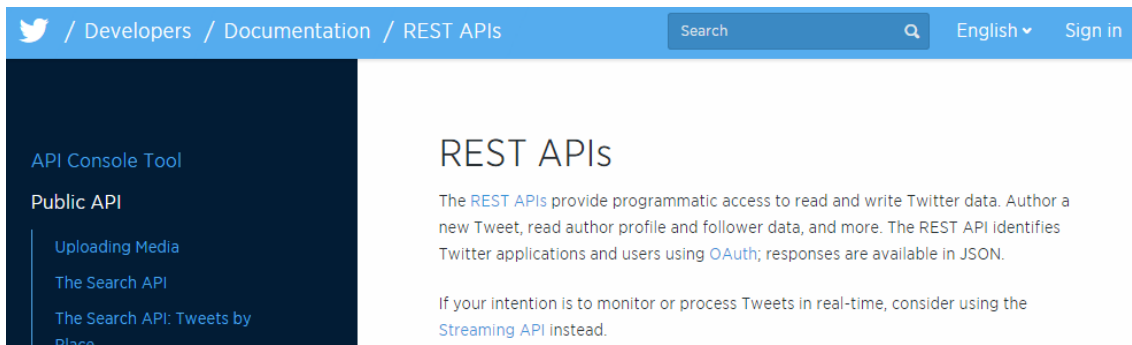


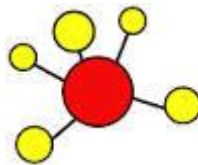
Figura 29. Página de la API de Twitter.

Realizando la consulta recibiremos un JSON que se parseará dentro del propio objeto y que podremos recorrer directamente con los bucles de Python. Gracias a esta característica generaremos, con los datos requeridos, los objetos necesarios de NeoModel y que se guardarán dentro de la base de datos.

Ya con la información guardada dentro de la base de datos. Generaremos los datos del grafo que se mostrará en la pantalla. Para generar este grafo usaremos dos frameworks, uno simple para poder preparar las fechas al formato correcto y otro para generar el grafo propiamente dicho.

El primero de ellos es Python-dateutil, un pequeño framework que nos facilita el tratamiento de las fechas entre los distintos formatos que existen. En este caso se utiliza para hacer parseos de fechas en las búsquedas a la base de datos y para, de nuevo, parsear las fechas que te devuelve la base de datos a las diferentes partes de la interfaz, para su interpretación y su muestra.

Para poder generar el grafo a partir de los datos recuperados de la base de datos y aplicarle las diferentes medidas seleccionadas se utiliza el framework para el tratamiento de grafos Igraph.



Igraph es un framework escrito en diferentes lenguajes de programación, entre ellos Python. Genera de forma interna un grafo con los datos que le suministras y con el que puedes luego realizar operaciones. En nuestro caso, todas las medidas de centralidad se pueden realizar gracias a la API del framework, realizando su correspondiente llamada.

Por ejemplo, la siguiente llamada guarda en el array “in_degree” el valor de entrada de los nodos correspondientes, para el grafo “g”.

```
in_degree = g.degree(type="in")
```

Por otra parte, si en la elección que hemos tomado a la hora de visualizar los datos es la visualización en stream se realizarán los pasos detallados a continuación.

Al igual que ocurre con el anterior caso, el framework TwitterAPI nos permite crear una conexión con la API de Twitter para realizar streams. TwitterAPI genera una instancia de conexión en la cual se reciben eventos que activan funciones definidas. Estas funciones llevan el código a ejecutar según se reciba el evento correspondiente de Twitter.

Lo que se hace en este caso es mantener esta conexión activa en segundo plano mientras se van actualizando los datos de la base de datos mientras el usuario sólo ve cómo se van actualizando los datos que se muestran.

Todos estos datos del stream se almacenan en la propia base de datos para poder realizar a continuación el análisis en éstos.



Interfaz

La interfaz para el proyecto será Web, por lo tanto esta parte se dividirá en los tres lenguajes de uso de los desarrollos de aplicaciones web: HTML para generar la estructura de la página, CSS para cambiar el estilo general así como de los diferentes componentes y, por último, Javascript, para incluir funcionalidad a la aplicación.



Nos centraremos en cada uno de ellos, ya que se tienen diferentes características. El primero que detallaremos será el lenguaje HTML. Es un lenguaje de etiquetado que nos permite estructurar y ordenar los diferentes componentes que se encontrarán en la página. En este caso, Django posee un motor de plantillas que nos facilita el desarrollo del código a partir de unas pautas.

