

Diseño e implementación de una herramienta para el análisis estructural y semántico de Twitter

Trabajo de Fin de Grado
Grado en Ingeniería Informática



Autor:

José Pertierra das Neves

Tutores:

Alejandro J. García del Amo Jiménez

Miguel Romance del Río

Septiembre de 2016

“El matrimonio es la principal causa del divorcio”

- Groucho Marx

Agradecimientos

A mis padres, en donde he encontrado la seguridad que muchas veces he necesitado. A mi hermano, aunque sea más de letras. A mi familia, la de aquí y la de allí. A mis amigos, otra familia más. Mis compañeros de universidad, los del Grado de Ingeniería Informática, los del Grado en Matemáticas y los del Doble Grado en concreto, una generación de gente increíble. A los maestros de colegio e instituto que han impulsado mis ganas de descubrir, a Fernando. A la Universidad Rey Juan Carlos y al departamento de Matemáticas que me han permitido avanzar muy rápido en poco tiempo ofreciendo este Doble Grado. A sus profesores, todos y cada uno han dejado huella en mí. En especial, a mis tutores de Trabajo de Fin de Grado, Miguel y Alejandro, quienes me han aportado este proyecto, casi tan interesante como ellos. A las increíbles personas que he conocido este último año, me han ayudado a encontrarme y definirme. A Lu, por todo.

A las demás personas que he conocido a lo largo de mi vida, de todas he aprendido algo y todas han influido. A mi yo del pasado y a mi yo del futuro.

Gracias.

Resumen

En el Trabajo de Final de Grado de título *Diseño e implementación de una herramienta para el análisis estructural y semántico de Twitter* se describen los procesos de análisis, diseño, implementación y evaluación de una aplicación de escritorio para el análisis de conjuntos de mensajes en Twitter.

La aplicación desarrollada se presenta como una herramienta creada con el objetivo de facilitar el tratamiento y estudio de características de los datos de Twitter por parte de cualquier posible usuario interesado en el ámbito de estudio de redes sociales.

Se ha concebido como una aplicación en Python que genera redes de interacciones entre mensajes utilizando también el posible sentimiento de los mismos. Sus características principales son la capacidad de rápida aplicación de clasificadores semánticos como el LSA o Naive-Bayes sobre los mensajes obtenidos y su exportación como red, y análisis de la misma. Para el fácil uso de estos métodos, se ha realizado una aplicación de escritorio.

La aplicación ha sido diseñada con el objetivo de facilitar el uso de análisis sobre cada uno de los tipos de datos: tuits, sentimientos y redes. Se ha implementado por tanto una arquitectura centrada en estos tres apartados por separado y comunicándolos entre sí realizando traspaso de información sobre la base de datos y documentos donde guardar las estructuras.

Después de realizar una revisión bibliográfica acerca de las posibles tecnologías que podrían utilizarse para el desarrollo de esta aplicación, se seleccionaron las siguientes: Python como lenguaje de programación, MongoDB como base de datos, JSON como comunicación entre módulos y XML como almacenamiento en documentos de estructuras para posterior uso.

La presente aplicación ayuda a visualizar, con un coste medianamente asequible de aprendizaje, conceptos relacionados con el datamining y estudio de redes sobre Twitter.

Palabras clave: Análisis semántico Latente, Clasificador Naive-Bayes, Descomposición en Valores Singulares, Grafos, MongoDB, Procesamiento del Lenguaje Natural, Python, Redes, Redes Sociales, Sentimiento, Twitter.

Abstract

In the Final Degree Project named *Diseño e implementación de una herramienta para el análisis estructural y semántico de Twitter* the analysis, design, implementation and evaluation processes of a desktop application for the analysis of groups of messages in Twitter are described.

The developed application is presented as a tool created with the aim of giving an easier treatment and study of the Twitter data characteristics to a possible user interested in the research of social networks.

This project has been conceived as a Python application that generates messages interactions networks using the sentiment of those messages. Its main characteristics are the capability of a quick semantic classifier application such as LSA or Naive-Bayes over the obtained messages and their exportation as a network as well as the analysis of that network. In order to facilitate the use of these methods, a desktop application has been created.

The application has been designed with the aim of giving an easier use of the analysis over every type of data: tweets, sentiments and networks. Because of that, a centered architecture in these three topics but in different areas with an interchange of information with the database and files in where the structures can be saved.

After doing a bibliographic revision about the possible technologies that could be used for the development of this application, some of them were selected: Python as programming language, MongoDB as database, JSON as module's communication and XML as the language in which the structures are saved in files.

The application helps visualizing, with a low learning cost, concepts related with the datamining and the study of Twitter networks.

Keywords: Graphs, Latent Semantic Analysis, MongoDB, Naive-Bayes Classifier, Natural Language Processing, Networks, Python, Sentiment, Singular Value Decomposition, Social Networks, Twitter.

Índice

Resumen.....	1
Abstract	3
Tablas.....	9
Tabla de Ilustraciones.....	11
1. Introducción	13
1.1. Twitter	13
1.2. Finalidad del proyecto	16
1.3. Organización del documento	17
2. Fundamentos Teóricos	19
2.1. Clasificación del Sentimiento en Textos	19
2.1.1. Clasificador Naive-Bayes.....	19
2.1.2. Análisis Semántico Latente.....	20
2.2. Estudio de Redes	21
3. Tecnologías empleadas	23
3.1. Documentos para datos	23
3.2. Base de Datos: MongoDB.....	25
3.3. Lenguaje de Programación: Python	29
3.4. Librerías para Python empleadas	30
3.4.1. Tweepy	30
3.4.2. Math	31
3.4.3. NumPy	31
3.4.4. Tkinter.....	31
3.4.5. PyMongo.....	31
3.4.6. Networkx	32
4. Descripción del proyecto.....	33
4.1. Requisitos	33
4.2. Diseño	35
5. Implementación	37
5.1. Datos obtenidos de Twitter.....	37
5.2. Guardado de los Datos	38
5.3. Módulo I: Administración de los datos desde Twitter	38
5.4. Módulo II: Clasificación de los sentimientos en los textos.....	39
5.5. Módulo III: Manejo de Redes	41

6.	Planificación	43
6.1.	Planning.....	43
6.2.	Diagrama de Gantt.....	46
7.	Pruebas y verificación	47
7.1.	Diseño de pruebas	47
7.1.1.	Infraestructura	47
7.1.2.	Participantes	47
7.1.3.	Protocolo.....	47
7.2.	Ejecución de pruebas y resultados	52
7.2.1.	Introducción	52
7.2.2.	Presentación	52
7.2.3.	Guiado por la aplicación.....	52
7.2.4.	Análisis del cuestionario.	52
7.2.5.	Análisis de las interacciones.....	56
7.2.6.	Errores.....	57
8.	Conclusiones y trabajo futuro	58
8.1.	Conclusiones	58
8.2.	Líneas de trabajo futuras	59
	Manual de Usuario.....	61
	Requisitos e instalación	61
	Gestión de la información de Twitter	61
	Configuración de la conexión.....	62
	Seguimiento de Temas.....	63
	Clasificación de Sentimiento	69
	Interfaz de clasificación manual.....	69
	Clasificación automática	71
	Clasificación de sentimiento sobre un seguimiento	72
	Redes.....	73
	Creación de la red	73
	Herramientas de análisis de la red.....	74
	Configuraciones visuales.....	75
	Representación de centralidad	80
	Plugins y herramientas externas.....	80
	Redes entre Hashtags y seguimientos	82
	Creación de redes de sentimiento	83
	Guión para la realización de pruebas	85

Tareas	85
Cuestionario	86
Glosario.....	87
Referencias	89

Tablas

Tabla 1 Clasificaciones de tuits por PearAnalytics, 2009	14
Tabla 2 Representación de las características de una base de datos orientada a documentos	27
Tabla 3 Ejemplos de limitaciones en las peticiones por parte de la API de Twitter.....	31
Tabla 4 Duración del estudio previo.....	43
Tabla 5 Duración de la revisión contextual sobre el problema.....	43
Tabla 6 Duración del desarrollo del Módulo de Twitter	44
Tabla 7 Duración del desarrollo del Módulo de Análisis de Sentimiento	44
Tabla 8 Duración del desarrollo del Módulo de Administración de Redes	44
Tabla 9 Duración del desarrollo de la unificación de los módulos desarrollados	45
Tabla 10 Duración de la fase principal de pruebas	45
Tabla 11 Duración de la fase de Documentación	45
Tabla 12 Resumen de las tareas principales en el proyecto	45
Tabla 13 Tabla a rellenar tras la realización de la prueba	49
Tabla 14 Referencias de comparación de datos obtenidos en la monitorización de las pruebas.....	50
Tabla 15 Puntuaciones obtenidas como respuesta a la pregunta 9 del cuestionario.....	55
Tabla 16 Tabla con los tiempos de ejecución de las tareas.....	56
Tabla 17 Tabla del número de clicks realizados por participante en cada tarea	57
Tabla 18 Fallos reportados en la realización de las pruebas	58

Tabla de Ilustraciones

Ilustración I Icono de la Red Social Twitter	13
Ilustración II Gráfico de clasificaciones de tuits por PearAnalytics, 2009	14
Ilustración III Perfil en Twitter de El País	15
Ilustración IV Representación de varios usuarios	16
Ilustración V Representación de la terna del Teorema CAP	26
Ilustración VI Logotipo de MongoDB.....	26
Ilustración VII Logotipo del lenguaje de programación Python	29
Ilustración VIII Representación de los distintos módulos que compondrán la aplicación.....	36
Ilustración IX Representación de las colecciones creadas en MongoDB	38
Ilustración X Representación del ciclo de guardado de tuits mediante un StreamListener	39
Ilustración XI Representación del manejo del módulo de clasificación de sentimiento.....	40
Ilustración XII Representación de la creación de la red básica a partir de los tuits	42
Ilustración XIII Método de obtención del clúster de un nodo.....	42
Ilustración XIV Diagrama de Gantt correspondiente al planning	46

1. Introducción

1.1. Twitter

Twitter es una red social creada en Marzo de 2006 destinada al microblogging cuenta con más de 310 millones de usuarios activos mensuales y ha tenido una expansión en más de 40 idiomas [1]. Se ha convertido de forma inequívoca en una de las plataformas más utilizadas para la transmisión y comunicación de contenidos de todo tipo, especialmente los relacionados con la actualidad y retransmisión en directo en tiempo real. De esta forma podemos encontrar cómo se ven reflejados eventos especiales como los Juegos Olímpicos de Río de 2016 en las redes sociales, generando en este caso 187 millones de tweets. Ésto no ha sido ajeno a los creadores de los directores en la compañía californiana, que dispone de numerosas estadísticas que miden y estudian los temas más populares. Volviendo al ejemplo, facilitan los 3 momentos más importantes, los 3 atletas más mencionados y los 3 deportes más comentados [2] para el uso como marketing y promoción de los eventos mostrando la influencia y la repercusión que han tenido. Además, Twitter también ha dispuesto para otras compañías análisis especializados que permiten medir la efectividad de la publicidad y la difusión de las campañas en la red social [3].



Ilustración 1 Icono de la Red Social Twitter

Como en muchos servicios o aplicaciones web que han ido surgiendo, tenemos una gran base de datos de información que los usuarios proporcionan de forma voluntaria. Por ejemplo, podemos aprovechar secciones de proveedores en e-commerce como Amazon donde los compradores evalúan los productos para obtener análisis de CRM (Customer Relationship Management) que nos facilite la búsqueda de patrones o estructuras de comportamiento. Si lo aplicamos a Twitter, el objeto principal de estudio son las grandes montañas de información generadas por los usuarios y ligadas a los tweets. Al no tratarse de pocos datos, uno de los objetivos a cubrir es el de la optimización de los métodos mediante la automatización. Volviendo al ejemplo de los patrones de comportamiento, un factor importante es el entendimiento o cuantificar los sentimientos dentro de los textos de los tuits por ejemplo. En 2009 , Pear Analytics [4] realiza un estudio sobre 2000 tuits donde se clasificaron según las siguientes categorías: noticas, spam, autopromoción, cháchara sin sentido, conversaciones y repeticiones de mensajes. Sus porcentajes de aparición en los análisis fueron los siguientes

<i>Categoría</i>	<i>Porcentaje</i>
<i>Noticias</i>	3.6
<i>Spam</i>	3.75
<i>Autopromoción</i>	5.85
<i>Cháchara sin sentido</i>	40.55
<i>Conversaciones</i>	37.55
<i>Repeticiones de mensajes</i>	8.7

Tabla 1 Clasificaciones de tuits por PearAnalytics, 2009

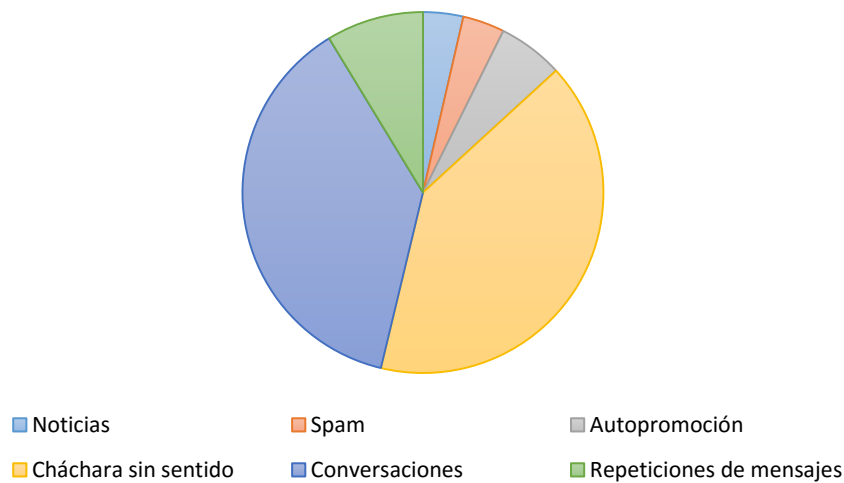


Ilustración II Gráfico de clasificaciones de tuits por PearAnalytics, 2009

Por tanto, observamos un gran filo correspondiente a las conversaciones que se crean, lo que nos puede ayudar a entender la estructura de Twitter. Y por otro lado, el estudio de sentimiento mediante el uso de modelos y herramientas que nos lo faciliten, que tiene una fuerte aplicación en multitud de campos donde el comportamiento y patrones de opinión de usuarios o clientes sea fundamental.

Una vez presentado Twitter como actor principal donde trabajaremos, deberemos entender su terminología propia. Ésta es común para otras aplicaciones, y nos ayudará a entender el contexto de estudio. Para ello, partiremos en casi todas las redes sociales de la idea de amistad y relación en la red. A continuación, tendremos las interacciones que se pueden realizar entre los usuarios de forma normal como el envío de información muy concreta. Y por último podemos apreciar algún tipo de interacción extra que se permita realizar en cada una de las diferentes aplicaciones. En cuanto a la estructura de Twitter definiremos

- Usuario: en representación a una cuenta que puede publicar información que firma como suya y que reciben todos los que están suscritos para recibir sus actualizaciones.
- Tuit: bloque de información que se intercambia. Pueden ser de tipo escrito (con un límite de 140 caracteres) o un documento de tipo multimedia, como imágenes, vídeos o enlaces con características dinámicas como noticias e incluso encuestas.

- Relación de seguimiento: como se ha comentado antes, el funcionamiento de Twitter se basa en la suscripción de cuentas, que puede ser unidireccional. Esto es, podremos encontrar perfiles que sigan a muchos usuarios, por lo que recibirán mucha información, y perfiles que tengan muchos seguidores, lo que aumenta su radio de difusión, o ambos a la vez. Cuantos más seguidores mayor será la probabilidad de calificar como influyente a un perfil o usuario. Además, podemos estudiar el ratio entre seguidos/seguidores. Por ejemplo, el diario español El País cuenta con más de 5.62 millones de seguidores y se encuentra suscrito a 720 cuentas lo que representa aproximadamente un 0.000128%.



Ilustración III Perfil en Twitter de El País

- Retuit: si un usuario recibe un tuit puede reenviarlo a todos sus seguidores. De esta forma, cuantas más veces se haya retuiteado o reenviado un tuit podremos intuir si ha sido influyente o no, también estudiar qué usuarios lo han realizado en contraposición a los que no.
- Me gusta: nos permite identificar o resaltar un tuit que se ha recibido y calificarlo positivamente. Nuevamente, cuantos más “Me gusta” obtenga un tuit, intuitivamente, más habrá interesado. La red social anteriormente realizaba esta opción mediante el denominado Favorito con el símbolo de una estrella, pero esto fue cambiado a finales de 2015 debido a que éste podría “generar confusión” y el corazón se encuentra entre los iconos más representativos en redes sociales [5]. En la Ilustración III Perfil en Twitter de El País podemos apreciar que la información de los tuits marcados como “Me gusta” son públicos, lo que permite analizar los gustos de cada usuario en concreto.
- Mensaje Directo o Privado: interacción entre dos usuarios de forma privada, a la que sólo tienen acceso el creador o el destinatario.

- Cada usuario tiene su página principal o TimeLine donde puede ver la información que llega de sus suscripciones ordenada de forma invertida cronológicamente. Además tiene una sección de información detallada sobre su cuenta como procedencia, fecha de registro, fecha de nacimiento, descripción, nombre asignado o identificador único de usuario.
- Menció: Dentro del denominado tuit, los usuarios pueden realizar contacto directo con otros añadiendo su identificador de usuario, ésta será nuestra principal interacción de estudio. Por lo que una representación muy básica de la estructura que podrían tener varios usuarios en Twitter sería la Ilustración IV Representación de varios usuarios, la cual se corresponde con una abstracción que se adapta a otras redes sociales cualesquiera.

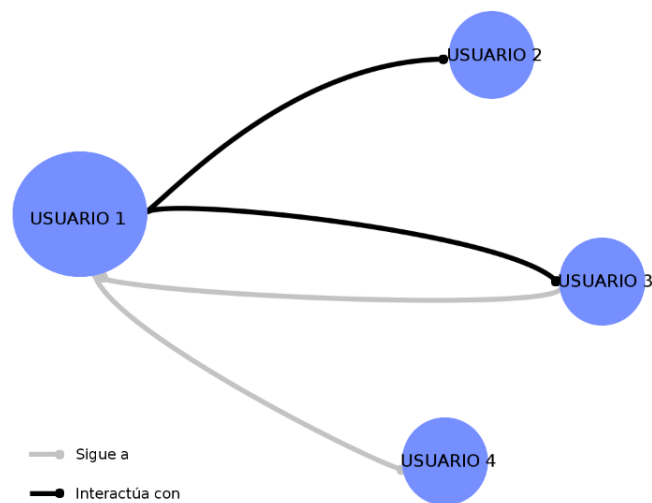


Ilustración IV Representación de varios usuarios

- HashTag: etiqueta en texto plano cuyo fin es el de representar una temática importante. Su estructura está conformada por el carácter “#” seguido de texto. Podemos realizar un seguimiento de cada hashtag como representación de un hito importante, donde se puede medir la cantidad de tuits que se realizan, la participación en número de usuarios o las conversaciones que se mantienen.
- Topics: son temas convertidos en populares ya que son o muy comentados, o patrocinados o de gran importancia en la actualidad. Pueden ser varias palabras y cualesquiera. Se utiliza el término Trending Topic para determinar cuándo un tema está siendo muy utilizado en el momento. Para medir esto la compañía utiliza algoritmos que tienen en cuenta lo novedoso de un tema y la velocidad de la repercusión [6].

1.2. Finalidad del proyecto

El proyecto se basa en la revisión bibliográfica donde buscar herramientas teóricas que satisfagan las necesidades de análisis sobre grandes datos presentadas y su implementación a nivel de aplicación. El resultado, debe consistir en una plataforma que permita el seguimiento o

descarga de tuits que se realizan en directo mediante un Hashtag. A partir de éstos, se deberá poder realizar un análisis de sentimiento de textos implementado dentro de la aplicación, que permita comunicar los mensajes escritos en la red social con otros formatos de visualización: mediante la asignación de los sentimientos inferidos sobre los tuits y su presentación en texto simple o como representación en forma de red. Aparte de estudiar las redes creadas mediante los sentimientos podemos realizar análisis de las mismas sin ellos, puesto que las conversaciones serán vistas como interacciones, como se ha comentado anteriormente.

En resumen, la finalidad principal es la de la creación de una herramienta software que nos permita un fácil tratamiento de los datos obtenidos de Twitter y su análisis mediante el estudio de sentimiento y la visión de estructura en forma de red.

1.3. Organización del documento

La organización del documento se corresponde con la de un proyecto para el desarrollo de una aplicación. Primeramente, deberemos realizar una revisión sobre los fundamentos teóricos que serán aplicados. Éstos son: métodos matemáticos para la clasificación semántica y de sentimiento en textos mediante el Análisis Semántico Latente y la aplicación del aprendizaje Bayesiano mediante los clasificadores Naive-Bayes, y el análisis y aplicación de redes en este ámbito. A continuación, una vez hemos visto qué herramientas matemáticas debemos implementar, se presentan las tecnologías a utilizar, justificando su uso. En este apartado se tratarán los documentos que se utilizan como medio de transporte de la información entre aplicaciones, la elección y características de la Base de Datos MongoDB, la selección del lenguaje de programación Python, así como un resumen de algunas de las librerías más importantes que se han utilizado. Después, el documento trata los pasos iniciales en el proyecto software: análisis de requisitos y diseño, donde se especifican las características que se desean implementar definiendo su finalidad. La implementación sigue a este paso, donde se tratará la arquitectura del sistema principalmente. Cómo se subdivide la aplicación en cuanto a nivel de desarrollo basado en la modularización definiendo tres principales partes: la gestión de los datos descargados, la interfaz encargada de facilitar la clasificación de sentimiento y la interfaz encargada de la representación y análisis de redes.

Deberemos especificar a continuación cómo se ha definido el proyecto mediante la división que se ha realizado en tareas y la repartición de éstas en el tiempo. Nos ayudaremos de un Diagrama de Gantt para su representación y fácil visualización.

Previo a toda finalización del desarrollo de una aplicación se deben realizar las pruebas pertinentes sobre el funcionamiento de la misma, así como test de verificación que nos ayudan a definir si se han cumplido objetivos sobre los requisitos que se deseaban implementar. La sección correspondiente se divide en dos fases principales: el diseño y planificación de las pruebas, y su ejecución y análisis de resultados, donde podremos descubrir qué mejoras realizar sobre la aplicación y en qué líneas de desarrollo posterior situarse. Esto entra en relación con el apartado siguiente donde se describirán las conclusiones obtenidas del proyecto y las posibles líneas de trabajo futuras.

Por último, se adjuntan en el documento un Manual de Usuario que representa la información básica de funcionamiento y uso de la aplicación además de documentos extra como

la guía utilizada para la definición de tareas en las pruebas y su supervisión, así como un cuestionario con el que concluyen las mismas.

2. Fundamentos Teóricos

En este apartado trataremos los principales principios teóricos y métodos matemáticos que nos permitan desarrollar correctamente la aplicación.

2.1. Clasificación del Sentimiento en Textos

Dos de las principales herramientas utilizadas para la clasificación de sentimientos en y textos en general son las basadas en el aprendizaje Bayesiano y el Análisis Semántico Latente [7].

2.1.1. Clasificador Naive-Bayes

El aprendizaje Bayesiano establece sus fundamentos en el Teorema de Bayes, el cual establece una relación entre la probabilidad condicionada de dos sucesos A y B aleatorios de forma que

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, (1)$$

donde:

- $P(A)$ es la probabilidad a priori,
- $P(B|A)$ es la probabilidad de B en la hipótesis A ,
- $P(A|B)$ es la probabilidad a posteriori.

A partir de aquí, el Clasificador Naive-Bayes utiliza un espacio de hipótesis y una función objetivo, que define los valores resultado, del cual estimará sus probabilidades para cada una de las hipótesis en su contexto. El resultado devuelto por el clasificador será el que mayor probabilidad tenga dado un conjunto de valores de entrada. En pseudocódigo el funcionamiento del clasificador Naive-Bayes es el siguiente [7]

AprendizajeDeNaiveBayes(ejemplosInput)

Para cada valor del resultado v_j

└ Obtener estimación $P'(v_j)$ de la probabilidad $P(v_j)$

└ Para cada valor a_i de cada atributo a

Obtener una estimación $P'(a_i|v_j)$ de $P(a_i|v_j)$

InstanciaDeClasificador(x)

Devolver $v_{NB} = \operatorname{argmax}_{v_j \in V} P(v_j) \prod_i P(a_i|v_j)$

En la clasificación del sentimiento de textos, cada uno de los diferentes sentimientos: "Negativo", "Positivo", ... serán valores v_j resultado. La aparición de cada uno de los términos o palabras que componen el texto serán los atributos de a .

2.1.2. Análisis Semántico Latente

Se encarga de determinar la similitud semántica mediante la representación vectorial de los documentos y términos para después analizar la similitud de una nueva consulta o texto a clasificar con respecto a ellos.

Los fundamentos del LSA se basan en el SVD, de sus siglas en inglés Singular Value Decomposition, que se encarga de obtener dichos vectores mediante la aplicación del Teorema de factorización de una matriz A rellena con las frecuencias de cada término en cada documento de forma [7]

$$A_{mn} = U_{mm} S_{mn} V_{nn}^T \quad (2)$$

Donde U y V son ortogonales (de orden m y n respectivamente) y S es una matriz diagonal (de tamaño $m * n$), $U^T U = I$, $V^T V = I$. Las columnas de U son autovectores ortonormales de AA^T , las columnas de V son autovectores ortonormales de $A^T A$, y S contiene las raíces cuadradas de los autovalores de U o V ordenados descendientemente.