El lenguaje es repetitivo en la manera de que cada página distinta de la aplicación Web repite el mismo esquema. Para ello se identifican las diferentes partes que se repetirán en la aplicación y Django, a partir de una nomenclatura especial será capaz de interpretar y autogenerar código repetitivo.

Además de esto, otra de las ventajas de utilizar el sistema de plantillas de Django es el añadido de funcionalidad propia de los lenguajes que no se encuentran en HTML. Cómo podría ser el uso de bucles, comparaciones o asignaciones. Esto hace que se puedan generar campos rellenos con la información de la base de datos, eligiendo el momento en el que se quiere mostrar.



Para la parte del estilo de la web se utilizará el framework Bootstrap. Este framework, desarrollado por los propios programadores de Twitter, nos facilita en gran medida esta parte.

Bootstrap se basa en un sistema de columnas para poder colocar y dimensionar los diferentes componentes que se han definido en el HTML de la aplicación web. El propio framework es capaz, gracias a este sistema de columnas, de adaptar el estilo a todos los diferentes tamaños de dispositivos que se encuentran actualmente.

Por otra parte, el framework también dispone de una gran variedad de elementos HTML adaptados a un estilo moderno, muy parecido al que nos ofrece la página web de Twitter. Además, debido a que se trata de un framework ampliamente utilizado y conocido, existen numerosos desarrollos que amplían la funcionalidad de este, adaptándose a cualquier tipo de desarrollo.



Por último, para el desarrollo dinámico de la web se hará uso de dos frameworks, jQuery y d3.js. jQuery es una biblioteca que se carga una vez generada la página web y que nos permite acceder a todos los elementos de la página web de una forma mucho más sencilla que el tradicional JavaScript.

Muchos desarrollos para añadir funcionalidad a las aplicaciones web parten con la base de jQuery, como por ejemplo la parte dinámica de Bootstrap. Que utilizaremos para realizar acciones en diferentes componentes de la aplicación, como pueden ser botones o filtros.



El framework d3.js que utilizaremos también en esta parte nos servirá para generar en un marco de dibujo el propio grafo. d3.js nos permite generar y actualizar según las acciones un grafo interactivo que se genera a partir de los datos leídos de la base de datos. Podemos cambiar el color y el tamaño de los nodos y las aristas para poder visualizar las diferentes medidas que existen en el análisis. Creando así una forma fácil de visualizar esta información.

Herramientas

Definidas todas las tecnologías para el desarrollo de la aplicación, en esta sección trataremos sobre las herramientas que utilizaremos para poder manejar, de la mejor forma posible, todas estas.

Para un proyecto de esta envergadura lo recomendable en estos casos es elegir un IDE (Ambiente de desarrollo integrado). Un IDE es una aplicación informática que proporciona servicios integrales para el desarrollo de aplicaciones. Gracias a este software tendremos gran cantidad de opciones integradas, por ejemplo análisis inteligente de código u opciones de debug.

El primer criterio será elegir un IDE que trabaje con lenguaje principal que hemos seleccionado, Python. En cambio, ya que Python es ampliamente utilizado en sectores web, seleccionaremos un IDE que además nos ayude en el desarrollo de las tecnologías web.



El IDE que utilizaremos por tanto será PyCharm, que soporta todos los lenguajes web utilizados y además está completamente integrado para desarrollar aplicaciones con el framework Django.

Evaluación

En la parte de evaluación se detallarán las pruebas que se han realizado en el desarrollo para la solución de errores así como poder asignar los límites propios de la aplicación. No es posible garantizar que un software o sistema falle jamás, pero al realizar estas se pretende que de estas, se disminuyan los posibles riesgos que se generen.

Para explicar esta parte realizaremos un plan de pruebas donde dividiremos las pruebas desde aspectos más específicos, pruebas directamente en código fuente, a más generales, analizando las salidas del sistema a determinadas entradas.

En nuestro proyecto, debido a la falta de tiempo, no se incluye el uso de herramientas de pruebas unitarias. La creación automática de pruebas unitarias nos ayudaría a tener un control sobre la aplicación a la hora de realizar mejoras posteriores. Sería recomendable su implantación en posteriores versiones.

Caja blanca

Para la primera parte se realizarán las pruebas en el código fuente de la aplicación, verificando parte por parte que las líneas funcionan según cómo se estaba definido.