A partir de aquí, la selección de una dimensionalidad, la cual suele ser 2, para la reducción de las matrices nos ofrece los vectores asociados a las palabras mediante el producto $U_2 S_2$ y escogiendo una fila de U_2 teniendo en cuenta el orden de los términos para saber a cuál representa. El producto $S_2 V^T$ nos ofrece los vectores de cada uno de los documentos. [7]

Una vez establecidos los vectores de documentos y términos, pasamos a clasificar un nuevo texto. El primer paso es escoger los términos que aparecen de forma que el punto en 2 dimensiones donde se representará se forma a partir de la media de los puntos asociados a los términos [7]. De forma

$$Consulta = \frac{\sum_{i=1}^n \vec{v}_i}{n} \quad (3)$$

A continuación, trabajaremos sobre una medida que nos informe de cómo de similares es la consulta al resto de documentos. Esto se realiza principalmente con la similitud coseno obtenida de la siguiente forma [7]

$$Similaridad_{Coseno}(\vec{d}_1, \vec{d}_2) = \frac{\langle \vec{d}_1, \vec{d}_2 \rangle}{|\vec{d}_1| \cdot |\vec{d}_2|} \quad (4)$$

Es interesante fijar un umbral relativo que tenga en cuenta los documentos más cercanos a la hora de realizar la asignación de sentimiento, aunque también podamos escoger el más cercano y copiarlo. El umbral se puede fijar utilizando un valor obtenido a partir de las distancias de todos los documentos a la consulta de forma [7]

$$Umbral = Media_{Distancias} + \alpha * Desviación\ Típica,$$

donde el valor α es una variable que permite configurar el umbral.

Por otro lado, y para mejorar la rapidez de los cálculos, podemos fijar un umbral constante.

2.2. Estudio de Redes

La aplicación de las redes a nuestro proyecto va ligado a su formación a partir de los mensajes realizados en conversación en la red Twitter. Para ello deberemos repasar lo conceptos básicos de definición [7]:

- Definimos un grafo G no dirigido como una dupla $G = (V, E)$ donde:
 - $V = \{v_1, v_2, \dots\}$ es un conjunto de vértices, puntos en el espacio.
 - $E = \{(v_i, v_j), (v_k, v_l), \dots\}$ es un conjunto de aristas es decir pares de elementos de V donde cada uno representa una relación o conexión.
- Un camino se define como una secuencia de vértices $\{i_1, i_2, \dots, i_n\}$ y de relaciones $\{(i_1, i_2), (i_2, i_3), \dots, (i_{n-1}, i_n)\}$ tal que $(i_{j-1}, i_j) \in E$.
- El grado de un nodo viene determinado por el cardinal de la vecindad, número de aristas incidentes en un vértice. Podemos encontrar dos tipos de grados si hablamos de grafos dirigidos:
 - Grado de entrada del nodo i donde se hace el recuento de los nodos conectados hasta éste $\sum_j g_{ji}$
 - Grado de salida del nodo i , en este caso partimos de éste y se dirige a otros. $\sum_j g_{ij}$
- La distribución de grado, $P(\text{grado})$, de una red es una descripción de la frecuencia de nodos que tienen diferentes grados. En las redes naturales, obtenidas mediante información real, esta representación es un histograma de frecuencias.
- El coeficiente de clustering, de agrupamiento o transitividad de una red mide cómo de relacionado se encuentra un nodo a la red. Si estamos tratando un grafo completo el valor es máximo, y es pequeño cuando el agrupamiento es pobre. Viene determinado por

$$Clustering(G) = \frac{3 * \text{numero de triángulos en la red}}{\text{número de ternas de nodos}}. \quad (5)$$

- El coeficiente de clustering de un nodo i se obtiene como la representación la fracción de los triángulos en los que se encuentra entre el número de ternas centradas en éste

$$Clustering_{\text{Nodo } i}(G) = \frac{\text{Número de triángulos conectados al vértice } v_i}{\text{número de ternas centradas en } v_i}. \quad (6)$$

- El clustering medio de la red se calcula como el promedio de los coeficientes de clustering de cada uno de los nodos en ésta. Ésto es

$$Clustering_{Medio}(G) = \frac{1}{n} \sum_i^n Clustering_i(G). \quad (7)$$

- La centralidad de grado para un nodo i , se define como la representación de la cantidad de relacionados con un nodo determinado, viene dada por

$$CentralidadGrado_i = \frac{Grado_i(G)}{n-1}. \quad (8)$$

- La centralidad de cercanía (Closeness) se encarga de indicar cómo de cercano está un nodo dado a cualquier otro

$$Cercanía(v_i) = \frac{1}{\sum_{j \neq i} distancia(i,j)}. \quad (9)$$

- La centralidad de intermediación (Betweenness) captura la información relacionada con cómo de bien situado se encuentra un nodo en términos de los caminos en los que se encuentra dentro de la red

$$Centralidad\ por\ intermediación(v) = \sum_{s \neq v \neq t \in V} \frac{\rho_{st}(v)}{\rho_{st}}. \quad (10)$$

- La función $Centralidad_{Autovectores}(v_i)$ que devuelve la centralidad de autovector del nodo v_i se corresponde con una suma de las demás centralidades utilizando las proporciones adecuadas. De la siguiente forma

$$Centralidad_{Autovectores}(v_i) = \frac{1}{\lambda} \sum_{j=1}^n A_{j,i} Centralidad_{Autovectores}(v_j). \quad (11)$$

3. Tecnologías empleadas

En esta sección, repasaremos los elementos necesarios para el desarrollo del proyecto desde el punto técnico. Como hemos explicado anteriormente, el objetivo es el de construir un conjunto de herramientas en forma de aplicación que nos ayuden a realizar una administración de datos de Twitter, el análisis de sentimiento y el análisis de redes que se forman. De esta manera, trataremos primero qué tecnologías utilizar para la persistencia de los datos o la transmisión de estos entre aplicaciones, su guardado en un almacén o base de datos, y la implementación de todos los módulos que los trate desarrollados en un lenguaje de programación determinado.

3.1. Documentos para datos

Por especificación de la herramienta o para la mejora de las comunicaciones entre los distintos módulos y el envío y obtención de información deberemos utilizar determinados documentos comunes en las aplicaciones informáticas donde almacenar datos. Estas tecnologías a nivel de aplicación las utilizaremos principalmente para la obtención de datos desde Twitter y la persistencia de entornos de trabajo en cada uno de los apartados, como puede ser: guardar las claves de acceso a la API de Twitter mediante archivos JSON, guardar una red o clasificación de sentimiento creada en un archivo XML, o realizar la exportación de una información en esta aplicación a otra externa mediante archivos CSV. Éstos por tanto son los principales tipos de documentos que vamos a describir.

3.1.1. JSON

JSON [8], de sus siglas en inglés Javascript Object Notation, es un formato de intercambio de datos. Destaca su facilidad de comprensión para los humanos debido a su intuitiva estructura, así como también la generación y lectura por parte de los computadores. Como su nombre indica, es un subconjunto de la notación de Javascript en su creación, 1999, aunque hoy en día se considera un lenguaje independiente y gran alternativa al XML [9].

Está formado por dos tipos de estructuras:

- Pares de nombre y valor. Como un diccionario o tabla hash en otros lenguajes. Determinado por la siguiente notación

$$\{nombre_1: valor_1, nombre_2: valor_2, \dots, nombre_n: valor_n\}$$

- Lista ordenada de valores. En otros ámbitos se suele denominar array, vectores o listas. Determinada por la siguiente notación

$$[valor_1, valor_2, \dots, valor_m]$$

3.1.2.CSV

CSV [10], de sus siglas en inglés Comma-Separated Value, se describe como un tipo de archivo en el que se realiza el guardado de datos en texto plano (números y texto). Como su nombre indica, cada registro se compone de uno o más campos separados por comas. Algunos de los problemas debido al uso de comas, es el hecho de no poder utilizarlas en los valores o realizar la inclusión de marcas de escapado para aceptarlas. Esto complica su estandarización. Además, han surgido ramificaciones basadas en CSV que permiten el uso de otros caracteres de separación.

3.1.3.XML

XML [11], de sus siglas en inglés Extensible Markup Language, se trata de un lenguaje de marcado diseñado para el almacenamiento y transporte de información. Ha sido diseñado para una fácil lectura por parte de los humanos y los computadores. Su nacimiento surgió a partir de los llamados GML, de sus siglas en inglés Lenguajes de Marcado Generalizado y tiene similitud con algunos otros lenguajes de este tipo como el conocido HTML [12].

La estructura de un archivo XML se basa en la clara definición en partes de la información, que se pueden subdividir en nuevas partes. Cada una de ellas se denomina **elemento** y se diferencia de las demás por estar señalada con una **etiqueta**. Como en casi todos los archivos de lenguajes de marcado podremos tener una cabecera donde se detallarán las declaraciones como documento XML, versión y codificación. A continuación, tendremos el cuerpo principal del documento, el cual no es opcional, y debe haber solamente un elemento raíz para que esté bien formado y sea legible.

Como hemos dicho anteriormente cada uno de los elementos viene definido por una etiqueta. Además, podremos añadir ciertos campos a ellos llamados atributos describiendo su valor. De esta forma un ejemplo básico del cuerpo de un documento XML resultaría

```
< nombreDeLaEtiqueta atributo1 = ' ... ' ... atributon = ' ... ' >
    < subElemento1 ... >
        ...
    </subElemento1 >
    < subElemento2 ... >
        ...
    </subElemento2 >
        ...
</nombreDeLaEtiqueta >
```

Algunos de los beneficios del uso de archivos XML es acerca del uso de las etiquetas. Como su nombre indica podremos hacer una extensión del lenguaje mediante la adición de nuevas etiquetas. Como ya hemos resaltado, el ser humano tiene gran facilidad a la hora de comprender su estructura, debido al paradigma de elementos y atributos que puede ser también fácilmente interpretado por analizadores.

El uso de XML por tanto se realiza principalmente para la transmisión de información entre diversas aplicaciones. En este proyecto deberemos tenerlo en cuenta si queremos almacenar datos temporales o plantillas de trabajo.

3.2. Base de Datos: MongoDB

3.2.1. Selección de la Base de Datos

Una de las principales características de las bases de datos NoSQL es el tipo de modelo de datos que se quiera implementar. Son perfectas para proyectos en los que se requiera escalabilidad y sean bases de datos distribuidas. Permiten el almacenamiento de los datos sin la necesidad de estructuras fijas como tablas, aunque no suelen soportar la operación JOIN. Podemos encontrar varios tipos si tenemos en cuenta el modelo de datos que se quiere representar: orientación a documentos, a columnas, a clave y valor o grafos, por ejemplo.

El uso de bases de datos NoSQL se ha incrementado en la tecnología del CloudComputing, donde es necesario el almacenamiento de grandes cantidades de datos, así como su eficiente movilidad y tratamiento frente a los sistemas SQL clásicos los cuales pueden salir victoriosos en cuanto a tiempos de ejecución pero no en escalabilidad y flexibilidad. Tenemos ejemplos muy claros como Google, Twitter, Amazon o Facebook.

Uno de los puntos en los que debemos apoyarnos a la hora de seleccionar cuál es nuestro mejor sistema de base de datos a utilizar en el proyecto es la selección de los conceptos de importancia según el Teorema CAP o de Eric Brewer [13], de los cuales es imposible para un sistema de cómputo distribuido garantizar simultáneamente.

- Consistencia. Los clientes que realicen peticiones deben disponer la misma información al mismo tiempo. Esto se soluciona mediante la replicación de la información.

- Disponibilidad. Siempre se puede leer o escribir en la información base. Esto se soluciona con la correcta actualización continua de la información en cada nodo.

- Tolerancia a particiones. O el correcto funcionamiento aunque tengan lugar particiones de información entre los nodos. Esto se soluciona con la definición de reglas para el control de las casuísticas de nodos fallo.

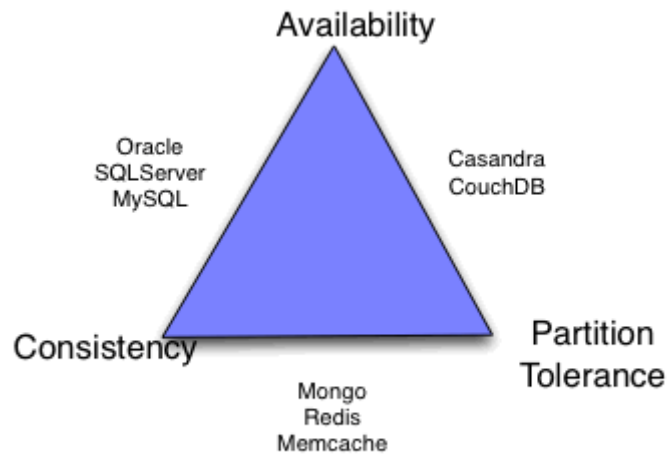


Ilustración V Representación de la terna del Teorema CAP

Sólo podremos tener no más de dos de estas características a la vez [14]. Y como se puede apreciar en el esquema, MongoDB proporciona la Consistencia y la Tolerancia a Particiones.

3.2.2. Visión General de MongoDB

MongoDB es un sistema de bases de datos NoSQL (llamado también “No SQL” o “No relacional”), desarrollado a partir de Octubre de 2007, orientada a documentos diseñada para una fácil evolución y escalabilidad, su desarrollo se realiza sobre el concepto código abierto. [15]. Su principal diferencia se basa en el guardado de los datos. A diferencia de las bases de datos relacionales, donde se guardan en estructuras de tablas, MongoDB realiza el almacenamiento mediante documentos BSON, los cuales se encuentran basados en los JSON, que cuentan con un esquema dinámico.



Ilustración VI Logotipo de MongoDB

Podemos resumir la clasificación de las características de una base de datos orientada a documentos, como MongoDB, en la siguiente tabla [16].

Característica	Nivel
Rendimiento	Alto
Escalabilidad	Alta
Flexibilidad	Alta
Complejidad	Baja
Funcionalidad	Variable (baja)

Tabla 2 Representación de las características de una base de datos orientada a documentos

3.2.3. Trabajando con MongoDB

En MongoDB podremos encontrar las mismas operaciones básicas que en el resto de bases de datos tradicionales como son las denominadas funciones CRUD [17]. Al no utilizar el Lenguaje Estructurado para Consultas (SQL) tradicional, se utiliza una sintaxis de funciones en JavaScript. Siempre deberemos utilizar el objeto "db" que representa la base de datos y a continuación el nombre de la colección, que puede asemejarse a una tabla, seguido de un punto. A continuación, de nuevo otro punto y la función a realizar. Las operaciones fundamentales son:

- Insert o inserción en una colección. Para insertar un documento en una colección previamente creada, o se creará automáticamente. Debemos simplemente especificar los distintos campos que lo componen en forma de diccionario. En SQL sabemos que la sentencia utilizaríamos *"INSERT INTO"*. En la que estamos tratando un ejemplo de sintaxis sería:

```
db.compradorCoche.insert({
  Comprador: 'Antonio Ramírez',
  Precio: '17500',
  Fecha: '10-08-2016'
});
```

Cada elemento en la colección debe tener un identificador "_id" (número hexadecimal de 12 bytes) que se puede especificar opcionalmente en la inserción. Si no se hace, MongoDB se encargará de generar un identificador único. Pero si se define un identificador deberá ser único o, como en SQL, nos devolverá un error de clave primaria duplicada.

- Find o búsqueda de elementos en la colección. Para ubicar registros donde se cumplen reglas descritas en los valores de los campos. En SQL su equivalente sería *"SELECT ... FROM ..."*. Su sintaxis más básica es la siguiente:

```
db.compradorCoche.find({});
```

Donde obtendríamos todos los elementos de la colección *compradorCoche*. Ahora bien, si queremos utilizar algún tipo de filtro que se cumplan en los campos,

valores concretos o rangos, sólo deberemos utilizar los parámetros pertinentes en el interior de la función*find(...parámetros...)*; .Donde podremos utilizar operaciones de comparación como: *\$seq* (para valores equivalentes), *\$gt* (para valores mayores que uno dado) o *\$ne* (para valores distintos a uno dado entre otros).

A continuación del Find, podemos utilizar también ciertas funciones como: *sort()* ,para la ordenación, o *limit()* ,para limitar el número de datos devueltos, entre otras.

- Remove y Drop nos permiten eliminar elementos o colecciones enteras directamente. En SQL su equivalente sería "*DELETE FROM ...*" y "*DROP TABLE ...*". Un ejemplo de su sintaxis en MongoDB sería la siguiente:

```
db.compradorCoche.remove({
  Precio:{$gte:'15000'}
});

db.compradorCoche.drop();
```

- Update o actualización de un elemento. Se encargará de reescribir la información sobre los elementos que coincidan con unos valores determinados. Por ejemplo, si queremos actualizar los compradores de coche que se llamen "Tony" y cambiarles en nombre por "Antonio" sería de la siguiente forma:

```
db.compradorCoche.update({
  {Comprador:"Tony"},
  {Comprador:"Antonio"}
});
```

Además, podemos realizar algunas otras operaciones útiles definidas en la función Aggregation, las cuales son más complejas en la aplicación a un conjunto de datos. Podemos especificar, por ejemplo:

- *\$group* para la agrupación de registros que cumplan unas determinadas características.
- *\$match* para reunir elementos cuyos campos que cumplan una condición. de forma similar a la búsqueda.
- *\$count* para realizar un conteo.

3.2.4. Ventajas de MongoDB

Podremos destacar algunas ventajas que nos ofrece el uso de MongoDB [18]

- Alta disponibilidad.
- Buena escalabilidad y distribución en arquitecturas de considerables clústers.
- El manejo de objetos JSON tiene la ventaja de que es el estándar popular en el desarrollo Web, por lo que existe cierta cohesión.

- Podemos utilizar el paradigma Map-Reduce que nos permite dividir la información y aplicar funciones map mediante sentencias en Javascript ejecutadas en el servidor.
- En cuanto a la seguridad, nos ofrece autenticación y autorización, además de una buena gestión de usuarios.

De esta forma MongoDB se muestra como una herramienta muy útil para almacenar de manera eficiente grandes conjuntos de tuits y los datos relacionados con éstos. Además, tiene muy buen soporte con casi todos los lenguajes de programación populares, lo que nos aporta versatilidad y flexibilidad en el proyecto.

3.3. Lenguaje de Programación: Python

Python [19] es un lenguaje de programación interpretado, orientado a objetos y de alto nivel con semánticas dinámicas. Su alto nivel construido en estructuras de datos y combinado con un tipado y referenciado dinámicos, lo convierte en un lenguaje muy sencillo y práctico para el desarrollo rápido de aplicaciones. Además, es común para el uso en scripting. Uno de sus puntos fuertes es el de tener una sintaxis clara, fácilmente interpretable (esto es uno de los principios de la comunidad alrededor del lenguaje), lo que permite una reducción del coste en mantenimiento por su buena legibilidad. También cuenta con una modularización lo que nos brinda un gran acceso a la reutilización de código.



Ilustración VII Logotipo del lenguaje de programación Python

Se ha reconocido que Python puede no desempeñar las tareas requeridas con una rapidez como la de otros lenguajes de programación como son Java y C++. De hecho, estos pueden llegar a ser entre 3 y 5 veces más rápidos. Sin embargo, con Python hay un increíble ahorro de tiempo en el desarrollo. Por ejemplo, algunos de los programas o algoritmos que implementemos en Java en 15 líneas podremos reducirlas a una tercera parte con este lenguaje interpretado. También hay algunas otras ventajas como la no declaración explícita de variables ya que se utiliza la asignación directa en la inicialización. Además, es conocido comúnmente como posible solución para la comunicación entre componentes aunque estén escritos en otros lenguajes.

“Python is powerful... and fast; plays well with others; runs everywhere; is friendly & easy to learn; is Open” [20]

Su creación se remonta a finales de los ochenta por Guido van Rossum en Holanda. Y, al ser pensado como un desarrollo de código abierto, bajo una licencia pública general de GNU, se ha creado una fuerte comunidad alrededor. La cual promueve su evolución, documenta frameworks y realizan proyectos con espíritu open-source. Esto ha favorecido el uso de Python en entornos científicos, donde también su facilidad de aprendizaje y versatilidad han jugado un buen papel. Creándose multitud de código y librerías relacionadas con aplicaciones de cálculo como Matlab o R; de visión artificial o de desarrollo gráfico; aplicación en ramas de la Computación o Matemáticas gracias a su posibilidad de desempeñar su trabajo permitiendo la programación funcional.

Python nos ofrece además muchas facilidades para el trabajo posterior y desarrollo de proyectos, principalmente en la creación de una aplicación web que sirva como herramienta que cubra algunas de las funciones de una aplicación propia de escritorio. En ese ámbito hablaríamos de Django donde fácilmente aplicaríamos el paradigma MVC, de sus siglas en inglés Modelo Vista Controlador.

3.4. Librerías para Python empleadas

Pasaremos a describir a continuación algunas de las librerías importantes que necesitaremos para el desarrollo del proyecto.

3.4.1. Tweepy

Tweepy [21] surge como un framework que permite la comunicación con Twitter mediante las APIs creadas por la compañía mediante numerosas funciones. Nos facilita las peticiones que hay que realizar al servidor para la búsqueda de tuits e información de usuarios. Para ello, a partir de la versión 1.1 de la API de Twitter es necesario realizar para todas las peticiones la autorización a través de OAuth, un protocolo abierto que permite la autorización segura en un simple y estándar método desde la web, entornos móviles y aplicaciones de escritorio [22] Para ello se deberá registrar una cuenta y rellenar los datos correspondientes a la aplicación que se quiera desarrollar. Nos proporcionará cuatro identificadores: *clave de cliente*, *clave secreta del cliente*, *token de acceso* y *token secreto de acceso*.

Algunas de las limitaciones por tiempo que se han establecido para la versión de la API que se utiliza en este proyecto (1.1) son las siguientes

<i>Consulta</i>	<i>Número de peticiones en ventana de 15 mins.</i>
<i>Búsqueda de tweets</i>	450

Obtención de las id de los followers	15
Obtener TimeLine de usuario	300
Tópicos populares por lugar	15

Tabla 3 Ejemplos de limitaciones en las peticiones por parte de la API de Twitter

También existen un tipo de objetos denominados Streams que nos permiten mantener búsquedas con conexiones de mayor duración. Esto nos ayudará, por ejemplo, a que si queremos realizar el seguimiento de un HashTag o palabra clave se mantenga a la escucha sin limitaciones por parte del servidor que proporciona la información.

Debemos recalcar que, aunque la librería nos facilita la forma de acceso para realizar las consultas, las respuestas desde la API son siempre mediante un documento JSON donde se identifica cada campo con su nombre y valor. [23] Lo que necesitará de un post tratamiento en el que deberemos escoger los datos necesarios

Se utilizará la versión 3.5.0 de esta librería.

3.4.2. Math

La librería Math [24] se encuentra siempre dentro de las distribuciones de Python y se encarga de proporcionar las funciones matemáticas más complejas y que no tienen representación sintáctica. Algunas de ellas pueden ser: tratamiento de números (como obtención del absoluto, comprobar si es negativo o positivo, si no es un número, truncar, factorial, redondeo al alza...), potencias o funciones logarítmicas, funciones trigonométricas (seno, coseno, tangente...) e hiperbólicas, además se incluyen constantes matemáticas como el número Π o e . Para el tratamiento de números complejos se debe utilizar el módulo *cmath*.

3.4.3. NumPy

NumPy [25] es un paquete de computación científica para Python. Entre las herramientas que nos ofrece podemos encontrar: métodos y objetos que traten arrays n-dimensionales de forma potente (útiles también para el trabajo con matrices), métodos muy útiles de álgebra lineal, transformadas de Fourier, herramientas de selección aleatoria de números... entre otros.

3.4.4. Tkinter

Se considera la biblioteca para el desarrollo de interfaces gráficas más populares mediante en el lenguaje de programación Python. Se considera estándar para éste y viene junto con las instalaciones del intérprete en las versiones para MS Windows. Tkinter proporciona una rápida y potente forma de desarrollo de aplicaciones con interfaz gráfica mediante la orientación a objetos. Donde deberemos crear una ventana e ir añadiendo diferentes elementos como: botones, lienzos, entradas, listas, menús, textos... También se facilita la configuración de estilos para cada uno de éstos.

3.4.5. PyMongo

PyMongo [26] es una biblioteca que contiene herramientas para el trabajo con la Base de Datos MongoDB, además es la vía más recomendable para el tratamiento de ésta mediante Python. Como casi todas las librerías desarrolladas para Python, se pueden consultar la documentación, código y contribuciones mediante los repositorios públicos.

Las funciones son muy similares a las que se utilizan para ejecutar directamente en el servidor. Hay que tener en cuenta la devolución en paquetes de Mongo para evitar cualquier fallo de falta de memoria. Para ello se utilizan cursores sobre los que se puede iterar.

3.4.6. Networkx

Networkx [27] es un paquete de software que facilita herramientas para la creación, manipulación y estudio de estructura, dinámicas y funciones sobre redes complejas. Permite la construcción de grafos de forma aleatoria o incrementalmente, además tiene un buen funcionamiento para realizar operaciones con grafos de gran escala del mundo real, por ejemplo, con 10 millones de nodos y 100 millones de relaciones [28].

4. Descripción del proyecto

En este apartado presentaremos los requerimientos del proyecto. Se analizará también la funcionalidad que será ofrecida por la aplicación.

Como hemos dicho anteriormente, el propósito de este proyecto es el de desarrollar un conjunto de herramientas que faciliten el tratamiento de datos provenientes de redes sociales para su análisis, en concreto Twitter. Esto conlleva varias partes: definir qué queremos, cómo lo vamos a almacenar y que análisis vamos a automatizar.

4.1. Requisitos

En casi todos los mecanismos y procesos del Big Data, el procesado de datos es una parte muy importante. Es por ello que una zona delicada es la relativa a la Ingeniería de Datos: extracción y pulido. Esto nos define un punto por el que comenzar.

1. Necesitamos un módulo que nos facilite el acceso a la información de Twitter y a través del cual podamos realizar una descarga de tuits sobre un hashtag o palabra clave. Algunos de los requisitos clave en este apartado son:
 - a. Se debe poder realizar un seguimiento y descarga de varios rastreos a la vez. Por ejemplo, si queremos escuchar¹ sobre “#Fútbol” y a la vez “#RealMadrid”.
 - b. El usuario deberá poder contabilizar o ver el número de tuits que tiene almacenados en cada momento.
 - c. El usuario deberá poder acceder a la información guardada de anteriores escuchas como si de una lista se tratara.
 - d. El usuario podrá concluir escuchas y retomarlas cuando quiera. Para esto se facilitarán los controles de “Pausa” e “Iniciar” o similares.
 - e. Se podrán configurar fácilmente las claves de acceso necesarias a la API de Twitter mediante algún menú de configuración.

Otro de los puntos fuertes de las redes sociales, como hemos dicho anteriormente, es la gran cantidad de información que se genera y que puede ser estudiada con técnicas de estadística descriptiva muy básicas. Para ello deberemos desarrollar un conjunto de facilidades que nos permitan:

2. Realizar un estudio básico sobre datos relacionados con los tuits. Esto nos dará mucha información acerca de la naturaleza de un hashtag o palabra clave. Para ello los principales segmentos de estudio son:
 - a. Proporcionar al usuario una visión temporal de evolución de la escucha. Esto es una gráfica del número de tuits con respecto al tiempo. Podremos verlo en

¹ Se utiliza “escuchar” como traducción literal del objeto listener, el cual se encarga de obtener todos los documentos en la API de Twitter que contengan una palabra clave.

distintas franjas temporales y el usuario debe poder seleccionar la que desee entre: 1 hora, 5 minutos o 1 minuto, principalmente.

- b. Realizar un estudio básico de frecuencias sobre las palabras y términos utilizados en esos documentos. Por ejemplo, palabras más utilizadas o hashtags más utilizados y cuántas veces.
- c. Presentar un estudio de frecuencias sobre los usuarios. Esto implica poder conocer cuántos usuarios han participado en el topic seleccionado y cuántas veces.

Una vez hemos planteado los cimientos del proyecto podemos atravesar la barrera hacia un estudio de sentimiento. En este caso vamos a proporcionar información muy útil, que también puede ir ligada a los dos anteriores apartados:

- 3. Utilizando las herramientas matemáticas descritas acerca de la clasificación de textos y de sentimiento mediante los métodos relacionados con el aprendizaje bayesiano y con el análisis semántico latente deberemos poder realizar las siguientes tareas:
 - a. Desarrollar un módulo individual en el que el usuario pueda introducir documentos, clasificarlos manualmente en 5, 3 o 2 opciones.
 - b. El usuario debe ser capaz de realizar un guardado de estas clasificaciones y documentos a clasificar para posteriores usos.
 - c. Uno de los fines de este módulo deberá ser brindar varias opciones de clasificación al usuario, principalmente las repasadas en este documento. Mediante clasificador Naive-Bayes o LSA.
 - d. Se deberá poder realizar un test de aciertos en el que los textos clasificados se probarán de forma que se introduzcan todos menos uno en el estadio de entrenamiento, a continuación, se clasificará el restante. La salida devuelta deberá ser el porcentaje de congruencias entre la clasificación esperada y la obtenida.

Otro de los elementos centrales del proyecto es el estudio mediante redes. Éstas nos permitirán descubrir mucha información relativa a la topología de las conexiones entre usuarios y tuits en el seguimiento. Para ello sería conveniente que el usuario pudiera tratar éstos de una forma casi tangible.

- 4. Basándonos en aplicaciones previas, podemos desarrollar una herramienta que nos ayude en el manejo de las redes formadas a partir de un seguimiento.
 - a. Las redes formadas se basarán en tratar como nodos a los usuarios y la existencia de un tuit como respuesta será una arista. Si un usuario no ha recibido ni realizado réplicas a otros tuits se le considerará aislado y se representará sólo con un nodo.
 - b. Se deberá poder guardar y cargar una red mediante ficheros o archivos.
 - c. El usuario podrá identificar cada nodo mediante algún tipo de acción que visibilice el identificador de éste.
 - d. Se podrá acceder a la información básica de cada nodo acerca de la cantidad de vecinos que tiene y los distintos grados de centralidad.

- e. Deberemos poder seleccionar un clúster a partir de un nodo que pertenezca a éste. Además, podremos estudiarlo en un entorno diferente al de la red principal.
- f. El usuario tendrá la opción de colorear los distintos nodos de la red de varias formas: mediante el mismo color para los nodos de cada clúster o mediante los grados de cada nodo.
- g. Será posible obtener información acerca de la red, como número de nodos, de aristas, distribución de grados o número de clusters, principalmente.
- h. Podrá organizar la red mediante el ratón de forma que arrastre los nodos a su voluntad para una cómoda visualización de la red.

Una vez hemos planteado estos requerimientos sobre la aplicación de forma independiente, debemos considerar aquellos que aúnan varios de los módulos.

5. A partir de los módulos creados de forma independiente, se desarrollarán los mecanismos pertinentes para las conexiones entre ellos. Esto permitirá al usuario:
 - a. Poder utilizar los tuits descargados y guardados para su clasificación de sentimiento. Para ello, el usuario deberá definir qué clasificador utilizar de entre los disponibles y podrá obtener una prueba de aciertos descrita en el punto 1.
 - b. Se podrá crear la red pertinente a partir de un seguimiento, como se ha descrito en el punto 4. Esto también se puede realizar en directo mientras se procesa la escucha, la red se comienza a crear a partir de que el usuario pulse “Inicio”.
 - c. Se deberán poder crear redes que tengan en cuenta el sentimiento de cada tuit. Esto nos permitirá la fácil visualización de una red donde el color de la arista es un factor que influye, ya que representa la polaridad obtenida.
 - d. Se podrán visualizar datos acerca del estudio de sentimiento, como número de tuits en cada tipo, porcentaje de tipo o porcentaje por usuario de tuits enviados de cada tipo, entre otros.

4.2. Diseño

Revisados los requisitos y las distintas funciones que necesitaremos desarrollar el siguiente paso es el planteamiento de la aplicación en módulos independientes, que, aunque ya hayan sido descritos con más o menos concreción, es necesario definir claramente.

5. Interfaz y módulo de control de descarga de datos, creación de estudio de sentimientos y redes a partir de éstos.
6. Biblioteca con las llamadas a base de datos necesarias para la obtención de los tuits.
7. Módulo de clasificación de documentos según sentimiento.
8. Módulo para el manejo y análisis de Redes.

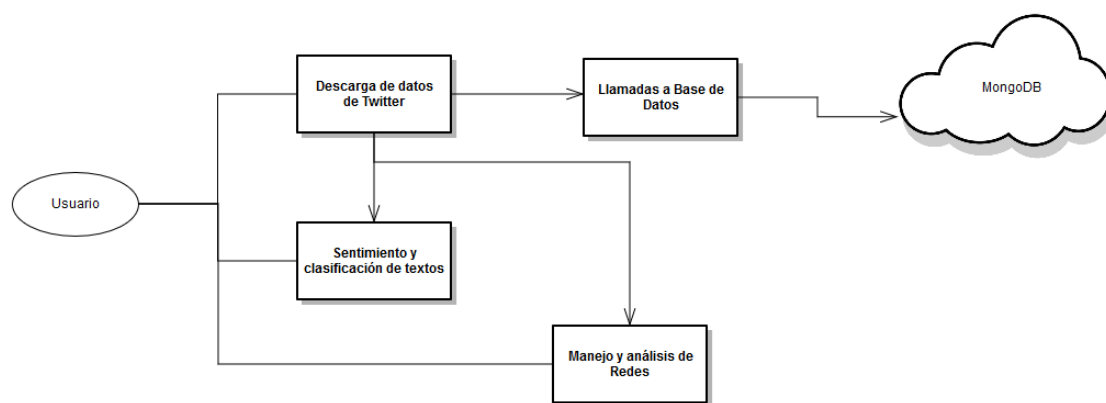


Ilustración VIII Representación de los distintos módulos que compondrán la aplicación

5. Implementación

En este capítulo describiremos las estructuras y metodologías necesarias para desarrollar la aplicación. Esto implica la organización de los datos y las rutas que siguen en su tratamiento, los métodos en los distintos módulos.

5.1. Datos obtenidos de Twitter

Como se ha especificado, los datos devueltos por Twitter tras la realización de peticiones serán recibidos como documentos con formato JSON. Dado que la principal fuente de datos será mediante la creación de *Stream Listeners* que devuelvan tuits destacaremos los campos que creemos necesarios [23]

- *id* o *id_str* del tuit. Representación del identificador único del tuit obtenido. El primero es un entero y el segundo está en formato string o texto.
- *in_reply_to_screen_name* o *in_reply_to_user_id* / *in_reply_to_user_id_str*. Si no se encuentra vacío o con valor nulo estamos hablando de una réplica y nos aporta el nombre o el identificador de usuario, respectivamente, al que hemos respondido.
- *in_reply_to_status_id* o *in_reply_to_status_id_str*. Similar al anterior, pero en este caso si es una réplica nos aportará el *id* del tuit al que se ha contestado.
- *retweet_count*. Nos indica el número de veces que ha sido retuiteado un tuit.
- *favourite_count*. Indica cuántas veces se ha presionado “Me gusta” sobre el documento.
- *text*. Un dato de tipo String que contiene el mensaje enviado en el tuit.
- *user*. Información que se obtiene relativa al usuario que ha realizado la publicación del tuit.
- *created_at*. Fecha en la cual se creó el tuit.

Es interesante realizar también estudios acerca de los usuarios y las posibles relaciones que tengan con determinados tuits. Para ello, podemos hacer peticiones que nos permitan obtener información concreta de un usuario a partir de su identificador. Esto nos devolverá nuevamente un documento JSON, del cual destacaremos los siguientes campos [23]

- *created_at*. Fecha en la cual se creó la cuenta.
- *description*. Texto escrito por el usuario que describe su cuenta.
- *id* o *id_str*. Similar al identificador de tuit.
- *name*. Nombre del usuario como lo ha definido
- *screen_name*. Alias que el usuario utiliza para identificarse. Es único pero puede cambiarse, por lo que se recomienda el uso del identificador como elemento principal para distinguir a los usuarios.
- *status*. Indica si es posible el último tuit o retuit que ha realizado el usuario.
- *statuses_count*. Cantidad de tuits (incluyendo los retuits) que ha realizado el usuario.

Además mediante la API podemos realizar búsquedas sobre otras relaciones relativas a los tuits y a los usuarios. Podremos obtener todos los tuits y retuits realizados por una cuenta, del más nuevo al más antiguo, recorriéndolos mediante un cursor que devuelve una lista con un tope de elementos indicado a partir del que continuar en siguientes peticiones. Además, es interesante poder obtener los followers. Esto nos puede permitir por ejemplo, realizar una red básica sobre un usuario y ver cuáles de los usuarios que le siguen se siguen entre sí.

5.2. Guardado de los Datos

Después de revisar qué datos son los que podemos obtener desde la red social, nos plantearemos cuáles deberemos guardar y cómo. Para ello lo más fácil e intuitivo, y es que por eso se ha escogido MongoDB, es mantener una estructura parecida a la de entrada. Podemos plantear dos colecciones.

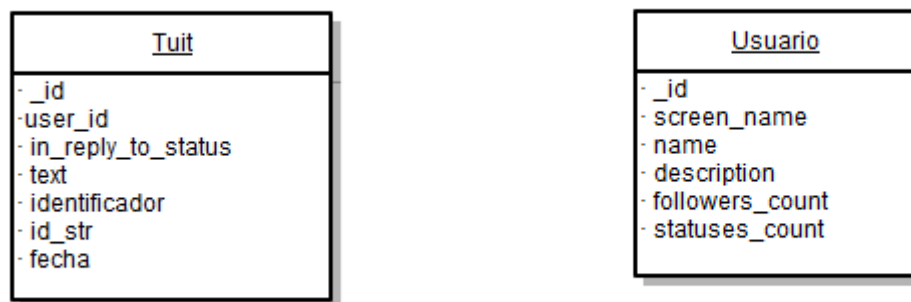


Ilustración IX Representación de las colecciones creadas en MongoDB

Si bien es cierto que podemos no guardar los datos de los usuarios puesto que podemos realizar una petición cuando queramos obtenerlos, podemos optimizar el proceso guardando los datos necesarios de cada usuario que realiza un tuit. En los procesos posteriores, una consulta vía http no consumirá poco tiempo, pero no cuando trabajemos con la descarga de información de 10.000 usuarios.

5.3. Módulo I: Administración de los datos desde Twitter

Planteamos ahora qué debemos desarrollar para conseguir que el usuario tenga un buen manejo sobre los seguimientos. Para esto describiremos diferentes metodologías y ciclos.

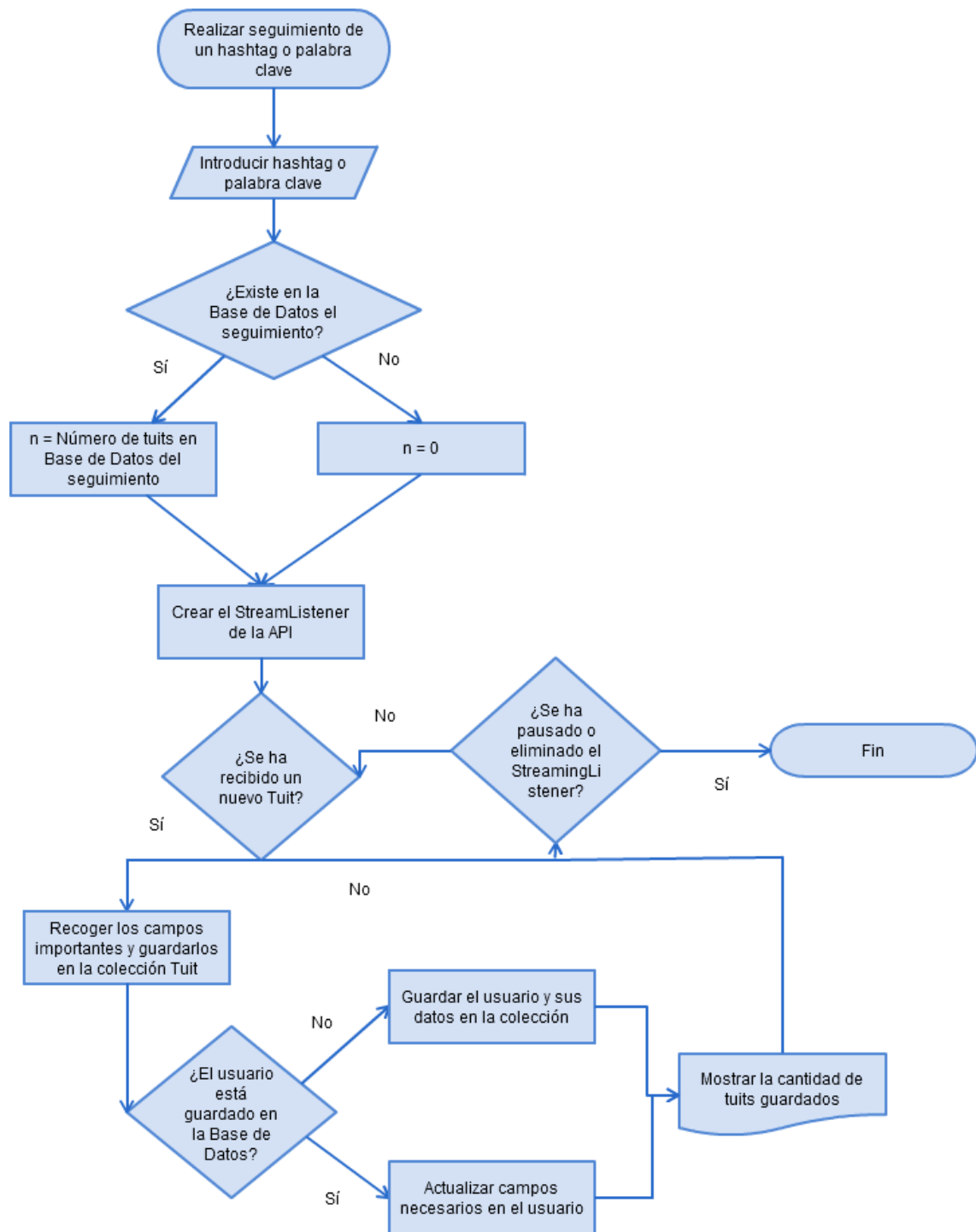


Ilustración X Representación del ciclo de guardado de tuits mediante un StreamListener

5.4. Módulo II: Clasificación de los sentimientos en los textos

En este caso vamos a describir las rutas que se seguirá para poder realizar la clasificación de los textos en la aplicación.

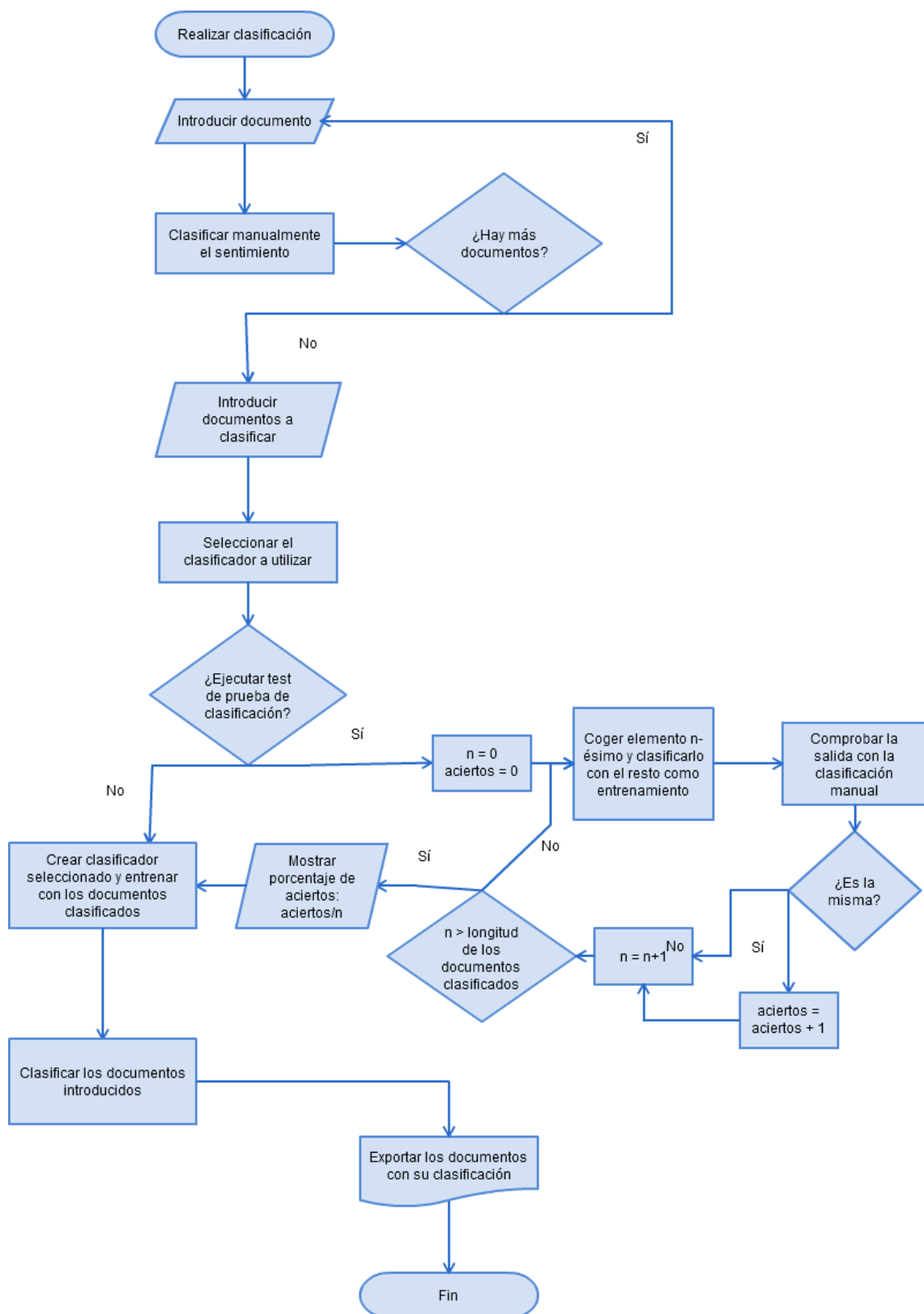
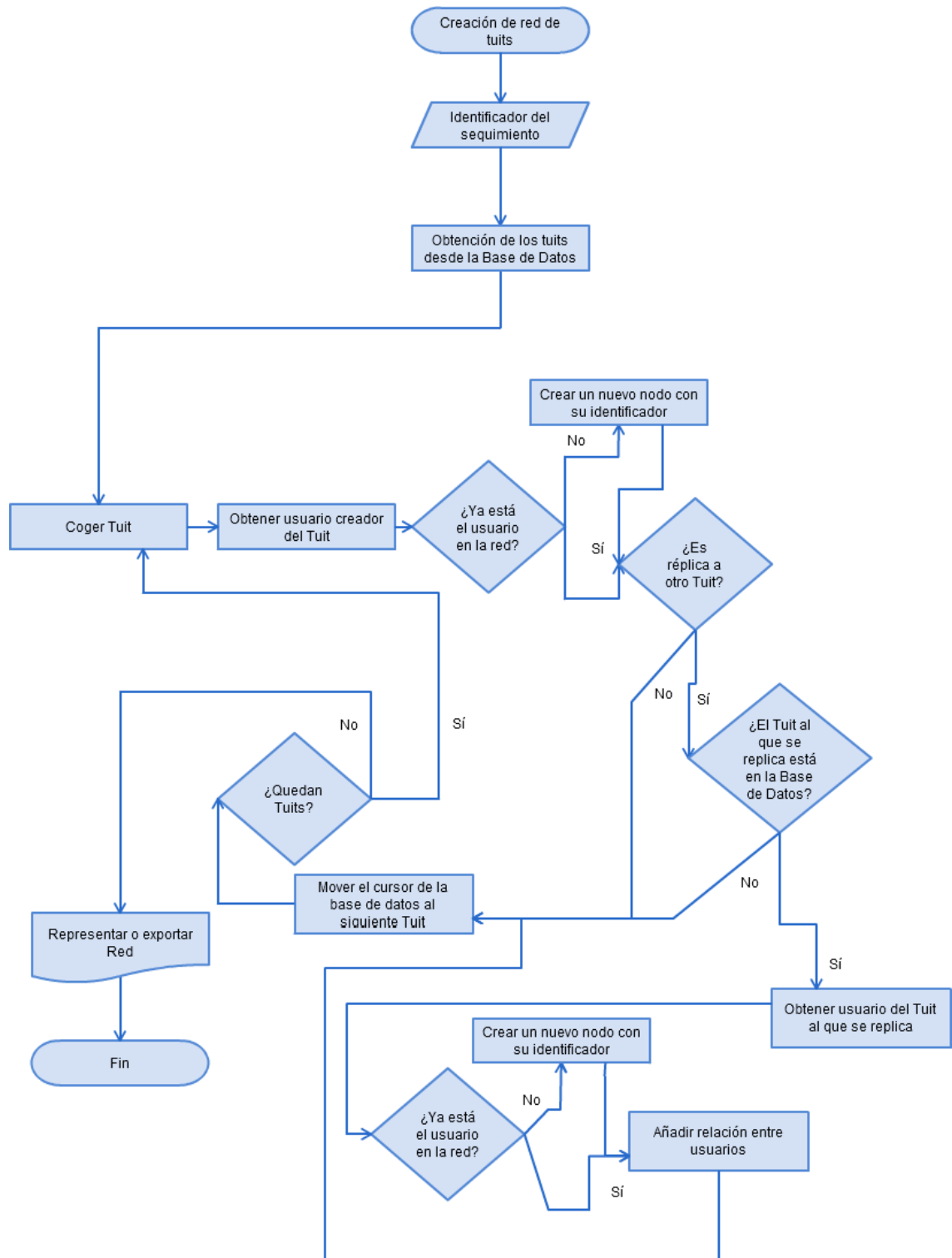


Ilustración XI Representación del manejo del módulo de clasificación de sentimiento

5.5. Módulo III: Manejo de Redes

En este módulo deberemos crear una estructura de red, como funcionalidad principal, a partir de un conjunto de tuits de un seguimiento escogido mediante el módulo de administración de datos descargados de Twitter. Procedamos a describir la construcción más básica de una red entre usuarios, se creará una relación si alguna vez han intercambiado réplicas.



Podemos determinar la metodología correspondiente a algunos mecanismos de estudio de las redes. Como puede ser la obtención del clúster al que pertenece un nodo seleccionado. Para ello aplicaremos un simple algoritmo de DFS, del inglés recorrido en profundidad.