Con estas pruebas intentaremos garantizar que los flujos de decisión que se toman en la aplicación se cumplen y además observaremos que no se superen los límites que se han especificado.

Para esto simularemos las clases que se han escrito introduciendo datos de prueba para simular el proceso que seguirían los datos. En el caso de que existieran datos que no corresponden con los esperados, se tendría que ver la causística del error, y realizar un análisis de este.

Para este caso nos podemos ayudar de la función de debug que nos ofrece el IDE PyCharm. Esta función nos ofrece paso a paso todos los datos que en ese momento están afectando al sistema. Pudiendo, con esta información, identificar donde ocurre el error y el porqué de este.

Ejemplo de una prueba de caja blanca sobre el código fuente de la función de almacenado

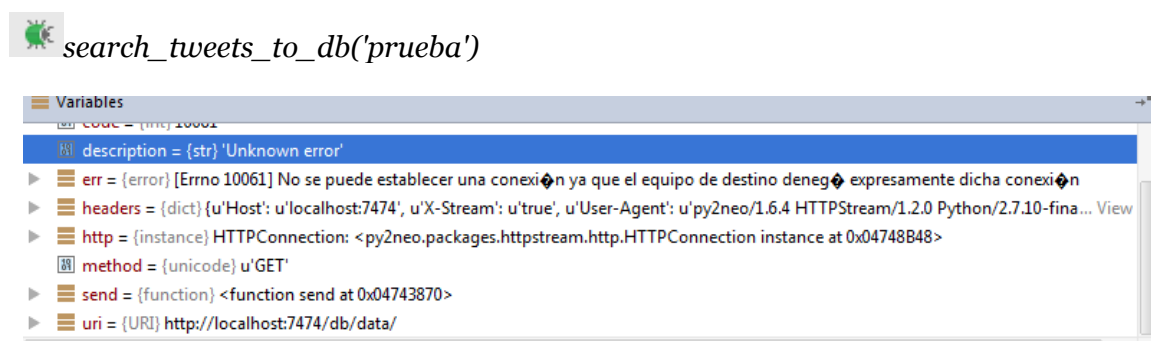


Figura 30. Modo Debug aplicación

Caja negra

Las pruebas de caja negra son aquellas en las que no nos fijamos en las partes específicas del programa, sino que visualizamos el sistema de una forma general. Nos volvemos a fijar en las entradas y en las salidas de estas, pero estas pruebas no analizan las partes de un determinado programa, si no que simulan un ambiente general.

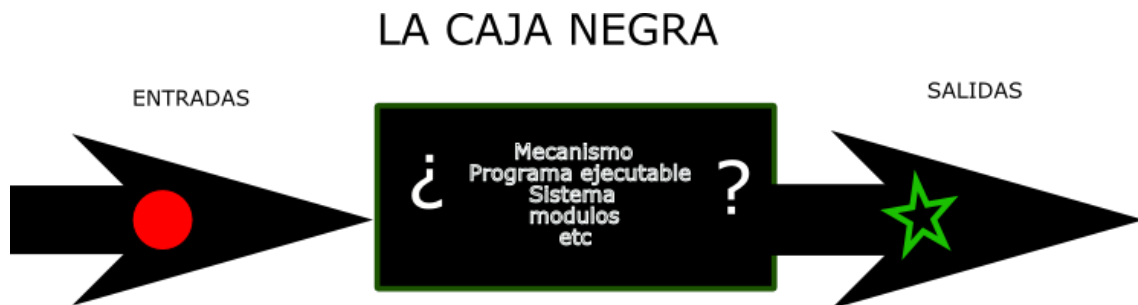


Figura 31: Caja negra

Se trabajan ya sobre las interfaces gráficas, intentando emular el que sería un usuario de la aplicación. Estas pruebas se dividen a su vez en dos. Primero haremos pruebas con entradas que son correctas y por lo tanto darán resultados que se pueden predecir y probar. A continuación haremos pruebas con entradas erróneas. Estas entradas erróneas tienen que estar controladas por el sistema, devolviendo en sus salidas los correspondientes avisos.

A continuación ponemos un ejemplo de datos erróneos, en este caso sobre rangos de fechas imposibles.