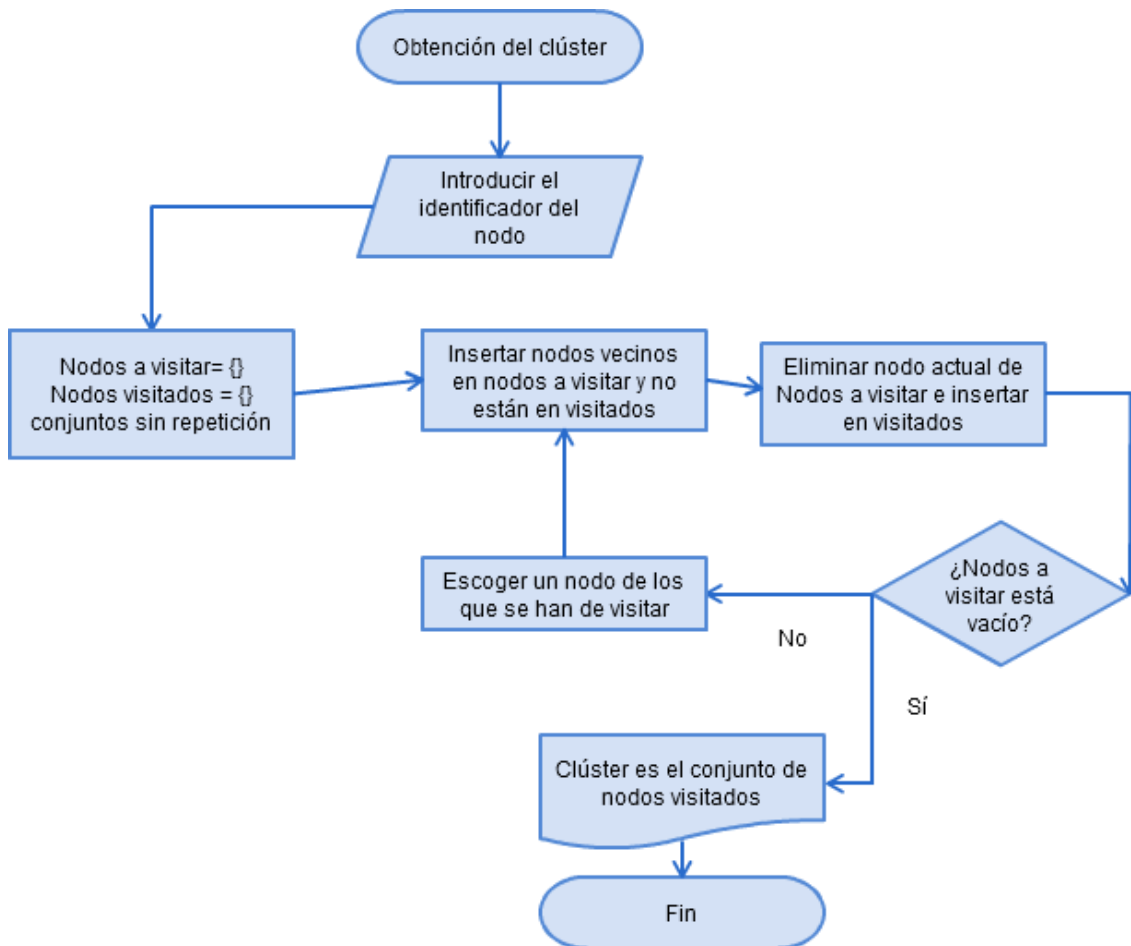


Ilustración XIII Método de obtención del clúster de un nodo

6. Planificación

En este capítulo deberemos analizar los principales pasos realizados en el proyecto, así como exponer su duración, inicio y fin.

6.1. Planning

La duración del proyecto deberá presentarse mediante las distintas fases que lo componen. Éstas se dividen a su vez en tareas, que serán el componente atómico representado en días de trabajo.

Primeramente, en la siguiente tabla se presenta la duración de la primera fase en el proyecto: el estudio previo.

Nombre de la Tarea	Duración	Fecha de Inicio	Fecha de Fin
Trabajo previo	11 días	09/10/2015	23/10/2015
Estudio del posible proyecto	6d	09/10/2015	16/10/2015
Estudio de viabilidad	5d	19/10/2015	23/10/2015

Tabla 4 Duración del estudio previo

A continuación, el siguiente bloque se dedica a la revisión de documentación acerca de los métodos matemáticos y su aplicación en el proyecto, así como el estudio del lenguaje de programación y la base de datos adecuada teniendo en cuenta los algoritmos que se deberán implementar.

Nombre de la Tarea	Duración	Fecha de Inicio	Fecha de Fin
Revisión bibliográfica y tecnologías	25 días	26/10/2015	27/11/2015
Documentación matemática que aborde el problema	20d	26/10/2015	20/11/2015
Selección del Lenguaje de Programación	5d	23/11/2015	27/11/2015
Selección de la Base de Datos	5d	23/11/2015	27/11/2015

Tabla 5 Duración de la revisión contextual sobre el problema

Posteriormente, tenemos la fase principal de Desarrollo de la Aplicación. El primer bloque de desarrollo es el del Módulo necesario para la obtención de datos desde Twitter. Su duración viene dada en la siguiente tabla.

Nombre de la Tarea	Duración	Fecha de Inicio	Fecha de Fin
Módulo de Twitter	29 días	30/11/2015	07/01/2016
Análisis y Requisitos	5 d	30/11/2015	04/12/2015
Diseño	5 d	07/12/2015	11/12/2015
Diseño de la Base de Datos	2 d	10/12/2015	11/12/2015
Implementación	15 d	14/12/2015	01/01/2016
Pruebas	5 d	01/01/2016	07/01/2016

Tabla 6 Duración del desarrollo del Módulo de Twitter

El siguiente módulo, de Análisis del Sentimiento, tiene la representación de su duración en la siguiente tabla.

Nombre de la Tarea	Duración	Fecha de Inicio	Fecha de Fin
Módulo de Análisis de Sentimiento	70 días	30/11/2015	04/03/2016
Análisis y Requisitos	5 d	30/11/2015	04/12/2015
Diseño	5 d	02/02/2016	08/02/2016
Implementación	15 d	09/02/2016	29/02/2016
Pruebas	5 d	29/02/2016	04/03/2016

Tabla 7 Duración del desarrollo del Módulo de Análisis de Sentimiento

Después, el Módulo de Manejo de Redes. El planning de esta parte se muestra a continuación.

Nombre de la Tarea	Duración	Fecha de Inicio	Fecha de Fin
Módulo de manejo de Redes	27 días	07/03/2016	12/04/2016
Análisis y Requisitos	5 d	07/03/2016	04/12/2015
Diseño	5 d	11/03/2016	17/03/2016
Implementación	15 d	18/03/2016	07/04/2016
Pruebas	5 d	06/04/2016	12/04/2016

Tabla 8 Duración del desarrollo del Módulo de Administración de Redes

El esquema de la unificación de estos módulos, última fase del desarrollo principal de la aplicación, se presenta en la siguiente tabla.

Nombre de la Tarea	Duración	Fecha de Inicio	Fecha de Fin
Módulo de Análisis de Sentimiento	27 días	18/04/2016	24/05/2016
Diseño	5 d	18/04/2016	22/04/2016
Implementación	15 d	22/04/2016	12/05/2016
Pruebas	5 d	18/05/2016	24/05/2016

Tabla 9 Duración del desarrollo de la unificación de los módulos desarrollados

El siguiente bloque es el de las pruebas. Su duración y composición se describen a continuación.

Nombre de la Tarea	Duración	Fecha de Inicio	Fecha de Fin
Pruebas	21 días	01/08/2016	29/08/2016
Diseño de Pruebas	5 d	01/08/2016	05/08/2016
Realización de Pruebas	2 d	05/08/2016	08/08/2016
Corrección y mejoras	10 d	09/08/2016	22/08/2016
Pruebas finales	5 d	23/08/2016	29/08/2016

Tabla 10 Duración de la fase principal de pruebas

Como conclusión del proyecto, se dedica la última fase a la elaboración de la documentación correspondiente a la aplicación y al desarrollo.

Nombre de la Tarea	Duración	Fecha de Inicio	Fecha de Fin
Documentación	37 días	15/07/2016	05/09/2016
Memorias de los TFGs	37 d	15/07/2016	05/09/2016
Manual de usuario	10 d	22/07/2016	02/09/2016

Tabla 11 Duración de la fase de Documentación

Para resumir, las principales fases se visualizan en la siguiente tabla.

Nombre de la Tarea	Duración	Fecha de Inicio	Fecha de Fin
Trabajo previo	11 días	09/10/2015	23/10/2015
Módulo de Twitter	29 días	30/11/2015	07/01/2016
Módulo de Análisis de Sentimiento	27 días	18/04/2016	24/05/2016
Módulo de manejo de Redes	27 días	07/03/2016	12/04/2016
Revisión bibliográfica y tecnologías	25 días	26/10/2015	27/11/2015
Pruebas	21 días	01/08/2016	29/08/2016
Documentación	37 días	15/07/2016	05/09/2016

Tabla 12 Resumen de las tareas principales en el proyecto

6.2. Diagrama de Gantt

Se representa en la Ilustración XIV Diagrama de Gantt correspondiente al planning las tareas y sus duraciones detalladas anteriormente.

Planificación

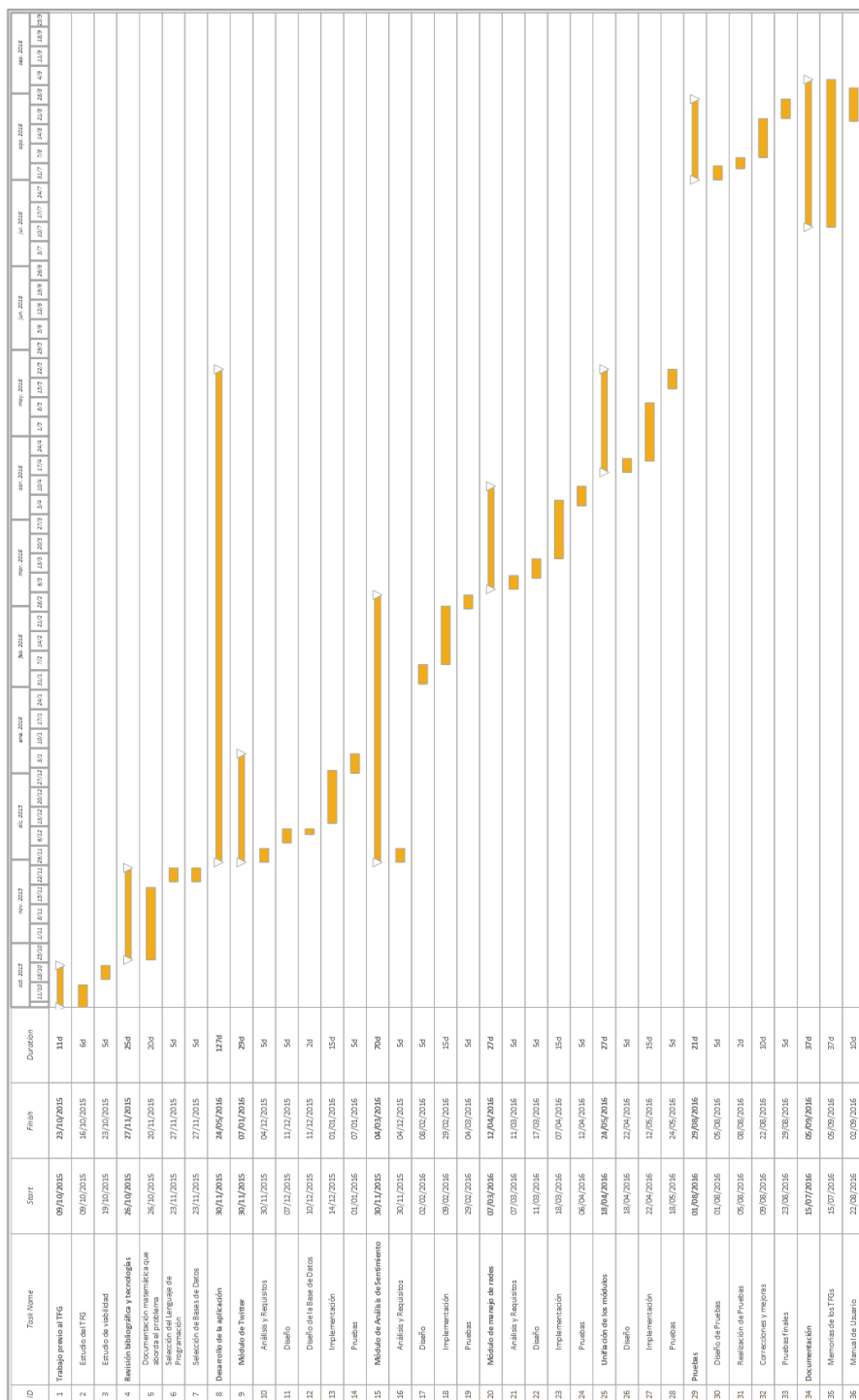


Ilustración XIV Diagrama de Gantt correspondiente al planning

7. Pruebas y verificación

En este apartado se definirán las pruebas a realizar sobre la aplicación que permitirán la búsqueda de fallos y posibles mejoras sobre la misma. Primero deberemos diseñarlas y especificarlas, especificando los objetivos de cada una y mostrando sus resultados y conclusiones.

Nos centraremos en evaluar la usabilidad de la aplicación siguiendo los estándares proporcionados por la norma ISO 9241-11, analizando la eficacia, eficiencia y la satisfacción de los usuarios al realizar las tareas con las distintas interfaces.

7.1. Diseño de pruebas

7.1.1. Infraestructura

La aplicación se ha desarrollado como aplicación de escritorio sobre el lenguaje de programación Python se utilizará la versión 3.5 sobre un ordenador de tipo PC portátil Hewlett Packard con sistema operativo Microsoft Windows 10 64 bits Education con las siguientes prestaciones: Intel(R) Core(TM) i7-2670QM CPU @ 2.20GHz, RAM: 6,00 GB, pantalla con resolución 1366 x 768 píxeles, tarjeta gráfica AMD Radeon HD 6700M Series.

Dado que la mayor parte del software para estudiar los mapas de calor en una página aplicación (clics) y el tiempo de interacción son de pago optamos por observar una grabación de la interacción del usuario para extraer los datos de estudio que se describirán posteriormente. Para esto utilizaremos el software Screencast-O-Matic (Versión de prueba) con el que guardaremos en formato vídeo la interacción de la pantalla.

7.1.2. Participantes

Los usuarios que participarán en la evaluación de la aplicación, serán usuarios de prueba, alguno de ellos posiblemente final. Su participación se hará de forma voluntaria sin beneficio monetario o de otro tipo por ello. De los participantes reportaremos su edad, sexo, nivel de educación y especialidad. Todos los usuarios deberán tener un conocimiento previo acerca de la red social Twitter y su funcionamiento, un filtro será que hayan tenido una cuenta en ésta anteriormente.

7.1.3. Protocolo

7.1.3.1. *Guiado por la aplicación y tareas*

A cada usuario se le dará un listado digital, formato PDF, con los ejercicios a realizar donde se describe su objetivo. Se han seleccionado un conjunto de tareas directamente relacionadas con los requisitos definidos en la fase de análisis. Éstas son, ordenadas por interfaz

- Fase de tareas 1: Módulo de Twitter
 - Tarea 1.1: Configuración de los datos de acceso a la API de Twitter. Además, se deberá probar la conexión. La finalización de esta tarea se efectuará al obtener una conexión correcta.
 - Tarea 1.2: Realizar un seguimiento de un Hashtag elegido por el usuario hasta haber almacenado más de 5 tuits. Esto incluye iniciarlo, observar la información de número de tuits almacenados y la pausa/finalización.
 - Tarea 1.3: Salir de la aplicación para volver a abrirla. Obtener el seguimiento realizado anteriormente e iniciar de nuevo. La tarea estará acabada cuando se pause el seguimiento.
 - Tarea 1.4: Obtener información acerca de los datos guardados: número de tuits y número de usuarios.
 - Tarea 1.5: Visualizar la evolución temporal y obtener el máximo de tuits por minuto realizados.
 - Tarea 1.6: Definir cuál ha sido el término más utilizado, así como otros hashtags mencionados.
- Fase de tareas 2: Módulo de clasificación de sentimiento.
 - Tarea 2.1: Agregar la frase de forma manual: "Esta frase me gusta", y clasificarla como positiva o negativa.
 - Tarea 2.2: Eliminar la frase del sentimiento asignado y clasificarla en el opuesto. Es decir, si se clasificó como negativa se deberá mover a positiva y viceversa.
 - Tarea 2.3: Introducir una nueva frase y clasificarla en el sentimiento con listado vacío, e introducir otra que se dejará tal cual. Realizar un guardado a un archivo y cerrar la aplicación.
 - Tarea 2.4: Abrir de nuevo la aplicación y el archivo recientemente guardado. Comprobar que las frases están en las mismas posiciones que en el paso anterior.
 - Tarea 2.5: Incluir las siguientes frases en un documento de texto. Importarlo a la aplicación las frases e importarlas a la aplicación. Clasificarlas todas al mismo sentimiento. Frases
 - El perro es bonito
 - El hombre es guapo
 - La mujer es bella
 - El edificio está bien construido
 - Tarea 2.6: Realizar un test de clasificación y definir el resultado.
 - Tarea 2.7: Realizar la clasificación automática de las frases restantes.
- Fase de tareas 3: Unificación y Redes
 - Tarea 3.1: Realizar una clasificación de sentimiento con el clasificador elegido por el usuario.
 - Tarea 3.2: Iniciar un seguimiento que muestre la red que se va creando, con un hashtag elegido por el usuario.
 - Tarea 3.3: Crear una red a partir de un Hashtag y obtener el número de nodos y de relaciones.
 - Tarea 3.4: Colorear mediante el grado de nodo. Obtener el identificador del nodo que mayor grado tiene y visualizar su clúster.
 - Tarea 3.5: Visualizar la información estructural de la red: distribución de grados, tamaño de mayor componente, diámetro.

- Tarea 3.6: Guardar como un archivo la red. Cerrar la aplicación y abrir el fichero creado. Comprobar que es la misma red. Opcional: exportar el archivo para aplicaciones Gephi²
- Tarea 3.7: Crear una red donde se tenga en cuenta el sentimiento de los tuits.

7.1.3.2. Puntos de evaluación

La siguiente tabla será rellenada con los datos obtenidos de la observación de los usuarios mientras realizaban las tareas

<i>Tarea</i>	<i>Éxito en Tarea</i>	<i>Clicks</i>	<i>Ayuda</i>	<i>Tiempo</i>
<i>Fase 1</i>				
<i>Tarea 1.1</i>				
<i>...</i>				
<i>Tarea 1.6</i>				
<i>Fase 2</i>				
<i>Tarea 2.1</i>				
<i>...</i>				
<i>Tarea 2.7</i>				
<i>Fase 3</i>				
<i>Tarea 3.1</i>				
<i>...</i>				
<i>Tarea 3.7</i>				
<i>Núm. Usuario</i>	<i>Sexo</i>	<i>Edad</i>	<i>Estudios</i>	<i>Especialidad</i>

Tabla 13 Tabla a rellenar tras la realización de la prueba

donde

- Éxito en Tarea. Indicación de si se ha cumplido o no el objetivo especificado en la descripción de la tarea en la lista. No cumplirlo significa que el participante ha abandonado la actividad sin el resultado esperado. Se debe especificar la razón. Esto nos permitirá descubrir posibles errores, bugs o uso no previsto de la aplicación por parte de algún usuario.
- Clicks. Número total de Clicks realizados durante la realización de la tarea. Podremos descubrir cómo de intuitiva es una interfaz con respecto al control con el ratón. Además de si las etiquetas en los menús son lo suficientemente descriptivos.
- Ayuda. Número de veces en las que el participante ha necesitado alguna ayuda en la tarea y de qué tipo.

² Software para el estudio y manejo de grafos y redes <https://gephi.org/>

- Tiempo. Diferencia entre la hora de inicio y la de final. Normalmente éste irá en escala de segundos o minutos. En caso de que la tarea no se complete no se tendrá en cuenta el tiempo.

Mediante la monitorización y grabación de la interacción con la aplicación se tratará de medir la eficiencia y la eficacia de las interfaces. Para ello se analizará la tasa de finalización con éxito de cada tarea y compararemos el tiempo y número de clicks invertidos con la siguiente tabla

<i>Tarea</i>	<i>Clicks</i>	<i>Tiempo (segundos)</i>
<i>Fase 1</i>	35	88
1.1	10	28
1.2	7	20
1.3	9	20
1.4	2	5
1.5	4	10
1.6	3	5
<i>Fase 2</i>	49	122
2.1	6	20
2.2	6	7
2.3	15	30
2.4	7	15
2.5	9	40
2.6	3	5
2.7	3	5
<i>Fase 3</i>	49	105
3.1	9	15
3.2	5	12
3.3	6	15
3.4	7	13
3.5	3	5
3.6	10	25
3.7	9	20

Tabla 14 Referencias de comparación de datos obtenidos en la monitorización de las pruebas

La Tabla 14 Referencias de comparación de datos obtenidos en la monitorización de las pruebas sido completada con datos muestrales procedentes de la realización por parte de los miembros del desarrollo de la aplicación. El conocimiento previo de la interfaz nos permite considerar éstos como el límite inferior o mínimo de cada tarea. Por ello añadiremos un margen al tiempo y al número de clicks, +15 segundos y +3 clicks.

En cuanto a la comunicación, en todas las pruebas se pide al participante la expresión de su opinión mientras utiliza la aplicación con el fin de que un miembro desarrollador que asiste en las pruebas pueda tomarlo como notas (Think Aloud).

Además, es necesario aprovechar el orden de las tareas para evaluar la facilidad de aprendizaje inherente a la aplicación. Esto es, los usuarios pares realizarán las tareas en este orden: Fase 1, Fase 2 y Fase 3, y los impares en el siguiente: Fase 2, Fase 3 y Fase 1. De esta forma, es valioso comparar el tiempo invertido, el número de clicks y las veces que ha consultado la ayuda en una misma tarea que se ha realizado con y sin experiencia previa a la aplicación viendo si disminuyen o se mantienen.

El grado de satisfacción del usuario es medido mediante la práctica del Think Aloud antes mencionado y el cuestionario que deberá ser completado como conclusión de las pruebas.

7.1.3.3. Cuestionario final

Las preguntas que realizaremos al finalizar serán respondidas por escrito en el computador proporcionado para la realización de la prueba mediante un fichero de texto.

1. ¿Le ha parecido fácil e intuitiva la interfaz a la hora de realizar las distintas tareas en el apartado de obtención de datos desde Twitter?
2. ¿Le ha parecido fácil e intuitiva la interfaz a la hora de realizar las distintas tareas en el apartado de análisis de sentimiento?
3. ¿Le ha parecido fácil e intuitiva la interfaz a la hora de realizar las distintas tareas en el apartado de manejo de redes?
4. ¿En algún momento se ha sentido inseguro o no tenía el control de la aplicación?
5. ¿Cree que los menús, botones y cómo se llega a ellos se encuentran correctamente dispuestos? ¿Cuáles modificaría?
6. ¿Le parece suficiente la ayuda aportada a lo largo de la aplicación?
7. ¿Cuáles son las características positivas de la aplicación?
8. ¿Cuáles son las características negativas de la aplicación?
9. Valore del 1 al 5, donde 1 es muy mala y 5 es muy buena, las interfaces relacionadas con cada fase según la interacción en la realización las tareas.
10. ¿Qué nuevas funcionalidades le gustaría poder cubrir?
11. Otros comentarios.

Las preguntas 1,2 y 3 representan una introducción al cuestionario de forma directa. Podemos ver con ellas, de nuevo, la relación en cuanto a facilidad de aprendizaje entre usuarios que han ejecutado las fases en distinto orden.

En la cuestión 4 podemos analizar la información visual que recibe el usuario, es decir, si realiza un control con el ratón pero no conoce sus efectos, esto se podría resolver con la mejora de iconos en el mouse y ToolTips. Similar con respecto a la pregunta posterior.

En la número 6, evaluamos si es necesaria la ayuda para las tareas. Además podemos anotar cuáles son más fundamentales que otras y encontrar mejoras para la aplicación. Es similar con respecto a las preguntas 8 y 10. El análisis sobre los puntos que el usuario considera como positivos en la aplicación también nos permitirá desarrollar mejoras teniéndolos en cuenta.

Mediante la pregunta 9, queremos diferenciar las posibles interfaces en cuanto a buena usabilidad o

mala. También, nos sirve como doble pregunta para la comprobación de una coherencia en las respuestas puesto que es parecida a la número 1.

7.2. Ejecución de pruebas y resultados

7.2.1. Introducción

Describimos a continuación el procedimiento seguido en las pruebas. Éstas se han realizado entre los días 6 y 8 de Agosto de 2016 en emplazamientos personales de los desarrolladores con un total de 6 voluntarios. Ésto permite el control del entorno por parte de los diseñadores de las pruebas, evitando además la necesidad de la relocalización de la infraestructura. Se intenta también que las características que puedan influir en los participantes externos al ordenador, como sonido ambiente o iluminación, sean similares para todos ellos.

7.2.2. Presentación

Como paso previo, el encargado de la realización de pruebas ha de presentar el proyecto explicando el recorrido a realizar en la sesión. Debemos incidir en la comodidad del participante para que pueda ejecutar todas las tareas sin necesidad de parar o descanso. A continuación, se le introduce al documento de guiado por la aplicación donde se listarán las tareas a realizar y en qué orden. Se procede a las tareas en sí mientras se anotan las posibles reacciones y pensamientos que se transmiten en voz alta, como se les ha pedido (Think Aloud). Sin embargo, esta práctica no se ha desarrollado con la obtención de mucha información puesto que la mayoría de reacciones estaban relacionadas con la búsqueda de botones con la lectura de las etiquetas en los menús. Finalmente, contestan al cuestionario de forma escrita mediante ordenador.

7.2.3. Guiado por la aplicación

Como se ha comentado antes, tras la presentación deberemos entregar al usuario el documento donde se listan las actividades a realizar. Para esto recordamos que el orden de las fases será alterado entre usuarios dependiendo de su número de participación en las pruebas. Siendo los impares los que realicen un guión de tipo A y los pares uno de tipo B. El guión A se compone de las partes: fase 1, 2 y 3, en este orden, y el B: fase 2, 3 y 1. En la primera tarea de la Fase 1, en la que el usuario debe conectarse mediante las claves de acceso de la API, tendrá éstas accesibles en el documento de tareas.

7.2.4. Análisis del cuestionario.

Por último, la realización de un cuestionario nos proporciona una forma de medir el grado de satisfacción de cada uno de los usuarios y global. Éste consta de 11 preguntas de las cuales se han obtenido las siguientes respuestas.

Pregunta 1

¿Le ha parecido fácil e intuitiva la interfaz a la hora de realizar las distintas tareas en el apartado de obtención de datos desde Twitter?

5 de los 6 participantes han respondido afirmativamente. Sin embargo, dos de los participantes (33,3%) han tenido problemas con respecto a la Tarea 1.5, donde indican que el doble click para la visualización de la información no les parece la más intuitiva. Uno de ellos realizó la ejecución de esta fase como la última y esperaba una opción con el despliegue del menú de opciones con el click derecho. 1 de los participantes (16,66%) afirma que lo primero que ha buscado ha sido una opción de “Ayuda” en los menús y la califica de importante.

Pregunta 2

¿Le ha parecido fácil e intuitiva la interfaz a la hora de realizar las distintas tareas en el apartado de análisis de sentimiento?

El 100% han afirmado que esta interfaz les ha resultado fácil e intuitiva para la realización de las tareas. Ha habido problemas en una de las participaciones en relación al control erróneo en las tareas donde se necesitaba clasificar una sentencia, pero se ha presionado la opción: “Eliminar”, en lugar de la que se quería: “Negativo”. 2 de los 6 participantes (33,33%) han afirmado que una vez realizadas tareas anteriores les han parecido más intuitivos los menús.

Pregunta 3

¿Le ha parecido fácil e intuitiva la interfaz a la hora de realizar las distintas tareas en el apartado de manejo de redes?

2 de los 6 participantes (33,3%) han descrito como difícil la realización de las tareas mediante ésta interfaz. Ambos participantes pertenecen al grupo donde se ha realizado la Fase 1 al final. 1 de los participantes (16,66%) la ha descrito como medianamente difícil, aunque no tanto como la primera parte por el aprendizaje.

Pregunta 4

¿En algún momento se ha sentido inseguro o no tenía el control de la aplicación?

2 de los participantes han afirmado sentirse inseguro por falta de información en las respuestas visuales en la Fase 3 (Interfaz de manejo de Redes).

Pregunta 5

¿Cree que los menús, botones y cómo se llega a ellos se encuentran correctamente dispuestos? ¿Cuáles modificaría?

4 de los 6 participantes (66,66%) afirman que los menús son correctos. 2 de ellos que se deberían modificar algunos botones para una mayor intuición con la lectura de su texto. Éstos son:

- Crear Stream,
- Coloreado,

- Distribución aleatoria.

Además 1 de los participantes aporta como idea el uso del icono en forma de mano en el puntero si éste se sitúa sobre un elemento con el que se puede interactuar, y de otro tipo si se puede mostrar un menú con el click-derecho.

Pregunta 6

¿Le parece suficiente la ayuda aportada a lo largo de la aplicación?

Todos los participantes han contestado afirmativamente. Uno de ellos ha añadido además que esperaba encontrarse con una sección de ayuda dentro de la aplicación con los puntos básicos de funcionamiento.

Pregunta 7

¿Cuáles son las características positivas de la aplicación?

Las ideas fundamentales aportadas en esta pregunta han sido

- Simplicidad en la interfaz. Lo que se traduce en un rápido acceso a las funciones que se quieren realizar.
- Los colores utilizados.
- El nivel de aprendizaje que se adquiere sobre la aplicación es relativamente alto en un tiempo pequeño.
- La interfaz de sentimientos es la más fácil de usar por su interfaz minimalista.

Pregunta 8

¿Cuáles son las características negativas de la aplicación?

Las ideas fundamentales aportadas en esta pregunta han sido

- Esta simplicidad se traduce en una falta de configuración en algunas interfaces como la de sentimiento, donde se puede establecer una opción para elección del número de sentimientos.
- En la creación de red a partir de las conversaciones, se utiliza el id de twitter, poco manejable si queremos identificar a un usuario para posibles análisis de su perfil.
- La posibilidad de configuración visual en la interfaz de redes como figuras y colores en la representación.
- En la interfaz de sentimiento el click derecho para la clasificación es poco intuitivo. Una opción podría ser arrastrar los textos
- Selección de dos nodos en la red y cálculo de camino más corto.
- Selección de un nodo en la red y visionar su influencia mediante la distancia.
- Haber recibido un mensaje "No responde del sistema"
- Algunos tiempos de espera son excesivos si no se recibe ningún tipo de mensaje

Pregunta 9

Valore del 1 al 5, donde 1 es muy mala y 5 es muy buena, las interfaces relacionadas con cada fase según la interacción en la realización las tareas.

Las calificaciones obtenidas se muestran en la siguiente tabla

	Particip.1	Particip.2	Particip.3	Particip.4	Particip.5	Particip.6	Promedio
Tarea 1.1	3	5	2	4	2	4	3.33
Tarea 1.2	5	5	5	5	3	5	4.66
Tarea 1.3	5	5	5	5	3	4	4.5
Tarea 1.4	4	5	5	4	3	4	4.166
Tarea 1.5	2	3	3	3	2	4	2.833
Tarea 1.6	1	3	1	2	2	5	2.833
Tarea 2.1	5	3	4	3	5	4	4
Tarea 2.2	5	4	4	5	5	4	4.5
Tarea 2.3	5	4	5	5	5	5	4.833
Tarea 2.4	5	3	5	3	4	4	4
Tarea 2.5	5	4	4	5	4	5	4.5
Tarea 2.6	4	3	5	2	5	3	3.66
Tarea 2.7	4	5	3	4	4	4	4
Tarea 3.1	3	2	4	4	3	3	3.166
Tarea 3.2	4	1	5	2	5	2	3.166
Tarea 3.3	4	3	3	3	4	3	3.33
Tarea 3.4	3	2	5	3	4	4	3.5
Tarea 3.5	5	5	5	4	5	5	4.833
Tarea 3.6	4	4	3	5	5	4	4.166
Tarea 3.7	5	4	4	5	4	3	4.166

Tabla 15 Puntuaciones obtenidas como respuesta a la pregunta 9 del cuestionario

Pregunta 10

¿Qué nuevas funcionalidades le gustaría poder cubrir?

Las principales ideas que han aportado los participantes han sido

- La creación de redes de seguimiento especificando un usuario.
- La posibilidad de eliminación de nodos concretos en la red.
- La búsqueda del nombre de usuario con su id sin recurrir a internet.
- *Saber si alguien se lleva bien con otra persona.*
- *Métodos o herramientas que detecte los distintos clústeres y los separe visualmente.*

Pregunta 11

Otros comentarios.

2 de los participantes (33,33%) han afirmado que algunas de los enunciados de las tareas les han parecido confusos y por ello han utilizado la ayuda del supervisor de las pruebas.

7.2.5. Análisis de las interacciones

En esta sección realizaremos un análisis y estudiaremos las posibles conclusiones que se pueden obtener de los datos medidos en las distintas pruebas. En concreto, la eficiencia y eficacia medida en tiempo de duración de la prueba, clicks realizados y éxito o no en la consecución del objetivo establecido. Los dos primeros podremos compararlos con los mínimos establecidos por las ejecuciones realizadas con miembros del equipo de desarrollo.

Los tiempos empleados, medidos en segundos, por parte de cada uno de los participantes en las diferentes tareas han sido los siguientes

	Particip.1	Particip.2	Particip.3	Particip.4	Particip.5	Particip.6	Promedio
Tarea 1.1	83	40	54	43	65	35	53,33
Tarea 1.2	47	35	30	30	40	34	36
Tarea 1.3	35	50	27	41	44	29	37,66
Tarea 1.4	120	70	40	80	50	31	65,166
Tarea 1.5	50	20	35	28	30	24	31,166
Tarea 1.6	90	40	37	25	40	15	41,16
Tarea 2.1	35	18	26	30	25	29	27,166
Tarea 2.2	5	15	9	18	12	25	14
Tarea 2.3	50	33	41	51	46	40	43,5
Tarea 2.4	15	25	20	18	21	26	20,833
Tarea 2.5	60	71	59	47	65	55	59,5
Tarea 2.6	10	12	15	18	13	20	14,66
Tarea 2.7	5	10	7	11	10	10	8,33
Tarea 3.1	80	60	55	61	45	25	54,33
Tarea 3.2	30	24	20	25	32	23	25,66
Tarea 3.3	30	25	23	27	30	32	27,833
Tarea 3.4	50	10	18	15	35	12	23,33
Tarea 3.5	64	10	20	30	25	28	29,5
Tarea 3.6	**	35	31	**	40	30	34
Tarea 3.7	90	58	40	50	45	33	52,66

Tabla 16 Tabla con los tiempos de ejecución de las tareas

Las tareas en las que se ha definido un doble asterisco (“**”) son las que han encontrado un error irrecuperable del sistema.

A continuación, vemos los datos relativos a los clicks que se han realizado por participante en cada tarea

	Particip.1	Particip.2	Particip.3	Particip.4	Particip.5	Particip.6	Promedio
Tarea 1.1	40	50	45	43	40	39	42,833

Tarea 1.2	15	20	16	15	14	13	15,5
Tarea 1.3	16	20	15	21	16	10	16,33
Tarea 1.4	40	11	17	16	30	9	20,5
Tarea 1.5	20	16	21	18	15	12	17
Tarea 1.6	20	19	15	20	21	8	17,166
Tarea 2.1	10	24	14	16	11	21	16
Tarea 2.2	30	35	25	30	20	40	30
Tarea 2.3	10	25	20	21	16	23	19,166
Tarea 2.4	5	12	10	19	11	8	10,833
Tarea 2.5	20	15	16	20	12	22	17,5
Tarea 2.6	3	7	5	5	4	6	5
Tarea 2.7	3	5	4	6	5	3	4,33
Tarea 3.1	20	25	18	26	16	22	21,166
Tarea 3.2	15	20	16	21	18	15	17,5
Tarea 3.3	5	7	6	5	8	6	6,166
Tarea 3.4	17	14	12	20	11	7	16,5
Tarea 3.5	20	10	16	5	15	10	12,66
Tarea 3.6	**	21	15	**	30	20	21,5
Tarea 3.7	18	12	20	18	25	21	19

Tabla 17 Tabla del número de clicks realizados por participante en cada tarea

Junto con las calificaciones obtenidas por parte de los usuarios en el cuestionario final, podemos observar que las 1.4 y 1.5 han presentado un cambio significativo en cuanto a confortabilidad del usuario utilizando la aplicación. Éstas tareas están basadas en la obtención de información bruta del seguimiento: número de usuarios que han participado, número de tweets y evolución temporal de éstos. Sin embargo, el principal problema, y así es como han declarado algunos de los participantes en el cuestionario, es el de la falta de ser intuitivo al tener que realizar un doble clicado sobre el ítem del listado de seguimientos y posteriormente sobre la gráfica básica de evolución temporal.

Otra de las tareas que se refleja cómo menos fluida es la 3.1. En ésta se ha pedido al participante que realice una clasificación de sentimiento en un seguimiento seleccionado. La diferencia entre el tiempo de comparación y la media es de casi 40 segundos. Observando el desarrollo en vivo del usuario, se han percibido algunos problemas con el tratamiento de apertura de ficheros con clasificaciones previas. Esto ha obligado a dos de los participantes a cerrar y volver a abrir la aplicación.

7.2.6. Errores

Los errores reportados, o bien porque el usuario considera una mejora o porque se ha encontrado un bug que ha influido negativamente en la experiencia, se resumen en la siguiente tabla:

Ubicación	Descripción	Veces encontrado	Importancia
Interfaz de Sentimientos			
	Fallo por carácter extraño en texto	2	Alta, puesto que es un fallo que quiebra la interfaz.
	Mensaje “No responde” del sistema al realizar click sobre la ventana después de solicitar el análisis de palabras	2	Alta, puesto que el usuario necesita cerrar y abrir de nuevo la aplicación.
Interfaz de datos de Twitter			
	Fallo en la visualización de la evolución temporal. Sale mal ordenado por fecha.	1	Media, puesto que se muestra la fecha exacta en la selección de cada franja temporal
	Fallo en la apertura de archivo para la clasificación de sentimiento sobre un seguimiento.	2	Alta, no se puede continuar sin el archivo específico ya que contiene los elementos de entrenamiento
Interfaz de manejo de redes			
	Fallo visual donde algunos nodos se han quedado inaccesibles.	2	Baja, ya hay una opción desarrollada para la distribución aleatoria de los nodos por la pantalla.

Tabla 18 Fallos reportados en la realización de las pruebas

8. Conclusiones y trabajo futuro

Trabajados los fundamentos matemáticos del proyecto y su implementación como herramientas software podemos extraer ciertas conclusiones y estudiar cuáles son las líneas de trabajo futuras.

8.1. Conclusiones

Las principales conclusiones que podemos extraer con el proyecto son las relacionadas con las tecnologías empleadas para desarrollarlo. Python es un excelente y muy utilizado

lenguaje de programación en proyectos de corte científico, y sus librerías han facilitado gran parte de las tareas en especial las relacionadas con la realización de operaciones sobre redes y el análisis de centralidad. De forma similar sucede con la implementación del LSA mediante la Descomposición en Valores Singulares y el clasificador Naive-Bayes.

Por otro lado, la librería Tkinter utilizada en la implementación la GUI ha dificultado el proceso de mejorar la apariencia de la aplicación. Como se ha descrito en las líneas de trabajo futuro, el proyecto puede continuarse en forma de adaptación como aplicación web, de esta forma tendría mayor nivel de configuración mediante la tecnología CSS, HTML y Javascript.

8.2. Líneas de trabajo futuras

Una vez los requisitos han sido ejecutados y verificados con pruebas, la aplicación está finalizada. El siguiente paso consiste en la implementación de nuevas funcionalidades o en la mejora de las mismas.

Algunas de las posibles líneas de trabajo futuro son:

- Mejora de los tiempos en la creación de la red y en los distintos análisis de sentimiento.
- Desarrollo de una aplicación web escalable. Esto nos permite utilizar los módulos desarrollados y su implementación en un servidor, por ejemplo con Django, donde sólo haya que realizar la inversión sobre el front-end de la web-app.
- Mejorar la escalabilidad de los análisis de sentimiento. Uno de los inconvenientes, principalmente en el Latent Semantic Analysis, es la gestión de la memoria por almacenar grandes cantidades de información, por ejemplo, matrices, y realizar operaciones con ellas.
- Es interesante la implementación de mejoras visuales en el apartado de redes. Por ejemplo:
 - Representación de los nodos mediante relaciones de repulsión y atracción dependiendo de las conexiones entre éstos (Campos de fuerzas)
 - Representación de centralidad representándola mediante la diferencia de tamaños en los nodos. Cuanta más centralidad tenga un vértice más grande aparecerá en la representación.

Manual de Usuario

Este documento está dividido en cuatro partes. La primera describe las instrucciones y requisitos de la aplicación. Las tres últimas se centrarán en cada uno de los diferentes módulos que componen la aplicación. Serán facilitadas imágenes y aclaraciones acerca de la interfaz relacionadas con los menús y botones.

Requisitos e instalación

Para el funcionamiento de la aplicación se deberán cumplir los siguientes requerimientos hardware y software:

- Tener instalado Python 3.5.
- Heber instalado las librerías de Python: Numpy (www.numpy.org), Tweepy (www.tweepy.org), PyMongo (<https://api.mongodb.com/python>) y Networkx (<https://networkx.github.io>).
- MongoDB configurado en el puerto número: 27017, y creada la base de datos: 'twitterAnalysis', y las colecciones: 'tweet' y 'usuario' (Sin comilla simple).
- Sistema Operativo Windows, Mac o Linux.
- Recomendable CPU mayor de 1.5 GHz y memoria RAM mayor de 512 MB.

Los componentes software que deben encontrarse en la carpeta de la aplicación se resumen en la siguiente estructura

```
Carpeta raíz
|- /AnálisisDatosDocumentos/
|- /AnálisisDeSentimiento/
|- /AnálisisSeguimiento/
|- /BBDD/
|- /EstudioDeDatos/
|- /Redes/
|- /Twitter/
Iniciar.py
persistencia.py
```

Para acceder al inicio de la aplicación deberá ejecutarse el archivo en Python: "Iniciar.py", a continuación, se mostrará la ventana de Gestión de la Información de Twitter.

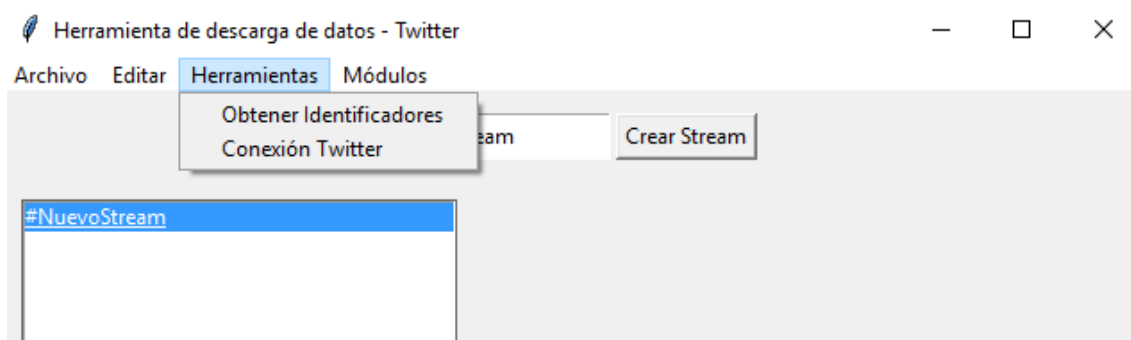
Gestión de la información de Twitter

Las principales herramientas que se facilitan en esta interfaz son las siguientes

Configuración de la conexión

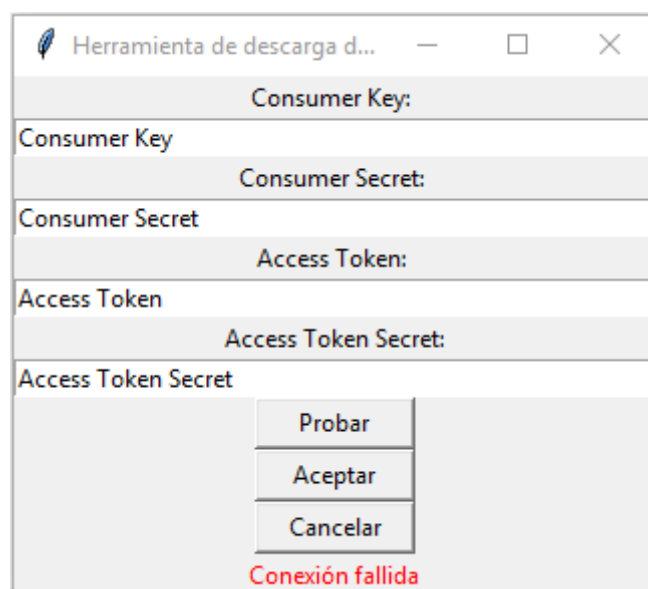
EL usuario podrá configurar las credenciales de conexión a Twitter. Para ello deberá realizar la solicitud previa en su página (<https://dev.twitter.com>) donde se debe acceder con una cuenta registrada.

Para el acceso a la ventana de credenciales deberemos desplegar el menú de “Herramientas” y posteriormente utilizar la opción “Conexión Twitter”. Como se indica en la siguiente imagen



Pantallazo 1 Despliegue del menú Herramientas

A continuación, se mostrará una ventana donde se ubica un formulario con las credenciales correspondientes: Consumer Key, Consumer Secret, Access Token y Access Token Secret, como se muestra en la imagen

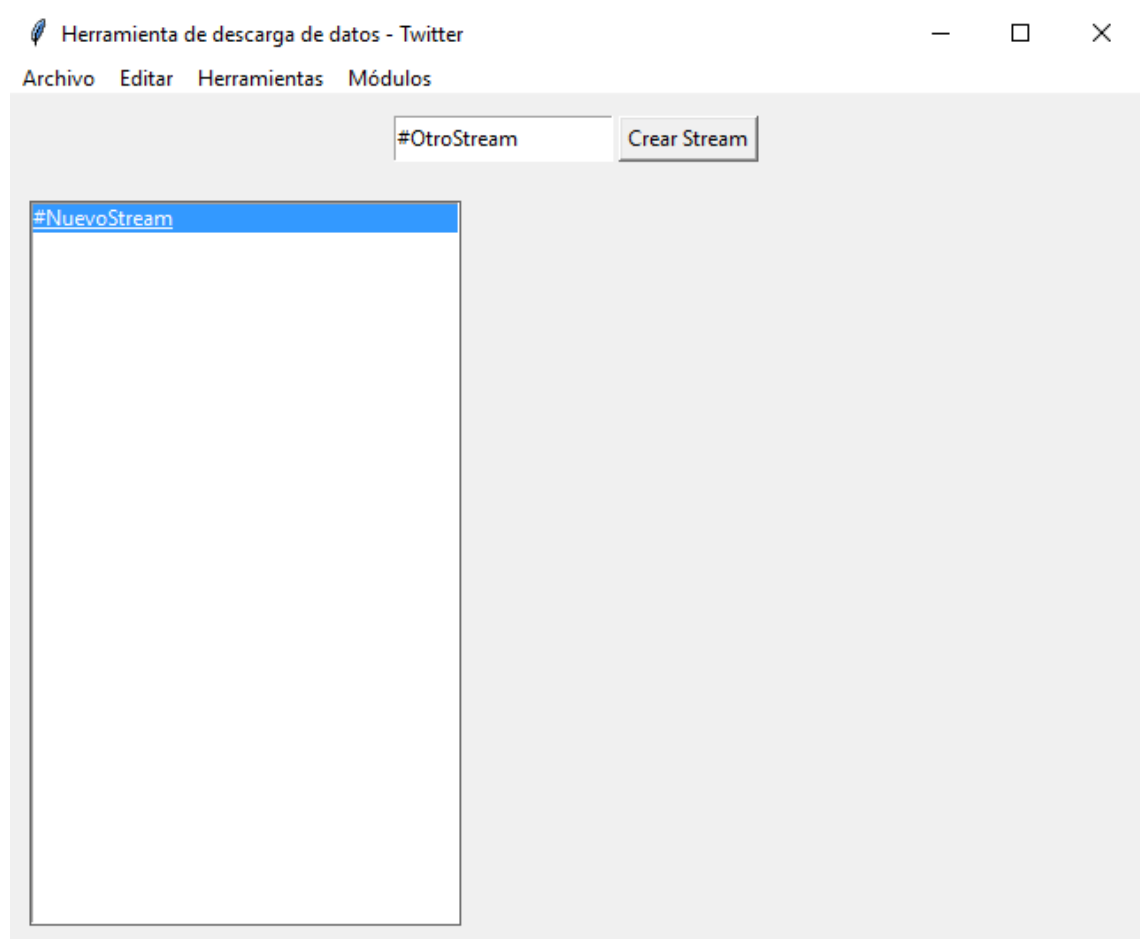
A screenshot of a configuration window titled "Herramienta de descarga d...". The window contains a form with four labeled input fields: "Consumer Key:", "Consumer Secret:", "Access Token:", and "Access Token Secret:". Below these fields are three buttons: "Probar", "Aceptar", and "Cancelar". At the bottom of the window, there is a red text message that says "Conexión fallida".

Pantallazo 2 Ventana de configuración de la conexión a Twitter

Después de insertar correctamente los datos pertinentes, podemos establecer una conexión de prueba mediante el botón “Probar”. Si las credenciales son correctas y se dispone de una conexión a la internet se recibirá el mensaje “Conexión Correcta” ,en caso contrario se mostrará un mensaje de “Conexión fallida” (como se muestra en la imagen anterior)

Seguimiento de Temas

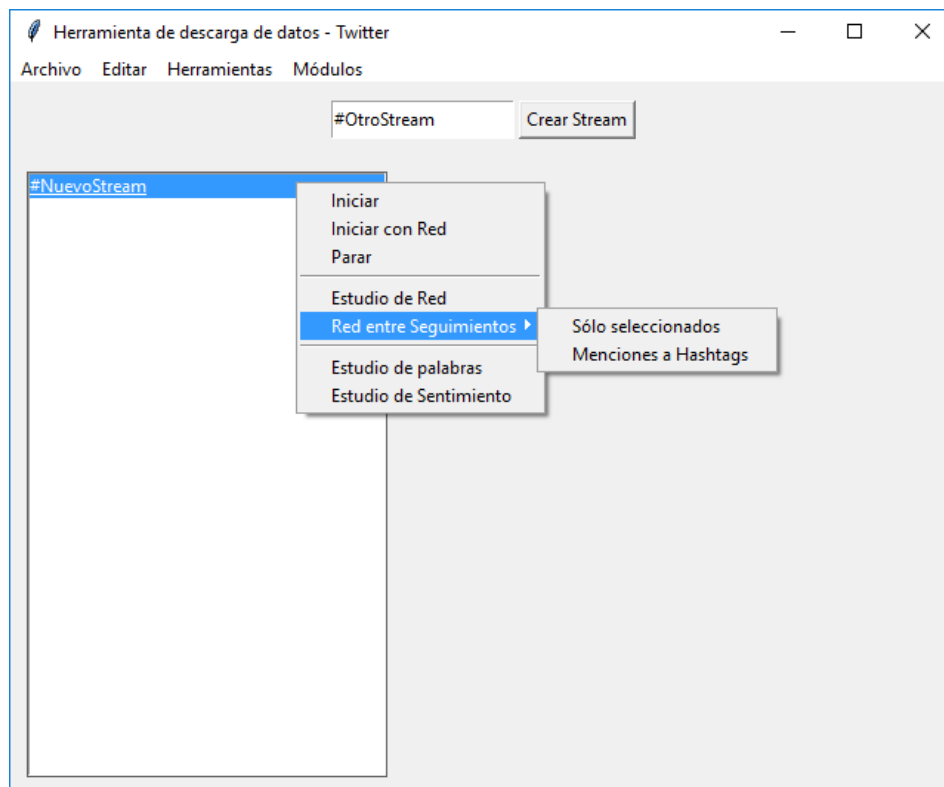
Para ejecutar un seguimiento de un tema deberemos elegir un término que se usará como filtrado en la búsqueda. Usualmente se utiliza un Hashtag. Éste debe ser especificado en el campo donde vemos “#OtroStream” y presionar el botón “Crear Stream”.