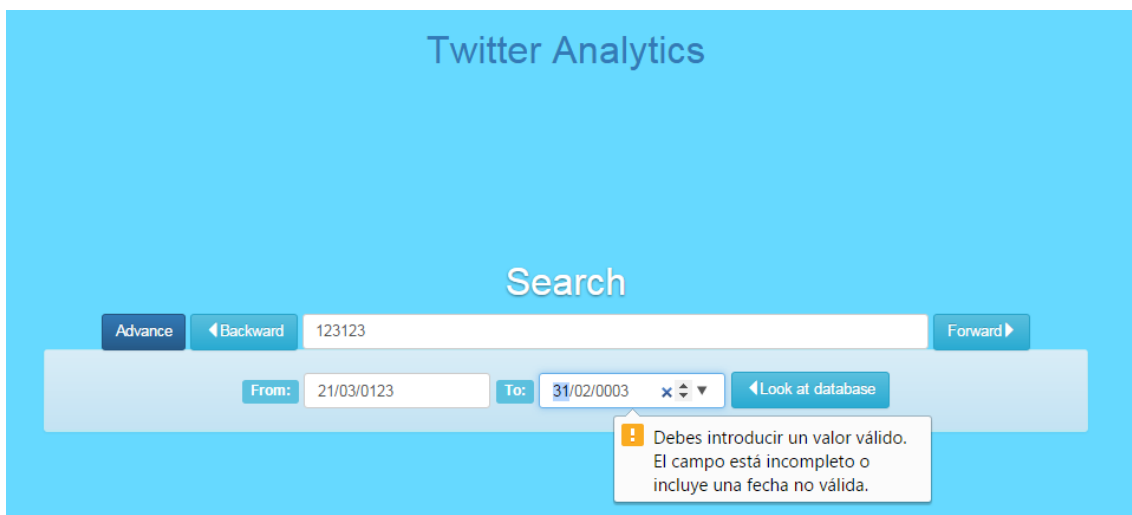


Figura 32: Error en fechas

Implantación

En el apartado de implantación se tratarán los temas que corresponden a la instalación de la aplicación dentro de cualquier sistema. Primero se hablará sobre los requisitos necesarios y luego se realizará una guía sobre los diferentes pasos a realizar.

Una vez instalado se hará un repaso sobre las pruebas a realizar para comprobar que el software se ha instalado correctamente, y se detallarán diferentes errores comunes que pueden ocurrir si no se ha instalado correctamente.

Instalación

Para la instalación de la aplicación serán necesarios una serie de requisitos mínimos para que la aplicación funcione de una forma correcta. El principal requisito al tratarse de una aplicación web es disponer de un servidor. Sería necesario un servidor físico conectado a la red y disponible las 24h, así como el servidor web que realizaría el proceso. Del lado del cliente sería necesario tanto una conexión estable con este servidor, cómo un navegador web para interpretar los diferentes sitios webs para su visualización.

En nuestro caso, para realizar el desarrollo, se instaló directamente el servidor en la propia máquina para acceder a él de forma local, ganando así ventajas en velocidad y acceso total a todas las partes de la aplicación.

Para poder levantar el servidor en modo local Django dispone su propio servidor de desarrollo. Este ligero servidor, cómo se dice en la documentación oficial de Django, sirve solo como servidor de pruebas para el desarrollo. Este servidor no contiene características propias de los servidores web, por lo que no se nombrará en la instalación.



A continuación se lista el proceso de instalación de la aplicación:

1. *Instalación de Neo4j*

Neo4j tiene un instalador disponible en su página web (<http://neo4j.com/download/>). Dispone de dos versiones, una Enterprise para su uso en ámbito comercial, y la versión Community, ya sea para aprender o el desarrollo aplicaciones de uso propio.

2. *Configuración de Neo4j*

Una vez instalada la base de datos, al iniciar la aplicación nos aparecerá la ventana para lanzar la base de datos.