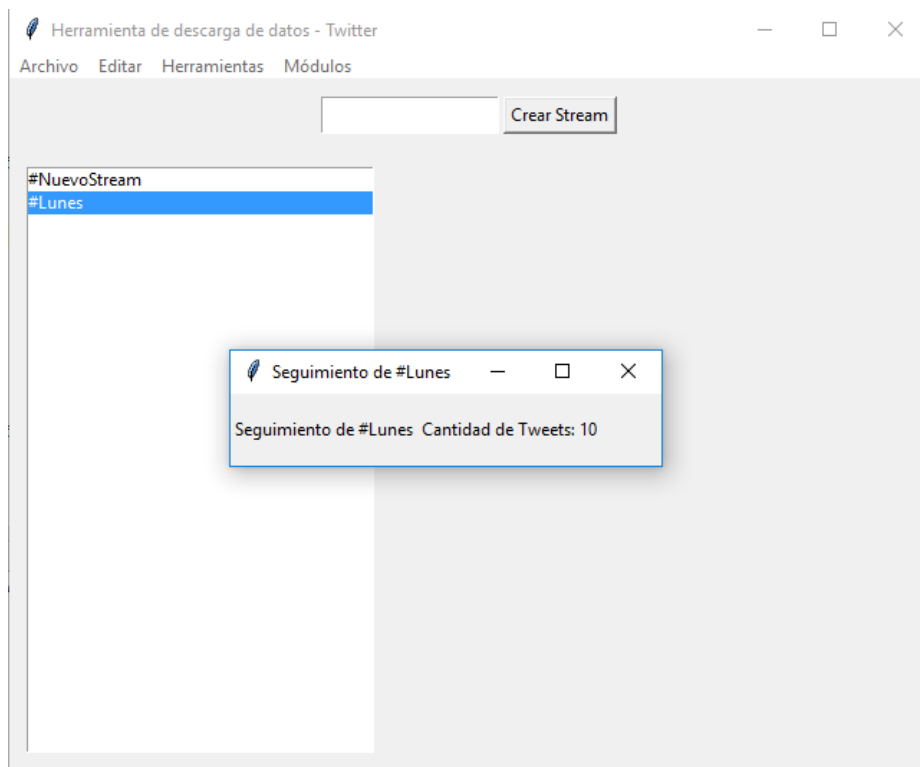
Pantallazo 3 Pantalla principal de la interfaz de gestión de datos de Twitter

Al realizar click derecho sobre una selección de la lista de seguimientos se desplegará un menú de herramientas (Pantallazo 4 Menú de herramientas sobre seguimiento). En él podremos encontrar los principales métodos que nos ayudarán a controlar la descarga de datos desde Twitter. Para comenzar el guardado de tuits sobre un identificador, deberemos presionar

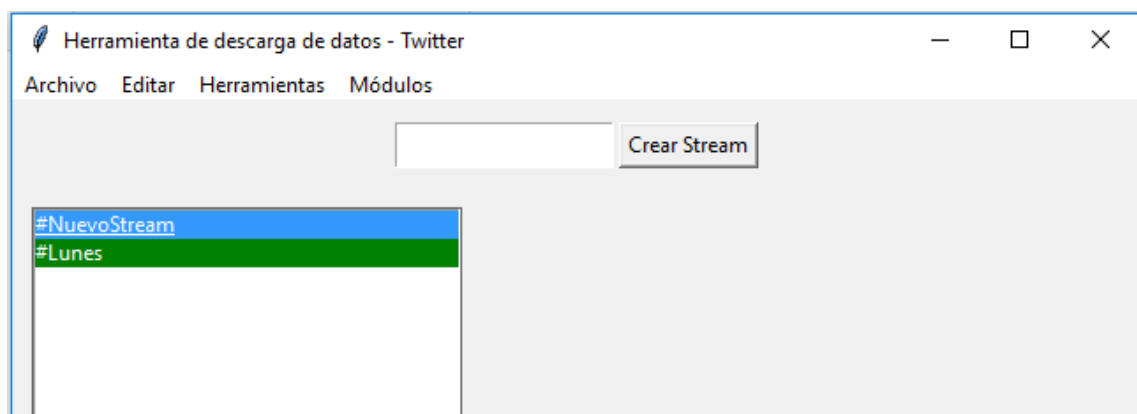
“Iniciar” o “Iniciar con Red”. Éste último se encarga de representar gráficamente las conversaciones que surgen en directo. Si elegimos “Iniciar” se abrirá una nueva ventana donde obtendremos información sobre el tema sobre el que se realiza el seguimiento y el número de tuits que se han descargado hasta el momento (como podemos ver en el Pantallazo 5). Además, si volvemos a la ventana de interfaz principal o cerramos la ventana de información actual del seguimiento visualizaremos los que se encuentran activos pero en segundo plano mediante sus ítems en la lista resaltados con color verde. Esto se puede contemplar en el Pantallazo 6.



Pantallazo 4 Menú de herramientas sobre seguimiento



Pantallazo 5 Visualización del estado actual del seguimiento



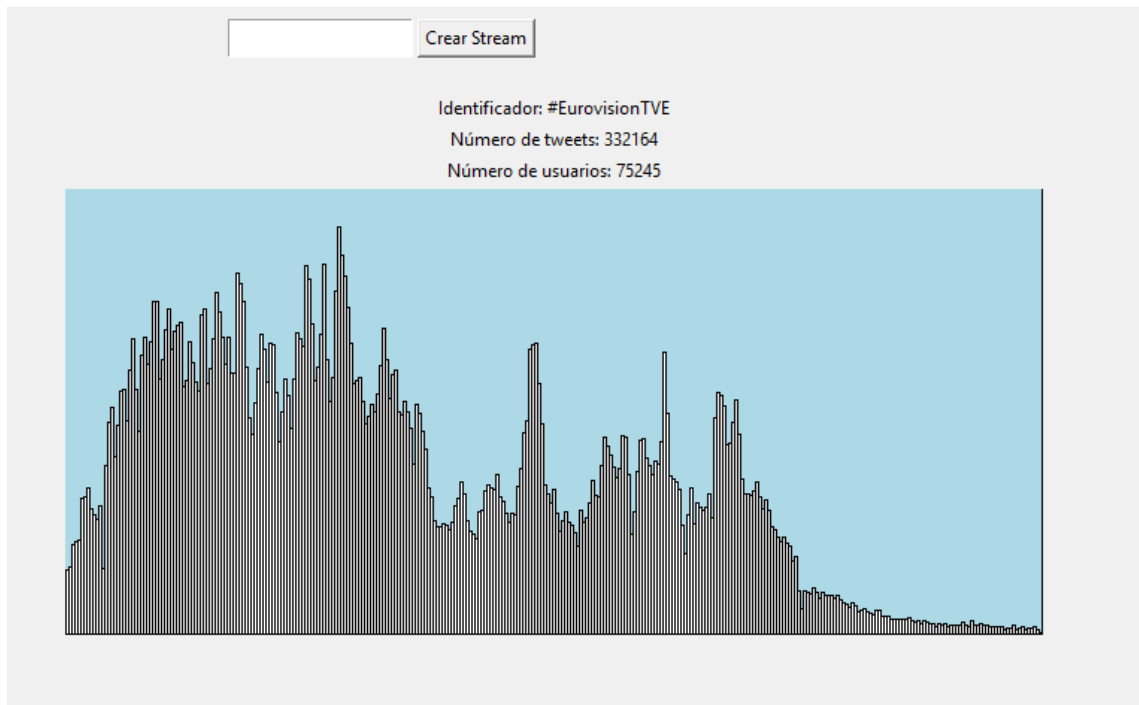
Pantallazo 6 Seguimiento realizado en segundo plano

Una vez hayamos obtenido algunos tuits podremos comenzar a obtener alguna información de éstos. Por ejemplo, la visualización de palabras más utilizadas y de hashtags mencionados. A ésta se accede mediante el menú desplegado mediante el click derecho con el ítem de seguimiento y el comando “Estudio de palabras”. La ventana con los datos se divide en dos columnas donde se puede realizar scroll. A la izquierda las palabras más utilizadas, de mayor a menor aparición, junto con la cantidad de veces registrada y a la derecha los hashtags mencionados en todos los tuits.

Herramienta de descarga de datos - Twitter	
que - 1667	#koparenbiladbb
de - 1615	#nvdemscaucus
el - 1275	#gottale...
a - 1219	#2
y - 903	#badalona
la - 875	#soto
en - 763	#gottalent2,
con - 649	#l6neconopactos
no - 633	#vote5h
me - 553	#frasesliberales
un - 499	#...
es - 468	#gotfrikki
lo - 460	#tusiquevales
se - 422	#tt
los - 352	#cepostaperte
las - 335	#l6nturnopactos
al - 300	#youtube
una - 294	#españa!!!
si - 285	#goldenbuzzer
por - 285	#go...
este - 277	#gritoporlalibertad...
programa - 258	#reto50machista
le - 218	#nivelazo
para - 206	#lista
del - 206	#reto150porrosa
ha - 204	#gottalentespana
como - 185	#gottalent2?
esto - 172	#oscars2016
más - 147	#reto50mac...
talento - 145	#l6nstanconruntos

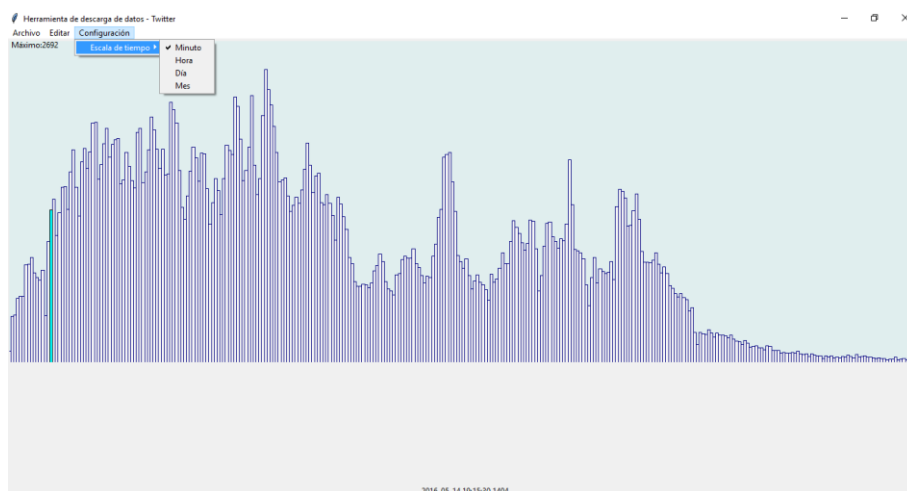
Pantallazo 7 Frecuencia de palabras y aparición de Hashtags en los mensajes

Además podemos obtener una visualización de la evolución temporal de los tuits, así como el número total de ellos y el número de usuarios que los han realizado. Para acceder a esta información se deberá realizar doble click en el listado de seguimientos, sobre el que queramos obtener los datos. Si queremos interactuar con la evolución temporal de nuevo interactuaremos con ella con doble click, esto nos abrirá una nueva ventana.



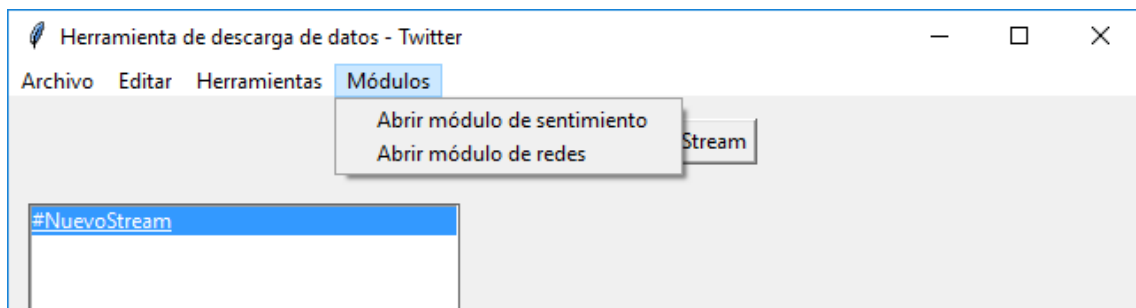
Pantallazo 8 Visualización de datos básicos: evolución temporal, tuits recogidos y usuarios que intervinieron

En la evolución temporal en detalle de los tuits podemos visualizar una gráfica de barras donde se especifica el número de tuits realizados en un intervalo de tiempo fijado. Si desplegamos el menú de “Configuración”, podremos seleccionar éste entre las opciones: “Minuto”, “Hora”, “Día” o “Mes”. En la parte inferior de la ventana obtendremos la fecha y cantidad de tuits realizados en el intervalo seleccionado ,pasando el mouse por encima, en la gráfica. Además se especifica el máximo de tuits realizados por intervalo de tiempo en la parte superior izquierda de la gráfica de barras.



Pantallazo 9 Evolución temporal en detalle

Una vez hemos utilizado las herramientas básicas de obtención de datos desde Twitter podremos acceder al resto de módulos mediante el despliegue del menú “Módulos”. Podemos acceder a la interfaz de Sentimiento mediante “Abrir módulo de sentimiento” y el de redes mediante “Abrir módulo de redes”. Éstos nos abrirán las nuevas ventanas correspondientes.



Pantallazo 10 Acceso al resto de módulos

Clasificación de Sentimiento

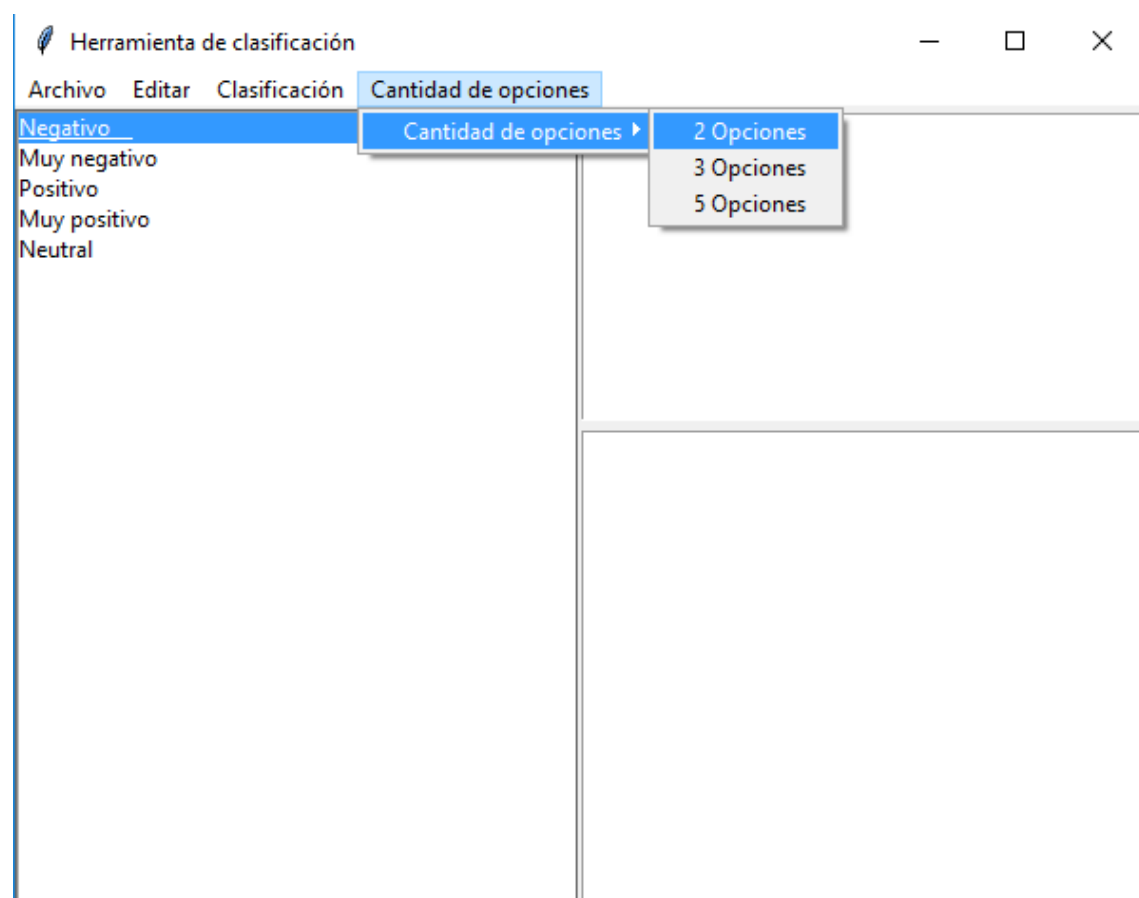
En este apartado describiremos el acceso y uso de las herramientas de clasificación de Sentimiento

Interfaz de clasificación manual

La interfaz de clasificación de sentimiento se divide en 2 partes principales. La columna izquierda muestra los textos que van a ser clasificados y la parte izquierda muestra los que ya lo están, de forma que se ubican en cada uno de los listados correspondientes, éstos serán indicados por el color que resalta los ítems clasificados en el tipo. Podemos definir el número de opciones a determinar en la clasificación de sentimiento:

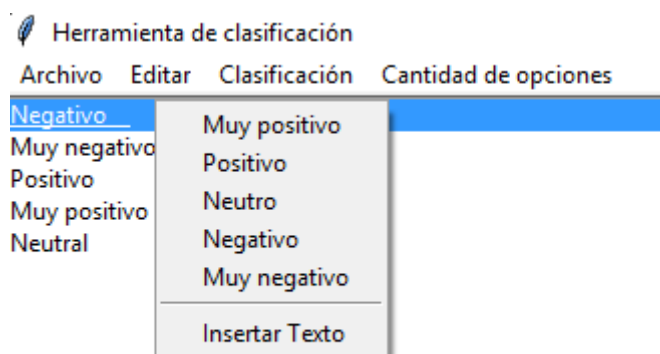
- 2 opciones: positivo y negativo
- 3 opciones: positivo, negativo y neutral
- 5 opciones: muy positivo, positivo, neutral, negativo y muy negativo

La gama de colores seleccionada para los textos positivos es el verde, negativos rojo y neutrales gris. Siendo los “Muy” los que mayor fuerza en el color tendrán.



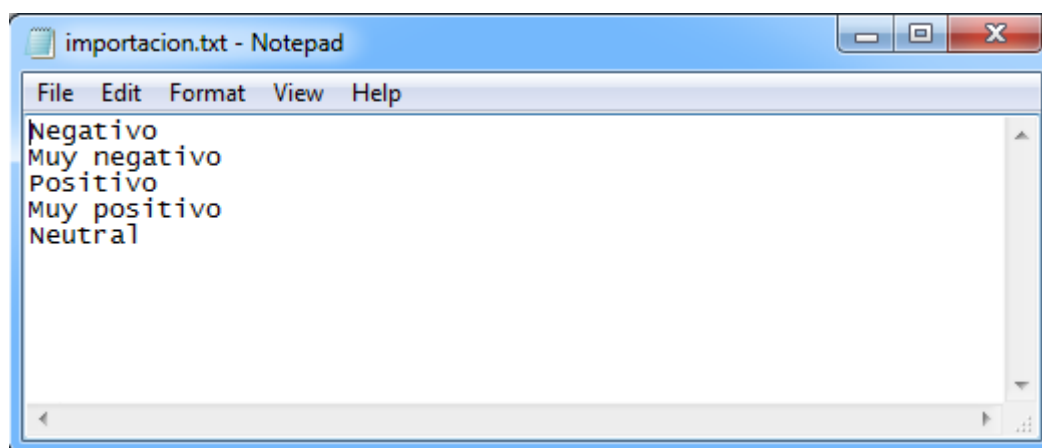
Pantallazo 11 Despliegue del menú para la selección de opciones

Podemos interactuar con el listado de documentos a clasificar mediante el click derecho después de la selección de alguno. Podemos realizar una multiselección arrastrando el clicado o mediante los atajos de teclado junto con clicado como *Shift* o *Ctrl*. Se desplegará un menú que nos dará la opción de insertar un nuevo texto, mediante una ventana donde insertarlo, o la clasificación de los ítems seleccionados. Si elegimos un sentimiento, la selección será movida al listado correspondiente. Además sólo se muestran las opciones correspondientes a la cantidad de opciones elegidas. En el caso del Pantallazo 12, se ha elegido la clasificación mediante 5 opciones y se muestra las diferentes opciones de clasificación.



Pantallazo 12 Despliegue de menú sobre los ítems a clasificar

Para la inserción de un nuevo texto podemos utilizar el método descrito anteriormente o realizar la importación de un fichero de texto plano donde cada línea será un documento a importar. Por ejemplo, el documento reflejado en la siguiente imagen



Pantallazo 13 Ejemplo de fichero de texto a importar

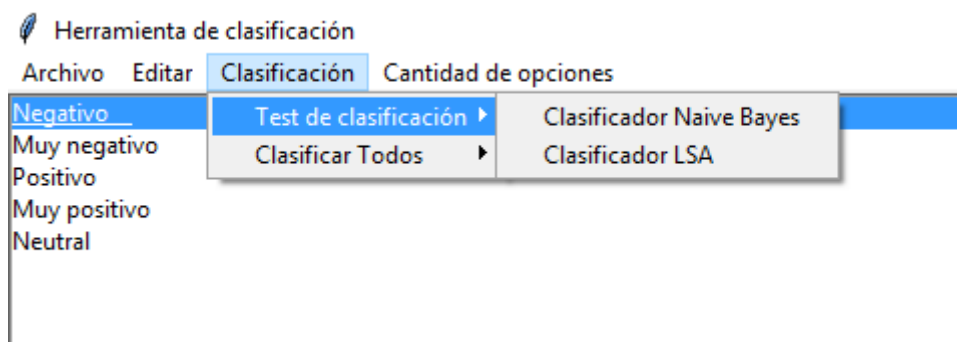
Otra de las formas de importar textos o clasificaciones ya realizadas y guardadas en archivos son mediante la apertura de éstos siguiendo los pasos "Archivo" -> "Abrir" y seleccionar la ruta y fichero de datos.

Una vez realizada la clasificación manual, podremos utilizarla para la clasificación automática de nuevos documentos. Esto se puede realizar mediante esta interfaz directamente introduciéndolos o sobre los tuits de un seguimiento guardando la clasificación actual mediante

“Archivo”->“Guardar” y la selección de la ruta y nombre de archivo. Éstos métodos los explicamos a continuación.

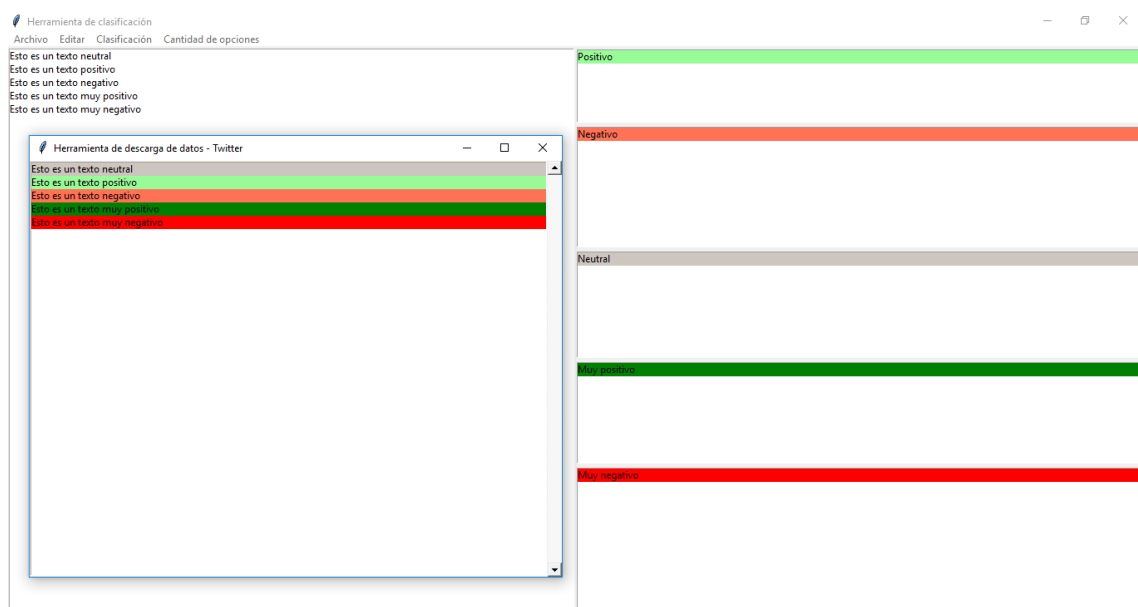
Clasificación automática

Si desplegamos el menú “Clasificación” podremos ejecutar la realización de Test de clasificación o directamente la clasificación de textos introducidos sin sentimiento. En ambos casos se deberá seleccionar el clasificador requerido: Naive-Bayes o LSA. En el caso de los test se abrirá una nueva ventana con la información porcentual de aciertos sobre la inferencia de documentos ya clasificados que se clasifican como si no lo estuvieran.



Pantallazo 14 Despliegue de menú para el tipo de clasificación

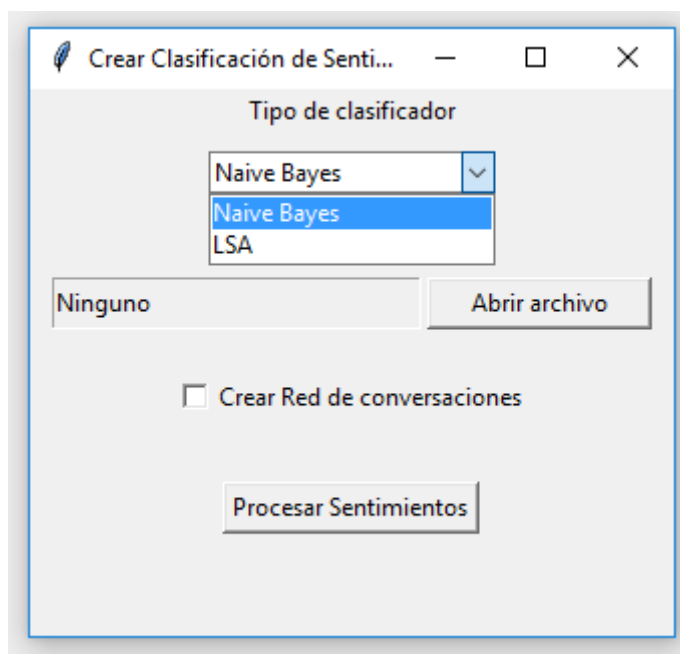
Si seleccionamos “Clasificar Todos” y el clasificador requerido obtendremos una visualización en una nueva ventana los textos a clasificar con su color correspondiente que los identifica ,explicado anteriormente.



Pantallazo 15 Ejecución y muestra del resultado de una clasificación de 5 opciones

Clasificación de sentimiento sobre un seguimiento

Partiendo de la interfaz principal, de gestión de datos de Twitter, deberemos acceder al despliegue del menú sobre el ítem que queramos en la lista de seguimientos. A continuación, deberemos utilizar el comando “Estudio de Sentimiento”. Se mostrará una nueva ventana donde se pedirá la información necesaria para realizar la clasificación de los tuits de un seguimiento. Ésta es: clasificador a utilizar, nuevamente (en la versión actual) Naive Bayes o LSA, el archivo de ejemplos de clasificación que hemos guardado en la interfaz de clasificación, como se describió anteriormente, y la opción de creación de red con sentimientos que describiremos en la sección dedicada a la interfaz de redes. Cuando se haya configurado correctamente se deberá presionar “Procesar Sentimiento”, lo que mostrará una nueva ventana donde se muestran los textos clasificados por colores.



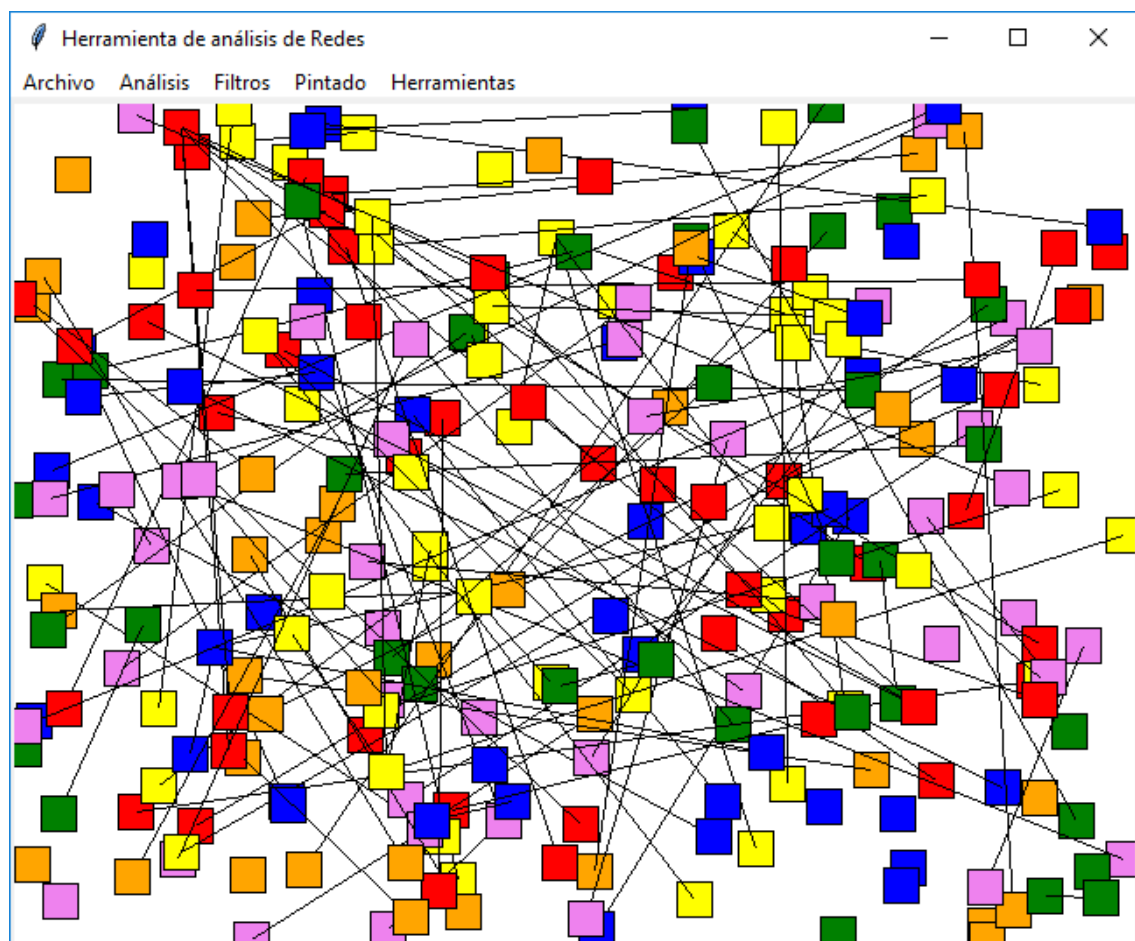
Pantallazo 16 Creación de un estudio de sentimiento

Redes

En esta sección revisaremos la funcionalidad y uso de la interfaz dedicada al manejo de redes creadas a partir de datos descargados mediante la interfaz de gestión de datos de Twitter.

Creación de la red

Desplegando el menú sobre el ítem de seguimiento seleccionado en la interfaz de gestión de datos de Twitter deberemos seleccionar la opción “Estudio de Red”, la cual nos creará una red de conversaciones creadas entre tuits y se mostrará en una nueva ventana. Los usuarios serán representados mediante cuadrados donde el color de cada uno será el mismo que el de los nodos a los que esté relacionados. Sin embargo, se establece un máximo número de colores a utilizar, por lo que no todos los nodos del mismo color están conectados.

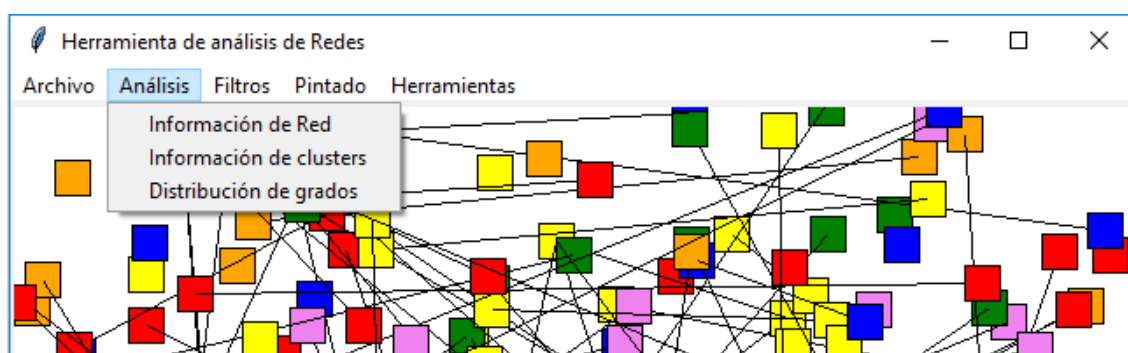


Pantallazo 17 Visualización de red creada

Sobre esta interfaz podremos realizar los mismos métodos de administración de archivos que en la de sentimiento. Podemos guardar la red mediante “Archivo”->”Guardar” y selección de la ruta y nombre del fichero, abrir una red guardada mediante “Archivo”->”Abrir” y selección del fichero , e incluso (en la versión actual) realizar una exportación a CSV que puede ser utilizada en programas externos como Gephi para el análisis de redes.

Herramientas de análisis de la red

Dentro de las herramientas que nos permiten analizar ciertos aspectos de la red tenemos: información básica, información de clústeres e información sobre la distribución de los grados. Para acceder a cada una de ellas se deberá desplegar el menú “Análisis”

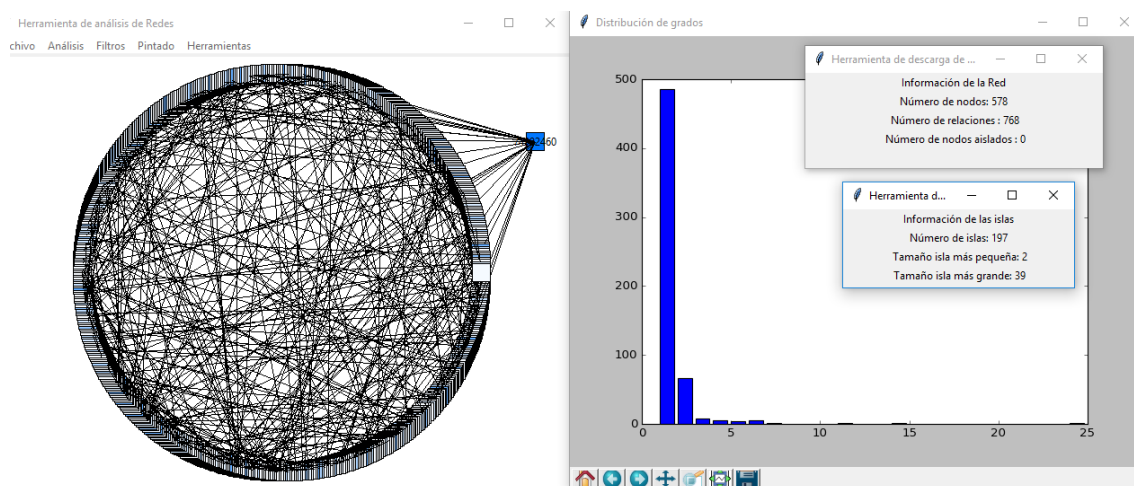


Pantallazo 18 Despliegue del menú de análisis

La selección de información básica mediante “Información de Red” nos mostrará una nueva ventana con el número de nodos, número de relaciones y número de nodos aislados (sin relación con otros diferentes) que componen la red.

La información sobre clústeres mediante “Información de clusters” nos ofrece el número de islas que componen la red, partes conexas en las que se puede dividir, el tamaño (en número de nodos) de la mayor de ellas y la menor de ellas. Si una red sólo tiene una isla nos encontramos ante una red conexas.

Por último podemos obtener información acerca de los grados y su distribución mediante “Distribución de grados” que nos muestra una nueva ventana donde se representa gráficamente el grado frente al número de nodos que lo tienen.



Pantallazo 19 Visualización de la información de la red, distribución de grados e información de clusters.

Configuraciones visuales

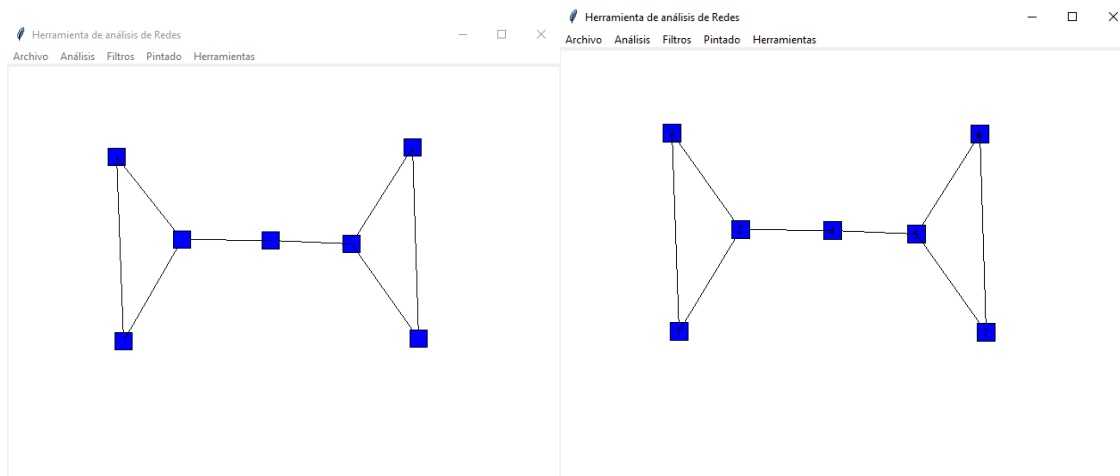
Podemos realizar configuraciones básicas sobre la representación de la red mediante el uso de filtros que nos permiten ocultar ciertas relaciones o eliminar ciertos nodos.

En la versión actual de la aplicación, podemos realizar una eliminación de los nodos que no se relacionan con un nodo distinto en la red mediante el despliegue del menú “Filtros” y la selección de “Nodos con ellos mismos” en “Restricciones de Nodos”. Además podemos realizar una ocultación/visualización de relaciones según su sentimiento: Neutral o Normal, Positivo, Muy positivo, Negativo o Muy negativo.



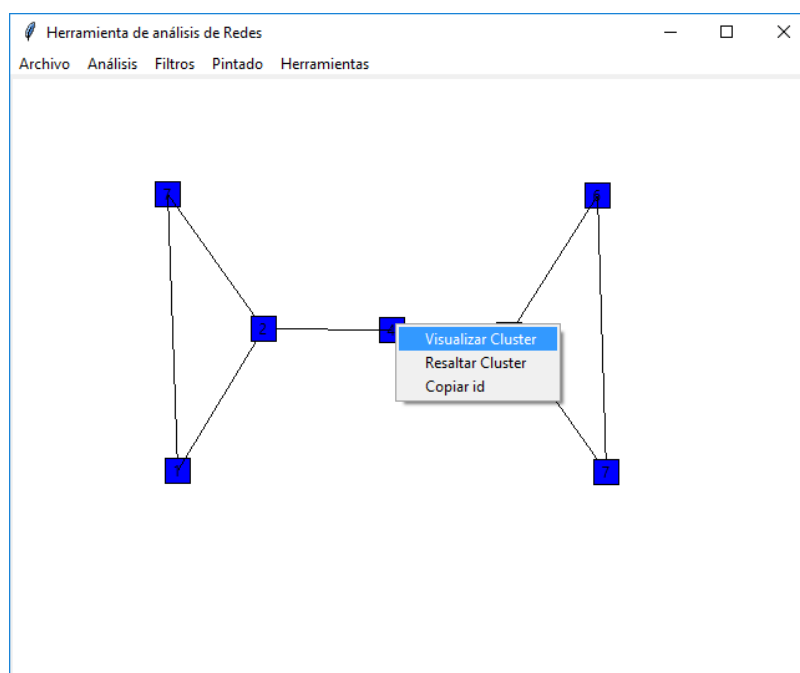
Pantallazo 20 Despliegue del menú de filtros

Podemos visualizar los identificadores de cada nodo dentro de la red para facilitar cualquier tarea de identificación. Únicamente se deberá realizar click con el botón central del mouse sobre el nodo a mostrar.

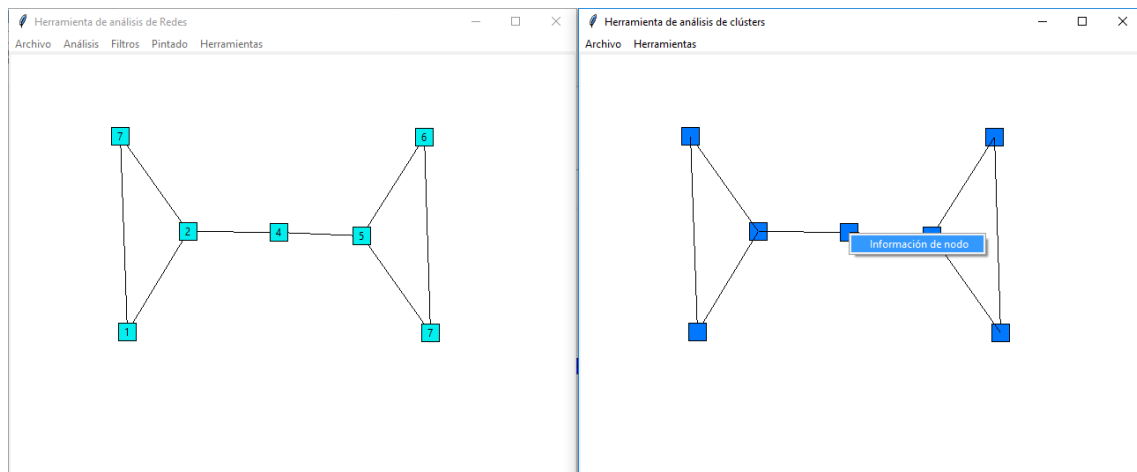


Pantallazo 21 Visualización del identificador de cada nodo

Además podemos interactuar con cada nodo desplegando un menú mediante el click derecho sobre el nodo que queramos. Tendremos varias opciones: “Resaltar clúster” que colorea diferente el grupo al que pertenece el nodo seleccionado, “Visualizar clúster” que se encarga de representar el clúster en una nueva ventana para una interacción más detallada, y por último “Copiar id” que se encarga de copiar al portapapeles el identificador del nodo seleccionado para posterior uso.



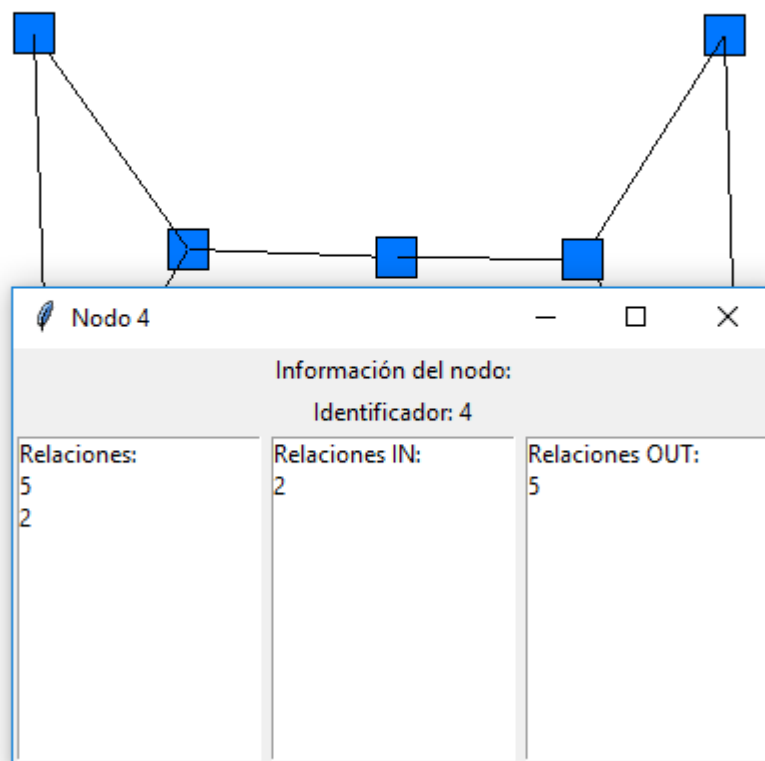
Pantallazo 22 Despliegue del menú sobre nodo de red



Pantallazo 23 Visualización del clúster del nodo seleccionado

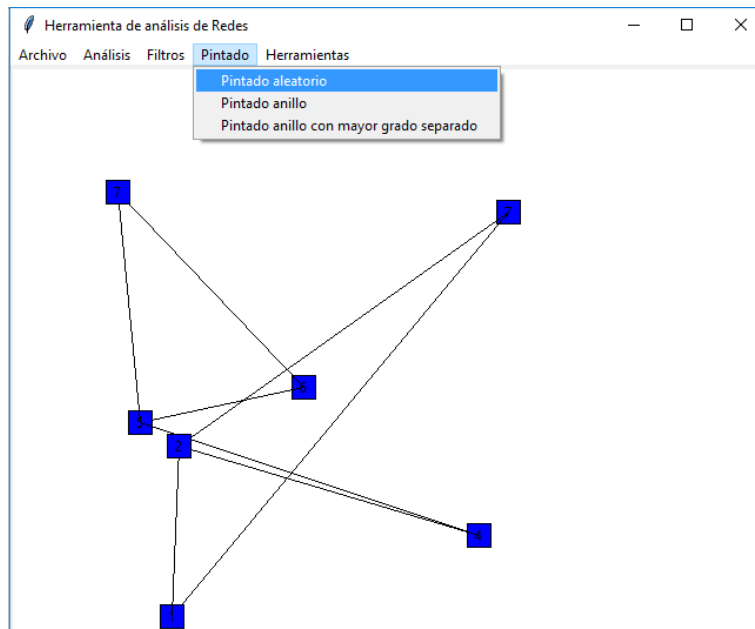
Sobre la ventana que se muestra donde se representa el grupo del nodo seleccionado podemos obtener información detallada sobre las relaciones. A ésta se accede nuevamente mediante el click derecho sobre el nodo a estudiar y la selección de la opción “Información de nodo” en el menú flotante.

En la ventana que se nos muestra sobre la información del nodo podremos observar su identificador, las relaciones totales de entrada y de salida, y las relaciones divididas según tipo.

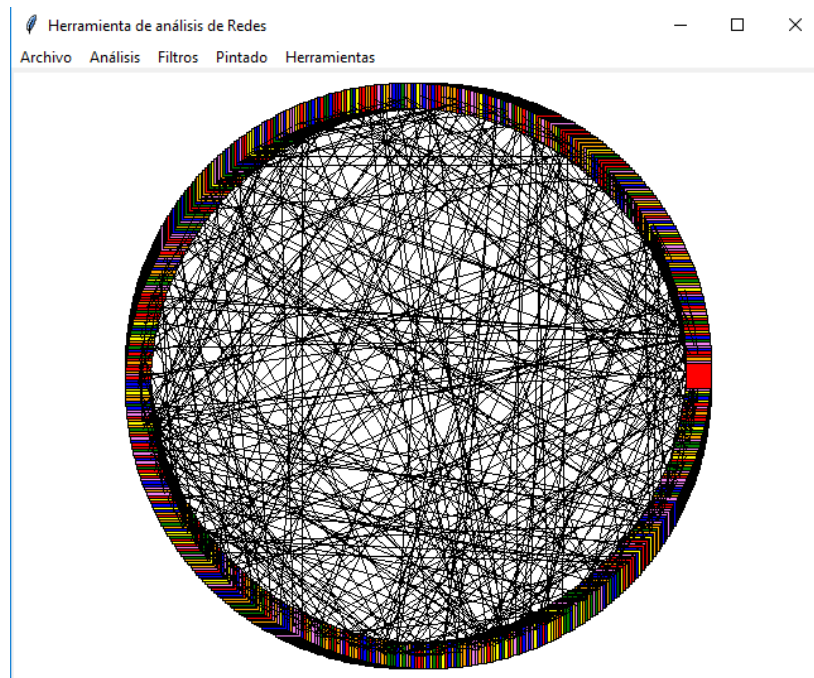


En la interfaz de red, también tenemos opciones de representación de la red según la posición de los nodos en el plano. Mediante el despliegue del menú “Pintado” podemos seleccionar las distintas formas:

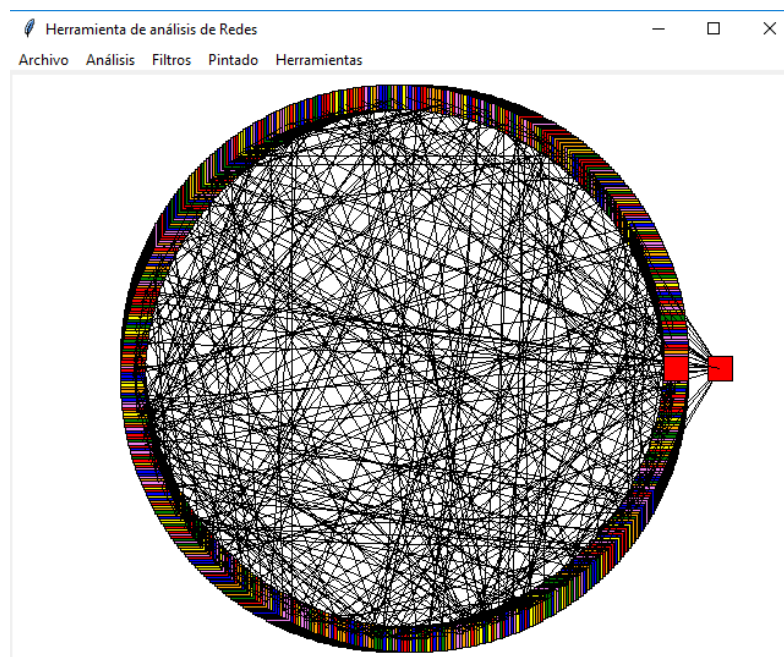
- Aleatoria. Estableciendo un punto aleatorio en el plano para cada nodo. Pantallazo 24.
- Anillo. En forma circular donde las relaciones quedan en el interior. Pantallazo 25.
- Anillo con mayor grado separado. De forma similar a la anterior se disponen los nodos en forma circular y se extrae el que mayor grado tenga. Pantallazo 26.