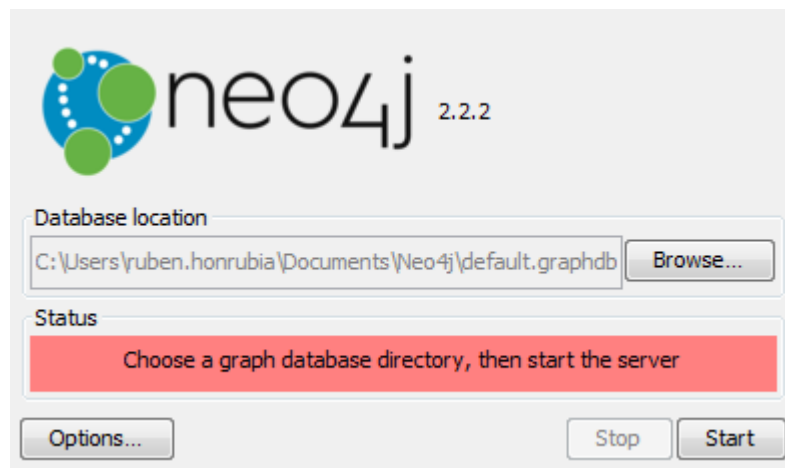


Figura 33: Panel de control Neo4j

Podremos seleccionar la localización de la base de datos seleccionando el botón Browse, configurar las opciones del programa pulsando el botón Options e iniciar/parar el servidor con los botones Start/Stop.

Existen gran cantidad de opciones de configuración de la base de datos, pero nos centraremos en una opción no aceptada por NeoModel y que aparece en las nuevas versiones de Neo4j.

En las opciones de configuración del servidor buscamos la opción:

```
# REQUIRE (OR DISABLE THE REQUIREMENT OF) AUTH TO ACCESS NEO4J
DBMS.SECURITY.AUTH_ENABLED=TRUE
```

Esta configuración por defecto no es aceptada por NeoModel ya que este no tiene todavía implementada la lógica de autenticación en la base de datos. Por lo tanto, colocaremos este flag a false.

3. *Instalación de Python*

En la página de descarga del instalador de Python (<https://www.python.org/downloads/>) lo primero que aparece son las dos versiones de Python. A la hora de instalar Python en la máquina es identificar que versión utilizar. Python mantiene dos versiones que difieren una de otra. Tenemos la versión 2 del lenguaje y la versión 3. En nuestro caso, y debido a compatibilidad con los paquetes que a utilizaremos, seleccionaremos la versión 2.

Una vez tengamos instalado Python en la máquina, podemos ejecutar el siguiente comando en la consola y comprobar la versión que tenemos instalado de Python.

```
C:\PFC\Python27>python --version
Python 2.7.10
C:\PFC\Python27>
```

4. *Instalación de paquetes Python*

a. **Django**

Para instalar Django utilizaremos Pip, un gestor de paquetes que viene incluido en la instalación de Python.

Para ello ejecutamos la siguiente instrucción en la consola:

```
“pip install Django==1.8.4”
```

De nuevo podemos comprobar la versión instalado con el comando dentro de la consola de Python:

```
“print(django.get_version())”
```

Que nos devolverá el resultado: 1.8

b. **TwitterAPI**

De nuevo instalaremos el paquete con el gestor de paquetes Pip con la siguiente instrucción:

```
“pip install TwitterAPI”.
```

c. **Neomodel**

Instrucción en Pip:

```
“pip install neomodel”
```

d. **Dateutil**

Instrucción en Pip:

```
“pip install Python-dateutil”
```

e. **iGraph**

En este caso tendremos que tener en cuenta que la instalación a través de Pip no funciona en el sistema operativo Windows.

En el caso de realizar la instalación en Linux o en Mac OS X, se utilizará la siguiente instrucción:

```
“pip install Python-igraph”.
```

Para el caso de Windows, la instalación a través de Pip fallará a mitad. Por ello, para poder instalar igraph en Windows será necesario descargar un instalador desde la siguiente dirección e instalar manualmente el paquete.

<https://pypi.python.org/pypi/igraph/0.1.8>



5. *Instalación de servidor web*

Django sugiere levantar el servidor de tres formas:

- Usando WSGI
- Usando FastCGI, SCGI o AJP
- mod_python

Podemos utilizar cualquiera de ellas para que la aplicación funcione correctamente, pero cómo bien se explica en la documentación de Django, sería recomendable utilizar las primeras dos, ya que la mod_python se encuentra obsoleta.

6. *Instalación de archivos de aplicación*

El proyecto necesita de una estructura de ficheros específica que te genera Django para que este funcione. Esta estructura ya viene creada en los archivos de la aplicación. Se copiarán los ficheros en la carpeta raíz del servidor y se levantará el servidor a partir de esta.

7. *Configuración final*

Para el paso a productivo de la aplicación es necesario cambiar la configuración que viene por defecto en los archivos de configuración. Estos archivos de configuración están preparados para un entorno de pruebas antes del movimiento a productivo por lo que es necesario saber identificar estas opciones.

Estas opciones de configuración se encuentran dentro del fichero settings.py. Este fichero almacena las opciones de configuración de la aplicación web Django. Para cambiar las opciones de prueba buscaremos las siguientes líneas en el fichero:

```
DEBUG = True
TEMPLATE_DEBUG = True
```

Cambiamos estas opciones y las colocamos a False. Una vez cambiadas podremos levantar el servidor.

Pruebas

Para comprobar que todo se encuentra correctamente instalado realizaremos un recorrido por las diferentes partes de la aplicación forzando que se ejecuten todas las partes de esta. Para ello iniciaremos el servidor, la base de datos y accederemos a la dirección web correspondiente.

Una vez dentro de la pantalla inicial nos debemos encontrar en la pantalla de búsqueda. En el caso de que esta pantalla no aparezca deberemos fijarnos en los logs generados por Django que nos darán detalles del error. El error más común que nos puede ocurrir en esta parte es que la base de datos no se encuentre disponible, por lo que se tendrá que averiguar si es posible que hayan problemas de conexión a esta, o que la base de datos no se encuentre bien instalada.

Si la pantalla inicial ha cargado correctamente, se harán pruebas en la interfaz. En el caso de que la pantalla se encuentre con texto, pero este aparezca sin estilo (la forma más fácil de verlo es que no hayan colores), será resultado de que el CSS de la aplicación no se ha cargado correctamente, por lo que habrá que asegurarse que las rutas de acceso a estos ficheros sean accesibles por la aplicación.

Una vez realizada la prueba del estilo en la página web, se debe realizar la prueba de la lógica de la aplicación web para comprobar que los ficheros javascript se han cargado también correctamente. La mejor forma de hacer esta prueba en la aplicación es seleccionando la opción *advance* para que nos aparezca la selección de fechas. Si esto ocurre, los archivos de javascript se habrán cargado correctamente. En el caso de que no hiciera nada, se tendría que revisar que el acceso de la aplicación a estos archivos es posible.

La siguiente prueba será el procesado por la parte del servidor, para ello escribiremos una url que contenga errores, como por ejemplo `<url_servidor>/analytiERROR`, para que el servidor nos devuelva un error 404 avisando de que la dirección no es correcta. En el caso de que aparezca la ventana de debug de Django, tendremos que quitar las opciones de prueba que estaban en el archivo `settings.py` y reiniciar el servidor.

Para finalizar, crearemos un evento de prueba que nos permitirá hacer una análisis rápido de Twitter para observar el procesado de la información, así como el guardado y acceso a la información en la base de datos. Si el resultado es correcto, nos aseguraremos de que se hayan pasado por todos los paquetes necesarios de la aplicación. En el caso de que se encuentren errores en este, se tendrían que revisar los logs generados por Django para encontrar el error.



Conclusiones

El desarrollo de una aplicación web, debido al amplio abanico de tecnologías utilizadas, es una tarea en la que tienes que ahondar en muchos ámbitos distintos. Empezando en las tecnologías de servidor, tienes que pensar cómo funciona la red. Tienes que resolver peticiones en un tiempo relativamente corto, para que el usuario que está utilizando en ese momento la aplicación, no piense que se ha colgado.

Una vez que tienes claro esto tienes que devolver en la propia respuesta toda la información necesario para que el usuario sepa lo que está viendo. Para el desarrollo de una web no vale enviar un texto plano. Este texto que le envías tiene que tener una visualización cómoda y sencilla para que el usuario la entienda. Pero ya no sólo eso, también tiene que interactuar con ella.

Cada una de estas características es una tecnología diferente. En cada una de ellas tienes que pensar en ese ámbito en el que se rodea. En el caso del CSS, para dar estilo a la web, estamos hablando de un aspecto tan grande que existen cursos sólo para ello. Y ya no solo de programación, sino de otros ámbitos, mucho más orientados a estos.

El mantenimiento de un servidor web, las rápidas actualizaciones de cualquier tecnología que utilices. Todo esto también está relacionado con la creación de una web, no es sólo crearla y ya está. Alrededor de estas existe un mundo que no se explica en el siguiente documento, pero que sería obligatorio tener en cuenta.

Conclusiones

Dificultades

Aunque ya se ha comentado en anteriores secciones, la dificultad más importante en la creación de una aplicación web es la gran diversidad de tecnologías que te encuentras en el desarrollo.