Pantallazo 25 Despliegue del menú de Pintado de la red



Pantallazo 26 Visualización de la representación en forma de anillo

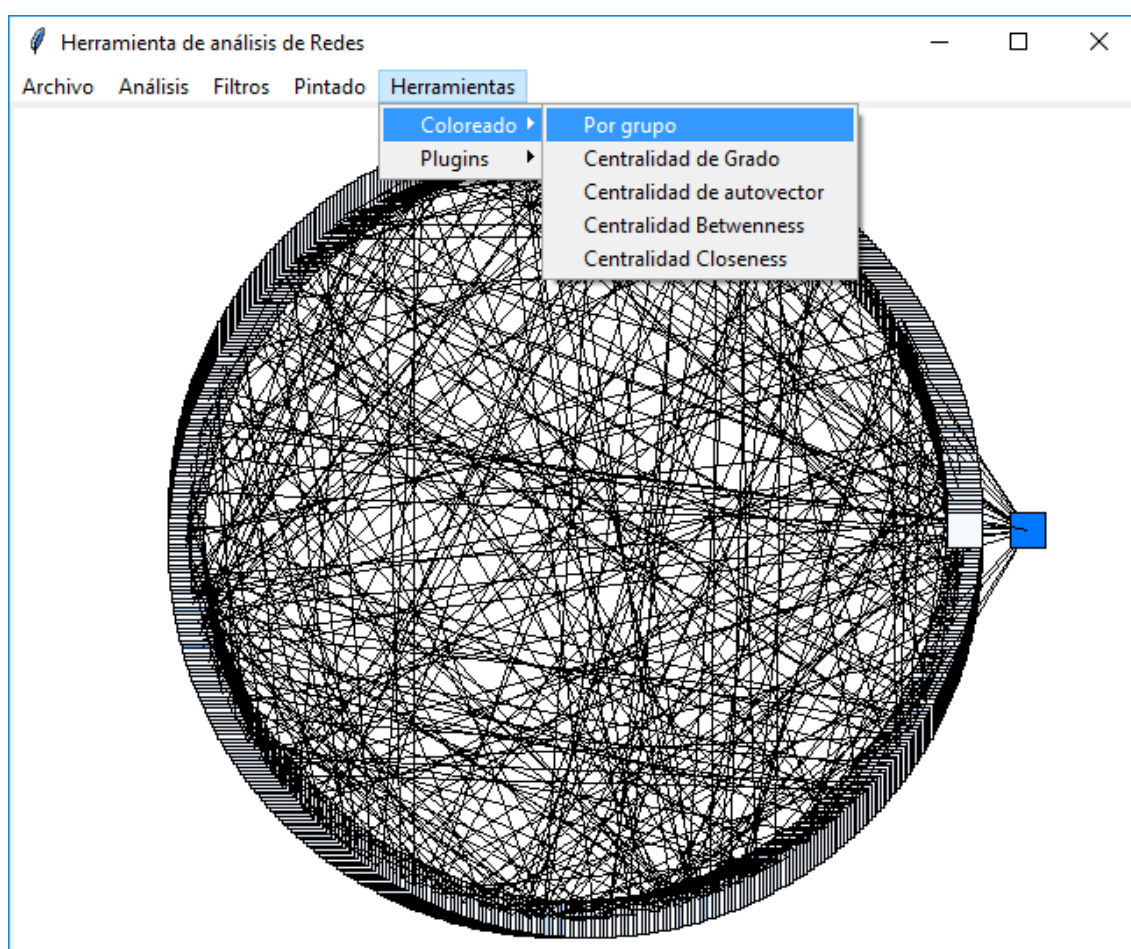


Pantallazo 27 Visualización de la representación en forma de anillo con el nodo de mayor grado separado

Representación de centralidad

Podemos seleccionar en la interfaz de red alguno de los métodos de coloreado presentados en el despliegue del menú “Herramientas”->”Coloreado”. Los principales son los siguientes

- Por grupo. Se utiliza el mismo color para pintar nodos que se encuentran en el mismo clúster.
- Centralidad. Se representa mediante gama de azul, cuanto más fuerte el tono de azul mayor es la centralidad en comparación con la de los demás nodos. En la versión actual se encuentran disponibles las centralidades por: grado, autovector, betweenness y closeness.

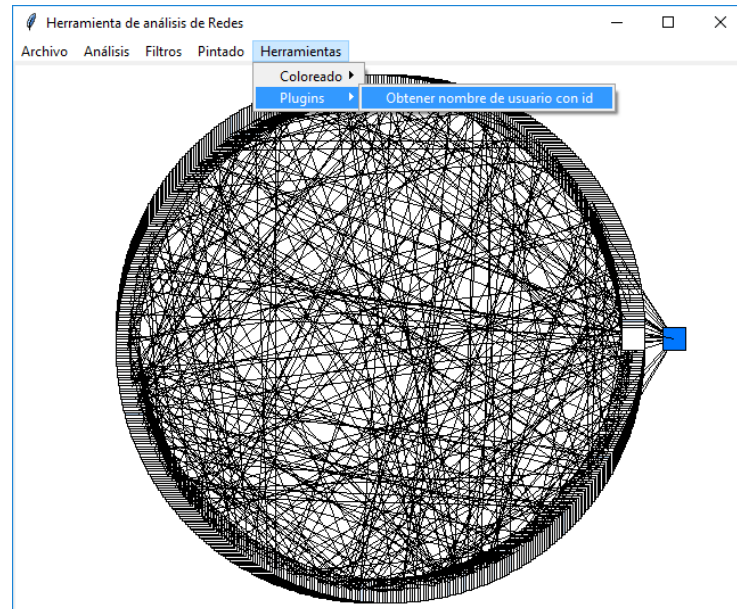


Pantallazo 28 Despliegue del menú de herramientas

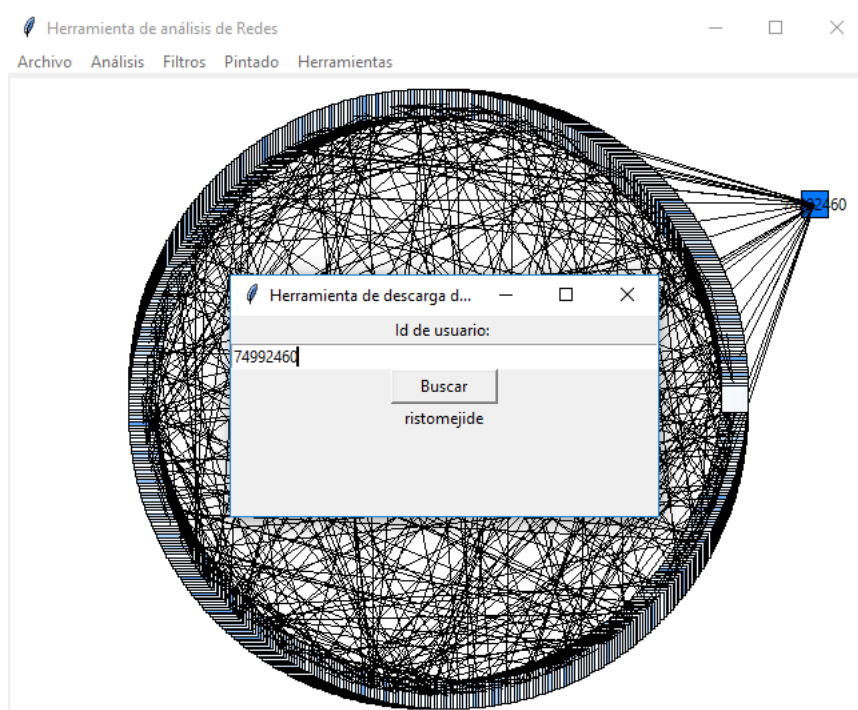
Plugins y herramientas externas

En la actual versión de la aplicación se encuentra disponible la opción de consulta de nombre de usuario a partir de un identificador de Twitter. Útil en combinación con la representación de centralidad por grado. Para esto deberemos desplegar el menú

“Herramientas”-> “Plugins” y “Obtener nombre de usuario con id”. A continuación, se mostrará una ventana con un campo de entrada donde deberemos insertar el identificador del usuario y presionar “Buscar”. Su screen_name aparecerá en la parte inferior.



Pantallazo 29 Despliegue del menú de plugins

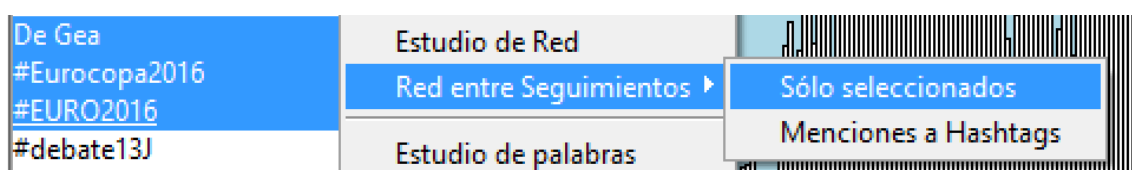


Pantallazo 30 Visualización de obtención de screen_name de usuario de Twitter a partir de id

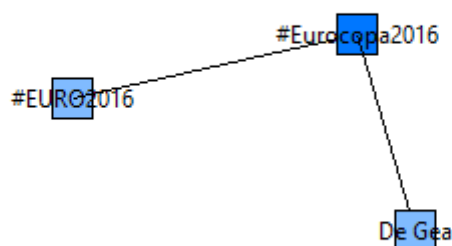
Redes entre Hashtags y seguimientos

En la interfaz principal de gestión de datos de Twitter podemos hacer uso de la agrupación de seguimientos para la creación de una red de relación entre ellos. Esto se encargará de crear relaciones si encontramos tuits que nombre dos términos diferentes.

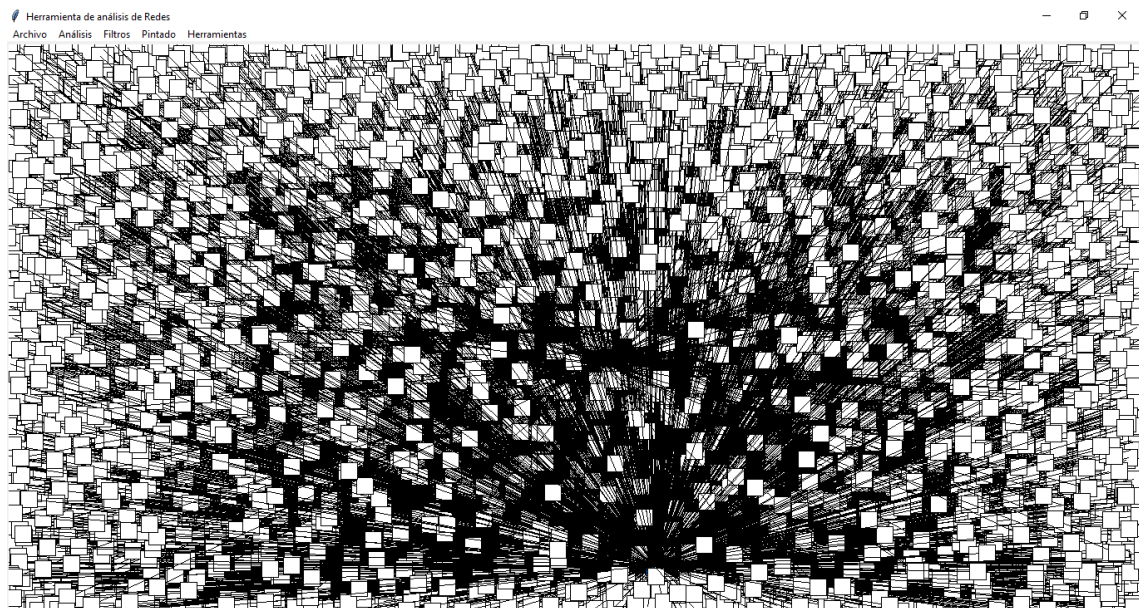
Para ello, debemos desplegar el menú mediante el click derecho sobre el ítem o ítems de seguimiento a estudiar (Pantallazo 29). Dentro del submenú “Red entre seguimientos” encontramos “Sólo seleccionados” con el que se visualizará la red descrita antes, y “Menciones a Hashtags” el cual creará una red donde cada nodo será un hashtag y una relación si se utiliza en un tuit del seguimiento o seguimientos seleccionados. Ejemplo de la primera red la encontramos en el Pantallazo 30 y de la segunda en el Pantallazo 31.



Pantallazo 31 Despliegue del menú de creación de red entre seguimientos



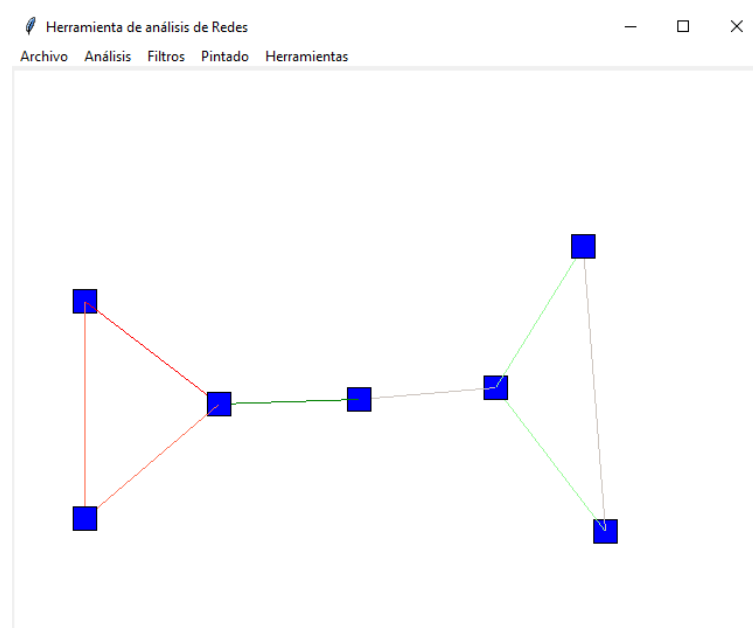
Pantallazo 32 Visualización de red entre hashtags



Pantallazo 33 Visualización de red creada por los hashtags mencionados en un seguimiento

Creación de redes de sentimiento

En la interfaz principal de la gestión de datos de Twitter permite la creación de redes mediante los sentimientos obtenidos de la clasificación de los tuits. Para esto se procede como en el apartado de Clasificación de Sentimiento Sobre un Seguimiento. Sin embargo, en este caso se debe seleccionar la opción “Crear red de conversaciones”. A continuación, se muestran dos ventanas: la que se corresponde con la salida de la clasificación de tuits ,como en el caso normal de clasificación de sentimiento, y una ventana correspondiente a la red creada representando los distintos sentimientos entre usuarios mediante los colores especificados en el apartado de Interfaz de Clasificación de Sentimiento.



Pantallazo 34 Visualización de red con sentimientos

Guión para la realización de pruebas

El usuario deberá leer atentamente las distintas tareas a realizar. Si tiene cualquier duda, puede realizar cualquier consulta al encargado que se encuentre vigilando la sesión. Cuando las finalice, proceda con el cuestionario final. En total se deberán ejecutar 20 ejercicios y 11 cuestiones. El inicio de cada sección le será indicado por el ayudante en las pruebas.

Importante: La elaboración de las pruebas es voluntaria, por lo que no será recompensado por su realización.

Tareas

- Tarea 1.1: Configuración de los datos de acceso a la API de Twitter. Además, se deberá probar la conexión. La finalización de esta tarea se efectuará al obtener una conexión correcta. Las claves a introducir son las siguientes

NOMBRE	CLAVE
CONSUMER KEY	MrOBuLfa1aJ4Yz7EpT4
CONSUMER SECRET	psSONPQbnkRf331GWORgEFASc1u8zZ3jYOKN
ACCESS TOKEN	253145620-W7P74YIOATn81s9w9SSBz7VD3GE
ACCESS TOKEN SECRET	qryUeBgU43YJ2etFqEA1QcpReZgg5Cv

- Tarea 1.2: Realizar un seguimiento de un Hashtag elegido por el usuario hasta haber almacenado más de 5 tuits. Esto incluye iniciarlo, observar la información de número de tuits almacenados y la pausa/finalización.
- Tarea 1.3: Salir de la aplicación para volver a abrirla. Obtener el seguimiento realizado anteriormente e iniciar de nuevo. La tarea estará acabada cuando se pause el seguimiento.
- Tarea 1.4: Obtener información acerca de los datos guardados: número de tuits y número de usuarios.
- Tarea 1.5: Visualizar la evolución temporal y obtener el máximo de tuits por minuto realizados.
- Tarea 1.6: Definir cuál ha sido el término más utilizado así como otros hashtags mencionados.
- Tarea 2.1: Agregar la frase de forma manual: "Esta frase me gusta", y clasificarla como positiva o negativa.
- Tarea 2.2: Eliminar la frase del sentimiento asignado y clasificarla en el opuesto. Es decir, si se clasificó como negativa se deberá mover a positiva y viceversa.
- Tarea 2.3: Introducir una nueva frase: "Esta frase es especial" y clasificarla en el sentimiento con listado vacío, e introducir otra que se dejará tal cual: "Esto es una frase que gusta". Realizar un guardado a un archivo y cerrar la aplicación.
- Tarea 2.4: Abrir de nuevo la aplicación y el archivo recientemente guardado. Comprobar que las frases están en las mismas posiciones que en el paso anterior.

- Tarea 2.5: Incluir las siguientes frases en un documento de texto. Importarlo a la aplicación las frases e importarlas a la aplicación. Clasificarlas todas al mismo sentimiento. Frases
 - o El perro es bonito
 - o El hombre es guapo
 - o La mujer es bella
 - o El edificio esta bien construido
- Tarea 2.6: Realizar un test de clasificación y definir el resultado.
- Tarea 2.7: Realizar la clasificación automática de las frases restantes.
- Tarea 3.1: Realizar una clasificación de sentimiento con el clasificador elegido por el usuario sobre un seguimiento.
- Tarea 3.2: Iniciar un seguimiento que muestre la red que se va creando, con un hashtag elegido por el usuario y pausar pasados 5 segundos.
- Tarea 3.3: Crear una red a partir de un Hashtag y obtener el número de nodos y de relaciones.
- Tarea 3.4: Colorear mediante el grado de nodo. Obtener el identificador del nodo que mayor grado tiene y visualizar su clúster.
- Tarea 3.5: Visualizar la información estructural de la red: distribución de grados, tamaño de mayor isla.
- Tarea 3.6: Guardar como un archivo la red. Cerrar la aplicación y abrir el fichero creado. Comprobar que es la misma red.
- Tarea 3.7: Crear una red donde se tenga en cuenta el sentimiento de los tuits.

Cuestionario

- ¿Le ha parecido fácil e intuitiva la interfaz a la hora de realizar las distintas tareas en el apartado de obtención de datos desde Twitter?
- ¿Le ha parecido fácil e intuitiva la interfaz a la hora de realizar las distintas tareas en el apartado de análisis de sentimiento?
- ¿Le ha parecido fácil e intuitiva la interfaz a la hora de realizar las distintas tareas en el apartado de manejo de redes?
- ¿En algún momento se ha sentido inseguro o no tenía el control de la aplicación?
- ¿Cree que los menús, botones y cómo se llega a ellos se encuentran correctamente dispuestos? ¿Cuáles modificaría?
- ¿Le parece suficiente la ayuda aportada a lo largo de la aplicación?
- ¿Cuáles son las características positivas de la aplicación?
- ¿Cuáles son las características negativas de la aplicación?
- Valore del 1 al 5, donde 1 es muy mala y 5 es muy buena, las interfaces relacionadas con cada fase según la interacción en la realización las tareas.
- ¿Qué nuevas funcionalidades le gustaría poder cubrir?
- Otros comentarios.

Glosario

API – Application Programming Interface

CRUD – Create Read Update and Delete

CSS – Cascading Style Sheets

CSV – Comma Separated Value

DFS – Depth First Search

GML – Generalized Markup Language

GNU – GNU is not Unix

GUI – Graphic User Interface

HTML – HyperText Markup Language

JSON – JavaScript Object Notation

MVC – Model-View-Controller

RDBMS – Relational Data Base Management System

SQL – Structured Query Language

XML – Extensible Markup Language

Referencias

- [1] «Twitter» 2016. [En línea]. Available: <https://about.twitter.com/es/company>. [Último acceso: Agosto 2016].
- [2] E. Filadelfo, 22 Agosto 2016. [En línea]. Available: <https://blog.twitter.com/2016/the-río2016-twitter-data-recap> . [Último acceso: Agosto 2016].
- [3] «Twitter Analytics» 2016. [En línea]. Available: <https://business.twitter.com/es/analytics.html>. [Último acceso: Agosto 2016].
- [4] PearAnalytics, Agosto 2009. [En línea]. Available: <http://web.archive.org/web/20120503013715/http://www.pearanalytics.com/blog/wp-content/uploads/2010/05/Twitter-Study-August-2009.pdf>. [Último acceso: Agosto 2016].
- [5] A. Kumar, «Blog Twitter» 3 Noviembre 2015. [En línea]. Available: <https://blog.twitter.com/es/2015/corazones-en-twitter>. [Último acceso: Agosto 2016].
- [6] «Wikipedia,» 2016. [En línea]. Available: https://es.wikipedia.org/wiki/Trending_topic. [Último acceso: Agosto 2016].
- [7] J. Pertierra das Neves, «Análisis semántico y estructural de Twitter basado en redes complejas» Móstoles, 2016.
- [8] «JSON.org» 2016. [En línea]. Available: <http://www.json.org/json-es.html>. [Último acceso: Agosto 2016].
- [9] «Wikipedia» 2016. [En línea]. Available: <https://es.wikipedia.org/wiki/JSON>. [Último acceso: Agosto 2016].
- [10] «Wikipedia» 2016. [En línea]. Available: https://en.wikipedia.org/wiki/Comma-separated_values. [Último acceso: Agosto 2016].
- [11] «w3schools.com» 2016. [En línea]. Available: http://www.w3schools.com/xml/xml_what.asp. [Último acceso: Agosto 2016].
- [12] «Wikipedia» 2016. [En línea]. Available: https://es.wikipedia.org/wiki/Extensible_Markup_Language. [Último acceso: Agosto 2016].
- [13] R. Puente, «Blog Rodrigo Puente» 3 Diciembre 2014. [En línea]. Available: <http://blog.rodrigopuente.com/bases-de-datos-teorema-cap-cual-base-de-datos-debo-usar/>. [Último acceso: Agosto 2016].
- [14] «Wikipedia» 2016. [En línea]. Available: https://es.wikipedia.org/wiki/Teorema_CAP. [Último acceso: Agosto 2016].
- [15] «Wikipedia» 2016. [En línea]. Available: <https://es.wikipedia.org/wiki/MongoDB>. [Último acceso: Agosto 2016].

- [16] B. Scofield, «NoSQL. Death to Relational Databases(?)» 2010.
- [17] «Wikipedia» 2016. [En línea]. Available: <https://es.wikipedia.org/wiki/CRUD>. [Último acceso: Agosto 2016].
- [18] D. San Juan Lopez, «Aplicación Web para el Almacenamiento y Visualización de Geodatos Meteorológicos mediante Spring y MongoDB. Análisis de Técnicas de Indexación NoSQL» Universidad Extremadura, 2015.
- [19] Python.org, «Python.org» 2016. [En línea]. Available: <https://www.python.org/doc/essays/blurbl/>. [Último acceso: Agosto 2016].
- [20] «Python.org» 2016. [En línea]. Available: <https://www.python.org/about/>. [Último acceso: Agosto 2016].
- [21] «Tweepy.org» 2016. [En línea]. Available: <http://www.tweepy.org/>. [Último acceso: Agosto 2016].
- [22] «OAuth.net» 2016. [En línea]. Available: <https://oauth.net/>. [Último acceso: Agosto 2016].
- [23] «Twitter Development» 2016. [En línea]. Available: <https://dev.twitter.com/rest/public>. [Último acceso: Agosto 2016].
- [24] «Python Documentation Math» 2016. [En línea]. Available: <https://docs.python.org/2/library/math.html>. [Último acceso: Agosto 2016].
- [25] «NumPy.org» 2016. [En línea]. Available: <http://www.numpy.org/>. [Último acceso: Agosto 2016].
- [26] «Api MongoDB» 2016. [En línea]. Available: <https://api.mongodb.com/python/current/>. [Último acceso: Agosto 2016].
- [27] «Networkx Python» 2016. [En línea]. Available: <https://networkx.github.io/>. [Último acceso: Agosto 2016].
- [28] A. Hagberg y D. Conway, «Hacking social network using the Python programming language» 30ª Conferencia de Sunbelt, 2010.
- [29] F. Ramírez Herrera, «Análisis de Redes Complejas. Identificación y análisis de vulnerabilidades en infraestructuras de hardware y software» Instituto Costarricense sobre Drogas.
- [30] L. Becchetti, E. Koutsoupias y S. Leonardi, «Online Social Networks and Networks Economics» Universidad de Roma, 2010.
- [31] R. Zafarani, M. Ali Abbasi y H. Liu, «Social Media Mining: An Introduction» Cambridge University Press, 2014.