Todas estas tecnologías se encuentran divididas en secciones bastantes claras que representan perfiles muy distintos. Cuando te decides a crear una página web no puedes simplemente centrarte en la programación, debes ampliar tus horizontes hacia otras ramas. Tienes que pensar, por ejemplo, cómo un diseñador gráfico, que tiene que mostrar la forma más sencilla de mostrar la información para que el usuario la entienda.

Es difícil saber siempre cuál será la mejor forma de realizar un desarrollo de estas características, en las que cada parte está separada pero a la vez unida del resto. En mi caso, es cierto que no habré elegido siempre la opción correcta a la hora de diseñar el proyecto, pero es bueno ver que cuando te decides a crear un proyecto de estas características, empiezas a tener claros los diferentes puntos de vista que existen.

Por otra parte, dificultades en el aprendizaje de nuevas tecnologías he tenido, pero eso es algo que pasará siempre. Aprender Python o JavaScript, lenguajes tan distintos por primera vez hace que resulte tedioso el tener que avanzar el proyecto y no saber muy bien cómo.

Aprender cómo sacar información valiosa de las redes sociales, todo ello a base de modelos matemáticos. Es cierto que no he tenido que profundizar en los modelos, pero sí saber cada uno a que se refieren y que aportan.

Pero se consigue a base de esfuerzo, de pelearse día a día. Viendo en perspectiva todo lo que he recorrido me hace sentirme generalmente contento con el proceso de aprendizaje. Hemos elegido una carrera que es así, siempre te sorprenderá y siempre tocará seguir aprendiendo cosas nuevas.



Aportaciones

Tanto el proyecto como la redacción de este documento me han aportado gran cantidad de conocimientos de cara al futuro, tanto técnicos como personales.

He aprendido nuevas tecnologías, aunque ya sea sólo la base. El desarrollo web es un mundo inmenso, en el que los desarrolladores se especializan en cada una de las partes en las que se divide. Otros se especializan también en cómo trabaja cierto framework y desarrollar todo a partir de él. Tener ahora una visión global de todo esto, y saber que detrás de cada una de las aplicaciones que veo tienen esa gran magnitud de trabajo por detrás hace que quiera seguir aprendiendo.

Pero no sólo técnicamente me ha aportado cosas la realización del proyecto. El tener que estudiar todos los ámbitos de la aplicación, ya no solo la programación, me hace sentir que tienes que tener las ideas claras antes de embarcarte a un proyecto. No puedes llegar a la pantalla del ordenador y ponerte a programar lo que se te ocurra, tienes que planificar bien lo que quieres hacer. Saber tus limitaciones. Estudiar la viabilidad, lo que te puede aportar. Cuando lo tengas claro, hacerlo.

Es sobretodo esto último es una de las aportaciones personales que más me han llenado. El acabar un proyecto así. Es cierto que en mi vida estudiantil desarrollé aplicaciones para aprender nuevas tecnologías. Pero estas aplicaciones eran de una magnitud muy reducida, que te asentaban unas bases sobre esa tecnología. Pero no es lo mismo. El desarrollar una aplicación grande, con fundamentos, con estudios de mercado. Muchas horas solucionando errores, aprendiendo de ellos.

Es difícil medir en un proyecto final de carrera todo el grado de aportación que te afecta directamente. Las sensaciones que me llevo son positivas. Es un trabajo duro, pero la recompensa es, a mi modo de ver, grande y merecida

Ampliaciones

Mejoras

En la aplicación quedan mejoras y correcciones en todas las partes. Pero esto ocurre como en cualquier otra aplicación. En el mundo de las nuevas tecnologías tienes que estar siempre actualizándote.

Las mejoras las podemos dividir en la siguiente lista:

- **Interfaz:** Esta es posiblemente una de las mejoras que siempre habría que ir actualizando. El diseño de la página está pensado en la sencillez de formas y colores tan de moda actualmente. En este aspecto también se tendría que realizar un estudio de estructura de la aplicación para que el usuario se sienta cómodo utilizando la aplicación.
También sería recomendable el crear un logo para la aplicación, que sea diferenciador del resto de aplicaciones.
- **Diseño:** La implementación de la aplicación ha intentado siempre mantenerse fiel al diseño de la aplicación por capas. El desarrollo se ha hecho de forma en las que las diferentes partes de la aplicación se encuentren separadas unas de otras y no exista una gran cohesión ni dependencia una de otras. El problema viene cuando esto, por unos u otros motivos, no ha podido mantenerse siempre. En algunos momentos puntuales del desarrollo se ha tenido que escribir lógica en partes de interfaz.
Esto se debería ir solucionando revisando el diseño del sistema y mejorándolo a la par para ir reduciendo hasta el mínimo estos problemas.

Monetización

Las herramientas de análisis de redes sociales que hay en el mercado ofrecen siempre una versión de pago con la que poder añadir más información a los análisis que realizan.

En la aplicación que hemos desarrollado no habría una forma clara de limitar la información a mostrar pero sí que se podrían buscar nuevas formas de generar dinero. Se podría considerar el hacer un análisis de seguimiento directo a las personas que nos contraten. Realizando estudios en profundidad en cada uno de los casos.

Ampliación de redes sociales a analizar

Redes sociales aparecen todos los días. El centrarse en una sola red social, aunque sea una de las más importantes del momento, no es algo beneficioso. Hemos visto redes sociales crecer y desaparecer en un espacio corto de tiempo. Incrementar el número de redes sociales serviría para no quedarse nunca estancado y ofrecería oportunidades de negocio nuevas.

Todo ello, además, pensando que los usuarios corrientes que existen en el mercado no están tampoco en una sola red social. El usuario medio estará conectado a diferentes redes sociales. El tener una herramienta que centralizara los análisis de todas las redes sociales sería un gran hito.



Referencias

Internet

- Twitter Developers. REST APIs. [Guía de referencia Web]. Versión 2.0. [Consulta Enero-Septiembre 2015].
<https://dev.twitter.com/>
- Django Software Foundation. Django [Página Web]. Versión 1.8.4. [Consulta Enero-Septiembre 2015].
<https://www.djangoproject.com/>
- Wikipedia. Modelo-vista-controlador. [Publicación seriada en línea]. Fecha última actualización: 2015-06-17, 15:38:00. [Consulta Abril 2015].
<https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador>
- NeoModel. Neomodel documentation. [Guía de referencia Web]. Versión 1.2.2. [Consulta Enero-Agosto 2015].
<https://github.com/robinedwards/neomodel>
- Neo Technology, Inc. Neo4j. [Programa] [Página Web] Versión 2.3.0 para Windows. [Consulta Enero-Julio 2015].
<http://neo4j.com/>
- Bootstrap. [Módulo software] Bootstrap Components. [Página Web]. Versión 3.3.5. [Consulta Julio-Septiembre 2015].
<http://getbootstrap.com/>
- jQuery [Módulo software] jQuery Foundation. [Página Web]. Versión 2.1.4 [Consulta Marzo-Septiembre 2015].
<https://jquery.com/>
- D3.js. [Módulo software]. D3 Data-Driven Documents. [Página Web]. Versión 3.5.6. [Consulta Julio-Septiembre 2015].
<http://d3js.org/>

- Autores Varios. Herramientas de análisis de redes sociales. [Blogs]
[Consulta Enero-Septiembre 2015]
<http://miposicionamientoweb.es/twitter-analytics-para-tus-tweets/>

<http://blogthinkbig.com/tweetchup-es-la-herramienta-para-conocer-tus-metricas-de-twitter/>

<https://documania20.wordpress.com/2013/04/03/herramientas-en-la-web-2-0-tweetreach/>
- ISO IEEE 830 [Documentación] Gonzalo Méndez. [Página Web]. Versión 1.
[Consulta Agosto 2015]
<https://www.fdi.ucm.es/profesor/gmendez/docs/iso809/ieee830.pdf>

Bibliografía

- ROBINSON, Ian, WEBBER, Jim, EIFREM, Emil. *Graph Databases*. Primera Edición. Estados Unidos de América: O'REILLY. ISBN:978-1-449-35626-2
- Lubanovic, Bill. *Introducing Python. Modern Computing in Simple Packages*. Sebastopol, California (United States of America). O'Reilly Media, Inc. Noviembre 2014. 478 p. ISBN 978-1-4493-5936-2.
- Cecil Martin, Robert. *UML for Java Programmers*. Englewood Cliffs, New Jersey (United States of America). Prentice-Hall, Inc. ca. 2002. 236 p. ISBN 0-13-203837-4
- Requena Santos, Félix. *Análisis de redes sociales*. Madrid. Centro de Investigaciones Sociológicas CIS. ISBN 8474763479
- Haro Ollé, Juan José de. *Manual imprescindible de redes sociales para la educación*. Madrid. Anaya Multimedia, D.L. 2010. ISBN 9788441527966.